

IETF 112

Constrained RESTful Environments WG (core)

Chairs:

Marco Tiloca <marco.tiloca@ri.se>

Jaime Jiménez <jaime.jimenez@ericsson.com>

Mailing list: core@ietf.org

Jabber: core@jabber.ietf.org



- We assume people have read the drafts
- Meetings serve to advance difficult issues by making good use of face-to-face communications
- Note Well: Be aware of the IPR principles, according to RFC 8179 and its updates
 - Blue sheets – Automatic
 - Jabber Scribe(s)
 - Note Taker(s)

Note Well

This is a reminder of IETF policies in effect on various topics such as patents or code of conduct. It is only meant to point you in the right direction. Exceptions may apply. The IETF's patent policy and the definition of an IETF "contribution" and "participation" are set forth in BCP 79; please read it carefully.

As a reminder:

- By participating in the IETF, you agree to follow IETF processes and policies.
- If you are aware that any IETF contribution is covered by patents or patent applications that are owned or controlled by you or your sponsor, you must disclose that fact, or not participate in the discussion.
- As a participant in or attendee to any IETF activity you acknowledge that written, audio, video, and photographic records of meetings may be made public.
- Personal information that you provide to IETF will be handled in accordance with the IETF Privacy Statement.
- As a participant or attendee, you agree to work respectfully with other participants; please contact the ombudsteam (<https://www.ietf.org/contact/ombudsteam/>) if you have questions or concerns about this.

Definitive information is in the documents listed below and other IETF BCPs. For advice, please talk to WG chairs or ADs:

- [BCP 9](#) (Internet Standards Process)
- [BCP 25](#) (Working Group processes)
- [BCP 25](#) (Anti-Harassment Procedures)
- [BCP 54](#) (Code of Conduct)
- [BCP 78](#) (Copyright)
- [BCP 79](#) (Patents, Participation)
- <https://www.ietf.org/privacy-policy/> (Privacy Policy)

<https://www.ietf.org/about/note-well/>

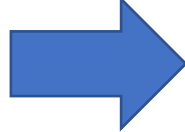


Practicalities

- Use the queue request on Meetecho
- Use of queuing at core@jabber.ietf.org
 - mic: to ask for relaying a question
- This meeting is recorded
- Bluesheets are automatically filled

All times are in UTC

Agenda (120 min)

- 
- 16:00–16:10 Intro, Agenda, Status
 - 16:10–16:25 HREF
 - 16:25–16:40 CoRAL
 - 16:40–16:55 Groupcomm-bis
 - 16:55–17:05 Group OSCORE
 - 17:05–17:20 Key update for OSCORE
 - 17:20–17:30 Cacheable OSCORE
 - 17:30–17:40 OSCORE-capable proxies
 - 17:40–17:50 Performance measurement option
 - 17:50–18:00 Flextime

Status

RFC 9100

- “Sensor Measurement Lists (SenML) Features and Versions”
- *Was draft-ietf-core-senml-versions*



RFC Queue

- draft-ietf-core-senml-data-ct-07
 - In RFC Ed Queue : EDIT
- draft-ietf-core-echo-request-tag-14
 - In RFC Ed Queue : RFC-EDITOR
- draft-ietf-core-resource-directory-28
 - In RFC Ed Queue : EDIT
- draft-ietf-core-new-block-14
 - In RFC Ed Queue : EDIT

RECENTLY
APPROVED

RECENTLY
APPROVED

IESG Processing

- draft-ietf-core-yang-cbor-17
 - IESG Evaluation::Revised I-D Needed
- draft-ietf-core-sid-17
 - IESG Evaluation::Revised I-D Needed

Post-WGLC Processing

- **draft-ietf-core-comi-11**
 - Waiting for Shepherd Write-Up; fixes required from authors
- **draft-ietf-core-yang-library-03**
 - Waiting for Shepherd Write-Up; need to synch with -comi
- **draft-ietf-core-osc core-groupcomm-13**
 - Version -12 addressed the 1st WGLC comments
 - Approaching the 2nd WGLC (see later presentation)

Next to come

- Update WG milestones in the Datatracker
 - Some have been indeed met
 - Add new ones to reflect ongoing activities
- Interim meetings, 15:00-16:30 UTC
 - 2021, December 8th
 - 2022, January 19th
 - 2022, February 2nd
 - 2022, February 16th
 - 2022, March 2nd

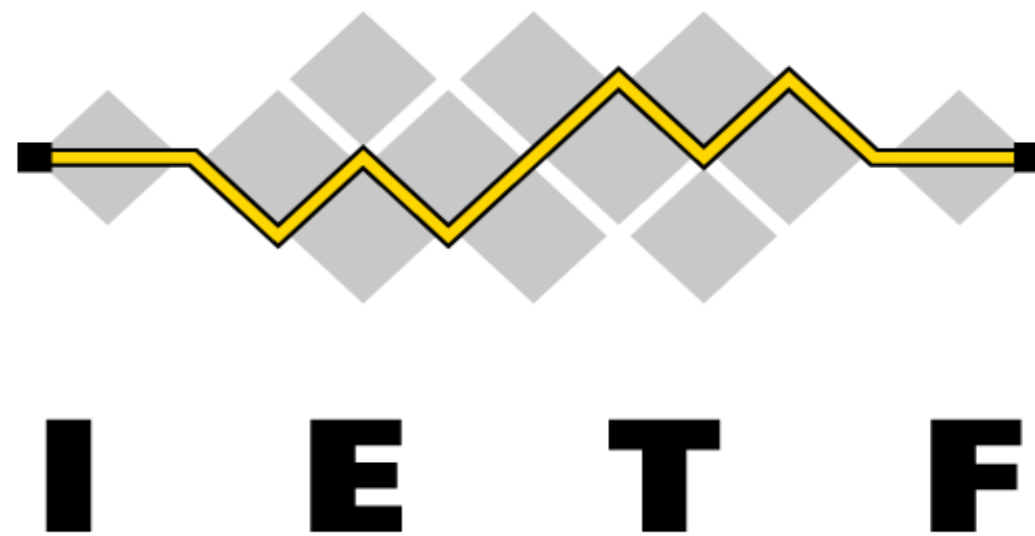
All times are in UTC

Agenda (120 min)

- 16:00–16:10 Intro, Agenda, Status
- • 16:10–16:25 HREF
- 16:25–16:40 CoRAL
- 16:40–16:55 Groupcomm-bis
- 16:55–17:05 Group OSCORE
- 17:05–17:20 Key update for OSCORE
- 17:20–17:30 Cacheable OSCORE
- 17:30–17:40 OSCORE-capable proxies
- 17:40–17:50 Performance measurement option
- 17:50–18:00 Flextime

Flextime

Thank you!
Comments/questions?



CoRE: CRI

November 8th, (Monday), 16:00–18:00 UTC
(17:00–19:00 CEST, 08:00–10:00 PDT)

The Web (~ 1990)

component

big web

hyperreferences

URI

transfer protocol

HTTP

representation format

HTML

The Thing Web (~ 2010): CoRE

component

thing web

hyperreferences

URI

transfer protocol

CoAP

representation format

(CBOR-based formats)

Time for a cleanup? (~ 2020)

component

thing web

hyperreferences

URI → CRI

transfer protocol

CoAP

representation format

(CBOR-based formats)

URI

[...] many implementations [...] support only an ad-hoc, informally-specified, bug-ridden, non-interoperable subset of half of RFC 3986.

— Klaus Hartke

RFC 3986, RFC 7252

RFC 3986: syntax of URIs, (implicit) data model
RFC 7252 maps that data model to CoAP options

Component	Structure	Separators
scheme	http:	
authority	//tzi.de	. — in hostnames
path	(path segments)	/
query	(query elements)	& — CoAP
fragment	#page-5	

URI references: RFC 3986 "resolution procedure"

URIs **relative** to a base (document URI): occur in **documents**, not in transfer protocols

Example	Resolution from https://tzi.de/pa/t/h	What happens to Path Segments?
foo	https://tzi.de/pa/t/foo	discard last 1
/foo	https://tzi.de/foo	discard all
../foo	https://tzi.de/pa/foo	discard last 2
?bar	https://tzi.de/pa/t/h?bar	discard 0
//tzi.org/foo	e.g., https://tzi.org/foo	(new authority), discard all

CRI

Concise Resource Identifier:

Concise equivalent of URIs and URI references (RFC 3986)

New **representation format** for **URI data model**

draft-ietf-core-href defines **CRIs** and CRI references

Evolution

- started by Klaus Hartke
- further developed with Jim Schaad, now CBOR-based

Abstract content:

```
[ ((scheme, authority) // discard), path, query, fragment ]
```

(path and query are arrays;
authority has address/name + optional port)

-06 ("authority anomaly")

URI	CRI
urn:x	["urn", null, ["x"]]
urn:/x/y	["urn", null, ["x", "y"]]
urn:/x	["urn", null, ["x"]]
Solution:	special-case non-rooted opaque
urn:x	["urn", true, ["x"]]

-06: parsed hostname

Component	representation	parsed out
scheme	"http" or -2	:
authority	["tzi", "de", 4711]	. and :
path	["pa", "th"]	/
query	["qu", "e=ry"]	? and &
fragment	"fragment"	#

http://tzi.de:4711/pa/th?qu&e=ry#fragment

– 08: Status

- consistent design
- feature-complete
- initial test vectors in PR (need updates)
- some implementations need updates
- (1) More implementer reviews, (2) WGLC?

CRI in CURIE

CURIEs (JSON-LD, ...): Put a prefix into one place and concatenate the rest of the URI in another place.

CRI: Can do this with CBOR-packed in certain cases:

`[-4, ["www", "w3", "org"]] + [{"ns", "td", "title"}]` ✓

`[-4, ["www", "w3", "org"], ["ns", "td"]] + ??? [...["title"]]` 🤔

Percent-Encoded Text

Constraints in Section 2 generally bearable, except maybe:

CoAP/CRI does not support percent-encoding except for its own delimiters: /path%2Fslash/foo, not urn:aa:bb%3Ac

New (optional?) proposal in CRI -08:

encode application percent-encoded text via arrays:

[-5, true, ["aa:bb:c"]] — unencoded →

[-5, true, ["aa:bb", ":", "c"]] — odd elements %-encoded 🙌

The Constrained RESTful Application Language (CoRAL)

`draft-ietf-core-coral-04`

Christian Amsüss, Thomas Fossati

2021-11-08, IETF 112

A data model and language for talking about resources and interactions with them, suitable for constrained devices

Potential users

- Problem details
- PubSub topic descriptions
- OSCORE Group Manager administration
- SDF
- ...and anything that uses link-format (e. g. discovery)

In contrast to ..., CoRAL is:

RDF Compact (numeric pre-arranged or ad-hoc shorthands for predicates), parsable using CBOR, no URI processor required.

RFC6690 Less string parsing, more depth to information, clear semantics.

CBOR Semantic keys over ad-hoc ones. High-level terminology for derived specs. Interaction model provided. Reuse of terminology.

But CoRAL can be used with them:

RDF can be round-tripped to unstructured CoRAL almost completely.

RFC6690 can be round-tripped to CoRAL, provided the common CoRE attributes are used to describe the targets.

CBOR's literals can be used in CoRAL.

Work areas

90%¹ Information model

Sea of triples, with optional structuring into a tree-like shape.

70% Interaction model

User agent searches document, decides which link (or form) to follow.

70% Dictionary setup

Packed CBOR now does the heavy lifting. Variations being discussed:
Per-document-format; ad-hoc (Basic Packed); importing named dictionary.
Document format can guide tree-like shape.

30% Binary serialization

To be revisited with corpus of use case examples.

? Text serialization

Currently using binary serialization with **EDN** (or **Turtle** when details like the optional structuring or which parts are compressed do not matter).

10% Queries, patches, provenance

¹Don't read too much into these numbers, they are for comparison between the items at best

Next steps

- Coordinate with users to validate current state against their models.
- Get corpus of examples for further dictionary and serialization work.
- What needs to be in for an initial usable version?

Thanks

Comments?

Questions?

Design team for CoRAL and CRIs meets roughly every 2 weeks.

Group Communication for the Constrained Application Protocol (CoAP)

draft-ietf-core-groupcomm-bis-05

Esko Dijk, IoTconsultancy.nl
Chonggang Wang, InterDigital
Marco Tiloca, RISE

IETF 112 - CoRE WG, November 8, 2021

Goal

- › Normative successor of experimental RFC 7390
 - Obsoletes RFC 7390, Updates RFC 7252 / 7641
- › New standard reference for implementations now based on RFC 7390
- › Scope
 - CoAP group communication, including latest features: Observe/Blockwise/Security ...
 - Unsecured & group-OSCORE-secured
 - Definition of group types & Secure group configuration

Overview of -05 updates

- › Removal of Multi-ETag Option → existing ETag Option seems ok & simpler
- › Section added (1.3) to clarify in detail what is Updated / Obsoleted in former RFCs – was [#20](#)
- › Detailed (2.2.1) how Application Group can be named in Group URI or in CoAP request [#28](#) – **pending WG review & approval!**

Application Group Naming #28

- › Application Group (AG) Name: any string, (integer) number, or URI
- › MAY be encoded in Group URI

coap://[COAP_GROUP_ADDR]/g/**Group1**/light → In path: **RECOMMENDED**

coap://[COAP_GROUP_ADDR]/light?**Group1** → In query

coap://[COAP_GROUP_ADDR]/light?g=**Group1**

coap://[COAP_GROUP_ADDR]/light?foo=bar&gp=**Group1**

coap://**Group1**.example.com/light → In host (part of CoAP Group)

coap://[COAP_GROUP_ADDR]:**43210**/light → In port (part of CoAP Group)

coap://[COAP_GROUP_ADDR]/light → Not in group URI: can be in a CoAP Option, or implicit

Overview of -05 updates

- › Detailed (2.2.3) what kinds of group discovery are possible using CoAP Discovery #29 – pending WG review & approval!
- › Stronger advice (4 / 6.1: “NOT RECOMMENDED”) on unsecured group communications – was #22 – ok for WG?
- › Editorial improvements & fixes
(e.g. group relations fix, explain forward/backward security, 6.3 amplification risk, ...)

Non-RD Group Discovery #29

- › Discover all AGs part of a CoAP Group ‘CG1’ and CG1 members

`coap://CG1/.well-known/core?href=/g/*`

Example

- › Discover all realm-local members of AG ‘Group1’

`coap://[ff03::fd]/.well-known/core?href=/grp/Group1`

Example

- › Discover all AGs of a particular type ‘Type1’ in the mesh and members

`coap://[ff03::fd]/.well-known/core?rt=Type1`

Example

(One more example in the I-D. All this: application-specific, i.e. examples only.)

Next steps

- › More reviews of the updated parts (diff)!
 - If all's well we can close [#28](#), [#29](#)
- › More reviews of entire document? As part of WGLC?
 - Promised @IETF 108: Christian, Francesca
- › All review comments (John, Christian) now addressed
 - Current version -05 may be ready for WGLC.

Thank you!

Comments/questions?

<https://github.com/core-wg/groupcomm-bis/>

Motivation (backup slide)

- › RFC 7390 was published in 2014
 - CoAP functionalities available by then were covered
 - No group security solution was available to indicate
 - It is an Experimental document (started as Informational)
- › What has changed?
 - More CoAP functionalities have been developed (Block-Wise, Observe)
 - RESTful interface for membership configuration is not really used
 - Group OSCORE provides group end-to-end security for CoAP
- › Practical considerations
 - Group OSCORE clearly builds on RFC 7390 normatively
 - However, it can refer RFC 7390 only informationally

Group OSCORE - Secure Group Communication for CoAP

draft-ietf-core-oscore-groupcomm-13

Marco Tiloca, RISE

Göran Selander, Ericsson

Francesca Palombini, Ericsson

John Mattsson, Ericsson

Jiye Park, Universität Duisburg-Essen

IETF 112, CoRE WG, November 8th, 2021

Update since IETF 111

- › Version -13 submitted
- › Terminology on formats of public keys
 - UCCS → CCS (CWT Claims Set)
 - Sufficient to refer to RFC 8392
 - Same as in *draft-ietf-lake-edhoc*
- › Group Mode: fix in the derivation of the “Group Encryption Key”
 - Used for generating a keystream, to separately encrypt the message signature
 - Now the right key size is indicated in the key derivation step

Update since IETF 111

- › Updated Section 10 on MTI compliance requirements
 - Constrained devices might not be able to support multiple signature algorithms
 - Goal: enable as much interoperability as we can reasonably achieve
 - Now following the same rationale of *draft-ietf-lake-edhoc*

If supporting the Group Mode

- Less constrained endpoints SHOULD implement both: the EdDSA signature algorithm with elliptic curve Ed25519; and the ECDSA signature algorithm with elliptic curve P-256.
- Constrained endpoints SHOULD implement: the EdDSA signature algorithm with elliptic curve Ed25519; or the ECDSA signature algorithm with elliptic curve P-256.

If supporting the Pairwise Mode

- Less constrained endpoints SHOULD implement both ECDH curves X25519 and P-256.
- Constrained endpoints SHOULD implement the X25519 or P-256 curve as ECDH curve.

Next steps

- › No open issues or open points we are aware of
 - Recently closed 4 Github issues, 3 of which already addressed in v -12
- › Updated implementation for Eclipse Californium
- › Ready for the 2nd WGLC
- › Started to produce test vectors, for both group mode and pairwise mode
 - Appendices to this draft would be pretty long. Alternative release venue?
 - › Just a CoRE Github repo?
 - › Separate informational draft, as in LAKE? Should it be published as RFC?

Thank you!

Comments/questions?

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

Key Update for OSCORE (KUDOS)

draft-hoeglund-core-oscore-key-limits-02

Rikard Höglund, RISE
Marco Tiloca, RISE

IETF 112, CoRE WG, November 8th, 2021

Recap

- › OSCORE (RFC8613) uses AEAD algorithms to provide security
 - Need to follow limits in key usage and number of failed decryptions, before rekeying
 - Excessive use of the same key can enable breaking security properties of the AEAD algorithm
 - Reference **draft-irtf-cfrg-aead-limits-03**
- › (1) Study of AEAD limits and their impact on OSCORE
 - Defining appropriate limits for OSCORE, for a variety of algorithms
 - Defining counters for key usage; message processing details; steps when limits are reached
 - Taking into account John Mattsson's input at the April CoRE interim [1]
- › (2) Defined a new method for rekeying OSCORE (KUDOS)
 - Loosely inspired by Appendix B.2 of OSCORE
 - Goal: renew the Master Secret and Master Salt; derive new Sender/Recipient keys from those
 - Achieves Perfect Forward Secrecy

[1] <https://datatracker.ietf.org/meeting/110/materials/slides-110-saag-analysis-of-usage-limits-of-aead-algorithms-00.pdf>

Key limits (1/3)

Confidentiality Advantage (CA):
Probability of breaking confidentiality properties

Integrity Advantage (IA):
Probability of breaking integrity properties

- › Recap on AEAD limits
 - Discussed in **draft-irtf-cfrg-aead-limits-03**
 - Limits key use for encryption (q) and invalid decryptions (v)
 - This draft defines fixed values for 'q', 'v', and 'l' and from those calculate CA & IA probabilities
 - › IA & CA probabilities must be acceptably low
- › Now explicit size limit of protected data to be sent in a new OSCORE message
 - The probabilities are influenced by 'l', i.e., maximum message size in cipher blocks
 - Implementations should not exceed 'l', and it has to be easy to avoid doing so
 - New text: *the total size of the COSE plaintext, authentication Tag, and possible cipher padding for a message may not exceed the block size for the selected algorithm multiplied with 'l'*
- › New table (Figure 3) showing values of 'l' not just in cipher blocks but actual bytes

Key limits (2/3)

- › Increased value of 'l' (message size in blocks) for algos except AES_128_CCM_8
 - Increasing 'l' from 2^8 to 2^{10} should maintain secure CA and IA probabilities
 - draft-irtf-cfrg-aead-limits mentions aiming for CA & IA lower than to 2^{-50}
 - › They have added a table in that document with calculated 'q' and 'v' values

$q = 2^{20}$, $v = 2^{20}$, and $l = 2^{10}$

Algorithm name	IA probability	CA probability
AEAD_AES_128_CCM	2^{-64}	2^{-66}
AEAD_AES_128_GCM	2^{-97}	2^{-89}
AEAD_AES_256_GCM	2^{-97}	2^{-89}
AEAD_CHACHA20_POLY1305	2^{-73}	-

- › Intent is to increase 'q', 'v' and/or 'l' further. Should we?
 - Since we are well below 2^{-50} for CA & IA currently

Key limits (3/3)

- Updated table of 'q', 'v' and 'l' for AES_128_CCM_8
 - Added new value for 'v', still leaving CA and IA less than 2^{-50}
 - Is it ideal to aim for CA & IA close to 2^{-50} as defined in the CRFG document?

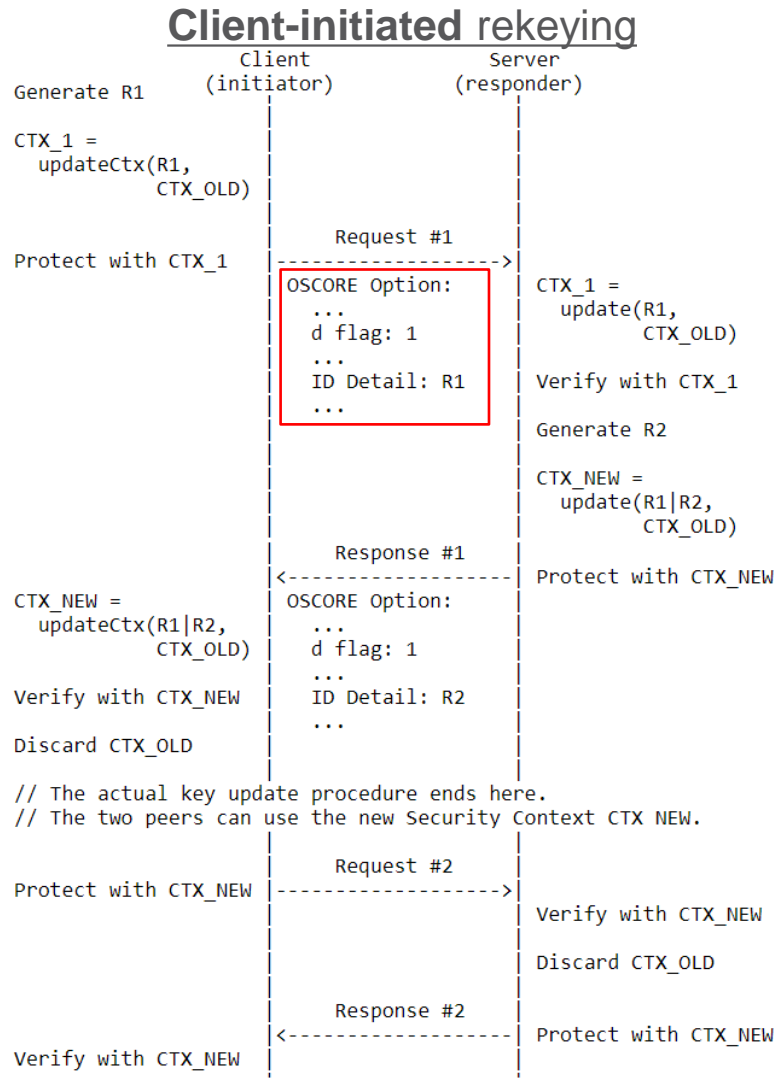
'q', 'v' and 'l'	IA probability	CA probability	'q', 'v' and 'l'	IA probability	CA probability
q=2 ²⁰ , v=2 ²⁰ , l=2 ⁸	2 ⁻⁴⁴	2 ⁻⁷⁰	q=2 ²⁰ , v=2 ²⁰ , l=2 ⁶	2 ⁻⁴⁴	2 ⁻⁷⁴
q=2 ¹⁵ , v=2 ²⁰ , l=2 ⁸	2 ⁻⁴⁴	2 ⁻⁸⁰	q=2 ¹⁵ , v=2 ²⁰ , l=2 ⁶	2 ⁻⁴⁴	2 ⁻⁸⁴
q=2 ¹⁰ , v=2 ²⁰ , l=2 ⁸	2 ⁻⁴⁴	2 ⁻⁹⁰	q=2 ¹⁰ , v=2 ²⁰ , l=2 ⁶	2 ⁻⁴⁴	2 ⁻⁹⁴
q=2 ²⁰ , v=2 ¹⁵ , l=2 ⁸	2 ⁻⁴⁹	2 ⁻⁷⁰	q=2 ²⁰ , v=2 ¹⁵ , l=2 ⁶	2 ⁻⁴⁹	2 ⁻⁷⁴
q=2 ¹⁵ , v=2 ¹⁵ , l=2 ⁸	2 ⁻⁴⁹	2 ⁻⁸⁰	q=2 ¹⁵ , v=2 ¹⁵ , l=2 ⁶	2 ⁻⁴⁹	2 ⁻⁸⁴
q=2 ¹⁰ , v=2 ¹⁵ , l=2 ⁸	2 ⁻⁴⁹	2 ⁻⁹⁰	q=2 ¹⁰ , v=2 ¹⁵ , l=2 ⁶	2 ⁻⁴⁹	2 ⁻⁹⁴
q=2 ²⁰ , v=2 ¹⁴ , l=2 ⁸	2 ⁻⁵⁰	2 ⁻⁷⁰	q=2 ²⁰ , v=2 ¹⁴ , l=2 ⁶	2 ⁻⁵⁰	2 ⁻⁷⁴
q=2 ¹⁵ , v=2 ¹⁴ , l=2 ⁸	2 ⁻⁵⁰	2 ⁻⁸⁰	q=2 ¹⁵ , v=2 ¹⁴ , l=2 ⁶	2 ⁻⁵⁰	2 ⁻⁸⁴
q=2 ¹⁰ , v=2 ¹⁴ , l=2 ⁸	2 ⁻⁵⁰	2 ⁻⁹⁰	q=2 ¹⁰ , v=2 ¹⁴ , l=2 ⁶	2 ⁻⁵⁰	2 ⁻⁹⁴
q=2 ²⁰ , v=2 ¹⁰ , l=2 ⁸	2 ⁻⁵⁴	2 ⁻⁷⁰	q=2 ²⁰ , v=2 ¹⁰ , l=2 ⁶	2 ⁻⁵⁴	2 ⁻⁷⁴
q=2 ¹⁵ , v=2 ¹⁰ , l=2 ⁸	2 ⁻⁵⁴	2 ⁻⁸⁰	q=2 ¹⁵ , v=2 ¹⁰ , l=2 ⁶	2 ⁻⁵⁴	2 ⁻⁸⁴
q=2 ¹⁰ , v=2 ¹⁰ , l=2 ⁸	2 ⁻⁵⁴	2 ⁻⁹⁰	q=2 ¹⁰ , v=2 ¹⁰ , l=2 ⁶	2 ⁻⁵⁴	2 ⁻⁹⁴

Key update (1/4)

- › Defined a new method for rekeying OSCORE
 - Key Update for OSCORE (KUDOS) - Named procedure
 - Client and server exchange two nonces R1 and R2
 - *UpdateCtx()* function for deriving new OSCORE Security Context using the nonces
 - Current Sec Ctx (to renew) ==> Intermediate Sec Ctx
==> **New Sec Ctx**

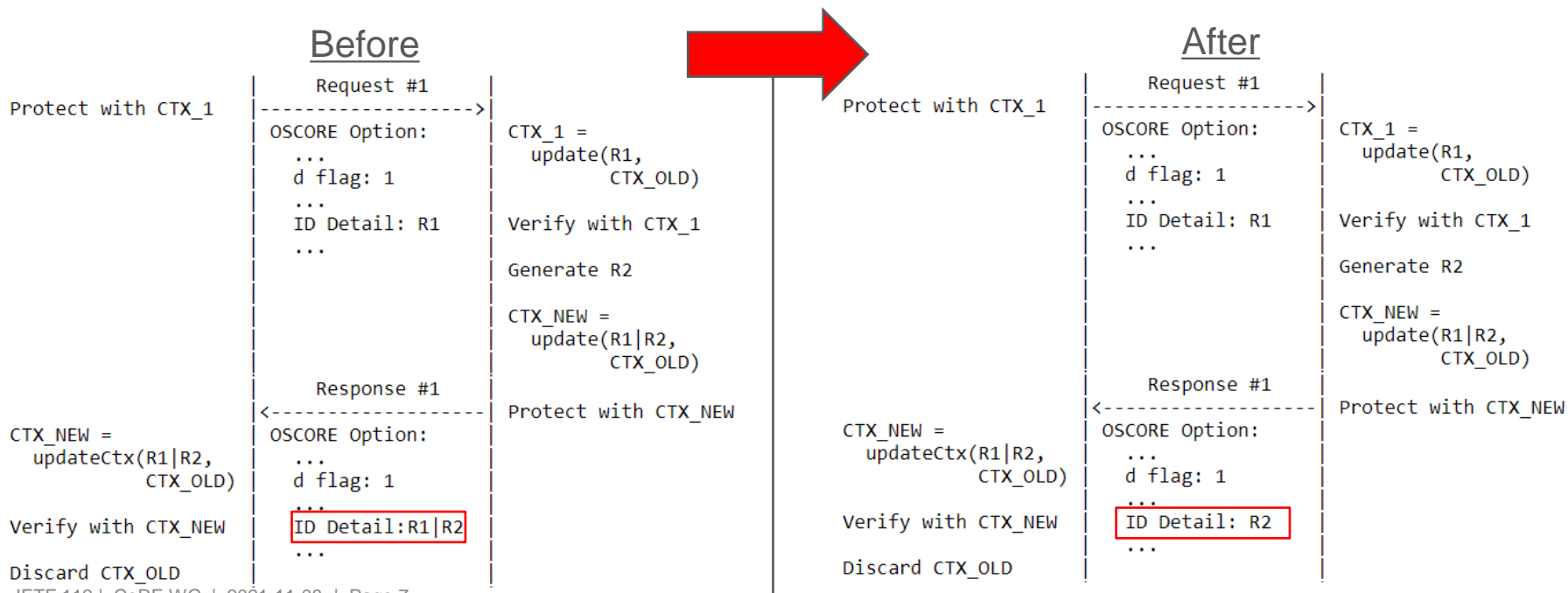
- › Properties

- › Can be initiated by either the client or server
- › Completes in one round-trip (after that, the new Security Context can be used)
- › Only one intermediate Security Context is derived
- › The ID Context does not change
- › Robust and secure against peer rebooting
- › Compatible with prior key establishment using the EDHOC protocol



Key update (2/4)

- › No more R1 in the Response #1 for the **client-initiated** rekeying
 - Just like in OSCORE Appendix B.2
 - Simply not needed: Response #1 correlates to Request #1 through the CoAP Token



Key update (3/4)

- › Recommendations on minimum length of R1 and R2 values
 - R1 and R1 | R2 are used as nonces
 - Motivation is based on similar considerations for Appendix B.2 in RFC8613
 - We now recommend minimum 8 bytes, **is this sufficient?**
 - Further text needs to be added as in Appendix B.2. e.g. mentioning the birthday paradox
- › Currently MUST terminate ongoing observations after rekeying (derived CTX_NEW)
 - Possible to keep them ongoing for a price, i.e. admitting an earlier use of large Partial IVs
 - Possible solution: after a rekeying, the client considers PIV* as the highest req_piv among all the ongoing observations. Then, when the client starts the first new observation, the SSN jumps to PIV*+1, thus every observation request has a PIV greater than PIV*.
 - Drawback: Big jumps in PIV, i.e., faster consumption and larger communication overhead
 - (More complicated solutions like reserving some PIVs in a bit-map is also possible)
 - **Is it worth keeping observations ongoing across a rekeying? Plan is to not keep observations**

Key update (4/4)

- › Added and discussed 6TiSCH as use case
 - 6TiSCH uses OSCORE Appendix B.2 to handle failure events
 - If the 6TiSCH JRC severely fails, it can use Appendix B.2 with the pledges (RECOMMENDED)
 - The new key update procedure is a good replacement, especially for 6TiSCH
 - Among its intrinsic advantages compared to Appendix B.2, **it preserves the ID Context across rekeying**
 - › 6TiSCH uses ID Context as pledge identifier, meaning that:
 - › → A key update would not change pledge identifier, which remains unchanged in the long run
 - › → The JRC does not need anymore to do a remapping between new ID Context and pledge identifier
 - › → **ID Contexts and pledge identifiers can be used as intended at setup/deploy time**
- › The update to RFC8613 includes also “deprecating and replacing” its Appendix B.2
 - **Ok with this?**

More general updates

› Improved Table of Content structure

- Key Limits
- Current rekeying methods
- New rekeying methods
 - › Building blocks
 - › Client-initiated procedure
 - › Server initiated procedure
 - › Policies
 - › Discussion

› Editorial improvements

- Terminology harmonization
- Alignment to most recent EDHOC interfaces
- Use of RFC8126 terminology in IANA considerations
- Updated title to *Key Update for OSCORE (KUDOS)* - Feedback on title?

Next steps

- › Address open points, including:
 - Material to save to disk to support rebooting
 - Reuse applicable considerations from OSCORE Appendix B.2
 - Update security considerations
 - Further refinement of key limits
- › The document foundation and the key update protocol are stable
- › Plan to implement
- › WG adoption?

Thank you!

Comments/questions?

<https://gitlab.com/rikard-sics/draft-hoeglund-oscore-rekeying-limits/>

OSCORE Option update

- › OSCORE Option: defined the use of **flag bit 1** to signal presence of **flag bits 8-15**
- › **Defined flag bit 15 -- 'd' -- to indicate:**
 - This is a OSCORE key update message
 - **"id detail"** is specified (**length + value**); used to transport a nonce for the key update

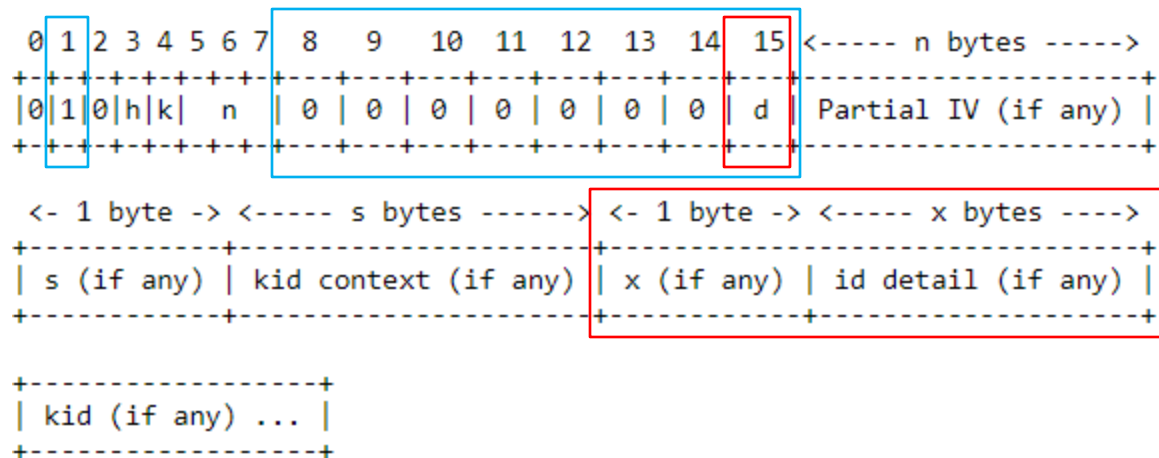


Figure 3: The OSCORE option value, including 'id detail'

Cacheable OSCORE

Or “What to do when numeric request-response binding fails us”.
`draft-amsuess-core-cacheable-oscore-03`

Christian Amsüss, Marco Tiloca

2021-11-08, IETF 112

Development since IETF110: It's really two topics

- I How is request-response binding provided –
when the server does not get source authentication?
- II Once we know that, what do we need for cacheability?

Split introduced late in -03 – not as big as feared, but ...directions?

Request-response binding in OSCORE

What would need to go wrong for response mismatch¹ to happen?

Client intends (and sends) $R1$.

Server processes (and answers to) $R2$.

OSCORE ensures sender and seqno match between $R1$ and $R2$.

Only client and server can produce such messages, and can thus trust them to be identical.

¹See draft-mattsson-core-coap-attacks-01: CoAP Attacks

Request-response binding in Group OSCORE

What would need to go wrong for response mismatch to happen?

Client C intends (and sends) $R1$.

Server S processes (and answers to) $R2$.

OSCORE ensures sender and seqno match² between $R1$ and $R2$.

Only C and S can produce such messages *because of source authentication in all messages*.

²...and KID context, but that doesn't matter much here

Who can use a response?

In group/group mode, every member can read responses.

A third party T can only trust a captured³ response when the original client *and* the server: Client C could have sent distinct $R1$ to be seen by T , and $R2$ to be seen by S .

³Or cached, we'll come to that

How can a response be made usable without trusting *C*?

- Full request is part of response
e. g. a Class E or Class I Response-For⁴
- Hash of request is part of response (Class I or E)
- Either is part of the AAD without being part of the message at all
e. g. by a “hidden Class I option” (currently in cacheable), or by extension of external_aad

...replacing / augmenting the (otherwise very practical) request-response binding mechanism.

⁴draft-bormann-core-responses-00: Non-traditional response forms

...and thus, Cacheable OSCORE is split

- I Request-Response binding can be thusly managed – with some caveats described for Cacheable OSCORE (no freshness)
- II Deterministic requests become a simple means to create common cache keys, and only deal with avoiding nonce reuse and limited request privacy

Questions

- Where else is part I useful?
- Is this simpler to follow when presented in split form inside a single document?

Answers? Other questions? Comments?

OSCORE-capable Proxies

draft-tiloca-core-oscore-capable-proxies-01

Marco Tilocca, RISE
Rikard Höglund, RISE

IETF 112, CoRE WG, November 8th, 2021

Recap

- › A CoAP proxy (P) can be used between client (C) and server (S)
 - A security association might be required between C and P --- examples in next slide
- › 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 $C \leftrightarrow S$ and over $C \leftrightarrow P$
- › This started as an Appendix of *draft-tiloca-core-groupcomm-proxy*
 - Agreed at IETF 110 [1] and at the June 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

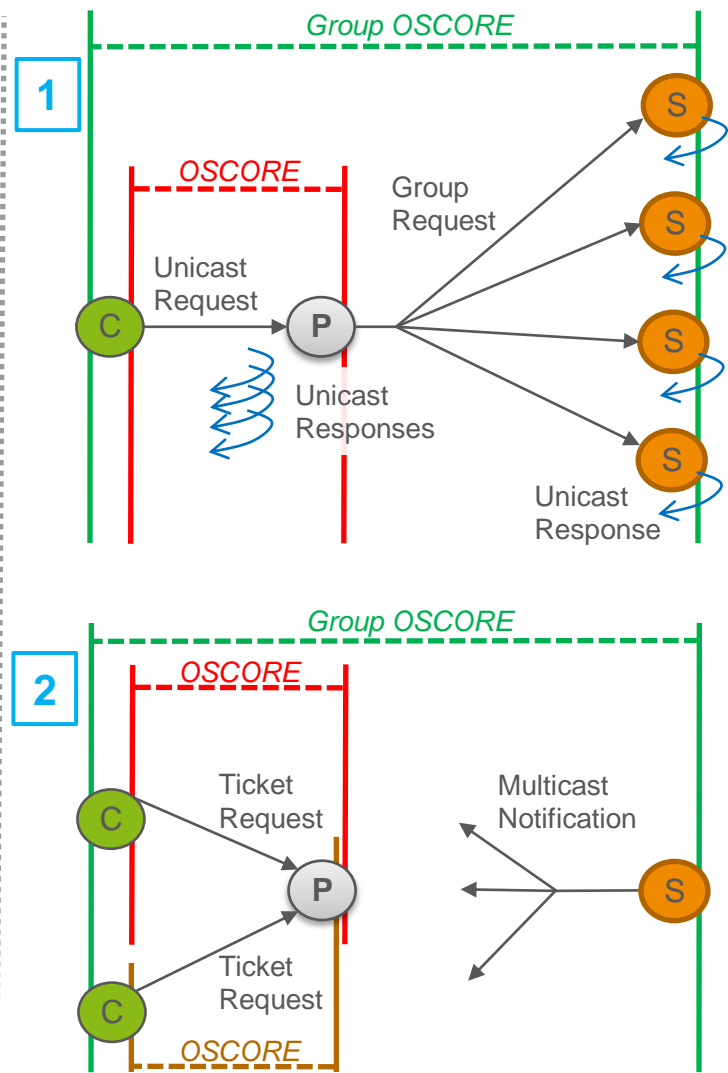
- *draft-tiloca-core-groupcomm-proxy*
- CoAP group communication through a proxy
- P must identify C through a security association

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 $C \leftrightarrow P$

3. LwM2M Client and External Application Server

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



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 $C \leftrightarrow S$, 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 OSCORE layers to the same message
- › Focus on OSCORE, but the same applies “as is” to Group OSCORE

Updates since v -00

- › Version -00 and planned updates presented at the September interim meeting [3]
- › Latest version -01 addresses comments from Göran and Christian – Thanks!
 - Suggestions for more uses case to mention
 - Lift the limit of 2 OSCORE layers applied to the same message
 - Main feedback: the original presentation of message processing was too complicated
- › Added more use cases, now in a new Section 2.4
 - Cross-proxy, as third party service to indicate transports available at the server [4][5]
 - Proxy as an entry point in a firewalled network, accessible only by authenticated clients
 - Privacy-oriented scenarios, with chain of proxies and >2 OSCORE layers per message

[3] <https://datatracker.ietf.org/doc/minutes-interim-2021-core-10-202109151600/>

[4] <https://datatracker.ietf.org/doc/draft-amsuess-core-transport-indication/>

[5] <https://mailarchive.ietf.org/arch/msg/core/RZH8pgyksEwtMYVE1MrPkj9opyg/>

Updates since v -00

- › Revised presentation of message processing
 - Now much shorter and simpler
 - High-level general algorithm, fitting a client, proxy or server as a message processor
 - Now clearly said: no need for an explicit signaling method to guide the message processing
- › Unlike RFC 8613, protect also these CoAP options when applying an OSCORE layer
 - An OSCORE Option, when present as the result of the immediately previous OSCORE layer
 - Options intended to the other OSCORE endpoint X, e.g., proxy related options when X is proxy
- › Processing of an outgoing request
 - More options are protected (see above)
 - The origin client uses the Security Context shared with the origin server as first one

Updates since v -00

- › Processing of an incoming request REQ, based on what it includes
 - **Case A** – Proxy-related options: **included**
 - › Forward to the next hop, possibly adding a further OSCORE layer
 - **Case B** – Proxy-related options: **not included**; OSCORE option: **not included**
 - › Deliver to the application, if any
 - **Case C** – Proxy-related options: **not included**; OSCORE option: **included**
 - › Decrypt REQ using the Security Context retrieved through the OSCORE option
 - › Repeat the (A/B/C) condition assessment over the decrypted request

Error handling is also documented in the draft

Updates since v -00

- › Processing of an outgoing response
 - More options are protected (see previous slide)
 - The origin server uses the Security Context shared with the origin client as first one
 - Apply the same OSCORE layers removed from the request
 - › In the reverse order than the one they were removed
 - › Only the successfully removed layers, if it is an error response

- › Processing of an incoming response
 - Remove the same OSCORE layers added to the request
 - › In the reverse order than the one they were added
 - The layers to remove are at most as many as the added ones

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”
- › Main update in v -01
 - Message processing simplified and generalized to >2 OSCORE layers
 - Removed detailed breakdown and heavy notation → document much shorter and simpler
- › Next steps
 - Add examples
 - Discuss caching of responses, building on *draft-amsuess-core-cachable-oscore*
 - Elaborate on applying >2 OSCORE layers to a same message
 - Look into CoAP header compression from RFC 8824. Use as is? Need for adaptations?
- › More comments and input are welcome!

Thank you!

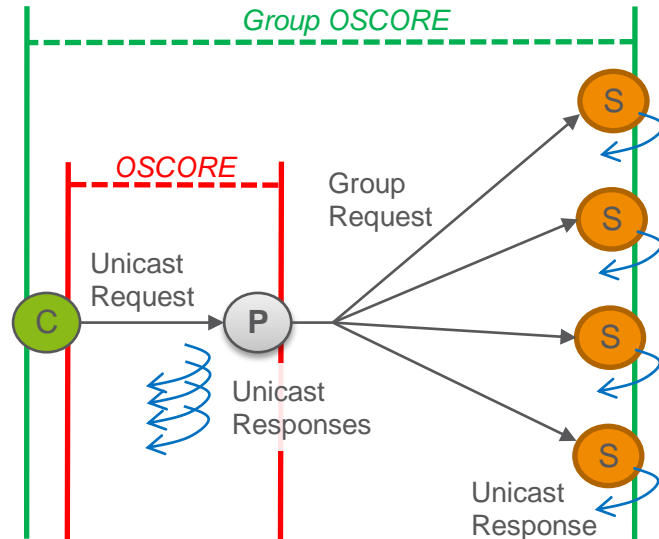
Comments/questions?

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

Some use cases

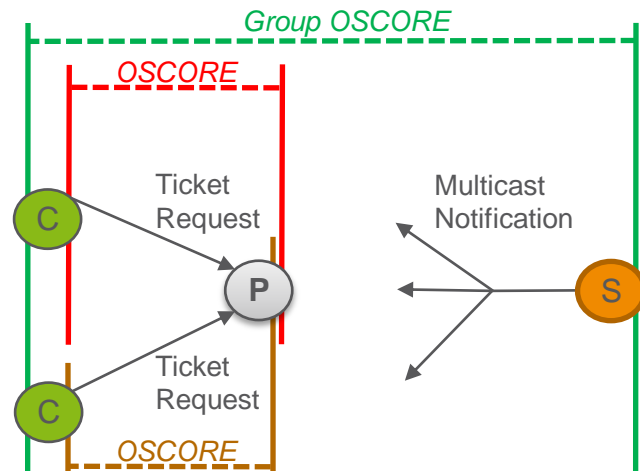
› CoAP Group Communication with Proxies

- *draft-tiloca-core-groupcomm-proxy*
- CoAP group communication through a proxy
- Possible e2e security with Group OSCORE
- P must identify C through a security association before forwarding a request to the group



› CoAP Observe Notifications over Multicast, with Group OSCORE for e2e security

- *draft-ietf-core-observe-multicast-notifications*
- C provides P with a Ticket Request obtained from S
- This allows P to correctly listen to multicast notifications sent by S
- The provisioning of the Ticket Request to P should be protected over $C \leftrightarrow P$



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 outside the LwM2M domain

Constrained Application Protocol (CoAP) Performance Measurement Option

draft-fz-core-coap-pm-00

Online, Nov 2021, IETF 112

Giuseppe Fioccola (Huawei)
Tianran Zhou (Huawei)
Mauro Cociglio (Telecom Italia)
Fabio Bulgarella (Telecom Italia)
Massimo Nilo (Telecom Italia)

Motivation

- In case of CoAP reliable mode, reliability is provided by marking a message as Confirmable (CON)
 - There are Message IDs and ACKs that can be used to identify packets and measure RTT. But it is resource-consuming for constrained nodes since they have to look at IDs and take timestamps.
- In case of CoAP unreliable mode, a message that does not require reliable transmission can be sent as a Non-confirmable message (NON)
 - No easy way to do measurements

It is resource consuming to read IDs / sequence numbers and store timestamps for constrained nodes.

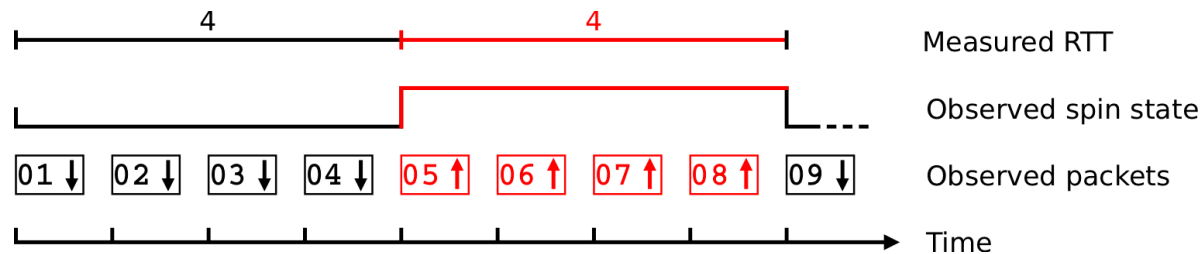
Performance Measurement in constrained environment needs simplified mechanisms!

Spin Bit and sSquare Bit

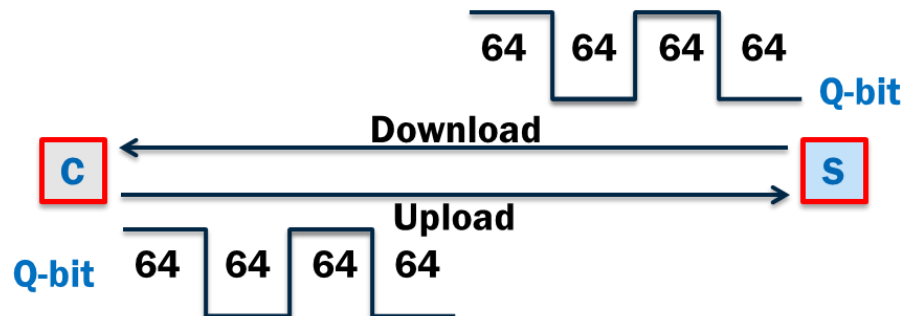
Explicit Flow Measurement (EFM) techniques employ few marking bits, inside the header of each packet, for loss and delay measurement.

These are described in [draft-ietf-ippm-explicit-flow-measurements](#), just adopted in IPPM

- The **Spin bit** idea is to create a square wave signal on the data flow, using a bit, whose length is equal to RTT. It is optional in QUIC (RFC9000)

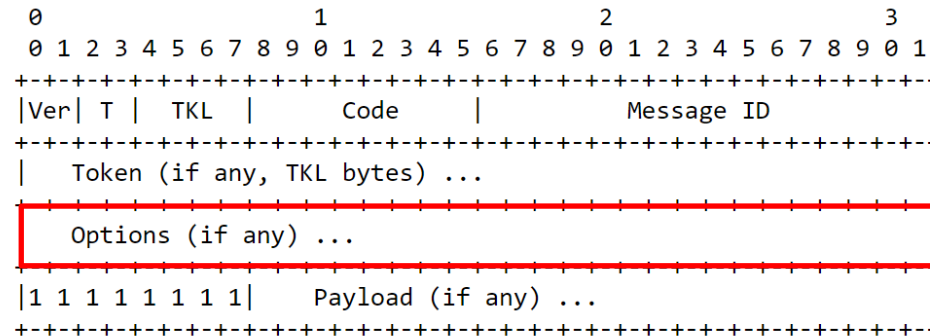


- The **sSquare bit** creates square waves of a known length as defined in the Alternate Marking (RFC8321). This can be used for packet loss measurements.

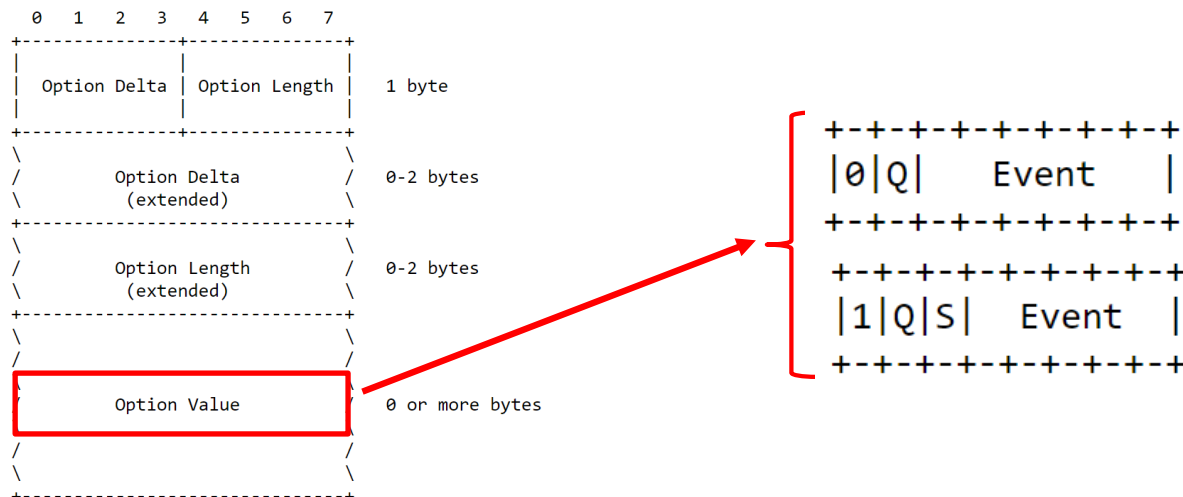


COAP PM Option

- A new option for CoAP carrying Performance Measurement (PM) bits (in particular Spin bit and sSquare Bit) can be defined



- The PM Option Value can be defined with 1 bit or 2 bits. 2 bits are defined as follows:
 - sSquare Bit (Q) for Packet Loss measurement in both Client-Server and Server-Client directions and for RTT measurement. It can be used alone.
 - Spin Bit (S) can also be added for RTT measurement (reinforced by the Q bit)



Key Points and Benefits

- No IDs/sequence numbers for packet loss and flexible timestamp handling to measure RTT. The method is simple to meet the requirement of constrained nodes.
 - Equip the CoAP with Performance Measurement bits to enable RTT and Loss metrics.
- Proposal to improve the Q bit mechanism and find a synergy with S bit in order to simplify the application. Q bit can also be used alone to measure loss and delay.
 - Constrained nodes need simple way to do performance measurements
- Possible advanced usage:

Addition of event signaling bits for on-path observers. The on-path observer can be the Proxy or a Gateway to interconnect disjointed CoAP networks.

 - This information could be used to adjust protocol parameters (e.g. timeout values) based on the real network performance.
 - It could also be possible to decide whether to use reliable or unreliable message transmission based on network conditions

Next Steps

- This draft is based on well-known methodologies applied in RFC9000 (Spin Bit) and RFC8321 (sSquare Bit).
- It aims to meet the limited resources of constrained environment.

Welcome questions, comments

Thank you