Group Communication for the Constrained Application Protocol (CoAP)

draft-ietf-core-groupcomm-bis-04

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

IETF 111, CoRE WG, July 28th, 2021

Goal

- Intended normative successor of experimental RFC 7390 (if approved)
  - As a Standards Track document
  - Obsoletes RFC 7390; Updates RFC 7252 and RFC 7641
- Be standard reference for implementations that are now based on RFC 7390, e.g.:
  - “Eclipse Californium 2.0.x” (Eclipse Foundation)
  - “Implementation of CoAP Server & Client in Go” (OCF)
- What’s in scope?
  - CoAP group communication (e.g., over UDP/IP), including latest developments
  - (Observe/Blockwise/Security ...)
  - Caching and re-validation of responses
  - Unsecured CoAP or Group-OSCORE-secured communication
  - Principles for secure group configuration
  - Use cases (appendix)

Update from v -04

Revised caching model – based on feedback from the June CoRE interim [1]

- Freshness model, for the origin client
  - New members can join the group at any time; a local cache entry for responses to a group request may not cover all the responses sent since the latest cache refresh. This needs rules.
  - The client always sends out a group request, unless the client has fresh responses cached for all group servers. This is possible only when the client has a full, up-to-date knowledge of the group membership.

- Validation model, between origin client and origin servers
  - Simple and based on the ETag Option in group request/response, as normally used.
  - The server SHOULD (but is not required to) embed a compact, server-specific ID as ETag value.
  - The client needs to handle potential cases of 'value conflict' in ETags from different servers.
    - If responses from two servers have the same ETag value, it's not possible to validate only one
  - «Legacy» servers not aware of this ETag feature will just ignore the option (=ok)

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

Update from v -04

Revised caching model – based on feedback from the June CoRE interim [1]

- Caching model at a proxy
  - Creation and maintenance of cache entries
  - Freshness model like for the origin client, with more details about how/when serving from a cache entry
  - Case with end-to-end security based on Cacheable OSCORE
    - https://datatracker.ietf.org/doc/draft-amsuess-core-cachable-oscore/
- Response re-validation between proxy and group Servers
  - Based on the ETag option, like between the origin client and the group Servers
- Response re-validation between client and proxy
  - Based on a new Group-ETag option
- All the above moved to *draft-tiloca-core-groupcomm-proxy* as more appropriate

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

Update from v -04

Processed review and comments from John Mattsson [2] – To be completed

1. Make more general to cover group communication – Done
   - Not necessarily UDP over IP multicast, although it is the default transport
2. Make more general to about security group communication – Done
   - Not necessarily Group OSCORE, although it is the default security solution
3. Expectations from Group OSCORE and Echo Option about amplification / DoS – Done
   - The problem is mitigated by using Echo, but not prevented altogether
4. Make it clearer what is added/replaced in the updated/obsoleted documents – TODO
5. Explicit dedicated considerations on amplification attacks and DoS
   - Added new Section 6.3 “Risk of amplification” – Need for feedback and possible additional input
   - The NoSec mode is NOT RECOMMENDED and strongly discouraged; examples are given when it can still be acceptable, as discussed in the June CoRE interim. In any other case, security MUST be used.

[2] https://mailarchive.ietf.org/arch/msg/core/xy3ImeWkbqziBhqs4NCGwNP6R7U/

Update from v -04

- New Section 5.3 – Valid security cases with forward/reverse proxies
  - With forward/reverse proxy → Group OSCORE for e2e security over client ↔ servers
  - With a totally trusted reverse proxy acting entirely on behalf of the client, admit also:
    - Hop-by-hop security over client ↔ proxy
    - Group OSCORE over proxy ↔ servers
  - Further details on security in different legs are left to *draft-tiloca-core-groupcomm-proxy*

- Clarified interaction between Observe and No-Response Options

- Added informative reference to *draft-ietf-core-new-block*
  - Servers MUST ignore multicast requests that contain the Q-Block2 Option.

- Open point on terminology – Issue #24
  - Change “backward/forward security” to “backward/forward secrecy” ? Opinions/input ?

Next steps

- Finish addressing the comments from John Mattsson [2]
  - Consider the latest points on amplification raised for *draft-mattsson-core-coap-attack*
  - Make it clearer what is added/replaced in the updated/obsoleted documents
- (Finish to) address the few remaining Github issues [3], also covering the points above
- Some specific functionalities left for testing in the CoAP implementation
  - Block2 in a multicast request, followed by Block2 unicast requests to each server
- Next version can be ready for WGLC

[2] https://mailarchive.ietf.org/arch/msg/core/xy3ImeWkbqziBhqs4NCGwNP6R7U/

[3] https://github.com/core-wg/groupcomm-bis/issues

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