Group Communication for the Constrained Application Protocol (CoAP)

draft-ietf-core-groupcomm-bis-01

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

IETF 108 - CoRE WG, July 31st, 2020

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 over UDP/IP, including latest developments (Observe/Blockwise/Security …)
  - Unsecured CoAP or group-OSCORE-secured communication
  - Principles for secure group configuration
  - Use cases (appendix)

Overview of -01 updates

- Mostly addressed Jim’s review at [1] – Thanks!

- Clarifications on group membership for client-only nodes
  - Don’t have to be in an application group or CoAP group
  - Have to be in the used security group

- Response suppression
  - No need to talk of “legitimate” requests (was issue #4)
  - Suppress if nothing to say, unless the application requires to respond anyway

- Token reuse
  - Clearer indications and differences compared to the unicast case

[1] https://mailarchive.ietf.org/arch/msg/core/CkoNseJhJgALEs3iOLMqUZhEarI/

Overview of -01 updates

- When proxies are used
  - Clarifications on stop accepting responses to group requests
  - The client has more (app-)context information to judge when stopping
- Multicast scope to use
  - Configure in advance, i.e. not up to the client to decide
- Clarification on cancelling group observations
- Usage of Group OSCORE
  - Mentioned both the group mode and the pairwise mode (was issue #5)
  - Creation/management of OSCORE groups addressed in other documents
  - Updated security considerations; reference to COSE-bis documents

Open Github issues

- The UDP port may change (issue #1)
  - Multicast request → Src: 59101   Dst: 9999
  - Unicast response → Src: 5683   Dst: 59101
- The outcome of the thread at [2] seems to converge to:
  - Both source address and source port number of the response are irrelevant to the successful processing at the client
- Planned update
  - The source port number of the response can differ from the destination port number of the request. A client MUST be able to handle this.
  - Issues with that?

[2] https://mailarchive.ietf.org/arch/msg/core/d2CJN0g-ksq9uf0hDqBCRqcHz5g/

Open Github issues

- Requirements for response suppression (issue #2)
  - Operate on Response Code Class, instead of Response Code
  - Planned to switch to Response Code Class, as NoResponse does
  - Issues with that?

- Use URI-Host for naming application groups (issue #3)
  - *“If encoded in the CoAP group URI, the information typically gets removed in the CoAP request sent over the wire. Then the receiving server cannot use it.”*
  - *”It can be added to an outgoing CoAP request (with the group URI already resolved to IP address). Then it influences the choice of application group, because each virtual server will have a different set of resources hosted.”*
  - Should the draft explicitly admit it?

More open points

- Client support for admin-local scope
  - It’s not in RFC 7252, but it’s in RFC 7390 for discovery use cases.
  - Keep it?

- Mapping of application groups and security groups, see [3]
  - Many app groups using one sec group is fine.
  - One app group using many sec groups is “delicate”.
  - **Case A**: the sec groups use different algorithms/parameters → A server joins all of them; a client joins any that it supports. This looks ok.
  - **Case B**: the sec groups express different access control properties → This is problematic and a trouble for applications; better rely on resource properties.
  - Proposal: include Case A as relevant example; not recommend Case B

[3] https://mailarchive.ietf.org/arch/msg/core/4JtUVaB-XG_g0i_8v8CEMGyNdO8/

Next steps

- Work on open Github issues

- Address open points
  - Two left from Jim’s review of -00
  - Others also raised today

- Interop of selected functions in CoAP implementations
  - “Observe + multicast” – Locally tested, w/ and w/o Group OSCORE
  - Limited usage of Blockwise (first multicast request with Block2)

Thank you!

Comments/questions?

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