Group Communication for the Constrained Application Protocol (CoAP)

draft-ietf-core-groupcomm-bis-09

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

IETF CoRE WG interim, October 11, 2023

Review comments John

- Selected 4 review comments to discuss in CoRE interim
- The rest have been processed and text is proposed (Github branch)

- 1. Using experimental/obsolete protocol from RFC 7390
- 2. Is there concrete use anywhere of unsecured discovery?
- 3. “Sensitive and mission-critical” applications
- 4. CoAP Group – are address/port optional or mandatory?

Using experimental/obsolete protocol 1/4

Context:

The IETF does not define a mandatory protocol to accomplish CoAP group creation. [RFC7390] defined an experimental protocol for configuration of group membership for unsecured group communication, based on JSON-formatted configuration resources. **However, using such experimental protocol is not a recommended approach.**

Remark:

I don't think something is not recommended just because it is an experimental RFC. What would the alternative be? To use something proprietary that is not even documented publicly.

- Sentence was originally added after WGLC comment that the past tense needs to be clarified – i.e. why "defined" and not "defines".
- Perhaps a reason to not recommend it, is that RFC 7390 is now to be obsoleted by groupcomm-bis.
- On the other hand, there's no harm in using the experimental protocol. It's still available.

Using experimental/obsolete protocol 1/4

Possible solutions:

- Remove the sentence, and change “defined” -> “defines” (present tense)
  - as an exercise left to the reader: if the obsoletion would have any impact on use of the RFC 7390 protocol
- Change sentence to “Despite the obsoletion of RFC 7390, using this experimental protocol is still a viable option.”
- ... any other?

Concrete use of unsecured discovery? 2/4

Context: (6.1)

For example, early discovery of devices and resources is a typical use case where the NoSec mode is relevant

Remark:

This seems like something people say to move the hard problem of secure credential management somewhere out of scope. Is there any *concrete* examples of any secure actual deployments where this message flow makes sense? Otherwise, I think this part should be removed.

- “Early discovery” here refers to today’s secure device bootstrap solutions. There are multiple variants, most don’t use CoAP today for unsecured discovery but something else.
- The unsecured discovery step is a necessary step in all secure bootstrap solutions: it can’t be removed. (The alternative is not using the method at all.)
- A concrete example that uses CoAP discovery (as one of the discovery options) is Constrained BRSKI (cBRSKI). draft-ietf-anima-constrained-voucher

Concrete use of unsecured discovery? 2/4

Possible solutions:

- Make an informative reference to draft-ietf-anima-constrained-voucher and explain that this method uses a CoAP NoSec discovery step during the bootstrap phase – still while preserving security of the overall method. The secured exchange and trust relation follows later.
  - During the operational phase, the CoAP NoSec is not used for application communication

- Refer in general to secure bootstrap methods as the only known reason, currently, to use CoAP NoSec. Explain why this initial discovery step is not secured.
  - Reason comes from logistics: when a device is made in the factory, its future customer is still unknown.
  - Also logistics & scale requires IoT device installation to be a “near zero effort” operation.

For context: secure bootstrap methods specified today are BRSKI, 6tisch CoJP. And being developed are variants cBRSKI, BRSKI-AE, BRSKI-PRM.

Sensitive and mission-critical 3/4

Context: (various sections)

"sensitive and mission-critical applications" – require security in this case

"non-sensitive and non-critical use cases" – allow NoSec in this case

Remark:

I think the current IETF view should be at least "Unless proven to be not sensitive". Zero trust mandates encryption everywhere without exceptions and modern protocols like QUIC delivers that.

› Agreed, though "encryption everywhere" is not possible for secure bootstrap methods (see previous slides for this topic) during the initial discovery phase.

Sensitive and mission-critical 3/4

Possible solution - We have rephrased that sentence as follows. (Github commit)

OLD

> For sensitive and mission-critical applications, CoAP group communication MUST be protected by using Group OSCORE as specified in [I-D.ietf-core-oscore-groupcomm].

NEW

> CoAP group communication MUST be protected by using Group OSCORE as specified in [I-D.ietf-core-oscore-groupcomm], with the possible exception of: applications that are proven to be neither sensitive nor critical; and specific, well-defined steps where security is not viable or is intrinsically unattainable (e.g., early discovery).

Also, in Section 4 "Unsecured Group Communication", we have rephrased one paragraph as follows.

OLD

> The possible, exceptional use of the NoSec mode ought to be limited to non-sensitive and non-critical applications for which it is relevant, such as early discovery of devices and resources (see Section 6.1).

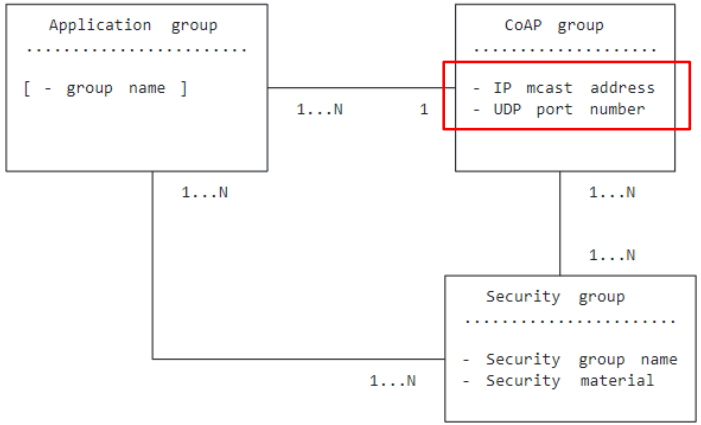
NEW

> The possible, exceptional use of the NoSec mode ought to be limited to: applications that are proven to be neither sensitive nor critical; and specific, well-defined steps where security is not viable or is intrinsically unattainable, e.g., early discovery of devices and resources (see Section 6.1).

CoAP group optional address/port? 4/4

Context/Remark (2.1.1 / 2.2.1.1)

- CoAP group shows **mandatory** mcast address and UDP port in Figure 1.
- Section 2.2.1.1 says UDP port is **optional**, and that IP mcast address is **optional**, in the CoAP group URI.

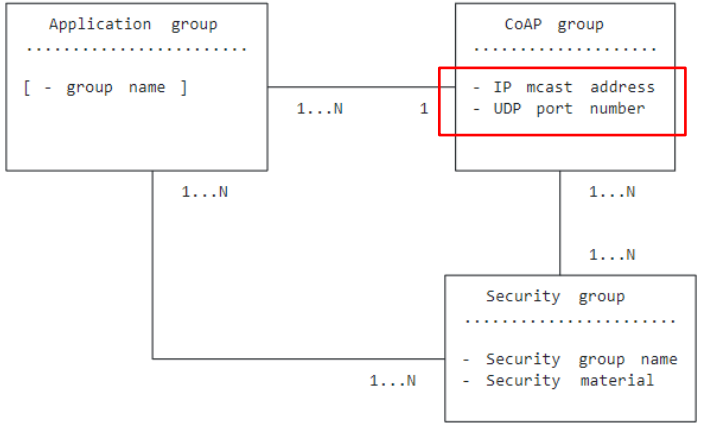


Authors' Response:

Intention was as follows:

- Section 2.1.1 defines the CoAP group and its associated attributes of IP multicast address and UDP port number, which are always there as attributes. (No optionality.)
- Figure 1 represents this definition of the CoAP group, with the 2 mandatory attributes.
- Section 2.2.1.1 defines how CoAP groups are named; and the rules for the name are different. It uses the URI for the name. In the name, the actual IP multicast address may be optionally present. The actual UDP port number may not be present.

CoAP group optional address/port? 4/4



Possible solution

› Update Section 2.2.1.1 text as below:

A CoAP group is always defined by the two properties of IP multicast address and UDP port number (see Section 2.1.1).

However, a CoAP group is for practical purposes identified and named by the authority component in the group URI. This component includes the host subcomponent and an optional UDP port number.

The host subcomponent directly defines the IP multicast address of the CoAP group, in case the host consists of an IP literal.

The host subcomponent indirectly defines the IP multicast address of the CoAP group, in case the host consists of a hostname: resolving the hostname to an IP address in this case produces the IP multicast address.