Discovery of OSCORE Groups with the CoRE Resource Directory

draft-tiloca-core-oscore-discovery-05

Marco Tiloca, RISE
Christian Amsüss
Peter van der Stok

IETF CoRE WG, Virtual Interim, April 8th, 2020

Recap

- A newly deployed device:
  - May not know the OSCORE groups and their Group Manager (GM)
  - May have to wait GMs to be deployed or OSCORE groups to be created

- Use the CoRE Resource Directory (RD):
  - Discover an OSCORE group and retrieve information to join it
  - Practically, discover the links to join the OSCORE group at its GM
  - CoAP Observe supports early discovery and changes in group information

- Use resource lookup, to retrieve:
  - The name of the OSCORE group
  - A link to the resource at the GM for joining the group

Updates from -04

- Addressed review from Jim – Thanks!
  - https://mailarchive.ietf.org/arch/msg/core/FoNCVZtIRzYhv4Imx6e87ZoFk0w/
  - Still one open point (later slide)

- Improved content organization
  - Registration of Group Manager endpoints
  - List and description of target attributes

- Registration of links to ACE Authorization Servers

- Added examples in CoRAL
  - Also asked by Jim

Link to Authorization Server

- When registering an OSCORE group to the RD
  - Possible to register related link to an Authorization Server (AS)
  - The AS is associated to the GM of the OSCORE group
- The joining node is able to retrieve the link to the AS
  - Avoid a first unauthorized access to the GM at joining time

```
Request: GM -> RD

Req: POST coap://rd.example.com/rd?ep=gm1
Content-Format: 40
Payload:
</group-oscore/feedca570000>;ct=41;rt="core.osc.mbr";
                              sec-gp="feedca570000";app-gp="group1";
                              cs_alg="-8";cs_alg_crv="6";
                              cs_key_kty="1";cs_key_crv=6";
                              cs_kenc="1",
<coap://as.example.com/token>;
      rel="authorization-server";
      anchor="coap://[2001:db8::ab]/group-oscore/feedca570000"

Response: RD -> GM

Res: 2.01 Created
Location-Path: /rd/4521
```

From Jim's review

- An application group can use multiple OSCORE groups
  - E.g., one for administration and one for normal communication

- Clarified meaning and usage of 'sec-gp'
  - Stable, invariant and plane name of the OSCORE group
  - This also makes *draft-ace-key-groupcomm-oscore* an informative reference

- Algorithm/key related parameters
  - Improved name and definitions

Examples in CoRAL

- Covered all the main examples
  - Registration, Update with re-registration, Lookup #1, Lookup #2
- Many things become easier
- Easier to specify the link to the AS
  - Easy to add information to such link
  - That link is not to be “navigated”. Ok?
- Currently as Appendix
  - Plan to move to the document body

```
Request: Joining node -> RD

Req: GET coap://rd.example.com/rd-lookup/res
  ?rt=core.osc.mbr&app-gp=group1
Accept: TBD123456 (application/coral+cbor)

Response: RD -> Joining node

Res: 2.05 Content
Content-Format: TBD123456 (application/coral+cbor)

Payload:
#using <http://coreapps.org/reef#>
#using <http://coreapps.org/core.rd#>

#base <coap://[2001:db8::ab]/>
rd-item </group-oscore/feedca570000> {
   rt "core.osc.mbr"
   sec-gp "feedca570000"
   app-gp "group1"
   cs_alg -8
   cs_alg_crv 6
   cs_key_kty 1
   cs_key_crv 6
   cs_kenc 1
   as-uri <coap://as.example.com/token>
}
```

Open point – BACnet example

- Explicit registration of node's membership to application groups
  - Nodes don't need to know their application groups in advance
- Issues
  - This results in multiple endpoint registrations
  - This is not a native functionality of the RD
- This document itself does not need this feature
  - But, it seems common practice in some deployments
- Possible way forward
  - Remove the membership registration from the BACnet example
  - Define the membership registration in a separate short document

Summary and next steps

- Addressed Jim's review; link to AS; examples in CoRAL
- Outcome from previous meetings
  - "Time to start reading it in order to decide for WGA" [1]
  - People volunteered to review: Jim (done); Carsten; Klaus; Bill [1]
  - "Reviewer volunteers are asked to provide reviews now" [2]
- Way forward
  - Close the open point on registration of node's membership (BACnet example)
  - CoRAL: move examples to the document body; translate the BACnet example
  - Process reviews as they come

[1] https://etherpad.ietf.org/p/notes-ietf-104-core?useMonospaceFont=true

[2] https://mailarchive.ietf.org/arch/msg/core/78LHFFyq9c1_t0-kAmuDKcTzc3c/

Thank you!

Comments/questions?

https://gitlab.com/crimson84/draft-tiloca-core-oscore-discovery

Backup

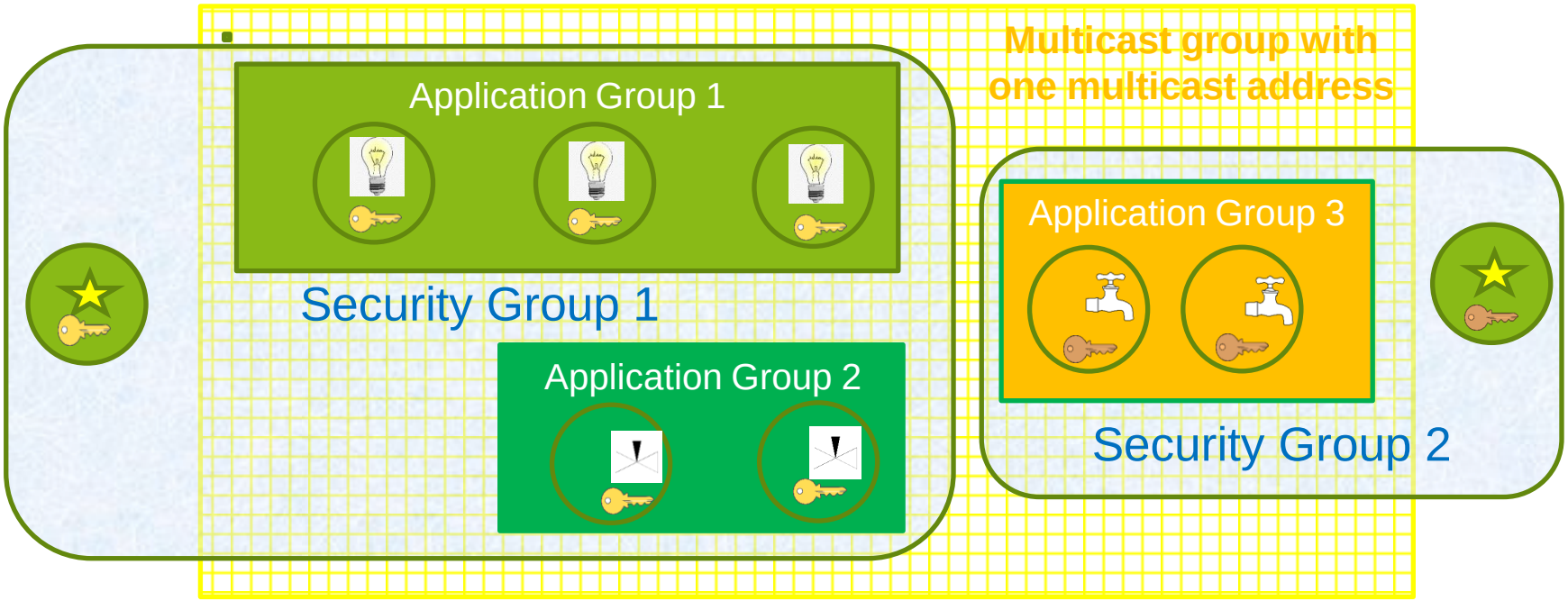
Application/CoAP/Security Groups

- Application group
  - Defined in {RD} and reused as is
  - Set of CoAP endpoints sharing a pool of resources
  - Registered and looked up just as per Appendix A of {RD}

- CoAP/Multicast Group
  - Defined in draft-dijk-core-groupcomm-bis
  - Set of CoAP endpoints listening to the same IP multicast address
  - The IP multicast address is the ‘base’ address of the link to the application group

- OSCORE Security Group
  - Set of CoAP endpoints sharing a common Group OSCORE Security Context
  - A GM registers the group-membership resources for accessing its groups

Application vs. Security Groups



Client of application group

Different key sets

Resources for given function

Alg/key related parameters

- New optional parameters for a registered join resource
  - (*)(**) *cs_alg* : countersignature algorithm, e.g. “EdDSA”
  - (*) *cs_alg_crv* : countersignature curve (if applicable), e.g. “Ed25519”
  - (*) *cs_key_kty* : countersignature key type, e.g. “OKP”
  - (*) *cs_key_crv* : countersignature curve (if applicable), e.g. “Ed25519”
  - (*) *cs_kenc* : encoding of public keys, e.g. “COSE_Key”
  - (**) *alg* : AEAD algorithm
  - (**) *hkdf* : HKDF algorithm

- Benefits for a joining node, when discovering the OSCORE group
  - (*) No need to ask the GM or to have a trial-and-error when joining the group
  - (**) Decide whether to join the group or not, based on supported the algorithms

Registration

- The GM registers itself with the RD
  - MUST include all its join resources, with their link attributes
  - New ‘rt’ value “core.osc.mbr” in the CoRE Parameters registry

```
Request: GM -> RD

Req: POST coap://rd.example.com/rd?ep=gm1
Content-Format: 40
Payload:
</group-oscore/feedca570000>;ct=41;rt="core.osc.mbr";
                             sec-gp="feedca570000";app-gp="group1";
                             cs_alg="-8";cs_alg_crv="6";
                             cs_key_kty="1";cs_key_crv=6";
                             cs_kenc="1",
<coap://as.example.com/token>;
      rel="authorization-server";
      anchor="coap://[2001:db8::ab]/group-oscore/feedca570000"

Response: RD -> GM

Res: 2.01 Created
Location-Path: /rd/4521
```

Discovery (1/2)

› The device performs a resource lookup at the RD
  – Known information: name of the Application Group, i.e. “group1”
  – Need to know: OSCORE Group Identifier; Join resource @ GM; Multicast IP address
  – ‘*app-gp*’ → Name of the Application Group, acting as tie parameter in the RD

```
Request: Joining node -> RD

Req: GET coap://rd.example.com/rd-lookup/res
  ?rt=core.osc.mbr&app-gp=group1

Response: RD -> Joining node


Res: 2.05 Content
Payload:
<coap://[2001:db8::ab]/group-oscore/feedca570000>;rt="core.osc.mbr";
    sec-gp="feedca570000";app-gp="group1";
    cs_alg="-8";cs_alg_crv="6";cs_key_kty="1";
    cs_key_crv=6";cs_kenc="1";anchor="coap://[2001:db8::ab]"
```

Discovery (2/2)

- The device performs an endpoint lookup at the RD
  - Still need to know the Multicast IP address
  - 'ep' // Name of the Application Group, value from '*app-gp*'
  - 'base' // Multicast IP address used in the Application Group

```
Request: Joining node -> RD

Req: GET coap://rd.example.com/rd-lookup/ep
  ?et=core.rd-group&ep=group1

Response: RD -> Joining node

Res: 2.05 Content
Payload:
</rd/501>;ep="group1";et="core.rd-group";
          base="coap://[ff35:30:2001:db8::23]"
```