Admin Interface for the OSCORE Group Manager

draft-ietf-ace-oscore-gm-admin-07

Marco Tiloca, RISE
Rikard Höglund, RISE
Peter van der Stok
Francesca Palombini, Ericsson

ACE WG Interim Meeting, January 23rd, 2023

Recap

- **RESTful admin interface at the OSCORE Group Manager**
  - Create, (re-)configure and delete OSCORE groups
  - Support for both: i) Link Format and CBOR ; ii) CoRAL

- **Two new types of resources at the Group Manager**
  - A <u>single</u> *group-collection* resource, at /manage
  - One *group-configuration* resource per group, at /manage/GROUPNAME

- **Using ACE for authentication and authorization**
  - The Administrator is the Client
  - The Group Manager is the Resource Server
  - For secure communication, use transport profiles of ACE

Overview

Configuration Properties

- Security algorithms and parameters

Status Properties

- Any other information (e.g., group name)

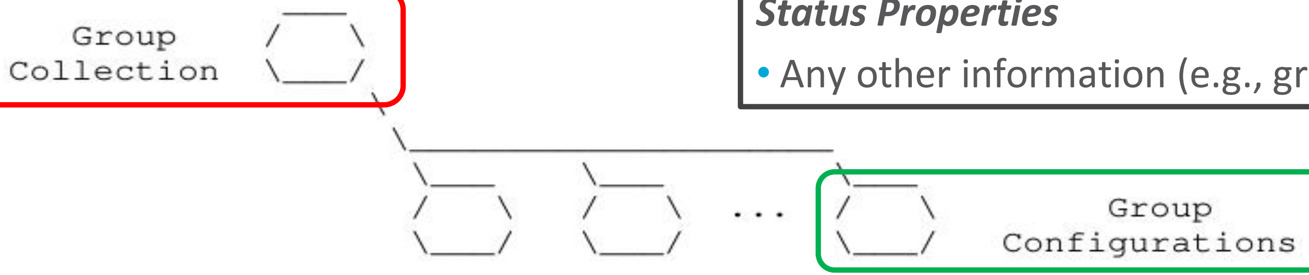


Figure 1: Resources of a Group Manager

***Group-collection* resource**

- Retrieve the list of OSCORE groups
  - All groups (GET)
  - Group selected by filters (FETCH)
- Create a new OSCORE group (POST)
  - A group-configuration resource is created
  - A group-membership for joining nodes is also created, see *draft-ietf-ace-key-groupcomm-oscore*

***Group-configuration* resource**

- Retrieve the group configuration (GET)
- Retrieve part of the group configuration (FETCH)
- Overwrite the group configuration (PUT)
- Update the group configuration (PATCH/iPATCH)
- Delete the group (DELETE)

Updates since IETF 113

- **Last presented: version -05 at IETF 113** (Vienna, March 2022)

- **Submitted version -06 for IETF 114** (Philadelphia, July 2022)

- **Submitted version -07 for IETF 115** (London, November 2022)

These updates address all open points and feedback from IETF 113

v -05 → v -06 (1/2)

› **AIF specific data model to use for scope**

– Use and extend AIF-OSCORE-GROUPCOMM introduced in *draft-ietf-ace-key-groupcomm-oscore*

– scope = << [+ scope_entry] >>

– scope_entry = [Toid, Tperm]

› **An "admin scope entry" as in this document has:**

– Toid : tstr / true / #6nnn(any)

; i.e., group name / wildcard / group name pattern

– Tperm : the rightmost bit is set to 1;

bits set to 1 enable an admin permission

› **The same scope can include both "admin scope entries" and "user scope entries"**

– The latter ones are defined in *-key-groupcomm-oscore*

– Entry types distinguishable by the rightmost bit in Tperm

Tperm

bit	7	6	5	4	3	2	1	0
								1

Permission granted if bit set to 1

0: Retrieve list of groups (bit always set to 1)

1: Create a group and its configuration

2: Read group configuration

3: Change group configuration

4: Delete a group and its configuration

5, 6, 7: Unassigned

v -05 → v -06 (2/2)

- **Revised interactions between Admin (C), AS and Group Manager (RS)**
  - Based on the revised AIF specific data model

- **Categorized operations at the Group Manager (GM), as required or optional to support**
  - In the same spirit of what was done in *draft-ietf-ace-key-groupcomm(-oscore)*

- **Added the group status parameter ‘gid_reuse’ (default = false)**
  - When creating an OSCORE group, the Administrator can request the GM to recycle Group IDs
  - If so, this actually happens if the Group Manager also supports the recycling of Group IDs

- **Detailed processing of group name patterns at the AS**
  - Moved to an Appendix, as just a possible example to consider for implementations
  - The document body is kept to the minimum and at high-level about this point (like in RFC 9200)

v -06 → v -07

- **Updated optional signaling of the “type” of scope claim**
  - Aligned with the recent change made in *draft-ietf-ace-key-groupcomm(-oscore)*
  - The scope claim (CBOR bstr) can be tagged to signal the use of AIF-OSCORE-GROUPCOMM
  - Thanks to RFC 9277, the CBOR tag is the one associated with the CoAP Content-Format defined in *draft-ietf-ace-key-groupcomm-oscore* for the Media-Type “application/aif+cbor”

- **New error code "No available group names“, and guidelines on how to follow-up**
  - Possible to use in response to the Administrator upon a group creation request

- **Categorized parameters as must/should/may be supported**
  - In the same spirit of what was done in *draft-ietf-ace-key-groupcomm(-oscore)*

- **Editorial fixes/improvements**
  - Fixes in the examples; alignment with parameter renaming in *draft-ietf-ace-key-groupcomm*
  - Split between IANA registration of parameters and their CBOR abbreviations

Next steps

1. **More explanations and guidelines on multiple Administrators for the same OSCORE group**
   – E.g., leverage notifications to Admin1 about actions performed by Admin2

2. **Extended security considerations, mostly about the OSCORE Group Manager**

3. **Fixes and clarifications for the case where CoRAL is used**

4. **Use CBOR diagnostic notation and Packed CBOR in the CoRAL examples**

5. **Avoid delay due to progress of *draft-ietf-core-coral*. Split this ACE document into two?**
   – **Doc 1**: current document minus the CoRAL-related content (use, special features, examples , …)
   – **Doc 2**: new WG document, with revised CoRAL-related content taken out of this ACE document

› **Goal for IETF 116**

– Points 1 and 2 completely addressed

– Points 3 and 4 addressed as much as possible

– Reach consensus on how to proceed with point 5

Thank you!

Comments/questions?

https://github.com/ace-wg/ace-oscore-gm-admin

Backup

Group Configuration Parameters

- **Configuration properties**
  - hkdf
  - cred_fmt
  - group_mode
  - sign_enc_alg
  - sign_alg
  - sign_params
  - pairwise_mode
  - alg
  - ecdh_alg
  - ecdh_params
  - det_req
  - det_hash_alg

- **Status properties**
  - rt = “core.osc.gconf”
  - active
  - group_name // Plain immutable identifier
  - group_title // Descriptive string
  - ace_groupcomm_profile
  - max_stale_sets
  - exp
  - gid_reuse
  - **app_groups** // Names of application groups
  - joining_uri
  - group_policies
  - as_uri // Link to the AS

- When using PATCH, easy “replacement” update for most parameters
  - Specify the pair (“label”, new_value), like when creating the group
- ‘app_groups’ is a list of names and requires special handling

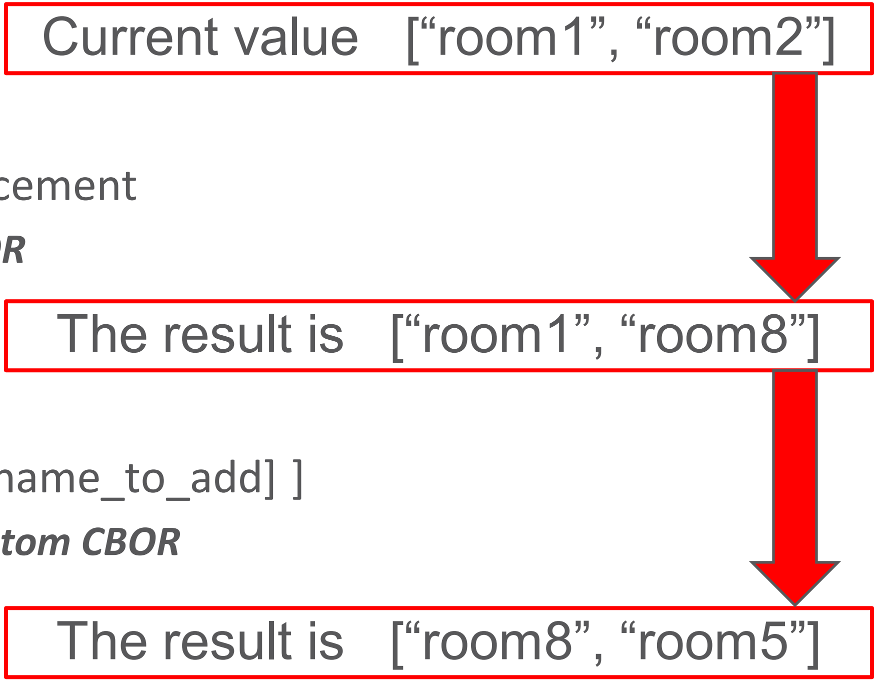
Configuration update with PATCH

- **Two ways to update ‘app_groups’**
  - List of associated applications groups

- **Overwrite** – New array of names as hard replacement
  - app_groups : [“room1”, “room8”] ***Custom CBOR***
  - app_group “room1”
    app_group “room8” ***CoRAL***

- **Addition/deletion** – [[*name_to_remove], [*name_to_add]]
  - app_groups_diff : [[“room1”], [“room5”]] ***Custom CBOR***
  - app_group_del “room1”
    app_group_add “room8” ***CoRAL***

- Overwrite and addition/deletion **not together** in the same PATCH payload



Current value [“room1”, “room2”]

The result is [“room1”, “room8”]

The result is [“room8”, “room5”]

Configuration update with PATCH

- **4.00 (Bad request)**
  - Any malformed or invalid payload
  - iPATCH is used as request method, but:
    - ‘app_groups_diff’ is included (Custom CBOR)
    - ‘app_group_del’ and/or ‘app_group_add’ are included (CoRAL)

- **4.09 (Conflict)**
  - New parameter values would yield an inconsistent group configuration

- **4.22 (Unprocessable entity)** might be returned just as per RFC 8132
  - The server is unable to or is incapable of processing the request