

Work in progress towards

Key Provisioning for Group Communication using ACE
draft-ietf-ace-key-groupcomm-12

Francesca Palombini, Ericsson
Marco Tiloca, RISE

ACE WG Interim Meeting, May 11th, 2021

Open point from previous interim

ace-key-groupcomm defines an 'sign_info_entry' as:

```
sign_info_entry = [  
  id : gname / [+ gname],  
  sign_alg : int / tstr,  
  sign_parameters : [any],  
  sign_key_parameters : [any],  
  pub_key_enc = int / nil,  
]
```

The new generalized format in *ace-key-groupcomm-oscore* is:

```
sign_info_entry = [  
  id : gname / [+ gname],  
  sign_alg : int / tstr,  
  sign_parameters : [alg_capab_1 : any,  
                    alg_capab_2 : any,  
                    ...,  
                    alg_capab_N : any],  
  sign_capab_1 : [any],  
  sign_capab_2 : [any],  
  ...  
  sign_capab_N : [any],  
  pub_key_enc = int / nil,  
]
```

In a new
Appendix

Ready for future
COSE algorithms
with multiple COSE
capabilities

In the document body

› Option 1: add the following in *ace-key-groupcomm* when defining 'sign_info_entry'

Profiles of this specification MAY define an alternative, extended format to use for each 'sign_info_entry', as including multiple elements between 'sign_parameters' and 'pub_key_enc', rather than only 'sign_key_parameters' (OPT13). The alternative format must still provide all the required information to successfully perform signing operations in the group, consistent with the algorithm specified in 'sign_alg'.

› Option 2: have the generalized format of 'sign_info_entry' already in a new appendix of *ace-key-groupcomm*

Open point from previous interim

- › Option 2: generalized format of ‘sign_info_entry’ already in *ace-key-groupcomm*
 - Preferable, less invasive, less conducive to bad usages in profiles of *ace-key-groupcomm*
 - Does not change the document body; does not change current implementations
 - No objections at the previous ACE interim meeting
 - Open point and proposal re-explained on the mailing list --- No objections
 - › <https://mailarchive.ietf.org/arch/msg/ace/aRwe1NIKjbHsGqNSaIn4ubtwGcQ/>
- › Option 2 now included in a new Appendix B of *ace-key-groupcomm*
 - Editor’s copy at: <https://github.com/ace-wg/ace-key-groupcomm/tree/v-12>
 - › <https://github.com/ace-wg/ace-key-groupcomm/commit/025e37429b1bf628abc2e6d94892c8cb04846ad1>
 - TODO: remove that content from *ace-key-groupcomm-oscore* where originally defined

Thank you!

Work in progress towards

Key Management for OSCORE Groups in ACE
draft-ietf-ace-key-groupcomm-oscore-11

Marco Tiloca, RISE
Jiye Park, Universitaet Duisburg-Essen
Francesca Palombini, Ericsson

Latest addition (with open point)

- › Processing of the Joining Response in *ace-key-groupcomm-oscore*
 - Alignment with work-in-progress v -12 of *draft-ietf-core-oscore-groupcomm*
 - If the OSCORE group uses the pairwise mode, the Group Manager ...
 - › Performs additional checks on the Ed25519/Ed448 public key of the joining node
 - › If the Y coordinate of the key is -1 or 1 (mod p), it cannot be used to derive pairwise keys
 - › If that's the case, the Group Manager **MAY** abort the joining
- › Already included in the Editor's copy of v -11
 - <https://github.com/ace-wg/ace-key-groupcomm-oscore/tree/v-11>
 - › <https://github.com/ace-wg/ace-key-groupcomm-oscore/commit/41201aab3689877731416d59df5b8711a07c684b>
- › Even if a “MAY”, is this excessive? Alternative: let the node join anyway
 - Pro: the node can become a group member and use at least the group mode with signatures
 - Con: possible derivation of pairwise keys with that node will be aborted at runtime

And more from the previous interim meeting ...

Updates since IETF 110 (1/3)

- › Alignments with draft-ietf-core-oscore-groupcomm
 - Enable recycling of Group IDs (was issue #46)
 - Remove redundancies about key type capabilities (was issue #47)
- › Recycling of Group IDs (GIDs) is now allowed to the Group Manager (GM)
 - When a node (re-)joins the group, it receives the GID used in the group
 - The GM stores that GID as the node's "Birth GID", until the node leaves the group
 - When rekeying the group and assigning a new GID*
 - › The GM evicts also the nodes with GID* as their Birth GID (and rekeys the group accordingly)

Updates since IETF 110 (2/3)

- › Removed redundancies about key type capabilities
 - To be stated only once, in the pertinent sets of parameters

General format

OLD CONTENT

NEW CONTENT

Response
from
/authz-info

sign_info_entry = [
...
sign_parameters : [any],
sign_key_parameters : [any],
...]



[[+sign alg capab], [+sign_key_type_capab]]



[+sign alg capab]



[+sign_key_type_capab]

[+sign_key_type_capab]

ecdh_info_entry = [
...
ecdh_parameters : [any],
ecdh_key_parameters : [any],
...]



[[+ecdh alg capab], [+ecdh_key_type_capab]]



[+ecdh alg capab]



[+ecdh_key_type_capab]

[+ecdh_key_type_capab]

key = {
...
cs_params : [+item],
cs_key_params : [+item],
... }



[[+sign alg capab], [+sign_key_type_capab]]



[[+sign alg capab], [+sign_key_type_capab]]



[+sign_key_type_capab]

DELETED PARAMETER

Joining

Response

Updates since IETF 110 (3/3)

- › Generalized format of parameters on COSE capabilities (was issue #48)
 - Current Appendix B in the Editor’s copy
 - Aligned with Appendix H of *draft-ietf-core-oscore-groupcomm*
 - Ready for future algorithms with more capabilities than the COSE Key Type
 - If applied to today’s algorithms, the result is just what already in the document body
- › Consistency check – This affects:
 - Fields in the ‘key’ map of the Joining Response
 - › Defined in this document → OK
 - ‘ecdh_info_entry’ in the response from /authz-info
 - › Defined in this document → OK
 - ‘sign_info_entry’ in the response from /authz-info
 - › Defined in *ace-key-groupcomm* → Thus handled in *ace-key-groupcomm*