Key Management for OSCORE Groups in ACE

draft-ietf-ace-key-groupcomm-oscore-10

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

IETF 110, ACE WG, March 12th, 2021
Recap

› Message content and exchanges for:
  – Provisioning keying material to joining nodes and groups (rekeying)
  – Joining an OSCORE group through its Group Manager (GM)
  – More operations for current members at the GM

› Builds on *draft-ietf-ace-key-groupcomm*
  – Agnostic of the ACE transport profile used by C and GM

› Out of Scope:
  – Authorizing access to resources at group members
    › *draft-tiloca-ace-group-oscore-profile*
  – Actual secure communication in the OSCORE group
    › *draft-ietf-core-oscore-groupcomm*
Updates from -10

Aligned with *ace-key-groupcomm-11*:

- Adopted new format of the ‘*get_pub_key*’ parameter
  - Already added to the Californium implementation

- Improved error handling, using also the new error types
  - Registered error type 7 (“Group currently not active”)
  - Additional error situations are handled and replied to, e.g.:
    - Do certain operations when the group is not active
    - Do certain operations that are not admitted for the current role
    - The Group Manager has no available Sender IDs to assign upon request
Updates from -10

Aligned with *ace-key-groupcomm-11*:

› Possible usage of the extended scope format
  — Registered integer identifier in the “ACE Scope Semantics” registry

› In the Joining Request, ‘control_path’ → ‘control_uri’

› Editorial improvement and renumbering of requirements

Aligned with *core-oscore-groupcomm-11*:

› The Group Manager can recycle Sender IDs, under a same Group ID

› Removed moot group policies on synchronization of sequence numbers
Recent updates to the Group OSCORE security context
– The COSE capabilities of the used Key Type are now stated only once

In this document, those capabilities are stated and sent twice
– We can remove this redundancy here as well. Objections?

<table>
<thead>
<tr>
<th>General format</th>
<th>OLD CONTENT</th>
<th>NEW CONTENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>sign_info_entry = [</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td></td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>sign_parameters : [any],</td>
<td>[+(sign alg capab), [+(sign_key_type_capab)]</td>
<td>[+(sign alg capab)</td>
</tr>
<tr>
<td>sign_key_parameters : [any],</td>
<td>[+(sign_key_type_capab)]</td>
<td>[+(sign_key_type_capab)]</td>
</tr>
<tr>
<td></td>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>

| ecdh_info_entry = [     | …                                                                          | …                                                                          |
|                         | …                                                                          | …                                                                          |
| ecdh_parameters : [any],| [+(ecdh alg capab), [+(ecdh_key_type_capab)]                               | [+(ecdh alg capab)                                                         |
| ecdh_key_parameters : [any],| [+(ecdh_key_type_capab)]                                                  | [+(ecdh_key_type_capab)]                                                  |
|                         | …                                                                          | …                                                                          |

| key = {                 | …                                                                          | …                                                                          |
|                         | …                                                                          | …                                                                          |
| cs_params : [+item],    | [+(sign alg capab), [+(sign_key_type_capab)]                               | [+(sign alg capab)                                                         |
| cs_key_params : [+item],| [+(sign_key_type_capab)]                                                  | [+(sign_key_type_capab)]                                                  |
|                         | …                                                                          | …                                                                          |
Next steps

› Align with core-oscore-groupcomm
   – Remove redundancy about key type capabilities
   – Recycling of Group IDs → Additional logic at the Group Manager
   – Signing keys and Diffie-Hellman keys
     › Comment from Ben [1][2] about the pairwise mode of Group OSCORE; required DH keys are now (derived from) the identity signing keys
     › (a) proof of correctness; or (b) separate provisioning of DH keys
     › Ongoing work on (a); option (b) would need to be in this document

› while (ace-key-groupcomm != ready) {
   align with updates in ace-key-groupcomm ;
}

[1] https://mailarchive.ietf.org/arch/msg/core/ujj_LlqW9fq_quh-YqKS0fF0/
[2] https://mailarchive.ietf.org/arch/msg/core/YRXvtiFmHLk5YkXK8-uJg-t3NU/
Thank you!

Comments/questions?

https://github.com/ace-wg/ace-key-groupcomm-oscore