Group OSCORE Profile of the Authentication and Authorization for Constrained Environments Framework

draft-tiloca-ace-group-oscore-profile-00

Marco Tiloca, RISE
Rikard Höglund, RISE
Ludwig Seitz, RISE
Francesca Palombini, Ericsson

IETF 105, ACE WG, Montreal, July 25th, 2019
Motivation (1/2)

› Application scenarios with group communication
  – Group OSCORE provides security also over multicast
  – What about access control for resources at group members?

› For very simple use cases
  – Straightforward and plain access control may be just fine
  – Joining the security group is enough to access resources
  – Any group member can do anything at any other group members’ resource

› For more complicated use cases
  – Different clients can have different access rights
  – Creating (many) more groups poorly scales and is hard to manage
  – Instead, use ACE to enforce fine-grained access control. However …
Motivation (2/2)

› Every current profile of ACE
  – Does not cover secure group communication between C and RSs
  – Relies on a single security protocol between C and RS

› OSCORE profile
  – C and RS must use OSCORE
  – The Token is bound to the OSCORE Security Context
  – Group OSCORE is simply not admitted

› We cannot use Group OSCORE and ACE-based access control of resources
Contribution

› New Group OSCORE profile of ACE
  – Builds on the OSCORE profile
  – Admits two security protocols: OSCORE and Group OSCORE
  – Assumes that C and RS have already joined a same OSCORE group

› Outcomes
  – Pairwise OSCORE Security Context $ctx$
  – Token bound to both $ctx$ and the Group OSCORE Security Context $g_ctx$
  – $ctx$ is bound to $g_ctx$, i.e. $ctx$ derivation relies also on $g_ctx$ parameters

› Properties
  – Proof-of-Possession of the OSCORE Master Secret in the Token
  – Server Authentication (through OSCORE or Group OSCORE)
  – Proof-of-Group-Membership for that exact Client (Token bound also to $g_ctx$)
The C-to-AS Access Token Request includes also:
- The **Sender ID** (‘kid’) of the Client in the OSCORE group
- The **Group ID** (‘kid_context’) of the OSCORE group
- New request parameters: ‘salt’ and ‘context_id’

The AS-to-C Access Token Response includes also:
- Namesake parameters of the OSCORE Sec Ctx Object
- Same OSCORE Sec Ctx Object in the Access Token

Token POST and response
- Exchanges of nonces N1 and N2 as in the OSCORE profile
- RS stores the Access Token with {**Sender ID; Group ID**}
Overview – $\Delta$s from OSCORE profile

› Derivation of the pairwise OSCORE Security Context $ctx$
  – Extended parameters, through more concatenations
  – Use also information related to the OSCORE Group

› Context ID = N1 | N2 | $<$Group ID of the OSCORE group$>$
  – The Group ID of the OSCORE group is also in the Access Token, as ‘context_id’

› Salt = $<$Sender ID of C in the OSCORE group$>$ | $<$Master Salt in the OSCORE group$>$
  – The Sender ID of C in the OSCORE group is also in the Access Token, as ‘salt’
  – The Master Salt in the OSCORE group is known to C and RS as group members

› Master Secret = $<$OSCORE Master Secret$>$ | $<$Master Secret of the OSCORE group$>$
  – The OSCORE Master Secret is in the Access Token, as ‘ms’ like in the OSCORE profile
  – The Master Secret of the OSCORE group is known to C and RS as group members
**C – RS1 pairing**

<table>
<thead>
<tr>
<th>C</th>
<th>RS1</th>
<th>RS2</th>
<th>AS</th>
</tr>
</thead>
<tbody>
<tr>
<td>[--- Resource Request ---]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[--- AS Information ------]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>POST /token</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(aud: RS1, sid: 0, gid: abcd0000)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Access Token + RS Information</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(aud: RS1, sid: 0, gid: abcd0000)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>POST /authz-info</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(access_token, N1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;---- 2.01 Created (N2) ------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/Pairwise OSORE Sec /Pairwise OSORE Sec</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Context Derivation/ Context Derivation/</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

0: Sender ID (‘kid’) of C in the OSCORE group

abcd0000: Group ID (‘kid_context’) of the OSCORE group
C – RS2 pairing

0: Sender ID (‘kid’) of C in the OSCORE group
abcd0000: Group ID (‘kid_context’) of the OSCORE group

C

RS1

RS2

AS

--- POST /token ---
(aud: RS2, sid: 0, gid: abcd0000)

--- Access Token + RS Information ----
(aud: RS2, sid: 0, gid: abcd0000)

--- POST /authz-info ---
(access_token, N1’)

--- 2.01 Created (N2’) ---

/Pairwise OSCORE Sec Context Derivation/

/Pairwise OSCORE Sec Context Derivation/
C can access RS1 and RS2 resources, as per the posted Access Token, using OSCORE or Group OSCORE.
Open point

› Risk for impersonation among group members
  – A node n1 asks for a Token, but using the Sender ID of a node n2
  – Then n1 performs authorized actions, yet “blaming” n2 for them

› Solution
  – Bind also the public key used in the group to the Access Token
  – Include the public key and a PoP signature in the Token Request
  – The AS includes also the public key in the Access Token

› Thanks to Jim for this discussion!
Summary

› New ACE profile for secure group communication
  – Two security protocols: OSCORE and Group OSCORE
  – The pairwise context and group context are bound to each other
  – The Access Token is bound also to the group context

› Benefits
  – Enables Group OSCORE together with ACE-based access control
  – Builds on the OSCORE profile and its context derivation

› Need for document reviews
Thank you!

Comments/questions?

https://gitlab.com/crimson84/draft-tiloca-ace-group-oscore-profile