Group OSCORE - Secure Group Communication for CoAP

draft-ietf-core-oscore-groupcomm-11

Marco Tiloca, RISE
Göran Selander, Ericsson
Francesca Palombini, Ericsson
John Mattsson, Ericsson
Jiye Park, Universität Duisburg-Essen

IETF 110, CoRE WG, March 8th, 2021
Update since the November meeting

› Version -11 submitted
  – Addressed more points discussed at the IETF 109 meeting

› Two main open points
  – Admitting to recycle Group IDs in the same group (Christian)
  – Security of using one identity key for both signing and Diffie-Hellman (Ben [3][4])

[1] https://mailarchive.ietf.org/arch/msg/core/pXEyxhbf-s2wgGDzrDhUNPsHZZc/
[2] https://mailarchive.ietf.org/arch/msg/core/quxfWG2mZnp--5gP10PAZOofPwbU/
[4] https://mailarchive.ietf.org/arch/msg/core/YRNXvtiFmHLk5YkXK8-uJg-t3NU/
Updates from -11

› Single format for the `external_aad`
  - For both encrypting and signing operations
  - Removed ‘par_countersign_key’
  - Improved description of last two fields

› Today, COSE algorithms have only “Key Type” as capability
  - In general, 0 or 2+ capabilities; that can happen with future algorithms

› New Appendix H, with future-friendly templates
  - For parameters of the Security Context
  - For ‘par_countersign’ in the external_aad
  - An instance with today’s algorithms produces the formats used in the document body

```latex
aad_array = [
  'oscore_version' : uint,
  'algorithms' : [alg_aead : int / tstr,
                 alg_countersign : int / tstr,
                 par_countersign : [countersign_alg_capab,
                                   countersign_key_type_capab]],
  'request_kid' : bstr,
  'request_piv' : bstr,
  'options' : bstr,
  'request_kid_context' : bstr,
  'OSCORE_option': bstr
]

par_countersign[
  countersign_alg_capab [ c_1 : any,
                         c_2 : any,
                         ...,
                         c_N : any],
  countersign_capab_1,
  countersign_capab_2,
  ...
  countersign_capab_N
]
```
 Updates from -11

› Usage of ‘kid’ in response messages
  – Must be included only if the request was protected in group mode
  – The mode used to protect the response plays no role

› Relaxed rules on recycling Sender IDs in a group
  – Now forbidden only under the same Group ID

› Revised examples of protected messages
Updates from -11

› Additional reason to lose part of the Security Context – Section 2.4.1.2
   – Reached the limit of Recipient Contexts, due to memory availability
   – Delete a current Recipient Context, to make room for a new one

› Hereafter, each new Recipient Context will start with an invalid Replay Window
   – Get rekeyed by the Group Manager; or
   – Run the Echo exchange in Appendix E, achieving also freshness as byproduct

› Overall, improved distinction between anti-replay and freshness
   – Server “synchronization” with a client is related to freshness, and achievable with Echo
Updates from -11

Some “major editorial” changes

› Reorganized Sections 2.4.*, to better stress cause-effect relations
  – Causes: loss of mutable Security Context; exhaustion of Sender Sequence Number
  – Effect: ask the Group Manager for new keying material; reset Sender Sequence Number

› Section 9 – Message processing in pairwise mode
  – Rewritten as delta from OSCORE (RFC 8613), plus few additions from the Group Mode

› Removed old Appendix E.1 and Appendix E.2 as moot
  – Revised Appendix E (was E.3), on the Echo exchange as only synchronization method
Open point – Observations and GIDs

› Text to explicitly add
  – If a group member re-joins the group, it MUST terminate all its ongoing observations

› Recycling of Group IDs in a same group
  – Currently forbidden, to avoid possible issues with long-lasting observations
  – Reminder: observations survive a change of Sender ID and Group ID

› A client C1 starts an observation with (GID1, KID1, PIV1)
  – C1 obtains a new ‘kid’ = KID2; its observation continues as (GID1, KID1, PIV1)
  – … The group is rekeyed many times … The gid “wraps” and becomes GID1 again
  – A client C2 with ‘kid’ = KID1 legitimately starts an observation (GID1, KID1, PIV1)

→ One notification would match and decrypt against two observations
Open point – Recycling Group IDs

› Solution to enable Group ID recycling
  – The Group Manager (GM) retains the Gid that a node obtains upon group joining, i.e. its “birth Gid”
  – Before rekeying the group, the GM checks if the new Gid is any current member’s “birth Gid”

› If such members are found, the GM removes them from the group and rekeys accordingly

› Those evicted nodes will ask the GM for the latest keying material
  – Since they are not group members anymore, they receive error responses
  – Eventually, they will re-join the group, terminating their observations

› If any of those nodes re-joins before another rekeying has happened
  – The Group Manager MUST NOT rekey the group again upon its joining

Recycling Group IDs is safe → A group can live forever – Objections?
Open point – Github issues #72 #73

› Using identity keys for both signing and Diffie-Hellman [3][4]
  – A DH secret is used to generate encryption keys for the pairwise mode
  – Both usages have the same goal and policy: group communication under a Security Context

› As deviating from common best practices, security has to be well proven
  – Ongoing work to prove this secure in Group OSCORE
  – Build on the paper at [5], as focused on (but not limited to) ECIES settings

› The pairwise mode per se is fine! This is actually about the derivation of pairwise keys
  – Problem alternatively solvable by providing and storing separate Diffie-Hellman keys
  – That’s a last resort, since it would mean more provisioning and storage overhead

[4] https://mailarchive.ietf.org/arch/msg/core/YRNXvtiFmHLk5YkXK8-uJg-t3NU/
Next steps

› Address the two open points
  – Recycling of Group IDs in the same group
  – Usage of identity keys for both signing and Diffie-Hellman

› Submit v -12
  – If no further issues arise, it should be ready to move on
Thank you!

Comments/questions?

https://github.com/core-wg/oscore-groupcomm