Group OSCORE - Secure Group Communication for CoAP

draft-ietf-core-oscore-groupcomm-09

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

IETF 108, CoRE WG, July 31st, 2020
Update since the April meeting

› Version -09 submitted in June
  – Addressed open points raised in April
  – Addressed remaining points from Jim’s and Christian’s reviews

› WGLC on -09, ended the 20th of July

› 2nd interop during this Hackathon

› New discussion item on separate pairwise space for PIVs

[1] https://mailarchive.ietf.org/arch/msg/core/VMhrAPEt4TE8jahatVd1EoDzdMI/
Main updates in -09

› Two different operating modes
  – **Group mode** – Main and usual mode
    › MUST be supported
    › Encryption with group keying material; signature included
  – **Pairwise mode**
    › MAY be supported – If so, use for unicast requests (e.g., Block-wise, Echo, …)
    › Encryption with derived pairwise keying material; no signature

› New Group Flag bit in the OSCORE option
  – Set to 1 if the message is protected in group mode
  – Set to 0 if the message is protected in pairwise mode (aligned with OSCORE)
Main updates in -09

› Pairwise key derivation
  – Same construction from 3.2.1 of RFC 8613
  – **Pairwise key = HKDF(Sender/Recipient Key, DH Shared Secret, info, L)**
    › Sender Key of the sender node, i.e. Recipient Key of the recipient side
    › Static-static DH shared secret, from one’s private key and the other’s public key
  – Compatible with ECDSA and EdDSA (after coordinate remapping)

› Major editorial revision of Section 2 “Security Context”
  – Improved presentation of Common/Sender/Recipient context
  – Derivation of keys for the pairwise mode explained here
  – Update and loss of the Security Context (e.g., in case of rekeying and reboot)

› Usage of update registries and COSE capabilities from COSE-bis
Report from IETF 108 Hackathon

› Tests with RISE and August Cellars implementations

› Successful interop tests
  – Communication in group mode
  – Derivation of pairwise keys

› Successful local tests
  – Communication in pairwise mode
Main points from WGLC

› Information is now replicated in the Security Context
  – Sufficient to keep ‘Counter Signature Parameters’
  – Delete ‘Counter Signature Key Parameters’ as redundant.
  – Issues with that?

› Curve remapping in the pairwise mode, for DH secret derivation
  – Current text Ed25519 (MTI) → Montgomery for X25519 (MTI if supporting pairwise mode)
  – **Jim**: consider remapping to the short-Weierstrass curve instead
  – Mention just as possible alternative? Or have Wei25519 and ECDH25519 as MTI?

› Wrap-around of Sender Sequence Number (SSN)
  – **Jim**: *is the wrap-around of the SSN or of the PIV?*
  – It should really be the SSN, which is used as PIV. *Anything missing to clarify?*
Main points from WGLC

› Support for Observe, across group rekeying
  – Now the client and server store the ‘kid’ of the original Observe request
  – That value is the ‘request_kid’ in the external_aad of notifications, also after rekeying
  – Jim: should we store also the kid context?
  – No need to, it’s not part of the ‘external_aad’. Keep as is?

› New Context established → Reset the Sender Sequence Number to 0?
  – Now it’s not reset, unless the application decides differently
  – Jim: having it reset simplifies the detection of group rekeying
  – Reset also Replay Windows and Observe Numbers of ongoing observations
  – Change to reset by default? Can the application do differently?
Separate SSN spaces

› Right now: every node has a single SSN space
  – Used for PIVs both in group mode and pairwise mode

› New proposal from Jim: two separate SSN spaces
  – One SSN for the group mode
  – For each associated recipient
    › One pairwise SSN – NEW
  – For each associated client
    › One group Replay Window
    › One pairwise Replay Window – NEW
Separate SSN spaces

Pros
  - Less frequent exhaustion of SSN values
  - Reuse of OSCORE code for the pairwise mode

Cons
  - Higher storage (extra SSNs and Replay Windows)
  - Might result in greater communication overhead (fresh PIV in some responses)

Issues
  1. The server might have to use its fresh PIV (no reusage of request PIV)
     › E.g., when request and response are protected in different modes
  2. Separate synchronization of the two spaces for servers
     › The synch method using Echo needs some adaptation (see Appendix E.3)
Separate SSN spaces - Issue #1

1. C → S : Request in Group Mode
   - kid: $SID_C$; piv: $gPIV_C$
   - Nonce built from $\{SID_C, gPIV_C \}$; Key: $gK_C$

2. S → C : Response in Pairwise Mode
   - kid: $SID_S$; piv: NONE
   - Nonce built from $\{SID_S, gPIV_C \}$; Key: $pK_{SC}$

3. C → S : Request in Pairwise Mode
   - kid: $SID_C$; piv: $pPIV_{CS}$
   - Nonce built from $\{SID_C, pPIV_{CS} \}$; Key: $pK_{CS}$

4. S → C : Response in Pairwise Mode
   - kid: $SID_S$; piv: NONE
   - Nonce built from $\{SID_S, pPIV_{CS} \}$; Key: $pK_{SC}$

Request and response are protected in different modes

AND

The server reuses the request PIV (PIV reflection)

If $gPIV_C == pPIV_{CS}$, in (1) and (3)

Nonce reusage with $pK_{SC}$, in (2) and (4)

$\{SID_S, gPIV_C\} == \{SID_S, pPIV_{CS}\}$
Separate SSN spaces - Issue #2

1. C → S : Request in group mode
   - With client’s group PIV

2. S → C : Response in pairwise mode
   - With server’s pairwise PIV and Echo option
   - S stores <kid, gid, piv> from the request at (1)

3. C → S : Request in pairwise mode
   - With client’s pairwise PIV and Echo option
   - Should also include the client’s group PIV

› Need more discussion, especially with implementers
   - Weigh pros/cons and performance tradeoffs

› Opinions about separate SSN spaces?

a) In a new CoAP option

b) In the payload, next to the ciphertext
   - Length signaled in the OSCORE option
   - Need to integrity protect?
   - How for (b)? Use the external_aad?
     - It deviates from OSCORE format
     - Not ideal for code reuse
Next steps

› Addressing WGLC comments in version -10
  – Jim
  – Peter

› More discussion on separate PIVs for the pairwise mode

› More interop tests in pairwise mode
Thank you!

Comments/questions?

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