Some more MLS Extensions

Rohan Mahy — rohan.ietf@gmail.com

IETF 119, 22-Mar-2024

Extensions for MIMI

- MIMI is currently planning to have room state shared as GroupContextExtension to get **group agreement**.
- Richard Barnes proposed a generic application state extension that enables this.
  - MIMI would use this for
    - the **room policy document** (ex: This room is a members-only room. You have to have the “admin” role to add and remove users); and
    - the **participation list** which contains users and their roles (ex: Alice is an admin, Bob is a regular participant)
  - The participation list is updated (patched?) via a new proposal type which does not require an *UpdatePath*
  - The actual policy will likely be defined in MIMI.
- What else do we need in MLS?
  - KeyPackage Context?
  - Pending Proposals in External Commits?
  - Improvements to Conveying ratchet tree?

draft-mahy-mls-kp-context

- KeyPackage extension which restricts the use of the KeyPackage to a specific context
  - Only use this KeyPackage to join a specific MLS group
  - This KeyPackage is only meant to be used if the Adder has the following “user” identity
  - This KeyPackage is only meant to be used if the Adder is in the following domain
  - This KeyPackage is only meant to be used if the Adder has the following public key
- Might add
  - Only use this KeyPackage to join a specific “room”
- Any other contexts we might be missing?
- Next steps? Add to extensions draft?
- Could be incorporated into **AppState general solution**

SelfRemove proposal in MLS Extensions

- Open Issue: User still cannot ensure that removing oneself is atomic
  - Option 1: User's client sends a commit with Remove proposals for other clients, then sends a SelfRemove Proposal. Takes 2 epochs to remove, with one client present
  - Option 2: User's client sends Remove proposals for other clients and SelfRemove proposal at the same time. External Commit still is obliged to ignore the Remove proposals.
- Solutions:
  - Option A: Add list of other client indexes to delete (must be clients of the same user) to the SelfRemove
  - Option B: Change the behavior of External Commit in presence of SelfRemove to commit valid Remove proposals. Makes the extension no longer a safe extension?
  - **Option C**: Generically support sending all valid pending proposals in an external commit.

Include Pending Proposals in External Commits

- Currently external commits don’t include otherwise valid pending proposals

- Extension to require external commits to include all valid pending proposals (from wherever new joiner got the GroupInfo)
  - DS is responsible to provide pending proposals too if it provides GroupInfo
  - Clients need to accept External Commits which include the pending proposals by reference
  - Clients sending External Commit need to fetch and include valid pending proposals
- Not a safe extension. Its an ordinary GroupContext extension.

Why?

  - Solves consistency problem
  - Means we don’t need the SelfRemove proposal type

Options sending Ratchet Tree and GroupInfo

- Conveying the ratchet tree
  - RFC9420 only describes how to convey entire ratchet tree as an extension in Welcome and/or GroupInfo.
  - Already describes that ratchet tree for Welcome can be out-of-band but not *how*.
    - Some MLS DS's reconstruct the ratchet tree from Commits/Proposals
    - Could be provided via an HTTPS URL or stapled to an MLS message
- Conveying the GroupInfo
  - Client needs to provide at least the GroupInfo signature and any GroupInfo extensions (external_pub). Otherwise DS can reconstruct a GroupInfo.
  - Provide either full GroupInfo or a PartialGroupInfo (signature + GroupInfo ext)

~~draft-mahy-mls-x25519kyber768draft00~~ → draft-mahy-mls-xwing-00

- Desire for handful of complimentary post-quantum (PQ) security extensions for MLS:
  - Straightforward MLS cipher suite: replace classical KEM with a hybrid PQ/traditional KEM. Drop-in replacement in many MLS libraries without changes to any other part of the MLS stack. Single KEM which is performant and works for the vast majority of implementations. Address harvest-now / decrypt-later using the simplest, most practicable solution available. ⬅ **You Are Here**
  - Versions of existing cipher suites that use PQ signatures; and specific guidelines on the construction, use, and validation of hybrid signatures.
  - One or more mechanisms which reduce bandwidth and/or storage requirements; or improve performance (ex: by updating post-quantum keys less frequently than classical keys, or by sharing portions of PQ keys across a large number of clients or groups.)
- NIST announced the ML-KEM standard based on Kyber. Need to use slightly more complicated combiner for ML-KEM vs plain concatenation with Kyber.
- Poll for adoption?