This is the MIMI Age!

Let's Massage MIMI into shape!

Rohan Mahy — rohan.ietf@gmail.com

19-Mar-2025

MIMI Protocol

Rohan Mahy — rohan.ietf@gmail.com

19-Mar-2025

MIMI protocol changes since IETF121

- Franking fixes (add franking integrity mechanism; use salt in MIMI content) #90, #96, #100
- Allow search for multiple fields #94
- Add minimal metadata pseudonyms #101 (preso shortly)
- Move in participant list and room metadata from draft-mahy-mimi-app-components #102
- Minor struct improvements
  - SignWithLabel was missing the name of its key #92
  - Fix type of encrypted_groupinfo_and_tree #97
  - When HandshakeBundle in update contained multiple proposals, fanout together #106
  - Fix GroupInfoResponse in error case #107
- CI, draft metadata, tag updates #98, #99, #105, #108

PRs

- Proxy download endpoint #111 (**more**)
- Improve integrity of KeyPackage requests #112
- Add nonce to GroupInfoResponse signature to prevent 2-time pad #113

Download proxy PR

a.com → storage.a.com
hub.net → assets.hub.net
b.org → assets.b-org.hub.net

- **Attachment** policy in a room has a specific host for each provider represented in the room. Clients check for sender Alice an attachment is only coming from `storage.a.com` and for sender Bob an attachment is only coming from `assets.b-org.hub.net`
    - Hub checks that target is an allowed storage domain. Prevents endpoint from becoming an open proxy
- `downloadProxy` endpoint on Hub prevents sender's local provider from seeing IP addresses of room members.
- My proposal to modify this PR
    - Make `downloadProxy` endpoint mandatory to implement on the Hub
    - Make OHTTP Gateway mandatory to implement on the Hub; OHTTP Relay mandatory to implement on local providers.
    - Clients can decide to fetch attachments directly, only use `downloadProxy` endpoint, or use OHTTP (RECOMMENDED)

Issues

- Stuff that has PRs already #24 #110
- Normative
    - Timestamp integrity for history #114
    - Figure out versioning #52
    - Do we need knocks #38
    - Tracking arbitrary state #23 - I think we now have this using app_data_dictionary, but need members-only (**more**)
    - Transport protocol #26 - draft has been using HTTPS for ages, no concrete counter proposal. Can we close please?
    - Binary encoding #25 - draft has been using TLS PL for ages, no concrete counter proposal. Can we close please?
- Non normative
    - MIMI threat model #93
    - Add Pseudonym flows #84 (depends on finishing MMR)
    - Make ASCII art fit in 72 chars #68

Arbitrary State (Issue #23)

- The new MLS AppDataUpdate and app_data_dictionary extensions in MLS extensions allows for applications to efficiently put arbitrary state in the MLS GroupContext or convey it in KeyPackages, LeafNodes, and GroupInfos.
- AppEphemeral allows members to send some arbitrary state to the group as well (ex: include a join code with a new joiner).
- This seems to satisfy most of the request in Issue #23
- What about for state that is “members-only”? We could encrypt it and leave the key distribution to members to another mechanism. Who else needs to solve that problem? Minimal Metadata Rooms!

Minimal Metadata Rooms

What's Next?

- Are there any other endpoints we absolutely need for MIMI protocol?
- Can we have this ready for WGLC by Madrid?

MIMI Content

Rohan Mahy — rohan.ietf@gmail.com

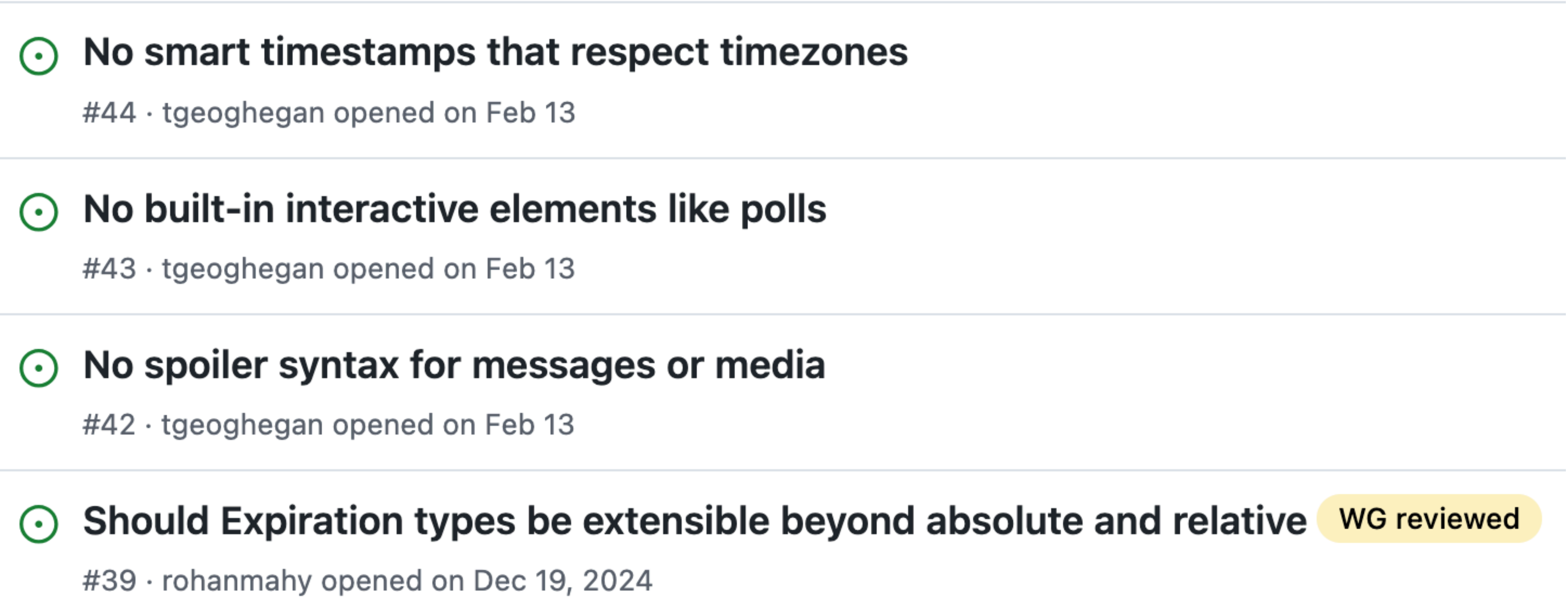
19-Mar-2025

Changes since -05

- Mega PR #46
  - Rebuilt the examples using script
  - Added salt once more to prevent SHA256 length extension attack
  - Future proof timestamps: either
    - integer: milliseconds since start of UNIX epoch OR
    - tagged: CBOR extended time 1001({ etc. }) from RFC 9581
  - Lots of consistency fixes
  - Uses mimi://example.com/u/alice-smith style names
- Removed lastSeen #37
- Remove delivery report timestamps #51
- Warn clients not to double render CID referenced parts #48 (Issue #30)
- IANA Register MIMI extensions #50
- Some typos / consistency fix in CDDL #40, #41
- Made draft build editor's copy (added Martin Thomson template) #52

Issues

- How we reference inline content / handle multiple parts
  - Issue #30 covers much of the discussion
  - Issue #49 contains a very stripped down content attachment mechanism
- Propose won’t fix - can add via extensions later if needed



No smart timestamps that respect timezones
#44 · tgeoghegan opened on Feb 13

No built-in interactive elements like polls
#43 · tgeoghegan opened on Feb 13

No spoiler syntax for messages or media
#42 · tgeoghegan opened on Feb 13

Should Expiration types be extensible beyond absolute and relative WG reviewed
#39 · rohanmahy opened on Dec 19, 2024

Next Steps

- Close these open issues
- Immediately start a WGLC
- Note: Delivery reports need some eyeballs. Move to another draft?

MIMI Room Policy

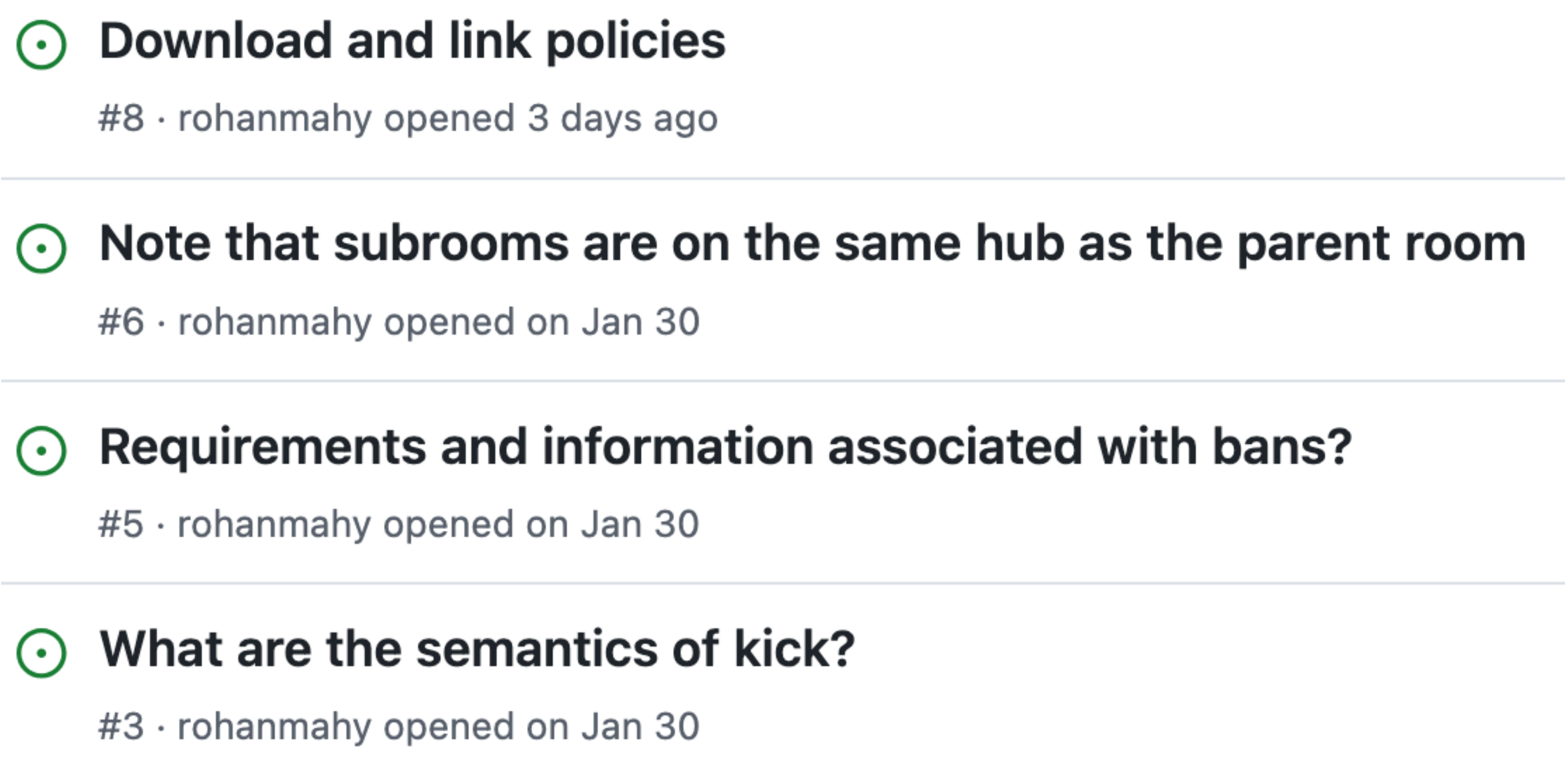
draft-ietf-mimi-room-policy

Rohan Mahy: rohan.ietf@gmail.com

MIMI interim, 12-Feb–2025

Changes and Issues

- Changed since IETF 121
  - Moved Roles, Capabilities, and Preauthorized users (from draft-mahy-mimi-app-components) into the draft.
  - Removed contradictory text/mechanism about roles and preauthorization
- Issues



⊙ **Download and link policies**

#8 · rohanmahy opened 3 days ago

⊙ **Note that subrooms are on the same hub as the parent room**

#6 · rohanmahy opened on Jan 30

⊙ **Requirements and information associated with bans?**

#5 · rohanmahy opened on Jan 30

⊙ **What are the semantics of kick?**

#3 · rohanmahy opened on Jan 30

Role Definitions

- Which capabilities a holder of the role has
- Constraints on min/max number of participants and active participants
- How holder may change the role of other participants (next slide)

- role_index
  - zero is for non-participants
  - one can be used for banned participants
- Role name
- Role description

Role-Based Access Control — role changes

- With some capabilities Holder may change the role of other participants:
  - canChangeUserRole
  - canChangeOwnRole
  - canAddParticipant
  - canAddOwnClient
  - canAddSelf
  - canUseJoinCode
  - canRemoveParticipant
  - canRemoveOwnClient
  - canRemoveSelf
  - canKick
  - canChangeUserRole
  - canChangeOwnRole
  - canBan
  - canUnban
- Role transitions
  - adding, moves from role 0 to a specific role; removing moves to role 0
  - banning moves to role 1; unbanning moves from role 1 (if role 1 exists)
  - authorized_role_changes = [(0,[1,2,3]), (1,[0,2,3]), (2,[0,1,3]), (3,[0,1,2])]

Role-Based Access Control — external senders

- **New**: Hub/Anti-spam external senders can be authorized via one or more roles (ex: `policy_enforcer`) in the participant list.
- They are authenticated via MLS `external_senders` credential, but authorized via RBAC mechanism.
    - Configuration can give the policy enforcer the capabilities to delete clients/users but not add them.
    - Constraints also can prevent policy enforcer from becoming an *active* participant

Preauthorized users

- **Used to give a role to users based on identity properties**
  - **When is it consulted?**
    - When attempted joiner is *neither* in Participant List *nor* has join code
    - When requester is in Participant List and tries to change its own role (ex: become a moderator or admin after being added as a regular user)
  - How do you determine if user is authorized?
    - List of claims per preauth entry. The client must match all claims (AND semantics) to be authorized for the role in an entry.
    - Client gets the first matching role. For example, entries that are banned go first, then the most powerful roles to least powerful.

MIMI Identity

draft-mahy-mimi-identity

Rohan Mahy — rohan.ietf@gmail.com

19-Mar-2025

Do we need this draft?

- This draft is still largely a survey of the problem space. But it exposes a few issues MIMI needs to solve
- If different providers use different MLS credential types, what do we need to make things work?
  - Each client needs to be able to validate all the credential types in the room
    - includes managing trust roots for issuers
    - includes any revocation / status lists used
  - They need to be able to find the signature public key in each credential
  - They need to be able to recognize the same client if it tries to replace itself in a room
  - They need to be able to construct a canonical participant identifier from each credential
    - Proposes using the mimi: URI scheme as canonical representation inside MIMI protocol (ex: participant lists). Can use whatever representation inside your credential / AS as you want.
  - Need consistent way to come up with display name and/or handle for participants as well

What's next?

What other policy areas need addressing?