Beyond Olm

About challenges and opportunities in Messaging Layer Security (MLS)

Hej, jeg er The one with the braid

- [matrix] Consultant & Software Architect
  - Client development
  - OIDC
  - Sliding Sync
  - End-to-End Encryption
- Cryptography
- Unix, Flutter & AArch64
- ask me about night trains 

Part 1: Olm in detail

Olm	Megolm
Encrypt to-device messages	Encrypt room events
Secure channel for key sharing	Keys for Megolm are shared using Olm

Key types

- Device fingerprint keys
- Device identity keys
- One time keys
- Fallback keys
- Megolm encryption keys
- Megolm signing keys

Device fingerprint keys

- Ed25519
- The device fingerprint
- Signs all other keys of the device
- Public key shared across the [matrix]

Device identity keys

- Curve25519
- Used to deviate shared secrets
- Signed using the fingerprint keys
- Public key shared across the [matrix]
- Could in future be rotated

One time keys

- Signed Curve25519
- Disposable, single-use keys
- There are looooots of them
- Claimed by other users to establish Olm sessions

Fallback keys

- Signed Curve25519
- Similar to one time keys but not disposable
- Used once one-time keys are consumed
- New fallback keys regenerated once device online

Megolm encryption keys

- Random secret
- Used to derive AES-256 and HMAC-SHA-256 keys
- After each sent message, a hash derives the next key
- Future messages can be decrypted but past ones can’t

Megolm signing key

- Ed25519
- Used to sign messages sent via Megolm
- Public key shared in the room along with encryption key

Part 2: MLS in detail

Terminology

- Group
- Epoch
- Member
- KeyPackage
- Group Context
- Proposal
- Commit

Group

- Group of clients
- Hold a common secret
- Sequence of *epochs*

Epoch

- Snapshot of a *group*
- Depending on their predecessor

Member

- A client included in shared group state
- Access to group’s secrets

KeyPackage

- Signed business card of a client
- Contains cryptographic identity
- Hybrid Public Key Encryption (HPKE)
- Used to introduce clients into groups

Group Context

- Collected public state of a group
- Used as introduction for new members

Proposal

- Message suggesting to add or remove a member

Commit

- Implements a set of proposals in a group

Secrets

- Commit Secret
- Epoch Secret
- Encryption Secret

Architecture

Services

Authentication Service (AS)

- Trusted service
- Authenticates presented credentials

Delivery Service (DS)

- Untrusted service
- Routes messages to the participants

Lifetime of an MLS group

Creation

1. Publish KeyPackage via DS
2. Propose KeyPackage in group as PublicMessage
3. Commit KeyPackage as PublicMessage
4. Welcome message for new member as direct PublicMessage

Part 3: Blockers of MLS in [matrix]

MLS on [matrix]

- DS
  - Directory with KeyPackages
  - Group Channels with MLS messages as Rooms
- AS
  - Basically Key Verification

A group has a single linear sequence of epochs.

But Federation & Partition 😥👉👈 ?

We generally assume that each participant maintains a complete and up-to-date view [...] of the group [...].

But Federation & Partition 🥺 👉 👈 ?

A group member that has observed proposals within an epoch MUST send a Commit message before sending application data.

But Federation & Partition 😟👉👈 ?

Key Differences

- Common view of the group
- Cryptographic membership vs. [matrix] membership + power levels
- State events vs. group state
- Rotated KeyPackages vs. OTKs
- Append-only approach
- Partition and divergence

Branching and Reunification ?

are we MLS yet ?

Not Yet.

MLS out there

- MSC2883: Matrix-flavored MLS
  - Decentralized MLS (DMLS)
- MSC4244: MLS for Matrix
  - Hub servers
- MSC4256: MLS mode Matrix
  - German BWI MLS work

The solution ?

Decentralize !

Get in touch

- I’m [matrix] consultant and Software Architect
- info@braid.business
- @braid:alsace.hair

🌈 rights are human rights !