MIMI Problem Statement

Rohan Mahy
rohan@wire.com

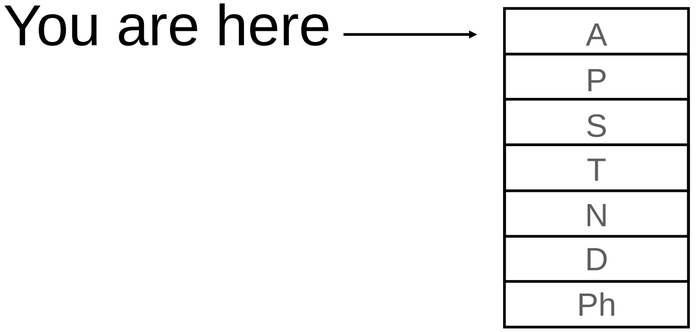
Background

Where we are today

- Instant messaging is **widespread** and **feature rich**: plain text and rich text messaging, delivery notifications, read receipts, replies, reactions, and more; sharing files/audio/videos; and many support calling and/or conferencing features
- Most modern messaging services are **end-to-end encrypted**. Some messaging vendors are implementing MLS
- The lack of interoperability is causing a **poor user experience**.

What's different from last time

- IETF standardized IM for XMPP and SIP decades ago, but the implementations and environment have both changed significantly. Messaging applications now share more common features; the MLS protocol has been developed to facilitate E2EE interoperable messaging; and external pressure to interoperate has increased significantly (ex: because of the EU Digital Markets Act).
- Alternative: publication of vendor-specific APIs without standardization
  - Would perpetuate suboptimal outcomes for users and app developers
  - supporting pairwise communication flows would create a patchwork of inconsistent user experience and many bugs
- The theory behind MIMI
  - a **minimal standardized framework** to enable cross-app communications will provide consistency while leaving app developers freedom to continue to make their own design choices

You are here 

Content / Message Formats

- Content / Messages in IM systems usually consists of:
  - plain text / rich text messages
  - replies / reactions / mentions
  - edits / deletes / expiring
  - read receipts
  - (links to) attachments / images / videos / recorded sound
  - bootstrapping for calls / links for conferences

- Content typically e2e encrypted—can’t translate it with gateway.
  Everyone in the group gets the same content. Need common format(s).
- Likely to be many formats. Communicate which are supported / required / sent

End-to-end Encryption MLS “Profile”

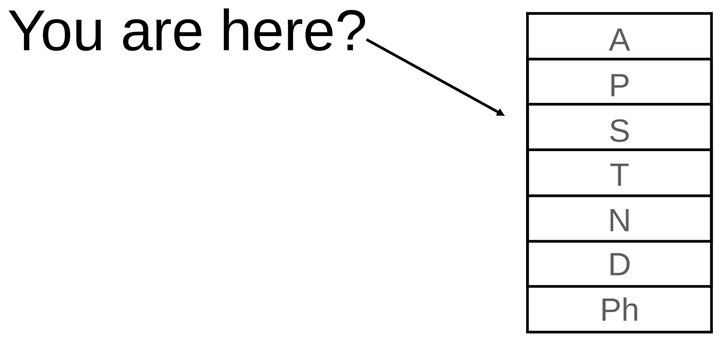
In MLS the Authentication Service (AS) and Distribution Service (DS) are abstract services.
In MIMI, the AS/DS likely include multiple services *in different administrative domains*.

Services can be split between clients and “servers”.

The MLS Architecture† is flexible and expects applications to specify:

- Authentication Service
  - Issuing credentials to clients
  - Enabling client to verify a credential presented by another client
- Distribution Service
  - Ordering of handshake message
  - Routing MLS messages among clients
- KeyPackage Storage
- Administrative/Policy Knobs (ex: ciphersuites, timers, acceptable/required extensions, authorization rules, etc.)

† https://www.ietf.org/archive/id/draft-ietf-mls-architecture-09.html



Message Transport Protocols

- Message Transport protocols in this context carry at least the MLS messages.
- Could include other useful features
- server-to-server
  - we could leave this undefined and create gateways
  - but desirable to have
  - determining how to resolve the server(s) for a specific service is related problem

- client-to-server
  - usually an implementation-specific matter
  - but needs to meet all the requirements for MLS

Identifiers in Instant Messaging

- Key items we are identifying in IM protocols:
  - users
  - client
    - in many IM systems, a user can have many clients (ex: mobile and desktop)
  - groups (conversations / chats / channels)

vs.

Items we display/search on:

- email addresses
- phone numbers
- full names
- service handles

- An identifier used internally is not always what the end-user sees in the UI

- The internal identifiers need to be compatible with the protocols we use:
  - message transport; MLS; content format

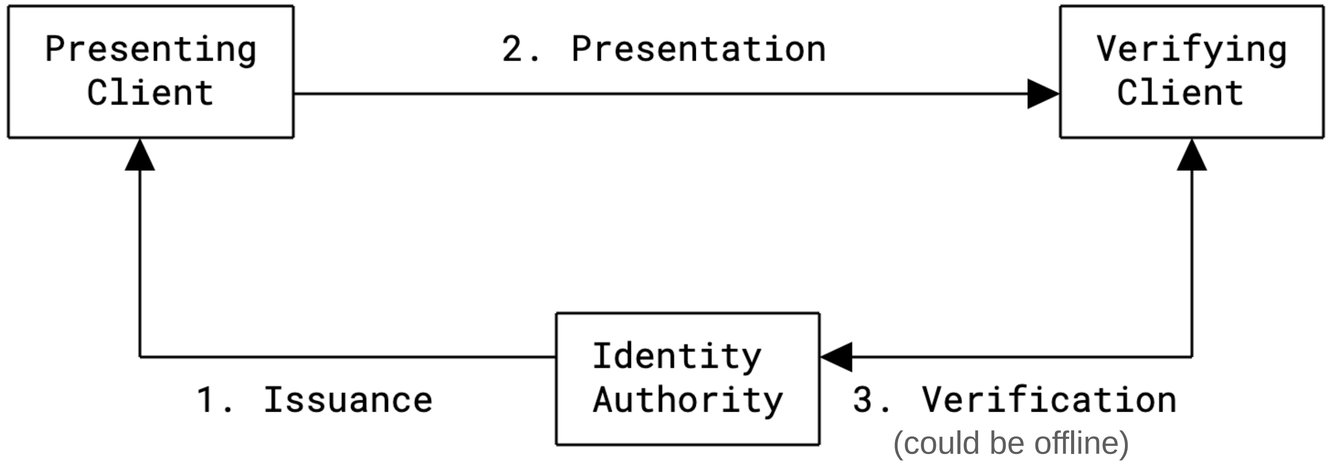
End-to-end Identity

- Prevent ***impersonation***
  - across services
  - within a service

- How do you strongly associate a client with a specific user identity?

- How does a client verify a presented identity?

- Assumes protocols defined elsewhere



Clients get a credential ❶ (for example a certificate) from an Identity Authority and include it in MLS ❷.

Other clients verify ❸ the credential of each client (online or offline)

The Introduction Problem

- How do you get the internal identifier and permission to send messages to someone? (assuming you know the target service)
  - Look up the target using some other info, ex:
    - email
    - phone number
    - name
    - handle
  - Need to preserve privacy and preferences of target
  - Often implemented with a connection request
- Discovery of the IM Service
  - How do we determine target’s preferred service? (ex: WhatsApp vs iMessage)

What’s left?

SPAM and Abuse prevention hooks

- Metadata processing to manage spam and abuse
  - ex: Communicate that a user was blocked and/or reported for suspected spam
  - Having a strong identity makes the problem easier.

Administration / Moderation

- Interoperable mechanisms for group administration or moderation across systems

Thanks!