MLS Federation

MLS interim Jan 2019

Emad Omara

Why ?

- To allow users who are using different applications operated by separate entities to securely exchange messages

- These applications can already communicate with each other, but they lack an encryption layer

How ?

- Define the wire format for all user messages (MLS currently define the handshake messages only)

- Define the protocol between the client and the server (both delivery and authentication) in a new document.
  - How to retrieve user init key and identity key
  - How to fan out the group messages

- Operator servers discovery is out of scope

How ?

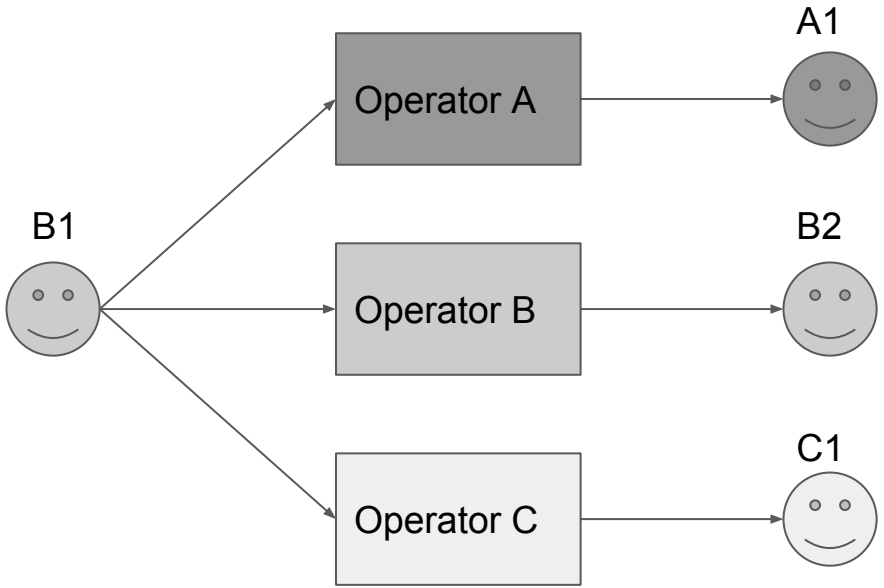
- Clients need to know how to retrieve key from servers operated by different entities
  - Either the client issues multiple requests for different servers, which means the client will also do fan out for user messages
  - Or proxy this through the client’s operator server

- Servers should store the minimum information needed to deliver the messages, as any state or metadata stored in multiple servers have to be in sync

Client fan-out

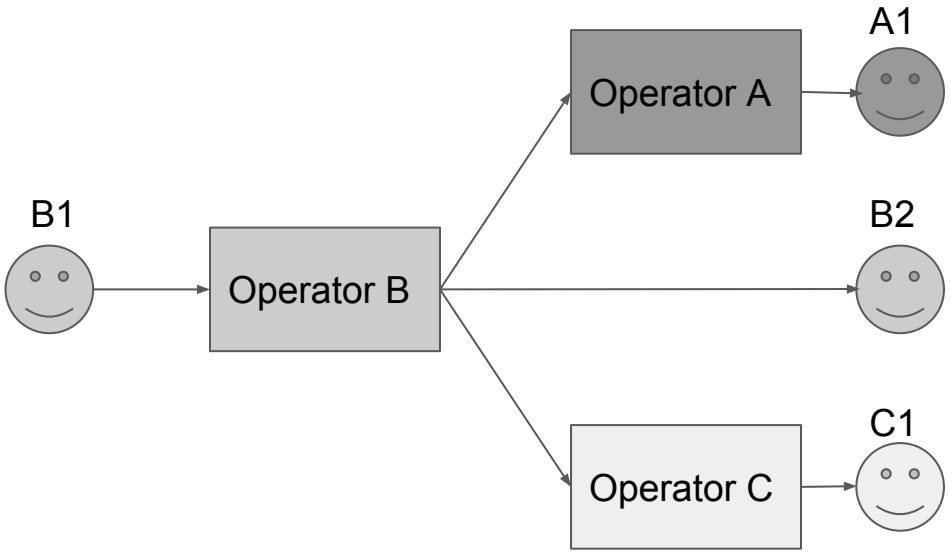
- Client B1 establishes multiple connections with different servers to retrieve keys and deliver messages

- Not scalable!



server fan-out

- Client B1 establishes one connection with its server
- Operator B server will proxy all key requests to other servers and fan out the messages



Challenges

- MLS protocol already relies on the server to store some metadata about the group and order enforcing. How this works with multiple servers ?

- Version negotiation becomes even harder

Next step?

- Should this be part of this WG?
- Prepare ID for the client-to-server and server-to-server protocols
- Update the charter to include application federation