Network Working Group
Internet-Draft
Intended status: Informational
Expires: 9 July 2026

K. Kohbrok
R. Robert
Phoenix R&D

5 January 2026

MIMI Portability
draft-kohbrok-mimi-portability-05

Abstract

This document describes MIMI Portability mechanisms.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 9 July 2026.

Table of Contents

1. Introduction

draft-robert-mimi-delivery-service and others describe a transport and delivery mechanism for messages in a federated environment that relies on the concept of a fixed Delivery Service that is in charge of orchestrating the communication for a given MLS group. All clients of a given MLS group agree on what Delivery Service to use and rely on it to solve the problem of Commit message ordering.

While having a fixed Delivery Service solves a class of synchronization problems, it can sometimes be a limiting factor, especially because it introduces reliance on a specific operator. There are legitimate scenarios where the operator of a Delivery Service no longer wants to operate it, or where the users of a group want to switch to a different Delivery Service.

This document describes mechanisms that allow users to switch an MLS group to a different Delivery Service.

1.1. Change Log

draft-01

* Version bump to prevent expiration

draft-02

* Version bump to prevent expiration

draft-03

* Version bump to prevent expiration

draft-04

* Version bump to prevent expiration

draft-05

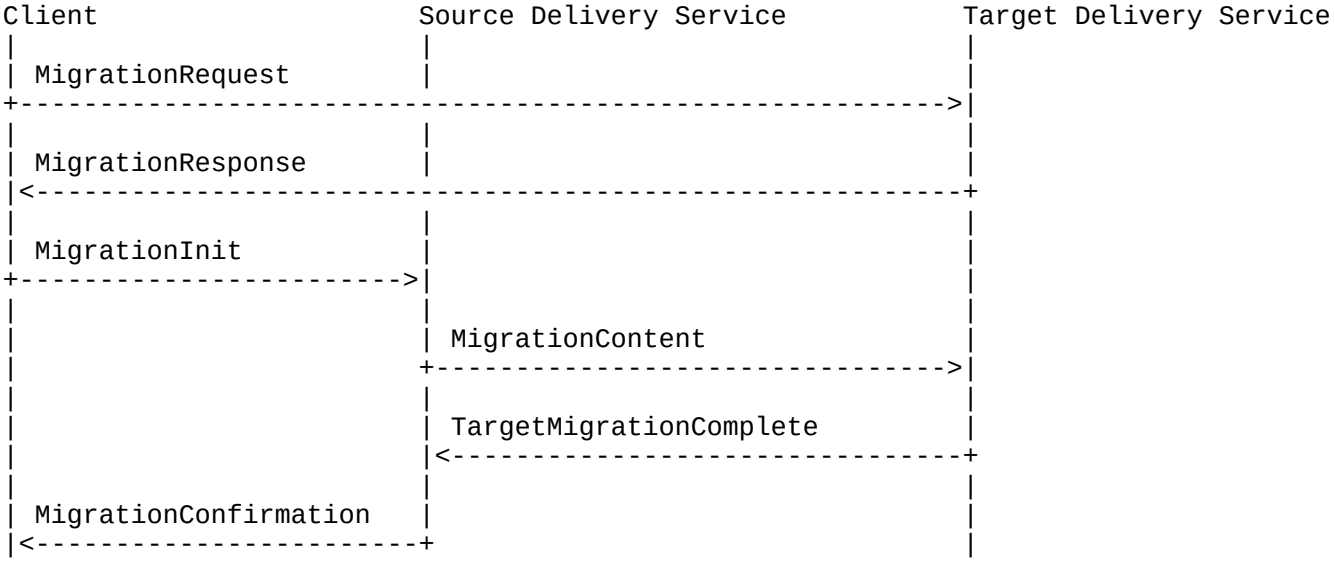
* Version bump to prevent expiration

2. Architecture

We consider two distinct Delivery Services, the _Source Delivery Service_ and the _Target Delivery Service_. The Source Delivery Service is where the MLS group currently resides, and the Target Delivery Service is where the MLS group should be moved to. The Source Delivery Service and the Target Delivery Service have distinct domain names.

The Source Delivery Service is responsible for exporting the MLS group state and sending it to the Target Delivery Service. The Target Delivery Service is responsible for importing the MLS group state and making it available to the members of the MLS group.

Once the MLS group state has been imported into the Target Delivery Service, the Source Delivery Service no longer has any responsibility for the MLS group. The Source Delivery Service can delete the MLS group state and all associated metadata.



TODO: The MLS group ID is fixed for the lifetime of a group and cannot be changed. This requires group IDs to be globally unique across all relevant Delivery Services. Even though this could be solved using a UUID, groups should also be tied to a specific owning delivery service.

3. Migration

TODO: Describe what keys are used for the signatures.

TODO: Describe that the crypto primitives should be aligned with the ciphersuite of the MLS group.

3.1. Requesting a migration

The migration process is initiated by a client of the MLS group. The client requests a MigrationResponse from the Target Delivery Service. The Target Delivery Service returns a MigrationResponse that can be used by the Source Delivery Service to transfer the MLS group state to the Target Delivery Service.

The client sends the following MigrationRequest message to the Target Delivery Service:

```
struct {
  opaque source_domain_name<V>;
  opaque target_domain_name<V>;
  opaque group_id<V>;
  uint64 epoch;
} MigrationRequest;
```

The Target Delivery Service responds with a MigrationResponse message:

```
struct {
  MigrationRequest migration_request;
  uint8[32] nonce;
  opaque signature<V>;
} MigrationResponse;
```

3.2. Initiating a migration

The client sends the following MigrationInit message to the Source Delivery Service:

```
struct {
  MigrationResponse migration_response;
  opaque signature<V>;
} MigrationInit;

struct {
  opaque target_ds_domain<V>;
} MigrationCommitAAD
```

The client also sends a Commit message to the group, where the AAD consists of a serialized MigrationCommitAAD struct.

The Source Delivery Service sends a MigrationContent message to the Target Delivery Service:

```
struct {
  MigrationResponse migration_response;
  opaque group_state<V>;
  opaque signature<V>;
} MigrationContent;
```

The Target Delivery Service responds with a TargetMigrationComplete message:

```
struct {
  opaque migration_content_hash<V>;
  opaque signature<V>;
} TargetMigrationComplete;
```

The Source Delivery Service proceeds to fan out the client's Commit message that includes the MigrationCommmitAAD to the group.

Finally, the Source Delivery Service responds to the client with a MigrationConfirmation message:

```
struct {
  opaque domain_name<V>;
  uint8[32] nonce;
  opaque group_id<V>;
  opaque signature<V>;
} MigrationConfirmation;
```

The Source Delivery Service can now delete the MLS group state and all associated metadata.

Clients can now send messages to the group using the Target Delivery Service.

4. Authorization

In general, whether a client is allowed to migrate an MLS group from one Delivery Service to another is a policy decision that is made by the operator of the Source Delivery Service. The Source Delivery Service MUST NOT allow a client to migrate an MLS group if the client is not authorized to do so.

As a default policy, a Source Delivery Service SHOULD allow any client to migrate an MLS group to another Delivery Service.

Conversely, the Target Delivery Service MUST NOT allow a client to import an MLS group if the client is not authorized to do so. The Target Delivery Service MUST ensure that the client is authorized to import the MLS group before issuing a MigrationResponse and importing the MLS group state through an MigrationContent message.

As a default policy, a Target Delivery Service SHOULD allow any client to migrate an MLS group to it when the client is also allowed to create new groups on the Target Delivery Service.

TODO: The default policies suggested above may be a bit liberal. We might want to restrict them s.t. only clients from the target DS can request migration to that DS.

Authors' Addresses

Konrad Kohbrok
Phoenix R&D
Email: konrad.kohbrok@datashrine.de

Raphael Robert
Phoenix R&D
Email: ietf@raphaelrobert.com