Workgroup: Network Working Group
Internet-Draft: draft-ietf-mls-federation-03
Published: 9 September 2023
Intended Status: Informational
Expires: 12 March 2024
Authors: E. Omara Google
R. Robert Phoenix R&D

The Messaging Layer Security (MLS) Federation

Abstract

This document describes how the Messaging Layer Security (MLS) protocol can be used in a federated environment.

Discussion Venues

This note is to be removed before publishing as an RFC.

Source for this draft and an issue tracker can be found at https://github.com/mlswg/mls-federation.

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 12 March 2024.

Table of Contents

1. Introduction

MLS Architecture draft [I-D.ietf-mls-architecture] describes the overall MLS system architecture, assuming the client and servers (Delivery Service and Authentication Service) are operated by the same entity. This is however not a strict requirement, the MLS protocol does not have an inherent dependency on one single entity and can instead be used between multiple entities.

The focus of this document is on the different components of the MLS architecture when used in a federated environment.

2. Federated environments

Federated environments are environments where multiple entities are operating independent MLS services. In particular, the assumption is that Delivery Services and Authentication Services are not necessarily operated by the same entity.

Entities operating independent MLS services are commonly called domains. In most cases these domains might correspond to the Domain

Name System, but it is not strictly required. Operating MLS services in a federated environment can therefore be regarded as federation between different domains.

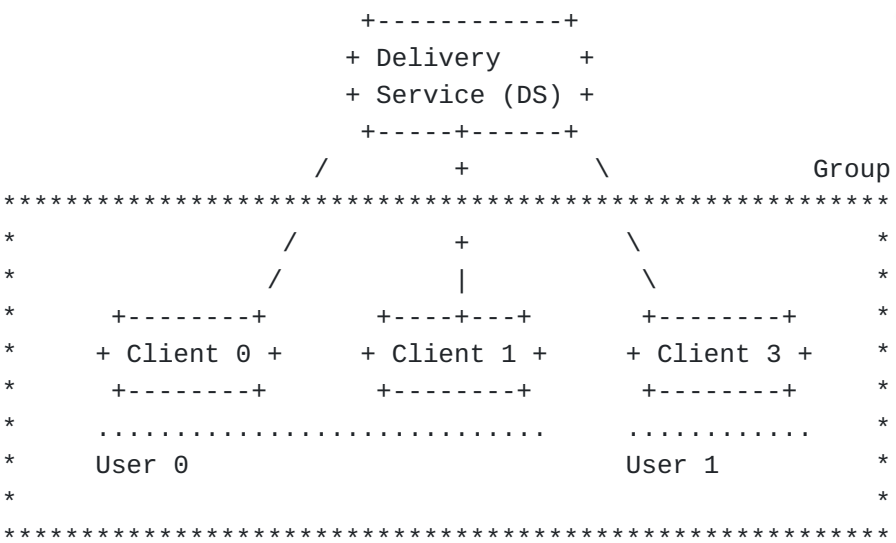
Federation is however not limited to the MLS services themselves. For example, a federated environment could also contain clients that are provided by different entities. Specifically, different vendors could provide different applications that differ in scope and functionality but are interoperable.

2.1. Delivery Services

Depending on the kind of federated environment, the two following types of federated Delivery Services are possible:

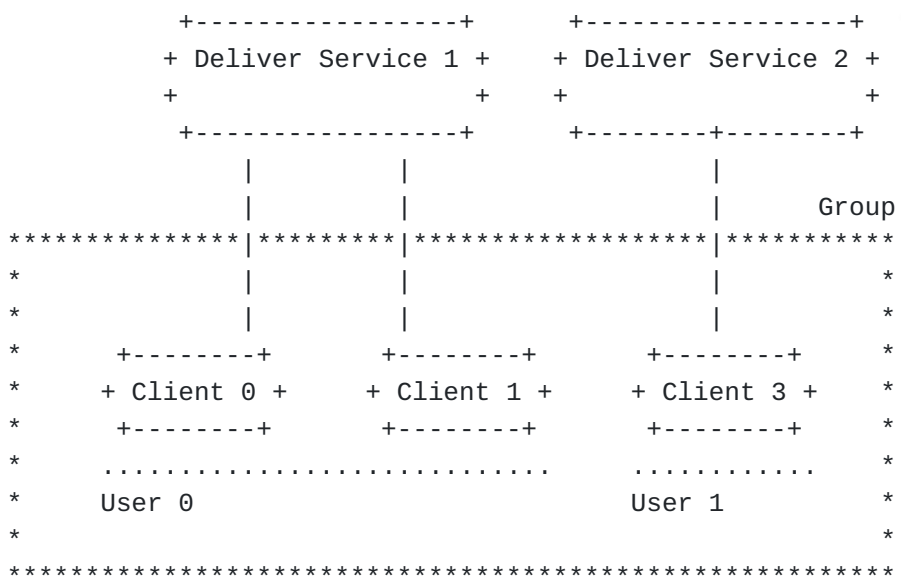
2.1.1. Different client applications

The diagram below shows an MLS group where all clients are provided by potentially different vendors but operate on the same Delivery Service:



2.1.2. Different Delivery Services

The diagram below shows a federated environment in which different or identical clients applications operate on different Delivery Services:



2.2. Authentication Service

In a federated environment, authentication becomes more important. While the specifics of an Authentication Service are out-of-scope for MLS in general, it is important that strong authentication is accessible to all clients of a federated environment. As an example, a shared transparency log like [keytransparency] could be used.

3. Further considerations

The following aspects of federated communication are important for successful federation but are not considered in scope of the MLS charter:

Common format for the content of application messages

The MLS protocol does not impose any format on the content of application messages. Instead, application messages are considered to be an opaque sequence of bytes. Applications in a federated environment are expected to agree on a common format. The negotiation can be done at the KeyPackage level, or through the MLS extension mechanism.

3.1. Network protocol between different domains

Cross-domain operations such as sending and receiving messages, fetching KeyPackages, and querying the Authentication Services have to be part of a common network protocol that is supported by all domains in a federated environment.

4. Discovery service for clients/users on a specific domain

Searching for users and other discovery services are typically part of messaging systems. In the context of MLS, this functionality might overlap with the fetching of KeyPackages and message routing.

5. Use cases

5.1. Different Delivery Servers

Different applications operated by different entities can use MLS to exchange end-to-end encrypted messages. For example, in a messaging applications, clients of messaging1.tld can encrypt and decrypt end-to-end encrypted messages from messaging2.tld.

5.2. Different client applications

Different client applications operating on the same server can use MLS to exchange end-to-end encrypted handshake and application messages. For example, different browsers can implement the MLS protocol, and web developers write web applications that use the MLS implementation in the browser to encrypt and decrypt the messages. This will require a new standard Web API to allow the client applications to set the address of the delivery service in the browser. A more concrete example is using MLS in the browser to negotiate SRTP keys for multi-party conference calls.

6. Functional Requirements

6.1. Delivery service

While there is no strict requirement regarding the network topology, there are practical advantages when clients only connect to their own Delivery Service rather than to the whole federated environment. This routing strategy can possibly make the design of a cross-domain network protocol easier in the context of access control.

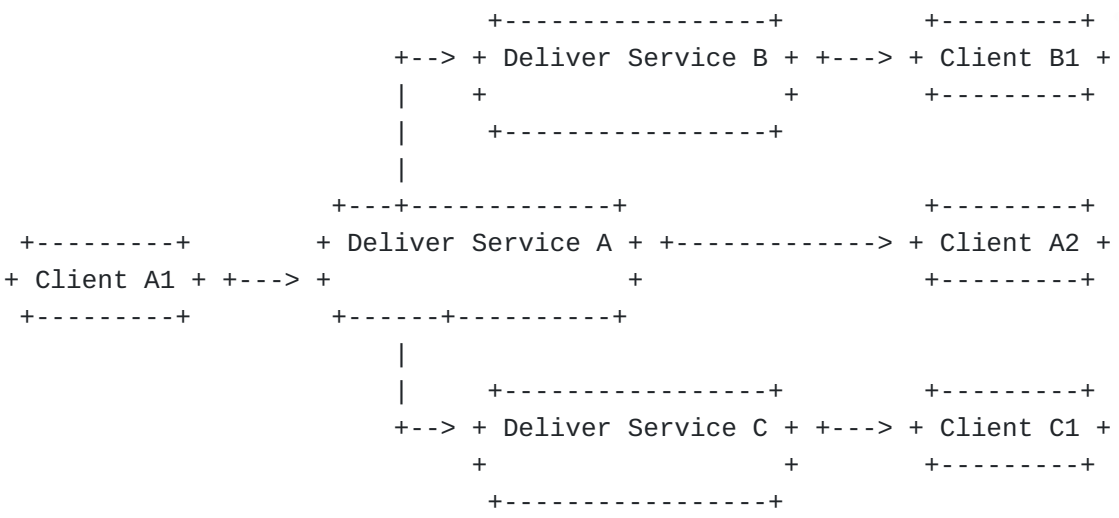
In such a topology, the client's own Delivery Service relays messages to the Delivery Service in charge for a specific group.

When several Delivery Services are involved in relaying messages, it is important that all of them agree on which one is responsible for ordering handshake messages of a specific group at any given time in order to enforce the total ordering of handshake messages required by the MLS protocol.

Depending on the functional requirements (such high availability and redundancy), the different Delivery Services can elect a dedicated Delivery Service to be responsible for ordering handshake messages

for a certain period of time. The election process can be repeated when the availability of Delivery Services change.

The diagram below shows a federated environment where a client connects to its own Delivery Service that in turn relays messages to other Delivery Services.



6.2. Authentication Service

The MLS specification only describes how the signatures on the contents of the leaf nodes of a given group can be verified and that clients have to support the signature schemes of all other clients in each group.

The credential (and thus the binding between identity of the group member and its signature public key) in each (non-blank) leaf node has to be authenticated by the AS. This becomes relevant in a federated setting, as the AS, and thus the authentication process of each member in a given group might differ.

This problem can be solved in a variety of ways, for example, by having all applications and/or service providers involved in a federation agree on a shared process, or by having clients advertise their authentication process in a similar way as their ciphersuite, with the requirement that all members of a group must support each others authentication processes.

Depending on the design of the AS of a given client, other, federated clients might have to trust that client's service provider to authenticate its credential. Confidence in authentication provided by service providers in general can be strengthened by using a scheme such as [keytransparency], which allows both local and federated clients to assert a shared view of the authentication information provided by the service.

In a federated environment, the AS should provide the same degree of transparency as in a non-federated environment, i.e. end-to-end authentication should be possible.

7. Security Considerations

7.1. Version & ciphersuite negotiation

In a federated environment, version & ciphersuite negotiation is more critical, to avoid forcing a downgrade attack by malicious third party Delivery Services. This is due to the fact that the thread model is extended to include the following:

* Different entities might have diverging security policies, e.g. they don't enforce validation of KeyPackages the same way.

* Entities might be malicious and act as a threat actor, e.g. might generate fake clients controlled by them.

The negotiation of version numbers & ciphersuites can be done at the KeyPackage level.

8. IANA Considerations

This document makes no requests of IANA.

9. References

9.1. Normative References

[I-D.ietf-mls-architecture] Beurdouche, B., Rescorla, E., Omara, E., Inguva, S., and A. Duric, "The Messaging Layer Security (MLS) Architecture", Work in Progress, Internet-Draft, draft-ietf-mls-architecture-11, 26 July 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-mls-architecture-11>.

9.2. Informative References

[keytransparency] "Key Transparency", n.d., <https://github.com/google/keytransparency>.

Authors' Addresses

Emad Omara
Google

Email: emadomara@google.com

Raphael Robert

Phoenix R&D

Email: ietf@raphaelrobert.com