Messaging Layer Security
Internet-Draft
Intended status: Informational
Expires: 3 September 2026

R. Mahy

2 March 2026

Including Pending Proposals in External Commits in the Messaging Layer Security protocol
draft-mahy-mls-ext-commit-pp-01

Abstract

The Messaging Layer Security (MLS) protocol allows authorized non-members to join a group via external commits, however it disallows most pending proposals in those commits, which causes unfortunate side effects. This document describes an MLS extension to include pending proposal in external commits when the extension is present in a group.

About This Document

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

The latest revision of this draft can be found at https://rohanmahy.github.io/mls-ext-commit-pp/draft-mahy-mls-ext-commit-pp.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-mahy-mls-ext-commit-pp/.

Discussion of this document takes place on the Messaging Layer Security Working Group mailing list (mailto:mls@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/mls/. Subscribe at https://www.ietf.org/mailman/listinfo/mls/.

Source for this draft and an issue tracker can be found at https://github.com/rohanmahy/mls-ext-commit-pp.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

MLS [RFC9420] allows external commits by authorized clients. The external committer needs a copy of the GroupInfo, either from an existing member or (when supported) from the MLS Distribution Service (DS). The reasoning was that the external committer can't access pending proposals, and that the external committer could not verify them in any case.

The problem is that two important use cases are negatively impacted when external committers join: leaving a group, and external policy enforcement.

This document describes an MLS extension, that when in the required_capabilities in the GroupContext, requires an external joiner to include any pending proposals by reference in its external commit. It assumes that the external joiner can get a suitable set of external proposals from whichever party supplies the GroupInfo.

2. Conventions and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

This document relies on many terms and struct names from [RFC9420].

3. Problem Use Cases

3.1. Leaving a group

MLS clients cannot leave a group without the assistance of another member of the group. A Remove proposal, or SelfRemove proposal (see Section 6.4 of [I-D.ietf-mls-extensions]) needs to be committed before it takes effect, but a client cannot commit its own remove, because a committer knows the epoch secret for the newly created epoch.

Instead, a leaver sends a Remove or SelfRemove proposal and then stays in the group it is trying to leave until it receives a commit showing it has been removed. The client will not be able to determine the epoch secret of the new group, but it will be able validate the tags of the commit. If an external commit is accepted by the DS before a Remove proposal is committed, the leaver needs to enter the new epoch and try to leave again. In practice, this can happen multiple times before the leaver’s proposal is finally committed.

The SelfRemove proposal addresses this problem partially, in that SelfRemove proposals can be included by reference in an external commit, however this does not solve the problem when there are related proposals that should be processed atomically.

When the leaver wants all the clients of the same "user" identity to leave simultaneously, it has a dilema. It can commit Remove proposals for all the user's other clients, then send a SelfRemove proposal immediately in the new epoch. Alternatively, it can send a bundle a proposals for itself and the user's other clients, but risk that only the SelfRemove proposal is committed (at which point the leaver cannot effect any other changes to the group), or that several epochs pass before all the leaving clients are removed.

This problem is exacerbated in the More Instant Message Interoperability (MIMI) protocol ([I-D.ietf-mimi-protocol]) since the removal of another client may not be authorized by itself, but would be otherwise acceptable when a user is removing itself completely.

3.2. External policy enforcer

In many architectures, an external policy enforcer can send external proposals to an MLS group. This could be relatively simple, as in an organization removing a compromised client, or a company removing the clients associated with an employee when she leaves the company.

In more complicated scenarios, such as in MIMI room policy [I-D.ietf-mimi-room-policy], the policy enforcer might change the role of a participant to ban them, to prevent them from sending messages, or to remove their moderator privileges.

If an external commit occurs after the pending external proposals, the commit would usually exclude the pending proposals. The policy enforcer would have to resend the external proposals in the new epoch.

A malicious client could try to send an external commit to rejoin the group immediately upon seeing a proposal to lower its privileges. The DS could refuse the external commit, but the participant would still have legitimate use cases where it may need to rejoin via an external commit.

A pair of malicious clients, A and B, could collude so if B sees an external proposal removing, banning, or reducing privileges of A, B sends an external commit. B could potentially delay the policy action against A by several epochs using this approach.

4. Mechanism

This document defines the external_pending_proposals extension type. When present in the capabilities of a LeafNode it indicates the client's support of the extension. When present in the required_capabilities in the GroupContext, it indicates that all clients MUST include all the pending proposals provided to it by the provider of the GroupInfo.

The external_pending_proposals extension type has no contents:

```
struct {
} ExternalPendingProposalsContents;
```

The provider of the GroupInfo is expected to provide all the pending proposals it has received at the point it provides the GroupInfo.

5. Security Considerations

5.1. Authentication of Proposals by Potential Joiner

If the source of the GroupInfo provides an invalid proposal, an external commit using that GroupInfo and proposal list will be rejected by the other group members. This is effectively a denial of service attack on the client that wants to join the group. However the source of the GroupInfo could more easily not provide the GroupInfo, or provide an invalid one. This problem could be eliminated if the external joiner can authenticate the request. This is already the case for most external proposals, which are typically used for policy enforcement.
[I-D.kohbrok-mls-leaf-operation-intents] describes a mechanism by which leavers can prove their intent to leave and, at the time, their membership in an MLS group.

5.2. Authentication of Proposals by a policy enforcing DS

A policy enforcing DS can validate most proposals and commits. A malicious or faulty client can generate an incorrect tag or an invalid UpdatePath without detection by the MLS DS, regardless of this mechanism. Most of the proposal types in [RFC9420] can be validates by a policy enforcing DS, however a proposal that members could determine was invalid, but a DS could not could lead to exactly the same types of problems already observed with faulty commits.

6. IANA Considerations

This document requests the addition of a new MLS Extension Type under the heading of "Messaging Layer Security".

RFC EDITOR: Please replace XXXX throughout with the RFC number assigned to this document

The external_pending_proposals MLS Extension Type is used inside GroupContext and LeafNode objects. When present it indicates support for this extension.

* Value: 0x0009 (suggested)

* Name: external_pending_proposals

* Message(s): GC, LN : This extension may appear in GroupContext and LeafNode objects

* Recommended: Y

* Reference: RFC XXXX

7. References

7.1. Normative References

[I-D.ietf-mls-extensions]
Robert, R., "The Messaging Layer Security (MLS) Extensions", Work in Progress, Internet-Draft, draft-ietf-mls-extensions-08, 21 July 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-mls-extensions-08>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

[RFC9420] Barnes, R., Beurdouche, B., Robert, R., Millican, J., Omara, E., and K. Cohn-Gordon, "The Messaging Layer Security (MLS) Protocol", RFC 9420, DOI 10.17487/RFC9420, July 2023, <https://www.rfc-editor.org/rfc/rfc9420>.

7.2. Informative References

[I-D.ietf-mimi-protocol]
Barnes, R., Hodgson, M., Kohbrok, K., Mahy, R., Ralston, T., and R. Robert, "More Instant Messaging

Interoperability (MIMI) using HTTPS and MLS", Work in Progress, Internet-Draft, draft-ietf-mimi-protocol-05, 20 October 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-mimi-protocol-05>.

[I-D.ietf-mimi-room-policy]
Mahy, R., "Room Policy for the More Instant Messaging Interoperability (MIMI) Protocol", Work in Progress, Internet-Draft, draft-ietf-mimi-room-policy-03, 18 December 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-mimi-room-policy-03>.

[I-D.kohbrok-mls-leaf-operation-intents]
Kohbrok, K., "Leaf Operation Intents", Work in Progress, Internet-Draft, draft-kohbrok-mls-leaf-operation-intents-01, 20 October 2025, <https://datatracker.ietf.org/doc/html/draft-kohbrok-mls-leaf-operation-intents-01>.

Author’s Address

Rohan Mahy
Email: rohan.ietf@gmail.com