MLS
Internet-Draft
Intended status: Informational
Expires: 24 October 2025

R. Mahy
Rohan Mahy Consulting Services

22 April 2025

Ways to convey the Ratchet Tree in Messaging Layer Security
draft-mahy-mls-ratchet-tree-options-02

Abstract

The Messaging Layer Security (MLS) protocol needs to share its ratchet_tree object to welcome new clients into a group and in external joins. While the protocol only defines a mechanism for sharing the entire tree, most implementations use various optimizations to avoid sending this structure repeatedly in large groups. This document describes a way to convey these improvements in a standardized way and to convey the parts of a GroupInfo object that are not visible to an intermediary server.

About This Document

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

Status information for this document may be found at https://datatracker.ietf.org/doc/draft-mahy-mls-ratchet-tree-options/.

Discussion of this document takes place on the MLS 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-ratchet-tree-options/.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

In the Messaging Layer Security (MLS) protocol [RFC9420], the members of a group are organized into a ratchet tree, the full representation of which is described in the ratchet_tree extension. The protocol specifies that the full ratchet_tree can be included in Welcome messages or shared externally, but describes no concrete way to convey it externally. Likewise, when non-member clients want to join

a group, they can do so using an external commit. They require the GroupInfo and the ratchet_tree.

Many MLS implementations allow external commits to get the GroupInfo from a central server. In the MIMI architecture [I-D.ietf-mimi-arch], this server is called the hub, and for brevity we will use that term generically to refer to any central server that provides either GroupInfo or ratchet_tree objects to new members (i.e. welcomed clients or externally joining clients).

When all handshake messages (commits and proposals) are sent as PublicMessages (or SemiPrivateMessages [I-D.mahy-mls-semiprivatemessage]), the hub can construct its own version of the ratchet_tree and most of the GroupInfo object as proposals and commits arrive.

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 assumes familiarity with terms and structs from the MLS specification ([RFC9420]).

3. Conveying the Ratchet Tree

The ratchet tree can be conveyed inline in its entirety. Alternatively, this document describes how it can be referred to via an HTTPS URI, or signaled that it is communicated out-of-band or reconstructed by the distribution service.

```
enum {
  reserved(0),
  full(1),
  httpsUri(2),
  outOfBand(3),
  distributionService(4),
  (255)
} RatchetTreeRepresentation;

struct {
  RatchetTreeRepresentation representation;
  select (representation) {
    case full:
      Node ratchet_tree<V>;
    case httpsUrl:
      /* an HTTPS URL */
      opaque ratchet_tree_url<V>;
      opaque tree_signature<V>;
    case outOfBand:
      opaque tree_signature<V>;
    case distributionService:
      struct {};
  };
} RatchetTreeOption;
```

* full indicates that the complete ratchet_tree extension is included in the RatchetTreeOption object.

* httpsUri indicates that the ratchet_tree can be downloaded from a URI using the https: scheme.

* outOfBand indicates that the ratchet_tree is communicated or reconstructed via an unspecified out-of-band application protocol.

* distributionService indicates that the ratchet_tree is reconstructed by the Distribution Service from the handshake in the group. This is not possible if any handshake messages are sent as an MLS PrivateMessage.

3.1. Conveying the ratchet tree using HTTPS

This document defines a new MLS GroupContext extension ratchet_tree_source_domains. When present, it contains a list of at least one domain name.

```
struct {
    opaque domain<V>;
} Domain;

struct {
    Domain domains<V>
} DomainList;

DomainList ratchet_tree_source_domains;
```

When this extension is included in the GroupContext of a group, the URL where the ratchet_tree is fetched MUST come from one of the domains in the ratchet_tree_source_domains.domains list.

4. Conveying the GroupInfo

In some systems the GroupInfo is sent to a hub with a full ratchet_tree extension always included with every commit. This is used in systems where the hub may or may not track the membership of the group, but does not keep the entire ratchet_tree data structure. As group size increases, the size of the ratchet_tree extension in the GroupInfo scales roughly linearly. Even using basic credentials, this object gets large quickly. If x509 credentials are used, the size increases much more rapidly, and if a post-quantum ciphersuite (for example [I-D.mahy-mls-xwing]) is used, the size will increase even more rapidly with each new member.

In some systems that require unencrypted handshake messages, the hub tracks commits as they are sent and constructs changes to the ratchet_tree as each handshake is accepted. The hub could also recreate most of the fields of a GroupInfo, with the exception of the GroupInfo signature and the GroupInfo extensions, by inspecting those same unencrypted handshake messages . This document defines a PartialGroupInfo struct that contains these missing fields. PartialGroupInfo can be included with a commit and any referenced proposals to reconstruct a GroupInfo and ratchet_tree from the GroupInfo and ratchet_tree included in the previous epoch.

```
enum {
  no_ratchet_tree(0),
  present(1),
  removed(2),
  added(3),
  (255)
} RatchetTreePresence;

struct {
  RatchetTreePresence ratchet_tree_presence;
  /* GroupInfo extensions excluding ratchet_tree */
  Extension group_info_extensions<V>;
  opaque signature<V>;
} PartialGroupInfo;
```

The value of ratchet_tree_presence is defined as follows:

* no_ratchet_tree: the ratchet_tree extension appears in neither the current nor previous epochs.

* present: there is a ratchet_tree extension in both the current and previous epochs.

* removed: there was a ratchet_tree extension in the previous epoch but none in the current epoch.

* added: there is a ratchet_tree extension in the current epoch but there was none in the previous epoch.

The group_info_extensions object is the list of GroupInfo extensions, omitting any ratchet_tree extension (if present). The only other GroupInfo extension defined in the base protocol is external_pub, the public key of the external commiter. The group_info_extensions is often an empty list.

The signature in the PartialGroupInfo is the signature produced by the committer (represented by its leaf index in the GroupInfo as the signer).

5. Security Considerations

TODO Security

6. IANA Considerations

6.1. ratchet_tree_source_domains MLS Extension Type

This document registers the ratchet_tree_source_domains Extension Type, using the template below:

* Value: TBD1 (new assignment by IANA)

* Name: ratchet_tree_source_domains

* Messages: GC

* Recommended: Y

* Reference: RFC XXXX

7. References

7.1. Normative References

[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-arch] Barnes, R., "An Architecture for More Instant Messaging Interoperability (MIMI)", Work in Progress, Internet-Draft, draft-ietf-mimi-arch-01, 21 November 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-mimi-arch-01>.

[I-D.mahy-mls-semiprivatemessage] Mahy, R., "Semi-Private Messages in the Messaging Layer Security (MLS) Protocol", Work in Progress, Internet-Draft, draft-mahy-mls-semiprivatemessage-05, 21 April 2025, <https://datatracker.ietf.org/doc/html/draft-mahy-mls-semiprivatemessage-05>.

[I-D.mahy-mls-xwing]
Mahy, R., "Messaging Layer Security Ciphersuite using XWing Key Exchange Mechanism", Work in Progress, Internet-Draft, draft-mahy-mls-xwing-00, 4 March 2024, <https://datatracker.ietf.org/doc/html/draft-mahy-mls-xwing-00>.

[I-D.robert-mimi-delivery-service]
Robert, R. and K. Kohbrok, "MIMI Delivery Service", Work in Progress, Internet-Draft, draft-robert-mimi-delivery-service-06, 6 November 2023, <https://datatracker.ietf.org/doc/html/draft-robert-mimi-delivery-service-06>.

Appendix A. Change Log

A.1. Changes between -01 and -02

* Added ratchet_tree_source_domains extension

A.2. Changes between -00 and -01

* Removed ratchet tree patch options and notation.

* Added ratchet_tree_presence options for out-of-band, via HTTPS, and reconstructed by the delivery service.

Acknowledgments

The PartialGroupInfo was first introduced in [I-D.robert-mimi-delivery-service].

Author’s Address

Rohan Mahy
Rohan Mahy Consulting Services
Email: rohan.ietf@gmail.com