Network Working Group
Internet-Draft
Intended status: Informational
Expires: 23 August 2026

K. Kohbrok
R. Robert
Phoenix R&D

19 February 2026

The MLS-TLS secure channel protocol
draft-kohbrok-mls-tls-00

Abstract

This document details how the Messaging Layer Security (MLS) protocol can be combined with the Transport Layer Security (TLS) record layer to yield the MLS-TLS secure channel protocol. In this composed protocol, MLS acts as a continuous key agreement protocol that allows initiator and responder to protect both past and future messages in case of key material compromise. As such, MLS-TLS is suitable for long-lived connections. MLS-TLS also inherits the modularity of MLS and can be configured with post-quantum secure ciphersuites.

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://kkohbrok.github.io/draft-kohbrok-mls-tls/draft-kohbrok-mls-tls.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-kohbrok-mls-tls/.

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

Status of This Memo

Table of Contents

1. Introduction

This document builds on the two-party profile for MLS [I-D.draft-kohbrok-mls-two-party-profile] and combines it with the record layer of TLS 1.3 [RFC8446] to yield a secure channel protocol. The composition of both protocols is seamless due to the modular design of TLS 1.3 and MLS's capability to act as a continuous key agreement protocol. After the initial handshake, initiator and responder can tunnel MLS commits through the secure channel to update their key material. Key material updates allow both parties to achieve forward-secrecy and post-compromise security.

2. MLS as a continuous key agreement protocol

MLS is a secure messaging protocol that can also act as a continuous key agreement protocol. In particular, all parties in an MLS group can not only exchange encrypted messages, but also export shared key material for use outside of MLS.

Within an MLS group, each member has distinct public key material that it can update by sending a _commit_ that updates its own key material and the key material exported from the group. Such an update allows a client to achieve forward secrecy (FS) and post-compromise security (PCS). FS and PCS can be intuitively understood as follows.

FS is the property that prevents an adversary from being able to decrypt messages sent in the past even if it obtains current key material. Group members achieve FS by deleting previously exported key material after an update.

PCS is the property that prevents an adversary from decrypting future messages even if it obtains current key material. Clients can achieve this property by generating new key material and encrypting it to other group members. PCS is achieved when other group members update the public keys of the updating client.

3. Composing MLS and TLS

The MLS-TLS protocol is a two-party protocol that establishes a secure channel between an initiator and a responder. The protocol makes use of MLS's capability to export key material and uses said key material to initialize the record protection layer of the TLS protocol as defined in Section 5 of [RFC8446]. As such MLS essentially replaces the TLS 1.3 handshake.

The specific protocol flow for how initiator and responder establish an MLS group and how the exported keys are subsequently updated is described in [I-D.draft-kohbrok-mls-two-party-profile], which defines a two-party profile for MLS. In particular, the two-party profile describes how updates are coordinated and when exported keys are ready to be used, in this case to initialize or update the TLS record layer.

4. Advantages of MLS

The ability of both parties to achieve PCS by sending MLS commits in-band is increasingly valuable the higher the risk of compromise of either party. As such, the MLS-TLS protocol is specifically suited for scenarios where connections tend to be long-lived. For example, in cases where connection (re)establishment is costly.

MLS is a modular design. When using it as a continuous key agreement protocol, this yields two advantages: Post-quantum capable cryptographic abstractions and arbitrary credential types.

MLS's ciphersuites use Key Encapsulation Mechanisms (KEMs) and signature algorithms as abstractions. This allows the use of a variety of cryptographic algorithms including post-quantum secure ones for both confidentiality and authentication. See [I-D.draft-ietf-mls-pq-ciphersuites] for hybrid and pure post-quantum secure ciphersuites.

MLS does not require clients to use any specific credential type as long as they are provided with signature keys to sign MLS messages. MLS-TLS is thus largely agnostic to how the application binds a client's identity to its signature public key.

5. Protocol Overview

The MLS-TLS protocol consists of three phases.

1. Initial key agreement, where initiator and responder agree on key material using the MLS two party profile. At the end of this phase, both parties are members of an MLS group that is used in the other phases to generate key material.

2. Secure channel phase, where initiator and responder can encrypt data using the TLS 1.3 record layer protocol. The encryption keys are derived from the MLS group created during the initial key agreement phase. During the secure channel phase, initiator and responder can tunnel MLS commits through the secure channel to update their key material.

3. Resumption, where initiator or responder can resume a previously interrupted connection without having to repeat phase 1, including the ability to send data in the first flight of messages.

For more details on the initial key agreement, resumption, as well as how message updates work, see [I-D.draft-kohbrok-mls-two-party-profile].

6. Deriving keys for record layer protection

Both after the initial key agreement phase and the resumption phase, initiator and responder derive key material from the MLS group created during the initial key agreement phase.

The client_application_traffic_secret and the server_application_traffic_secret required by the record layer are derived as follows.

```
server_application_traffic_secret =
  MLS-Exporter("MLS-TLS s ap traffic", [], Length)

client_application_traffic_secret =
  MLS-Exporter("MLS-TLS c ap traffic", [], Length)
```

Where MLS-Exporter is defined in [RFC9420] and Length is the size of the secret required by the TLS record layer.

7. Protecting application data on the wire

Application data is encoded as described in Section 5.1 of [RFC8446] and protected as described in Section 5.2 of [RFC8446]. The keys derived from the MLS group as described in Section Section 6 are used to calculate the traffic keys for record protection as described in Section 7.3 of [RFC8446].

The application traffic secrets MUST be deleted after deriving the traffic keys.

8. Updating record layer protection keys

The MLS two party profile allows both parties to update the MLS group created in the initial key agreement phase. In the context of MLS-TLS, these ConnectionUpdate and EpochKeyUpdate messages (as defined in [I-D.draft-kohbrok-mls-two-party-profile]) are sent in place of key update messages defined in [RFC8446]. Concretely, they are serialized and sent as the content of a TLSInnerPlaintext message with type set to handshake.

TODO: The two-party profile I-D should define a message similar to [RFC9420]’s MLSMessage, which contains either a ClientHello, ServerHello, ConnectionUpdate or EpochKeyUpdate. This is to allow recipients to know what to deserialize when they receive an in-band update message.

When a party is ready to start using the key material as specified in Section 4 of [I-D.draft-kohbrok-mls-two-party-profile], the client derives new application traffic secrets as specified in Section 6. These secrets are then used to further derive the traffic keys for record protection as described in Section 7.3 of [RFC8446].

Key material that is replaced in this way MUST be deleted.

9. Preventing cross-protocol attacks

TODO: Define a component that the initiator needs to include in its leaf node and that MUST be included in the context of the MLS group. This purpose of the component is to clearly mark the individual messages for use in the context of MLS-TLS.

10. Security Considerations

TODO Security

11. IANA Considerations

This document has no IANA actions.

12. Normative References

[I-D.draft-ietf-mls-pq-ciphersuites]
Mahy, R. and R. Barnes, "ML-KEM and Hybrid Cipher Suites for Messaging Layer Security", Work in Progress, Internet-Draft, draft-ietf-mls-pq-ciphersuites-01, 4 November 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-mls-pq-ciphersuites-01>.

[I-D.draft-kohbrok-mls-two-party-profile]
Kohbrok, K. and R. Robert, "A two-party profile for MLS", Work in Progress, Internet-Draft, draft-kohbrok-mls-two-party-profile-00, 3 February 2026, <https://datatracker.ietf.org/doc/html/draft-kohbrok-mls-two-party-profile-00>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/rfc/rfc8446>.

[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>.

Acknowledgments

TODO acknowledge.

Authors' Addresses

Konrad Kohbrok
Phoenix R&D
Email: konrad@ratchet.ing

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