

MLS
Internet-Draft
Intended status: Informational
Expires: 15 December 2025

E. Fondevik
Kongsberg Defense & Aerospace
B. Hale
X. Tian
Naval Postgraduate School
13 June 2025

Paired MLS - PCS in Limited Modes
draft-fondevik-mls-pairedmls-03

Abstract

This document describes the Paired Messaging Layer Security (MLS) extension that improves Post Compromise Security for devices that are unable to self-update using a trusted paired device.

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 15 December 2025.

Copyright Notice

Copyright (c) 2025 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	2
2. About This Document	3
3. Status of this Memo	3
4. Copyright Notice	3
5. Terminology	4
6. Extension Execution	5
6.1. Issuing a Paired Update	7
6.1.1. Termination of Pairing	8
7. Security Considerations	8
7.1. Transport Security	8
7.2. Security of Shared Randomness	9
7.3. Post Compromise Security and Forward Secrecy	9
7.4. Discontinuation of Pairings	9
7.5. Impersonation to the Group	10
7.6. Visability of paired devices to Delivery Service	10
8. Extension Requirements to MLS	11
8.1. Leaf Node Contents	11
8.2. Notifications Between Paired Devices	11
8.3. Multiple Signature Keys per MLS NODE	11
9. IANA Considerations	11
10. References	12
10.1. Normative References (i.e. RFCs)	12
10.2. Informational References	12
Acknowledgments	12
Authors' Addresses	12

1. Introduction

Paired MLS allows a trusted device to update the security parameters of another group member. The trusted paired device can be added to the group or can be another existing group member. The Paired MLS extension builds upon MLS (see [1]). This document presents two operational modes for the Paired MLS extension; interested readers can learn about other cases that were evaluated at [FHX23].

The Paired MLS extension describes a standard case where each device possesses its own signature key. To enable the standard Paired MLS extension, the MLS anchor node must accommodate being shared by at least two devices. If the anchor node is an MLS leaf node, this means that the leaf node would store at least two sets of signature keys. An additional operational mode is described, `_Hidden_ mode`, where the paired devices share a signing key and the paired device is able to issue digital signatures on behalf of the partner device in addition to PCS updates. Caveats to Hidden mode are discussed further under Security Considerations.

2. About This Document

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

Status information for this document may be found at `_[Todo]_`.

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/PairedMLS/draft-pairedMLS>.

3. 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 XX May 2024.

4. Copyright Notice

Copyright (c) 2023 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

5. Terminology

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], and [RFC8174] when, and only when, they appear in all capitals, as shown here.

The terms MLS client, MLS member, MLS group, Leaf Node, GroupContext, KeyPackage, Signature Key, Handshake Message, Private Message, Public Message, and RequiredCapabilities have the same meanings as in the [MLS protocol] <https://www.rfc-editor.org/rfc/rfc9420.html> (<https://www.rfc-editor.org/rfc/rfc9420.html>).

Generally, Paired MLS involves two paired devices, where one device can perform PCS updates in an MLS group on behalf of the other device. Without loss of generality, we define the one performing updates as the active device and the device not performing the update as the passive device:

Passive Device: A passive device is a user equipment device that is not issuing updates for a given period. Such a device may operate in receive-only mode or in another limited fashion such that sending regular keying updates is impractical or even impossible. A passive device may be offline or online, i.e., if the passive device is online it can receive application and group management messages, but is restricted from issuing updates.

Active Device: An active device is another user equipment device designated that is able to issue updates during the given period.

A passive device may be pre-paired with an active device such that the active device can issue updates on behalf of the paired passive device. The active device's updates enable a passive device to PCS heal after a compromise for improved post-compromise security. Paired devices may switch roles between active and passive. Also a device may be paired with multiple others, such that it can issue updates on behalf of several paired passive devices.

Anchor: The MLS node that acts as a shared access node between the paired and passive device is called an anchor. The anchor can be a leaf node of a group member, where the paired devices both have access to the leaf node information but our nominally outside of the MLS tree. The anchor can also be a shared intermediate node on the path to the root of an MLS tree, and as such the the anchor may be shared with multiple only MLS members in addition to the paired devices. Any MLS members that share the anchor node in their parent tree MUST be paired.

Shared Randomness: In order for one device to perform cryptographic updates on behalf of another, they must share a source of randomness that is kept in secure storage. This is in addition to each of the devices' own randomness source. In cryptographic analysis terms, such shared randomness must be stored with similar protections as signature keys in order to not be assumed as compromised in the event of other state compromise. Practically, this is accomplished by sharing a seed for pseudorandom number generation between the two. This random seed IS RECOMMENDED be shared via secure hardware or sharing MAY be bootstrapped over a 1-to-1 channel, where the added MLS PCS guarantees from this draft are contingent on the security of the 1-to-1 channel. Readers are encouraged to see [FHX23] for security tradeoff analysis.

6. Extension Execution

The extension assumes the use of the MLS protocol where the device that desires to execute the extension is already an MLS group member and thus has access to an MLS leaf node. The group member initiating this extension MUST first negotiate the shared randomness with the device it will pair with: this SHOULD be done via secure hardware and MAY be done through an out-of-band, 1-to-1 channel. This extension assumes that the randomness is stored securely, similarly to signature private keys.

Signature key management determines whether the extension is used in standard mode or with hidden mode. In standard mode, both of the paired devices must have their own signing keys, distinct from the anchor. This is the case whether the paired devices are both MLS group members with their distinct leaf nodes, or if the anchor node is an MLS group member leaf node. In the latter case, the extension would require the ability to associate multiple signature public keys to a leaf node. In hidden mode, the paired devices may use the anchor's signing key, thus obfuscating the actions of the individual devices. The private signing key MAY be shared among the paired devices offline or out-of-band.

After sharing randomness and establishing an anchor node, the devices are considered "paired" and either device may update on the other's behalf. When the active device issues an update to the group on behalf of the passive device, it will also issue a _Notify_ notification message to the passive device to ratchet forward its group key using the shared randomness. This ensures that the passive device stays synchronized with the group epoch so it can process updates and commits made by other group members. This notification message is sent in place of a normal update to the paired device, i.e., such that the _Notify_ message does not contain secret keying material. Since, unlike the update message the _Notify_ does not

contain information about the key update, an adversary that has compromised the passive device and tries to decrypt the message learns nothing about the new epoch state, achieving PCS for the passive device. The `_Notify_` notification message is formatted similarly to an update message, such that the distinction between the two is opaque to the DS.

Messages transmitted in the Paired MLS extension are those inherited from MLS [1] with the following changes:

- * If an `*Update*` message is sent by A, such that A is an active device and is sending an update on behalf of B, then A computes the update as in MLS except for the KEM to B. Instead of the KEM for B, A computes the `_Notify_` message. <!--
- * A `*CeasePair*` message is sent from a paired device to its paired devices with whom the initiating device wishes to discontinue paired MLS extension. The command is followed by a self remove then group addition.

Optionally, if randomness between paired devices is transmitted online the following commands are additionally utilized: * A `_ShareRand_` message is sent to negotiate the shared randomness between pairing devices. * A `_InitPairing_` message notifies the recipient about devices that wish to pair. The message contains the identities of the pairing devices and a flag for standard or hidden mode operation. If hidden mode is set, the message MUST be sent directly to the target device in a secure 1-to-1 channel and MUST contain the signature key pair of the anchor leaf node. If standard mode is set, the recipient MUST be the directory, and the directory will associate the public signatures of the requesting devices to the anchor node of the initiating device. * An `_Accept_` message is sent back to the initiator to confirm successful pairing. This means that both devices have shared randomness and that signing keys have been provisioned accordingly. -->

[TODO] Add context on pairing remove messages (if these are allowed to be transparent to the group) MLS commands such as Remove, GroupInfo KeyPackage and Welcome take the form and are processed as according to [1].

6.1. Issuing a Paired Update

Once A and B have been paired, active device A can issue an update on behalf of passive device B. A sends the update to the rest of the MLS group as a normal commit. From the perspective of other MLS group members this update will be indistinguishable from any other MLS update preformed by A. Furthermore, in hidden mode, updates by the paired devices A or B will appear to come from the anchor, due to the shared signing key. A will send the `_Notify_` message to B, where `_Notify_` is indistinguishable to other group members from a commit message to B. The `_Notify_` message signals to B how it should use the shared randomness to derive the necessary update for the new group key in order to stay in sync with the new group epoch.

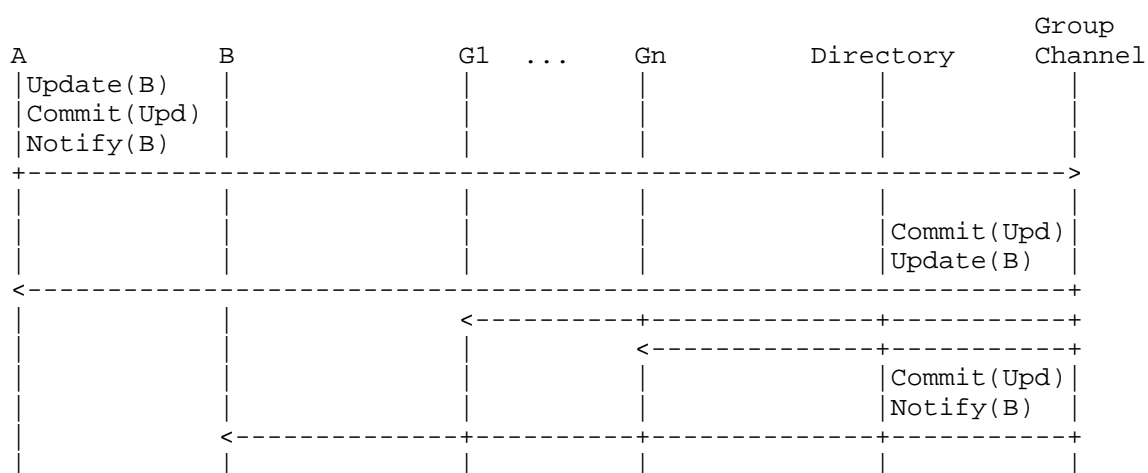


Figure 1 Active device A updates with a commit on behalf of B to the group. The commit message is process as in MLS for all members (A, B, G1, ..., Gn). The Update message is processed as in MLS, but with the change that the update for B is computed as a notify message instead. The notify message is formatted the same as a commit message form the view of the DS. Remaining MLS group members, which are labeled G1, ..., Gn, will receive the standard update messages from the DS.

If any other MLS group member sends proposals or commits to the paired devices the process will follow the flow as defined in RFC9420 [1].

6.1.1. Termination of Pairing

To end Paired MLS extension, either A or B may issue an out-of-band request to its paired device to cease pairing. This request notifies the other device to stop using the shared randomness to update on the other's behalf. In standard mode, pairing termination can be enforced through a self-remove and re-add to the group. In hidden mode, an out-of-band cease pairing request can similarly be issued, but enforcing the termination is more challenging since removing either device is opaque to the MLS given the shared signature key. This will be discussed further under Security Considerations. It is possible, however, to enforce termination of the pairing and hidden mode by removing both devices and re-adding under separate signature keys.

7. Security Considerations

The goal of the MLS extension is to reduce the PCS security risk in cases when group members are unable to update, or updates are seldom. The extension allows other MLS group member devices, or other additional devices belonging to the same user to update on the passive device's behalf. The structure of a shared anchor node in the MLS tree and various devices under that in a subtree can be attractive for practical operational reasons, and the hidden mode could further allow a user to have multiple devices listed under their user identity leaf node; however there are security caveats to exploiting such structures and we will summarize trade-offs here.

7.1. Transport Security

Recommendations for preventing denial of service (DoS) attacks, or restricting transmitted messages are inherited from MLS. Furthermore, message integrity and confidentiality is, as for MLS, protected. <!-- An adversary that can observe network traffic will be able to discern group membership. The MLS packets for the extension are designed to be indistinguishable from regular MLS packets for anyone but the paired devices. As such, a network observer should not be able to determine which devices are paired based solely on packet analysis, however, since paired devices communicate using a separate channel, a network observer might be able to discern general communication from pairing by observing timing and frequency. To prevent the separated communication from leaking information directly, this channel MUST be encrypted. We RECOMMEND using a TLS connection as a minimum.

7.2. Security of Shared Randomness

If the shared randomness between paired devices is leaked then any entity in possession of this information will be able to generate the group session key when either of the devices update on behalf of the other. As such, the shared randomness MUST be stored, securely and encrypted on all applicable end devices when not in use. Furthermore, we strongly RECOMMEND that the random seeds are loaded offline through hardware. If this is not possible a secure, and encrypted channel MUST be utilized to negotiate, or distribute the randomness.

-->

7.3. Post Compromise Security and Forward Secrecy

The main goal of the extension is to reduce epoch sizes when a group member is unable to update. A full security analysis pertaining PCS and FS can be found in [FHX23]. If the extension is not utilized or if paired devices are simultaneously unable to update, FS and PCS security is reduced to that of the original underlying MLS protocol. The PCS benefits from active device updates are contingent on how the shared randomness is stored; if the passive device stores the shared randomness in active memory with other MLS state, then the PCS benefits cannot be assumed. Instead, the shared randomness MUST be stored more securely as with the signature private keys. Furthermore, we strongly RECOMMEND that the random seeds are loaded offline through hardware. If this is not possible, then the out-of-band 1-to-1 channel utilized to negotiate or distribute the randomness is critical to the security benefits; compromise of that negotiation or distribution reduces the PCS guarantees to that of RFC4920 [1].

7.4. Discontinuation of Pairings

[Todo] currently operating under a single paired device. If multiple all need to be removed and then re-added later. Termination of pairing can be signaled as described above; in standard extension mode, if a malicious or unwitting device A ignores the signaling and continues to update on behalf of device B, there is no negative impact on security as B can still issue its own updates using its unique randomness. A can of course disrupt the key schedule if it ignores the signaling to terminate pairing and uses the shared randomness after B deletes it, but this is similar as in MLS [1]. For such reasons, it is RECOMMENDED that B maintain the shared randomness after signaling termination of pairing until confirmation has been received from A. This does not affect forward secrecy.

In hidden mode, where devices share a signature key, termination of pairing requires removal and re-addition of both devices, such that they are registered with separate signature keys. It is not possible to remove only one device, as any removed device will maintain access to the signature private key in the group.

7.5. Impersonation to the Group

In Paired MLS standard mode, distinct signing keys are used by the main device and its paired device when issuing an update. Impersonation of other MLS group members is therefore not feasible given that the signature public keys are known.

In hidden mode, a single signing key is used by all paired devices. This could allow one or more paired devices to be opaque to the MLS group, which inhibits the MLS goals of transparency of group membership but supports possible user side goals of limiting tracking (e.g., if Alice possesses multiple devices that are group members). Thus, devices using the Paired MLS extension in hidden mode MUST be associated with the same group membership user identity, i.e., the paired devices may all belong to Alice but they should not belong to separate users Alice and Bob.

Without the ability to interrogate the delivery service for anonymous hidden pairings, compromised or malicious paired devices may eavesdrop undetected in hidden mode. If a group key is leaked somehow, PCS can be achieved through an update by either of the paired devices. However, if the shared randomness source is compromised on one device, then both devices are irrevocably compromised as the attacker could duplicate generation of the update secrets used on either device. Similarly, the shared signature key in hidden mode means that it is impossible to remove a hidden device member and a hidden member can easily start new groups, impersonating other members; this is similar to signature key compromise in MLS [CHK21]. It is for these reasons that it is required that hidden mode is only used for devices associated with the same MLS user.

7.6. Visability of paired devices to Delivery Service

The detection of the active/passive status of the paired devices to the rest of the group is possible in standard mode, but the detection of pairing is not. Thus other group members may see that device A has updated frequently without knowing that it is on behalf of B. This is because standard mode uses distinct signature keys for each device to issue signed updates to the group.

TODO If there is a consideration that the lack of pairing awareness in the group may result in a devices ejection from the group, it is possible to signal to the group that devices are paired and updates have been performed on behalf of B.

In hidden mode, the DS is still aware of the devices A and B but may not be aware of the pairing status. The anchoring node's signature key is used by both devices, but whether or not they possess shared randomness to perform updates on the behalf of the other is not known to the DS.

8. Extension Requirements to MLS

8.1. Leaf Node Contents

The MLS leaf node will need to support multiple signature keys for the public guardian. The leaf node content is modified by changing `signature_key` to a vector of `SignaturePublicKey`. *<!-- ## Shared Randomness Establishment* The security of this extension is based upon the security of the out-of-band channel to used to establish shared randomness. We RECOMMEND the shared-randomness be installed via protected hardware in the same way that long-term signing keys stored such that it is infeasible to be accessed by an adversary. The shared-randomness MAY be shared via a secure 1-to-1 channel such as a key encapsulation mechanism between the devices.

8.2. Notifications Between Paired Devices

The notification message sent to the passive device to forward ratchet its group key must be secured from forgery and replay attacks. If an attacker were able to to prompt either devices to update, then they would fall out-of-sync and be unable to decrypt future group messages.

8.3. Multiple Signature Keys per MLS NODE

-->

9. IANA Considerations

[TODO] Determine an extension code to use

10. References

[I-D.ietf-mls-protocol] Barnes, R., Beurdouche, B., Robert, R., Millican, J., Omara, E., and K. Cohn-Gordon. "The Messaging Layer Security (MLS) Protocol". Work in Progress, Internet-Draft, draft-ietf-mls-protocol-20, 27 March 2023.
<https://datatracker.ietf.org/doc/html/draft-ietf-mls-protocol-20>
(<https://datatracker.ietf.org/doc/html/draft-ietf-mls-protocol-20>)

10.1. Normative References (i.e. RFCs)

[1] <https://www.rfc-editor.org/info/rfc9420> (<https://www.rfc-editor.org/info/rfc9420>) "MLS RFC" [2] <https://www.rfc-editor.org/info/rfc5246> (<https://www.rfc-editor.org/info/rfc5246>) "TLS RFC"

10.2. Informational References

[FHX23] E. M. Fondevik, B. Hale, and X. Tian. "Guardianship in Group Key Exchange for Limited Environments". Cryptology ePrint Archive, Paper 2023/1761. 11 November 2023.
<https://eprint.iacr.org/2023/1761> (<https://eprint.iacr.org/2023/1761>)

[CHK21] C. Cremers, B. Hale, K. Kohbrok. "The Complexities of Healing in Secure Group Messaging: Why Cross-Group Effects Matter". USENIX Security Symposium 2021: 1847-1864

Acknowledgments

Contributors ## Authors

Authors' Addresses

Elsie Fondevik
Kongsberg Defense & Aerospace
Email: elsie.fondevik@kongsberg.com

Britta Hale
Naval Postgraduate School
Email: britta.hale@nps.edu

Xisen Tian
Naval Postgraduate School
Email: xisen.tian1@nps.edu