

Messaging Layer Security
Internet-Draft
Intended status: Informational
Expires: 23 April 2026

R. Robert
K. Kohbrok
Phoenix R&D
20 October 2025

Single Signature KeyPackages
draft-kohbrok-mls-single-signature-keypackages-00

Abstract

Single Signature KeyPackages improve the overhead of creating, transmitting and verifying MLS KeyPackages by removing one signature.

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-single-signature-keypackages/draft-kohbrok-mls-single-signature-keypackages.html>. Status information for this document may be found at <https://datatracker.ietf.org/doc/draft-kohbrok-mls-single-signature-keypackages/>.

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/kkohbrok/draft-kohbrok-single-signature-keypackages>.

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 23 April 2026.

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. Single Signature KeyPackages	2
3. KeyPackage core hash component	3
4. Security Considerations	4
5. IANA Considerations	4
5.1. Component Type	4
5.2. WireFormat	4
6. Normative References	5
Acknowledgments	5
Authors' Addresses	5

1. Introduction

MLS KeyPackages require two signatures: One over the LeafNode and one over the KeyPackage around it. This draft introduced a LeafNode component that contains a hash over the KeyPackage fields surrounding the LeafNode. As a consequence, verifying the LeafNode also verifies the KeyPackage.

Saving a signature is significant, especially in the context of PQ-secure signature schemes such as ML-DSA, where signatures are multiple orders of magnitude larger than those of most non-PQ signature schemes.

2. Single Signature KeyPackages

A SingleSignatureKeyPackage (SSKP) functions much like a regular KeyPackage with two exceptions: It lacks the signature around the outer KeyPackage and requires a component inside the LeafNode that contains a hash of the KeyPackage around the inner LeafNode.

Since both parsing and processing of an SSKP is different from that of a regular KeyPackage, this document introduces a new WireFormat `mls_single_signature_key_package` and extends the select statement in the definition of `MLSMMessage` in Section 6 of [RFC9420] as follows.

```
struct {  
    ...  
    select (MLSMMessage.wire_format) {  
        ...  
        case mls_single_signature_key_package:  
            SingleSignatureKeyPackage key_package;  
    };  
} MLSMessage;  
  
struct {  
    ProtocolVersion version;  
    CipherSuite cipher_suite;  
    HPKEPublicKey init_key;  
    Extension extensions<V>;  
} KeyPackageCore  
  
struct {  
    KeyPackageCore core;  
    LeafNode leaf_node;  
} SingleSignatureKeyPackage
```

A `SingleSignatureKeyPackage` is created and processed like a regular `KeyPackage` with the following exceptions.

- * The signature around the outer `KeyPackage` is omitted upon creation
- * As there is no signature around the outer `KeyPackage`, verification is skipped during verification
- * The `app_data_dictionary` in the `leaf_node` must contain a valid `KeyPackageCoreHash` as defined in Section 3 under the `component_id` TBD.

The original purpose of the signature over the `KeyPackage` is now served by the signature over the `LeafNode`, which by inclusion of the `KeyPackageCoreHash` provides authenticity for both the `LeafNode` itself and the `KeyPackageCore` around it.

3. KeyPackage core hash component

```
struct {  
    opaque key_package_core_hash;  
} KeyPackageCoreHash
```

The `KeyPackageCoreHashComponent` can be created by hashing the TLS-serialized core of a `SingleSignatureKeyPackage` using the hash function of the `LeafNode`'s ciphersuite.

A `KeyPackageCoreHash` is only valid if two conditions are met.

- * The `leaf_node_source` of the `LeafNode` is `KeyPackage`
- * If the `LeafNode` is verified in the context of a `SingleSignatureKeyPackage`, the `key_package_core_hash` is the hash of the core of that `SingleSignatureKeyPackage`.

4. Security Considerations

Security considerations around `SingleSignatureKeyPackages` are the same as regular `KeyPackages`, except that content of the `KeyPackageCore` should not be trusted until the signature of the `LeafNode` was verified and the `KeyPackageCoreHash` validated.

5. IANA Considerations

5.1. Component Type

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

- * Value: TBD
- * `key_package_core_hash`
- * Where: LN
- * Recommended: Y
- * Reference: TBD

5.2. WireFormat

This document requests the addition of a new `WireFormat` under the heading of "Messaging Layer Security".

The `mls_single_signature_key_package` allows saving the creation and verification of a signature that is necessary when creating a regular `KeyPackage`.

- * Value: TBD
- * Name: `mls_single_signature_key_package`

* Recommended: Y

* Reference: TBD

6. Normative References

[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

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

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