CFRG
Internet-Draft
Intended status: Informational
Expires: 23 April 2026

S. Josefsson
20 October 2025

Mothma: Generic Instantiated PQ/T Hybrid Signatures
draft-josefsson-cfrg-mothma-00

Abstract

This document specify Mothma as a generic family of instantiated Post-Quantum/Traditional (PQ/T) Hybrid Digital Signatures. The goal is to provide a generic hybrid signature pattern that can be analysed separately for security assurance, and to offer concrete instantiated algorithms for integration into protocol and implementations. Identified instances are provided based on combinations of the traditional EdDSA, ECDSA and RSA methods with the post-quantum methods of ML-DSA, SLH-DSA, XMSS and LMS.

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-josefsson-cfrg-mothma/.

Discussion of this document takes place on the Crypto Forum Research Group (CFRG) Research Group mailing list (mailto:cfrg@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/cfrg/.

Source for this draft and an issue tracker can be found at https://gitlab.com/jas/ietf-cfrg-mothma.

Status of This Memo

This Internet-Draft will expire on 23 April 2026.

Table of Contents

1. Introduction

To hedge against attacks on a traditional digital signature methods such as Ed25519 [RFC8032] or a post-quantum digital signature method such as SLH-DSA [NIST.FIPS.205], it is possible to combine both algorithms to define a combined method as a joint signature method. Using the terminology of [RFC9794], this combination forms a PQ/T Hybrid Digital Signature.

Mothma is a generic pattern to create a PQ/T Hybrid Digital Signature methods based on at least one post-quantum algorithm and at least one traditional algorithm. The idea is that Mothma can be analyzed generally and some assurance can be had that it behaves well. For ease of presentation, this document combine one traditional algorithm with one post-quantum algorithm.

While a naive approach would be to integrate a generic Mothma combiner into protocols and have the protocol and implementation negotiate parameters, that leads to complexity detrimental to security. Therefor this document describe specific instances of Mothma applied on selected algorithms.

Mothma is based on the hybrid signature scheme suggested in [DJB-HYBRID-SIGNATURE].

We initially suggest Mothma as combinations of traditional EdDSA, ECDSA and RSA methods with the post-quantum methods of ML-DSA, SLH-DSA, XMSS and LMS. Other combinations may be added following the same generic pattern, and may be proposed for addition to this document.

2. Requirements Language

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.

3. Mothma

Mothma is defined as follows:

The signed message is (s2,s1,r,h,m) where

```
m = the message being signed,
r = H(fresh randomness chosen during signing),
h = H(r,H(hybridpk),hybridsigname,appname,appcontext,m),
s1 = traditional signature of (r,h),
s2 = post-quantum signature of (s1,r,h),
H = SHA3-256.
```

The hash function SHA3-256 is defined in [NIST.FIPS.202].

Using H(hybridpk) instead of hybridpk makes clear that H(hybridpk) can be saved alongside hybridpk, guaranteeing that the key is hashed only once when it’s generated or received.

Here the fresh randomness MUST be 16 bytes, and only to be used for one signature.

The ’hybridsigname’ value is specified by each instantiated variant, whereas the ’appname’ and ’appcontext’ will come from the application using Mothma. These are 0-255 octet long strings, and when text values are put into these fields the encoding is ASCII [RFC0020].

The signed message (s2,s1,r,h,m) is the concatenation of its values.

The signature SHOULD NOT be detached from its corresponding message ’m’ because this leads to fragile implementation, although we recognize that Mothma variants MAY be integrated into existing legacy protocols this way.

Verification is done by confirming that the value of ’h’ and invoking the verify functions for the traditional and post-quantum system using the received values as follows, and taking the logical AND of their verification outputs.

```
Signed message is (s2,s1,r,h,m)
h’ = H(r,H(hybridpk),hybridsigname,appname,appcontext,m),
v1 = Ed25519 verification of s1 on message (r,h),
v2 = ML-DSA-65 verification of s2 on message (s1,r,h),
verify = h == h’ && v1 && v2
```

4. Naming

Protocols wishing to utilize PQ/T Hybrid Signatures described in this document MUST refer to one of the derived instantiated algorithm identifiers and MUST NOT adopt a generic facility where the individual algorithms are parameters.

The convention for identifiers is "Mothma-TSIG-PQSIG" replacing "TSIG" and "PQSIG" with a brief mnemonic identifying the traditional and post-quantum algorithm respectively.

5. EdDSA

EdDSA [RFC8032] is a digital signature system, with variants Ed25519 and Ed448. This protocol always uses the ’pure’ version of EdDSA.

The Ed448 ’context’ input MUST be the Mothma name, e.g., "Mothma-Ed25519-ML-DSA-65".

6. ECDSA

ECDSA [NIST.FIPS.186] is a digital signature system, with variants for the P256, P384 and P521 curves, and the Brainpool curves [RFC5639] brainpoolP256r1, brainpoolP384r1, and brainpoolP512r1. This protocol always uses the ’pure’ version of ECDSA.

7. RSA

RSA [RFC8017] is a digital signature system, with two variants RSASSA-PSS and RSASSA-PKCS1-v1_5. This document do not define any Mothma RSA variants, pending a decision on how to map its arbitrary public key and signature output sizes into Mothma’s fixed-size approach, and pending interest from anyone who desire to use RSA for PQ/T hybrid signatures.

8. ML-DSA

CRYSTALS-Dilithium [PQ-CRYSTALS] is a post-quantum digital signature system, standardized in [NIST.FIPS.204] as Module-Lattice-Based Digital Signature Standard (ML-DSA).

This protocol always uses the ’pure’ version of ML-DSA (where ML-DSA signs the message), and not the ’prehashed’ variant (where ML-DSA signs a previously hashed message). ML-DSA may be used in deterministic or hedged mode.

The ML-DSA ’context’ input MUST be the Mothma name, e.g., "Mothma-Ed25519-ML-DSA-65".

9. SLH-DSA

Sphincs+ [SPHINCS] is a stateless hash-based digital signature system, standardized in [NIST.FIPS.205] as Stateless Hash-Based Digital Signature Algorithm (SLH-DSA).

The SLH-DSA 'context' input MUST be the Mothma name, e.g., "Mothma-Ed25519-ML-DSA-65".

10. XMSS & LMS

XMSS [RFC8391] and LMS [RFC8554] are stateful hash-based digital signature systems, discussed in [NIST.SP.800-208] as Recommendation for Stateful Hash-Based Signature Schemes.

11. Mothma variants

11.1. Mothma-Ed25519-ML-DSA-65

The 'hybridsigname' field is "Mothma-Ed25519-ML-DSA-65".

The ML-DSA-65 signature 's2' is 3309 octets, the Ed25519 signature 's1' is 64 octets, 'r' is 16 octets, 'h' is 32 octets, therefor the signature size is 3421 octets plus the message itself.

11.2. Mothma-Ed25519-SLH-DSA-SHAKE-128S, Mothma-Ed25519-SLH-DSA-SHAKE-128F, Mothma-Ed448-SLH-DSA-SHAKE-256S, Mothma-Ed448-SLH-DSA-SHAKE-256F

The following table describe the mapping from Mothma name to the EdDSA and SLH-DSA variant used.

Mothma variant	EdDSA variant	SLH-DSA variant
Mothma-Ed25519-SLH-DSA-SHAKE-128S	Ed25519	SLH-DSA-SHAKE-128S
Mothma-Ed25519-SLH-DSA-SHAKE-128F	Ed25519	SLH-DSA-SHAKE-128F
Mothma-Ed448-SLH-DSA-SHAKE-256S	Ed448	SLH-DSA-SHAKE-256S
Mothma-Ed448-SLH-DSA-SHAKE-256F	Ed448	SLH-DSA-SHAKE-256F

Table 1: Mothma EdDSA/SLH-DSA mappings

The 'hybridsigname' field to use is as follows.

Mothma variant	hybridsigname
Mothma-Ed25519-SLH-DSA-SHAKE-128S	"Mothma-Ed25519-SLH-DSA-SHAKE-128S"
Mothma-Ed25519-SLH-DSA-SHAKE-256F	"Mothma-Ed25519-SLH-DSA-SHAKE-128F"
Mothma-Ed448-SLH-DSA-SHAKE-256S	"Mothma-Ed448-SLH-DSA-SHAKE-256S"
Mothma-Ed448-SLH-DSA-SHAKE-256F	"Mothma-Ed448-SLH-DSA-SHAKE-256F"

Table 2: Value for hybridsigname

11.3. Mothma-ECDSA-P256-ML-DSA-44, Mothma-ECDSA-P384-ML-DSA-65, Mothma-ECDSA-P521-ML-DSA-87

11.4. Mothma-ECDSA-brainpoolP256r1-ML-DSA-44, Mothma-ECDSA-brainpoolP384r1-ML-DSA-65, Mothma-ECDSA-brainpoolP521-ML-DSA-87

11.5. Mothma-Ed25519-XMSS-SHA2_10_256, Mothma-Ed448-XMSS-SHA2_20_256

11.6. Mothma-Ed25519-LMS_SHA256_M32_H5, Mothma-Ed448-LMS_SHA256_M32_H25

12. Acknowledgments

The method was suggested by Daniel J. Bernstein. This document re-use ideas and some text from [CHEMPAT].

13. IANA Considerations

None.

14. Security Considerations

The security considerations of all references apply.

The intention is that Mothma hybrid signatures should be secure if at least one of the traditional and post-quantum algorithms is secure.

Cryptographic algorithms and parameters are usually broken or weakened over time. Implementers and users need to continously re-evaluate that cryptographic algorithms continue to provide the expected level of security.

15. References

15.1. Normative References

[NIST.FIPS.186]
NIST, "Federal Information Processing Standards Publication: digital signature standard (DSS)", NIST Federal Information Processing Standards Publications 186, DOI 10.6028/NIST.FIPS.186, 1994, <https://nvlpubs.nist.gov/nistpubs/Legacy/FIPS/fipspub186.pdf>.

[NIST.FIPS.202]
Dworkin, M., Dworkin, M. J., and NIST, "SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions", FIPS PUB 202, NIST Federal Information Processing Standards Publications 202, DOI 10.6028/nist.fips.202, DOI 10.6028/NIST.FIPS.202, August 2015, <http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf>.

[NIST.FIPS.204]
"*** BROKEN REFERENCE ***".

[NIST.FIPS.205]
"*** BROKEN REFERENCE ***".

[RFC0020] Cerf, V., "ASCII format for network interchange", STD 80, RFC 20, DOI 10.17487/RFC0020, October 1969, <https://www.rfc-editor.org/info/rfc20>.

[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/info/rfc2119>.

[RFC5639] Lochter, M. and J. Merkle, "Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation", RFC 5639, DOI 10.17487/RFC5639, March 2010, <https://www.rfc-editor.org/info/rfc5639>.

[RFC8017] Moriarty, K., Ed., Kaliski, B., Jonsson, J., and A. Rusch, "PKCS #1: RSA Cryptography Specifications Version 2.2", RFC 8017, DOI 10.17487/RFC8017, November 2016, <https://www.rfc-editor.org/info/rfc8017>.

[RFC8032] Josefsson, S. and I. Liusvaara, "Edwards-Curve Digital Signature Algorithm (EdDSA)", RFC 8032, DOI 10.17487/RFC8032, January 2017, <https://www.rfc-editor.org/info/rfc8032>.

[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/info/rfc8174>.

[RFC8391] Huelsing, A., Butin, D., Gazdag, S., Rijneveld, J., and A. Mohaisen, "XMSS: eXtended Merkle Signature Scheme", RFC 8391, DOI 10.17487/RFC8391, May 2018, <https://www.rfc-editor.org/info/rfc8391>.

[RFC8554] McGrew, D., Curcio, M., and S. Fluhrer, "Leighton-Micali Hash-Based Signatures", RFC 8554, DOI 10.17487/RFC8554, April 2019, <https://www.rfc-editor.org/info/rfc8554>.

15.2. Informative References

[CHEMPAT] Josefsson, S., "Chempat: Generic Instantiated PQ/T Hybrid Key Encapsulation Mechanisms", Work in Progress, Internet-Draft, draft-josefsson-chempat-03, 18 March 2025, <https://datatracker.ietf.org/doc/html/draft-josefsson-chempat-03>.

[DJB-HYBRID-SIGNATURE] Bernstein, D., "How to construct a hybrid signature combiner?", March 2024, <https://mailarchive.ietf.org/arch/msg/cfrg/LdvasJBpseekZtQkQF1nuPPDH_s/>.

[NIST.SP.800-208] Cooper, D. A., Apon, D. C., Dang, Q. H., Davidson, M. S., Dworkin, M. J., Miller, C. A., and NIST, "Recommendation for Stateful Hash-Based Signature Schemes", NIST Special Publications (General) 800-208, DOI 10.6028/NIST.SP.800-208, 29 October 2020, <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-208.pdf>.

[PQ-CRYSTALS] CRYSTALS Team, "Cryptographic Suite for Algebraic Lattices (CRYSTALS)", December 2017, <https://pq-crystals.org/>.

[RFC9794] Driscoll, F., Parsons, M., and B. Hale, "Terminology for Post-Quantum Traditional Hybrid Schemes", RFC 9794, DOI 10.17487/RFC9794, June 2025, <https://www.rfc-editor.org/info/rfc9794>.

[SPHINCS] SPHINCS+ Team, "SPHINCS+", November 2017, <https://sphincs.org/>.

Author’s Address

Simon Josefsson
Email: simon@josefsson.org