| | |
|---|---|
| Workgroup: | CFRG |
| Internet-Draft: | draft-josefsson-chempat-01 |
| Published: | 14 April 2024 |
| Intended Status: | Informational |
| Expires: | 16 October 2024 |
| Author: | S. Josefsson |


# Chempat: Generic Instantiated PQ/T Hybrid Key Encapsulation Mechanisms


## Abstract

This document specify Chempat as a generic family of instantiated Post-Quantum/Traditional (PQ/T) Hybrid Key Exchange Methods (KEMs). The goal is to provide a generic combiner construct 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 traditional Diffie-Hellman key agreement using curves P-256, P-384, X25519, X448, brainpoolP256, brainpoolP384 combined with post quantum methods ML-KEM-768, ML-KEM-1024, Streamlined NTRU Prime sntrup761, and Classic McEliece.


The RFC Editor will remove this note

## 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-chempat/.

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/. Subscribe at https://www.ietf.org/mailman/listinfo/cfrg/.

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

## 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 16 October 2024.

## Copyright Notice



## Table of Contents

# 1. Introduction

To hedge against attacks on a traditional key agreement algorithm such as X25519 [RFC7748] and a post-quantum key encapsulation mechanism (KEM) such as ML-KEM-768 [MLKEM], it is possible to combine both algorithms to derive a shared secret [GHP18] and define the combination mechanism as a new KEM. Using the terminology of [I-D.driscoll-pqt-hybrid-terminology], this combination forms a PQ/T Hybrid Key Encapsulation Mechanism.

Chempat is a generic pattern to create a PQ/T Hybrid Key Encapsulation Mechanism based on at least one post-quantum algorithm and at least one traditional algorithm. The idea is that the Chempat combiner can be analyzed generally and some assurance can be had that it behaves well. For ease of presentation, this document combine one traditional DH-Based KEM algorithm with one post-quantum KEM algorithm.

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

# 2. Motivation

There are many choices that can be made when specifying a hybrid KEM: the constituent KEMs; their security levels; the combiner; and the hash within, to name but a few. Having too many similar options are a burden to the ecosystem.

The above argues for having carefully selected instantiated hybrid KEMs. Each hybrid KEM should be analysed to meet security targets. If that analysis assume specific behaviour of the combiner, or if the analysis become more complex due to the combiner, that leads to more work to re-use the analysis for other combinations. While it would be preferrable to only specify one hybrid KEM and analyse that, such as [XWING], cryptographic history suggests that algorithm preferences varies over time.

The argument then is to establish a generic method that can be analysed independent of its component algorithms, such as [KEMCOMBINER]. Generic methods can lead to parametrized protocols and implementations that is more difficult to analyse, and a lack of instantiated algorithm identifiers.

While non-hybrid approaches may eventually be preferrable, there are doubts on what properties protocols demand from cryptographic primitives, and some of the properties are different from what have been expected from traditional algorithms [CDM23]. This suggests that some post-quantum KEM's should be used together with a other algorithms to strengthen the properties.

Finally this leads up to our approach to describe a generic method that can be analysed independently of the individual components, with as few parameters as possible in the generic combiner, and to instantiate it with common algorithm choices that make sense for protocols and implementations. That is the essence of Chempat.

## 3. Comparison to X-Wing

X-Wing [XWING] is a Hybrid PQ/T KEM based on X25519 and ML-KEM-768. Main differences:

- Chempat is applicable to other algorithm combinations, X-Wing's combiner does not extend securely to other KEM combinations.
- Chempat on X25519 with ML-KEM-768 will hash the ML-KEM ciphertext and public key.
- Chempat on X25519 with ML-KEM-768 can provide a per-protocol key-domain separation context string.

## 4. Comparison to HPKE X25519Kyber768Draft00

HPKE's X25519Kyber768Draft00 [XYBERHPKE] is similar to X-Wing. Main differences to Chempat:

- Chempat is applicable to other algorithm combinations, X25519Kyber768Draft00's combiner does not extend securely to other KEM combinations.
- Chempat hashes the shared secret, to be usable outside of HPKE.
- Chempat hashes the combined ciphertext and public keys.

There is also a different KEM called X25519Kyber768Draft00 [XYBERTLS] which is used in TLS. This one should not be used outside of TLS, as it assumes the presence of the TLS transcript to ensure non malleability.

## 5. Comparison to KEM Generic Combiner

Chempat is most similar to the generic combiner in [KEMCOMBINER]. Main differences:

- Chempat offers instantiated identified Hybrid KEMs for direct use in protocols and implementations.
- Chempat offers the possibility of a generic simpler security argument for the combiner, whereas [KEMCOMBINER] is parametrized with several algorithm choices and any security analysis needs to be parametrized over the numerous options permitted.
- Chempat has a fixed 32 byte shared secret instead of a variable length shared secret.
- Chempat hashes the public keys of the component KEM's.

# 6. Design Goals

While Chempat share a lot with [XWING], [XYBERHPKE] and [KEMCOMBINER] the following goals set it apart:

- Allow generic security analysis independent of combinations.
- Provide concrete instantiated algorithm identifiers for several anticipated uses of Hybrid KEM combinations.

We aim for instantiated algorithms of Chempat to be usable for most applications, including specifically HPKE [RFC9180], TLS [RFC8446], OpenPGP [RFC4880] and SSH [RFC4251].

# 7. 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.

The following terms are used throughout this document:

string - array of bytes

func1(), func2(a,b) - denote functions called FUNC1 and FUNC2 that takes no parameters and two parameters a and b, respectively.

concat(x0, ..., xN): returns the concatenation of byte strings. concat(0x01, 0x0203, 0x040506) = 0x010203040506.

random(n): return a pseudorandom byte string of length n bytes produced by a cryptographically-secure random number generator.

# 8. Chempat

Chempat is defined as follows:

```
H = SHA3-256

hybrid_pk = concat(receiver_pk_TKEM, receiver_pk_PQKEM)

hybrid_ct = concat(sender_ct_TKEM, sender_ct_PQKEM)

hybrid_ss = H(concat(ss_TKEM,
                     ss_PQKEM,
                     H(hybrid_ct),
                     H(hybrid_pk),
                     context))
```

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

The hybrid_pk string is the concatenation of the serialized public-key output from the traditional (receiver_pk_TEM) and post-quantum (receiver_pk_PQKEM) respectively. To reduce memory usage it is possible to hash the public keys to pre-compute H(hybrid_pk) directly when hybrid_pk is received.

The hybrid_ct string is the concatenation of the serialized ciphertext output from the traditional (receiver_ct_TEM) and post-quantum (receiver_ct_PQKEM) respectively. To reduce memory usage it is possible to hash the ciphertext to pre-compute H(hybrid_ct) directly when hybrid_ct is received.

The hybrid_ss string is the 32-byte output shared secret, formed as the output of the SHA3-256 hash function. The inputs to the hash function is a concatenation of the shared secrets from the traditional (ss_TKEM) and post-quantum (ss_PQKEM) KEMs with the hashes of the ciphertexts (H(hybrid_ct)) and public keys (H(hybrid_pk)) together with a variable-length protocol-specific context string.

The context string can be chosen uniquely by the protocol referencing this document. The purpose is to provide protocol domain separation of the generated keys. The content is arbitrary, and in practice the name of the protocol will suffice. Since this results in a new Chempat instance, to reduce combinatorical complexity of parameters, we provide one instance with the context variable set to the name of the Chempat instance, for example "Chempat-X25519-sntrup761".

# 9. Naming

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

The convention for identifiers is "Chempat-TKEM-PQKEM" replacing "TKEM" and "PQKEM" with a brief mnemonic identifying the traditional and post-quantum algorithm respectively.

# 10. Use in HPKE

Each Chempat instance satisfy the HPKE KEM interface as follows.

The SerializePublicKey, DeserializePublicKey, SerializePrivateKey and DeserializePrivateKey are concatenation and splitting of the known-length component strings.

```
H = SHA3-256

def GenerateKeyPair():
  (pk_T, sk_T) = DHKEM.KeyGen()
  (pk_PQ, sk_PT) = PQKEM.KeyGen()
  return (concat(sk_T, sk_PQ, pk_T, pk_PQ), concat(pk_T, pk_PQ))

# TBA DeriveKeyPair

def Chempat(ss_T, ss_PQ, ct_T, ct_PQ, pk_T, pk_PQ):
  return H(concat(ss_T,
                  ss_PQ,
                  H(concat(ct_T, ct_PQ)),
                  H(concat(pk_T, pk_PQ)),
                  Context))

def Encapsulate(pk):
  pk_T = pk[0:DHKEM.Npk]
  pk_PQ = pk[DHKEM.Npk:PQKEM.Npk-DHKEM.Npk]
  (ss_T, ct_T) = DHKEM.Encap(pk_T)
  (ss_PQ, ct_PQ) = PQKEM.Encap(pk_PQ)
  ss = Chempat(ss_T, ss_PQ, ct_T, ct_PQ, pk_T, pk_PQ)
  ct = concat(ct_T, ct_PQ)
  return (ss, ct)

def Decapsulate(ct, sk):
  ct_T = ct[0:DHKEM.Nenc]
  ct_T = ct[DHKEM.Nenc:PQKEM.Nenc-DHKEM.Nenc]
  sk_PQ = sk[0:DHKEM.Nsecret]
  sk_T = sk[DHKEM.Nsecret:PQKEM.Nsecret-DHKEM.Nsecret]
  pk_T = sk[0:DHKEM.Npk]
  pk_PQ = sk[DHKEM.Npk:PQKEM.Npk-DHKEM.Npk]
  ss_T = DHKEM.Decap(ct_T, sk_T)
  ss_PQ = PQKEM.Decap(ct_PQ, sk_PQ)
  return Chempat(ss_T, ss_PQ, ct_T, ct_PQ, pk_T, pk_PQ)
```

Chempat does not provide authenticeted KEMs and does not support AuthEncap() or AuthDecap() of [RFC9180].

Context is a string provided by the protocol referencing this document, or if not provided corresponds to the name of the Chempat instance, such as "Chempat-X25519-sntrup761".

Nsecret is 32 for all Chempat instances, and Nenc, Npk, and Nsk depends on the underlying components.

# 11. Chempat-X25519-sntrup761

This algorithm is instantiated using the TKEM as DHKEM(X25519, HKDF-SHA256) from [RFC9180] and PQKEM as a HPKE variant of sntrup761 from [NTRUPrimePQCS] [NTRUPrime].

The DHKEM.Nsecret, DHKEM.Nenc, DHKEM.Npk, DHKEM.Nsk are all 32 for X25519 per Section 7.1 of [RFC9180].

The PQKEM.Nsecret is 32, PQKEM.Nenc is 1039, PQKEM.Npk is 1158 and PQKEM.Nsk is 1763 for sntrup761 per [NTRUPrimePQCS].

Thus Nenc is 1071, Npk is 1190 and Nsk is 1795 for Chempat-X25519-sntrup761.

# 12. Chempat with Classic McEliece with X448 and X25519

This is a set of mechanisms implemented the same way but with different component algorithms and parameter lengths.

This algorithm is instantiated using the TKEM as DHKEM(X, HKDF-SHA512) from [RFC9180] and PQKEM as a HPKE variant of M from [MCELIECE] [CM-spec], substituting X and M for the particular algorithm from the tables below. Sizes for DHKEM for X25519 and X448 as per Section 7.1 of [RFC9180], and sizes for PQKEM as per [CM-spec].

The f and non-f versions are interoperable. The f versions have faster key generation, while the non-f versions have simpler key generation. For example, a key generated with mceliece6688128f can decapsulate ciphertexts that were encapsulated with mceliece6688128, and vice versa. The secret-key sizes (and formats) are the same, the encapsulation functions are the same, and the decapsulation functions are the same. Implementations of this protocol can chose between f and non-f variants, however the name of the hybrid will use the non-f names.

| DHKEM variant | Nsecret | Nenc | Npk | Nsk |
|---|---|---|---|---|
| X25519 | 32 | 32 | 32 | 32 |
| X448 | 64 | 56 | 56 | 56 |

Table 1: X25519/X448 DHKEM size

| PQKEM variant | Nsecret | Nenc | Npk | Nsk |
|---|---|---|---|---|
| mceliece348864 | 32 | 96 | 261120 | 6492 |
| mceliece460896 | 32 | 156 | 524160 | 13608 |
| mceliece6688128 | 32 | 208 | 1044992 | 13932 |
| mceliece6960119 | 32 | 194 | 1047319 | 13948 |
| mceliece8192128 | 32 | 208 | 1357824 | 14120 |

Table 2: Classic McEliece sizes

Names and sizes of the Chempat hybrids are per table below.

| Variant | Nenc | Npk | Nsk |
|---|---|---|---|
| Chempat-X25519-mceliece348864 | 128 | 261152 | 6524 |
| Chempat-X25519-mceliece460896 | 188 | 524192 | 13640 |

| Variant | Nenc | Npk | Nsk |
|---|---|---|---|
| Chempat-X25519-mceliece6688128 | 240 | 1045024 | 13964 |
| Chempat-X25519-mceliece6960119 | 226 | 1047351 | 13980 |
| Chempat-X25519-mceliece8192128 | 240 | 1357856 | 14152 |
| Chempat-X448-mceliece348864 | 160 | 261176 | 6548 |
| Chempat-X448-mceliece460896 | 220 | 524216 | 13664 |
| Chempat-X448-mceliece6688128 | 272 | 1045048 | 13988 |
| Chempat-X448-mceliece6960119 | 258 | 1047375 | 14004 |
| Chempat-X448-mceliece8192128 | 272 | 1357880 | 14176 |

Table 3: Classic McEliece with X25519/X448

# 13. Chempat-X25519-ML-KEM-768

This algorithm is instantiated using the TKEM as DHKEM(X25519, HKDF-SHA256) from [RFC9180] and PQKEM as a HPKE variant of ML-KEM-768 from [MLKEM].

Protocols and implementation **MAY** consider [XWING] instead of Chempat-X25519-ML-KEM-768, and the definition of Chempat-X25519-ML-KEM-768 is here for situations when some property of X-Wing is not wanted. Informally and non-conclusively, X-Wing offers better performance and Chempat-X25519-ML-KEM-768 offers re-use of the generic security claims on Chempat and a per-protocol key-separation context string.

The DHKEM.Nsecret, DHKEM.Nenc, DHKEM.Npk, DHKEM.Nsk are all 32 for X25519 per Section 7.1 of [RFC9180].

The PQKEM.Nsecret is 32, PQKEM.Nenc is 1088, PQKEM.Npk is 1184 and PQKEM.Nsk is 2400 for ML-KEM-768 per [MLKEM].

Thus Nenc is 1120, Npk is 1216 and Nsk is 2432 for Chempat-X25519-ML-KEM-768.

# 14. Chempat-X448-ML-KEM-1024

This algorithm is instantiated using the TKEM as DHKEM(X448, HKDF-SHA512) from [RFC9180] and PQKEM as a HPKE variant of ML-KEM-1024 from [MLKEM].

For X448 DHKEM.Nsecret is 64, DHKEM.Nenc is 56, DHKEM.Npk is 56, DHKEM.Nsk is 56 per Section 7.1 of [RFC9180].

The PQKEM.Nsecret is 32, PQKEM.Nenc is 864, PQKEM.Npk is 1568 and PQKEM.Nsk is 2400 for ML-KEM-1024 per [MLKEM].

Thus Nenc is 1120, Npk is 1624 and Nsk is 2456 for Chempat-X25519-ML-KEM-1024.

# 15. Chempat-P256-ML-KEM-768

This algorithm is instantiated using the TKEM as DHKEM(P-256, HKDF-SHA256) from [RFC9180] and PQKEM as a HPKE variant of ML-KEM-768 from [MLKEM].

For P256 DHKEM.Nsecret is 32, DHKEM.Nenc is 65, DHKEM.Npk is 65, DHKEM.Nsk is 32 per Section 7.1 of [RFC9180].

The PQKEM.Nsecret is 32, PQKEM.Nenc is 1088, PQKEM.Npk is 1184 and PQKEM.Nsk is 2400 for ML-KEM-768 per [MLKEM].

Thus Nenc is 1153, Npk is 1249 and Nsk is 2432 for Chempat-P256-ML-KEM-768.

# 16. Chempat-P384-ML-KEM-1024

This algorithm is instantiated using the TKEM as DHKEM(P-384, HKDF-SHA384) from [RFC9180] and PQKEM as a HPKE variant of ML-KEM-1024 from [MLKEM].

For P384 DHKEM.Nsecret is 48, DHKEM.Nenc is 97, DHKEM.Npk is 97, DHKEM.Nsk is 48 per Section 7.1 of [RFC9180].

The PQKEM.Nsecret is 32, PQKEM.Nenc is 864, PQKEM.Npk is 1568 and PQKEM.Nsk is 2400 for ML-KEM-1024 per [MLKEM].

Thus Nenc is 961, Npk is 1665 and Nsk is 2448 for Chempat-P384-ML-KEM-1024.

# 17. Chempat-brainpoolP256-ML-KEM-768

This algorithm is instantiated using the TKEM as DHKEM(brainpoolP256, HKDF-SHA256) from [RFC9180] [RFC5639] and PQKEM as a HPKE variant of ML-KEM-768 from [MLKEM].

For brainpoolP256 DHKEM.Nsecret is 32, DHKEM.Nenc is 65, DHKEM.Npk is 65, DHKEM.Nsk is 32.

The PQKEM.Nsecret is 32, PQKEM.Nenc is 1088, PQKEM.Npk is 1184 and PQKEM.Nsk is 2400 for ML-KEM-768 per [MLKEM].

Thus Nenc is 1153, Npk is 1249 and Nsk is 2432 for Chempat-brainpoolP256-ML-KEM-768.

# 18. Chempat-brainpoolP384-ML-KEM-1024

This algorithm is instantiated using the TKEM as DHKEM(brainpoolP384, HKDF-SHA384) from [RFC9180] [RFC5639] and PQKEM as a HPKE variant of ML-KEM-1024 from [MLKEM].

For brainpoolP384 DHKEM.Nsecret is 48, DHKEM.Nenc is 97, DHKEM.Npk is 97, DHKEM.Nsk is 48. The PQKEM.Nsecret is 32, PQKEM.Nenc is 864, PQKEM.Npk is 1568 and PQKEM.Nsk is 2400 for ML-KEM-1024 per [MLKEM].

Thus Nenc is 961, Npk is 1665 and Nsk is 2448 for Chempat-brainpoolP384-ML-KEM-1024.

# 19. Security Considerations

Chempat is intended to be secure if SHA3 is secure and either the traditional algorithm is secure or the post-quantum algorithm is secure.

The security considerations of each component algorithm are inherited.

Cryptographic algorithms and parameters will be broken or weakened over time. Blindly implementing supported groups listed here is not advised. Implementers and users need to check that the cryptographic algorithms listed continue to provide the expected level of security.

# 20. IANA Considerations

Protocols that provide a Context variable will need to register their own key-domain separate identifiers. The registrations below are when Chempat instances are used with their default value of Context.

This document requests/registers new entries to the "HPKE KEM Identifiers" registry as follows.

| Value | KEM | Nsecret | Nenc | Npk | Nsk | Auth | Reference |
|---|---|---|---|---|---|---|---|
| TBD | Chempat-X25519-sntrup761 | 32 | 1071 | 1190 | 1795 | No | THISRFC |
| TBD | Chempat-X25519-mceliece348864 | 32 | 128 | 261152 | 6524 | No | THISRFC |
| TBD | Chempat-X25519-mceliece460896 | 32 | 188 | 524192 | 13640 | No | THISRFC |
| TBD | Chempat-X25519-mceliece6688128 | 32 | 240 | 1045024 | 13964 | No | THISRFC |
| TBD | Chempat-X25519-mceliece6960119 | 32 | 226 | 1047351 | 13980 | No | THISRFC |
| TBD | Chempat-X25519-mceliece8192128 | 32 | 240 | 1357856 | 14152 | No | THISRFC |
| TBD | Chempat-X448-mceliece348864 | 32 | 160 | 261176 | 6548 | No | THISRFC |

| Value | KEM | Nsecret | Nenc | Npk | Nsk | Auth | Reference |
|---|---|---|---|---|---|---|---|
| TBD | Chempat-X448-mceliece460896 | 32 | 220 | 524216 | 13664 | No | THISRFC |
| TBD | Chempat-X448-mceliece6688128 | 32 | 272 | 1045048 | 13988 | No | THISRFC |
| TBD | Chempat-X448-mceliece6960119 | 32 | 258 | 1047375 | 14004 | No | THISRFC |
| TBD | Chempat-X448-mceliece8192128 | 32 | 272 | 1357880 | 14176 | No | THISRFC |
| TBD | Chempat-X25519-ML-KEM-768 | 32 | 1120 | 1216 | 2432 | No | THISRFC |
| TBD | Chempat-X448-ML-KEM-1024 | 32 | 1120 | 1624 | 2456 | No | THISRFC |
| TBD | Chempat-P256-ML-KEM-768 | 32 | 1153 | 1249 | 2432 | No | THISRFC |
| TBD | Chempat-P384-ML-KEM-1024 | 32 | 961 | 1665 | 2448 | No | THISRFC |
| TBD | Chempat-brainpoolP256-ML-KEM-768 | 32 | 1153 | 1249 | 2432 | No | THISRFC |
| TBD | Chempat-brainpoolP384-ML-KEM-1024 | 32 | 961 | 1665 | 2448 | No | THISRFC |

Table 4: Chempat HPKE KEM Identifiers

This document requests/registers a new entry to the TLS Supported Group registry as follows.

| Value | Description | DTLS-OK | Recommended | Reference | Comment |
|---|---|---|---|---|---|
| TBD | Chempat-X25519-sntrup761 | Y | Y | THISRFC | PQ/T hybrid of X25519 and sntrup761 |

Table 5: Chempat TLS Supported Groups

## 21. Acknowledgments

The combiner function was suggested by Daniel J. Bernstein. The document re-use ideas and some text from [XWING], [KEMCOMBINER], [XYBERHPKE] and [RFC9180].

# 22. References

## 22.1. 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>.

## 22.2. Informative References

**[CDM23]** Cremers, C., Dax, A., and N. Medinger, "Keeping Up with the KEMs: Stronger Security Notions for KEMs and automated analysis of KEM-based protocols", 2023, <https://eprint.iacr.org/2023/1933>.

**[CM-spec]** Classic McEliece Team, "Classic McEliece: conservative code-based cryptography: cryptosystem specification", October 2022, <https://classic.mceliece.org/mceliece-spec-20221023.pdf>.

**[GHP18]** Giacon, F., Heuer, F., and B. Poettering, "KEM Combiners", 2018, <https://doi.org/10.1007/978-3-319-76578-5_7>.

**[I-D.driscoll-pqt-hybrid-terminology]** D, F., "Terminology for Post-Quantum Traditional Hybrid Schemes", Work in Progress, Internet-Draft, draft-driscoll-pqt-hybrid-terminology-02, 7 March 2023, <https://datatracker.ietf.org/doc/html/draft-driscoll-pqt-hybrid-terminology-02>.

**[KEMCOMBINER]** Ounsworth, M., Wussler, A., and S. Kousidis, "Combiner function for hybrid key encapsulation mechanisms (Hybrid KEMs)", Work in Progress, Internet-Draft, draft-ounsworth-cfrg-kem-combiners-05, 31 January 2024, <https://datatracker.ietf.org/doc/html/draft-ounsworth-cfrg-kem-combiners-05>.

**[MCELIECE]** Josefsson, S., "Classic McEliece", Work in Progress, Internet-Draft, draft-josefsson-mceliece-01, 14 April 2024, <https://datatracker.ietf.org/doc/html/draft-josefsson-mceliece-01>.

**[MLKEM]** National Institute of Standards and Technology, "FIPS 203 (Initial Draft): Module-Lattice-Based Key-Encapsulation Mechanism Standard", n.d., <https://csrc.nist.gov/pubs/fips/203/ipd>.

**[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>.

**[NTRUPrime]** Bernstein, D. J., Chuengsatiansup, C., Lange, T., and C. van Vredendaal, "NTRU Prime: reducing attack surface at low cost", August 2017, <https://ntruprime.cr.yp.to/ntruprime-20170816.pdf>.

**[NTRUPrimePQCS]** Daniel J Bernstein, Billy Bob Brumley, Ming-Shing Chen, Chitchanok Chuengsatiansup, Tanja Lange, Adrian Marotzke, Bo-Yuan Peng, Nicola Tuveri, Christine van Vredendaal, and Bo-Yin Yang, "NTRU Prime: round 3, Submission to the NIST PQC Standardization Round 3 Process", October 2020, <https://csrc.nist.gov/CSRC/media/Projects/post-quantum-cryptography/documents/round-3/submissions/NTRU-Prime-Round3.zip>.

**[RFC4251]** Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH) Protocol Architecture", RFC 4251, DOI 10.17487/RFC4251, January 2006, <https://www.rfc-editor.org/rfc/rfc4251>.

**[RFC4880]** Callas, J., Donnerhacke, L., Finney, H., Shaw, D., and R. Thayer, "OpenPGP Message Format", RFC 4880, DOI 10.17487/RFC4880, November 2007, <https://www.rfc-editor.org/rfc/rfc4880>.

**[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/rfc/rfc5639>.

**[RFC7748]** Langley, A., Hamburg, M., and S. Turner, "Elliptic Curves for Security", RFC 7748, DOI 10.17487/RFC7748, January 2016, <https://www.rfc-editor.org/rfc/rfc7748>.

**[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>.

**[RFC9180]** Barnes, R., Bhargavan, K., Lipp, B., and C. Wood, "Hybrid Public Key Encryption", RFC 9180, DOI 10.17487/RFC9180, February 2022, <https://www.rfc-editor.org/rfc/rfc9180>.

**[XWING]** Connolly, D., Schwabe, P., and B. Westerbaan, "X-Wing: general-purpose hybrid post-quantum KEM", Work in Progress, Internet-Draft, draft-connolly-cfrg-xwing-kem-02, 26 March 2024, <https://datatracker.ietf.org/doc/html/draft-connolly-cfrg-xwing-kem-02>.

**[XYBERHPKE]** Westerbaan, B. and C. A. Wood, "X25519Kyber768Draft00 hybrid post-quantum KEM for HPKE", Work in Progress, Internet-Draft, draft-westerbaan-cfrg-hpke-xyber768d00-02, 4 May 2023, <https://datatracker.ietf.org/doc/html/draft-westerbaan-cfrg-hpke-xyber768d00-02>.

**[XYBERTLS]** Westerbaan, B. and D. Stebila, "X25519Kyber768Draft00 hybrid post-quantum key agreement", Work in Progress, Internet-Draft, draft-tls-westerbaan-xyber768d00-03, 24 September 2023, <https://datatracker.ietf.org/doc/html/draft-tls-westerbaan-xyber768d00-03>.

## Author's Address

**Simon Josefsson**
Email: simon@josefsson.org