JOSE
Internet-Draft
Intended status: Standards Track
Expires: 12 June 2026

T. Reddy
A. Banerjee
Nokia
H. Tschofenig
H-BRS

9 December 2025

Post-Quantum Key Encapsulation Mechanisms (PQ KEMs) for JOSE and COSE
draft-ietf-jose-pqc-kem-05

Abstract

This document describes the conventions for using Post-Quantum Key Encapsulation Mechanisms (PQ-KEMs) within JOSE and COSE.

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-ietf-jose-pqc/.

Discussion of this document takes place on the jose Working Group mailing list (mailto:jose@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/cose/. Subscribe at https://www.ietf.org/mailman/listinfo/jose/.

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 12 June 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

Quantum computing is no longer perceived as a consequence of computational sciences and theoretical physics. Considerable research efforts and enormous corporate and government funding for the development of practical quantum computing systems are being invested currently. As such, as quantum technology advances, there is the potential for future quantum computers to have a significant impact on current cryptographic systems.

Researchers have developed Post-Quantum Key Encapsulation Mechanisms (PQ-KEMs) to provide secure key establishment resistant against an adversary with access to a quantum computer.

As the National Institute of Standards and Technology (NIST) is still in the process of selecting the new post-quantum cryptographic algorithms that are secure against both quantum and classical computers, the purpose of this document is to propose a PQ-KEMs to protect the confidentiality of content encrypted using JOSE and COSE against the quantum threat.

Although this mechanism could thus be used with any PQ-KEM, this document focuses on Module-Lattice-based Key Encapsulation Mechanisms (ML-KEMs). ML-KEM is a one-pass (store-and-forward) cryptographic mechanism for an originator to securely send keying material to a recipient using the recipient’s ML-KEM public key. Three parameters sets for ML-KEMs are specified by [FIPS203]. In order of increasing security strength (and decreasing performance), these parameter sets are ML-KEM-512, ML-KEM-768, and ML-KEM-1024.

2. 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.

This document makes use of the terms defined in [I-D.ietf-pquip-pqt-hybrid-terminology]. The following terms are repeately used in this specification:

* KEM: Key Encapsulation Mechanism

* PQ-KEM: Post-Quantum Key Encapsulation Mechanism

* CEK: Content Encryption Key

* ML-KEM: Module-Lattice-based Key Encapsulation Mechanism

For the purposes of this document, it is helpful to be able to divide cryptographic algorithms into two classes:

"Traditional Algorithm": An asymmetric cryptographic algorithm based on integer factorisation, finite field discrete logarithms or elliptic curve discrete logarithms. In the context of JOSE, examples of traditional key exchange algorithms include Elliptic Curve Diffie-Hellman Ephemeral Static [RFC6090] [RFC8037]. In the context of COSE, examples of traditional key exchange algorithms include Ephemeral-Static (ES) DH and Static-Static (SS) DH [RFC9052].

"Post-Quantum Algorithm": An asymmetric cryptographic algorithm that is believed to be secure against attacks using quantum computers as well as classical computers. Post-quantum algorithms can also be called quantum-resistant or quantum-safe algorithms. Examples of Post-Quantum Algorithm include ML-KEM.

2.1. Key Encapsulation Mechanisms

For the purposes of this document, we consider a Key Encapsulation Mechanism (KEM) to be any asymmetric cryptographic scheme comprised of algorithms satisfying the following interfaces [PQCAPI].

* def kemKeyGen() -> (pk, sk)

* def kemEncaps(pk) -> (ct, ss)

* def kemDecaps(ct, sk) -> ss

where pk is public key, sk is secret key, ct is the ciphertext representing an encapsulated key, and ss is shared secret.

KEMs are typically used in cases where two parties, hereby refereed to as the "encapsulater" and the "decapsulater", wish to establish a shared secret via public key cryptography, where the decapsulater has an asymmetric key pair and has previously shared the public key with the encapsulater.

3. Design Rationales

Section 4.6 of the JSON Web Algorithms (JWA) specification, see [RFC7518], defines two ways of using a key agreement:

* When Direct Key Agreement is employed, the shared secret established through the Traditional Algorithm will be the content encryption key (CEK).

* When Key Agreement with Key Wrapping is employed, the shared secret established through the Traditional Algorithm will wrap the CEK.

For efficient use with multiple recipients, the key wrap approach is used since the content can be encrypted once with the CEK, while each recipient receives an individually encrypted CEK. Similarly, Section 8.5.4 and Section 8.5.5 of COSE [RFC9052] define the Direct Key Agreement and Key Agreement with Key Wrap, respectively. This document proposes the use of PQ-KEMs for these two modes.

It is essential to note that in the PQ-KEM, one needs to apply Fujisaki-Okamoto [FO] transform or its variant [HHK] on the PQC KEM part to ensure that the overall scheme is IND-CCA2 secure, as mentioned in [I-D.ietf-tls-hybrid-design]. The FO transform is performed using the KDF such that the PQC KEM shared secret achieved is IND-CCA2 secure. As a consequence, one can re-use PQC KEM public keys but there is an upper bound that must be adhered to.

Note that during the transition from traditional to post-quantum algorithms, there may be a desire or a requirement for protocols that incorporate both types of algorithms until the post-quantum algorithms are fully trusted. HPKE [RFC9180] is a KEM that can be extended to support hybrid post-quantum KEMs and the specification for the use of PQ/T Hybrid Key Encapsulation Mechanism (KEM) in Hybrid Public-Key Encryption (HPKE) for integration with JOSE and COSE is described in [I-D.reddy-cose-jose-pqc-hybrid-hpke].

4. KEM PQC Algorithms

At time of writing, NIST have standardized three PQC algorithms, with more expected to be standardised in the future ([NISTFINAL]). These algorithms are not necessarily drop-in replacements for traditional asymmetric cryptographic algorithms. For instance, RSA [RSA] and ECC [RFC6090] can be used as both a key encapsulation method (KEM) and as a signature scheme, whereas there is currently no post-quantum algorithm that can perform both functions.

4.1. ML-KEM

ML-KEM offers several parameter sets with varying levels of security and performance trade-offs. This document specifies the use of the ML-KEM algorithm at three security levels: ML-KEM-512, ML-KEM-768, and ML-KEM-1024. ML-KEM key generation, encapsulation and decaspulation functions are defined in [FIPS203]. The main security property for KEMs standardized in the NIST Post-Quantum Cryptography Standardization Project is indistinguishability under adaptive chosen ciphertext attacks (IND-CCA2) (see Section 10.2 of

[I-D.ietf-pquip-pqc-engineers]). The public/private key sizes, ciphertext key size, and PQ security levels of ML-KEM are detailed in Section 12 of [I-D.ietf-pquip-pqc-engineers].

4.2. PQ-KEM Encapsulation

The encapsulation process is as follows:

1. Generate an inital shared secret SS’ and the associated ciphertext CT using the KEM encapsulation function and the recipient’s public key recipPubKey.

   (SS’, CT) = kemEncaps(recipPubKey)

1. Derive a final shared secret SS of length SSLen bytes from the initial shared secret SS’ using the underlying key derivation function:

   SS = KDF(SS’, SSLen)

In Direct Key Agreement mode, the output of the KDF MUST be a key of the same length as that used by encryption algorithm. In Key Agreement with Key Wrapping mode, the output of the KDF MUST be a key of the length needed for the specified key wrap algorithm.

When Direct Key Agreement is employed, SS is the CEK. When Key Agreement with Key Wrapping is employed, SS is used to wrap the CEK.

4.3. PQ-KEM Decapsulation

The decapsulation process is as follows:

1. Decapsulate the ciphertext CT using the KEM decapsulation function and the recipient’s private key to retrieve the initial shared secret SS’:

   SS’ = kemDecaps(recipPrivKey, CT)

If the decapsulation operation outputs an error, output "decryption error", and stop.

1. Derive the final shared secret SS of length SSLen bytes from the inital secret SS’ using the underlying key derivation function:

   SS = KDF(SS’, SSLen)

5. KDF

5.1. Key Derivation for JOSE

The key derivation for JOSE is performed using the KMAC defined in NIST SP 800-108r1-upd1 [SP-800-108r1]. The KMAC(K, X, L, S) parameters are instantiated as follows:

* K: the input key-derivation key. In this document this is the initial shared secret (SS') outputted from the kemEncaps() or kemDecaps() functions.

* X: The context-specific data used for key derivation includes the concatenation of AlgorithmID, SuppPubInfo, and SuppPrivInfo, as defined in [NIST.SP.800-56Ar3]. The fields AlgorithmID and SuppPubInfo are defined in Section 4.6.2 of [RFC7518] The fields PartyUInfo and PartyVInfo, also defined in that section, are intentionally excluded. PartyUInfo is omitted because post-quantum KEMs do not support sender authentication. PartyVInfo is excluded because the recipient' s identity is already bound to the public key used for encapsulation, making its inclusion unnecessary. If mutually known private information is required, both parties MUST agree out-of-band to include it as SuppPrivInfo.

* L: length of the output key in bits and it would be set to match the length of the key required for the AEAD operation.

* S: the optional customization label. In this document this parameter is unused, that is it is the zero-length string "".

For all security levels of ML-KEM, KMAC256 is used.

5.2. Key Derivation for COSE

The key derivation for COSE is performed using the KMAC defined in NIST SP 800-108r1-upd1 [SP-800-108r1]. The KMAC(K, X, L, S) parameters are instantiated as follows:

* K: the input key-derivation key. In this document this is the initial shared secret (SS') outputted from the kemEncaps() or kemDecaps() functions.

* X: The context structure defined in Section 5.2 of [RFC9053] excluding PartyUInfo and PartyVInfo fields. PartyUInfo is omitted because sender authentication is not available in PQ KEMs. PartyVInfo is excluded because the recipient's identity is already bound to the public key used for encapsulation, making its inclusion redundant. If mutually known private information is to

be included, both the sender and the recipient MUST agree out-of-band to include it as SuppPrivInfo in the key derivation function, as defined in [NIST.SP.800-56Ar3].

* L: length of the output key in bits and it would be set to match the length of the key required for the AEAD operation.

* S: the optional customization label. In this document this parameter is unused, that is it is the zero-length string "".

For all security levels of ML-KEM, KMAC256 is used.

6. Post-quantum KEM in JOSE

As explained in Section 3 JWA defines two ways to use public key cryptography with JWE:

* Direct Key Agreement

* Key Agreement with Key Wrapping

This specification describes these two modes of use for PQ-KEM in JWE. Unless otherwise stated, no changes to the procedures described in [RFC7516] have been made.

6.1. Direct Key Agreement

* The "alg" header parameter MUST be a PQ-KEM algorithm chosen from the JSON Web Signature and Encryption Algorithms registry defined in [JOSE-IANA].

* The CEK will be generated using the process explained in Section 4.2. The output of the Section 4.2 MUST be a secret key of the same length as that used by the "enc" algorithm.

* The usage for the "alg" and "enc" header parameters remain the same as in JWE [RFC7516]. Subsequently, the plaintext will be encrypted using the CEK, as detailed in Step 15 of Section 5.1 of [RFC7516].

* The header parameter encapsulated key "ek" defined in [I-D.ietf-jose-hpke-encrypt] MUST include the output (’ct’) from the PQ-KEM algorithm, encoded using base64url.

* The recipient MUST base64url decode the ciphertext from the "ek" header parameter and then use it to derive the CEK using the process defined in Section 4.3.

* The JWE Encrypted Key MUST be absent.

Note that when using Direct Key Agreement in JOSE Compact Serialization, inefficiency arises due to double encoding of the KEM ciphertext. In this mode, the "epk" parameter inside the protected header carries the KEM ciphertext, already base64url-encoded. Then, the entire protected header is base64url-encoded again as part of the compact serialization.

6.2. Key Agreement with Key Wrapping

* The derived key is generated using the process explained in Section 4.2 and used to encrypt the CEK.

* The parameter "ek" MUST include the output (’ct’) from the PQ-KEM algorithm, encoded using base64url.

* The JWE Encrypted Key MUST include the base64url-encoded encrypted CEK.

* The ’enc’ (Encryption Algorithm) header parameter MUST specify a content encryption algorithm from the JSON Web Signature and Encryption Algorithms registry, as defined in [JOSE-IANA].

* The recipient MUST base64url decode the ciphertext from "ek". Subsequently, it is used to derive the key, through the process defined in Section 4.3. The derived key will then be used to decrypt the CEK.

7. Post-Quantum KEM in COSE

This specification supports two uses of PQ-KEM in COSE, namely

* PQ-KEM in a Direct Key Agreement mode.

* PQ-KEM in a Key Agreement with Key Wrap mode.

In both modes, the COSE header parameter ’ek’ defined in Section 7.2 of [I-D.ietf-cose-hpke], is used to convey the output (’ct’) from the PQ KEM Encaps algorithm.

7.1. Direct Key Agreement

The CEK will be generated using the process explained in Section 4.2. Subsequently, the plaintext will be encrypted using the CEK. The resulting ciphertext is either included in the COSE_Encrypt or is detached. If a payload is transported separately then it is called "detached content". A nil CBOR object is placed in the location of the ciphertext. See Section 5 of [RFC9052] for a description of detached payloads.

The COSE_Recipient structure for the recipient is organized as follows:

* The sender MUST set the ’alg’ parameter to indicate the use of the PQ-KEM algorithm.

* This documents RECOMMENDS the use of the ’kid’ parameter (or other parameters) to explicitly identify the recipient public key used by the sender. If the COSE_Encrypt contains the ’kid’ then the recipient may use it to select the appropriate private key.

7.2. Key Agreement with Key Wrap

With the two layer structure the PQ-KEM information is conveyed in the COSE_recipient structure, i.e. one COSE_recipient structure per recipient.

In this approach the following layers are involved:

* Layer 0 (corresponding to the COSE_Encrypt structure) contains the content (plaintext) encrypted with the CEK. This ciphertext may be detached, and if not detached, then it is included in the COSE_Encrypt structure.

* Layer 1 (corresponding to a recipient structure) contains parameters needed for PQ-KEM to generate a shared secret used to encrypt the CEK. This layer conveys the encrypted CEK in the "ciphertext" field (Section 5.1 of [RFC9052]). The unprotected header MAY contain the kid parameter to identify the static recipient public key the sender has been using with PQ-KEM.

This two-layer structure is used to encrypt content that can also be shared with multiple parties at the expense of a single additional encryption operation. As stated above, the specification uses a CEK to encrypt the content at layer 0.

8. JOSE Ciphersuite Registration

This specification registers a number of PQ-KEM algorithms for use with JOSE.

All security levels of ML-KEM internally utilize SHA3-256, SHA3-512, SHAKE128, and SHAKE256. This internal usage influences the selection of the KDF as described in this document.

ML-KEM-512 MUST be used with a KDF capable of outputting a key with at least 128 bits of security and with a key wrapping algorithm with a key length of at least 128 bits.

ML-KEM-768 MUST be used with a KDF capable of outputting a key with at least 192 bits of security and with a key wrapping algorithm with a key length of at least 192 bits.

ML-KEM-1024 MUST be used with a KDF capable of outputting a key with at least 256 bits of security and with a key wrapping algorithm with a key length of at least 256 bits.

* In Direct key agreement, the parameter "alg" MUST be specified, and its value MUST be one of the values specified in Figure 1. (Note that future specifications MAY extend the list of algorithms.)

alg	Description
ML-KEM-512	ML-KEM-512
ML-KEM-768	ML-KEM-768
ML-KEM-1024	ML-KEM-1024

Figure 1: Direct Key Agreement: Algorithms.

* In Key Agreement with Key Wrapping, the parameter "alg" MUST be specified, and its value MUST be one of the values specified in the table Figure 2.

alg	Description
ML-KEM-512+A128KW	ML-KEM-512 + AES128KW
ML-KEM-768+A192KW	ML-KEM-768 + AES192KW
ML-KEM-1024+A256KW	ML-KEM-1024 + AES256KW

Figure 2: Key Agreement with Key Wrapping: Algorithms.

9. COSE Ciphersuite Registration

Figure 3 maps the JOSE algorithm names to the COSE algorithm values (for the PQ-KEM ciphersuites defined by this document).

JOSE	COSE ID	Description	Recommended
ML-KEM-512	TBD1	ML-KEM-512	No
ML-KEM-768	TBD2	ML-KEM-768	No
ML-KEM-1024	TBD3	ML-KEM-1024	No
ML-KEM-512+A128KW	TBD4	ML-KEM-512 + AES128KW	No
ML-KEM-768+A192KW	TBD5	ML-KEM-768 + AES192KW	No
ML-KEM-1024+A256KW	TBD6	ML-KEM-1024 + AES256KW	No

Figure 3: Mapping between JOSE and COSE PQ-KEM Ciphersuites.

10. Use of AKP Key Type for PQC KEM Keys in JOSE and COSE

The "AKP" (Algorithm Key Pair) key type, defined in [I-D.ietf-cose-dilithium] is used in this specification to represent PQC KEM keys for JOSE and COSE. When used with JOSE or COSE algorithms that rely on PQC KEMs, a key with "kty" set to "AKP" represents an PQC KEM key pair. The public key is carried in the "pub" parameter. If included, the private key is carried in the "priv" parameter. When expressed as a JWK, the "pub" and "priv" values are base64url-encoded.

The "AKP" key type mandates inclusion of the "alg" parameter, and applying AKP to PQC KEMs requires distinguishing between keys used for Direct Key Agreement and those used for Key Agreement with Key Wrap, which is in line with [NIST.SP.800-57pt1r5] guidance. Note that the NIST guidance refers to using a key for a single purpose, and both KEM and KEM+KW fall under the same overall purpose of key establishment. In this draft, they are treated as distinct algorithm usages to ensure clear operational separation.

For ML-KEM algorithms, as specified in [FIPS203], there are two possible representations of a private key: a seed and a fully expanded private key derived from the seed. This document specifies the use of only the seed form for private keys. To promote interoperability, this specification mandates that the "priv" parameter MUST contain the 32-byte seed used to generate the ML-KEM key pair. It does not support the expanded private key representation defined by NIST. This approach ensures consistency with other PQC algorithms used in JOSE/COSE, and avoids ambiguity.

11. Security Considerations

PQC KEMs used in the manner described in this document MUST explicitly be designed to be secure in the event that the public key is reused, such as achieving IND-CCA2 security. ML-KEM has such security properties.

12. IANA Considerations

12.1. JOSE

The following entries are added to the "JSON Web Signature and Encryption Algorithms" registry:

* Algorithm Name: ML-KEM-512
* Algorithm Description: PQ-KEM that uses ML-KEM-512 PQ-KEM.
* Algorithm Usage Location(s): "alg"
* JOSE Implementation Requirements: Optional
* Change Controller: IESG
* Specification Document(s): [[TBD: This RFC]]
* Algorithm Analysis Documents(s): TODO
* Algorithm Name: ML-KEM-768

* Algorithm Description: PQ-KEM that uses ML-KEM-768 PQ-KEM.
* Algorithm Usage Location(s): "alg"
* JOSE Implementation Requirements: Optional
* Change Controller: IESG
* Specification Document(s): [[TBD: This RFC]]
* Algorithm Analysis Documents(s): TODO
* Algorithm Name: ML-KEM-1024
* Algorithm Description: PQ-KEM that uses ML-KEM-1024 PQ-KEM.
* Algorithm Usage Location(s): "alg"
* JOSE Implementation Requirements: Optional
* Change Controller: IESG
* Specification Document(s): [[TBD: This RFC]]
* Algorithm Analysis Documents(s): TODO
* Algorithm Name: ML-KEM-512+A128KW
* Algorithm Description: PQ-KEM that uses ML-KEM-512 PQ-KEM and CEK wrapped with "A128KW".
* Algorithm Usage Location(s): "alg"
* JOSE Implementation Requirements: Optional
* Change Controller: IESG
* Specification Document(s): [[TBD: This RFC]]
* Algorithm Analysis Documents(s): TODO
* Algorithm Name: ML-KEM-768+A192KW
* Algorithm Description: PQ-KEM that uses ML-KEM-768 and CEK wrapped with "A192KW".
* Algorithm Usage Location(s): "alg"

* JOSE Implementation Requirements: Optional
* Change Controller: IESG
* Specification Document(s): [[TBD: This RFC]]
* Algorithm Analysis Documents(s): TODO
* Algorithm Name: ML-KEM-1024+A256KW
* Algorithm Description: PQ-KEM that uses ML-KEM-1024 and CEK wrapped with "A256KW".
* Algorithm Usage Location(s): "alg"
* JOSE Implementation Requirements: Optional
* Change Controller: IESG
* Specification Document(s): [[TBD: This RFC]]
* Algorithm Analysis Documents(s): TODO

12.2. COSE

The following has to be added to the "COSE Algorithms" registry:

* Name: ML-KEM-512
* Value: TBD1
* Description: PQ-KEM that uses ML-KEM-512 PQ-KEM.
* Capabilities: [kty]
* Change Controller: IESG
* Reference: This document (TBD)
* Recommended: No
* Name: ML-KEM-768
* Value: TBD2
* Description: PQ-KEM that uses ML-KEM-768 PQ-KEM.
* Capabilities: [kty]

* Change Controller: IESG
* Reference: This document (TBD)
* Recommended: No
* Name: ML-KEM-768
* Value: TBD3
* Description: PQ-KEM that uses ML-KEM-1024 PQ-KEM.
* Capabilities: [kty]
* Change Controller: IESG
* Reference: This document (TBD)
* Recommended: No
* Name: ML-KEM-512+A128KW
* Value: TBD4
* Description: PQ-KEM that uses ML-KEM-512 PQ-KEM and CEK wrapped with "A128KW".
* Capabilities: [kty]
* Change Controller: IESG
* Reference: This document (TBD)
* Recommended: No
* Name: ML-KEM-768+192KW
* Value: TBD5
* Description: PQ-KEM that uses ML-KEM-768 and CEK wrapped with "A192KW".
* Capabilities: [kty]
* Change Controller: IESG
* Reference: This document (TBD)

* Recommended: No

* Name: ML-KEM-1024+A256KW

* Value: TBD6

* Description: PQ-KEM that uses ML-KEM-1024 and CEK wrapped with "A256KW".

* Capabilities: [kty]

* Change Controller: IESG

* Reference: This document (TBD)

* Recommended: No

Acknowledgments

The authors thank AJITOMI Daisuke, Brian Campbell, Daniel Huigens, Filip Skokan, Ilari Liusvaara, Neil Madden, and Stepan Yakimovich for their contributions to this specification.

References

Normative References

[COSE-IANA-Curves]
IANA, "COSE Elliptic Curves", n.d., <https://www.iana.org/assignments/cose>.

[I-D.ietf-jose-hpke-encrypt]
Reddy.K, T., Tschofenig, H., Banerjee, A., Steele, O., and M. B. Jones, "Use of Hybrid Public Key Encryption (HPKE) with JSON Web Encryption (JWE)", Work in Progress, Internet-Draft, draft-ietf-jose-hpke-encrypt-15, 30 November 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-jose-hpke-encrypt-15>.

[JOSE-IANA]
IANA, "JSON Web Signature and Encryption Algorithms", n.d., <https://www.iana.org/assignments/jose/jose.xhtml>.

[JOSE-IANA-Curves]
IANA, "JSON Web Key Elliptic Curve", n.d., <https://www.iana.org/assignments/jose/jose.xhtml>.

[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>.

[RFC7516] Jones, M. and J. Hildebrand, "JSON Web Encryption (JWE)", RFC 7516, DOI 10.17487/RFC7516, May 2015, <https://www.rfc-editor.org/rfc/rfc7516>.

[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>.

[RFC8949] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", STD 94, RFC 8949, DOI 10.17487/RFC8949, December 2020, <https://www.rfc-editor.org/rfc/rfc8949>.

Informative References

[FIPS203] "FIPS-203: Module-Lattice-based Key-Encapsulation Mechanism Standard", <https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.203.pdf>.

[FO] "Secure Integration of Asymmetric and Symmetric Encryption Schemes", <https://link.springer.com/article/10.1007/s00145-011-9114-1>.

[HHK] "A Modular Analysis of the Fujisaki-Okamoto Transformation", <https://link.springer.com/chapter/10.1007/978-3-319-70500-2_12>.

[I-D.ietf-cose-dilithium] Prorock, M. and O. Steele, "ML-DSA for JOSE and COSE", Work in Progress, Internet-Draft, draft-ietf-cose-dilithium-11, 15 November 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-cose-dilithium-11>.

[I-D.ietf-cose-hpke] Tschofenig, H., Steele, O., Daisuke, A., and L. Lundblade, "Use of Hybrid Public-Key Encryption (HPKE) with CBOR Object Signing and Encryption (COSE)", Work in Progress, Internet-Draft, draft-ietf-cose-hpke-18, 19 October 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-cose-hpke-18>.

[I-D.ietf-pquip-pqc-engineers]
Banerjee, A., Reddy.K, T., Schoinianakis, D., Hollebeek, T., and M. Ounsworth, "Post-Quantum Cryptography for Engineers", Work in Progress, Internet-Draft, draft-ietf-pquip-pqc-engineers-14, 25 August 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-pquip-pqc-engineers-14>.

[I-D.ietf-pquip-pqt-hybrid-terminology]
D, F., P, M., and B. Hale, "Terminology for Post-Quantum Traditional Hybrid Schemes", Work in Progress, Internet-Draft, draft-ietf-pquip-pqt-hybrid-terminology-06, 10 January 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-pquip-pqt-hybrid-terminology-06>.

[I-D.ietf-tls-hybrid-design]
Stebila, D., Fluhrer, S., and S. Gueron, "Hybrid key exchange in TLS 1.3", Work in Progress, Internet-Draft, draft-ietf-tls-hybrid-design-16, 7 September 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-tls-hybrid-design-16>.

[I-D.reddy-cose-jose-pqc-hybrid-hpke]
Reddy.K, T. and H. Tschofenig, "PQ/T Hybrid KEM: HPKE with JOSE/COSE", Work in Progress, Internet-Draft, draft-reddy-cose-jose-pqc-hybrid-hpke-08, 7 July 2025, <https://datatracker.ietf.org/doc/html/draft-reddy-cose-jose-pqc-hybrid-hpke-08>.

[NIST.SP.800-56Ar3]
National Institute of Standards and Technology, "Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography, NIST Special Publication 800-56A Revision 3", April 2018, <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Ar3.pdf>.

[NIST.SP.800-57pt1r5]
National Institute of Standards and Technology, "Recommendation for Key Management, Part 1 General, NIST Special Publication 800-57 Part 1 Revision 5", May 2020, <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r5.pdf>.

[NISTFINAL]
"NIST Releases First 3 Finalized Post-Quantum Encryption Standards", n.d., <https://www.nist.gov/news-events/news/2024/08/nist-releases-first-3-finalized-post-quantum-encryption-standards>.

[PQCAPI] "PQC - API notes", <https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/example-files/api-notes.pdf>.

[RFC6090] McGrew, D., Igoe, K., and M. Salter, "Fundamental Elliptic Curve Cryptography Algorithms", RFC 6090, DOI 10.17487/RFC6090, February 2011, <https://www.rfc-editor.org/rfc/rfc6090>.

[RFC7518] Jones, M., "JSON Web Algorithms (JWA)", RFC 7518, DOI 10.17487/RFC7518, May 2015, <https://www.rfc-editor.org/rfc/rfc7518>.

[RFC8037] Liusvaara, I., "CFRG Elliptic Curve Diffie-Hellman (ECDH) and Signatures in JSON Object Signing and Encryption (JOSE)", RFC 8037, DOI 10.17487/RFC8037, January 2017, <https://www.rfc-editor.org/rfc/rfc8037>.

[RFC9052] Schaad, J., "CBOR Object Signing and Encryption (COSE): Structures and Process", STD 96, RFC 9052, DOI 10.17487/RFC9052, August 2022, <https://www.rfc-editor.org/rfc/rfc9052>.

[RFC9053] Schaad, J., "CBOR Object Signing and Encryption (COSE): Initial Algorithms", RFC 9053, DOI 10.17487/RFC9053, August 2022, <https://www.rfc-editor.org/rfc/rfc9053>.

[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>.

[RSA] "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems+", <https://dl.acm.org/doi/pdf/10.1145/359340.359342>.

[SP-800-108r1]
"Recommendation for Key Derivation Using Pseudorandom Functions", <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-108r1-upd1.pdf>.

Authors' Addresses

Tirumaleswar Reddy
Nokia
Bangalore
Karnataka
India
Email: kondtir@gmail.com

Aritra Banerjee
Nokia
London
United Kingdom
Email: aritra.banerjee@nokia.com

Hannes Tschofenig
University of Applied Sciences Bonn-Rhein-Sieg
Germany
Email: hannes.tschofenig@gmx.net