COSE
Internet-Draft
Intended status: Standards Track
Expires: 20 August 2026

T. Reddy
Nokia
H. Tschofenig
16 February 2026

Post-Quantum and Hybrid KEMs for HPKE with JOSE and COSE
draft-reddy-cose-jose-pqc-hybrid-hpke-11

Abstract

This document specifies the use of Post-Quantum (PQC) and Post-Quantum/Traditional (PQ/T) Hybrid Key Encapsulation Mechanisms (KEMs) within the Hybrid Public Key Encryption (HPKE) for JOSE and COSE. It defines algorithm identifiers and key formats to support pure post-quantum algorithms (ML-KEM) and their PQ/T hybrid combinations.

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-reddy-cose-jose-pqc-hybrid/.

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

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 20 August 2026.

Table of Contents

1. Introduction

The migration to Post-Quantum Cryptography (PQC) is unique in the history of modern digital cryptography in that neither the traditional algorithms nor the post-quantum algorithms are fully trusted to protect data for the required data lifetimes. The traditional algorithms, such as RSA and elliptic curve cryptography (ECC), will fall to quantum cryptanalysis, while the post-quantum

algorithms face uncertainty about the underlying mathematics, compliance issues, unknown vulnerabilities, hardware and software implementations that have not had sufficient maturing time to rule out classical cryptanalytic attacks and implementation bugs.

During this transition, deployments may adopt different strategies depending on their security posture, risk tolerance, and external constraints. Hybrid key exchange is generally preferred over pure PQC key exchange because it provides defense in depth by combining the strengths of both traditional and post-quantum algorithms. This approach ensures continued security even if one of the component algorithms is compromised during the transitional period.

However, pure PQC key exchange may be required for specific deployments with regulatory or compliance mandates that necessitate the exclusive use of post-quantum cryptography. Such requirements may arise in environments governed by stringent cryptographic standards that prohibit reliance on traditional public-key algorithms.

Hybrid Public Key Encryption (HPKE) specifies a scheme for encrypting arbitrary-length plaintexts to a recipient's public key. The use of HPKE with JOSE and COSE is specified in [I-D.ietf-jose-hpke-encrypt] and [I-D.ietf-cose-hpke], respectively. HPKE can be extended to support both pure PQC and post-quantum/traditional (PQ/T) hybrid Key Encapsulation Mechanisms (KEMs), as defined in [I-D.ietf-hpke-pq]. This document specifies the use of these KEMs in HPKE for JOSE and COSE.

Supporting both pure PQC and PQ/T hybrid KEMs enables flexible deployment choices: hybrid mechanisms provide a conservative transition strategy with defense in depth, while pure PQC mechanisms accommodate deployments with regulatory, compliance, or policy-driven requirements for exclusive use of PQC.

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]. 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, elliptic curve discrete logarithms, or related mathematical problems. 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. Examples of PQC key exchange algorithms include ML-KEM.

"Post-Quantum Traditional (PQ/T) Hybrid Scheme": A multi-algorithm scheme where at least one component algorithm is a post-quantum algorithm and at least one is a traditional algorithm.

"PQ/T Hybrid Key Encapsulation Mechanism": A multi-algorithm KEM made up of two or more component KEM algorithms where at least one is a post-quantum algorithm and at least one is a traditional algorithm.

3. Construction

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 parameter 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. For pure PQC, the ML-KEM algorithms are used as standalone KEMs within the HPKE framework as defined in [I-D.ietf-hpke-pq].

While this document defines ciphersuites for all three parameter sets, implementers should follow the guidance in Section 3 of [I-D.ietf-hpke-pq] regarding the selection of security levels. Specifically, it is noted that while ML-KEM-512 provides NIST security category 1, the use of ML-KEM-768 or ML-KEM-1024 is generally preferred to provide a higher security margin against potential future cryptanalysis.

PQ/T algorithms for HPKE [I-D.ietf-hpke-pq] use a multi-algorithm scheme, where one component algorithm is a post-quantum algorithm and another one is a traditional algorithm. The hybrid combiner construction, such as the C2PRICombiner defined in [I-D.irtf-cfrg-hybrid-kems], combines the shared secrets and public values from a post-quantum KEM and a traditional KEM to derive a single shared secret for HPKE.

4. Alignment with JOSE HPKE Modes

The JOSE HPKE specification [I-D.ietf-jose-hpke-encrypt] and the COSE HPKE specification [I-D.ietf-cose-hpke] define the use of HPKE with two Key Management Modes:

* Key Encryption, and

* Integrated Encryption.

In both JOSE and COSE, the selected Key Management Mode determines how HPKE is applied at the message layer. In Key Encryption mode, HPKE is used to encrypt a Content Encryption Key (CEK), which is then used to encrypt the payload. In Integrated Encryption mode, HPKE is used directly to encrypt the payload, and no separate CEK is employed.

Each Key Management Mode is identified by a distinct algorithm identifier (alg) in both JOSE and COSE. This document registers separate HPKE algorithm identifiers for Key Encryption and Integrated Encryption for both pure PQC and PQ/T hybrid HPKE instantiations.

This separation ensures that JOSE and COSE implementations can determine the intended HPKE Key Management Mode solely from the alg value, without ambiguity, and preserves compatibility with existing HPKE processing models.

5. Ciphersuite Registration

This specification registers a set of pure PQC and PQ/T hybrid KEMs for use with HPKE. In this context, an HPKE ciphersuite is defined as a combination of the following algorithm components:

* KEM algorithm, which may be either:

  - a pure PQC KEM (for example, ML-KEM-768), or

  - a PQ/T hybrid KEM that combines a PQC KEM with a traditional key-exchange algorithm (for example, ML-KEM-768 + X25519, defined as "MLKEM768-X25519" in [I-D.ietf-hpke-pq])

* KDF algorithm

* AEAD algorithm

The values for KEM, KDF, and AEAD are drawn from the HPKE IANA registry [HPKE-IANA]. Consequently, JOSE and COSE can only use algorithm combinations that are already defined and registered for HPKE.

The HPKE ciphersuites defined for use with JOSE and COSE, including both pure PQC and PQ/T hybrid KEMs, are specified in Section 9.

Note that the pure PQC and PQ/T hybrid KEMs defined for HPKE are not authenticated KEMs. As a result, only the HPKE Base mode is supported when using these KEMs, in accordance with the HPKE and JOSE/COSE HPKE specifications.

6. AKP Key Type for Use with PQC and PQ/T Hybrid HPKE Algorithms

This section describes the required parameters for an "AKP" key type, as defined in [I-D.ietf-cose-dilithium], and its use with pure PQC and PQ/T hybrid algorithms for HPKE, as defined in {#XWING} and {#XWING-KE}. An example key representation is also provided for illustration.

6.1. Required Parameters

A JSON Web Key (JWK) or COSE_Key with a key type (kty) for use with pure PQC or PQ/T hybrid algorithms for HPKE includes the following parameters:

* kty (Key Type)
  The kty parameter MUST be present and MUST be set to "AKP".

* alg (Algorithm)
  The alg parameter MUST be present and MUST identify the pure PQC or PQ/T hybrid algorithm for HPKE, as defined in {#XWING} or {#XWING-KE}.
  HPKE algorithms using pure PQC or PQ/T hybrid KEMs are those registered in the "JSON Web Signature and Encryption Algorithms" registry and the "COSE Algorithms" registry, and are derived from the corresponding KEM identifiers in the HPKE IANA registry.

* pub (Public Key)
  The pub parameter MUST be present and MUST contain the public encapsulation key (pk) as defined in Section 5.1 of [I-D.irtf-cfrg-hybrid-kems]. For hybrid KEMs, the PQC KEM public key MUST be placed first, followed immediately by the traditional public key, as specified in [I-D.ietf-hpke-pq]. No padding or delimiters are used between the keys.

  When represented as a JWK, this value MUST be base64url-encoded.

   * priv (Private Key)
      When representing a private key, the priv parameter MUST be
      present and MUST contain the private decapsulation key (sk) as
      defined in Section 5.1 of [I-D.irtf-cfrg-hybrid-kems]. For hybrid
      KEMs, the PQC KEM private key MUST be placed first, followed
      immediately by the traditional private key.
      When represented as a JWK, this value MUST be base64url-encoded.

6.1.1. Example

   The following is an example JWK representation of an "AKP" key for
   the "MLKEM768-X25519-SHAKE256-AES-256-GCM" algorithm:

```
   {
       "kty"  : "AKP",
       "alg"  : "HPKE-8",
       "pub"  : "4iNrNajCSz...tmrrIzQSQQO9lNA",
       "priv" : "f5wrpOiP...rPpm7yY"
   }
```

7. Security Strength Analysis of Registered HPKE Ciphersuites

   This section provides an analysis of the security strength of the
   HPKE ciphersuites defined in this document.

Alg	Mode	KEM	Trad	KDF	AEAD	Sec
HPKE-8	Hybrid	MLKEM-768	P-256	SHAKE256	AES-256-GCM	Level 3/C128
HPKE-9	Hybrid	MLKEM-768	P-256	SHAKE256	ChaCha20-Poly1305	Level 3/C128
HPKE-10	Hybrid	MLKEM-768	X25519	SHAKE256	AES-256-GCM	Level 3/C128
HPKE-11	Hybrid	MLKEM-768	X25519	SHAKE256	ChaCha20-Poly1305	Level 3/C128
HPKE-12	Hybrid	MLKEM-1024	P-384	SHAKE256	AES-256-GCM	Level 5/C192
HPKE-13	Hybrid	MLKEM-1024	P-384	SHAKE256	ChaCha20-Poly1305	Level 5/C192
HPKE-14	Level	MLKEM-512	-	SHAKE256	AES-128-GCM	Level 1
HPKE-15	Level	MLKEM-768	-	SHAKE256	AES-256-GCM	Level 3
HPKE-16	Level	MLKEM-1024	-	SHAKE256	AES-256-GCM	Level 5

   * Level x = NIST post-quantum security level

   * Cyyy = Approximate classical security strength in bits for
      traditional algorithms.

   * Hybrid classical strength is bounded by the traditional component

   * -KE variants share identical cryptographic properties and are
      omitted

NIST post-quantum security Levels 1, 3, and 5 are defined by reference to the classical cost of breaking AES-128, AES-192, and AES-256, respectively. These levels apply to post-quantum algorithm targets and do not constitute classifications of the AES primitive itself.

KDF Selection: SHAKE256 is selected as the HPKE KDF to align with the SHAKE-based extendable-output functions (XOFs) used internally by ML-KEM. This avoids introducing an additional hash primitive into the cryptographic processing pipeline. The security capacity of SHAKE256 is sufficient to support NIST security levels 1, 3, and 5 without introducing a bottleneck in key derivation. This choice does not imply that alternative KDFs are cryptographically incompatible with ML-KEM; rather, using the same hash family reduces the number of distinct cryptographic primitives that implementations must support.

AEAD Selection: AES-128-GCM provides approximately 128-bit symmetric security strength and is sufficient for Level 3 constructions. As discussed in Section 3.1 of [I-D.ietf-pquip-pqc-engineers], symmetric cryptography such as AES remains secure in a post-quantum setting, and 128-bit symmetric algorithms are considered quantum-safe for the foreseeable future.

Nevertheless, this specification mandates AES-256-GCM for all Level 3 ciphersuites to maintain consistent symmetric key sizing across higher-strength suites and to avoid introducing AES-128 into Level 3 configurations. Because the HPKE AEAD registry does not define an identifier for AES-192-GCM, AES-256-GCM is selected as the stronger available AEAD option. AES-128-GCM, used in HPKE-14, limits the symmetric security strength to approximately 128 bits, which is consistent with ML-KEM-512 (Level 1).

PQ/T Hybrid: ML-KEM-1024 is paired with P-384 rather than P-521. While P-521 offers higher classical security strength, P-384 already provides a strong classical fallback (192-bit security), is widely implemented, and aligns with existing TLS PQ/T hybrid construction.

8. Security Considerations

The security considerations in [I-D.ietf-hpke-pq], [I-D.ietf-jose-hpke-encrypt] and [I-D.ietf-cose-hpke] are to be taken into account.

The shared secrets computed in the hybrid key exchange must be computed in a way that achieves the "hybrid" property: the resulting secret is secure as long as at least one of the component key exchange algorithms is unbroken. 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.

8.1. Post-Quantum Security for Multiple Recipients

In HPKE JWE Key Encryption, when encrypting the Content Encryption Key (CEK) for multiple recipients, it is crucial to consider the security requirements of the message to safeguard against "Harvest Now, Decrypt Later" attacks. For messages requiring post-quantum security, all recipients MUST use algorithms supporting post-quantum cryptographic methods, such as PQC KEMs or Hybrid PQ/T KEMs. Using traditional algorithms (e.g., ECDH-ES) for any recipient in these scenarios compromises the overall security of the message.

9. IANA Considerations

9.1. JOSE

This document requests IANA to add new values to the "JSON Web Signature and Encryption Algorithms" registry.

9.1.1. JOSE Algorithms for Integrated Encryption

* Algorithm Name: HPKE-8
* Algorithm Description: Integrated Encryption with HPKE using ML-KEM-768 + P-256 Hybrid KEM, the SHAKE256 KDF, and the AES-256-GCM AEAD.
* Algorithm Usage Location(s): "alg"
* JOSE Implementation Requirements: Optional
* Change Controller: IANA
* Specification Document(s): [[TBD: This RFC]]
* Algorithm Analysis Documents(s): TODO
* Algorithm Name: HPKE-9

* Algorithm Description: Integrated Encryption with HPKE using ML-KEM-768 + P-256 Hybrid KEM, the SHAKE256 KDF, and the ChaCha20Poly1305 AEAD.

* Algorithm Usage Location(s): "alg"

* JOSE Implementation Requirements: Optional

* Change Controller: IANA

* Specification Document(s): [[TBD: This RFC]]

* Algorithm Analysis Documents(s): TODO

* Algorithm Name: HPKE-10

* Algorithm Description: Integrated Encryption with HPKE using ML-KEM-768 + X25519 Hybrid KEM, the SHAKE256 KDF, and the AES-256-GCM AEAD.

* Algorithm Usage Location(s): "alg"

* JOSE Implementation Requirements: Optional

* Change Controller: IANA

* Specification Document(s): [[TBD: This RFC]]

* Algorithm Analysis Documents(s): TODO

* Algorithm Name: HPKE-11

* Algorithm Description: Integrated Encryption with HPKE using ML-KEM-768 + X25519 Hybrid KEM, the SHAKE256 KDF, and the ChaCha20Poly1305 AEAD.

* Algorithm Usage Location(s): "alg"

* JOSE Implementation Requirements: Optional

* Change Controller: IANA

* Specification Document(s): [[TBD: This RFC]]

* Algorithm Analysis Documents(s): TODO

* Algorithm Name: HPKE-12

* Algorithm Description: Integrated Encryption with HPKE using ML-KEM-1024 + P-384 Hybrid KEM, the SHAKE256 KDF, and the AES-256-GCM AEAD.

* Algorithm Usage Location(s): "alg"

* JOSE Implementation Requirements: Optional

* Change Controller: IANA

* Specification Document(s): [[TBD: This RFC]]

* Algorithm Analysis Documents(s): TODO

* Algorithm Name: HPKE-13

* Algorithm Description: Integrated Encryption with HPKE using ML-KEM-1024 + P-384 Hybrid KEM, the SHAKE256 KDF, and the ChaCha20Poly1305 AEAD.

* Algorithm Usage Location(s): "alg"

* JOSE Implementation Requirements: Optional

* Change Controller: IANA

* Specification Document(s): [[TBD: This RFC]]

* Algorithm Analysis Documents(s): TODO

* Algorithm Name: HPKE-14

* Algorithm Description: Integrated Encryption with HPKE using ML-KEM-512 KEM, the SHAKE256 KDF, and the AES-128-GCM AEAD.

* Algorithm Usage Location(s): "alg"

* JOSE Implementation Requirements: Optional

* Change Controller: IANA

* Specification Document(s): [[TBD: This RFC]]

* Algorithm Analysis Document(s): TODO

* Algorithm Name: HPKE-15

   * Algorithm Description: Integrated Encryption with HPKE using ML-KEM-768 KEM, the SHAKE256 KDF, and the AES-256-GCM AEAD.

   * Algorithm Usage Location(s): "alg"

   * JOSE Implementation Requirements: Optional

   * Change Controller: IANA

   * Specification Document(s): [[TBD: This RFC]]

   * Algorithm Analysis Document(s): TODO

   * Algorithm Name: HPKE-16

   * Algorithm Description: Integrated Encryption with HPKE using ML-KEM-1024 KEM, the SHAKE256 KDF, and the AES-256-GCM AEAD.

   * Algorithm Usage Location(s): "alg"

   * JOSE Implementation Requirements: Optional

   * Change Controller: IANA

   * Specification Document(s): [[TBD: This RFC]]

   * Algorithm Analysis Document(s): TODO

9.1.2. JOSE Algorithms for Key Encryption

   * Algorithm Name: HPKE-8-KE

   * Algorithm Description: Key Encryption with HPKE using ML-KEM-768 + P-256 Hybrid KEM, the SHAKE256 KDF, and the AES-256-GCM AEAD.

   * Algorithm Usage Location(s): "alg"

   * JOSE Implementation Requirements: Optional

   * Change Controller: IANA

   * Specification Document(s): [[TBD: This RFC]]

   * Algorithm Analysis Document(s): TODO

   * Algorithm Name: HPKE-9-KE

* Algorithm Description: Key Encryption with HPKE using ML-KEM-768 + P-256 Hybrid KEM, the SHAKE256 KDF, and the ChaCha20Poly1305 AEAD.

* Algorithm Usage Location(s): "alg"

* JOSE Implementation Requirements: Optional

* Change Controller: IANA

* Specification Document(s): [[TBD: This RFC]]

* Algorithm Analysis Document(s): TODO

* Algorithm Name: HPKE-10-KE

* Algorithm Description: Key Encryption with HPKE using ML-KEM-768 + X25519 Hybrid KEM, the SHAKE256 KDF, and the AES-256-GCM AEAD.

* Algorithm Usage Location(s): "alg"

* JOSE Implementation Requirements: Optional

* Change Controller: IANA

* Specification Document(s): [[TBD: This RFC]]

* Algorithm Analysis Document(s): TODO

* Algorithm Name: HPKE-11-KE

* Algorithm Description: Key Encryption with HPKE using ML-KEM-768 + X25519 Hybrid KEM, the SHAKE256 KDF, and the ChaCha20Poly1305 AEAD.

* Algorithm Usage Location(s): "alg"

* JOSE Implementation Requirements: Optional

* Change Controller: IANA

* Specification Document(s): [[TBD: This RFC]]

* Algorithm Analysis Document(s): TODO

* Algorithm Name: HPKE-12-KE

* Algorithm Description: Key Encryption with HPKE using ML-KEM-1024 + P-384 Hybrid KEM, the SHAKE256 KDF, and the AES-256-GCM AEAD.

   * Algorithm Usage Location(s): "alg"

   * JOSE Implementation Requirements: Optional

   * Change Controller: IANA

   * Specification Document(s): [[TBD: This RFC]]

   * Algorithm Analysis Document(s): TODO

   * Algorithm Name: HPKE-13-KE

   * Algorithm Description: Key Encryption with HPKE using ML-KEM-1024
      + P-384 Hybrid KEM, the SHAKE256 KDF, and the ChaCha20Poly1305
      AEAD.

   * Algorithm Usage Location(s): "alg"

   * JOSE Implementation Requirements: Optional

   * Change Controller: IANA

   * Specification Document(s): [[TBD: This RFC]]

   * Algorithm Analysis Document(s): TODO

   * Algorithm Name: HPKE-14-KE

   * Algorithm Description: Key Encryption with HPKE using ML-KEM-512
      KEM, the SHAKE256 KDF, and the AES-128-GCM AEAD.

   * Algorithm Usage Location(s): "alg"

   * JOSE Implementation Requirements: Optional

   * Change Controller: IANA

   * Specification Document(s): [[TBD: This RFC]]

   * Algorithm Analysis Document(s): TODO

   * Algorithm Name: HPKE-15-KE

   * Algorithm Description: Key Encryption with HPKE using ML-KEM-768
      KEM, the SHAKE256 KDF, and the AES-256-GCM AEAD.

   * Algorithm Usage Location(s): "alg"

* JOSE Implementation Requirements: Optional

* Change Controller: IANA

* Specification Document(s): [[TBD: This RFC]]

* Algorithm Analysis Document(s): TODO

* Algorithm Name: HPKE-16-KE

* Algorithm Description: Key Encryption with HPKE using ML-KEM-1024 KEM, the SHAKE256 KDF, and the AES-256-GCM AEAD.

* Algorithm Usage Location(s): "alg"

* JOSE Implementation Requirements: Optional

* Change Controller: IANA

* Specification Document(s): [[TBD: This RFC]]

* Algorithm Analysis Document(s): TODO

9.2. COSE

This document requests IANA to add new values to the 'COSE Algorithms' registry.

9.2.1. COSE Algorithms Registry

* Name: HPKE-8

* Value: TBD1

* Description: COSE HPKE Integrated Encryption using ML-KEM-768 + P-256 Hybrid KEM, the SHAKE256 KDF, and the AES-256-GCM AEAD.

* Capabilities: [kty]

* Change Controller: IESG

* Reference: [[TBD: This RFC]]

* Recommended: No

* Name: HPKE-9

* Value: TBD2

* Description: COSE HPKE Integrated Encryption using ML-KEM-768 + P-256 Hybrid KEM, the SHAKE256 KDF, and the ChaCha20Poly1305 AEAD.

* Capabilities: [kty]

* Change Controller: IESG

* Reference: [[TBD: This RFC]]

* Recommended: No

* Name: HPKE-10

* Value: TBD3

* Description: COSE HPKE Integrated Encryption using ML-KEM-768 + X25519 Hybrid KEM, the SHAKE256 KDF, and the AES-256-GCM AEAD.

* Capabilities: [kty]

* Change Controller: IESG

* Reference: [[TBD: This RFC]]

* Recommended: No

* Name: HPKE-11

* Value: TBD4

* Description: COSE HPKE Integrated Encryption using ML-KEM-768 + X25519 Hybrid KEM, the SHAKE256 KDF, and the ChaCha20Poly1305 AEAD.

* Capabilities: [kty]

* Change Controller: IESG

* Reference: [[TBD: This RFC]]

* Recommended: No

* Name: HPKE-12

* Value: TBD5

* Description: COSE HPKE Integrated Encryption using ML-KEM-1024 + P-384 Hybrid KEM, the SHAKE256 KDF, and the AES-256-GCM AEAD.

* Capabilities: [kty]

* Change Controller: IESG

* Reference: [[TBD: This RFC]]

* Recommended: No

* Name: HPKE-13

* Value: TBD6

* Description: COSE HPKE Integrated Encryption using ML-KEM-1024 + P-384 Hybrid KEM, the SHAKE256 KDF, and the ChaCha20Poly1305 AEAD.

* Capabilities: [kty]

* Change Controller: IESG

* Reference: [[TBD: This RFC]]

* Recommended: No

* Name: HPKE-14

* Value: TBD7

* Description: COSE HPKE Integrated Encryption using ML-KEM-512 KEM, the SHAKE256 KDF, and the AES-128-GCM AEAD.

* Capabilities: [kty]

* Change Controller: IESG

* Reference: [[TBD: This RFC]]

* Recommended: No

* Name: HPKE-15

* Value: TBD8

* Description: COSE HPKE Integrated Encryption using ML-KEM-768 KEM, the SHAKE256 KDF, and the AES-256-GCM AEAD.

* Capabilities: [kty]

* Change Controller: IESG

   * Reference: [[TBD: This RFC]]

   * Recommended: No

   * Name: HPKE-16

   * Value: TBD9

   * Description: COSE HPKE Integrated Encryption using ML-KEM-1024
      KEM, the SHAKE256 KDF, and the AES-256-GCM AEAD.

   * Capabilities: [kty]

   * Change Controller: IESG

   * Reference: [[TBD: This RFC]]

   * Recommended: No

   * Name: HPKE-8-KE

   * Value: TBD10

   * Description: COSE HPKE Key Encryption using ML-KEM-768 + P-256
      Hybrid KEM, the SHAKE256 KDF, and the AES-256-GCM AEAD.

   * Capabilities: [kty]

   * Change Controller: IESG

   * Reference: [[TBD: This RFC]]

   * Recommended: No

   * Name: HPKE-9-KE

   * Value: TBD11

   * Description: COSE HPKE Key Encryption using ML-KEM-768 + P-256
      Hybrid KEM, the SHAKE256 KDF, and the ChaCha20Poly1305 AEAD.

   * Capabilities: [kty]

   * Change Controller: IESG

   * Reference: [[TBD: This RFC]]

   * Recommended: No

* Name: HPKE-10-KE

* Value: TBD12

* Description: COSE HPKE Key Encryption using ML-KEM-768 + X25519 Hybrid KEM, the SHAKE256 KDF, and the AES-256-GCM AEAD.

* Capabilities: [kty]

* Change Controller: IESG

* Reference: [[TBD: This RFC]]

* Recommended: No

* Name: HPKE-11-KE

* Value: TBD13

* Description: COSE HPKE Key Encryption using ML-KEM-768 + X25519 Hybrid KEM, the SHAKE256 KDF, and the ChaCha20Poly1305 AEAD.

* Capabilities: [kty]

* Change Controller: IESG

* Reference: [[TBD: This RFC]]

* Recommended: No

* Name: HPKE-12-KE

* Value: TBD14

* Description: COSE HPKE Key Encryption using ML-KEM-1024 + P-384 Hybrid KEM, the SHAKE256 KDF, and the AES-256-GCM AEAD.

* Capabilities: [kty]

* Change Controller: IESG

* Reference: [[TBD: This RFC]]

* Recommended: No

* Name: HPKE-13-KE

* Value: TBD15

* Description: COSE HPKE Key Encryption using ML-KEM-1024 + P-384 Hybrid KEM, the SHAKE256 KDF, and the ChaCha20Poly1305 AEAD.

* Capabilities: [kty]

* Change Controller: IESG

* Reference: [[TBD: This RFC]]

* Recommended: No

* Name: HPKE-14-KE

* Value: TBD16

* Description: COSE HPKE Key Encryption using ML-KEM-512 KEM, the SHAKE256 KDF, and the AES-128-GCM AEAD.

* Capabilities: [kty]

* Change Controller: IESG

* Reference: [[TBD: This RFC]]

* Recommended: No

* Name: HPKE-15-KE

* Value: TBD17

* Description: COSE HPKE Key Encryption using ML-KEM-768 KEM, the SHAKE256 KDF, and the AES-256-GCM AEAD.

* Capabilities: [kty]

* Change Controller: IESG

* Reference: [[TBD: This RFC]]

* Recommended: No

* Name: HPKE-16-KE

* Value: TBD18

* Description: COSE HPKE Key Encryption using ML-KEM-1024 KEM, the SHAKE256 KDF, and the AES-256-GCM AEAD.

* Capabilities: [kty]

* Change Controller: IESG

* Reference: [[TBD: This RFC]]

* Recommended: No

Acknowledgments

Thanks to Ilari Liusvaara and Orie Steele for the discussion and comments.

References

Normative References

[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., Lundblade, L., and M. B. Jones, "Use of Hybrid Public-Key Encryption (HPKE) with CBOR Object Signing and Encryption (COSE)", Work in Progress, Internet-Draft, draft-ietf-cose-hpke-21, 2 February 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-cose-hpke-21>.

[I-D.ietf-hpke-pq]
Barnes, R. and D. Connolly, "Post-Quantum and Post-Quantum/Traditional Hybrid Algorithms for HPKE", Work in Progress, Internet-Draft, draft-ietf-hpke-pq-03, 6 November 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-hpke-pq-03>.

[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>.

[I-D.irtf-cfrg-hybrid-kems]
Connolly, D., Barnes, R., and P. Grubbs, "Hybrid PQ/T Key Encapsulation Mechanisms", Work in Progress, Internet-Draft, draft-irtf-cfrg-hybrid-kems-08, 27 January 2026, <https://datatracker.ietf.org/doc/html/draft-irtf-cfrg-hybrid-kems-08>.

[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>.

Informative References

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

[HPKE-IANA]
IANA, "Hybrid Public Key Encryption (HPKE) IANA Registry", <https://www.iana.org/assignments/hpke/hpke.xhtml>.

[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>.

[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>.

[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>.

Authors' Addresses

Tirumaleswar Reddy
Nokia
Bangalore
Karnataka
India
Email: k.tirumaleswar_reddy@nokia.com

Hannes Tschofenig
Germany
Email: hannes.tschofenig@gmx.net