IPSECME
Internet-Draft
Intended status: Standards Track
Expires: 2 April 2026

P. Kampanakis
Amazon Web Services

29 September 2025

Post-quantum Hybrid Key Exchange with ML-KEM in the Internet Key Exchange Protocol Version 2 (IKEv2) draft-ietf-ipsecme-ikev2-mlkem-03

Abstract

NIST recently standardized ML-KEM, a new key encapsulation mechanism, which can be used for quantum-resistant key establishment. This draft specifies how to use ML-KEM by itself or as an additional key exchange in IKEv2 along with a traditional key exchange. These options allow for negotiating IKE and Child SA keys which are safe against cryptographically relevant quantum computers and theoretical weaknesses in ML-KEM or implementation issues.

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 2 April 2026.

Copyright Notice

Table of Contents

1. Introduction

A Cryptographically Relevant Quantum Computer (CRQC), if it became a reality, could threaten today's public key establishment algorithms. Someone storing encrypted communications that use (Elliptic Curve) Diffie-Hellman ((EC)DH) to establish keys could decrypt these communications in the future after a CRQC became available to them. Such communications include Internet Key Exchange Protocol Version 2 (IKEv2).

To address this concern, the Mixing Preshared Keys in IKEv2 specification [RFC8784] introduced Post-quantum Preshared Keys (PPK) as a temporary option for stirring a pre-shared key of adequate entropy in the derived Child SA encryption keys in order to provide quantum-resistance. This specification can be used in conjunction with PPK as defined in [RFC8784]. Alternatively, [I-D.ietf-ipsecme-ikev2-qr-alt] can be used for mixing pre-shared keys in IKEv2, as it provides better security properties than [RFC8784] and, since the PPK negotiation can be combined with additional ML-KEM key exchanges and the extra round trip penalty can be avoided.

Since then, NIST has been working on a public project [NIST-PQ] for standardizing quantum-resistant algorithms which include key encapsulation and signatures. At the end of Round 3, they picked Kyber as the first Key Encapsulation Mechanism (KEM) for standardization. . Kyber was then standardized as Module-Lattice-based Key-Encapsulation Mechanism (ML-KEM) in 2024 [FIPS203].

As post-quantum public keys and ciphertexts may make UDP packet sizes larger than common network Maximum Transport Units (MTU), the Intermediate Exchange in IKEv2 document [RFC9242] defined how to do additional large message exchanges by using new IKE_INTERMEDIATE messages. IKE_INTERMEDIATE messages can only be used after IKE_SA_INIT. The Multiple Key Exchanges in IKEv2 specification [RFC9370] defined how to do up to seven additional key exchanges by using IKE_INTERMEDIATE or IKE_FOLLOWUP_KE messages and by deriving new SKEYSEED and KEYMAT key materials. These messages can be fragmented at the IKEv2 layer before causing IP fragmentation [RFC7383]. If a post-quantum KEM does not fit inside IKE_SA_INIT without causing IP fragmentation, then it can be used after IKE_SA_INIT in an IKE_INTERMEDIATE, CREATE_CHILD_SA, or IKE_FOLLOWUP_KE message as an additional key establishment algorithm.

This document describes how ML-KEM can be used as a quantum-resistant KEM in IKEv2 in an IKE_SA_INIT or CREATE_CHILD_SA exchange, or in one additional IKE_INTERMEDIATE or IKE_FOLLOWUP_KE key exchange after an initial IKE_SA_INIT or CREATE_CHILD_SA respectively. This approach of combining a quantum-resistant with a traditional algorithm, is commonly called Post-Quantum Traditional (PQ/T) Hybrid [RFC9794] key exchange and combines the security of a well-established algorithm with relatively new quantum-resistant algorithms. The result is a new Child SA key or an IKE or Child SA rekey with keying material which is safe against a CRQC. Another use of a PQ/T Hybrid key exchange in IKEv2 is for someone that wants to exchange keys using the high security parameter of ML-KEM. As these may not fit in common network packet payload sizes, they will need to be sent in a IKE_FOLLOWUP_KE or CREATE_CHILD_SA key exchange which can be

fragmented. This specification is a profile of the Multiple Key Exchanges in IKEv2 specification [RFC9370] and registers new algorithm identifiers for ML-KEM key exchanges in IKEv2.

1.1. KEMs

In the context of the NIST Post-Quantum Cryptography Standardization Project [NIST-PQ], key exchange algorithms are formulated as KEMs, which consist of three steps:

* 'KeyGen() -> (pk, sk)': A probabilistic key generation algorithm, which generates a public / encapsulation key 'pk' and a private / decapsulation key 'sk'. The resulting pk is sent to the responder in the KEi payload.

* 'Encaps(pk) -> (ct, ss)': A probabilistic encapsulation algorithm, which takes as input a public key pk (from the KEi) and outputs a ciphertext 'ct' and shared secret 'ss'. The ct is sent back to intiator in the KEr payload.

* 'Decaps(sk, ct) -> ss': A decapsulation algorithm, which takes as input a secret key sk and ciphertext ct (from the KEr) and outputs a shared secret ss, or in some rare cases a distinguished error value.

1.2. ML-KEM

ML-KEM is a standardized lattice-based key encapsulation mechanism [FIPS203]. It uses Module Learning with Errors as its underlying primitive which is a structured lattices variant that offers good performance and relatively small and balanced key and ciphertext sizes. ML-KEM was standardized with three parameters, ML-KEM-512, ML-KEM-768, and ML-KEM-1024. These were mapped by NIST to the three security levels defined in the NIST PQC Project, Level 1, 3, and 5. These levels correspond to the hardness of breaking AES-128, AES-192 and AES-256 respectively.

ML-KEM-512, ML-KEM-768 and ML-KEM-1024 key exchanges will not have noticeable performance impact on IKEv2/IPsec tunnels which usually stay up for long periods of time and transfer sizable amounts of data. Since the ML-KEM-768 and ML-KEM-1024 public key and ciphertext sizes can exceed the network MTU, these key exchanges could require two or three network IP packets from both the initiator and the responder.

1.3. 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.

2. ML-KEM in IKEv2

2.1. ML-KEM in IKE_INTERMEDIATE, CREATE_CHILD_SA, or IKE_FOLLOWUP_KE messages

ML-KEM key exchanges can be negotiated in IKE_INTERMEDIATE or IKE_FOLLOWUP_KE messages as defined in the Multiple Key Exchanges in IKEv2 specification [RFC9370]. We summarize them here for completeness.

Section 2.2.2 of [RFC9370] specifies that KEi(0), KEr(0) are regular key exchange messages in the first IKE_SA_INIT exchange which end up generating a set of keying material, SK_d, SK_a[i/r], and SK_e[i/r]. The peers then perform an IKE_INTERMEDIATE exchange, carrying new Key Exchange payloads. These are protected with the SK_e[i/r] and SK_a[i/r] keys which were derived from the IKE_SA_INIT as per Section 3.3.1 of the Intermediate Exchange in IKEv2 document [RFC9242]. The initiator generates an ML-KEM keypair (pk, sk) using KeyGen(), and sends the public key (pk) to the responder inside a KEi(1) payload. The responder will encapsulate a shared secret ss using Encaps(pk) and the resulting ciphertext (ct) is sent to initiator using the KEr(1). After the initiator receives KEr(1), it will decapsulate it using Decaps(sk, ct). Both Encaps and Decaps return the shared secret (ss) and both peers have a common shared secret SK(1) at the end of this KE(1) exchange. The ML-KEM shared secret is stirred into new keying material SK_d, SK_a[i/r], and SK_e[i/r] as defined in Section 2.2.2 of the Multiple Key Exchanges in IKEv2 document [RFC9370]. Afterwards the peers can perform more exchanges if necessary and then continue to the IKE_AUTH exchange phase as defined in Section 3.3.2 of the Intermediate Exchange in IKEv2 specification [RFC9242].

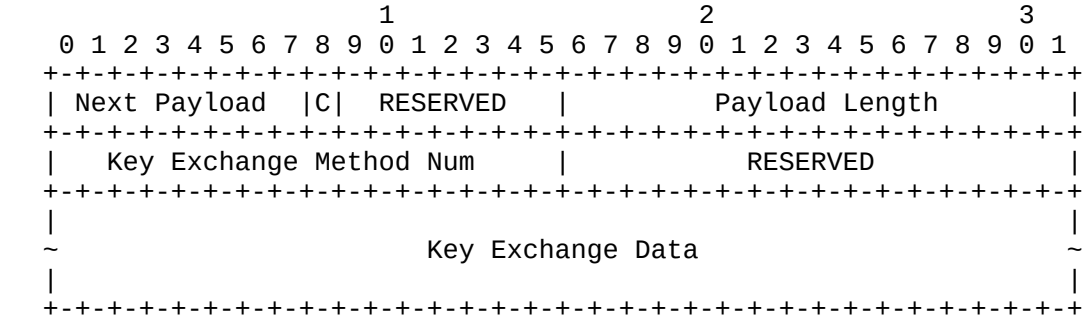
ML-KEM can also be used to create or rekey a Child SA or rekey the IKE SA in a PQ/T Hybrid approach by using a IKE_FOLLOWUP_KE exchange which follows a traditional CREATE_CHILD_SA. After the additional ML-KEM key exchange KE(1) has taken place in the IKE_FOLLOWUP_KE exchange, the IKE or Child SA are rekeyed by stirring the new ML-KEM shared secret SK(1) in SKEYSEED and KEYMAT as specified in Section 2.2.4 of [RFC9370]. Alternatively, ML-KEM can still be used on its own in a CREATE_CHILD_SA that rekeys the IKE or IPsec SAs without any other key exchanges as per [RFC7296].

ML-KEM-768 and ML-KEM-1024 public keys and ciphertexts may make UDP packet sizes larger than typical network MTUs. Thus, IKE_INTERMEDIATE or IKE_FOLLOWUP_KE messages carrying ML-KEM public keys and ciphertexts may be IKEv2 fragmented as per the IKEv2 Message Fragmentation specification [RFC7383].

Although, this document focuses on using ML-KEM as the second key exchange in a PQ/T Hybrid KEM [RFC9794] scenario, ML-KEM-512 and ML-KEM-768 Key Exchange Method identifiers 35 and 36 respectively MAY be used in IKE_SA_INIT as a quantum-resistant-only key exchange. The encapsulation key and ciphertext sizes for these ML-KEM parameters may not push the UDP packet to size larger than typical network MTUs. On the other hand, IKE_SA_INIT messages using ML-KEM-1024 Key Exchange Method identifier 37 could exceed typical network MTUs and could not be IKEv2 fragmented. Thus, implementations transporting IKE over UDP and not performing Path MTU (PMTU) discovery SHOULD NOT use ML-KEM-1024 in the IKE_SA_INIT exchange on networks where the PMTU is unknown or restricted. However, when reliable transport is used for IKE (e.g. [RFC9329], [I-D.smyslov-ipsecme-ikev2-reliable-transport]) or a sufficient PMTU is guaranteed, implementations MAY use ML-KEM-1024 in an IKE_SA_INIT exchange.

2.2. Key Exchange Payload

The KE payload is shown below and the fields inside it has meaning as defined in Section 3.4 of the IKEv2 standard [RFC7296]:



The Key Exchange Data from the initiator to the responder contains the public key (pk) from the KeyGen() operation encoded as a raw byte array (i.e., output of ByteEncode) as defined in Section 7.1 of Module-Lattice-Based KEM standard [FIPS203].

The Key Exchange Data from the responder to the initiator contains the ciphertext (ct) from the Encaps operation encoded as a raw byte array.

Table 1 shows the Payload Length, Key Exchange Method Num identifier and the Key Exchange Data Size in octets for Key Exchange Payloads from the initiator and the responder for the ML-KEM variants specified in this document.

KEM	Payload Length (initiator / responder)	Key Exchange Method Num	Data Size in Octets (initiator / responder)
ML-KEM-512	808 / 776	35	800 / 768
ML-KEM-768	1192 / 1096	36	1184 / 1088
ML-KEM-1024	1576 / 1576	37	1568 / 1568

Table 1: Key Exchange Payload Fields

2.3. Recipient Tests

Receiving and handling of malformed ML-KEM public keys or ciphertexts must follow the input validation described in the Module-Lattice-Based KEM standard [FIPS203].

Responders MUST perform the checks on the initiator public key specified in section 7.2 of the Module-Lattice-Based KEM standard [FIPS203] before the Encaps(pk) operation. If the checks fail, the responder SHOULD send a Notify payload of type INVALID_SYNTAX as a response to the request from initiator.

Initiators MUST perform the Ciphertext type check specified in section 7.3 of the Module-Lattice-Based KEM standard [FIPS203] before the Decaps(sk, ct) operation. If the check fails, the initiator MUST reject the ciphertext and MUST fail the exchange, log the error, and stop creating the SA (i.e. not initiate IKE_AUTH or next IKE_INTERMEDIATE). If the error occurs in the CREATE_CHILD_SA or IKE_FOLLOWUP_KE exchanges, the initiator MUST delete the existing IKE SA and send a Delete payload in a new INFORMATIONAL exchange for the responder to also remove it.

Note that during decapsulation, ML-KEM uses implicit rejection which leads the decapsulating entity to implicitly reject the decapsulated shared secret by setting it to a hash of the ciphertext together with a random value stored in the ML-KEM secret when the re-encrypted shared secret does not match the original one.

Section 4 of [SP800227] includes guidelines for using KEMs securely in applications.

3. Security Considerations

All security considerations from [RFC9242] and [RFC9370] apply to the ML-KEM exchanges described in this specification.

The main security property for KEMs standardized by NIST is indistinguishability under adaptive chosen ciphertext attacks (IND-CCA2) [FIPS203], which means that shared secret values should be indistinguishable from random strings even given the ability to have arbitrary ciphertexts decapsulated. IND-CCA2 corresponds to security against an active attacker, and the public key / secret key pair can be treated as a long-term key or reused. A weaker security notion is indistinguishability under chosen plaintext attacks (IND-CPA), which means that the shared secret values should be indistinguishable from random strings given a copy of the public key. IND-CPA roughly corresponds to security against a passive attacker, and sometimes corresponds to one-time key exchange. Generating an ephemeral keypair and ciphertext for each ML-KEM key exchange is REQUIRED by this specification. Note that this is also common practice for (EC)DH keys today. Responders also MUST NOT reuse randomness in the generation of KEM ciphertexts.

The ML-KEM public key generated by the initiator and the ciphertext generated by the responder use randomness (usually a seed) which MUST be independent of any other random seed used in the IKEv2 negotiation. For example, at the initiator, the ML-KEM and (EC)DH keypairs used in a PQ/T Hybrid key exchange MUST NOT be generated from the same seed.

When using PQ/T Hybrid key exchanges, SKEYSEED and KEYMAT in this specification are generated by using shared secrets, nonces, and SPIs with a pseudorandom function as defined in [RFC9370]. As discussed in [PQ-PROOF2], such PQ/T Hybrid key derivations are IND-CPA, but not proven to be IND-CCA2 secure.

IKEv2 is susceptible to downgrade attacks where an active man-in-the-middle could force the peers to negotiate the weakest key exchange method supported by both. In particular, if both peers support some sequence of key exchanges that involve only traditional algorithms, an active, on-path attacker with a CRQC may be able to convince the peers to use it even if they both support ML-KEM as well. Note that to achieve such a downgrade, the adversary needs to break traditional (EC)DH IKE_SA_INIT ephemeral exchanges while the negotiation is still taking place and completely control the flow to delay or drop legitimate IKEv2 messages. IKEv2 downgrades is a known issue [DOWN-RES] caused by the way IKEv2 authenticates messages only in one direction of the exchange; [PQIKEV2-FA] concluded that IKE_INTERMEDIATE [RFC9370] does not introduce additional attacks with respect to IKEv2’s original security model.

The simplest way to prevent such active attacks is to disable support for traditional-only sequences of key exchanges whenever possible. If the responder knows out-of-band that initiators support ML-KEM, then it SHOULD reject any proposal that doesn’t include ML-KEM in the IKE_SA_INIT or IKE_INTERMEDIATE. Likewise, if the initiator knows out-of-band that a responder supports ML-KEM, it SHOULD only include proposals for ML-KEM or abort the negotiation if the responder selects a proposal that doesn’t include ML-KEM. A long-term solution for the downgrade issue in IKEv2 is proposed in [I-D.smyslov-ipsecme-ikev2-downgrade-prevention].

As an alternative, in cases where only a subset of peer identities is known to have been upgraded to support ML-KEM the peers can enforce a policy to not encrypt any data until an ML_KEM exchange has taken place. [RFC9370] supports Childless IKE SAs which can be followed by a new Child SA after doing more key exchanges. To ensure that data is encrypted over a quantum-resistant IPsec Child SA, the peers could enforce a policy which first establishes a Childless IKE SA [RFC6023] (or a Child SA which does not encrypt any data) with a traditional key exchange and without an IKE_INTERMEDIATE exchange. Subsequently

the peers can rekey the initial IKE SA and derive a new Child SA (or rekey the existing Child SA that did not encrypt any data) with ML-KEM in a CREATE_CHILD_SA exchange or with ML-KEM as an additional key exchange in a IKE_FOLLOWUP_KE exchange which follows a traditional CREATE_CHILD_SA exchange. Section 2.2.5.1 of [RFC9370] discusses the details of the latter PQ/T Hybrid approach. This approach has the disadvantage that an adversary with a CRQC that could decrypt the IKE_SA_INIT exchange has access to all the information exchanged over the initial IKE SA or Child SA before the rekey. This information includes the identities of the peers, configuration parameters, and all negotiated SA information (including traffic selectors), but not the information and data encrypted after the CREATE_CHILD_SA (and IKE_FOLLOWUP_KE with ML-KEM)

4. IANA Considerations

IANA is requested to assign three values for the names "ml-kem-512", "ml-kem-768", and "ml-kem-1024" in the IKEv2 "Transform Type 4 - Key Exchange Method Transform IDs" and has listed this document as the reference. The Recipient Tests field should also point to this document:

Number	Name	Status	Recipient Tests	Reference
35	ml-kem-512		[TBD, this RFC, Section 2.3],	[TBD, this RFC]
36	ml-kem-768		[TBD, this RFC, Section 2.3],	[TBD, this RFC]
37	ml-kem-1024		[TBD, this RFC, Section 2.3],	[TBD, this RFC]

Table 2: Updates to the IANA "Transform Type 4 - Key Exchange Method Transform IDs" table

5. References

5.1. Normative References

[FIPS203] National Institute of Standards and Technology (NIST), "Module-Lattice-Based Key-Encapsulation Mechanism Standard", NIST Federal Information Processing Standards, 13 August 2024, <https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.203.pdf>.

[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>.

[RFC7296] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., and T. Kivinen, "Internet Key Exchange Protocol Version 2 (IKEv2)", STD 79, RFC 7296, DOI 10.17487/RFC7296, October 2014, <https://www.rfc-editor.org/info/rfc7296>.

[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>.

[RFC9242] Smyslov, V., "Intermediate Exchange in the Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 9242, DOI 10.17487/RFC9242, May 2022, <https://www.rfc-editor.org/info/rfc9242>.

[RFC9370] Tjhai, CJ., Tomlinson, M., Bartlett, G., Fluhrer, S., Van Geest, D., Garcia-Morchon, O., and V. Smyslov, "Multiple Key Exchanges in the Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 9370, DOI 10.17487/RFC9370, May 2023, <https://www.rfc-editor.org/info/rfc9370>.

5.2. Informative References

[DOWN-RES] Bhargavan, K., Brzuska, C., Fournet, C., Green, M., Kohlweiss, M., and S. Zanella-B迪guelin, "Downgrade Resilience in Key-Exchange Protocols", 2016, <https://ieeexplore.ieee.org/document/7546520>.

[I-D.ietf-ipsecme-ikev2-qr-alt] Smyslov, V., "Mixing Preshared Keys in the IKE_INTERMEDIATE and in the CREATE_CHILD_SA Exchanges of IKEv2 for Post-quantum Security", Work in Progress, Internet-Draft, draft-ietf-ipsecme-ikev2-qr-alt-10, 23 May 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-ipsecme-ikev2-qr-alt-10>.

[I-D.smyslov-ipsecme-ikev2-downgrade-prevention] Smyslov, V. and C. Patton, "Prevention Downgrade Attacks on the Internet Key Exchange Protocol Version 2 (IKEv2)", Work in Progress, Internet-Draft, draft-smyslov-ipsecme-ikev2-downgrade-prevention-02, 28 August 2025, <https://datatracker.ietf.org/doc/html/draft-smyslov-ipsecme-ikev2-downgrade-prevention-02>.

[I-D.smyslov-ipsecme-ikev2-reliable-transport]
Smyslov, V. and T. Reddy.K, "Separate Transports for IKE and ESP", Work in Progress, Internet-Draft, draft-smyslov-ipsecme-ikev2-reliable-transport-04, 15 April 2025, <https://datatracker.ietf.org/doc/html/draft-smyslov-ipsecme-ikev2-reliable-transport-04>.

[IKEv2-A] Petcher, A. and E. Assuncao, "Analyzing IKEv2: Security Proofs, Known Attacks, and Other Insights", 2025, <https://ethz.ch/content/dam/ethz/special-interest/infk/inst-infsec/appliedcrypto/education/theses/semester-project_eduarda-assuncao.pdf>.

[NIST-PQ] National Institute of Standards and Technology (NIST), "Post-Quantum Cryptography", https://csrc.nist.gov/projects/post-quantum-cryptography .

[PQ-PROOF2]
Petcher, A. and M. Campagna, "Security of Hybrid Key Establishment using Concatenation", 2023, <https://eprint.iacr.org/2023/972>.

[PQIKEV2-FA]
Gazdag, S., Grundner-Culemann, S., Guggemos, T., Heider, T., and D. Loebenberger, "A formal analysis of IKEv2寰洂 post-quantum extension", 2021, <https://www.mnm-team.org/pub/Publikationen/gggh21b/PDF-Version/gggh21b.pdf>.

[RFC6023] Nir, Y., Tschofenig, H., Deng, H., and R. Singh, "A Childless Initiation of the Internet Key Exchange Version 2 (IKEv2) Security Association (SA)", RFC 6023, DOI 10.17487/RFC6023, October 2010, <https://www.rfc-editor.org/info/rfc6023>.

[RFC7383] Smyslov, V., "Internet Key Exchange Protocol Version 2 (IKEv2) Message Fragmentation", RFC 7383, DOI 10.17487/RFC7383, November 2014, <https://www.rfc-editor.org/info/rfc7383>.

[RFC8784] Fluhrer, S., Kampanakis, P., McGrew, D., and V. Smyslov, "Mixing Preshared Keys in the Internet Key Exchange Protocol Version 2 (IKEv2) for Post-quantum Security", RFC 8784, DOI 10.17487/RFC8784, June 2020, <https://www.rfc-editor.org/info/rfc8784>.

[RFC9329] Pauly, T. and V. Smyslov, "TCP Encapsulation of Internet Key Exchange Protocol (IKE) and IPsec Packets", RFC 9329, DOI 10.17487/RFC9329, November 2022, <https://www.rfc-editor.org/info/rfc9329>.

[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>.

[SP800227] National Institute of Standards and Technology (NIST), "Recommendations for Key-Encapsulation Mechanisms", NIST Federal Information Processing Standards, 18 September 2025, <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-227.pdf>.

Acknowledgments

The authors would like to thank Valery Smyslov, Graham Bartlett, Scott Fluhrer, Ben S, Leonie Bruckert, Tero Kivinen, Rebecca Guthrie, Wang Guilin, Michael Richardson, John Mattsson, and Gerardo Ravago for their valuable feedback. Special thanks to Chris Patton for bringing up the downgrade issue.

Author’s Address

Panos Kampanakis
Amazon Web Services
Email: kpanos@amazon.com