IP Security Maintenance and Extensions
Internet-Draft
Intended status: Standards Track
Expires: 3 September 2026

G. Wang, Ed.
Huawei Int. Pte Ltd
L. Bruckert
secunet Security Networks
V. Smyslov
ELVIS-PLUS
M. Chen
China Mobile

2 March 2026

Post-quantum Hybrid Key Exchange in IKEv2 with FrodoKEM
draft-ietf-ipsecme-hybrid-kem-ikev2-frodo-00

Abstract

FrodoKEM is an unstructured lattice based Key Encapsulation Mechanism (KEM). Compared to ML-KEM, it is considered with more conservative security. This draft specifies how to use FrodoKEM by itself or as an additional key exchange in IKEv2 along with a traditional key exchange. These options enable to negotiate IKE and Child SA keys that are safe against a Cryptographically Relevant Quantum Computer (CRQC).

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 3 September 2026.

Table of Contents

1. Notes of Change

Changes made in version draft-ietf-ipsecme-kem-auth-ikev2-00:

* A new co-author is added.

* Clarifications are added that FrodoKEM can be used either as an additional key exchange method, or as the only key exchange method for IKE SA (provided there is no transport issues).

* References are re-arranged and updated.

* Github address of this document added: https://github.com/smyslov/draft-wang-ipsecme-hybrid-kem-ikev2-frodo/.

* Editorial changes throughout the document: Shorten Introduction, aligned Security Discussions with [W-D.K25], ...

Changes made in version draft-wang-ipsecme-kem-auth-ikev2-03:

* This specification has switched 4 variants of eFrodoKEM (ephemeral mode) to those of FrodoKEM (standard mode). The reasons are given in Section 4.2.

* Other Sections are updated correspondingly.

Three main changes have been made in version draft-wang-ipsecme-kem-auth-ikev2-02, as a response to comments received since July of 2025.

* Reduced code points from 6 in v01 to 4 now (revised Sections 3.3 and 6).

* Explained why variants for both AES and SHAKE are necessary (revised Section 3.2).

* Added KEi and KEr payloads for using FrodoKEM in IKEv2 (added new Sections 4.3).

Two main changes have been made in version draft-wang-ipsecme-kem-auth-ikev2-01, as a response to comments received at 122 meeting:

* Restructured the draft.

* Reduced the point codes from 12 to 6 (eFrodoKEM).

2. Introduction

Cryptographically-relevant quantum computers (CRQCs) pose a threat to data protected using traditional security algorithms. In particular, the so-called harvest-now-and-decrypt-later (HNDL) attack is considered an imminent threat. To mitigate this threat, the concept of hybrid key encapsulation mechanisms (KEMs) has been proposed to achieve secure key exchange if at least one of the KEMs is still secure [RFC9794]. “Multiple key exchanges in the Internet Key Exchange Protocol Version 2 (IKEv2) [RFC9370] specifies a framework to perform hybrid key encapsulation in IKEv2 by allowing multiple key exchanges for deriving shared secret keys during a Security Association (SA) setup. This framework employs the IKE_INTERMEDIATE exchange, which is a new IKEv2 exchange introduced in “Intermediate Exchange in the Internet Key Exchange Protocol Version 2 (IKEv2)” [RFC9242], so that multiple key exchanges can be run to establish an IKE SA. RFC 9370 also introduces IKE_FOLLOWUP_KE, a new IKEv2 exchange for realizing the same purpose when the IKE SA is being rekeyed or additional Child SAs are created.

[RFC9370] just specifies the framework of hybrid KEMs and has to be instantiated for concrete KEMs by separate documents. [W-D.K25] describes how the framework can be run with ML-KEM [FIPS203], previously called Kyber, which has been standardized by NIST in August 2024. However, for some applications (e.g. financial services) demanding high security level, additional PQ KEMs may be desired for use with [RFC9370]. Currently, ISO is standardizing three PQ KEM algorithms (EDNOTE: we may want to change the wording since the ISO standard will be finished eventually): Kyber, FrodoKEM, and Classic McEliece. Note that FrodoKEM [FrodoKEM] [I-D.LBES25] is an unstructured lattice based KEM, whose security is more conservative compared to ML-KEM based on structured lattices. This specification is a profile of the Multiple Key Exchanges in IKEv2 specification [RFC9370] and registers new algorithm identifiers for FrodoKEM key exchanges in IKEv2.

While this document focuses on using FrodoKEM as an additional key exchange in a hybrid KEM scenario, in some scenarios it is possible to also use FrodoKEM as a quantum-resistant-only key exchange. Since its encapsulation key and ciphertext sizes make UDP packet size larger than typical network MTUs, using FrodoKEM in the IKE_SA_INIT will most probably lead to IP fragmentation of these messages. However, when reliable transport is used for IKE (e.g. [RFC9329], [I-D.ietf-ipsecme-ikev2-reliable-transport]) or IP fragmentation is not an issue in a given network, implementations MAY use FrodoKEM in the IKE_SA_INIT exchange.

EDITOR NOTE: This document is being developed at https://github.com/smyslov/draft-wang-ipsecme-hybrid-kem-ikev2-frodo/.

3. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

4. KEMs and FrodoKEM

4.1. KEMs

Key encapsulation mechanism (KEM) is a kind of key exchange, which allows one entity to encapsulate a secret key under a (long-term or ephemeral) public key of another entity. By following the definiton given in [W-D.K25], a KEM consists of three algorithms:

* KeyGen(k) -> (pk, sk): A probabilistic key generation algorithm, which generates a public encapsulation key pk and a secret decapsulation key sk, when a security parameter k is given.

* Encaps(pk) -> (ct, ss): A probabilistic encapsulation algorithm, which takes as input a public encapsulation key pk and outputs a ciphertext ct and a shared secret ss.

* Decaps(sk, ct) -> ss: A decapsulation algorithm, which takes as input a secret decapsulation key sk and ciphertext ct and outputs a shared secret ss.

4.2. FrodoKEM

The security of FrodoKEM is based on a well-studied hard problem in unstructured lattices, called the learning with errors (LWE) problem. The algorithm details of FrodoKEM are specified in [I-D.LBES25] and [FrodoKEM] .

FrodoKEM [FrodoKEM] has 12 variants. It offers 3 NIST security levels 1, 3, and 5; its pseudorandom generator (PRG) is AES128 or SHAKE 128; and its KEM public key can be a long-term (standard mode) or a short-term ones (ephemeral mode).

In this document, FrodoKEM, rather than eFrodoKEM, is specified for key exchange in IKEv2, based on the following three reasons. First of all, the performance difference between standard mode and ephemeral mode is negligible. Secondly, FrodoKEM (standard mode) has no restriction on the reuse of a public key (Section 8 in [I-D.LBES25]). Finally, ephemeral public keys are expected for the key exchange in IKEv2, but it is not required in [RFC7296] that they never repeat. In fact, Section 2.12 in [RFC7296] describes the reason and several reasonable strategies for reuse of public keys (in the setting of Diffie-Hellman exponentials) in IKEv2, together with conditions and security implication.

FrodoKEM has both AES and SHAKE variants to offer optimized performance across different hardware platforms. AES variants are highly suitable for devices with hardware acceleration for AES (like AES-NI on Intel processors). SHAKE variants provide competitive or better performance on platforms lacking AES hardware acceleration (such as many embedded systems and general-purpose CPUs). To cover both scenarios, this specification include both variants.

According to the current standardization progress in ISO, FrodoKEM will be standardized for the eight variants for NIST security levels 3 and 5. Namely, they are (e)FrodoKEM-976 and (e)FrodoKEM-1344, but not (e)FrodoKEM-640 for security level 1. To align with ISO, this document specifies the use of FrodoKEM varaints for security levels 3 and 5 only, not variants for security level 1.

Based on the above, this document specifies only four variants of FrodoKEM (standard mode) for IKEv2 key exchange. Namely, FrodoKEM-976-<AES> and FrodoKEM-976-<SHAKE> for security level 3, and FrodoKEM-1344-<AES> and FrodoKEM-1344-<SHAKE> for security level 5.

4.3. Comparison to ML-KEM

ML-KEM and FrodoKEM are two well-known post-quantum KEMs based on structured and unstructured lattices. The performance of FrodoKEM is not as good as ML-KEM. As shown in Table 1, the sizes of public encapsulation key and ciphtertext of FrodoKEM (Table A.5 in [FrodoKEM]) are roughly 13 times larger than those of ML-KEM (Table 3 in [FIPS203]). Consequently, this will almost unavoidably trigger IKE fragmentation [RFC7383] [RFC9242], when FrodoKEM is used in IKEv2 as additional key exchange [RFC9370].

Algorithms	decapsulation key sk	encapsulation key pk	ciphterext ct	shared secret ss
ML-KEM-768	2,400	1,184	1,088	32
ML-KEM-1024	3,168	1,568	1,568	32
FrodoKEM-976	31,296	15,632	15,792	24
FrodoKEM-1344	43,088	21,520	21,696	32

Table 1: Size (in bytes) of keys and ciphertexts of ML-KEM and FrodoKEM

5. FrodoKEM in IKEv2

5.1. Recipient Tests

Different from ML-KEM [FIPS203], there is no input validation for FrodoKEM public keys or ciphertexts [I-D.LBES25] [FrodoKEM]. Therefore, the recipient tests are not required for using FrodoKEM in IKEv2.

5.2. FrodoKEM in IKE_INTERMEDIATE

As specified in [RFC9370], to run PQ KEMs in IEKv2, the initiator and the responder run traditional key exchange first and then PQ KEMs. This is because the size of PQ KEM public key or the ciphertext is normally large, such that the first exchange in IKEv2 cannot accommodate them (together with other necessary information) without exceeding MTU (Maximum Transmission Unit), which is generally set as 1500 bytes.

Namely, in the first IKE_SA_INIT exchanges, the initiator sends KEi(0) payload to the responder, and the responder sends KEr(0) payload to the initiator for completing traditional ephemeral DH or ephemeral ECDH key exhange. Once these procedures are done successfully, the two entities will share the same raw key SK(0). And from SK(0), a series of keying materials are derived, which are called as SKEYSEED(0), SK_d(0), SK_a[i/r](0), SK_e[i/r](0), and SK_p[i/r](0) (refers Section 2.2.2 in [RFC9370]).

To run FrodoKEM (or any PQ KEM) as an additional key exchange in IKEv2, both the initiator and the responder MUST declare their support of both the ADDKE Transform Types and the IKE_INTERMEDIATE exchange in the IKE_SA_INIT exchanges between them. At the same time, the initiator SHALL present its intended FrodoKEM variants via one or more ADDKE Transform Types, which are expressed in one or more Proposals. Then, the responder MAY select a variant of FrodoKEM (or more PQ KEMs) from the initiator’s Proposals, and then sends the corresponding ADDKE Transform ID (or IDs) to the initiator.

Once the initiator receives one ADDKE Transform ID, which denotes FrodoKEM (or any PQ KEM), it will run KeyGen(k) to generate an ephemeral public and private FrodoKEM key pairs (pk, sk) or select one of its existing public keys pk, and sends the value of public key pk to the responder via KEi(1) payload. Correspondingly, once retrieving the public key pk from KEi(1) payload, the responder will run Encaps(pk) to obtain a pair (ct1, ss1). Here, ss1 is the raw key to be shared, and ct1 is the ciphertext encapsulated ss1. Then, the responder will send ct1 via KEr(1) payload that contains ct1 to the initiator. After that, the initiator can retrieve ct1 from KEr(1) payload and then decapsulate ct1 to obtain the shared secret ss1. Here, both KEi(1) and KEr(1) payloads SHALL be sent via the IKE_INTERMEDIATE exchanges between the two entities. Also, note that during these procedures, KEi(1) and KEr(1) payloads SHALL be protected via using keys SK_a[i/r](0) and SK_e[i/r](0).

Once ss1 is successfully shared, the two entities will set SK(1)=ss1, and then stir SK(1) with SK_d(0) to derive SKEYSEED(1). And then, from SKEYSEED(1), a series of SK_d(1), SK_a[i/r](1), SK_e[i/r](1),

and SK_p[i/r](1) will be derived. If there are more ADDKE exchanges for PQ KEMs, these procedures will continue until the final ADDKE finishes. Then, the final updated key values, SKEYSEED(n), SK_d(n), SK_a[i/r](n), SK_e[i/r](n), and SK_p[i/r](n), SHALL be used to protect the following IKEv2 exchanges, including the IKEv2 authentication messages.

The structure of KEi(1) and KEr(1) payloads and their lengths for FrodoKEMs listed in Table 1 will be given in Section 5.4.

Following general examples in Appendix A of [RFC9370], here is an example to show that the initiator proposes to use additional key exchanges for establishing an IKE SA. Here, the initiator proposes three sets of additional key exchanges. Namely, the first set is TBD38 (FrodoKEM-976-<AES>), TBD39 (FrodoKEM-976-<SHAKE>) or NONE (refers Section 7); the second set is 36 (ml-kem-768), 37 (ml-kem-1024) [W-D.K25] or NONE; and the third set is TBD41 (FrodoKEM-1344-<SHAKE>) or NONE (refers Section 7). As all of the three additional key exchanges are optional, the responder can choose NONE for some or all of the additional exchanges if the proposed key exchange methods are not supported by the responder, or for whatever reasons the responder decides not to perform the additional key exchange.

```
   Initiator                      Responder
   -------------------------------------------------------------------
   HDR(IKE_SA_INIT), SAi1(...ADDKE*...), --->
   KEi(Curve25519), Ni, N(IKEV2_FRAG_SUPPORTED),
   N(INTERMEDIATE_EXCHANGE_SUPPORTED)
       Proposal #1
       Transform ENCR (ID = ENCR_AES_GCM_16,
                       256-bit key)
       Transform PRF (ID = PRF_HMAC_SHA2_512)
       Transform KE (ID = Curve25519)
       Transform ADDKE1 (ID = TBD38)
       Transform ADDKE1 (ID = TBD39)
       Transform ADDKE1 (ID = NONE)
       Transform ADDKE2 (ID = ml-kem-768)
       Transform ADDKE2 (ID = ml-kem-1024)
       Transform ADDKE2 (ID = NONE)
       Transform ADDKE3 (ID = TBD41)
       Transform ADDKE3 (ID = NONE)

                     <--- HDR(IKE_SA_INIT), SAr1(...ADDKE*...),
                          KEr(Curve25519), Nr, N(IKEV2_FRAG_SUPPORTED),
                          N(INTERMEDIATE_EXCHANGE_SUPPORTED)
                          Proposal #1
                            Transform ENCR (ID = ENCR_AES_GCM_16,
                                            256-bit key)
                            Transform PRF (ID = PRF_HMAC_SHA2_512)
                            Transform KE (ID = Curve25519)
                            Transform ADDKE1 (ID = TBD38)
                            Transform ADDKE2 (ID = ml-kem-768)
                            Transform ADDKE3 (ID = NONE)

   HDR(IKE_INTERMEDIATE), SK {KEi(1)(TBD38)} -->
                     <--- HDR(IKE_INTERMEDIATE), SK {KEr(1)(TBD38)}
   HDR(IKE_INTERMEDIATE), SK {KEi(2)(ml-kem-768)} -->
                     <--- HDR(IKE_INTERMEDIATE), SK {KEr(2)(ml-kem-768)}

   HDR(IKE_AUTH), SK{ IDi, AUTH, SAi2, TSi, TSr } --->
                     <--- HDR(IKE_AUTH), SK{IDr, AUTH, SAr2,TSi, TSr}
```

Figure 1: Hybrid KEMs of ECDH, TBD38 (FrodoKEM-976-<AES>), and ml-kem-768

In the above example, the responder chooses to run two additional key exchanges. Namely, it selects TBD38 (FrodoKEM-976-<AES>), 36 (ml-kem-768), and NONE, respectively for the first, second, and third additional key exchanges. According to the IKEv2 specification [RFC7296], a set of keying materials will be derived, in particular SK_d, SK_a[i/r], and SK_e[i/r], when the IKE_SA_INIT exchange has

been completed by the initiator and the responder with a successful execution of ECDH based on the curve 25519. After that, both peers will perform an IKE_INTERMEDIATE exchange, carrying TBD38 payload, which is protected with SK_e[i/r] and SK_a[i/r] keys. After the completion of this IKE_INTERMEDIATE exchange, the SKEYSEED is updated using SK(1), which is the TBD38 shared secret. Next, an IKE_INTERMEDIATE exchange for 36 payload will be performed so that the SKEYSEED will be updated again.

After the completion of both IKE_INTERMEDIATE exchanges for TBD38 and 36, the initiator and the responder will continue the IKE_AUTH exchange phase.

Note that similar to the above example running FrodoKEM in IKE_INTERMEDIATE with ECDH as a traditional key exchange, FrodoKEM can also be executed in IKE_INTERMEDIATE as an additional (PQ) KEM with a Post-quantum Preshared Keys (PPK) as a traditional key exchange. In this case, the traditional exchange part with PPK SHOULD be implemented as specified by [RFC8784] or [RFC9867].

5.3. FrodoKEM in IKE_FOLLOWUP_KE

FrodoKEM can also be used for creating additional Child SAs and rekeying the IKE SA or Child SAs. FrodoKEM may be used as the only key exchange in CREATE_CHILD_SA exchange or as an additional key exchange method. In the latter case, the IKE_FOLLOWUP_KE exchange as defined in [RFC9370] is used.

IKE_FOLLOWUP_KE is an additional exchange for the purpose of using multiple key exchanges with the CREATE_CHILD_SA Exchange. If the use of additional key exchange methods is negotiated in the CREATE_CHILD_SA exchange, these are performed subsequently in a series of IKE_FOLLOWUP_KE exchanges. After all key exchanges are completed, SKEYSEED or KEYMAT are computed as specified in Section 2.2.4 of [RFC9370].

5.4. IKEv2 Payloads for FrodoKEM

For completeness, the KE (Key Exchange) payload is given below and all fields inside keep the same meaning as specified in Section 3.4 of the IKEv2 standard [RFC7296]. This also means that the initiator SHALL prepare KEi(0) and KEi(1) payloads according to Figure 2. Namely, the Key Exchange Data will be filled with KEi(0) or KEi(1). This applies for the responder to prepare KEr(0) and KEr(1) as well.

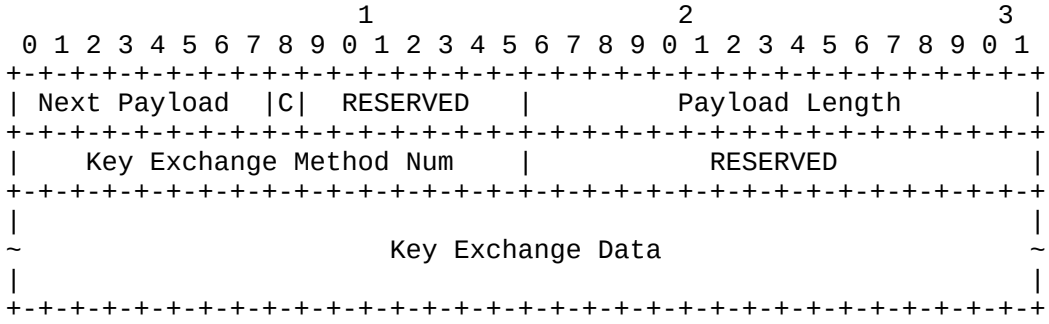

Figure 2: Key Exchange Payload

Table 2 lists the lengths in octets for the KE payload with four variants of FrodoKEM specified in this document.

KE Method No.	KEM	Payload Length (for pk/ct)	Data Size in octets (KEi/ KEr)
TBD38	FrodoKEM-976-<AES>	15,640/15,800	15,632/15,792
TBD39	FrodoKEM-976-<SHAKE>	15,640/15,800	15,632/15,792
TBD40	FrodoKEM-1344-<AES>	21,528/21,704	21,520/21,696
TBD41	FrodoKEM-1344-<SHAKE>	21,528/21,704	21,520/21,696

Table 2: Lengths of Key Payload for 4 variants of FrodoKEM

6. Security Considerations

Basically, security considerations from [RFC7383], [RFC9242] and [RFC9370] apply to key exchange of ECDH, FrodoKEM, and their hybrid described in this specification.

The security discussions in Section 10 of [W-D.K25] apply here as well. First of all, FrodoKEM is designed to be a post-quantum KEM with IND-CCA2 security. Namely, it has indistinguishability under adaptive chosen ciphertext attacks, which means that shared secret values should be indistinguishable from random strings even an active attacker is given the ability to access arbitrary ciphertexts decapsulated.

Next, generating a new ciphertext for each FrodoKEM key exchange is REQUIRED by this specification. However, it is OPTIONAL for generating a new ephemeral key pair for each FrodoKEM key exchange, as IKEv2 [RFC7296] does not require that the public keys never repeat. Note that when the same FrodoKEM public key is resued to encapsulate multiple shared secrets, forward security may not be guaranteed, as the compromise of the corresponding private decapsulation key may lead to the compromise of shared secrets exchanged in previous sessions using the same encapsulation key. Section 2.12 in [RFC7296] gives the reasons for reuse of a public key and discusses the corresponding security implications.

Thirdly, the independency is REQUIRED among the randomness for generating FrodoKEM encapsulation key and for generating ciphertexts, and other random seeds used in IKEv2 negotiation. For example, the responder MUST NOT reuse the same randomness to generate multiple FrodoKEM ciphertexts, and the initiator also MUST NOT use the same seed to generate the FrodoKEM and (EC)DH keypairs in a PQ/T Hybrid key exchange.

Finally, downgrade attacks on the authentication part of IKEv2 has been identified and repaired in "Prevention Downgrade Attacks on the Internet Key Exchange Protocol Version 2 (IKEv2)" [W-D.SP25]. Due to a flaw without authenticating the whole message received from the other peer, these attacks may allow an active attacker to mislead the two peers to finally negotiating a weak KEM for key exchange. These attacks apply to the IKEv2 [RFC7296] and all its extensions, including [RFC9370]. So, this specification MUST be implemented with the updated authentication mechanism given by [W-D.SP25].

7. IANA Considerations

As specified in Section 4.2, this draft is to asking 4 values for registration in the "Transform Type 4 - Key Exchange Method Transform IDs" registry [IANA-IKEv2], maintained by IANA. Namely, they are: "FrodoKEM-976-<AES>", "FrodoKEM-976-<SHAKE>", "FrodoKEM-1344-<AES>", and "FrodoKEM-1344-<SHAKE>".

Table 3 below gives the list of 4 IANA values for the 4 versions of FrodoKEM (standard mode).

Number	Name	Status	Recipient Tests	Reference
TBD38	FrodoKEM-976-<AES>		[TBD, this draft, Section 5.1]	[TBD, this draft]
TBD39	FrodoKEM-976-<SHAKE>		[TBD, this draft, Section 5.1]	[TBD, this draft]
TBD40	FrodoKEM-1344-<AES>		[TBD, this draft, Section 5.1]	[TBD, this draft]
TBD41	FrodoKEM-1344-<SHAKE>		[TBD, this draft, Section 5.1]	[TBD, this draft]

Table 3: Updates to the IANA "Transform Type 4 - Key Exchange Method Transform IDs"

8. Acknowledgments

The authors would like to thank the following experts for their valuable comments: Scott Fluhrer, Christopher Patton, Kev Kitchen, Panos Kampanakis, Paul Wouters, Thom Wiggers, Michael Richardson, John Mattsson, Marc Penninga, Tirumal Reddy, and Jun Hu.

9. References

9.1. Normative References

[I-D.LBES25]
Longa, P., Bos, J. W., Ehlen, S., and D. Stebila, "FrodoKEM: key encapsulation from learning with errors", Work in Progress, WG Document, September 2025, <https://datatracker.ietf.org/doc/draft-longa-cfrg-frodokem/>.

[IANA-IKEv2] "Internet Key Exchange Version 2 (IKEv2) Parameters", the Internet Assigned Numbers Authority (IANA). , <https://www.iana.org/assignments/ikev2-parameters/ikev2-parameters.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/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>.

[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>.

[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>.

[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>.

[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>.

[RFC9867] Smyslov, V., "Mixing Preshared Keys in the IKE_INTERMEDIATE and CREATE_CHILD_SA Exchanges of the Internet Key Exchange Protocol Version 2 (IKEv2) for Post-Quantum Security", RFC 9867, DOI 10.17487/RFC9867, November 2025, <https://www.rfc-editor.org/info/rfc9867>.

9.2. Informative References

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

[FrodoKEM] Alkim, E., Bos, J. W., Ducas, L., Longa, P., Mironov, I., Naehrig, N., Nikolaenko, V., Peikert, C., Raghunathan, A., and D. Stebila, "FrodoKEM: Learning With Errors Key Encapsulation", Preliminary Standardization Proposal submitted to ISO , December 2024, <https://frodokem.org/files/FrodoKEM_standard_proposal_20250929.pdf>.

[I-D.ietf-ipsecme-ikev2-reliable-transport] Smyslov, V. and T. Reddy.K, "Separate Transports for IKE and ESP", Work in Progress, Internet-Draft, draft-ietf-ipsecme-ikev2-reliable-transport-00, 6 October 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-ipsecme-ikev2-reliable-transport-00>.

[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>.

[W-D.K25] Kampanakis, K., "Post-quantum Hybrid Key Exchange with ML-KEM in the Internet Key Exchange Protocol Version 2 (IKEv2)", Work in Progress, Internet-Draft (Group Documet of IPSECME, IETF), October 2025, <https://datatracker.ietf.org/doc/draft-ietf-ipsecme-ikev2-mlkem/>.

[W-D.SP25] Smyslov, V. and C. Patton, "Prevention Downgrade Attacks on the Internet Key Exchange Protocol Version 2 (IKEv2)", Work in Progress, Internet-Draft (Group Documet of IPSECME, IETF), November 2025, <https://datatracker.ietf.org/doc/draft-ietf-ipsecme-ikev2-downgrade-prevention/>.

Authors' Addresses

Guilin Wang (editor)
Huawei Int. Pte Ltd
9 North Buona Vista Drive, #13-01
The Metropolis Tower 1
SINGAPORE 138588
Singapore
Email: wang.guilin@huawei.com

Leonie Bruckert
secunet Security Networks
Ammonstr. 74
01067 Dresden
Germany
Email: Leonie.Bruckert@secunet.com

Valery Smyslov
ELVIS-PLUS
PO Box 81
Moscow (Zelenograd)
124460
Russian Federation
Phone: +7 495 276 0211
Email: svan@elvis.ru

Meiling Chen
China Mobile
BeiJing
China
Email: chenmeiling@chinamobile.com