

IP Security Maintenance and Extensions
Internet-Draft
Intended status: Standards Track
Expires: 27 June 2026

G. Wang, Ed.
Huawei Int. Pte Ltd
L. Bruckert
secunet Security Networks
V. Smyslov
ELVIS-PLUS
24 December 2025

Post-quantum Hybrid Key Exchange in IKEv2 with FrodoKEM
draft-wang-ipsecme-hybrid-kem-ikev2-frodo-03

Abstract

Multiple key exchanges in the Internet Key Exchange Protocol Version 2 (IKEv2) [RFC9370] specifies a framework, which supports multiple key encapsulation mechanisms (KEMs) in IKEv2 by allowing up to 7 layers of additional KEMs to derive the final shared secret keys for IPsec protocols. The primary goal is to mitigate the “harvest now and decrypt later” threat posed by Cryptographically Relevant Quantum Computers (CRQCs). For this purpose, one or more post-quantum KEMs are usually performed in addition to the traditional (EC)DH key exchange. This draft specifies how the post-quantum KEM algorithm FrodoKEM is instantiated for IKEv2 as an additional key exchange mechanism.

[EDNOTE: IANA KE code points for FrodoKEM may need to be assigned, as the code points for ML-KEM has been considered in [W-D.K25].]

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 27 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	2
1.1. Notes of Change	2
1.2. Introduction	3
2. Requirements Language	4
3. KEMs and FrodoKEM	4
3.1. KEMs	5
3.2. FrodoKEM	5
3.3. Comparison to ML-KEM	6
4. FrodoKEM in IKEv2	6
4.1. FrodoKEM in IKE_INTERMEDIATE	7
4.2. FrodoKEM in IKE_FOLLOWUP_KEY	10
4.3. IKEv2 Payloads for FrodoKEM	10
5. Security Considerations	11
6. IANA Considerations	11
7. Acknowledgments	12
8. Normative References	12
9. Informative References	13
Authors' Addresses	14

1. Introduction

1.1. Notes of Change

Changes made in version 03:

- * This specification has switched 4 variants of eFrodoKEM (ephemeral mode) to those of FrodoKEM (standard mode). The reasons are given in Section 3.2.
- * Other Sections are updated correspondingly.

Three main changes have been made in version 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 01, as a response to comments received at 122 meeting:

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

1.2. Introduction

Cryptographically-relevant quantum computers (CRQCs) pose a threat to cryptographically protected data. 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. "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 to take place for deriving shared secret keys during a Security Association (SA) setup. Essentially, this framework employs the IKE_INTERMEDIATE exchange, which is a new IKEv2 message 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 via exchanging additional PQ public keys and ciphertexts between a client and a server. 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.

However, [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 the ML-KEM [FIPS203], previously called Kyber, which has been standardized by NIST in August 2024. However, on the one hand, RFC 9370 allows up to 7 layers of additional KEMs to derive final shared secret keys in IKEv2. On the other hand, 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, which is based on structured lattice. Therefore, this specification is motivated to describe concretely how the framework of hybrid KEMs for IKEv2 specified in RFC 9370 can be instantiated with FrodoKEM. FrodoKEM should be used together with a traditional key exchange mechanism such as ECDH and in addition, may be used with other KEMs, e.g. ML-KEM.

Here are a few reasons for explaining why such diversity of KEMs is important for IKEv2 (and also other security protocols).

- * The availability of various PQ algorithms is beneficial to applications as different PQ algorithms could be selected according to practical performance and security requirements.
- * Generally speaking, post-quantum algorithms are still not mature yet. Some algorithms may turn out to be insecure after a number of years' study and/or standardization. An example is SIKE, which had been in the NIST standardization progress for several years until it was totally broken in July of 2022 [CD22].
- * Cryptographic agility shall play a crucial role in the PQ migration [OPM23]. To facilitate cryptographic agility, not only should the systems and protocols be engineered agile but also there should be a good number of standardized PQC algorithms available, which may be based on different hard problems.

However, the performance of FrodoKEM is not as good as ML-KEM. In particular, the sizes of public key and ciphertext of FrodoKEM are roughly 13 times larger than those of ML-KEM. Consequently, this will almost unavoidably trigger IKE fragmentation.

2. 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.

3. KEMs and FrodoKEM

3.1. 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 definition 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`.

3.2. FrodoKEM

FrodoKEM [I-D.LBES25] is one of three KEMs in the process of ISO standardization [FrodoKEM]. Its security is based on a well-studied hard problem in unstructured lattices, called the learning with errors problem. The algorithm details of FrodoKEM are specified in [I-D.LBES25].

In total, FrodoKEM [FrodoKEM] has 12 variants. Namely, it offers 3 NIST security levels 1, 3, and 5; the pseudorandom generator (PRG) uses AES128 or SHAKE 128; and the KEM public key can be a long-term key (standard mode) or a short-term key (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, the recommendation of using FrodoKEM (Section 8 in [I-D.LBES25]) also allows to use FrodoKEM (standard mode) with ephemeral public keys. 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 SHALL include both variants.

According to the current standardization progress in ISO, FrodoKEM will be standardized for only 8 variants for NIST security levels 3 and 5. Namely, there are (e)FrodoKEM-976 and (e)FrodoKEM-1344, but not (e)FrodoKEM-640 for security level 1. To align with ISO, this specification specifies the use of FrodoKEM variants 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.

3.3. Comparison to ML-KEM

ML-KEM and FrodoKEM are two well-known post-quantum KEMs based on lattices. More specifically, ML-KEM [FIPS203], originally called Kyber, has been standardized as the only one KEM scheme by NIST in August of 2024. It is a Module-Lattice based Key-Encapsulation Mechanism, so called ML-KEM.

However, the performace of FrodoKEM is not as good as ML-KEM. Specifically, 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.

Algorithms	decapsulation key sk	encapsulation key pk	ciphtertext 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

4. FrodoKEM in IKEv2

4.1. FrodoKEM in IKE_INTERMEDIATE

As specified in [RFC9370], to run PQ KEMs in IEKv2, the initiator and the responder SHALL 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 in IKEv2, 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 exchange. Once these procedures are done successfully, the two entities SHALL 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) (refere to section 2.2.2 in [RFC9370]).

To run FrodoKEM (or any PQ KEM) 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 presented 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 is for FrodoKEM (or any PQ KEM), it SHALL run KeyGen(k) to generate an ephemeral public and private KEM key pairs (pk, sk), and sends the value of public key pk to the responder via KEi(1) payload. Correspondingly, once retrived public key pk from KEi(1) payload, the responder SHALL run Encaps(pk) to obtain a pair (ct1, ssl). Here, ssl is the raw key to be shared, and ct1 is the ciphertext encapsulated ssl. Then, the responder SHALL 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 sceret ssl. 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 ssl is successfully shared, the two entities SHALL set SK(1)=ssl, 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) SHALL be derived. If there are more ADDKE exchanges for PQ KEMs, these procedures SHALL 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, in particular 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 4.3.

Following general examples given 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 (refer to Section 6); 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 (refer to Section 6). 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 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 = 36)	
Transform ADDKE2 (ID = 37)	
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 = 36)
	Transform ADDKE3 (ID = NONE)
HDR(IKE_INTERMEDIATE), SK {KEi(1)(TBD38)} -->	
	<--- HDR(IKE_INTERMEDIATE), SK {KEr(1)(TBD38)}
HDR(IKE_INTERMEDIATE), SK {KEi(2)(36)} -->	
	<--- HDR(IKE_INTERMEDIATE), SK {KEr(2)(36)}
HDR(IKE_AUTH), SK{ IDi, AUTH, SAI2, TSi, TSr } --->	
	<--- HDR(IKE_AUTH), SK{IDr, AUTH, SAR2, TSi, TSr}

Fig. 1 Hybrid KEMs of ECDH, TBD38 (FrodoKEM-976-<AES>), and 36 (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 can 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.

4.2. 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].

4.3. 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 Fig. 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.

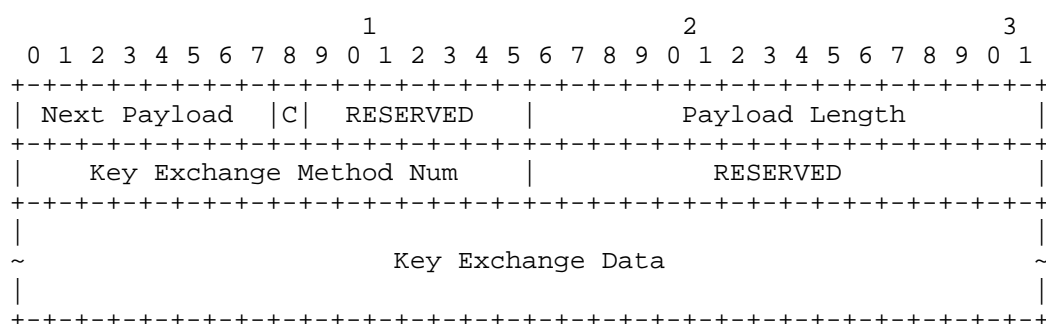


Fig. 2 Key Exchange Payload

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.

5. Security Considerations

Basically, security considerations from [RFC7383], [RFC9242] and [RFC9370] apply to hybrid KEM exchange of ECDH, ML-KEM, and FrodoKEM described in this draft.

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 lead an attacker to misleading the two peers to finally negotiate a weak KEM for key exchange. These attacks apply to the IKEv2 [RFC7296], [RFC9370], and all extensions. So, this specification MUST be implemented with the updated mechanism for authentication given by [W-D.SP25].

6. IANA Considerations

As specified in Section 3.2, 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 2 below gives the list of 4 IANA values for the 4 versions of FrodoKEM (standard mode). The Recipient Tests field should point to this document as well.

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

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

7. Acknowledgments

The authors would like to thank Scott Fluhrer, Christopher Patton, and Kev Kitchen for their valuable comments.

8. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/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/info/rfc8174>>.
- [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>>.

- [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, C.J., 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>>.
- [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>>.
- [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_20241205.pdf>.

9. Informative References

- [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 Document of IPSECME, IETF), October 2025, <<https://datatracker.ietf.org/doc/draft-ietf-ipsecme-ikev2-mlkem/>>.
- [OPM23] Ott, D., Paterson, K., and D. Moreau, "Where Is the Research on Cryptographic Transition and Agility?", Communications of the ACM, 66(4): 29-32, January 2023.

[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/>>.

[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](https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.203.pdf)>.

[CD22]

Castruck, W. and T. Decru, "An Efficient Key Recovery
Attack on SIDH", Formal version published in the
proceedings of EUROCRYPT 2023 , July 2022,
<<https://eprint.iacr.org/2022/975>>.

[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 Document of
IPSECME, IETF), November 2025,
<[https://datatracker.ietf.org/doc/draft-ietf-ipsecme-
ikev2-downgrade-prevention/](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
Russian Federation

Email: smyslov.ietf@gmail.com