IP Security Maintenance and Extensions
Internet-Draft
Intended status: Standards Track
Expires: 8 January 2026

G. Wang, Ed.
Huawei Int. Pte Ltd

7 July 2025

KEM based Authentication for the IKEv2 with Post-quantum Security
draft-wang-ipsecme-kem-auth-ikev2-01

Abstract

This draft specifies a new authentication mechanism, called KEM based authentication, for the Internet Key Exchange Protocol Version 2 (IKEv2) [RFC7296]. This is motivated by the fact that ML-KEM is much more efficient that ML-DSA, which are the post-quantum algoirhtms for mitigating the pontential security threats again quantum computers. The KEM based authenticationth for the IKV2 is achieved via introduing a new value of the IKEv2 Authentication Method registry mantained by IANA. For using the new authentication method, two peers MUST send the SUPPORTED_AUTH_METHODS Notify, defined by [RFC9593], to negotiate the supported KEM algorithms. After that, the correponding KEM certificates and cipthertext are exchanged via the INTERMEDIATE Exchange. Finally, the IKE messages are authenticated via the shared secret encapsulated between the two peers. This documents also specifies the instantiation with ML-KEM for this new general authenticaiton method for the IKEv2.

[EDNOTE: Code points for KEM-based authentication may need to be assigned in the IKEv2 Authenticaion Method registry, maintained by IANA]

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 8 January 2026.

Table of Contents

1. Introduction

1.1. Notes of Change

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

* More details abouth how each side does for running KEM authentication in Section 5.1.

* Added Section 5.2 for KEM authentication with preshared public key.

1.2. Introduction

Cryptographically-relevant quantum computers (CRQC) 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 again the Internet Key Exchange Protocol Version 2 (IKEv2) [RFC7296], multiple key exchanges in the IKEv2 [RFC9370] was introduced to achieve post-quantum (PQ) security for the exchange. To offering post-quantum security for the authentication in the IKEv2, "Announcing Supported Authentication Methods in the Internet Key Exchange Protocol Version 2 (IKEv2)" [RFC9593] specifies a new Notify type, called the SUPPORTED_AUTH_METHODS, which allows two peers to indicate the list of supported authentication methods while establishing IKEv2 SA. One purpose of [RFC9593] is to support post-quantum signature algorithms for authenticaion in the IKEv2, as further described by the following drafts.

"Signature Authentication in the Internet Key Exchange Version 2 (IKEv2)" [I-D.RSF25] specifies how NIST PQ digital algorithms ML-DSA [FIPS204] and SLH-DSA [FIPS205] can be used in the IKEv2 by indicating the supported signature algorithms via exchanging the Notify SIGNATURE_HASH_ALGORITHMS, defined in [RFC7427]. Similarly, "IKEv2 Support of ML-DSA" [I-D.Flu25] specifies how ML-DSA can be run in the IKEv2, by indicating the supported signature algorithms via exchanging the SUPPORTED_AUTH_METHODS Notify, defined in [RFC9593]. On the other hand, "Post-Quantum Traditional (PQ/T) Hybrid PKI Authentication in the Internet Key Exchange Version 2 (IKEv2)" [I-D.HMW] specifies how to run general hybrid PQ/T digital algorithms in the IKEv2 via introducing some extensions in the SUPPORTED_AUTH_METHODS Notify.

For all of those Internet standard drafts, it is the same that the corresponding public key certificates and singatures for the involved siganture algorithms are exchanged via the INTERMEDIATE Exchange, defined in [RFC9242].

Motivated by the fact that ML-KEM [FIPS203] is much more efficient than ML-DSA [FIPS204], "KEM-based Authentication for TLS 1.3" [I-D.WCS] [I-D.WCSSS] specifies how KEM can be used to achieve post-quantum secure authentication for the TLS 1.3 protocol. The basic idea is as follows: when one entity A receives the certified long term public key of another entity B, A can authenticate B by encapsulating a secret key k using B’s KEM public key, and confirming that the communicating entity is indeed B if the entity can successfully return the correct k to A. The seminal idea for TLS is presented in [SSW20], followed by a number of research papers then. Besides saving communication overhead and computational time, as

pointed out in [I-D.WCS], KEM based authentication also benefits to reduce implementation code size, as the code for PQ singature may not need, and KEM based authentication can re-use the KEM code for ephemeral key establishment.

This draft specifies how to realise KEM based authentication for the Internet Key Exchange Protocol Version 2 (IKEv2) [RFC7296]. This is achieved by defining a new authenication method, called KEM based Authentication. Namely, this draft asks to add a new value of the "IKEv2 Authenticaion Method" registry [IANA-IKEv2], mantained by IANA. To run this new authenticaiton method, two peers MUST send the SUPPORTED_AUTH_METHODS Notify, defined by [RFC9593], in the IKE_SA_INIT Exchange, to negotiate the supported KEM algorithms. After that, the correponding KEM certificates and cipthertext are exchanged via the INTERMEDIATE Exchange [RFC9593]. This documents also specified the instantiation with ML-KEM for this new general authenticaiton method for the IKEv2.

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. Key Encapsulation Mechanism and Digital Signature

Key encapsulation mechanism(KEM) is a kind of key exchange, which allows one entity to encapsulate a secret key under a (longterm or ephemeral) public key of another entity. By following the definiton given in [I-D.KR24], 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 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.

By following the definiton given in [I-D.BHCD], a signature scheme consists of the following three algorithms:

* SKeyGen(k) -> (spk, ssk): A probabilistic key generation algorithm, which generates a public verifying key spk and a private signing key ssk, when a security parameter k is given..

* Sign(ssk, m) -> s: A deterministic or probabilistic signature generation algorithm, which takes as input a private signing key ssk and a message m, and outputs a signature s.

* Verify(spk, s, m) -> b: A deterministic verification algorithm, which takes as input a public verifying key spk, a signature s and a message m, and outputs a bit b indicating accept (b=1) or reject (b=0) of the signature-message pair (s, m).

In August of 2024, NIST released ML-KEM [FIPS203] and ML-DSA [FIPS204], which are both module-lattice based post-quantum algorithms. Each of both algorithms has three sets of parameters corresponding to three NIST security levels. Namely, ML-KEM-512 for Level 1, ML-KEM-768 for Level 3, and ML-KEM-1024 for for Level 5, while ML-DSA-44 for Level 2, ML-DSA-65 for Level 3, and ML-DSA-87 for Level 5.

4. Comparison of ML-KEM and ML-DSA

This section compares ML-KEM and ML-DSA, with respect to the sizes of key, ciphertext and signature, and also the computational overhead of key generation, encapsulaion, signing, decasulation, and verifying.

First of all, to compare the communication overhead when using ML-DSA and ML-KEM, data from Table 2 in [FIPS204] and Table 3 in [FIPS204] are used to generate the following Table 1. The results show the comparison of keys and signature over ciphertext between ML-DSA and ML-KEM. Namely, for all corresponding security levels, the following statements can be conluded:

* The private key size of ML-DSA is about 1.5 times of the decapsulation key size of ML-KEM,

* The public size of ML-DSA is about 1.6 times of that of ML-KEM, and

* The signature size of ML-DSA is about 3 times of the ciphertext size of ML-KEM.

	private key/ decapaaulation key	public key/ encapsulation key	singature/ ciphertext
ML-DSA-44/ ML-KEM-512	2,650/1,632 =157%	1,312/800 =164%	2,420/768 =315%
ML-DSA-65/ ML-KEM-768	4,032/2,400 =168%	1,952/1,184 =165%	3,309/1,088 =304%
ML-DSA-87/ ML-KEM-1024	4,896/3,168 =155%	2,592/1,568 =165%	4,627/1,568 =295%

Table 1: Communication Overhead Comparison of ML-DSA and ML-KEM

Specifically, for the three security levels, when ML-KEM is used to replace ML-DSA for authentication, the saved communication overhead, namely public key+signature for ML-DSA and encapsulation key+ciphertext for ML-KEM, is 2,164, 2,989, and 4,083 bytes, respectively. Those savings are helpful to improve the performance of IKEv2. In fact, as shown in the experiment in simulated environment, the average time delay for IKEv2 can increase a few times due to large size of PQ public key, ciphertext and signature, especially when the IP packet loss rate reaches about 1%.

Next, the computation overhead comparison will be given now. In [HAZ24], the authors present their implementation results of Dilithium and Kyber, via various optimization techniques for Keccak and the two PQC algorithms. Concretely, Table 6 in [HAZ24] shows the performance comparison of Dilithium and Kyber on ARMv7 Cortex-M4 processor, presented by clock cycles. By ignoring the small difference between ML-KEM and its informal verion Kyber, also ML-DSA and its informal verion Dilithium, the followin Table 2 is obtained.

	KeyGen speed/ SKeyGen speed	Encap speed/ Sign speed	Decap speed/ Verify speed
ML-DSA-44/ ML-KEM-512	1,357k/369k =368%	3,487k/448k =778%	1,350k/409k =330%
ML-DSA-65/ ML-KEM-768	2,394k/604k =396%	5,574k/732k =761%	2,302k/674k =342%
ML-DSA-87/ ML-KEM-1024	4,069k/962k =423%	7,730k/1,119k =691%	3,998k/1,043k =383%

Table 2: Computational Overhead Comparison of ML-DSA and ML-KEM

By assuming that the copmtuational comparison shown in Table 1 reasonably presents the performance difference of ML-KEM and ML-DSA at the same platform with similar implementation techniques, for all three corresponding security levels, the following statements can be derived:

* The private key generation time of ML-DSA is about 4 times of that of ML-KEM.

* The signature signing time of ML-DSA is about 7 times of the encapsulation speed of ML-KEM.

* The signature verifying time of ML-DSA is over 3 times of the decapsulation speed of ML-KEM.

In the scenario of KEM based authentication, both the private keys for ML-DSA and ML-KEM will be long term keys, so the private key generation time can be ignored, as it is just one time overhead. Therefore, by combining signature signing and verifying together, and also combining encapsulation and decapsulation together, we can simply say that the computaional time of ML-DSA is about 5 times of that of ML-KEM.

5. Protocl Details for KEM based Authentication

By following [RFC9593], two communicating peers send ech other the Notify Message Type SUPPORTED_AUTH_METHODS to negotiate which authentication method will be used to authenticate them to each other. Bascially, the authentication method can be any one registered in the "IKEv2 Authentication Method" registry under "Internet Key Exchange Version 2 (IKEv2) Parameters" [IANA-IKEv2], maintained by IANA. To run KEM based Authentication, the draft is

supposed to apply the value of 16 (TBD) for "KEM based Authentication" in the "IKEv2 Authentication Method" registry (Section 7).

5.1. Exchanges for KEM based Authentication

After the initiator starts the IKE_SA_INIT exchange as usual, the responder sents the notify SUPPORTED_AUTH_METHODS with value of 16 (TBD) to indicate that the responder wants to run KEM based Authentication with respect to some specific KEM algorithms, which the responder supports. These KEMs will be listed in the SUPPORTED_AUTH_METHODS Notify Payload (Section 5.3), ordered by the responder’s preference, among other possible authentication methods.

After the initiator receives SUPPORTED_AUTH_METHODS from the resonder, it will select the KEM algorithm from the list of KEMs sent by the responder that has the highest preference and is supported by the initator as well. After that, the responder SHALL use this KEM algorithm to authenticate itself to the initiator.

Fig. 1 below gives an example to show how two peers use the SUPPORTED_AUTH_METHODS notification to run KEM based authentication. In the protocol, the IKE_INTERMEDIATE exchange may be used to faciliate the hybrid key exchange in the IKEv2 as specified in [RFC9370], and to transfer PQ certifates between the responder and the intitator for completing KEM based authentication.

Once the responder and the inititator have negotiated to run KEM based authentication, shown as the value of 16 in the SUPPORTED_AUTH_METHODS notification, one specific KEM algorithm SHALL be selected by the initiator. After that, both parties SHALL encapsulate a shared secret under the public key of the other party and send out the resulting ciphertext. Then, the peer SHALL decapsualte the ciphertext received to obtain the shared secret key encapsulated by the other party. Next, the AUTH data SHALL be calculated according to the specification [RFC7296] via using the MAC algorithm selected. Finally, once each party successfully verifies the MAC code for the AUTH data received from the other party, the whole IKE key exchange and authentication is successful.

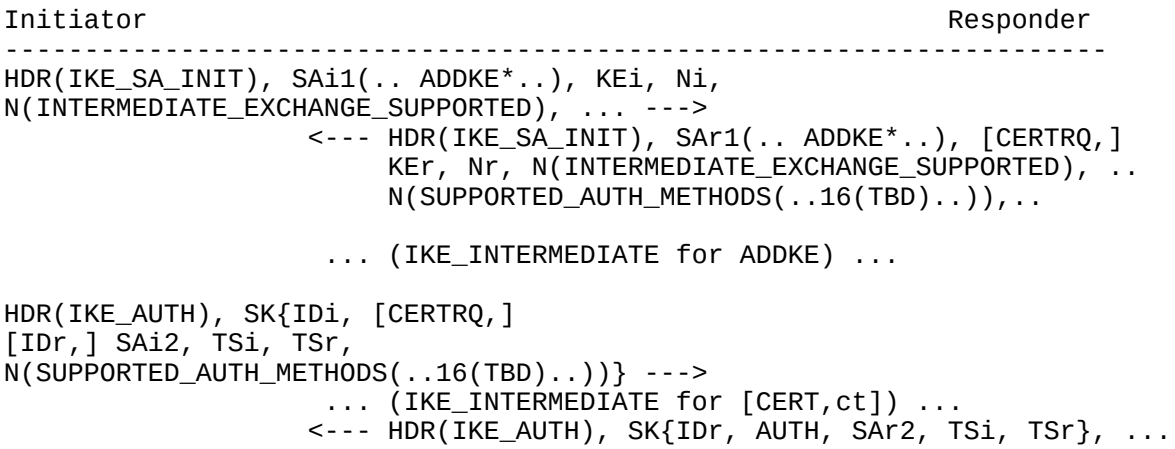

Fig. 1 An Example of Running KEM based Authentication between Two Peers

If the resulting SUPPORTED_AUTH_METHODS notification with list of autehnticiton methods is too long such that IP fragmentation [RFC7383] of the IKE_SA_INIT response may happen, the responder MAY choose to send empty SUPPORTED_AUTH_METHODS notification in the IKE_SA_INIT exchange response. Then, the responder and the intiatior can send ech other the SUPPORTED_AUTH_METHODS notification with list of authentictatin methods they support by using the IKE_INTERMEDIATE exchange, as desribed in Section 3.1 of [RFC9593].

[EDNOTE: More examples may be provided later.]

5.2. KEM based Authentication with Preshared Public Key

In Section 5.1, the protocol may need to run one additional round as KEM based authentication has to first send out one’s KEM public key and the associated certificate, so that the other party can return back an encapsualted secret as a challenge, before finally releasing the MAC for the AUTH data. This is naturally not as efficient as signature authentication, which can be sent out in one message consisting of message being authenticated, the digital signarure, and also the corresponing public key certificate used to verify the singature.

However, the situation will be changed when both parties have already acquied each other’s public key (and the associated certificate) before running KEM based authentication. Such a public key may be pre-installed, cached, or provisoned via out-bound ways. For TLS authentication, the situation has been specified in [I-D.WCSSS]. Tnhis KEM based authentication with preshared public key is expected to be useful in scenarios where one or both parties have conctrained capabilities, e.g, IoT devices.

[EDNOTE: An example may be provided later.]

5.3. Payload Format for KEM based Authentication

For easy reference, the SUPPORTED_AUTH_METHODS Notify payload format is shown in the following, as specified in Section 3.2 of [RFC9593]. Correspondigly, here, Protocol ID field MUST be set to 0, the SPI Size MUS be set to 0 (meaning there is no SPI field), and the Notify Message Type MUST set to 16443.

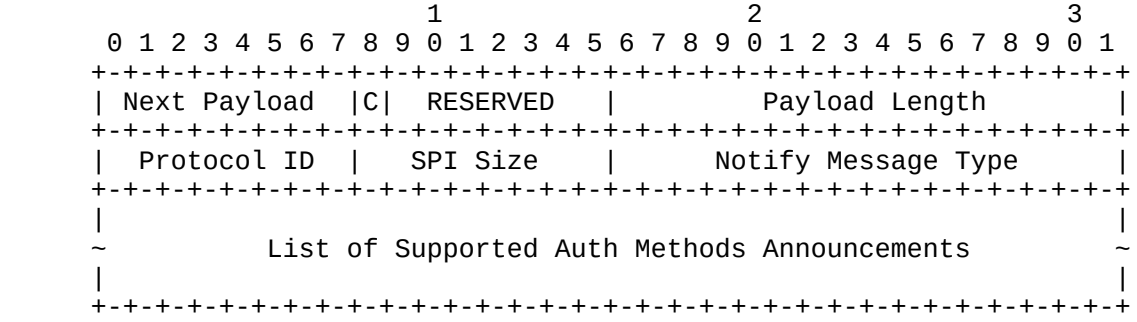

Fig.2 SUPPORTED_AUTH_METHODS Notify Payload Format

Payload Format for KEM based Authentication Announcement is defined in Fig. 3, which is treated as part of the Supported Auth Methods Announcements shown in Fig. 2. Namely, for this part, a number (N) of KEM based authentication methods are listed, as desribed below.

* Length: Length of the whole blob of one announcement in octets; must be greater than 5.

* Auth Method: Announced authentication method, which is supposed to 16 (TBD), standing for "KEM based authentication" for the IKEv2.

* Cert Link: Links this announcement with a particular CA, which issued the KEM certificate for the KEM algorithm identified in AlgorithmIdentifiers below; see Section 3.2.2 of [RFC9593] for detail.

* Alg Len 1: Length of the KEM algorithm idenfied in AlgorithmIdentifiers below, in octets.

* Alg Len 2: Length of the keyed MAC algorithm idenfied in AlgorithmIdentifiers below, in octets.

* AlgorithmIdentifiers: The concatenation of two variable-length ASN.1 objects that are encoded using Distinguished Encoding Rules (DER) [X.690] and identify one specific KEM aglorithm and one keyed MAC algorithm. Those two ASN.1 objects are combined here temporary for this instance of KEM-based authenticaion.

For example, if the concatenation of ML-KEM-768 and SHAKE128 (denoted as ML-KEM-768+SHAKE128 for simplicity), this means that ML-KEM-768 will be used to encapsulate and decapsulate the shared secret ss in KEM-based authenticaion, while SHAKE128 with key ss will be used to calculate the IKE authentication code for the IKE data to be authenticated, according to [RFC7296].

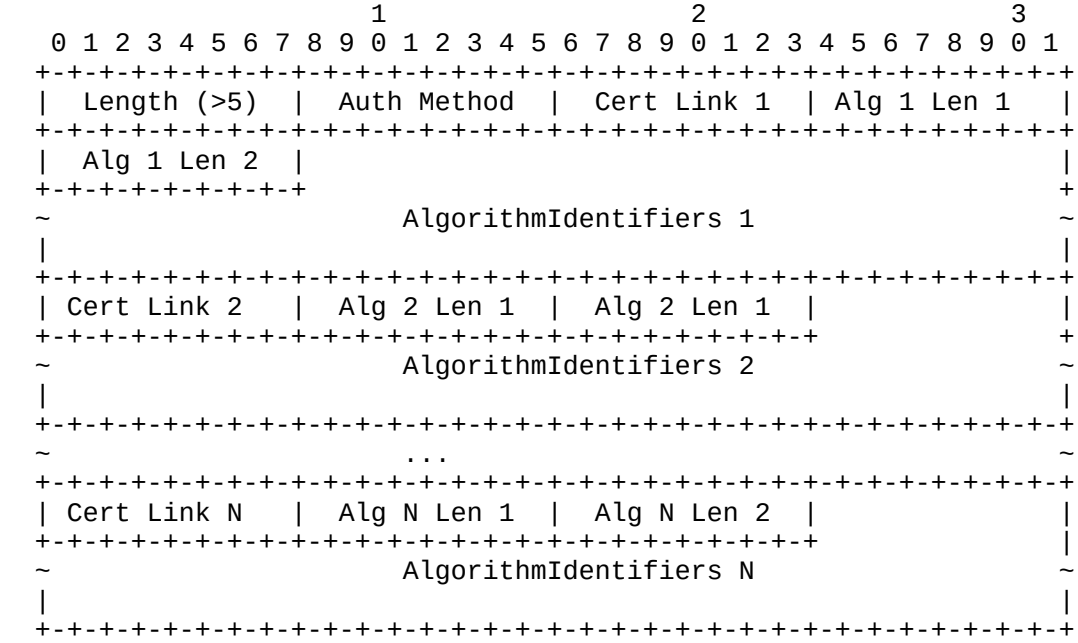

Fig.3 Payload Format for KEM based Authentication Announcement

[EDNOTE: Moreover, the ss obtained from KEM based Authentication can be twisted with the some key derived from the IKEv2 KE part, as does in [I-D.WCS]. Comments and suggestions are welcome]

Alternatively, fixed algorithm combination for KEM based authentication, say ML-KEM-768_SHA128, MAY be defined by adding a new value in the "IKEv2 Authenticaion Method" registry [IANA-IKEv2]. Similar to the above definition for AlgorithmIdentifiers, ML-KEM-768_SHA128 means that ML-KEM-768 will be used to encapsulate and decapsulate the shared secret ss in KEM-based authenticaion, while SHA128 with key ss will be used to calculate the IKE authentication code for the IKE data to be authenticated, according to [RFC7296]. Compared to AlgorithmIdentifiers, such a fixed algorithm combination can easily be recognized and used, but it sits on the same level as KEM based authentication, the general scheme specified the above. Anyway, this happens for general "Digital Signature" (14) and a specific digital scheme, e.g., "ECDSA with SHA-256 on the P-256 curve" (16) [IANA-IKEv2]. Once this happens, such KEM based Authentication Announcements with different Auth Method MUST be merged to the Authentication Announcements described in Fig. 3.

[EDNOTE: Comments and suggestions are welcome for fixed algorithm combinations.]

[EDNOTE: More examples may be provided later.]

6. Security Considerations

To be done later. 1) It may be not a good idea by directly showing the decapsulated secret ss as the means of authentication here. The reason is that the entity being authenticated may employ the owner of the KEM public/private key pair as an oracle to decapsualte the secret via running a different session, normally within a separate protocol or scenario where directly showing such a secret is harmless. 2) It may be preferable or MUST limit the use of such a KEM certificate only for KEM authentication.

7. IANA Considerations

This document is supposed to define a new type in the "IKEv2 Authentication Method" registry under "Internet Key Exchange Version 2 (IKEv2) Parameters" [IANA-IKEv2], maintained by IANA: .

Value	IKEv2 Authentication Method	Reference
16 (TBD)	KEM based Authentication	This draft

8. Acknowledgments

To be added later.

9. 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>.

[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>.

[RFC7427] Kivinen, T. and J. Snyder, "Signature Authentication in the Internet Key Exchange Version 2 (IKEv2)", RFC 7427, DOI 10.17487/RFC7427, January 2015, <https://www.rfc-editor.org/info/rfc7427>.

[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>.

[RFC9593] Smyslov, V., "Announcing Supported Authentication Methods in the Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 9593, DOI 10.17487/RFC9593, July 2024, <https://www.rfc-editor.org/info/rfc9593>.

[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>.

[FIPS204] National Institute of Standards and Technology, "FIPS 204: Module-Lattice-Based Digital Signature Standard", Federal Information Processing Standards Publication , August 2024, <https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.204.pdf>.

[FIPS205] National Institute of Standards and Technology, "FIPS 205: Stateless Hash-Based Digital Signature Standard", Federal Information Processing Standards Publication , August 2024, <https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.204.pdf>.

[X.690] ITU-T, "Information Technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)", ISO/IEC 8825-1:2021 (E), ITU-T Recommendation X.690, February 2021.

[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>.

10. Informative References

[HAZ24] Huang, J., Adomnicai, A., Zhang, J., Dai, W., Liu, Y., Cheung, R. C. C., Koc, C. K., and D. Chen, "Revisiting Keccak and Dilithium Implementations on ARMv7-M", IACR Transactions on Cryptographic Hardware and Embedded Systems. ISSN 2569-2925, Vol. 2024, No. 2, pp. 124., March 2024, <https://tches.iacr.org/index.php/TCHES/article/view/11419>.

[SSW20] Schwabe, P., Stebila, D., and T. Wiggers, "Post-quantum TLS without handshake signatures", In the Proceedings of ACM CCS 2020, pages 14611480. ACM Press. doi:10.1145/3372297.3423350., November 2020.

[I-D.RSF25] Reddy, T., Smyslov, V., and S. Fluhrer, "Signature Authentication in the Internet Key Exchange Version 2 (IKEv2) using PQC", Work in Progress, Internet-Draft, v04, February 2025, <https://datatracker.ietf.org/doc/draft-reddy-ipsecme-ikev2-pqc-auth/>.

[I-D.Flu25] Fluhrer, S., "IKEv2 Support of ML-DSA", Work in Progress, Internet-Draft, v00, January 2025, <https://datatracker.ietf.org/doc/draft-sfluhrer-ipsecme-ikev2-mldsa/>.

[I-D.HMW] Hu, J., Morioka, Y., and G. Wang, "Post-Quantum Traditional (PQ/T) Hybrid PKI Authentication in the Internet Key Exchange Version 2 (IKEv2)", Work in Progress, Internet-Draft, v01, November 2024, <https://datatracker.ietf.org/doc/draft-hu-ipsecme-pqt-hybrid-auth/>.

[I-D.WCS] Wiggers, T., Celi, S., Schwabe, P., Stebila, D., and N. Sullivan, "KEM-based Authentication for TLS 1.3", Work in Progress, Individual Draft, v04, April 2025, <https://datatracker.ietf.org/doc/draft-celi-wiggers-tls-authkem/>.

[I-D.WCSSS] Wiggers, T., Celi, S., Schwabe, P., Stebila, D., and N. Sullivan, "KEM-based pre-shared-key handshakes for TLS 1.3", Work in Progress, Individual Draft, v03, April 2025, <https://datatracker.ietf.org/doc/draft-wiggers-tls-authkem-psk/>.

[I-D.BHCD] Bindel, B., Hale, B., Connolly, D., and F. Driscoll, "Hybrid signature spectrums", Work in Progress, Internet Draft, v06, January 2025, <https://datatracker.ietf.org/doc/draft-ietf-pquip-hybrid-signature-spectrums/>.

[I-D.KR24] Kampanakis, K. and G. Ravago, "Post-quantum Hybrid Key Exchange with ML-KEM in the Internet Key Exchange Protocol Version 2 (IKEv2)", Work in Progress, Internet-Draft,, November 2024, <https://datatracker.ietf.org/doc/draft-kampanakis-ml-kem-ikev2/>.

Author’s Address

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