IP Security Maintenance and Extensions
Internet-Draft
Intended status: Standards Track
Expires: 4 September 2025

G. Wang, Ed.
Huawei Int. Pte Ltd

3 March 2025

Composite ML-DSA Authentication in the IKEv2
draft-wang-ipsecme-composite-mldsa-auth-ikev2-00

Abstract

This draft specifies composite ML-DSA authentication in the Internet Key Exchange Protocol Version 2 (IKEv2) [RFC7296]. Namely, the authenticaiton in the IKEv2 is completed by using a compiste signature of ML-DSA [FIPS203], the newly post-quantum digital singature standard, and one of the following traditional singature algorithms, SA-PKCS#1v1.5, RSA-PSS, ECDSA, Ed25519, and Ed448. These concrete composite algorithm specifications follow [OGPKF24]. Composite ML-DSA authenticatio is achieved by asking to add a new value in the "IKEv2 Authentication Method" registry [IANA-IKEv2], mantained by IANA. After that, two peers MUST send the SUPPORTED_AUTH_METHODS Notify, defined in [RFC9593], to negotiate the specific composite ML-DSA algoithms.

[EDNOTE: Code points for composite ML-DSA authentication may need to be assigned in the "IKEv2 Authentication 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 4 September 2025.

Table of Contents

1. 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 discribed 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.HM25] 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 correponding KEM certificates and singatures for the involved siganture algorithms are exchanged via the INTERMEDIATE Exchange, defined in [RFC9242].

Motivated by the fact that no concrete composite signature authentication has been proposed, this draft specifies how the authentication in the IKEv2 is completed by using composite ML-DSA singature algorithms, defined [OGPKF24]. Namely, those algorithms include all the following compsite signatures of ML-DSA [FIPS203], the newly post-quantum digital singature standard released by NIST, and one of the following traditional singature algorithms, SA-PKCS#1v1.5, RSA-PSS, ECDSA, Ed25519, and Ed448. In function, composite ML-DSA authenticatio is achieved by asking to add a new value (TBD) in the "IKEv2 Authentication Method" registry [IANA-IKEv2], mantained by IANA. After that, two peers MUST send the SUPPORTED_AUTH_METHODS Notify, defined in [RFC9593], to negotiate the specific composite ML-DSA algoithms. Finally, using the the specific composite ML-DSA algoithms selected, not necessarily the same algoithm in bidirections, the peers SHALL sign and verify the IKE data to be authenticated, according the specfication in [RFC7296].

The whole procedure is just the same as using one of the current signature algorithm for the IKEv2 authenticaitno. Therefore, a compsote algorithm achieves "protocol backwards-compatibility" by simply replacing one existing algorithm without requiring any modification at the protocol level, ranther than at the cryptography level, as further explained in Section 2, Composite Design Philosophy, in [OGPKF24].

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. Composite ML-DSA Signatures

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

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

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

* Verify(pk, s, m) -> b: A deterministic verification algorithm, which takes as input a public verifying key pk, 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).

Composite signature follows the same syntax as the above for normal signature algorithm, with the signing key sk and the verifying key pk consisted of two compsonent keys. The detailed structures of composite signature and its keys follow the specification given in [OGPKF24]. For simplicy, all these structures for composit signature can be treated as the concatenation of those of two component signature algorithm.

ML-DSA [FIPS204], Module-Lattice-Based Digital Signature Standard, was released in August of 2024 by NIST. The algorithm has three sets of parameters for three NIST security levels. Namely, ML-DSA-44 for Level 2, ML-DSA-65 for Level 3, and ML-DSA-87 for Level 5.

Moreover, ML-DSA [FIPS204] is specified in pure and pre-hashed signing modes, referred to as "ML-DSA" and "HashML-DSA" respectively. Following the specifications in [OGPKF24], this draft also uses "Composite-ML-DSA" and "HashComposite-ML-DSA" to refer the composite signature obtained by using "ML-DSA" and "HashML-DSA" respectivley. In total, [OGPKF24] specifies 27 composite ML-DSA algorithms: Namely, 13 for "Composite-ML-DSA" with OIDs from <CompSig>.21 to <CompSig>.43, listed in Section 7.1 in [OGPKF24], and 14 for "HashComposite-ML-DSA" with OIDs from <CompSig>.40 to <CompSig>.53, listed in Section 7.2 in [OGPKF24]. Here, "<CompSig>" is equal to "2.16.840.1.114027.80.8.1".

For the purpose of this document, the following uderstanding shall be enough. Namely, to generate a ML-DSA composite signature, the given message M MUST be first processed as following, according to ML-DSA is used as pure or pre-hashed mode.

* For the pure mode "ML-DSA": M'=Domain||len(ctx)||ctx||M.

* For the pre-hashed mode "HashML-DSA": M'=Domain||len(ctx)||ctx||HashOID||PH(M).

Here, according to the specification given in [OGPKF24],

* Domain: It is the domain separator, which is defiend as the OID of the this specific composite siganture algorithm, eg., <CompSig>.28 for pure mode MLDSA65-ECDSA-P384, or <CompSig>.51 for pre-hashed mode HashMLDSA87-ECDSA-P384-SHA512. >.

* ctx: The context string, which defaults to the empty string.

* len(ctx): The length of ctx.

* PH(M): The hash value of applying pre-hash PH to message M, where PH is the assocated hash algorithm with this specific composite siganture algorithm, e.g., SHA512 for the compsite siganture HashMLDSA87-ECDSA-P384-SHA512.

* HashOID: The OID of the pre-hash PH assocated with this specific composite siganture algorithm, e.g., the OID of SHA512 for the compsite siganture algorithm HashMLDSA87-ECDSA-P384-SHA512.

After that, the processed message M’ is sent to ML-DSA signing algorithm to sign, i.e. s1=ML-DSA(mldsaSK, M’, ctx=Domain) and also to the traditional digital signature algorithm to sign, e.g, s2=Trad.Sign(tradSK, M’). Then, those two component signatures SHALL be concatenated together as the composite signature s, i.e., s=s1||s2. Note that the domain separator Domain used here is to achieve the weak non-separability (WNS), defined in [I-D.BHCD].

Finally, to verify a composite signature s=s1||s2, s SHALL be parsed as s1 and s2. Then, message M SHALL be processed just as above to output M’, according to if ML-DSA used here is pure mode or pre-hashed mode. Finally, (s, M) is a valid compsite signature and message pair if and only if both (s1, M’) is valid singature for ML-DSA and (s2, M’) is valid for the traditional signature algorithm used here.

4. Protocl Details for Composite ML-DSA 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 Composite ML-DSA Authentication, this document is supposed to apply the value of 15 (TBD) for "Composite ML-DSA Authentication" in the "IKEv2 Authentication Method" registry (Section 6).

4.1. Exchanges for Composite ML-DSA Authentication

After the initiator starts the IKE_SA_INIT exchange as usual, the responder sents the notify SUPPORTED_AUTH_METHODS with value of 15 (TBD) to indicate that the responder wants to run Composite ML-DSA Authentication with respect to some specific Composite ML-DSA algorithms, which the responder supports. These compsite algoithms will be listed in the SUPPORTED_AUTH_METHODS Notify Payload (Section 4.2), 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 Composite ML-DSA algorithm, with highest preference of the responder, from the list of all Composite ML-DSA algorithms sent by the responder which the initator supports as well, to authenticate itself to the initiator.

Table 1 below gives an example to show how two peers use the SUPPORTED_AUTH_METHODS notification to run Composite ML-DSA 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 Composite ML-DSA authentication.

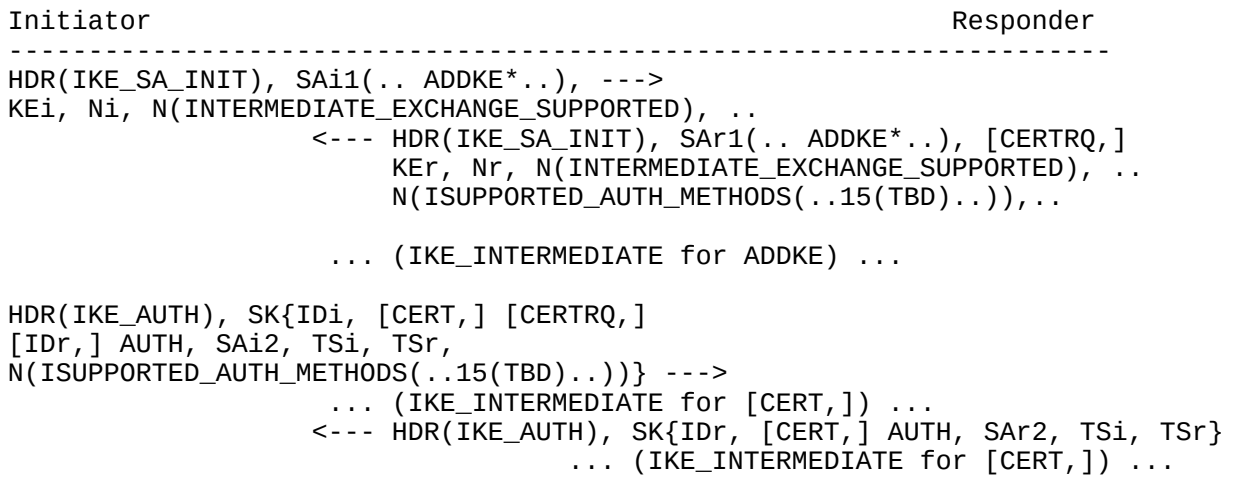

Fig. 1 An Example of Running Composite ML-DSA Authentication between Two Peers

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.]

4.2. Payload Format for Composite ML-DSA 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.

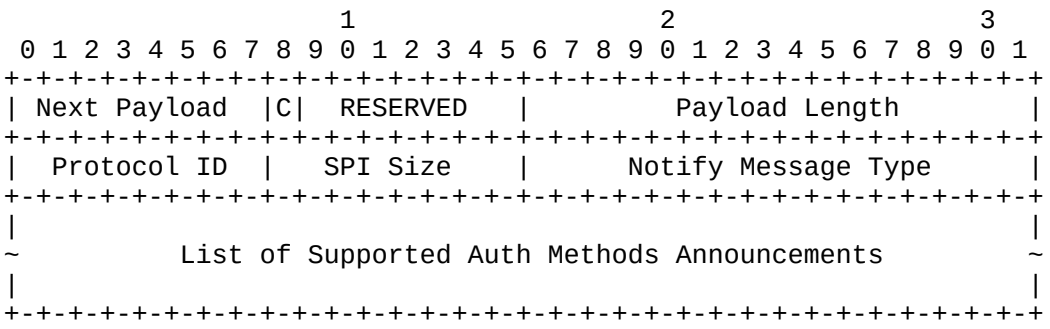

Fig.2 SUPPORTED_AUTH_METHODS Notify Payload Format

Payload Format for Composite ML-DSA 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 Composite ML-DSA 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 15 (TBD), standing for "Composite ML-DSA authentication" for the IKEv2.

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

* Alg Len 1: Length of the Composite ML-DSA algorithm idenfied in AlgorithmIdentifiers below, in octets.

* AlgorithmIdentifier: One variable-length ASN.1 objects that are encoded using Distinguished Encoding Rules (DER) [X.690] and identify one specific Composite ML-DSA algorithm.

By the above definition, AlgorithmIdentifier is just <CompSig>.i [OGPKF24], where i specifies which specific Composite ML-DSA algorithm was announced here. For example, if AlgorithmIdentifier is <CompSig>.28, this means that pure mode MLDSA65-ECDSA-P384 is selected. Namely, pure mode MLDSA65ML and ECDSA-P384 will be used to compositely sign the IKE data to be authenticated, according to [RFC7296].

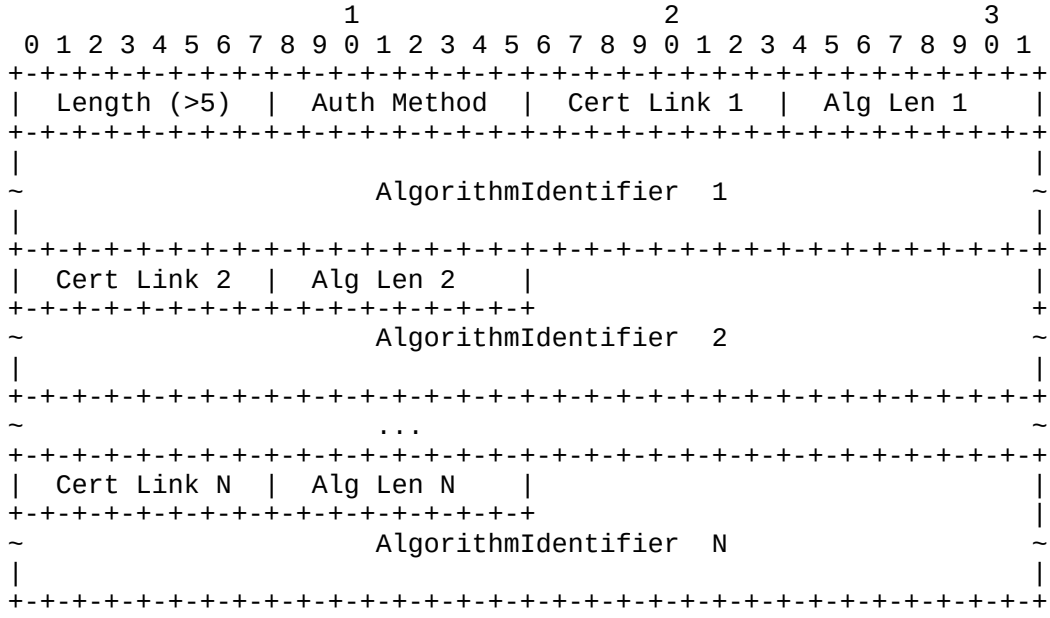

Fig.3 Payload Format for Composite ML-DSA Authentication Announcement

In this document, fixed composite ML-DSA algorithms defined in [OGPKF24] are registered as a set of authentication methods in the "IKEv2 Authentication Method" registry [IANA-IKEv2], mantained by IANA. This approach is parallel to what the general "Digital Signature" (14) is regisered in the same registry. It is also possible to choose register some specific composite ML-DSA algorithms ((popular ones, for example) at the same level as "Digital Signature" (14). However, if too many such algorithms registered there, this will make the limited codes in the "IKEv2 Authentication Method" registry (only 256 possibilities) even scarce.

[EDNOTE: More examples may be provided later.]

5. Security Considerations

To be done later.

6. 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
15 (TBD)	Composite ML-DSA Authentication	This draft

7. Acknowledgments

To be added later.

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>.

[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>.

9. Informative References

[OGPKF24] M. Ounsworth, M., Gray, J., Pala, M., J. Klaussner, J., and S. S. Fluhrer, "Composite ML-DSA For use in X.509 Public Key Infrastructure and CMS", Work in Progress, Internet-Draft, v04., October 2024, <https://datatracker.ietf.org/doc/draft-ietf-lamps-pq-composite-sigs/>.

[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.HM25] Morioka, Y., "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.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/>.

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