IPSECME
Internet-Draft
Intended status: Standards Track
Expires: 8 January 2026

Y. Fukagawa
NTT Com
H. Kosuge
NTT
M. Saito
NTT Data Intellilink
S. Fluhrer
Cisco Systems
A. Nagai
NTT

7 July 2025

Post-quantum Hybrid Key Exchange with NTRU in the Internet Key Exchange Protocol Version 2 (IKEv2)
draft-skyline-ipsecme-ntru-ikev2-00

Abstract

This document specifies the use of NTRU in the Internet Key Exchange Protocol Version 2 (IKEv2), following the framework defined in RFC 9370. RFC 9370 introduces a mechanism that enables multiple key encapsulation mechanisms (KEMs) to be used within IKEv2, allowing up to seven additional key exchange methods to be negotiated alongside the initial key exchange. This document defines how NTRU can be used as an additional key exchange method to improve the post-quantum security of IKEv2 by broadening algorithmic diversity.

[EDNOTE: IANA KE code points for NTRU will be needed to be assigned.]

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

The IKEv2 as specified in [RFC7296], uses the Diffie-Hellman (DH) or the Elliptic Curve Diffie-Hellman (ECDH) algorithm to establish a shared secret between an initiator and a responder. However, the advent of a Cryptographically Relevant Quantum Computer (CRQC) defined in [I-D.D25] would pose a significant threat to the security of DH and ECDH. To address this problem, the Mixing Preshared Keys in IKEv2 specification [RFC8784] was introduced as a temporary option. After that, PQ/T Hybrid Key Encapsulation Mechanism (KEM) in IKEv2 was specified, which combines post-quantum and classical KEMs to ensure security against both classical and quantum adversaries. To enable PQ/T Hybrid KEMs in IKEv2, [RFC9370] extends the IKEv2 specification to allow multiple key exchanges to take place during the computation of a shared secret in the Security Association (SA) setup.

The Multiple Key Exchanges defined in [RFC9370] specifies how to perform up to seven additional key exchanges using the IKE_INTERMEDIATE or IKE_FOLLOWUP_KE messages, along with procedures for deriving new SKEYSEED and KEYMAT key materials.The reason RFC 9370 allows the use of multiple post-quantum KEMs is to hedge against the risk that any single KEM may be broken in the future. While RFC 9370 provides a general framework that enables hybrid KEMs, it does not define specific instantiations for concrete combinations of multiple KEMs. Currently, two Internet-Drafts aim to define such additional key exchange methods by building upon the framework established in [RFC9370]. First, [I-D.KR24] adds ML-KEM [FIPS203], also known as Kyber. Second, [I-D.W25] introduces FrodoKEM for reasons of algorithm diversity and cryptographic agility.

To improve the diversity of post-quantum KEM algorithms, this document proposes to include NTRU [I-D.F25]-specifically, the version of NIST Post-Quantum Cryptography (PQC) round 3 [NTRU] as one of the algorithms usable within the framework defined in [RFC9370]. The inclusion of NTRU increases algorithmic diversity for two main reasons. First, although NTRU is lattice-based, its security is based on the NTRU problem and does not rely on the (Module) Learning With Errors ((M)LWE) problems. Second, it provides a tighter security proof for the desired security notion: indistinguishability under adaptive chosen-ciphertext attacks (IND-CCA2). In addition to increasing algorithmic versatility, the absence of patents is beneficial to users. NTRU is patent-free, which has the advantage of allowing users to avoid patent issues. NTRU is also a valuable choice from an implementation perspective, as its overall performance is comparable to that of ML-KEM, with notably faster encapsulation.

1.1. KEMs

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

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

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

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

1.2. NTRU

NTRU [I-D.F25] is a KEM based on lattices over polynomial rings. It offers good performance, as well as small and balanced key and ciphertext sizes. The parameters of NTRU were chosen based on the three security levels defined in the NIST PQC Project: Levels 1, 3, and 5. These levels correspond to the hardness of breaking AES-128, AES-192 and AES-256. Also, NTRU defines two families of parameters: NTRU-HPS and NTRU-HRSS. The NTRU-HPS consists of three parameter sets: ntruhps2048677, ntruhps4096821, and ntruhps40961229, corresponding to security levels 1, 3, and 5, respectively. The NTRU-HRSS includes two parameter sets: ntruhrss701 and ntruhrss1373, corresponding to security levels 1 and 5, respectively.

This specification introduces five parameters for IKEv2 key exchanges to allow flexible selection based on security levels and implementation requirements, such as performance, bandwidth, and packet size constraints. Since the ntruhps40961229 and ntruhrss1373 public key and ciphertext sizes can exceed the typical network MTU, these key exchanges could require two or three network IP packets from both the initiator and the responder.

1.3. Conventions and Definitions

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

2. NTRU in IKEv2

2.1. NTRU in IKE_INTERMEDIATE or CREATE_CHILD_SA messages

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

Section 2.2.2 of [RFC9370] specifies that KEi(0), KEr(0) are regular key exchange messages in the first IKE_SA_INIT exchange which end up generating a set of keying material, SK_d, SK_a[i/r], and SK_e[i/r]. The peers then perform an IKE_INTERMEDIATE exchange, carrying new Key

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

NTRU can also be used to create or rekey a Child SA or rekey the IKE SA by using an IKE_FOLLOWUP_KE message after a CREATE_CHILD_SA message. After the NTRU additional key exchange KE(1) has taken place using and IKE_FOLLOWUP_KE exchange, the IKE or Child SA are rekeyed by stirring the new NTRU shared secret SK(1) in SKEYSEED and KEYMAT as specified in Section 2.2.4 of [RFC9370].

The ntruhps4096821, ntruhps40961229, and ntruhrss1373 public keys and ciphertexts may make UDP packet sizes larger typical network MTUs (1500 bytes). Thus, IKE_INTERMEDIATE or IKE_FOLLOWUP_KE messages carrying NTRU public keys and ciphertexts may be IKEv2 fragmented as per the IKEv2 Message Fragmentation specification [RFC7383].

As the payload lengths of ntruhps2048677, ntruhrss701, and ntruhps4096821 are less than the typical network MTUs (1500 bytes), these parameters MAY be used in IKE_SA_INIT, resulting in a configuration where only post-quantum KEMs are used. The other parameters, ntruhps40961229 and ntruhrss1373 SHOULD NOT be used in IKE_SA_INIT messages that could exceed typical network MTUs and cannot be fragmented using IKEv2 fragmentation.

2.2. Key Exchange Payload

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

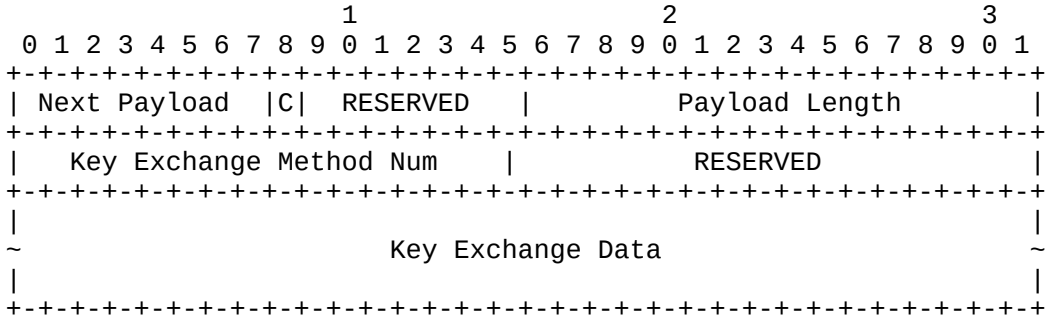

Figure 1: Key Exchange Payload

The Key Exchange Data from the initiator to the responder contains the public key (pk) from the KeyGen() operation encoded as a raw byte array as defined in [I-D.F25].

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

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

KEM	Payload Length (initiator / responder)	Key Exchange Method Num	Data Size in Octets (initiator / responder)
ntruhps2048677	938 / 938	TBD50	930 / 930
ntruhrss701	1146 / 1146	TBD51	1138 / 1138
ntruhps4096821	1238 / 1238	TBD52	1230 / 1230
ntruhps40961229	1850 / 1850	TBD53	1842 / 1842
ntruhrss1373	2409 / 2409	TBD54	2401 / 2401

Table 1: Key Exchange Payload Fields

3. Examples of Running Hybrid KEMS with NTRU

Following general exmaples given in Appendix A of [RFC9370], here is an example to show that the initiator proposes the use of additional key exchanges to establish an IKE SA. Here, the initiator proposes three sets of additional key exchanges. Namely, the first set is TBD36 (ml-kem-768), TBD37 (ml-kem-1024) [I-D.KR24] or NONE; the second set is TBD50 (ntruhps2048677), TBD52 (ntruhps4096821) or NONE. As both of the two additional key exchanes 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 ECR (ID = ENCR_AES_GCM_16,
                   256-bit key)
    Transform PRF (ID = PRF_HMAC_SHA2_512)
    Transform KE (ID = Curve25519)
    Transform ADDKE1 (ID = TBD36)
    Transform ADDKE1 (ID = TBD37)
    Transform ADDKE1 (ID = NONE)
    Transform ADDKE2 (ID = TBD50)
    Transform ADDKE2 (ID = TBD52)
    Transform ADDKE2 (ID = NONE)

                    <--- HDR(IKE_SA_INIT), SAr1(.. ADDKE*...),
                         KEr(Curve25519), Nr, N(IKEV2_FRAG_SUPPORTED),
                         N(INTERMEDIATE_EXCHANGE_SUPPORTED)
                         Proposal #1
                           Transform ECR (ID = ENCR_AES_GCM_16,
                                          256-bit key)
                           Transform PRF (ID = PRF_HMAC_SHA2_512)
                           Transform KE (ID = Curve25519)
                           Transform ADDKE1 (ID = NONE)
                           Transform ADDKE2 (ID = TBD50)

HDR(IKE_INTERMEDIATE), SK {KEi(1)(TBD50)} -->
                    <--- HDR(IKE_INTERMEDIATE), SK {KEr(1)(TBD50)}

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

Figure 2: Hybrid KEMs of ECDH and TBD50 (ntruhps2048677)

In the above specific example, the responder chooses to run one additional key exchanges. Namely, it selects NONE and TBD50 (ntruhps2048677), respectively for the first and second additional key exchanges. According to [RFC7296], a set of keying materials can be derived, in particular SK_d, SK_a[i/r], and SK_e[i/r]. After that, both peers will perform an IKE_INTERMEDIATE exchange, carrying TBD50 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 TBD50 shared secret.

After the completion of both IKE_INTERMEDIATE exchanges, the initiator and the responder continue to the IKE_AUTH exchange phase.

4. Security Considerations

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

The NTRU public key generated by the initiator and the ciphertext generated by the responder use randomness (usually a seed) which MUST be independent of any other random seed used in the IKEv2 negotiation. For example, at the initiator, the NTRU and (EC)DH key pairs should not be generated from the same seed. Also, nonces used in the derivation of SKEYSEED and KEYMAT MUST NOT be reused, in order to ensure the pseudorandomness of key derivation using the pseudorandom function defined in [RFC9370].

5. IANA Considerations

IANA is requested to assign five values for the names "ntruhps2048677", "ntruhrss701", "ntruhps40961229", "ntruhrss1373", and "ntruhps4096821"in the IKEv2 "Transform Type 4 - Key Exchange Method Transform IDs" and has listed this document as the reference. The Recipient Tests field should also point to this document:

[EDNOTE: IANA KE code points for FrodoKEM might be assigned, we assume those code points in this spefication.

Number	Name	Status	Recipient Tests	Reference
TBD50	ntruhps2048677		[TBD, this draft],	[TBD, this draft]
TBD51	ntruhrss701		[TBD, this draft],	[TBD, this draft]
TBD52	ntruhps4096821		[TBD, this draft],	[TBD, this draft]
TBD53	ntruhps40961229		[TBD, this draft],	[TBD, this draft]
TBD54	ntruhrss1373		[TBD, this draft],	[TBD, this draft]
55-1023	Unassigned			

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

6. References

6.1. 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/rfc/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/rfc/rfc7296>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/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/rfc/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/rfc/rfc9370>.

6.2. Informative References

[FIPS203] National Institute of Standards and Technology, "FIPS 203(Initial Draft): Module-Lattice-Based Key-Encapsulation Mechanism Standard", FIPS Standard (Draft) , August 2023, <https://csrc.nist.gov/pubs/fips/203/ipd>.

[I-D.D25] D, F., P, M., and B. Hale, "Terminology for Post-Quantum Traditional Hybrid Schemes", Work in Progress, Internet-Draft, draft-ietf-pquip-pqt-hybrid-terminology-06, 10 January 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-pquip-pqt-hybrid-terminology-06>.

[I-D.F25] Fluhrer, S., Prorock, M., Celi, S., Gray, S., Xagawa, K., and H. Kosuge, "NTRU Key Encapsulation", Work in Progress, Internet-Draft, draft-fluhrer-cfrg-ntru-03, 3 March 2025, <https://datatracker.ietf.org/doc/html/draft-fluhrer-cfrg-ntru-03>.

[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,, February 2024, <https://datatracker.ietf.org/doc/draft-kampanakis-ml-kem-ikev2/>.

[I-D.W25] Wang, G., "Post-quantum Hybrid Key Exchange in the IKEv2 with FrodoKEM", Work in Progress, Internet-Draft,, April 2025, <https://datatracker.ietf.org/doc/draft-wang-hybrid-kem-ikev2-frodo-02/>.

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

[NTRU] Chen, C., Bos, O., Ducas, J., Longa, A., Mironov, J., Naehrig, J., Nikolaenko, T., Peikert, P., Raghunathan, W., Stebila, K., Stebila, T., and Z. Stebila, "NTRU Algorithm Specifications And Supporting Documentation", Preliminary Standardization Proposal submitted to ISO , September 2020, <https://info.isl.ntt.co.jp/crypt/ntru/files/ntru.pdf>.

[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/rfc/rfc7383>.

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

Acknowledgments

To be added later.

Authors' Addresses

Yuta Fukagawa
NTT Com
Email: y.fukagawa@ntt.com

Hisaharu Kosuge
NTT
Email: hrhs@ntt.com

Masataka Saito
NTT Data Intellilink
Email: saitomst@intellilink.co.jp

Scott Fluhrer
Cisco Systems
Email: sfluhrer@cisco.com

Akira Nagai
NTT
Email: akira.nagai@ntt.com