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

G. Wang, Ed.
Huawei Int. Pte Ltd
V. Smyslov
ELVIS-PLUS

2 March 2026

KEM-based Authentication for IKEv2 with Post-quantum Security
draft-wang-ipsecme-kem-auth-ikev2-03

Abstract

This draft specifies a new authentication mechanism, called KEM (Key Encapsulation Mechanism) -based authentication, for the Internet Key Exchange Protocol Version 2 (IKEv2). This is motivated by the fact that some post-quantum KEMs (like ML-KEM) are more efficient than post-quantum signature algorithms (like ML-DSA).

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.

Copyright Notice

Table of Contents

1. Introduction

1.1. Notes of Change

Version -02 was completely rewritten to accommodate ideas expressed in PQuAKE protocol.

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

* More details about how each side does for running KEM authentication in Section 6.1.

* Added section about 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 against the Internet Key Exchange Protocol Version 2 (IKEv2) [RFC7296], an extension allowing multiple key exchanges in IKEv2 [RFC9370] was introduced to achieve post-quantum (PQ) security for the key exchange.

"Signature Authentication in the Internet Key Exchange Version 2 (IKEv2) using PQC" [I-D.ietf-ipsecme-ikev2-pqc-auth] specifies how NIST PQ digital algorithms ML-DSA [FIPS204] and SLH-DSA [FIPS205] can be used in IKEv2. On the other hand, "Post-Quantum Traditional (PQ/T) Hybrid PKI Authentication in the Internet Key Exchange Version 2 (IKEv2)" [I-D.hu-ipsecme-pqt-hybrid-auth] specifies how to run general hybrid PQ/T digital algorithms in IKEv2.

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.celi-wiggers-tls-authkem] [I-D.wiggers-tls-authkem-psk] specifies how a 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. A similar 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.celi-wiggers-tls-authkem], KEM-based authentication also reduces implementation code size, as the code for PQ signature may not be needed, and KEM-based authentication can re-use the KEM code for ephemeral key establishment.

This draft specifies how to use KEM-based authentication for the Internet Key Exchange Protocol Version 2 (IKEv2) [RFC7296].

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

A key encapsulation mechanism (KEM) is a set of algorithms that allows key exchange to be performed, which allows one entity to encapsulate a secret key under a (longterm or ephemeral) public key of another entity. By following the definition given in [I-D.ietf-ipsecme-ikev2-mlkem], 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 definition given in [I-D.ietf-pquip-hybrid-signature-spectrums], 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, encapsulation, signing, decapsulation, and verifying.

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 concluded:

* 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/ Decapsulation Key	Public Key/ Encapsulation Key	Signature/ 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 PQ 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 version Kyber, also ML-DSA and its informal version Dilithium, the following Table 2 is obtained.

	KeyGen Time/ SKeyGen Time	Encap Time/ Sign Time	Decap Time/ Verify Time
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 computational comparison shown in Table 2 reasonably presents a performance difference of ML-KEM and ML-DSA on the same platform with optimal implementations, 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 time of ML-KEM.

* The signature verifying time of ML-DSA is over 3 times of the decapsulation time 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 computational time of ML-DSA is about 5 times of that of ML-KEM.

5. Protocol Overview for KEM-based Authentication

With a KEM-based authentication in IKEv2 peers first perform an ephemeral key exchange that allows them to derive session keys for protecting subsequent exchanges. This ephemeral key exchange may take place entirely in the IKE_SA_INIT exchange or be accompanied by a series of subsequent IKE_INTERMEDIATE exchanges [RFC9370], in which case it would be a hybrid key exchange.

To run KEM-based Authentication in IKEv2, the two peers first exchange their long-term KEM certificates, and then exchange random values encapsulated with each other’s long-term KEM public key from the corresponding KEM certificate. Here, both long-term KEM certificates and the KEM ciphertexts encapsulating the random values are sent via the IKE_INTERMEDIATE exchanges. After decapsulating these random values the two peers cam demonstrate their knowledge of the random values sent by the other side in the subsequent IKE_AUTH exchange, thus confirming that they do possess the corresponding KEM private keys linked to their long-term KEM certificates.

Mode details about the KEM-based authentication are provided in <citations for PQuAKE and other papers needed>.

6. Protocol Details for KEM-based Authentication

The KEM-based authentication in IKEv2 relies on the IKE_INTERMEDIATE exchange [RFC9242] that MUST be supported. In addition, the following extensions can contribute to the security and the robustness of the protocol.

* IKE fragmentation [RFC7383] SHOULD be negotiated in case peers use UDP as a transport for IKE, since KEM public key and KEM ciphertext can be of significant size.

* Multiple key exchanges [RFC9370] has to be utilized if a hybrid key exchange combining traditional and post-quantum key exchange methods is required.

* Using group Postquantum Preshared Key (PPK) [RFC9867] can contribute to the identity hiding, thus it is RECOMMENDED to use.

* The ability to authenticate the full protocol transcript [I-D.smyslov-ipsecme-ikev2-downgrade-prevention] protects peers against downgrade attacks and SHOULD be used.

6.1. Exchanges for KEM-based Authentication

6.1.1. The IKE_SA_INIT Exchange

The initiator wishing to use the KEM-based authentication indicates this to the responder by including a new status type notification USE_AUTHKEM (with type <TBA2>) into the IKE_SA_INIT request message. This notification has the Protocol ID and SPI Size both set to 0 (meaning no SPI is present), and its notification data is empty. To be able to use the KEM-based authentication the initiator MUST also include the INTERMEDIATE_EXCHANGE_SUPPORTED notification.

If the responder receives the IKE_SA_INIT request containing the USE_AUTHKEM notification and it supports and is configured to use the KEM-based authentication, then it sends this notification back in the response. The responder can optionally include the Certificate Request payload (as defined in [RFC7296]) to help the initiator to select the right KEM certificate. The responder can also indicate which KEMs it is willing to use for authentication by including the SUPPORTED_AUTH_METHODS notification [RFC9593] in the response. In this case the notification SHOULD NOT contain announcements with the Auth Method other than a newly defined "KEM-based Authentication" (<TBA1>). The format of announcement is defined in Section 6.3.2.

Peers can also negotiate other IKEv2 extensions during IKE_SA_INIT. Some of them are listed in Section 6. In particular, the figures below show the optional use of PPK [RFC9867] during KEM-based authentication.

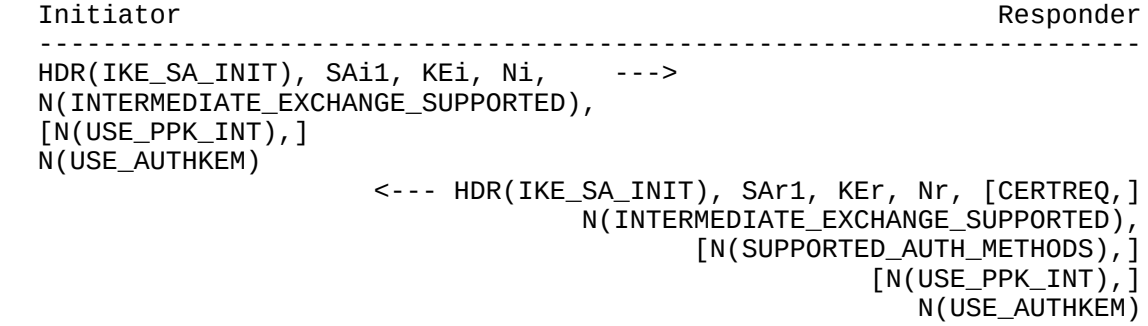

Figure 1: The IKE_SA_INIT Exchange

Peers MUST ignore the USE_AUTHKEM notification in the message if it is not accompanied by the INTERMEDIATE_EXCHANGE_SUPPORTED notification.

If using KEM-based authentication is mandatory for the responder and the initiator did not propose it or failed to make a valid proposal, then the responder MUST send the NO_PROPOSAL_CHOSEN notification indicating that the proposed parameters are unacceptable. Otherwise the responder can continue establishing IKE SA using other authentication method.

If using the KEM-based authentication is mandatory for the initiator and it failed to negotiate it with the responder, then the initiator MUST cancel establishing IKE SA. Otherwise the initiator can continue using other authentication method.

After the IKE_SA_INIT exchange completes, peers calculate the session keys as defined in Section 2.14 of [RFC7296] and use these keys to protect the next exchange.

6.1.2. The IKE_INTERMEDIATE Exchanges

Once the KEM-based authentication is negotiated in IKE_SA_INIT, a series of the IKE_INTERMEDIATE exchanges takes place. These exchanges can be classified as those for the purpose of the KEM-based authentication and those for other purposes. The exchanges for the KEM base authentication MUST follow all other IKE_INTERMEDIATE exchanges, so that they take place right before the IKE_AUTH exchange.

6.1.2.1. The IKE_INTERMEDIATE Exchanges for Other Purposes

First, if multiple key exchanges were negotiated, then one or more IKE_INTERMEDIATE exchanges are performed, right after the IKE_SA_INIT exchange, as defined in [RFC9370].

```
Initiator                                                  Responder
-----------------------------------------------------------------------
HDR(IKE_INTERMEDIATE), SK {KEi(n)}   --->
                             <---   HDR(IKE_INTERMEDIATE), SK {KEr(n)}
```

Figure 2: Additional Key Exchange(s)

Each of these exchanges updates the session keys as defined in Section 2.2.2 of [RFC9370].

If the use of PPKs as defined in [RFC9867] is negotiated, then the IKE_INTERMEDIATE exchange negotiating the PPK to use is performed then as defined in Section 3.1 of [RFC9867].

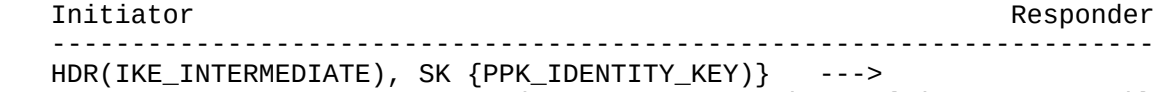

Figure 3: PPK Negotiation

After this exchange the session keys are updated as defined in Section 3.1.1 of [RFC9867].

If both additional key exchange(s) and the use of PPKs as defined in [RFC9867] are negotiated, then the last additional key exchange MAY alternatively be combined with negotiating the PPK in a single IKE_INTERMEDIATE exchange as shown in Figure 3. Note, that in many cases only one additional key exchange will be negotiated (as a part of PQ/T hybrid key exchange), thus in this case the last additional key exchange will also be the only one.

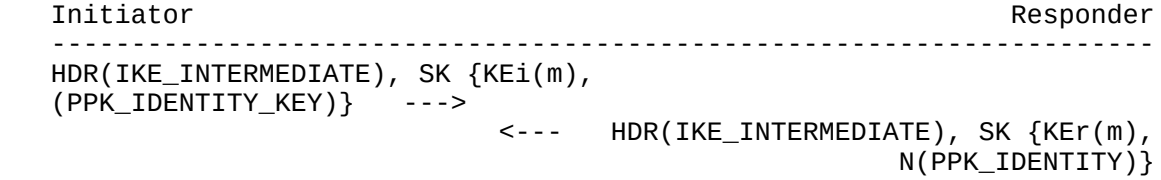

Figure 4: Last Additional Key Exchange Combined with PPK Negotiation

If this combined IKE_INTERMEDIATE exchange takes place, then upon its completion session keys are updated twice - fist as defined in Section 2.2.2 of [RFC9370] and then, provided the peers negotiate the PPK to use, as defined in Section 3.1.1 of [RFC9867].

Note, that while [RFC9867] generally requires that the PPK negotiation is performed in the last IKE_INTERMEDIATE exchange right before the IKE_AUTH exchange, it also allows this requirement to be overridden by other specification. This document specifically overrides this requirement by allowing the IKE_INTERMEDIATE exchanges concerned with the KEM-based authentication to take place between the negotiation of PPK and the IKE_AUTH exchange.

6.1.2.2. The IKE_INTERMEDIATE Exchanges for the KEM-based Authentication

The next two IKE_INTERMEDIATE exchanges perform the necessary actions for the KEM-based authentication.

In the first exchange the responder sends its KEM certificate to the initiator. The initiator can optionally include the Certificate Request payload in the request message to help the responder to select the right KEM certificate. The Certificate Request payload can also be used to indicate which format(s) of certificate are acceptable for the sender of this payload. This is done by setting the Cert Encoding field to appropriate value. For KEM-based authentication this field MUST be either "X.509 bundle" (new certificate encoding defined in this document, it value is <TBA5 by IANA>) or "Hash and URL of X.509 bundle" (value 13). If both encoding are acceptable, then two Certificate Request payloads MUST be included (with possibly different list of CAs) each with its own value in the Cert Encoding field. Since the Certificate Request payload is optional, its absence leaves the choice of the certificate and its encoding on the certificate sender’s discretion. Note that these rules are also applicable to the Certificate Request payload sent by the responder in the IKE_SA_INIT exchange (see Section 6.1.1).

The initiator can also indicate which KEMs it is willing to use for authentication by including the SUPPORTED_AUTH_METHODS notification [RFC9593] in the request. In this case the notification SHOULD NOT contain announcements with the Auth Method other than the "KEM Authentication" (<TBA1>). The format of announcement is defined in Section 6.3.2.

The responder sends its KEM certificate in the response. Besides the KEM certificate, additional intermediate signing certificates and CRLs can be sent. All these certificates and CRLs are packed into the CertificateBundle ASN.1 structure defined in Section 3.6 of [RFC7296]. The KEM certificate MUST be the only KEM certificate in the bundle. The bundle can either be present in the Certificate payload (with the Cert Encoding field set to a new value "X.509 bundle") or provided by reference (with the Cert Encoding field set to "Hash and URL of X.509 bundle"). If certificate bundle is provided by reference, then a host can avoid downloading the peer’s certificates if it cached them from a previous connection. Only one Certificate payload MUST be present in the response.

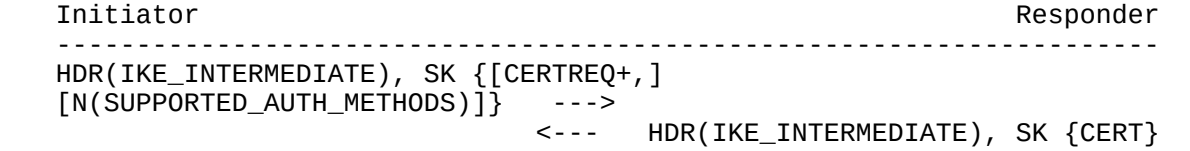

Figure 5: Obtaining Responder’s KEM Certificate

This exchange does not update the current session keys.

The initiator verifies the responder’s certificate to make sure that it is valid, properly signed by a certificate chain leaded to a trusted CA, and that it is acceptable for responder authentication based on the initiator’s local policy. The exact verification steps are out of scope of this document, but generally should follow those defined in [RFC5280].

Then the initiator starts the next IKE_INTERMEDIATE exchange. In the request message it sends the ciphertext CT_i, produced as a result of encapsulation of a random value SS_i using the KEM public keys from the responder’s certificate. CT_i is sent in a new status type notification AUTHKEM_CT (with type <TBA3>). The Protocol ID and SPI Size fields are set to 0 (meaning no SPI is present). The notification data contains the ciphertext CT_i.

In the same request the initiator also sends its KEM certificate. Besides the KEM certificate, additional intermediate signing certificates and CRLs can be sent. All these certificates and CRLs are packed into the CertificateBundle ASN.1 structure defined in Section 3.6 of [RFC7296]. The KEM certificate MUST be the only KEM certificate in the bundle. The bundle can either be present in the Certificate payload (with the Cert Encoding field set to a new value "X.509 bundle") or provided by reference (with the Cert Encoding field set to "Hash and URL of X.509 bundle"). If certificate bundle is provided by reference, then a host can avoid downloading the peer’s certificates if it cached them from a previous connection.

This content (either bundle or hash & URL) is sent in a new payload Encrypted Certificate (with type <TBA0 by IANA>). The content is encrypted using key derived from SS_i. The details of the Certificate payload encryption are provided in Section 6.3.3. Encryption provides protection of initiator’s identity against active attackers, since only a responder a responder with the private key corresponding to the responder’s certificate is able to decrypt the initiator’s KEM certificate.

> | Note, that if the bundle is provided by reference and an
> | attacker can monitor the responder's network activity, then
> | this identity protection might not work.

Exactly one Encrypted Certificate payload MUST be present in the message.

The responder decrypts the initiator's certificate, verifies it to make sure that it is valid, properly signed by a certificate chain leaded to a trusted CA, and that it is acceptable for initiator authentication based on the responder's local policy. The exact verification steps are out of scope of this document, but generally should follow those defined in [RFC5280].

The initiator's certificate and the responder's one, sent in the previous exchange, MAY contain public keys for different KEMs, provided that peers support these KEMs, - there is no requirement that they are for the same KEM.

The responder then sends a response message containing the ciphertext CT_r, produced as a result of encapsulation of a random value SS_r using the KEM public keys from the initiator's certificate. CT_r is sent in the AUTHKEM_CT notification in the same manner as SS_i.

The responder also includes a new status type notification AUTHKEM_SSCONF (with type <TBA3>). The Protocol ID and SPI Size fields are set to 0 (meaning no SPI is present). The notification data is computed as prf(SK_pr, SS_r), where prf is the negotiated PRF for this SA and SK_pr is from the current session keys (see Section 3.3.1 of [RFC9242]).

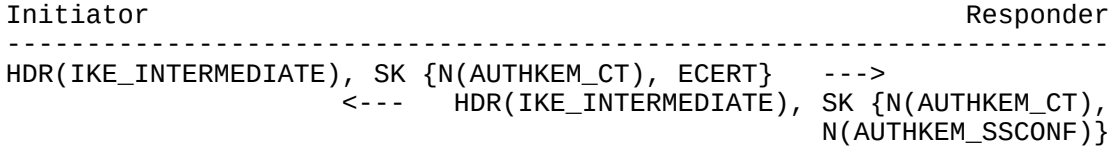

Figure 6: Obtaining Initiator's KEM Certificate and Exchange of Encrypted Random Values

Upon receiving the response message the initiator decapsulates the received ciphertext CT_r and obtains the SS_r key. Then the initiator computes prf(SK_pr, SS_r) using the obtained SS_r and verifies that the value matches the content of the AUTHKEM_SSCONF notification. This ensures that there are no mis-configurations and the the received SS_r value is indeed the same that was encapsulated.

Upon completion of this exchange the session keys are updated. A new value of SKEYSEED is computed as:

SKEYSEED' = prf(SK_d, SS_i | SS_r)

Then all the session keys (SK_d, SK_ai, SK_ar, SK_ei, SK_er, SK_pi, and SK_pr) are computed as defined in Section 2.14 of [RFC7296]. These session keys are then used for protecting all subsequent exchanges starting from the IKE_AUTH exchange.

6.1.3. The IKE_AUTH Exchange

The IKE_AUTH exchange immediately follows the IKE_INTERMEDIATE exchange where peers sent each other encrypted SS values. The IKE_AUTH exchange is basically unchanged from that defined in [RFC7296]. The Auth Method field in the AUTH payloads MUST be "KEM Authentication". The format of the AUTH payload is computed as defined in Section 6.3.1.

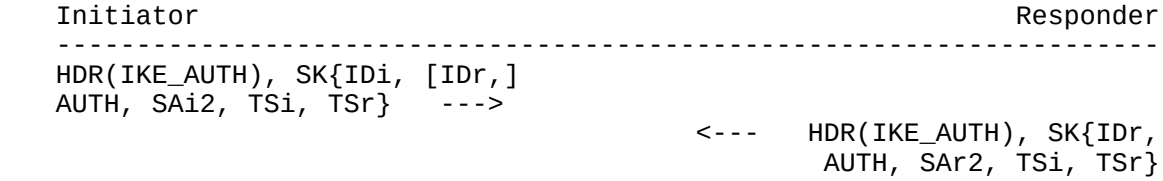

Figure 7: The IKE_AUTH Exchange

6.1.3.1. Identities in the IKE_AUTH Exchange

With KEM based authentication peers have to exchange their KEM certificates before actual authentication of each other. Certificates usually contain some identity information, that the CA asserts the included public key is bound to. This makes peers aware of each other's identity before the IKE_AUTH exchange starts.

However, IKEv2 doesn't rely on the identities from certificates when performing authentication. Instead, Identification payloads (IDi and IDr) are mandatory in the IKE_AUTH exchange and exactly they are included into the AUTH payload calculation. The content of the IDi and IDr payload may differ from the identification information in the corresponding certificate, as stated in Section 3.5 of [RFC7296].

6.2. Alternative Sequences of Exchanges

This section is to be removed eventually. It provides ideas for alternative design of the protocol.

The presented sequence of exchanges requires two additional exchanges (namely IKE_INTERMEDIATE exchanges) for KEM-based authentication compared to signature-based authentication. It is possible to only have one additional IKE_INTERMEDIATE exchange, but this either results in some degradation of security properties or leads to additional protocol complexity. Note, that the amount of data exchanged over the wire would not changed, the different items would just be combined in one message. This would increase the size of these messages that might not be desirable from a reliable delivery point of view.

6.2.1. Alternative 1: Send Responder Certificate with Protection Only by the IKE_SA_INIT Keys

It is possible to exchange certificates before all additional key exchanges (and optional negotiation of the PPK) are completed. In this case the certificates would only be protected with session keys computed after the IKE_SA_INIT exchange (in case of hybrid key exchange that would be traditional key exchange).

In this case, the exchange shown in Figure 5 is not needed, since the certificates would be exchanged in the previous exchange. For example, the exchange shown in Figure 4 would look like:

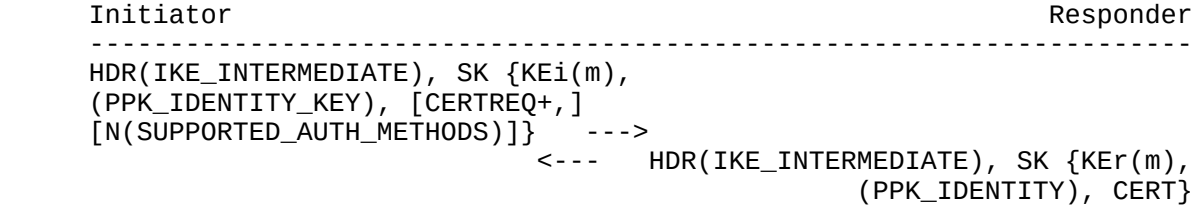

Figure 8: Last Additional Key Exchange Combined with PPK Negotiation and Responder’s Certificate

This design would decrease the level of identity hiding for responder, since the session keys used to protect the responder’s certificates in case of hybrid key exchange could be cracked by an attacker using quantum computer.

6.2.2. Alternative 2: Use "Future" Keys to protect Responder's Certificate

In this case the exchange shown in Figure 5 is also not needed. The responder sends its certificate in the previous IKE_INTERMEDIATE exchange using new Encrypted Certificate payload. Note, that this exchange contains the last additional key exchange and optionally the PPK negotiation, thus its messages are not yet protected with the session keys that would result in this exchange - the new keys are calculated only once the exchange completes. However, the responder can calculate the updated keys before it sends the response message. The idea is that the responder calculates new session keys and uses these "future" keys (for the next exchange) to protect its certificate in the Encrypted Certificate payload, that itself is protected with the "current" session keys in the Encrypted payload. The Encrypted Certificate payload is protected using keys derived from the SK_d from the next exchange (see Section 6.3.3 for details).

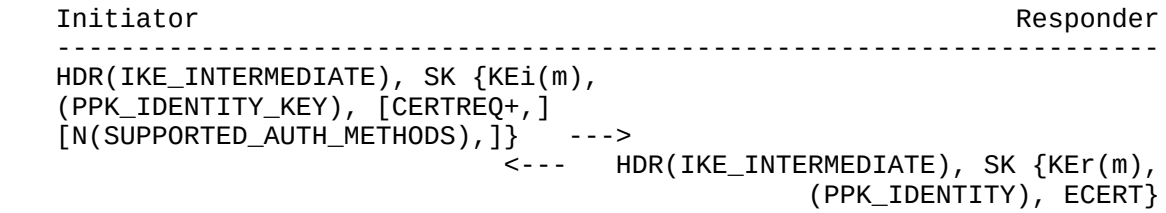

Figure 9: Last Additional Key Exchange Combined with PPK Negotiation and Encrypted Responder's Certificate

This sequence of exchanges allows certificates to be protected at the same level as in the original sequence of exchanges (in particular, both certificates can be protected with group PPK). The drawback is the added complexity for implementations, since it could break the traditional state transitions logic of IKE.

6.3. Data Formats

6.3.1. Authentication Payload for KEM-based Authentication

The format of the Authentication payload is defined in Section 3.8 of [RFC7296]. In case of KEM-based authentication the Auth Method is set to "KEM Authentication" (<TBA1>). The Authentication Data is computed as:

For the initiator:

```
AUTH = prf( prf(SK_pi, "Key Pad for IKEv2"),
                  <InitiatorSignedOctets>)
```

For the responder:

```
AUTH = prf( prf(SK_pr, "Key Pad for IKEv2"),
                  <ResponderSignedOctets>)
```

The content of the InitiatorSignedOctets and the ResponderSignedOctets is defined in Section 3.3.2 of [RFC9242]. If full transcript authentication is employed as specified in [I-D.smyslov-ipsecme-ikev2-downgrade-prevention], then the InitiatorSignedOctets and the ResponderSignedOctets are additionally modified as defined in [I-D.smyslov-ipsecme-ikev2-downgrade-prevention].

6.3.2. Authentication Announcement for KEM Authentication

The format of the announcement for the "KEM Authentication" is the same as for "Digital Signature" as shown in Section 3.2.3 of [RFC9593]. For convenience the format is also shown below.

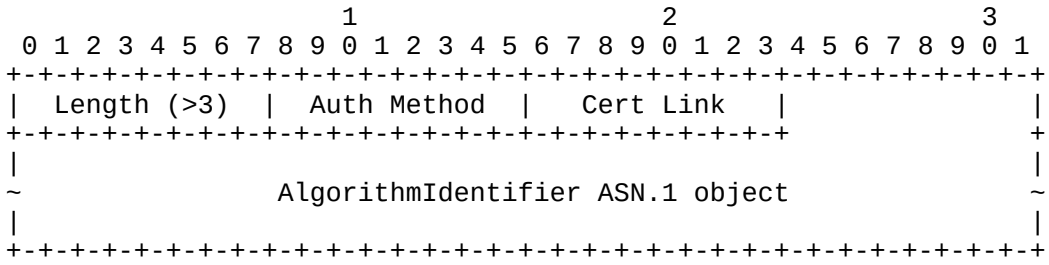

Figure 10: Supported Authentication Method

* Length - Length of the announcement in octets, must be greater than 3.

* Auth Method - Announced authentication method, in this case it is "KEM Authentication".

* Cert Link - Links this announcement with particular CA; see Section 3.2.2 of [RFC9593] for details.

* AlgorithmIdentifier ASN.1 object - the AlgorithmIdentifier of PKIX (see Section 4.1.1.2 of [RFC5280]), encoded using distinguished encoding rules (DER) [X.690].

For "KEM Authentication" announcement the AlgorithmIdentifier MUST contain identifier of a KEM algorithm (and not, for example, of a signature algorithm). Identifiers for KEM algorithms are specified in the corresponding documents for these algorithms (e.g., for ML-KEM see [I-D.ietf-lamps-kyber-certificates]).

6.3.3. Encrypted Certificate Payload

The Encrypted Certificate payload, denoted ECERT in this document, provides a means to transport certificates or other authentication-related information in encrypted form. The payload type for the Encrypted Certificate payload is <TBA0 by IANA>. This payload is semantically equivalent to the Certificate payload (defined in Section 3.6 of [RFC7296], but it provides a means to encrypt the certificate information. For this purpose the format of the Encrypted Certificate payload mirrors the format of the Encrypted payload (defined in Section 3.14 of [RFC7296]) with the difference, that it contains not the encrypted IKE payloads, but the encrypted Cert Encoding and Certificate Data as defined in Section 3.6 of [RFC7296].

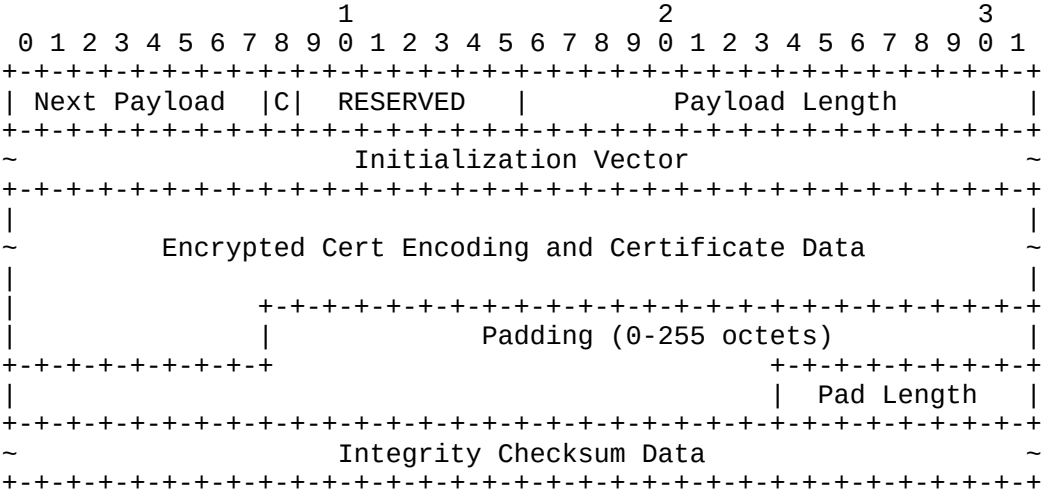

Figure 11: Encrypted Certificate Payload Format

Initialization Vector, Padding, Pad Length and Integrity Checksum Data are defined in Section 3.14 of [RFC7296].

This document restricts the possible values of Cert Encoding in the ECERT payload to either "X.509 bundle" (<TBA5 by IANA>) or "Hash and URL of X.509 bundle" (13).

The payload is encrypted and authenticated using the same encryption and integrity algorithms as were negotiated for IKE SA. The encryption and authentication keys are calculated as follows:

```
EC_a | EC_e = prf+ ( SK_d, K )
```

where SK_d is from the current session keys (see Section 3.3.1 of [RFC9242]) and K is the key to protect the certificate data (like SS_i in Section 6.1.2.2).

7. Interaction with Other IKEv2 Extensions

To be added.

8. 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.

9. IANA Considerations

IANA is requested to make the following changes in the IKEv2 registries "Internet Key Exchange Version 2 (IKEv2) Parameters" [IANA-IKEv2]:

1. Define a new payload type in the "IKEv2 Payload Types" registry:

Value	Next Payload Type	Notation
TBA0	Encrypted Certificate	ECERT

Table 3

2. Define a new authentication method in the "IKEv2 Authentication Method" registry:

Value	IKEv2 Authentication Method
TBA1	KEM Authentication

Table 4

3. Define three new status type notifications in the "IKEv2 Notify Message Status Types" registry:

Value	Notify Message Status Type
TBA2	USE_AUTHKEM
TBA3	AUTHKEM_CT
TBA4	AUTHKEM_SSCONF

Table 5

4. Define a new certificate encoding in the "IKEv2 Certificate Encodings" registry:

Value	Certificate Encoding
TBA5	X.509 bundle

Table 6

10. References

10.1. Normative References

[IANA-IKEv2] "Internet Key Exchange Version 2 (IKEv2) Parameters", <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>.

[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>.

10.2. 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>.

[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>.

[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>.

[I-D.celi-wiggers-tls-authkem] Wiggers, T., Celi, S., Schwabe, P., Stebila, D., and N. Sullivan, "KEM-based Authentication for TLS 1.3", Work in Progress, Internet-Draft, draft-celi-wiggers-tls-authkem-

06, 4 November 2025,
<https://datatracker.ietf.org/doc/html/draft-celi-wiggers-tls-authkem-06>.

[I-D.hu-ipsecme-pqt-hybrid-auth]
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, draft-hu-ipsecme-pqt-hybrid-auth-04, 27 February 2026, <https://datatracker.ietf.org/doc/html/draft-hu-ipsecme-pqt-hybrid-auth-04>.

[I-D.ietf-ipsecme-ikev2-mlkem]
Kampanakis, P., "Post-quantum Hybrid Key Exchange with ML-KEM in the Internet Key Exchange Protocol Version 2 (IKEv2)", Work in Progress, Internet-Draft, draft-ietf-ipsecme-ikev2-mlkem-03, 29 September 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-ipsecme-ikev2-mlkem-03>.

[I-D.ietf-ipsecme-ikev2-pqc-auth]
Reddy.K, T., Smyslov, V., and S. Fluhrer, "Signature Authentication in the Internet Key Exchange Version 2 (IKEv2) using PQC", Work in Progress, Internet-Draft, draft-ietf-ipsecme-ikev2-pqc-auth-06, 20 October 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-ipsecme-ikev2-pqc-auth-06>.

[I-D.ietf-lamps-kyber-certificates]
Turner, S., Kampanakis, P., Massimo, J., and B. Westerbaan, "Internet X.509 Public Key Infrastructure - Algorithm Identifiers for the Module-Lattice-Based Key-Encapsulation Mechanism (ML-KEM)", Work in Progress, Internet-Draft, draft-ietf-lamps-kyber-certificates-11, 22 July 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-lamps-kyber-certificates-11>.

[I-D.ietf-pquip-hybrid-signature-spectrums]
Bindel, N., Hale, B., Connolly, D., and F. D, "Hybrid signature spectrums", Work in Progress, Internet-Draft, draft-ietf-pquip-hybrid-signature-spectrums-07, 20 June 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-pquip-hybrid-signature-spectrums-07>.

[I-D.smyslov-ipsecme-ikev2-downgrade-prevention]
Smyslov, V. and C. Patton, "Prevention Downgrade Attacks on the Internet Key Exchange Protocol Version 2 (IKEv2)",

Work in Progress, Internet-Draft, draft-smyslov-ipsecme-ikev2-downgrade-prevention-02, 28 August 2025, <https://datatracker.ietf.org/doc/html/draft-smyslov-ipsecme-ikev2-downgrade-prevention-02>.

[I-D.wiggers-tls-authkem-psk] Wiggers, T., Celi, S., Schwabe, P., Stebila, D., and N. Sullivan, "KEM-based pre-shared-key handshakes for TLS 1.3", Work in Progress, Internet-Draft, draft-wiggers-tls-authkem-psk-04, 4 November 2025, <https://datatracker.ietf.org/doc/html/draft-wiggers-tls-authkem-psk-04>.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <https://www.rfc-editor.org/info/rfc5280>.

[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>.

[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>.

[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>.

[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.

[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.

Acknowledgements

Thom Wiggers shared an idea to provide identity protection of one of the peers against active attackers.

Contributors

Uri Blumenthal and Brandon Luo contributed a lot to the design of this protocol as authors of PQuAKE protocol.

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

Valery Smyslov
ELVIS-PLUS
PO Box 81
Moscow (Zelenograd)
124460
Russian Federation
Email: svan@elvis.ru