Workgroup: Network Working Group
Internet-Draft:
draft-guthrie-ikev2-hybrid-non-composite-auth
Published: 25 March 2022
Intended Status: Standards Track
Expires: 26 September 2022
Authors: R. Guthrie
NSA

Hybrid Non-Composite Authentication in IKEv2

Abstract

This document describes how to extend the Internet Key Exchange Protocol Version 2 (IKEv2) to allow hybrid non-composite authentication. The intended purpose for this extension is to enable the use of a Post-Quantum (PQ) digital signature and X.509 certificate in addition to the use of a traditional authentication method. This document enables peers to signify support for hybrid non-composite authentication, and send additional CERTREQ, AUTH, and CERT payloads to perform multiple authentications.

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 26 September 2022.

Copyright Notice

Table of Contents

1. Introduction

This document describes how to extend the Internet Key Exchange Protocol Version 2 (IKEv2) to allow negotiation of authentication methods, including hybrid authentication. The intended purpose for this extension is to enable the use of a Post-Quantum (PQ) digital signature and X.509 certificate in addition to the use of a traditional authentication method. This document is motivated by [I-D.draft-becker-guthrie-noncomposite-hybrid-auth] and the multiple authentication mechanism for IKEv2 introduced in [RFC4739], and specifies how to perform multiple authentications, with each authentication using its own CERT AND AUTH payloads. This document also leverages the supported authentication method announcement specified in [I-D.draft-ietf-ipsecme-ikev2-auth-announce].

2. Terminology and Notation

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

3. Protocol Details

3.1. Exchanges

If the responder is willing to use this extension, it includes a new HYBRID_AUTH Notify payload in the response message of the IKE_SA_INIT exchange. The inclusion of N(HYBRID_AUTH) in the responder's IKE_SA_INIT message indicates to the initiator that the responder can perform multiple authentications using multiple AUTH and CERT payloads. Additionally, the responder includes in IKE_SA_INIT a SUPPORTED_AUTH_METHODS Notify payload as defined in [I-D.draft-ietf-ipsecme-ikev2-auth-announce]. If a peer sends N(HYBRID_AUTH), it MUST also send N(SUPPORTED_AUTH_METHODS). If the initiator does not support this extension and the extension indicated through inclusion of N(SUPPORTED_AUTH_METHODS), it MUST ignore the received N(HYBRID_AUTH) notification. If the initiator supports this extension, it MAY include N(HYBRID_AUTH) and N(SUPPORTED_AUTH_METHODS) in its IKE_AUTH message, indicating to the responder that it can perform multiple authentications using multiple AUTH and CERT payloads. Additionally, the initiator MAY send in the IKE_AUTH message additional AUTH and CERT payloads based on information conveyed in the responder's SUPPORTED_AUTH_METHODS Notify payload, in order for the responder to perform multiple authentications. If the initiator includes N(HYBRID_AUTH) and N(SUPPORTED_AUTH_METHODS) in its IKE_AUTH message, the responder MAY also send additional AUTH and CERT payloads based on these, in order for the initiator to perform multiple authentications. Note that Figure 1 illustrates the scenario where both initiator and responder support N(HYBRID_AUTH) and both choose to do a single additional authentication. Section 3.5 illustrates what the responder IKE_AUTH message looks like in the case that more than two AUTH payloads and corresponding CERT payloads are sent.

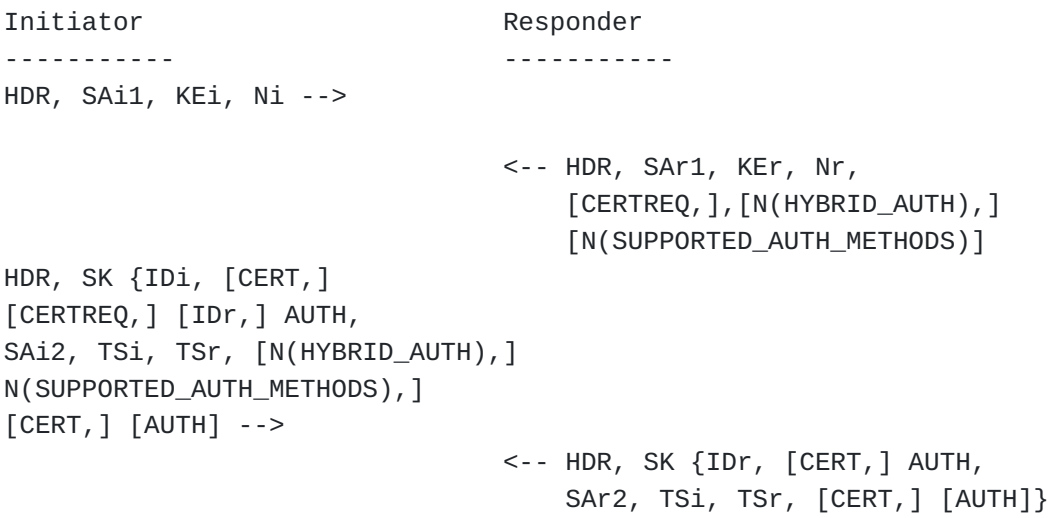

Figure 1: IKE_SA_INIT and IKE_AUTH Exchanges

Figure 1

If the responder sends N(HYBRID_AUTH) in IKE_SA_INIT or the initiator sends N(HYBRID_AUTH) in IKE_AUTH but N(SUPPORTED_AUTH_METHODS) is missing from the message, the responding peer SHOULD ignore the N(HYBRID_AUTH) Notify Payload and proceed as if the other peer does not support this extension.

3.1.1. Exchanges using IKE_INTERMEDIATE

When PQ cryptography is incorporated into IKEv2, either during the key establishment phase or for authentication, it is suspected that the increased size of PQ KEMs and digital signatures will cause IP fragmentation. Though [RFC7383] mitigates this issue for the IKE_AUTH exchange through deploying fragmentation at the IKEv2 layer instead, its fragmentation mechanism functions only on encrypted payloads, and therefore does not extend to the IKE_SA_INIT exchange.

[I-D.draft-ietf-ipsecme-ikev2-intermediate] introduces an IKE_INTERMEDIATE exchange that follows IKE_SA_INIT and precedes IKE_AUTH. IKE_INTERMEDIATE leverages the key establishment of the IKE_SA_INIT exchange and can be used to send larger data that would not fit in an IKE_SA_INIT message without causing IP fragmentation.

In the case that N(SUPPORTED_AUTH_METHODS) is large enough to cause fragmentation of the responder's IKE_SA_INIT message, or in the case that the peers are using IKE_INTERMEDIATE for some other purpose, the responder will send the data from N(SUPPORTED_AUTH_METHODS) in IKE_INTERMEDIATE instead of IKE_SA_INIT, as described in [I-D.draft-ietf-ipsecme-ikev2-auth-announce]. In this case, the responder sends an empty N(SUPPORTED_AUTH_METHODS) payload in IKE_SA_INIT, which signals to the initiator to begin the IKE_INTERMEDIATE. In the

responder's IKE_INTERMEDIATE response, it will again send N(SUPPORTED_AUTH_METHODS), but with a non-empty Notification Data field, where it lists supported authentication methods announcements.

When IKE_INTERMEDIATE is used, the responder MUST use it to send N(HYBRID_AUTH) in the same manner as N(SUPPORTED_AUTH_METHODS). That is, the responder will send an empty HYBRID_AUTH Notify Payload in IKE_SA_INIT, and then send a non-empty N(HYBRID_AUTH) in its IKE_INTERMEDIATE response message.

Figure 2 shows the IKE_SA_INIT, IKE_INTERMEDIATE, and IKE_AUTH exchanges when N(HYBRID_AUTH) and N(SUPPORTED_AUTH_METHODS) are sent using IKE_INTERMEDIATE. Note that both Notify Payloads in the responder's IKE_SA_INIT message are empty, and both Notify Payload's in the responder's IKE_INTERMEDIATE message contain data.

```
Initiator                          Responder
-----------                        -----------
HDR, SAi1, KEi, Ni -->
                                   <-- HDR, SAr1, KEr, Nr,
                                       [CERTREQ,]
                                       [N(HYBRID_AUTH),]
                                       [N(SUPPORTED_AUTH_METHODS)]
HDR, SK {…} -->
                                   <-- HDR, SK{…
                                       [N(HYBRID_AUTH),]
                                       [N(SUPPORTED_AUTH_METHODS)}
HDR, SK {IDi, [CERT,]
[CERTREQ,] [IDr,] AUTH,
SAi2, TSi, TSr,[N(HYBRID_AUTH),]
[N(SUPPORTED_AUTH_METHODS),]
[CERT,] [AUTH]} -->
                                   <-- HDR, SK {IDr, [CERT,] AUTH,
                                       SAr2, TSi, TSr, [CERT,] [AUTH]}
```

Figure 2: IKE_SA_INIT, IKE_INTERMEDIATE, and IKE_AUTH Exchanges

Figure 2

Furthermore, the use of IKE_INTERMEDIATE alters IKEv2's authentication mechanism, as specified in [I-D.draft-ietf-ipsecme-ikev2-intermediate]. If the IKE_INTERMEDIATE exchange is used, care must be taken to apply this modified authentication mechanism to all authentications that are performed with this extension.

3.2. SUPPORTED_AUTH_METHODS Notify Payload

The SUPPORTED_AUTH_METHODS Notify payload as defined in [I-D.draft-ietf-ipsecme-ikev2-auth-announce] is a status notification payload with type TBA; it has a protocol ID of 0 and no Security Parameter Index (SPI). The Notification Data field is defined in [I-D.draft-ietf-ipsecme-ikev2-auth-announce], and is called List of Supported Auth Methods Announcements. It contains the list of supported authentication methods, where each item in the list is called an announcement. Each announcement is a variable-sized blob, whose format depends on the announced authentication method. Authentication methods are represented as values from the "IKEv2 Authentication Method" registry defined in [IKEV2IANA]. [I-D.draft-ietf-ipsecme-ikev2-auth-announce] defines three formats for announcements, each of different lengths. The shortest (2 octets) is used for authentication methods "Shared Key Message Integrity Code" (2) and "NULL Authentication" (13). The second (3 octets) is used for "RSA Digital Signature" (1), "DSS Digital Signature" (3), "ECDSA with SHA-256 on the P-256 curve" (9), "ECDSA with SHA-384 on the P-384 curve" (10) and "ECDSA with SHA-512 on the P-521 curve (11). The last (multi-octet) is used with the "Digital Signature" (14) authentication method defined in [RFC7427].

If a peer sends N(HYBRID_AUTH), it MUST also send N(SUPPORTED_AUTH_METHODS). The peer includes announcements for all supported authentication methods in N(SUPPORTED_AUTH_METHODS), and the data in N(HYBRID_AUTH) provides the context necessary for the receiving peer to parse the authentication methods presented in N(SUPPORTED_AUTH_METHODS) in the context of performing multiple authentications.

N(SUPPORTED_AUTH_METHODS) contains a list of authentication methods the sender supports. For each authentication the sender would like performed, the options for that authentication should be listed consecutively. The options for that authentication should also be listed in order of most preferred to least preferred. The sets of options should themselves appear in order of most preferred authentication to least preferred authentication (i.e., options for the authentication that would be most preferable if only one authentication would occur should be listed first, and so on).

For example, if a peer would like two authentications to be performed, where options for the first authentication are "ECDSA with SHA-384 on the P-384 curve (10)" or "ECDSA with SHA-512 on the P-521 curve (11) (where ECDSA with SHA-512 on the P-521 curve is most preferred) and options for the second authentication are three choices of PQ digital signature: PQ_a, PQ_b, PQ_c (where PQ_b is most preferred, followed by PQ_c, then PQ_a), and with a preference for PQ authentication over traditional authentication in the case

that the receiving peer only performs a single authentication, the announcements for these methods should appear in the following order: PQ_b, PQ_c, PQ_a, ECDSA with SHA-512 on the P-521 curve, ECDSA with SHA-384 on the P-384 curve.

Author's Note: What authentication method will be used for PQ signatures? Will a new IANA value be defined, or will PQ signatures use the Digital Signature (14) Authentication Method value? If it is the former, announcements for PQ authentication may fit into the 3 octet announcement template (along with the other certificate-based authentication methods).

3.3. HYBRID_AUTH Notify Payload

The HYBRID_AUTH Notify payload is a status notification payload with the type TBA. It has a protocol ID of 0 and no Security Parameter Index (SPI). Data consists of two fields. The first is one octet and is used to indicate how many authentications a peer would prefer the other peer select from the supported authentication methods it lists in the N(SUPPORTED_AUTH_METHODS) payload. The second field tells a peer how to select authentication methods from the list of announcements made in N(SUPPORTED_AUTH_METHODS).

The value of the # of Auths field MUST be at least two. If the value of this field is 0 of 1, this Notify Payload SHOULD be ignored and the receiving peer should proceed as if the sending peer does not support this extension. In the case that the receiving peer decides not to ignore this Notify Payload, it MUST check the Indices field and determine whether the Indices field is a reasonable length (i.e., contains between one and seven indices). If the Indices field is a reasonable length, the receiving peer MAY ignore only the # of Auths field and proceed based on the values in the Indices field. Otherwise, the receiving peer MUST ignore the Notify Payload.

The value(s) in the subsequent Indices field tells the peer which authentication methods it may select from N(SUPPORTED_AUTH_METHODS) if it agrees to using this extension. It works as follows: for each authentication the sending peer would like to have performed, the Indices field lists the index of the top choice for each authentication, with the exception of the top choice for the first authentication (which will always coincide with the first announcement). Then, for each authentication that the receiving peer agrees to, it can appropriately select an authentication method from each sub-list. If a peer receives the list enumerated in the previous section, the # of Auths field in the corresponding HYBRID_AUTH Notify Payload will be two, and the Indices field will be 3. Then, if this peer agrees to perform two authentications and supports at least one authentication method presented for each authentication, it will select one authentication method from the

first sub-list, which is announcements 0, 1, and 2, and one authentication method from the second sub-list, which is announcements 3 and 4. If the receiving peer does not support at least one authentication method from each sub-list or does not wish to perform the number of authentications preferred by the sending peer, it MAY select an authentication method from a subset of these sub-lists, rather than an authentication method from each. If the receiving peer wishes to perform only one authentication, it can perform, for example, only the PQ_b authentication, rather than the PQ_a/b/c authentication in conjunction with either ECDSA with SHA-512 on the P-521 curve or ECDSA with SHA-384 on the P-384 curve. If the receiving peer does not support at least one authentication method from each sub-list or does not wish to perform as many authentications as preferred by the sending peer, it SHOULD attempt to choose an authentication method that is preferred by the sending peer.

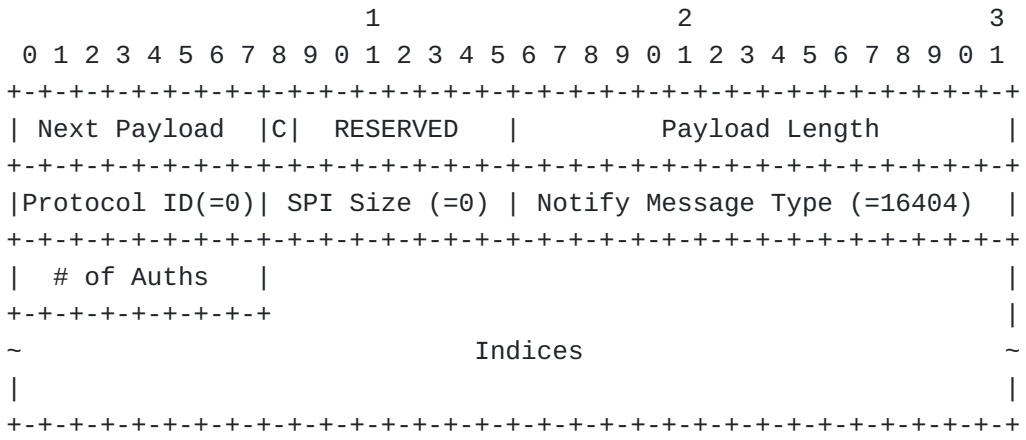

Figure 3: HYBRID_AUTH Notify Payload Format

Figure 3

3.4. CERTREQ Payload

The CERTREQ payload contains the IKE header, the certificate encoding being requested, and the encoding of an acceptable certification authority (CA) for the type of certificate requested [RFC7296]. The CA field is a concatenated list of hashes of the public keys of trusted CAs, where each is encoded as the SHA-1 hash of the Subject Public Key Info element from each Trust Anchor certificate. Subject Public Key Info contains signatureAlgorithm which identifies the cryptographic algorithm used by the CA to sign the certificate. Multiple CERTREQ payloads MAY be sent in order to accommodate multiple values for certificate encodings, but a single CERTREQ payload can contain requests corresponding to certificates used with both traditional and PQ authentication, provided that they use the same certificate encoding.

3.5. Additional AUTH Payload

The AUTH payload, as specified in [RFC7296], contains an IKE header, the authentication method, reserved bits, and authentication data. Additional AUTH payloads MUST use the same AUTH payload format as is defined in [RFC7296]. AUTH payloads MAY use the same authentication method. AUTH payloads sent by a peer SHOULD use authentication methods announced by the other peer in N(SUPPORTED_AUTH_METHODS). For each AUTH payload a peer sends that is using an authentication method that requires a CERT payload, there MUST be at least one CERT payload accompanying that AUTH payload. There may be more than one CERT payload per AUTH payload if certificate chains are sent.

When additional AUTH and CERT payloads are sent in support of multiple authentications, all additional AUTH and CERT payloads MUST be sent at the end of the IKE_AUTH message. Each additional AUTH payload MUST be directly preceded by the CERT payloads that are used during that authentication.

When a peer receives multiple sets of AUTH and CERT payloads, they SHOULD perform all authentications. It is left to the individual implementation to decide whether or not to proceed if some but not all authentications are performed, or some but not all authentications succeed. If no authentications succeed, the connection MUST be dropped.

3.6. Additional CERT Payload

The CERT payload contains the IKE header, the certificate encoding, and the certificate data [RFC7296].

Though this document refers to a single traditional CERT payload and a single PQ CERT payload, it is often the case that multiple CERT payloads are sent in response to a single CERTREQ in order to provide a certificate chain.

[RFC7296] states that if more than one CERT payload is used for authentication, the first CERT payload MUST contain the public key used to verify the AUTH payload. The remaining CERT payloads need not be in any particular order.

If additional AUTH and CERT payloads are sent in support of multiple authentications, all additional AUTH and CERT payloads MUST be sent at the end of the IKE_AUTH message. Each set of CERT payloads used in a single authentication MUST be listed consecutively, beginning with the end entity certificate, and be immediately followed by the relevant AUTH payload. If more than two sets of AUTH and CERT payloads are sent, each additional AUTH payload acts as a delimiter which groups together CERT payloads containing certificates that belong to the same certificate chain.

For example, if the responder sent three sets of AUTH and CERT payloads, the responder's IKE_AUTH message appear as shown in Figure 4.

```
Initiator                       Responder
-----------                     -----------
                                <-- HDR, SK {IDr, [CERT,] AUTH,
                                    SAr2, TSi, TSr, [CERT,] [AUTH,]
                                    [CERT,] [AUTH]}
```

Figure 4: Responder's IKE_AUTH message with three authentications

Figure 4

In the case that more than one authentication uses X.509 certificates, the peer in receipt of these certificates MUST confirm that the SANs match in all end entity certificates.

For guidance on performing validation of multiple certificate chains, refer to [I-D.draft-becker-guthrie-noncomposite-hybrid-auth].

4. Security Considerations

It is likely that the Post-Quantum AUTH and CERT payloads will cause the IKE_AUTH message to exceed the supported message size, requiring use of [RFC7383]. Thus, this document inherits the security concerns of both [RFC7296] and [RFC7383]. This document also incorporates [I-D.draft-ietf-ipsecme-ikev2-intermediate] and [I-D.draft-ietf-ipsecme-ikev2-auth-announce], so it inherits these security considerations as well.

All hybrid implementations are vulnerable to a downgrade attack in which a malicious peer does not express support for PQ algorithms, resulting in an exchange that can only rely upon traditional algorithms for security. Other concerns may arise through the use of multiple certificate chains and digital signatures, as considered in [I-D.draft-becker-guthrie-noncomposite-hybrid-auth].

Last, it is worth noting that a DoS attack could be conducted through this document's use of the N(SUPPORTED_AUTH_METHODS) sent in the IKE_SA_INIT exchange, where a malicious responder could send a long list of authentication announcements.

5. IANA Considerations

This document defines a new Notify Message Type in the "IKEv2 Notify Message Types - Status Types" registry [IKEV2IANA]:

6. References

6.1. Normative References

[I-D.draft-ietf-ipsecme-ikev2-auth-announce]
Smyslov, V., "Announcing Supported Authentication Methods in IKEv2", draft-ietf-ipsecme-ikev2-auth-announce-00 (work in progress), February 2022, <https://datatracker.ietf.org/doc/draft-ietf-ipsecme-ikev2-auth-announce/>.

[I-D.draft-ietf-ipsecme-ikev2-intermediate]
Smyslov, V., "Intermediate Exchange in the IKEv2 Protocol", draft-ietf-ipsecme-ikev2-intermediate-10 (work in progress), March 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-ipsecme-ikev2-intermediate-10>.

[IKEV2IANA] IANA, "Internet Key Exchange Version 2 (IKEv2) Parameters", <https://www.iana.org/assignments/ikev2-parameters/>.

[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>.

6.2. Informative References

[I-D.draft-becker-guthrie-noncomposite-hybrid-auth]

Becker, A., Guthrie, R., and M. Jenkins, "Non-Composite Hybrid Authentication in PKIX and Applications to Internet Protocols", draft-becker-guthrie-noncomposite-hybrid-auth-00 (work in progress), March 2022, <https://www.ietf.org/id/draft-becker-guthrie-noncomposite-hybrid-auth-00.html?msclkid=8114e302aa0611ecbea583d810632940>.

[RFC4739] Eronen, P. and J. Korhonen, "Multiple Authentication Exchanges in the Internet Key Exchange (IKEv2) Protocol", RFC 4739, DOI 10.17487/RFC4739, November 2006, <https://www.rfc-editor.org/info/rfc4739>.

Author's Address

Rebecca Guthrie
National Security Agency

Email: rmguthr@uwe.nsa.gov