Stream: Internet Engineering Task Force (IETF)
RFC: 9593
Category: Standards Track
Published: June 2024
ISSN: 2070-1721
Author: V. Smyslov
ELVIS-PLUS

RFC 9593
Announcing Supported Authentication Methods in the Internet Key Exchange Protocol Version 2 (IKEv2)

Abstract

This specification defines a mechanism that allows implementations of the Internet Key Exchange Protocol Version 2 (IKEv2) to indicate the list of supported authentication methods to their peers while establishing IKEv2 Security Associations (SAs). This mechanism improves interoperability when IKEv2 partners are configured with multiple credentials of different types for authenticating each other.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction 2

2. Terminology and Notation 3

3. Protocol Details 3

   3.1. Exchanges 3

   3.2. SUPPORTED_AUTH_METHODS Notify Message Type 6

      3.2.1. 2-Octet Announcement 6

      3.2.2. 3-Octet Announcement 7

      3.2.3. Multi-octet Announcement 8

4. Interaction with IKEv2 Extensions concerning Authentication 8

5. IANA Considerations 9

6. Security Considerations 9

7. References 10

   7.1. Normative References 10

   7.2. Informative References 10

Appendix A. Examples of Announcing Supported Authentication Methods 11

   A.1. No Need to Use the IKE_INTERMEDIATE Exchange 11

   A.2. With Use of the IKE_INTERMEDIATE Exchange 12

Acknowledgments 12

Author's Address 13

1. Introduction

The Internet Key Exchange Protocol Version 2 (IKEv2), defined in [RFC7296], performs authenticated key exchange in IPsec. IKEv2, unlike its predecessor IKEv1, defined in [RFC2409], doesn't include a mechanism to negotiate an authentication method that the peers would use to authenticate each other. It is assumed that each peer selects whichever authentication method it thinks is appropriate, depending on authentication credentials it has.

This approach generally works well when there is no ambiguity in selecting authentication credentials. SA establishment failure between peers may occur when there are several credentials of different types configured on one peer, while only some of them are supported on the other peer. Another problem situation is when a single credential may be used to produce different types of authentication tokens (e.g., signatures of different formats). Since IKEv2 requires that each peer use exactly one authentication method, and it doesn't provide means for peers to indicate to the other side which authentication methods they support, the peer that supports a wider range of authentication methods (or authentication token formats) could improperly select a method (or format) that is not supported by the other side.

Emerging post-quantum signature algorithms may bring additional challenges for implementations, especially if so-called hybrid schemes are used (e.g., see [COMPOSITE-SIGS]).

This specification defines an extension to the IKEv2 protocol that allows peers to announce their supported authentication methods, thus decreasing risks of SA establishment failure in situations when there are several ways for the peers to authenticate themselves.

2. Terminology and Notation

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. Protocol Details

When establishing an IKE SA, each party may send to its peer a list of the authentication methods it supports and is configured to use. For this purpose, this specification introduces a new Notify Message Type SUPPORTED_AUTH_METHODS. The Notify payload with this Notify Message Type is utilized to convey the supported authentication methods of the party sending it. The sending party may additionally specify that some of the authentication methods are only for use with the particular trust anchors. The receiving party may take this information into consideration when selecting an algorithm for its authentication (i.e., the algorithm used for calculation of the AUTH payload) if several alternatives are available. To simplify the receiver's task of linking the announced authentication methods with the trust anchors, the protocol ensures that the SUPPORTED_AUTH_METHODS notification is always co-located with the CERTREQ payload in the same message.

3.1. Exchanges

The initiator starts the IKE_SA_INIT exchange as usual. If the responder is willing to use this extension, it includes a new notification SUPPORTED_AUTH_METHODS in the IKE_SA_INIT response message. This notification contains a list of authentication methods supported by the responder, ordered by their preference.

```
Initiator                              Responder
-----------                            -----------
HDR, SAi1, KEi, Ni -->
                                   <-- HDR, SAr1, KEr, Nr, [CERTREQ,]
                                     [N(SUPPORTED_AUTH_METHODS)(...)]
```

Figure 1: The IKE_SA_INIT Exchange

If the initiator doesn't support this extension, it ignores the received notification as an unknown status notify.

Regardless of whether the notification is received, if the initiator supports and is willing to use this extension, it includes the SUPPORTED_AUTH_METHODS notification in the IKE_AUTH request message, with a list of authentication methods supported by the initiator, ordered by their preference.

```
Initiator                              Responder
-----------                            -----------
HDR, SK {IDi, [CERT,] [CERTREQ,]
[IDr,] AUTH, SAi2, TSi, TSr,
[N(SUPPORTED_AUTH_METHODS)(...)] }  -->
                                   <-- HDR, SK {IDr, [CERT,]
                                            AUTH, SAr2, TSi, TSr }
```

Figure 2: The IKE_AUTH Exchange

Because the responder sends the SUPPORTED_AUTH_METHODS notification in the IKE_SA_INIT exchange, it must take into account that the response message could grow so much that the IP fragmentation might take place.

- the SUPPORTED_AUTH_METHODS notification to be included is so large, that the responder suspects that IP fragmentation of the resulting IKE_SA_INIT response message may happen;
- both peers support the IKE_INTERMEDIATE exchange, defined in [RFC9242] (i.e., the responder has received and is going to send the INTERMEDIATE_EXCHANGE_SUPPORTED notification);

then the responder **MAY** choose not to send an actual list of the supported authentication methods in the IKE_SA_INIT exchange and instead ask the initiator to start the IKE_INTERMEDIATE exchange for the list to be sent in. This would allow using IKE fragmentation [RFC7383] for long messages (which cannot be used in the IKE_SA_INIT exchange), thus avoiding IP fragmentation. In this case, the responder includes a SUPPORTED_AUTH_METHODS notification containing no data in the IKE_SA_INIT response.

If the initiator receives the empty SUPPORTED_AUTH_METHODS notification in the IKE_SA_INIT exchange, it means that the responder is going to send the list of the supported authentication methods in the IKE_INTERMEDIATE exchange. If this exchange is to be initiated anyway for some other reason, then the responder **MAY** use it to send the SUPPORTED_AUTH_METHODS notification. Otherwise, the initiator **MAY** start the IKE_INTERMEDIATE exchange for this sole purpose by sending an empty IKE_INTERMEDIATE request. The initiator **MAY** also indicate its identity (and possibly the perceived responder's identity too) by including the IDi payload (possibly along with the IDr payload) in the IKE_INTERMEDIATE request. This information could help the responder to send back only those authentication methods that are configured to be used for authentication of this particular initiator. If these payloads are sent, they **MUST** be identical to the IDi/IDr payloads sent later in the IKE_AUTH request.

If the responder has sent any CERTREQ payload in the IKE_SA_INIT, then it **SHOULD** resend the same payload(s) in the IKE_INTERMEDIATE response containing the SUPPORTED_AUTH_METHODS notification if any of the included Announcements has a non-zero Cert Link field (see Sections 3.2.2 and 3.2.3). This requirement allows peers to have a list of Announcements and a list of CAs in the same message, which simplifies their linking. Note that this requirement is always fulfilled for the IKE_SA_INIT and IKE_AUTH exchanges. However, if for any reason the responder doesn't resend CERTREQ payload(s) in the IKE_INTERMEDIATE exchange, then the initiator **MUST NOT** abort negotiation. Instead, the initiator **MAY** either link the Announcements to the CAs received in the IKE_SA_INIT response, or it **MAY** ignore the Announcements containing links to CAs.

If multiple IKE_INTERMEDIATE exchanges take place during IKE SA establishments, it is **RECOMMENDED** that the responder use the last IKE_INTERMEDIATE exchange (the one just before IKE_AUTH) to send the list of supported authentication methods. However, it is not always possible for the responder to know how many IKE_INTERMEDIATE exchanges the initiator will use. In this case the responder **MAY** send the list in any IKE_INTERMEDIATE exchange. If the initiator sends IDi/IDr in an IKE_INTERMEDIATE request, then it is **RECOMMENDED** that the responder sends back the list of authentication methods in the response.

```
Initiator                              Responder
-----------                            -----------
HDR, SAi1, KEi, Ni -->
                                   <-- HDR, SAr1, KEr, Nr, [CERTREQ,]
                                       [N(SUPPORTED_AUTH_METHODS)()]

HDR, SK {..., [IDi, [IDr,]]}  -->
                                   <-- HDR, SK {..., [CERTREQ,]
                                   [N(SUPPORTED_AUTH_METHODS)(...)] }

HDR, SK {IDi, [CERT,] [CERTREQ,]
[IDr,] AUTH, SAi2, TSi, TSr,
[N(SUPPORTED_AUTH_METHODS)(...)] }  -->
                                   <-- HDR, SK {IDr, [CERT,]
                                            AUTH, SAr2, TSi, TSr }
```

Figure 3: Using the IKE_INTERMEDIATE Exchange for Sending Authentication Methods

Note that sending the SUPPORTED_AUTH_METHODS notification and using information obtained from it are optional for both the initiator and the responder. If multiple SUPPORTED_AUTH_METHODS notifications are included in a message, all their announcements form a single ordered list, unless overridden by other extension (see Section 4).

3.2. SUPPORTED_AUTH_METHODS Notify Message Type

The format of the SUPPORTED_AUTH_METHODS Notify payload is shown below.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Payload  |C|  RESERVED   |         Payload Length        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Protocol ID  |   SPI Size    |      Notify Message Type      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~          List of Supported Auth Methods Announcements         ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: SUPPORTED_AUTH_METHODS Notify Payload Format

The Notify payload format is defined in Section 3.10 of [RFC7296]. When a Notify payload of type SUPPORTED_AUTH_METHODS is sent, the Protocol ID field is set to 0, the SPI Size is set to 0 (meaning there is no SPI field), and the Notify Message Type is set to 16443.

Notification data contains the list of supported authentication methods announcements. Each individual announcement is a variable-size data blob whose format depends on the announced authentication method. The blob always starts with an octet containing the length of the blob followed by an octet containing the authentication method. Authentication methods are represented as values from the "IKEv2 Authentication Method" registry defined in [IKEV2-IANA]. The meaning of the remaining octets of the blob, if any, depends on the authentication method. Note that, for the currently defined authentication methods, the length octet fully defines both the format and the semantics of the blob.

If more authentication methods are defined in the future, the corresponding documents must describe the semantics of the announcements for these methods. Implementations **MUST** ignore announcements whose semantics they don't understand.

3.2.1. 2-Octet Announcement

If the announcement contains an authentication method that is not concerned with public key cryptography, then the following format is used.

```
                     1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Length (=2)  |  Auth Method  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: 2-Octet Announcement Format

Length: Length of the blob in octets; must be 2 for this case.

Auth Method: Announced authentication method.

This format is applicable for the authentication methods "Shared Key Message Integrity Code" (2) and "NULL Authentication" (13). Note that the authentication method "Generic Secure Password Authentication Method" (12) would also fall in this category; however, it is negotiated separately (see [RFC6467]), and for this reason there is no point to announce it via this mechanism. See also Section 4.

3.2.2. 3-Octet Announcement

If the announcement contains an authentication method that is concerned with public key cryptography, then the following format is used. This format allows linking the announcement with a particular trust anchor from the Certificate Request payload.

```
                     1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Length (=3)  |  Auth Method  |   Cert Link   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: 3-Octet Announcement Format

Length: Length of the blob in octets; must be 3 for this case.

Auth Method: Announced authentication method.

Cert Link: Links this announcement with a particular CA.

If the Cert Link field contains a non-zero value N, it means that the announced authentication method is intended to be used only with the N-th trust anchor (CA certificate) from the Certificate Request payload(s) sent by this peer. If it is zero, then this authentication method may be used with any CA. If multiple CERTREQ payloads were sent, the CAs from all of them are treated as a single list for the purpose of the linking. If no Certificate Request payload were received, the content of this field **MUST** be ignored and treated as zero.

This format is applicable for the authentication methods "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). Note, however, that these authentication methods are currently superseded by the "Digital Signature" (14) authentication method, which has a different announcement format, described below.

3.2.3. Multi-octet Announcement

The following format is currently used only with the "Digital Signature" (14) authentication method.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Length (>3)  |  Auth Method  |   Cert Link   |               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               +
|                                                               |
~                      AlgorithmIdentifier                      ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 7: Multi-octet Announcement Format

Length: Length of the blob in octets; must be greater than 3 for this case.

Auth Method: Announced authentication method. At the time of writing this document, only value 14 ("Digital Signature") is allowed.

Cert Link: Links this announcement with a particular CA; see Section 3.2.2 for details.

AlgorithmIdentifier: The variable-length ASN.1 object that is encoded using Distinguished Encoding Rules (DER) [X.690] and identifies the signature algorithm (see Section 4.1.1.2 of [RFC5280]).

The "Digital Signature" authentication method, defined in [RFC7427], supersedes previously defined signature authentication methods. In this case, the real authentication algorithm is identified via AlgorithmIdentifier ASN.1 object. Appendix A of [RFC7427] contains examples of commonly used ASN.1 objects.

4. Interaction with IKEv2 Extensions concerning Authentication

Generally in IKEv2 each party independently determines the way it authenticates itself to the peer. In other words, authentication methods selected by the peers need not be the same. However, some IKEv2 extensions break this rule.

The prominent example is "Secure Password Framework for Internet Key Exchange Version 2" [RFC6467], which defines a framework for using secure password authentication in IKEv2. With this framework, peers negotiate using one of the secure password methods in the IKE_SA_INIT exchange -- the initiator sends a list of supported methods in the request, and the responder picks one of them and sends it back in the response.

If peers negotiate secure password authentication, then the selected method is used by both initiator and responder, and no other authentication methods are involved. For this reason, there is no point to announce supported authentication methods in this case. Thus, if the peers choose to go with secure password authentication, they **MUST NOT** send the SUPPORTED_AUTH_METHODS notification.

In the situation when peers are going to use Multiple Authentication Exchanges [RFC4739], they **MAY** include multiple SUPPORTED_AUTH_METHODS notifications (instead of one), each containing authentication methods appropriate for each authentication round. The notifications are included in the order of the preference of performing authentication rounds.

5. IANA Considerations

This document defines a new type in the "IKEv2 Notify Message Status Types" registry:

Value	Notify Message Status Type	Reference
16443	SUPPORTED_AUTH_METHODS	RFC 9593

Table 1

6. Security Considerations

Security considerations for the IKEv2 protocol are discussed in [RFC7296]. Security properties of different authentication methods vary. Refer to corresponding documents, listed in the "IKEv2 Authentication Method" registry on [IKEV2-IANA] for discussion of security properties of each authentication method.

Announcing authentication methods gives an eavesdropper additional information about peers' capabilities. If a peer advertises "NULL Authentication" along with other methods, then an active on-path attacker can encourage peers to use NULL authentication by removing all other announcements. Note that this is not a real "downgrade" attack, since authentication methods in IKEv2 are not negotiated, and in this case NULL authentication should be allowed by local security policy.

Similarly, if an on-path attacker can break some of the announced authentication methods online, then the attacker can encourage peers to use one of these weaker methods by removing all other announcements, and if this succeeds, then perform a person-in-the-middle attack.

7. References

7.1. Normative References

[IKEV2-IANA] 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>.

[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>.

[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>.

[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>.

[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.

7.2. Informative References

[COMPOSITE-SIGS] Ounsworth, M., Gray, J., Pala, M., and J. Klaussner, "Composite Signatures For Use In Internet PKI", Work in Progress, Internet-Draft, draft-ietf-lamps-pq-composite-sigs-01, 24 May 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-lamps-pq-composite-sigs-01>.

[RFC2409] Harkins, D. and D. Carrel, "The Internet Key Exchange (IKE)", RFC 2409, DOI 10.17487/RFC2409, November 1998, <https://www.rfc-editor.org/info/rfc2409>.

[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>.

[RFC6467] Kivinen, T., "Secure Password Framework for Internet Key Exchange Version 2 (IKEv2)", RFC 6467, DOI 10.17487/RFC6467, December 2011, <https://www.rfc-editor.org/info/rfc6467>.

[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>.

Appendix A. Examples of Announcing Supported Authentication Methods

This appendix shows some examples of announcing authentication methods. This appendix is purely informative; if it disagrees with the body of this document, the other text is considered correct. Note that some payloads that are not relevant to this specification may be omitted for brevity.

A.1. No Need to Use the IKE_INTERMEDIATE Exchange

This example illustrates the situation when the SUPPORTED_AUTH_METHODS Notify payload fits into the IKE_SA_INIT message, and thus the IKE_INTERMEDIATE exchange is not needed. In this scenario, the responder announces that it supports the "Shared Key Message Integrity Code" and the "NULL Authentication" authentication methods. The initiator informs the responder that it supports only the "Shared Key Message Integrity Code" authentication method.

```
Initiator                              Responder
-----------                            -----------
                     IKE_SA_INIT
HDR, SAi1, KEi, Ni -->
                                   <-- HDR, SAr1, KEr, Nr,
                                       N(SUPPORTED_AUTH_METHODS(
                                       PSK, NULL))

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

A.2. With Use of the IKE_INTERMEDIATE Exchange

This example illustrates the situation when the IKE_INTERMEDIATE exchange is used. In this scenario, the responder announces that it supports the "Digital signature" authentication method using the RSASSA-PSS algorithm with CA1 and CA2 and the same method using the ECDSA algorithm with CA3. The initiator supports only the "Digital signature" authentication method using the RSASSA-PSS algorithm with no link to a particular CA.

```
Initiator                              Responder
-----------                            -----------
                     IKE_SA_INIT
HDR, SAi1, KEi, Ni,
N(SIGNATURE_HASH_ALGORITHMS) -->
                                   <-- HDR, SAr1, KEr, Nr,
                                       CERTREQ(CA1, CA2, CA3),
                                       N(SIGNATURE_HASH_ALGORITHMS),
                                       N(SUPPORTED_AUTH_METHODS())

                   IKE_INTERMEDIATE
HDR, SK {..., IDi]}  -->
                                   <-- HDR, SK {...,
                                       CERTREQ(CA1, CA2, CA3),
                                       N(SUPPORTED_AUTH_METHODS(
                                       SIGNATURE(RSASSA-PSS:1),
                                       SIGNATURE(RSASSA-PSS:2),
                                       SIGNATURE(ECDSA:3)))}

                      IKE_AUTH
HDR, SK {IDi, CERT, CERTREQ(CA2),
AUTH, SAi2, TSi, TSr,
N(SUPPORTED_AUTH_METHODS(
SIGNATURE(RSASSA-PSS:0))))}  -->
                                   <-- HDR, SK {IDr, CERT,
                                       AUTH, SAr2, TSi, TSr}
```

Acknowledgments

The author would like to thank Paul Wouters for his valuable comments and proposals. The author is also grateful to Daniel Van Geest, who made proposals for protocol improvement. Reese Enghardt and Rifaat Shekh-Yusef contributed to the clarity of the document.

Author's Address

Valery Smyslov
ELVIS-PLUS
PO Box 81
Moscow (Zelenograd)
124460
Russian Federation
Phone: +7 495 276 0211
Email: svan@elvis.ru