Workgroup: Network Working Group
Internet-Draft:
draft-smyslov-ipsecme-ikev2-qr-alt-09
Published: 19 October 2023
Intended Status: Standards Track
Expires: 21 April 2024
Authors: V. Smyslov
ELVIS-PLUS

Alternative Approach for Mixing Preshared Keys in IKEv2 for Post-quantum Security

Abstract

An Internet Key Exchange protocol version 2 (IKEv2) extension defined in RFC8784 allows IPsec traffic to be protected against someone storing VPN communications today and decrypting it later, when (and if) cryptographically relevant quantum computers are available. The protection is achieved by means of Post-quantum Preshared Key (PPK) which is mixed into the session keys calculation. However, this protection doesn't cover an initial IKEv2 SA, which might be unacceptable in some scenarios. This specification defines an alternative way to get protection against quantum computers, which is similar to the solution defined in RFC8784, but protects the initial IKEv2 SA too.

Besides, RFC8784 assumes that PPKs are static and thus they are only used when an initial IKEv2 Security Association (SA) is created. If a fresh PPK is available before the IKE SA is expired, then the only way to use it is to delete the current IKE SA and create a new one from scratch, which is inefficient. This specification also defines a way to use PPKs in active IKEv2 SA for creating additional IPsec SAs and for rekeys operations.

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 21 April 2024.

Table of Contents

1. Introduction

The Internet Key Exchange protocol version 2, defined in [RFC7296], is used in the IPsec architecture for performing authenticated key exchange. [RFC8784] defines an IKEv2 extension for protecting today's IPsec traffic against future quantum computers. The protection is achieved by means of using a Post-quantum Preshared Key (PPK) which is mixed into the session keys calculation. At the time this extension was being developed, it was a consensus in the IPSECME WG that only IPsec traffic needs to have such a protection. It was believed that no sensitive information is transferred over IKE SA and extending the protection to also cover IKE SA traffic would require serious modifications to core IKEv2 protocol, that contradicted to one of the goals to minimize such changes. For the cases when this protection is needed it was suggested to immediately rekey IKE SA once it is created.

In some situations it is desirable to have this protection for IKE SA from the very beginning, when an initial IKE SA is created. An example of such situation is Group Key Management protocol using IKEv2, defined in [I-D.ietf-ipsecme-g-ikev2]. In this protocol session keys are transferred from Group Controller/Key Server (GCKS) to Group Members (GM) immediately once an initial IKE SA is created. While it is possible to postpone transfer of the keys until the IKE SA is rekeyed (and [I-D.ietf-ipsecme-g-ikev2] specifies how to do this), the needed sequence of actions introduces an additional delay and adds unnecessary complexity to the protocol.

Since [RFC8784] was written, a new IKE_INTERMEDIATE exchange for IKEv2 was defined in [RFC9242]. While the primary motivation for developing this exchange was to allow multiple key exchanges to be used in IKEv2 (which is defined in [RFC9370]), the IKE_INTERMEDIATE exchange itself can be used for other purposes too.

This specification makes use of the IKE_INTERMEDIATE exchange to define an alternative approach to [RFC8784], which allows getting protection against quantum computers for initial IKE SA.

Another issue with [RFC8784] is that it assumes that PPKs are static entities, which are changed very infrequently. For this reason PPKs are only used once - when an initial IKE SA is established. This restriction makes it difficult to use [RFC8784] when PPKs are changed relatively frequently, for example as a result of Quantum Key Distribution (QKD). If a fresh PPK becomes available before the IKE SA is expired, there is no way to use it except for deleting this IKE SA and re-creating a new once from scratch using the fresh PPK.

This specification defines the use of PPKs in the CREATE_CHILD_SA exchange for creating additional IPsec SAs and for rekey of IKE and IPsec SAs. This allows to leverage fresh PPKs without the need to delete IKE SA and create it from scratch.

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 Description

3.1. Creating Initial IKE SA

The IKE initiator which supports the IKE_INTERMEDIATE exchange and wants to use PPK to protect initial IKE SA includes the

INTERMEDIATE_EXCHANGE_SUPPORTED notification and a notification of type USE_PPK_ALT in the IKE_SA_INIT request. If the responder supports the IKE_INTERMEDIATE exchange and is willing to use PPK for initial IKE SA protection, it includes both these notifications in the IKE_SA_INIT response.

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

The USE_PPK_ALT is a Status Type IKEv2 notification. Its Notify Message Type is <TBA by IANA>, Protocol ID and SPI Size are both set to 0. This specification doesn't define any data that this notification may contain, so the Notification Data is left empty. However, future extensions of this specification may make use of it. Implementations **MUST** ignore any data they don't understand.

Note, that this negotiation is independent from negotiation of using PPK defined in [RFC8784]. The initiator that supports both RFC8784 and this specification **MAY** include both the USE_PPK_ALT (along with the INTERMEDIATE_EXCHANGE_SUPPORTED) and the USE_PPK notifications if it is configured to use either specification. However, the responder supporting both specifications have to choose one to use, thus it **MUST** return either USE_PPK_ALT or USE_PPK notification in the response, but not both.

If the negotiation was successful, the initiator includes one or more PPK_IDENTITY_KEY notification containing PPK identities the initiator believes are appropriate for the IKE SA being created, into the IKE_INTERMEDIATE request.

The PPK_IDENTITY_KEY is a Status Type IKEv2 notification. Its Notify Message Type is <TBA by IANA>, Protocol ID and SPI Size fields are both set to 0. The format of the notification data is shown below on Figure 1.

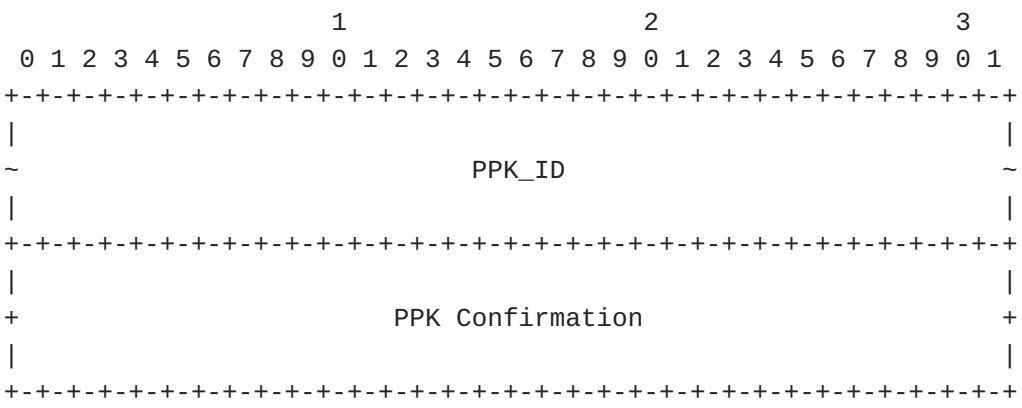

Figure 1: PPK_IDENTITY_KEY Notification Data Format

Where:

- PPK_ID (variable) -- PPK_ID as defined in Section 5.1 of [RFC8784].

- PPK Confirmation (8 octets) -- value, which allows the responder to check whether it has the same PPK as the initiator for a given PPK_ID. This field contains the first 8 octets of a string computed as prf(PPK, Ni | Nr | SPIi | SPIr), where prf is the negotiated PRF; PPK is the key value for a specified PPK_ID; Ni, Nr, SPIi, SPIr -- nonces and IKE SPIs for the SA being established.

If a series of the IKE_INTERMEDIATE exchanges takes place, the PPK_IDENTITY_KEY notification(s) **MUST** be sent in the last one, i.e. in the IKE_INTERMEDIATE exchange immediately preceding the IKE_AUTH exchange. If the last IKE_INTERMEDIATE exchange contains other payloads aimed for some other purpose, then the notification(s) **MAY** be piggybacked with these payloads.

```
Initiator                          Responder
-------------------------------------------------------------------
HDR, SK { ... N(PPK_IDENTITY_KEY, PPK_ID_1)
           [, N(PPK_IDENTITY_KEY, PPK_ID_2)] ...
           [, N(PPK_IDENTITY_KEY, PPK_ID_n)]}   --->
```

Depending on the responder's capabilities and policy the following situations are possible.

a. If the responder is configured with one of the PPKs which IDs were sent by the initiator and this PPK matches the initiator's one (based on the information from the PPK Confirmation field), then the responder selects this PPK and returns back its identity in the PPK_IDENTITY notification. The PPK_IDENTITY notification is defined in [RFC8784].

```
Initiator                          Responder
-------------------------------------------------------------------
               <---    HDR, SK { ... N(PPK_IDENTITY, PPK_ID_i)}
```

In this case the IKE_AUTH exchange is performed as defined in [RFC7296]. However, the keys for the IKE SA are computed using PPK, as described in Section 3.1.1. If the responder returns PPK identity that was not proposed by the initiator, then the initiator should treat this as a fatal error and **MUST** abort the IKE SA establishment.

b. If the responder doesn't have any of the PPKs which IDs were sent by the initiator or it has some of proposed PPKs, but their values mismatch the initiator's ones (based on the information from the PPK Confirmation field), and using PPK is mandatory for the responder, then it **MUST** return AUTHENTICATION_FAILED notification and abort creating the IKE SA.

```
Initiator                          Responder
-------------------------------------------------------------------
                 <---    HDR, SK {... N(AUTHENTICATION_FAILED)}
```

c. If the responder doesn't have any of the PPKs which IDs were sent by the initiator or it has some of proposed PPKs, but their values mismatch the initiator's ones (based on the information from the PPK Confirmation field), and using PPK is optional for the responder, then it doesn't include any PPK_IDENTITY notification to the response.

```
Initiator                          Responder
-------------------------------------------------------------------
                         <---    HDR, SK {...}
```

In this case the initiator cannot achieve quantum computer resistance using the proposed PPKs. If this is a requirement for the initiator, then it **MUST** abort creating IKE SA. Otherwise, the initiator continues with the IKE_AUTH exchange as described in [RFC7296].

Since the responder selects PPK before it knows the identity of the initiator, a situation may occur, when the responder agrees to use some PPK in the IKE_INTERMEDIATE exchange, but during the IKE_AUTH exchange discovers that this particular PPK is not associated with the initiator's identity in its local policy. Note, that the responder does have this PPK, but it is just not listed among the PPKs for using with this initiator. In this case the responder **SHOULD** abort negotiation and return back the AUTHENTICATION_FAILED notification to be consistent with its policy. However, if using PPK

with this initiator is marked optional in the local policy, then the responder **MAY** continue creating IKE SA using the negotiated "wrong" PPK.

3.1.1. Computing IKE SA Keys

Once the PPK is negotiated in the last IKE_INTERMEDIATE exchange, the IKE SA keys are recalculated. Note that if the IKE SA keys are also recalculated as the result of the other actions performed in the IKE_INTERMEDIATE exchange (for example, as defined in [RFC9370], then applying PPK **MUST** be done after all of them, so that recalculating IKE SA keys with PPK is the last action before they are used in the IKE_AUTH exchange.

The IKE SA keys are computed differently compared to [RFC8784]. A new SKEYSEED' value is computed using the negotiated PPK and the most recently computed SK_d key. Note, that the PPK is applied to SK_d exactly how it is specified in [RFC8784], and the result is used as SKEYSEED'.

```
SKEYSEED' = prf+ (PPK, SK_d)
```

Then the SKEYSEED' is used to recalculate all SK_* keys as defined in Section 2.14 of [RFC7296].

```
{SK_d | SK_ai | SK_ar | SK_ei | SK_er | SK_pi | SK_pr}
                          = prf+ (SKEYSEED', Ni | Nr | SPIi | SPIr )
```

In the formula above Ni and Nr are nonces from the IKE_SA_INIT exchange and SPIi, SPIr - SPIs of the IKE SA being created. Note, that SK_d, SK_pi, and SK_pr are not individually recalculated using PPK, as it is defined in [RFC8784].

The resulting keys are then used in the IKE_AUTH exchange and in the created IKE SA.

3.2. Using PPKs in the CREATE_CHILD_SA Exchange

If a fresh PPK is available to both peers at the time when IKE SA created using old PPK is still active, peers **MAY** use this PPK without re-creating the IKE SA. In this case the PPK can be used for creating additional IPsec SAs and rekeying both IKE and IPsec SAs. Since the content of the CREATE_CHILD_SA messages is similar in all these cases, all the payloads not relevant to this specifications are omitted from the diagrams below for brevity. Refer to Section 1.3 of [RFC7296] for the content of the CREATE_CHILD_SA messages.

If the initiator wants to use a PPK in the CREATE_CHILD_SA exchange, it includes one or more PPK_IDENTITY_KEY notification containing PPK

identities the initiator believes are appropriate for the SA being created, into the CREATE_CHILD_SA request. The responder sends back the PPK_IDENTITY notification containing the ID of the selected PPK.

```
Initiator                          Responder
-------------------------------------------------------------------
HDR, SK { ... N(PPK_IDENTITY_KEY, PPK_ID_1)
        [, N(PPK_IDENTITY_KEY, PPK_ID_2)] ...
        [, N(PPK_IDENTITY_KEY, PPK_ID_n)]}   --->
                  <---    HDR, SK { ... N(PPK_IDENTITY, PPK_ID_i)}
```

In case the responder doesn't support (or is not configured for) using PPKs in the CREATE_CHILD_SA exchange, or doesn't have any of the PPKs which IDs were sent by the initiator, or it has some of proposed PPKs, but their values mismatch the initiator's ones (based on the information from the PPK Confirmation field), then it doesn't include any PPK_IDENTITY notification in the response and new SA is created as defined in [RFC7296]. If this is inappropriate for the initiator, it **MAY** immediately delete this SA.

Otherwise the new SA is created using the selected PPK.

3.2.1. Computing Keys

For the purpose of calculation session keys for the new SA, the current SK_d key is first mixed with the selected PPK:

```
SK_d' = prf+ (PPK, SK_d)
```

The resulted key SK_d' is then used instead of SK_d in all formulas for computing keys for the new SA (Sections 2.17 and 2.18 of [RFC7296], Section 2.2.4 of [RFC9370]).

Note, that if the PPK that was used for the IKE SA establishment is not changed, then there is no point to use it in the CREATE_CHILD_SA exchange.

4. Security Considerations

Security considerations of using Post-quantum Preshared Keys in the IKEv2 protocol are discussed in [RFC8784]. Compared to [RFC8784] this specification makes even initial IKE SA quantum secure. In addition, a PPK is mixed into the SK_* keys calculation before the IKE_AUTH exchange starts, and since PPK is used in authentication too, that gives this exchange a QR protection even against active attacker.

This specification relies on the IKE_INTERMEDIATE exchange. Refer to [RFC9242] for discussion of related security issues.

Section 4 of [RFC9370] discusses the potential impact of appearing a CRQC to various cryptographic primitives used in IKEv2. It is worth to repeat here that it is believed that security of symmetric key cryptographic primitives will not be affected by CRQC.

5. IANA Considerations

This document defines two new Notify Message Types in the "IKEv2 Notify Message Types - Status Types" registry:

<TBA>	USE_PPK_ALT
<TBA>	PPK_IDENTITY_KEY

6. Acknowledgements

The author would like to thank Paul Wouters for valuable comments and Tero Kivinen for pointing out to the problem of mismatched preshared keys. Thanks to Rebecca Guthrie for providing comments and proposals for the document.

7. References

7.1. 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/info/rfc2119>.

[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>.

[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>.

[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/info/rfc8784>.

[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>.

7.2. Informative References

[I-D.ietf-ipsecme-g-ikev2]
Smyslov, V. and B. Weis, "Group Key Management using IKEv2", Work in Progress, Internet-Draft, draft-ietf-ipsecme-g-ikev2-09, 19 April 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-ipsecme-g-ikev2-09>.

[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>.

Appendix A. Comparison this Specification with RFC8784

This specification isn't intended to be a replacement for [RFC8784]. Instead, it is supposed to be used in situations where the approach defined there has a significant shortcomings. However, if the partners support both [RFC8784] and this specification, then the latter **MAY** also be used in situations where [RFC8784] suffices.

The approach defined in this document has the following advantages:

1. The main advantage of this specification compared to [RFC8784] is that it allows an initial IKE SA to be protected against quantum computers. This is important for those IKE extensions which transfer sensitive information, e.g. cryptographic keys, over initial IKE SA. The prominent example of such extensions is [I-D.ietf-ipsecme-g-ikev2].

2. This specification allows the initiator to specify several appropriate PPKs and the responder to choose one of them. This feature could simplify PPK rollover.

3. With this specification there is no need for the initiator to calculate the content of the AUTH payload twice (with and without PPK) to support a situation when using PPK is optional for both sides.

The main disadvantage of the approach defined in this document is that it requires an additional round trip (the IKE_INTERMEDIATE exchange) to set up IKE SA. However, if the IKE_INTERMEDIATE exchange has to be used for some other purposes in any case, then PPK stuff can be piggybacked with other payloads, thus eliminating this penalty.

Author's Address

Valery Smyslov
ELVIS-PLUS

PO Box 81
Moscow (Zelenograd)
124460
Russian Federation

Phone: +7 495 276 0211
Email: svan@elvis.ru