

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: 24 November 2025

V. Smyslov  
ELVIS-PLUS  
23 May 2025

Mixing Preshared Keys in the IKE\_INTERMEDIATE and in the CREATE\_CHILD\_SA  
Exchanges of IKEv2 for Post-quantum Security  
draft-ietf-ipsecme-ikev2-qr-alt-10

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 them later, when (and if) a Cryptographically Relevant Quantum Computer (CRQC) is available. The protection is achieved by means of a Post-quantum Preshared Key (PPK) which is mixed into the session keys calculation. However, this protection does not cover an initial IKEv2 Security Association (SA), which might be unacceptable in some scenarios. This specification defines an alternative way to provide protection against quantum computers, which is similar to the solution defined in RFC8784, but also protects the initial IKEv2 SA.

RFC8784 assumes that PPKs are static and thus they are only used when an initial IKEv2 SA is created. If a fresh PPK is available before the IKE SA 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 defines a way to use PPKs in active IKEv2 SAs for creating additional IPsec SAs and rekey 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 24 November 2025.

## Copyright Notice

Copyright (c) 2025 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Terminology and Notation . . . . .	4
3. Protocol Description . . . . .	4
3.1. Creating Initial IKE SA . . . . .	4
3.1.1. Computing IKE SA Keys . . . . .	9
3.2. Using PPKs in the CREATE_CHILD_SA Exchange . . . . .	9
3.2.1. Computing Keys . . . . .	11
4. Security Considerations . . . . .	11
5. IANA Considerations . . . . .	11
6. Acknowledgements . . . . .	12
7. References . . . . .	12
7.1. Normative References . . . . .	12
7.2. Informative References . . . . .	12
Appendix A. Comparison of this Specification with RFC8784 . . .	13
Author's Address . . . . .	14

## 1. Introduction

The Internet Key Exchange protocol version 2, defined in [RFC7296], is used in the IPsec architecture for performing authenticated key exchange. An extension to IKEv2 for mixing preshared keys for post-quantum security is defined in [RFC8784]. This extension allows today's IPsec traffic to be protected 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, the consensus in the IPsecME WG was that the IPsec traffic was more important to be protected than the IKE traffic. It was believed that information transferred over IKE SA (including peers' identities) is less important and extending the protection to also cover initial IKE SA would require serious modifications to core IKEv2 protocol. One of the goals was to minimize such changes. It was also decided that

immediate rekey of initial IKE SA would add this protection to the new IKE SA (albeit it would not provide protection of the identity of the peers).

However, in some situations it is desirable to have this protection for the IKE SA from the very beginning, when an initial IKE SA is created. An example of such a situation is the Group Key Management protocol using IKEv2, defined in [I-D.ietf-ipsecme-g-ikev2]. In this protocol group policy and session keys are transferred from a Group Controller/Key Server (GCKS) to the Group Members (GM) immediately once an initial IKE SA is created. While session keys are additionally protected with a key derived from SK\_d (and thus are immune to quantum computers if PPKs [RFC8784] are employed), the other sensitive data, including group policy, is not.

Another issue with using PPKs as it is defined in [RFC8784] is that this approach 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 PPKs as defined in [RFC8784] when they are changed relatively frequently, for example via the use 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 one from scratch using the fresh PPK.

Some time after the protocol extension for mixing preshared keys in IKEv2 for post-quantum security was defined in [RFC8784], a new IKE\_INTERMEDIATE exchange for IKEv2 [RFC9242] was developed. 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 defines the use of PPKs in the IKE\_INTERMEDIATE exchange of IKEv2 for post-quantum security, which allows getting full protection against quantum computers for initial IKE SA.

This specification also defines the use of PPKs in the CREATE\_CHILD\_SA exchange for creating additional IPsec SAs and for rekeying of IKE and IPsec SAs. This allows implementations to leverage fresh PPKs without the need to delete IKE SA and create it from scratch.

This specification does not replace the approach defined in RFC 8784. Both approaches for using PPKs in IKEv2 can be used depending on the circumstances (see Appendix A).

## 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.

This document uses the terms defined in [RFC7296]. In particular, readers should be familiar with the terms "initiator" and "responder" as used in that document.

The approach defined in RFC 8784 is referred to as "using PPKs in the IKE\_AUTH exchange" or simply "using PPKs in IKE\_AUTH" throughout this document.

## 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\_INT 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_INT)	<div style="text-align: right;">           ----&gt;            &lt;--- HDR, SAr1, KEr, Nr, [CERTREQ,]                  N(INTERMEDIATE_EXCHANGE_SUPPORTED),                  N(USE_PPK_INT)         </div>

The USE\_PPK\_INT is a Status Type IKEv2 notification. Its Notify Message Type is <TBA1 by IANA>, Protocol ID and SPI Size are both set to 0. This specification does not 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 in the notification they do not understand.

Note that this negotiation is independent from negotiation of using PPKs as specified in [RFC8784]. An initiator that supports both the use of PPKs in IKE\_AUTH [RFC8784] and in IKE\_INTERMEDIATE MAY include

both the USE\_PPK\_INT and the USE\_PPK notifications if configured to so. However, if the responder supports both specifications and is configured to use PPKs, it has to choose one to use, thus it MUST return either USE\_PPK\_INT or USE\_PPK notification in the response, but not both.

If the initiator did not propose using this extension in the IKE\_SA\_INIT request and responder's policy mandates protecting initial IKE SA with a PPK, then the responder MUST return the NO\_PROPOSAL\_CHOSEN notification.

If the negotiation was successful, the initiator includes one or more PPK\_IDENTITY\_KEY notification into the IKE\_INTERMEDIATE request with PPK identities the initiator believes are appropriate for the IKE SA being created,

The PPK\_IDENTITY\_KEY is a Status Type IKEv2 notification. Its Notify Message Type is <TBA2 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.

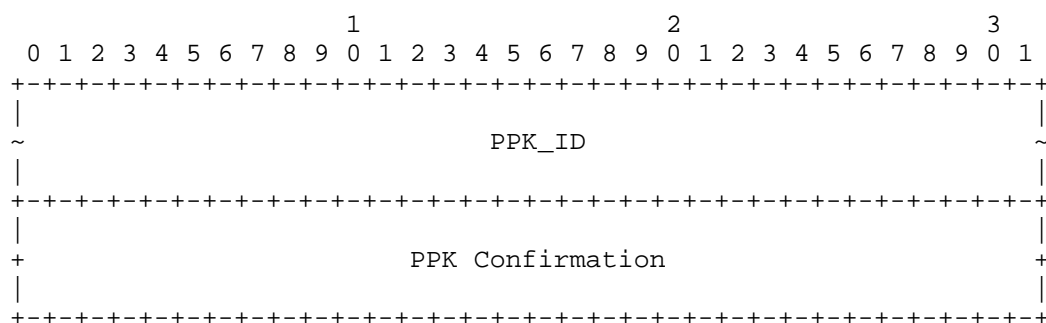


Figure 1: PPK\_IDENTITY\_KEY Notification Data Format

Where:

- \* PPK\_ID (variable) -- PPK\_ID as defined in Section 5.1 of [RFC8784]. The receiver can determine the length of PPK\_ID by subtracting 8 (the length of PPK Confirmation) from the Notification Data length.
- \* 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.

1. 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 IKEv2 [RFC7296]. However, the keys for the IKE SA are computed using PPK, as described in Section 3.1.1. If the responder returns a PPK identity that was not proposed by the initiator, then the initiator MUST treat this as a fatal and abort the IKE SA establishment.

2. If the responder does not have any of the PPKs which IDs were sent by the initiator or it has some of the 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)}

3. If the responder does not have any PPKs proposed by the initiator or it has some of the 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 does not 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 the IKE SA. Otherwise, the initiator continues with the IKE\_AUTH exchange as described in IKEv2 [RFC7296].

Table 1 summarizes the above logic for the responder:

Received USE_PPK_INT	Supports USE_PPK_INT	Has one of proposed PPKs	PPK is mandatory for initial IKE SA	Action
No	*	*	No	[RFC8784] (if proposed) or standard IKEv2 protocol
No	Yes	*	Yes	Send NO_PROPOSAL_CHOSEN
Yes	Yes	Yes	*	Section 3.1, Paragraph 16, Item 1 (use this extension)
Yes	Yes	No	Yes	Section 3.1, Paragraph 16, Item 2 (abort negotiation)
Yes	Yes	No	No	Section 3.1, Paragraph 16, Item 3 (standard IKEv2 protocol)

Table 1: Responder's behavior

Since the responder selects a 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, the responder MAY continue creating IKE SA using the negotiated "wrong" PPK if this is acceptable according to its local policy.



### 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 the PPK MUST be done after all of them, so that recalculating IKE SA keys with the PPK is the last action before they are used in the IKE\_AUTH exchange.

The IKE SA keys are computed differently compared to how PPKs are used in IKE\_AUTH. 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'.

$$\text{SKEYSEED}' = \text{prf+}(\text{PPK}, \text{SK}_d)$$

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

$$\{\text{SK}_d \mid \text{SK}_{ai} \mid \text{SK}_{ar} \mid \text{SK}_{ei} \mid \text{SK}_{er} \mid \text{SK}_{pi} \mid \text{SK}_{pr}\} \\ = \text{prf+}(\text{SKEYSEED}', \text{Ni} \mid \text{Nr} \mid \text{SPIi} \mid \text{SPIr})$$

In the formula above, Ni and Nr are nonces from the IKE\_SA\_INIT exchange, and SPIi and SPIr are the 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 an IKE SA is active, peers MAY use this fresh PPK without creating a new IKE SA from scratch when they have a need to create additional IPsec SAs or to rekey existing SAs. In this case the PPK can be used for creating additional IPsec SAs and for rekeying both IKE and IPsec SAs regardless whether the current IKE SA was created with the use of a PPK (no matter how: in IKE\_AUTH, in IKE\_INTERMEDIATE or in CREATE\_CHILD\_SA) or not.

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 PPK Confirmation

field in this case contains the first 8 octets of a string computed as  $\text{prf}(\text{PPK}, \text{Ni} \parallel \text{SPIi} \parallel \text{SPIr})$ , where  $\text{Ni}$  is the initiator's nonce from the `CREATE_CHILD_SA` request and  $\text{SPIi}/\text{SPIr}$  - SPIs of the current IKE SA. If the responder supports using PPKs in the `CREATE_CHILD_SA` exchange and is configured and ready to do it, then it sends back the `PPK_IDENTITY` notification containing the ID of the selected PPK, as depicted in figures below.

Initiator	Responder
-----	
HDR, SK { [N(REKEY_SA),] SA, Ni, [KEi,] TSr,	
N(PPK_IDENTITY_KEY, PPK_ID_1)	
[, N(PPK_IDENTITY_KEY, PPK_ID_2)] ...	
[, N(PPK_IDENTITY_KEY, PPK_ID_n)] }   --->	
	<--- HDR, SK {SA, Nr [KEr,] TSr, TSr,
	N(PPK_IDENTITY, PPK_ID_i) }

Figure 2: `CREATE_CHILD_SA` Exchange for Creating or Rekeying Child SAs

Initiator	Responder
-----	
HDR, SK {SA, Ni, KEi,	
N(PPK_IDENTITY_KEY, PPK_ID_1)	
[, N(PPK_IDENTITY_KEY, PPK_ID_2)] ...	
[, N(PPK_IDENTITY_KEY, PPK_ID_n)] }   --->	
	<--- HDR, SK {SA, Nr, KEr,
	N(PPK_IDENTITY, PPK_ID_i) }

Figure 3: `CREATE_CHILD_SA` Exchange for Rekeying IKE SA

In case the responder does not support (or is not configured for) using PPKs in the `CREATE_CHILD_SA` exchange, or does not 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 does not include any `PPK_IDENTITY` notification in the response and new SA is created as defined in IKEv2 [RFC7296]. If this is inappropriate for the initiator, it can immediately delete this SA.

If using PPKs in `CREATE_CHILD_SA` is mandatory for the responder and the initiator does not include any `PPK_IDENTITY_KEY` notification in the request or the responder does not 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 the responder MUST return the `NO_PROPOSAL_CHOSEN` notification.

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:

$$SK\_d' = \text{prf+} (PPK, SK\_d)$$

The resulting 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]. Unlike using PPKs in IKE\_AUTH, 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 the PPK is used in authentication too, this exchange is quantum secure even against an 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 Status Types" registry:

<TBA1>	USE_PPK_INT
<TBA2>	PPK_IDENTITY_KEY

## 6. Acknowledgements

Author would like to thank Paul Wouters for valuable comments and Tero Kivinen who made a thorough review of the document and proposed a lot of text improvements, and who also pointed out to the problem of mismatched preshared keys. Thanks to Rebecca Guthrie for providing comments and proposals for the document and to Mikhail Borodin for discovering the problem of calculating PPK Confirmation in CREATE\_CHILD\_SA.

## 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. Smysov, "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] Smysov, 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] Smysov, V. and B. Weis, "Group Key Management using IKEv2", Work in Progress, Internet-Draft, draft-ietf-ipsecme-g-ikev2-22, 16 March 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-ipsecme-g-ikev2-22>>.

[RFC9370] Tjhai, C.J., Tomlinson, M., Bartlett, G., Fluhner, 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 of this Specification with RFC8784

This specification is not intended to be a replacement for using PPKs in IKE\_AUTH as defined in [RFC8784]. Instead, it is supposed to be used in situations where the approach defined there does not meet the requirements, like the need to make the initial IKE SA quantum-secure or the need to choose between several available PPKs. However, if the peers support both using PPKs in IKE\_AUTH and this specification, then the latter may also be used in situations where using PPKs in IKE\_AUTH suffices (e.g., when initial IKE SA is not required to be quantum-protected).

The approach defined in this document has the following advantages:

1. The main advantage of using PPK in the IKE\_INTERMEDIATE exchange instead of the IKE\_AUTH exchange is that it allows IKE\_AUTH to be fully protected. This means that the ID payloads and any other sensitive content sent in the IKE\_AUTH are protected against quantum computers. The same is true for the sensitive data sent in the GSA\_AUTH exchange is the G-IKEv2 protocol [I-D.ietf-ipsecme-g-ikev2].
2. In addition to the IKE\_AUTH exchange being fully protected, the initial IKE SA is also fully protected, which is important when sensitive information is transferred over initial IKE SA. Examples of such situation are the CREATE\_CHILD\_SA exchange of IKEv2 and the GSA\_REGISTRATION exchange of G-IKEv2 [I-D.ietf-ipsecme-g-ikev2].
3. As the PPK exchange happens as separate exchange before IKE\_AUTH this means that initiator can propose several PPKs and responder can pick one. This is not possible when PPK exchange happens in the IKE\_AUTH. This feature could simplify PPK rollover.
4. 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 always requires an additional round trip (the IKE\_INTERMEDIATE exchange) to set up IKE SA and initial IPsec SA.

However, if the IKE\_INTERMEDIATE exchange has to be used for some other purposes in any case, then the PPK related payloads 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](mailto:svan@elvis.ru)