Network
Internet-Draft
Intended status: Standards Track
Expires: 18 April 2026

P. Wouters
Aiven
V. Smyslov
ELVIS-PLUS

15 October 2025

IKEv2 Support for Child SA PFS Policy Information
draft-pwouters-ipsecme-child-pfs-info-02

Abstract

This document defines an extension for the Internet Key Exchange Protocol Version 2 (IKEv2) to support negotiation at the time of initial Child Security Association (SA) establishing of Key Exchange (KE) method that could be used in subsequent rekeys of this SA.

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 18 April 2026.

Table of Contents

1. Introduction

The IKEv2 [RFC7296] protocol uses the Key Exchange (KE) payload to perform an ephemeral key exchange. During an IKEv2 rekey operations, a new KE payload is used to create a new ephemeral key, resulting in Perfect Forward Secrecy (PFS).

A Child Security Association (SA) optionally uses its own PFS settings by including its own KE payload and list of acceptable Key Exchange methods. During Child SA rekeys, KE payloads of acceptable Key Exchange methods are exchanged to create PFS.

The Initial Exchanges establish both an IKE SA and a Child SA using the Key Exchange method negotiated for the IKE SA. Thus, after the Initial exchanges, the peers are not aware of each other PFS requirements for the existing Child SA. It is common practice to either not perform PFS for Child SAs, or to only perform the same KE methods for both the IKE SA and all Child SAs. The situation is even more complex when Post-Quantum Key Exchange methods are used that might contain multiple KE payloads, which might not all be desired for rekeying the Initial Child SA. It is currently not possible to know the desired PFS configuration for rekey of the initial Child SA. The peers find out about this problem only at the first Child SA rekey, which can be substantially (several hours) later than initial Child SA is created.

This document defines a method for peers to negotiate a Key Exchange method that is compliant with peers' Child SA PFS policy at the time an initial Child SA is being established.

1.1. 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.

The document uses a term "initial Child SA" to refer to a Child SA created in the IKE_AUTH exchange. A Child SA created in the CREATE_CHILD_SA exchange cannot be considered as "initial Child SA" even if it is the first ever Child SA created in the IKE SA (e.g., in case of childless IKE SA [RFC6023]).

2. Protocol Overview

In IKEv2, when an IKE SA is created, an initial Child SA is also created as described in [RFC7296]. Negotiation of the Child SA parameters for the initial Child SA is different than for any other Child SA. In particular, since the IKE_AUTH exchange does not contain Key Exchange (KE) payloads, key exchange method for initial Child SA cannot be negotiated. Section 1.2 of [RFC7296] states that the SA payloads in the IKE_AUTH exchange cannot contain Transform Type 4 (Key Exchange Method) with any value other than NONE (and that [RFC7296] recommends implementations to omit the whole transform substructure if its Transform ID is NONE). This is the only difference in the negotiation of Child SA parameters between the IKE_AUTH (for initial Child SA) and the CREATE_CHILD_SA (for all other Child SA) exchanges.

In order to allow peers to negotiate the Key Exchange method for use in successive rekey operations during initial IKE exchanges, this document allows supporting peers to negotiate key exchange method (or methods in case of multiple key exchanges [RFC9370]) in the IKE_AUTH exchange as they would do it in the CREATE_CHILD_SA exchange. Note, that this negotiation does not affect the way session keys are derived for an initial Child SA and is only purposed for agreeing on the mutually acceptable key exchange method for successive rekey of initial Child SA.

3. Protocol Details

To be able to use this extensions peers first need to negotiate support for it. This is done in the IKE_SA_INIT exchange by exchanging new notification CHILD_SA_PFS_INFO_SUPPORTED (<TBA>). The Protocol ID and SPI Size fields of this notification are set to 0, the Notification Data is empty. The initiator wishing to use this extension includes this notification in the IKE_SA_INIT request. If

the responder receives the CHILD_SA_PFS_INFO_SUPPORTED and supports this extension, it sends this notification back in the response.

If peers successfully exchanged the CHILD_SA_PFS_INFO_SUPPORTED notification in the IKE_SA_INIT exchange then the initiator MAY include the Key Exchange Method (KE) transform(s) that it is wishing to use for subsequent rekey operations according to its Child SA PFS policy in the SA payload in the IKE_AUTH exchange. Additional Key Exchange Method (AKE*) transforms are also included if the initiator proposes multiple key exchanges [RFC9370]. The responder selects one of the proposed KE (and each of AKE*, if present) transform according to its Child SA PFS policy and returns back its selection in the response along with transforms of other types, as specified in Section 2.7 of [RFC7296]. If the responder fails to find mutually acceptable set of transforms then it returns the NO_PROPOSAL_CHOSEN notification and the initial Child SA is not created, as defined in Section 2.7 of [RFC7296].

Note that this extension may cause initial Child SA to fail even when it would be created if peers didn’t use this extension (in situation when no peer’s Child SA PFS policies have no mutually acceptable key exchange methods). For this reason, if any of the peers does not intends to rekey the initial Child SA (e.g., it plans to create a short-lived Child SA), then this peer SHOULD NOT negotiate support for this extension in the IKE_SA_INIT exchange, so that the extension is not used.

The negotiated key exchange method along with additional key exchange methods (if any) are not used in the key derivation for the initial Child SA. Instead, peers keep this information for later use. When one of the peers wishes to rekey the initial Child SA, it SHOULD propose the negotiated KE transform and AKE* transforms (if they were negotiated) in the SA payload in the CREATE_CHILD_SA exchange. In this case the proposing host can be sure that the peer supports this key exchange method and these additional key exchange methods (if any). Note, that other KE (and AKE*) transforms MAY additionally be proposed in this case, for example when the Child SA PFS policy has been updated.

4. Operational Considerations

This document is a result of cases from operational experience that have shown peers can run into broken IPsec connections at rekey time. These are not obvious to the administrators as these usually do not sit around for a few hours to wait and see if the rekey process worked successfully. The method defined in this document results in immediate negotiation failure that can be repaired before taking the IPsec connection into production.

During rekey, the cryptographic strength of a rekeyed Child SA SHOULD remain at least as strong as the Child SA being rekeyed. In practice this often means the negotiated algorithms remain the same. But some deployments use stronger settings for the IKE SA compared to its Child SAs, which means technically the initial Child SA uses a stronger KE method than for rekeys. The negotiation of KE method during initial Child SA establishing exposes such settings to the peers at the time IKE SA is being created, and peers can at that time accept or reject the child proposal. Once the KE method is negotiated during initial Child SA establishing, rekey proposals using this method are guaranteed to be acceptable to both parties.

Deployments with a large number of Child SAs often use no PFS for their Child SAs. It is computationally much cheaper to establish the large number of Child SAs and then immediately rekey the IKE SA. This method can also be used if the peer's Child SA KE methods are unacceptable. If both peers accept the KE method of 0 (NONE), it can decide to rekey the Child SA without PFS and immediately rekey the IKE SA using its accepted KE method to gain PFS on the Child SA.

5. Security Considerations

This document introduces no new security considerations, as it only causes an increased awareness of peer capabilities with respect to KE methods.

6. Implementation Status

[Note to RFC Editor: Please remove this section and the reference to [RFC7942] before publication.]

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in [RFC7942]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

According to [RFC7942], "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

Authors are requested to add a note to the RFC Editor at the top of this section, advising the Editor to remove the entire section before publication, as well as the reference to [RFC7942].

7. IANA Considerations

This document defines one new IKEv2 Notify Message Type payload for the IANA "IKEv2 Notify Message Status Types" registry.

Value	Notify Message Status Type	Reference
TBA	CHILD_SA_PFS_INFO_SUPPORTED	[this document]

8. References

8.1. 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>.

[RFC7296] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., and T. Kivinen, "Internet Key Exchange Protocol Version 2 (IKEv2)", STD 79, RFC 7296, DOI 10.17487/RFC7296, October 2014, <https://www.rfc-editor.org/info/rfc7296>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

8.2. Informative References

[RFC6023] Nir, Y., Tschofenig, H., Deng, H., and R. Singh, "A Childless Initiation of the Internet Key Exchange Version 2 (IKEv2) Security Association (SA)", RFC 6023, DOI 10.17487/RFC6023, October 2010, <https://www.rfc-editor.org/info/rfc6023>.

[RFC7942] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", BCP 205, RFC 7942, DOI 10.17487/RFC7942, July 2016, <https://www.rfc-editor.org/info/rfc7942>.

[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>.

Authors' Addresses

Paul Wouters
Aiven
Email: paul.wouters@aiven.io

Valery Smyslov
ELVIS-PLUS
Email: svan@elvis.ru