Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 17 August 2026

V. Smyslov
ELVIS-PLUS

13 February 2026

Using IKE Fragmentation for Large Messages
draft-smyslov-ipsecme-ikev2-fragm-large-msg-01

Abstract

This document describes describes issues with using Internet Key Exchange version 2 (IKEv2) fragmentation for transmitting large messages on unreliable transport. The document proposes several approaches for dealing with these issues: randomizing the order the fragments are sent, dispersing sending fragments over time and selective retransmission of fragments based on information from the peer about their receipt.

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 17 August 2026.

Copyright Notice

Table of Contents

1. Introduction

The Internet Key Exchange version 2 (IKEv2) protocol [RFC7296] is used for key management in IPsec architecture. IKEv2 was originally defined on UDP transport, and, while later IKEv2 extensions added TCP as a possible transport ([RFC9329], [I-D.ietf-ipsecme-ikev2-reliable-transport]), UDP is still a preferred and mostly used transport for IKEv2.

When UDP is used as a transport, any IKEv2 message that exceeds MTU size is fragmented at IP layer. IP fragmentation is known to cause problems with some intermediate devices that cannot correctly procesess any IP fragments other than the first one. To deal with this, a protocol extension was developed that allows fragmenting messages at IKE layer [RFC7383]. While IKEv2 fragmentation allows to avoid IP fragmentation of large messages, it lacks any congestion control mechanisms, that may cause issues when the fragmented message

is large (with some definition of "large") and the network (or receiver's) capacity is limited (with some definition of "limited"). This document defines several approaches that can be used to mitigate these issues.

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.

All multi-octet fields representing integers in this document are laid out in big endian order (also known as "most significant byte first", or "network byte order").

3. Issues with Using IKE Fragmentation for Large Messages

At the time the IKE fragmentation mechanism was being developed, it was considered that most IKEv2 messages would fit into the typical MTU size and only few of them could exceed it. In particular, the IKE_AUTH messages containing certificates were concerned and it was assumed at that time that "large" IKEv2 messages would be less than few Kbytes in size, so that the number of IKE fragments for a message (with a typical MTU size) would be small. Due to these considerations it was decided that no mechanism for acknowledging of receipt of individual fragments is needed - all fragments of a message are transmitted (and retransmitted) at once, as with IP fragmentation.

When postquantum cryptographic mechanisms started to be incorporated into IKEv2, the notion of "large" for an IKEv2 message has changed from few Kbytes to several tens of Kbytes [I-D.wang-ipsecme-hybrid-kem-ikev2-frodo] and up to several hundreds of Kbytes [I-D.smyslov-ipsecme-ikev2-mceliece]. When messages of this size are fragmented and all fragments are sent at once, it can happen that either intermediate network devices or the final recipient are uncapable of handling that much data at the rate it was sent, causing some fragments to be lost. Sender will eventually retransmit all the message fragments, but since the disproportion between sending rate and network/recipient limitations remain, it is likely that some fragments will be lost again. Since the restransmissions are not adaptive, there is no guarantee that the message will eventually be drlivered even after several retransmissions.

4. Proposed Techniques

This document defines several techniques that can be used to improve reliability of transmitting large messages using IKEv2 fragmentation mechanism. These techniques can be used independently of each other. The techniques can be classified as unilateral actionsm, which do not require any actiona from the peer, and bilateral actions that require mutual support by both sender and receiver.

4.1. Unilateral Techniques

4.1.1. Sending Fragments in different Order

The simplest technique is to change the order in which fragments are sent with each retransmission. This will help in situation the recipient have limited buffer size for incoming packets and cannot process the received fragments quickly enough to free the buffer. This means that the some number of the first sent fragments will be processed while the rest will be dropped. Changing the order in which the fragments are sent with each retransmission ensures that the first sent packets will change each time, thus increasing the chances all fragments are delivered.

4.1.2. Reducing Sending Rate

Another simple technique is to add some delay between sending each fragment, thus reducing the rate with which data is being transmitted. Since the sender does not know what rate is OK for the receiver, this document gives no advice what this delay should be. If the sender uses exponential back-off when retransmitting, a sensible approach would be to also increase delay between sending fragments with each retransmission.

4.1.3. Reducing the Amount of Data Sent by Initiator

In IKEv2 it is always the initiator that plays an active role in ensuring that both the request and the response messages of an exchange are delivered (Section 2.1 of [RFC7296]). In particular, the initiator periodically retransmits the request until it receives the response. With IKE fragmentation in case the response message is fragmented, a situation is possible that the initiator receives only some of these fragments. In this situation the initiator must retransmit the request, but if the request message has been fragmented too (the usual case), there is no need to retransmit all fragments of the request message, it is enough to retransmit only the first one (i.e., with the Fragment Number field equal to 1). This works because only this fragment triggers re-sending the response (as specified in Section 2.6.1 of [RFC7383], all other fragments are

discarded and would only waste network resources.

4.2. Bilateral Techniques

4.2.1. Selective Retransmission of Fragments

Selective Retransmission of Fragments is a protocol extension that allows, when supported by both peers, to selectively retransmit only those fragments that weren't delivered.

4.2.1.1. Negotiation

No negotiation is required to use this extension. See Appendix A.

4.2.1.2. Handling Fragmented Request

Initially the initiator sends all fragments as defined in [RFC7383]. If the responder supports this extension and it received only some of the sent fragments, then the responder can indicate which fragments are missing. For this purpose, it sends an IKEv2 response message formatted in accordance with Section 4.2.1.4.

4.2.1.3. Handling Fragmented Response

The initiator keeps retransmitting the request message (fragmented or not, with selective fragment retransmission or not) until it receives at least one response fragment message. When this happens and if there are some missing response message fragments, the initiator supporting this extension can indicate which fragments are missing. For this purpose, it sends an IKEv2 request message for the same exchange (with the same Message ID) formatted in accordance with Section 4.2.1.4.

4.2.1.4. Receipt Status Message

Receipt Status Message is an IKEv2 message that contains only the Encrypted Fragment payload (see Section 2.5 of [RFC7383]) formatted as follows:

* The Next Payload field is set to 0.

* The Fragment Number field is set to 0xffff.

* The Total Fragments field is set to 0xffff.

* The Encrypted content field contains the information about the missing fragments represented in the form defined in Figure 1.

All other fields are filled in accordance with [RFC7383] and [RFC7296]. However, when the Integrity Checksum Data is calculated (perhaps as part of AEAD encryption), the Fragment Number field is temporary set to zero. Once the ICV calculation is done, this field is restored to 0xffff and the message is sent.

The formatting of this message indicates that it is the last fragment of the message fragmented to 65535 parts. If the receiver of this message does not support this extension, it will interpret this message as above, but the message will have no effect since its ICV check will fail.

If the receiver supports this extension, it will recognize that this message contains the receipt status data, since it is unlikely that any real IKEv2 message sent over UDP is fragmented to 65535 fragments. In this case the receiver will check the ICV making sure that the Fragment Number field is set to zero before this. If the the ICV check passes and the message is successfully decrypted, then the receiver extracts the receipt status data and obtains the information about the fragments that need to be retransmitted.

If this ICV check fails, then the receiver MAY re-run the ICV check after restoring the Fragment Number field to its original value 0xffff. This is to handle the extremely rare case when the sender of this message does not support this extension and indeed fragmented the IKEv2 message into 65535 fragments, and it happened that the last fragment come first.

> | I admit that the trick with invalid ICV is a protocol hack.
> | Mea culpa. See Appendix A.

The content of the Encrypted Fragment payload contains the receipt status data in the following format.

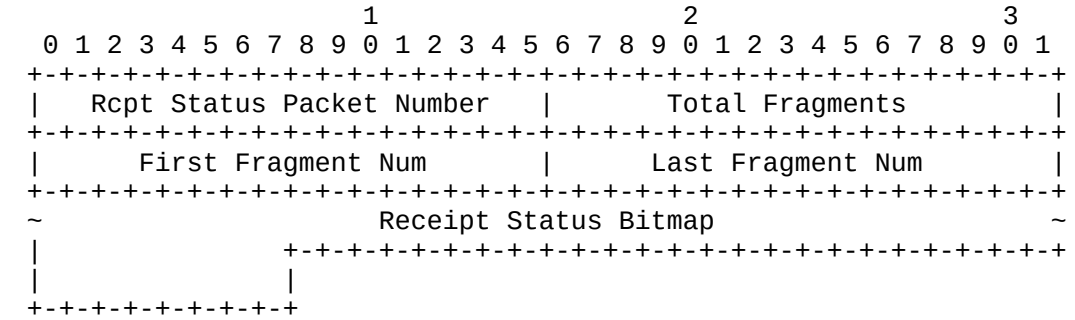

Figure 1: Receipt Status Data Format

* Rcpt Status Packet Number (2 octets, unsigned integer) -- the number of this receipt status packet. Each new receipt status packet MUST have this value greater than previous receipt status packets sent by the host in the context of the current IKEv2 message exchange, i.e. with the same IKE SPIs, Exchange Type and Message ID. This field MUST NOT be zero, packets with zero value MUST be ignored on receipt.

* Total Fragments (2 octets, unsigned integer) -- when sending, this field MUST be set to the value from the Total Fragments field in the Encrypted Fragment payloads (see Section 2.5 in [RFC7383] that the sender of this packet currently uses to reconstruct the fragmented message. On receipr, if the value in this field is not equal to the value in the Total Fragments field in the in the Encrypted Fragment payload most recently sent to the peer, then this receipt status packet MUST be silently ignored.

* First Fragment Num (2 octets, unsigned integer) -- the number of the fragment that corresponds to the first significant bit (the leftmost bit of the first octet) in the Receipt Status Bitmap. This field MUST NOT be zero and MUST NOT be greater than the value in the Total Fragments field. If these conditions are not met in the received packet, the packet MUST be silently discarded.

* Last Fragment Num (2 octets, unsigned integer) -- the number of the fragment that corresponds to the last significant bit in the Receipt Status Bitmap.This field MUST NOT be zero, MUST NOT be smaller than the value in the First Fragment Num field and MUST NOT be greater than the value in the Total Fragments field. If these conditions are not met in the received packet, the packet MUST be silently discarded.

* Receipt Status Bitmap (variable) -- this field contains a bitmap that represents receipt status for fragments with numbers starting from the value in the from First Fragment Num field till the value from the Last Fragment Num field. The bitmap is interpreted as starting from the most significant (the leftmost) bit of the first octet of the field. The length of the bitmap in octets is calculated as: ((Last Fragment Num) - (First Fragment Num) / 8) + 1. The last octet of the bitmap may contain some unused bits, these bits are ignored. Each bit in the bitmap represents the receipt status of the corresponding fragment - it is set to 1 if the fragment was received and successfully processed and set to 0 if the fragment is missing.

4.2.1.5. Implementation Details

TBD. Should discuss when to send messages with receipt status data, how to handle them, how to keep these messages small when the number of fragments is large, how to cope with PMTU discovery, etc.

5. Security Considerations

Security of IKEv2 and the IKEv2 fragmentation extension is discussed in [RFC7296] and [RFC7383] respectfully. Techniques proposed in this document do not affect security properties of the protocol.

6. IANA Considerations

This document makes no requests to IANA.

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>.

[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>.

7.2. Informative References

[RFC9329] Pauly, T. and V. Smyslov, "TCP Encapsulation of Internet Key Exchange Protocol (IKE) and IPsec Packets", RFC 9329, DOI 10.17487/RFC9329, November 2022, <https://www.rfc-editor.org/info/rfc9329>.

[I-D.ietf-ipsecme-ikev2-reliable-transport]
Smyslov, V. and T. Reddy.K, "Separate Transports for IKE and ESP", Work in Progress, Internet-Draft, draft-ietf-ipsecme-ikev2-reliable-transport-00, 6 October 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-ipsecme-ikev2-reliable-transport-00>.

[I-D.wang-ipsecme-hybrid-kem-ikev2-frodo]
WANG, G., Bruckert, L., and V. Smyslov, "Post-quantum Hybrid Key Exchange in IKEv2 with FrodoKEM", Work in Progress, Internet-Draft, draft-wang-ipsecme-hybrid-kem-ikev2-frodo-03, 24 December 2025, <https://datatracker.ietf.org/doc/html/draft-wang-ipsecme-hybrid-kem-ikev2-frodo-03>.

[I-D.smyslov-ipsecme-ikev2-mceliece]
Smyslov, V. and Y. Nir, "Using Classic McEliece in the Internet Key Exchange Protocol Version 2 (IKEv2)", Work in Progress, Internet-Draft, draft-smyslov-ipsecme-ikev2-mceliece-01, 6 October 2025, <https://datatracker.ietf.org/doc/html/draft-smyslov-ipsecme-ikev2-mceliece-01>.

[I-D.antony-ipsecme-ikev2-fragment-acknowledgment]
Antony, A., Klassert, S., and T. Brunner, "IKEv2 Fragment Acknowledgment Extension", Work in Progress, Internet-Draft, draft-antony-ipsecme-ikev2-fragment-acknowledgment-03, 21 January 2026, <https://datatracker.ietf.org/doc/html/draft-antony-ipsecme-ikev2-fragment-acknowledgment-03>.

Appendix A. Design Rationale for Selective Retransmission of Fragments Extension

The extension is not negotiated for the following reasons. First, there are currently many IKEv2 extensions, most of them are negotiated via exchange of some notifications. The number of extensions grows, thus the number of notifications exchanged in initial IKE exchange grows as well. This increases the size of the IKE_SA_INIT messages (not much, but still). Second, this extension is tightly tied to the IKEv2 fragmentation extension, thus, if it were negotiatable, peers should make sure that it is not negotiated alone, without IKEv2 fragmentation been negotiated too. Not a big deal, but still some additional checks. This could be avoided if RFC 7383 explicitly allowed notification data to be non-empty for future extensions (e.g., as in [I-D.ietf-ipsecme-ikev2-reliable-transport]), but that was not the case. On the other hand, it appeared that this extension can be defined in such a way, that no negotiation is needed

by only relying on incoming fragment message checks defined in [RFC7383].

It is possible to avoid the hack with intentionally invalid ICV if this extension defined that the Fragment Number field is set to zero (as it is currently defined for the purpose of ICV check). This would make the message invalid as per checks from Section 2.6 of [RFC7383], so the message would be dropped by unsupporting initiator. The only concern here is that some network intermediate devices could do a DPI of IKEv2 traffic and drop invalid messages, thus the message containing receipt status data is made looking valid on the wire and a hack with invalid ICV is used to force old implementations to discard it. But this is purely theoretical concert not backed up by any real data.

Acknowledgements

Author is grateful to Antony Antony, Steffen Klassert and Tobias Brunner, whose document "IKEv2 Fragment Acknowledgment Extension" [I-D.antony-ipsecme-ikev2-fragment-acknowledgment] resurrected the ideas that were in the author’s mind at the time RFC 7383 was being developed, but were abandoned at that time due to unnecessary (as was believed then) complexity.

Author’s Address

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