Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: July 29, 2018

V. Smyslov
ELVIS-PLUS

January 25, 2018

Auxiliary Exchange in the IKEv2 Protocol
draft-smyslov-ipsecme-ikev2-aux-00

Abstract

This documents defines a new exchange, called Auxiliary Exchange, for the Internet Key Exchange protocol Version 2 (IKEv2). This exchange can be used for transferring large amount of data in the process of IKEv2 Security Association (SA) establishment. Introducing Auxiliary Exchange allows to re-use existing IKE Fragmentation mechanism, that helps to avoid IP fragmentation of large IKE messages, but cannot be used in the initial IKEv2 exchange.

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 http://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 July 29, 2018.

Copyright Notice

Table of Contents

1. Introduction

The Internet Key Exchange protocol version 2 (IKEv2) defined in [RFC7296] uses UDP as a transport for its messages. If size of the messages is large enough, IP fragmentation may take place that may interfere badly with some network devices. The problem is described in more detail in [RFC7383], which also defines an extension to the IKEv2 called IKE Fragmentation. This extension allows IKE messages to be fragmented at IKE level, which eliminates possible issues caused by IP fragmentation. However, the IKE Fragmentation cannot be used in the initial IKEv2 exchange, IKE_SA_INIT. This limitation in most cases is not a problem, since the IKE_SA_INIT messages used to be small enough to not cause IP fragmentation.

Recent progress in Quantum Computing has brought a concern that classical Diffie-Hellman key exchange methods will become insecure in a relatively near future and should be replaced with Quantum Computer (QC) resistant ones. Currently most of QC-resistant key exchange methods have large public keys. If these keys are exchanged in the IKE_SA_INIT, then most probably IP fragmentation would take place, therefore all the problems caused by it would become inevitable.

A possible solution would be to use TCP as a transport for IKEv2, as described in [RFC8229]. However this approach has significant drawbacks and is intended to be a "last resort" when UDP transport is blocked by intermediate network devices.

This document defines a new exchange for the IKEv2 protocol, called Auxiliary Exchange or IKE_AUX. One or more these exchanges may take place right after the IKE_SA_INIT exchange and prior to the IKE_AUTH exchange. These exchanges may be used to exchange large amounts of data, which don't fit into the IKE_SA_INIT exchange without causing IP fragmentation. The IKE_AUX messages can be fragmented using IKE Fragmentation mechanism.

While ability to transfer large public keys of QC-resistant methods was a primary motivation for the Auxiliary Exchange, its application is not limited to this use case. This exchange may be used whenever large messages need to be exchanged before the IKE_AUTH exchange. It is expected that separate specifications will define how and when the IKE_AUX exchange is used in the IKEv2.

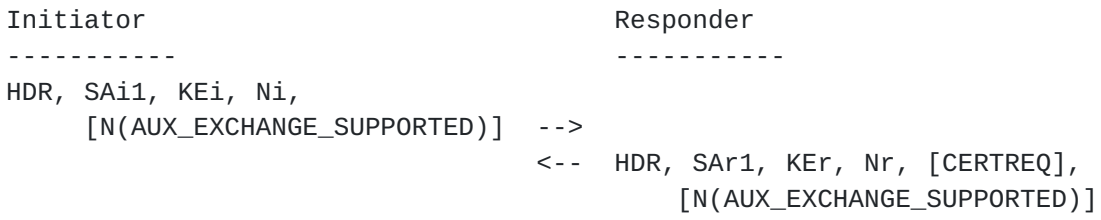
2. Terminology and Notation

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. Auxiliary Exchange Details

3.1. Support for Auxiliary Exchange Negotiation

The initiator indicates its support for Auxiliary Exchange by including a notification of type AUX_EXCHANGE_SUPPORTED in the IKE_SA_INIT request message. If the responder also supports this exchange, it includes this notification in the response message.



The AUX_EXCHANGE_SUPPORTED 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 this notification may contain, so the Notification Data is left empty. However, other specifications may override this. Implementations MUST ignore the non-empty Notification Data if they don't understand its purpose.

3.2. Using Auxiliary Exchange

If both peers indicated their support for Auxiliary Exchange, the initiator may use one or more these exchanges to transfer additional data, which are not fit into the IKE_SA_INIT exchange. Using the IKE_AUX exchange is optional, the initiator may find it unnecessary after completing the IKE_SA_INIT exchange.

The Auxiliary Exchange is denoted as IKE_AUX, its Exchange Type is <TBA by IANA>.

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

The initiator may use several IKE_AUX exchanges if necessary. Since initiator's Window Size is initially set to one (Section 2.3 of [RFC7296]), These exchanges MUST follow each other and MUST all be completed before the IKE_AUTH exchange is initiated. The IKE SA MUST NOT be considered as established until the IKE_AUTH exchange is successfully completed.

The Message IDs for the IKE_AUX exchanges MUST be chosen by the standard IKEv2 rule, described in the Section 2.2. of [RFC7296], i.e. it is set to 1 for the first IKE_AUX exchange, 2 for the next (if any) and so on. The message ID for the first pair of the IKE_AUTH messages is one more than the last IKE_AUX Message ID.

The content of the IKE_AUX messages depends on the data being transferred and will be defined by specifications utilizing this exchange. However, since the main motivation for IKE_AUX is to avoid IP fragmentation when large amount of data need to be transferred prior to IKE_AUTH, the Encrypted payload SHOULD be present in the IKE_AUX messages and payloads containing large data SHOULD be placed inside. This will allow IKE Fragmentation [RFC7383] to take place, provided it is supported by the peers and negotiated in the initial exchange.

3.3. Keying Material and Authentication

The keys SK_e and SK_a for the Encrypted payload in the IKE_AUX exchanges are computed in a standard fashion, as defined in the Section 2.14 of [RFC7296]. Note that this may be redefined by other specifications utilizing the IKE_AUX exchange (e.g. in case the IKE_AUX is used to exchange additional keys which must later be stirred into the SKEYSEED).

The data transferred in the IKE_AUX exchanges must be authenticated in the IKE_AUTH exchange. For this purpose the definition of the blob to be signed (or MAC'ed) from the Section 2.15 of [RFC7296] is modified as follows in case of at least one IKE_AUX exchange takes place:

```
InitiatorSignedOctets = RealMessage1 | AUX_I | NonceRData | MACedIDForI
AUX_I = ICV_INIT_1 | ICV_INIT_2 | ICV_INIT_3 ...

ResponderSignedOctets = RealMessage2 | AUX_R | NonceIData | MACedIDForR
AUX_R = ICV_RESP_1 | ICV_RESP_2 | ICV_RESP_3 ...
```

ICV_INIT_1, ICV_INIT_2, ICV_INIT_3, etc. represent the content of the Integrity Checksum Data field from the Encrypted payloads (or Encrypted Fragment payloads) from all the IKE_AUX messages sent by the initiator in an order of increasing MessageIDs (and increasing Fragment Numbers for the same Message ID). ICV_RESP_1 | ICV_RESP_2 | ICV_RESP_3 etc. are defined similarly for the messages sent by the responder.

3.4. IKE Fragmentation

If both peers indicated their support for IKE Fragmentation, then some additional restrictions are applied to ensure that the values of Integrity Checksum Data is unambiguous. These restrictions MUST be applied to the IKE_AUX exchanges only and MAY be lifted once all these exchanges are over.

The responder MUST send the IKE_AUX response in the same form (fragmented or not) as the request message. The initiator MUST NOT switch from unfragmented to fragmented request in a single IKE_AUX exchange - either the request is sent unfragmented and retransmitted until unfragmented response is received (applicable if message size is small and no IP fragmentation is expected), or the request is fragmented from the beginning of exchange. The initiator MAY however send either fragmented or unfragmented messages in different IKE_AUX exchanges. The initiator SHOULD use IKE Fragmentation if the size of request (or the expected size of response) is large enough to cause IP fragmentation. The PMTU discovery for IKE Fragmentation as defined in Section 2.5.2 of [RFC7383] MUST NOT be used for the IKE_AUX exchanges.

4. Interaction with other IKEv2 Extensions

The IKE_AUTH exchanges may be used in the IKEv2 Session Resumption [RFC5723] between the IKE_SESSION_RESUME and the IKE_AUTH exchanges.

5. Security Considerations

The data that is transferred by means of the IKE_AUX exchanges is not authenticated until the subsequent IKE_AUTH exchange is completed. However, if the data is placed inside the Encrypted payload, then it is protected from passive eavesdroppers. In addition the peers can be certain that they receives messages from the party he/she performed the IKE_SA_INIT with if they can successfully verify the Integrity Checksum Data of the Encrypted payload.

The main application for Auxiliary Exchange is to transfer large amount of data before IKE SA is set up without causing IP fragmentation. For that reason it is expected that in most cases IKE Fragmentation will be employed in the IKE_AUX exchanges. Section 5 of [RFC7383] contains security considerations for IKE Fragmentation.

Note, that if an attacker was able to break key exchange from the IKE_SA_INIT in real time (e.g. by means of Quantum Computer), then the security of IKE_AUX would degrage. In particular, such an attacker would be able both to read data contained in the Encrypted payload and to forge it. THe forgery would become evident in the IKE_AUTH exchange (provided the attacker caanot break employed authentication mechanism), but the ability to inject forged IKE_AUX messages with valid ICV would allow the attacker to mount Denial-of-Service attack.

6. IANA Considerations

This document defines a new Exchange Type in the "IKEv2 Exchange Types" registry:

<TBA>	IKE_AUX

This document also defines a new Notify Message Types in the "Notify Message Types - Status Types" registry:

<TBA>	AUX_EXCHANGE_SUPPORTED

7. Acknowledgements

The idea to use an intermediate exchange between IKE_SA_INIT and IKE_AUTH was first suggested by Tero Kivinen.

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>.

[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>.

8.2. Informative References

[RFC8229] Pauly, T., Touati, S., and R. Mantha, "TCP Encapsulation of IKE and IPsec Packets", RFC 8229, DOI 10.17487/RFC8229, August 2017, <https://www.rfc-editor.org/info/rfc8229>.

[RFC5723] Sheffer, Y. and H. Tschofenig, "Internet Key Exchange Protocol Version 2 (IKEv2) Session Resumption", RFC 5723, DOI 10.17487/RFC5723, January 2010, <https://www.rfc-editor.org/info/rfc5723>.

Author's Address

Valery Smyslov
ELVIS-PLUS
PO Box 81
Moscow (Zelenograd) 124460
RU

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