

**Intermediate Exchange in the IKEv2 Protocol**  
**draft-ietf-ipsecme-ikev2-intermediate-02**

Abstract

This documents defines a new exchange, called Intermediate 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 Intermediate Exchange allows re-using 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 <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 January 25, 2020.

Copyright Notice

Copyright (c) 2019 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 Simplified BSD License text as described in [Section 4](#).e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

<a href="#">1.</a>	<a href="#">Introduction</a>	<a href="#">2</a>
<a href="#">2.</a>	<a href="#">Terminology and Notation</a>	<a href="#">3</a>
<a href="#">3.</a>	<a href="#">Intermediate Exchange Details</a>	<a href="#">3</a>
<a href="#">3.1.</a>	<a href="#">Support for Intermediate Exchange Negotiation</a>	<a href="#">3</a>
<a href="#">3.2.</a>	<a href="#">Using Intermediate Exchange</a>	<a href="#">4</a>
<a href="#">3.3.</a>	<a href="#">The IKE_INTERMEDIATE Exchange Protection and Authentication</a>	<a href="#">5</a>
<a href="#">3.3.1.</a>	<a href="#">Protection of the IKE_INTERMEDIATE Messages</a>	<a href="#">5</a>
<a href="#">3.3.2.</a>	<a href="#">Authentication of the IKE_INTERMEDIATE Exchanges</a>	<a href="#">5</a>
<a href="#">3.4.</a>	<a href="#">Error Handling in the IKE_INTERMEDIATE Exchange</a>	<a href="#">8</a>
<a href="#">4.</a>	<a href="#">Interaction with other IKEv2 Extensions</a>	<a href="#">9</a>
<a href="#">5.</a>	<a href="#">Security Considerations</a>	<a href="#">9</a>
<a href="#">6.</a>	<a href="#">IANA Considerations</a>	<a href="#">10</a>
<a href="#">7.</a>	<a href="#">Acknowledgements</a>	<a href="#">10</a>
<a href="#">8.</a>	<a href="#">References</a>	<a href="#">10</a>
<a href="#">8.1.</a>	<a href="#">Normative References</a>	<a href="#">10</a>
<a href="#">8.2.</a>	<a href="#">Informative References</a>	<a href="#">11</a>
	<a href="#">Author's Address</a>	<a href="#">11</a>

## 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 takes 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, eliminating 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 not to cause IP fragmentation.

However, the situation has been changing recently. One example of the need to transfer large amount of data before IKE SA is created is using Quantum Computer resistant key exchange methods in IKEv2. 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 will take place, therefore all the problems caused by it will become inevitable.

Smyslov

Expires January 25, 2020

[Page 2]

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

This specification describes a way to transfer large amount of data in IKEv2 using UDP transport. For this purpose the document defines a new exchange for the IKEv2 protocol, called Intermediate Exchange or IKE\_INTERMEDIATE. One or more these exchanges may take place right after the IKE\_SA\_INIT exchange and prior to the IKE\_AUTH exchange. The IKE\_INTERMEDIATE exchange messages can be fragmented using IKE Fragmentation mechanism, so these exchanges may be used to transfer large amounts of data which don't fit into the IKE\_SA\_INIT exchange without causing IP fragmentation.

The Intermediate Exchange can be used to transfer large public keys of QC-resistant key exchange methods, but its application is not limited to this use case. This exchange can also be used whenever some data need to be transferred before the IKE\_AUTH exchange and for some reason the IKE\_SA\_INIT exchange is not suited for this purpose. This document defines the IKE\_INTERMEDIATE exchange without tying it to any specific use case. It is expected that separate specifications will define for which purposes and how the IKE\_INTERMEDIATE 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. Intermediate Exchange Details**

### **3.1. Support for Intermediate Exchange Negotiation**

The initiator indicates its support for Intermediate Exchange by including a notification of type INTERMEDIATE\_EXCHANGE\_SUPPORTED in the IKE\_SA\_INIT request message. If the responder also supports this exchange, it includes this notification in the response message.

Initiator	Responder
-----	-----
HDR, SAi1, KEi, Ni,	
[N(INTERMEDIATE_EXCHANGE_SUPPORTED)] -->	
	<-- HDR, SAR1, KEr, Nr, [CERTREQ],
	[N(INTERMEDIATE_EXCHANGE_SUPPORTED)]

Smyslov

Expires January 25, 2020

[Page 3]

The `INTERMEDIATE_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, future enhancements of this specification may override this. Implementations **MUST** ignore the non-empty Notification Data if they don't understand its purpose.

### 3.2. Using Intermediate Exchange

If both peers indicated their support for the Intermediate Exchange, the initiator may use one or more these exchanges to transfer additional data. Using the `IKE_INTERMEDIATE` exchange is optional, the initiator may find it unnecessary after completing the `IKE_SA_INIT` exchange.

The Intermediate Exchange is denoted as `IKE_INTERMEDIATE`, its Exchange Type is <TBA by IANA>.

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

The initiator may use several `IKE_INTERMEDIATE` 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_INTERMEDIATE` exchanges **MUST** be chosen according to the standard IKEv2 rule, described in the [Section 2.2. of \[RFC7296\]](#), i.e. it is set to 1 for the first `IKE_INTERMEDIATE` 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 one that was used in the last `IKE_INTERMEDIATE` exchange.

If the presence of NAT is detected in the `IKE_SA_INIT` exchange via `NAT_DETECTION_SOURCE_IP` and `NAT_DETECTION_DESTINATION_IP` notifications, then the peers **MUST** switch to port 4500 immediately once this exchange is completed, i.e. in the first `IKE_INTERMEDIATE` exchange.

The content of the `IKE_INTERMEDIATE` exchange messages depends on the data being transferred and will be defined by specifications utilizing this exchange. However, since the main motivation for the `IKE_INTERMEDIATE` exchange is to avoid IP fragmentation when large

Smyslov

Expires January 25, 2020

[Page 4]

amount of data need to be transferred prior to IKE\_AUTH, the Encrypted payload MUST be present in the IKE\_INTERMEDIATE exchange messages and payloads containing large data MUST 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. The IKE\_INTERMEDIATE Exchange Protection and Authentication**

#### **3.3.1. Protection of the IKE\_INTERMEDIATE Messages**

The keys SK\_e[i/r] and SK\_a[i/r] for the Encrypted payload in the IKE\_INTERMEDIATE exchanges are computed in a standard fashion, as defined in the [Section 2.14 of \[RFC7296\]](#). Every subsequent IKE\_INTERMEDIATE exchange uses the most recently calculated IKE SA keys before this exchange is started. So, the first IKE\_INTERMEDIATE exchange always uses SK\_e[i/r] and SK\_a[i/r] keys that were computed as a result of the IKE\_SA\_INIT exchange. If the first IKE\_INTERMEDIATE exchange performs additional key exchange resulting in the update of SK\_e[i/r] and SK\_a[i/r], then these updated keys are used for encryption and authentication of the next IKE\_INTERMEDIATE exchange, otherwise the current keys are used, and so on.

#### **3.3.2. Authentication of the IKE\_INTERMEDIATE Exchanges**

The content of the IKE\_INTERMEDIATE 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:

```
InitiatorSignedOctets = RealMsg1 | NonceRData | MACedIDForI [| IntAuth]
ResponderSignedOctets = RealMsg2 | NonceIDData | MACedIDForR [| IntAuth]
```

```
IntAuth = IntAuth_1 | [| IntAuth_2 [| IntAuth_3]] ...
```

```
IntAuth_1 = IntAuth_1_I | IntAuth_1_R
IntAuth_2 = IntAuth_2_I | IntAuth_2_R
IntAuth_3 = IntAuth_3_I | IntAuth_3_R...
```

```
IntAuth_1_I = prf(SK_pi_1, IntAuth_1_I_A [| IntAuth_1_I_P])
IntAuth_2_I = prf(SK_pi_2, IntAuth_2_I_A [| IntAuth_2_I_P])
IntAuth_3_I = prf(SK_pi_3, IntAuth_3_I_A [| IntAuth_3_I_P])
...
```

```
IntAuth_1_R = prf(SK_pr_1, IntAuth_1_R_A [| IntAuth_1_R_P])
IntAuth_2_R = prf(SK_pr_2, IntAuth_2_R_A [| IntAuth_2_R_P])
IntAuth_3_R = prf(SK_pr_3, IntAuth_3_R_A [| IntAuth_3_R_P])
...
```



Smyslov

Expires January 25, 2020

[Page 5]

IntAuth\_1\_I/IntAuth\_1\_R, IntAuth\_2\_I/IntAuth\_2\_R, IntAuth\_3\_I/IntAuth\_3\_R, etc. represent the results of applying the negotiated prf to the content of the IKE\_INTERMEDIATE messages sent by the initiator (IntAuth\*\_I) and by the responder (IntAuth\*\_R) in an order of increasing Message IDs (i.e. in an order the IKE\_INTERMEDIATE exchanges took place). The prf is applied to the two chunks of data: mandatory IntAuth\*\_[I/R]\_A and optional IntAuth\*\_[I/R]\_P. The IntAuth\*\_[I/R]\_A chunk lasts from the first octet of the IKE Header (not including prepended four octets of zeros, if port 4500 is used) to the last octet of the Encrypted Payload header. The IntAuth\*\_[I/R]\_P chunk is present if the Encrypted payload is not empty. It consists of the not yet encrypted content of the Encrypted payload, excluding Initialization Vector, Padding, Pad Length and Integrity Checksum Data fields (see 3.14 of [\[RFC7296\]](#) for description of the Encrypted payload). In other words, the IntAuth\*\_[I/R]\_P chunk is the inner payloads of the Encrypted payload in plaintext form.





Smyslov

Expires January 25, 2020

[Page 7]

in Figure 1) is set to the length of IntAuth\_\*\_[I/R]\_P plus the size of the Payload header (four octets).

The prf calculations MUST be applied to whole messages only, before possible IKE Fragmentation. This ensures that the IntAuth will be the same regardless of whether IKE Fragmentation takes place or not. This is important since [\[RFC7383\]](#) allows sending first unfragmented message and then resending it in fragmented form in case of no reply is received. If the message was received in fragmented form, it should be reconstructed before calculating prf as if it were received unfragmented. The RESERVED field in the reconstructed Encrypted Payload header MUST be set to the value of the RESERVED field in the Encrypted Fragment payload header from the first fragment (that with Fragment Number equal to 1).

Note that it is possible to avoid actual reconstruction of the message by incrementally calculating prf on decrypted fragments. However care must be taken to properly replace the content of the Next Header and the Length fields so that the result of computing prf is the same as if it were computed on reconstructed message.

Each calculation of IntAuth\_\*\_[I/R] uses its own keys SK\_p[i/r]\_\*, which are the most recently updated SK\_p[i/r] keys available before the corresponded IKE\_INTERMEDIATE exchange is started. The first IKE\_INTERMEDIATE exchange always uses SK\_p[i/r] keys that were computed in the IKE\_SA\_INIT as SK\_p[i/r]\_1. If the first IKE\_INTERMEDIATE exchange performs additional key exchange resulting in SK\_p[i/r] update, then this updated SK\_p[i/r] are used as SK\_p[i/r]\_2, otherwise the original SK\_p[i/r] are used, and so on. Note, that if keys are updated then for any given IKE\_INTERMEDIATE exchange the keys SK\_e[i/r] and SK\_a[i/r] used for its messages protection (see [Section 3.3.1](#)) and the keys SK\_p[i/r] for its authentication are always from the same generation.

### **3.4. Error Handling in the IKE\_INTERMEDIATE Exchange**

Since messages of the IKE\_INTERMEDIATE exchange are not authenticated until the IKE\_AUTH exchange successfully completes, possible errors need to be handled with care. There is a trade-off between providing a better diagnostics of the problem and a risk to become a part of DoS attack. See [Section 2.21.1](#) and 2.21.2 of [\[RFC7296\]](#) describe how errors are handled in initial IKEv2 exchanges, these considerations are also applied to the IKE\_INTERMEDIATE exchange.



#### **4. Interaction with other IKEv2 Extensions**

The IKE\_INTERMEDIATE exchanges MAY be used during the IKEv2 Session Resumption [[RFC5723](#)] between the IKE\_SESSION\_RESUME and the IKE\_AUTH exchanges. To be able to use it peers MUST negotiate support for intermediate exchange by including INTERMEDIATE\_EXCHANGE\_SUPPORTED notifications in the IKE\_SESSION\_RESUME messages. Note, that a flag whether peers supported the IKE\_INTERMEDIATE exchange is not stored in the resumption ticket and is determined each time from the IKE\_SESSION\_RESUME exchange.

#### **5. Security Considerations**

The data that is transferred by means of the IKE\_INTERMEDIATE 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 Intermediate 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\_INTERMEDIATE exchanges. [Section 5 of \[RFC7383\]](#) contains security considerations for IKE Fragmentation.

Note, that if an attacker was able to break key exchange in real time (e.g. by means of Quantum Computer), then the security of the IKE\_INTERMEDIATE exchange would degrade. 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 cannot break employed authentication mechanism), but the ability to inject forged the IKE\_INTERMEDIATE exchange messages with valid ICV would allow the attacker to mount Denial-of-Service attack. Moreover, if in this situation the negotiated prf was not secure against preimage attack with known key, then the attacker could forge the IKE\_INTERMEDIATE exchange messages without later being detected in the IKE\_AUTH exchange. To do this the attacker should find the same IntAuth\_\*\_[I|R] value for the forged message as for original.





## **6. IANA Considerations**

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

<TBA>           IKE\_INTERMEDIATE

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

<TBA>           INTERMEDIATE\_EXCHANGE\_SUPPORTED

## **7. Acknowledgements**

The idea to use an intermediate exchange between IKE\_SA\_INIT and IKE\_AUTH was first suggested by Tero Kivinen. Scott Fluhrer and Daniel Van Geest identified a possible problem with authentication of the IKE\_INTERMEDIATE exchange and helped to resolve it. Author is also grateful to Tobias Brunner for raising good points concerning authentication of the IKE\_INTERMEDIATE exchange.

## **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](mailto:svan@elvis.ru)

