

**Compression in the Internet Key Exchange Protocol Version 2 (IKEv2)**  
**draft-smyslov-ipsecme-ikev2-compression-00**

Abstract

This document describes a method for reducing the size of IKEv2 messages by means of lossless compression. Making IKEv2 messages smaller is desirable for low power consumption battery powered IoT devices. It also helps avoid IP fragmentation of IKEv2 messages. This document describes how compression is negotiated maintaining backward compatibility and how it is used in IKEv2.

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 June 27, 2016.

Copyright Notice

Copyright (c) 2015 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 (<http://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="#">3</a>
<a href="#">2.</a>	<a href="#">Requirements Language . . . . .</a>	<a href="#">5</a>
<a href="#">3.</a>	<a href="#">Protocol Description . . . . .</a>	<a href="#">6</a>
<a href="#">3.1.</a>	<a href="#">Using Compression in the IKE_SA_INIT Exchange . . . . .</a>	<a href="#">6</a>
<a href="#">3.2.</a>	<a href="#">Using Compression in Subsequent Exchanges . . . . .</a>	<a href="#">8</a>
<a href="#">4.</a>	<a href="#">Payload Formats . . . . .</a>	<a href="#">10</a>
<a href="#">4.1.</a>	<a href="#">Compressed Payload . . . . .</a>	<a href="#">10</a>
<a href="#">4.2.</a>	<a href="#">INVALID_COMPRESSION_ALGORITHM Notification . . . . .</a>	<a href="#">11</a>
<a href="#">5.</a>	<a href="#">Interaction with other IKEv2 Extensions . . . . .</a>	<a href="#">12</a>
<a href="#">5.1.</a>	<a href="#">Interaction IKE Fragmentation . . . . .</a>	<a href="#">12</a>
<a href="#">5.2.</a>	<a href="#">Interaction IKE Resumption . . . . .</a>	<a href="#">12</a>
<a href="#">6.</a>	<a href="#">Security Considerations . . . . .</a>	<a href="#">13</a>
<a href="#">7.</a>	<a href="#">IANA Considerations . . . . .</a>	<a href="#">14</a>
<a href="#">8.</a>	<a href="#">References . . . . .</a>	<a href="#">15</a>
<a href="#">8.1.</a>	<a href="#">Normative References . . . . .</a>	<a href="#">15</a>
<a href="#">8.2.</a>	<a href="#">Informative References . . . . .</a>	<a href="#">15</a>
	<a href="#">Author's Address . . . . .</a>	<a href="#">17</a>

## 1. Introduction

The Internet Key Exchange protocol version 2 (IKEv2) defined in [\[RFC7296\]](#) is used in the IP Security (IPsec) architecture for the purposes of Security Association (SA) parameters negotiation and authenticated key exchange. The protocol uses UDP as the transport for its messages. The size of the messages varies from hundreds bytes to several kBytes.

Sending large UDP messages may cause IP fragmentation to take place, which may interact badly with some Network Address Translators (NAT). One of the possible solutions to the problem is IKEv2 fragmentation described in [\[RFC7383\]](#). However, the IKEv2 fragmentation cannot be used for unencrypted messages and thus cannot be used in the initial IKEv2 exchange called IKE\_SA\_INIT exchange. Usually the messages of the IKE\_SA\_INIT exchange are relatively small and this restriction doesn't cause problems. However with adoption more and more new algorithms and new IKEv2 extensions there is a tendency for these messages to become larger and larger for the implementations that support new features.

The lossless compression can be used to reduce the size of IKEv2 messages. Each IKEv2 message contains different types of data structured in payloads. Depending on the type of payload the compressibility of the data it contains varies greatly. Some types of payloads, like the Nonce payload, contain data that are almost uncompressable. On the other hand, such payloads like the Security Association payload or Notification payload usually have a lot of redundancy in their encoding and hence are highly compressable. Since many emerging IKEv2 extensions add new type of notification or new parameter to the Security Association payload contained in the IKE\_SA\_INIT messages, the ability to compress these messages would help keep their size bounded.

Compression can also be applied to the messages followed the IKE\_SA\_INIT exchange. In this case the reduced size of the messages would make the necessity to use the IKEv2 fragmentation less likely or would decrease the number of fragments the messages are divided into, which would increase the protocol reliability and productivity.

The other field where using compression may be useful is the Internet of Things (IoT) devices utilizing a lower power consumption technology. For many such devices the power consumption for transmitting extra bits over network is much higher than the power consumption for spending extra CPU cycles to compress data before transmission. The [appendix A](#) of [\[IPSEC-IOT-REQS\]](#) gives some estimate data. Since many such devices are battery powered without an ability to recharge or to replace the battery which serves for the lifecycle



of the device (a few years), the task of reducing the power consumption for such devices is very important.

This document specifies how lossless compression is used in IKEv2. In order to enable compression in the IKE\_SA\_INIT exchange a new payload is introduced that contains other payloads in compressed form. The processing of the Encrypted payload is modified to accommodate compression in subsequent exchanges. The document also specifies how the use of compression is negotiated between the peers maintaining backward compatibility.

## **2. Requirements Language**

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [[RFC2119](#)].

### 3. Protocol Description

Compression is accommodated differently in the initial IKEv2 exchange and in subsequent exchanges. The difference comes out from the fact, that the messages of all the IKEv2 exchanges except the initial exchange contain the Encrypted payload. In this case the compression is added as an additional step while constructing the Encrypted payload. The initial IKEv2 exchange requires introduction a new payload, that would contain other payloads in compressed form.

#### 3.1. Using Compression in the IKE\_SA\_INIT Exchange

The use of compression is not negotiated in a usual for IKEv2 manner - by exchanging appropriate Notification or Vendor ID payloads. Instead a different negotiation mechanism is used.

If an Initiator wants to use compression for the IKE SA being created, it constructs the IKE\_SA\_INIT request message in a following way. A new payload which is called Compressed payload and described in the [Section 4.1](#) is included into the request message. This payload contains other payloads in compressed form as well as an indication of what compression algorithm was used. When selecting compression algorithm the Initiator must guess what algorithms are supported by the peer and choose an appropriate one. If its guess is wrong the Responder will inform about this situation and the mutually appropriate algorithm will be negotiated by the cost of an extra round trip and a message recompression. The Critical bit in the Compressed payload header MUST be set to 1.

Initiator

-----

HDR, C!{SA, KE, [N+], [V+]}, Ni, [N+], [V+] -->

Not all payloads that are usually present in the IKE\_SA\_INIT messages are subject for compression. Some payloads may contain random or pseudo-random data that is almost uncompressable. Other payloads must be processed as early as possible, before the responder spends resources decompressing them. In particular, the Nonce payload and the COOKIE notification payload MUST NOT be included into the Compressed payload. Obviously, if the compression algorithm ID is from private range (241-255), then the corresponding Vendor ID payload MUST NOT be compressed. See [Section 5](#) for more details about interaction compression with other IKEv2 extensions.

If the Responder doesn't support IKEv2 compression, then it is expected to return the UNSUPPORTED\_CRITICAL\_PAYLOAD notification in response to such request message, as prescribed in the [Section 2.5](#) of





[RFC7296]. Depending on the implementation it may also return the INVALID\_SYNTAX notification or doesn't respond at all.

```

                                         Legacy Responder
                                         -----
<--  HDR, N(UNSUPPORTED_CRITICAL_PAYLOAD)

                                         or

<--  HDR, N(INVALID_SYNTAX)

                                         or

                                         (No response)
```

If the Initiator receives the UNSUPPORTED\_CRITICAL\_PAYLOAD notification with the Compressed payload type in its notification data or if it receives the INVALID\_SYNTAX notification or if it receives no response after several retransmissions then the Initiator MUST restart the IKE\_SA\_INIT exchange with no compression.

If the Responder supports IKEv2 compression, but doesn't support the particular compression algorithm the Initiator has chosen, then the Responder sends back a new error notification: INVALID\_COMPRESSION\_ALGORITHM. This notification is describe in the [Section 4.2](#). Its notification data contains the list of IDs of compression algorithms supported by the Responder.

```

                                         Responder
                                         -----
<--  HDR, N(INVALID_COMPRESSION_ALGORITHM)
```

If the Initiator receives the INVALID\_COMPRESSION\_ALGORITHM notification, then it looks through the list of algorithms included into the notification data and selects the appropriate one. After that it MUST restart the IKE\_SA\_INIT exchange using the newly selected algorithm for compression. If no mutually appropriate algorithm found, then the Initiator MUST restart the IKE\_SA\_INIT exchange with no compression.

Once the Responder receives the IKE\_SA\_INIT request with the compression algorithm in the Compressed payload that is appropriate to it, the compressed payloads are decompressed and along with the outer payloads form the uncompressed request message, which is then processed as usual. If the Responder agrees to use compression in the SA being created then the Responder MUST include the Compressed



payload in response message. The compression algorithm indicated in the Compressed payload MUST be the algorithm from the request message.

```

                                     Responder
                                     -----
<-- HDR, C!{SA, KE, [N+], [V+]}, Nr, [N+], [V+]

```

If for some reason the Responder doesn't want to use compression for the SA being created (e.g. using compression is disabled by administrator) then it MUST send back an uncompressed IKE\_SA\_INIT response message. In this case the endpoints MUST NOT use compression in subsequent exchanges.

### **3.2. Using Compression in Subsequent Exchanges**

Once the endpoints have used compression in the IKE\_SA\_INIT exchange, they may continue to use it in subsequent exchanges. However compression is used differently in this exchanges. Messages of every IKEv2 exchange apart from the initial exchange are protected by an Encrypted payload. With compression the algorithms for forming and processing of an Encrypted payload are modified as follows.

The content of an Encrypted payload is compressed before it is encrypted and authenticated. The Next Payload field in an Encrypted payload usually indicates the payload type of the first payload inside an Encrypted payload. If compression is used then the Next Payload field in the Encrypted payload MUST be set to XXX (TBA by IANA) - the value for the payload type of a Compressed payload. However, the Compressed payload itself MUST NOT appear inside the Encrypted payload, only its payload type is used to indicate that the content of the Encrypted payload was compressed before encryption.

Since in this case the Next Payload field in an Encrypted payload no longer indicates a type of the first inner payload, this information is moved to the Next Payload field of the last inner payload (which must be zero in IKEv2). This modification is done before the payloads are compressed.

```

Uncompressed: SK(Next=P1) {P1(Next=P2), P2(Next=P3), ... Pn(Next=0)}
Compressed:   SK(Next=C) {P1(Next=P2), P2(Next=P3), ... Pn(Next=P1)}

```

#### **Preparing payloads for compression**

This modification doesn't cause ambiguity on the receiver, since the total size of the inner payloads can be easily determined after decryption, and while walking through the list of them the receiver always knows whether the current payload is the last or not.

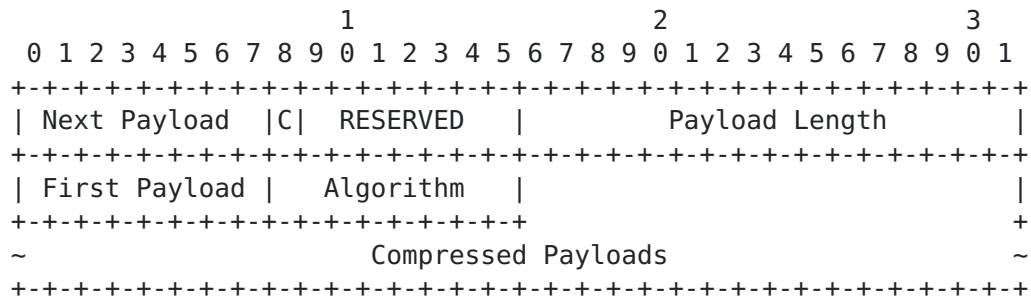


After the use of compression is negotiated in the initial exchange each endpoint is free to decide whether to apply compression or not on per-message basis. However, if applying compression to the content of an Encrypted payload doesn't reduce its size then the compression **MUST NOT** be used for this message. Implementations **MUST** be prepared to receive both compressed and uncompressed messages.

## 4. Payload Formats

### 4.1. Compressed Payload

The Compressed payload, denoted C!{...} in this document (the exclamation mark means that this payload is critical), contains other payloads in compressed form. The payload type for the Compressed payload is XXX (TBA by IANA).



#### Compressed Payload

- o Next Payload (1 octet) - Identifier for the payload type of the next payload in the message.
- o Critical (1 bit) - MUST be set to 1.
- o RESERVED (7 bits) - MUST be sent as zero; MUST be ignored on receipt (as specified in [RFC7296](#)).
- o Payload Length (2 octets, unsigned integer) - Length in octets of the current payload, including the generic payload header.
- o First Payload (1 octet) - Identifier for the payload type of the first payload contained in Compressed Payloads field.
- o Algorithm (1 octet) - ID of the algorithm used to compress inner payloads. The possible values for compression algorithm ID are listed in "IKEv2 Notification IPCOMP Transform IDs" registry in [\[IKEV2-IANA\]](#).
- o Compressed Payloads (variable length) - This field contains IKEv2 payloads in compressed form. The Next Payload field of the last included payload MUST be set to 0.

There MUST NOT be more than one Compressed payloads in a message. The Compressed payload MUST NOT appear inside the Encrypted payload and the Encrypted payload payload MUST NOT appear inside the Compressed payload.



#### 4.2. INVALID\_COMPRESSION\_ALGORITHM Notification

The INVALID\_COMPRESSION\_ALGORITHM notification is sent by Responder if the compression algorithm chosen by Initiator is inappropriate. The Notification Data contains the list of supported compression algorithm IDs.

```

      1             2             3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| Next Payload |C|  RESERVED   |          Payload Length          |
+-+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| Protocol ID(=0)| SPI Size(=0) |      Notify Message Type      |
+-+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
~                Supported Compression Algorithms                ~
+-+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

```

#### INVALID\_COMPRESSION\_ALGORITHM Notification

- o Protocol ID (1 octet) - MUST be 0.
- o SPI Size (1 octet) - MUST be 0, meaning no SPI is present.
- o Notify Message Type (2 octets) - MUST be XXX (TBA by IANA), the value assigned for the INVALID\_COMPRESSION\_ALGORITHM notification.
- o Supported Compression Algorithms (variable length) - List of compression algorithm IDs supported by the Responder. Each algorithm ID occupies one octet. The possible values for compression algorithm IDs are listed in "IKEv2 Notification IPCOMP Transform IDs" registry in [[IKEV2-IANA](#)].



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

IKE Compression is compatible with most of IKE extensions, since It neither affect their operation, nor is affected by them. However, some IKE extensions require special handling.

### **5.1. Interaction IKE Fragmentation**

When IKE Compression is used with IKE Fragmentation [[RFC7383](#)] the compression MUST take place before splitting the original content of the Encrypted payload into chunks. In other words, the content of the Encrypted payload must be compressed as a whole, before it is fragmented.

The Compressed payload MUST NOT appear inside the Encrypted Fragment payload and the Encrypted Fragment payload payload MUST NOT appear inside the Compressed payload.

### **5.2. Interaction IKE Resumption**

IKE Resumption [[RFC5723](#)] defines a mechanism for restoring an IKE SA state after a failure. If the peers support IKE Compression then the flag whether the compression was negotiated and the compression algorithm MUST be restored from the resumption ticket while resuming IKE SA. In other words the use of compression must not be re-negotiated in the IKE\_SESSION\_RESUME exchange and thus the Compressed payload MUST NOT appear in this exchange.

## 6. Security Considerations

It was shown in [[COMP-LEAK](#)] that using compression inside an encrypted channel may result in a leakage of some information about a plaintext. Recently some practical exploits were discovered that rely on using compression in security protocols ([[CRIME](#)], [[BREACH](#)]). However, it is believed that the way a compression is added to the IKEv2 would not weaken the protocol security. The existing exploits rely on an ability for an attacker to insert data into an encrypted stream, i.e. to perform a chosen-plaintext attack. IKEv2 messages don't contain application data, which restricts attacker's ability to perform chosen-plaintext attack. Moreover, the data usually exchanged over the IKEv2 SA contain no secret information and in most cases no sensitive information. The possible exceptions could be some weak Extensible Authentication Protocol (EAP) methods, which might transfer secret information within an IKE SA. It is RECOMMENDED that implementations don't use the IKE Compression for the messages containing the EAP payload if there is a possibility that the EAP method transfers secret information.

## 7. IANA Considerations

This document defines new Payload in the "IKEv2 Payload Types" registry:

<TBA>	Compressed	C
-------	------------	---

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

<TBA>	INVALID_COMPRESSION_ALGORITHM
-------	-------------------------------

## 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, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC5723] Sheffer, Y. and H. Tschofenig, "Internet Key Exchange Protocol Version 2 (IKEv2) Session Resumption", [RFC 5723](#), DOI 10.17487/RFC5723, January 2010, <<http://www.rfc-editor.org/info/rfc5723>>.
- [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, <<http://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, <<http://www.rfc-editor.org/info/rfc7383>>.
- [IKEV2-IANA] "Internet Key Exchange Version 2 (IKEv2) Parameters", <<http://www.iana.org/assignments/ikev2-parameters>>.

### 8.2. Informative References

- [IPSEC-IOT-REQS] Migault, D. and T. Guggemos, "Requirements for Diet-ESP the IPsec/ESP protocol for IoT", [draft-mglt-6lo-diet-esp-requirements-01](#) (work in progress), February 2015.
- [COMP-LEAK] Kelsey, J., "Compression and Information Leakage of Plaintext", <<http://www.iacr.org/cryptodb/archive/2002/FSE/3091/3091.pdf>>.
- [CRIME] Rizzo, J. and T. Duong, "The CRIME attack", <[https://docs.google.com/presentation/d/11eBmGiHbYcHR9gL5nDyZChu\\_lCa2Gizeu0faLU2HOU](https://docs.google.com/presentation/d/11eBmGiHbYcHR9gL5nDyZChu_lCa2Gizeu0faLU2HOU)>.
- [BREACH] Prado, A., Harris, N., and Y. Gluck, "SSL, gone in 30 seconds: A BREACH beyond CRIME", <<https://media.blackhat.com/us-13/>>.



US-13-Prado-SSL-Gone-in-30-seconds-A-BREACH-beyond-CRIME-  
Slides.pdf>.

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)