

Internet Engineering Task Force (IETF)
Request for Comments: 5857
Category: Standards Track
ISSN: 2070-1721

E. Ertekin
C. Christou
R. Jasani
Booz Allen Hamilton
T. Kivinen
AuthenTec, Inc.
C. Bormann
Universitaet Bremen TZI
May 2010

IKEv2 Extensions to Support Robust Header Compression over IPsec

Abstract

In order to integrate Robust Header Compression (ROHC) with IPsec, a mechanism is needed to signal ROHC channel parameters between endpoints. Internet Key Exchange (IKE) is a mechanism that can be leveraged to exchange these parameters. This document specifies extensions to IKEv2 that will allow ROHC and its associated channel parameters to be signaled for IPsec Security Associations (SAs).

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <http://www.rfc-editor.org/info/rfc5857>.

Copyright Notice

Copyright (c) 2010 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.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction	3
2. Terminology	3
3. ROHC Channel Initialization for ROHCoIPsec	3
3.1. ROHC_SUPPORTED Notify Message	3
3.1.1. ROHC Attributes	5
3.1.2. ROHC Attribute Types	6
3.2. ROHC Channel Parameters That Are Implicitly Set	9
4. Security Considerations	9
5. IANA Considerations	9
6. Acknowledgments	10
7. References	11
7.1. Normative References	11
7.2. Informative References	12

1. Introduction

Increased packet header overhead due to IPsec [IPSEC] can result in the inefficient utilization of bandwidth. Coupling ROHC [ROHC] with IPsec offers an efficient way to transfer protected IP traffic.

ROHCoIPsec [ROHCOIPSEC] requires configuration parameters to be initialized at the compressor and decompressor. Current specifications for hop-by-hop ROHC negotiate these parameters through a link-layer protocol such as the Point-to-Point Protocol (PPP) (i.e., ROHC over PPP [ROHC-PPP]). Since key exchange protocols (e.g., IKEv2 [IKEV2]) can be used to dynamically establish parameters between IPsec peers, this document defines extensions to IKEv2 to signal ROHC parameters for ROHCoIPsec.

2. Terminology

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 RFC 2119 [BRA97].

3. ROHC Channel Initialization for ROHCoIPsec

The following subsections define extensions to IKEv2 that enable an initiator and a responder to signal parameters required to establish a ROHC channel for a ROHCoIPsec session.

3.1. ROHC_SUPPORTED Notify Message

ROHC channel parameters MUST be signaled separately for each ROHC-enabled IPsec SA. Specifically, a new Notify message type MUST be included in the IKE_AUTH and CREATE_CHILD_SA exchanges whenever a new ROHC-enabled IPsec SA is created, or an existing one is rekeyed.

The Notify payload sent by the initiator MUST contain the channel parameters for the ROHC session. These parameters indicate the capabilities of the ROHC decompressor at the initiator. Upon receipt of the initiator's request, the responder will either ignore the payload (if it doesn't support ROHC or the proposed parameters) or respond with a Notify payload that contains its own ROHC channel parameters.

Note that only one Notify payload is used to convey ROHC parameters. If multiple Notify payloads containing ROHC parameters are received, all but the first such Notify payload MUST be dropped. If the initiator does not receive a Notify payload with the responder's ROHC channel parameters, ROHC MUST NOT be enabled on the Child SA.

A new Notify Message Type value, denoted ROHC_SUPPORTED, indicates that the Notify payload is conveying ROHC channel parameters (Section 4).

The Notify payload (defined in RFC 4306 [IKEV2]) is illustrated in Figure 1.

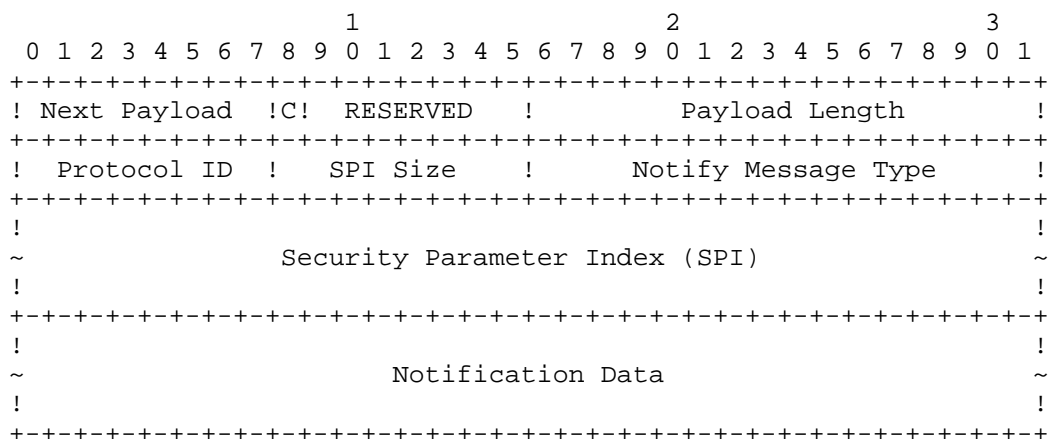


Figure 1. Notify Payload Format

The fields of the Notify payload are set as follows:

Next Payload (1 octet)

Identifier for the payload type of the next payload in the message. Further details can be found in RFC 4306 [IKEV2].

Critical (1 bit)

Since all IKEv2 implementations support the Notify payload, this value MUST be set to zero.

Payload Length (2 octets)

As defined in RFC 4306 [IKEV2], this field indicates the length of the current payload, including the generic payload header.

Protocol ID (1 octet)

Since this notification message is used during the creation of a Child SA, this field MUST be set to zero.

SPI Size (1 octet)

This value MUST be set to zero, since no SPI is applicable (ROHC parameters are set at SA creation; thus, the SPI has not been defined).

Notify Message Type (2 octets)

This field MUST be set to ROHC_SUPPORTED.

Security Parameter Index (SPI)

Since the SPI Size field is 0, this field MUST NOT be transmitted.

Notification Data (variable)

This field MUST contain at least three ROHC Attributes (Section 3.1.1).

3.1.1. ROHC Attributes

The ROHC_SUPPORTED Notify message is used to signal channel parameters between ROHCoIPsec compressor and decompressor. The message contains a list of "ROHC Attributes", which contain the parameters required for the ROHCoIPsec session.

The format for signaling ROHC Attributes takes a similar format to the Transform Attributes described in Section 3.3.5 of RFC 4306 [IKEV2]. The format of the ROHC Attribute is shown in Figure 2.

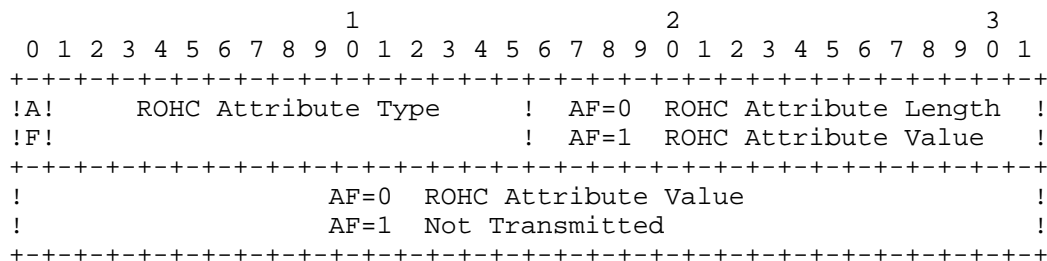


Figure 2. Format of the ROHC Attribute

- o Attribute Format (AF) (1 bit) - If the AF bit is a zero (0), then the ROHC Attribute is expressed in a Type/Length/Value format. If the AF bit is a one (1), then the ROHC attribute is expressed in a Type/Value (TV) format.

- o ROHC Attribute Type (15 bits) - Unique identifier for each type of ROHC attribute (Section 3.1.2).
- o ROHC Attribute Length (2 octets) - Length (in octets) of the Attribute Value. When the AF bit is a one (1), the ROHC Attribute Value is 2 octets and the ROHC Attribute Length field is not present.
- o ROHC Attribute Value (variable length) - Value of the ROHC Attribute associated with the ROHC Attribute Type. If the AF bit is a zero (0), this field's length is defined by the ROHC Attribute Length field. If the AF bit is a one (1), the length of the ROHC Attribute Value is 2 octets.

3.1.2. ROHC Attribute Types

This section describes five ROHC Attribute Types: MAX_CID, ROHC_PROFILE, ROHC_INTEG, ROHC_ICV_LEN, and MRRU. The value allocated for each ROHC Attribute Type is specified in Section 4.

MAX_CID (Maximum Context Identifier, AF = 1)

The MAX_CID attribute is a mandatory attribute. Exactly one MAX_CID attribute MUST be sent. The MAX_CID field indicates the maximum value of a context identifier supported by the ROHCoIPsec decompressor. This attribute value is 2 octets in length. The value for MAX_CID MUST be at least 0 and at most 16383. Since CIDs can take values between 0 and MAX_CID, the actual number of contexts that can be used are MAX_CID+1. If MAX_CID is 0, this implies having one context. The recipient of the MAX_CID Attribute MUST only use context identifiers up to MAX_CID for compression.

Note that the MAX_CID parameter is a one-way notification (i.e., the sender of the attribute indicates what it can handle to the other end); therefore, different values for MAX_CID may be announced in each direction.

ROHC_PROFILE (ROHC Profile, AF = 1)

The ROHC_PROFILE attribute is a mandatory attribute. Each ROHC_PROFILE attribute has a fixed length of 4 octets, and its attribute value is a 2-octet long ROHC Profile Identifier [ROHC_PROF]. There MUST be at least one ROHC_PROFILE attribute included in the ROHC_SUPPORTED Notify message. If multiple ROHC_PROFILE attributes are sent, the order is arbitrary. The recipient of a ROHC_PROFILE attribute(s) MUST only use the profile(s) proposed for compression.

Several common profiles are defined in RFCs 3095 [ROHCV1] and 5225 [ROHCV2]. Note, however, that two versions of the same profile MUST NOT be signaled. For example, if a ROHCoIPsec decompressor supports both ROHCv1 UDP (0x0002) and ROHCv2 UDP (0x0102), both profiles MUST NOT be signaled. This restriction is needed, as packets compressed by ROHC express only the 8 least-significant bits of the profile identifier; since the 8 least-significant bits for corresponding profiles in ROHCv1 and ROHCv2 are identical, the decompressor is not capable of determining the ROHC version that was used to compress the packet.

Note that the ROHC_PROFILE attribute is a one-way notification; therefore, different values for ROHC_PROFILE may be announced in each direction.

ROHC_INTEG (Integrity Algorithm for Verification of Decompressed Headers, AF = 1)

The ROHC_INTEG attribute is a mandatory attribute. There MUST be at least one ROHC_INTEG attribute contained within the ROHC_SUPPORTED Notify message. The attribute value contains the identifier of an integrity algorithm that is used to ensure the integrity of the decompressed packets (i.e., ensure that the decompressed packet headers are identical to the original packet headers prior to compression).

Authentication algorithms that MUST be supported are specified in the "Authentication Algorithms" table in Section 3.1.1 ("ESP Encryption and Authentication Algorithms") of RFC 4835 [CRYPTO-ALG] (or its successor).

The integrity algorithm is represented by a 2-octet value that corresponds to the value listed in the IKEv2 Parameters registry [IKEV2-PARA], "Transform Type 3 - Integrity Algorithm Transform IDs" section. Upon receipt of the ROHC_INTEG attribute(s), the responder MUST select exactly one of the proposed algorithms; the chosen value is sent back in the ROHC_SUPPORTED Notify message returned by the responder to the initiator. The selected integrity algorithm MUST be used in both directions. If the responder does not accept any of the algorithms proposed by the initiator, ROHC MUST NOT be enabled on the SA.

It is noted that:

1. The keys (one for each direction) for this integrity algorithm are derived from the IKEv2 KEYMAT (see [IKEV2], Section 2.17). For the purposes of this key derivation, ROHC is considered to

be an IPsec protocol. When a ROHC-enabled CHILD_SA is rekeyed, the key associated with this integrity algorithm is rekeyed as well.

2. A ROHCv1 initiator MAY signal a value of zero (0x0000) in a ROHC_INTEG attribute. This corresponds to "NONE" in the "IKEv2 Integrity Algorithm Transform IDs" registry. The ROHCv1 responder MAY select this value by responding to the initiator with a ROHC_INTEG attribute of zero (0x0000). In this scenario, no integrity algorithm is applied in either direction.
3. The ROHC_INTEG attribute is a parameter that is negotiated between two ends. In other words, the initiator indicates what it supports, the responder selects one of the ROHC_INTEG values proposed and sends the selected value to the initiator.

ROHC_ICV_LEN (Integrity Algorithm Length, AF = 1)

The ROHC_ICV_LEN attribute is an optional attribute. There MAY be zero or one ROHC_ICV_LEN attribute contained within the ROHC_SUPPORTED Notify message. The attribute specifies the number of Integrity Check Value (ICV) octets the sender expects to receive on incoming ROHC packets. The ICV of the negotiated ROHC_INTEG algorithms MUST be truncated to ROHC_ICV_LEN bytes by taking the first ROHC_ICV_LEN bytes of the output. Both the initiator and responder announce a single value for their own ICV length. The recipient of the ROHC_ICV_LEN attribute MUST truncate the ICV to the length contained in the message. If the value of the ROHC_ICV_LEN attribute is zero, then an ICV MUST NOT be sent. If no ROHC_ICV_LEN attribute is sent at all or if the ROHC_ICV_LEN is larger than the length of the ICV of selected algorithm, then the full ICV length as specified by the ROHC_INTEG algorithm MUST be sent.

Note that the ROHC_ICV_LEN attribute is a one-way notification; therefore, different values for ROHC_ICV_LEN may be announced in each direction.

MRRU (Maximum Reconstructed Reception Unit, AF = 1)

The MRRU attribute is an optional attribute. There MAY be zero or one MRRU attribute contained within the ROHC_SUPPORTED Notify message. The attribute value is 2 octets in length. The attribute specifies the size of the largest reconstructed unit in octets that the ROHCv1 decompressor is expected to reassemble from ROHC segments (see Section 5.2.5 of [ROHCv1]). This size includes the Cyclic Redundancy Check (CRC) and the ROHC ICV. If

MRRU is 0 or if no MRRU attribute is sent, segment headers MUST NOT be transmitted on the ROHCoIPsec channel.

Note that the MRRU attribute is a one-way notification; therefore, different values for MRRU may be announced in each direction.

If an unknown ROHC Attribute Type Value is received, it MUST be silently ignored.

3.2. ROHC Channel Parameters That Are Implicitly Set

The following ROHC channel parameters MUST NOT be signaled:

- o `LARGE_CIDS`: This value is implicitly determined by the value of `MAX_CID` (i.e., if `MAX_CID` is ≤ 15 , `LARGE_CIDS` is assumed to be 0).
- o `FEEDBACK_FOR`: When a pair of SAs is created (one in each direction), the ROHC channel parameter `FEEDBACK_FOR` MUST be set implicitly to the other SA of the pair (i.e., the SA pointing in the reverse direction).

4. Security Considerations

The ability to negotiate the length of the ROHC ICV may introduce vulnerabilities to the ROHCoIPsec protocol. Specifically, the capability to signal a short ICV length may result in scenarios where erroneous packets are forwarded into the protected domain. This security consideration is documented in further detail in Section 6.1.4 of [ROHCOIPSEC] and Section 5 of [IPSEC-ROHC].

This security consideration can be mitigated by using longer ICVs, but this comes at the cost of additional overhead, which reduces the overall benefits offered by ROHCoIPsec.

5. IANA Considerations

This document defines a new Notify message (Status Type). Therefore, IANA has allocated one value from the "IKEv2 Notify Message Types" registry to indicate `ROHC_SUPPORTED`.

In addition, IANA has created a new "ROHC Attribute Types" registry in the "Internet Key Exchange Version 2 (IKEv2) Parameters" registry [IKEV2-PARA]. Within the "ROHC Attribute Types" registry, this document allocates the following values:

Registry:

Value	ROHC Attribute Type	Format	Reference
0	RESERVED		[RFC5857]
1	Maximum Context Identifier (MAX_CID)	TV	[RFC5857]
2	ROHC Profile (ROHC_PROFILE)	TV	[RFC5857]
3	ROHC Integrity Algorithm (ROHC_INTEG)	TV	[RFC5857]
4	ROHC ICV Length in bytes (ROHC_ICV_LEN)	TV	[RFC5857]
5	Maximum Reconstructed Reception Unit (MRRU)	TV	[RFC5857]
6-16383	Unassigned		
16384-32767	Private use		[RFC5857]

Following the policies outlined in [IANA-CONSIDERATIONS], the IANA policy for assigning new values for the ROHC Attribute Types registry shall be Expert Review.

For registration requests, the responsible IESG Area Director will appoint the Designated Expert. The Designated Expert will post a request to both the ROHC and IPsec mailing lists (or a successor designated by the Area Director) for comment and review. The Designated Expert will either approve or deny the registration request and publish a notice of the decision to both mailing lists (or their successors), as well as informing IANA. A denial notice must be justified by an explanation.

6. Acknowledgments

The authors would like to thank Sean O'Keefe, James Kohler, and Linda Noone of the Department of Defense, as well as Rich Espy of OPnet for their contributions and support in the development of this document.

The authors would also like to thank Yoav Nir and Robert A Stangarone Jr.: both served as committed document reviewers for this specification.

In addition, the authors would like to thank the following for their numerous reviews and comments to this document:

- o Magnus Westerlund
- o Stephen Kent
- o Lars-Erik Jonsson
- o Pasi Eronen
- o Jonah Pezeshki

- o Carl Knutsson
- o Joseph Touch
- o David Black
- o Glen Zorn

Finally, the authors would also like to thank Tom Conkle, Michele Casey, and Etzel Brower.

7. References

7.1. Normative References

- [IPSEC] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.
- [ROHC] Sandlund, K., Pelletier, G., and L-E. Jonsson, "The RObust Header Compression (ROHC) Framework", RFC 5795, March 2010.
- [IKEV2] Kaufman, C., "Internet Key Exchange (IKEv2) Protocol", RFC 4306, December 2005.
- [BRA97] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [ROHCV1] Bormann, C., Burmeister, C., Degermark, M., Fukushima, H., Hannu, H., Jonsson, L-E., Hakenberg, R., Koren, T., Le, K., Liu, Z., Martensson, A., Miyazaki, A., Svanbro, K., Wiebke, T., Yoshimura, T., and H. Zheng, "RObust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed", RFC 3095, July 2001.
- [ROHCV2] Pelletier, G. and K. Sandlund, "RObust Header Compression Version 2 (ROHCv2): Profiles for RTP, UDP, IP, ESP and UDP-Lite", RFC 5225, April 2008.
- [IPSEC-ROHC] Ertekin, E., Christou, C., and C. Bormann, "IPsec Extensions to Support Robust Header Compression over IPsec", RFC 5858, May 2010.
- [IANA-CONSIDERATIONS] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

7.2. Informative References

- [ROHCOIPSEC] Ertekin, E., Jasani, R., Christou, C., and C. Bormann, "Integration of Header Compression over IPsec Security Associations", RFC 5856, May 2010.
- [ROHC-PPP] Bormann, C., "Robust Header Compression (ROHC) over PPP", RFC 3241, April 2002.
- [ROHCPROF] IANA, "RObust Header Compression (ROHC) Profile Identifiers", <<http://www.iana.org>>.
- [CRYPTO-ALG] Manral, V., "Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH)", RFC 4835, April 2007.
- [IKEV2-PARA] IANA, "Internet Key Exchange Version 2 (KEv2) Parameters", <<http://www.iana.org>>.

Authors' Addresses

Emre Ertekin
Booz Allen Hamilton
5220 Pacific Concourse Drive, Suite 200
Los Angeles, CA 90045
US

EMail: ertekin_emre@bah.com

Chris Christou
Booz Allen Hamilton
13200 Woodland Park Dr.
Herndon, VA 20171
US

EMail: christou_chris@bah.com

Rohan Jasani
Booz Allen Hamilton
13200 Woodland Park Dr.
Herndon, VA 20171
US

EMail: ro@breakcheck.com

Tero Kivinen
AuthenTec, Inc.
Fredrikinkatu 47
HELSINKI
FI

EMail: kivinen@iki.fi

Carsten Bormann
Universitaet Bremen TZI
Postfach 330440
Bremen D-28334
Germany

EMail: cabo@tzi.org

