Network Working Group	E. Ertekin	
Internet-Draft	C. Christou	
Expires: February 13, 2010	R. Jasani	
	Booz Allen Hamilton	
	T. Kivinen	
	Safenet, Inc.	
	C. Bormann	
	Universitaet Bremen TZI	
	August 12, 2009	

TOC

IKEv2 Extensions to Support Robust Header Compression over IPsec (ROHCoIPsec)
draft-ietf-rohc-ikev2-extensions-hcoipsec-09

Status of this Memo

Abstract

In order to integrate ROHC with IPsec [ROHCOIPSEC], a mechanism is needed to signal ROHC channel parameters between end-points. Internet Key Exchange (IKE) is a mechanism which can be leveraged to exchange these parameters. This document specifies extensions to IKEv2 [IKEV2] that will allow ROHC and its associated channel parameters to be signaled for IPsec security associations (SAs).

Table of Contents

---

1. Introduction

TOC

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 Point-to-Point Protocol (PPP) (i.e. ROHC over PPP

[ROHC-PPP]). Since key exchange protocols (e.g. IKEv2) can be used to dynamically establish parameters between IPsec peers, this document defines extensions to IKEv2 to signal ROHC parameters for ROHCoIPsec.

---

2. ROHC Channel Initialization for ROHCoIPsec

TOC

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

2.1. ROHC Channel Parameters that are Signaled

TOC

ROHC channel parameters will be signaled at either the establishment or rekeying of a Child SA. Specifically, a new Notify message type is used during the IKE_AUTH and CREATE_CHILD_SA exchanges to convey these parameters.

The Notify payload sent by the initiator contains the channel parameters for the ROHC implementation. Specifically, 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. The value for the ROHC_SUPPORTED message is specified in Section 4.

The Notify Payload (defined in [IKEV2]) is illustrated in Figure 1.

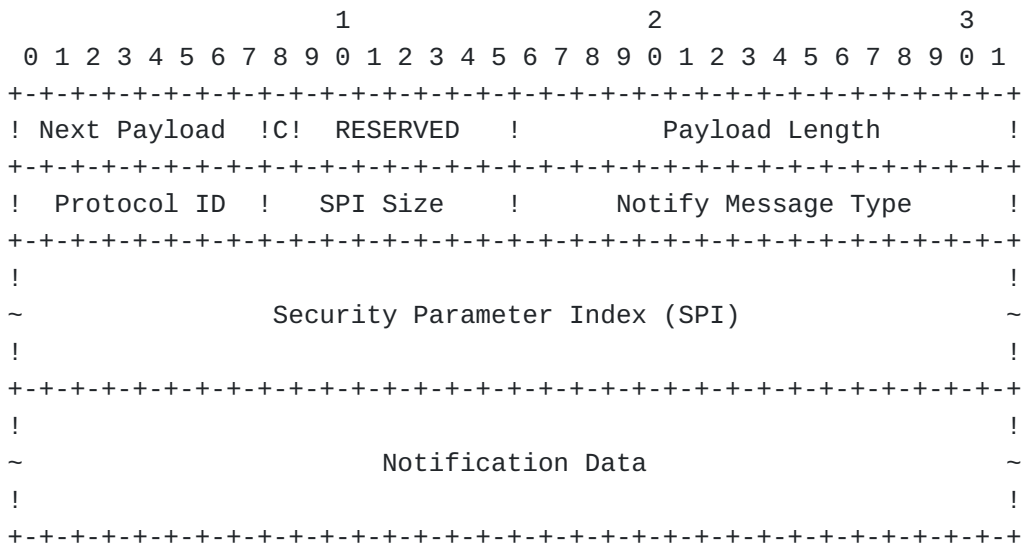

Figure 1. Notify Payload format.

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 [IKEV2].

Critical (1 bit)

> Since all IKEv2 implementations must support the Notify Payload, this value is zero.

Payload Length (2 octets)

> As defined in [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.

2.1.1. ROHC_SUPPORTED Notify Message

TOC

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 [IKEV2]. The ROHC Attribute is shown in Figure 2.

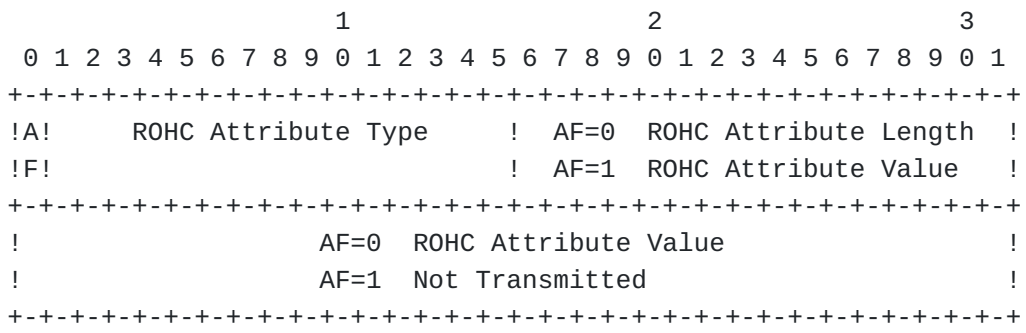

Figure 2. Format of the ROHC Attribute.

*ROHC Attribute Type (2 octets) - Unique identifier for each type of ROHC attribute (see Section 2.1.2). The most significant bit in the field is the Attribute Format (AF) bit. If the AF bit is a zero (0), then the ROHC Attribute is expressed in a Type/Length/Value (TLV) format. If the AF bit is a one (1), then the ROHC Attribute is expressed in a Type/Value (TV) format.

*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.

*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.

TOC

2.1.2. 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.
Maximum Context Identifier (MAX_CID, 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 two octets in length. The range of values for MAX_CID must be at least 0 and at most 16383 (the value 0 implies having one context). The recipient of the MAX_CID Attribute must only use up to MAX_CID context identifiers for compression.

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 two-octet long profile identifier. There may be one or more ROHC_PROFILE attribute(s) 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 [ROHCV1] and [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.

Integrity Algorithm for Verification of Decompressed Headers (ROHC_INTEG, 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 contains an integrity algorithm that is used to ensure the integrity of the decompressed packets (i.e. ensure that the packet headers are properly decompressed).

> Authentication algorithms that must be supported are specified in Section 3.2 of [CRYPTO-ALG]. More explicitly, the implementation conformance requirements for authentication algorithms are as follows:

Requirement	Algorithm
Must	AUTH_HMAC_SHA1_96
Should+	AUTH_AES_XCBC_MAC_96
May	AUTH_HMAC_MD5_96

The integrity algorithm is represented by a two octet value that corresponds to the value listed in [IKEV2-PARA] "For Transform Type 3 (Integrity Algorithm)" section. Upon receipt of the ROHC_INTEG attribute(s), the responder must select exactly one of proposed algorithms and send the selected algorithm back to the initiator. The selected integrity algorithm must be used in both directions.

It is noted that:

1. The key for this Integrity Algorithm is computed using the same method as is used to compute IPsec's Integrity Algorithm key ([IKEV2], Section 2.17). When a ROHC-enabled CHILD_SA is rekeyed, the key associated with this integrity algorithm is rekeyed as well.

2. A ROHCoIPsec initiator may signal a value of zero (0x0000) in a ROHC_INTEG attribute. This corresponds to "NONE" in the Integrity Algorithm Transform ID registry. The ROHCoIPsec 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.

Integrity Algorithm Length (ROHC_ICV_LEN, 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 ICV octets the sender expects to receive on incoming ROHC packets. The ICV of the negotiated ROHC_INTEG algorithms are truncated to ROHC_ICV_LEN bytes by taking the first ROHC_ICV_LEN bytes of the output. Both the initiator and responder announce their preference 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 ROHC_ICV_LEN length is zero, then no ICV is calculated or sent. If no ROHC_ICV_LEN attribute is sent at all or 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 is sent.

Maximum reconstructed reception unit (MRRU, AF = 1)

> The MRRU attribute is an optional attribute. There may be zero or one MRRU attribute contained within the ROHC_SUPPORTED Notify message. If present, the attribute value is two octets in length. The attribute specifies the size of the largest reconstructed unit in octets that the ROHCoIPsec decompressor is expected to reassemble from ROHC segments. This size includes the CRC, and the ROHC ICV. If MRRU is 0 or if no MRRU attribute is sent, no segment headers are allowed on the ROHCoIPsec channel.

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

2.2. ROHC Channel Parameters that are Implicitly Set

TOC

The following ROHC channel parameters are not signaled:

- LARGE_CIDS: This value is implicitly determined by the value of MAX_CID (e.g. if MAX_CID is <= 15, LARGE_CIDS is assumed to be 0).

- FEEDBACK_FOR: When a pair of SAs are created (one in each direction), the ROHC channel parameter FEEDBACK_FOR is set implicitly to the other SA of the pair (i.e. the SA pointing in the reverse direction).

---

3. Security Considerations

TOC

The ROHC channel parameters signaled via IKEv2 do not add any new vulnerabilities beyond those associated with the normal operation of IKEv2.

---

4. IANA Considerations

TOC

This document defines a new Notify Message (Status Type). Therefore, IANA is requested to allocate one value from the IKEv2 Notify Message registry to indicate ROHC_SUPPORTED. Note that, since this Notify Message is a Status Type, values ranging from 0 to 16383 must not be allocated for ROHC_SUPPORTED.
In addition, IANA is requested to allocate a "ROHC Attribute Types" registry in the IKEv2 Parameters Registry [IKEV2-PARA]. Within the

"ROHC Attribute Types" registry, this document allocates the following values:

Registry Value	ROHC Attribute Type	Reference
0	RESERVED	[rfcThis]
1	Maximum Context Identifier (MAX_CID)	[rfcThis]
2	ROHC Profile (ROHC_PROFILE)	[rfcThis]
3	ROHC Integrity Algorithm (ROHC_INTEG)	[rfcThis]
4	ROHC ICV Length in bytes (ROHC_ICV_LEN)	[rfcThis]
5	Maximum Reconstructed Reception Unit (MRRU)	[rfcThis]
6-65536	Unassigned	

Following the policies outlined in [IANA-CONSIDERATIONS], the IANA policy for assigning new values for the ROHC Attribute Types registry shall be Specification Required: values and their meanings must be documented in a permanent and readily available public specification, in sufficient detail so that interoperability between independent implementations is possible.

---

5. Acknowledgments

TOC

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

*Mr. Magnus Westerlund

*Dr. Stephen Kent

*Mr. Lars-Erik Jonsson

*Mr. Pasi Eronen

*Dr. Jonah Pezeshki

*Mr. Carl Knutsson

*Dr. Joseph Touch

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

6. References

TOC

6.1. Normative References

TOC

[IPSEC]	Kent, S. and K. Seo, “Security Architecture for the Internet Protocol,” RFC 4301, December 2005.
[ROHC]	Jonsson, L-E., Pelletier, G., and K. Sandlund, “The RObust Header Compression (ROHC) Framework,” RFC 4995, July 2007.
[IKEV2]	Kaufman, C., “Internet Key Exchange (IKEv2) Protocol,” RFC 4306, December 2005.
[ROHCV1]	Bormann, C., Burmeister, C., Degermark, M., Fukushima, H., Hannu, H., Jonsson, L., 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.

6.2. Informative References

TOC

[ROHCOIPSEC]	Ertekin, E., Jasani, R., Christou, C., and C. Bormann, “Integration of Header Compression over IPsec Security Associations,” work in progress , August 2009.
[ROHC-PPP]	Bormann, C., “Robust Header Compression (ROHC) over PPP,” RFC 3241, April 2002.
[CRYPTO-ALG]	Manral, V., “Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH),” RFC 4835, April 2007.
[IKEV2-PARA]	IANA, “IKEv2 Parameters, http://www.iana.org/assignments/ikev2-parameters,” January 2008.
[IANA-CONSIDERATIONS]	Narten, T. and H. Alvestrand, “Guidelines for Writing an IANA Considerations Section in RFCs,” RFC 5226, October 1998.

Authors' Addresses

TOC

	Emre Ertekin
	Booz Allen Hamilton
	13200 Woodland Park Dr.
	Herndon, VA 20171
	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
	Safenet, Inc.
	Fredrikinkatu 47
	HELSINKI
	FI
Email:	kivinen@safenet-inc.com
	Carsten Bormann
	Universitaet Bremen TZI
	Postfach 330440
	Bremen D-28334
	Germany
Email:	cabo@tzi.org