Internet Engineering Task Force (IETF)
Request for Comments: 5856
Category: Informational
ISSN: 2070-1721

E. Ertekin
R. Jasani
C. Christou
Booz Allen Hamilton
C. Bormann
Universitaet Bremen TZI

May 2010

Integration of Robust Header Compression over IPsec Security Associations

Abstract

IP Security (IPsec) provides various security services for IP traffic. However, the benefits of IPsec come at the cost of increased overhead. This document outlines a framework for integrating Robust Header Compression (ROHC) over IPsec (ROHCoIPsec). By compressing the inner headers of IP packets, ROHCoIPsec proposes to reduce the amount of overhead associated with the transmission of traffic over IPsec Security Associations (SAs).

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are a candidate for any level of Internet Standard; see 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/rfc5856.

Copyright Notice

Table of Contents

1. Introduction

This document outlines a framework for integrating ROHC [ROHC] over IPsec [IPSEC] (ROHCoIPsec). The goal of ROHCoIPsec is to reduce the protocol overhead associated with packets traversing between IPsec SA endpoints. This can be achieved by compressing the transport layer header (e.g., UDP, TCP, etc.) and inner IP header of packets at the ingress of the IPsec tunnel, and decompressing these headers at the egress.

For ROHCoIPsec, this document assumes that ROHC will be used to compress the inner headers of IP packets traversing an IPsec tunnel. However, since current specifications for ROHC detail its operation on a hop-by-hop basis, it requires extensions to enable its operation over IPsec SAs. This document outlines a framework for extending the usage of ROHC to operate at IPsec SA endpoints.

ROHCoIPsec targets the application of ROHC to tunnel mode SAs. Transport mode SAs only protect the payload of an IP packet, leaving the IP header untouched. Intermediate routers subsequently use this IP header to route the packet to a decryption device. Therefore, if ROHC is to operate over IPsec transport-mode SAs, (de)compression functionality can only be applied to the transport layer headers, and not to the IP header. Because current ROHC specifications do not include support for the compression of transport layer headers alone, the ROHCoIPsec framework outlined by this document describes the application of ROHC to tunnel mode SAs.

2. Audience

The authors target members of both the ROHC and IPsec communities who may consider extending the ROHC and IPsec protocols to meet the requirements put forth in this document. In addition, this document is directed towards vendors developing IPsec devices that will be deployed in bandwidth-constrained IP networks.

3. Terminology

ROHC Process

Generic reference to a ROHC instance (as defined in RFC 3759 [ROHC-TERM]) or any supporting ROHC components.

Compressed Traffic

Traffic that is processed through the ROHC compressor and decompressor instances. Packet headers are compressed and decompressed using a specific header compression profile.

Uncompressed Traffic

Traffic that is not processed by the ROHC compressor instance. Instead, this type of traffic bypasses the ROHC process.

IPsec Process

Generic reference to the Internet Protocol Security (IPsec) process.

Next Header

Refers to the Protocol (IPv4) or Next Header (IPv6, Extension) field.

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].

4. Problem Statement: IPsec Packet Overhead

IPsec mechanisms provide various security services for IP networks. However, the benefits of IPsec come at the cost of increased per-packet overhead. For example, traffic flow confidentiality (generally leveraged at security gateways) requires the tunneling of IP packets between IPsec implementations. Although these IPsec tunnels will effectively mask the source-destination patterns that an intruder can ascertain, tunneling comes at the cost of increased packet overhead. Specifically, an Encapsulating Security Payload (ESP) tunnel mode SA applied to an IPv6 flow results in at least 50 bytes of additional overhead per packet. This additional overhead may be undesirable for many bandwidth-constrained wireless and/or satellite communications networks, as these types of infrastructure are not overprovisioned. ROHC applied on a per-hop basis over bandwidth-constrained links will also suffer from reduced performance when encryption is used on the tunneled header, since encrypted headers cannot be compressed. Consequently, the additional overhead incurred by an IPsec tunnel may result in the inefficient utilization of bandwidth.

Packet overhead is particularly significant for traffic profiles characterized by small packet payloads (e.g., various voice codecs). If these small packets are afforded the security services of an IPsec tunnel mode SA, the amount of per-packet overhead is increased. Thus, a mechanism is needed to reduce the overhead associated with such flows.

5. Overview of the ROHCoIPsec Framework

5.1. ROHCoIPsec Assumptions

The goal of ROHCoIPsec is to provide efficient transport of IP packets between IPsec devices without compromising the security services offered by IPsec. The ROHCoIPsec framework has been developed based on the following assumptions:

- ROHC will be leveraged to reduce the amount of overhead associated with unicast IP packets traversing an IPsec SA.

- ROHC will be instantiated at the IPsec SA endpoints, and it will be applied on a per-SA basis.

- Once the decompression operation completes, decompressed packet headers will be identical to the original packet headers before compression.

5.2. Summary of the ROHCoIPsec Framework

ROHC reduces packet overhead in a network by exploiting intra- and inter-packet redundancies of network and transport-layer header fields of a flow.

Current ROHC protocol specifications compress packet headers on a hop-by-hop basis. However, IPsec SAs are instantiated between two IPsec endpoints. Therefore, various extensions to both ROHC and IPsec need to be defined to ensure the successful operation of the ROHC protocol at IPsec SA endpoints.

The specification of ROHC over IPsec SAs is straightforward, since SA endpoints provide source/destination pairs where (de)compression operations can take place. Compression of the inner IP and upper layer protocol headers in such a manner offers a reduction of packet overhead between the two SA endpoints. Since ROHC will now operate between IPsec endpoints (over multiple intermediate nodes that are transparent to an IPsec SA), it is imperative to ensure that its performance will not be severely impacted due to increased packet reordering and/or packet loss between the compressor and decompressor.

In addition, ROHC can no longer rely on the underlying link layer for ROHC channel parameter configuration and packet identification. The ROHCoIPsec framework proposes that ROHC channel parameter configuration is accomplished by an SA management protocol (e.g., Internet Key Exchange Protocol version 2 (IKEv2) [IKEV2]), while identification of compressed header packets is achieved through the

Next Header field of the security protocol (e.g., Authentication Header (AH) [AH], ESP [ESP]) header.

Using the ROHCoIPsec framework proposed below, outbound and inbound IP traffic processing at an IPsec device needs to be modified. For an outbound packet, a ROHCoIPsec implementation will compress appropriate packet headers, and subsequently encrypt and/or integrity protect the packet. For tunnel mode SAs, compression may be applied to the transport layer and the inner IP headers. For inbound packets, an IPsec device must first decrypt and/or integrity check the packet. Then, decompression of the inner packet headers is performed. After decompression, the packet is checked against the access controls imposed on all inbound traffic associated with the SA (as specified in RFC 4301 [IPSEC]).

> Note: Compression of inner headers is independent from compression of the security protocol (e.g., ESP) and outer IP headers. ROHC profiles have been defined to allow for the compression of the security protocol and the outer IP header on a hop-by-hop basis. The applicability of ROHCoIPsec and hop-by-hop ROHC on an IPv4 ESP-processed packet [ESP] is shown below in Figure 1.

```
          -----------------------------------------------------------
    IPv4  | new IP hdr  |     | orig IP hdr   |   |    | ESP   | ESP|
          |(any options)| ESP | (any options) |TCP|Data|Trailer| ICV|
          -----------------------------------------------------------
          |<-------(1)------->|<------(2)-------->|

          (1) Compressed hop-by-hop by the ROHC [ROHC]
              ESP/IP profile
          (2) Compressed end-to-end by the ROHCoIPsec [IPSEC-ROHC]
              TCP/IP profile
```

Figure 1. Applicability of hop-by-hop ROHC and ROHCoIPsec on an IPv4 ESP-processed packet.

If IPsec NULL encryption is applied to packets, ROHC may still be used to compress the inner headers at IPsec SA endpoints. However, compression of these inner headers may pose challenges for intermediary devices (e.g., traffic monitors, sampling/management tools) that are inspecting the contents of ESP-NULL packets. For example, policies on these devices may need to be updated to ensure that packets that contain the "ROHC" protocol identifier are not dropped. In addition, intermediary devices may require additional functionality to determine the content of the header compressed packets.

In certain scenarios, a ROHCoIPsec implementation may encounter UDP-encapsulated ESP or IKE packets (i.e., packets that are traversing NATs). For example, a ROHCoIPsec implementation may receive a UDP-encapsulated ESP packet that contains an ESP/UDP/IP header chain. Currently, ROHC profiles do not support compression of the entire header chain associated with this packet; only the UDP/IP headers can be compressed.

6. Details of the ROHCoIPsec Framework

6.1. ROHC and IPsec Integration

Figure 2 illustrates the components required to integrate ROHC with the IPsec process, i.e., ROHCoIPsec.

```
                  +-------------------------------+
                  | ROHC Module                   |
                  |                               |
                  |                               |
        +-----+   |     +-----+     +---------+   |
        |     |   |     |     |     |  ROHC   |   |
      --|  A  |---------|  B  |-----| Process |------> Path 1
        |     |   |     |     |     |         |   |   (ROHC-enabled SA)
        +-----+   |     +-----+     +---------+   |
           |      |        |                      |
           |      |        |-------------------------> Path 2
           |      |                               |   (ROHC-enabled SA,
           |      +-------------------------------+  but no compression)
           |
           |
           |
           |
           +-----------------------------------------> Path 3
                                                      (ROHC-disabled SA)
```

Figure 2. Integration of ROHC with IPsec

The process illustrated in Figure 2 augments the IPsec processing model for outbound IP traffic (protected-to-unprotected). Initial IPsec processing is consistent with RFC 4301 [IPSEC] (Section 5.1, Steps 1-2).

Block A: The ROHC data item (part of the SA state information) retrieved from the "relevant SAD entry" ([IPSEC], Section 5.1, Step3a) determines if the traffic traversing the SA is handed to the ROHC module. Packets selected to a ROHC-disabled SA MUST follow normal IPsec processing and MUST NOT be sent to the ROHC module

(Figure 2, Path 3). Conversely, packets selected to a ROHC-enabled SA MUST be sent to the ROHC module.

Block B: This step determines if the packet can be compressed. If the packet is compressed, an integrity algorithm MAY be used to compute an Integrity Check Value (ICV) for the uncompressed packet ([IPSEC-ROHC], Section 4.2; [IKE-ROHC], Section 3.1). The Next Header field of the security protocol header (e.g., ESP, AH) MUST be populated with a "ROHC" protocol identifier [PROTOCOL], inner packet headers MUST be compressed, and the computed ICV MAY be appended to the packet (Figure 2, Path 1). However, if it is determined that the packet will not be compressed (e.g., due to one of the reasons described in Section 6.1.3), the Next Header field MUST be populated with the appropriate value indicating the next-level protocol (Figure 2, Path 2), and ROHC processing MUST NOT be applied to the packet.

After the ROHC process completes, IPsec processing resumes, as described in Section 5.1, Step3a, of RFC 4301 [IPSEC].

The process illustrated in Figure 2 also augments the IPsec processing model for inbound IP traffic (unprotected-to-protected). For inbound packets, IPsec processing is performed ([IPSEC], Section 5.2, Steps 1-3) followed by AH or ESP processing ([IPSEC], Section 5.2, Step 4).

Block A: After AH or ESP processing, the ROHC data item retrieved from the SAD entry will indicate if traffic traversing the SA is processed by the ROHC module ([IPSEC], Section 5.2, Step 3a). Packets traversing a ROHC-disabled SA MUST follow normal IPsec processing and MUST NOT be sent to the ROHC module. Conversely, packets traversing a ROHC-enabled SA MUST be sent to the ROHC module.

Block B: The decision at Block B is made using the value of the Next Header field of the security protocol header. If the Next Header field does not indicate a ROHC header, the decompressor MUST NOT attempt decompression (Figure 2, Path 2). If the Next Header field indicates a ROHC header, decompression is applied. After decompression, the signaled ROHCoIPsec integrity algorithm MAY be used to compute an ICV value for the decompressed packet. This ICV, if present, is compared to the ICV that was calculated at the compressor. If the ICVs match, the packet is forwarded by the ROHC module (Figure 2, Path 1); otherwise, the packet MUST be dropped. Once the ROHC module completes processing, IPsec processing resumes, as described in Section 5.2, Step 4, of RFC 4301 [IPSEC].

When there is a single SA between a compressor and decompressor, ROHC MUST operate in unidirectional mode, as described in Section 5 of RFC 3759 [ROHC-TERM]. When there is a pair of SAs instantiated between

ROHCoIPsec implementations, ROHC MAY operate in bi-directional mode, where an SA pair represents a bi-directional ROHC channel (as described in Sections 6.1 and 6.2 of RFC 3759 [ROHC-TERM]).

Note that to further reduce the size of an IPsec-protected packet, ROHCoIPsec and IPComp [IPCOMP] can be implemented in a nested fashion. This process is detailed in [IPSEC-ROHC], Section 4.4.

6.1.1. Header Compression Protocol Considerations

ROHCv2 [ROHCV2] profiles include various mechanisms that provide increased robustness over reordering channels. These mechanisms SHOULD be adopted for ROHC to operate efficiently over IPsec SAs.

A ROHC decompressor implemented within IPsec architecture MAY leverage additional mechanisms to improve performance over reordering channels (either due to random events or to an attacker intentionally reordering packets). Specifically, IPsec’s sequence number MAY be used by the decompressor to identify a packet as "sequentially late". This knowledge will increase the likelihood of successful decompression of a reordered packet.

Additionally, ROHCoIPsec implementations SHOULD minimize the amount of feedback sent from the decompressor to the compressor. If a ROHC feedback channel is not used sparingly, the overall gains from ROHCoIPsec can be significantly reduced. More specifically, any feedback sent from the decompressor to the compressor MUST be processed by IPsec and tunneled back to the compressor (as designated by the SA associated with FEEDBACK_FOR). As such, some implementation alternatives can be considered, including the following:

- Eliminate feedback traffic altogether by operating only in ROHC Unidirectional mode (U-mode).

- Piggyback ROHC feedback messages within the feedback element (i.e., on ROHC traffic that normally traverses the SA designated by FEEDBACK_FOR).

6.1.2. Initialization and Negotiation of the ROHC Channel

Hop-by-hop ROHC typically uses the underlying link layer (e.g., PPP) to negotiate ROHC channel parameters. In the case of ROHCoIPsec, channel parameters can be set manually (i.e., administratively configured for manual SAs) or negotiated by IKEv2. The extensions required for IKEv2 to support ROHC channel parameter negotiation are detailed in [IKE-ROHC].

If the ROHC protocol requires bi-directional communications, two SAs MUST be instantiated between the IPsec implementations. One of the two SAs is used for carrying ROHC-traffic from the compressor to the decompressor, while the other is used to communicate ROHC-feedback from the decompressor to the compressor. Note that the requirement for two SAs aligns with the operation of IKE, which creates SAs in pairs by default. However, IPsec implementations will dictate how decompressor feedback received on one SA is associated with a compressor on the other SA. An IPsec implementation MUST relay the feedback received by the decompressor on an inbound SA to the compressor associated with the corresponding outbound SA.

6.1.3. Encapsulation and Identification of Header Compressed Packets

As indicated in Section 6.1, new state information (i.e., a new ROHC data item) is defined for each SA. The ROHC data item MUST be used by the IPsec process to determine whether it sends all traffic traversing a given SA to the ROHC module (ROHC-enabled) or bypasses the ROHC module and sends the traffic through regular IPsec processing (ROHC-disabled).

The Next Header field of the IPsec security protocol (e.g., AH or ESP) header MUST be used to demultiplex header-compressed traffic from uncompressed traffic traversing a ROHC-enabled SA. This functionality is needed in situations where packets traversing a ROHC-enabled SA contain uncompressed headers. Such situations may occur when, for example, a compressor only supports up to n compressed flows and cannot compress a flow number n+1 that arrives. Another example is when traffic is selected to a ROHC-enabled SA, but cannot be compressed by the ROHC process because the appropriate ROHC Profile has not been signaled for use. As a result, the decompressor MUST be able to identify packets with uncompressed headers and MUST NOT attempt to decompress them. The Next Header field is used to demultiplex these header-compressed and uncompressed packets where the "ROHC" protocol identifier will indicate that the packet contains compressed headers. To accomplish this, IANA has allocated value 142 to "ROHC" from the Protocol ID registry [PROTOCOL].

It is noted that the use of the "ROHC" protocol identifier for purposes other than ROHCoIPsec is currently not defined. In other words, the "ROHC" protocol identifier is only defined for use in the Next Header field of security protocol headers (e.g., ESP, AH).

The ROHC Data Item, IANA Protocol ID allocation, and other IPsec extensions to support ROHCoIPsec are specified in [IPSEC-ROHC].

6.1.4. Motivation for the ROHC ICV

Although ROHC was designed to tolerate packet loss and reordering, the algorithm does not guarantee that packets reconstructed at the decompressor are identical to the original packet. As stated in Section 5.2 of RFC 4224 [REORDR], the consequences of packet reordering between ROHC peers may include undetected decompression failures, where erroneous packets are constructed and forwarded to upper layers. Significant packet loss can have similar consequences.

When using IPsec integrity protection, a packet received at the egress of an IPsec tunnel is identical to the packet that was processed at the ingress (given that the key is not compromised, etc.).

When ROHC is integrated into the IPsec processing framework, the ROHC processed packet is protected by the AH/ESP ICV. However, bits in the original IP header are not protected by this ICV; they are protected only by ROHC's integrity mechanisms (which are designed for random packet loss/reordering, not malicious packet loss/reordering introduced by an attacker). Therefore, under certain circumstances, erroneous packets may be constructed and forwarded into the protected domain.

To ensure the integrity of the original IP header within the ROHCoIPsec-processing model, an additional integrity check MAY be applied before the packet is compressed. This integrity check will ensure that erroneous packets are not forwarded into the protected domain. The specifics of this integrity check are documented in Section 4.2 of [IPSEC-ROHC].

6.1.5. Path MTU Considerations

By encapsulating IP packets with AH/ESP and tunneling IP headers, IPsec increases the size of IP packets. This increase may result in Path MTU issues in the unprotected domain. Several approaches to resolving these path MTU issues are documented in Section 8 of RFC 4301 [IPSEC]; approaches include fragmenting the packet before or after IPsec processing (if the packet's Don't Fragment (DF) bit is clear), or possibly discarding packets (if the packet's DF bit is set).

The addition of ROHC within the IPsec processing model may result in similar path MTU challenges. For example, under certain circumstances, ROHC headers are larger than the original uncompressed headers. In addition, if an integrity algorithm is used to validate packet headers, the resulting ICV will increase the size of packets. Both of these properties of ROHCoIPsec increase the size of packets, and therefore may result in additional challenges associated with path MTU.

Approaches to addressing these path MTU issues are specified in Section 4.3 of [IPSEC-ROHC].

6.2. ROHCoIPsec Framework Summary

To summarize, the following items are needed to achieve ROHCoIPsec:

- IKEv2 Extensions to Support ROHCoIPsec

- IPsec Extensions to Support ROHCoIPsec

7. Security Considerations

Several security considerations associated with the use of ROHCoIPsec are covered in Section 6.1.4. These considerations can be mitigated by using a strong integrity-check algorithm to ensure the valid decompression of packet headers.

A malfunctioning or malicious ROHCoIPsec compressor (i.e., the compressor located at the ingress of the IPsec tunnel) has the ability to send erroneous packets to the decompressor (i.e., the decompressor located at the egress of the IPsec tunnel) that do not match the original packets emitted from the end-hosts. Such a scenario may result in decreased efficiency between compressor and decompressor, or may cause the decompressor to forward erroneous packets into the protected domain. A malicious compressor could also intentionally generate a significant number of compressed packets, which may result in denial of service at the decompressor, as the decompression of a significant number of invalid packets may drain the resources of an IPsec device.

A malfunctioning or malicious ROHCoIPsec decompressor has the ability to disrupt communications as well. For example, a decompressor may simply discard a subset of (or all) the packets that are received, even if packet headers were validly decompressed. Ultimately, this could result in denial of service. A malicious decompressor could also intentionally indicate that its context is not synchronized with the compressor’s context, forcing the compressor to transition to a lower compression state. This will reduce the overall efficiency gain offered by ROHCoIPsec.

8. IANA Considerations

All IANA considerations for ROHCoIPsec are documented in [IKE-ROHC] and [IPSEC-ROHC].

9. Acknowledgments

The authors would like to thank Sean O'Keeffe, 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:

- Magnus Westerlund
- Stephen Kent
- Pasi Eronen
- Joseph Touch
- Tero Kivinen
- Jonah Pezeshki
- Lars-Erik Jonsson
- Jan Vilhuber
- Dan Wing
- Kristopher Sandlund
- Ghyslain Pelletier
- David Black
- Tim Polk
- Brian Carpenter

Finally, the authors would also like to thank Tom Conkle, Renee Esposito, Etzel Brower, and Michele Casey of Booz Allen Hamilton for their assistance in completing this work.

10. Informative References

[ROHC] Sandlund, K., Pelletier, G., and L-E. Jonsson, "The RObust Header Compression (ROHC) Framework", RFC 5795, March 2010.

[IPSEC] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[ROHC-TERM] Jonsson, L-E., "Robust Header Compression (ROHC): Terminology and Channel Mapping Examples", RFC 3759, April 2004.

[BRA97] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[IKEV2] Kaufman, C., "Internet Key Exchange (IKEv2) Protocol", RFC 4306, December 2005.

[ESP] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.

[AH] Kent, S., "IP Authentication Header", RFC 4302, December 2005.

[IPSEC-ROHC] Ertekin, E., Christou, C., and C. Bormann, "IPsec Extensions to Support Robust Header Compression over IPsec", RFC 5858, May 2010.

[IKE-ROHC] Ertekin, E., Christou, C., Jasani, R., Kivinen, T., and C. Bormann, "IKEv2 Extensions to Support Robust Header Compression over IPsec", RFC 5857, May 2010.

[PROTOCOL] IANA, "Assigned Internet Protocol Numbers", <http://www.iana.org>.

[IPCOMP] Shacham, A., Monsour, B., Pereira, R., and M. Thomas, "IP Payload Compression Protocol (IPComp)", RFC 3173, September 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.

[REORDR] Pelletier, G., Jonsson, L-E., and K. Sandlund, "RObust Header Compression (ROHC): ROHC over Channels That Can Reorder Packets", RFC 4224, January 2006.

Authors' Addresses

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

EMail: ertekin_emre@bah.com

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

EMail: ro@breakcheck.com

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

EMail: christou_chris@bah.com

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

EMail: cabo@tzi.org