Internet Engineering Task Force (IETF)
Request for Comments: 6261
Category: Experimental
ISSN: 2070-1721

A. Keranen
Ericsson

May 2011

Encrypted Signaling Transport Modes for the Host Identity Protocol

Abstract

This document specifies two transport modes for Host Identity Protocol (HIP) signaling messages that allow them to be conveyed over encrypted connections initiated with the Host Identity Protocol.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for examination, experimental implementation, and evaluation.

This document defines an Experimental Protocol for the Internet community. 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). 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/rfc6261.

Table of Contents

1. Introduction

Host Identity Protocol (HIP) [RFC5201] signaling messages can be exchanged over plain IP using the protocol number reserved for this purpose, or over UDP using the UDP port reserved for HIP NAT traversal [RFC5770]. When two hosts perform a HIP base exchange, they set up an encrypted connection between them for data traffic, but continue to use plain IP or UDP for HIP signaling messages.

This document defines how the encrypted connection can be used also for HIP signaling messages. Two different modes are defined: HIP over Encapsulating Security Payload (ESP) and HIP over TCP. The benefit of sending HIP messages over ESP is that all signaling traffic (including HIP headers) will be encrypted. If HIP messages are sent over TCP (which in turn is transported over ESP), TCP can handle also message fragmentation where needed.

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 [RFC2119].

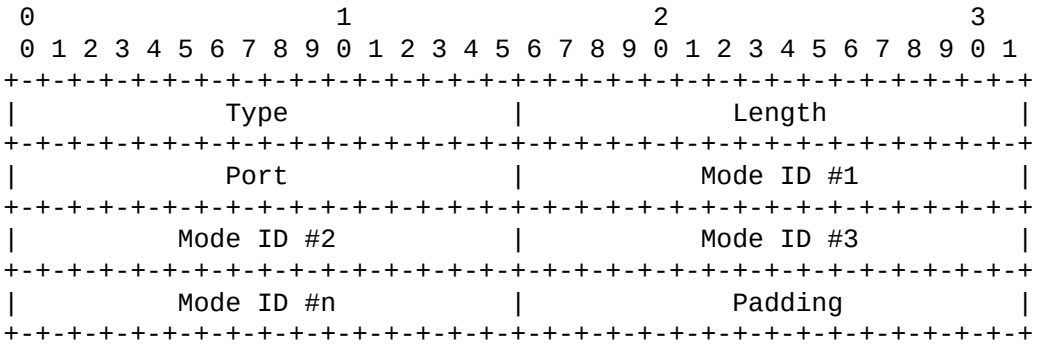
3. Transport Mode Negotiation

This section defines how support for different HIP signaling message transport modes is indicated and how the use of different modes is negotiated.

3.1. Mode Negotiation in the HIP Base Exchange

A HIP host implementing this specification SHOULD indicate the modes it supports, and is willing to use, in the base exchange. The HIP signaling message transport mode negotiation is similar to HIP NAT traversal mode negotiation: first the Responder lists the supported modes in a HIP_TRANSPORT_MODE parameter (see Figure 1) in the R1 packet. The modes are listed in priority order, the more preferred mode(s) first. If the Initiator supports, and is willing to use, any of the modes proposed by the Responder, it selects one of the modes by adding a HIP_TRANSPORT_MODE parameter containing the selected mode to the I2 packet. Finally, if the Initiator selected one of the modes and the base exchange succeeds, hosts MUST use the selected mode for the following HIP signaling messages sent between them for the duration of the HIP association or until another mode is negotiated.

If the Initiator cannot, or will not, use any of the modes proposed by the Responder, the Initiator SHOULD include an empty HIP_TRANSPORT_MODE parameter to the I2 packet to signal that it supports this extension but will not use any of the proposed modes. Depending on local policy, the Responder MAY either abort the base exchange or continue HIP signaling without using an encrypted connection, if there was no HIP_TRANSPORT_MODE parameter in I2 or the parameter was empty. If the Initiator selects a mode that the Responder does not support (and hence was not included in R1), the Responder MUST abort the base exchange. If the base exchange is aborted due to (possibly lack of) HIP_TRANSPORT_PARAMETER, the Responder SHOULD send a NO_VALID_HIP_TRANSPORT_MODE notification (see Section 3.3) to the Initiator.



Type 7680
Port transport layer port number (or zero if not used)
Length length in octets, excluding Type, Length, and Padding
Mode ID defines the proposed or selected transport mode(s)

The following HIP Transport Mode IDs are defined:

ID name	Value
RESERVED	0
DEFAULT	1
ESP	2
ESP-TCP	3

Figure 1: Format of the HIP_TRANSPORT_MODE Parameter

The mode DEFAULT indicates that the same transport mode (e.g., plain IP or UDP) that was used for the base exchange should be used for subsequent HIP signaling messages. In the ESP mode, the messages are sent as such on the encrypted ESP connection; in the ESP-TCP mode, TCP is used within the ESP tunnel. If a mode that uses a transport layer connection within the ESP tunnel (e.g., ESP-TCP) is offered, the Port field MUST contain the local port number the host will use for the connection. If none of the modes utilize a transport layer protocol, the Port field SHOULD be set to zero when the parameter is sent and ignored when received. The Port and Mode ID fields are encoded as unsigned integers using network byte order.

The HIP_TRANSPORT_MODE parameter resides on the signed part of the HIP packets, and hence it is covered by the signatures of the R1, I2, and UPDATE packets.

3.2. Mode Negotiation after the HIP Base Exchange

If a HIP host wants to change to a different transport mode (or start using a transport mode) some time after the base exchange, it sends a HIP UPDATE packet with a HIP_TRANSPORT_MODE parameter containing the mode(s) it would prefer to use. The host receiving the UPDATE SHOULD respond with an UPDATE packet containing the mode that is selected as in the negotiation during the base exchange. If the receiving host does not support, or is not willing to use, any of the listed modes, it SHOULD respond with an UPDATE packet where the HIP_TRANSPORT_MODE parameter contains only the currently used transport mode (even if that was not included in the previous UPDATE packet) and continue using that mode.

Since the HIP_TRANSPORT_MODE parameter's type is not critical (as defined in Section 5.2.1 of [RFC5201]), a host not supporting this extension would simply reply with an acknowledgement UPDATE packet without a HIP_TRANSPORT_MODE parameter. In such a case, depending on local policy as in mode negotiation during the base exchange, the host that requested the new transport mode MAY close the HIP association. If the association is closed, the host closing the association SHOULD send a NO_VALID_HIP_TRANSPORT_MODE NOTIFY packet to the other host before closing the association.

3.3. Error Notifications

During a HIP signaling transport mode negotiation, if a HIP_TRANSPORT_MODE parameter does not contain any mode that the receiving host is willing to use, or a HIP_TRANSPORT_MODE parameter does not exist in a HIP packet where the receiving host expected to see it, the receiving host MAY send back a NOTIFY packet with a NOTIFICATION parameter [RFC5201] error type NO_VALID_HIP_TRANSPORT_MODE (value 100). The Notification Data field for the error notifications SHOULD contain the HIP header of the rejected packet.

4. HIP Messages on Encrypted Connections

This specification defines two different transport modes for sending HIP packets over encrypted ESP connections. These modes require that the ESP transport format [RFC5202] is negotiated to be used between the hosts. If the ESP transport format is not used, these modes MUST NOT be offered in the HIP_TRANSPORT_MODE parameter. If a HIP_TRANSPORT_MODE parameter containing an ESP transport mode is received but the ESP transport format is not used, a host MUST NOT select such a mode but act as specified in Section 3.1 (if performing a base exchange) or Section 3.2 (if performing an UPDATE) when no valid mode is offered.

The ESP mode provides simple protection for all the signaling traffic and can be used as a generic replacement for the DEFAULT mode in cases where all signaling traffic should be encrypted. If the HIP messages may become so large that they would need to be fragmented, e.g., because of HIP certificates [RFC6253] or DATA messages [RFC6078], it is RECOMMENDED to use the ESP-TCP mode that can handle message fragmentation at the TCP level instead of relying on IP-level fragmentation.

When HIP NAT traversal [RFC5770] is used, the ESP and HIP packets are sent UDP encapsulated. The use of different NAT traversal modes, and in particular UDP encapsulation, is independent of the transport mode (as specified in this document) of HIP packets. However, when HIP packets are sent over an ESP connection, no additional UDP encapsulation (i.e., within the ESP connection) for the HIP packets is needed and MUST NOT be used since the ESP packets are already UDP encapsulated, if needed for NAT traversal. For example, if UDP encapsulation is used as defined in [RFC5770], and the ESP-TCP transport mode is used as defined in this document, the HIP packets are sent over IP, UDP, ESP, and TCP (in that order).

HIP messages that result in changing or generating new keying material, i.e., the base exchange and re-keying UPDATE messages, MUST NOT be sent over the encrypted connection that is created using the keying material that is being changed, nor over an encrypted connection using the newly created keying material.

It should be noted that when HIP messages are sent using an encrypted connection, on-path network elements (e.g., firewalls and HIP-aware NATs) that would normally see the HIP headers and contents of the unencrypted parameters, cannot see any part of the messages unless they have access to the encryption keying material. The original HIP design made an explicit decision to expose some of this information to HIP-aware NATs. If an encrypted transport mode is used, only the base exchange or update without encryption is visible to such NATs.

4.1. ESP Mode

If the ESP mode is selected in the base exchange, both hosts MUST listen for incoming HIP signaling messages and send outgoing messages on the encrypted connection. The ESP header's next header value for HIP messages sent over ESP MUST be set to HIP (139).

4.2. ESP-TCP Mode

If the ESP-TCP mode is selected, the host with the larger HIT (calculated as defined in Section 6.5 of [RFC5201]) MUST start to listen for an incoming TCP connection on the encrypted connection

(i.e., to the HIT of the host) on the port it used in the Port field of the transport mode parameter. The other host MUST create a TCP connection to that port and the host MAY use the port it sent in the transport mode parameter as the source port for the connection. Once the TCP connection is established, both hosts MUST listen for incoming HIP signaling messages and send the outgoing messages using the TCP connection. The ESP next header value for messages sent using the ESP-TCP mode TCP connections MUST be set to TCP (6).

If the hosts are unable to create the TCP connection, the host that initiated the mode negotiation MUST restart the negotiation with the UPDATE message and SHOULD NOT propose the ESP-TCP mode. If local policy does not allow use of any mode other than ESP-TCP, the HIP association SHOULD be closed. The UPDATE or CLOSE message MUST be sent using the same transport mode that was used for negotiating the use of the ESP-TCP mode.

Since TCP provides reliable transport, the HIP messages sent over TCP MUST NOT be retransmitted. Instead, a host SHOULD wait to detect that the TCP connection has failed to retransmit the packet successfully in a timely manner (such detection is platform- and policy-specific) before concluding that there is no response.

5. Recovering from Failed Encrypted Connections

If the encrypted connection fails for some reason, it can no longer be used for HIP signaling and the hosts SHOULD re-establish the connection using HIP messages that are sent outside of the encrypted connection. Hence, while listening for incoming HIP messages on the encrypted connection, hosts MUST still accept incoming HIP messages using the same transport method (e.g., UDP or plain IP) that was used for the base exchange. When responding to a HIP message sent outside of the encrypted connection, the response MUST be sent using the same transport method as the original message used. If encryption was previously used, hosts SHOULD send outside of the encrypted connection only HIP messages that are used to re-establish the encrypted connection. In particular, when the policy requires that only encrypted messages (e.g., DATA messages using an encrypted transport mode) be sent, they MUST be sent using an encrypted connection. Note that a policy MUST NOT prevent sending unencrypted UPDATE messages used for re-establishing the encrypted connection, since that would prevent recovering from failed encrypted connections.

The UPDATE messages used for re-establishing the encrypted connection MUST contain a HIP_TRANSPORT_MODE parameter and the negotiation proceeds as described in Section 3.2.

6. Host Mobility and Multihoming

If a host obtains a new address, a new Security Association (SA) pair may be created for (or an existing SA pair may be moved to) the new address, as described in [RFC5206]. If the ESP or ESP-TCP transport mode is used, HIP signaling continues using the (new) SA pair and the same transport mode as before.

With the ESP mode, the first mobility UPDATE message SHOULD be sent using the old SA, and the following messages, including the response to the first UPDATE, SHOULD be sent using the new SAs. Retransmissions of the UPDATE messages use the same SA as the original message. If the ESP-TCP mode is used, the HIP signaling TCP connection is moved to the new SA pair like any other TCP connection. However, the mobility UPDATE messages SHOULD NOT be sent over the TCP connection, but using plain ESP as in the ESP mode, and consequently hosts MUST be prepared to receive UPDATE messages over plain ESP even if the ESP-TCP mode is used.

In some cases, the host may not be able to send the mobility UPDATE messages using the encrypted connection before it breaks. This results in a similar situation as if the encrypted connection had failed and the hosts need to renegotiate the new addresses using unencrypted UPDATE messages and possibly rendezvous [RFC5204] or HIP relay [RFC5770] servers. Also, these UPDATE messages MUST contain the HIP_TRANSPORT_MODE parameter and perform the transport mode negotiation.

Examples of the signaling flows with mobility and multihoming are shown in Appendix A.

7. Security Considerations

By exchanging the HIP messages over an ESP connection, all HIP signaling data (after the base exchange but excluding keying material (re)negotiation and some of the mobility UPDATE messages) will be encrypted, but only if NULL encryption is not used. Thus, a host requiring confidentiality for the HIP signaling messages must check that encryption is negotiated for use on the ESP connection. Moreover, the level of protection provided by the ESP transport modes depends on the selected ESP transform; see [RFC5202] and [RFC4303] for security considerations of the different ESP transforms.

While this extension to HIP allows for negotiation of security features, there is no risk of downgrade attacks since the mode negotiation happens using signed (R1/I2 or UPDATE) packets and only after both hosts have been securely identified in the base exchange. If an attacker would attempt to change the modes listed in the

HIP_TRANSPORT_MODE parameter, that would break the signatures and the base exchange (or update) would not complete. Furthermore, since both "secure" modes (ESP and ESP-TCP) defined in this document are equally secure, the only possible downgrade attack would be to make both hosts accept the DEFAULT mode. If the local policy (of either host) requires using a secure mode, the base exchange or update would again simply fail (as described in Section 3.1).

8. Acknowledgements

Thanks to Gonzalo Camarillo, Kristian Slavov, Tom Henderson, Miika Komu, Jan Melen, and Tobias Heer for reviews and comments.

9. IANA Considerations

This section is to be interpreted according to [RFC5226].

This document updates the IANA maintained "Host Identity Protocol (HIP) Parameters" registry [RFC5201] by assigning a new HIP Parameter Type value (7680) for the HIP_TRANSPORT_MODE parameter (defined in Section 3.1).

The HIP_TRANSPORT_MODE parameter has 16-bit unsigned integer fields for different modes, for which IANA has created and now maintains a new sub-registry entitled "HIP Transport Modes" under the "Host Identity Protocol (HIP) Parameters" registry. Initial values for the transport mode registry are given in Section 3.1; future assignments are to be made through IETF Review or IESG Approval [RFC5226]. Assignments consist of a transport mode identifier name and its associated value.

This document also defines a new HIP NOTIFICATION message type [RFC5201] NO_VALID_HIP_TRANSPORT_MODE (100) in Section 3.3.

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC5201] Moskowitz, R., Nikander, P., Jokela, P., and T. Henderson, "Host Identity Protocol", RFC 5201, April 2008.

[RFC5202] Jokela, P., Moskowitz, R., and P. Nikander, "Using the Encapsulating Security Payload (ESP) Transport Format with the Host Identity Protocol (HIP)", RFC 5202, April 2008.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

10.2. Informational References

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.

[RFC5204] Laganier, J. and L. Eggert, "Host Identity Protocol (HIP) Rendezvous Extension", RFC 5204, April 2008.

[RFC5206] Nikander, P., Henderson, T., Vogt, C., and J. Arkko, "End-Host Mobility and Multihoming with the Host Identity Protocol", RFC 5206, April 2008.

[RFC5770] Komu, M., Henderson, T., Tschofenig, H., Melen, J., and A. Keranen, "Basic Host Identity Protocol (HIP) Extensions for Traversal of Network Address Translators", RFC 5770, April 2010.

[RFC6078] Camarillo, G. and J. Melen, "Host Identity Protocol (HIP) Immediate Carriage and Conveyance of Upper-Layer Protocol Signaling (HICCUPS)", RFC 6078, January 2011.

[RFC6253] Heer, T. and S. Varjonen, "Host Identity Protocol Certificates", RFC 6253, May 2011.

Appendix A. Mobility and Multihoming Examples

When changing interfaces due to mobility or multihoming, the hosts use HIP messages to notify the other host about the new address and to check that the host with the new address is still reachable. The following examples show the signaling performed during the address change in two different scenarios. Note that not all HIP parameters nor all the content of the parameters is shown in the examples. This section and the examples are not normative; for normative behavior, see previous sections.

In the examples, host A uses two different addresses (a1 and a2) while host B has just a single address (b1). In the first example, "Make before Break" (Figure 2), host A starts to use the new address but can still use the old address (due to multihoming) for signaling. In the second example, "Break before Make" (Figure 3), host A loses the first address before obtaining the second address (e.g., due to mobility), and the mobility HIP signaling is done without the encrypted connection.

The following notations are used in the examples:

- ESPx(y): data y sent encapsulated in ESP with SA x; if ESP-encapsulation is not used, the data is sent over plain IP or UDP
- UPDATE(x,y,z): HIP UPDATE message [RFC5201] with parameters x,y,z
- LOCATOR(x): HIP LOCATOR parameter [RFC5206] with locator x
- ESP_INFO(x,y): HIP ESP_INFO parameter [RFC5202] with "old SPI" value x and "new SPI" value y
- ACK, ECHO_REQ, and ECHO_RSP: HIP ACK, ECHO_REQUEST, and ECHO_RESPONSE parameters [RFC5201]

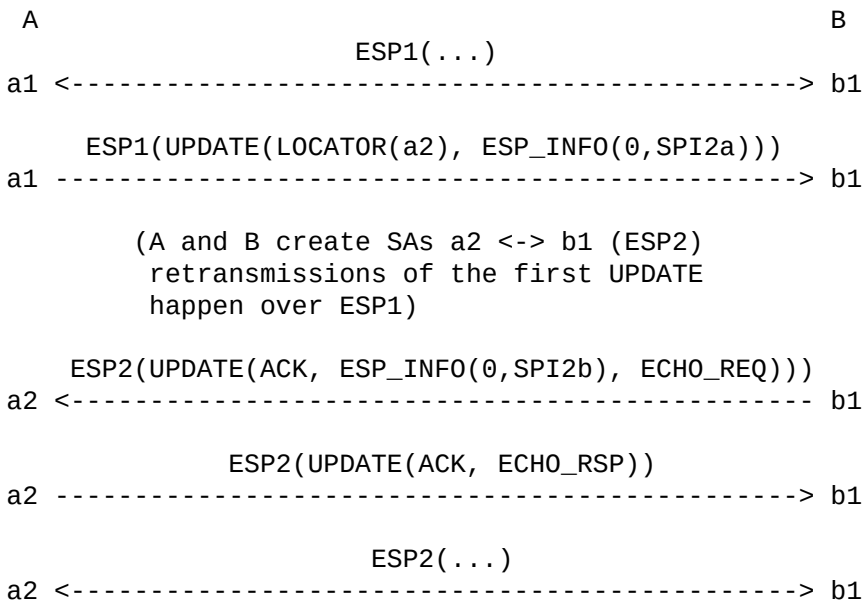


Figure 2: Make Before Break

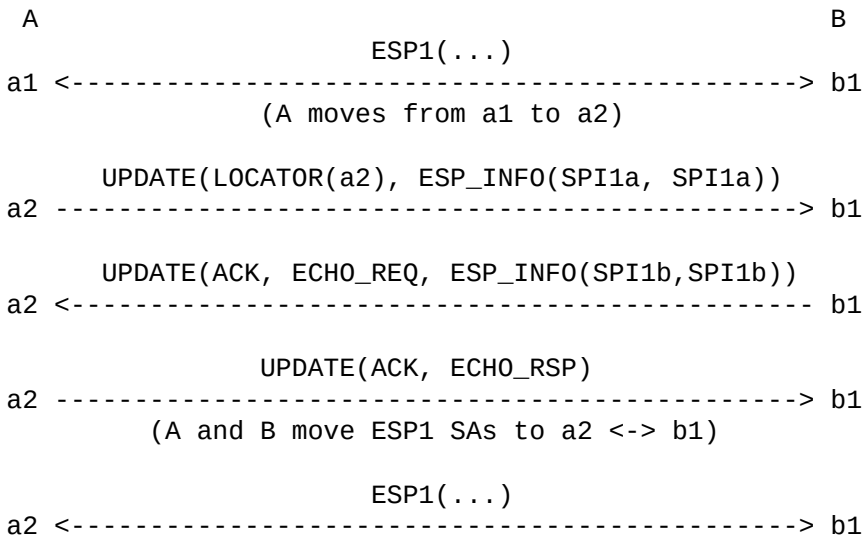


Figure 3: Break Before Make

When the ESP-TCP mode is used, the signaling flows are similar since TCP is not used for the mobility UPDATE messages as described in Section 6.

Author's Address

Ari Keranen
Ericsson
Hirsalantie 11
02420 Jorvas
Finland

EMail: ari.keranen@ericsson.com