

Network Working Group  
Internet-Draft  
Expires: January 10, 2005

L. Eggert  
NEC  
J. Laganier  
LIP / Sun Microsystems  
July 12, 2004

Host Identity Protocol (HIP) Rendezvous Extensions  
draft-eggert-hip-rvs-00

Status of this Memo

By submitting this Internet-Draft, I certify that any applicable patent or other IPR claims of which I am aware have been disclosed, and any of which I become aware will be disclosed, in accordance with [RFC 3668](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on January 10, 2005.

Copyright Notice

Copyright (C) The Internet Society (2004). All Rights Reserved.

## Abstract

This document discusses rendezvous extensions for the Host Identity Protocol (HIP). Rendezvous mechanisms extend HIP for communication with HIP Rendezvous Servers. Rendezvous Servers improve operation when HIP nodes are multi-homed or mobile. The first part of this document motivates the need for rendezvous mechanisms; the second part describes the protocol extensions in detail.

## Table of Contents

<a href="#">1.</a>	<a href="#">Introduction . . . . .</a>	<a href="#">4</a>
<a href="#">2.</a>	<a href="#">Communication Between HIP Nodes . . . . .</a>	<a href="#">5</a>
<a href="#">3.</a>	<a href="#">Communication Between Mobile or Multi-Homed HIP Nodes . . . . .</a>	<a href="#">7</a>
<a href="#">3.1</a>	<a href="#">Mobility and Multi-Homing with DNS Updates . . . . .</a>	<a href="#">7</a>
<a href="#">3.2</a>	<a href="#">Mobility and Multi-Homing with Rendezvous Servers . . . . .</a>	<a href="#">8</a>
<a href="#">4.</a>	<a href="#">HIP Extensions for Rendezvous Servers . . . . .</a>	<a href="#">10</a>
<a href="#">4.1</a>	<a href="#">Additional Control Fields in the HIP Base Header . . . . .</a>	<a href="#">10</a>
<a href="#">4.1.1</a>	<a href="#">RVS Control Field . . . . .</a>	<a href="#">10</a>
<a href="#">4.1.2</a>	<a href="#">CONCEAL_IP Control Field . . . . .</a>	<a href="#">10</a>
4.2	<a href="#">Additional HIP Parameters for Communication with     Rendezvous Servers . . . . .</a>	<a href="#">11</a>
<a href="#">4.2.1</a>	<a href="#">RVA_REQUEST Parameter Format and Processing . . . . .</a>	<a href="#">11</a>
<a href="#">4.2.2</a>	<a href="#">RVA_REPLY Parameter Format and Processing . . . . .</a>	<a href="#">11</a>
<a href="#">4.2.3</a>	<a href="#">RVA_HMAC Parameter Format and Processing . . . . .</a>	<a href="#">12</a>
<a href="#">4.2.4</a>	<a href="#">FROM Parameter Format and Processing . . . . .</a>	<a href="#">13</a>
<a href="#">4.2.5</a>	<a href="#">TO Parameter Format and Processing . . . . .</a>	<a href="#">14</a>
<a href="#">4.2.6</a>	<a href="#">VIA_RVS Parameter Format and Processing . . . . .</a>	<a href="#">15</a>
<a href="#">4.3</a>	<a href="#">Use of Existing HIP Messages and Parameters . . . . .</a>	<a href="#">15</a>
<a href="#">4.3.1</a>	<a href="#">ECHO_REQUEST and ECHO_REPLY Parameters . . . . .</a>	<a href="#">15</a>
<a href="#">4.3.2</a>	<a href="#">REA Parameter . . . . .</a>	<a href="#">16</a>
<a href="#">4.3.3</a>	<a href="#">NES Parameter . . . . .</a>	<a href="#">16</a>
<a href="#">5.</a>	<a href="#">Diagram Notation . . . . .</a>	<a href="#">17</a>
<a href="#">6.</a>	<a href="#">Establishing Rendezvous Associations . . . . .</a>	<a href="#">17</a>
<a href="#">7.</a>	<a href="#">Establishing HIP Associations via Rendezvous Servers . . . . .</a>	<a href="#">19</a>
<a href="#">7.1</a>	<a href="#">Sending a Redirect in Reply to I1 . . . . .</a>	<a href="#">19</a>

<a href="#">7.2</a>	Relaying I1 Only . . . . .	<a href="#">20</a>
<a href="#">7.2.1</a>	Passing I1 Through an ESP SA . . . . .	<a href="#">20</a>
<a href="#">7.2.2</a>	Rewriting I1 Destination IP Address . . . . .	<a href="#">21</a>
<a href="#">7.2.3</a>	Rewriting I1 Source and Destination IP Addresses . . . . .	<a href="#">22</a>
<a href="#">7.3</a>	Relaying Additional HIP Packets . . . . .	<a href="#">23</a>
<a href="#">7.3.1</a>	Concealing the Responder IP Address . . . . .	<a href="#">24</a>
<a href="#">7.3.2</a>	Concealing the Initiator IP Address . . . . .	<a href="#">25</a>
<a href="#">7.3.3</a>	Concealing Initiator and Responder IP Addresses . . . . .	<a href="#">26</a>
<a href="#">7.4</a>	Cascading Rendezvous Servers . . . . .	<a href="#">27</a>
<a href="#">7.5</a>	Opportunistic Initiators . . . . .	<a href="#">29</a>
<a href="#">7.6</a>	Implication on the HIP integrity checks . . . . .	<a href="#">29</a>
<a href="#">7.6.1</a>	Checksum . . . . .	<a href="#">29</a>
<a href="#">7.6.2</a>	HMAC and SIGNATURE . . . . .	<a href="#">29</a>
<a href="#">7.6.3</a>	Example . . . . .	<a href="#">29</a>
<a href="#">8.</a>	Security Considerations . . . . .	<a href="#">30</a>
<a href="#">9.</a>	Acknowledgments . . . . .	<a href="#">30</a>
<a href="#">10.</a>	References . . . . .	<a href="#">30</a>
<a href="#">10.1</a>	Normative References . . . . .	<a href="#">30</a>
<a href="#">10.2</a>	Informative References . . . . .	<a href="#">31</a>
	Authors' Addresses . . . . .	<a href="#">31</a>
<a href="#">A.</a>	Document Revision History . . . . .	<a href="#">32</a>

## 1. Introduction

The current Internet uses two global namespaces: domain names and IP addresses. The Domain Name System (DNS) provides a two-way lookup service between the two [[1](#)]. Domain names are symbolic identifiers for sets of IP addresses.

IP addresses have two uses. First, they are topological locators for network attachment points. Second, they act as names for the attached network interfaces. Saltzer [[10](#)] discusses these naming concepts in detail.

Routing and other network-layer mechanisms are based on the locator aspects of IP addresses. Transport-layer protocols and mechanisms typically use IP addresses in their role as names for communication endpoints.

This dual use of IP addresses limits the flexibility of the Internet architecture. The need to avoid readdressing in order to maintain existing transport-layer connections complicates advanced functionality, such as mobility, multi-homing, or network composition.

The Host Identity Protocol (HIP) architecture [[2](#)] defines a new third namespace. The Host Identity namespace decouples the name and locator roles currently filled by IP addresses. Instead of mapping domain names directly into IP addresses, HIP maps domain names into Host Identities, and Host Identities into IP addresses. Transport-layer mechanisms operate on Host Identities instead of using IP addresses as endpoint names. Network-layer mechanisms continue to use IP addresses as pure locators.

Without HIP, nodes establish transport-layer connections by first looking up the fully-qualified domain name (FQDN) of a peer in the DNS. A successful DNS lookup returns the peer's IP addresses. A node uses one of the returned IP addresses to initiate transport-layer communication with a peer node.

HIP nodes will also look up the domain name of desired peers in the DNS. When a successful lookup includes a peer's Host Identities, HIP nodes perform a HIP Base Exchange before establishing transport-layer connections. The HIP Base Exchange authenticates the end hosts and can bootstrap encryption of the subsequent communication with IPsec [[11](#)]. The HIP specification [[3](#)] discusses the details of the Base Exchange and the related protocol exchanges.

After the Base Exchange, HIP nodes use Host Identities instead of IP addresses for transport-layer connections with a peer. The HIP layer

in the network stack internally translates Host Identities (HI) into network-layer IP addresses. This additional mapping between Host Identities and IP addresses (HI->IP) is logically separate from the first mapping between fully-qualified domain names and Host Identities (FQDN->HI).

For application and transport-layer compatibility, the FQDN->HI mapping must remain in the DNS. However, the HI->IP mapping is internal to the HIP layer and may be performed in a number of ways. Different lookup mechanism may support communication between two mobile or multi-homed HIP nodes better [4].

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](#) [5].

## [2.](#) Communication Between HIP Nodes

In the current Internet, the DNS provides a FQDN->IP mapping. With HIP, it must continue to provide a mapping based on domain names. This allows transport-layer connections to bind to Host Identities instead of IP addresses transparently.

Instead of mapping domain names directly into IP addresses (FQDN->IP), with HIP the DNS maps them to Host Identities (FQDN->HI). In a second step, another lookup that is internal to the HIP-layer translates the Host Identities into IP addresses for network-layer delivery (HI->IP).

Several alternative approaches are possible for maintaining the HI->IP information. The DNS can maintain this mapping along with the FQDN->HI mapping. Alternatively, a database separate from the DNS can manage this information. This section discusses the different approaches and their implications on communication between two HIP nodes.

The HIP architecture and protocol specifications suggest storing Host Identities along with a node's IP addresses in the DNS [2][3]. The index for both tables will be domain names. Logically, the DNS will thus contain two separate mappings: FQDN->HI and FQDN->IP.

Figure 1 shows the lookup steps and HIP Base Exchange when a node's Host Identities are stored alongside its IP addresses. In step #1, the initiator I performs a DNS lookup on R's domain name FQDN(R). The DNS server responds with both R's Host Identities HI(R) and its IP addresses IP(R) in step #2.

The initiator I uses both pieces of information to perform the HIP

Base Exchange with R in step #3. (The details of the Base Exchange, specified in [3], are not relevant to this discussion and will thus be omitted.)

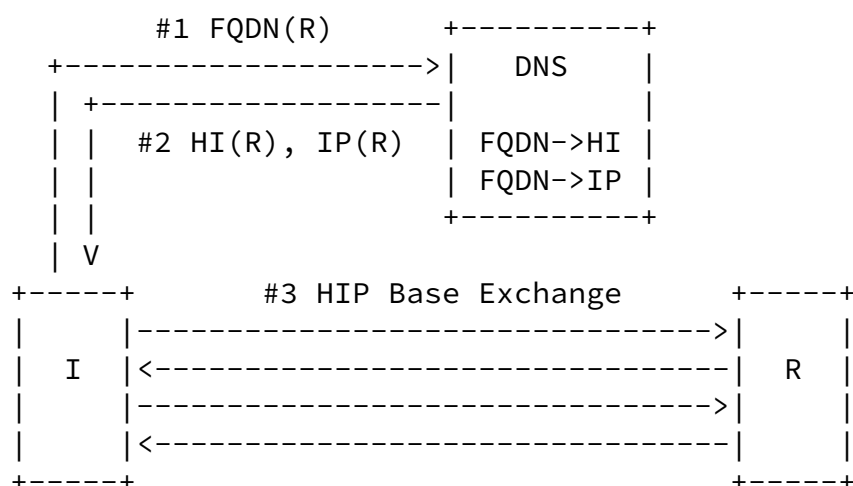


Figure 1: HIP Lookup and Base Exchange

Note that the DNS does not currently store the HI->IP mapping

directly. Instead, a DNS lookup on a domain name returns both its FQDN->HI and FQDN->IP entries. The HIP stack then implicitly constructs the HI->IP mapping based on the HI and IP information returned by the DNS lookup. In the example in Figure 1, the FQDN(R) lookup in step #1 returns both HI(R) and IP(R) in step #2. HIP implicitly constructs the HI(R)->IP(R) mapping based on the assumption that HI(R) is reachable at IP(R).

One disadvantage of this approach is that a node's domain name is required to obtain both its Host Identities and its IP addresses. Even if a HIP node already knows the Host Identity of a HIP peer through other means, it cannot currently obtain the peer's IP addresses through the DNS. The DNS does not maintain an explicit HI->IP table, but instead indexes Host Identities only by domain names.

A reverse HI->FQDN DNS mapping could address this limitation. HIP nodes would then look up a HIP peer's domain name through its Host Identity. They would then use the returned domain name to find the peer's IP addresses in a second lookup. However, the DNS may not be structurally suited to maintain the reverse HIP->FQDN mapping. As the main Internet-wide database, the DNS is already being overloaded with functionality that might be better handled with new mechanisms [12]. Finally, the additional reverse lookup would increase the latency of the HIP Base Exchange.

### [3.](#) Communication Between Mobile or Multi-Homed HIP Nodes

HIP decouples domain names from IP addresses. Because transport protocols bind to Host Identities, they remain unaware if the set of IP addresses associated with a Host Identity changes. This change can have various reasons, including, but not limited to, mobility and multi-homing.



Proposed extensions for mobility and multi-homing [4] allow a HIP node to notify its peers about changes in its set of IP addresses. These extensions require an established HIP association between two nodes, i.e., a completed HIP Base Exchange.

In addition to notifying its current peers about changes in its IP addresses, a HIP node must also update its HI->IP mapping in response to IP address changes. Otherwise, HIP Base Exchanges from new peers could fail because they try to contact the node at an IP address it is no longer reachable at.

### 3.1 Mobility and Multi-Homing with DNS Updates

If the DNS indirectly maintains the HI->IP mapping in a FQDN->IP table, nodes can dynamically update their DNS entry in a secure fashion [6][7]. The DNS server maintaining the information will then sign and distribute the updated zone.

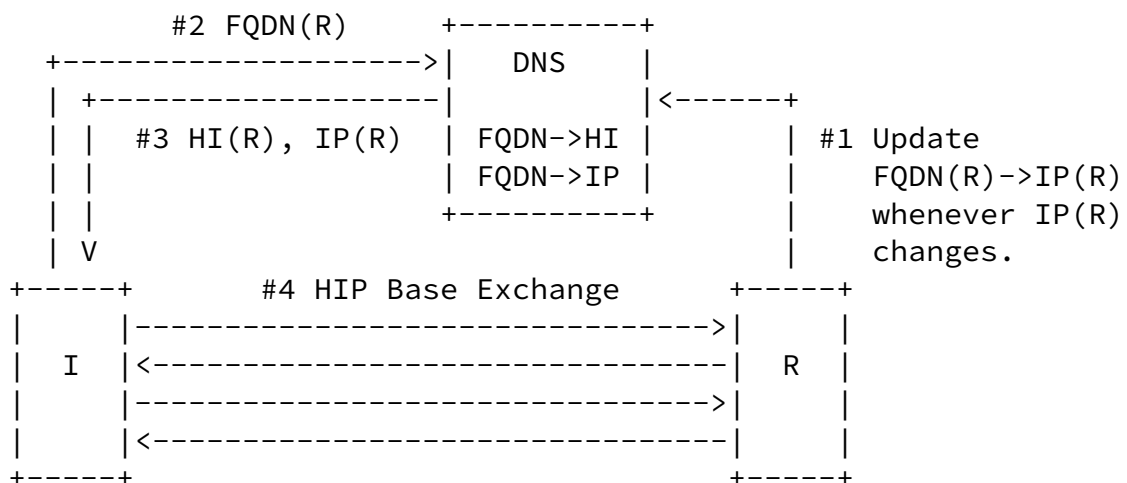


Figure 2: HIP Lookup and Base Exchange with DNS Updates

Figure 2 shows an example of this scenario. In step #1, R registers its FQDN(R)→IP(R) entry in the DNS. It will dynamically update the DNS entry whenever its IP addresses IP(R) change. Because the DNS always contains R's current IP addresses, node I can perform a HIP Base Exchange with R at its new IP address (steps #2-4).

One drawback of using dynamic DNS updates in this way is the cost of updating secure zones. Re-signing an entire zone whenever the IP addresses of one entry change places a high cost on the DNS server. Using dynamic DNS to update HI->IP mappings may thus not be appropriate when changes of IP addresses are frequent.

A simple, operational change could help limit the costs of frequent DNS updates. Instead of recomputing a zone after each dynamic update, a DNS server could aggregate the modifications and only perform zone updates periodically. The disadvantage of this approach is that HIP nodes may be unreachable until the DNS server distributes the updated zone.

Another concern with using the DNS to support HIP node mobility is the propagation time of updated DNS entries. DNS servers frequently cache DNS responses to reduce the load on the primary servers. During the time-to-live associated with a DNS response, DNS servers may answer additional requests for the same DNS entry from their local caches instead of contacting the primary servers. Thus, even after a HIP node updates its DNS entry, the DNS can still serve the old entry until the cached responses expire. This can lead to communication problems, because peers may try to contact a HIP node at an IP address it is no longer reachable at.

### [3.2](#) Mobility and Multi-Homing with Rendezvous Servers

The HIP architecture tries to greatly reduce the frequency of Dynamic DNS updates by introducing Rendezvous Servers [\[2\]](#). Instead of registering its current set of IP addresses in its HI->IP entry in the DNS, a HIP node may instead register the IP addresses of its Rendezvous Servers. Because the IP addresses of Rendezvous Servers are assumed to change only infrequently, this approach can significantly reduce the load on DNS servers.

Rendezvous Servers maintain a mapping between the Host Identities of HIP nodes for which they provide service and the node's current IP addresses. HIP nodes must notify their Rendezvous Servers about any changes in their IP addresses. This approach effectively relocates the HI->IP information - and the burden of keeping it current - from the DNS to the Rendezvous Servers. This can reduce update costs

under the assumption that Rendezvous Servers provide more efficient ways of maintaining HI->IP tables.

When a packet destined for one of its HIP nodes arrives at a Rendezvous Server, it relays the packet to one of the HIP node's current IP addresses. Due to the specifics of the HIP, only the first packet of a HIP Base Exchange will require such relaying [2]. Subsequent packet of the HIP Base Exchange and all further data

packets will directly flow between the HIP nodes, bypassing the Rendezvous Server.

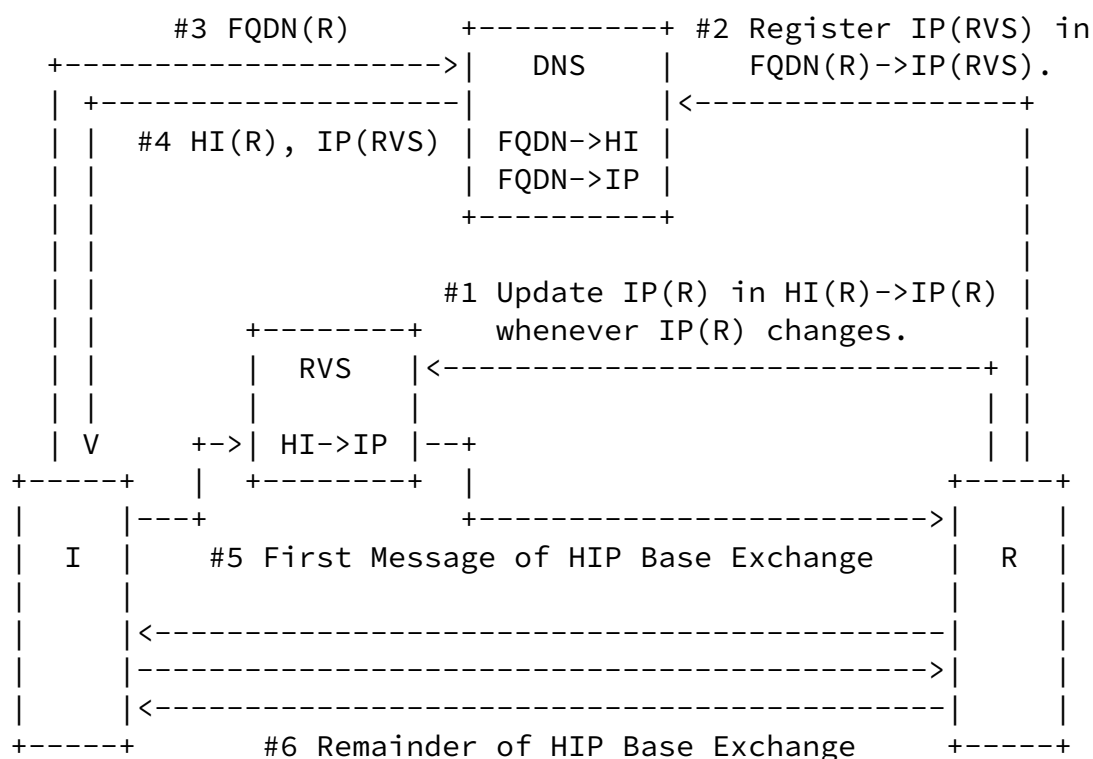


Figure 3: HIP Lookup and Base Exchange with Rendezvous Server

Figure 3 shows a HIP lookup and Base Exchange involving a Rendezvous

Server. Here, HIP node R is using Rendezvous Server RVS. In step #1, it updates RVS with its current IP addresses IP(R). Then, in step #2, R registers the Rendezvous Server's IP addresses IP(RVS) in its FQDN(R)->IP(RVS) DNS entry.

In step #3, a second HIP node I issues a DNS lookup on FQDN(R) to obtain R's Host Identities HI(R) and IP addresses. The lookup returns R's Host Identities HI(R) in step #4. The DNS reply also includes the IP addresses of the Rendezvous Server IP(RVS) (instead of IP(R), because R's current addresses are unknown to the DNS.)

In step #5, node I initiates the HIP Base Exchange. It addresses the first packet of the HIP Base Exchange to IP(RVS). Upon receipt, the Rendezvous Server relays the packet to one of R's current IP addresses IP(R). The remainder of the HIP Base Exchange then occurs directly between I and R in step #6.

When Rendezvous Servers maintain the HI->IP information, they may support more efficient update operations compared to dynamic DNS updates ([Section 3.1](#)). Unlike the DNS, Rendezvous Servers do not provide a lookup service. Instead, they use the HI->IP information

to actively relay traffic between HIP nodes.

This approach changes the role of the IP addresses stored in a DNS entry. Traditionally, nodes were directly reachable at the IP addresses listed in their DNS entry. HIP Rendezvous Server change this basic property by replacing the IP addresses of their client nodes in the DNS with their own. The IP addresses in a DNS entry hence no longer directly designate interfaces of an endpoint. Instead, they identify interfaces of a node that can relay packets to the endpoint.

#### [4.](#) HIP Extensions for Rendezvous Servers

The following sections describe HIP extensions for communication with Rendezvous Servers. These extensions allow:

- o A HIP Rendezvous Server to advertise its RVS capabilities to its correspondents.
- o A HIP node to create a Rendezvous Association (RVA) with its Rendezvous Server, i.e., to register its current set of IP address(es).
- o two HIP nodes to establish a HIP Association (HA) between them via one or more Rendezvous Server.

#### [4.1](#) Additional Control Fields in the HIP Base Header

RVS mechanisms make use of two new Control Fields in the HIP Control Field: RVS\_CAPABLE and CONCEAL\_IP Control Fields. These new fields are used to, respectively, advertise Rendezvous Server capabilities, and query downstream RVS for concealing source IP addresses.

##### [4.1.1](#) RVS Control Field

The RVS\_CAPABLE Control Field ("R") allows a Rendezvous Server to advertise its rendezvous capabilities to the HIP nodes it associates with.

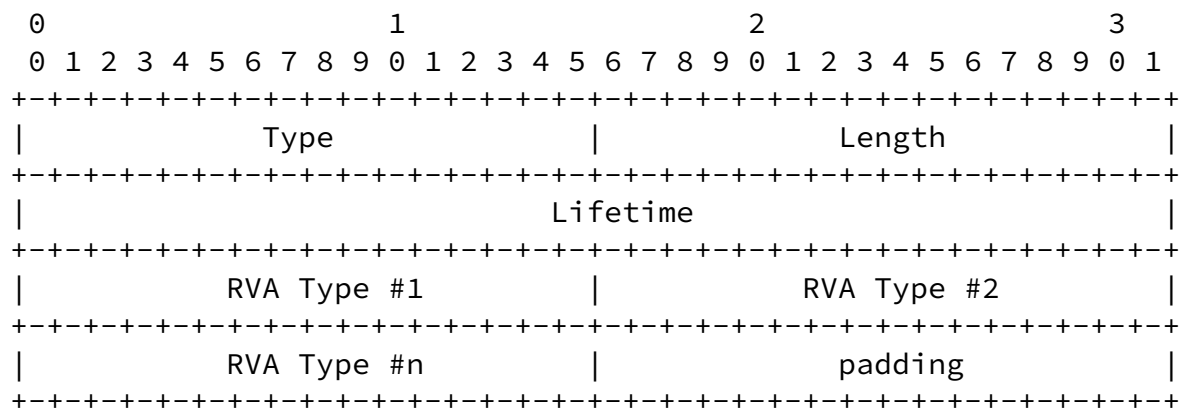
##### [4.1.2](#) CONCEAL\_IP Control Field

The CONCEAL\_IP Control Field ("C") is used by a HIP node to query downstream Rendezvous Servers to conceal its IP address. The RVS conceals the sender's IP address of a HIP packet by (1) replacing the packet's source IP address field with its own address, and by (2) omitting to add a FROM parameter containing the sender's IP address.

A RVS receiving a HIP packet with the CONCEAL\_IP Control Field set MUST NOT augment the packet with a FROM parameter while relaying it. If the relaying cannot be accomplished without FROM parameter, the RVS MUST drop the packet, and MAY notify the original sender.

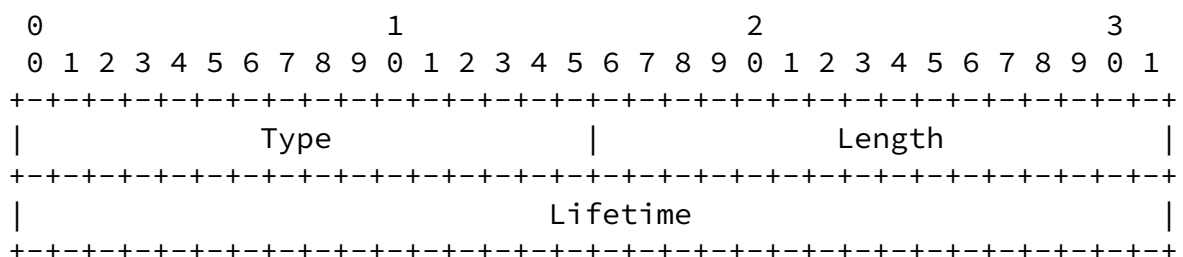
## [4.2](#) Additional HIP Parameters for Communication with Rendezvous Servers

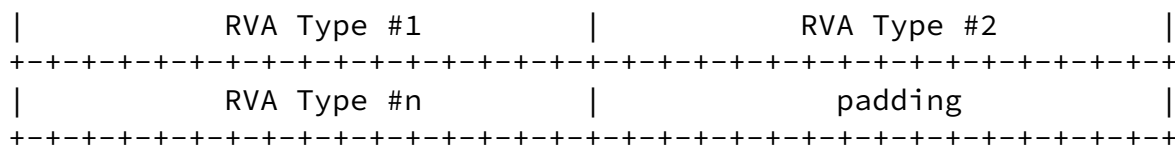
### [4.2.1](#) RVA\_REQUEST Parameter Format and Processing



Type	100
Length	Length in octets, excluding Type, Length and Padding
Lifetime	This field encode, the desired RVA validity time.
RVA Type	This field encode, in order of preference, the preferred rendezvous service types.

### [4.2.2](#) RVA\_REPLY Parameter Format and Processing





Type102

LengthLength in octets, excluding Type, Length and Padding

LifetimeThis field encode the offered RVA validity time

RVA Type

This field encode, in order of preference, the preferred rendezvous service types.

The following RVA Type are defined:

Type number	RVA Type
0	Reserved
1	RELAY_I1
2	RELAY_I1R1
3	RELAY_I1R1I2
4	RELAY_I1R1I2R2
5	RELAY_ESP_I1
6	REDIRECT_I1

4.2.3 RVA\_HMAC Parameter Format and Processing

The RVA\_HMAC is an OPTIONAL parameter whose only difference with the HMAC parameter defined in [3] is the Type code:

Type65320

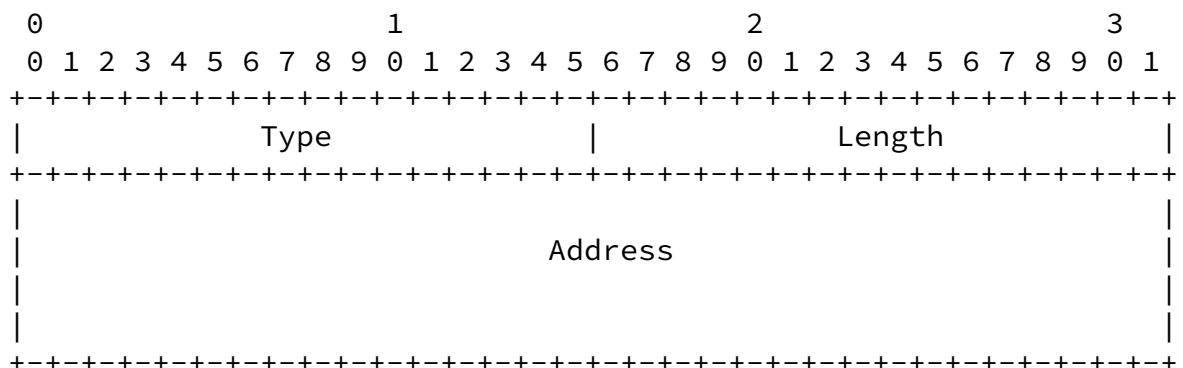
Length20

HMAC                    160 low order bits of a HMAC keyed with the appropriate HIP integrity keys (HIP\_lg or HIP\_gl) of the corresponding Rendezvous Association or HIP Association. This HMAC is computed over the HIP packet excluding RVA\_HMAC and any other following parameter. The checksum field **MUST** be set to zero and the HIP header length in the HIP common header **MUST** be calculated not to cover any excluded parameter when the Authenticator field is calculated.

To allow a HIP node and any of its RVS to verify the integrity of packets flowing between them, both use an RVA\_HMAC parameter keyed with a HMAC of HIP\_lg and HIP\_gl integrity keys. One RVA\_HMAC **SHOULD** be present on every packets flowing between a HIP node and any of its RVS and **MUST** be present when FROM and TO parameters are processed.

On the receiving side, when an RVA\_HMAC is validated, it **SHOULD** be removed from the packet and if so, packet length and checksum **MUST** be recomputed accordingly.

#### [4.2.4](#) FROM Parameter Format and Processing





Type	65100 (under signature) or 65300 (after signature)
Length	16
Address	An IPv6 address or an IPv4-in-IPv6 format IPv4 address

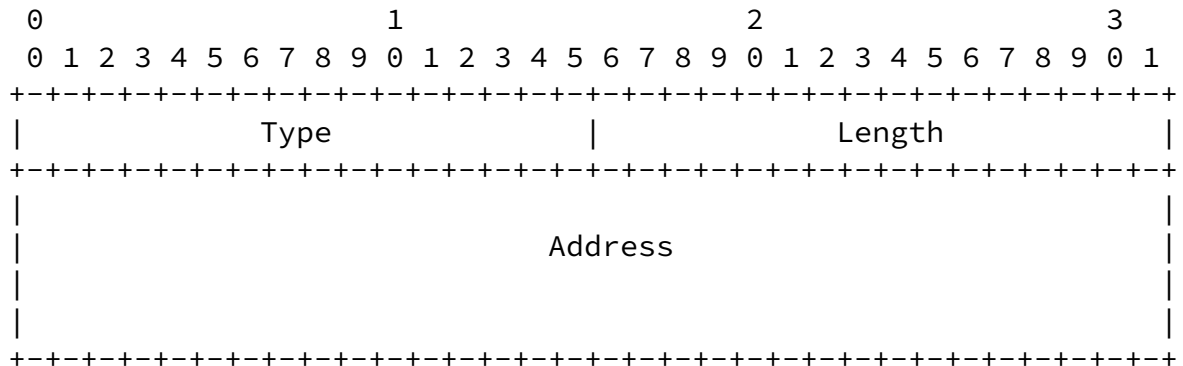
A Rendezvous Server MAY add a FROM parameter containing the original source IP address of a HIP packet (I1, R1, I2 or R2) whose source IP address has been rewritten. If one or more FROM parameters are already present, the new FROM parameter MUST be appended after the existing ones. Each time an RVS inserts a FROM parameter, it MUST also insert additional parameters that will be used to validate this and the subsequent HIP packets. These parameters are:

- o An ECHO\_REQUEST, containing a chunk of opaque data allowing to validate, in a possible subsequent answer, a T0 parameter which MUST be protected by an ECHO\_RESPONSE containing the same opaque data.
- o A valid RVA\_HMAC, protecting the packet integrity.

When a HIP node validates a FROM parameter, it is removed from the packet and recorded for later use (i.e., for building the corresponding T0 parameter to be piggybacked onto a subsequent answer). The packet's source IP address is also replaced by the address included in the first occurrence of FROM parameter.

For each FROM parameter, a HIP node MAY add to its replies a T0 parameter containing the IP address included in the FROM. These replies will be sent via the RVS, which MUST remove the outer T0 parameter from the packet and replace its destination address with the address contained in the T0 parameter before relaying it.

#### 4.2.5 T0 Parameter Format and Processing



Type	65102 (under signature) or 65302 (after signature)
Length	16
Address	An IPv6 address or an IPv4-in-IPv6 format IPv4 address

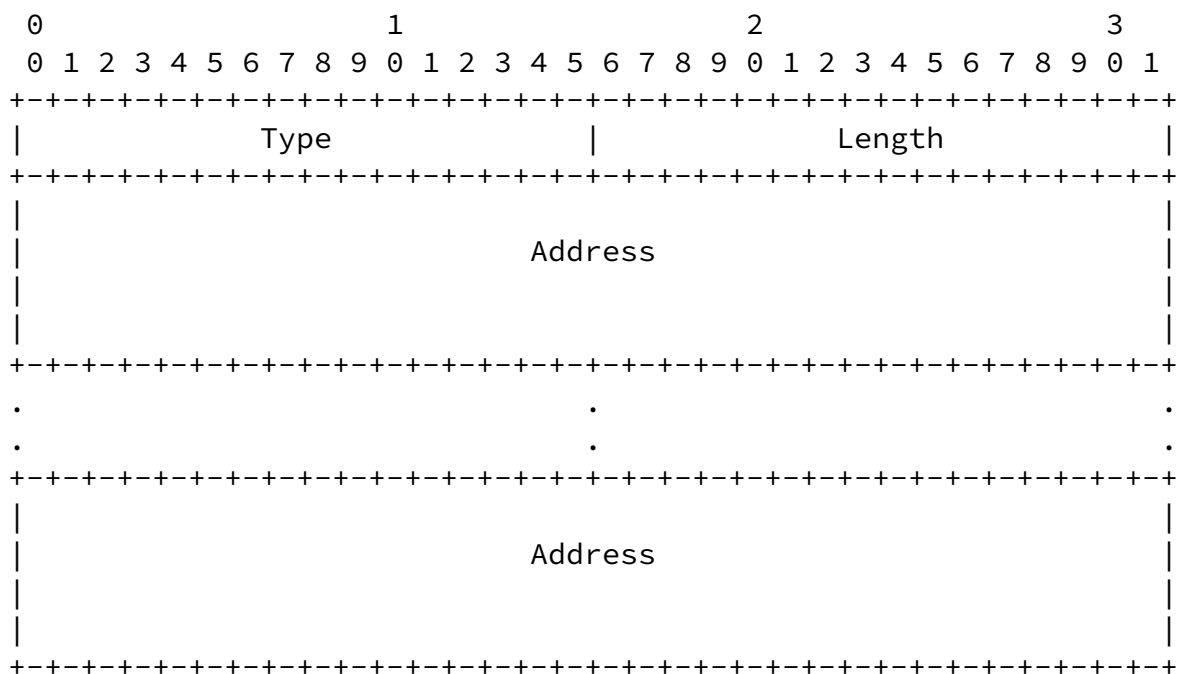
A HIP node MAY add one or more T0 parameter containing the final destination IP address of a HIP packet (I1, R1, I2 or R2) whose destination IP address needs to be rewritten by an RVS. This is essentially equivalent to loose source-routing. If one or more T0 parameters are already present, the new T0 parameter MUST be appended after the existing ones. Each time a node inserts a T0 parameter, it MUST also insert additional parameters that will be used by the RVS for validation. These parameters are:

- o An ECHO\_RESPONSE, containing a chunk of opaque data allowing the RVS to validate the address contained in the T0 parameter.
- o A valid RVA\_HMAC, protecting the packet integrity.

When the RVS validates a T0 parameter, SHALL remove it from the packet, and SHALL replace the packet destination IP address with the address included in the T0 parameter. Packet length and checksum MUST then be recomputed accordingly.

For each FROM parameter, a HIP node MAY add to its replies a T0 parameter containing the IP address included in the FROM. These replies will be sent via the RVS, which MUST remove the outer T0 parameter from the packet and replace its destination address field with the address contained in the T0 parameter before relaying it.

#### [4.2.6](#) VIA\_RVS Parameter Format and Processing



Type	65500
Length	Variable
Address	An IPv6 address or an IPv4-in-IPv6 format IPv4 address

At some point a, HIP endpoint might be in position to begin to send HIP packets directly towards the remote HIP endpoint's IP address, without further assistance from one or more of its RVS(s). In that case, it MAY include in these packets a subset of the IP address(es)

of its Rendezvous Servers by either:

- o Add its IP address into an existing VIA\_RVS parameter situated at the end of the HIP packet, while modifying accordingly the size of the parameter.
- o Appending a newly created VIA\_RVS parameter at the end of the HIP packet if it does not already contain a VIA\_RVS parameter.

Note that the main goal of the using the VIA\_RVS parameter is to allow operators to diagnose possible issues encountered while establishing a HIP association via a RVS.

### [4.3](#) Use of Existing HIP Messages and Parameters

#### [4.3.1](#) ECHO\_REQUEST and ECHO\_REPLY Parameters

A FROM parameter MAY be augmented by including an ECHO\_REQUEST

Eggert & Laganier	Expires January 10, 2005	[Page 15]
-------------------	--------------------------	-----------

---

Internet-Draft	HIP Rendezvous Extensions	July 2004
----------------	---------------------------	-----------

parameter to the carrying packet. The contents of the ECHO\_REQUEST might then be echoed back in ECHO\_RESPONSE.

A TO parameter SHOULD be augmented and authenticated by including an ECHO\_REPLY parameter to the carrying packet. The contents of the ECHO\_REPLY MUST be copied from a previously received ECHO\_RESPONSE.

All the HIP packets requiring RVS relaying facility to carry an answer packet SHOULD be augmented by the RVS with an ECHO\_REQUEST parameter.

A possible packet answered via the RVS, thus requiring relaying

facility, SHOULD be authenticated by an ECHO\_REPLY parameter. The contents of the ECHO\_REPLY MUST be copied from a previously received ECHO\_RESPONSE.

On the receiving side, when a HIP node validates an ECHO\_REPLY located after the signatures, it MUST remove it from the packet and recompute packet length and checksum accordingly.

#### [4.3.2](#) REA Parameter

A HIP node associated via an RVS MAY use a REA parameter to make its correspondent aware of its veritable current IP address. If used, the REA parameter MUST be used in conformance with the guidelines specified in [\[4\]](#). In addition, a HIP node MAY initiate the protocol later during the base exchange by using the REA parameter in the R2 packet. This R2-with-REA packet MUST be treated as a UPDATE-with-REA, i.e., trigger a Routability Return check by generating and sending a new SPI stored in a NES parameter included in an UPDATE packet.

#### [4.3.3](#) NES Parameter

A HIP node receiving a REA packet later than I2 MUST perform a Routability Return check before sending data to the new IP address. This check is performed by replying to an incoming REA with a NES parameter containing a new SPI to be used, as described in [\[4\]](#).

## 5. Diagram Notation

Notation -----	Significance -----
I, R	I and R are the respective source and destination IP addresses of the IP header
HIT-I, HIT-R	HIT-I and HIT-R are respectively the Initiator and the Responder HIT of the packet
R	The RVS_CAPABLE Control Field is set into the Control Field of the HIP header
C	The CONCEAL_IP Control Field is set into the Control Field of the HIP header
REA:I	A REA parameter containing the IP address i is present in the HIP header
FROM:I	A FROM parameter containing the IP address I is present in the HIP header
TO:I	A TO parameter containing the IP address I is present in the HIP header
VIA_RVS:RVS	A VIA_RVS parameter containing IP addresses RVS is present in the HIP header
EREQ	An ECHO_REQUEST parameter is present in the HIP header
EREP	An ECHO_REPLY parameter is present in the HIP header
RREQ	A RVA_REQUEST parameter is present in the HIP header
RREP	A RVA_REPLY parameter is present in the HIP header

## 6. Establishing Rendezvous Associations

A HIP node that wants to register its IP address with its RVS MAY simply establish a HIP association with it. It MUST then keep its IP address current with the server by sending UPDATE packets whenever its set of IP addresses changes.

However, for the sake of economizing RVS resources, which can possibly be used by several thousands of different HIP nodes, we

Eggert & Laganier

Expires January 10, 2005

[Page 17]

---

Internet-Draft

HIP Rendezvous Extensions

July 2004

define a new sort of "soft state" HIP association called a Rendezvous Association (RVA). In order to maintain this RVA established, a HIP Association need not remain established.

A HIP node MAY establish an RVA with its RVS by establishing a HA while adding an RVA\_REQUEST parameter in an I2, possibly preceded by an I1 containing the same RVA\_REQUEST. The possibility offered to initiate the protocol in I1 allows a HIP node to query a RVS for the set of offered rendezvous service types before completing the establishment of the Rendezvous association (in case the desired service type isn't available on this RVS). A RVS MUST then reply with, respectively, an R2 possibly preceded by an R1, which will both have the RVS\_CAPABLE control field set, and contain a RVA\_REPLY parameter specifying the characteristics of the offered RVA (validity time, type, etc.). Then, the RVS and the HIP node MAY delete most of the HIP Association state, retaining only the Lifetime, Initiator's HIT and IP address(es), as well as HIP\_lg and HIP\_gl integrity keys.

When a HA is established via an RVS, the integrity of HIP packets flowing between a HIP node and its RVS is protected by an additional RVA\_HMAC keyed with these keys.

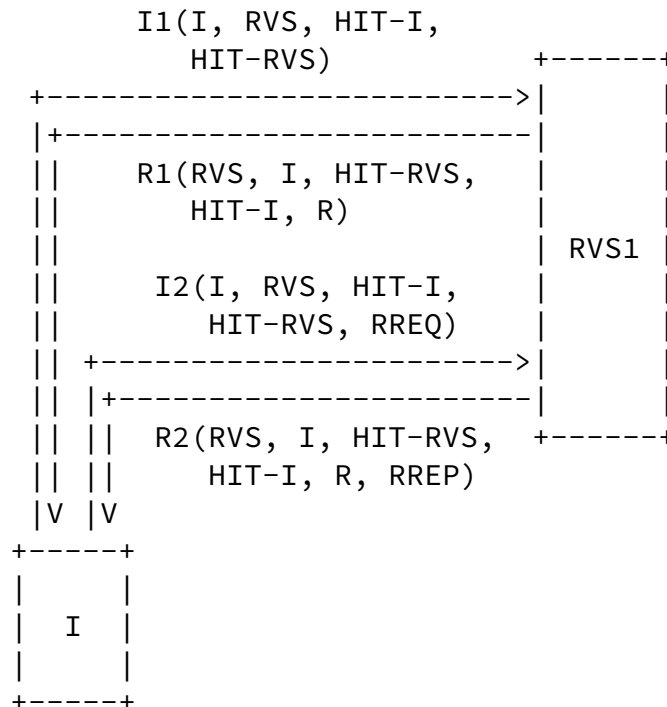


Figure 12: Establishing a Rendezvous Association

There is nothing to prevent an RVS node to advertise its RVS capabilities to the peers it associates with, nor to establish an RVA with another RVS.

If a HIP node wants to associate with several cascaded Rendezvous Servers  $RVS_i$  ( $0 < i < n+1$ ), it SHALL sequentially create RVAs ( $RVA_i$ ) with each of them, starting from the "nearest" ( $RVS_1$ ) to the "farthest" ( $RVS_n$ ). Apart from  $RVA_1$ , a node SHOULD create any such  $RVA_i$  ( $1 < i < n+1$ ) by sending an I1 to  $RVS_i$  via each of the RVS which precede it, i.e.,  $RVS_j$  ( $1 < j < i$ ).

This is achieved by using  $(i - 1)$  different T0 parameters containing, in order, the IP address of each RVS preceding  $RVS_i$ , i.e.,  $RVS_j$  ( $1$



< j < i). This process is similar to IP loose source-routing. Hence, A RVS accepting to be part of a cascade MAY relay an incoming I1 from one its clients to any given address and HIT. Those I1s MUST be protected by a valid RVA\_HMAC parameter.

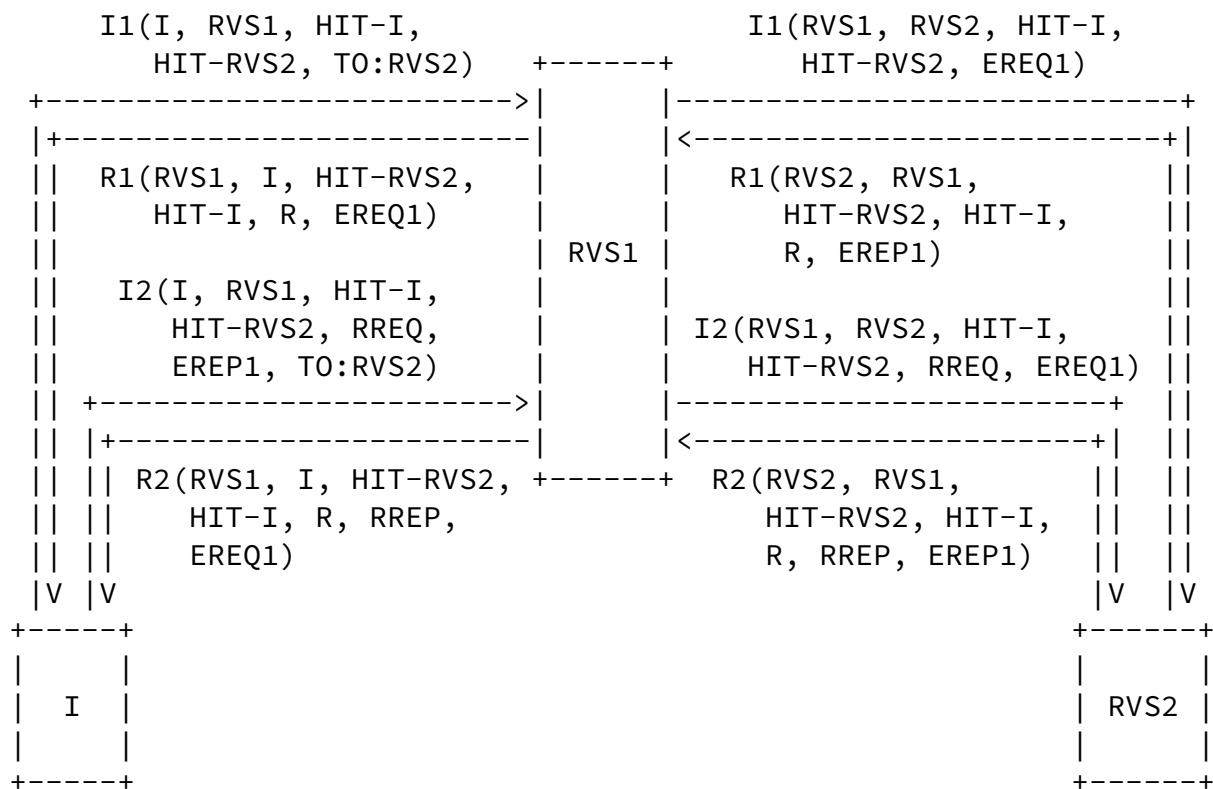


Figure 13: Establishing Cascaded Rendezvous Associations

## 7. Establishing HIP Associations via Rendezvous Servers

### 7.1 Sending a Redirect in Reply to I1

Instead of having the RVS relay incoming I1s to the correct Responder, one possibility is to answer with a redirect packet when a HIP packet destined for one of the Rendezvous Server's HIP nodes arrives. This redirect packet contains the IP address and packet

signature of the Responder.

The Responder cannot sign the redirect packets delivered by the RVS in real time. When the RVA is set up, the Responder sends the signed redirect packet to the RVS, who stores it until the RVA expires.

This redirect packet can be implemented by using a REA parameter embedded into a NOTIFY packet that includes a SIGNATURE2 parameter for protection. Note that this may expose the Initiator to replay attacks, but this is not very different from the situation where the Initiator receives a signed R1 whose signature omits Receiver HIT.

However, because an initiator might be unaware of the HI of the responder, knowing only its HIT, it might not be able to verify this SIGNATURE2. Hence, it is necessary to include in this redirect packet the HI of the responder, thus allowing the initiator to verify the signatures based on a previously known HIT.

## [7.2](#) Relaying I1 Only

### [7.2.1](#) Passing I1 Through an ESP SA

If a HIP node and one of its Rendezvous Servers maintain a HIP Association, the Rendezvous Server MAY tunnel I1s incoming to this node's HIT into the corresponding ESP SA. The main drawbacks of this approach are that, (1) middleboxes cannot see the encrypted I1 passing from an RVS to its clients, and (2) the source IP address of I1 is lost. In particular, (2) implies that the RVS MUST transmit to the responder the original source IP address by either of the following:

- o add a FROM parameter to the HIP header
- o include the whole original IP header in the ESP payload (very similar to ESP tunnel mode)
- o route back the subsequent R1 via the RVS

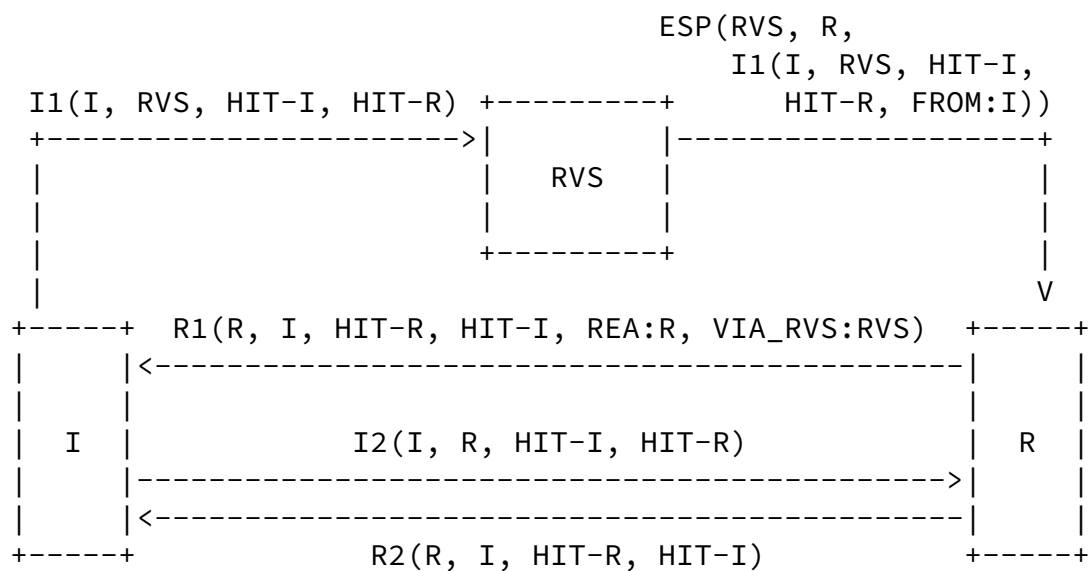


Figure 14: Rendezvous Server Forwarding I1 Through an ESP SA

### [7.2.2](#) Rewriting I1 Destination IP Address

When a HIP packet destined for one of its HIP nodes arrives at a Rendezvous Server, it relays the packet to one of the HIP node's current IP addresses. In most case, it is expected that only the first packet of a HIP Base Exchange (i.e., I1) will require such

relaying [2]. Subsequent packet of the HIP Base Exchange and all further data packets will directly flow between the HIP nodes, bypassing the Rendezvous Server.

In the simplest case, the Rendezvous Server can relay an I1 towards its true destination by merely replacing the destination IP address of the I1 by one of the destination HIT owner's IP address(es). Note, however, that such I1s might be subject to egress filtering on the Rendezvous Server's network [8], thus causing I1 packet to be dropped (source IP address does not belong to the RVS network).

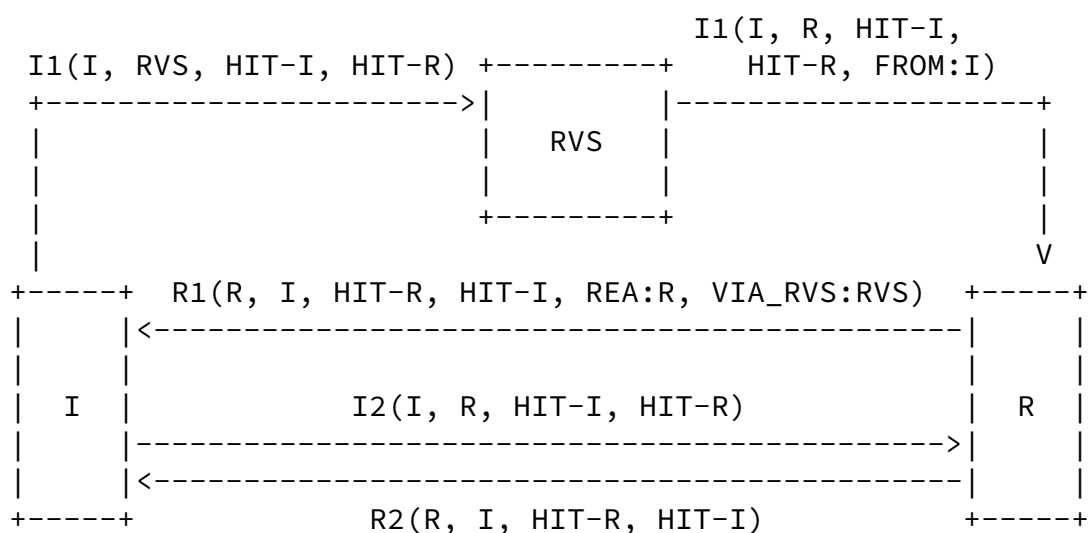


Figure 15: Rendezvous Server Rewriting I1 Destination IP Address

### [7.2.3](#) Rewriting I1 Source and Destination IP Addresses

Because of egress filtering, a HIP Rendezvous Server might need to replace the original source IP address of an I1 by its own IP address, thus concealing the Initiator's IP address to the Responder.

While this might be desirable, one of the extension described in this document allows a Rendezvous Server to piggy-back incoming HIP packets with an OPTIONAL FROM parameter containing the original source IP address of the packet. A HIP node receiving a packet containing such a FROM parameter has two possibilities for answering back. It might answer back either:

- o Directly to the IP address included in the FROM parameter, thus disclosing its IP address.
- o Via the Rendezvous Server IP address, adding to the HIP header a TO parameter containing the IP address included in the FROM parameter, thus being able to conceal its IP address to its correspondent.

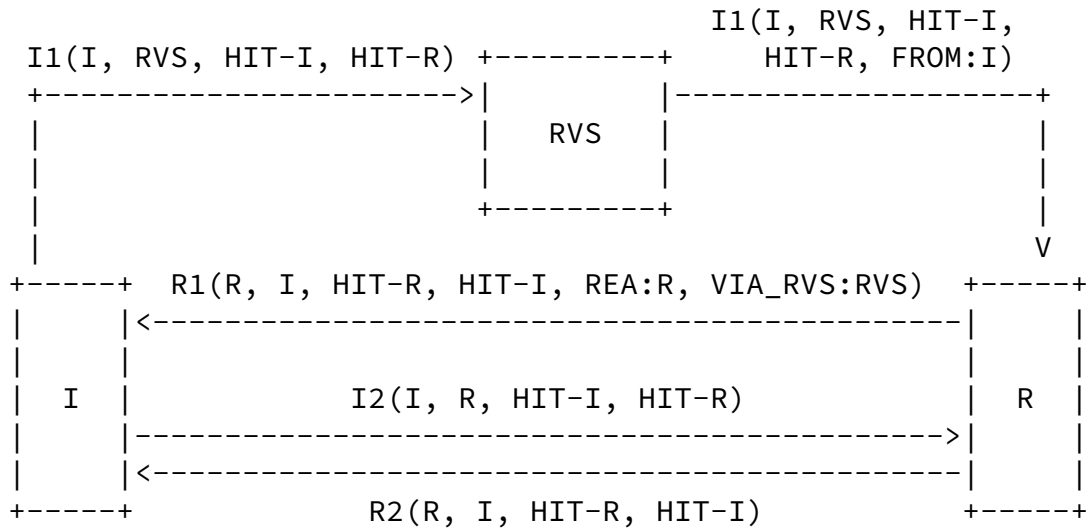


Figure 16: Rendezvous Server Rewriting I1 Source and Destination IP Addresses

### 7.3 Relaying Additional HIP Packets

It might be useful to relay further HIP packets (i.e., R1, I2 and R2) via the RVS, for example for concealing HIP nodes IP addresses until they have authenticated each other.

Because these packets are larger than I1 (they contain public keys and signatures), the relaying of such packet create an opportunity for denial of service attacks. To defend against these attacks, the Rendezvous Server needs to differentiate between legitimate HIP packets (i.e., I1 and subsequent HIP packets triggered by an I1) and illegitimate ones.

For the sake of reducing the load incurred on the RVS, an RVS is not required to keep track of IP addresses and other pieces of state associated with ongoing HIP exchanges. Such behavior is OPTIONAL. Instead, the relaying facility MAY make use of ECHO\_REQUEST and ECHO\_RESPONSE parameters.

Each time a packet is being relayed, the RVS MAY augment it with an ECHO\_REQUEST parameter containing a chunk of opaque data. The receiver of such a packet SHOULD augment any packet answering to this packet with an ECHO\_REPLY parameter containing the same chunk of opaque data. This opaque data allows an RVS to find and validate the answered packet IP addresses and HITs. When successfully validated, ECHO\_REPLY parameters SHOULD be removed from the packet before

relaying.

### [7.3.1](#) Concealing the Responder IP Address

As mentioned before, a Responder MAY want to conceal its IP address(es) to an Initiator whose Host Identity has not yet been validated by an I2. Such a Responder SHOULD set the CONCEAL\_IP Control Field in the HIP packets (R1 and R2) it sends. The Rendezvous Server then MUST replace the source IP address of relayed HIP packets with its own one without appending a FROM parameter.

The Responder MUST NOT include a REA parameter before it receives a valid I2. This situation also requires the Responder to send back via the RVS an R1 to the Initiator. Then, the Initiator will send via the RVS an I2 to the Responder, causing the Responder to send directly to the Initiator an R2 containing a REA parameter with its current IP address(es).

[4] does not describe any method to initiate the readdressing protocol in an R2 (by adding a REA parameter). A Responder MAY initiate the readdressing protocol in R2. The Initiator SHOULD then perform a Routability Return check by answering with an UPDATE packet including a NES. The Responder will then use the new SPI to send ESP packet to the Initiator.

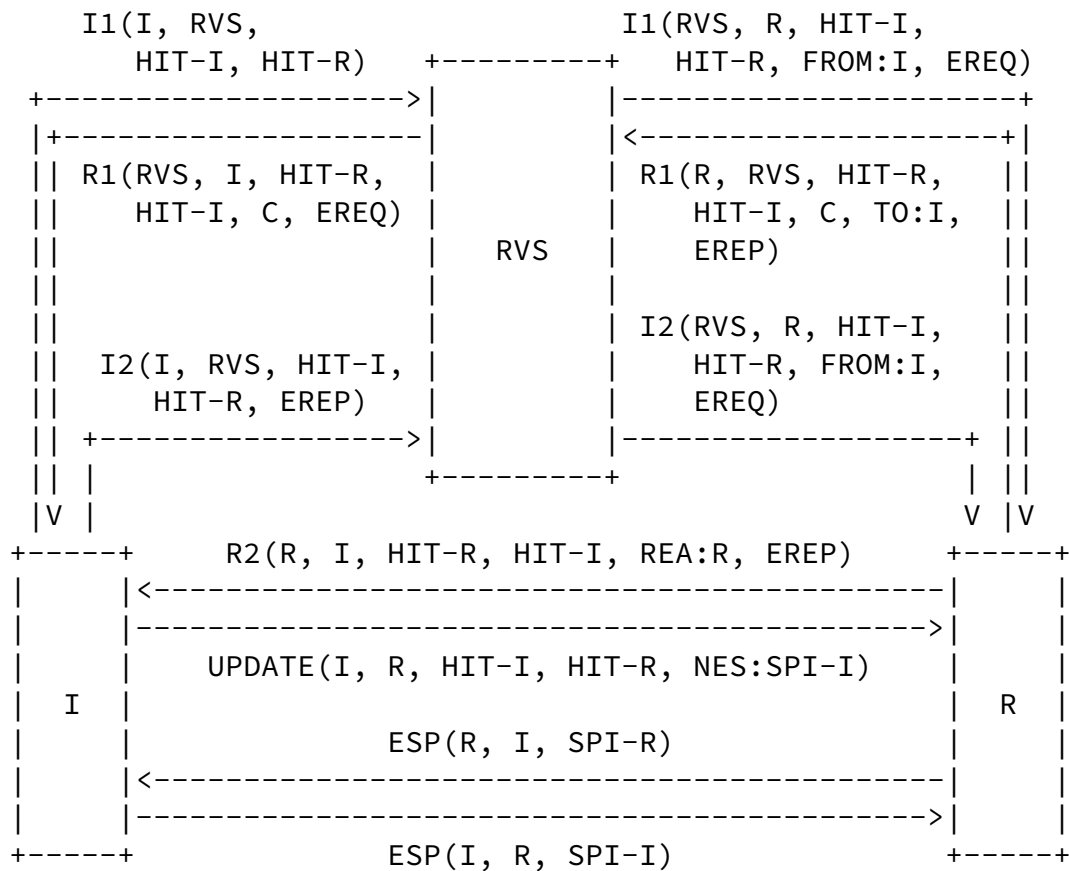




Figure 17: Responder Concealing its IP address

### [7.3.2](#) Concealing the Initiator IP Address

Similarly, an Initiator might want to conceal its IP address(es) to a Responder whose Host Identity has not yet been validated by R2. Such an Initiator set the CONCEAL\_IP Control Field in the HIP packets (I1 and I2) it sends.

The Rendezvous Server then replace the source IP address of relayed HIP packets with its own one without appending a FROM parameter.

The Initiator MUST NOT include a REA parameter before he received a valid I2. This situation also requires the Responder to send back via the RVS an R1 to the Initiator. Then, the Initiator will sends via the RVS an I2 to the Responder. This will cause the Responder to send via the RVS to the Initiator an R2 containing a REA parameter with its current IP address(es).

[4] does not describe any method to initiate the readdressing protocol in an R2 (by adding a REA parameter). A Responder MAY initiate the readdressing protocol in R2. The Initiator SHOULD then

perform a Routability Return check by answering with an UPDATE packet including a NES. The Responder will then use the new SPI to sends ESP packet to the Initiator.

The Initiator should then initiate a "classic" readdressing protocol by sending UPDATE packets including a REA parameter, as per [\[4\]](#).

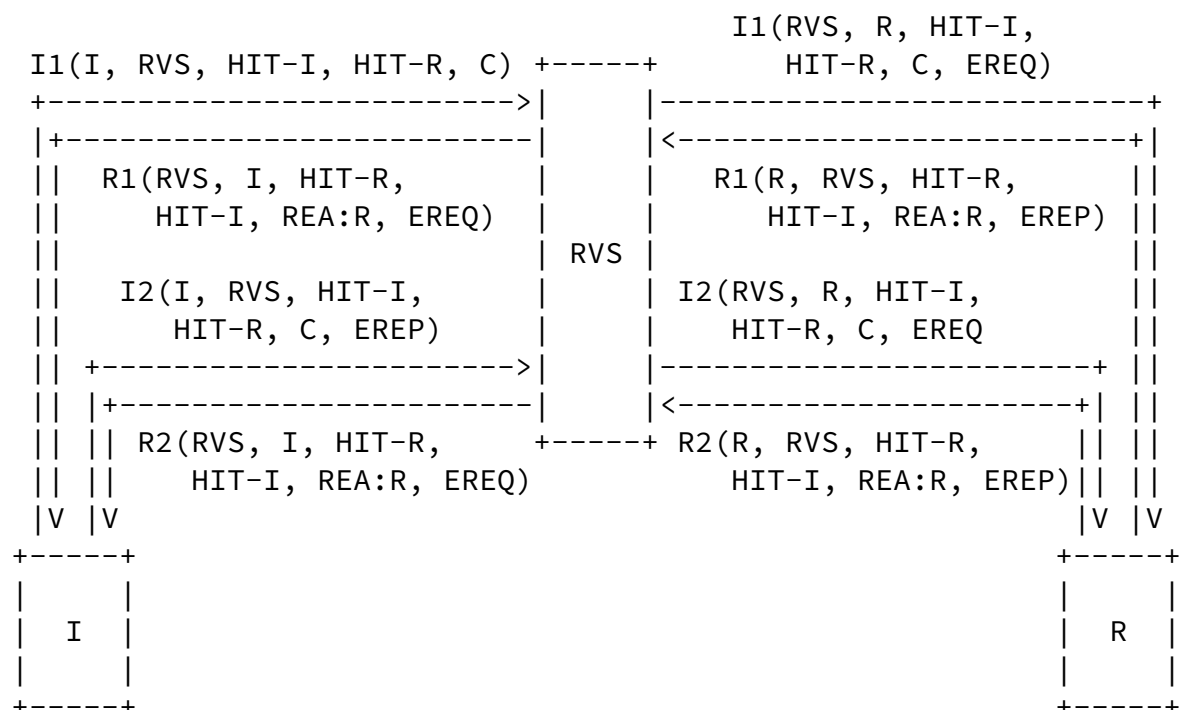


Figure 18: Initiator Concealing its IP address

At this point, the functionality described here has not been verified to not introduce new opportunities for DoS and DDoS attacks, because the responder is unaware of the original source IP address of a packet. Hence, it is questionable if a responder accepting concealed initiator(s) should be able, while establishing an RVA with it RVS, to negotiate a rate-limit on the throughput of relayed IIs. This might be done by adding a Rate Limit field in the RVA\_REQUEST and RVA\_REPLY parameter.

### 7.3.3 Concealing Initiator and Responder IP Addresses

This situation combines the two variant of IP address concealing described previously: both Initiator and Responder want to conceal their IP addresses until their correspondent's Host Identity is validated by, respectively, a R2 and an I2. All the HIP packets prior to, and including, R2, MUST be exchanged via the RVS with the CONCEAL\_IP Control Field set.

The Rendezvous Server then replace the source IP address of relayed HIP packets with its own one without appending a FROM parameter.

Both Initiator and Responder MUST NOT include a REA parameter before they received and validated, respectively, an R2 or a I2. This situation also requires the Responder to send back via the RVS R1 and R2 to the Initiator. Then, the Initiator will sends via the RVS an I2 to the Responder. This will cause the Responder to send via the RVS to the Initiator an R2 containing a REA parameter with its current IP address(es).

[4] does not describe any method to initiate the readdressing protocol in an R2 (by adding a REA parameter). A Responder MAY initiate the readdressing protocol in R2. The Initiator SHOULD then (1) perform a Routability Return check by answering with an UPDATE packet including a NES as in [4], and (2), SHOULD initiate a readdressing protocol with the same update, as in [4]. The Initiator and Responder MUST then use the new SPIs for future ESP packets.

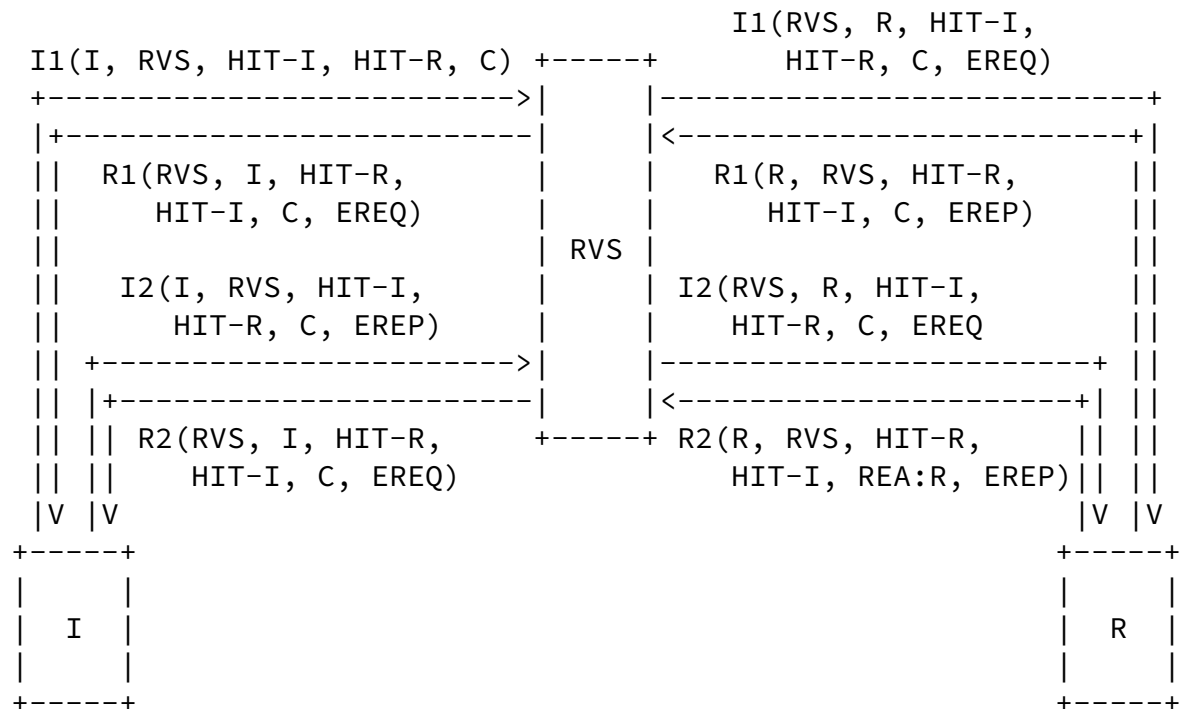


Figure 19: Initiator and Responder Concealing their IP addresses

## 7.4 Cascading Rendezvous Servers

In some situations, it might be useful to use cascaded Rendezvous Servers to establish RVS associations. A typical scenario would be a small number of "trusted" Rendezvous Servers and a larger number of

Eggert & Laganier

Expires January 10, 2005

[Page 27]

Internet-Draft

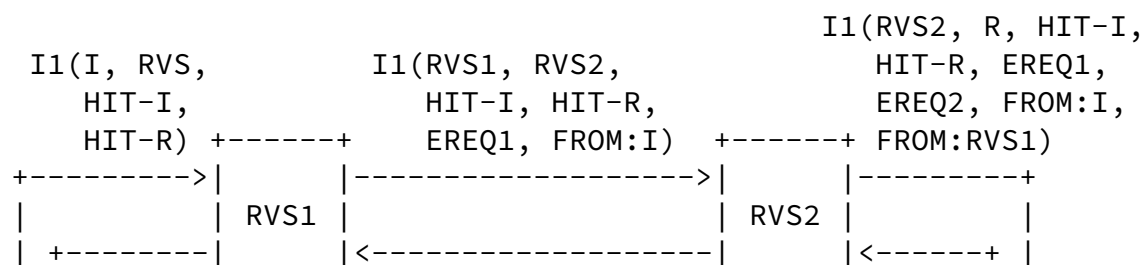
HIP Rendezvous Extensions

July 2004

"untrusted" Rendezvous Servers. Only the trusted Rendezvous Servers are aware of the IP addresses of the Responders. The untrusted servers know only the IP addresses of other (un)trusted Rendezvous Servers. Untrusted Rendezvous Servers are changed periodically, in order to lower the opportunity for flooding-type attacks on their IP addresses.

In the case of cascaded Rendezvous Servers, the parameters added to the HIP base exchange, like FROM, TO, VIA\_RVS, ECHO\_REQUEST/REPLY or RVA\_HMAC, MUST be "aggregated" or "clustered" on a per-type basis. This means that, when an RVS needs to add onto a HIP packet a parameter which is already present in it, this parameter MUST be added just after the existing parameter(s) of the same type. For instance, a FROM parameter MUST be added just after the existing FROM(s) parameter(s). The same applies to TO, VIA\_RVS, ECHO\_REQUEST/REPLY or RVA\_HMAC.

Another solution to cascaded Rendezvous Servers may be to encapsulate the original packet into a PAYLOAD and then piggyback it with additional parameters. This scheme has not been evaluated further.



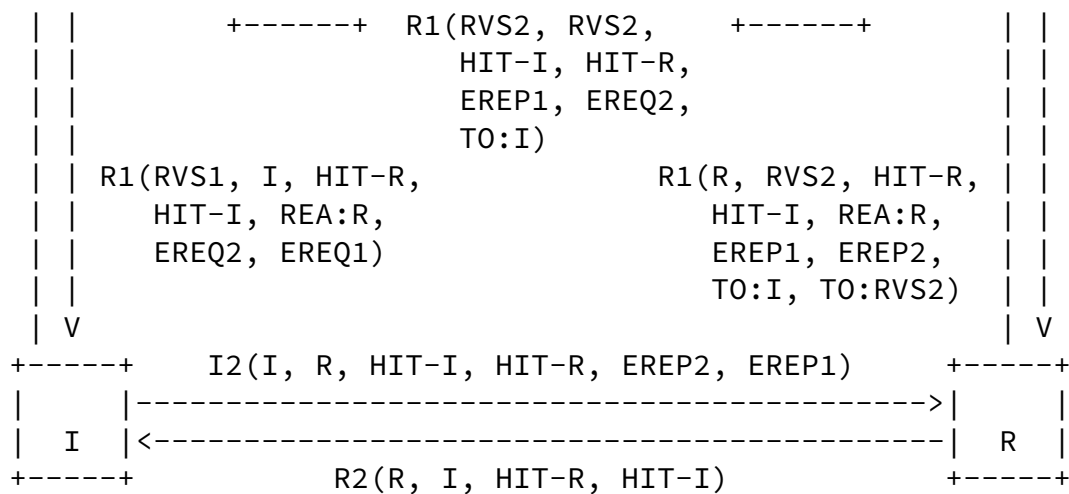


Figure 20: Two Cascaded Rendezvous Servers Relaying an I1-R1 Message Pair

## 7.5 Opportunistic Initiators

Because an opportunistic initiator uses the unspecified IPv6 address (i.e., ::0) as a placeholder for the Responder HIT in I1s it sends, an RVS cannot use this Responder HIT to demultiplex incoming "opportunistic" I1s. The only way to properly relay such an opportunistic I1 to the appropriate responder is to lease per-client (hence per-HIT) relay IP addresses. That way, the RVS MAY use the destination IP address as an indicator to determine the correct responder.

In order to avoid trivial spoofing attacks with R1s, a HIP node receiving an opportunistic I1 from a Rendezvous Server SHOULD reply with its R1 via the same Rendezvous Server.

## [7.6](#) Implication on the HIP integrity checks

The establishment of HIP associations through one or more Rendezvous Servers causes HIP packets flowing between the HIP nodes to be modified during transmission. Several kinds of modifications to both the IP and HIP headers are possible. The HIP protocol uses two kinds of packet integrity checks: hop-by-hop and end-to-end. The HIP checksum is a hop-by-hop check and SHOULD be verified and recomputed by each of the on-path HIP middleboxes (e.g., Rendezvous Servers). The HMAC and SIGNATURE are end-to-end checks and MUST be computed by the sender and verified by the receiver.

### [7.6.1](#) Checksum

The checksum field of a HIP header to be modified MUST be verified before applying the modification and recomputed accordingly after.

### [7.6.2](#) HMAC and SIGNATURE

The HMAC and SIGNATURE field of a HIP header MUST be computed and verified based on a "sender view" or "receiver view" of the HIP header. In particular, this implies that SIGNATURE and HMAC MUST NOT cover FROM and TO parameters added or removed by Rendezvous Servers and that the HIP pseudo-header used to compute and verify them MUST contain the IP addresses as seen by the remote HIP peer. In case of IP address concealment, this means that the IP address(es) of the Rendezvous Servers MUST be used in the pseudo-header in place of the IP address(es) of the end hosts.

### [7.6.3](#) Example

Here is an example showing how to compute the different integrity checks (end-to-end and hop-by-hop) when two Rendezvous Servers are

cascaded and when both peers conceals their IP addresses (packet flowing along the path I -> RVS1 -> RVS2 -> R)

End-to-end integrity checks: HMAC and SIGNATURE are computed with a pseudo-header containing (two times) RVS1 as place holder for source and destination IP addresses. The rationale being that the initiator is concealing its IP address behind that of RVS1. Therefore, R will verify the signature using RVS1 as the source IP address in the pseudo-header. Similarly, the responder is concealing its IP address behind that of RVS1, so I will verify the signature using RVS1 as a source IP address in the pseudo-header.

hop-by-hop integrity checks: Checksum is computed hop-by-hop; first with I and RVS1, then with RVS1 and RVS2, and finally with RVS2 and R.

## 8. Security Considerations

The security aspects of different HIP rendezvous mechanisms are currently being investigated. They will be discussed in a future revision of this document.

## 9. Acknowledgments

The following people have provided thoughtful and helpful discussions and/or suggestions that have improved this document: Marcus Brunner, Tom Henderson, Miika Komu, Mika Kousa, Pekka Nikander, Simon Schuetz, Tim Shepard, Kristian Slavov, Martin Stiernerling, and Juergen Quittek.

## 10. References

### 10.1 Normative References

- [1] Mockapetris, P., "Domain names - concepts and facilities", STD 13, [RFC 1034](#), November 1987.
- [2] Moskowitz, R., "Host Identity Protocol Architecture", [draft-moskowitz-hip-arch-05](#) (work in progress), October 2003.

- [3] Moskowitz, R., Nikander, P. and P. Jokela, "Host Identity Protocol", [draft-moskowitz-hip-09](#) (work in progress), February 2004.
- [4] Nikander, P., "End-Host Mobility and Multi-Homing with Host Identity Protocol", [draft-nikander-hip-mm-01](#) (work in progress), January 2004.

Eggert & Laganier Expires January 10, 2005 [Page 30]

---

Internet-Draft HIP Rendezvous Extensions July 2004

- [5] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [6] Vixie, P., Thomson, S., Rekhter, Y. and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)", [RFC 2136](#), April 1997.
- [7] Wellington, B., "Secure Domain Name System (DNS) Dynamic Update", [RFC 3007](#), November 2000.
- [8] Killalea, T., "Recommended Internet Service Provider Security Services and Procedures", [BCP 46](#), [RFC 3013](#), November 2000.
- [9] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", [BCP 38](#), [RFC 2827](#), May 2000.

## [10.2](#) Informative References

- [10] Saltzer, J., "On the Naming and Binding of Network Destinations", [RFC 1498](#), August 1993.



- [11] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", [RFC 2401](#), November 1998.
- [12] Klensin, J., "Role of the Domain Name System (DNS)", [RFC 3467](#), February 2003.
- [13] Nikander, P., "A Bound End-to-End Tunnel (BEET) mode for ESP", [draft-nikander-esp-beet-mode-00](#) (work in progress), October 2003.

#### Authors' Addresses

Lars Eggert  
NEC Network Laboratories  
Kurfuersten-Anlage 36  
Heidelberg 69115  
DE

Phone: +49 6221 90511 43  
Fax: +49 6221 90511 55  
EMail: [lars.eggert@netlab.nec.de](mailto:lars.eggert@netlab.nec.de)  
URI: <http://www.netlab.nec.de/>

Eggert & Laganier	Expires January 10, 2005	[Page 31]
-------------------	--------------------------	-----------

---

Internet-Draft	HIP Rendezvous Extensions	July 2004
----------------	---------------------------	-----------

Julien Laganier  
Sun Labs (Sun Microsystems) & LIP (CNRS/INRIA/ENSL/UCBL)  
180, Avenue de l'Europe  
Saint Ismier CEDEX 38334  
FR

Phone: +33 476 188 815

EMail: [ju@sun.com](mailto:ju@sun.com)  
URI: <http://research.sun.com/>

## [Appendix A](#). Document Revision History

Revision	Comments
00	Compared to <a href="#">draft-eggert-hip-rendezvous-00</a> : Minor fixes to figures and their descriptive text. Added RVS protocol specification. Removed sections related to communications between HIP and non-HIP nodes. Use boilerplate from <a href="#">RFC 3668</a> .

## Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at [ietf-ipr@ietf.org](mailto:ietf-ipr@ietf.org).

## Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

## Copyright Statement

Copyright (C) The Internet Society (2004). This document is subject

to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

#### Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.