HIP Working Group J. Laganier
Internet-Draft LIP / Sun Microsystems
Expires: August 19, 2005 L. Eggert
NEC
February 18, 2005

Host Identity Protocol (HIP) Rendezvous Extension
draft-ietf-hip-rvs-01

Status of this Memo

Copyright Notice

Abstract

This document discusses a Rendezvous extension for the Host Identity Protocol (HIP). The Rendezvous extension extend HIP and the HIP registration extension for initiating communication between HIP nodes via HIP Rendezvous Servers. Rendezvous Servers improve operation when HIP nodes are multi-homed or mobile.

Table of Contents

1. Introduction

The current Internet has a dual use of IP addresses. First, they are topological locators for network attachment points. Second, they act as names for the attached network interfaces. Saltzer [6] 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 [1] defines a new third namespace. The Host Identity namespace decouples the name and locator roles currently filled by 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. Because of this decoupling the HIP layer needs to map Host Identities into IP addresses.

Without HIP, a node needs to know its peer IP address to make an initial contact. The Host Identity Protocol architecture [1] introduces an additional piece of infrastructure, the Rendezvous Server (RVS), which serves as an initial contact point (rendezvous) for nodes trying to reach the RVS clients. A RVS offers to a peer it serves to relay to its IP address the first packet of a HIP exchange incoming at the RVS IP address and with the peer receiver HIT. A peer uses the HIP Registration Protocol [2] to register its HIT->IP address mapping with its RVS. Then an initiator and responder can have rendezvous together at the RVS IP address. The initiator would send a I1 packet to the RVS IP address, which would then relay the I1 to the responder IP address. Then, further communications would typically occurs directly without further assistance from the RVS.

After the Base Exchange, HIP nodes use Host Identities instead of IP addresses to name transport-layer endpoints. The HIP layer in the network stack internally translates Host Identities (HI) into network-layer IP addresses.

2. Terminology

This section defines terms used throughout the remainder of this specification.

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 [3].

Rendezvous Service : A HIP Service provided by a HIP Rendezvous Server to its Rendezvous Clients. The Rendezvous Server offers to relay some of the incoming HIP packets exchanged during a HIP exchange to the owner of their receiver HIT (i.e. the Rendezvous Client or one of its correspondent nodes).

Rendezvous Server (RVS): A HIP Registrar providing the Rendezvous Service.

Rendezvous Client (RVC): A HIP Requester which has registered for the Rendezvous Service at a Rendezvous Server.

Rendezvous Registration (RVR): A HIP Registration for the Rendezvous Service, established between a Rendezvous Server and a Rendezvous Client.

3. Overview of Rendezvous Server Operation

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.

```
+-----+                +-----+
|     |-------I1------>|     |
|  I  |<------R1-------|  R  |
|     |-------I2------>|     |
|     |<------R2-------|     |
+-----+                +-----+
```

Figure 1: HIP Base Exchange without Rendezvous Server

Figure 2 shows a simple HIP Base Exchange (without Rendezvous Server) in which the initiator initiates the exchange directly with the responder by sending an I1 packet to the responder IP address, as per the HIP base specification [4].

Proposed extensions for mobility and multi-homing [5] 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.

However, such a HIP node might also want to be still reachable by potential future correspondent peers unaware of its location change. The HIP architecture [1] introduces Rendezvous Servers at which a HIP

node register its current HIT and IP addresses. The RVS basically relays HIP packet incoming at this HIT to the node IP address. Thus, a peer publishing its RVS IP address instead of its own is reachable by means of rendezvous at its RVS IP address.

```
            +-----+
   +--I1--->| RVS |---I1--+
   |        +-----+       |
   |                      v
+-----+                +-----+
|     |<------R1-------|     |
|  I  |-------I2------>|  R  |
|     |<------R2-------|     |
+-----+                +-----+
```

Figure 2: HIP Base Exchange with Rendezvous Server

Figure 2 shows a HIP Base Exchange involving a Rendezvous Server RVS. It is assumed that HIP node R precedently used the HIP registration protocol [2] to register with the RVS its HIT and IP address. When the initiator I tries to establish contact with the responder, it does not need to know the current IP address of R. Instead, I is aware of the RVS IP address of R, at which it sends an I1 packet. The RVS, noticing that the receiver HIT is not its own, but the HIT of a HIP node registered for the rendezvous Service, would relay the I1 to the responder IP address. Typically the responder would then complete the exchange without further assistance from the RVS by sending an R1 directly to the initiator IP address.

3.1 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
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	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 REG_REQUEST parameter is present in the HIP header
RRES	A REG_RESPONSE parameter is present in the HIP header
RVR:t1,t2	A RVR_TYPE parameter with Type value t1 and t2 is present in the HIP header.

3.2 Rendezvous Client Registering with a Rendezvous Server

Before the Rendezvous Server starts to relay HIP packets to their receiver HIT owner (i.e. a Rendezvous Client or one of its correspondent node), the Rendezvous Client needs to register with its Server for the Rendezvous Service, as shown in the following schema:

```
   +-----+                           +-----+
   |     |            I1             |     |
   |     |--------------------------->|     |
   |     |<---------------------------|     |
   | RVC |    R1(REG_INFO,RVR:1,2)    | RVS |
   |     |      I2(REG_REQ,RVR:2)     |     |
   |     |--------------------------->|     |
   |     |<---------------------------|     |
   |     |      R2(REG_RES,RVR:2)     |     |
   +-----+                           +-----+
```

3.3 Establishing HIP Associations via Rendezvous Servers

3.3.1 Encapsulating I1 into a Tunnel

If a HIP node and one of its Rendezvous Servers have a Rendezvous Registration of type TUNNEL_I1, the Rendezvous Server tunnels up to this node I1s incoming to this node's HIT using the appropriate encapsulation technique. The technique to be used is determined based on the kind of association established between the RVS and its client, and differs only by the type of header prepended to the HIP packet (e.g. HIP, ESP or UDP).

```
                                          ENCAP(RVS, R)[ I1(I, RVS,    ]
                                                       [ HIT-I, HIT-R, ]
    I1(I, RVS, HIT-I, HIT-R) +---------+               [ FROM:I)       ]
    +----------------------->|         |--------------------+
    |                        |   RVS   |                    |
    |                        |         |                    |
    |                        +---------+                    |
    |                                                       V
   +-----+    R1(R, I, HIT-R, HIT-I, REA:R, VIA:RVS)    +-----+
   |     |<---------------------------------------------|     |
   |     |                                              |     |
   |  I  |            I2(I, R, HIT-I, HIT-R)            |  R  |
   |     |--------------------------------------------->|     |
   |     |<---------------------------------------------|     |
   +-----+             R2(R, I, HIT-R, HIT-I)           +-----+
```

Figure 5: I1_TUNNEL: Rendezvous Server Encapsulating I1 into a Tunnel

3.3.2 Rewriting I1 IP Header

If a HIP node and one of its Rendezvous Servers have a Rendezvous Registration of type REWRITE_I1, the Rendezvous Server relays up to this node I1s incoming to this node's HIT by merely rewrite the IP header. The destination IP address of the I1 is replaced by the IP address of the receiver HIT owner (i.e. the Rendezvous Client).

However, because of egress filtering, a HIP Rendezvous Server might also 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. Hence, such a node MUST append I1 packets with a FROM parameter containing the original source IP address of the packet. This FROM parameter MUST be integrity protected by a RVR_HMAC keyed with the corresponding rendezvous registration integrity key [2].

```
                                             I1(I, RVS, HIT-I,
       I1(I, RVS, HIT-I, HIT-R) +---------+     HIT-R, FROM:I, VIA:RVS)
       +----------------------->|         |--------------------+
       |                        |   RVS   |                    |
       |                        |         |                    |
       |                        +---------+                    |
       |                                                       V
      +-----+     R1(R, I, HIT-R, HIT-I, REA:R, VIA:RVS)   +-----+
      |     |<---------------------------------------------|     |
      |     |                                              |     |
      |  I  |            I2(I, R, HIT-I, HIT-R)            |  R  |
      |     |--------------------------------------------->|     |
      |     |<---------------------------------------------|     |
      +-----+             R2(R, I, HIT-R, HIT-I)           +-----+
```

Figure 6: I1_REWRITE: Rendezvous Server Rewriting I1 Source and Destination IP Addresses

3.3.3 Bidirectional Forwarding of HIP packets

In some cases it is useful to have a RVS which relay further HIP packets in a bidirectional manner, i.e. from the initiator to the responder but also from the responder to the initiator. These further packets would typically be either an R1 or an UPDATE. A RVS behaves accordingly when the Rendezvous Registration Type is BIDIRECTIONAL.

However, because such packets are larger than I1 (they contain a signature), their relaying 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 SHOULD make use of ECHO_REQUEST and ECHO_RESPONSE parameters.

Each time a packet is being relayed and will possibly trigger an answer, the RVS MUST augment it with an ECHO_REQUEST parameter containing a chunk of opaque data. The receiver of such a packet MUST 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 MUST be removed from the packet before relaying.

```
       I1(I, RVS,                    I1(RVS, R, HIT-I, HIT-0
          HIT-I, HIT-0)   +---------+   FROM:I, EREQ)
    +--------------------->|         |----------------------+
    |+--------------------|         |<--------------------+|
    || R1(RVS, I, HIT-R,  |   RVS   | R1(R, RVS, HIT-R,   ||
    ||    HIT-I, REA:R,   |         |    HIT-I, REA:R,    ||
    ||    VIA:RVS)        |         |    TO:I, EREP)      ||
    ||                    |         |                     ||
    ||                    +---------+                     ||
    |V                                                    |V
   +-----+             I2(R, I, HIT-R, HIT-I)          +-----+
   |     |-------------------------------------------->|     |
   |  I  |<--------------------------------------------|  R  |
   |     |             R2(I, R, HIT-I, HIT-R)          |     |
   +-----+                                             +-----+
```

Figure 7: BIDIRECTIONAL: Responder replying via the RVS to Initiator

3.3.4 Implication on the HIP integrity checks

The establishment of HIP associations via 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 middle-boxes (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.

3.3.4.1 Checksum

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

3.3.4.2 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 by the RVS, this means that the IP address of this RVS MUST be used in the pseudo-header in place of the IP address of the end host it conceals.

3.3.4.3 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 conceals the Responder IP address (packet flowing along the path I -> RVS1 -> RVS2 -> R)

End-to-end integrity checks: HMAC and SIGNATURE are computed with a pseudo-header containing RVS1 as a place holder for the destination IP address, the rationale being that RVS1 is concealing the Responder IP address. Therefore, R will verify the signature using RVS1 as the destination 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.

4. RVS Extensions Definition

The following sections describe extensions to:

- The HIP registration protocol [2], allowing a HIP node to register with its Rendezvous Server for the Rendezvous Service and maintain the RVS aware of its current location.

- The HIP protocol [4] itself, allowing to establish an HIP association via one or more HIP Rendezvous Server(s).

4.1 Usage and Processing of Existing Parameters

4.1.1 ECHO_REQUEST and ECHO_REPLY Parameters

A FROM parameter MAY be augmented by including an ECHO_REQUEST parameter to the carrying packet. The contents of the ECHO_REQUEST MUST then be echoed back in ECHO_RESPONSE.

A TO parameter MUST 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 MUST be augmented by the RVS with an ECHO_REQUEST parameter.

A possible packet answered via the RVS, thus requiring relaying facility, MUST 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.

4.1.2 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 [5].

4.2 New Registration Type

This specification defines an additional Registration Type to use within the HIP Registration protocol [2] while registering with a Rendezvous Server for the Rendezvous Service.

Number	Registration Type
1	RENDEZVOUS

4.3 New Parameter Formats and Processing

4.3.1 RVR_TYPE Parameter

The RVR_RYPE is an OPTIONAL parameter allowing a Rendezvous Server and its Requesters to negotiate the type of Rendezvous Service provided by a Rendezvous Registration.

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |             Type              |             Length            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  RVR Type #1  |  RVR Type #2  |                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+            Padding            |
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type   [TBD by IANA {110)]
Length   8
RVR Type An 8 bit number indicating the specific type of the Rendezvous Server/Service.

Number	RVR Type	Definition
1	TUNNEL_I1	Tunneling I1 - Section 3.3.1
2	REWRITE_I1	Rewriting I1 - Section 3.3.2
3	BIDIRECTIONAL	Rewriting I1 and followers - Section 3.3.3
3-200	Reserved by IANA	
201-255	Reserved by IANA for private use	

A Requester of a Rendezvous Registration SHOULD include the RVR_RYPE parameter along with any REG_REQUEST for the Rendezvous Service. This parameter specifies the desired RVS Type (i.e. TUNNEL_I1, REWRITE_I1 or BIDIRECTIONAL). It SHOULD NOT include the parameter unless there is a REG_REQUEST parameter included along.

A Rendezvous Server SHOULD include a RVR_TYPE parameter along with any REG_INFO announcing support for the Rendezvous Service. This parameter SHOULD specify all the RVR Types supported by the Rendezvous Server, in preference order.

A Rendezvous Server MUST include a RVR_RYPE parameter along with any

REG_RESPONSE establishing a Rendezvous Registration. This parameter MUST specify a single RVR Type for the established Registration.

A Rendezvous Server SHOULD NOT include the parameter unless there is a REG_INFO or REG_REQUEST parameter included along.

4.3.2 RVR_HMAC Parameter

The RVR_HMAC is an OPTIONAL parameter whose only difference with the HMAC parameter defined in [4] is the Type code, making it situated after the TO and FROM parameters (as opposed to HMAC):

Type [TBD by IANA {65320)]
Length 20
HMAC 160 low order bits of a HMAC keyed with the appropriate HIP integrity keys (HIP_lg or HIP_gl) of the corresponding HIP Association. This HMAC is computed over the HIP packet excluding RVR_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 Rendezvous Client and its RVS to verify the integrity of packets flowing between them, both use an RVR_HMAC parameter keyed with a HMAC of HIP_lg and HIP_gl integrity keys. One RVR_HMAC SHOULD be present on every packets flowing between a client and a server and MUST be present when FROM and TO parameters are processed.

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

4.3.3 FROM Parameter

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |             Type              |             Length            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    |                             Address                           |
    |                                                               |
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
   Type          [ TBD by IANA {65100 under signature, 65300 after) ]
   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 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 an RVR_HMAC protecting the packet integrity that the Rendezvous Client will use to validate this packet.

If the type of the RVR allows the Rendezvous Client to answer to a relayed packet via the RVS, an ECHO_REQUEST MUST be included along with the FROM parameter. It contains a chunk of opaque data allowing to validate TO parameters included in a subsequent answer. These TO parameters MUST be protected by an ECHO_RESPONSE containing the same opaque data.

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 TO parameter to be piggy-backed 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 TO parameter containing the IP address included in the FROM. These replies will be sent via the RVS, which MUST remove the outer TO parameter from the packet and replace its destination address with the address contained in the TO parameter before relaying it.

4.3.4 TO Parameter

```
      0                   1                   2                   3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |             Type              |             Length            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                                                               |
     |                            Address                            |
     |                                                               |
     |                                                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type   [TBD by IANA {65102 under signature, 65302 after)]
Length   16
Address   An IPv6 address or an IPv4-in-IPv6 format IPv4 address

A HIP node MAY add one or more TO parameter containing the final destination IP address of a HIP packet whose destination IP address needs to be rewritten by an RVS. This is essentially equivalent to loose source-routing. If one or more TO parameters are already present, the new TO parameter MUST be appended after the existing ones. Each time a node inserts a TO 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 TO parameter.

o A valid RVR_HMAC, protecting the packet integrity.

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

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

4.3.5 VIA_RVS Parameter

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |             Type              |             Length            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    |                            Address                            |
    |                                                               |
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    .                               .                               .
    .                               .                               .
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    |                            Address                            |
    |                                                               |
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
Type         [ TBD by IANA {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 RVSs for debugging purposes.

Similarly, a RVS relaying an I1 to the Responder or an R1 to the Initiator MAY include in these packets its IP address for debugging as well.

When the IP address of a RVS need to be included in a packet, by either an end-node or the RVS itself, one of these two methods is used:

- o Add RVS 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 Append 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 using the VIA_RVS parameter is to allow operators to diagnose possible issues encountered while establishing a HIP association via a RVS.

5. Security Considerations

The security aspects of different HIP rendezvous mechanisms are currently being investigated. This section describes the known threats introduced by these HIP extensions, and implications on the overall security of HIP and IP. In particular, the following tries to show that the extensions described in this document do not introduce additional threats in the Internet infrastructure.

It is difficult to encompass the whole scope of threats introduced by Rendezvous Servers because their presence have implications both at the IP and HIP layer. In particular, the extensions hereby described might allow for redirection, amplification and reflection attacks at the IP layer, as well as attacks on the HIP layer itself, for example Man-in-the-Middle attacks against the cryptographic core-protocol SIGMA used by HIP.

If an Initiator has an a priori knowledge of the Responder's HI when it first contacts it via the RVS, it has a means to verify the signatures in the HIP exchange, thus conforming to the SIGMA protocol which is resilient to Man-in-the-Middle attacks.

If an Initiator has not an a priori knowledge of the Responder's HI (so called Opportunistic Initiators), it is almost impossible to defend the HIP exchange against MitM attacks (cannot authenticate public keys exchanged). The only solution is to mitigate hijacking threats on the HIP state by requiring an R1 answering an Opportunistic I1 to come from the IP address where the I1 was initially sent. That way we retain a level of security which is equivalent to what exists today in the Internet: By sending an IP packet to an IP address, and receiving an answered IP packet from this same IP address, I know that the routing fabric trusts my correspondent to be represented by this IP address. While it is true that such security is weak, it is better than none, and avoids to introduce additional threats at the IP layer.

6. IANA Considerations

This document updates the IANA Registry for HIP Parameters Types by assigning new HIP Parameter Types values for the new HIP Parameters defined in Section 4.3:

- RVR_TYPE (defined in Section 4.3.1)

- RVR_HMAC (defined in Section 4.3.2)

- FROM (defined in Section 4.3.3)

- TO (defined in Section 4.3.4)

- VIA_RVS (defined in Section 4.3.5)

IANA needs to open a new registry specific to the HIP Rendezvous Extensions, for the Rendezvous Registration (RVR) Types values defined in Section 4.3.1:

Type number	RVR Type
0	Reserved by IANA
1	TUNNEL_I1
2	REWRITE_I1
3	BIDIRECTIONAL
3-200	Reserved by IANA
201-255	Reserved by IANA for private use

Adding new reservations requires IETF consensus RFC2434 [7].

7. 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 Stiemerling, and Juergen Quittek.

Part of this work is a product of the Ambient Networks project, partially supported by the European Commission under its Sixth Framework Programme. It is provided "as is" and without any express or implied warranties, including, without limitation, the implied warranties of fitness for a particular purpose. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Ambient Networks project or the European Commission.

8. References

8.1 Normative References

[1] Moskowitz, R. and P. Nikander, "Host Identity Protocol Architecture", draft-ietf-hip-arch-00 (work in progress), October 2004.

[2] Laganier, J., Koponen, T. and L. Eggert, "Host Identity Protocol (HIP) Registration Extensions", draft-koponen-hip-registration-00 (work in progress), January 2005.

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

[4] Moskowitz, R., Nikander, P. and P. Jokela, "Host Identity Protocol", draft-ietf-hip-base-01 (work in progress), October 2004.

[5] Nikander, P., "End-Host Mobility and Multi-Homing with Host Identity Protocol", draft-ietf-hip-mm-00 (work in progress), October 2004.

8.2 Informative References

[6] Saltzer, J., "On the Naming and Binding of Network Destinations", RFC 1498, August 1993.

[7] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

[8] Nikander, P. and J. Laganier, "Host Identity Protocol (HIP) Domain Name System (DNS) Extensions", draft-ietf-hip-rvs-00 (work in progress), October 2004.

[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] Killalea, T., "Recommended Internet Service Provider Security Services and Procedures", BCP 46, RFC 3013, November 2000.

Authors' Addresses

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
URI: http://research.sun.com/

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
URI: http://www.netlab.nec.de/

Appendix A. Document Revision History

Revision	Comments
01	Splitted out the registration sub-protocol. Simplify typology of relaying techniques (keep only TUNNEL, REWRITE, BIDIRECTIONAL). Rewrote IANA Considerations.
00	Initial version as a HIP WG item.

Intellectual Property Statement

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

Acknowledgment

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