Network Working Group
Internet-Draft
Expires: April 17, 2005

P. Nikander
J. Arkko
Ericsson Research Nomadic Lab
T. Henderson
The Boeing Company

October 17, 2004

End-Host Mobility and Multi-Homing with Host Identity Protocol
draft-ietf-hip-mm-00

Status of this Memo

Copyright Notice

Abstract

This document specifies basic end-host mobility and multi-homing mechanisms for the Host Identity Protocol.

Table of Contents

1. Introduction

This document specifies an extension to the Host Identity Protocol [3] (HIP). The extension provides a means for hosts to keep their communications on-going while having multiple IP addresses, either at the same time or one after another. That is, the extension provides basic end-to-end support for multi-homing, mobility, and simultaneous multi-homing and mobility. Additionally, the extension allows communications to continue even when multi-homing or mobility causes a change of the IP version that is available in the network; that is, if one of the communicating hosts has both IPv4 and IPv6 connectivity, either directly or through a proxy, the other host can alternate between IPv4 and IPv6, without needing to tear down and re-establish upper layer protocol connections or associations. In other words, the way upper layer protocols need to react to cross-IP-version handovers does not differ from the way they need to react to intra-IP-version handovers.

This document does not specify any rendezvous or proxy services. Those are subject to other specifications. Hence, this document alone does not necessarily allow two mobile hosts to communicate, unless they have other means for initial rendezvous and for solving the simultaneous movement problem.

The Host Identity Protocol [3] (HIP) defines a mechanism that decouples the transport layer (TCP, UDP, etc) from the internetworking layer (IPv4 and IPv6), and introduces a new Host Identity namespace. When a host uses HIP, the transport layer sockets and IPsec Security Associations are not bound to IP addresses but to Host Identifiers. This document specifies how the mapping from Host Identifiers to IP addresses can be extended from a static one-to-one mapping into a dynamic one-to-many mapping, thereby enabling end-host mobility and multi-homing.

In practice, the HIP base exchange [3] creates a pair of IPsec Security Associations (SA) between a pair of HIP enabled hosts. These SAs are not bound to IP addresses, but to the Host Identifiers (public keys) used to create them. However, the hosts must also know at least one IP address where their peers are reachable. Initially these IP addresses are the ones used during the HIP base exchange.

Since the SAs are not bound to IP addresses, the host is able to receive packets that are protected using a HIP created ESP SA from any address. Thus, a host can change its IP address and continue to send packets to its peers. However, unless the host is sufficiently trusted, the peers are not able to reply before they can reliably and securely update the set of addresses that they associate with the sending host. Furthermore, mobility may change the path

characteristics in such a manner that reordering occurs and packets fall outside the ESP anti-replay window.

This document specifies a mechanism that allows a HIP host to update the set of addresses that its peers associate with it. The address update is implemented with new HIP parameter types. Due to the danger of flooding attacks (see [4]), the peers must always check the reachability of the host at a new address, unless sufficient level of trust exists between the hosts.

The reachability check is implemented by the challenger sending some piece of unguessable information to the new address, and waiting for some acknowledgment from the responder that indicates reception of the information at the new address. This may include exchange of a nonce, or generation of a new SPI and observing data arriving on the new SPI.

There are a number of situations where the simple end-to-end readdressing functionality is not sufficient. These include the initial reachability of a mobile host, location privacy, end-host and site multi-homing with legacy hosts, and NAT traversal. In these situations there is a need for some helper functionality in the network. This document does not address those needs.

Finally, making underlying IP mobility transparent to the transport layer has implications on the proper response of transport congestion control, path MTU selection, and QoS. Transport-layer mobility triggers, and the proper transport response to a HIP mobility or multi-homing address change, are outside the scope of this document.

2. Conventions used in this document

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 RFC2119 [1].

3. Terminology

Preferred address An address on which a host prefers to receive data. With respect to a given peer, a host always has one active preferred address. By default, the source address used in the HIP base exchange is the preferred address.

New preferred address A new preferred address sent by a host to its peers. The reachability of the new preferred address often needs to be verified before it can be taken into use. Consequently, there may simultaneously be an active preferred address, being used, and a new preferred address, the reachability of which is being verified.

4. Overview of HIP basic mobility and multi-homing functionality

HIP mobility and multi-homing is fundamentally based on the HIP architecture [4], where the transport and internetworking layers are decoupled from each other by an interposed host identity protocol layer. In the HIP architecture, the transport layer sockets are bound to the Host Identifiers (through HIT or LSI in the case of legacy APIs), and the Host Identifiers are translated to the actual IP address.

The HIP base protocol specification [3] defines how two hosts exchange their Host Identifiers and establish a pair of ESP Security Associations (SA). The ESP SAs are then used to carry the actual payload data between the two hosts, by wrapping TCP, UDP, and other upper layer packets into transport mode ESP payloads. The IP header uses the actual IP addresses in the network.

The base specification does not contain any mechanisms for changing the IP addresses that were used during the base HIP exchange. Hence, in order to remain connected, any systems that implement only the base specification and nothing else must retain the ability to receive packets at their primary IP address; that is, those systems cannot change the IP address on which they are using to receive packets without causing loss of connectivity until a base exchange is performed from the new address.

4.1 Informing the peer about multiple or changed address(es)

This document specifies a new HIP protocol parameter, the REA parameter (see Section 6.1), that allows the hosts to exchange information about their IP address(es), and any changes in their address(es). The logical structure created with REA parameters has three levels: hosts, IPsec Security Associations (SAs) indexed by Security Parameter Indices (SPIs), and addresses.

The relation between these entities for an association negotiated as defined in the base specification [3] is illustrated in Figure 1.

```
           -<- SPI1a --                         -- SPI2a ->-
   host1 <              > addr1a <---> addr2a <              > host2
           ->- SPI2a --                         -- SPI1a -<-
```

Figure 1: Relation between hosts, SPIs, and addresses (base specification)

In Figure 1, host1 and host2 negotiate two unidirectional IPsec SAs, and each host selects the SPI value for its inbound SA. The addresses addr1a and addr2a are the source addresses that each host

uses in the base HIP exchange. These are the "preferred" (and only) addresses conveyed to the peer for each SA; even though packets sent to any of the hosts' interfaces can arrive on an inbound SPI, when a host sends packets to the peer on an outbound SPI, it knows of a single destination address associated with that outbound SPI (for host1, it sends a packet on SPI2a to addr2a to reach host2), unless other mechanisms exist to learn of new addresses.

In general, the bindings that exist in an implementation corresponding to this draft can be depicted as shown in Figure 2. In this figure, a host can have multiple inbound SPIs (and, not shown, multiple outbound SPIs) between itself and another host. Furthermore, each SPI may have multiple addresses associated with it. These addresses bound to an SPI are not used as IPsec selectors. Rather, the addresses are those addresses that are provided to the peer host, as hints for which addresses to use to reach the host on that SPI. The REA parameter is used to change the set of addresses that a peer associates with a particular SPI.

```
                         address11
                       /
                SPI1   - address12
              /
             /           address21
        host -- SPI2   <
             \           address22
              \
                SPI3   - address31
                       \
                         address32
```

Figure 2: Relation between hosts, SPIs, and addresses (general case)

A host may establish any number of security associations (or SPIs) with a peer. The main purpose of having multiple SPIs is to group the addresses into collections that are likely to experience fate sharing. For example, if the host needs to change its addresses on SPI2, it is likely that both address21 and address22 will simultaneously become obsolete. In a typical case, such SPIs may correspond with physical interfaces; see below. Note, however, that especially in the case of site multi-homing, one of the addresses may become unreachable while the other one still works. In the typical case, however, this does not require the host to inform its peers about the situation, since even the non-working address still logically exists.

A basic property of HIP SAs is that the inbound IP address is not used as a selector for the SA. Therefore, in Figure 2, it may seem

unnecessary for address31, for example, to be associated only with SPI3-- in practice, a packet may arrive to SPI1 via destination address address31 as well. However, the use of different source and destination addresses typically leads to different paths, with different latencies in the network, and if packets were to arrive via an arbitrary destination IP address (or path) for a given SPI, the reordering due to different latencies may cause some packets to fall outside of the IPsec ESP anti-replay window. For this reason, HIP provides a mechanism to affiliate destination addresses with inbound SPIs, if there is a concern that replay windows might be violated otherwise. In this sense, we can say that a given inbound SPI has an "affinity" for certain inbound IP addresses, and this affinity is communicated to the peer host. Each physical interface SHOULD have a separate SA, unless the ESP reordering window is loose.

Moreover, even if the destination addresses used for a particular SPI are held constant, the use of different source addresses may also cause packets to fall outside of the ESP replay window, since the path traversed is often affected by the source address or interface used. A host has no way to influence the source address on which a peer uses to send its packets on a given SPI. Hosts SHOULD consistently use the same source address when sending to a particular destination IP address and SPI. For this reason, a host may find it useful to change its SPI or at least reset its ESP replay window when the peer host readdresses.

An address may appear on more than one SPI. This creates no ambiguity since the receiver will ignore the IP addresses as IPsec selectors anyway.

A single REA parameter contains data only about one SPI. To simultaneously signal changes on several SPIs, it is necessary to send several REA parameters. The packet structure supports this.

If the REA parameter is sent in an UPDATE packet, then the receiver will respond with an UPDATE acknowledgment. If the REA parameter is sent in a NOTIFY, I2, or R2 packet, then the recipient may consider the REA as informational, and act only when it needs to activate a new address. The use of REA in a NOTIFY message may not be compatible with middleboxes.

4.2 Address verification

When a HIP host receives a set of IP addresses from another HIP host in a REA, it does not necessarily know whether the other host is actually reachable at the claimed addresses. In fact, a malicious peer host may be intentionally giving a bogus addresses in order to cause a packet flood towards the given address [7]. Thus, before the

HIP host can actually use a new address, it must first check that the peer is reachable at the new address.

A second benefit of performing an address check is to allow any possible middleboxes in the network along the new path to obtain the peer host's inbound SPI.

A simple technique to verify addresses is to send an UPDATE to the host at the new address. The UPDATE packet SHOULD include a nonce, unguessable by anyone not on the path to the new address, that forces the host to reply in a manner that confirms reception of the nonce. One direct way to perform this is to include an ECHO_REQUEST parameter with some piece of unguessable information such as a random number. If the host is sending a NES parameter, the ECHO_REQUEST MAY contain the new SPI, for example. If the peer host is rekeying by sending an UPDATE with NES to the new address, the arrival of data on the new SPI can also be used to verify the address.

If middlebox traversal is possible along the path, and the peer host is not rekeying, the peer host SHOULD include a SPI parameter as part of its UPDATE, with the SPI corresponding to its active inbound SPI. It is not specified how a host knows whether or not middleboxes might lie on its path, so a conservative assumption may be to always include the SPI parameter.

In certain networking scenarios, hosts may be trusted enough to bypass performing address verification. In such a case, the host MAY bypass the address verification step and put the addresses into immediate service. Note that this may not be compatible with middlebox traversal.

4.3 Preferred address

When a host has multiple addresses and SPIs, the peer host must decide upon which to use as a destination address. It may be that a host would prefer to receive data on a particular inbound interface. HIP allows a particular address to be designated as a preferred address, and communicated to the peer.

In general, when multiple addresses are used for a session, there is the question of using multiple addresses for failover only or for load-balancing. Due to the implications of load-balancing on the transport layer that still need to be worked out, this draft assumes that multiple addresses are used primarily for failover. An implementation may use ICMP interactions, reachability checks, or other means to detect the failure of an address.

4.4 Address data structure and status

In a typical implementation, each outgoing address is represented as a piece of state that contains the following data:

the actual bit pattern representing the IPv4 or IPv6 address,
lifetime (seconds),
status (UNVERIFIED, ACTIVE, DEPRECATED).

The status is used to track the reachability of the address:

UNVERIFIED indicates that the reachability of the address has not been verified yet,

ACTIVE indicates that the reachability of the address has been verified and the address has not been deprecated,

DEPRECATED indicates that the address lifetime has expired

The following state changes are allowed:

UNVERIFIED to ACTIVE The reachability procedure completes successfully.

UNVERIFIED to DEPRECATED The address lifetime expires while it is UNVERIFIED.

ACTIVE to DEPRECATED The address lifetime expires while it is ACTIVE.

ACTIVE to UNVERIFIED There has been no traffic on the address for some time, and the local policy mandates that the address reachability must be verified again before starting to use it again.

DEPRECATED to UNVERIFIED The host receives a new lifetime for the address.

If a host is verifying reachability with another host, a DEPRECATED address MUST NOT be changed to ACTIVE without first verifying its reachability. If reachability is not being verified, then the UNVERIFIED state is a transient state that transitions immediately to ACTIVE.

5. Protocol overview

In this section we briefly introduce a number of usage scenarios where the HIP mobility and multi-homing facility is useful. To understand these usage scenarios, the reader should be at least minimally familiar with the HIP protocol specification [3]. However, for the (relatively) uninitiated reader it is most important to keep in mind that in HIP the actual payload traffic is protected with ESP, and that the ESP SPI acts as an index to the right host-to-host context.

Each of the scenarios below assumes that the HIP base exchange has completed, and the hosts each have a single outbound SA to the peer host. Associated with this outbound SA is a single destination address of the peer host-- the source address used by the peer during the base exchange.

The readdressing protocol is an asymmetric protocol where one host, called the mobile host, informs another host, called the peer host, about changes of IP addresses on affected SPIs. The readdressing exchange is designed to be piggybacked on a number of existing HIP exchanges. The main packets on which the REA parameters are expected to be carried on are UPDATE packets. However, some implementations may want to experiment with sending REA parameters also on other packets, such as R1, I2, and NOTIFY.

5.1 Mobility with single SA pair

A mobile host must sometimes change an IP address bound to an interface. The change of an IP address might be needed due to a change in the advertised IPv6 prefixes on the link, a reconnected PPP link, a new DHCP lease, or an actual movement to another subnet. In order to maintain its communication context, the host must inform its peers about the new IP address. This first example considers the case in which the mobile host has only one interface, IP address, and a single pair of SAs (one inbound, one outbound).

1. The mobile host is disconnected from the peer host for a brief period of time while it switches from one IP address to another. Upon obtaining a new IP address, the mobile host sends a REA parameter to the peer host in an UPDATE message. The REA indicates the following: the new IP address, the SPI associated with the new IP address, the address lifetime, and whether the new address is a preferred address. The mobile host may optionally send a NES to create a new inbound SA, in which case it transitions to state REKEYING. In this case, the REA contains the new SPI to use. Otherwise, the existing SPI is identified in the REA parameter, and the host waits for its UPDATE to be

acknowledged.

2. Depending on whether the mobile host initiated a rekey, and on whether the peer host itself wants to rekey or verify the mobile host's new address, a number of responses are possible. Figure 3 illustrates an exchange for which neither side initiates a rekeying, but for which the peer host performs an address check. If the peer host chooses not to perform an address check, the UPDATE that it sends will only acknowledge the mobile host's update but will not solicit a response from the mobile host. If the mobile host is rekeying, the peer will also rekey, as shown in Figure 4. If the mobile host did not decide to rekey but the peer desires to do so, then it initiates a rekey as illustrated in Figure 5. The UPDATE messages sent from the peer back to the mobile are sent to the newly advertised address.
3. If the peer host is verifying the new address, the address is marked as UNVERIFIED in the interim. Once it has successfully received a reply to its UPDATE challenge, or optionally, data on the new SA, it marks the new address as ACTIVE and removes the old address.

```
     Mobile Host                         Peer Host

                UPDATE(REA, SEQ)
        ----------------------------------->
                UPDATE(SPI, SEQ, ACK, ECHO_REQUEST)
        <-----------------------------------
                UPDATE(ACK, ECHO_RESPONSE)
        ----------------------------------->
```

Figure 3: Readdress without rekeying, but with address check

```
     Mobile Host                         Peer Host

                UPDATE(REA, NES, SEQ, [DIFFIE_HELLMAN])
        ----------------------------------->
                UPDATE(NES, SEQ, ACK, [DIFFIE_HELLMAN,] ECHO_REQUEST)
        <-----------------------------------
                UPDATE(ACK, ECHO_RESPONSE)
        ----------------------------------->
```

Figure 4: Readdress with mobile-initiated rekey

```
     Mobile Host                            Peer Host

                UPDATE(REA, SEQ)
        ----------------------------------->
                UPDATE(NES, SEQ, ACK, [DIFFIE_HELLMAN], ECHO_REQUEST)
        <-----------------------------------
                UPDATE(NES, SEQ, ACK, [DIFFIE_HELLMAN,] ECHO_RESPONSE)
        ----------------------------------->
                UPDATE(ACK)
        <-----------------------------------
```

Figure 5: Readdress with peer-initiated rekey

Hosts that use link-local addresses as source addresses in their HIP handshakes may not be reachable by a mobile peer. Such hosts SHOULD provide a globally routable address either in the initial handshake or via the REA parameter.

5.2 Host multihoming

A (mobile or stationary) host may sometimes have more than one interface. The host may notify the peer host of the additional interface(s) by using the REA parameter. To avoid problems with the ESP reordering window, a host SHOULD use a different SA for each interface used to receive packets from the peer host.

When more than one address is provided to the peer host, the host SHOULD indicate which address is preferred. By default, the addresses used in the base exchange are preferred until indicated otherwise.

Although the protocol may allow for configurations in which there is an asymmetric number of SAs between the hosts (e.g., one host has two interfaces and two inbound SAs, while the peer has one interface and one inbound SA), it is RECOMMENDED that inbound and outbound SAs be created pairwise between hosts. When a NES arrives to rekey a particular outbound SA, the corresponding inbound SA should be also rekeyed at that time. Although asymmetric SA configurations might be experimented with, their usage may constrain interoperability at this time. However, it is recommended that implementations attempt to support peers that prefer to use non-paired SAs. It is expected that this section and behavior will be modified in future revisions of this protocol, once the issue and its implications are better understood.

To add both an additional interface and SA, the host sends a REA with a NES. The host uses the same (new) SPI value in the REA and both the "Old SPI" and "New SPI" values in the NES-- this indicates to the

peer that the SPI is not replacing an existing SPI. The multihomed host transitions to state REKEYING, waiting for a NES from the peer and an ACK of its own UPDATE. As in the mobility case, the peer host can perform an address check while it is rekeying. Figure 6 illustrates the basic packet exchange.

```
     Multi-homed Host                    Peer Host

                UPDATE(REA, NES, SEQ, [DIFFIE_HELLMAN])
        ----------------------------------->
                UPDATE(NES, SEQ, ACK, [DIFFIE_HELLMAN,] ECHO_REQUEST)
        <-----------------------------------
                UPDATE(ACK, ECHO_RESPONSE)
        ----------------------------------->
```

Figure 6: Basic multihoming scenario

For the case in which multiple addresses are advertised in a REA, the peer does not need to send ACK for the UPDATE(REA) in every subsequent message used for the address check procedure of the multiple addresses. Therefore, a sample packet exchange might look as shown in Figure 7.

```
     Multi-homed Host                    Peer Host

                UPDATE(REA(addr_1,addr_2), SEQ)
        ----------------------------------->
                UPDATE(ACK)
        <-----------------------------------

        sent to addr_1:UPDATE(SPI, SEQ, ECHO_REQUEST)
        <-----------------------------------
                UPDATE(ACK, ECHO_RESPONSE)
        ----------------------------------->

        sent to addr_2:UPDATE(SPI, SEQ, ECHO_REQUEST)
        <-----------------------------------
                UPDATE(ACK, ECHO_RESPONSE)
        ----------------------------------->
```

Figure 7: REA with multiple addresses

When processing inbound REAs that establish new security associations, a host uses the destination address of the UPDATE containing REA as the local address to which the REA plus NES is targeted. Hosts may send REA with the same IP address to different peer addresses-- this has the effect of creating multiple inbound SAs implicitly affiliated with different source addresses.

When rekeying in a multihoming situation in which there is an asymmetric number of SAs between two hosts, a respondent to the NES/UPDATE procedure may have some ambiguity as to which inbound SA it should update in response to the peer's UPDATE. In such a case, the host SHOULD choose an SA corresponding to the inbound interface on which the UPDATE was received.

5.3 Site multi-homing

A host may have an interface that has multiple globally reachable IP addresses. Such a situation may be a result of the site having multiple upper Internet Service Providers, or just because the site provides all hosts with both IPv4 and IPv6 addresses. It is desirable that the host can stay reachable with all or any subset of the currently available globally routable addresses, independent on how they are provided.

This case is handled the same as if there were different IP addresses, described above in Section 5.2. Note that a single interface may experience site multi-homing while the host itself may have multiple interfaces.

Note that a host may be multi-homed and mobile simultaneously, and that a multi-homed host may want to protect the location of some of its interfaces while revealing the real IP address of some others.

This document does not presently specify additional site multihoming extensions to HIP to further align it with the requirements of the multi6 working group.

5.4 Dual host multi-homing

Consider the case in which both hosts would like to add an additional address after the base exchange completes. In Figure 8, consider that host1 wants to add address addr1b. It would send a REA to host2 located at addr2a, and a new set of SPIs would be added between hosts 1 and 2 (call them SPI1b and SPI2b). Next, consider host2 deciding to add addr2b to the relationship. host2 now has a choice of which of host1's addresses to initiate REA to. It may choose to initiate a REA to addr1a, addr1b, or both. If it chooses to send to both, then a full mesh (four SA pairs) of SAs would exist between the two hosts. This is the most general case; it may be often the case that hosts primarily establish new SAs only with the peer's preferred address. The readdressing protocol is flexible enough to accommodate this choice.

```
              -<- SPI1a --                         -- SPI2a ->-
      host1 <              > addr1a <---> addr2a <              > host2
              ->- SPI2a --                         -- SPI1a -<-

                             addr1b <---> addr2b
```

Figure 8: Dual multihoming case in which each host uses REA to add a second address

5.5 Combined mobility and multi-homing

It looks likely that in the future many mobile hosts will be simultaneously mobile and multi-homed, i.e., have multiple mobile interfaces. Furthermore, if the interfaces use different access technologies, it is fairly likely that one of the interfaces may appear stable (retain its current IP address) while some other(s) may experience mobility (undergo IP address change).

The use of REA plus NES should be flexible enough to handle most such scenarios, although more complicated scenarios have not been studied so far.

5.6 Network renumbering

It is expected that IPv6 networks will be renumbered much more often than most IPv4 networks are. From an end-host point of view, network renumbering is similar to mobility.

5.7 Initiating the protocol in R1 or I2

A Responder host MAY include one or more REA parameters in the R1 packet that it sends to the Initiator. These parameters MUST be protected by the R1 signature. If the R1 packet contains REA parameters, the Initiator SHOULD send the I2 packet to the new preferred address. The I1 destination address and the new preferred address may be identical.

```
            Initiator                                 Responder

                              R1 with REA
                  <-----------------------------------
   record additional addresses
   change responder address
                     I2 with new SPI in SPI parameter
                  ----------------------------------->
                                                     (process normally)
                                  R2
                  <-----------------------------------
   (process normally)
```

Figure 9: REA inclusion in R1

An Initiator MAY include one or more REA parameters in the I2 packet, independent on whether there was REA parameter(s) in the R1 or not. These parameters MUST be protected by the I2 signature. Even if the I2 packet contains REA parameters, the Responder MUST still send the R2 packet to the source address of the I2. The new preferred address SHOULD be identical to the I2 source address.

```
            Initiator                                 Responder

                             I2 with REA
                  ----------------------------------->
                                                     (process normally)
                                                     record additional
addresses
                       R2 with new SPI in SPI parameter
                  <-----------------------------------
   (process normally)
                           data on new SA
                  ------------------------------------>
                                                      (process normally)
```

Figure 10: REA inclusion in I2

6. Parameter and packet formats

6.1 REA 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             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                              SPI                              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                       Address Lifetime                        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |P|                          Reserved                           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                            Address                            |
   |                                                               |
   |                                                               |
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   .                                                               .
   .                                                               .
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                       Address Lifetime                        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                           Reserved                            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                            Address                            |
   |                                                               |
   |                                                               |
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type: 3
Length: Length in octets, excluding Type and Length fields.
SPI: Security Parameter Index (SPI) corresponding to Addresses
P: Preferred address. Set to one if the first address in this REA is the new preferred address; otherwise set to zero.
Reserved: Zero when sent, ignored when received.
Address Lifetime: Address lifetime, in seconds.
Address: An IPv6 address or an IPv4-in-IPv6 format IPv4 address [2].

The SPI field identifies the SPI that this parameter applies to. It is implicitly qualified by the Host Identity of the sending host. The sending host is free to introduce new SPIs at will. That is, if a received REA has a new SPI, it means that all the old addresses, assigned to the other SPIs, are also supposed to still work, while

the new addresses in the newly received REA are supposed to be associated with a new SPI. On the other hand, if a received REA has an SPI that the receiver already knows about, it would replace (all) the address(es) currently associated with the SPI with the new one(s).

The Address Lifetime indicates how long the following address is expected to be valid. The lifetime is expressed in seconds. Each address MUST have a non-zero lifetime. The address is expected to become deprecated when the specified number of seconds has passed since the reception of the message. A deprecated address SHOULD NOT be used as an destination address if an alternate (non-deprecated) is available and has sufficient scope. Since IP addresses are ignored upon reception, deprecation status does not have any affect on the receiver.

The Address field contains an IPv6 address, or an IPv4 address in the IPv4-in-IPv6 format [2]. The latter format denotes a plain IPv4 address that can be used to reach the Mobile Host.

6.2 UPDATE packet with included REA

A number of combinations of parameters in an UPDATE packet are possible (e.g., see Section 5). Any UPDATE packet that includes a REA parameter SHOULD include both an HMAC and a HIP_SIGNATURE parameter.>

If there are multiple REA parameters to be sent in a single UPDATE, and at least one of the REA parameters is matched with a NES parameter, then each REA must be matched with a NES parameter, to avoid ambiguity:

```
   IP ( HIP ( REA1, REA2, NES1, NES2, [ DH, ] ... ) )
```

If there are multiple REA parameters to be sent and not all are paired with a NES, then multiple UPDATEs must be used (some with NES, some without) to avoid ambiguity in the pairing of REA with NES.

7. Processing rules

7.1 Sending REAs

The decision of when to send REAs is basically a local policy issue. However, it is RECOMMENDED that a host sends a REA whenever it recognizes a change of its IP addresses, and assumes that the change is going to last at least for a few seconds. Rapidly sending conflicting REAs SHOULD be avoided.

When a host decides to inform its peers about changes in its IP addresses, it has to decide how to group the various addresses, and whether to include any addresses on multiple SPIs. Since each SPI is associated with a different Security Association, the grouping policy may be based on IPsec replay protection considerations. In the typical case, simply basing the grouping on actual kernel level physical and logical interfaces is often the best policy. Virtual interfaces, such as IPsec tunnel interfaces or Mobile IP home addresses SHOULD NOT be announced.

Note that the purpose of announcing IP addresses in a REA is to provide connectivity between the communicating hosts. In most cases, tunnels (and therefore virtual interfaces) provide sub-optimal connectivity. Furthermore, it should be possible to replace most tunnels with HIP based "non-tunneling", therefore making most virtual interfaces fairly unnecessary in the future. On the other hand, there are clearly situations where tunnels are used for diagnostic and/or testing purposes. In such and other similar cases announcing the IP addresses of virtual interfaces may be appropriate.

Once the host has decided on the groups and assignment of addresses to the SPIs, it creates a REA parameter for each group. If there are multiple REA parameters, the parameters MUST be ordered so that the new preferred address is in the first REA parameter. Only one address (the first one, if at all) may be indicated as preferred in the REA parameter.

If addresses are being added to an existing SPI, the REA parameter indicates the existing SPI and the full set of valid addresses for that SPI. Any addresses previously ACTIVE on that SPI that are not included in the REA will be set to DEPRECATED by the receiver.

If a mobile host decides to change the SPI upon a readdress, it sends a REA with the SPI field within the REA set to the new SPI, and also a NES parameter with the Old SPI field set to the previous SPI and the New SPI field set to the new SPI. If multiple REA and NES parameters are included, the NES MUST be ordered such that they appear in the same order as the set of corresponding REAs. The

decision as to whether to rekey and send a new Diffie-Hellman parameter while performing readdressing is a local policy decision.

If new addresses and new SPIs are being created, the REA parameter's SPI field contains the new SPI, and the NES parameter's the Old SPI field and New SPI fields are both set to the new SPI, indicating that this is a new and not a replacement SPI.

If there are multiple REA parameters leading to a packet size that exceeds the MTU, the host SHOULD send multiple packets, each smaller than the MTU. In the case of R1 and I2, the additional packets should be UPDATE packets that are sent after the base exchange has been completed.

7.2 Handling received REAs

A host SHOULD be prepared to receive REA parameters in any HIP packets, excluding I1.

When a host receives a REA parameter, it first performs the following operations:

1. The host checks if the SPI listed is a new one. If it is a new one, it creates a new SPI that contains no addresses. If it is an existing one, it prepares to change the address set on the existing SPI.
2. For each address listed in the REA parameter, check that the address is a legal unicast or anycast address. That is, the address MUST NOT be a broadcast or multicast address. Note that some implementations MAY accept addresses that indicate the local host, since it may be allowed that the host runs HIP with itself.
3. For each address listed in the REA parameter, check if the address is already bound to the SPI. If the address is already bound, its lifetime is updated. If the status of the address is DEPRECATED, the status is changed to UNVERIFIED. If the address is not already bound, the address is added, and its status is set to UNVERIFIED. Mark all addresses on the SPI that were NOT listed in the REA parameter as DEPRECATED. As a result, the SPI now contains any addresses listed in the REA parameter either as UNVERIFIED or ACTIVE, and any old addresses not listed in the REA parameter as DEPRECATED.
4. If the REA is paired with a NES parameter, the NES parameter is processed. If the REA is replacing the address on an existing SPI, the SPI itself may be changed-- in this case, the host proceeds according to HIP rekeying procedures. This case is indicated by the NES parameter including an existing SPI in the Old SPI field and a new SPI in the New SPI field, and the SPI field in the REA matching the New SPI in the NES. If instead the REA corresponds to a new SPI, the NES will include the same SPI

in both its Old SPI and New SPI fields.

5. Mark all addresses at the address group that were NOT listed in the REA parameter as DEPRECATED.

Once the host has updated the SPI, if the REA parameter contains a new preferred address, the host SHOULD initiate a change of the preferred address. This usually requires that the host first verifies reachability of the address, and only then changes the preferred address. See Section 7.4.

7.3 Verifying address reachability

A host MAY want to verify the reachability of any UNVERIFIED address at any time. It typically does so by sending a nonce to the new address. For example, if the host is changing its SPI and is sending a NES to the peer, the new SPI value SHOULD be random and the value MAY be copied into an ECHO_REQUEST sent in the rekeying UPDATE. If the host is not rekeying, it MAY still use the ECHO_REQUEST parameter in an UPDATE message sent to the new address. A host MAY also use other message exchanges as confirmation of the address reachability. Note that in the case of receiving a REA on an R1 and replying with an I2, receiving the corresponding R2 is sufficient for marking the Responder's primary address active.

In some cases, it may be sufficient to use the arrival of data on a newly advertised SA as implicit address reachability verification, instead of waiting for the confirmation via a HIP packet (e.g., Figure 13). In this case, a host advertising a new SPI as part of its address reachability check SHOULD be prepared to receive traffic on the new SA. Marking the address active as a part of receiving data on the SA is an idempotent operation, and does not cause any harm.

```
  Mobile host                                    Peer host

                                                 prepare incoming SA
                   new SPI in R2, or UPDATE
             <-----------------------------------
switch to new outgoing SA
                        data on new SA
             ----------------------------------->
                                                 mark address ACTIVE
```

Figure 13: Address activation via use of new SA

7.4 Changing the preferred address

A host MAY want to change the preferred outgoing address for different reasons, e.g., because traffic information or ICMP error messages indicate that the currently used preferred address may have become unreachable. Another reason is receiving a REA parameter that has the P-bit set.

To change the preferred address, the host initiates the following procedure:

1. If the new preferred address has ACTIVE status, the preferred address is changed and the procedure succeeds.
2. If the new preferred address has UNVERIFIED status, the host starts to verify its reachability. Once the verification has succeeded, the preferred address change is completed, unless a new change has been initiated in the meantime.
3. If the peer host has not indicated a preference for any address, then the host picks one of the peer's ACTIVE addresses randomly or according to policy. This case may arise if, for example, ICMP error messages arrive that deprecate the preferred address, but the peer has not yet indicated a new preferred address.
4. If the new preferred address has DEPRECATED status and there is at least one non-deprecated address, the host selects one of the non-deprecated addresses as a new preferred address and continues.

8. Policy considerations

XXX: This section needs to be written.

The host may change the status of unused ACTIVE addresses into UNVERIFIED after a locally configured period of inactivity.

9. Security Considerations

XXX: This section requires lots of more work.

(Initial text by Jari Arkko): If not controlled in some manner, messaging related to address changes would create the following types of vulnerabilities:

Revealing the contents of the (cleartext) communications

Hijacking communications and man-in-the-middle attacks

Denial of service for the involved nodes, by disabling their ability to receive the desired communications

Denial of service for third parties, by redirecting a large amount of traffic to them

Revealing the location of the nodes to other parties

In HIP, communications are bound to the public keys of the end-points and not to IP addresses. The REA message is signed with the sender's public key, and hence it becomes impossible to hijack the communications of another host through the use of the REA message. Similarly, since only the host itself can sign messages to move its traffic flows to a new IP address, denial of service attacks through REA can not cause the traffic flows to be sent to an IP address that the host did not wish to use. Finally, in HIP all communications are encrypted with ESP, so a hijack attempt would also be unable to reveal the contents of the communications.

Malicious nodes that use HIP can, however, try to cause a denial of service attack by establishing a high-volume traffic flow, such as a video stream, and then redirecting it to a victim. However, the address reachability check provides some assurance that the given address is willing to accept the new traffic. Only attackers who are on the path between the peer and the new address could respond to the test.

10. IANA Considerations

11. Acknowledgments

12. References

12.1 Normative references

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

[2] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 2373, July 1998.

[3] Moskowitz, R., Nikander, P. and P. Jokela, "Host Identity Protocol", draft-moskowitz-hip-09 (work in progress), February 2004.

[4] Moskowitz, R., "Host Identity Protocol Architecture", draft-moskowitz-hip-arch-05 (work in progress), October 2003.

12.2 Informative references

[5] Bellovin, S., "EIDs, IPsec, and HostNAT", IETF 41th, March 1998.

[6] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", draft-iab-sec-cons-00 (work in progress), August 2002.

[7] Nikander, P., "Mobile IP version 6 Route Optimization Security Design Background", draft-nikander-mobileip-v6-ro-sec-02 (work in progress), December 2003.

Authors' Addresses

Pekka Nikander
Ericsson Research Nomadic Lab
JORVAS FIN-02420
FINLAND

Phone: +358 9 299 1
EMail: pekka.nikander@nomadiclab.com

Jari Arkko
Ericsson Research Nomadic Lab
JORVAS FIN-02420
FINLAND

Phone: +358 9 299 1
EMail: jari.arkko@nomadiclab.com

Tom Henderson
The Boeing Company
P.O. Box 3707
Seattle, WA
USA

EMail: thomas.r.henderson@boeing.com

Appendix A. Changes from previous versions

A.1 From nikander-hip-mm-00 to nikander-hip-mm-01

The actual protocol has been largely revised, based on the new symmetric New SPI (NES) design adopted in the base protocol draft version -08. There are no more separate REA, AC or ACR packets, but their functionality has been folded into the NES packet. At the same time, it has become possible to send REA parameters in R1 and I2.

The Forwarding Agent functionality was removed, since it looks like that it will be moved to the proposed HIP Research Group. Hence, there will be two other documents related to that, a simple Rendezvous server document (WG item) and a Forwarding Agent document (RG item).

A.2 From nikander-hip-mm-01 to nikander-hip-mm-02

Alignment with base-00 draft (use of UPDATE and NOTIFY packets).

The "logical interface" concept was dropped, and the SA/SPI was identified as the protocol component to which a HIP association binds addresses to.

The RR was (again) made recommended, not mandatory, able to be administratively overridden.

A.3 From -02 to draft-ietf-hip-mm-00

REA parameter type value is now "3" (was TBD before).

Recommend that in multihoming situations, that inbound/outbound SAs are paired to avoid ambiguity when rekeying them.

Clarified that multihoming scenario for now was intended for failover instead of load-balancing, due to transport layer issues.

Clarified that if HIP negotiates base exchange using link local addresses, that a host SHOULD provide its peer with a globally reachable address.

Clarified whether REAs sent for existing SPIs update the full set of addresses associated with that SPI, or only perform an incremental (additive) update. REAs for an existing SPI should list all current addresses for that SPI, and any addresses previously in use on the SPI but not in the new REA parameter should be DEPRECATED.

Clarified that address verification pertains to *outgoing* addresses.

When discussing incluson of REA in I2, the draft stated "The Responder MUST make sure that the puzzle solution is valid BOTH for the initial IP destination address used for I1 and for the new preferred address." However, this statement conflicted with Appendix D of the base specification, so it has been removed for now.

Appendix B. Implementation experiences

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.