                 Host Mobility with the Host Identity Protocol

Abstract

   This document defines a mobility extension to the Host Identity
   Protocol (HIP). Specifically, this document defines a "LOCATOR_SET"
   parameter for HIP messages that allows for a HIP host to notify peers
   about alternate addresses at which it may be reached. This document
   also defines how the parameter can be used to preserve communications
   across a change to the IP address used by one or both peer hosts.
   The same LOCATOR_SET parameter can also be used to support end-host
   multihoming (as specified in RFC 8047). This document obsoletes RFC
   5206.

Status of This Memo

   This is an Internet Standards Track document.

   This document is a product of the Internet Engineering Task Force
   (IETF). It represents the consensus of the IETF community. It has
   received public review and has been approved for publication by the
   Internet Engineering Steering Group (IESG). Further information on
   Internet Standards is available in Section 2 of RFC 7841.

   Information about the current status of this document, any errata,
   and how to provide feedback on it may be obtained at
   http://www.rfc-editor.org/info/rfc8046.

Copyright Notice

Table of Contents

1. Introduction and Scope

   The Host Identity Protocol (HIP) [RFC7401] supports an architecture
   that decouples the transport layer (TCP, UDP, etc.) from the
   internetworking layer (IPv4 and IPv6) by using public/private key
   pairs, instead of IP addresses, as host identities. When a host uses
   HIP, the overlying protocol sublayers (e.g., transport-layer sockets
   and Encapsulating Security Payload (ESP) Security Associations (SAs))
   are instead bound to representations of these host identities, and
   the IP addresses are only used for packet forwarding. However, each
   host needs to also know at least one IP address at which its peers
   are reachable. Initially, these IP addresses are the ones used
   during the HIP base exchange.

   One consequence of such a decoupling is that new solutions to
   network-layer mobility and host multihoming are possible. There are
   potentially many variations of mobility and multihoming possible.
   The scope of this document encompasses messaging and elements of
   procedure for basic network-level host mobility, leaving more
   complicated mobility scenarios, multihoming, and other variations for
   further study. More specifically, the following are in scope:

      This document defines a LOCATOR_SET parameter for use in HIP
      messages. The LOCATOR_SET parameter allows a HIP host to notify a
      peer about alternate locators at which it is reachable. The
      locators may be merely IP addresses, or they may have additional
      multiplexing and demultiplexing context to aid with the packet
      handling in the lower layers. For instance, an IP address may
      need to be paired with an ESP Security Parameter Index (SPI) so
      that packets are sent on the correct SA for a given address.

      This document also specifies the messaging and elements of
      procedure for end-host mobility of a HIP host. In particular,
      message flows to enable successful host mobility, including
      address verification methods, are defined herein.

      The HIP rendezvous server (RVS) [RFC8004] can be used to manage
      simultaneous mobility of both hosts, initial reachability of a
      mobile host, location privacy, and some modes of NAT traversal.
      Use of the HIP RVS to manage the simultaneous mobility of both
      hosts is specified herein.

The following topics are out of scope:

   While the same LOCATOR_SET parameter supports host multihoming
   (simultaneous use of a number of addresses), procedures for host
   multihoming are out of scope and are specified in [RFC8047].

   While HIP can potentially be used with transports other than the
   ESP transport format [RFC7402], this document largely assumes the
   use of ESP and leaves other transport formats for further study.

   We do not consider localized mobility management extensions (i.e.,
   mobility management techniques that do not involve directly
   signaling the correspondent node); this document is concerned with
   end-to-end mobility.

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

The main sections of this document are organized as follows.
Section 3 provides a summary overview of operations, scenarios, and
other considerations. Section 4 specifies the messaging parameter
syntax. Section 5 specifies the processing rules for messages.
Section 6 describes security considerations for this specification.

2. Terminology and Conventions

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].

LOCATOR_SET. A HIP parameter containing zero or more Locator fields.

locator. A name that controls how the packet is routed through the
   network and demultiplexed by the end host. It may include a
   concatenation of traditional network addresses such as an IPv6
   address and end-to-end identifiers such as an ESP SPI. It may
   also include transport port numbers or IPv6 Flow Labels as
   demultiplexing context, or it may simply be a network address.

Locator. When capitalized in the middle of a sentence, this term
   refers to the encoding of a locator within the LOCATOR_SET
   parameter (i.e., the 'Locator' field of the parameter).

   Address. A name that denotes a point of attachment to the network.
      The two most common examples are an IPv4 address and an IPv6
      address. The set of possible addresses is a subset of the set of
      possible locators.

   Preferred locator. A locator on which a host prefers to receive
      data. Certain locators are labeled as preferred when a host
      advertises its locator set to its peer. By default, the locators
      used in the HIP base exchange are the preferred locators. The use
      of preferred locators, including the scenario where multiple
      address scopes and families may be in use, is defined more in
      [RFC8047] than in this document.

   Credit-Based Authorization (CBA). A mechanism allowing a host to
      send a certain amount of data to a peer's newly announced locator
      before the result of mandatory address verification is known.

3. Protocol Model

   This section is an overview; a more detailed specification follows
   this section.

3.1. Operating Environment

   HIP [RFC7401] is a key establishment and parameter negotiation
   protocol. Its primary applications are for authenticating host
   messages based on host identities and establishing SAs for the ESP
   transport format [RFC7402] and possibly other protocols in the
   future.

```
   +-------------------+                  +-------------------+
   |                   |                  |                   |
   |   +-----------+   |                  |   +-----------+   |
   |   | Key       |   |      HIP         |   | Key       |   |
   |   | Management| <-+------------------+-> | Management|   |
   |   | Process   |   |                  |   | Process   |   |
   |   +-----------+   |                  |   +-----------+   |
   |        ^          |                  |        ^          |
   |        |          |                  |        |          |
   |        v          |                  |        v          |
   |   +-----------+   |                  |   +-----------+   |
   |   | IPsec     |   |      ESP         |   | IPsec     |   |
   |   | Stack     | <-+------------------+-> | Stack     |   |
   |   |           |   |                  |   |           |   |
   |   +-----------+   |                  |   +-----------+   |
   |                   |                  |                   |
   |                   |                  |                   |
   |     Initiator     |                  |     Responder     |
   +-------------------+                  +-------------------+
```
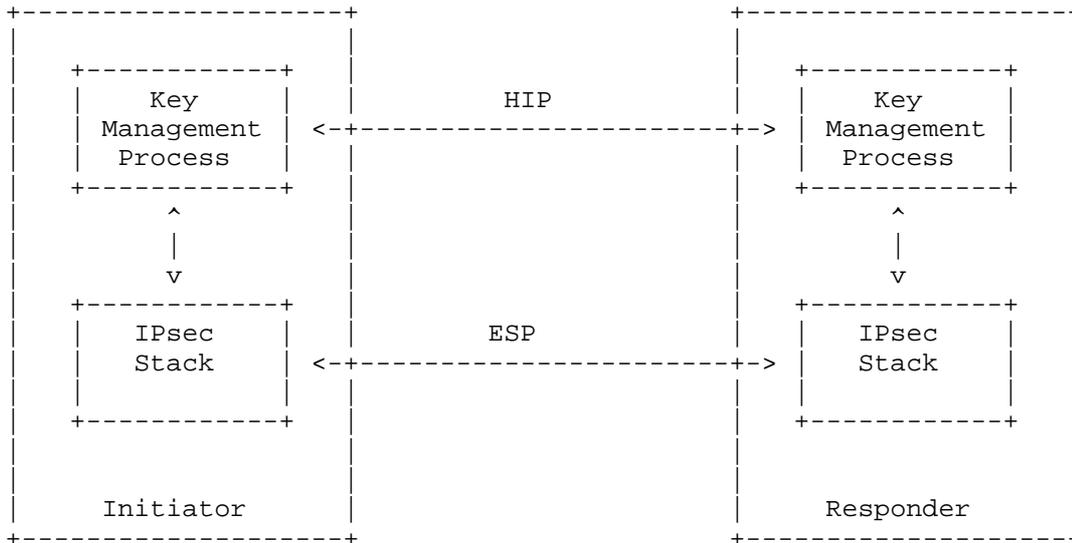
                     Figure 1: HIP Deployment Model

   The general deployment model for HIP is shown above, assuming
   operation in an end-to-end fashion. This document specifies an
   extension to HIP to enable end-host mobility. In summary, these
   extensions to the HIP base protocol enable the signaling of new
   addressing information to the peer in HIP messages. The messages are
   authenticated via a signature or keyed Hash Message Authentication
   Code (HMAC) based on its Host Identity (HI). This document specifies
   the format of this new addressing (LOCATOR_SET) parameter, the
   procedures for sending and processing this parameter to enable basic
   host mobility, and procedures for a concurrent address verification
   mechanism.

```
                       ---------
                      | TCP    |  (sockets bound to HITs)
                       ---------
                          |
                       ---------
          ----> | ESP    |  {HIT_s, HIT_d} <-> SPI
          |      ---------
          |          |
        ----      ---------
       | MH |-> | HIP    |  {HIT_s, HIT_d, SPI} <-> {IP_s, IP_d, SPI}
        ----      ---------
                          |
                       ---------
                      |  IP    |
                       ---------
```

            Figure 2: Architecture for HIP Host Mobility and Multihoming

Figure 2 depicts a layered architectural view of a HIP-enabled stack
using the ESP transport format. In HIP, upper-layer protocols
(including TCP and ESP in this figure) are bound to Host Identity
Tags (HITs) and not IP addresses. The HIP sublayer is responsible
for maintaining the binding between HITs and IP addresses. The SPI
is used to associate an incoming packet with the right HITs. The
block labeled "MH" corresponds to the function that manages the
bindings at the ESP and HIP sublayers for mobility (specified in this
document) and multihoming (specified in [RFC8047]).

Consider first the case in which there is no mobility or multihoming,
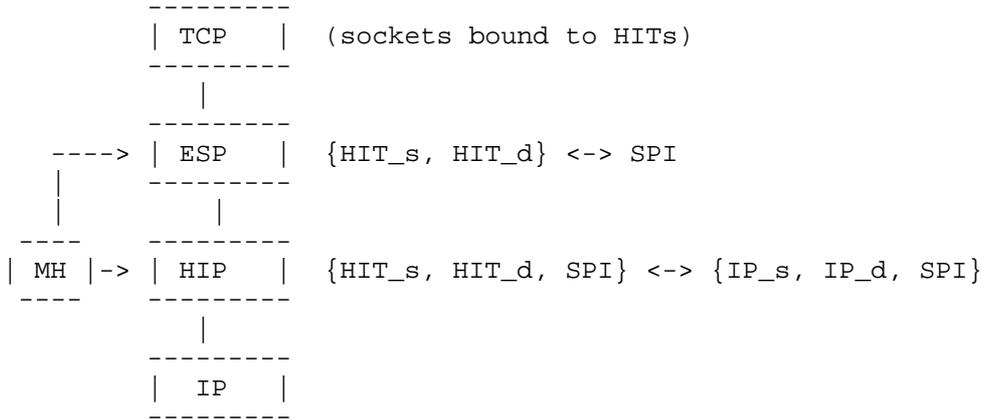as specified in the base protocol specification [RFC7401]. The HIP
base exchange establishes the HITs in use between the hosts, the SPIs
to use for ESP, and the IP addresses (used in both the HIP signaling
packets and ESP data packets). Note that there can only be one such
set of bindings in the outbound direction for any given packet, and
the only fields used for the binding at the HIP layer are the fields
exposed by ESP (the SPI and HITs). For the inbound direction, the
SPI is all that is required to find the right host context. ESP
rekeying events change the mapping between the HIT pair and SPI, but
do not change the IP addresses.

Consider next a mobility event, in which a host moves to another IP
address. Two things need to occur in this case. First, the peer
needs to be notified of the address change using a HIP UPDATE
message. Second, each host needs to change its local bindings at the
HIP sublayer (new IP addresses). It may be that both the SPIs and IP
addresses are changed simultaneously in a single UPDATE; the protocol
described herein supports this. Although internal notification of
transport-layer protocols regarding the path change (e.g., to reset

congestion control variables) may be desired, this specification does
not address such internal notification. In addition, elements of
procedure for traversing network address translators (NATs) and
firewalls, including NATs and firewalls that may understand HIP, may
complicate the above basic scenario and are not covered by this
document.

3.1.1. Locator

   This document defines a generalization of an address called a
   "locator". A locator specifies a point of attachment to the network
   but may also include additional end-to-end tunneling or a per-host
   demultiplexing context that affects how packets are handled below the
   logical HIP sublayer of the stack. This generalization is useful
   because IP addresses alone may not be sufficient to describe how
   packets should be handled below HIP. For example, in a host
   multihoming context, certain IP addresses may need to be associated
   with certain ESP SPIs to avoid violating the ESP anti-replay window.
   Addresses may also be affiliated with transport ports in certain
   tunneling scenarios. Locators may simply be traditional network
   addresses. The format of the Locator fields in the LOCATOR_SET
   parameter is defined in Section 4.

3.1.2. Mobility Overview

   When a host moves to another address, it notifies its peer of the new
   address by sending a HIP UPDATE packet containing a single
   LOCATOR_SET parameter and a single ESP_INFO parameter. This UPDATE
   packet is acknowledged by the peer. For reliability in the presence
   of packet loss, the UPDATE packet is retransmitted as defined in the
   HIP specification [RFC7401]. The peer can authenticate the contents
   of the UPDATE packet based on the signature and keyed hash of the
   packet.

   When using the ESP transport format [RFC7402], the host may, at the
   same time, decide to rekey its security association and possibly
   generate a new Diffie-Hellman key; all of these actions are triggered
   by including additional parameters in the UPDATE packet, as defined
   in the base protocol specification [RFC7401] and ESP extension
   [RFC7402].

   When using ESP (and possibly other transport modes in the future),
   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 without necessarily
   rekeying. However, the peers are not able to send packets to these
   new addresses 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
for the SA, thereby requiring rekeying.

3.2. Protocol Overview

   In this section, we briefly introduce a number of usage scenarios for
   HIP host mobility. These scenarios assume that HIP is being used
   with the ESP transform [RFC7402], although other scenarios may be
   defined in the future. To understand these usage scenarios, the
   reader should be at least minimally familiar with the HIP
   specification [RFC7401] and with the use of ESP with HIP [RFC7402].
   According to these specifications, the data traffic in a HIP session
   is protected with ESP, and the ESP SPI acts as an index to the right
   host-to-host context. More specification details are found later in
   Sections 4 and 5.

   The scenarios below assume that the two hosts have completed a single
   HIP base exchange with each other. Therefore, both of the hosts have
   one incoming and one outgoing SA. Further, each SA uses the same
   pair of IP addresses, which are the ones used in the base exchange.

   The readdressing protocol is an asymmetric protocol where a mobile
   host informs a peer host about changes of IP addresses on affected
   SPIs. The readdressing exchange is designed to be piggybacked on
   existing HIP exchanges. In support of mobility, the LOCATOR_SET
   parameter is carried in UPDATE packets.

   The scenarios below at times describe addresses as being in either an
   ACTIVE, UNVERIFIED, or DEPRECATED state. From the perspective of a
   host, newly learned addresses of the peer need to be verified before
   put into active service, and addresses removed by the peer are put
   into a deprecated state. Under limited conditions described below
   (Section 5.6), an UNVERIFIED address may be used. The addressing
   states are defined more formally in Section 5.1.

   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 LOCATOR_SET parameter.

3.2.1. Mobility with a Single SA Pair (No Rekeying)

   A mobile host sometimes needs to 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 needs to inform

its peers about the new IP address. This first example considers the
case in which the mobile host has only one interface, one IP address
in use within the HIP session, a single pair of SAs (one inbound, one
outbound), and no rekeying occurring on the SAs. We also assume that
the new IP addresses are within the same address family (IPv4 or
IPv6) as the previous address. This is the simplest scenario,
depicted in Figure 3. Note that the conventions for message
parameter notations in figures (use of parentheses and brackets) is
defined in Section 2.2 of [RFC7401].

```
   Mobile Host                            Peer Host

           UPDATE(ESP_INFO, LOCATOR_SET, SEQ)
      ---------------------------------->
           UPDATE(ESP_INFO, SEQ, ACK, ECHO_REQUEST)
      <---------------------------------
           UPDATE(ACK, ECHO_RESPONSE)
      ---------------------------------->
```
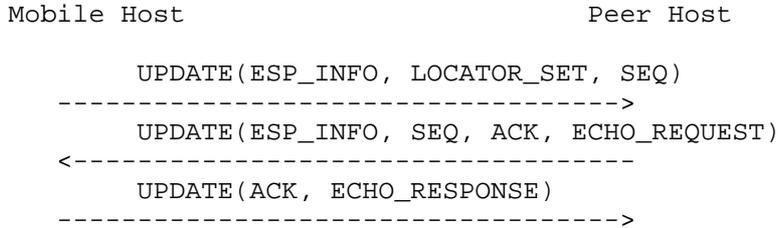
        Figure 3: Readdress without Rekeying but with Address Check

The steps of the packet processing are as follows:

1. The mobile host may be disconnected from the peer host for a
    brief period of time while it switches from one IP address to
    another; this case is sometimes referred to in the literature as
    a "break-before-make" case. The host may also obtain its new IP
    address before losing the old one ("make-before-break" case). In
    either case, upon obtaining a new IP address, the mobile host
    sends a LOCATOR_SET parameter to the peer host in an UPDATE
    message. The UPDATE message also contains an ESP_INFO parameter
    containing the values of the old and new SPIs for a security
    association. In this case, both the OLD SPI and NEW SPI
    parameters are set to the value of the preexisting incoming SPI;
    this ESP_INFO does not trigger a rekeying event but is instead
    included for possible parameter-inspecting firewalls on the path
    ([RFC5207] specifies some such firewall scenarios in which the
    HIP-aware firewall may want to associate ESP flows to host
    identities). The LOCATOR_SET parameter contains the new IP
    address (embedded in a Locator Type of "1", defined below) and a
    lifetime associated with the locator. The mobile host waits for
    this UPDATE to be acknowledged, and retransmits if necessary, as
    specified in the base specification [RFC7401].

2. The peer host receives the UPDATE, validates it, and updates any
    local bindings between the HIP association and the mobile host's
    destination address. The peer host MUST perform an address
    verification by placing a nonce in the ECHO_REQUEST parameter of
    the UPDATE message sent back to the mobile host. It also
    includes an ESP_INFO parameter with both the OLD SPI and NEW SPI
    parameters set to the value of the preexisting incoming SPI and
    sends this UPDATE (with piggybacked acknowledgment) to the mobile
    host at its new address. This UPDATE also acknowledges the
    mobile host's UPDATE that triggered the exchange. The peer host
    waits for its UPDATE to be acknowledged, and retransmits if
    necessary, as specified in the base specification [RFC7401]. The
    peer MAY use the new address immediately, but it MUST limit the
    amount of data it sends to the address until address verification
    completes.

3. The mobile host completes the readdress by processing the UPDATE
    ACK and echoing the nonce in an ECHO_RESPONSE, containing the ACK
    of the peer's UPDATE. This UPDATE is not protected by a
    retransmission timer because it does not contain a SEQ parameter
    requesting acknowledgment. Once the peer host receives this
    ECHO_RESPONSE, it considers the new address to be verified and
    can put the address into full use.

While the peer host is verifying the new address, the new address is
marked as UNVERIFIED (in the interim), and the old address is
DEPRECATED. Once the peer host has received a correct reply to its
UPDATE challenge, it marks the new address as ACTIVE and removes the
old address.

3.2.2. Mobility with a Single SA Pair (Mobile-Initiated Rekey)

The mobile host may decide to rekey the SAs at the same time that it
notifies the peer of the new address. In this case, the above
procedure described in Figure 3 is slightly modified. The UPDATE
message sent from the mobile host includes an ESP_INFO with the OLD
SPI set to the previous SPI, the NEW SPI set to the desired new SPI
value for the incoming SA, and the KEYMAT Index desired. Optionally,
the host may include a DIFFIE_HELLMAN parameter for a new Diffie-
Hellman key. The peer completes the request for a rekey as is
normally done for HIP rekeying, except that the new address is kept
as UNVERIFIED until the UPDATE nonce challenge is received as
described above. Figure 4 illustrates this scenario.

```
   Mobile Host                           Peer Host

         UPDATE(ESP_INFO, LOCATOR_SET, SEQ, [DIFFIE_HELLMAN])
       ----------------------------------->
         UPDATE(ESP_INFO, SEQ, ACK, [DIFFIE_HELLMAN,] ECHO_REQUEST)
       <-----------------------------------
         UPDATE(ACK, ECHO_RESPONSE)
       ----------------------------------->
```

          Figure 4: Readdress with Mobile-Initiated Rekey

3.2.3. Mobility Messaging through the Rendezvous Server

   Section 6.11 of [RFC7401] specifies procedures for sending HIP UPDATE
   packets. The UPDATE packets are protected by a timer subject to
   exponential backoff and resent UPDATE_RETRY_MAX times. It may be,
   however, that the peer is itself in the process of moving when the
   local host is trying to update the IP address bindings of the HIP
   association. This is sometimes called the "double-jump" mobility
   problem; each host's UPDATE packets are simultaneously sent to a
   stale address of the peer, and the hosts are no longer reachable from
   one another.

   The HIP Rendezvous Extension [RFC8004] specifies a rendezvous service
   that permits the I1 packet from the base exchange to be relayed from
   a stable or well-known public IP address location to the current IP
   address of the host. It is possible to support double-jump mobility
   with this rendezvous service if the following extensions to the
   specifications of [RFC8004] and [RFC7401] are followed.

   1. The mobile host sending an UPDATE to the peer, and not receiving
       an ACK, MAY resend the UPDATE to an RVS of the peer, if such a
       server is known. The host MAY try the RVS of the peer up to
       UPDATE_RETRY_MAX times as specified in [RFC7401]. The host MAY
       try to use the peer's RVS before it has tried UPDATE_RETRY_MAX
       times to the last working address (i.e., the RVS MAY be tried in
       parallel with retries to the last working address). The
       aggressiveness of a host replicating its UPDATEs to multiple
       destinations, to try candidates in parallel instead of serially,
       is a policy choice outside of this specification.

   2. An RVS supporting the UPDATE forwarding extensions specified
       herein MUST modify the UPDATE in the same manner as it modifies
       the I1 packet before forwarding. Specifically, it MUST rewrite
       the IP header source and destination addresses, recompute the IP
       header checksum, and include the FROM and RVS_HMAC parameters.

   3. A host receiving an UPDATE packet MUST be prepared to process the
       FROM and RVS_HMAC parameters and MUST include a VIA_RVS parameter
       in the UPDATE reply that contains the ACK of the UPDATE SEQ.

   4. An Initiator receiving a VIA_RVS in the UPDATE reply should
       initiate address reachability tests (described later in this
       document) towards the end host's address and not towards the
       address included in the VIA_RVS.

   This scenario requires that hosts using RVSs also take steps to
   update their current address bindings with their RVS upon a mobility
   event. [RFC8004] does not specify how to update the RVS with a
   client host's new address. Section 3.2 of [RFC8003] describes how a
   host may send a REG_REQUEST in either an I2 packet (if there is no
   active association) or an UPDATE packet (if such association exists).
   According to procedures described in [RFC8003], if a mobile host has
   an active registration, it may use mobility updates specified herein,
   within the context of that association, to readdress the association.

3.2.4. Network Renumbering

   It is expected that IPv6 networks will be renumbered much more often
   than most IPv4 networks. From an end-host point of view, network
   renumbering is similar to mobility, and procedures described herein
   also apply to notify a peer of a changed address.

3.3. Other Considerations

3.3.1. Address Verification

   When a HIP host receives a set of locators from another HIP host in a
   LOCATOR_SET, 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 bogus addresses in order to
   cause a packet flood towards the target addresses [RFC4225].
   Therefore, the HIP host needs to first check that the peer is
   reachable at the new address.

   Address verification 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 the exchange of
   a nonce or the generation of a new SPI and observation of data
   arriving on the new SPI. More details are found in Section 5.4 of
   this document.

An additional potential benefit of performing address verification is
to allow NATs and firewalls in the network along the new path to
obtain the peer host's inbound SPI.

3.3.2. Credit-Based Authorization

CBA allows a host to securely use a new locator even though the
peer's reachability at the address embedded in the locator has not
yet been verified. This is accomplished based on the following three
hypotheses:

1. A flooding attacker typically seeks to somehow multiply the
    packets it generates for the purpose of its attack because
    bandwidth is an ample resource for many victims.

2. An attacker can often cause unamplified flooding by sending
    packets to its victim, either by directly addressing the victim
    in the packets or by guiding the packets along a specific path by
    means of an IPv6 Routing header, if Routing headers are not
    filtered by firewalls.

3. Consequently, the additional effort required to set up a
    redirection-based flooding attack (without CBA and return
    routability checks) would pay off for the attacker only if
    amplification could be obtained this way.

On this basis, rather than eliminating malicious packet redirection
in the first place, CBA prevents amplifications. This is
accomplished by limiting the data a host can send to an unverified
address of a peer by the data recently received from that peer.
Redirection-based flooding attacks thus become less attractive than,
for example, pure direct flooding, where the attacker itself sends
bogus packets to the victim.

Figure 5 illustrates CBA: Host B measures the amount of data recently
received from peer A and, when A readdresses, sends packets to A's
new, unverified address as long as the sum of the packet sizes does
not exceed the measured, received data volume. When insufficient
credit is left, B stops sending further packets to A until A's
address becomes ACTIVE. The address changes may be due to mobility,
multihoming, or any other reason. Not shown in Figure 5 are the
results of credit aging (Section 5.6.2), a mechanism used to dampen
possible time-shifting attacks.

```
            +-------+                        +-------+
            |   A   |                        |   B   |
            +-------+                        +-------+
                |                                |
    address     |------------------------------->| credit += size(packet)
      ACTIVE    |                                |
                |------------------------------->| credit += size(packet)
                |<-------------------------------| do not change credit
                |                                |
                |                                |
                + address change                |
                + address verification starts   |
    address     |<-------------------------------| credit -= size(packet)
  UNVERIFIED    |------------------------------->| credit += size(packet)
                |<-------------------------------| credit -= size(packet)
                |                                |
                |                                |
                |<-------------------------------| credit -= size(packet)
                |                              X credit < size(packet)
                |                                | => do not send packet!
                + address verification concludes |
    address     |                                |
      ACTIVE    |<-------------------------------| do not change credit
                |                                |
```
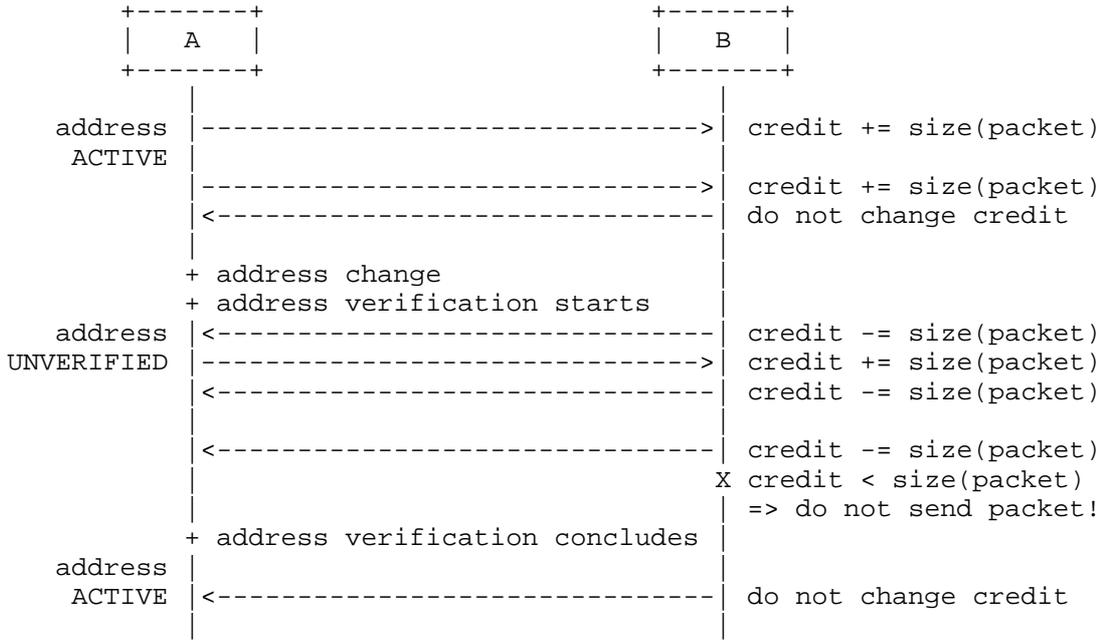
                   Figure 5: Readdressing Scenario

   This document does not specify how to set the credit limit value, but
   the goal is to allow data transfers to proceed without much
   interruption while the new address is verified. A simple heuristic
   to accomplish this, if the sender knows roughly its round-trip time
   (RTT) and current sending rate to the host, is to allow enough credit
   to support maintaining the sending rate for a duration corresponding
   to two or three RTTs.

3.3.3. Preferred Locator

   When a host has multiple locators, the peer host needs to decide
   which to use for outbound packets. It may be that a host would
   prefer to receive data on a particular inbound interface. HIP allows
   a particular locator to be designated as a preferred locator and
   communicated to the peer (see Section 4).

4. LOCATOR_SET Parameter Format

   The LOCATOR_SET parameter has a type number value that is considered
   to be a "critical parameter" as per the definition in [RFC7401]; such
   parameter types MUST be recognized and processed by the recipient.
   The parameter consists of the standard HIP parameter Type and Length
   fields, plus zero or more Locator sub-parameters. Each Locator sub-

parameter contains a Traffic Type, Locator Type, Locator Length,
preferred locator bit ("P" bit), Locator Lifetime, and a Locator
encoding. A LOCATOR_SET containing zero Locator fields is permitted
but has the effect of deprecating all addresses.

```
 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             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Traffic Type  | Locator Type | Locator Length | Reserved   |P|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Locator Lifetime                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Locator                              |
|                                                               |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                                                               .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Traffic Type  | Locator Type | Locator Length | Reserved   |P|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Locator Lifetime                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Locator                              |
|                                                               |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
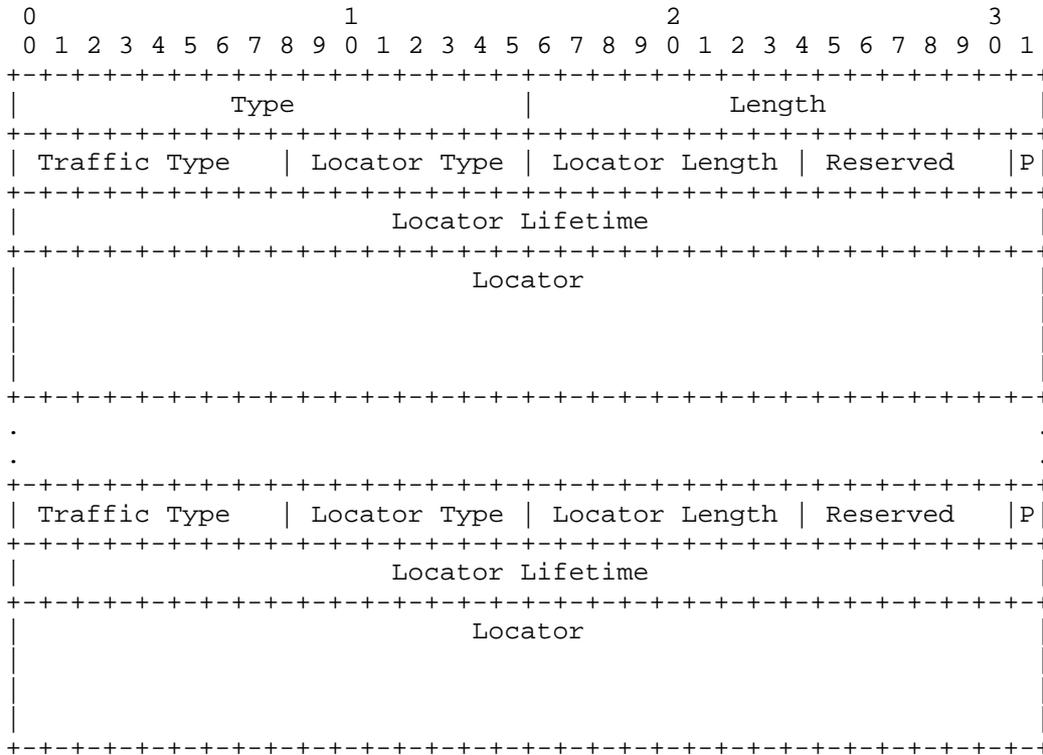
Figure 6: LOCATOR_SET Parameter Format

Type: 193

Length: Length in octets, excluding Type and Length fields, and
   excluding padding.

Traffic Type: Defines whether the locator pertains to HIP signaling,
   user data, or both.

Locator Type: Defines the semantics of the Locator field.

Locator Length: Defines the length of the Locator field, in units of
   4-byte words (Locators up to a maximum of 4*255 octets are
   supported).

Reserved: Zero when sent, ignored when received.

P: Preferred locator. Set to one if the locator is preferred for
   that Traffic Type; otherwise, set to zero.

Locator Lifetime: Lifetime of the locator, in seconds.

Locator: The locator whose semantics and encoding are indicated by
   the Locator Type field. All sub-fields of the Locator field are
   integral multiples of four octets in length.

The Locator Lifetime (lifetime) indicates how long the following
locator is expected to be valid. The lifetime is expressed in
seconds. Each locator 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 a destination address if an alternate
(non-deprecated) is available and has sufficient address scope.

4.1. Traffic Type and Preferred Locator

The following Traffic Type values are defined:

0:   Both signaling (HIP control packets) and user data.

1:   Signaling packets only.

2:   Data packets only.

The "P" bit, when set, has scope over the corresponding Traffic Type.
That is, when a "P" bit is set for Traffic Type "2", for example, it
means that the locator is preferred for data packets. If there is a
conflict (for example, if the "P" bit is set for an address of Type
"0" and a different address of Type "2"), the more specific Traffic
Type rule applies (in this case, "2"). By default, the IP addresses
used in the base exchange are preferred locators for both signaling
and user data, unless a new preferred locator supersedes them. If no
locators are indicated as preferred for a given Traffic Type, the
implementation may use an arbitrary destination locator from the set
of active locators.

4.2. Locator Type and Locator

   The following Locator Type values are defined, along with the
   associated semantics of the Locator field:

   0: An IPv6 address or an IPv4-in-IPv6 format IPv4 address [RFC4291]
       (128 bits long). This Locator Type is defined primarily for
       non-ESP-based usage.

   1: The concatenation of an ESP SPI (first 32 bits) followed by an
       IPv6 address or an IPv4-in-IPv6 format IPv4 address (an
       additional 128 bits). This IP address is defined primarily for
       ESP-based usage.

4.3. UPDATE Packet with Included LOCATOR_SET

   A number of combinations of parameters in an UPDATE packet are
   possible (e.g., see Section 3.2). In this document, procedures are
   defined only for the case in which one LOCATOR_SET and one ESP_INFO
   parameter are used in any HIP packet. Any UPDATE packet that
   includes a LOCATOR_SET parameter SHOULD include both an HMAC and a
   HIP_SIGNATURE parameter.

   The UPDATE MAY also include a HOST_ID parameter (which may be useful
   for HIP-aware firewalls inspecting the HIP messages for the first
   time). If the UPDATE includes the HOST_ID parameter, the receiving
   host MUST verify that the HOST_ID corresponds to the HOST_ID that was
   used to establish the HIP association, and the HIP_SIGNATURE MUST
   verify with the public key associated with this HOST_ID parameter.

   The relationship between the announced Locators and any ESP_INFO
   parameters present in the packet is defined in Section 5.2. This
   document does not support any elements of procedure for sending more
   than one LOCATOR_SET or ESP_INFO parameter in a single UPDATE.

5. Processing Rules

   This section describes rules for sending and receiving the
   LOCATOR_SET parameter, testing address reachability, and using CBA on
   UNVERIFIED locators.

5.1. Locator Data Structure and Status

   Each locator announced in a LOCATOR_SET parameter is represented by a
   piece of state that contains the following data:

   o the actual bit pattern representing the locator,

o the lifetime (seconds),

o the status (UNVERIFIED, ACTIVE, DEPRECATED),

o the Traffic Type scope of the locator, and

o whether the locator is preferred for any particular scope.

The status is used to track the reachability of the address embedded
within the LOCATOR_SET parameter:

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, and

DEPRECATED: indicates that the locator's lifetime has expired.

The following state changes are allowed:

UNVERIFIED to ACTIVE: The reachability procedure completes
   successfully.

UNVERIFIED to DEPRECATED: The locator's lifetime expires while the
   locator is UNVERIFIED.

ACTIVE to DEPRECATED: The locator's lifetime expires while the
   locator 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 needs to be verified again before starting to use it
   again.

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

A DEPRECATED address MUST NOT be changed to ACTIVE without first
verifying its reachability.

Note that the state of whether or not a locator is preferred is not
necessarily the same as the value of the preferred bit in the Locator
sub-parameter received from the peer. Peers may recommend certain
locators to be preferred, but the decision on whether to actually use
a locator as a preferred locator is a local decision, possibly
influenced by local policy.

In addition to state maintained about status and remaining lifetime
for each locator learned from the peer, an implementation would
typically maintain similar state about its own locators that have
been offered to the peer.

A locator lifetime that is unbounded (does not expire) can be
signified by setting the value of the lifetime field to the maximum
(unsigned) value.

Finally, the locators used to establish the HIP association are by
default assumed to be the initial preferred locators in ACTIVE state,
with an unbounded lifetime.

5.2. Sending the LOCATOR_SET

The decision of when to send the LOCATOR_SET is a local policy issue.
However, it is RECOMMENDED that a host send a LOCATOR_SET whenever it
recognizes a change of its IP addresses in use on an active HIP
association and assumes that the change is going to last at least for
a few seconds. Rapidly sending LOCATOR_SETs that force the peer to
change the preferred address SHOULD be avoided.

The sending of a new LOCATOR_SET parameter replaces the locator
information from any previously sent LOCATOR_SET parameter;
therefore, if a host sends a new LOCATOR_SET parameter, it needs to
continue to include all active locators. Hosts MUST NOT announce
broadcast or multicast addresses in LOCATOR_SETs.

We now describe a few cases introduced in Section 3.2. We assume
that the Traffic Type for each locator is set to "0" (other values
for Traffic Type may be specified in documents that separate the HIP
control plane from data-plane traffic). Other mobility cases are
possible but are left for further study.

1. Host mobility with no multihoming and no rekeying. The mobile
    host creates a single UPDATE containing a single ESP_INFO with a
    single LOCATOR_SET parameter. The ESP_INFO contains the current
    value of the SPI in both the OLD SPI and NEW SPI fields. The
    LOCATOR_SET contains a single Locator with a Locator Type of "1";
    the SPI MUST match that of the ESP_INFO. The preferred bit
    SHOULD be set and the "Locator Lifetime" is set according to
    local policy. The UPDATE also contains a SEQ parameter as usual.
    This packet is retransmitted as defined in the HIP specification
    [RFC7401]. The UPDATE should be sent to the peer's preferred IP
    address with an IP source address corresponding to the address in
    the LOCATOR_SET parameter.

   2. Host mobility with no multihoming but with rekeying. The mobile
       host creates a single UPDATE containing a single ESP_INFO with a
       single LOCATOR_SET parameter (with a single address). The
       ESP_INFO contains the current value of the SPI in the OLD SPI,
       the new value of the SPI in the NEW SPI, and a KEYMAT Index as
       selected by local policy. Optionally, the host may choose to
       initiate a Diffie-Hellman rekey by including a DIFFIE_HELLMAN
       parameter. The LOCATOR_SET contains a single Locator with a
       Locator Type of "1"; the SPI MUST match that of the NEW SPI in
       the ESP_INFO. Otherwise, the steps are identical to the case in
       which no rekeying is initiated.

5.3. Handling Received LOCATOR_SETs

   A host SHOULD be prepared to receive a single LOCATOR_SET parameter
   in a HIP UPDATE packet. Reception of multiple LOCATOR_SET parameters
   in a single packet, or in HIP packets other than UPDATE, is outside
   of the scope of this specification.

   Because a host sending the LOCATOR_SET may send the same parameter in
   different UPDATE messages to different destination addresses,
   including possibly the RVS of the host, the host receiving the
   LOCATOR_SET MUST be prepared to handle the possibility of duplicate
   LOCATOR_SETs sent to more than one of the host's addresses. As a
   result, the host MUST detect and avoid reprocessing a LOCATOR_SET
   parameter that is redundant with a LOCATOR_SET parameter that has
   been recently received and processed.

   This document describes sending both ESP_INFO and LOCATOR_SET
   parameters in an UPDATE. The ESP_INFO parameter is included when
   there is a need to rekey or key a new SPI, and is otherwise included
   for the possible benefit of HIP-aware NATs and firewalls. The
   LOCATOR_SET parameter contains a complete listing of the locators
   that the host wishes to make or keep active for the HIP association.

   In general, the processing of a LOCATOR_SET depends upon the packet
   type in which it is included. Here, we describe only the case in
   which ESP_INFO is present and a single LOCATOR_SET and ESP_INFO are
   sent in an UPDATE message; other cases are for further study. The
   steps below cover each of the cases described in Section 5.2.

   The processing of ESP_INFO and LOCATOR_SET parameters is intended to
   be modular and support future generalization to the inclusion of
   multiple ESP_INFO and/or multiple LOCATOR_SET parameters. A host
   SHOULD first process the ESP_INFO before the LOCATOR_SET, since the
   ESP_INFO may contain a new SPI value mapped to an existing SPI, while
   a Locator Type of "1" will only contain a reference to the new SPI.

When a host receives a validated HIP UPDATE with a LOCATOR_SET and
ESP_INFO parameter, it processes the ESP_INFO as follows. The
ESP_INFO parameter indicates whether an SA is being rekeyed, created,
deprecated, or just identified for the benefit of HIP-aware NATs and
firewalls. The host examines the OLD SPI and NEW SPI values in the
ESP_INFO parameter:

1. (no rekeying) If the OLD SPI is equal to the NEW SPI and both
    correspond to an existing SPI, the ESP_INFO is gratuitous
    (provided for HIP-aware NATs and firewalls) and no rekeying is
    necessary.

2. (rekeying) If the OLD SPI indicates an existing SPI and the NEW
    SPI is a different non-zero value, the existing SA is being
    rekeyed and the host follows HIP ESP rekeying procedures by
    creating a new outbound SA with an SPI corresponding to the NEW
    SPI, with no addresses bound to this SPI. Note that locators in
    the LOCATOR_SET parameter will reference this new SPI instead of
    the old SPI.

3. (new SA) If the OLD SPI value is zero and the NEW SPI is a new
    non-zero value, then a new SA is being requested by the peer.
    This case is also treated like a rekeying event; the receiving
    host MUST create a new SA and respond with an UPDATE ACK.

4. (deprecating the SA) If the OLD SPI indicates an existing SPI and
    the NEW SPI is zero, the SA is being deprecated and all locators
    uniquely bound to the SPI are put into the DEPRECATED state.

If none of the above cases apply, a protocol error has occurred and
the processing of the UPDATE is stopped.

Next, the locators in the LOCATOR_SET parameter are processed. For
each locator listed in the LOCATOR_SET parameter, check that the
address therein 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.

The below assumes that all Locators are of Type "1" with a Traffic
Type of "0"; other cases are for further study.

For each Type "1" address listed in the LOCATOR_SET parameter, the
host checks whether the address is already bound to the SPI
indicated. 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
corresponding to the SPI that were NOT listed in the LOCATOR_SET
parameter as DEPRECATED.

As a result, at the end of processing, the addresses listed in the
LOCATOR_SET parameter have a state of either UNVERIFIED or ACTIVE,
and any old addresses on the old SA not listed in the LOCATOR_SET
parameter have a state of DEPRECATED.

Once the host has processed the locators, if the LOCATOR_SET
parameter contains a new preferred locator, the host SHOULD initiate
a change of the preferred locator. This requires that the host first
verify reachability of the associated address, and only then change
the preferred locator; see Section 5.5.

If a host receives a locator with an unsupported Locator Type, and
when such a locator is also declared to be the preferred locator for
the peer, the host SHOULD send a NOTIFY error with a Notify Message
Type of LOCATOR_TYPE_UNSUPPORTED, with the Notification Data field
containing the locator(s) that the receiver failed to process.
Otherwise, a host MAY send a NOTIFY error if a (non-preferred)
locator with an unsupported Locator Type is received in a LOCATOR_SET
parameter.

A host MAY add the source IP address of a received HIP packet as a
candidate locator for the peer even if it is not listed in the peer's
LOCATOR_SET, but it SHOULD prefer locators explicitly listed in the
LOCATOR_SET.

5.4. Verifying Address Reachability

A host MUST verify the reachability of an UNVERIFIED address. The
status of a newly learned address MUST initially be set to UNVERIFIED
unless the new address is advertised in an R1 packet as a new
preferred locator. A host MAY also want to verify the reachability
of an ACTIVE address again after some time, in which case it would
set the status of the address to UNVERIFIED and reinitiate address
verification. A typical verification that is protected by
retransmission timers is to include an ECHO REQUEST within an UPDATE
sent to the new address.

A host typically starts the address-verification procedure by sending
a nonce to the new address. A host MAY choose from different message
exchanges or different nonce values so long as it establishes that
the peer has received and replied to the nonce at the new address.

For example, when the host is changing its SPI and sending an
ESP_INFO to the peer, the NEW SPI value SHOULD be random and the
random value MAY be copied into an ECHO_REQUEST sent in the rekeying
UPDATE. However, if the host is not changing its SPI, it MAY still
use the ECHO_REQUEST parameter for verification but with some other
random value. A host MAY also use other message exchanges as
confirmation of the address reachability.

In some cases, it MAY be sufficient to use the arrival of data on a
newly advertised SA as implicit address reachability verification as
depicted in Figure 7, instead of waiting for the confirmation via a
HIP packet. 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.

```
   Mobile Host                                         Peer Host

              UPDATE(ESP_INFO, LOCATOR_SET, ...)
           ---------------------------------->

                                              prepare incoming SA
              UPDATE(ESP_INFO, ...) with new SPI
           <---------------------------------
 switch to new outgoing SA
                      data on new SA
           ---------------------------------->
                                              mark address ACTIVE
              UPDATE(ACK, ECHO_RESPONSE) later arrives
           ---------------------------------->
```
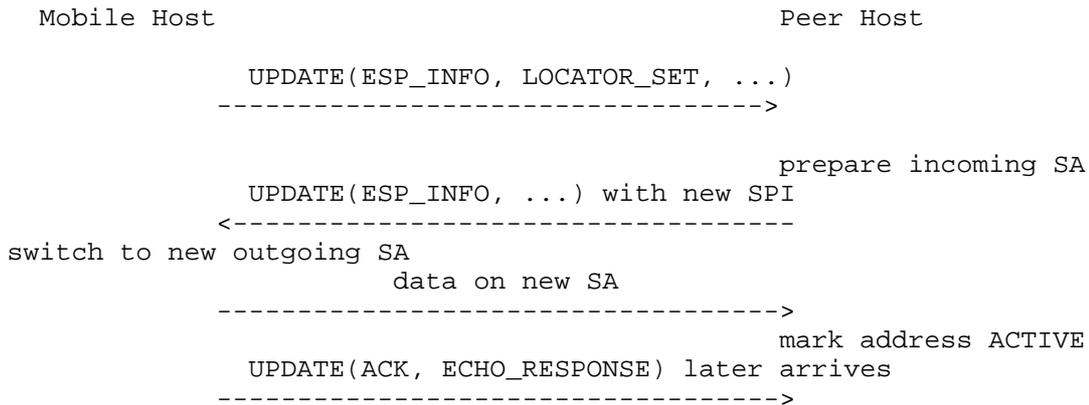
        Figure 7: Address Activation via Use of a New SA

When address verification is in progress for a new preferred locator,
the host SHOULD select a different locator listed as ACTIVE, if one
such locator is available, to continue communications until address
verification completes. Alternatively, the host MAY use the new
preferred locator while in UNVERIFIED status to the extent CBA
permits. CBA is explained in Section 5.6. Once address verification
succeeds, the status of the new preferred locator changes to ACTIVE.

5.5. Changing the Preferred Locator

   A host MAY want to change the preferred outgoing locator 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 may be due to receiving a
   LOCATOR_SET parameter that has the "P" bit set.

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

   1. If the new preferred locator has an ACTIVE status, the preferred
       locator is changed and the procedure succeeds.

   2. If the new preferred locator has an UNVERIFIED status, the host
       starts to verify its reachability. The host SHOULD use a
       different locator listed as ACTIVE until address verification
       completes if one such locator is available. Alternatively, the
       host MAY use the new preferred locator, even though in UNVERIFIED
       status, to the extent CBA permits. Once address verification
       succeeds, the status of the new preferred locator changes to
       ACTIVE, and its use is no longer governed by CBA.

   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 local policy. This case may arise if, for
       example, ICMP error messages that deprecate the preferred locator
       arrive, but the peer has not yet indicated a new preferred
       locator.

   4. If the new preferred locator has a DEPRECATED status and there is
       at least one non-deprecated address, the host selects one of the
       non-deprecated addresses as a new preferred locator and
       continues. If the selected address is UNVERIFIED, the address
       verification procedure described above will apply.

5.6. Credit-Based Authorization

   To prevent redirection-based flooding attacks, the use of a CBA
   approach MUST be used when a host sends data to an UNVERIFIED
   locator. The following algorithm addresses the security
   considerations for prevention of amplification and time-shifting
   attacks. Other forms of credit aging, and other values for the
   CreditAgingFactor and CreditAgingInterval parameters in particular,
   are for further study, and so are the advanced CBA techniques
   specified in [CBA-MIPv6].

5.6.1. Handling Payload Packets

   A host maintains a "credit counter" for each of its peers. Whenever
   a packet arrives from a peer, the host SHOULD increase that peer's
   credit counter by the size of the received packet. When the host has
   a packet to be sent to the peer, and when the peer's preferred
   locator is listed as UNVERIFIED and no alternative locator with
   status ACTIVE is available, the host checks whether it can send the
   packet to the UNVERIFIED locator. The packet SHOULD be sent if the
   value of the credit counter is higher than the size of the outbound
   packet. If the credit counter is too low, the packet MUST be
   discarded or buffered until address verification succeeds. When a
   packet is sent to a peer at an UNVERIFIED locator, the peer's credit
   counter MUST be reduced by the size of the packet. The peer's credit
   counter is not affected by packets that the host sends to an ACTIVE
   locator of that peer.

   Figure 8 depicts the actions taken by the host when a packet is
   received. Figure 9 shows the decision chain in the event a packet is
   sent.

```
        Inbound
        Packet
           |
           |         +----------------+              +---------------+
           |         |    Increase    |              |    Deliver    |
        +-----> | credit counter |-------------> | packet to     |
           |         | by packet size |              | application   |
           |         +----------------+              +---------------+
```
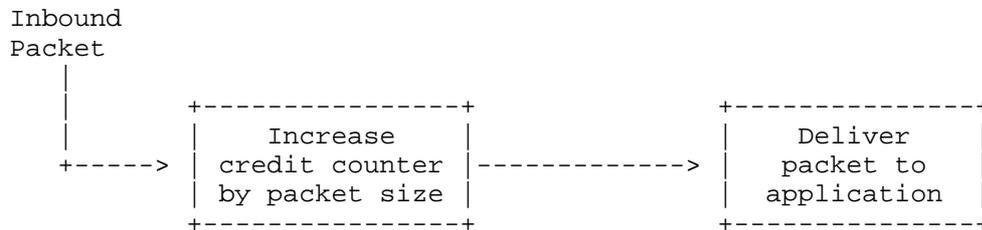
     Figure 8: Receiving Packets with Credit-Based Authorization

```
Outbound
 Packet
   |
   |             _____
   |            /                   \                  +---------------+
   |           /   Is the preferred  \        No       | Send packet   |
 +-----> | destination address |-------------> | to preferred  |
   |           \     UNVERIFIED?      /                 |   address     |
               _____/                   +---------------+
                        |
                        | Yes
                        |
                        v
                _____
               /                   \                   +---------------+
              /   Does an ACTIVE    \        Yes        | Send packet   |
             | destination address |-------------> |  to ACTIVE    |
              \       exist?        /                   |   address     |
               _____/                   +---------------+
                        |
                        | No
                        |
                        v
                _____
               /                   \                   +---------------+
              /  Is credit counter  \        No         |               |
             |          >=         |-------------> | Drop or       |
              \     packet size?    /                   | buffer packet |
               _____/                   +---------------+
                        |
                        | Yes
                        |
                        v
             +---------------+                         +---------------+
             | Reduce credit |                         | Send packet   |
             |  counter by   |----------------> | to preferred  |
             |  packet size  |                         |   address     |
             +---------------+                         +---------------+
```

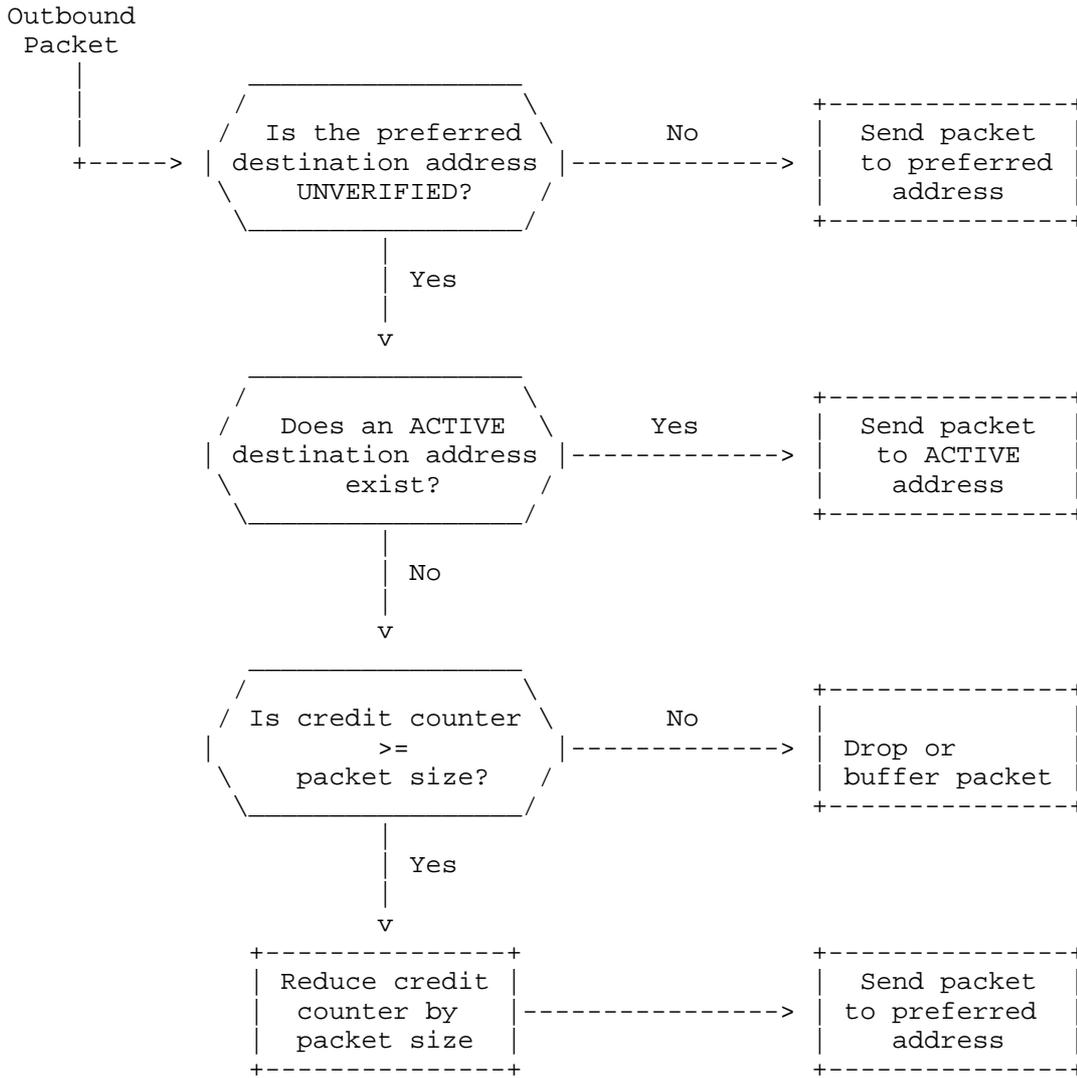   Figure 9: Sending Packets with Credit-Based Authorization

5.6.2. Credit Aging

   A host ensures that the credit counters it maintains for its peers
   gradually decrease over time. Such "credit aging" prevents a
   malicious peer from building up credit at a very slow speed and using
   this, all at once, for a severe burst of redirected packets.

   Credit aging may be implemented by multiplying credit counters with a
   factor, CreditAgingFactor (a fractional value less than one), in
   fixed-time intervals of CreditAgingInterval length. Choosing
   appropriate values for CreditAgingFactor and CreditAgingInterval is
   important to ensure that a host can send packets to an address in
   state UNVERIFIED even when the peer sends at a lower rate than the
   host itself. When CreditAgingFactor or CreditAgingInterval are too
   small, the peer's credit counter might be too low to continue sending
   packets until address verification concludes.

   The parameter values proposed in this document are as follows:

      CreditAgingFactor   7/8
      CreditAgingInterval   5 seconds

   These parameter values work well when the host transfers a file to
   the peer via a TCP connection, and the end-to-end round-trip time
   does not exceed 500 milliseconds. Alternative credit-aging
   algorithms may use other parameter values or different parameters,
   which may even be dynamically established.

6. Security Considerations

   The HIP mobility mechanism provides a secure means of updating a
   host's IP address via HIP UPDATE packets. Upon receipt, a HIP host
   cryptographically verifies the sender of an UPDATE, so forging or
   replaying a HIP UPDATE packet is very difficult (see [RFC7401]).
   Therefore, security issues reside in other attack domains. The two
   we consider are malicious redirection of legitimate connections as
   well as redirection-based flooding attacks using this protocol. This
   can be broken down into the following:

      1) Impersonation attacks

         - direct conversation with the misled victim

         - man-in-the-middle (MitM) attack

      2) Denial-of-service (DoS) attacks

         - flooding attacks (== bandwidth-exhaustion attacks)

            * tool 1: direct flooding

            * tool 2: flooding by botnets

            * tool 3: redirection-based flooding

         - memory-exhaustion attacks

         - computational-exhaustion attacks

      3) Privacy concerns

   We consider these in more detail in the following sections.

   In Sections 6.1 and 6.2, we assume that all users are using HIP. In
   Section 6.3, we consider the security ramifications when we have both
   HIP and non-HIP hosts.

6.1. Impersonation Attacks

   An attacker wishing to impersonate another host will try to mislead
   its victim into directly communicating with them or carry out a MitM
   attack between the victim and the victim's desired communication
   peer. Without mobility support, such attacks are possible only if
   the attacker resides on the routing path between its victim and the
   victim's desired communication peer or if the attacker tricks its
   victim into initiating the connection over an incorrect routing path
   (e.g., by acting as a router or using spoofed DNS entries).

   The HIP extensions defined in this specification change the situation
   in that they introduce an ability to redirect a connection, both
   before and after establishment. If no precautionary measures are
   taken, an attacker could potentially misuse the redirection feature
   to impersonate a victim's peer from any arbitrary location. However,
   the authentication and authorization mechanisms of the HIP base
   exchange [RFC7401] and the signatures in the UPDATE message prevent
   this attack. Furthermore, ownership of a HIP association is securely
   linked to a HIP HI/HIT. If an attacker somehow uses a bug in the
   implementation to redirect a HIP connection, the original owner can
   always reclaim their connection (they can always prove ownership of
   the private key associated with their public HI).

MitM attacks are possible if an on-path attacker is present during
the initial HIP base exchange and if the hosts do not authenticate
each other's identities. However, once such an opportunistic base
exchange has taken place, a MitM attacker that comes later to the
path cannot steal the HIP connection because it is very difficult for
an attacker to create an UPDATE packet (or any HIP packet) that will
be accepted as a legitimate update. UPDATE packets use HMAC and are
signed. Even when an attacker can snoop packets to obtain the SPI
and HIT/HI, they still cannot forge an UPDATE packet without
knowledge of the secret keys. Also, replay attacks on the UPDATE
packet are prevented as described in [RFC7401].

6.2. Denial-of-Service Attacks

6.2.1. Flooding Attacks

   The purpose of a DoS attack is to exhaust some resource of the victim
   such that the victim ceases to operate correctly. A DoS attack can
   aim at the victim's network attachment (flooding attack), its memory,
   or its processing capacity. In a flooding attack, the attacker
   causes an excessive number of bogus or unwanted packets to be sent to
   the victim, which fills their available bandwidth. Note that the
   victim does not necessarily need to be a node; it can also be an
   entire network. The attack functions the same way in either case.

   An effective DoS strategy is distributed denial of service (DDoS).
   Here, the attacker conventionally distributes some viral software to
   as many nodes as possible. Under the control of the attacker, the
   infected nodes (e.g., nodes in a botnet) jointly send packets to the
   victim. With such an "army", an attacker can take down even very
   high bandwidth networks/victims.

   With the ability to redirect connections, an attacker could realize a
   DDoS attack without having to distribute viral code. Here, the
   attacker initiates a large download from a server and subsequently
   uses the HIP mobility mechanism to redirect this download to its
   victim. The attacker can repeat this with multiple servers. This
   threat is mitigated through reachability checks and CBA. When
   conducted using HIP, reachability checks can leverage the built-in
   authentication properties of HIP. They can also prevent redirection-
   based flooding attacks. However, the delay of such a check can have
   a noticeable impact on application performance. To reduce the impact
   of the delay, CBA can be used to send a limited number of packets to
   the new address while the validity of the IP address is still in
   question. Both strategies do not eliminate flooding attacks per se,
   but they preclude: (i) their use from a location off the path towards
   the flooded victim; and (ii) any amplification in the number and size

of the redirected packets. As a result, the combination of a
reachability check and CBA lowers a HIP redirection-based flooding
attack to the level of a direct flooding attack in which the attacker
itself sends the flooding traffic to the victim.

6.2.2. Memory/Computational-Exhaustion DoS Attacks

We now consider whether or not the proposed extensions to HIP add any
new DoS attacks (consideration of DoS attacks using the base HIP
exchange and updates is discussed in [RFC7401]). A simple attack is
to send many UPDATE packets containing many IP addresses that are not
flagged as preferred. The attacker continues to send such packets
until the number of IP addresses associated with the attacker's HI
crashes the system. Therefore, a HIP association SHOULD limit the
number of IP addresses that can be associated with any HI. Other
forms of memory/computationally exhausting attacks via the HIP UPDATE
packet are handled in the base HIP document [RFC7401].

A central server that has to deal with a large number of mobile
clients MAY consider increasing the SA lifetimes to try to slow down
the rate of rekeying UPDATEs or increasing the cookie difficulty to
slow down the rate of attack-oriented connections.

6.3. Mixed Deployment Environment

We now assume an environment with hosts that are both HIP and non-HIP
aware. Four cases exist:

1. A HIP host redirects its connection onto a non-HIP host. The
    non-HIP host will drop the reachability packet, so this is not a
    threat unless the HIP host is a MitM that could somehow respond
    successfully to the reachability check.

2. A non-HIP host attempts to redirect their connection onto a HIP
    host. This falls into IPv4 and IPv6 security concerns, which are
    outside the scope of this document.

3. A non-HIP host attempts to steal a HIP host's session (assume
    that Secure Neighbor Discovery is not active for the following).
    The non-HIP host contacts the service that a HIP host has a
    connection with and then attempts to change its IP address to
    steal the HIP host's connection. What will happen in this case
    is implementation dependent, but such a request should fail by
    being ignored or dropped. Even if the attack were successful,
    the HIP host could reclaim its connection via HIP.

   4. A HIP host attempts to steal a non-HIP host's session. A HIP
       host could spoof the non-HIP host's IP address during the base
       exchange or set the non-HIP host's IP address as its preferred
       address via an UPDATE. Other possibilities exist, but a solution
       is to prevent the local redirection of sessions that were
       previously using an unverified address, but outside of the
       existing HIP context, into the HIP SAs until the address change
       can be verified.

6.4. Privacy Concerns

   The exposure of a host's IP addresses through HIP mobility extensions
   may raise privacy concerns. The administrator of a host may be
   trying to hide its location in some context through the use of a VPN
   or other virtual interfaces. Similar privacy issues also arise in
   other frameworks such as WebRTC and are not specific to HIP.
   Implementations SHOULD provide a mechanism to allow the host
   administrator to block the exposure of selected addresses or address
   ranges. While this issue may be more relevant in a host multihoming
   scenario in which multiple IP addresses might be exposed [RFC8047],
   it is worth noting also here that mobility events might cause an
   implementation to try to inadvertently use a locator that the
   administrator would rather avoid exposing to the peer host.

7. IANA Considerations

   [RFC5206], obsoleted by this document, specified an allocation for a
   LOCATOR parameter in the "Parameter Types" subregistry of the "Host
   Identity Protocol (HIP) Parameters" registry, with a type value of
   193. IANA has renamed the parameter to "LOCATOR_SET" and has updated
   the reference from [RFC5206] to this specification.

   [RFC5206], obsoleted by this document, specified an allocation for a
   LOCATOR_TYPE_UNSUPPORTED type in the "Notify Message Types" registry,
   with a type value of 46. IANA has updated the reference from
   [RFC5206] to this specification.

8. Differences from RFC 5206

   This section summarizes the technical changes made from [RFC5206].
   This section is informational, intended to help implementors of the
   previous protocol version. If any text in this section contradicts
   text in other portions of this specification, the text found outside
   of this section should be considered normative.

This document specifies extensions to the HIP Version 2 protocol,
while [RFC5206] specifies extensions to the HIP Version 1 protocol.
[RFC7401] documents the differences between these two protocol
versions.

[RFC5206] included procedures for both HIP host mobility and basic
host multihoming. In this document, only host mobility procedures
are included; host multihoming procedures are now specified in
[RFC8047]. In particular, multihoming-related procedures related to
the exposure of multiple locators in the base exchange packets; the
transmission, reception, and processing of multiple locators in a
single UPDATE packet; handovers across IP address families; and other
multihoming-related specifications have been removed.

The following additional changes have been made:

o The LOCATOR parameter in [RFC5206] has been renamed to
   LOCATOR_SET.

o Specification text regarding the handling of mobility when both
   hosts change IP addresses at nearly the same time (a "double-jump"
   mobility scenario) has been added.

o Specification text regarding the mobility event in which the host
   briefly has an active new locator and old locator at the same time
   (a "make-before-break" mobility scenario) has been added.

o Specification text has been added to note that a host may add the
   source IP address of a received HIP packet as a candidate locator
   for the peer even if it is not listed in the peer's LOCATOR_SET,
   but that it should prefer locators explicitly listed in the
   LOCATOR_SET.

o This document clarifies that the HOST_ID parameter may be included
   in UPDATE messages containing LOCATOR_SET parameters, for the
   possible benefit of HIP-aware firewalls.

o The previous specification mentioned that it may be possible to
   include multiple LOCATOR_SET and ESP_INFO parameters in an UPDATE.
   This document only specifies the case of a single LOCATOR_SET and
   ESP_INFO parameter in an UPDATE.

o The previous specification mentioned that it may be possible to
   send LOCATOR_SET parameters in packets other than the UPDATE.
   This document only specifies the use of the UPDATE packet.

o This document describes a simple heuristic for setting the credit
   value for CBA.

   o This specification mandates that a host must be able to receive
      and avoid reprocessing redundant LOCATOR_SET parameters that may
      have been sent in parallel to multiple addresses of the host.

9. References

9.1. Normative References

   [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
              Requirement Levels", BCP 14, RFC 2119,
              DOI 10.17487/RFC2119, March 1997,
              <http://www.rfc-editor.org/info/rfc2119>.

   [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing
              Architecture", RFC 4291, DOI 10.17487/RFC4291, February
              2006, <http://www.rfc-editor.org/info/rfc4291>.

   [RFC7401] Moskowitz, R., Ed., Heer, T., Jokela, P., and T.
              Henderson, "Host Identity Protocol Version 2 (HIPv2)",
              RFC 7401, DOI 10.17487/RFC7401, April 2015,
              <http://www.rfc-editor.org/info/rfc7401>.

   [RFC7402] Jokela, P., Moskowitz, R., and J. Melen, "Using the
              Encapsulating Security Payload (ESP) Transport Format with
              the Host Identity Protocol (HIP)", RFC 7402,
              DOI 10.17487/RFC7402, April 2015,
              <http://www.rfc-editor.org/info/rfc7402>.

   [RFC8003] Laganier, J. and L. Eggert, "Host Identity Protocol (HIP)
              Registration Extension", RFC 8003, DOI 10.17487/RFC8003,
              October 2016, <http://www.rfc-editor.org/info/rfc8003>.

   [RFC8004] Laganier, J. and L. Eggert, "Host Identity Protocol (HIP)
              Rendezvous Extension", RFC 8004, DOI 10.17487/RFC8004,
              October 2016, <http://www.rfc-editor.org/info/rfc8004>.

9.2. Informative References

   [CBA-MIPv6]
              Vogt, C. and J. Arkko, "Credit-Based Authorization for
              Mobile IPv6 Early Binding Updates", Work in Progress,
              draft-vogt-mobopts-credit-based-authorization-00, February
              2005.

   [RFC4225] Nikander, P., Arkko, J., Aura, T., Montenegro, G., and E.
              Nordmark, "Mobile IP Version 6 Route Optimization Security
              Design Background", RFC 4225, DOI 10.17487/RFC4225,
              December 2005, <http://www.rfc-editor.org/info/rfc4225>.

   [RFC5206] Nikander, P., Henderson, T., Ed., Vogt, C., and J. Arkko,
              "End-Host Mobility and Multihoming with the Host Identity
              Protocol", RFC 5206, DOI 10.17487/RFC5206, April 2008,
              <http://www.rfc-editor.org/info/rfc5206>.

   [RFC5207] Stiemerling, M., Quittek, J., and L. Eggert, "NAT and
              Firewall Traversal Issues of Host Identity Protocol (HIP)
              Communication", RFC 5207, DOI 10.17487/RFC5207, April
              2008, <http://www.rfc-editor.org/info/rfc5207>.

   [RFC8047] Henderson, T., Ed., Vogt, C., and J. Arkko, "Host
              Multihoming with the Host Identity Protocol", RFC 8047,
              DOI 10.17487/RFC8047, February 2017,
              <http://www.rfc-editor.org/info/rfc8047>.

   [SIMPLE-CBA]
              Vogt, C. and J. Arkko, "Credit-Based Authorization for
              Concurrent Reachability Verification", Work in Progress,
              draft-vogt-mobopts-simple-cba-00, February 2006.

Authors' Addresses

   Thomas R. Henderson (editor)
   University of Washington
   Campus Box 352500
   Seattle, WA
   United States of America

   Email: tomhend@u.washington.edu

   Christian Vogt
   Independent
   3473 North First Street
   San Jose, CA 95134
   United States of America

   Email: mail@christianvogt.net

   Jari Arkko
   Ericsson
   Jorvas, FIN-02420
   Finland

   Phone: +358 40 5079256
   Email: jari.arkko@piuha.net