Network Working Group   R. Moskowitz, Ed.
Internet-Draft   HTT Consulting
Obsoletes: 4423 (if approved)   M. Komu
Intended status: Informational   Ericsson
Expires: August 19, 2017   February 15, 2017

                    Host Identity Protocol Architecture
                      draft-ietf-hip-rfc4423-bis-16

Abstract

   This memo describes a new namespace, the Host Identity namespace, and
   a new protocol layer, the Host Identity Protocol, between the
   internetworking and transport layers. Herein are presented the
   basics of the current namespaces, their strengths and weaknesses, and
   how a new namespace will add completeness to them. The roles of this
   new namespace in the protocols are defined.

   This document obsoletes RFC 4423 and addresses the concerns raised by
   the IESG, particularly that of crypto agility. It incorporates
   lessons learned from the implementations of RFC 5201 and goes further
   to explain how HIP works as a secure signaling channel.

Status of This Memo

   This Internet-Draft is submitted in full conformance with the
   provisions of BCP 78 and BCP 79.

   Internet-Drafts are working documents of the Internet Engineering
   Task Force (IETF). Note that other groups may also distribute
   working documents as Internet-Drafts. The list of current Internet-
   Drafts is at http://datatracker.ietf.org/drafts/current/.

   Internet-Drafts are draft documents valid for a maximum of six months
   and may be updated, replaced, or obsoleted by other documents at any
   time. It is inappropriate to use Internet-Drafts as reference
   material or to cite them other than as "work in progress."

   This Internet-Draft will expire on August 19, 2017.

Copyright Notice

Table of Contents

1. Introduction

   The Internet has two important global namespaces: Internet Protocol
   (IP) addresses and Domain Name Service (DNS) names. These two
   namespaces have a set of features and abstractions that have powered
   the Internet to what it is today. They also have a number of
   weaknesses. Basically, since they are all we have, we try and do too
   much with them. Semantic overloading and functionality extensions
   have greatly complicated these namespaces.

   The proposed Host Identity namespace fills an important gap between
   the IP and DNS namespaces. A Host Identity conceptually refers to a
   computing platform, and there may be multiple such Host Identities
   per computing platform (because the platform may wish to present a
   different identity to different communicating peers). The Host
   Identity namespace consists of Host Identifiers (HI). There is

exactly one Host Identifier for each Host Identity (although there
may be transient periods of time such as key replacement when more
than one identifier may be active). While this text later talks
about non-cryptographic Host Identifiers, the architecture focuses on
the case in which Host Identifiers are cryptographic in nature.
Specifically, the Host Identifier is the public key of an asymmetric
key-pair. Each Host Identity uniquely identifies a single host,
i.e., no two hosts have the same Host Identity. If two or more
computing platforms have the same Host Identifier, then they are
instantiating a distributed host. The Host Identifier can either be
public (e.g. published in the DNS), or unpublished. Client systems
will tend to have both public and unpublished Host Identifiers.

   There is a subtle but important difference between Host Identities
   and Host Identifiers. An Identity refers to the abstract entity that
   is identified. An Identifier, on the other hand, refers to the
   concrete bit pattern that is used in the identification process.

   Although the Host Identifiers could be used in many authentication
   systems, such as IKEv2 [RFC4306], the presented architecture
   introduces a new protocol, called the Host Identity Protocol (HIP),
   and a cryptographic exchange, called the HIP base exchange; see also
   Section 6. HIP provides for limited forms of trust between systems,
   enhance mobility, multi-homing and dynamic IP renumbering, aid in
   protocol translation / transition, and reduce certain types of
   denial-of-service (DoS) attacks.

   When HIP is used, the actual payload traffic between two HIP hosts is
   typically, but not necessarily, protected with ESP [RFC7402]. The
   Host Identities are used to create the needed ESP Security
   Associations (SAs) and to authenticate the hosts. When ESP is used,
   the actual payload IP packets do not differ in any way from standard
   ESP protected IP packets.

   Much has been learned about HIP [RFC6538] since [RFC4423] was
   published. This document expands Host Identities beyond use to
   enable IP connectivity and security to general interhost secure
   signalling at any protocol layer. The signal may establish a
   security association between the hosts, or simply pass information
   within the channel.

2. Terminology

```
    +--------------+-----------------------------------------------+
    | Term         | Explanation                                   |
    +--------------+-----------------------------------------------+
    | Public key   | The public key of an asymmetric cryptographic key |
    |              | pair.  Used as a publicly known identifier for |
    |              | cryptographic identity authentication.  Public is |
    |              | a a relative term here, ranging from "known to |
    |              | peers only" to "known to the world."          |
    | Private key  | The private or secret key of an asymmetric    |
    |              | cryptographic key pair.  Assumed to be known only |
    |              | to the party identified by the corresponding  |
    |              | public key.  Used by the identified party to  |
    |              | authenticate its identity to other parties.   |
    | Public key   | An asymmetric cryptographic key pair consisting |
    | pair         | of public and private keys.  For example, Rivest- |
    |              | Shamir-Adleman (RSA), Digital Signature Algorithm |
    |              | (DSA) and Elliptic Curve DSA (ECDSA) key pairs |
    |              | are such key pairs.                            |
    | End-point    | A communicating entity.  For historical reasons, |
    |              | the term 'computing platform' is used in this |
    |              | document as a (rough) synonym for end-point.   |
    +--------------+-----------------------------------------------+
```

2.2. Terms specific to this and other HIP documents

   It should be noted that many of the terms defined herein are
   tautologous, self-referential or defined through circular reference
   to other terms. This is due to the succinct nature of the
   definitions. See the text elsewhere in this document and the base
   specification [RFC7401] for more elaborate explanations.

   +---------------+--+
   | Term | Explanation |
   +---------------+--+
Computing	An entity capable of communicating and computing,
platform	for example, a computer. See the definition of
	'End-point', above.
HIP base	A cryptographic protocol; see also Section 6.
exchange	
HIP packet	An IP packet that carries a 'Host Identity
	Protocol' message.
Host Identity	An abstract concept assigned to a 'computing
	platform'. See 'Host Identifier', below.
Host Identity	A name space formed by all possible Host
namespace	Identifiers.
Host Identity	A protocol used to carry and authenticate Host

```
| Protocol       | Identifiers and other information.            |
| Host Identity  | The cryptographic hash used in creating the Host |
| Hash           | Identity Tag from the Host Identity.          |
| Host Identity  | A 128-bit datum created by taking a cryptographic |
| Tag            | hash over a Host Identifier plus bits to identify |
|                | which hash used.                              |
| Host           | A public key used as a name for a Host Identity. |
| Identifier     |                                               |
| Local Scope    | A 32-bit datum denoting a Host Identity.      |
| Identifier     |                                               |
| Public Host    | A published or publicly known Host Identifier |
| Identifier     | used as a public name for a Host Identity, and |
| and Identity   | the corresponding Identity.                   |
| Unpublished    | A Host Identifier that is not placed in any   |
| Host           | public directory, and the corresponding Host  |
| Identifier     | Identity.  Unpublished Host Identities are     |
| and Identity   | typically short lived in nature, being often  |
|                | replaced and possibly used just once.         |
| Rendezvous     | A mechanism used to locate mobile hosts based on |
| Mechanism      | their HIT.                                     |
+--------------+-------------------------------------------------+
```

3. Background

The Internet is built from three principal components: computing
platforms (end-points), packet transport (i.e., internetworking)
infrastructure, and services (applications). The Internet exists to
service two principal components: people and robotic services
(silicon-based people, if you will). All these components need to be
named in order to interact in a scalable manner. Here we concentrate
on naming computing platforms and packet transport elements.

There are two principal namespaces in use in the Internet for these
components: IP addresses, and Domain Names. Domain Names provide
hierarchically assigned names for some computing platforms and some
services. Each hierarchy is delegated from the level above; there is
no anonymity in Domain Names. Email, HTTP, and SIP addresses all
reference Domain Names.

The IP addressing namespace has been overloaded to name both

interfaces (at layer-3) and endpoints (for the endpoint-specific part
of layer-3, and for layer-4). In their role as interface names, IP
addresses are sometimes called "locators" and serve as an endpoint
within a routing topology.

IP addresses are numbers that name networking interfaces, and
typically only when the interface is connected to the network.
Originally, IP addresses had long-term significance. Today, the vast
number of interfaces use ephemeral and/or non-unique IP addresses.
That is, every time an interface is connected to the network, it is
assigned an IP address.

In the current Internet, the transport layers are coupled to the IP
addresses. Neither can evolve separately from the other. IPng
deliberations were strongly shaped by the decision that a
corresponding TCPng would not be created.

There are three critical deficiencies with the current namespaces.
Firstly, dynamic readdressing cannot be directly managed. Secondly,
confidentiality is not provided in a consistent, trustable manner.
Finally, authentication for systems and datagrams is not provided.
All of these deficiencies arise because computing platforms are not
well named with the current namespaces.

3.1. A desire for a namespace for computing platforms

An independent namespace for computing platforms could be used in
end-to-end operations independent of the evolution of the
internetworking layer and across the many internetworking layers.
This could support rapid readdressing of the internetworking layer
because of mobility, rehoming, or renumbering.

If the namespace for computing platforms is based on public-key
cryptography, it can also provide authentication services. If this
namespace is locally created without requiring registration, it can
provide anonymity.

Such a namespace (for computing platforms) and the names in it should
have the following characteristics:

o The namespace should be applied to the IP 'kernel' or stack. The

IP stack is the 'component' between applications and the packet
      transport infrastructure.

   o The namespace should fully decouple the internetworking layer from
      the higher layers. The names should replace all occurrences of IP
      addresses within applications (like in the Transport Control
      Block, TCB). This replacement can be handled transparently for
      legacy applications as the LSIs and HITs are compatible with IPv4
      and IPv6 addresses [RFC5338]. However, HIP-aware applications
      require some modifications from the developers, who may employ
      networking API extensions for HIP [RFC6317].

   o The introduction of the namespace should not mandate any
      administrative infrastructure. Deployment must come from the
      bottom up, in a pairwise deployment.

   o The names should have a fixed length representation, for easy
      inclusion in datagram headers and existing programming interfaces
      (e.g the TCB).

   o Using the namespace should be affordable when used in protocols.
      This is primarily a packet size issue. There is also a
      computational concern in affordability.

   o Name collisions should be avoided as much as possible. The
      mathematics of the birthday paradox can be used to estimate the
      chance of a collision in a given population and hash space. In
      general, for a random hash space of size n bits, we would expect
      to obtain a collision after approximately $1.2*sqrt(2**n)$ hashes
      were obtained. For 64 bits, this number is roughly 4 billion. A
      hash size of 64 bits may be too small to avoid collisions in a
      large population; for example, there is a 1% chance of collision
      in a population of 640M. For 100 bits (or more), we would not
      expect a collision until approximately $2**50$ (1 quadrillion)
      hashes were generated.

   o The names should have a localized abstraction so that they can be
      used in existing protocols and APIs.

   o It must be possible to create names locally. When such names are
      not published, this can provide anonymity at the cost of making
      resolvability very difficult.

   o The namespace should provide authentication services.

   o The names should be long lived, but replaceable at any time. This
      impacts access control lists; short lifetimes will tend to result

in tedious list maintenance or require a namespace infrastructure
for central control of access lists.

In this document, a new namespace approaching these ideas is called
the Host Identity namespace. Using Host Identities requires its own
protocol layer, the Host Identity Protocol, between the
internetworking and transport layers. The names are based on public-
key cryptography to supply authentication services. Properly
designed, it can deliver all of the above stated requirements.

4. Host Identity namespace

A name in the Host Identity namespace, a Host Identifier (HI),
represents a statistically globally unique name for naming any system
with an IP stack. This identity is normally associated with, but not
limited to, an IP stack. A system can have multiple identities, some
'well known', some unpublished or 'anonymous'. A system may self-
assert its own identity, or may use a third-party authenticator like
DNSSEC [RFC2535], PGP, or X.509 to 'notarize' the identity assertion
to another namespace. It is expected that the Host Identifiers will
initially be authenticated with DNSSEC and that all implementations
will support DNSSEC as a minimal baseline.

In theory, any name that can claim to be 'statistically globally
unique' may serve as a Host Identifier. In the HIP architecture, the
public key of a private-public key pair has been chosen as the Host
Identifier because it can be self managed and it is computationally
difficult to forge. As specified in the Host Identity Protocol
[RFC7401] specification, a public-key-based HI can authenticate the
HIP packets and protect them from man-in-the-middle attacks. Since
authenticated datagrams are mandatory to provide much of HIP's
denial-of-service protection, the Diffie-Hellman exchange in HIP base
exchange has to be authenticated. Thus, only public-key HI and
authenticated HIP messages are supported in practice.

In this document, the non-cryptographic forms of HI and HIP are
presented to complete the theory of HI, but they should not be
implemented as they could produce worse denial-of-service attacks
than the Internet has without Host Identity. There has been past
research in challenge puzzles to use non-cryptographic HI, for Radio
Frequency IDentification (RFID), in an HIP exchange tailored to the
workings of such challenges (as described further in [urien-rfid] and
[urien-rfid-draft]).

4.1. Host Identifiers

   Host Identity adds two main features to Internet protocols. The
   first is a decoupling of the internetworking and transport layers;
   see Section 5. This decoupling will allow for independent evolution
   of the two layers. Additionally, it can provide end-to-end services
   over multiple internetworking realms. The second feature is host
   authentication. Because the Host Identifier is a public key, this
   key can be used for authentication in security protocols like ESP.

   The only completely defined structure of the Host Identity is that of
   a public/private key pair. In this case, the Host Identity is
   referred to by its public component, the public key. Thus, the name
   representing a Host Identity in the Host Identity namespace, i.e.,
   the Host Identifier, is the public key. In a way, the possession of
   the private key defines the Identity itself. If the private key is
   possessed by more than one node, the Identity can be considered to be
   a distributed one.

   Architecturally, any other Internet naming convention might form a
   usable base for Host Identifiers. However, non-cryptographic names
   should only be used in situations of high trust - low risk. That is
   any place where host authentication is not needed (no risk of host
   spoofing) and no use of ESP. However, at least for interconnected
   networks spanning several operational domains, the set of
   environments where the risk of host spoofing allowed by non-
   cryptographic Host Identifiers is acceptable is the null set. Hence,
   the current HIP documents do not specify how to use any other types
   of Host Identifiers but public keys. For instance, Back-to-My-Mac
   [RFC6281] from Apple comes pretty close to the functionality of HIP,
   but unlike HIP, it is based on non-cryptographic identifiers.

   The actual Host Identifiers are never directly used at the transport
   or network layers. The corresponding Host Identifiers (public keys)
   may be stored in various DNS or other directories as identified
   elsewhere in this document, and they are passed in the HIP base
   exchange. A Host Identity Tag (HIT) is used in other protocols to
   represent the Host Identity. Another representation of the Host
   Identities, the Local Scope Identifier (LSI), can also be used in

protocols and APIs.

4.2. Host Identity Hash (HIH)

The Host Identity Hash is the cryptographic hash algorithm used in
producing the HIT from the HI. It is also the hash used throughout
the HIP protocol for consistency and simplicity. It is possible to
for the two hosts in the HIP exchange to use different hash
algorithms.

Multiple HIHs within HIP are needed to address the moving target of
creation and eventual compromise of cryptographic hashes. This
significantly complicates HIP and offers an attacker an additional
downgrade attack that is mitigated in the HIP protocol [RFC7401].

4.3. Host Identity Tag (HIT)

A Host Identity Tag is a 128-bit representation for a Host Identity.
It is created from an HIH, an IPv6 prefix [RFC7343] and a hash
identifier. There are two advantages of using the HIT over using the
Host Identifier in protocols. Firstly, its fixed length makes for
easier protocol coding and also better manages the packet size cost
of this technology. Secondly, it presents the identity in a
consistent format to the protocol independent of the cryptographic
algorithms used.

In essence, the HIT is a hash over the public key. As such, two
algorithms affect the generation of a HIT: the public-key algorithm
of the HI and the used HIH. The two algorithms are encoded in the
bit presentation of the HIT. As the two communicating parties may
support different algorithms, [RFC7401] defines the minimum set for
interoperability. For further interoperability, the responder may
store its keys in DNS records, and thus the initiator may have to
couple destination HITs with appropriate source HITs according to
matching HIH.

In the HIP packets, the HITs identify the sender and recipient of a
packet. Consequently, a HIT should be unique in the whole IP
universe as long as it is being used. In the extremely rare case of
a single HIT mapping to more than one Host Identity, the Host
Identifiers (public keys) will make the final difference. If there
is more than one public key for a given node, the HIT acts as a hint

for the correct public key to use.

4.4. Local Scope Identifier (LSI)

An LSI is a 32-bit localized representation for a Host Identity. The
purpose of an LSI is to facilitate using Host Identities in existing
APIs for IPv4-based applications. Besides facilitating HIP-based
connectivity for legacy IPv4 applications, the LSIs are beneficial in
two other scenarios [RFC6538].

In the first scenario, two IPv4-only applications are residing on two
separate hosts connected by IPv6-only network. With HIP-based
connectivity, the two applications are able to communicate despite of
the mismatch in the protocol families of the applications and the
underlying network. The reason is that the HIP layer translates the

LSIs originating from the upper layers into routable IPv6 locators
before delivering the packets on the wire.

The second scenario is the same as the first one, but with the
difference that one of the applications supports only IPv6. Now two
obstacles hinder the communication between the application: the
addressing families of the two applications differ, and the
application residing at the IPv4-only side is again unable to
communicate because of the mismatch between addressing families of
the application (IPv4) and network (IPv6). With HIP-based
connectivity for applications, this scenario works; the HIP layer can
choose whether to translate the locator of an incoming packet into an
LSI or HIT.

Effectively, LSIs improve IPv6 interoperability at the network layer
as described in the first scenario and at the application layer as
depicted in the second example. The interoperability mechanism
should not be used to avoid transition to IPv6; the authors firmly
believe in IPv6 adoption and encourage developers to port existing
IPv4-only applications to use IPv6. However, some proprietary,
closed-source, IPv4-only applications may never see the daylight of
IPv6, and the LSI mechanism is suitable for extending the lifetime of
such applications even in IPv6-only networks.

The main disadvantage of an LSI is its local scope. Applications may

violate layering principles and pass LSIs to each other in
application-layer protocols. As the LSIs are valid only in the
context of the local host, they may represent an entirely different
host when passed to another host. However, it should be emphasized
here that the LSI concept is effectively a host-based NAT and does
not introduce any more issues than the prevalent middlebox based NATs
for IPv4. In other words, the applications violating layering
principles are already broken by the NAT boxes that are ubiquitously
deployed.

4.5. Storing Host Identifiers in directories

The public Host Identifiers should be stored in DNS; the unpublished
Host Identifiers should not be stored anywhere (besides the
communicating hosts themselves). The (public) HI along with the
supported HIHs are stored in a new RR type. This RR type is defined
in HIP DNS Extension [RFC8005].

Alternatively, or in addition to storing Host Identifiers in the DNS,
they may be stored in various other directories. For instance, a
directory based on the Lightweight Directory Access Protocol (LDAP)
or a Public Key Infrastructure (PKI) [I-D.ietf-hip-rfc6253-bis] may
be used. Alternatively, Distributed Hash Tables (DHTs) [RFC6537]

have successfully been utilized [RFC6538]. Such a practice may allow
them to be used for purposes other than pure host identification.

Some types of applications may cache and use Host Identifiers
directly, while others may indirectly discover them through symbolic
host name (such as FQDN) look up from a directory. Even though Host
Identities can have a substantially longer lifetime associated with
them than routable IP addresses, directories may be a better approach
to manage the lifespan of Host Identities. For example, an LDAP-
based directory or DHT can be used for locally published identities
whereas DNS can be more suitable for public advertisement.

5. New stack architecture

One way to characterize Host Identity is to compare the proposed new
architecture with the current one. Using the terminology from the
IRTF Name Space Research Group Report [nsrg-report] and, e.g., the
unpublished Internet-Draft Endpoints and Endpoint Names

[chiappa-endpoints], the IP addresses currently embody the dual role
of locators and end-point identifiers. That is, each IP address
names a topological location in the Internet, thereby acting as a
routing direction vector, or locator. At the same time, the IP
address names the physical network interface currently located at the
point-of-attachment, thereby acting as a end-point name.

In the HIP architecture, the end-point names and locators are
separated from each other. IP addresses continue to act as locators.
The Host Identifiers take the role of end-point identifiers. It is
important to understand that the end-point names based on Host
Identities are slightly different from interface names; a Host
Identity can be simultaneously reachable through several interfaces.

The difference between the bindings of the logical entities are
illustrated in Figure 1. Left side illustrates the current TCP/IP
architecture and right side the HIP-based architecture.

```
   Transport ---- Socket              Transport ------ Socket
   association       |                association         |
                     |                                    |
                     |                                    |
                     |                                    |
   End-point         |                End-point --- Host Identity
          \          |                                    |
           \         |                                    |
            \        |                                    |
             \ |                                          |
   Location --- IP address            Location --- IP address
```
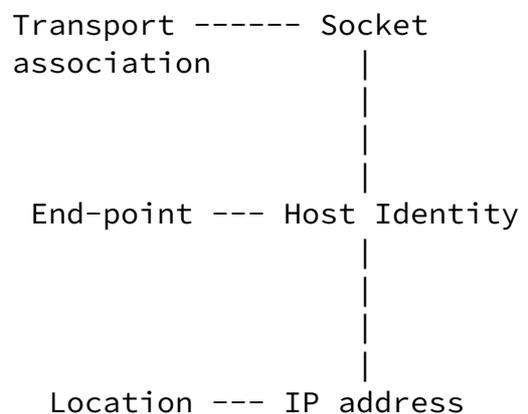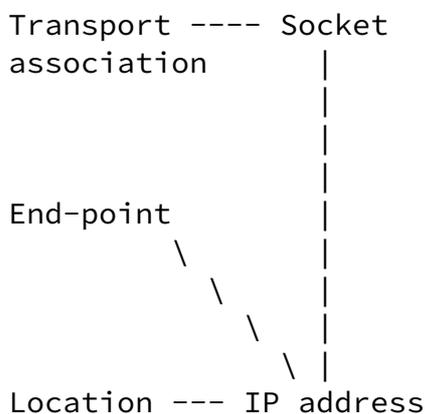
Figure 1

Architecturally, HIP provides for a different binding of transport-
layer protocols. That is, the transport-layer associations, i.e.,
TCP connections and UDP associations, are no longer bound to IP
addresses but rather to Host Identities. In practice, the Host
Identities are exposed as LSIs and HITs for legacy applications and
the transport layer to facilitate backward compatibility with
existing networking APIs and stacks.

5.1. On the multiplicity of identities

A host may have multiple identities both at the client and server
side. This raises some additional concerns that are addressed in
this section.

For security reasons, it may be a bad idea to duplicate the same Host
Identity on multiple hosts because the compromise of a single host
taints the identities of the other hosts. Management of machines
with identical Host Identities may also present other challenges and,
therefore, it is advisable to have a unique identity for each host.

Instead of duplicating identities, HIP opportunistic mode can be
employed, where the initiator leaves out the identifier of the
responder when initiating the key exchange and learns it upon the
completion of the exchange. The tradeoffs are related to lowered
security guarantees, but a benefit of the approach is to avoid
publishing of Host Identifiers in any directories [komu-leap]. The
approach could also be used for load balancing purposes at the HIP
layer because the identity of the responder can be decided
dynamically during the key exchange. Thus, the approach has the
potential to be used as a HIP-layer "anycast", either directly
between two hosts or indirectly through the rendezvous service
[komu-diss].

At the client side, a host may have multiple Host Identities, for
instance, for privacy purposes. Another reason can be that the
person utilizing the host employs different identities for different
administrative domains as an extra security measure. If a HIP-aware

middlebox, such as a HIP-based firewall, is on the path between the
client and server, the user or the underlying system should carefully
choose the correct identity to avoid the firewall to unnecessarily
drop HIP-base connectivity [komu-diss].

Similarly, a server may have multiple Host Identities. For instance,
a single web server may serve multiple different administrative
domains. Typically, the distinction is accomplished based on the DNS
name, but also the Host Identity could be used for this purpose.
However, a more compelling reason to employ multiple identities are
HIP-aware firewalls that are unable see the HTTP traffic inside the
encrypted IPsec tunnel. In such a case, each service could be
configured with a separate identity, thus allowing the firewall to
segregate the different services of the single web server from each
other [lindqvist-enterprise].

6. Control plane

HIP decouples control and data plane from each other. Two end-hosts
initialize the control plane using a key exchange procedure called
the base exchange. The procedure can be assisted by new
infrastructural intermediaries called rendezvous or relay servers.
In the event of IP address changes, the end-hosts sustain control
plane connectivity with mobility and multihoming extensions.
Eventually, the end-hosts terminate the control plane and remove the
associated state.

6.1. Base exchange

The base exchange is a key exchange procedure that authenticates the
initiator and responder to each other using their public keys.
Typically, the initiator is the client-side host and the responder is
the server-side host. The roles are used by the state machine of a
HIP implementation, but discarded upon successful completion.

The exchange consists of four messages during which the hosts also
create symmetric keys to protect the control plane with Hash-based
message authentication codes (HMACs). The keys can be also used to
protect the data plane, and IPsec ESP [RFC7402] is typically used as
the data-plane protocol, albeit HIP can also accommodate others.
Both the control and data plane are terminated using a closing
procedure consisting of two messages.

In addition, the base exchange also includes a computational puzzle
[RFC7401] that the initiator must solve. The responder chooses the
difficulty of the puzzle which permits the responder to delay new
incoming initiators according to local policies, for instance, when
the responder is under heavy load. The puzzle can offer some
resiliency against DoS attacks because the design of the puzzle
mechanism allows the responder to remain stateless until the very end
of the base exchange [aura-dos]. HIP puzzles have also been studied
under steady-state DDoS attacks [beal-dos], on multiple adversary
models with varying puzzle difficulties [tritilanunt-dos] and with
ephemeral Host Identities [komu-mitigation].

6.2. End-host mobility and multi-homing

HIP decouples the transport from the internetworking layer, and binds
the transport associations to the Host Identities (actually through
either the HIT or LSI). After the initial key exchange, the HIP
layer maintains transport-layer connectivity and data flows using its
mobility [RFC8046] and multihoming [RFC8047] extensions.
Consequently, HIP can provide for a degree of internetworking
mobility and multi-homing at a low infrastructure cost. HIP mobility
includes IP address changes (via any method) to either party. Thus,
a system is considered mobile if its IP address can change
dynamically for any reason like PPP, DHCP, IPv6 prefix reassignments,
or a NAT device remapping its translation. Likewise, a system is
considered multi-homed if it has more than one globally routable IP
address at the same time. HIP links IP addresses together, when
multiple IP addresses correspond to the same Host Identity. If one
address becomes unusable, or a more preferred address becomes
available, existing transport associations can easily be moved to
another address.

When a node moves while communication is already on-going, address
changes are rather straightforward. The peer of the mobile node can
just accept a HIP or an integrity protected ESP packet from any
address and ignore the source address. However, as discussed in
Section 12.2 below, a mobile node must send a HIP UPDATE packet to
inform the peer of the new address(es), and the peer must verify that
the mobile node is reachable through these addresses. This is
especially helpful for those situations where the peer node is
sending data periodically to the mobile node (that is, re-starting a
connection after the initial connection).

6.3. Rendezvous mechanism

Establishing a contact to a mobile, moving node is slightly more
involved. In order to start the HIP exchange, the initiator node has

to know how to reach the mobile node. For instance, the mobile node

can employ Dynamic DNS [RFC2136] to update its reachability
information in the DNS. To avoid the dependency to DNS, HIP provides
its own HIP-specific alternative: the HIP rendezvous mechanism as
defined in HIP Rendezvous specifications [RFC8004].

Using the HIP rendezvous extensions, the mobile node keeps the
rendezvous infrastructure continuously updated with its current IP
address(es). The mobile nodes trusts the rendezvous mechanism in
order to properly maintain their HIT and IP address mappings.

The rendezvous mechanism is especially useful in scenarios where both
of the nodes are expected to change their address at the same time.
In such a case, the HIP UPDATE packets will cross each other in the
network and never reach the peer node.

6.4. Relay mechanism

The HIP relay mechanism [I-D.ietf-hip-native-nat-traversal] is an
alternative to the HIP rendezvous mechanism. The HIP relay mechanism
is more suitable for IPv4 networks with NATs because a HIP relay can
forward all control and data plane communications in order to
guarantee successful NAT traversal.

6.5. Termination of the control plane

The control plane between two hosts is terminated using a secure two
message exchange as specified in base exchange specification
[RFC7401]. The related state (i.e. host associations) should be
removed upon successful termination.

7. Data plane

The encapsulation format for the data plane used for carrying the
application-layer traffic can be dynamically negotiated during the
key exchange. For instance, HICCUPS extensions [RFC6078] define one
way to transport application-layer datagrams directly over the HIP
control plane, protected by asymmetric key cryptography. Also, S-RTP
has been considered as the data encapsulation protocol [hip-srtp].
However, the most widely implemented method is the Encapsulated
Security Payload (ESP) [RFC7402] that is protected by symmetric keys

derived during the key exchange. ESP Security Associations (SAs)
offer both confidentiality and integrity protection, of which the
former can be disabled during the key exchange. In the future, other
ways of transporting application-layer data may be defined.

The ESP SAs are established and terminated between the initiator and
the responder hosts. Usually, the hosts create at least two SAs, one
in each direction (initiator-to-responder SA and responder-to-

initiator SA). If the IP addresses of either host changes, the HIP
mobility extensions can be used to re-negotiate the corresponding
SAs.

On the wire, the difference in the use of identifiers between the HIP
control and data plane is that the HITs are included in all control
packets, but not in the data plane when ESP is employed. Instead,
the ESP employs SPI numbers that act as compressed HITs. Any HIP-
aware middlebox (for instance, a HIP-aware firewall) interested in
the ESP based data plane should keep track between the control and
data plane identifiers in order to associate them with each other.

Since HIP does not negotiate any SA lifetimes, all lifetimes are
subject to local policy. The only lifetimes a HIP implementation
must support are sequence number rollover (for replay protection),
and SA timeout. An SA times out if no packets are received using
that SA. Implementations may support lifetimes for the various ESP
transforms and other data-plane protocols.

8. HIP and NATs

Passing packets between different IP addressing realms requires
changing IP addresses in the packet header. This may occur, for
example, when a packet is passed between the public Internet and a
private address space, or between IPv4 and IPv6 networks. The
address translation is usually implemented as Network Address
Translation (NAT) [RFC3022] or NAT Protocol translation (NAT-PT)
[RFC2766].

In a network environment where identification is based on the IP
addresses, identifying the communicating nodes is difficult when NATs
are employed because private address spaces are overlapping. In
other words, two hosts cannot be distinguished from each other solely

based on their IP address. With HIP, the transport-layer end-points
(i.e. applications) are bound to unique Host Identities rather than
overlapping private addresses. This allows two end-points to
distinguish one other even when they are located in different private
address realms. Thus, the IP addresses are used only for routing
purposes and can be changed freely by NATs when a packet between two
HIP capable hosts traverses through multiple private address realms.

NAT traversal extensions for HIP [I-D.ietf-hip-native-nat-traversal]
can be used to realize the actual end-to-end connectivity through NAT
devices. To support basic backward compatibility with legacy NATs,
the extensions encapsulate both HIP control and data plane in UDP.
The extensions define mechanisms for forwarding the two planes
through an intermediary host called HIP relay and procedures to
establish direct end-to-end connectivity by penetrating NATs.

Besides this "native" NAT traversal mode for HIP, other NAT traversal
mechanisms have been successfully utilized, such as Teredo
[varjonen-split].

Besides legacy NATs, a HIP-aware NAT has been designed and
implemented [ylitalo-spinat]. For a HIP-based flow, a HIP-aware NAT
or NAT-PT system tracks the mapping of HITs, and the corresponding
ESP SPIs, to an IP address. The NAT system has to learn mappings
both from HITs and from SPIs to IP addresses. Many HITs (and SPIs)
can map to a single IP address on a NAT, simplifying connections on
address poor NAT interfaces. The NAT can gain much of its knowledge
from the HIP packets themselves; however, some NAT configuration may
be necessary.

8.1. HIP and Upper-layer checksums

There is no way for a host to know if any of the IP addresses in an
IP header are the addresses used to calculate the TCP checksum. That
is, it is not feasible to calculate the TCP checksum using the actual
IP addresses in the pseudo header; the addresses received in the
incoming packet are not necessarily the same as they were on the
sending host. Furthermore, it is not possible to recompute the
upper-layer checksums in the NAT/NAT-PT system, since the traffic is
ESP protected. Consequently, the TCP and UDP checksums are
calculated using the HITs in the place of the IP addresses in the
pseudo header. Furthermore, only the IPv6 pseudo header format is

used. This provides for IPv4 / IPv6 protocol translation.

9. Multicast

   A number of studies investigating HIP-based multicast have been
   published (including [shields-hip], [xueyong-hip], [xueyong-hip],
   [amir-hip], [kovacshazi-host] and [xueyong-secure]). In particular,
   so-called Bloom filters, that allow compressing of multiple labels
   into small data structures, may be a promising way forward
   [sarela-bloom]. However, the different schemes have not been adopted
   by the HIP working group (nor the HIP research group in IRTF), so the
   details are not further elaborated here.

10. HIP policies

   There are a number of variables that influence the HIP exchange that
   each host must support. All HIP implementations should support at
   least 2 HIs, one to publish in DNS or similar directory service and
   an unpublished one for anonymous usage. Although unpublished HIs
   will be rarely used as responder HIs, they are likely to be common
   for initiators. Support for multiple HIs is recommended. This

   provides new challenges for systems or users to decide which type of
   HI to expose when they start a new session.

   Opportunistic mode (where the initiator starts a HIP exchange without
   prior knowledge of the responder's HI) presents a security tradeoff.
   At the expense of being subject to MITM attacks, the opportunistic
   mode allows the initiator to learn the identity of the responder
   during communication rather than from an external directory.
   Opportunistic mode can be used for registration to HIP-based services
   [RFC8003] (i.e. utilized by HIP for its own internal purposes) or by
   the application layer [komu-leap]. For security reasons, especially
   the latter requires some involvement from the user to accept the
   identity of the responder similar to how SSH prompts the user when
   connecting to a server for the first time [pham-leap]. In practice,
   this can be realized in end-host based firewalls in the case of
   legacy applications [karvonen-usable] or with native APIs for HIP
   APIs [RFC6317] in the case of HIP-aware applications.

   Many initiators would want to use a different HI for different

responders. The implementations should provide for a policy mapping
of initiator HITs to responder HITs. This policy should also include
preferred transforms and local lifetimes.

Responders would need a similar policy, describing the hosts allowed
to participate in HIP exchanges, and the preferred transforms and
local lifetimes.

11. Design considerations

11.1. Benefits of HIP

In the beginning, the network layer protocol (i.e., IP) had the
following four "classic" invariants:

1. Non-mutable: The address sent is the address received.

2. Non-mobile: The address doesn't change during the course of an
    "association".

3. Reversible: A return header can always be formed by reversing the
    source and destination addresses.

4. Omniscient: Each host knows what address a partner host can use
    to send packets to it.

Actually, the fourth can be inferred from 1 and 3, but it is worth
mentioning explicitly for reasons that will be obvious soon if not
already.

In the current "post-classic" world, we are intentionally trying to
get rid of the second invariant (both for mobility and for multi-
homing), and we have been forced to give up the first and the fourth.
Realm Specific IP [RFC3102] is an attempt to reinstate the fourth
invariant without the first invariant. IPv6 is an attempt to
reinstate the first invariant.

Few client-side systems on the Internet have DNS names that are
meaningful. That is, if they have a Fully Qualified Domain Name
(FQDN), that name typically belongs to a NAT device or a dial-up
server, and does not really identify the system itself but its
current connectivity. FQDNs (and their extensions as email names)

are application-layer names; more frequently naming services than
particular systems. This is why many systems on the Internet are not
registered in the DNS; they do not have services of interest to other
Internet hosts.

DNS names are references to IP addresses. This only demonstrates the
interrelationship of the networking and application layers. DNS, as
the Internet's only deployed and distributed database, is also the
repository of other namespaces, due in part to DNSSEC and application
specific key records. Although each namespace can be stretched (IP
with v6, DNS with KEY records), neither can adequately provide for
host authentication or act as a separation between internetworking
and transport layers.

The Host Identity (HI) namespace fills an important gap between the
IP and DNS namespaces. An interesting thing about the HI is that it
actually allows a host to give up all but the 3rd network-layer
invariant. That is to say, as long as the source and destination
addresses in the network-layer protocol are reversible, HIP takes
care of host identification, and reversibility allows a local host to
receive a packet back from a remote host. The address changes
occurring during NAT transit (non-mutable) or host movement (non-
omniscient or non-mobile) can be managed by the HIP layer.

With the exception of High-Performance Computing applications, the
Sockets API is the most common way to develop network applications.
Applications use the Sockets API either directly or indirectly
through some libraries or frameworks. However, the Sockets API is
based on the assumption of static IP addresses, and DNS with its
lifetime values was invented at later stages during the evolution of
the Internet. Hence, the Sockets API does not deal with the lifetime
of addresses [RFC6250]. As the majority of the end-user equipment is
mobile today, their addresses are effectively ephemeral, but the
Sockets API still gives a fallacious illusion of persistent IP
addresses to the unwary developer. HIP can be used to solidify this

illusion because HIP provides persistent surrogate addresses to the
application layer in the form of LSIs and HITs.

The persistent identifiers as provided by HIP are useful in multiple
scenarios (see, e.g., [ylitalo-diss] or [komu-diss], for a more

elaborate discussion):

o When a mobile host moves physically between two different WLAN
   networks and obtains a new address, an application using the
   identifiers remains isolated regardless of the topology changes
   while the underlying HIP layer re-establishes connectivity (i.e. a
   horizontal handoff).

o Similarly, the application utilizing the identifiers remains again
   unaware of the topological changes when the underlying host
   equipped with WLAN and cellular network interfaces switches
   between the two different access technologies (i.e. a vertical
   handoff).

o Even when hosts are located in private address realms,
   applications can uniquely distinguish different hosts from each
   other based on their identifiers. In other words, it can be
   stated that HIP improves Internet transparency for the application
   layer [komu-diss].

o Site renumbering events for services can occur due to corporate
   mergers or acquisitions, or by changes in Internet Service
   Provider. They can involve changing the entire network prefix of
   an organization, which is problematic due to hard-coded addresses
   in service configuration files or cached IP addresses at the
   client side [RFC5887]. Considering such human errors, a site
   employing location-independent identifiers as promoted by HIP may
   experience less problems while renumbering their network.

o More agile IPv6 interoperability can be achieved, as discussed in
   Section 4.4. IPv6-based applications can communicate using HITs
   with IPv4-based applications that are using LSIs. Additionally,
   the underlying network type (IPv4 or IPv6) becomes independent of
   the addressing family of the application.

o HITs (or LSIs) can be used in IP-based access control lists as a
   more secure replacement for IPv6 addresses. Besides security, HIT
   based access control has two other benefits. First, the use of
   HITs can potentially halve the size of access control lists
   because separate rules for IPv4 are not needed [komu-diss].
   Second, HIT-based configuration rules in HIP-aware middleboxes
   remain static and independent of topology changes, thus
   simplifying administrative efforts particularly for mobile

environments. For instance, the benefits of HIT based access
control have been harnessed in the case of HIP-aware firewalls,
but can be utilized directly at the end-hosts as well [RFC6538].

While some of these benefits could be and have been redundantly
implemented by individual applications, providing such generic
functionality at the lower layers is useful because it reduces
software development effort and networking software bugs (as the
layer is tested with multiple applications). It also allows the
developer to focus on building the application itself rather than
delving into the intricacies of mobile networking, thus facilitating
separation of concerns.

HIP could also be realized by combining a number of different
protocols, but the complexity of the resulting software may become
substantially larger, and the interaction between multiple possibly
layered protocols may have adverse effects on latency and throughput.
It is also worth noting that virtually nothing prevents realizing the
HIP architecture, for instance, as an application-layer library,
which has been actually implemented in the past [xin-hip-lib].
However, the tradeoff in moving the HIP layer to the application
layer is that legacy applications may not be supported.

11.2. Drawbacks of HIP

In computer science, many problems can be solved with an extra layer
of indirection. However, the indirection always involves some costs
as there is no such a thing as "free lunch". In the case of HIP, the
main costs could be stated as follows:

o In general, a new layer and a new namespace always involve some
   initial effort in terms of implementation, deployment and
   maintenance. Some education of developers and administrators may
   also be needed. However, the HIP community at the IETF has spent
   years in experimenting, exploring, testing, documenting and
   implementing HIP to ease the adoption costs.

o HIP decouples identifier and locator roles of IP addresses.
   Consequently, a mapping mechanism is needed to associate them
   together. A failure to map a HIT to its corresponding locator may
   result in failed connectivity because a HIT is "flat" by its
   nature and cannot be looked up from the hierarchically organized
   DNS. HITs are flat by design due to a security tradeoff. The
   more bits are allocated for the hash in the HIT, the less likely
   there will be (malicious) collisions.

o From performance viewpoint, HIP control and data plane processing
   introduces some overhead in terms of throughput and latency as
   elaborated below.

The key exchange introduces some extra latency (two round trips) in
the initial transport layer connection establishment between two
hosts. With TCP, additional delay occurs if the underlying network
stack implementation drops the triggering SYN packet during the key
exchange. The same cost may also occur during HIP handoff
procedures. However, subsequent TCP sessions using the same HIP
association will not bear this cost (within the key lifetime). Both
the key exchange and handoff penalties can be minimized by caching
TCP packets. The latter case can further be optimized with TCP user
timeout extensions [RFC5482] as described in further detail by
Schuetz et al [schuetz-intermittent].

The most CPU-intensive operations involve the use of the asymmetric
keys and Diffie-Hellman key derivation at the control plane, but this
occurs only during the key exchange, its maintenance (handoffs,
refreshing of key material) and tear down procedures of HIP
associations. The data plane is typically implemented with ESP
because it has a smaller overhead due to symmetric key encryption.
Naturally, even ESP involves some overhead in terms of latency
(processing costs) and throughput (tunneling) (see e.g.
[ylitalo-diss] for a performance evaluation).

11.3. Deployment and adoption considerations

   This section describes some deployment and adoption considerations
   related to HIP from a technical perspective.

11.3.1. Deployment analysis

   HIP has commercially been utilized at Boeing airplane factory for
   their internal purposes [paine-hip]. It has been included in a
   security product called Tofino to support layer-two Virtual Private
   Networks [henderson-vpls] to facilitate, e.g, supervisory control and
   data acquisition (SCADA) security. However, HIP has not been a "wild
   success" [RFC5218] in the Internet as argued by Levae et al
   [leva-barriers]. Here, we briefly highlight some of their findings
   based on interviews with 19 experts from the industry and academia.

From a marketing perspective, the demand for HIP has been low and
substitute technologies have been favored. Another identified reason
has been that some technical misconceptions related to the early
stages of HIP specifications still persist. Two identified
misconceptions are that HIP does not support NAT traversal, and that
HIP must be implemented in the OS kernel. Both of these claims are

untrue; HIP does have NAT traversal extensions
[I-D.ietf-hip-native-nat-traversal], and kernel modifications can be
avoided with modern operating systems by diverting packets for
userspace processing.

The analysis by Levae et al clarifies infrastructural requirements
for HIP. In a minimal set up, a client and server machine have to
run HIP software. However, to avoid manual configurations, usually
DNS records for HIP are set up. For instance, the popular DNS server
software Bind9 does not require any changes to accommodate DNS
records for HIP because they can be supported in binary format in its
configuration files [RFC6538]. HIP rendezvous servers and firewalls
are optional. No changes are required to network address points,
NATs, edge routers or core networks. HIP may require holes in legacy
firewalls.

The analysis also clarifies the requirements for the host components
that consist of three parts. First, a HIP control plane component is
required, typically implemented as a userspace daemon. Second, a
data plane component is needed. Most HIP implementations utilize the
so called BEET mode of ESP that has been available since Linux kernel
2.6.27, but is included also as a userspace component in a few of the
implementations. Third, HIP systems usually provide a DNS proxy for
the local host that translates HIP DNS records to LSIs and HITs, and
communicates the corresponding locators to HIP userspace daemon.
While the third component is not mandatory, it is very useful for
avoiding manual configurations. The three components are further
described in the HIP experiment report [RFC6538].

Based on the interviews, Levae et al suggest further directions to
facilitate HIP deployment. Transitioning the HIP specifications to
the standards track may help, but other measures could be taken. As
a more radical measure, the authors suggest to implement HIP as a
purely application-layer library [xin-hip-lib] or other kind of
middleware. On the other hand, more conservative measures include

focusing on private deployments controlled by a single stakeholder.
As a more concrete example of such a scenario, HIP could be used by a
single service provider to facilitate secure connectivity between its
servers [komu-cloud].

11.3.2. HIP in 802.15.4 networks

The IEEE 802 standards have been defining MAC layered security. Many
of these standards use EAP [RFC3748] as a Key Management System (KMS)
transport, but some like IEEE 802.15.4 [IEEE.802-15-4.2011] leave the
KMS and its transport as "Out of Scope".

HIP is well suited as a KMS in these environments:

o HIP is independent of IP addressing and can be directly
   transported over any network protocol.

o Master Keys in 802 protocols are commonly pair-based with group
   keys transported from the group controller using pair-wise keys.

o AdHoc 802 networks can be better served by a peer-to-peer KMS than
   the EAP client/server model.

o Some devices are very memory constrained and a common KMS for both
   MAC and IP security represents a considerable code savings.

11.3.3. HIP and Internet of Things

HIP requires certain amount computational resources from a device due
to cryptographic processing. HIP scales down to phones and small
system-on-chip devices (such as Raspberry Pis, Intel Edison), but
small sensors operating with small batteries have remained
problematic. Different extensions to the HIP have been developed to
scale HIP down to smaller devices, typically with different security
tradeoffs. For example, the non-cryptographic identifiers have been
proposed in RFID scenarios. The slimfit approach [hummen] proposes a
compression layer for HIP to make it more suitable for constrained
networks. The approach is applied to a light-weight version of HIP
(i.e. "Diet HIP") in order to scale down to small sensors.

The HIP Diet Exchange [I-D.ietf-hip-dex] design aims at reducing the
overhead of the employed cryptographic primitives by omitting public-

key signatures and hash functions. In doing so, the main goal is to
still deliver similar security properties to the Base Exchange (BEX).

DEX is primarily designed for computation or memory- constrained
sensor/actuator devices. Like BEX, it is expected to be used
together with a suitable security protocol such as the Encapsulated
Security Payload (ESP) for the protection of upper layer protocol
data. In addition, DEX can also be used as a keying mechanism for
security primitives at the MAC layer, e.g., for IEEE 802.15.9
networks ([IEEE.802-15-9]).

The main differences between HIP BEX and DEX are:

1. Minimum collection of cryptographic primitives to reduce the
    protocol overhead.

    *   Static Elliptic Curve Diffie-Hellman key pairs for peer
        authentication and encryption of the session key.

    *   AES-CTR for symmetric encryption and AES-CMAC for MACing
        function.

    *   A simple fold function for HIT generation.

2. Forfeit of Perfect Forward Secrecy with the dropping of an
    ephemeral Diffie-Hellman key agreement.

3. Forfeit of digital signatures with the removal of a hash
    function. Reliance on ECDH derived key used in HIP_MAC to prove
    ownership of the private key.

4. Diffie-Hellman derived key ONLY used to protect the HIP packets.
    A separate secret exchange within the HIP packets creates the
    session key(s).

5. Optional retransmission strategy tailored to handle the
    potentially extensive processing time of the employed
    cryptographic operations on computationally constrained devices.

11.4. Answers to NSRG questions

The IRTF Name Space Research Group has posed a number of evaluating questions in their report [nsrg-report]. In this section, we provide answers to these questions.

1. How would a stack name improve the overall functionality of the Internet?

    HIP decouples the internetworking layer from the transport layer, allowing each to evolve separately. The decoupling makes end-host mobility and multi-homing easier, also across IPv4 and IPv6 networks. HIs make network renumbering easier, and they also make process migration and clustered servers easier to implement. Furthermore, being cryptographic in nature, they provide the basis for solving the security problems related to end-host mobility and multi-homing.

2. What does a stack name look like?

    A HI is a cryptographic public key. However, instead of using the keys directly, most protocols use a fixed size hash of the public key.

3. What is its lifetime?

    HIP provides both stable and temporary Host Identifiers. Stable HIs are typically long lived, with a lifetime of years

    or more. The lifetime of temporary HIs depends on how long the upper-layer connections and applications need them, and can range from a few seconds to years.

4. Where does it live in the stack?

    The HIs live between the transport and internetworking layers.

5. How is it used on the end points?

    The Host Identifiers may be used directly or indirectly (in the form of HITs or LSIs) by applications when they access network services. Additionally, the Host Identifiers, as public keys, are used in the built in key agreement protocol,

called the HIP base exchange, to authenticate the hosts to
each other.

6. What administrative infrastructure is needed to support it?

    In some environments, it is possible to use HIP
    opportunistically, without any infrastructure. However, to
    gain full benefit from HIP, the HIs must be stored in the DNS
    or a PKI, and a new rendezvous mechanism is needed [RFC8005].

7. If we add an additional layer would it make the address list in
SCTP unnecessary?

    Yes

8. What additional security benefits would a new naming scheme
offer?

    HIP reduces dependency on IP addresses, making the so called
    address ownership [Nik2001] problems easier to solve. In
    practice, HIP provides security for end-host mobility and
    multi-homing. Furthermore, since HIP Host Identifiers are
    public keys, standard public key certificate infrastructures
    can be applied on the top of HIP.

9. What would the resolution mechanisms be, or what characteristics
of a resolution mechanisms would be required?

    For most purposes, an approach where DNS names are resolved
    simultaneously to HIs and IP addresses is sufficient.
    However, if it becomes necessary to resolve HIs into IP
    addresses or back to DNS names, a flat resolution
    infrastructure is needed. Such an infrastructure could be

    based on the ideas of Distributed Hash Tables, but would
    require significant new development and deployment.

12. Security considerations

This section includes discussion on some issues and solutions related
to security in the HIP architecture.

. MiTM Attacks

   HIP takes advantage of the new Host Identity paradigm to provide
   secure authentication of hosts and to provide a fast key exchange for
   ESP. HIP also attempts to limit the exposure of the host to various
   denial-of-service (DoS) and man-in-the-middle (MitM) attacks. In so
   doing, HIP itself is subject to its own DoS and MitM attacks that
   potentially could be more damaging to a host's ability to conduct
   business as usual.

   Resource exhausting denial-of-service attacks take advantage of the
   cost of setting up a state for a protocol on the responder compared
   to the 'cheapness' on the initiator. HIP allows a responder to
   increase the cost of the start of state on the initiator and makes an
   effort to reduce the cost to the responder. This is done by having
   the responder start the authenticated Diffie-Hellman exchange instead
   of the initiator, making the HIP base exchange 4 packets long. The
   first packet sent by the responder can be prebuilt to further
   mitigate the costs. This packet also includes a computational puzzle
   that can optionally be used to further delay the initiator, for
   instance, when the responder is overloaded. The details are
   explained in the base exchange specification [RFC7401].

   Man-in-the-middle (MitM) attacks are difficult to defend against,
   without third-party authentication. A skillful MitM could easily
   handle all parts of the HIP base exchange, but HIP indirectly
   provides the following protection from a MitM attack. If the
   responder's HI is retrieved from a signed DNS zone or securely
   obtained by some other means, the initiator can use this to
   authenticate the signed HIP packets. Likewise, if the initiator's HI
   is in a secure DNS zone, the responder can retrieve it and validate
   the signed HIP packets. However, since an initiator may choose to
   use an unpublished HI, it knowingly risks a MitM attack. The
   responder may choose not to accept a HIP exchange with an initiator
   using an unknown HI.

   Other types of MitM attacks against HIP can be mounted using ICMP
   messages that can be used to signal about problems. As a overall
   guideline, the ICMP messages should be considered as unreliable
   "hints" and should be acted upon only after timeouts. The exact

attack scenarios and countermeasures are described in full detail the
base exchange specification [RFC7401].

The need to support multiple hashes for generating the HIT from the
HI affords the MitM to mount a potentially powerful downgrade attack
due to the a-priori need of the HIT in the HIP base exchange. The
base exchange has been augmented to deal with such an attack by
restarting on detecting the attack. At worst this would only lead to
a situation in which the base exchange would never finish (or would
be aborted after some retries). As a drawback, this leads to an
6-way base exchange which may seem bad at first. However, since this
only occurs in an attack scenario and since the attack can be handled
(so it is not interesting to mount anymore), we assume the subsequent
messages do not represent a security threat. Since the MitM cannot
be successful with a downgrade attack, these sorts of attacks will
only occur as 'nuisance' attacks. So, the base exchange would still
be usually just four packets even though implementations must be
prepared to protect themselves against the downgrade attack.

In HIP, the Security Association for ESP is indexed by the SPI; the
source address is always ignored, and the destination address may be
ignored as well. Therefore, HIP-enabled Encapsulated Security
Payload (ESP) is IP address independent. This might seem to make
attacking easier, but ESP with replay protection is already as well
protected as possible, and the removal of the IP address as a check
should not increase the exposure of ESP to DoS attacks.

12.2. Protection against flooding attacks

Although the idea of informing about address changes by simply
sending packets with a new source address appears appealing, it is
not secure enough. That is, even if HIP does not rely on the source
address for anything (once the base exchange has been completed), it
appears to be necessary to check a mobile node's reachability at the
new address before actually sending any larger amounts of traffic to
the new address.

Blindly accepting new addresses would potentially lead to flooding
Denial-of-Service attacks against third parties [RFC4225]. In a
distributed flooding attack an attacker opens high volume HIP
connections with a large number of hosts (using unpublished HIs), and
then claims to all of these hosts that it has moved to a target
node's IP address. If the peer hosts were to simply accept the move,
the result would be a packet flood to the target node's address. To
prevent this type of attack, HIP mobility extensions include a return
routability check procedure where the reachability of a node is
separately checked at each address before using the address for
larger amounts of traffic.

A credit-based authorization approach for host mobility with the Host Identity Protocol [RFC8046] can be used between hosts for sending data prior to completing the address tests. Otherwise, if HIP is used between two hosts that fully trust each other, the hosts may optionally decide to skip the address tests. However, such performance optimization must be restricted to peers that are known to be trustworthy and capable of protecting themselves from malicious software.

12.3. HITs used in ACLs

At end-hosts, HITs can be used in IP-based access control lists at the application and network layers. At middleboxes, HIP-aware firewalls [lindqvist-enterprise] can use HITs or public keys to control both ingress and egress access to networks or individual hosts, even in the presence of mobile devices because the HITs and public keys are topologically independent. As discussed earlier in Section 7, once a HIP session has been established, the SPI value in an ESP packet may be used as an index, indicating the HITs. In practice, firewalls can inspect HIP packets to learn of the bindings between HITs, SPI values, and IP addresses. They can even explicitly control ESP usage, dynamically opening ESP only for specific SPI values and IP addresses. The signatures in HIP packets allow a capable firewall to ensure that the HIP exchange is indeed occurring between two known hosts. This may increase firewall security.

A potential drawback of HITs in ACLs is their 'flatness' means they cannot be aggregated, and this could potentially result in larger table searches in HIP-aware firewalls. A way to optimize this could be to utilize Bloom filters for grouping of HITs [sarela-bloom]. However, it should be noted that it is also easier to exclude individual, misbehaving hosts out when the firewall rules concern individual HITs rather than groups.

There has been considerable bad experience with distributed ACLs that contain public key related material, for example, with SSH. If the owner of a key needs to revoke it for any reason, the task of finding all locations where the key is held in an ACL may be impossible. If the reason for the revocation is due to private key theft, this could be a serious issue.

A host can keep track of all of its partners that might use its HIT in an ACL by logging all remote HITs. It should only be necessary to

log responder hosts. With this information, the host can notify the
various hosts about the change to the HIT. There have been attempts
to develop a secure method to issue the HIT revocation notice
[zhang-revocation].

Some of the HIP-aware middleboxes, such as firewalls
[lindqvist-enterprise] or NATs [ylitalo-spinat], may observe the on-
path traffic passively. Such middleboxes are transparent by their
nature and may not get a notification when a host moves to a
different network. Thus, such middleboxes should maintain soft state
and timeout when the control and data plane between two HIP end-hosts
has been idle too long. Correspondingly, the two end-hosts may send
periodically keepalives, such as UPDATE packets or ICMP messages
inside the ESP tunnel, to sustain state at the on-path middleboxes.

One general limitation related to end-to-end encryption is that
middleboxes may not be able to participate to the protection of data
flows. While the issue may affect also other protocols, Heer at al
[heer-end-host] have analyzed the problem in the context of HIP.
More specifically, when ESP is used as the data-plane protocol for
HIP, the association between the control and data plane is weak and
can be exploited under certain assumptions. In the scenario, the
attacker has already gained access to the target network protected by
a HIP-aware firewall, but wants to circumvent the HIP-based firewall.
To achieve this, the attacker passively observes a base exchange
between two HIP hosts and later replays it. This way, the attacker
manages to penetrate the firewall and can use a fake ESP tunnel to
transport its own data. This is possible because the firewall cannot
distinguish when the ESP tunnel is valid. As a solution, HIP-aware
middleboxes may participate to the control plane interaction by
adding random nonce parameters to the control traffic, which the end-
hosts have to sign to guarantee the freshness of the control traffic
[heer-midauth]. As an alternative, extensions for transporting data
plane directly over the control plane can be used [RFC6078].

12.4. Alternative HI considerations

The definition of the Host Identifier states that the HI need not be
a public key. It implies that the HI could be any value; for example
a FQDN. This document does not describe how to support such a non-
cryptographic HI, but examples of such protocol variants do exist

([urien-rfid], [urien-rfid-draft]). A non-cryptographic HI would
still offer the services of the HIT or LSI for NAT traversal. It
would be possible to carry HITs in HIP packets that had neither
privacy nor authentication. Such schemes may be employed for
resource constrained devices, such as small sensors operating on
battery power, but are not further analyzed here.

If it is desirable to use HIP in a low security situation where
public key computations are considered expensive, HIP can be used
with very short Diffie-Hellman and Host Identity keys. Such use
makes the participating hosts vulnerable to MitM and connection
hijacking attacks. However, it does not cause flooding dangers,

since the address check mechanism relies on the routing system and
not on cryptographic strength.

13. IANA considerations

This document has no actions for IANA.

14. Acknowledgments

For the people historically involved in the early stages of HIP, see
the Acknowledgments section in the Host Identity Protocol
specification.

During the later stages of this document, when the editing baton was
transferred to Pekka Nikander, the comments from the early
implementers and others, including Jari Arkko, Tom Henderson, Petri
Jokela, Miika Komu, Mika Kousa, Andrew McGregor, Jan Melen, Tim
Shepard, Jukka Ylitalo, Sasu Tarkoma, and Jorma Wall, were
invaluable. Also, the comments from Lars Eggert, Spencer Dawkins and
Dave Crocker were also useful.

The authors want to express their special thanks to Tom Henderson,
who took the burden of editing the document in response to IESG
comments at the time when both of the authors were busy doing other
things. Without his perseverance original document might have never
made it as RFC4423.

This main effort to update and move HIP forward within the IETF
process owes its impetuous to a number of HIP development teams. The

authors are grateful for Boeing, Helsinki Institute for Information
Technology (HIIT), NomadicLab of Ericsson, and the three
universities: RWTH Aachen, Aalto and University of Helsinki, for
their efforts. Without their collective efforts HIP would have
withered as on the IETF vine as a nice concept.

Thanks also for Suvi Koskinen for her help with proofreading and with
the reference jungle.

15. Changes from RFC 4423

In a nutshell, the changes from RFC 4423 [RFC4423] are mostly
editorial, including clarifications on topics described in a
difficult way and omitting some of the non-architectural
(implementation) details that are already described in other
documents. A number of missing references to the literature were
also added. New topics include the drawbacks of HIP, discussion on
802.15.4 and MAC security, HIP for IoT scenarios, deployment
considerations and description of the base exchange.

16. References

16.1. Normative References

[I-D.ietf-hip-dex]
           Moskowitz, R. and R. Hummen, "HIP Diet EXchange (DEX)",
           draft-ietf-hip-dex-05 (work in progress), February 2017.

[I-D.ietf-hip-native-nat-traversal]
           Keranen, A., Melen, J., and M. Komu, "Native NAT Traversal
           Mode for the Host Identity Protocol", draft-ietf-hip-
           native-nat-traversal-16 (work in progress), February 2017.

[I-D.ietf-hip-rfc6253-bis]
           Heer, T. and S. Varjonen, "Host Identity Protocol
           Certificates", draft-ietf-hip-rfc6253-bis-09 (work in
           progress), July 2016.

[RFC5482] Eggert, L. and F. Gont, "TCP User Timeout Option",
           RFC 5482, DOI 10.17487/RFC5482, March 2009,
           <http://www.rfc-editor.org/info/rfc5482>.

   [RFC7343] Laganier, J. and F. Dupont, "An IPv6 Prefix for Overlay
              Routable Cryptographic Hash Identifiers Version 2
              (ORCHIDv2)", RFC 7343, DOI 10.17487/RFC7343, September
              2014, <http://www.rfc-editor.org/info/rfc7343>.

   [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>.

   [RFC8005] Laganier, J., "Host Identity Protocol (HIP) Domain Name
              System (DNS) Extension", RFC 8005, DOI 10.17487/RFC8005,
              October 2016, <http://www.rfc-editor.org/info/rfc8005>.

   [RFC8046] Henderson, T., Ed., Vogt, C., and J. Arkko, "Host Mobility
              with the Host Identity Protocol", RFC 8046,
              DOI 10.17487/RFC8046, February 2017,
              <http://www.rfc-editor.org/info/rfc8046>.

   [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>.

16.2. Informative references

[amir-hip]
          Amir, K., Forsgren, H., Grahn, K., Karvi, T., and G.
          Pulkkis, "Security and Trust of Public Key Cryptography
          for HIP and HIP Multicast", International Journal of
          Dependable and Trustworthy Information Systems (IJDTIS),
          2(3), 17-35, DOI: 10.4018/jdtis.2011070102, 2013.

[aura-dos]
          Aura, T., Nikander, P., and J. Leiwo, "DOS-resistant
          Authentication with Client Puzzles", 8th International
          Workshop on Security Protocols, pages 170-177. Springer, ,
          April 2001.

[beal-dos]
          Beal, J. and T. Shephard, "Deamplification of DoS Attacks
          via Puzzles", , October 2004.

[chiappa-endpoints]
          Chiappa, J., "Endpoints and Endpoint Names: A Proposed
          Enhancement to the Internet Architecture",
          URL http://www.chiappa.net/~jnc/tech/endpoints.txt, 1999.

[heer-end-host]
          Heer, T., Hummen, R., Komu, M., Goetz, S., and K. Wehre,
          "End-host Authentication and Authorization for Middleboxes
          based on a Cryptographic Namespace", ICC2009 Communication
          and Information Systems Security Symposium, , 2009.

[heer-midauth]
          Heer, T. and M. Komu, "End-Host Authentication for HIP
          Middleboxes", Working draft draft-heer-hip-middle-auth-02,
          September 2009.

[henderson-vpls]
          Henderson, T. and D. Mattes, "HIP-based Virtual Private
          LAN Service (HIPLS)", Working draft draft-henderson-hip-
          vpls-07, Dec 2013.

[hip-srtp]
          Tschofenig, H., Muenz, F., and M. Shanmugam, "Using SRTP
          transport format with HIP", Working draft draft-

                  tschofenig-hiprg-hip-srtp-01, October 2005.

   [hummen]   Hummen, R., Hiller, J., Henze, M., and K. Wehrle, "Slimfit
                  - A HIP DEX Compression Layer for the IP-based Internet of
                  Things", Wireless and Mobile Computing, Networking and
                  Communications (WiMob), 2013 IEEE 9th International
                  Conference on , page 259-266. DOI:
                  10.1109/WiMOB.2013.6673370, October 2013.

   [IEEE.802-15-4.2011]
                  "Information technology - Telecommunications and
                  information exchange between systems - Local and
                  metropolitan area networks - Specific requirements - Part
                  15.4: Wireless Medium Access Control (MAC) and Physical
                  Layer (PHY) Specifications for Low-Rate Wireless Personal
                  Area Networks (WPANs)", IEEE Standard 802.15.4, September
                  2011, <http://standards.ieee.org/getieee802/
                  download/802.15.4-2011.pdf>.

   [IEEE.802-15-9]
                  "IEEE Draft Recommended Practice for Transort of Key
                  Management Protocol (KMP) Datagrams", IEEE P802.15.9/D04,
                  May 2015.

   [karvonen-usable]
                  Karvonen, K., Komu, M., and A. Gurtov, "Usable Security
                  Management with Host Identity Protocol", 7th ACS/IEEE
                  International Conference on Computer Systems and
                  Applications, (AICCSA-2009), 2009.

   [komu-cloud]
                  Komu, M., Sethi, M., Mallavarapu, R., Oirola, H., Khan,
                  R., and S. Tarkoma, "Secure Networking for Virtual
                  Machines in the Cloud", International Workshop on Power
                  and QoS Aware Computing (PQoSCom2012), IEEE, ISBN:
                  978-1-4244-8567-3, September 2012.

   [komu-diss]

Komu, M., "A Consolidated Namespace for Network
Applications, Developers, Administrators and Users",
Dissertation, Aalto University, Espoo, Finland ISBN:
978-952-60-4904-5 (printed), ISBN: 978-952-60-4905-2
(electronic). , December 2012.

[komu-leap]
Komu, M. and J. Lindqvist, "Leap-of-Faith Security is
Enough for IP Mobility", 6th Annual IEEE Consumer
Communications and Networking Conference IEEE CCNC 2009,
Las Vegas, Nevada, , January 2009.

[komu-mitigation]
Komu, M., Tarkoma, S., and A. Lukyanenko, "Mitigation of
Unsolicited Traffic Across Domains with Host Identities
and Puzzles", 15th Nordic Conference on Secure IT Systems
(NordSec 2010), Springer Lecture Notes in Computer
Science, Volume 7127, pp. 33-48, ISBN: 978-3-642-27936-2,
October 2010.

[kovacshazi-host]
Kovacshazi, Z. and R. Vida, "Host Identity Specific
Multicast", International conference on Networking and
Services (ICNS'06), IEEE Computer Society, Los Alamitos,
CA, USA, http://doi.ieeecomputersociety.org/10.1109/
ICNS.2007.66, 2007.

[leva-barriers]
Levae, A., Komu, M., and S. Luukkainen, "Adoption Barriers
of Network-layer Protocols: the Case of Host Identity
Protocol", The International Journal of Computer and
Telecommunications Networking, ISSN: 1389-1286, March
2013.

[lindqvist-enterprise]
Lindqvist, J., Vehmersalo, E., Manner, J., and M. Komu,
"Enterprise Network Packet Filtering for Mobile
Cryptographic Identities", International Journal of
Handheld Computing Research, 1 (1), 79-94, , January-March
2010.

[Nik2001]   Nikander, P., "Denial-of-Service, Address Ownership, and
Early Authentication in the IPv6 World", in Proceesings
of Security Protocols, 9th International Workshop,
Cambridge, UK, April 25-27 2001, LNCS 2467, pp. 12-26,
Springer, 2002.

[nsrg-report]
          Lear, E. and R. Droms, "What's In A Name:Thoughts from the
          NSRG", draft-irtf-nsrg-report-10 (work in progress),
          September 2003.

[paine-hip]
          Paine, R., "Beyond HIP: The End to Hacking As We Know It",
          BookSurge Publishing, ISBN: 1439256047, 9781439256046,
          2009.

[pham-leap]
          Pham, V. and T. Aura, "Security Analysis of Leap-of-Faith
          Protocols", Seventh ICST International Conference on
          Security and Privacy for Communication Networks, ,
          September 2011.

[RFC2136] Vixie, P., Ed., Thomson, S., Rekhter, Y., and J. Bound,
          "Dynamic Updates in the Domain Name System (DNS UPDATE)",
          RFC 2136, DOI 10.17487/RFC2136, April 1997,
          <http://www.rfc-editor.org/info/rfc2136>.

[RFC2535] Eastlake 3rd, D., "Domain Name System Security
          Extensions", RFC 2535, DOI 10.17487/RFC2535, March 1999,
          <http://www.rfc-editor.org/info/rfc2535>.

[RFC2766] Tsirtsis, G. and P. Srisuresh, "Network Address
          Translation - Protocol Translation (NAT-PT)", RFC 2766,
          DOI 10.17487/RFC2766, February 2000,
          <http://www.rfc-editor.org/info/rfc2766>.

[RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network
          Address Translator (Traditional NAT)", RFC 3022,
          DOI 10.17487/RFC3022, January 2001,
          <http://www.rfc-editor.org/info/rfc3022>.

[RFC3102] Borella, M., Lo, J., Grabelsky, D., and G. Montenegro,
          "Realm Specific IP: Framework", RFC 3102,
          DOI 10.17487/RFC3102, October 2001,
          <http://www.rfc-editor.org/info/rfc3102>.

[RFC3748] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H.
          Levkowetz, Ed., "Extensible Authentication Protocol
          (EAP)", RFC 3748, DOI 10.17487/RFC3748, June 2004,
          <http://www.rfc-editor.org/info/rfc3748>.

   [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>.

   [RFC4306] Kaufman, C., Ed., "Internet Key Exchange (IKEv2)
              Protocol", RFC 4306, DOI 10.17487/RFC4306, December 2005,
              <http://www.rfc-editor.org/info/rfc4306>.

   [RFC4423] Moskowitz, R. and P. Nikander, "Host Identity Protocol
              (HIP) Architecture", RFC 4423, DOI 10.17487/RFC4423, May
              2006, <http://www.rfc-editor.org/info/rfc4423>.

   [RFC5218] Thaler, D. and B. Aboba, "What Makes For a Successful
              Protocol?", RFC 5218, DOI 10.17487/RFC5218, July 2008,
              <http://www.rfc-editor.org/info/rfc5218>.

   [RFC5338] Henderson, T., Nikander, P., and M. Komu, "Using the Host
              Identity Protocol with Legacy Applications", RFC 5338,
              DOI 10.17487/RFC5338, September 2008,
              <http://www.rfc-editor.org/info/rfc5338>.

   [RFC5887] Carpenter, B., Atkinson, R., and H. Flinck, "Renumbering
              Still Needs Work", RFC 5887, DOI 10.17487/RFC5887, May
              2010, <http://www.rfc-editor.org/info/rfc5887>.

   [RFC6078] Camarillo, G. and J. Melen, "Host Identity Protocol (HIP)
              Immediate Carriage and Conveyance of Upper-Layer Protocol
              Signaling (HICCUPS)", RFC 6078, DOI 10.17487/RFC6078,
              January 2011, <http://www.rfc-editor.org/info/rfc6078>.

   [RFC6250] Thaler, D., "Evolution of the IP Model", RFC 6250,
              DOI 10.17487/RFC6250, May 2011,
              <http://www.rfc-editor.org/info/rfc6250>.

   [RFC6281] Cheshire, S., Zhu, Z., Wakikawa, R., and L. Zhang,
              "Understanding Apple's Back to My Mac (BTMM) Service",
              RFC 6281, DOI 10.17487/RFC6281, June 2011,
              <http://www.rfc-editor.org/info/rfc6281>.

    [RFC6317] Komu, M. and T. Henderson, "Basic Socket Interface
               Extensions for the Host Identity Protocol (HIP)",
               RFC 6317, DOI 10.17487/RFC6317, July 2011,
               <http://www.rfc-editor.org/info/rfc6317>.

    [RFC6537] Ahrenholz, J., "Host Identity Protocol Distributed Hash
               Table Interface", RFC 6537, DOI 10.17487/RFC6537, February
               2012, <http://www.rfc-editor.org/info/rfc6537>.

    [RFC6538] Henderson, T. and A. Gurtov, "The Host Identity Protocol
               (HIP) Experiment Report", RFC 6538, DOI 10.17487/RFC6538,
               March 2012, <http://www.rfc-editor.org/info/rfc6538>.

    [sarela-bloom]
               Saerelae, M., Esteve Rothenberg, C., Zahemszky, A.,
               Nikander, P., and J. Ott, "BloomCasting: Security in Bloom
               filter based multicast", , Lecture Notes in Computer
               Science 2012, , pages 1-16, Springer Berlin Heidelberg,
               2012.

    [schuetz-intermittent]
               Schuetz, S., Eggert, L., Schmid, S., and M. Brunner,
               "Protocol enhancements for intermittently connected
               hosts", SIGCOMM Comput. Commun. Rev., 35(3):5-18, , July
               2005.

    [shields-hip]
               Shields, C. and J. Garcia-Luna-Aceves, "The HIP protocol
               for hierarchical multicast routing", Proceedings of the
               seventeenth annual ACM symposium on Principles of
               distributed computing, pages 257-266. ACM, New York, NY,
               USA, ISBN: 0-89791-977-7, DOI: 10.1145/277697.277744,
               1998.

    [tritilanunt-dos]
               Tritilanunt, S., Boyd, C., Foo, E., and J. Nieto,
               "Examining the DoS Resistance of HIP", OTM Workshops (1),
               volume 4277 of Lecture Notes in Computer Science, pages
               616-625,Springer , 2006.

    [urien-rfid]

Urien, P., Chabanne, H., Bouet, M., de Cunha, D., Guyot, V., Pujolle, G., Paradinas, P., Gressier, E., and J. Susini, "HIP-based RFID Networking Architecture", IFIP International Conference on Wireless and Optical Communications Networks, DOI: 10.1109/WOCN.2007.4284140, July 2007.

[urien-rfid-draft]
Urien, P., Lee, G., and G. Pujolle, "HIP support for RFIDs", IRTF Working draft draft-irtf-hiprg-rfid-07, April 2013.

[varjonen-split]
Varjonen, S., Komu, M., and A. Gurtov, "Secure and Efficient IPv4/IPv6 Handovers Using Host-Based Identifier-Location Split", Journal of Communications Software and Systems, 6(1), 2010, ISSN: 18456421, 2010.

[xin-hip-lib]
Xin, G., "Host Identity Protocol Version 2.5", Master's Thesis, Aalto University, Espoo, Finland, , June 2012.

[xueyong-hip]
Xueyong, Z., Zhiguo, D., and W. Xinling, "A Multicast Routing Algorithm Applied to HIP-Multicast Model", Proceedings of the 2011 International Conference on Network Computing and Information Security - Volume 01 (NCIS '11), Vol. 1. IEEE Computer Society, Washington, DC, USA, pages 169-174, DOI: 10.1109/NCIS.2011.42, 2011.

[xueyong-secure]
Xueyong, Z. and J. Atwood, "A Secure Multicast Model for Peer-to-Peer and Access Networks Using the Host Identity Protocol", Consumer Communications and Networking Conference. CCNC 2007. 4th IEEE, pages 1098,1102, DOI: 10.1109/CCNC.2007.221, January 2007.

[ylitalo-diss]
          Ylitalo, J., "Secure Mobility at Multiple Granularity
          Levels over Heterogeneous Datacom Networks", Dissertation,
          Helsinki University of Technology, Espoo, Finland ISBN
          978-951-22-9531-9, 2008.

[ylitalo-spinat]
          Ylitalo, J., Salmela, P., and H. Tschofenig, "SPINAT:
          Integrating IPsec into overlay routing", Proceedings of
          the First International Conference on Security and Privacy
          for Emerging Areas in Communication Networks (SecureComm
          2005). Athens, Greece. IEEE Computer Society, pages
          315-326, ISBN: 0-7695-2369-2, September 2005.

[zhang-revocation]
          Zhang, D., Kuptsov, D., and S. Shen, "Host Identifier
          Revocation in HIP", IRTF Working draft draft-irtf-hiprg-
          revocation-05, Mar 2012.

Authors' Addresses

   Robert Moskowitz (editor)
   HTT Consulting
   Oak Park
   Michigan
   USA

   Email: rgm@labs.htt-consult.com

   Miika Komu
   Ericsson
   Hirsalantie 11
   02420 Jorvas
   Finland

   Email: miika.komu@ericsson.com