Network Working Group
Internet-Draft
Expires: January 14, 2006

P. Nikander
Ericsson Research Nomadic Lab
J. Laganier
DoCoMo Euro-Labs
July 11, 2005

Host Identity Protocol (HIP) Domain Name System (DNS) Extensions
draft-ietf-hip-dns-02

Status of this Memo

Copyright Notice

Abstract

This document specifies two new resource records (RRs) for the Domain Name System (DNS), and how to use them with the Host Identity Protocol (HIP). These RRs allow a HIP node to store in the DNS its Host Identity (HI, the public component of the node public-private key pair), Host Identity Tag (HIT, a truncated hash of its public key), and the Domain Name or IP addresses of its Rendezvous Servers (RVS).

Table of Contents

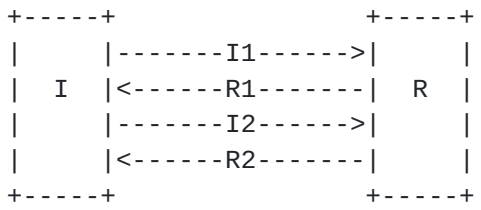
1. Introduction

This document specifies two new resource records (RRs) for the Domain Name System (DNS) [1], and how to use them with the Host Identity Protocol (HIP) [11]. These RRs allow a HIP node to store in the DNS its Host Identity (HI, the public component of the node public-private key pair), Host Identity Tag (HIT, a truncated hash of its HI), and the Domain Name or IP addresses of its Rendezvous Servers (RVS) [14].

The current Internet architecture defines two global namespaces: IP addresses and domain names. The Domain Name System provides a two way lookup between these two namespaces. The HIP architecture [12] defines a new third namespace, called the Host Identity Namespace. This namespace is composed of Host Identifiers (HI) of HIP nodes. The Host Identity Tag (HIT) is one representation of an HI. This representation is obtained by taking the output of a secure hash function applied to the HI, truncated to the IPv6 address size. HITs are supposed to be used in the place of IP addresses within most ULPs and applications.

The Host Identity Protocol [11] allows two HIP nodes to establish together a HIP Association. A HIP association is bound to the nodes HIs rather than to their IP address(es).

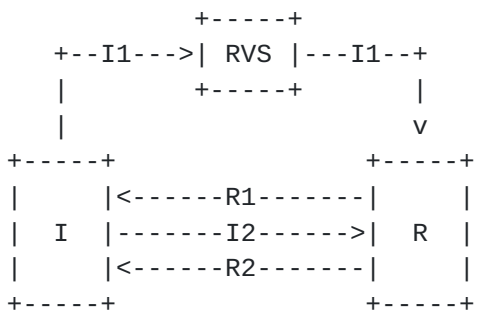
A HIP node establish a HIP association by initiating a 4 way handshake where two parties, the Initiatior and Responder, exchange the I1, I2, R1 and R2 HIP packets (see section 5.3 of [11])



Although a HIP node can initiate HIP communication "opportunistically", i.e., without a priori knowledge of its peer's HI, doing so exposes both endpoints to Man-in-the-Middle attacks on the HIP handshake and its cryptographic protocol. Hence, there is a desire to gain knowledge of peers' HI before applications and ULPs initiate communication. Because many applications use the Domain Name System [1] to name nodes, DNS is a straightforward way to provision nodes with the HIP informations (i.e. HI and possibly RVS) of nodes named in the DNS tree, without introducing or relying on an additional piece of infrastructure. Note that without DNSSEC [3] the Man-in-the-Middle attack evocated before has moved from the

opportunistic HIP handshake to the DNS name resolution; See also Section 8.

The proposed HIP multi-homing mechanisms [13] allow a node to dynamically change its set of underlying IP addresses while maintaining IPsec SA and transport layer session survivability. The HIP rendezvous extensions [14] proposal allows a HIP node to maintain HIP reachability while it is changing its current location (the node IP address(es)). This rendezvous service is provided by a third party, the node's Rendezvous Server (RVS).



An initiator (I) willing to establish a HIP association with a responder (R) would typically initiate a HIP exchange by sending an I1 towards the RVS IP address rather than towards the responder IP address. Then, the RVS, noticing that the receiver HIT is not its own, but the HIT of a HIP node registered for the rendezvous Service, would relay the I1 to the responder. Typically the responder would then complete the exchange without further assistance from the RVS by sending an R1 directly to the initiator IP address.

Currently, most of the Internet applications that need to communicate with a remote host first translate a domain name (often obtained via user input) into one or more IP address(es). This step occurs prior to communication with the remote host, and relies on a DNS lookup.

With HIP, IP addresses are expected to be used mostly for on-the-wire communication between end hosts, while most ULPs and applications uses HIs or HITs instead (ICMP might be an example of an ULP not using them). Consequently, we need a means to translate a domain name into an HI. Using the DNS for this translation is pretty straightforward: We define a new HIPHI (HIP HI) resource record. Upon query by an application or ULP for a FQDN -> IP lookup, the resolver would then additionally perform an FQDN -> HI lookup, and use it to construct the resulting HI -> IP mapping (which is internal to the HIP layer). The HIP layer uses the HI -> IP mapping to translate HIs and their local representations (HITs, IPv4 and IPv6-compatible LSIs) into IP addresses and vice versa.

This draft introduces the following new DNS Resource Records:

- HIPHI, for storing Host Identifiers and Host Identity Tags
- HIPRVS, for storing rendezvous server information

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC2119 [4].

3. Usage Scenarios

In this section, we briefly introduce a number of usage scenarios where the DNS is useful with the Host Identity Protocol.

With HIP, most application and ULPs are unaware of the IP addresses used to carry packets on the wire. Consequently, a HIP node could take advantage of having multiple IP addresses for fail-over, redundancy, mobility, or renumbering, in a manner which is transparent to most ULPs and applications (because they are bound to HIs, hence they are agnostic to these IP address changes).

In these situations, a node wishing to be reachable by reference to its FQDN should store the following informations in the DNS:

- A set of IP address(es) through A and AAAA RRs.
- A Host Identity (HI) and/or Host Identity Tag (HIT) through HIPHI RRs.
- An IP address or DNS name of its Rendezvous Server(s) (RVS) through HIPRVS RRs.

When a HIP node wants to initiate a communication with another HIP node, it first needs to perform a HIP base exchange to set-up a HIP association towards its peer. Although such an exchange can be initiated opportunistically, i.e., without a priori knowledge of the responder's HI, by doing so both nodes knowingly risk man-in-the-middle attacks on the HIP exchange. To prevent these attacks, it is recommended that the initiator first obtain the HI of the responder, and then initiate the exchange. This can be done, for example, through manual configuration or DNS lookups. Hence, a new HIPHI RR is introduced.

When a HIP node is frequently changing its IP address(es), the dynamic DNS update latency may prevent it from publishing its new IP address(es) in the DNS. For solving this problem, the HIP architecture introduces Rendezvous Servers (RVS). A HIP host uses a Rendezvous Server as a Rendezvous point, to maintain reachability with possible HIP initiators. Such a HIP node would publish in the DNS its RVS IP address or DNS name in a HIPRVS RR, while keeping its RVS up-to-date with its current set of IP addresses.

When a HIP node wants to initiate a HIP exchange with a responder it will perform a number of DNS lookups. First the initiator will need to query for an A or AAAA record at the responders FQDN.

If the query for the A and/or AAAA was responded to with a DNS answer

with RCODE=3 (Name Error), then the responder's information is not present in the DNS and further queries SHOULD NOT be made.

In case the query for the address records returned a DNS answer with RCODE=0 (No Error), then the initiator sends out two queries: One for the HIPHI type and one for the HIPRVS type at the responder's FQDN.

Depending on the combinations of answer the situations described in Section 3.1, Section 3.2 and Section 3.3 can occur.

Note that storing HIP RR information in the DNS at a FQDN which is assigned to a non-HIP node might have ill effects on its reachability by HIP nodes.

3.1 Simple static singly homed end-host

A HIP node (R) with a single static network attachment, wishing to be reachable by reference to its FQDN (www.example.com), would store in the DNS, in addition to its IP address(es) (IP-R), its Host Identity (HI-R) in a HIPHI resource record.

An initiator willing to associate with a node would typically issue the following queries:

QNAME=www.example.com, QTYPE=A

(QCLASS=IN is assumed and ommitted from the examples)

Which returns a DNS packet with RCODE=0 and one or more A RRs A with the address of the responder (e.g. IP-R) in the answer section.

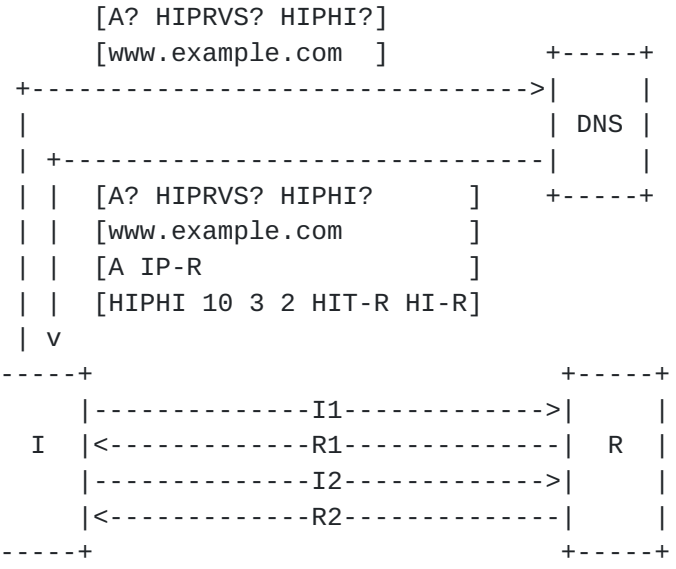
QNAME=www.example.com, QTYPE=HIPHI

Which returns a DNS packet with RCODE=0 and one or more HIPHI RRs with the HIT and HI (e.g. HIT-R and HI-R) of the responder in the answer section.

QNAME=www.example.com, QTYPE=HIPRVS

Which returns a DNS packet with RCODE=0 and an empty answer section.

Caption: In the remainder of this document, for the sake of keeping diagrams simple and concise, several DNS queries and answers are represented as one single transaction, while in fact there are several queries and answers flowing back and forth, as described in the textual examples.



3.2 Mobile end-host

A mobile HIP node (R) wishing to be reachable by reference to its FQDN (www.example.com) would store in the DNS, possibly in addition to its IP address(es) (IP-R), its HI (HI-R) in a HIPHI RR, and the IP address(es) of its Rendezvous Server(s) (IP-RVS) in HIPRVS resource record(s). The mobile HIP node also need to notify its Rendezvous Servers of any change in its set of IP address(es).

An initiator willing to associate with such mobile node would typically issue the following queries:

QNAME=www.example.com, QTYPE=A

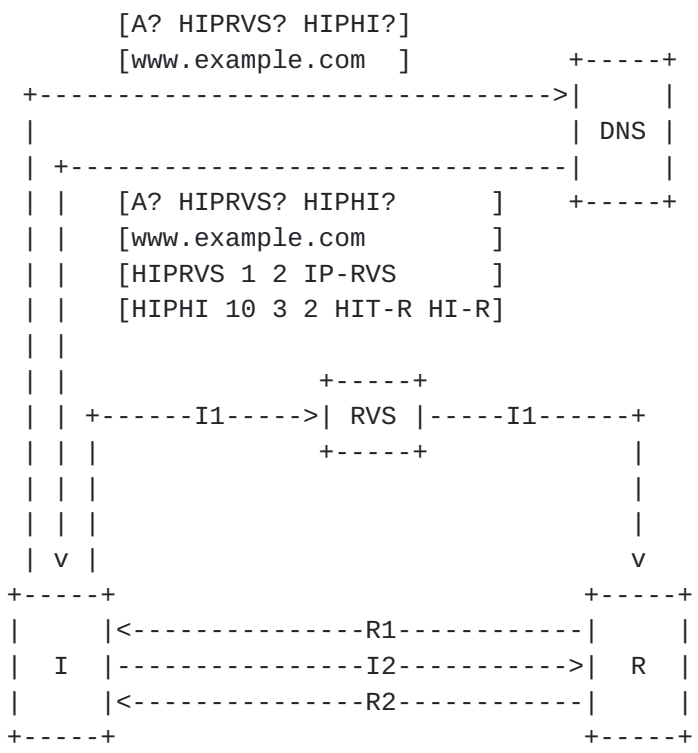
Which returns a DNS packet with RCODE=0 and an empty answer section.

QNAME=www.example.com, QTYPE=HIPHI

Which returns a DNS packet with RCODE=0 and one or more HIPHI RRs with the HIT and HI (e.g HIT-R and HI-R) of the responder in the answer section.

QNAME=www.example.com, QTYPE=HIPRVS

Which returns a DNS packet with RCODE=0 and one or more HIPRVS RRs containing IP address(es) (e.g IP-RVS) or FQDN(s) of RVS(s).



The initiator would then send an I1 to one of its RVS. Following, the RVS will relay the I1 up to the mobile node, which will complete the HIP exchange.

3.3 Mixed Scenario

A HIP node might be configured with more than one IP address (multi-homed), or Rendezvous Server (multi-reachable). In these cases, it is possible that the DNS returns multiple A or AAAA RRs, as well as HIPRVS containing one or multiple Rendezvous Servers. In addition to its set of IP address(es) (IP-R1, IP-R2), a multi-homed end-host would store in the DNS its HI (HI-R) in a HIPHI RR, and possibly the IP address(es) of its RVS(s) (IP-RVS1, IP-RVS2) in HIPRVS RRs.

An initiator willing to associate with such a node would typically issue the following queries:

QNAME=www.example.com, QTYPE=A

Which returns a DNS packet with RCODE=0 and one or more A or AAAA RRs

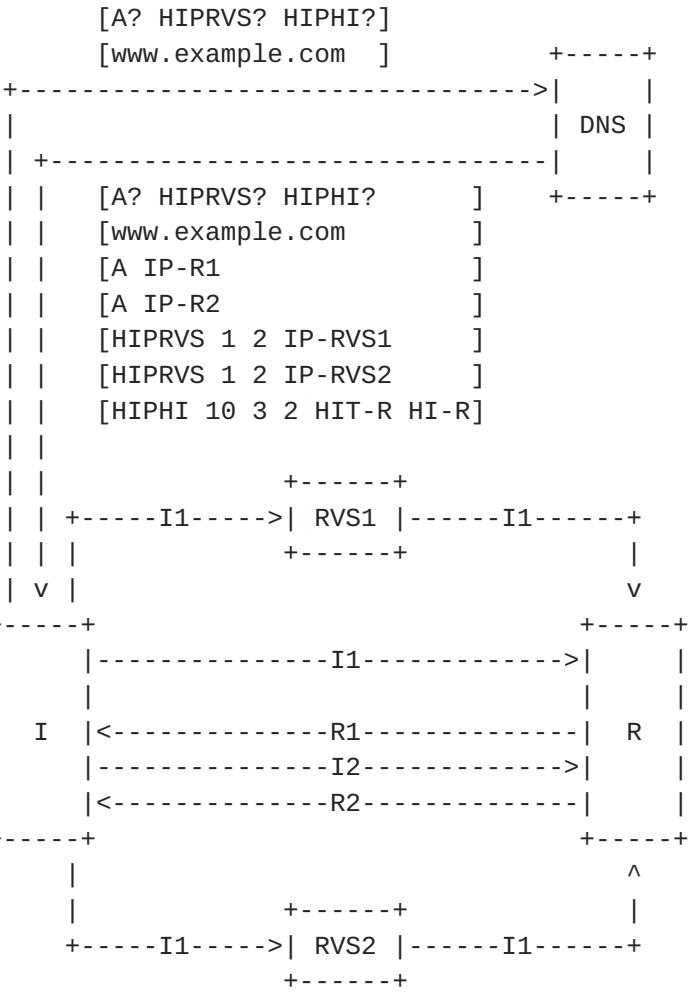
containing IP address(es) (e.g IP-R1 and IP-R2) in the answer section.

QNAME=www.example.com, QTYPE=HIPHI

Which returns a DNS packet with RCODE=0 and one or more HIPHI RRs with the HIT and HI (e.g HIT-R and HI-R) of the responder in the answer section.

QNAME=www.example.com, QTYPE=HIPRVS

Which returns a DNS packet with RCODE=0 and one or more HIPRVS RRs containing IP address(es) (e.g IP-RVS1, IP-RVS2) or FQDN(s) of RVS(s).



4. Overview of using the DNS with HIP

4.1 Storing HI and HIT in DNS

Any conforming implementation may store Host Identifiers in a DNS HIPHI RDATA format. An implementation may also store a HIT along with its associated HI. If a particular form of an HI or HIT does not already have a specified RDATA format, a new RDATA-like format SHOULD be defined for the HI or HIT.

4.1.1 Different types of HITs

There are two types of HITs. HITs of the first type, called Type 1 HIT, consist of an 8-bit prefix followed by 120 bits of the hash of the public key. HITs of the second type (Type 2 HIT) consist of a Host Assigning Authority Field (HAA), and only the last 64 bits come from a SHA-1 hash of the Host Identity. This latter format for HIT is recommended for 'well known' systems. It is possible to support a resolution mechanism for these names in hierarchical directories, like the DNS.

This document fully specifies only Type 2 HITs. Type 1 HITs are specified in Section 3.1 of [11].

Note that the format how HITs are stored in the HIPHI RRs may be different form the format actually used in protocols, the HIP base exchange [11] included. This is because the DNS RR explicitly contains the HIT type and algorithm, while some protocols may prefer to use a prefix to indicate the HIT type. The implementations are expected to use the actual HI when comparing Host Identities.

4.1.1.1 Host Assigning Authority (HAA) field

The 64 bits of HAA supports two levels of delegation. The first is a registered assigning authority (RAA). The second is a registered identity (RI, commonly a company). The RAA is 24 bits with values assign sequentially by ICANN. The RI is 40 bits, also assigned sequentially but by the RAA.

4.1.1.2 Storing HAA in HIPHI Resource Records

Any conforming implementation may store a domain name Host Assigning Authority (HAA) in a DNS HIPHI RDATA format. A HAA MUST be stored like a Type 2 HIT, while the least significant bits of the HIT extracted from the HI hash output are set to zero, the Host Identity Length is set zero, and the Host Identity field is omitted. If a particular form of a HAA does not already have an associated HIT specified RDATA format, a new RDATA-like format SHOULD be defined for

the HIT/HAA.

4.1.1.3 HI and HIT verification

Upon return of a HIPHI RR, a host MUST always calculate the HI-derivative HIT to be used in the HIP exchange, as specified in the HIP architecture [12], while the HIT possibly embedded along SHOULD only be used as an optimization (e.g. table lookup).

4.2 Storing Rendezvous Servers in the DNS

The HIP Rendezvous server (HIPRVS) resource record indicates an address or a domain name of a RendezVous Server, towards which a HIP I1 packet might be sent to trigger the establishment of an association with the entity named by this resource record [14].

An RVS receiving such an I1 would then relay it to the appropriate responder (the owner of the I1 receiver HIT). The responder will then complete the exchange with the initiator, typically without ongoing help from the RVS.

Any conforming implementation may store Rendezvous Server's IP address(es) or DNS name in a DNS HIPRVS RDATA format. If a particular form of a RVS reference does not already have a specified RDATA format, a new RDATA-like format SHOULD be defined for the RVS.

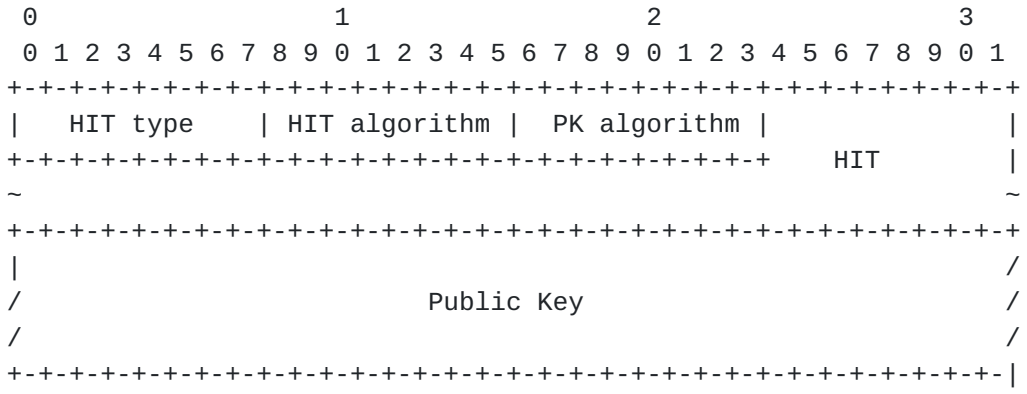
4.3 Initiating connections based on DNS names

On a HIP node, a Host Identity Protocol exchange SHOULD be initiated whenever an Upper Layer Protocol attempt to communicate with an entity and the DNS lookup returns HIPHI and/or HIPRVS resource records. If a DNS lookup returns one or more HIPRVS RRs and no A nor AAAA RRs, the afore mentioned HIP exchange SHOULD be initiated towards one of these RVS [11]. Since some hosts may choose not to have HIPHI information in DNS, hosts MAY implement support for opportunistic HIP.

5. Storage Format

5.1 HIPHI RDATA format

The RDATA for a HIPHI RR consists of a HIT type, an algorithm type, a HIT, and a public key.



5.1.1 HIT type format

The HIT type field indicates the Host Identity Tag (HIT) type and the implied HIT format.

The following values are defined:

0	No HIT is present
1	A Type 1 HIT is present
2	A Type 2 HIT is present
3-6	Unassigned
7	A HAA is present

5.1.2 HIT algorithm format

The HIT algorithm indicates the hash algorithm used to generate the Host Identity Tag (HIT) from the HI.

The following values are defined:

0	Reserved
1	SHA1
2-255	Unassigned

5.1.3 PK algorithm format

The PK algorithm field indicates the public key cryptographic algorithm and the implied public key field format. This document reuse the values defined for the 'algorithm type' of the IPSECKEY RR [10] 'gateway type' field.

The presently defined values are given only informally:

1 A DSA key is present, in the format defined in RFC2536 [5].

2 A RSA key is present, in the format defined in RFC3110 [6].

5.1.4 HIT format

There's currently two types of HITs, and a single type of HAA. Both of them are stored in network byte order within a self-describing variable length wire-encoded <character-string> (as per Section 3.3 of [2]):

- A *Type 1* HIT: least significant bits of the hash (e.g. SHA1) of the public key (Host Identity), which is possibly following in the HIPHI RR.
- A *Type 2* HIT: binary prefix (HAA) concatenated with a the least significant bits of the hash (e.g. SHA1) of the public key (Host Identity), which is possibly following in the HIPHI RR.
- A HAA: binary prefix (HAA) concatenated with 0, up to the associated HIT length.

5.1.5 Public key format

Both of the public key types defined in this document (RSA and DSA) reuse the public key formats defined for the IPSECKEY RR [10] (which in turns contains the algorithm-specific portion of the KEY RR RDATA, all of the KEY RR DATA after the first four octets, corresponding to the same portion of the KEY RR that must be specified by documents that define a DNSSEC algorithm).

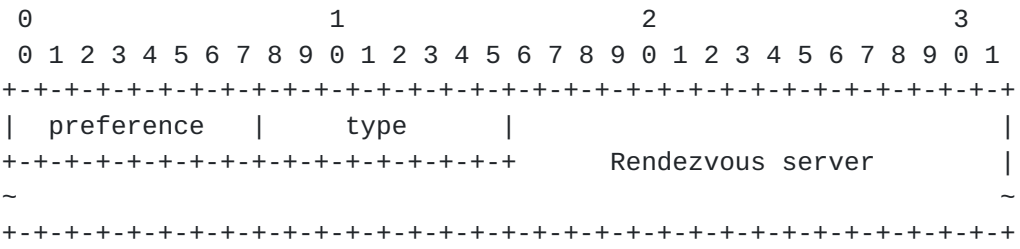
In the future, if a new algorithm is to be used both by IPSECKEY RR and HIPHI RR, it would probably use the same public key encodings for both RRs. Unless specified otherwise, the HIPHI public key field would use the same public key format as the IPSECKEY RR RDATA for the corresponding algorithm.

The DSA key format is defined in RFC2536 [5].

The RSA key format is defined in RFC3110 [6] and the RSA key size limit (4096 bits) is relaxed in the IPSECKEY RR [10] specification.

5.2 HIPRVS RDATA format

The RDATA for a HIPRVS RR consists of a preference value, a Rendezvous server type and either one or more Rendezvous server address, or one Rendezvous server domain name.



5.2.1 Preference format

This is an unsigned 8-bit value, used to specify the preference given to the RVS in the HIPRVS RR amongst others at the same owner. RVSs with lower values are preferred. If there is a tie within some RR subset, the initiating HIP host should pick one of the RVS randomly from the set of RRs, such that the requester load is fairly balanced amongst all RVSs of the set.

5.2.2 Rendezvous server type format

The Rendezvous server type indicates the format of the information stored in the Rendezvous server field.

This document reuses the type values for the 'gateway type' field of the IPSECKEY RR [10]. The presently defined values are given only informally:

1. One or more 4-byte IPv4 address(es) in network byte order are present.

2. One or more 16-byte IPv6 address(es) in network byte order are present.

3. One or more variable length wire-encoded domain names as described in section 3.3 of RFC1035 [2]. The wire-encoded format is self-describing, so the length is implicit. The domain names MUST NOT be compressed.

5.2.3 Rendezvous server format

The Rendezvous server field indicates one or more Rendezvous Server(s) IP address(es), or domain name(s). A HIP I1 packet sent to any of these RVS would reach the entity named by this resource record.

This document reuses the format used for the 'gateway' field of the IPSECKEY RR [10], but allows to concatenate several IP (v4 or v6) addresses. The presently defined formats for the data portion of the Rendezvous server field are given only informally:

- One or more 32-bit IPv4 address(es) in network byte order.

- One or more 128-bit IPv6 address(es) in network byte order.

- One or more variable length wire-encoded domain names as described in section 3.3 of RFC1035 [2]. The wire-encoded format is self-describing, so the length is implicit. The domain names MUST NOT be compressed.

6. Presentation Format

This section specifies the representation of the HIPHI and HIPRVS RR in a zone data master file.

6.1 HIPHI Representation

The HIT Type, HIT algorithm, PK algorithm, HIT and Public Key are REQUIRED.

The HIT Type, HIT algorithm, and PK algorithm are represented as unsigned integers.

The HIT field is represented as the Base16 encoding [8] (a.k.a. hex or hexadecimal) of the public key hash. The encoding MUST NOT contains whitespace. If no HIT is to be indicated, then the HIT algorithm MUST be zero and the HIT field must be "." (a single dot character).

The Public Key field is represented as the Base64 encoding [8] of the public key. The encoding MAY contains whitespace(s), and such whitespaces MUST be ignored.

The complete representation of the HPIHI record is:

```
IN  HIPHI ( hit-type hit-algorithm pk-algorithm
            base16-encoded-hit
            base64-encoded-public-key )
```

6.2 HIPRVS Representation

The Preference and RVS Type fields are REQUIRED. At least one RVS field MUST be present.

The HIT Type, HIT algorithm, and PK algorithm are represented as unsigned integers.

The RVS field is represented by one or more:

- IPv4 dotted decimal address(es)
- IPv6 colon hex address(es)
- uncompressed domain name(s)

The complete representation of the HPIRVS record is:

```
   IN  HIPRVS  ( preference rendezvous-server-type
                 rendezvous-server[1]
                         ...
                 rendezvous-server[n] )
```

6.3 Examples

Example of a node with a HI but no HIT:

```
www.example.com  IN  HIPHI ( 0 1 2
                             .
                             AB3NzaC1kc3MAAACBAOBhKn
                             TCPOuFBzZQX/N3O9dm9P9iv
                             UIMoId== )
```

Example of a node with a HI and a HIT:

```
www.example.com  IN  HIPHI ( 1 1 2
                             AB3NzaC1kc3MAAACBAOBhKn
                             TCPOuFBzZQX/N3O9dm9P9iv
                             UIMoId== )
```

Example of a node with an IPv6 RVS:

```
www.example.com  IN  HIPRVS (10 2 2001:db8:200:1:20c:f1ff:feb:a533 )
```

Example of a node with three IPv4 RVS:

```
www.example.com  IN  HIPRVS ( 10 1 192.0.1.2 192.0.2.2 192.0.3.2 )
```

Example of a node with two named RVS:

```
www.example.com  IN  HIPRVS ( 10 3 rvs.uk.example.com
                                   rvs.us.example.com )
```

7. Retrieving Multiple HITs and IPs from the DNS

If a host receives multiple HITs in a response to a DNS query, those HITs MUST be considered to denote a single service, and be semantically equivalent from that point of view. When initiating a base exchange with the denoted service, the host SHOULD be prepared to accept any of HITs as the peer's identity. A host MAY implement this by using the opportunistic mode (destination HIT null in I1), or by sending multiple I1s, if needed.

In particular, if a host receives multiple HITs and multiple IP addresses in response to a DNS query, the host cannot know how the HITs are reachable at the listed IP addresses. The mapping may be any, i.e., all HITs may be reachable at all of the listed IP addresses, some of the HITs may be reachable at some of the IP addresses, or there may even be one-to-one mapping between the HITs and IP addresses. In general, the host cannot know the mapping and MUST NOT expect any particular mapping.

It is RECOMMENDED that if a host receives multiple HITs, the host SHOULD first try to initiate the base exchange by using the opportunistic mode. If the returned HIT does not match with any of the expected HITs, the host SHOULD retry by sending further I1s, one at time, trying out all of the listed HITs. If the host receives an R1 for any of the I1s, the host SHOULD continue to use the successful IP address until an R1 with a listed HIT is received, or the host has tried all HITs, and try the other IP addresses only after that. A host MAY also send multiple I1s in parallel, but sending such I1s MUST be rate limited to avoid flooding (as per Section 8.4.1 of [11]).

8. Security Considerations

Though the security considerations of the HIP DNS extensions still need to be more investigated and documented, this section contains a description of the known threats involved with the usage of the HIP DNS extensions.

In a manner similar to the IPSECKEY RR [10], the HIP DNS Extensions allows to provision two HIP nodes with the public keying material (HI) of their peer. These HIs will be subsequently used in a key exchange between the peers. Hence, the HIP DNS Extensions introduce the same kind of threats that IPSECKEY does, plus threats caused by the possibility given to a HIP node to initiate or accept a HIP exchange using "Opportunistic" or "Unpublished Initiator HI" modes.

A HIP node SHOULD obtain both the HIPHI and HIPRVS RRs from a trusted party trough a secure channel insuring proper data integrity of the RRs. DNSSEC [3] provides such a secure channel.

In the absence of a proper secure channel, both parties are vulnerable to MitM and DoS attacks, and unrelated parties might be subject to DoS attacks as well. These threats are described in the following sections.

8.1 Attacker tampering with an unsecure HIPHI RR

The HIPHI RR contains public keying material in the form of the named peer's public key (the HI) and its secure hash (the HIT). Both of these are not sensitive to attacks where an adversary gains knowledge of them. However, an attacker that is able to mount an active attack on the DNS, i.e., tampers with this HIPHI RR (e.g. using DNS spoofing) is able to mount Man-in-the-Middle attacks on the cryptographic core of the eventual HIP exchange (responder's HIPHI and HIPRVS rewritten by the attacker).

8.2 Attacker tampering with an unsecure HIPRVS RR

The HIPRVS RR contains a destination IP address where the named peer is reachable by an I1 (HIP Rendezvous Extensions IPSECKEY RR [14]). Thus, an attacker able to tamper with this RRs is able to redirect I1 packets sent to the named peer to a chosen IP address, for DoS or MitM attacks. Note that this kind of attacks are not specific to HIP and exist independently to whether or not HIP and the HIPRVS RR are used. Such an attacker might tamper with A and AAAA RRs as well.

An attacker might obviously use these two attacks in conjunction: It will replace the responder's HI and RVS IP address by its owns in a spoofed DNS packet sent to the initiator HI, then redirect all

exchanged packets through him and mount a MitM on HIP. In this case HIP won't provide confidentiality nor initiator HI protection from eavesdroppers.

8.3 Opportunistic HIP

A HIP initiator may not be aware of its peer's HI, and/or its HIT (e.g. because the DNS does not contains HIP material, or the resolver isn't HIP-enabled), and attempt an opportunistic HIP exchange towards its known IP address, filling the responder HIT field with zeros in the I1 header. Such an initiator is vulnerable to a MitM attack because it can't validate the HI and HIT contained in a replied R1. Hence, an implementation MAY choose not to use opportunistic mode.

8.4 Unpublished Initiator HI

A HIP initiator may choose to use an unpublished HI, which is not stored in the DNS by means of a HIPHI RR. A responder associating with such an initiator knowingly risks a MitM attack because it cannot validate the initiator's HI. Hence, an implementation MAY choose not to use unpublished mode.

8.5 Hash and HITs Collisions

As many cryptographic algorithm, some secure hashes (e.g. SHA1, used by HIP to generate a HIT from an HI) eventually become insecure, because an exploit has been found in which an attacker with a reasonable computation power breaks one of the security features of the hash (e.g. its supposed collision resistance). This is why a HIP end-node implementation SHOULD NOT authenticate its HIP peers based solely on a HIT retrieved from DNS, but SHOULD rather use HI-based authentication.

8.6 DNSSEC

In the absence of DNSSEC, the HIPHI and HIPRVS RRs are subject to the threats described in RFC 3833 [17].

9. IANA Considerations

IANA needs to allocate two new RR type code for HIPHI and HIPRVS from the standard RR type space.

IANA needs to open a new registry for the HIPHI RR HIT type. Defined types are:

0	No HIT is present
1	A Type 1 HIT is present
2	A Type 2 HIT is present
3-6	Unassigned
7	A HAA is present

Adding new reservations requires IETF consensus RFC2434 [16].

IANA needs to open a new registry for the HIPHI RR HIT algorithm. Defined types are:

0	Reserved
1	SHA1
2-255	Unassigned

Adding new reservations requires IETF consensus RFC2434 [16].

IANA does not need to open a new registry for the HIPHI RR type for public key algorithms because the HIPHI RR reuse 'algorithms types' defined for the IPSECKEY RR [10]. The presently defined numbers are given here only informally:

0 is reserved

1 is RSA

2 is DSA

IANA does not need to open a new registry for the HIPRVS RR Rendezvous server type because the HIPHI RR reuse the 'gateway types' defined for the IPSECKEY RR [10]. The presently defined numbers are given here only informally:

0 is reserved

1 is IPv4

2 is IPv6

3 is a wire-encoded uncompressed domain name

10. Acknowledgments

As usual in the IETF, this document is the result of a collaboration between many people. The authors would like to thanks the author (Michael Richardson), contributors and reviewers of the IPSECKEY RR [10] specification, which this document was framed after. The authors would also like to thanks the following people, who have provided thoughtful and helpful discussions and/or suggestions, that have helped improving this document: Rob Austein, Hannu Flinck, Tom Henderson, Olaf Kolkman, Miika Komu, Andrew McGregor, Erik Nordmark, and Gabriel Montenegro. Some parts of this draft stem from [11].

Julien Laganier is partly funded by Ambient Networks, a research project supported by the European Commission under its Sixth Framework Program. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Ambient Networks project or the European Commission.

11. References

11.1 Normative references

[1] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

[2] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

[3] Eastlake, D. and C. Kaufman, "Domain Name System Security Extensions", RFC 2065, January 1997.

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

[5] Eastlake, D., "DSA KEYs and SIGs in the Domain Name System (DNS)", RFC 2536, March 1999.

[6] Eastlake, D., "RSA/SHA-1 SIGs and RSA KEYs in the Domain Name System (DNS)", RFC 3110, May 2001.

[7] Bush, R., Durand, A., Fink, B., Gudmundsson, O., and T. Hain, "Representing Internet Protocol version 6 (IPv6) Addresses in the Domain Name System (DNS)", RFC 3363, August 2002.

[8] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 3548, July 2003.

[9] Thomson, S., Huitema, C., Ksinant, V., and M. Souissi, "DNS Extensions to Support IP Version 6", RFC 3596, October 2003.

[10] Richardson, M., "A Method for Storing IPsec Keying Material in DNS", RFC 4025, March 2005.

[11] Moskowitz, R., "Host Identity Protocol", draft-ietf-hip-base-03 (work in progress), June 2005.

[12] Moskowitz, R., "Host Identity Protocol Architecture", draft-ietf-hip-arch-02 (work in progress), January 2005.

[13] Nikander, P., "End-Host Mobility and Multi-Homing with Host Identity Protocol", draft-ietf-hip-mm-01 (work in progress), February 2005.

[14] Laganier, J. and L. Eggert, "Host Identity Protocol (HIP) Rendezvous Extension", draft-ietf-hip-rvs-02 (work in progress), June 2005.

11.2 Informative references

[15] Jokela, P., "Using ESP transport format with HIP", draft-jokela-hip-esp-00 (work in progress), February 2005.

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

[17] Atkins, D. and R. Austein, "Threat Analysis of the Domain Name System (DNS)", RFC 3833, August 2004.

Authors' Addresses

Pekka Nikander
Ericsson Research Nomadic Lab
JORVAS FIN-02420
FINLAND

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

Julien Laganier
DoCoMo Communications Laboratories Europe GmbH
Landsberger Strasse 312
Munich 80687
Germany

Phone: +49 89 56824 231
Email: julien.ietf@laposte.net
URI: http://www.docomolab-euro.com/

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

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