Network Working Group P. Nikander
Internet-Draft Ericsson Research Nomadic Lab
Expires: March 6, 2006 J. Laganier
DoCoMo Euro-Labs
F. Dupont
Point6
September 2, 2005

A Non-Routable IPv6 Prefix for Keyed Hash Identifiers (KHI)
draft-laganier-ipv6-khi-00

Status of this Memo

Copyright Notice

Abstract

This document introduces Keyed Hash Identifiers (KHI) as a new, experimental class of IPv6-address-lookalike identifiers. They are

constructed to be statistically globally unique. They are intended to be used as identifiers only, and not as locators. They should not appear in actual IPv6 headers. Consequently, they are considered as non-routable addresses from the IPv6 point of view.

These identifiers are expected to be used at the existing IPv6 API and application protocols between consenting hosts. They may be defined and used in different contexts, suitable for different protocols. Examples of these include Host Identity Tags (HIT) in the Host Identity Protocol (HIP) and Temporary Mobile Identifiers (TMI) for Mobile IPv6 Privacy Extension.

This document requests IANA to allocate a temporary prefix out of the IPv6 addressing space for Keyed Hash Identifiers.

1. Introduction

This document introduces Keyed Hash Identifiers (KHI), a new class of IPv6-address-lookalike identifiers. They are intended to be statistically unique and non-routable at the IP layer. The identifiers are designed to be securely bound to a bitstring used as input to a secure hash function, keyed with a context identifier. Typically, but not necessarily, the input bitstring will include a suitably encoded public cryptographic key.

These identifiers have the following properties:

- Statistically unique (i.e. high probability uniqueness.)
- Securely bound to the input parameters used for their generation (i.e., the context identifier and a bitstring.)
- Conforming with the IPv6 global unicast address format defined in Section 2.5.4 of [RFC3513].
- Aggregated under the TBD IPv6 prefix.
- Non-routable as IPv6 addresses, due to their structure and identifier-only nature.

As mentioned above, KHIs are intended to be generated and used in different contexts, as suitable for different mechanisms and protocols. The context identifier is meant to be used to differentiate between the different contexts; see Section 4 for a discussion of the related API and kernel level implementation issues, and Section 5 for the design choices behind it.

Examples of identifiers and protocols that are expected to adopt the KHI format include Host Identity Tags (HIT) in the Host Identity Protocol [I-D.ietf-hip-base] and the Temporary Mobile Identifiers (TMI) in the Simple Privacy Extension for Mobile IPv6 [I-D.dupont-mip6-privacyext]. The format is designed to be extensible to allow other experimental proposals to share the same name space.

This document requests IANA to allocate a temporary prefix out of the IPv6 addressing space for Keyed Hash Identifiers. By default, the prefix will be returned to IANA in January 1st 2009, continued use requiring IETF consensus.

2. Keyed Hash Identifier Construction

A KHI is generated using the algorithm below, which takes as input a bitstring and a context identifier:

```
Input      :=  any bitstring
Hash Input :=  Context ID | Input
Hash       :=  SHA1( Expand( Hash Input ) )
KHI        :=  Prefix | Encode_n( Hash )
```

where:

| : Denotes concatenation of bitstrings

Input : A bitstring unique or statistically unique within a given context intended to be associated with the to-be-created KHI in the given context.

Context ID : A randomly generated value defining the expected usage context the the particular KHI.

As a baseline (TO BE DISCUSSED), we propose sharing the name space introduced for CGA Type Tags; see http://www.iana.org/assignments/cga-message-types and RFC 3972.

Expand( ) : An expansion function designed to overcome recent attacks on SHA1.

As a baseline (TO BE DISCUSSED), we propose inserting four (4) zero (0) bytes after every twelve (12) bytes of the argument bitstring.

Encode_n( ): An extraction function which output is obtained by extracting an <n>-bits-long bitstring from the argument bitstring.

As a baseline (TO BE DISCUSSED), we propose taking <n> middlemost bits from the SHA1 output.

Prefix : A constant (128 - <n>) bits long bitstring value, TBD, assigned by IANA.

3. Routing Considerations

Keyed Hash Identifiers are designed to serve as identifiers rather than locators. Therefore, routers SHOULD NOT forward any packets containing a KHI as a source or a destination address. If the destination address is a KHI but the source address is a valid unicast source address, an ICMP Destination Unreachable, Administratively Prohibited message MAY be generated.

Note that while KHIs are designed to be non-routable at the IP layer, there are ongoing research efforts for creating overlay routing for these kinds of identifiers. For example, the Host Identity Indirection Infrastructure (Hi3) proposal outlines a way for using a Distributed Hash Table to forward HIP packets based on the Host Identity Tag.

4. Collision Considerations

As noted above, KHIs are expected to be used at the legacy IPv6 APIs between consenting hosts. The context ID is intended to differentiate between the various mechanisms, or contexts, sharing the same name space. However, that context ID not being present in the KHI itself, but only in front of the input bitstring as an input to the hash function, might lead to certain implementation-related complications.

Because KHIs are not routable, in order to send packets using KHIs at the API level, the sending host must have additional state within the stack in order to determine parameters (e.g. what locators) to use in the outgoing packet. An underlying assumption here, and a matter of fact in the proposals that the authors are aware of, is that there is a protocol for setting up and maintaining the additional state. It is assumed that the state-set-up protocol carries the input bitstring, and that the resulting KHI-related state in the stack can be associated back with the appropriate context and state-set-up protocol.

Even though KHI collisions are expected to be extremely rare, two kinds of collisions may happen. Firstly, it is possible that two different input bitstrings within the same context may map to the same KHI. In that case, the state-set-up might be able to resolve the conflict.

A second type of collision may happen if two different input bitstrings, used in different usage contexts, map to the same KHI. In this case the main confusion is about which context to use. In order to prevent these types of collisions, it is RECOMMENDED that implementations that simultaneously support multiple different

contexts maintain a host-wide unified database of known KHIs, and indicate a conflict if any of the mechanisms attempt to register a KHI that is already in use. For example, if a given KHI is already being used as a HIT in HIP, it cannot be simultaneously used as a TMI in Mobile IP. Instead, if Mobile IP attempts to use the KHI, it will be notified (by the kernel) that the KHI in question is already in use.

5. Design Choices

The design of this name space faces two competing forces:

As many bits as possible should be preserved for the hash result.
It should be possible to share the name space between multiple mechanisms.

The desire to have a long hash result requires the prefix to be as short as possible, and to use few (if any) bits for additional encoding. The present design takes this desire to the maxim: all the bits beyond the prefix are used as hash output. This leaves no bits in the KHI itself available for separating the context. Additionally, due to security considerations, the present design REQUIRES that the hash function used in constructing KHIs is constant; see Section 6.

The authors explicitly considered including a hash extension mechanism, similar to the one in CGA [RFC3972], but decided to leave it out. There were two reasons: desire for simplicity, and the somewhat unclear IPR situation around the hash extension mechanism. If there is a future revision of this document, we strongly advice the future authors to reconsider the situation.

The desire to allow multiple mechanism to share the name space has been resolved by including the context identifier in the hash function input. While this does not allow the mechanism to be directly inferred from a KHI, it allows one to verify that a given input bitstring and KHI belong to a given context, with high probability; but see also Section 6.

6. Security Considerations

Keyed Hash Identifiers are designed to be securely bound to the context identifier and the bitstring used as the input parameters during their generation. To provide this property, the KHI generation algorithm relies on the second-preimage resistance (a.k.a. one-way) property of the hash function used in the generation [I-D.hoffman-hash-attacks]. To have this property, and to avoid collisions, it is important that the allocated prefix is as short as possible, leaving as many bits as possible for the hash output.

All mechanism using KHIs MUST use exactly the same mechanism for generating a KHI from the input bitstring. Allowing different mechanisms, without explicitly encoding the mechanism in the KHI itself, would allow so called bidding down attacks. That is, if multiple different hash functions were allowed in constructing KHIs in a given shared name space, and if one of the hash functions became insecure, that would allow attacks against even those KHIs that had been constructed using with the other, still secure hash functions.

Due to the desire to keep the hash output value as long as possible, the present design allows only one method for constructing KHIs from input bitstrings. If other methods (perhaps using more secure hash functions) are later needed, they MUST use a different prefix. Consequently, the suggested method to react to the hash result becoming too short due to increased computational power or the used hash function becoming insecure due to advances in cryptology is to allocate a new prefix and cease to use the present one.

As of today, SHA1 applied in conjunction with a proper expansion function of the hash input is considered as satisfying the second-preimage resistance requirement [I-D.hoffman-hash-attacks]. Hash output of at least 100 bits, but preferably up to 120 bits, is considered to have a low enough probability of collisions.

In order to preserve low enough probability of collisions (see Section 4), each method MUST utilize a mechanism that makes sure that the distinct input bitstrings are either unique or statistically unique, within that context. There are several possible methods to ensure that; for example, one can include into the input bitstring a globally maintained counter value, a pseudo- random number of sufficient entropy (minimum 120 bits), or a randomly generated public cryptographic key. The Context ID makes sure that input bitstrings from different contexts never overlap. These together make sure that the probability of collisions is determined only by the probability of natural collisions in the hash space and not increased by a possibility of colliding input bit strings.

7. IANA Considerations

IANA is requested to allocate a temporary non-routable prefix from the IPv6 address space, to be defaulted back to "Reserved by IETF" by January 1st 2009. As per Sections 2.5.1 and 2.5.4 of [RFC3513], the prefix must be allocated from the 0000::/3 block, since KHIs do not have a 64-bit interface identifier part. The allocation will require updating http://www.iana.org/assignments/ipv6-address-space

As a baseline (TO BE DISCUSSED), we propose an 8-bit prefix to be allocated from the 1000::/4 block.

The Context Identifier (or Context ID) is a randomly generated value defining the usage context of a KHI. This document defines no specific value.

As a baseline (TO BE DISCUSSED), we propose sharing the name space introduced for CGA Type Tags. Hence, defining new values would follow the rules of Section 8 of [RFC3972], i.e., on a First Come First Served basis. The policy will require updating the policy for http://www.iana.org/assignments/cga-message-types

8. Acknowledgments

Julien Laganier is partly funded by Ambient Networks, a research project supported by the European Commission under its Sixth Framework Program.

9. References

9.1 Normative references

[RFC3513] Hinden, R. and S. Deering, "Internet Protocol Version 6 (IPv6) Addressing Architecture", RFC 3513, April 2003.

[RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, March 2005.

9.2 Informative references

[I-D.dupont-mip6-privacyext]
Castelluccia, C., Dupont, F., and G. Montenegro, "A Simple Privacy Extension for Mobile IPv6", draft-dupont-mip6-privacyext-02 (work in progress), July 2005.

[I-D.hoffman-hash-attacks]
Hoffman, P. and B. Schneier, "Attacks on Cryptographic Hashes in Internet Protocols", draft-hoffman-hash-attacks-04 (work in progress), June 2005.

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

[RFC3587] Hinden, R., Deering, S., and E. Nordmark, "IPv6 Global Unicast Address Format", RFC 3587, August 2003.

Authors' Addresses

Pekka Nikander
Ericsson Research Nomadic Lab
JORVAS FI-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/

Francis Dupont
Point6
c/o GET/ENST Bretagne
2 rue de la Chataigneraie
CS 17607
35576 Cesson-Sevigne Cedex
France

Fax: +33 2 99 12 70 30
Email: Francis.Dupont@enst-bretagne.fr

Intellectual Property Statement

Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Copyright Statement

Acknowledgment

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