Internet Draft
Document: draft-van-beijnum-multi6-cbhi-00.txt
Expires: July 2004

I. van Beijnum

January 2004

Crypto Based Host Identifiers

Abstract

This memo specifies a 64-bit crypto-based host identifier that can be used as an interface identifier in protocols that allow address agility, such as [ODT]. The cryptographic nature of the host identifier makes it possible to determine whether a correspondent is legitimately using said host identifier or not.

The host identifiers can be used as regular interface identifiers in protocols that don't require an identifier that is separate from locators, or they can be expanded to 128-bit IPv6 address like values for use with protocols that do need such an identifier-only value.

1. Introduction

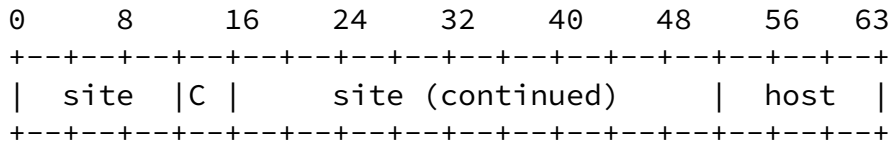
In many types of interactions across the network it is important to know the identity of the correspondent. This is especially true in multihoming and mobility, where a correspondent may change its address during a session. In [MIPv6], [NOID] and [ODT] it has been shown to be possible to solve mobility and multihoming without introducing a long-lived host or stack name identifier. However,

this doesn't mean that having such an identifier would be without benefits. This document explores the possibility of adding a means

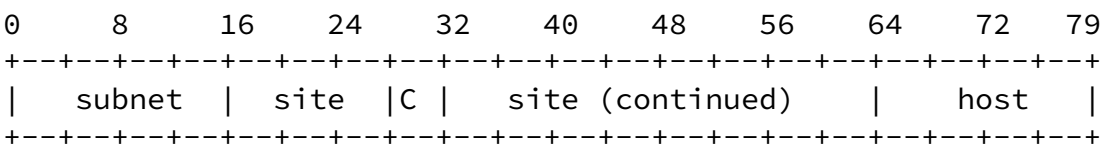
to identify a host independent of the full IPv6 address used by the host and independent of a specific multihoming or mobility solution.

2. Overview

There are two types of crypto-based host identifiers 64 bit and 80 bit. The 64 bit type consists of 4 control bits, 48 site key bits and a 12 bit host number:



The 64 bit host identifiers are appropriate in cases where the subnet bits (bits 48 - 63 in the IPv6 address) are subject to change, for instance in a host multihoming or mobility situation. When the subnet bits are fixed, which is likely to be the case with site multihoming or when no address changes are expected, 80 bit host identifiers that include the subnet bits are more appropriate, as these allow significantly more hosts to be grouped together in a site. The 80 bit host identifier consists of 4 control bits, 44 site key bits, a 16 bit host number and a 16 bit subnet number:



The control bits are:

- reserved
- 80 bit identifier (1) or 64 bit identifier (0)
- u/l and g bits as outlined below

3. EUI-64 Compatibility

Generally, a host will create IPv6 addresses for interfaces based on the interface's EUI-64 as outlined in [RFC 2462]. In order to

avoid overlap from addresses generated by [RFC 3041] and from regular EUI-64 interface identifiers, for crypto-based host identifiers the universal/global bit is set to "universal" and the group bit is set to indicate a group (multicast) address. Note that the resulting EUI-64 value is only valid for the purposes of generating IPv6 addresses in accordance with [RFC 2462]. Under no circumstances may such a value be assigned to an interface for use as a link address.

4. The Site Identifier

The site identifier is created by generating a key pair using a public key crypto algorithm (to be decided). Then the SHA-1 algorithm is used to calculate a hash over the public key. If 80 bit host identifiers are to be used, the site identifier consists of the first 44 bits of the SHA-1 hash. If 64 bit host identifiers are to be used, the site identifier consists of the last 48 bits of the SHA-1 hash. (The terms "site identifier" and "site key bits" are used interchangeably.)

All hosts that hold a host identifier must have a set of public key cryptographic keys. The host's public key is signed using the site secret key.

Site identifiers, along with the full self-signed public key and other pertinent information, are registered publicly to avoid and resolve site identifier collisions. When a newly generated site identifier collides with an existing one, the new key pair is discarded and a new one is generated. This is the only required use of a public registry. All other use of such a registry is optional.

Since it is computationally expensive to generate working keys that match a specific site identifier, possession of the secret key provides a "proof of ownership" of a site identifier that is good enough to fend off denial of service attacks and to provide authentication with a strength level somewhere between a simple encrypted password and full-out IPsec. An important feature is that the site identifier registry doesn't require rooted authority: any mechanism that makes a full list of site identifiers and public keys along with serial numbers available to anyone who wants to do a lookup within a reasonable timeframe after new identifiers have been generated is sufficient. A small number of repositories that

accept new site identifiers and accompanying material after checking the signature would work well. Each repository could work independently but they could exchange new site identifiers for the sake of completeness. Repositories can then make their contents available through mirroring and direct querying mechanisms. A good way to allow direct queries to the site identifier database would be by publishing a copy of an up to date repository in the DNS.

A fully populated 44 or 48 bit range of values is too large to store in the DNS without additional hierarchical structure. However, these ranges will never be fully populated, both because such a large number of site identifiers isn't necessary and because at some point, the chance of successive collisions becomes too large to be able to generate a new site identifier efficiently. A target for optimum performance would be a population somewhere between one in a million (approximately 17 million and 260 million

site identifiers respectively) and one in a thousand (17 / 260 billion site identifiers). Current practice shows that the DNS can handle flat spaces with up to several tens of millions entries, so a modest growth rate (well below Moore's Law) maxing out at around one to ten billion sites in 2050 shouldn't be a problem.

5. The Challenge/Response Mechanism

When a host wants to authenticate a correspondent using a crypto-based host identifiers, it issues a challenge to the correspondent. The layout of the challenge and the way it is transmitted to the correspondent is to be decided later.

When checking a response, a host may optionally take advantage of information published in the DNS or through other means. This allows the host to detect whether it's dealing with the "real" holder of a site locator rather than an impostor that stumbled on a key pair that maps to an existing site identifier. It also allows for retiring a compromised host key: if the published site serial number is higher than that presented by the correspondent, the host key is invalid.

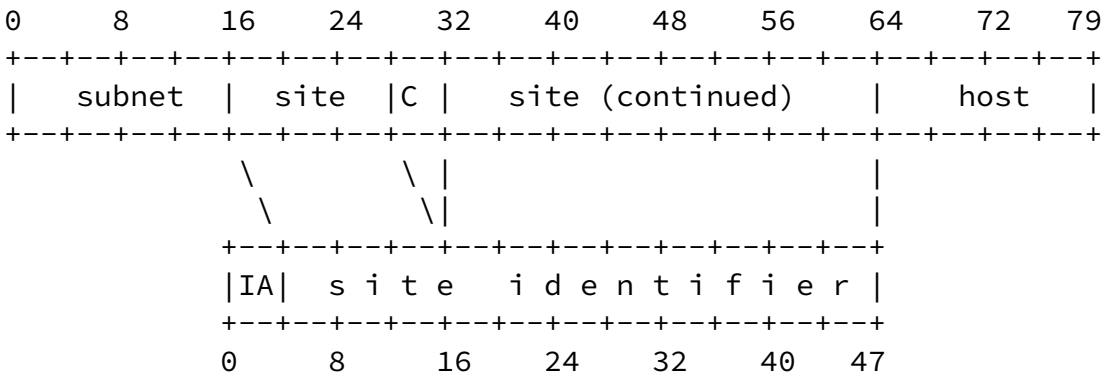
6. Turning the Site Identifier into an Address Range

In certain types of multihoming solutions, such as [ODELL96], the locator and identifier functions of the IP address are separated.

In these cases, the upper layer protocols such as TCP and UDP only see the identifier. In [ODELL96] the identifier consists of the lower 64 bits of the IPv6 address, which is compatible with what is proposed here. However, intra-site connectivity using just the lower 64 bits of the IPv6 address is problematic. To avoid this problem, and in order to provide a range of stable addresses a site may use regardless of its connectivity to the Internet, the site identifier may be transformed into a site prefix.

The procedure for transforming a 80 bit host identifier into a site prefix is to take the site identifier bits and concatenate those to a 4 bit prefix assigned by IANA. The resulting 48 bit value is the provider independent site prefix. This prefix is combined with a 80 bit host identifier to form a complete IPv6 address.

Example of how a 80 bit host identifier is turned into a 48 bit site prefix:

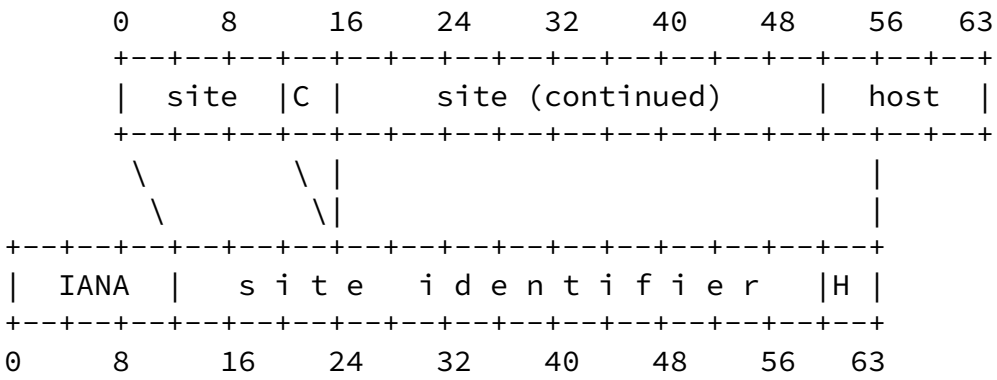


(IA = 4 bit prefix assigned by IANA)

A 64 bit host identifier is turned into a site prefix by concatenating the site bits with a 12 bit prefix assigned by IANA. This results in a 60 bit provider independent prefix. To avoid being limited to a single subnet, the top 4 bits of the host number are copied to bits 60 - 63 in the IPv6 address. The full 64 bit host identifier is present in the lower 64 bits to arrive at a full

IPv6 address. This allows for 16 subnets with 256 possible hosts each.

Example of how a 64 bit host identifier is turned into a 60 bit site prefix / 64 bit subnet prefix:



(IANA = 12 bit prefix assigned by IANA)
(H = top 4 bits of the host number)

Use of an EUI-64 that isn't a host identifier as outlined in this document in combination with one of the above provider independent prefixes is undefined and not recommended.

7. Operational Overview

If the host identifiers described here are used with ODT, then the ODT challenge/response interactions are changed as follows. A and B are hosts communicating using ODT, holding addresses A1, A2 and B1, B2 respectively. After A sees an address change from B1 to B2 in

incoming packets, it issues a challenge to B. Note that unlike with unmodified ODT there is no need to perform ODT interactions before a change of address happens, although a highly security conscious implementation may want to do so anyway.

When A wants to challenge B, it needs to encrypt a nonce using B's public key. If A doesn't have B's public key yet, it requests B's public key which must be signed by the site key, along with a copy of the site public key. A then checks whether the site identifier bits are indeed the same as the truncated hash derived from B's site public key, and if so, uses the site public key to check the signature over B's public key. If this procedure succeeds, A has

B's public key so it can encrypt a nonce and send it to B. B decrypts the nonce and returns it to A. When A receives back the nonce, it knows that B holds the matching private key so B's identity is verified.

8. IANA Considerations

IANA is requested to allocate a /4 and a /12 for crypto-based site identifier derived provider independent address ranges.

9. Security Considerations

Since the length of the hash over the public key is only 44 or 48 bits, even though finding a key for a known hash is extremely difficult, there is a significant chance of accidental collisions. As such, this authentication scheme on its own isn't secure enough for use with very sensitive applications.

10. Author's Address

Iljitsch van Beijnum
Karel Roosstraat 95
2571 BG The Hague
Netherlands

Phone: +31-70-3103790

Email: iljitsch@muada.com

11. References

[RFC 2462] S. Thomson and T. Narten, "IPv6 Stateless Address Autoconfiguration", December 1998

[RFC 3041] T. Narten and R. Draves, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", Januari 2001

[ODT] I. van Beijnum, "On Demand Tunneling For Multihoming", work in progress, January 2004

[MIPv6] "Mobility Support in IPv6",

draft-ietf-mobileip-ipv6-24.txt, work in progress

[NOID] E. Nordmark and T. Li, "Multihoming without IP Identifiers", draft-nordmark-multi6-noid-00.txt, work in progress, October 2003

[M6SEC] Nordmark, E., and T. Li, "Threats relating to IPv6 multihoming solutions", draft-nordmark-multi6-threats-00.txt, work in progress, October 2003.

[ODELL96] O'Dell M., "8+8 - An Alternate Addressing Architecture for IPv6", draft-odell-8+8-00.txt, work in progress, October 1996