James Kempf
Craig Gentry
Alice
Silverberg

Internet Draft
Document: draft-kempf-secure-nd-01.txt

Expires: December 2002

June 2002

Securing IPv6 Neighbor Discovery Using Address Based Keys (ABKs)

Status of this Memo

Abstract

When an IPv6 node receives a Router Advertisement, how does it know that the node which sent the advertisement is authorized to announce that it routes the prefix? When an IPv6 node receives a Neighbor Advertisement message, how does it know that the node sending the message is, in fact, authorized to claim the binding? The answer is, in the absence of a preconfigured IPsec security association among the nodes on the link and the routers, they don't. In this draft, a lightweight protocol is described for securing the signaling involved in IPv6 Neighbor Discovery. The protocol allows a node receiving a Router Advertisement or a Neighbor Advertisement to have the confidence that the message was authorized by the legitimate owner of the address or prefix being advertised without requiring a preconfigured IPsec security association. A certain degree of infrastructural support is required, but not any more than is currently common for public access IP networks. The protocol is based on some results in identity based cryptosystems that allow a publicly known identifier to function as a public key.

Contents

1.0 Introduction

The IPv6 Neighbor Discovery protocol described in RFC 2461 [1] plays a critical role in last hop network access for IPv6 nodes. The protocol allows a IP node joining a link to discover a default router, and for nodes on the link, including the routers, to discover the link layer address of an IP node on the link to which IP traffic must be delivered. Disruption of this protocol can have a serious impact on the ability of nodes to send and receive IP traffic.

Yet, security on the protocol is weak. As stated in the Security Considerations section of RFC 2461:

> The protocol contains no mechanism to determine which neighbors are authorized to send a particular type of message...; any neighbor, presumably even in the presence of authentication, can send Router Advertisement messages thereby being able to cause denial of service. Furthermore, any neighbor can send proxy Neighbor Advertisements as well as unsolicited Neighbor Advertisements as a potential denial of service attack.

In [2], a list of threats to IPv6 Neighbor Discovery on multi-access links is outlined. The threats don't occur on point to point links because the default router and IP address for a host are determined by PPP negotiation and so Neighbor Discovery is not required. These threats can occur for both wired and wireless public multi-access links. They are a particular problem for wireless links, however, because even private multi-access links over shared access (as opposed to switched) media with link level authentication mechanisms such as 802.1x [22] are subject to disruption if an authenticated node decides to play the trickster.

There are two underlying causes of these threats: a router advertising a prefix that it is not authorized to route or a node claiming an IPv6 address that it is not authorized to claim. These threats occur because the messaging involved in Neighbor Discovery by default contains no authentication information allowing the receiver to authenticate the sender. RFC 2461 recommends using IPsec AH authentication [4] if a security association exists, but this is a fairly heavyweight solution and is unlikely to be widely applicable to public access networks. In particular, a roaming node in a foreign public access network is unlikely to have a security association with a local access router or with other nodes on the same link. Indeed, most of the nodes on the same link may not even have the same home ISP as the roaming node. In addition, using IKE [28] or any other IPv6 protocol to establish a dynamic security association won't work if the protocol requires unsecured Neighbor Discovery. Manual keying can be used, but is impractical for public access networks.

In this document, a lightweight protocol that secures IPv6 Neighbor Discovery is described. The protocol allows IP nodes to verify that a node advertising routing for a local subnet prefix is authorized to advertise the prefix, and that information provided in a Neighbor Discovery message is authorized by the sending node. A certain amount of infrastructure is required, but no more than is currently needed for public access IP networks. In particular, no extension of the current NAS-based AAA infrastructure [24] nor a global PKI are necessary. The protocol depends on some results in identity based cryptosystems whereby a publicly known identifier, in this case, parts of a node's IP address, can serve as a public key. The technique whereby addresses are used to generate public/private key pairs is called Address Based Keys (ABKs).

2.0 Terminology

Address Based Keys (ABKs) - A technique whereby an identity based cryptosystem is used to generate a node's public and private key from its IPv6 subnet prefix or interface identifier.

Identity based cryptosystem - A cryptographic technique that allows a publicly known identifier, such as the IPv6 address,

to be used as a public key for authentication, key agreement, and encryption.

Identity based Private Key Generator (IPKG) - An agent that is capable of executing an identity based cryptographic algorithm to generate a private key when presented with the public identifier that will act as the public key. The IPKG is the root of trust in identity based crytosystems.

Public cryptographic parameters - A collection of publicly known parameters which the IPKG uses to generate the node's private key and which two nodes involved in securing or encrypting a message use to perform cryptographic operations. The public cryptographic parameters are formed from chosen constants and a secret key known only to the IPKG, specific to the identity based cryptographic algorithm.

Network Access Server (NAS) - A server that performs Authentication, Authorization, and Accounting (AAA) for nodes in a public access network.

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 [23].

3.0 What are Identity Based Cryptosystems?

Identity based cryptosystems are a collection of cryptographic techniques that allow a publicly known identifier, such as the email address or (particularly important in this application) the IP address of a node, to function as the public key part of a public/private key pair for purposes of digital signature calculation, key agreement, and encryption. Section 9.0 provides a quick overview of the available algorithms, with an extensive reference list. While identity based cryptosystems have been investigated for almost 20 years in the cryptographic community, they have not been widely discussed in the network security community. The reason is unclear, but it might have to do with the popularity and algorithmic simplicity of the reigning standard Diffie-Hellman technique, or possibly to the fact that, until recently, there have been no known identity based cryptographic algorithms that can be used to perform encryption. The existing algorithms have been restricted to digital signature calculation, and therefore have been fairly limited in scope. Hopefully, should identity based cryptosystems prove useful to the network security community, increased communication between the cryptography and network security communities will lead to a refinement of the algorithms and applications of identity based algorithms for application to securing IPv6 signaling.

Elliptic curve (EC) algorithms are particularly attractive for identity based keys because they work well with small key sizes, are computationally efficient on small nodes, such as small wireless

devices, and may generate smaller signatures. In addition, while non-EC algorithms have been proposed for identity based digital signature calculation, at the time of this writing, the most efficient way of performing identity based encryption is an EC algorithm.

Identity based cryptosystems work in the following way. A publicly known identifier is submitted to an IPKG. In this application, the publicly known identifier is either the 64 bit subnet prefix or the unique 64 bit interface identifier of an IPv6 address. The IPKG uses a particular algorithm to generate the private key and returns it. The public and private key can now be used for authentication and encryption as is typical in cryptosystems.

4.0 Digital Signature Calculations

Digital signatures MUST be calculated using the following algorithm:

```
sig = SIGN(hash(contents),IPrK,Params)
```

where:

```
sig      - The digital signature.
SIGN     - The identity based digital signature algorithm used
           to calculate the signature.
hash     - The HMAC-SHA1 one-way hash algorithm.
IPrK     - The Identity based Private Key.
Params   - The public cryptographic parameters.
contents - The message contents to be signed.
```

The digital signature MUST be verified using the following algorithm:

```
IPuK  = IBC-HASH(ID)
valid = VERIFY(hash(contents),sig,IPuK)
```

where:

```
IBC-HASH  - A hash function specific to the identity based
            algorithm that generates the public key from the
            public identifier.
ID        - The publicly known identifier used to generate the
            key.
IPuK      - The Identity based Public Key.
sig       - The digital signature.
VERIFY    - The identity based public key algorithm used to
            verify the signature.
Params    - The public cryptographic parameters.
valid     - 1 if the signature is verified, 0 if not.
```

5.0 Host and Router Configuration

Hosts and last hop routers participating in Neighbor Discovery require configuration with the identity based private key and with cryptographic parameters before they can secure messaging.

5.1 Router Configuration

When the ISP or network owner sets up its last hop routers, the routers are configured with the 64 bit subnet prefix or prefixes that they should advertise. In addition, the ISP uses its IPKG to generate a private key per prefix. The router uses this key in generating digital signatures on Router Advertisements. The private key and the public cryptographic parameters MUST be installed on the router through a secure channel. Examples of possible secure channels include configuration by a network administrator, installation via an NAS-based AAA network capable of secure key distribution, installation via a secure message exchange to a server with which the router has an IPsec security association, etc.

5.2 Host Configuration

Hosts require an identity based private key associated with their 64 bit interface identifier [3] in the IPv6 address, and the public cryptographic parameters. There are two possible ways in which the host can be configured:

- Dynamically, when the host is initially authenticated and authorized for network access through a secure connection with the local network's NAS,

- Statically, when its home ISP initially assigns the interface identifier.

If the dynamic configuration method is used, the local network must keep track of interface identifiers to avoid duplicates. If the static configuration method is used, the cryptographic parameters for the local network's router must be installed on a roaming host, since the router's parameters may not be the same as those for the roaming host. Dynamic and static configuration are discussed in the next two paragraphs.

Most public access networks currently require a host to undergo a secure authentication and authorization exchange through a NAS prior to being able to use the network. Since this exchange is typically performed at Layer 2 before any IP signaling, it can be done prior to any Neighbor Discovery signaling. The host includes its interface identifier in a message to the NAS. The NAS sends the interface identifier to the IPKG, where the private key is generated. The private key and public cryptographic parameters are then securely transferred back to the host where they are installed. The host uses

this private key for securing IPv6 Neighbor Discovery traffic on the foreign network, not for securing any private data, because the key belongs to the foreign network. After router discovery, the host uses the interface id and subnet prefix from the router to construct

the router's IP address using IPv6 Stateless Address Autoconfiguration. The hosts on the local link and the last hop router then use the public cryptographic parameters and the private keys given to them by the network to secure IPv6 Neighbor Discovery signaling.

Some public access networks may not perform secure Layer 2 authentication and authorization prior to allowing the host to perform Neighbor Discovery. In order to accommodate these kinds of networks, hosts MUST be configured with public cryptographic parameters and a private key by their home ISPs or network operators. The messaging for securing Neighbor Discovery includes an identifier based on the realm portion of the NAI [25]. The realm identifies the host's home ISP. This identifier allows the hosts and routers on the local link to authenticate the signaling of guest hosts. However, some method is needed to co-ordinate distribution of public cryptographic parameters between ISPs.

ISPs commonly use roaming consortia to provide remote access in areas where they do not have POPs. A group of ISPs organize into a roaming consortium to facilitate billing settlement and authentication. Roaming consortia can be used to support ABKs as well. A group of ISPs in a roaming consortium co-ordinate IPKGs so that the various ISPs in the consortium can accommodate guest hosts. The IPKGs use the same public cryptographic parameters, or are organized into an IPKG hierarchy [29]. Any private information (like a secret key) would need to be distributed between ISPs by secure means, such as a secure AAA connection or by hand.

6.0 Securing Router Advertisement

In this section, a protocol for securing the IPv6 Router Advertisement messages is discussed.

6.1 Router Advertisement Signature

A Router Advertisement sent by a router configured with a 64 bit prefix key contains a digital signature. The signature MUST sign the entire message.

In the signing algorithm described in Section 4.0, the input into the HMAC-SHA1 algorithm is the following:

```
contents = (chl,fl,rol,rel,rtt,sllao,mtuo,pro,dso,...)
```

IPrK in the signing algorithm is the private key having the router's 64 bit subnet prefix as its public key.

The digital signature MUST be included in an Identity Digital

Signature option (see Section 8.1) with the signature, algorithm, and realm identifier. An ICMP option is used instead of IPsec AH [4] because Neighbor Discovery options that are not recognized by a host

are ignored, so a host that can't verify the signature but is interested in risking using an unsecured Router Advertisement can simply ignore the option as a consequence of normal Neighbor Discovery processing, as opposed to having the Router Advertisement rejected by IPsec processing.

The Router Advertisement MUST contain a single Prefix option with the prefix for which the key was assigned. If the router also announces other prefixes, it MUST advertise them using separate Router Advertisements. If the router supports multiple identity based algorithms, it MAY include multiple Identity Digital Signature options with signatures calculated by the various algorithms, up to the path MTU.

6.2 Verifying a Router Advertisement

An IPv6 host receiving a Router Advertisement with an Identity Digital Signature Option verifies that the advertising node is authorized to send the advertisement in the following way. If the Router Advertisement does not contain a routing prefix option, or if it contains more than one routing prefix option, the host SHOULD discard the Router Advertisement, unless the host wants to risk using an unsecured Router Advertisement. If the host does not support one of the algorithms used for signing the message, it SHOULD discard the Router Advertisement, unless the host wants to risk using an unsecured Router Advertisement.

The host locates the single routing prefix option and extracts the subnet prefix which the sending node claims it is allowed to route. The host then uses the verification algorithm in Section 4.0 to verify the digital signature using the same value for contents as in Section 6.1. In this calculation, ID is the subnet prefix in the Prefix option. The identity based algorithm and router public cryptographic parameters depend on the algorithm and realm identifier in the Identity Digital Signature option.

6.3 Negotiating an Identity based Algorithm

A lengthy negotiation process for determining which identity based algorithm to use is obviously not in the interest of supporting a lightweight protocol. However, algorithms do change over time, and therefore it is necessary to have some way whereby a host can indicate in a Router Solicitation which algorithms it supports. If the router cannot provide an authenticator for any of the algorithms, it can simply return an unauthenticated Router Advertisement and the host can take its chances. For this purpose, the host uses an Identity Algorithm option (see Section 8.2).

For multicast Router Advertisements, the router can include Identity

Digital Signature options for each algorithm it supports, up to the path MTU. Alternatively, the host can be required to solicit the Router Advertisement and tell the router what algorithms it supports in an Identity Algorithm option.

7.0 Securing Neighbor Discovery

A similar procedure is used for securing IPv6 Neighbor Discovery messages.

7.1 Neighbor Advertisement Signature

A Neighbor Advertisement sent message contains a digital signature calculated with the private key generated from the 64 bit interface identifier and the host public cryptographic parameters. The signature MUST be calculated over the entire message.

The Target Link Layer Address option MUST be included.

In the signing algorithm described in Section 4.0, the input into the hash algorithm is the following:

```
contents = (flg,addr,l2addr)
```

IPrK is the interface identifier private key.

The digital signature MUST be included in an Identity Digital Signature option (see Section 8.1) with the signature, algorithm, and realm identifier. Again, an ICMP option is used instead of IPsec AH because Neighbor Discovery options that are not recognized by a node are ignored.

7.2 Verifying a Neighbor Advertisement

An IPv6 node receiving a Neighbor Advertisement with an Identity Digital Signature option verifies that the advertising node is authorized to send the advertisement in the following way. If the receiving node does not support one of the algorithms used for encrypting the signature, it SHOULD discard the Neighbor Advertisement, unless the node wants to risk using an unsecured Neighbor Advertisement.

The node uses the verification algorithm in Section 4.0 to verify the digital signature using the same value for contents as in Section 7.1. In this calculation, ID is the sending node's 64 bit interface identifier. The identity based algorithm and node public cryptographic parameters depend on the algorithm and realm identifier in the Identity Digital Signature option.

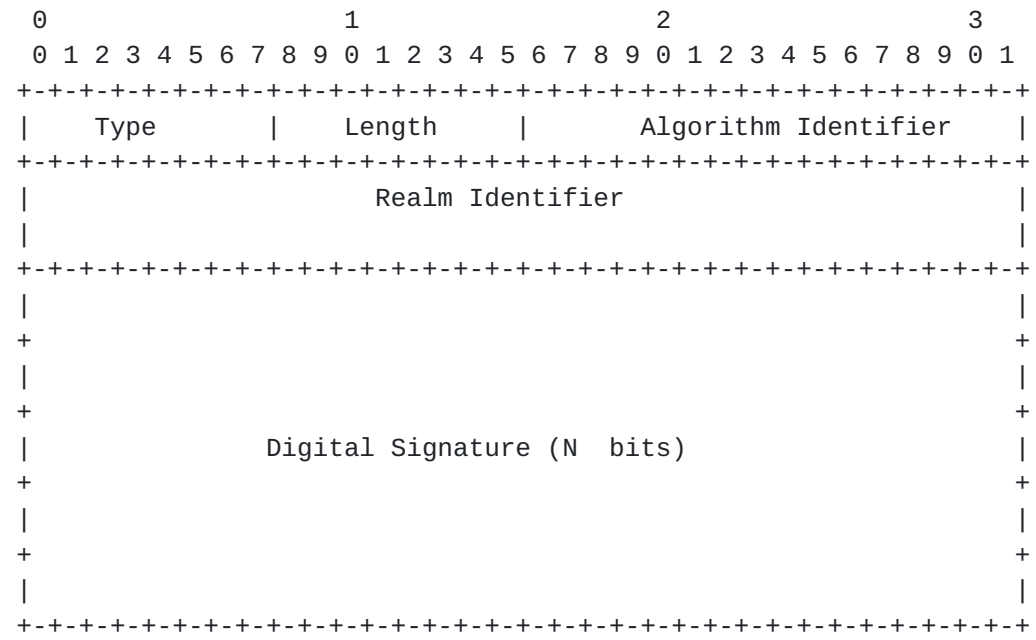
7.3 Negotiating an Identity Based Algorithm

A node sending a Neighbor Solicitation message can indicate what algorithms it is capable of accepting by including an Identity Algorithm option in the message.

8.0 Option Formats

8.1 Identity Digital Signature Option

The Identity Digital Signature Option contains a digital signature calculated using address based private key. It is always the last option in the list. The format of this option, after [1], is:



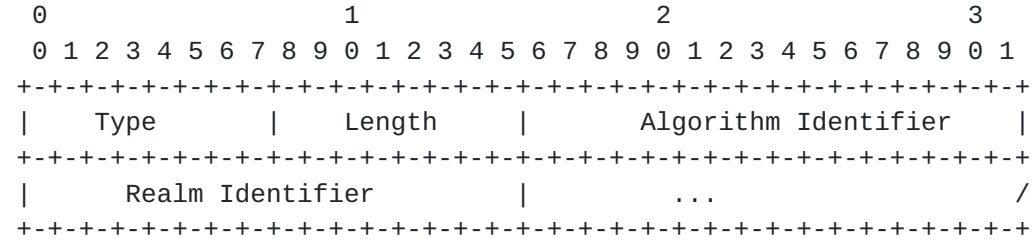
where:

Type	8 bit identifier for the option type, assigned by IANA.
Length	8 bit unsigned integer giving the option length (including type and length fields) in units of 8 octets.
Algorithm Identifier	16 bit nonzero algorithm identifier,assigned by IANA, indicating the identity based algorithm used to sign the message.
Realm Identifier	Either the 64 bit nonzero HMAC-SHA1 hash of the realm part of the NAI [25], or zero to indicate that the current network's IPKG and public cryptographic parameters should be used.
Digital Signature	An N bit field containing the digital signature. The field is zero aligned to the nearest 8 byte boundary. The exact number of bits is depends on the

identity based algorithm and use.

8.2 Identity Algorithm Option

The Identity Algorithm Option allows a node to indicate which identity based keying algorithms it supports for particular realms when requesting a Router Advertisement or Neighbor Advertisement. The Identity Algorithm Option has the following format:



where:

Type	8 bit identifier for the option type, assigned by IANA.
Length	8 bit unsigned integer giving the option length (including type and length fields) in units of 8 octets.
Algorithm Identifier	16 bit nonzero algorithm identifier,assigned by IANA, indicating the identity based algorithm used to sign the message.
Realm Identifier	Either the 64 bit nonzero HMAC-SHA1 hash of the realm part of the NAI [25], or zero to indicate the current network's algorithm.

and the option contains as many algorithm identifier-realm identifier pairs, in order of preference, as the node supports. The option is zero padded to multiples of 8 bytes. The

9.0 Identity Based Key Algorithms

Shamir [19] introduced the idea of identity based cryptography in 1984. Practical, provably secure identity based signature schemes [12], [11], [13] and Key Agreement Protocols [16] soon followed. Practical, provably secure identity based encryption schemes [8], [10] have only very recently been found.

In identity based signature protocols, the node signs a message using its private key supplied by its IPKG and the public cryptographic parameters. The signature is then verified using the

node's identity together with the public cryptographic parameters. In identity based key agreement protocols, two parties share a secret. Each party constructs the secret by using its own private

key and the other party's public identity. In identity based encryption, the encryptor uses the recipient's public identity to encrypt a message, and the recipient uses its private key to decrypt the ciphertext.

As is generally the case with public-key cryptography, the security of the systems is based on the difficulty of solving a hard number theory problem, such as factoring or a discrete log (or Diffie-Hellman) problem.

Elliptic curves and associated pairings have solved the problem of how to do identity based encryption [8], and are used to construct identity based signature [18][14][9] and key agreement [18][21] protocols.

There are a number of advantages to using identity based systems that are based on elliptic curves and their pairings. One is that there are compatible elliptic curve-based signature, key agreement, and encryption schemes. This means firstly that the same public key/private key pair and public cryptographic parameters can be used to do signatures, key agreement, and encryption. Secondly, these protocols overlap, so that results of computations and pre-computations done for one system can be used in the others. Further, there are usually efficiency advantages in using elliptic curves, over using other public-key methods. Generally, one obtains shorter signatures, shorter ciphertexts, and shorter key lengths for the same security as other systems. Efficiency can be further enhanced by using abelian varieties in place of elliptic curves [20].

There are identity based signature schemes [9] using elliptic curves and pairings that base their security on the difficulty of solving the elliptic curve Diffie-Hellman problem. This is the same classical hard problem on which standard Elliptic Curve Cryptography (ECC) [17][15] is based. Identity based encryption and key agreement schemes using elliptic curves (or abelian varieties) and pairings rely on the difficulty of solving the bilinear Diffie-Hellman problem.

Identity based cryptosystems can be constructed with or without key escrow. Protocols with key escrow can be performed in fewer passes than corresponding systems that do not provide for key escrow.

Techniques from threshold cryptography allow the master key information to be distributed or shared among a number of IPKGs so that all of them would have to collude for a node's private key to be known to them. Such a scenario would allow for key escrow if necessary, by agreement among all the IPKGs, but guards against knowledge of the private keys by the IPKGs without their mutual agreement.

10.0 Previous Work

RFC 3401 [27] describes a protocol for generating randomized interface identifiers for the bottom 64 bits of the IPv6 address. RFC 3401 is not designed to address any of the security concerns raised in RFC 2461; however, it is just designed to provide a measure of privacy to users by frustrating attempts to correlate particular addresses with particular network activity. Randomized interface identifiers can be used if the host is re-keyed every time it changes its interface identifier. In practice, this may be somewhat impractical in public access networks, unless the ABK is being provided by the local network and not the home ISP.

Cryptographically Generated Addresses (CGAs) [6], also called SUCV identifiers [7], are another way to construct a cryptographic binding for addresses. In CGAs, the interface identifier is generated from the public key, rather than the other way around as in ABKS. The primary difference between CGAs and ABKs are the following:

- CGAs use the hash of the public key as the interface id in the address suffix, whereas ABKs hash the interface id or subnet prefix to form the public key.
- CGAs allow the node to generate the public key/private key pair on its own, whereas ABKs require that the node be provided with a private key by the entity that assigns its address.
- ABKs require configuration with the public cryptographic parameters because the IPKG uses a master secret to perform the private key generation, and the master secret might expire or be compromised.

The consequences of the first point are that CGAs are not cryptographically active and therefore a separate mechanism is required to distribute the public key. This may be as simple as including it as a separate field in the message. In addition, CGAs are not "topologically active" and therefore cannot be used to sign the subnet prefix in routing.

The consequences of the second point are that there is less computational load on the node for ABKs, since it only has to perform signature verification, not public key/private key pair generation. However, CGAs can be used in the absence of any infrastructure whereas ABKs require the node to be assigned an address-based private key.

The consequences of the third point are nodes must be preconfigured with the private key and public cryptographic parameters for the operation. In principle, this is no different than key distribution in Diffie-Hellman. In this case, either

dynamic or static configuration of the private key and public cryptographic parameters is performed, but in a way that doesn't require Neighbor Discovery.

11.0 Infrastructure Requirements

As mentioned previously, ABKs require a certain amount of infrastructure to generate the private keys from the subnet prefix and interface ids. This requirement, in and of itself, is a hindrance for ad hoc networking designs that call for nodes to simply autoconfigure their addresses without requiring an ISP or network operator to be involved. For networks that are run by ISPs or enterprises, this requirement is not likely to be a problem, however.

ABKs place certain constraints on address provisioning. In particular, an address used for ABK cannot be assigned using DHCP [30]. To the extent DHCP requires Neighbor Discovery, there is a bootstrapping problem in using a DHCP address for ABK. An address used for ABK can be constructed using IPv6 Stateless Address Autoconfiguration [26] as long as the node performing the Stateless Address Autoconfiguration has an ABK interface id and private key for the suffix 64 bits of the address and no duplicate is detected. Indeed, the same mechanism described here to secure Neighbor Discovery could also be used to secure Stateless Address Autoconfiguration.

With some identity based algorithms, the IPKG maintains a copy of the private key, the so-called "key escrow" property. Consequently, the address assignor's IPKG knows the private keys for every address, and can potentially snoop authenticated or encrypted traffic. However, the ABK is only being used to secure IPv6 signaling traffic and not sensitive private data. Both the network operator and the legitimate client/user have an interest in seeing efficient operation of the network. Most users today are happy to trust their ISPs with their credit card number, trusting their ISP to guard their ABK is probably of equal or lesser extent.

If a group of ISPs in a roaming consortium choose to support ABKs, they have to co-ordinate in order to share a master key. There are techniques that allow secure generation of ABKs in such circumstances, but in principle ISPs in a roaming consortium must trust each other for billing and settlement, so business procedures and computational mechanisms for guarding privileged information are likely to be in place. A collection of ISPs that share a contract for IPKGs will allow their customers to securely use their networks, others will either get insecure or no service, just as is the case currently with roaming. The practical considerations involving co-ordinating the IPKGs between ISPs can be considerably reduced by using a hierarchical key generation system, such as is described in [29].

12.0 Security Considerations

The computation involved in verifying Neighbor Discovery messages could be utilized by an attacker to mount a "computational DoS

attack." The attacker bombards the victim with bogus Neighbor Discovery messages, which the victim is forced to verify. This ties the victim up in performing cryptography on the messages.

13.0 References

[1] Narten, T., Nordmark, E., and Simpson, W., "Neighbor Discovery for IP Version 6 (IPv6)", RFC 2461, December, 1998.

[2] Kempf, J., and Nordmark, E., "Threat Analysis for IPv6 Public Multi-Access Links," draft-kempf-netaccess-threats-00.txt, a work in progress.

[3] Hinden, R., and Deering,S., " IP Version 6 Addressing Architecture", RFC 2373, July 1998.

[4] Kent, S., and Atkinson, R., " IP Authentication Header," RFC 2402, November 1998.

[5] Droms, R. (ed), " Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", draft-ietf-dhc-dhcpv6-23.txt, a work in progress.

[6] O'Shea, G., and Roe, M., "Child-proof Authentication for MIPv6 (CAM)", ACM Computer Communications Review, April, 2001.

[7] Montenegro, G., and Castellucia, C., "SUCV Identifiers and Addresses," draft-montenegro-sucv-02.txt, a work in progress.

[8] D. Boneh and M. Franklin, "Identity based encryption from the Weil pairing", Advances in Cryptology --- Crypto 2001, Lecture Notes in Computer Science 2139, (2001), Springer, 213-229, http://www.cs.stanford.edu/~dabo/papers/ibe.pdf

[9] J. C. Cha and J. H. Cheon, "An Identity-Based Signature from Gap Diffie-Hellman Problem", Cryptology ePrint Archive: Report 2002/018, http://eprint.iacr.org/2002/018/

[10] C. Cocks, "An identity based encryption scheme based on quadratic residues", http://www.cesg.gov.uk/technology/id-pkc/media/ciren.

[11] U. Feige, A. Fiat, and A. Shamir, "Zero-knowledge Proofs of Identity", Journal of Cryptology 1, (1988), 77-94.

[12] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems", Advances in Cryptology --- Crypto '86, Lecture Notes in Computer Science 263, 1986), Springer, 186-194.

[13] L. C. Guillou and J.-J. Quisquater, "A practical zero-knowledge protocol fitted to security microprocessors minimizing both transmission and memory", Advances in Cryptology --- EUROCRYPT '88, Lecture Notes in Computer Science 330, (1988), Springer, 123-128.

[14] F. Hess, "Exponent Group Signature Schemes and Efficient Identity Based Signature Schemes Based on Pairings", Cryptology ePrint Archive: Report 2002/012, http://eprint.iacr.org/2002/012/

[15] N. Koblitz, "Elliptic curve cryptosystems", Mathematics of Computation 48 (1987), 203-209.

[16] U. Maurer and Y. Yacobi, "Non-interactive public-key

cryptography," Advances in Cryptology --- Eurocrypt '92, Lecture Notes in Computer Science 658,(1993), Springer, 458-460.

[17] V. S. Miller, "Uses of elliptic curves in cryptography", Advances in Cryptology --- Crypto'85, Lecture Notes in Computer Science 218, (1986), Springer, 417-426.
[18] R. Sakai, K. Ohgishi, and M. Kasahara, "Cryptosystems based on pairing", SCIC 2000-C20, Okinawa, Japan, January 2000
[19] A. Shamir, "Identity-Based Cryptosystems and Signature Schemes", Advances in Cryptology --- Crypto '84, Lecture Notes in Computer Science 196, (1984), Springer, 47-53.
[20] A. Silverberg and K. Rubin, "The best and worst of supersingular abelian varieties in cryptology", Cryptology e-Print Archive: Report 2002/006, http://eprint.iacr.org/2002/006/
[21] N. P. Smart, "An identity Based authenticated key agreement protocol based on the Weil pairing", Cryptology ePrint Archive: Report 2001/111, http://eprint.iacr.org/2001/111/
[22] "802.1x - Port Based Access Control", IEEE Standard for Local and Metropolitan Area Networks, 2001.
[23] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997.
[24] Mitton, D., and Beadles, M., "Network Access Server Requirements Next Generation (NASREQNG) NAS Model", RFC 2881, July 2000.
[25] Aboba, B., and Beadles, M., "The Network Access Identifier", RFC 2486, January, 1999.
[26] Thomas, S., and Narten, T., "IPv6 Address Stateless Autoconfiguration", RFC 2462, December, 1998.
[27] Narten, T., and Draves, R., "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 3041, January, 2001.
[28] Harkins, D., and Carrel, D., "The Internet Key Exchange (IKE)", RFC 2409, November, 1998.
[29] Gentry, C., and Silverberg. A., "Hierarchical ID-based Cryptography," http://eprint.iacr.org/2002/056/.
[30] Droms, R., et. al., "Dynamic Host Configuration Protocol for IPv6," draft-ietf-dhc-dhcpv6-26.txt, a work in progress.

14.0 Author's Contact Information

James Kempf
DoCoMo Labs USA
180 Metro Drive, Suite 300
San Jose, CA 95430
USA
Phone: +1 408 451 4711
Email: kempf@docomolabs-usa.com

Craig Gentry
DoCoMo Labs USA
180 Metro Drive, Suite 300
San Jose, CA 95430
USA
Phone: +1 408 451 4723
Email: cgentry@docomolabs-usa.com

Alice Silverberg
Department of Mathematics
Ohio State University
Columbus, OH 43210

Phone: +1 614 292 4975
Email: silver@math.ohio-state.edu

USA

15.0 Full Copyright Statement