Network Working Group
INTERNET-DRAFT
Category: Informational
<draft-arkko-send-cga-00.txt>

Jari Arkko
Pekka Nikander
Vesa-Matti Mantyla
Ericsson
June 2002

Securing IPv6 Neighbor Discovery Using Cryptographically Generated Addresses (CGAs)

Status of this Memo

Abstract

IPv6 nodes use the Neighbor Discovery (ND) protocol to discover other nodes on the link, to determine each other's link-layer addresses, to find routers and to maintain reachability information about the paths to active neighbors. The original ND specifications called for the use of IPsec for protecting the ND messages. However, in this particular application the use of IPsec may not always be feasible, mainly due to difficulties in key management. If not secured, ND protocol is vulnerable to various attacks. This document specifies a lightweight security solution for ND that does not rely on pre-

configuration or trusted third parties. The presented solution uses Cryptographically Generated Addresses.

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 [RFC 2119].

Table of contents

1. Introduction

IPv6 defines the Neighbor Discovery (ND) protocol in RFC 2461 [ND98]. Nodes on the same link use the ND protocol to discover each other's presence, to determine each other's link-layer addresses, to find

routers and to maintain reachability information about the paths to active neighbors. The ND protocol is used both by hosts and routers. Its functions include Router Discovery (RD), Address Auto-configuration, Address Resolution, Neighbor Unreachability Detection (NUD), Duplicate Address Detection (DAD), and Redirection.

Section 4 gives a description of different approaches for securing the ND protocol. This shows that the specified IPsec method isn't always applicable. In Sections 6 to 8 we present a new, lightweight solution for ND security. Our approach is based on the use of Cryptographically Generated Addresses (CGAs) [CAM01].

The CGA method provides a proof of address ownership, i.e., it provides assurance that the node we are talking to is indeed the one that came up with the very address. In our solution, there is no need for an external key management infrastructure. All the used keys can be self-generated, and can be presented without external credentials. Section 5 briefly introduces the CGA method.

Our requirements call for secure ND signaling to be possible both in private networks as well as in public access networks. In the public access networks we can't trust all parties to behave in an appropriate manner.

At the moment this specification considers only the case of networks that are fully protected with our secure ND approach. That is, we do not yet deal with the problem of securing some ND messages with our approach while allow some other messages to be secured with the traditional IPsec approach or even be left unsecured.

2. Definitions

Cryptographically Generated Addresses (CGAs) - A technique where the address of the node is cryptographically generated from the public key of the node and some other parameters using a one-way hash function [CAM01]. Also called SUCD Identities in [SUCV].

Internet Control Message Protocol version 6 (ICMPv6) - The IPv6 control signaling protocol. Neighbor Discovery is a part of ICMPv6.

Neighbor Discovery (ND) - The IPv6 Neighbor Discovery protocol[ND98].

3. Neighbor Discovery

The main functions of IPv6 Neighbor Discovery are as follows:

- Neighbor Unreachability Detection (NUD) is used for tracking the reachability of neighbors, both local destinations and routers [ND98, Section 7.3].

- Duplicate Address Detection (DAD) is used for preventing address collisions [ND98, AUTOCONF98]. A node that intends to assign a new address to one of its interfaces runs first the DAD procedure to verify that other nodes are not using the same address.

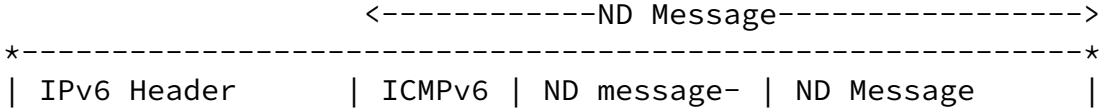
- Address Resolution is similar to IPv4 ARP [ARP82]. The Address Resolution function resolves a node's IPv6 address to the corresponding link-layer address for nodes on the link [ND98, Section 7.2]. Address Resolution is used for hosts and routers alike.

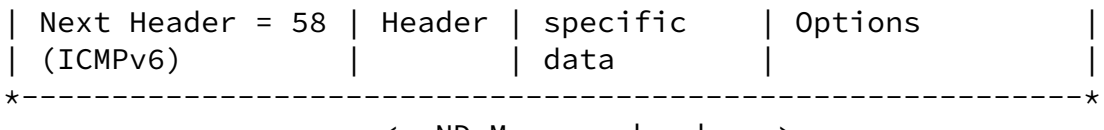
- Address Autoconfiguration is used for automatically assigning addresses to a host [AUTOCONF98]. This allows hosts to operate without configuration related to IP connectivity. The Address Autoconfiguration mechanism is stateless, where the hosts use prefix information delivered to them during Router Discovery to create addresses, and then test these addresses for uniqueness using the DAD procedure. A stateful mechanism, DHCPv6, provides additional Autoconfiguration features.

- The Redirection function is used for automatically redirecting hosts to an alternate router [ND98, Section 8]. It is similar to the ICMPv4 Redirect message [POS81].

- The Router Discovery function allows IPv6 hosts to discover the local routers on an attached link [ND98, Section 6].

The Neighbor Discovery messages follow the ICMPv6 message format and ICMPv6 types from 133 to 137. The IPv6 Next Header value for ICMPv6 is 58. The actual Neighbor Discovery message includes an ND message header consisting of ICMPv6 header and ND message-specific data, and zero or more ND options:





The ND message options are formatted in the Type-Length-Value format.

All IPv6 ND protocol functions are realized using the following messages:

ICMPv6 Type	Message
133	Router Solicitation (RS)
134	Router Advertisement (RA)
135	Neighbor Solicitation (NS)
136	Neighbor Advertisement (NA)
137	Redirect

The functions of the ND protocol are realized using these messages as follows:

- Router Discovery uses the RS and RA messages.
- Duplicate Address Detection uses the NS and NA messages.
- Address Autoconfiguration uses the NS, NA, RS, and RA messages.
- Address Resolution uses the NS and NA messages.
- Neighbor Unreachability Detection uses the NS and NA messages.
- Redirect uses the Redirect message.

All functions but Address Auto-configuration are explained in RFC 2461. The Address Auto-configuration is presented in RFC 2462.

In Section 8 we define two new ND message options that are used in our security solution.

4. Approaches for Securing Neighbor Discovery

When ND is not secured, attackers can cause, for instance, the following types of problems [Ark01, Ark02]:

- Making hosts adopt bogus prefixes. This leads to Denial-of-Service.
- Making hosts adopt bogus default routers. This leads to Denial-of-Service and can also be used in an attempt to place the attacker as a Man-in-the-Middle for all communications.

- Sending spoofed answers to DAD queries in an attempt to prevent a host from acquiring any address.
- Sending spoofed Address Resolution messages in an attempt to cause Denial-of-Service or to place the attacker as a Man-in-the-Middle between the neighbors.
- Sending spoofed NUD messages. This can be used to make the neighbor believe the node is reachable when it is not.
- Sending spoofed Redirect messages, again in an attempt to cause Denial-of-Service or to place the attacker as a Man-in-the-Middle.

The ND protocol ignores packets received from off-link nodes by verifying that the Hop Limit field contains the value 255. Since every forwarding router reduces this value by one, an ND packet containing the Hop Limit value of 255 must originate from a neighboring IPv6 node. However, this does not protect against malicious neighbors.

It is possible to authenticate the ND protocol packet exchanges using the IPSec Authentication Header, if a suitable SA exists. This is the approach specified in the original specification [ND98]. However, three different types of problems exist with this approach:

- Running an automatic keying protocol such as IKE would involve sending IP traffic, which may be impossible in the initial stages of some the ND procedures. For instance, we can't send IKE UDP packets before we have an address. Also, the node needs to discover the link layer address of the neighbor. This is a chicken-and-egg problem, since getting an address or finding the link layer address of the peer would require running ND, which in turn would require the security associations to be already in place.

- Manual SAs can be configured, but as [Ark01] points out this may lead to a large number of SAs. The definition of IPsec requires a different SA for every IP address that is used as a destination.

- According to RFC 2461 [ND98], the ND protocol "provides no mechanism to determine, which neighbors are authorized to send a particular type of message", e.g a Router Advertisement. The current set of IPsec policy selectors do not allow us to define which nodes are allowed to send which particular ND messages.

- IPsec in general can prove that the peers are among the intended users of the link. However, we also need to authorize the contents of the messages. For instance, even if a particular node is authorized to send Neighbor Advertisements, it is usually authorized to send them only on its own behalf. Manually configured SAs with security policy entries that limit the use of source address can in some cases handle this, but it isn't expected that trusted third parties and certificate infrastructures can keep up with the right IP address identities of all users at all times.

Due to the above, the use of IPsec in securing ND for public access networks is hard, and it isn't clear that attacks can be avoided even when IPsec is used.

Various new approaches to securing ND have been designed. One of these approaches is the Address Based Keys method which is discussed in [ABK02].

5. Cryptographically Generated Addresses

The purpose of the CGA method is to ensure that only the node that "owns" an address has the right to make statements about this address, e.g., for the purpose of address resolution.

In the CGA method a node first generates a private-public key pair. The public key and some other parameters are used to generate the CGA address. The private key is used for signing signaling messages related to this address. The public key has to be distributed to the receiving node(s) for the signature verification to be possible. It isn't necessary to protect the transfer of the public key.

The following text gives a more detailed description of the calculations the nodes have to perform when using the CGA.

5.1. Generation

An IPv6 address is composed of two parts:

```
Address = Routing Prefix | Interface ID
```

In the CGA scheme, the Route Prefix is derived in a usual manner, whereas the Interface ID of the node is created using the following procedure:

1. Select the security parameter Sec = 0, 1 , 2, 3.

2. Calculate

```
Hash = HASH("CGA" | Sec | Routing Prefix | PK | Random),
```

where

HASH A one-way hash algorithm.

PK The Public Key of the node.

"CGA" A three octet long string consisting of the ASCII characters 'C', 'G', and 'A'.

Sec One octet security parameter that can be used to tune the amount of work needed to create CGA addresses. The rationale for Sec is discussed more in depth in [Ark02].

Routing Prefix
The 64 routing prefix.

Random A 64 bit long random number.

3. Select 64+20 x Sec rightmost bits of the hash output and compare the 20 x Sec leftmost bits to zero. If not zero, proceed to generate a new Random value and go back to Step 2.

4. Concatenate the 64-bit Routing Prefix and the rightmost 64 bits of the Hash to obtain the Address.

5. Set the group and the universal bits [ADD98] to 1 and the two rightmost bits to Sec.

6. Perform Duplicate Address Detection. If collision is detected, proceed to generate a new Random value and return to Step 2. After three collisions, stop and report error.

Note that the Sec parameter is included in both in the address and in the hash input. The indication to use CGA-based addresses is encoded by setting the group and universal bits to 1.

5.2. Signatures

SIG = SIGALG(HASH(Contents),SK),

Where

SIGALG A public key signature algorithm.

Contents Some statement relating to the address in question.

SK Secret Key of the node.

For ND messages, the Contents is formed from the following parts:

1. The IPv6 header with the exception of the destination address field. (The purpose of omitting the destination address field is to avoid the CPU intensive signature generation when responding with the same message to different nodes.)
2. The ICMPv6/ND message with the exception of the Signature ND option. (The rest of the IPv6 message, e.g. the IPv6 Payload length or the ICMPv6 Length field are not modified as a result of omitting this part.)

5.3. Verification

In order to verify that a given address has been formed using CGA, the receiver performs the following steps:

1. Check that the group and the universal bits are 1. If not, the verification process fails.
2. Retrieve the Routing Prefix from the highest 64 bits of the address, Sec from the lowest 2 bits of the address, and PK and Random from an option accompanying the ND message. If the necessary option is not present in the message, the verification process fails.
3. Calculate the hash as defined in Section 5.1, set the group and universal bits, set the two lowest bits to Sec, and compare to the 64 lowest bits in the given address. If the values are not the same, the verification process fails.
4. The process succeeds.

In order to verify a given statement about a particular address, the following process is used:

1. Check that the address in the source field of the message has been formed using CGA, as explained above. If this fails, the verification process fails.
2. Construct a Contents string as described in Section 5.2.
3. Verify that the signature found in the Signature ND option has been produced using the private key corresponding to the public key found from the Public Key ND option. If this fails, the verification process fails.
4. The process succeeds.

5.4. Algorithms

In this specification, the one-way hash algorithm used in CGA generation and signature calculation is the SHA1 algorithm [SHA1]. As the public key algorithm we use the RSA algorithm [RSA78].

6. Securing Neighbor Discovery

The following text describes the procedures involved in securing ND with CGA. The method affects the transmitting and reception of Neighbor Advertisement (NA) messages.

All nodes MUST acquire a CGA-based address on all interfaces they communicate according to the procedure presented in Section 5.1.

All nodes follow the rules defined below when transmitting NA messages:

1. The node MUST use a CGA-based address as the source address in the IPv6 header that carries the ND message.
2. The node MUST attach the Public Key ND option to the NA message with the same parameters that were used in the construction of address in the source address field.
3. The node MUST attach the Signature ND option to the NA message, and calculates this signature as specified in Section 5.2.

All nodes follow the rules defined below when receiving NA messages:

1. The NA message MUST have a Public Key and Signature ND options. Messages that do not have these options MUST be silently discarded.
2. The receiver MUST verify the signature as described in Section 5.3. Messages that fail this verification MUST be silently discarded.

In the following we discuss how the above procedures affect the security of ND functions. We will also describe function-specific rules for the treatment of the NA messages.

6.1. Duplicate Address Detection

To the extent that CGA is secure, only the owner of the address can reply with a verifiable signature to a DAD query. This prevents other parties from sending replies in an attempt to prevent the host from getting an address.

After receiving an NS message, in this case the DAD query, and detecting an address collision, a node uses the above procedures for sending the NA message.

A node that receives the DAD reply, i.e. the NA message, uses the above procedures for receiving the NA message. If no valid replies are received, the tentative address is set to the VALID state. If the verification succeeded, the tentative address of the host is set to the DISABLED state.

6.2. Address Resolution

Here, the CGA method is used to assure that only the real owner of the address can produce a valid response.

The rules for sending and receiving the NA message have again been described earlier. Note that the link-layer address ND option is also protected with the signature, preventing a Man-in-the-Middle from replacing another link-layer address to a legitimate reply.

6.3. Neighbor Unreachability Detection

Here the CGA method makes sure that attackers cannot claim that a node is reachable when it is not.

For the procedure to process NA messages, see the beginning of Section 6. After a successful verification of the NA message, the node is marked as REACHABLE by the host.

7. Securing Router Discovery

For Router Discovery, we use CGA to ensure that a given message comes from the claimed IP address. However, this does not offer any information about the ability and willingness of the router to act as a router, or that the advertised network prefixes are correct. We use a heuristic process to verify these properties.

All routers MUST acquire a CGA-based address on all interfaces they communicate according to the procedure presented in Section 5.1. The router MUST use the same CGA-based address for both Neighbor Discovery and Router Discovery purposes.

7.1. Router Discovery

All routers follow the rules defined below when transmitting RA messages:

1. The router MUST use a CGA-based address as the source address in the IPv6 header that carries the RA message.

2. The router MUST attach the Public Key ND option to the RA message with the same parameters that were used in the construction of address in the source address field.
3. The router MUST attach the Signature ND option to the RA message, and calculates this signature as specified in Section 5.2.

All hosts follow the rules defined below when receiving RA messages:

1. The RA message MUST have a Public Key and Signature ND options. Messages that do not have these options MUST be silently discarded.
2. The receiver MUST verify the signature as described in Section 5.3. Messages that fail this verification MUST be silently discarded.

Still, even if the signature is verified correctly the host MUST check that the node claiming to be a router acts as a real router. We propose the following heuristic method:

- Each entry in the default router list of the host is marked either as an UNTESTED, TESTED, or FAILED router. All new entries start from the UNTESTED state.

- All communications from a host SHOULD use a router in the TESTED state, unless there are only UNTESTED ones available. FAILED routers SHOULD NOT be used for communications.

- When communicating to a non-local destination through the designated router, the host SHOULD keep track of the upper layer forward progress, in the same manner as is used in avoiding NUD [ND98, Section 7.3]. If such forward progress is being made, the router in question SHOULD be marked as TESTED.

- If no forward progress is being made, the host MAY attempt to send an ICMPv6 Echo request to verify that the router is working. Such requests MUST be addressed to a non-local destination known to the host, and MUST be rate-limited in the usual manner. A reply moves the router to the TESTED state.

- If no forward progress is being made, and no replies have been seen, the router SHOULD be marked as FAILED.

- Routers that have not been used by the host for a period of 120

seconds SHOULD be marked as UNTESTED.

- Routers in the FAILED state may be periodically tested with the ICMPv6 Echo request.

A similar process SHOULD be applied to test the prefixes advertised by the router.

Note that this heuristic process is inherently weak in the sense that a smart attacker could spoof response messages to unprotected communications or to the Echo requests. For this purpose it is strongly recommended that the hosts use only IPsec or TLS-protected communications as an indication of forward progress. This requires, however, that the hosts share a security association with another node in the Internet. Public web servers with TLS support and a certificate from a trusted root server are one possibility for arranging this security association in an easy manner.

7.2. Redirect

Routers use CGA to prove that the Redirect message comes from the right router.

All routers follow the rules defined below when transmitting Redirect messages:

1. The router MUST use a CGA-based address as the source address in the IPv6 header that carries the Redirect message.

2. The router MUST attach the Public Key ND option to the Redirect message with the same parameters that were used in the construction of address in the source address field.
3. The router MUST attach the Signature ND option to the Redirect message, and calculates this signature as specified in Section 5.2.

All hosts follow the rules defined below when receiving Redirect messages:

1. The Redirect message MUST have a Public Key and Signature ND options. Messages that do not have these options MUST be silently discarded.
2. The receiver MUST verify the signature as described in Section 5.3. Messages that fail this verification MUST be silently

discarded.

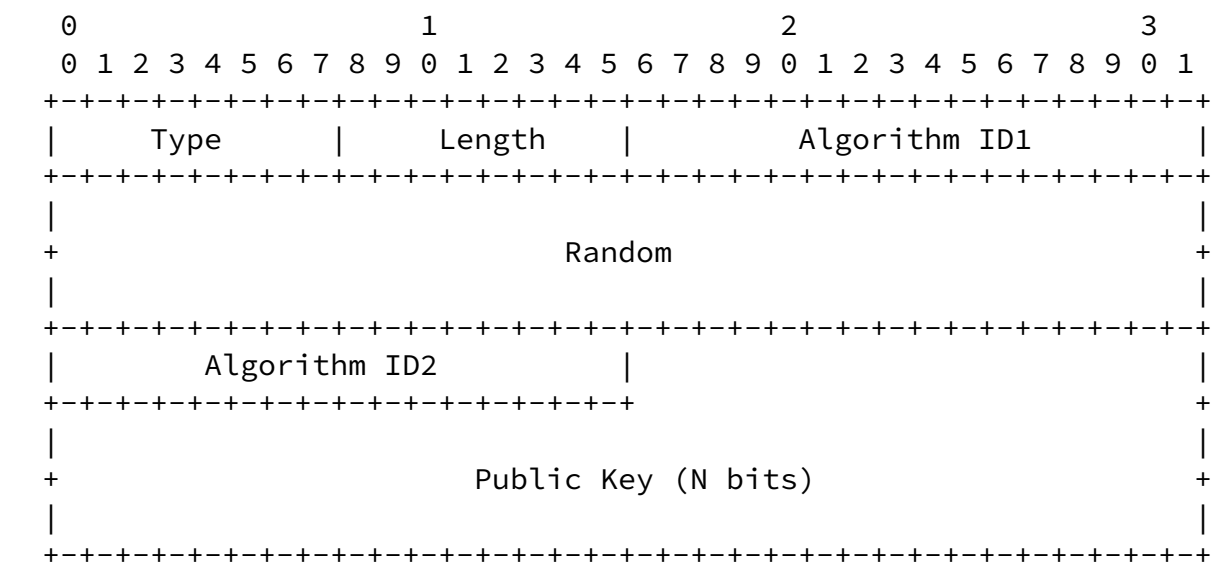
3. The receiver MUST verify that the Redirect message comes from an IP address to which the host may have earlier sent the packet that the Redirect message now partially returns. That is, the source address of the Redirect message must be the default router for traffic sent to the destination of the returned packet. If this is not the case, the message MUST be silently discarded.

Step 3 prevents a bogus router from sending a Redirect message when the host is not using the bogus router as a default router.

8. Option Formats

8.1. Public Key Option

The Public Key Option carries a public key of the node. This option follows the format presented in [ND98]:



where,

Type — An IANA assigned 8 bit identifier TBD for the option type.

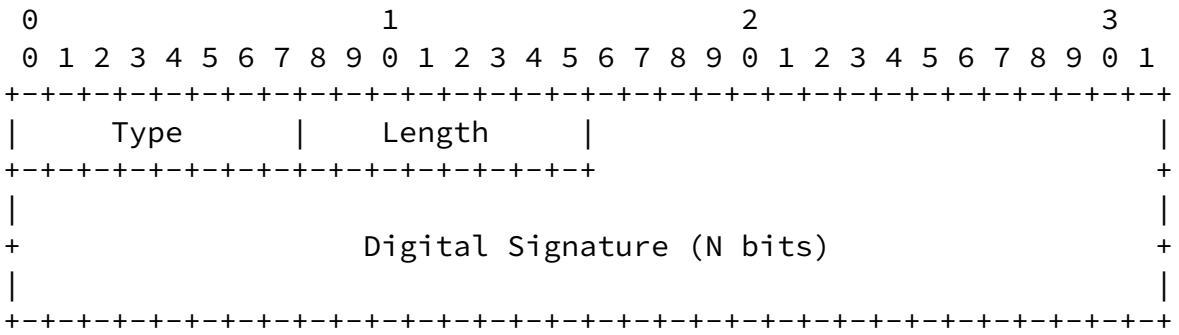
Length — An 8 bit unsigned integer indicating the option length (type + length fields) in units of 8 octets.

Algorithm ID1 — An IANA assigned 16 bit identifier for

	the signature algorithm. The currently defined values are: 1 RSA
Random	A 64 bit random number used in the creation of the address from the public key.
Algorithm ID2	An IANA assigned 16 bit identifier for the hash algorithm. The currently defined values are: 1 SHA1
Public Key	An N bit field carrying the public key, zero-padded to nearest 8 byte units. The public key is in the format implied by Algorithm ID1.

8.2. Signature Option

The Signature Option carries a digital signature calculated using the private key of the node. This option follows the format presented in [ND98]:



where,

Type	An IANA assigned 8 bit identifier TBD for the option type.
Length	An 8 bit unsigned integer indicating the option length (type + length fields) in units of 8 octets.
Digital Signature	An N bit field carrying the digital signature, zero-padded to nearest 8 byte units. The signature is calculated using the algorithm implied by Algorithm ID1 in the public key option, and is also in the format implied by this Algorithm ID1.

9. Security Considerations

The CGA method assures that the received messages are coming from the owner of the address. However, this method does not eliminate all security vulnerabilities related to the ND functions.

CGA prevents spoofed answers to DAD queries. An attacker may still be able to prevent valid responses or requests from reaching the intended recipient. As a result both participants are forced to believe that no address collision exists, when there in fact is.

Within Address Resolution and NUD functions CGA can be used to prevent spoofed responses. However, it is still possible to prevent the Address Resolution and NUD from completing for a given address. For the NUD, this means that a node is claimed to be unreachable, when it really is not.

Hosts can use CGA to show that the Redirect messages come from their current router. Still, we cannot say anything about the other router mentioned in the Redirect message. It is not clear if this is necessary, however. (If necessary, we could cross-certify routers without involving hosts.)

Within the Router Discovery functionality the CGA method ensures that we are communicating with the same router all the time, and prevents spoofing of the link-layer address of the router. But it does not help to verify that the router is connected to the Internet or that it is authorized to advertise a specific route prefix. A proper verification of these properties will not be possible without involving a trusted third party. However, we propose a heuristic method to test these properties in Section 7.1.

10. Acknowledgments

The authors would like to thank James Kempf, Gabriel Montenegro, Erik Nordmark, Tuomas Aura and Mike Roe for interesting discussions in this problem space.

11. References

11.1. Normative References

[ND98] Narten, T., Nordmark, E., and Simpson, W., "Neighbor discovery for IP Version 6 (IPv6)", RFC 2461, December, 1998.

[AUTOCONF98] Thomson, S., Narten, T., "IPv6 Stateless Address Autoconfiguration", RFC 2462, December 1998.

[ADD98] Hinden, R., Deering, S., "IP Version 6 Addressing Architecture", RFC 2372, July 1998.

[RSA78] Rivest, R., Shamir, A., Adleman, L. "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", Communications of the ACM, 21 (2), pp. 120-126, February 1978.

[SHA1] Eastlake, D., Jones, P., "US Secure Hash Algorithm 1 (SHA1)", RFC 3174, September 2001.

11.2. Non-Normative References

[ABK02] Kempf, J., Gentry, G., Silverberg, A., "Securing IPv6 Neighbor Discovery Using Address Based Keys (ABKs)", Internet Draft (work in progress), February 2002.

[Ark01] Arkko, J., Nikander, P., Kivinen, T., Rossi, M., "Manual SA Configuration for IPv6 link Local Messages", Internet Draft (work in progress), January 2001.

[Ark02] Arkko, J., Aura, T., Kempf, T., Mantyla, V.-M., Nikander, P., Roe, M. "Securing IPv6 Neighbor Discovery". Submitted for publication, 2002.

[ARP82] Plummer, D. C., "An Ethernet Address Resolution Protocol -- or -- Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware", RFC 826, November 1982.

[CAM01] O'Shea, G., Roe, M., "Child-proof Authentication for MIPv6 (CAM), Computer Communications Review", April 2001.

[POS81] Postel, J., "Internet Control Message Protocol", RFC 792, September 1981.

[SUCV] Montenegro, G., Castelluccia, C., "SUCV Identifiers and Addresses", Internet Draft (work in progress), November 2001.

12. IPR Considerations

The presented protocol uses public keys and hashes to prove address ownership. Ericsson's IPR may apply on such methods. The Ericsson policy on IPR issues can be checked from the Ericsson General IPR statement for IETF, http://www.ietf.org/ietf/IPR/ERICSSON-General.

13. Authors' Addresses

Jari Arkko
Oy LM Ericsson Ab
02420 Jorvas
Finland

EMail: jari.arkko@ericsson.com

Pekka Nikander
Oy LM Ericsson Ab
02420 Jorvas
Finland

EMail: pekka.nikander@nomadiclab.com

Vesa-Matti Mantyla
Oy LM Ericsson Ab
Tutkijantie 2C
90570 Oulu
Finland

EMail: vesa-matti.mantyla@ericsson.fi

Full Copyright Statement