

CGA and SeND Maintenance (CSI)
Internet-Draft
Intended status: Standards Track
Expires: January 30, 2010

W. Haddad
Ericsson
M. Naslund
Ericsson Research
July 29, 2009

On Secure Neighbor Discovery Proxying Using 'Symbiotic' Relationship
draft-haddad-csi-symbiotic-sendproxy-01

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on January 30, 2010.

Copyright Notice

Copyright (c) 2009 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents in effect on the date of publication of this document (<http://trustee.ietf.org/license-info>). Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Internet-Draft

Proxy SeND

July 2009

Abstract

This document introduces a simple mechanism which enables a host using a cryptographically generated IPv6 address to delegate the task of secure neighbor discovery to another node, i.e., proxying, by means of establishing a 'symbiotic' relationship with that node.

Table of Contents

1.	Introduction	3
2.	Conventions used in this document	4
3.	Motivation	5
4.	What is a 'Symbiotic' Relationship?	6
5.	Applying SR in a SeND environment	8
5.1.	Using SR for SeND Proxying	9
6.	New Option	12
7.	Security Considerations	13
8.	Normative References	14
	Authors' Addresses	15

1. Introduction

Secure neighbor discovery protocol [[RFC3971](#)] enables establishing a trust relationship between hosts attached to the same link and/or between a host and its access router(s) (ARs). SeND achieves its goal by using a cryptographically generated IPv6 address ([RFC3972](#)) on the host side and deploying a local public key infrastructure in the access network.

When SeND protocol is applied, all router advertisement (RtAdv) as well as any neighbor discovery protocol (described in [[RFC4861](#)]) messages sent by the AR are signed. In addition, any host can configure a CGA-based IPv6 address and use its properties to provide a "proof of ownership" when exchanging NDP messages with other hosts located on the same link.

This document introduces a simple mechanism which enables a host using CGA IPv6 address to delegate the task of "SeND proxying" to its AR and/or to another node(s) by means of establishing a new and unique form of "strong but distant" relationship that we refer to in the rest of this document as 'symbiotic'.

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](#)].

[3.](#) Motivation

Our motivation behind this work is three-fold:

- o provide a secure assistance for mobile nodes (MNs) while being active but away from their home link, e.g., case of mobile IPv6 protocol (more below).
- o enable a weak (still to be improved) form of anonymity on the link by preventing a particular host from disclosing its CGA public key especially when switching between different links.
- o extend SeND proxying assistance in some particular scenarios, to static/mobile host which is not CGA enabled.

[4.](#) What is a 'Symbiotic' Relationship?

A 'symbiotic' relationship (SR) is a unidirectional association between two nodes A and B. This means that when node A establishes an SR with node B, then node B is assumed to be the only node which is able to advertise such relationship to a third party and to provide a "proof of relationship (PoR)" (i.e., similar to providing a CGA proof of ownership) with node A. Consequently, establishing an SR with a node B can empower it, if/when needed, to act on behalf of node A and regardless of the latter's current location.

It follows that establishing a bidirectional SR between nodes A and B enables any of them to act on behalf of the other and also to provide each a different PoR to a third party.

Furthermore, a node is also able to establish different SRs with a

group of nodes in a single operation. Once established, each node from the group has to extract the specific SR which shows its own involvement, i.e., by re-arranging the parameters, in order to prove it to a third party.

A key element in the CGA mechanism is to generate a 128-bit random RAN(128) parameter which must be sent, among others, to the receiver in order to enable it to re-compute the CGA IPv6 address before verifying the signature.

When establishing an SR from node A to node B, the only required modification involves the RAN(128) generated by A when configuring its CGA address. Such modification consists on replacing the RAN(128) with another new random 128-bit (we call it SR_RAN(128)) which is generated from the RAN(128) and B's public key (Kp). The SR_RAN(128), is obtained from concatenating the previous RAN(128) with B's Kp then hashing the concatenation. Then, A takes the first 128 bits of the resulting hash and uses it as the SR_RAN(128) which will replace the previous RAN(128) when computing the 64-bit interface identifier (IID) for its CGA IPv6 address. In summary the previous RAN(128) used to generate the IID without SR is in fact dissolved in the new one, i.e., SR_RAN(128), which is now used for generating A's IID.

The rule for computing an SR_RAN(128) when establishing an SR with a node using Kp as public key is as follows:

$$\text{SR_RAN}(128) = \text{First}[128, \text{Hash}(\text{Kp} \mid \text{RAN}(128))]$$

Where:

- First(size, input) indicates a truncation of the "input" data so that only the first "size" bits remain to be used.
- RAN(128) is the 'previous' 128-bit random parameter which was supposed to be used for configuring a CGA address without an SR.

- "|" denotes a concatenation
- The recommended hash function is SHA256

We assume in the following that a CGA-enabled host (H) is attaching itself to a link protected with SeND protocol in which case, the AR is signing its router advertisement (RtAdv) messages. This means first and foremost, that (H) can securely get a copy of AR's certificate and trust its content.

As previously shown, establishing an SR between a host and its AR is a simple procedure which does not introduce any change in the mechanism designed for configuring a CGA IPv6 address per se. In fact, the introduced modification occurs in the "background" of the CGA mechanism. An important feature of such design is that it does not constrain the host (H) to disclose the elements behind the SR, i.e., how the SR_RAN(128) has been computed from the AR's Kp. This means that the host can keep using CGA technique by simply presenting its SR_RAN(128) as a normal RAN(128) parameter and avoid disclosing its SR except when needed.

After computing the new SR_RAN(128) parameter, (H) proceeds to generate its IPv6 address as defined in the CGA mechanism. When (H) needs to disclose the SR to its AR, e.g., to request proxying services, then it has to insert the RAN(128) used to generate the SR_RAN(128) in a new option (called SRO) to be carried, for example, in the router solicitation (RtSol) message sent to the AR or in an NDP message. In addition, (H) SHOULD encrypt SRO using the AR's Kp.

Upon receiving a RtSol message carrying an SRO, the AR should first check the SR validity. This is achieved by decrypting the SRO then checking if the IPv6 address is generated from using its own Kp. If the check is valid, then the AR should proceed to check the address ownership and verify the signature. After that, the AR SHOULD store the host's RAN(128) together with its IP address, public key and all associated CGA parameters. It follows that if (H) decides not to reveal its SR to its AR, then it can continue using SeND protocol by disclosing only its new SR_RAN(128) parameter (i.e., as a RAN(128)).

It follows that an AR MUST NOT accept an SR sent by a node which has configured a CGA IPv6 address unless the message carrying the SR is signed by the node's CGA private key.

When establishing different SRs with a group of nodes having each a public key, the host needs to concatenate all group nodes public keys with the RAN(128) before hashing the concatenation and taking the first 128 bits resulting from the hash as its SR_RAN(128). As mentioned earlier, each node from the group should be able to extract the specific SR which involves its public key and uses other group nodes public keys together with the RAN(128) as parameters to be sent to a third party when disclosing the specific SR.

As an SR is mainly about creating a crypto-relationship with another node, its key feature is that disclosing it to a third party, i.e., by providing a proof of relationship, makes sense only when it is done by the owner of the public key (Kp) hashed with the RAN(128) in order to produce SR_RAN(128). This is due to the fact that without a proof of ownership of Kp itself, the third party MUST reject the proof of relationship. In fact, when such situation arises, e.g., AR needs to act on behalf of (H), then it SHOULD add (H)'s IPv6 address and all CGA components that (H) has used to generate it. These components MUST include RAN(128) and the AR's public key instead of the SR_RAN(128) parameter. In addition, the AR MUST sign the message. It follows that no other node can claim the same privilege of acting on behalf of (H) unless it can discover AR's private key in order to correctly sign the message. We assume such scenario to be highly unlikely. The other alternative for a malicious node to claim the same SR with (H) is to generate another key pair then try to rebuild the whole chain of parameters which leads to the same IPv6 address used by (H).

Another potential scenario to explore is to use SR by a non-CGA host in a SeND environment. One possibility is for (H) to derive its IPv6 address by applying the same rule mentioned earlier with the difference that it has to take the first 64-bit (instead of 128 bits) and use them directly as interface identifier for configuring its IPv6 address. In such scenario, the host has to send a RtSol message to the AR in which, it has to include the SRO and encrypts it with AR's Kp. Note however that in such scenario, the RtSol message won't be signed.

A second potential path which also requires more investigation is related to manipulating SR in stateful address configuration and in a SeND environment. In fact, it may be possible to have an SR automatically established between a host and its AR when stateful address configuration is in place. This can be done by enabling the DHCP to generate IPv6 addresses to be allocated to hosts, in the same way as described for non-CGA host. The CGA MUST then share the RAN(128) with the AR without the host knowledge nor involvement. In such scenario, the AR may signal to the host its ability to act on its behalf by setting a bit in the RtAdv message.

[5.1.](#) Using SR for SeND Proxying

It follows from the above that SR simplicity and efficiency makes it a suitable candidate for enabling SeND proxying to mobile/static hosts. In order to do so, each host has to establish an SR with the secure NDP proxying node(s) (which may be the AR itself). In case

the AR is not empowered to offer NDP proxying services, then it SHOULD advertise -at least- the public key(s) of the node(s) which is

playing this role. Upon receiving an additional public key(s) in the RtAdv message sent by AR, (H) may decide to use it to establish an SR with its holder either directly, i.e., if the NDP proxying node's IP address is known, or via the AR.

In fact, as we're assuming that SeND protocol is deployed, which means first and foremost that (H) can trust the access infrastructure and the information that it is sending (and also validate it), then we can also assume with the same level of trust that the additional public key(s) advertised by the AR is also genuine and is owned by the real node offering proxying services.

Following a decision to delegate secure NDP proxying services to the owner of the public key sent in the RtAdv messages, (H) applies the rule described earlier to establish an SR with the proxying node when configuring its CGA IPv6 address. Once the CGA address uniqueness is checked, (H) can start using it as a normal CGA address as long as it does not need to request a proxying service.

One way to trigger delegating SeND NDP proxying task is to disclose the SR parameter to the AR and/or the NDP proxying node. This can be done for example, by sending a RtSol message which carries the RAN(128) in an SR0. Note that (H) SHOULD encrypt the RAN(128) with the proxying node's public key. After sending the RtSol message, (H) SHOULD stop replying to any NDP query. In other words, (H) will be able to decide when to "vanish" from the link whenever it sees it appropriate.

Furthermore, and for privacy purposes, the MN may decide to delegate the proxying task even while being physically attached to the link, in order to avoid disclosing its own CGA public key when signing NDP messages. In fact, disclosing the public key can severely harm the unlinkability aspect especially when the MN is using pseudo-IPv6 address(es) when switching between different links. This may be the case for example, when performing IP handoff between different ARs while trying to keep a certain level of location privacy which should not be broken by disclosing the CGA public key.

When acting as a SeND NDP proxy on behalf of (H), the dedicated node MAY include in the NDP message sent on behalf of the host all its CGA

parameters as well as the RAN(128) in order to enable the querier node to derive the host's IPv6 address before checking the NDP message validity. However, as the proxying node's public key is advertised by the trusted AR, such node can simply sign the NDP message sent on behalf of (H). In order to protect against replay attacks, the querier node MUST always use a nonce in each query sent to the proxying node. The nonce MUST be returned in the response sent by the proxying node in order to put an implicit lifetime, i.e., by associating the response to the query which triggered it. Note

that, in case the queried IPv6 address cannot be computed from parameters sent by the AR, the querier node MUST silently discard the message.

Mobile IPv6 protocol (described in [[I-D.ietf-mext-rfc3775bis](#)]) is a typical scenario where a mobile node (MN) needs to stay attached to its home link, i.e., via its home agent (HA), even when being physically attached to a foreign one. In this case, the HA is supposed to act on behalf of the MN and respond to any NDP query sent on the home link. Based on the above, all what the MN needs to do is to establish and activate an SR with its HA, regardless of its topological location (i.e., the MN may bootstrap while being attached to a foreign link).

Haddad & Naslund

Expires January 30, 2010

[Page 11]

Internet-Draft

Proxy SeND

July 2009

[6.](#) New Option

TBD

7. Security Considerations

This memo describes a mechanism which enables a host to delegate the task of performing SeND NDP proxying to another node by means of establishing a new type of relationship with that node. In its current form, the new mechanism is built on top of CGA technology.

The security of such delegation is inherited from the existence of a SeND environment which enables a host to establish a form of trust with the access router(s). In our proposal, we assume that such trust will be expanded to the relation between the host and the proxying node(s), i.e., in case such node is not the AR itself. It also means that the same trust can be assumed to reign between any host located on the same link than the 'proxied' host and the proxy node. Under such assumption, whenever the proxy node needs to disclose the SR relationship to a third party, e.g., querier node, it can only show the SR components in a message that must be signed with the proxy node's private key.

However, in the absence of the assumed trust between the querying node(s) and the proxy node(s), then the latter must include the proxied node signature in the proof of relationship that it may need to disclose. Furthermore, in such environment, the proxied's node signature cannot have an unlimited lifetime. Consequently, the proxied node has to bind its signature to a fixed lifetime after which, it becomes obsolete unless it is refreshed by the proxied node. An alternative may be to announce a pre-defined lifetime for any proxying request. It follows that in such scenario, the proxied node's public key has to be disclosed to the queried node, which in turn may preserve the queried node's location privacy but certainly hurt any anonymity and unlinkability goals. Note that a direct consequence of a binding between signature and lifetime is a requirement for synchronization between the proxying node and the querying one(s).

8. Normative References

[I-D.ietf-mext-rfc3775bis]

Johnson, D., Perkins, C., and J. Arkko, "Mobility Support in IPv6", [draft-ietf-mext-rfc3775bis-03](#) (work in progress), March 2009.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.

[RFC3971] Arkko, J., Kempf, J., Zill, B., and P. Nikander, "SEcure

Neighbor Discovery (SEND)", [RFC 3971](#), March 2005.

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

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", [RFC 4861](#), September 2007.

6210 Spine Road
Boulder, CO 80301
US

Phone: +303 473 6963
Email: Wassim.Haddad@ericsson.com

Mats Naslund
Ericsson Research
Torshamnsgatan 23
SE-164 80 Stockholm
Sweden

Phone: +46 8 58533739
Email: Mats.Naslund@ericsson.com