Public Key Algorithm Agility in SEND
draft-shen-csi-ecc-00.txt

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of 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/1id-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 2, 2009.

Abstract

This draft proposes an upgrade to the CGA/SEND protocols to incorporate the use of elliptic curve cryptography.
# Table of Contents

1. Requirements notation ........................................ 3  
2. Background ..................................................... 3  
3. Motivation ....................................................... 3  
4. New NDP Options .................................................. 4  
   4.1. ECC signature option ....................................... 4  
5. New CGA parameters specification .............................. 6  
   5.1. New CGA Parameter Data Structure .......................... 6  
   5.2. ECC recommended parameters ............................... 8  
6. Compatibility Issues .............................................. 8  
7. Security Considerations ........................................ 9  
   7.1. Hash Function in ECDSA ................................... 9  
8. IANA Considerations ............................................. 9  
9. References ....................................................... 10  
   9.1. Normative References ..................................... 10  
   9.2. Informative References .................................. 10  
Authors’ Addresses ................................................. 11  

Intellectual Property and Copyright Statements ................. 12
1. Requirements notation

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].

This draft follows the terminology that has been defined in [RFC3971] and [RFC3972]. In addition, this draft uses the following terms:

Elliptic Curve Cryptography (ECC)  A public-key cryptosystem based on the properties of elliptic curves over finite fields. The operations relevant for this work are digital signature derivation and verification.

Elliptic Curve Digital Signature Algorithm (ECDSA)  A standard digital signature algorithm based on ECC.

2. Background

The Secure Neighbor Discovery (SEND) protocol [RFC3971] was designed out of a need to secure IPv6 Neighbor Discovery messages against spoofing of addresses, in an effort to guard against several attacks, notably Denial of Service attacks. The neighbor discovery problem lends itself to the use of public key cryptosystems.

At the time when SEND was designed the public key cryptosystem of choice was RSA-based, due to its widespread use, availability of implementation, etc. At about the same time [RFC3971] was published, the National Security Agency (NSA) announced the incorporation of ECC in its set of recommended algorithms for modern use ("Suite B").

3. Motivation

Since the publication of SEND, the Internet has witnessed several trends: On the one hand, some existing protocols have been upgraded to support elliptic curve cryptography -- for example TLS (see [RFC4492]), and IKE/IKEv2 (see [RFC4754]). On the other hand, those interested in maintaining protocols based on RSA are inclined to heed the cryptographic community’s (e.g. NIST) recommendation to deprecate 1024-bit RSA algorithms by the year 2010, in favor of longer keys or else alternate algorithms. Finally, an increasing number of power or memory-limited devices have joined the Internet, and to them the computational burden of RSA-based protocols is undesirable.

The efficiency of ECC is well-known, since it provides similar...
It should be clarified that this document does not intend to make the strength of the ECC signature match that of the currently specified in [RFC3971] RSA signature; RSA algorithms based on 1024 bit keys provide similar cryptographic strength to ECC algorithms that use 163 bit keys, but 163 bit keys are not widely used, nor recommended for ECC. Instead, we recommend that the default key length for ECDSA be set to 256 bits.

Given the well-known strength and relatively low computational cost for ECC-based signature generation/verification, it is then imperative for the SEND/CGA protocols to also allow an option to employ ECC.

4. New NDP Options

The SEND protocol calls for the use of the "CGA option" to secure its messages. This option contains a field of "CGA parameters", which is taken from the CGA specification. Therefore, the CGA option of SEND does not need modification in order for SEND to support ECC-based options. However, the CGA specification ([RFC3972]) does and it will be addressed in section 5.

The SEND protocol calls for the use of public key cryptography in the construction and validation of the "RSA signature option". This option is required in order to validate that the CGA present in the "CGA option" was actually generated by the owner of the public key present in the CGA parameters list. Therefore, the SEND protocol needs to be updated to support an "ECC signature option".

4.1. ECC signature option

The ECC Signature option is very similar to the RSA Signature option. The ECC Signature option allows public key-based signatures to be attached to NDP messages. The format of the ECC Signature option is described in the following diagram:
### Figure 1

<table>
<thead>
<tr>
<th>0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1</th>
</tr>
</thead>
</table>
| +---------------------------------------------------+
| | Type | Length | Reserved |
| +---------------------------------------------------+
| | Key Hash |
| +---------------------------------------------------+
| | Digital Signature |
| +---------------------------------------------------+
| | Padding |
| +---------------------------------------------------+

**Type**

17

**Length**

8-bit unsigned integer. The length of the option (including the Type, Length, Reserved, Key Hash, Digital Signature, and Padding fields) in units of 8 octets. The value 0 is invalid.

**Reserved**

A 16-bit field reserved for future use. The value MUST be initialized to zero by the sender, and MUST be ignored by the receiver.

**Key hash**

A 128-bit field containing the most significant (leftmost) 128 bits of a SHA-256 [FIPS.180-2] hash of the public key used for constructing the signature. The SHA-2 hash is taken over the presentation used in the Public Key field of the CGA Parameters data structure carried in the CGA option. Its purpose is to
associate the signature with a particular key known by the receiver. Such a key can either be stored in the certificate cache of the receiver or be received in the CGA option in the same message.

**Digital Signature**

A variable-length field containing an ECDSA as defined in [SEC1], constructed by using the sender’s private key over the following sequence of octets:

- The 128-bit CGA Message Type tag [RFC3972] value for SEND, specified in section 5.2 of [RFC3972];
- The 128-bit Source Address field from the IP header;
- The 128-bit Destination Address field from the IP header;
- The 8-bit Type, 8-bit Code, and 16-bit Checksum fields from the ICMP header;
- The NDP message header, starting from the octet after the ICMP Checksum field and continuing up to but not including NDP options;
- All NDP options preceding the ECC Signature option.

The signature value is computed as defined in [SEC1]. The length of the Digital Signature field is determined by the length of the ECC Signature option minus the length of the other fields (including the variable length Padding field).

**Padding**

This variable-length field contains padding, as many bytes long as remain after the end of the signature, so that the total length is a multiple of 8 octets.

5. New CGA parameters specification

The ECDSA signature should be calculated via the algorithm defined in [SEC1] or other compatible standards ([FIPS.186-3], [ANSI-X962]).

5.1. New CGA Parameter Data Structure

The new CGA parameters data structure associated with ECC has the following format:
### Modifier

This field contains a 128-bit unsigned integer, which can be any value. The modifier is used during CGA generation to implement the hash extension and to enhance privacy by adding randomness to the address.

### Subnet Prefix

This field contains the 64-bit subnet prefix of the CGA.

### Collision Count

This is an eight-bit unsigned integer that MUST be 0, 1, or 2. The collision count is incremented during CGA generation to recover from an address collision detected by duplicate address detection.

### Public Key

This is a variable-length field containing the public key of the address owner. The public key MUST be formatted as a DER-encoded...
[ITU-X690-2002] ASN.1 structure of the type SubjectPublicKeyInfo, defined in [SEC1].

Extension Fields

This is an optional variable-length field that is not used in the current specification.

5.2. ECC recommended parameters

As specified by [RFC3971] and [RFC3972], all nodes that support the verification of the CGA option MUST record a minimum acceptable key length for RSA public keys. The default RSA key length SHOULD be 1024 bits. Implementations MAY also set an upper limit for the amount of computation needed when verifying packets that use these security associations. The upper limit SHOULD be at least 2048 bits. Meanwhile, the minimum RSA key length for SEND is required to be 384 bits.

The key size of ECDSA used in SeND and CGA MUST be 256, 384 or 521; the default key size SHOULD be 256-bit. As indicated in [NIST-800-57], ECC with a 256-bit key size provides similar strength to RSA with a 3072-bit key size. Hence the widely used ECDSA with at least 256-bit key size will provide security strength no less than the RSA signature defined SeND/CGA.

To ensure interoperability, we recommend a few widely used curves to narrow the set. We recommend three curves with above-mentioned key sizes (from [SEC2]): secp256r1, secp384r1 and secp521r1.

6. Compatibility Issues

To ensure backward compatibility, and pave the way to interoperability between new and old implementations, the RSA signature SHOULD be the default for both CGA and SeND. ECDSA support in CGA and SeND is OPTIONAL.

The negotiation of the PKI algorithms -- RSA, ECDSA ---to be used in CGA and SeND between two network nodes is outside the scope of this draft. It is expected that future versions of the SeND protocol will specify the negotiation process and sender/receiver’s action. It is expected that the PKI algorithm agility scheme is just dealing with the choice between RSA and ECDSA, but aiming to be a general scheme dealing with all possible PKI algorithms.
7. Security Considerations

The security considerations relating to CGAs are also applicable here. They are outlined in [RFC3971].

As discussed in section 5, ECDSA signatures provide increased security strength given the same key size compared to RSA signature.

It should be clarified that ECDSA signatures do not provide any additional security features compared to RSA signatures, when used in the context of CGA/SEND as discussed above.

7.1. Hash Function in ECDSA

Both CGA and SeND specify SHA-1 as the hash function used in address computation and respectively digital signature. The cryptography community no longer recommends the use of SHA-1 for new security protocols. According to [NIST-800-57], SHA-256 and SHA-384 should be used with ECDSA of 256 and 384 key size, respectively.

NIST intends to deprecate or at least replace the SHA-2 variants with a new hash function, SHA-3. The new hash function is not expected to be standardized until 2012.

Since the proposed changes to CGA/SEND rely on existing ECDSA standards, and these standards all make use of SHA-1, this document does not upgrade the hash function. In the future, however, the CGA/SEND specifications should be updated to keep up with the new NIST standard.

Part of work on agility of Hash function in SeND and CGA has been done [RFC4982] and more related work is in process in CSI Working Group.

8. IANA Considerations

This document defines one new Neighbor Discovery Protocol [RFC2461] option, which must be assigned Option Type values within the option numbering space for Neighbor Discovery Protocol messages:

The ECC Signature option (17), described in Section 4.1.

9. References
9.1. Normative References


9.2. Informative References


[RFC4754] Fu, D. and J. Solinas, "IKE and IKEv2 Authentication Using
the Elliptic Curve Digital Signature Algorithm (ECDSA)", RFC 4754, January 2007.


Authors’ Addresses

Sean Shuo
Huawei

Email: sshen@huawei.com

Michaela Vanderveen
Qualcomm

Email: mvandervn@yahoo.com