DHC Working Group
Internet Draft
Intended status: Standards Track
Update: RFC3315
Expires: September 18, 2012

Sheng Jiang
Huawei Technologies Co., Ltd
Sean Shen
CNNIC
March 12, 2012

# Secure DHCPv6 Using CGAs
# draft-ietf-dhc-secure-dhcpv6-06.txt

## Status of this Memo



## Copyright Notice

Abstract

The Dynamic Host Configuration Protocol for IPv6 (DHCPv6) enables DHCPv6 servers to pass configuration parameters. It offers configuration flexibility. If not secured, DHCPv6 is vulnerable to various attacks, particularly spoofing attack. This document analyzes the security issues of DHCPv6 and specifies a Secure DHCPv6 mechanism based on using CGAs.

Table of Contents

## 1. Introduction

The Dynamic Host Configuration Protocol for IPv6 (DHCPv6 [RFC3315]) enables DHCPv6 servers to pass configuration parameters. It offers configuration flexibility. If not secured, DHCPv6 is vulnerable to various attacks, particularly spoofing attack.

This document analyzes the security issues of DHCPv6 in details. This document is aiming to provide mechanisms for improving the security of DHCPv6, thus the address of a DHCPv6 message sender, which can be a DHCPv6 server, a relay agent or a client, can be verified by a receiver. It improves communication security of DHCPv6 interaction. The security mechanisms specified in this document is mainly based on the Cryptographically Generated Addresses (CGA [RFC3972]).

Secure DHCPv6 is applicable in environments where physical security on the link is not assured (such as over wireless) and attacks on DHCPv6 are a concern.

## 2. Terminology

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. Security Overview of DHCPv6

DHCPv6 is a client/server protocol that provides managed configuration of devices. It enables DHCPv6 server to auto-configure relevant network parameters on clients through the DHCPv6 message exchanging mechanisms. In the basic DHCPv6 specifications [RFC3315], security of DHCPv6 message can be improved in a few aspects.

a) In the basic DHCPv6 specifications, the DHCPv6 server uses a "regular" IPv6 address for itself. It is possible for a malicious attacker to use a fake address to spoof or launch an attack. See Section 23, "Security Considerations" of [RFC3315] for more details.

   Furthermore, if DHCPv6 servers play the role of updating DNS and other directory services, attackers may spoof DHCPv6 servers to register incorrect information in those services.

   CGA-based security mechanism can provide source address ownership proofing, which prevents such attacks.

b) The basic DHCPv6 specifications achieve message origin authentication and message integrity via an authentication option

with a symmetric key pair. For the key of the hash function, there are two key management mechanisms. Firstly, the key management is out of band, usually manual, i.e. operators set up key database for both server and client before running DHCPv6. Usually multiple keys are deployed once a time and key id is used to specify which key is used. Manual key distribution runs counter to the goal of minimizing the configuration data needed at each host. Secondly, a DHCPv6 server sends a reconfigure key to the client in the initial exchange of DHCPv6 messages for future use, in this case security is not guaranteed because this key is transmitted in plaintext.

Comparing to this, CGA-based security mechanism does not request any key management mechanisms.

c) Communication between a server and a relay agent, and communication between relay agents, can be secured through the use of IPsec, as described in section 21.1 in [RFC3315]. However, IPsec is quite complicated. A simpler security mechanism may have better deploy ability.

## 4. Secure DHCPv6 Overview

To solve the abovementioned security issues, we introduce CGAs into DHCPv6. CGAs are introduced in [RFC3972]. The usage of CGAs combining with associated signatures can verify the address ownership and protect messages "without a certification authority or any security infrastructure." [RFC3972]

This documentation introduces a Secure DHCPv6 mechanism that uses CGAs to secure the DHCPv6 protocol. It assumes the secured DHCPv6 message sender has already haven CGAs and their correspondent CGA parameters; and the receiver has already been given the CGAs of the sender.

In this document, a CGA option with an address ownership proof mechanism and a signature option with a corresponding verification mechanism are introduced. A DHCPv6 message (from either a server, a relay agent or a client) with a CGA as source address, can carry the CGA Parameters data structure and a digital signature. The receiver of this DHCPv6 message, who has already known the CGA of the sender, can verify both the CGA and signature, then process the payload of the DHCPv6 message only if the validation is successful.

With them, the receiver of a DHCPv6 message can verify the sender address of the DHCPv6 message, which improves communication security of DHCPv6 messages. The verification of data integrity and replay protections can also be achieved without the authentication option.

The sender can be a DHCPv6 server, a relay agent or a client. So, the

end-to-end security protection can be from DHCPv6 servers to relay

agents or clients, or from clients to relay agent or DHCPv6 servers. Relay agents MAY add its own Secure DHCPv6 options, too.

### 4.1. New Components

The components of the solution specified in this document are as follows:

- CGAs are used to make sure that the sender of a DHCPv6 message is the "owner" of the claimed address. A public-private key pair has been generated by a node itself before it can claim an address. A new DHCPv6 option, the CGA Parameter Option, is used to carry the public key and associated parameters.

- Public key signatures protect the integrity of the DHCPv6 messages and authenticate the identity of their sender.

- Server Address type of DUID is used to carry server's source address in the relay scenarios. The receiver gets the server's source CGA address for CGA verification.

### 4.2. Support for algorithm agility

Hash functions are the fundamental of security mechanisms, including CGAs in this document. "...they have two security properties: to be one way and collision free." "The recent attacks have demonstrated that one of those security properties is not true." [RFC4270] It is theoretically possible to perform collision attack Attacks against the "collision-free" property.

Following the approach recommended by [RFC4270] and [NewHash], recent analysis shows none of these attacks are currently doable [RFC6273]. "The broken security property will not affect the overall security of many specific Internet protocols, the conservative security approach is to change hash algorithms." [RFC4270]

However, these attacks indicate the possibility of future real-world attacks. Therefore, we have to take into account that future attacks will be improved and provide a support for multiple hash algorithms. Our mechanisms, in this document, support not only hash algorithm agility but also signature algorithm agility.

The support for hash agility within CGAs has been defined in [RFC4982]. The usage of CGAs in this document SHOULD also obey [RFC4982], too.

The support for algorithm agility in this document is mainly unilateral notification model from a sender to a receiver. If the receiver cannot support the algorithm provided by the sender, it

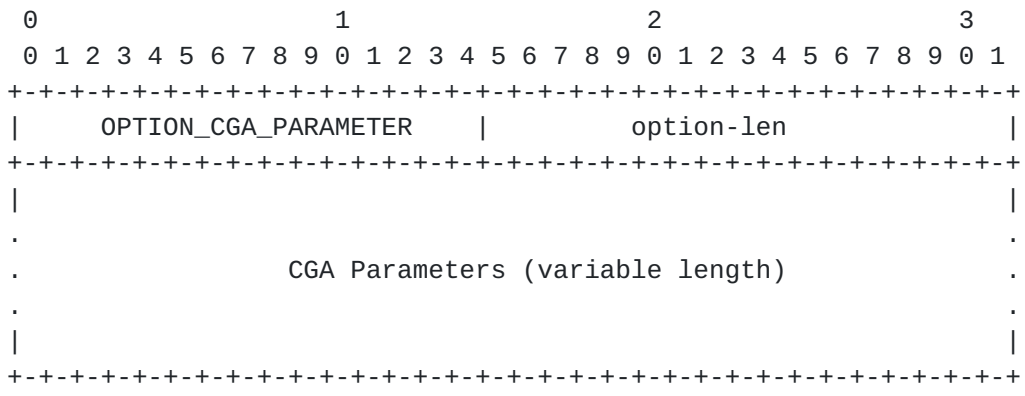
takes the risk itself. Senders in a same network do not have to upgrade to a new algorithm simultaneously.

## 5. Extensions for Secure DHCPv6

This section extends DHCPv6. Two new options and a new DUID type have been defined. The new options MUST be supported in the Secure DHCPv6 message exchanging. The new DUID type MUST be supported in the relay scenarios.

### 5.1. CGA Parameter Option

The CGA option allows the verification of the sender's CGAs. The format of the CGA option is described as follows:



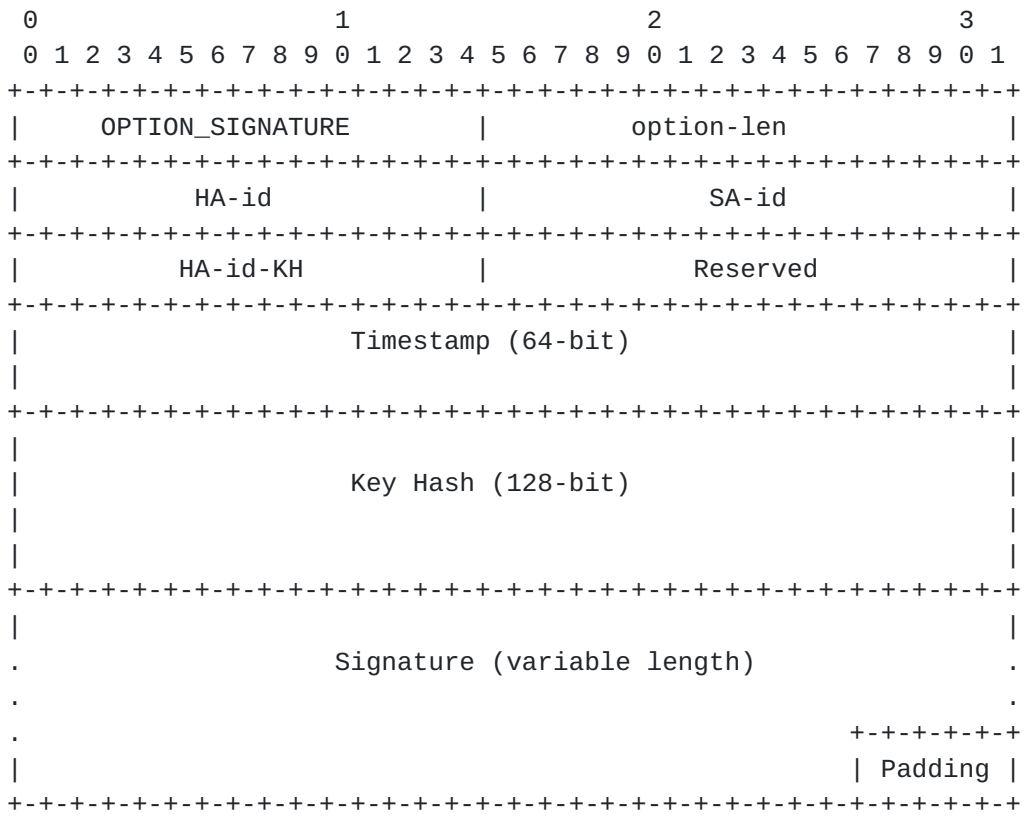

option-code OPTION_CGA_PARAMETER (TBA1).

option-len Length of CGA Parameters in octets.

CGA Parameters A variable-length field containing the CGA Parameters data structure described in Section 4 of [RFC3972]. This specification requires that the public key found from the CGA Parameters field in the CGA option MUST be that referred by the Key Hash field in the Signature option. Packets received with two different keys MUST be silently discarded.

### 5.2. Signature Option

The Signature option allows public key-based signatures to be attached to a DHCPv6 message. The Signature option could be any place within the DHCPv6 message. It protects all the DHCPv6 header and options, particularly including the CGA option, except for the Signature option and the Authentication Option. The format of the Signature option is described as follows:




option-code OPTION_SIGNATURE (TBA2).

option-len 32 + Length of Signature field and Padding field in octets.

HA-id Hash Algorithm id. The hash algorithm is used for computing the signature result. This design is adopted in order to provide hash algorithm agility.

SA-id Signature Algorithm id. The signature algorithm is used for computing the signature result. This design is adopted in order to provide hash algorithm agility.

HA-id-KH Hash Algorithm id for Key Hash. Hash algorithm used for producing the Key Hash field in the Signature option. This design is adopted in order to provide hash algorithm agility.

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.

Timestamp — The current time of day (NTP-format timestamp [RFC5905], a 64-bit unsigned fixed-point number, in seconds relative to 0h on 1 January 1900.). It can reduce the danger of replay attacks.

Key Hash — A 128-bit field containing the most significant (leftmost) 128 bits of the hash value of the public key used for constructing the signature. The hash algorithm is indicated in the HA-id-KH field. The field 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 to 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.

Signature — A variable-length field containing a digital signature. The signature value is computed with the hash algorithm and the signature algorithm, as described in HA-id and SA-id. The signature constructed by using the sender's private key protects the following sequence of octets:

1. The 128-bit CGA Message Type tag value for Secure DHCPv6, 0x81be a1eb 0021 ce7e caa9 4090 0665 d2e0 02c2.

2. The 128-bit Source IPv6 Address.

3. The 128-bit Destination IPv6 Address.

4. The DHCPv6 message header.

5. All DHCPv6 options except for the Signature option and the Authentication Option.

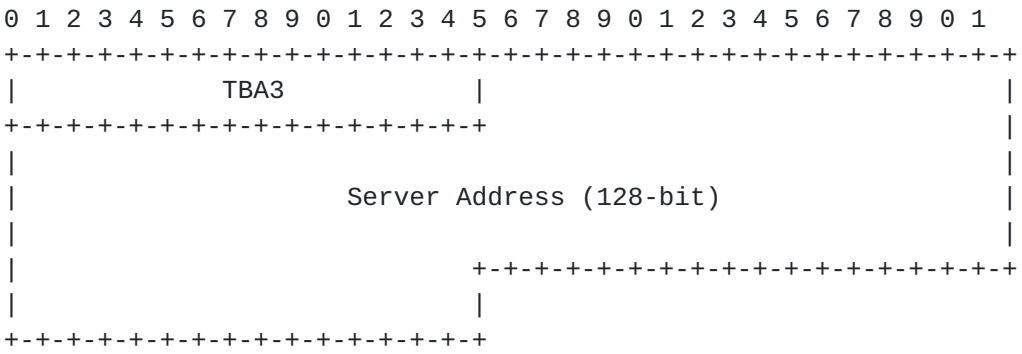
6. The content between the option-len field and the signature field in this Signature option, in the format described above.

Padding — This variable-length field contains padding, as many bits long as remain after the end of the signature. This padding is only needed if the length of signature is not a multiple of 8 bits.

### 5.3. DUID-SA Type

Server Address Type DUID (DUID-SA) allows IP address of DHCPv6 servers can be carried in DHCPv6 message payload.

The following diagram illustrates the format of a DUID-SA:




Type-code DUID-SA Type (TBA3)

Server Address The 128-bit IPv6 address of the DHCPv6 server.

The Server Address field of DUID-SA, which is the IPv6 address of the DHCPv6 server, MUST be a CGA.

In the server-relay-client scenarios, a DHCPv6 server knows a client is behind relay(s) if it receives a Relay-forward DHCPv6 message. Then it will reply a Relay-reply message with the server's source CGA address being carried in the Server Address Type DUID, which is in the payload. In this way, the receiver, a DHCPv6 client can get the server's source CGA address for CGA verification.

All the payloads, including DUID-SA, are protected by signature option by the definition of section 5.1 and 5.2.

## 6. Processing Rules and Behaviors

### 6.1. Processing Rules of Sender

The sender of a Secure DHCPv6 message could be a DHCPv6 server, a DHCPv6 relay agent or a DHCPv6 client.

The node MUST have the following information in order to create Secure DHCPv6 messages:

CGA parameters Any information required to construct CGAs, as described in [RFC3972].

| | |
|---|---|
| Keypair | A public-private key pair. The public key used for constructing the signature MUST be the same in CGA parameters. |
| CGA flag | A flag that indicates whether CGA is used or not. |

To support Secure DHCPv6, the Secure DHCPv6 enabled sender MUST construct the DHCPv6 message following the rules defined in [RFC3315]. The sender MUST use a CGA, which be constructed as specified in Section 4 of [RFC3972], as the source address, unless they are sent with the unspecified source address.

A Secure DHCPv6 message MUST contains both the CGA option and the Signature option.

The CGA option is constructed according to the rules presented in Section 5.1 and in [RFC3972]. The public key in the field is the one associated with the CGA, which is also the source address in the message header.

The Signature option MUST be constructed as explained in Section 5.2. It protects all DHCPv6 options (including the CGA option) except for the Signature option itself and the Authentication Option, the message header and the message payload

When constructing a Relay-reply message, a DHCPv6 server MUST include an OPTION_SERVERID [RFC3315] and put its CGA in the Server Address field of the DUID in the OPTION_SERVERID. By applying this rule, the CGA of the DHCPv6 server will not be lost when the Relay-reply message passes relay agents so that the client can verify CGA address and signature.

### 6.2. Processing Rules of Receiver

By receiving a DHCPv6 message, a Secure DHCPv6 enabled receiver MUST discard the DHCPv6 message if either the CGA option or the Signature option absents.

The receiving node MUST verify the source CGA address of the DHCPv6 message by using the public key of the DHCPv6 message sender, CGA Parameters and the algorithm described in Section 5 of [RFC3972]. The inputs to the algorithm are the source address, as used in IP header, and the CGA Parameters field. In the relay scenarios, a DHCPv6 server obtains the CGA of a client from the peer address field in the Relay-forward message. A DHCPv6 client obtains the CGA of a server from the Server Address field of the DUID in the OPTION_SERVERID.

If the CGA verification is successful, the recipient proceeds with a more time-consuming cryptographic check of the signature. Note that

even if the CGA verification succeeds, no claims about the validity of the use can be made until the signature has been checked.

The receiving node MUST verify the Signature option as follows: the Key Hash field MUST indicate the use of a known public key, the one learned from a preceding CGA option in the same message. The signature field verification MUST show that the signature has been calculated as specified in Section 5.2.

Only the messages that get through both CGA and signature verifications are accepted as secured DHCPv6 messages and continue to be handled for their contained DHCPv6 options as defined in [RFC3315]. Messages that do not pass all the above tests MUST be discarded.

Furthermore, the node that supports the verification of the Secure DHCPv6 messages MAY record the following information:

| | |
|---|---|
| Minbits | The minimum acceptable key length for public keys used in the generation of CGAs. An upper limit MAY also be set for the amount of computation needed when verifying packets that use these security associations. The appropriate lengths SHOULD be set according to the signature algorithm and also following prudent cryptographic practice. For example, minimum length 1024 and upper limit 2048 may be used for RSA [RSA]. |

### 6.3. Processing Rules of Relay Agent

To support Secure DHCPv6, Relay Agents MUST follow the same processing rules defined in [RFC3315].

A relay agent MAY verify the CGA and signature as a receiver before relay the DHCPv6 message further, following verification procedure define in Section 6.2. In the case of failure, it MUST discard the DHCPv6 message.

In the relay scenarios, because relay agent restructures the DHCPv6 messages, a downstream receiver would not find the sender's source CGA address in the DHCPv6 message header.

In the client-relay-server scenarios, "The relay agent copies the source address from the IP datagram in which the message was received from the client into the peer-address field in the Relay-forward message" [RFC3315]. Therefore, the CGA of a client will not be lost during the relay processing from the client to the server. The receiver, a DHCPv6 server, can find the sender's source CGA address in the peer-address field for CGA verification.

During the relay processing from the server to the client, when the relay agent constructs the Relay-reply message the server's IP address is replaced by the relay's IP address. In order to make the CGA of the DHCPv6 server reach the client, DUID-SA, described in Section 5.3, MUST be used. A relay will not change the OPTION_SERVERID when processing Relay-reply message from a DHCPv6 server, so that the CGA of the DHCPv6 server will not be lost when the Relay-reply message passes the Relay Agent.

Relay agents MAY also added its own CGA option and signature option in the Relay-forward or Relay-reply messages. By receiving such messages, the downstream receiver MUST verify CGA and signature from the relay agent, and CGA and signature from the original sender.

## 7. Security Considerations

This document provides new security features to the DHCPv6 protocol.

Using CGA as source addresses of DHCPv6 servers, relays or, also in DHCPv6 message exchanging provides the source address ownership verification and data integrity protection.

The Secure DHCPv6 mechanism is based on the precondition that the receiver has known the CGA of senders. For example, to prevent DHCPv6 server spoofing, the clients should be pre-notified the DHCPv6 server CGA. The clients may decline the DHCPv6 messages from other servers, which may be fake servers. The pre-notification operation also needs to be protected, which is out of scope.

DHCPv6 nodes without CGAs or the DHCPv6 messages that use unspecific addresses cannot be protected.

Downgrade attacks cannot be avoided if nodes are configured to accept both secured and unsecured messages. A future specification may provide a mechanism on how to treat unsecured DHCPv6 messages. One simple solution may be that Secure DHCPv6 is mandated on all servers, relay agents and clients on a certain link.

As stated in CGA definition [RFC3972], link-local CGAs are more vulnerable because the same prefix is used by all IPv6 nodes. Therefore, when link-local CGAs are used by the DHCPv6 clients, it is recommended to use a slightly higher Sec value, for example Sec=1 for now. When higher Sec values are used, the relative advantage of attacking link-local addresses becomes insignificant.

Impacts of collision attacks on current uses of CGAs are analyzed in [RFC4982]. The basic idea behind collision attacks, as described in Section 4 of [RFC4270], is on the non-repudiation feature of hash algorithms. However, CGAs do not provide non-repudiation features. Therefore, as [RFC4982] points out CGA-based protocols, including

Secure DHCPv6 defined in this document, are not affected by collision attacks on hash functions.

[RFC6273] has analyzed possible threats to the hash algorithms used in SEND. Since the Secure DHCPv6 defined in this document uses the same hash algorithms in similar way like SEND (except that Secure DHCPv6 has not used PKIX Certificate), analysis results could be applied as well: current attacks on hash functions do not constitute any practical threat to the digital signatures used in the signature algorithm in the Secure DHCPv6. Attacks on CGAs, as described in [RFC4982], will compromise the security of Secure DHCPv6 and they need to be addressed by encoding the hash algorithm information into the CGA as specified in [RFC4982].

## 8. IANA Considerations

This document defines two new DHCPv6 [RFC3315] options, which MUST be assigned Option Type values within the option numbering space for DHCPv6 messages:

The CGA Parameter Option (TBA1), described in Section 5.1.

The Signature Option (TBA2), described in Section 5.2.

This document defines a new DHCPv6 DUID, which MUST be assigned DUID Type values within the DHCPv6 DUID Type numbering space:

The DUID-SA (TBA3), described in Section 5.3.

This document defines three new registries that have been created and are maintained by IANA. Initial values for these registries are given below. Future assignments are to be made through Standards Action [RFC5226]. Assignments for each registry consist of a name, a value and a RFC number where the registry is defined.

Hash Algorithm id (HA-id). The values in this name space are 16-bit unsigned integers. The following initial values are assigned for HA-id in this document:

| Name | Value | RFCs |
|---|---|---|
| SHA-1 | 0x0000 | this document |

Signature Algorithm id (SA-id). The values in this name space are 16-bit unsigned integers. The following initial values are assigned for SA-id in this document:

| Name | Value | RFCs |
|---|---|---|
| RSASSA-PKCS1-v1_5 | 0x0000 | this document |

Hash Algorithm id for Key Hash (HA-id-KH). The values in this name space are 16-bit unsigned integers. The following initial values are assigned for HA-id-KH in this document:

| Name | Value | RFCs |
|---|---|---|
| SHA-1 | 0x0000 | this document |

This document defines a new 128-bit value under the CGA Message Type [RFC3972] namespace, 0x81be a1eb 0021 ce7e caa9 4090 0665 d2e0 02c2. (The tag value has been generated randomly by the editor of this specification. It may replaced by any IANA-allocated value when the specification is published.)

## 9. Acknowledgments

The authors would like to thank Bernie Volz, Ted Lemon, Ralph Dorms, Jari Arkko, Sean Turner, Stephen Kent, Thomas Huth, David Schumacher and other members of the IETF DHC & CSI working groups for their valuable comments.

## 10. References

### 10.1. Normative References

[RFC3315] R. Droms, et al., "Dynamic Host Configure Protocol for IPv6", RFC3315, July 2003.

[RFC3972] T. Aura, "Cryptographically Generated Address", RFC3972, March 2005.

[RFC4982] M. Bagnulo, J. Arkko, "Support for Multiple Hash Algorithms in Cryptographically Generated Addresses (CGAs)", RFC4982, July 2007.

[RFC5905] D. Mills, J. Martin, Ed., J. Burbank and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, June 2010.

### 10.2. Informative References

[RFC2119] S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", c, March 1997.

[RFC4270] Hoffman, P. and B. Schneier, "Attacks on Cryptographic Hashes in Internet Protocols", RFC 4270, November 2005.

[RFC5226] T. Narten and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 5226, May 2008.

[RFC6273] A. Kukec, S. Krishnan and S. Jiang "The Secure Neighbor Discovery (SEND) Hash Threat Analysis", RFC 6274, June 2011.

[NewHash] S.Bellovin and E. Rescorla, "Deploying a New Hash Algorithm", November 2005.

[RSA] RSA Laboratories, "RSA Encryption Standard, Version 2.1", PKCS 1, November 2002.

[sha-1] National Institute of Standards and Technology, "Secure Hash Standard", FIBS PUB 180-1, April 1995, http://www.itl.nist.gov/fipspubs/fip180-1.htm.

Author's Addresses

Sheng Jiang
Huawei Technologies Co., Ltd
Q14, Huawei Campus
No.156 Beiqing Road
Hai-Dian District, Beijing 100095
P.R. China
EMail: jiangsheng@huawei.com

Sean Shen
CNNIC
4, South 4th Street, Zhongguancun
Beijing 100190
P.R. China
EMail: shenshuo@cnnic.cn