DHC Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 25, 2017

L. Li
Tsinghua University
S. Jiang
Huawei Technologies Co., Ltd
Y. Cui
Tsinghua University
T. Jinmei
Infoblox Inc.
T. Lemon
Nominum, Inc.
D. Zhang
February 21, 2017

Secure DHCPv6
draft-ietf-dhc-sedhcpv6-21

Abstract

DHCPv6 includes no deployable security mechanism that can protect end-to-end communication between DHCP clients and servers. This document describes a mechanism for using public key cryptography to provide such security. The mechanism provides encryption in all cases, and can be used for authentication based on pre-sharing of authorized certificates.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The Dynamic Host Configuration Protocol for IPv6 (DHCPv6, [RFC3315]) allows DHCPv6 servers to flexibly provide addressing and other configuration information relating to local network infrastructure to DHCP clients. The protocol provides no deployable security mechanism, and consequently is vulnerable to various attacks.

This document provides a brief summary of the security vulnerabilities of the DHCPv6 protocol and then describes a new extension to the protocol that provides two additional types of security:

- o authentication of the DHCPv6 client and the DHCPv6 server to defend against active attacks, such as spoofing.

- o encryption between the DHCPv6 client and the DHCPv6 server in order to protect the DHCPv6 communication from pervasive monitoring.

The extension specified in this document applies only to end-to-end communication between DHCP servers and clients. Options added by relay agents in Relay-Forward messages, and options other than the client message in Relay-Reply messages sent by DHCP servers, are not protected. Such communications are already protected using the mechanism described in [I-D.ietf-dhc-relay-server-security].

This extension introduces two new DHCPv6 messages: the Encrypted-Query and the Encrypted-Response messages. It defines six new DHCPv6 options: the Algorithm, Certificate, Signature, Increasing-number, Encryption-Key-Tag option and Encrypted-message options.

2. Requirements Language

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] when they appear in ALL CAPS. When these words are not in ALL CAPS (such as "should" or "Should"), they have their usual English meanings, and are not to be interpreted as [RFC2119] key words.

3. Terminology

This section defines terminology specific to secure DHCPv6 used in this document.

secure DHCPv6 client: A node that initiates a DHCPv6 request on a link to obtain DHCPv6 configuration parameters from

one or more DHCPv6 servers using the encryption and optional authentication mechanisms defined in this document.

secure DHCPv6 server: A DHCPv6 server that implements the authentication and encryption mechanisms defined in this document, and is configured to use them.

4. Security Issues of DHCPv6

[RFC3315] defines an authentication mechanism with integrity protection. This mechanism uses a symmetric key that is shared by the client and server for authentication. It does not provide any key distribution mechanism.

For this approach, operators can set up a key database for both servers and clients from which the client obtains a key before running DHCPv6. However, manual key distribution runs counter to the goal of minimizing the configuration data needed at each host. Consequently, there are no known deployments of this security mechanism.

[RFC3315] provides an additional mechanism for preventing off-network timing attacks using the Reconfigure message: the Reconfigure Key authentication method. However, this method protects only the Reconfigure message. The key is transmitted in plaintext to the client in earlier exchanges and so this method is vulnerable to on-path active attacks.

Anonymity Profile for DHCP Clients [RFC7844] explains how to generate DHCPv4 or DHCPv6 requests that minimize the disclosure of identifying information. However, the anonymity profile limits the use of the certain options. It also cannot anticipate new options that may contain private information. In addition, the anonymity profile does not work in cases where the client wants to maintain anonymity from eavesdroppers but must identify itself to the DHCP server with which it intends to communicate.

Privacy consideration for DHCPv6 [RFC7824] presents an analysis of the privacy issues associated with the use of DHCPv6 by Internet users. No solutions are presented.

Current DHCPv6 messages are still transmitted in cleartext and the privacy information within the DHCPv6 message is not protected from passive attack, such as pervasive monitoring [RFC7258]. The privacy information of the IPv6 host, such as DUID, may be gleaned to find location information, previous visited networks and so on. [RFC7258]

claims that pervasive monitoring should be mitigated in the design of IETF protocol, where possible.

To better address the problem of passive monitoring and to achieve authentication without requiring a symmetric key distribution solution for DHCP, this document defines an asymmetric key authentication and encryption mechanism. This protects against both active attacks, such as spoofing, and passive attacks, such as pervasive monitoring.

5. Secure DHCPv6 Overview

5.1. Solution Overview

The following figure illustrates the secure DHCPv6 procedure. Briefly, this extension establishes the server's identity with an anonymous Information-Request exchange. Once the server's identity has been established, the client may either choose to communicate with the server or not. Not communicating with an unknown server avoids revealing private information, but if there is no known server on a particular link, the client will be unable to communicate with a DHCP server.

If the client chooses to communicate with the selected server(s), it uses the Encrypted-Query message to encapsulate its communications to the DHCP server. The server responds with Encrypted-Response messages. Normal DHCP messages are encapsulated in these two new messages using the new defined Encrypted-message option. Besides the Encrypted-message option, the Signature option is defined to verify the integrity of the DHCPv6 messages and then authentication of the client and the server. The Increasing number option is defined to detect a replay attack.

```
     +-------------+                           +-------------+
     |DHCPv6 Client|                           |DHCPv6 Server|
     +-------------+                           +-------------+
            |            Information-request          |
            |---------------------------------------->|
            |             Algorithm option            |
            |           Option Request option         |
            |                                         |
            |                    Reply                |
            |<----------------------------------------|
            |             Certificate option          |
            |             Signature option            |
            |          Increasing-number option       |
            |         Server Identifier option        |
            |                                         |
            |            Encryption-Query             |
            |---------------------------------------->|
            |          Encrypted-message option       |
            |          Server Identifier option       |
            |         Encryption-Key-Tag option       |
            |                                         |
            |            Encryption-Response          |
            |<----------------------------------------|
            |          Encrypted-message option       |
            |                                         |
```

Figure 1: Secure DHCPv6 Procedure

5.2. New Components

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

o Servers and clients that use certificates first generate a public/private key pair and then obtain a certificate that signs the public key. The Certificate option is defined to carry the certificate of the sender.

o The algorithm option is defined to carry the algorithms lists for algorithm agility.

o The signature is generated using the private key to verify the integrity of the DHCPv6 messages. The Signature option is defined to carry the signature.

o The increasing number is used to detect replayed packet. The Increasing-number option is defined to carry a strictly-increasing serial number.

- o The encryption key Tag is calculated from the public key data. The Encryption-Key-Tag option is defined to identify the used public/private key pair.

- o The Encrypted-message option is defined to contain the encrypted DHCPv6 message.

- o The Encrypted-Query message is sent from the secure DHCPv6 client to the secure DHCPv6 server. The Encrypted-Query message MUST contain the Encrypted-message option and Encryption-Key-Tag option. In addition, the Server Identifier option MUST be included if it is contained in the original DHCPv6 message. The Encrypted-Query message MUST NOT contain any other options.

- o The Encrypted-Response message is sent from the secure DHCPv6 server to the secure DHCPv6 client. The Encrypted-Response message MUST contain the Encrypted-message option. The Encrypted-Response message MUST NOT contain any other options.

5.3. Support for Algorithm Agility

In order to provide a means of addressing problems that may emerge with existing hash algorithms, signature algorithm and encryption algorithms in the future, this document provides a mechanism to support algorithm agility. The support for algorithm agility in this document is mainly a algorithm notification mechanism between the client and the server. The same client and server MUST use the same algorithm in a single communication session. The client can offer a set of algorithms, and then the server selects one algorithm for the future communication.

5.4. Impact on RFC3315

For secure DHCPv6, the Solicit and Rebind messages can be sent only to the selected server(s) which share one common certificate. If the client doesn't like the received Advertise(s) it could restart the whole process and selects another certificate, but it will be more expensive, and there's no guarantee that other servers can provide better Advertise(s).

[RFC3315] provides an additional mechanism for preventing off-network timing attacks using the Reconfigure message: the Reconfigure Key authentication method. Secure DHCPv6 can protect the Reconfigure message using the encryption method. So the Reconfigure Key authentication method SHOULD NOT be used if Secure DHCPv6 is applied.

5.5. Applicability

In principle, secure DHCPv6 is applicable in any environment where physical security on the link is not assured and attacks on DHCPv6 are a concern. In practice, however, authenticated and encrypted DHCPv6 configuration will rely on some operational assumptions mainly regarding public key distribution and management. In order to achieve the wider use of secure DHCPv6, opportunistic security [RFC7435] can be applied to secure DHCPv6 deployment, which allows DHCPv6 encryption in environments where support for authentication or a key distribution mechanism is not available.

Secure DHCPv6 can achieve authentication and encryption based on pre-sharing of authorized certificates. One feasible environment in an early deployment stage would be enterprise networks. In enterprise networks, the client is manually pre-configured with the trusted servers' public key and the server can also be manually pre-configured with the trusted clients' public keys. In some scenario, such as coffee shop where the certificate cannot be validated and one wants access to the Internet, then the DHCPv6 configuration process can be encrypted without authentication.

Note that this deployment scenario based on manual operation is not much different from the existing, shared-secret based authentication mechanisms defined in [RFC3315] in terms of operational costs. However, Secure DHCPv6 is still securer than the shared-secret mechanism in that even if clients' keys stored for the server are stolen that does not mean an immediate threat as these are public keys. In addition, if some kind of Public Key Infrastructure (PKI) is used with Secure DHCPv6, even if the initial installation of the certificates is done manually, it will help reduce operational costs of revocation in case a private key (especially that of the server) is compromised.

6. DHCPv6 Client Behavior

The secure DHCPv6 client is pre-configured with a certificate and its corresponding private key for client authentication. If the client does not obtain a certificate from Certificate Authority (CA), it can generate the self-signed certificate.

The secure DHCPv6 client sends an Information-request message as per [RFC3315]. The Information-request message is used by the DHCPv6 client to request the server's certificate information without having addresses, prefixes or any non-security options assigned to it. The contained Option Request option MUST carry the option code of the Certificate option. In addition, the contained Algorithm option MUST be constructed as explained in Section 10.1.1. The Information-

request message MUST NOT include any other DHCPv6 options except the above options to minimize the client's privacy information leakage.

When receiving the Reply messages from the DHCPv6 servers, a secure DHCPv6 client discards any DHCPv6 message that meets any of the following conditions:

- the Signature option is missing,
- multiple Signature options are present,
- the Certificate option is missing.

And then the client first checks acknowledged hash, signature and encryption algorithms that the server supports. The client checks the signature/encryption algorithms through the certificate option and checks the signature/hash algorithms through the signature option. The SA-id in the certificate option must be equal to the SA-id in the signature option. If they are different, then the client drops the Reply message. The client uses the acknowledged algorithms in the subsequent messages.

Then the client checks the authority of the server. In some scenario where non-authenticated encryption can be accepted, such as coffee shop, then authentication is optional and can be skipped. For the certificate check method, the client validates the certificates through the pre-configured local trusted certificates list or other methods. A certificate that finds a match in the local trust certificates list is treated as verified. If the certificate check fails, the Reply message is dropped.

The client MUST now authenticate the server by verifying the signature and checking increasing number, if there is a Increasing-number option. The order of two procedures is left as an implementation decision. It is RECOMMENDED to check increasing number first, because signature verification is much more computationally expensive. The client checks the Increasing-number option according to the rule defined in Section 9.1. For the message without an Increasing-number option, according to the client's local policy, it MAY be acceptable or rejected. The Signature field verification MUST show that the signature has been calculated as specified in Section 10.1.3. Only the messages that get through both the signature verification and increasing number check (if there is a Increasing-number option) are accepted. Reply message that does not pass the above tests MUST be discarded.

If there are multiple authenticated DHCPv6 certs, the client selects one DHCPv6 cert for the following communication. The selected

certificate may correspond to multiple DHCPv6 servers. If there are no authenticated DHCPv6 certs or existing servers fail authentication, the client should retry a number of times. The client conducts the server discovery process as per section 18.1.5 of [RFC3315] to avoid a packet storm. In this way, it is difficult for a rogue server to beat out a busy "real" server. And then the client takes some alternative action depending on its local policy, such as attempting to use an unsecured DHCPv6 server.

Once the server has been authenticated, the DHCPv6 client sends the Encrypted-Query message to the DHCPv6 server. The Encrypted-Query message contains the Encrypted-message option, which MUST be constructed as explained in Section 10.1.6. The Encrypted-message option contains the encrypted DHCPv6 message using the public key contained in the selected cert. In addition, the Server Identifier option MUST be included if it is in the original message (i.e. Request, Renew, Decline, Release) to avoid the need for other servers receiving the message to attempt to decrypt it. The Encrypted-Query message MUST include the Encryption-Key-Tag option to identify the used public/private key pair, which is constructed as explained in Section 10.1.5. The Encrypted-Query message MUST NOT contain any other DHCPv6 option except the Server Identifier option, Encryption-Key-Tag option, Encrypted-Message option.

The first DHCPv6 message sent from the client to the server, such as Solicit message, MUST contain the related information for client authentication. The encryption text SHOULD be formatted as explain in [RFC5652]. The Certificate option MUST be constructed as explained in Section 10.1.2. In addition, one and only one Signature option MUST be contained, which MUST be constructed as explained in Section 10.1.3. One and only one Increasing-number option SHOULD be contained, which MUST be constructed as explained in Section 10.1.4. In addition, the subsequent encrypted DHCPv6 message sent from the client can also contain the Increasing-number option to defend against replay attack.

For the received Encrypted-Response message, the client MUST drop the Encrypted-Response message if other DHCPv6 option except Encrypted-message option is contained. If the transaction-id is 0, the client also try to decrypt it. Then, the client extracts the Encrypted-message option and decrypts it using its private key to obtain the original DHCPv6 message. In this document, it is assumed that the client will not have multiple DHCPv6 sessions with different DHCPv6 servers using different key pairs and only one key pair is used for the encrypted DHCPv6 configuration process. After the decryption, it handles the message as per [RFC3315].If the decrypted DHCPv6 message contains the Increasing-number option, the DHCPv6 client checks it according to the rule defined in Section 9.1.

If the client fails to get the proper parameters from the chosen server(s), it can select another authenticated certificate and send the Encrypted-Query message to another authenticated server(s) for parameters configuration until the client obtains the proper parameters.

When the decrypted message is Reply message with an error status code, the error status code indicates the failure reason on the server side. According to the received status code, the client MAY take follow-up action:

- o Upon receiving an AuthenticationFail error status code, the client is not able to build up the secure communication with the server. However, there may be other DHCPv6 servers available that successfully complete authentication. The client MAY use the AuthenticationFail as a hint and switch to other DHCPv6 server if it has another one. The client SHOULD retry with another authenticated certificate. However, if the client decides to retransmit using the same certificate after receiving AuthenticationFail, it MUST NOT retransmit immediately and MUST follow normal retransmission routines defined in [RFC3315].

- o Upon receiving a ReplayDetected error status code, the client MAY resend the message with an adjusted Increasing-number option according to the returned number from the DHCPv6 server.

- o Upon receiving a SignatureFail error status code, the client MAY resend the message following normal retransmission routines defined in [RFC3315].

7. DHCPv6 Server Behavior

The secure DHCPv6 server is pre-configured with a certificate and its corresponding private key for server authentication. If the server does not obtain the certificate from Certificate Authority (CA), it can generate the self-signed certificate.

When the DHCPv6 server receives the Information-request message and the contained Option Request option identifies the request is for the server's certificate information, it SHOULD first check the hash, signature, encryption algorithms sets that the client supports. The server selects one hash, signature, encryption algorithm from the acknowledged algorithms sets for the future communication. And then, the server replies with a Reply message to the client. The Reply message MUST contain the requested Certificate option, which MUST be constructed as explained in Section 10.1.2, and Server Identifier option. In addition, the Reply message MUST contain one and only one Signature option, which MUST be constructed as explained in

Section 10.1.3. Besides, the Reply message SHOULD contain one and only one Increasing-number option, which MUST be constructed as explained in Section 10.1.4.

Upon the receipt of Encrypted-Query message, the server MUST drop the message if the other DHCPv6 option is contained except Server Identifier option, Encryption-Key-Tag option, Encrypted-message option. Then, the server checks the Server Identifier option. The DHCPv6 server drops the message that is not for it, thus not paying cost to decrypt messages. If it is the target server, according to the Encryption-Key-Tag option, the server identifies the used public/private key pair and decrypts the Encrypted-message option using the corresponding private key. It is essential to note that the encryption key tag is not a unique identifier. It is theoretically possible for two different public keys to share one common encryption key tag. The encryption key tag is used to limit the possible candidate keys, but it does not uniquely identify a public/private key pair. The server MUST try all corresponding key pairs. If the server cannot find the corresponding private key of the key tag or the corresponding private key of the key tag is invalid for decryption, then the server drops the received message.

If secure DHCPv6 server needs client authentication and decrypted message is a Solicit/Information-request message which contains the information for client authentication, the secure DHCPv6 server discards the received message that meets any of the following conditions:

o the Signature option is missing,

o multiple Signature options are present,

o the Certificate option is missing.

For the signature failure, the server SHOULD send an encrypted Reply message with an UnspecFail (value 1, [RFC3315]) error status code to the client.

The server validates the client's certificate through the local pre-configured trusted certificates list. A certificate that finds a match in the local trust certificates list is treated as verified. If the server does not know the certificate and can accept the non-authenticated encryption, then the server skips the authentication process and uses it for encryption only. The message that fails authentication validation MUST be dropped. In such failure, the DHCPv6 server replies with an encrypted Reply message with an AuthenticationFail error status code, defined in Section 10.3, back

to the client. At this point, the server has either recognized the authentication of the client, or decided to drop the message.

If the decrypted message contains the Increasing-number option, the server checks it according to the rule defined in Section 9.1. If the check fails, an encrypted Reply message with a ReplayDetected error status code, defined in Section 10.3, should be sent back to the client. In the Reply message, a Increasing-number option is carried to indicate the server's stored number for the client to use. According to the server's local policy, the message without an Increasing-number option MAY be acceptable or rejected.

The Signature field verification MUST show that the signature has been calculated as specified in Section 10.1.3. If the signature check fails, the DHCPv6 server SHOULD send an encrypted Reply message with a SignatureFail error status code. Only the clients that get through both the signature verification and increasing number check (if there is a Increasing-number option) are accepted as authenticated clients and continue to be handled their message as defined in [RFC3315].

Once the client has been authenticated, the DHCPv6 server sends the Encrypted-response message to the DHCPv6 client. If the DHCPv6 message is Reconfigure message, then the server set the transaction-id of the Encrypted-Response message to 0. The Encrypted-response message MUST only contain the Encrypted-message option, which MUST be constructed as explained in Section 10.1.6. The encryption text SHOULD be formatted as explain in [RFC5652]. The Encrypted-message option contains the encrypted DHCPv6 message that is encrypted using the authenticated client's public key. To provide the replay protection, the Increasing-number option SHOULD be contained in the encrypted DHCPv6 message.

8. Relay Agent Behavior

When a DHCPv6 relay agent receives an Encrypted-query or Encrypted-response message, it may not recognize this message. The unknown messages MUST be forwarded as described in [RFC7283].

When a DHCPv6 relay agent recognizes the Encrypted-query and Encrypted-response messages, it forwards the message according to section 20 of [RFC3315]. There is nothing more the relay agents have to do, it neither needs to verify the messages from client or server, nor add any secure DHCPv6 options. Actually, by definition in this document, relay agents MUST NOT add any secure DHCPv6 options.

Relay-forward and Relay-reply messages MUST NOT contain any additional Certificate option or Increasing-number option, aside from

those present in the innermost encapsulated messages from the client or server.

9. Processing Rules

9.1. Increasing Number Check

In order to check the Increasing-number option, defined in Section 10.1.4, the client/server has one stable stored number for replay attack detection. The server should keep a record of the increasing number forever. And the client keeps a record of the increasing number during the DHCPv6 configuration process with the DHCPv6 server. And the client can forget the increasing number information after the transaction is finished. The client's initial locally stored increasing number is set to zero.

It is essential to remember that the increasing number is finite. All arithmetic dealing with sequence numbers must be performed modulo 2^64. This unsigned arithmetic preserves the relationship of sequence numbers as they cycle from 2^64 - 1 to 0 again.

In order to check the Increasing-number option, the following comparison is needed.

NUM.STO = the stored number in the client/server

NUM.REC = the acknowledged number from the received message

The Increasing-number option in the received message passes the increasing number check if NUM.REC is more than NUM.STO. And then, the value of NUM.STO is changed into the value of NUM.REC.

The increasing number check fails if NUM.REC is equal with or less than NUM.STO.

9.2. Encryption Key Tag Calculation

The generation method of the encryption key tag adopts the method define in Appendix B in [RFC4034].

The following reference implementation calculates the value of the encryption key tag. The input is the data of the public key. The code is written for clarity not efficiency.

```
/*
    * First octet of the key tag is the most significant 8 bits of the
    * return value;
    * Second octet of the key tag is the least significant 8 bits of the
    * return value.
*/

   unsigned int
   keytag (
           unsigned char key[],  /* the RDATA part of the DNSKEY RR */
           unsigned int keysize  /* the RDLENGTH */
          )
   {
           unsigned long ac;     /* assumed to be 32 bits or larger */
           int i;                /* loop index */

           for ( ac = 0, i = 0; i < keysize; ++i )
                   ac += (i & 1) ? key[i] : key[i] << 8;
           ac += (ac >> 16) & 0xFFFF;
           return ac & 0xFFFF;
   }
```

10. Extensions for Secure DHCPv6

This section describes the extensions to DHCPv6. Six new DHCPv6 options, two new DHCPv6 messages and six new status codes are defined.

10.1. New DHCPv6 Options

10.1.1. Algorithm Option

The Algorithm option carries the algorithms sets for algorithm agility, which is contained in the Information-request message.

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |      OPTION_ALGORITHM         |          option-len           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   .                           EA-id List                          .
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   .                           SHA-id List                         .
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: Algorithm Option

o option-code: OPTION_ALGORITHM (TBA1).

o option-len: length of EA-id List + length of SHA-id List in octets.

o EA-id: The format of the EA-id List field is shown in Figure 3.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           EA-len              |             EA-id             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                              ...                              .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               EA-id           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

EA-len The length of the following EA-ids.

EA-id 2-octets value to indicate the Encryption Algorithm id. The client enumerates the list of encryption algorithms it supports to the server. The encryption algorithm is used for the encrypted DHCPv6 configuration process. This design is adopted in order to provide encryption algorithm agility. The value is from the Encryption Algorithm for Secure DHCPv6 registry in IANA. A registry of the initial assigned values is defined in Section 12. The RSA algorithm, as the mandatory encryption algorithm, MUST be included.

Figure 3: EA-id List Field

o SHA-id List: The format of the SHA-id List field is shown in Figure 4. The SHA-id List contains multiple pair of (SA-id, HA-id). Each pair of (SA-id[i], HA-id[i]) is considered to specify a specific signature method.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           SHA-len             |               SA-id[1]        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           HA-id[1]            |               SA-id[2]        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           HA-id[2]            |               ...             .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                              ...                              .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           SA-id[n]            |                HA-id[n]       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

SHA-len The length of the following SA-id and HA-id pairs.

SA-id 2-octets value to indicate the Signature Algorithm id. The client enumerates the list of signature algorithms it supports to the server. This design is adopted in order to provide signature algorithm agility. The value is from the Signature Algorithm for Secure DHCPv6 registry in IANA. The support of RSASSA-PKCS1-v1_5 is mandatory. A registry of the initial assigned values is defined in Section 12. The mandatory signature algorithms MUST be included.

HA-id 2-octets value to indicate the Hash Algorithm id. The client enumerates the list of hash algorithms it supports to the server. This design is adopted in order to provide hash algorithm agility. The value is from the Hash Algorithm for Secure DHCPv6 registry in IANA. The support of SHA-256 is mandatory. A registry of the initial assigned values is defined in Section 12. The mandatory hash algorithms MUST be included.

Figure 4: SHA-id List Field

10.1.2. Certificate Option

The Certificate option carries the certificate of the client/server, which is contained in the Reply message. The format of the Certificate option is described as follows:

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |      OPTION_CERTIFICATE       |          option-len           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |              EA-id            |            SA-id              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    .                           Certificate                         .
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: Certificate Option

- option-code: OPTION_CERTIFICATE (TBA2).
- option-len: 4 + length of Certificate in octets.
- EA-id: Encryption Algorithm id which is used for the certificate. If the value of the EA-id is 0, then the public key in the certificate is not used for encryption calculation.
- SA-id: Signature Algorithm id which is used for the certificate. If the value of the EA-id is 0, then the public key in the certificate is not used for signature calculation.
- Certificate: A variable-length field containing certificates. The encoding of certificate and certificate data MUST be in format as defined in Section 3.6, [RFC7296]. The support of X.509 certificate is mandatory.

It should be noticed that the scenario where the values of EA-id and SA-id are both 0 makes no sense and the client MUST discard a message with such values.

10.1.3. Signature option

The Signature option contains a signature that is signed by the private key to be attached to the Reply message. The Signature option could be in any place within the DHCPv6 message while it is logically created after the entire DHCPv6 header and options. It protects the entire DHCPv6 header and options, including itself. The format of the Signature option is described as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     OPTION_SIGNATURE          |         option-len            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         SA-id                 |           HA-id               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.                    Signature (variable length)                .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: Signature Option

- option-code: OPTION_SIGNATURE (TBA3).

- option-len: 4 + length of Signature field in octets.

- SA-id: Signature Algorithm id. The signature algorithm is used for computing the signature result. This design is adopted in order to provide signature algorithm agility. The value is from the Signature Algorithm for Secure DHCPv6 registry in IANA. The support of RSASSA-PKCS1-v1_5 is mandatory. A registry of the initial assigned values is defined in Section 12.

- 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. The value is from the Hash Algorithm for Secure DHCPv6 registry in IANA. The support of SHA-256 is mandatory. A registry of the initial assigned values is defined in Section 12.

- 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 field MUST be padded, with all 0, to the next octet boundary if its size is not a multiple of 8 bits. The padding length depends on the signature algorithm, which is indicated in the SA-id field.

Note: If Secure DHCPv6 is used, the DHCPv6 message is encrypted in a way that the authentication mechanism defined in RFC3315 does not understand. So the Authentication option SHOULD NOT be used if Secure DHCPv6 is applied.

10.1.4. Increasing-number Option

The Increasing-number option carries the strictly increasing number for anti-replay protection, which is contained in the Reply message and the encrypted DHCPv6 message. It is optional.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   OPTION_INCREASING_NUM       |         option-len            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                  Increasing-Num (64-bit)                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

option-code    OPTION_INCREASING_NUM (TBA4).

option-len     8, in octets.

Increasing-Num A strictly increasing number for the replay attack detection
               which is more than the local stored number.
```

Figure 7: Increasing-number Option

10.1.5. Encryption-Key-Tag Option

The Encryption-Key-Tag option carries the key identifier which is calculated from the public key data. The Encrypted-Query message MUST contain the Encryption-Key-Tag option to identify the used public/private key pair.

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |    OPTION_ENCRYPTION_KEY_TAG  |           option-len          |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |  encryption key tag(16-bit)   |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 8: Encryption-Key-Tag Option

option-code OPTION_ENCRYPTION_KEY_TAG (TBA5).

option-len 2, in octets.

encryption key tag A 16 bits field containing the encryption key tag sent from the client to server to identify the used public/private key pair. The encryption key tag is calculated from the public

key data, like fingerprint of a specific public key. The specific calculation method of the encryption key tag is illustrated in Section 9.2.

10.1.6. Encrypted-message Option

The Encrypted-message option carries the encrypted DHCPv6 message, which is calculated with the recipient's public key. The Encrypted-message option is contained in the Encrypted-Query message or the Encrypted-Response message.

The format of the Encrypted-message option is:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          option-code          |           option-len          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.                  encrypted DHCPv6 message                     .
.                       (variable)                              .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 9: Encrypted-message Option

option-code OPTION_ENCRYPTED_MSG (TBA6).

option-len Length of the encrypted DHCPv6 message in octets.

encrypted DHCPv6 message A variable length field containing the encrypted DHCPv6 message. In Encrypted-Query message, it contains encrypted DHCPv6 message sent from a client to server. In Encrypted-response message, it contains encrypted DHCPv6 message sent from a server to client.

10.2. New DHCPv6 Messages

Two new DHCPv6 messages are defined to achieve the DHCPv6 encryption: Encrypted-Query and Encrypted-Response. Both the DHCPv6 messages defined in this document share the following format:

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |    msg-type   |               transaction-id                  |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    .                            options                            .
    .                          (variable)                           .
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 10: The format of Encrypted-Query and Encrypted-Response Messages

msg-type: Identifier of the message type. It can be either Encrypted-Query (TBA7) or DHCPv6-Response (TBA8).

transaction-id: The transaction ID for this message exchange.

options: The Encrypted-Query message MUST contain the Encrypted-message option, Encryption-Key-Tag option and Server Identifier option if the message in the Encrypted-message option has a Server Identifier option. The Encrypted-Response message MUST only contain the Encrypted-message option.

10.3. Status Codes

The following new status codes, see Section 5.4 of [RFC3315] are defined.

- AuthenticationFail (TBD9): indicates that the message from the DHCPv6 client fails authentication check.

- ReplayDetected (TBD10): indicates the message from DHCPv6 client fails the increasing number check.

- SignatureFail (TBD11): indicates the message from DHCPv6 client fails the signature check.

11. Security Considerations

This document provides the authentication and encryption mechanisms for DHCPv6.

There are some mandatory algorithm for encryption algorithm in this document. It may be at some point that the mandatory algorithm is no longer safe to use.

A server or a client, whose local policy accepts messages without a Increasing-number option, may have to face the risk of replay attacks.

Since the algorithm option isn't protected by a signature, the list can be forged without detection, which can lead to a downgrade attack.

Likewise, since the Encryption-Key-Tag Option isn't protected, an attacker that can intercept the message can forge the value without detection.

If the client tries more than one cert for client authentication, the server can easily get a client that implements this to enumerate its entire cert list and probably learn a lot about a client that way. For this security item, It is RECOMMENDED that client certificates could be tied to specific server certificates by configuration.

12. IANA Considerations

This document defines six new DHCPv6 [RFC3315] options. The IANA is requested to assign values for these six options from the DHCPv6 Option Codes table of the DHCPv6 Parameters registry maintained in http://www.iana.org/assignments/dhcpv6-parameters. The six options are:

The Algorithm Option (TBA1), described in Section 10.1.2.

The Certificate Option (TBA2), described in Section 10.1.2.

The Signature Option (TBA3), described in Section 10.1.3.

The Increasing-number Option (TBA4),described in Section 10.1.4.

The Encryption-Key-Tag Option (TBA5),described in Section 10.1.5.

The Encrypted-message Option (TBA6), described in Section 10.1.6.

The IANA is also requested to assign value for these two messages from the DHCPv6 Message Types table of the DHCPv6 Parameters registry maintained in http://www.iana.org/assignments/dhcpv6-parameters. The two messages are:

The Encrypted-Query Message (TBA7), described in Section 10.2.

The Encrypted-Response Message (TBA8), described in Section 10.2.

The IANA is also requested to add three new registry tables to the DHCPv6 Parameters registry maintained in http://www.iana.org/assignments/dhcpv6-parameters. The three tables are the Hash Algorithm for Secure DHCPv6 table, the Signature Algorithm for Secure DHCPv6 table and the Encryption Algorithm for Secure DHCPv6 table.

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 for Secure DHCPv6. The values in this table are 16-bit unsigned integers. The following initial values are assigned for Hash Algorithm for Secure DHCPv6 in this document:

Name	Value	RFCs
SHA-256	0x01	this document
SHA-512	0x02	this document

Signature Algorithm for Secure DHCPv6. The values in this table are 16-bit unsigned integers. The following initial values are assigned for Signature Algorithm for Secure DHCPv6 in this document:

Name	Value	RFCs
Non-SigAlg	0x00	this document
RSASSA-PKCS1-v1_5	0x01	this document

Encryption algorithm for Secure DHCPv6. The values in this table are 16-bit unsigned integers. The following initial values are assigned for encryption algorithm for Secure DHCPv6 in this document:

Name	Value	RFCs
Non-EncryAlg	0x00	this document
RSA	0x01	this document

IANA is requested to assign the following new DHCPv6 Status Codes, defined in Section 10.3, in the DHCPv6 Parameters registry maintained in http://www.iana.org/assignments/dhcpv6-parameters:

Code	Name	Reference
TBD9	AuthenticationFail	this document
TBD10	ReplayDetected	this document
TBD11	SignatureFail	this document

13. Acknowledgements

The authors would like to thank Tomek Mrugalski, Bernie Volz, Jianping Wu, Randy Bush, Yiu Lee, Sean Shen, Ralph Droms, Jari Arkko, Sean Turner, Stephen Farrell, Christian Huitema, Stephen Kent, Thomas Huth, David Schumacher, Francis Dupont, Gang Chen, Suresh Krishnan, Fred Templin, Robert Elz, Nico Williams, Erik Kline, Alan DeKok, Bernard Aboba, Sam Hartman, Zilong Liu and other members of the IETF DHC working group for their valuable comments.

This document was produced using the xml2rfc tool [RFC2629].

14. Change log [RFC Editor: Please remove]

draft-ietf-dhc-sedhcpv6-21: Add the reference of draft-ietf-dhc-relay -server-security. Change the SA-ID List as SHA-ID List and delete the HA-id List. The SHA-id List contains the SA-id and HA-id pairs. Add some statements about the Reconfigure message process. Add some specific text on the encryption key tag calculation method; Add more text on security consideration; Changes some mistakes and grammar mistakes

draft-ietf-dhc-sedhcpv6-20: Correct a few grammar mistakes.

draft-ietf-dhc-sedhcpv6-19: In client behavior part, we adds some description about opportunistic security. In this way, in some scenario, authentication is optional. Add the reference of RFC 4034 for the encryption key tag calculation. Delete the part that the relay agent cache server announcements part. Add the assumption that the client's initial stored increasing number is set to zero. In this way, for the first time increasing number check in the Reply message, the check will always succeed, and then the locally stored number is changed into the contained number in the Reply message. Correct many grammar mistakes.

draft-ietf-dhc-sedhcpv6-18: Add the Algorithm option. The algorithm option contains the EA-id List, SA-id List, HA-id List, and then the certificate and signature options do not contain the algorithm list; Add the Encryption Key Tag option to identify the used public/private key pair; Delete the AlgorithmNotSupported error status code; Delete some description on that secure DHCPv6 exchanges the server selection method; Delete the DecryptionFail error status code; For the case where the client's certificate is missed, then the server discards the received message. Add the assumption that: For DHCPv6 client, just one certificate is used for the DHCPv6 configuration. Add the statement that: For the first Encrypted-Query message, the server needs to try all the possible private keys and then records the relationship between the public key and the encryption key tag.

draft-ietf-dhc-sedhcpv6-17: Change the format of the certificate option according to the comments from Bernie.

draft-ietf-dhc-sedhcpv6-16: For the algorithm agility part, the provider can offer multiple EA-id, SA-id, HA-id and then receiver choose one from the algorithm set.

draft-ietf-dhc-sedhcpv6-15: Increasing number option only contains the strictly increasing number; Add some description about why encryption is needed in Security Issues of DHCPv6 part;

draft-ietf-dhc-sedhcpv6-14: For the deployment part, Tofu is out of scope and take Opportunistic security into consideration; Increasing number option is changed into 64 bits; Increasing number check is a separate section; IncreasingnumFail error status code is changed into ReplayDetected error status code; Add the section of "caused change to RFC3315";

draft-ietf-dhc-sedhcpv6-13: Change the Timestamp option into Increasing-number option and the corresponding check method; Delete the OCSP stampling part for the certificate check; Add the scenario where the hash and signature algorithms cannot be separated; Add the comparison with RFC7824 and RFC7844; Add the encryption text format and reference of RFC5652. Add the consideration of scenario where multiple DHCPv6 servers share one common DHCPv6 server. Add the statement that Encrypted-Query and Encrypted-Response messages can only contain certain options: Server Identifier option and Encrypted-message option. Add opportunistic security for deployment consideration. Besides authentication+encyrption mode, encryption-only mode is added.

draft-ietf-dhc-sedhcpv6-12: Add the Signature option and timestamp option during server/client authentication process. Add the hash function and signature algorithm. Add the requirement: The Information-request message cannot contain any other options except ORO option. Modify the use of "SHOULD"; Delete the reference of RFC5280 and modify the method of client/server cert verification; Add the relay agent cache function for the quick response when there is no authenticated server. 2016-4-24.

draft-ietf-dhc-sedhcpv6-11: Delete the Signature option, because the encrypted DHCPv6 message and the Information-request message (only contain the Certificate option) don't need the Signature option for message integrity check; Rewrite the "Applicability" section; Add the encryption algorithm negotiation process; To support the encryption algorithm negotiation, the Certificate option contains the EA-id(encryption algorithm identifier) field; Reserve the Timestamp option to defend against the replay attacks for encrypted DHCPv6

configuration process; Modify the client behavior when there is no authenticated DHCPv6 server; Add the DecryptionFail error code. 2016-3-9.

draft-ietf-dhc-sedhcpv6-10: merge DHCPv6 authentication and DHCPv6 encryption. The public key option is removed, because the device can generate the self-signed certificate if it is pre-configured the public key not the certificate. 2015-12-10.

draft-ietf-dhc-sedhcpv6-09: change some texts about the deployment part.2015-12-10.

draft-ietf-dhc-sedhcpv6-08: clarified what the client and the server should do if it receives a message using unsupported algorithm; refined the error code treatment regarding to AuthenticationFail and TimestampFail; added consideration on how to reduce the DoS attack when using TOFU; other general editorial cleanups. 2015-06-10.

draft-ietf-dhc-sedhcpv6-07: removed the deployment consideration section; instead, described more straightforward use cases with TOFU in the overview section, and clarified how the public keys would be stored at the recipient when TOFU is used. The overview section also clarified the integration of PKI or other similar infrastructure is an open issue. 2015-03-23.

draft-ietf-dhc-sedhcpv6-06: remove the limitation that only clients use PKI- certificates and only servers use public keys. The new text would allow clients use public keys and servers use PKI-certificates. 2015-02-18.

draft-ietf-dhc-sedhcpv6-05: addressed comments from mail list that responsed to the second WGLC. 2014-12-08.

draft-ietf-dhc-sedhcpv6-04: addressed comments from mail list. Making timestamp an independent and optional option. Reduce the serverside authentication to base on only client's certificate. Reduce the clientside authentication to only Leaf of Faith base on server's public key. 2014-09-26.

draft-ietf-dhc-sedhcpv6-03: addressed comments from WGLC. Added a new section "Deployment Consideration". Corrected the Public Key Field in the Public Key Option. Added consideration for large DHCPv6 message transmission. Added TimestampFail error code. Refined the retransmission rules on clients. 2014-06-18.

draft-ietf-dhc-sedhcpv6-02: addressed comments (applicability statement, redesign the error codes and their logic) from IETF89 DHC WG meeting and volunteer reviewers. 2014-04-14.

draft-ietf-dhc-sedhcpv6-01: addressed comments from IETF88 DHC WG meeting. Moved Dacheng Zhang from acknowledgement to be co-author. 2014-02-14.

draft-ietf-dhc-sedhcpv6-00: adopted by DHC WG. 2013-11-19.

draft-jiang-dhc-sedhcpv6-02: removed protection between relay agent and server due to complexity, following the comments from Ted Lemon, Bernie Volz. 2013-10-16.

draft-jiang-dhc-sedhcpv6-01: update according to review comments from Ted Lemon, Bernie Volz, Ralph Droms. Separated Public Key/ Certificate option into two options. Refined many detailed processes. 2013-10-08.

draft-jiang-dhc-sedhcpv6-00: original version, this draft is a replacement of draft-ietf-dhc-secure-dhcpv6, which reached IESG and dead because of consideration regarding to CGA. The authors followed the suggestion from IESG making a general public key based mechanism. 2013-06-29.

15. References

15.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <http://www.rfc-editor.org/info/rfc2460>.

[RFC3315] Droms, R., Ed., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, DOI 10.17487/RFC3315, July 2003, <http://www.rfc-editor.org/info/rfc3315>.

[RFC3971] Arkko, J., Ed., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, DOI 10.17487/RFC3971, March 2005, <http://www.rfc-editor.org/info/rfc3971>.

[RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, DOI 10.17487/RFC4034, March 2005, <http://www.rfc-editor.org/info/rfc4034>.

[RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 4443, DOI 10.17487/RFC4443, March 2006, <http://www.rfc-editor.org/info/rfc4443>.

[RFC5652] Housley, R., "Cryptographic Message Syntax (CMS)", STD 70, RFC 5652, DOI 10.17487/RFC5652, September 2009, <http://www.rfc-editor.org/info/rfc5652>.

[RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, DOI 10.17487/RFC5905, June 2010, <http://www.rfc-editor.org/info/rfc5905>.

[RFC6840] Weiler, S., Ed. and D. Blacka, Ed., "Clarifications and Implementation Notes for DNS Security (DNSSEC)", RFC 6840, DOI 10.17487/RFC6840, February 2013, <http://www.rfc-editor.org/info/rfc6840>.

[RFC7283] Cui, Y., Sun, Q., and T. Lemon, "Handling Unknown DHCPv6 Messages", RFC 7283, DOI 10.17487/RFC7283, July 2014, <http://www.rfc-editor.org/info/rfc7283>.

[RFC7296] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., and T. Kivinen, "Internet Key Exchange Protocol Version 2 (IKEv2)", STD 79, RFC 7296, DOI 10.17487/RFC7296, October 2014, <http://www.rfc-editor.org/info/rfc7296>.

[RFC7435] Dukhovni, V., "Opportunistic Security: Some Protection Most of the Time", RFC 7435, DOI 10.17487/RFC7435, December 2014, <http://www.rfc-editor.org/info/rfc7435>.

[RFC7824] Krishnan, S., Mrugalski, T., and S. Jiang, "Privacy Considerations for DHCPv6", RFC 7824, DOI 10.17487/RFC7824, May 2016, <http://www.rfc-editor.org/info/rfc7824>.

[RFC7844] Huitema, C., Mrugalski, T., and S. Krishnan, "Anonymity Profiles for DHCP Clients", RFC 7844, DOI 10.17487/RFC7844, May 2016, <http://www.rfc-editor.org/info/rfc7844>.

15.2. Informative References

[I-D.ietf-dhc-relay-server-security]
Volz, B. and Y. Pal, "Security of Messages Exchanged Between Servers and Relay Agents", draft-ietf-dhc-relay-server-security-03 (work in progress), February 2017.

[RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, DOI 10.17487/RFC2629, June 1999, <http://www.rfc-editor.org/info/rfc2629>.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, DOI 10.17487/RFC5226, May 2008, <http://www.rfc-editor.org/info/rfc5226>.

[RFC6273] Kukec, A., Krishnan, S., and S. Jiang, "The Secure Neighbor Discovery (SEND) Hash Threat Analysis", RFC 6273, DOI 10.17487/RFC6273, June 2011, <http://www.rfc-editor.org/info/rfc6273>.

[RFC7258] Farrell, S. and H. Tschofenig, "Pervasive Monitoring Is an Attack", BCP 188, RFC 7258, DOI 10.17487/RFC7258, May 2014, <http://www.rfc-editor.org/info/rfc7258>.

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

Authors' Addresses

Lishan Li
Tsinghua University
Beijing 100084
P.R.China

Phone: +86-15201441862
Email: lilishan48@gmail.com

Sheng Jiang
Huawei Technologies Co., Ltd
Q14, Huawei Campus, No.156 Beiqing Road
Hai-Dian District, Beijing, 100095
CN

Email: jiangsheng@huawei.com

Yong Cui
Tsinghua University
Beijing 100084
P.R.China

Phone: +86-10-6260-3059
Email: yong@csnet1.cs.tsinghua.edu.cn

Tatuya Jinmei
Infoblox Inc.
3111 Coronado Drive
Santa Clara, CA
US

Email: jinmei@wide.ad.jp

Ted Lemon
Nominum, Inc.
2000 Seaport Blvd
Redwood City, CA 94063
USA

Phone: +1-650-381-6000
Email: Ted.Lemon@nominum.com

Dacheng Zhang
Beijing
CN

Email: dacheng.zhang@gmail.com