DHC Working Group
Internet-Draft
Intended status: Standards Track
Expires: October 26, 2016

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

April 24, 2016

Secure DHCPv6
draft-ietf-dhc-sedhcpv6-12

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. This document analyzes the security issues of DHCPv6 and specifies the secure DHCPv6 mechanism for authentication and encryption of messages between a DHCPv6 client and a DHCPv6 server.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on October 26, 2016.

Copyright Notice

Table of Contents

1. Introduction

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

This document analyzes the security issues of DHCPv6 and provides the following mechanisms for improving the security of DHCPv6 between the DHCPv6 client and the DHCPv6 server:

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

- the encryption between the DHCPv6 client and the DHCPv6 server in order to protect the DHCPv6 from passive attacks, such as pervasive monitoring.

Note: this secure mechanism in this document does not protect outer options in Relay-Forward and Relay-Reply messages, either added by a relay agent toward a server or added by a server toward a relay agent. Communication between a server and a relay agent, and communications between relay agents, may be secured through the use of IPsec, as described in section 21.1 in [RFC3315].

The security mechanism specified in this document achieves DHCPv6 authentication and encryption based on the sender's certificate. We introduce two new DHCPv6 messages: Encrypted-Query message and Encrypted-Response message and Four new DHCPv6 options: Certificate option, Signature option, Timestamp option and Encrypted-message option for DHCPv6 authentication and encryption. The Certificate option, Signature option, Timestamp option are used for DHCPv6 client/server authentication. The Encryption-Query message, Encryption-Response message and Encrypted-message option are used for DHCPv6 encryption.

2. Requirements Language and 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] 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 the DHCPv6 request on a link to obtain the DHCPv6 configuration parameters from one or more DHCPv6 servers. The configuration process is authenticated and encrypted using the defined mechanisms in this document.

secure DHCPv6 server: A node that responds to requests from clients using the authentication and encryption mechanism defined in this document.

4. Security Issues of DHCPv6

DHCPv6 is a client/server protocol that provides managed configuration of devices. It enables a DHCPv6 server to automatically configure relevant network parameters on clients. The basic DHCPv6 specification [RFC3315] defines security mechanisms, but they have some flaws and can be improved.

The basic DHCPv6 specifications can optionally authenticate the origin of messages and validate the integrity of messages using an authentication option with a symmetric key pair. [RFC3315] relies on pre-established secret keys. For any kind of meaningful security, each DHCPv6 client would need to be configured with its own secret key; [RFC3315] provides no mechanism for doing this.

For the out of band approach, operators can set up a key database for both servers and clients from which the client obtains a key before running DHCPv6. Manual key distribution runs counter to the goal of minimizing the configuration data needed at each host.

[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 active attacks.

In addition, the 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. The IETF has expressed strong agreement that pervasive monitoring is an attack that needs to be mitigated where possible in [RFC7258].

In comparison, the security mechanisms defined in this document provides for authentication and encryption based on the public key certificates of the client and server. The DHCPv6 authentication can protect DHCPv6 from active attacks, such as spoofing attack. And the DHCPv6 encryption defends against passive attacks, such as pervasive monitoring attack.

5. Secure DHCPv6 Overview

5.1. Solution Overview

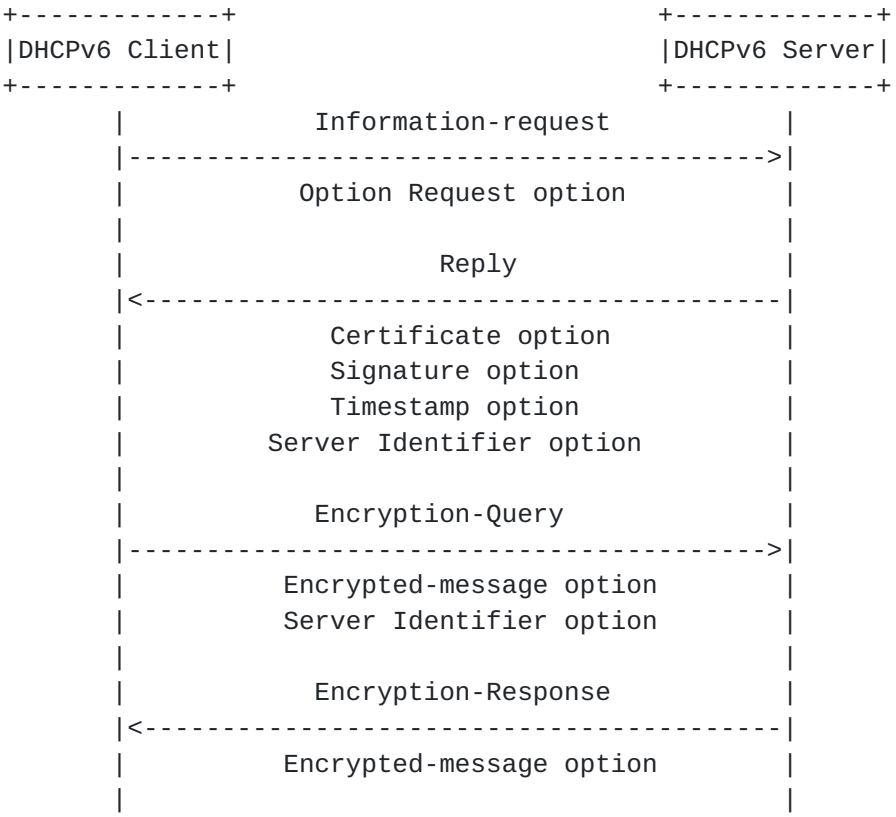
This solution provides authentication and encryption mechanisms based on the certificates of the DHCPv6 client and server. Before the standard DHCPv6 configuration process, the Information-request and Reply messages are exchanged to select the authenticated DHCPv6 server. After mutual authentication between the DHCPv6 client and server, the following DHCPv6 configuration process is encrypted to avoid the privacy information disclosure. We introduce two new DHCPv6 messages: Encrypted-Query message, Encrypted-Response message and four new DHCPv6 options: Encrypted-message option, Certificate option, Signature option, Timestamp option. Based on the new defined messages and options, the corresponding authentication and encryption mechanisms are achieved.

The following figure illustrates secure DHCPv6 procedure. The DHCPv6 client first sends Information-request message as per [RFC3315]. The Information-request message is used to request the servers for the servers' certificates information, without going through any address, prefix or non-security option assignment process. The Information-request contains no DHCPv6 options except ORO option to avoid client's privacy information disclosure. When receiving the Information-request message, the server sends the Reply message that contains the server's Certificate option, Signature option, Timestamp option and Server Identifier option. Upon the receipt of the Reply message, the DHCPv6 client verifies the server's identity according to the contained options in the Reply message. If there are multiple authenticated DHCPv6 server certs, the client selects one authenticated DHCPv6 server for the following DHCPv6 configuration process. If there are no authenticated DHCPv6 server cert or existing server certs fails authentication, the client should retry a number of times. In this way, it is difficult for a rogue server to beat out a busy "real" server. And then the client takes some other alternative action depending on its local policy.

After the server's authentication, the first DHCPv6 message sent from the client to the server, such as Solicit message, contains the client's Certificate information for client authentication. The DHCPv6 client sends the Encrypted-Query message to server, which

carries the Encrypted-message option and the Server Identifier option. The Encrypted-message option contains the encrypted DHCPv6 message sent from the client to the server. When the DHCPv6 server receives the Encrypted-Query message, it decrypts the message using its private key. If the decrypted message contains the client's Certificate option, the DHCPv6 server verifies the client's identity according to the contained client certificate information.

After the client's authentication, the server sends the Encrypted-Response message to the client, which contains the Encrypted-message option. The Encrypted-message option contains the encrypted DHCPv6 message sent from server to client, which is encrypted using the client's public key. If the message fails client authentication, then the server sends the corresponding error status code to the client. During the encrypted DHCPv6 configuration process, the Timestamp option can be contained in the encrypted DHCPv6 messages to defend against replay attacks.



Secure DHCPv6 Procedure

5.2. New Components

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

- 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.

- A signature generated using the private key which is used by the receiver to verify the integrity of the DHCPv6 messages and then authentication of the client/server. Another option is defined to carry the signature.

- A timestamp that can be used to detect replayed packet. The Timestamp option is defined to carry the current time of the client/server. The secure DHCPv6 client/server need to meet some accuracy requirements and be synced to global time, while the timestamp checking mechanism allows a configurable time value for clock drift. The real time provision is out of scope of this document.

- The Encrypted-message option that contains the encrypted DHCPv6 message.

- The Encrypted-Query message that is sent from the secure DHCPv6 client to the secure DHCPv6 server. The Encrypted-Query message contains the Encrypted-message option and Server Identifier option.

- The Encrypted-Response message that is sent from the secure DHCPv6 server to the secure DHCPv6 client. The Encrypted-Response message contains the Encrypted-message option.

5.3. Support for Algorithm Agility

In order to provide a means of addressing problems that may emerge in the future with existing hash algorithms, as recommended in [RFC4270], this document provides a mechanism for negotiating the use of more secure hashes in the future.

In addition to hash algorithm agility, this document also provides a mechanism for signature algorithm and encryption algorithm agility.

The support for algorithm agility in this document is mainly a unilateral notification mechanism from sender to recipient. A recipient MAY support various algorithms simultaneously among

different senders, and the different senders in a same administrative domain may be allowed to use various algorithms simultaneously. It is NOT RECOMMENDED that the same sender and recipient use various algorithms in a single communication session.

If the server does not support the algorithm used by the client, the server SHOULD reply with an AlgorithmNotSupported status code (defined in Section 10.3) to the client. Upon receiving this status code, the client MAY resend the message protected with the mandatory algorithm.

5.4. 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, it will rely on some operational assumptions mainly regarding public key distribution and management, until more lessons are learned and more experiences are achieved.

One feasible environment in an early deployment stage would be enterprise networks. In such networks the security policy tends to be strict and it will be easier to manage client hosts. One trivial deployment scenario is therefore to manually pre-configure client with the trusted servers' public key and manually register clients' public keys for the server. It may also be possible to deploy an internal PKI to make this less reliant on manual operations, although it is currently subject to future study specifically how to integrate such a PKI into the DHCPv6 service for the network.

Note that this deployment scenario based on manual operation is not different very much 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 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.

It is believed that Secure DHCPv6 could be more widely applicable with integration of generic PKI so that it will be more easily deployed. But such a deployment requires more general issues with PKI deployment be addressed, and it is currently unknown whether we can find practical deployment scenarios. It is subject to future study and experiments, and out of scope of this document.

6. DHCPv6 Client Behavior

For the secure DHCPv6 client, a certificate is needed for client authentication. The client is pre-configured with a certificate and its corresponding private key. If the client is pre-configured with public key but not with a certificate, it can generate the self-signed certificate for client authentication.

The secure DHCPv6 client sends Information-request message as per [RFC3315]. The Information-request message is used by the DHCPv6 client to request the server's identity verification information without having addresses, prefixes or any non-security options assigned to it. The Information-request message MUST NOT include any DHCPv6 options except ORO option to minimize client's privacy information leakage. The Option Request option in the Information-request message MUST contain the option code of the Certificate option.

When receiving the Reply messages from DHCPv6 servers, a secure DHCPv6 client discards any DHCPv6 messages that meet 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 the support of the hash function, signature algorithm and encryption algorithm that the server used. If the check fails, the Reply message is dropped. If all the algorithms are supported, the client then checks the authority of this server. The client also uses the same algorithms in the return messages.

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 certificate list is treated as verified. If the client want to check server's certificate to see whether it has been revoked, the OCSP stapling can be used. The message transaction-id is used as the identifier of the authenticated server's public key for encryption. At this point, the client has either recognized the certificate of the server, or decided to drop the message.

The client MUST now authenticate the server by verifying the signature and checking timestamp (see details in Section 9.1), if there is a Timestamp option. The order of two procedures is left as

an implementation decision. It is RECOMMENDED to check timestamp first, because signature verification is much more computationally expensive.

The Signature field verification MUST show that the signature has been calculated as specified in Section 10.1.2. Only the messages that get through both the signature verification and timestamp check (if there is a Timestamp option) are accepted. Reply message that does not pass the above tests MUST be discarded.

If there are multiple authenticated DHCPv6 servers, the client selects one DHCPv6 server for the following network parameters configuration. The client can also choose other implementation method depending on the client's local policy if the defined protocol can also run normally. For example, the client can try multiple transactions (each encrypted with different public key) at the "same" time. If there are no authenticated DHCPv6 servers or existing servers failed authentication, the client should retry a number of times. In this way, it is difficult for the 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. The client conducts the server discovery process as per section 18.1.5 of [RFC3315] to avoid the packet storm.

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 10.1.4, and Server Identifier option. The Encrypted-message option contains the DHCPv6 message that is encrypted using the selected server's public key. The Server Identifier option is externally visible to avoid decryption cost by those unselected servers.

For the encrypted DHCPv6 message sent from the DHCPv6 client to the DHCPv6 server, the first DHCPv6 message, such as Solicit message, MUST contain the Certificate option, Signature option and Timestamp option for client authentication. The Certificate option MUST be constructed as explained in Section 10.1.1. In addition, one and only one Signature option MUST be contained, which MUST be constructed as explained in Section 10.1.2. One and only one Timestamp option SHOULD be contained, which MUST be constructed as explained in Section 10.1.3. The Timestamp field SHOULD be set to the current time, according to sender's real time clock.

If the client has multiple certificates with different public/private key pairs, the message transaction-id is used as the identifier of the client's private key for decryption. In addition, the subsequent

encrypted DHCPv6 message can contain the Timestamp option to defend against replay attack.

For the received Encrypted-Response message, the client extracts the Encrypted-message option and decrypts it using its private key to obtain the original DHCPv6 message. Then it handles the message as per [RFC3315]. If the decrypted DHCPv6 message contains the Timestamp option, the DHCPv6 client checks the timestamp according to the rule defined in Section 9.1. The DHCPv6 message, which fails the timestamp check, MUST be discarded. If the client fails to get the proper parameters from the chosen server, it sends the Encrypted-Query message to another authenticated server for parameters configuration until the client obtains the proper parameters.

When the client receives a 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:

- Upon receiving an AlgorithmNotSupported error status code, the client SHOULD resend the message protected with one of the mandatory algorithms.

- 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 certificate if it has another one; but otherwise treat the message containing the status code as if it had not been received. But it SHOULD NOT retry with the same 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].

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

- Upon receiving a TimestampFail error status code, the client MAY resend the message with an adjusted timestamp according to the returned clock from the DHCPv6 server. The client SHOULD NOT change its own clock, but only compute an offset for the communication session.

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

7. DHCPv6 Server Behavior

For the secure DHCPv6 server, a certificate is needed for server authentication. The server is pre-configured with a certificate and its corresponding private key. If the server is pre-configured with public key but not with a certificate, it can generate the self-signed certificate for server authentication.

When the DHCPv6 server receives the Information-request message and the contained Option Request option identifies the request is for the server certificate information, it 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 10.1.1, 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 10.1.2. Besides, the Reply message SHOULD contain one and only one Timestamp option, which MUST be constructed as explained in Section 10.1.3. The Timestamp field SHOULD be set to the current time, according to server's real time clock.

Upon the receipt of Encrypted-Query message, the server checks the Server Identifier option. It decrypts the Encrypted-message option using its private key if it is the target server. The DHCPv6 server drops the message that is not for it, thus not paying cost to decrypt messages not for it.

If the decrypted message is a Solicit/Information-request message, the secure DHCPv6 server discards the received message that meets any of the following conditions:

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

In such failure, the server replies with an UnspecFail (value 1, [RFC3315]) error status code.

The server SHOULD first check the support of the hash function, signature algorithm, encryption algorithm that the client used. If the check fails, the server SHOULD reply with an AlgorithmNotSupported error status code, defined in Section 10.3, back to the client. If all the algorithms are supported, the server then checks the authority of this client.

The server validates the client's public key through the local pre-configured trusted public keys list. A public key that finds a match

in the local trust public keys list is treated as verified. The message that fails public key validation MUST be dropped. In such failure, the DHCPv6 server replies 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 Timestamp option, the server checks the timestamp according to the rule defined in Section 9.1. If the timestamp check fails, a TimestampFail error status code, defined in Section 10.3, should be sent back to the client. Depending on server's local policy, the message without a Timestamp option MAY be acceptable or rejected. If the server rejects such a message, a TimestampFail error status code should be sent back to the client. The Reply message that carries the TimestampFail error status code carries a Timestamp option, which indicates the server's clock for the client to use.

If the server does not send the Timestamp option, the client ignores the timestamp check and verifies the signature. If there is a timestamp option, the server MUST now authenticate the client by verifying the signature and checking timestamp (see details in Section 9.1). The order of two procedures is left as an implementation decision. It is RECOMMENDED to check timestamp first, because signature verification is much more computationally expensive. Depending on server's local policy, the message without a Timestamp option MAY be acceptable or rejected. If the server rejects such a message, a TimestampFail error status code, defined in Section 10.3, should be sent back to the client. The reply message that carries the TimestampFail error status code SHOULD carry a Timestamp option, which indicates the server's clock for the client to use.

The Signature field verification MUST show that the signature has been calculated as specified in Section 10.1.2. Only the clients that get through both the signature verification and timestamp check (if there is a Timestamp option) are accepted as authenticated clients and continue to be handled their message as defined in [RFC3315]. Clients that do not pass the above tests MUST be treated as unauthenticated clients. The DHCPv6 server SHOULD reply a SignatureFail error status code, defined in Section 10.3, for the signature verification failure; or a TimestampFail error status code, defined in Section 10.3, for the timestamp check failure, back to the client.

Once the client has been authenticated, the DHCPv6 server sends the Encrypted-response message to the DHCPv6 client. The Encrypted-response message contains the Encrypted-message option, which MUST be

constructed as explained in Section 10.1.4. 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 Timestamp option can 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 Timestamp option, aside from those present in the innermost encapsulated messages from the client or server.

Relay agent is RECOMMENDED to cache server announcements to form the list of the available DHCPv6 server certs. If the relay agent receives the Information-request message, then it replies with a list of server certs available locally. In this way, the client can be confident of a quick response, and therefore treat the lack of a quick response as an indication that no authenticated DHCP servers exist.

9. Processing Rules

9.1. Timestamp Check

In order to check the Timestamp option, defined in Section 10.1.3, recipients SHOULD be configured with an allowed timestamp Delta value, a "fuzz factor" for comparisons, and an allowed clock drift parameter. The recommended default value for the allowed Delta is 300 seconds (5 minutes); for fuzz factor 1 second; and for clock drift, 0.01 second.

Note: the Timestamp mechanism is based on the assumption that communication peers have roughly synchronized clocks, within certain allowed clock drift. So, an accurate clock is not necessary. If one has a clock too far from the current time, the timestamp mechanism would not work.

To facilitate timestamp checking, each recipient SHOULD store the following information for each sender, from which at least one accepted secure DHCPv6 message is successfully verified (for timestamp check and signature verification):

- The receive time of the last received and accepted DHCPv6 message. This is called RDlast.

- The timestamp in the last received and accepted DHCPv6 message. This is called TSlast.

A verified (for timestamp check and signature verification) secure DHCPv6 message initiates the update of the above variables in the recipient's record.

Recipients MUST check the Timestamp field as follows:

- When a message is received from a new peer (i.e., one that is not stored in the cache), the received timestamp, TSnew, is checked, and the message is accepted if the timestamp is recent enough to the reception time of the packet, RDnew:

  -Delta < (RDnew - TSnew) < +Delta

  After the signature verification also succeeds, the RDnew and TSnew values SHOULD be stored in the cache as RDlast and TSlast.

- When a message is received from a known peer (i.e., one that already has an entry in the cache), the timestamp is checked against the previously received Secure DHCPv6 message:

  TSnew + fuzz > TSlast + (RDnew - RDlast) x (1 - drift) - fuzz

  If this inequality does not hold or RDnew < RDlast, the recipient SHOULD silently discard the message. If, on the other hand, the inequality holds, the recipient SHOULD process the message.

  Moreover, if the above inequality holds and TSnew > TSlast, the recipient SHOULD update RDlast and TSlast after the signature verification also successes. Otherwise, the recipient MUST NOT update RDlast or TSlast.

An implementation MAY use some mechanism such as a timestamp cache to strengthen resistance to replay attacks. When there is a very large number of nodes on the same link, or when a cache filling attack is in progress, it is possible that the cache holding the most recent timestamp per sender will become full. In this case, the node MUST remove some entries from the cache or refuse some new requested

entries. The specific policy as to which entries are preferred over others is left as an implementation decision.

An implementation MAY statefully record the latest timestamps from senders. In such implementation, the timestamps MUST be strictly monotonously increasing. This is reasonable given that DHCPv6 messages are rarely misordered.

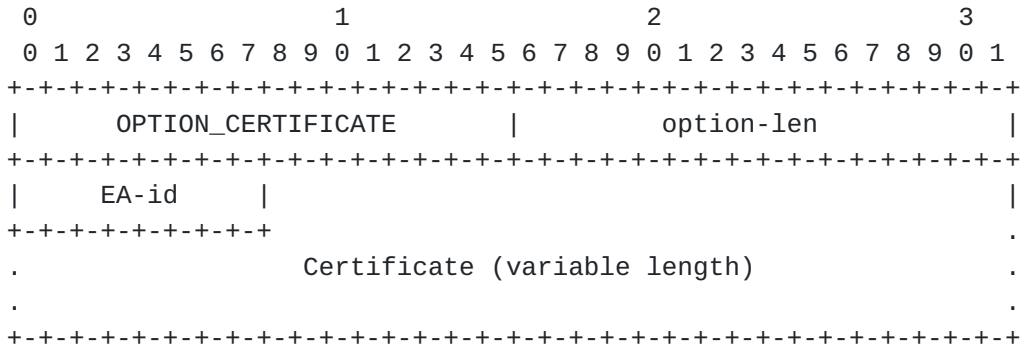
10. Extensions for Secure DHCPv6

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

10.1. New DHCPv6 Options

10.1.1. Certificate Option

The Certificate option carries the certificate of the client/server. The format of the Certificate option is described as follows:



option-code OPTION_CERTIFICATE (TBA1).

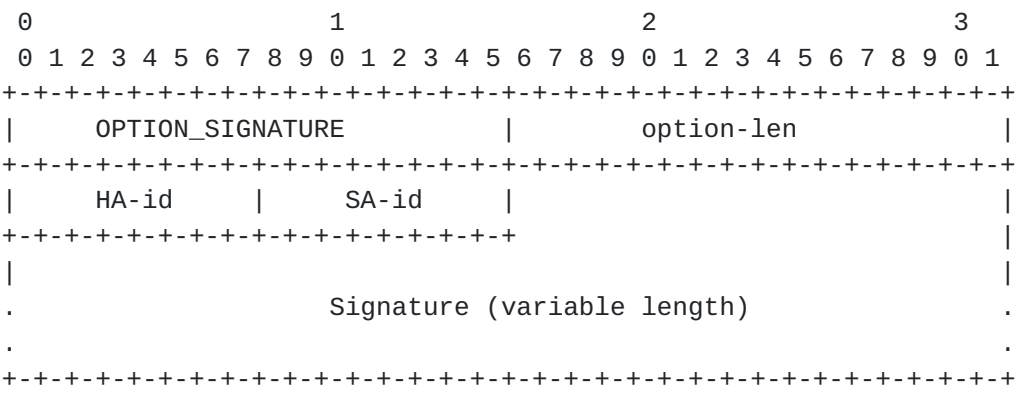
option-len 1 + Length of certificate in octets.

EA-id Encryption Algorithm id. 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.

Certificate A variable-length field containing certificate. 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.

10.1.2. Signature option

The Signature option allows a signature that is signed by the private key to be attached to a DHCPv6 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:



option-code OPTION_SIGNATURE (TBA2).

option-len 2 + Length of Signature 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. 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.

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.

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 DHCPv6 message header.

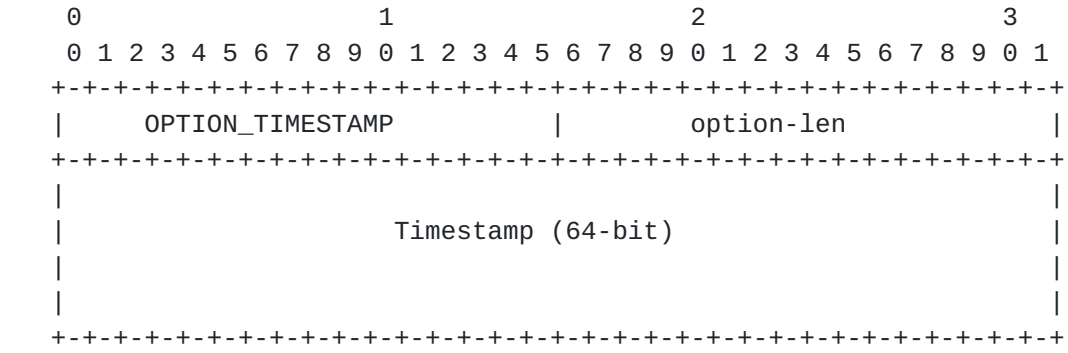
2. All DHCPv6 options including the Signature option (fill the Signature field with zeroes) except for the Authentication Option.

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.

10.1.3. Timestamp Option

The Timestamp option carries the current time on the sender. It adds the anti-replay protection to the DHCPv6 messages. It is optional.



option-code OPTION_TIMESTAMP (TBA3).

option-len 8, in octets.

Timestamp The current time of day (SeND-format timestamp in UTC (Coordinated Universal Time). It can reduce the danger of replay attacks. The timestamp data MUST be in format as defined in Section 5.3.1, [RFC3971].

10.1.4. Encrypted-message Option

The Encrypted-message option carries the encrypted DHCPv6 message with the recipient's public key.

The format of the Encrypted-message option is:

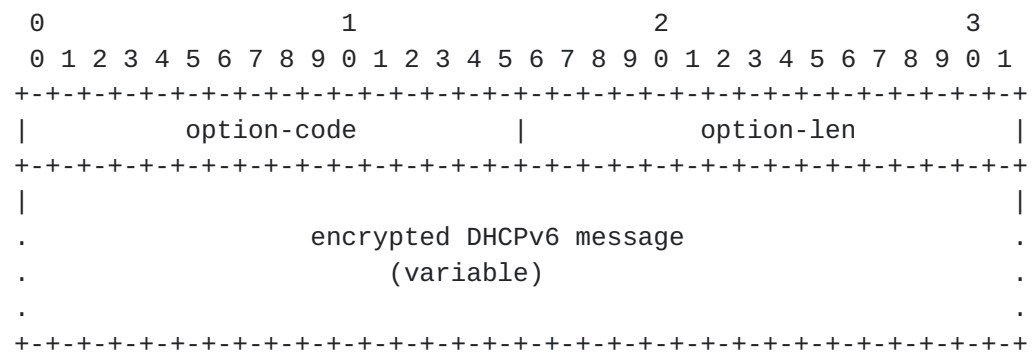


Figure 1: Encrypted-message Option Format

option-code OPTION_ENCRYPTED_MSG (TBA4).

option-len Length of the encrypted DHCPv6 message.

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

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:

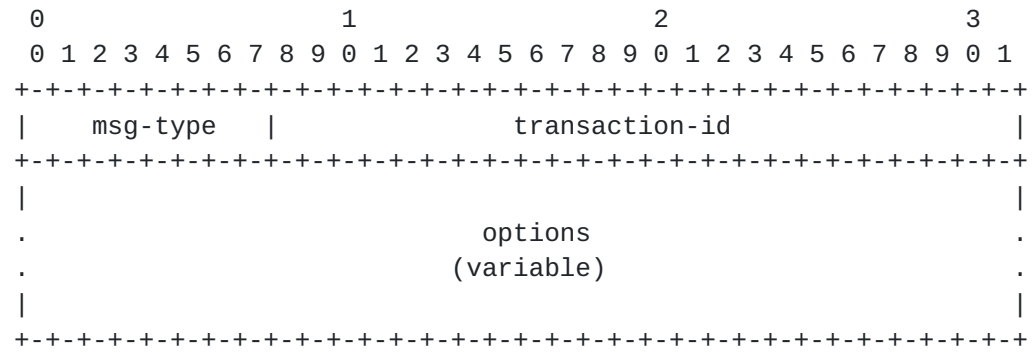


Figure 2: The format of Encrypted-Query and Encrypted-Response Messages

msg-type Identifier of the message type. It can be either Encrypted-Query (TBA5) or DHCPv6-Response (TBA6).

transaction-id The transaction ID for this message exchange.

options	The Encrypted-Query message MUST contain the Server Identifier option and Encrypted-message option. The Encrypted-Response message MUST contain the Encrypted-message option.

10.3. Status Codes

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

- AlgorithmNotSupported (TBD7): indicates that the DHCPv6 server does not support algorithms that sender used.
- AuthenticationFail (TBD8): indicates that the message from the DHCPv6 client fails authentication check.
- TimestampFail (TBD9): indicates the message from DHCPv6 client fails the timestamp check.
- SignatureFail (TBD10): indicates the message from DHCPv6 client fails the signature check.
- DecryptionFail (TBD11): indicates the message from DHCPv6 client fails the DHCPv6 message decryption.

11. Security Considerations

This document provides the authentication and encryption mechanisms for DHCPv6.

[RFC6273] has analyzed possible threats to the hash algorithms used in SEND. Since Secure DHCPv6 defined in this document uses the same hash algorithms in similar way to SEND, 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 Secure DHCPv6.

A server, whose local policy accepts messages without a Timestamp option, may have to face the risk of replay attacks.

A window of vulnerability for replay attacks exists until the timestamp expires. Secure DHCPv6 nodes are protected against replay attacks as long as they cache the state created by the message containing the timestamp. The cached state allows the node to protect itself against replayed messages. However, once the node flushes the state for whatever reason, an attacker can re-create the state by replaying an old message while the timestamp is still valid. In addition, the effectiveness of timestamps is largely dependent

upon the accuracy of synchronization between communicating nodes. However, how the two communicating nodes can be synchronized is out of scope of this work.

Attacks against time synchronization protocols such as NTP [RFC5905] may cause Secure DHCPv6 nodes to have an incorrect timestamp value. This can be used to launch replay attacks, even outside the normal window of vulnerability. To protect against these attacks, it is recommended that Secure DHCPv6 nodes keep independently maintained clocks or apply suitable security measures for the time synchronization protocols.

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.

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.

12. IANA Considerations

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

The Certificate Option (TBA1), described in Section 10.1.1.

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

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

The Encrypted-message Option (TBA4), described in Section 10.1.4.

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 (TBA5), described in Section 10.2.

The Encrypted-Response Message (TBA6), 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 8-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 8-bit unsigned integers. The following initial values are assigned for Signature Algorithm for Secure DHCPv6 in this document:

Name	Value	RFCs
RSASSA-PKCS1-v1_5	0x01	this document

Encryption algorithm for Secure DHCPv6. The values in this table are 8-bit unsigned integers. The following initial values are assigned for encryption algorithm for Secure DHCPv6 in this document:

Name	Value	RFCs
RSA	0	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
TBD7	AlgorithmNotSupported	this document
TBD8	AuthenticationFail	this document
TBD9	TimestampFail	this document
TBD10	SignatureFail	this document
TBD11	DecryptionFail	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, Qi Sun, 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-sedhcpv6-12: 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. Open Issues [RFC Editor: Please remove]

this protocol changes DHCPv6 message exchanges quite substantially: previously, the client first sends a Solicit message, gets possibly multiple Advertise messages, chooses the server (= sender of one of the Advertises) that would be best for the client, and then sends a Request to that chosen server. Now the server selection is done at the key exchange phase (the initial Information-request and Reply exchange), and the Solicit can be sent only to a single server. If the client doesn't like the Advertise it could restart the whole process, but it will be more expensive, and there's no guarantee that other servers can provide a better Advertise.

One might argue that it's okay as "secure DHCPv6" is an "optional" extension. But, with keeping in mind that the current IETF trend is to make everything privacy-aware (often by making everything encrypted), I'd personally say we should consider it to be the standard mode of DHCPv6 operation even if users can still disable it. From this point of view, I think we should either

- A. make the server selection behavior more compatible with the pre-encryption protocol, or

- B. accept we give up the previous server selection feature for privacy (after careful assessment of its effect and with clear wg consensus), and explicitly note that. we might even have to reflect that in rfc3315bis.

16. References

16.1. 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>.

[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>.

[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>.

[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>.

16.2. Informative References

[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>.

[RFC4270] Hoffman, P. and B. Schneier, "Attacks on Cryptographic Hashes in Internet Protocols", RFC 4270, DOI 10.17487/RFC4270, November 2005, <http://www.rfc-editor.org/info/rfc4270>.

[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

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

Email: jiangsheng@huawei.com

Lishan Li
Tsinghua University
Beijing 100084
P.R.China

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