DHC Working Group
Internet-Draft
Intended status: Standards Track
Expires: June 12, 2016

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

December 10, 2015

Secure DHCPv6
draft-ietf-dhc-sedhcpv6-10

Abstract

The Dynamic Host Configuration Protocol for IPv6 (DHCPv6) enables DHCPv6 servers to pass configuration parameters. It offers configuration flexibility. If not being secured, DHCPv6 is vulnerable to various attacks. This document analyzes the security issues of DHCPv6 and specifies a secure DHCPv6 mechanism for the authentication and encryption between DHCPv6 client and 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 June 12, 2016.

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 in details and provides the following mechanisms for improving the security of DHCPv6 between client and server:

- the authentication of the DHCPv6 client and the DHCPv6 server to defend against active attack, such as spoofing attack.
- the encryption between the DHCPv6 client and the DHCPv6 server in order to protect the DHCPv6 from passive attack, such as pervasive monitoring.
- the integrity check of DHCPv6 messages by the recipient of the message based on signature.
- anti-replay protection based on timestamps.

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, because they are only transported within operator networks and considered less vulnerable. 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 mechanisms specified in this document achieves the DHCPv6 authentication and encryption based on the sender's public key 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 the DHCPv6 authentication and encryption. The certificate option is used for the DHCPv6 authentication. It also integrates signature option for the integrity check and timestamps option for anti-replay protection. The Encryption-Query message, Encryption-Response message, and encrypted-message option are used for the 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 significant flaws and can be improved

The basic DHCPv6 specifications can optionally authenticate the origin of message 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 provides little message integrity or source integrity check, and it protects only the Reconfigure message. This key is transmitted in plaintext.

In addition, the current DHCPv6 messages are still transmitted in clear text 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 PM is an attack that needs to be mitigated where possible in [RFC7258].

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

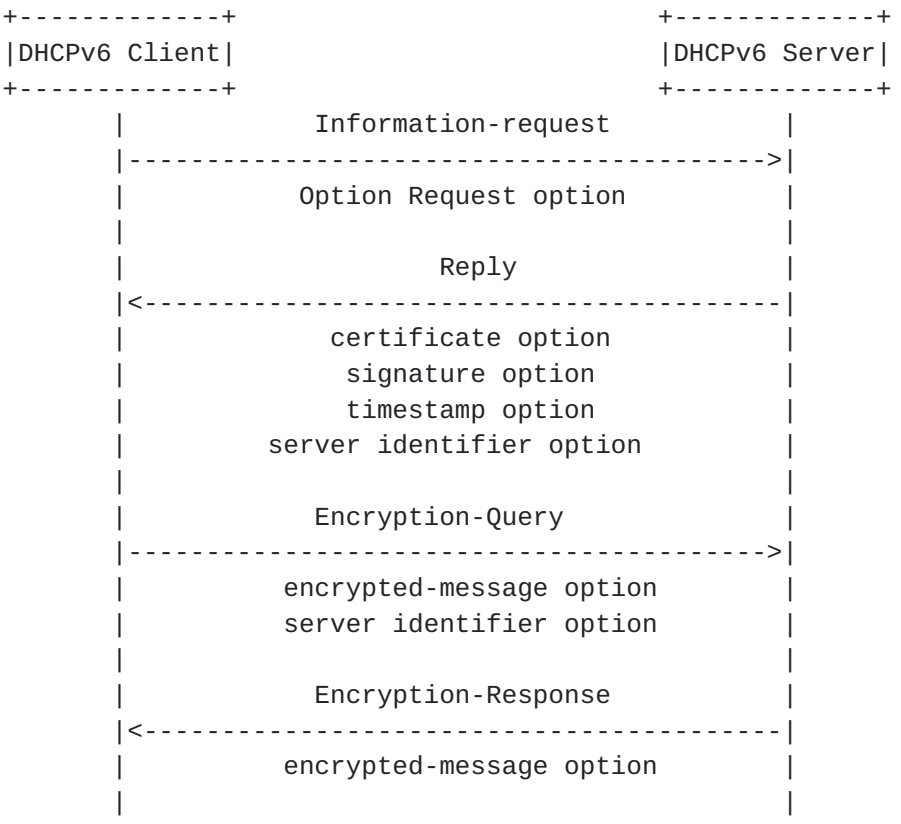
5. secure DHCPv6 overview

5.1. Solution Overview

This solution provides the authentication and encryption mechanisms based on the public certificates of the DHCPv6 client and server. Before the standard DHCPv6 configuration process, the Information-request and Reply messages are exchanged to select one authenticated DHCPv6 server. The following DHCPv6 configuration process is encrypted to avoid the privacy 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 proposed.

The following figure illustrates the secure DHCPv6 procedure. The DHCPv6 client first sends an Information-request message to the standard multicast address to all DHCPv6 servers. The Information-request message is used to request the servers for server authentication information, without going through any address, prefix or non-security option assignment process. The information-request is sent without client's privacy information, such as client identifier option to minimize information leak and increase client's privacy. 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 server authentication information in Reply message. If there are multiple authenticated DHCPv6 servers, the client selects one authenticated DHCPv6 server for the following DHCPv6 configuration process. If there are no authenticated DHCPv6 servers or existing servers failed authentication, the client behavior is policy specific. Depending on its policy, it can choose to connect repeat the server discovery process after certain delay or attempt to connect to a different network.

After the server's authentication, the first DHCPv6 message sent from client to server, such as Solicit message, contains the client's certificate option, signature option and timestamp option for client authentication. The DHCPv6 message sent from client to server is encrypted with the server's public key and encapsulated into the encrypted-message option. The DHCPv6 client sends the Encrypted-Query message to server, which carries the server identifier option and the encrypted-message option. 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, signature option, timestamp option, the DHCPv6 server verifies the client's identity according to the contained client authentication 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. The message that fails client authentication, MUST be dropped. And the server sends the corresponding error status code to client.



Secure DHCPv6 Procedure

It is worth noticing that the signature on a Secure DHCPv6 message can be expected to significantly increase the size of the message.

One example is normal DHCPv6 message length plus a 1 KB for a X.509 certificate and signature and 256 Byte for a signature. IPv6 fragments [RFC2460] are highly possible. In practise, the total length would be various in a large range. Hence, deployment of Secure DHCPv6 should also consider the issues of IP fragment, PMTU, etc. Also, if there are firewalls between secure DHCPv6 clients and secure DHCPv6 servers, it is RECOMMENDED that the firewalls are configured to pass ICMP Packet Too Big messages [RFC4443].

5.2. New Components

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

- Servers and clients that use certificates first generate a public/private key pair and then obtain a public key certificate from a Certificate Authority that signs the public key. One option is defined to carry the certificate.

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

- A timestamp that can be used to detect replayed packet. 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. Another option is defined to carry the current time of the client/server.

- An encrypted-message option that contains the encrypted DHCPv6 message.

- An Encrypted-Query message that sent from client to server. The Encrypted-Query message contains the encrypted-message option and server identifier option.

- An Encrypted-Response message that sent from server to client. The Encrypted-Response message contains the encrypted-message option.

5.3. Support for Algorithm Agility

Hash functions are used to provide message integrity checks. 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 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 the 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 recipient does not support the algorithm used by the sender, it cannot authenticate the message. In the client-to-server case, the server SHOULD reply with an AlgorithmNotSupported status code (defined in Section 10.3). Upon receiving this status code, the client MAY resend the message protected with the mandatory algorithm (defined in Section 10.1.2).

5.4. Imposed Additional Constraints

The client/server that supports the identity verification MAY impose additional constraints for the verification. For example, it may impose limits on minimum and maximum key lengths.

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

5.5. Applicability

Secure DHCPv6 is applicable in environments where physical security on the link is not assured and attacks on DHCPv6 are a concern, such as enterprise network. In enterprise network, the security policy is strict and the clients are stable terminals. The PKI model is used for the secure DHCPv6 deployment. The deployment of PKI is out of the scope of this document. The server is always considered to have connectivity to authorized CA and verify the clients' certificates. The client performs the server authentication locally. The trusted servers' certificates or trusted CAs' certificates, which form a certification path [RFC5280], is deployed in the client to achieve the server authentication. The DHCPv6 client obtains the trusted

certificates through the pre-configuration method or out of band, such as QR code. After the mutual authentication, the DHCPv6 message is encrypted with the recipient's public key, which is contained in the certificate.

6. DHCPv6 Client Behavior

For the security DHCPv6 client, it must have a public certificate. The client may be pre-configured with a public key certificate, which is signed by a CA trusted by the server, and its corresponding private key.

The DHCPv6 client multicasts the Information-request message to the DHCPv6 servers. The Information-request message MUST NOT include any option which may reveal the private information of the client, such as the client identifier option. 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 Option Request option in the Information-request message MUST contain the option code of certificate option, signature option, timestamp option, and server identifier option.

When receiving the Reply messages from DHCPv6 servers, a secure DHCPv6 client SHOULD discard 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 SHOULD first check the support of the hash and signature algorithms that the server used. If the check fails, the Reply message SHOULD be dropped. If both hash and signature algorithms are supported, the client then checks the authority of this server. The client SHOULD also use the same algorithms in the return messages.

The client SHOULD validate the certificate according to the rules defined in [RFC5280]. An implementation may create a local trust certificate record for verified certificates in order to avoid repeated verification procedure in the future. A certificate that finds a match in the local trust certificate list is treated as verified. At this point, the client has either recognized the authentication 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. If there are no authenticated DHCPv6 servers or existing servers failed authentication, the client behavior is policy specific. Depending on its policy, it can choose to connect using plain, unencrypted DHCPv6, repeat the server discovery process after certain delay or attempt to connect to a different network. The client MUST NOT conduct the server discovery process immediately 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 is constructed with 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 extra of decryption cost by those unselected servers.

The information for client authentication is contained in the Solicit/Information-request message, which is encrypted and then encapsulated into the Encrypted-Query message to avoid client privacy disclosure. The Solicit/Information-request message MUST contain the certificate option, which 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. It protects the message header and all DHCPv6 options except for the Authentication Option. 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 sender's real time clock.

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 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 recipient. 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 public key 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 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, it also MUST have a public certificate. The server may be pre-configured a public key certificate, which is signed by a CA trusted by the server, and its corresponding private key.

When the DHCPv6 server receives the Information-request message and the contained Option Request option informs the request for the server authentication information, it replies the 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. In addition, the Reply message MUST contain one and only one Signature option, which MUST be constructed as explained in Section 10.1.2. It protects the message header and all DHCPv6 options except for the Authentication Option. 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 the message.

If the decrypted message is Solicit/Information-request message, the secure DHCPv6 server SHOULD discard the received message that meet 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 SHOULD reply an UnspecFail (value 1, [RFC3315]) error status code.

The server SHOULD first check the support of the hash and signature algorithms 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 both hash and signature algorithms are supported, the server then checks the authority of this client.

If a certificate option is provided, the server SHOULD validate the certificate according to the rules defined in [RFC5280]. An implementation may create a local trust certificate record for verified certificates in order to avoid repeated verification procedure in the future. A certificate that finds a match in the local trust certificate list is treated as verified.

The message that fails certificate validation, MUST be dropped. In such failure, the DHCPv6 server SHOULD reply 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 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.

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 describes 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 SHOULD NOT add any secure DHCPv6 options.

Relay-forward and Relay-reply messages MUST NOT contain any additional certificate option or signature Option or timestamp Option, aside from those present in the innermost encapsulated messages from the client or server.

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, with certain allowed clock drift. So, 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 both 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 both 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.

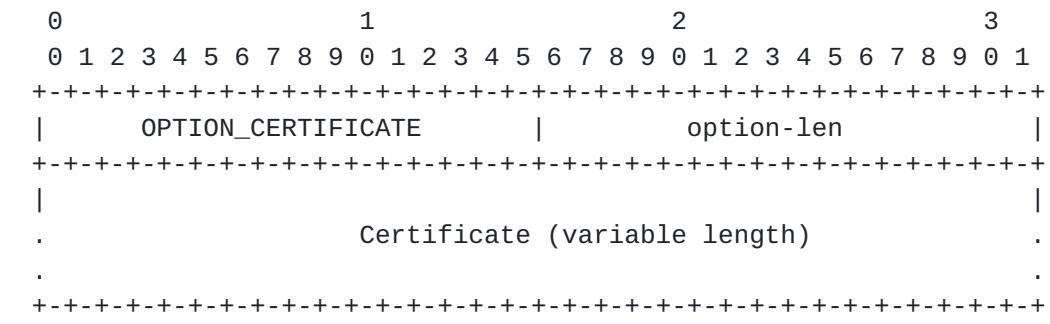
10. Extensions for Secure DHCPv6

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

10.1. New DHCPv6 Options

10.1.1. Certificate Option

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



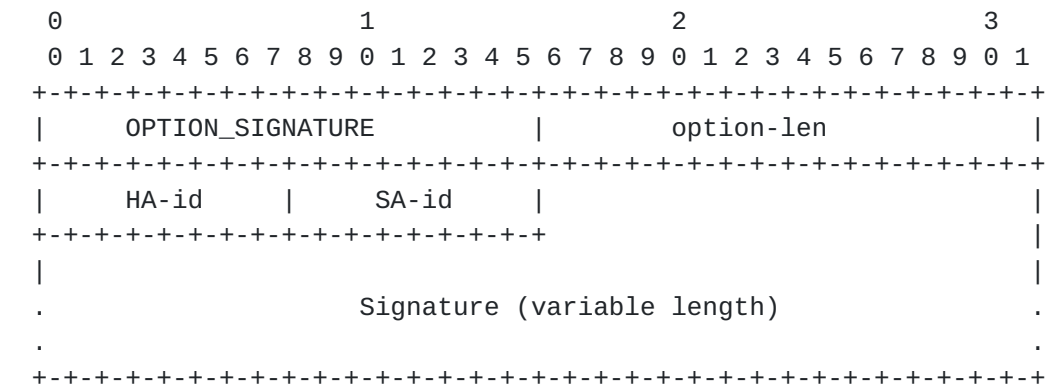
option-code OPTION_CERTIFICATE (TBA1).

option-len Length of certificate in octets.

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 - Signature (4) 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 any place within the DHCPv6 message while it is logically created after the entire DHCPv6 header and options, except for the Authentication Option. It protects the entire DHCPv6 header and options, including itself, except for the Authentication Option. 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 8.

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 8.

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.

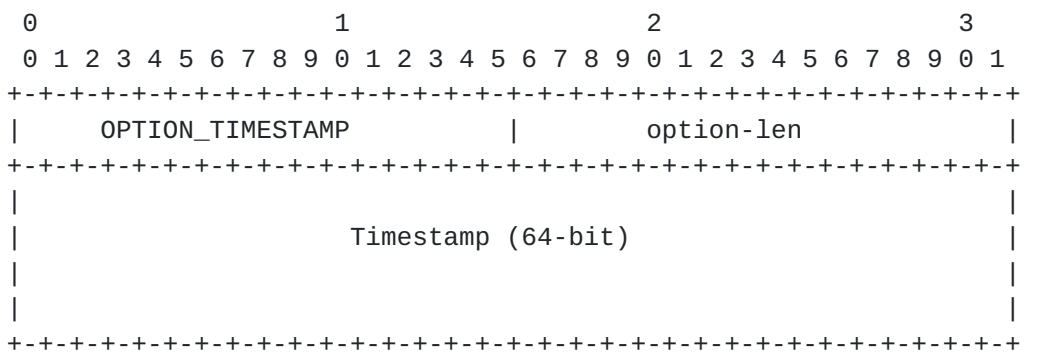
2. All DHCPv6 options including the Signature option (fill the signature field with zeroes) except for the Authentication Option.

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 both signature and authentication option are present, signature option does not protect the Authentication Option. It allows the Authentication Option be created after signature has been calculated and filled with the valid signature. It is because both options need to apply hash algorithm to whole message, so there must be a clear order and there could be only one last-created option. changing auth option, the authors chose not include authentication option in the signature.

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.

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:

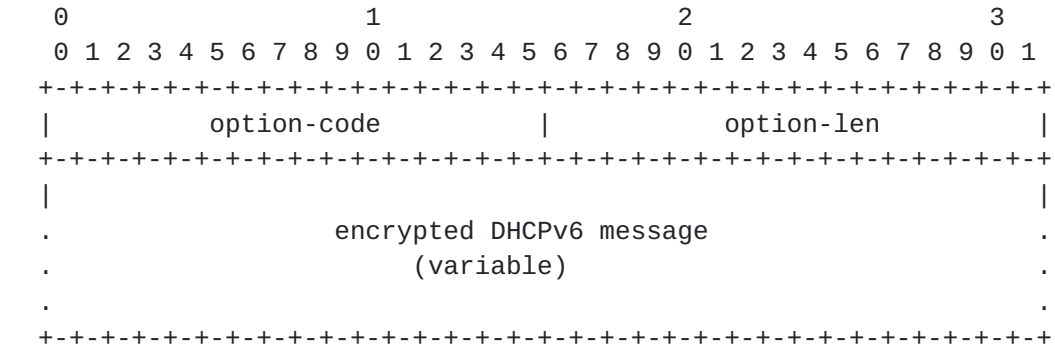


Figure 1: encrypted-message Option Format

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

10.2.1. Encrypted-Query Message

The Encrypted-Query message is sent from DHCPv6 client to DHCPv6 server, which contains the server identifier option and encrypted-message option.

The format of the Encrypted-Query message is:

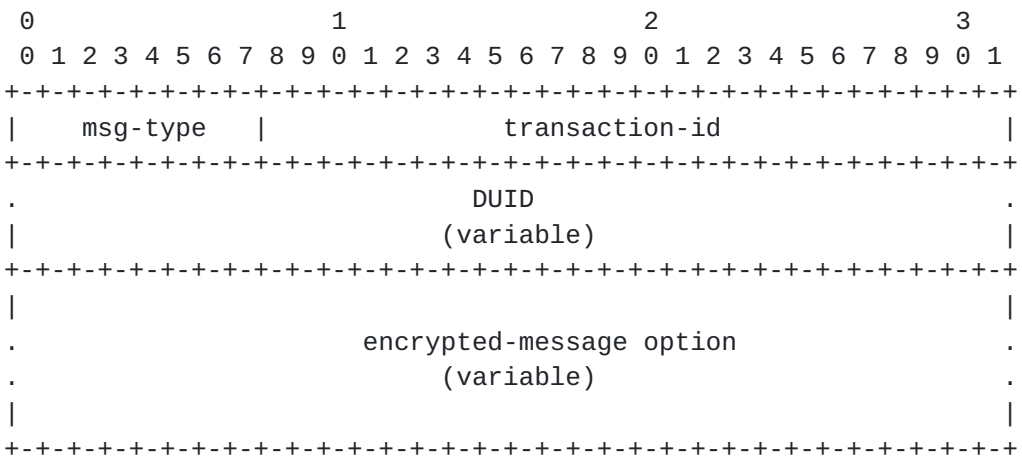


Figure 2: The format of Encrypted-Query Message

msg-type ENCRYPTED-QUERY (TBA5)

transaction-id The transaction ID for this message exchange.

DUID The DUID for the server.

encrypted-message option The encrypted DHCPv6 message.

10.2.2. Encrypted-Response Message

The Encrypted-Response message is sent from DHCPv6 server to DHCPv6 client, which contains the encrypted-message option.

The format of the Encrypted-Response message is:

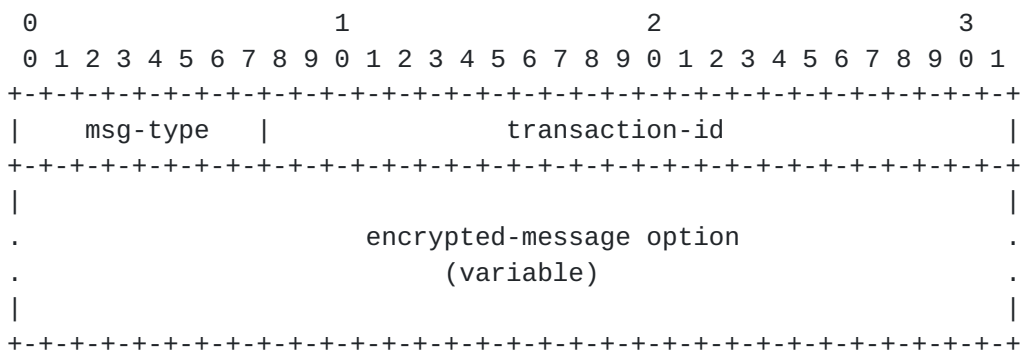


Figure 3: The format of Encrypted-Response Message

msg-type ENCRYPTED-RESPONSE (TBA6).

transaction-id The transaction ID for this message exchange.

encrypted-message option The encrypted DHCPv6 message.

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 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.

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 the 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 the 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.

12. IANA Considerations

This document defines five new DHCPv6 [RFC3315] options. The IANA is requested to assign values for these five options from the DHCPv6 Option Codes table of the DHCPv6 Parameters registry maintained in http://www.iana.org/assignments/dhcpv6-parameters. The five 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 10.2.1.

The Encrypted-Response Message (TBA6), described in Section 10.2.2.

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

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

13. Acknowledgements

The authors would like to thank Tomek Mrugalski, Bernie Volz, Randy Bush, Yiu Lee, Jianping Wu, 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. References

14.1. 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>.

[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>.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <http://www.rfc-editor.org/info/rfc5280>.

[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>.

14.2. 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: lilishan9248@126.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