DHC Working Group
Internet Draft
Intended status: Proposed Standard
Update: RFC3315
Expires: December 31, 2013

Sheng Jiang
Huawei Technologies Co., Ltd
Sean Shen
CNNIC
June 29, 2013

Secure DHCPv6 with Public Key
draft-jiang-dhc-sedhcpv6-00.txt

Status of this Memo

Copyright Notice

Abstract

The Dynamic Host Configuration Protocol for IPv6 (DHCPv6) enables DHCPv6 servers to pass configuration parameters. It offers configuration flexibility. If not secured, DHCPv6 is vulnerable to various attacks, particularly spoofing attacks. This document analyzes the security issues of DHCPv6 and specifies a Secure DHCPv6 mechanism. This mechanism is based on public/private key pairs. The authority of the sender may depend on either pre-configuration mechanism or Public Key Infrastructure.

Table of Contents

1. Introduction

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

This document analyzes the security issues of DHCPv6 in details. This document provides mechanisms for improving the security of DHCPv6:

- the identity of a DHCPv6 message sender, which can be a DHCPv6 server, a relay agent or a client, can be verified by a receiver.

- The integrity of DHCPv6 messages can be checked by the receiver of the message.

The security mechanisms specified in this document is based on self-generated public/private key pairs. It also integrates timestamps for anti-replay. The authentication procedure defined in this document may depend on either deployed Public Key Infrastructure (PKI, [RFC5280]) or pre-configured sender's public key. However, the deployment of PKI or pre-configuration is out of the scope.

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

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Security Overview of DHCPv6

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

a) 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 key of the hash function, there are two key management mechanisms. Firstly, the key management is out of band, usually manual, i.e., operators set up key database for both server and client before running DHCPv6. Usually multiple keys are deployed one a time and key id is used to specify which key is used.

Manual key distribution runs counter to the goal of minimizing the configuration data needed at each host. [RFC3315] provides an additional mechanism for preventing off-network timing attacks using the Reconfigure message: the Reconfigure Key authentication method. However, this method provides no message integrity or source integrity check. This key is transmitted in plaintext.

Comparing to this, the public/private key pair security mechanism only require a key pair on the sender. The key management mechanism is very simple.

b) Communication between a server and a relay agent, and communication between relay agents, can be secured through the use of IPsec, as described in section 21.1 in [RFC3315]. However, IPsec is quite complicated. A simpler security mechanism, which can be easier to deploy, is desirable.

4. Secure DHCPv6 Overview

To solve the above mentioned security issues, we introduce the use of public/private key pair mechanism into DHCPv6, also with timestamp. The authority of the sender may depend on either pre-configuration mechanism or PKI. By combining with the signatures, sender identity can be verified and messages protected.

This document introduces a Secure DHCPv6 mechanism that uses the public/private key pair to secure the DHCPv6 protocol. It assumes: a) the secured DHCPv6 message sender already has a public/private key pair; b) the receiver has already been have the public key of the sender, which may be pre-configured or recorded from previous communications, or the public key of CA (Certificate Authority), which issues the sender's certificate and is trusted by the receiver.

In this document, we introduce a key/certificate option and two signature options with a corresponding verification mechanism. Timestamp is integrated into signature options. A DHCPv6 message (from a server, a relay agent or a client), with a key/certificate option and carry a digital signature, can be verified by the receiver for both the timestamp and authentication, then process the payload of the DHCPv6 message only if the validation is successful.

This improves communication security of DHCPv6 messages. The authentication options [RFC3315] may also be used for replay

protection.

Because the sender can be a DHCPv6 server, a relay agent or a client, the end-to-end security protection can be from DHCPv6 servers to relay agents or clients, or from clients to DHCPv6 servers. Relay agents MAY add its own Secure DHCPv6 options in Relay-Forward messages when transmitting client messages to the server.

4.1. New Components

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

- A public/private key pair has been generated by a node itself. The node may request a CA to sign its public key to get a trustable certificate, which contains the original public key. Two new DHCPv6 option are defined to carry the public key or the certificate of the sender.

- Signatures signed by private key protect the integrity of the DHCPv6 messages and authenticate the identity of the sender.

- Timestamp, a value that indicates the relative time in second.

4.2. Support for algorithm agility

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

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

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

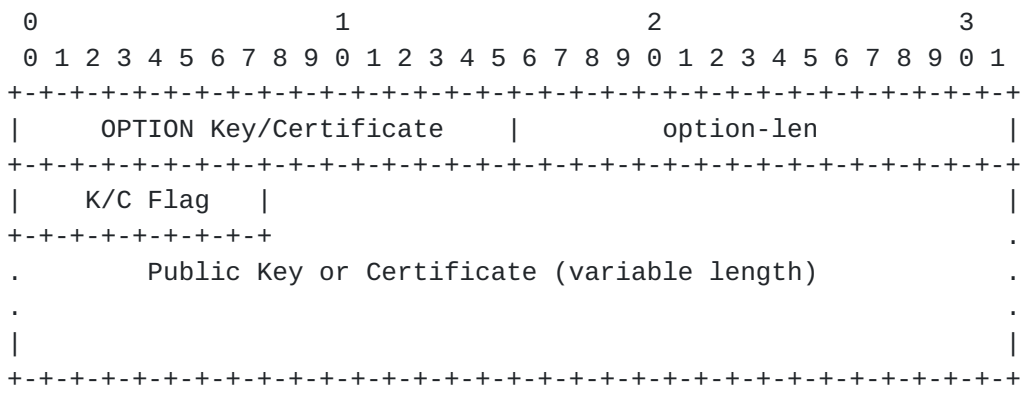
The support for algorithm agility in this document is mainly a unilateral notification model from a sender to a receiver. If the receiver cannot support the algorithm provided by the sender, it takes the risk itself. Senders in a same network do not have to upgrade to a new algorithm simultaneously.

5. Extensions for Secure DHCPv6

This section extends DHCPv6. Three new options have been defined. The new options MUST be supported in the Secure DHCPv6 message exchange.

5.1. Key/Certificate Option

The Key/Certificate option carries the public key or certificate of the sender. The format of the Public Key option is described as follows:



option-code OPTION_KC_PARAMETER (TBA1).

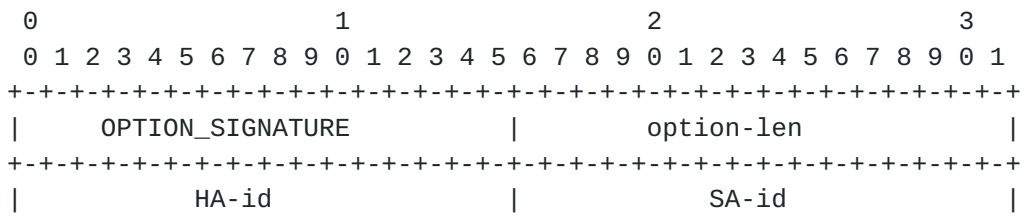
option-len 1+ length of public key/certificate in octets.

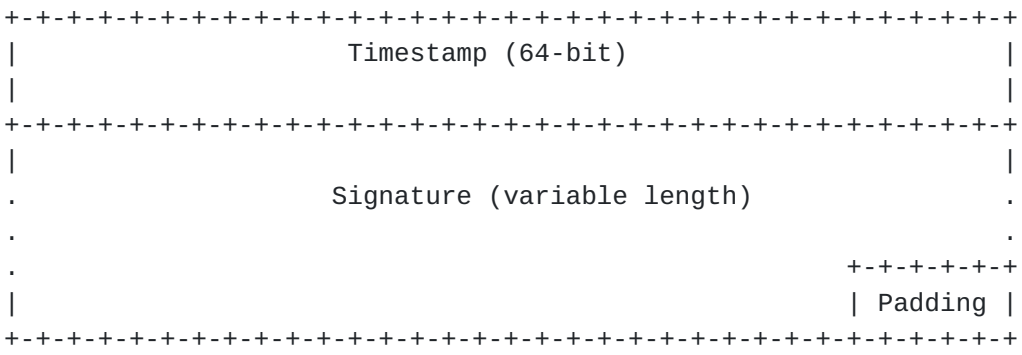
K/C Flag Flag to indicate whether the value is a public key or certificate. 00x for public key; FFx for certificate. Other values may be extended in the future.

Public key A variable-length field containing public key or certificate.

5.2. Signature Option

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





option-code OPTION_SIGNATURE (TBA2).

option-len 12 + Length of Signature field and Padding field in octets.

HA-id Hash Algorithm id. The hash algorithm is used for computing the signature result. This design is adopted in order to provide hash algorithm agility. The value is from the Hash Algorithm for Secure DHCPv6 registry in IANA. The initial values are assigned for SHA-1 is 0x0001.

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 initial values are assigned for RSASSA-PKCS1-v1_5 is 0x0001.

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

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

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

1. The 128-bit Source IPv6 Address.

2. The 128-bit Destination IPv6 Address.

3. The DHCPv6 message header.

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

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

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

Note: a Relay-Reply message is constructed by a DHCPv6 server in segments. The server first constructs the server message for client, which includes a Signature Option that covers the server message. In the signed data, the destination address is the address of the client. It then constructs the Relay-Reply message by encapsulating the server message into a Relay Message Option. If there is additional option for relay, the server MUST include another Signature Option, which covers the entire Relay-Reply message. In the signed data, the destination address is the address of the target relay agent.

6. Processing Rules and Behaviors

6.1. Processing Rules of Sender

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

The node MUST have a public/private key pair in order to create Secure DHCPv6 messages. The node may have a certificate which is signed by a CA trusted by both sender and receiver.

To support Secure DHCPv6, the Secure DHCPv6 enabled sender MUST construct the DHCPv6 message following the rules defined in [RFC3315].

A Secure DHCPv6 message MUST contain both the Key/Certificate option and the Signature option, except for Relay-forward and Relay-reply Messages.

Senders SHOULD set the Timestamp field to the current time, according to their real time clocks.

If a relay agent adds its own options in a Relay-forward message, it MUST contain the Key/Certificate option and the Signature option. If it does not any add new options it MUST NOT add either the Key/Certificate option or the Signature option into Relay-forward message. If there are more than a number of Relay agents (the number depends on the lengths of public key and signature, typical number is four) in the way and each of them adds their own options, it may exceed the IPv6 MTU. However, this can be considered as a rare deployment scenario.

Relay-reply Messages MUST NOT contain the Key/Certificate option since it appears in the Relay Message Option. If a server adds addition options for relay agents in Relay-reply message, it MUST contain a Signature Option. If it does not add any addition options, it MUST NOT add the Signature Option into the Relay-reply message.

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

6.2. Processing Rules of Receiver

When receiving a DHCPv6 message (except for Relay-Forward and Relay-Reply messages), a Secure DHCPv6 enabled receiver SHOULD discard the DHCPv6 message if either the Key/Certificate option or the Signature option is absent. If both options are absent, the receiver MAY fall back the unsecure DHCPv6 model.

The receiver SHOULD first check the authority of this sender. If the sender uses public key in the Key/Certificate option, the receiver SHOULD trust it by finding a match public key from the local trust public key list, which is pre-configured or recorded from previous communications. If the sender uses certificate in the Key/Certificate option, the receiver SHOULD validation the sender's certificate following the rules defined in [RFC5280]. An implementation may then create a local trust certificate record, too. The receiver may choose to further process the message from an unauthorized sender so that a leap of faith may be built up.

Then, the receiver MUST verify the Signature and check timestamp. The order of two procedures is left as an implementation decision. It is RECOMMENDED to check timestamp first, because signature verification is much more computational expensive.

The signature field verification MUST show that the signature has been calculated as specified in Section 5.2.

Only the messages that get through both the signature verifications

and timestamp check are accepted as secured DHCPv6 messages and

continue to be handled for their contained DHCPv6 options as defined in [RFC3315]. Messages that do not pass the above tests MUST be discarded or treated as unsecure messages.

The receiver MAY record the verified public key or certificate for future authentications.

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

Minbits The minimum acceptable key length for public keys. 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].

A Relay-forward message without any addition option to Relay Message option or a Relay-forward message with both addition options and the Signature option is accepted for a Secure DHCPv6 enabled server. Otherwise, the message SHOULD be discarded or treated as unsecure message. If Signature option is presented in the Relay-forward message, the signature verification and timestamp check are needed. The server MUST also verify signature for the encapsulated client DHCPv6 message in the Relay Message Option.

A Relay-reply message without any addition option to Relay Message option or a Relay-reply message with both addition options and the Signature Option is accepted for a Secure DHCPv6 enabled server. Otherwise, the message SHOULD be discarded or treated as unsecure message. If the Signature Option is presented in the Relay-reply message, the signature verification and timestamp check are needed. The relay agents obtain the public key or certificate of the server from the Key/Certificate option encapsulated in the Relay Message option.

6.3. Processing Rules of Relay Agent

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

In the client-relay-server scenario, the relay agent MAY verify the signature as a receiver before relaying the client message further, following verification procedure define in Section 6.2. In the case of failure, it MUST discard the DHCPv6 message. However, the verification procedure on relay agents does not save the load of the

DHCPv6 server. The server still MUST verify the signature by itself in order to prevent the attack between the relay agent and server.

In the server-relay-client scenario, if the Signature Option and addition options are presented, the relay agent MUST verify the signature before relaying the server message further, following verification procedure define in Section 6.2. In the case of failure, it MUST discard the DHCPv6 message.

The relay agent MAY also verify the signature for the encapsulated DHCPv6 message in the Relay Message Option. This can be helpful if the DHCPv6 response traverses a separate administrative domain, or if the relay agent is in a separate administrative domain. However, this is not necessary because the DHCPv6 client validation will catch any modification to the response.

6.4. Timestamp Check

Receivers 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.

To facilitate timestamp checking, each receiver SHOULD store the following information for each sender:

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

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

An accepted DHCPv6 message is any successfully verified (for both timestamp check and signature verification) DHCPv6 message from the given peer. It initiates the update of the above variables.

Receivers SHOULD then 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

  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 SEND message:

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

  If this inequality does not hold, the receiver SHOULD silently discard the message. If, on the other hand, the inequality holds, the receiver SHOULD process the message.

  Moreover, if the above inequality holds and TSnew > TSlast, the receiver SHOULD update RDlast and TSlast. Otherwise, the receiver 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.

7. Security Considerations

This document provides new security features to the DHCPv6 protocol.

Using public key based security mechanism and its verification mechanism in DHCPv6 message exchanging provides the authentication and data integrity protection. Timestamp mechanism provides anti-replay function.

The Secure DHCPv6 mechanism is based on the pre-condition that the receiver knows the public key of senders or the sender's certificate can be verified through a trust CA. It prevents DHCPv6 server spoofing. The clients may decline the DHCPv6 messages from unknown/unverified servers, which may be fake servers; or may prefer DHCPv6 messages from known/verified servers over unsigned messages or messages from unknown/unverified servers. The pre-configuration operation also needs to be protected, which is out of scope. The deployment of PKI is also out of scope.

However, when a DHCPv6 client first encounters a new public key or new unverified certificate, it can make a leap of faith. If the DHCPv6 server that used that public key/certificate is in fact legitimate, then all future communication with that DHCPv6 server can be protected by caching the public key. This does not provide complete security, but it limits the opportunity to mount an attack on a specific DHCPv6 client to the first time it communicates with a new DHCPv6 server.

Downgrade attacks cannot be avoided if nodes are configured to accept both secured and unsecured messages. A future specification may provide a mechanism on how to treat unsecured DHCPv6 messages.

[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 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.

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 SEND nodes keep independently maintained clocks or apply suitable security measures for the time synchronization protocols.

8. IANA Considerations

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

The Key/Certificate Parameter Option (TBA1), described in Section 5.1.

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

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

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

Name	Value	RFCs

Reserved	0x0000	this document
SHA-1	0x0001	this document
SHA-256	0x0002	this document

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

Name	Value	RFCs
Reserved	0x0000	this document
RSASSA-PKCS1-v1_5	0x0001	this document

9. Acknowledgments

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

10. References

10.1. Normative References

[RFC3315] R. Droms, et al., "Dynamic Host Configure Protocol for IPv6", RFC 3315, July 2003.

[RFC5280] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.

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

10.2. Informative References

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

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

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

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

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

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

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

Author's Addresses

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

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