

Network Working Group
Internet Draft
Intended status: Standards Track
Expires: April 25, 2010

Sheng Jiang
Sam(Zhongqi) Xia
Huawei Technologies Co., Ltd
October 26, 2009

Configuring Cryptographically Generated Addresses (CGA) using DHCPv6
draft-jiang-csi-cga-config-dhcpv6-01.txt

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/ietf/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at
<http://www.ietf.org/shadow.html>

This Internet-Draft will expire on April 25, 2010.

Copyright Notice

Copyright (c) 2009 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents in effect on the date of publication of this document (<http://trustee.ietf.org/license-info>). Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Internet-Draft [draft-jiang-csi-cga-config-dhcpv6-01.txt](#)

October 2009

Abstract

A Cryptographically Generated Address (CGA) is an IPv6 addresses binding with a public/private key pair. However, the current CGA specifications are lack of procedures to enable proper management of CGA generation. Administrators should be able to configure parameters used to generate CGA. The Dynamic Host Configuration Protocol for IPv6 (DHCPv6), which enables network management to dynamically configure hosts, can be used in the CGA configuration. Furthermore, CGA generation consumes large computation power. This computational burden can be delegated to the DHCPv6 server. A new DHCPv6 options are also defined in this document to enable hosts delegate CGA generation to a DHCPv6 server.

Table of Contents

1.	Introduction.....	3
2.	Terminology.....	3
3.	Requirements.....	4
	3.1. Configuration of the parameters required for the generation of CGA.....	4
	3.2. Offloading the large computational burden.....	5
4.	DHCPv6 Approach.....	5
	4.1. Node requests CGA-related configuration parameters to the DHCPv6 server.....	6
	4.2. Node requests CGA generation to the DHCPv6 server.....	6
5.	New CGA-related DHCPv6 Options.....	6
	5.1. DHCPv6 CGA Sec Option.....	6
	5.2. DHCPv6 CGA Generation Request Option.....	7
6.	Security Considerations.....	8
7.	IANA Considerations.....	9
8.	Acknowledgments.....	9
9.	References.....	9
	9.1. Normative References.....	9
	9.2. Informative References.....	10
	Author's Addresses.....	11

Internet-Draft [draft-jiang-csi-cga-config-dhcpv6-01.txt](#)

October 2009

1. Introduction

Cryptographically Generated Addresses (CGA, [[RFC3972](#)]) provide means to verify the ownership of IPv6 addresses without requiring any security infrastructure such as a certification authority. The use of CGAs allows identity verification in different protocols, such as SEure Neighbor Discovery (SEND, [[RFC3971](#)]), Enhanced Route Optimization for MIPv6 [[RFC4866](#)] or Site Multihoming by IPv6 Intermediation (SHIM6 [[RFC5533](#)]).

However, as [[PS-DC](#)] analyses, in the current specifications, there is a lack of procedures to enable proper management of CGA generation, in particular, in the configuration of the parameters that define the security properties of the addresses. Administrators should be able to configure parameters used to generate CGA. The Dynamic Host Configuration Protocol for IPv6 (DHCPv6), which enables network management to dynamically configure hosts, can be used in the CGA configuration. For example, DHCPv6 server should be able to assign subnet prefix or other relevant parameters to CGA address owner. In some scenarios, the administrator may further want to enforce some parameters, particularly, the demanded security related parameters such as SEC value.

Additionally, CGA generation is computational consumption. It can be a heavy burden for end-user devices, particular slow or battery-dependant devices. Currently, there are no means to delegate the computation of the modifier, a CPU intensive operation, to faster or non battery-dependant resources. It is possible that the whole or part of CGA generation procedure is delegated to the DHCPv6 server.

This draft analyses the requirements raised by CGA configuration and computational delegation for CGA generation. This draft provides solutions for CGA configuration and delegated CGA generation. Two existing DHCPv6 options are re-used. Two new DHCPv6 options are also defined in this document.

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](#) [[RFC2119](#)].

3. Requirements

The CGA specifications [[RFC3972](#)] define the procedure to generate a CGA. However, these procedures do not allow the enforcement of a given configuration to a group of hosts. It does also not consider the delegation of CPU-intensive operations to other nodes. In this section, we analyze the scenarios in which these operations are required.

3.1. Configuration of the parameters required for the generation of CGA

The CGA associated Parameters used to generate a CGA includes several parameters [[RFC3972](#)]:

- a Public Key,
- a Subnet Prefix,
- a 3-bit security parameter Sec. Additionally, it should be noted that the hash algorithm to be used in the generation of the CGA is also defined by the Sec value [[RFC4982](#)],
- a modifier that is selected so that the result of a hash to comply with the requirements introduced by the value of a security parameter Sec in order to provide protection against brute-force attacks,
- a Collision Count value, increased each time the address generated results in a collision in the subnet considered,
- any Extension Fields that could be used.

Currently, there are convenient mechanisms for allowing an administrator to configure the subnet prefix for a host, by Router Advertisement [[RFC4862](#)]. But other parameters used for generating the CGA could not be configured by the administrator.

It would be useful if these parameters could also be configured by the administrator. For instance, the administrator can determine, according to the type of infrastructure and the security needs, the Sec value that should be used by the hosts to generate the CGA. When appropriate, the Extension Fields could also be mandated by the administrator.

Upon reception of this information, the end hosts SHOULD generate addresses compliant with the received parameters. If the parameters

change, the end hosts SHOULD generate new addresses compliant with the parameters propagated.

3.2. Offloading the large computational burden

An important case to consider is the large computational consumption of the generation of the modifier field. The modifier is a 128 unsigned integer that is selected so that the Hash2 operation defined in [RFC 3972](#) results in the required number of leftmost 0 bits. The higher the number of bits required being 0, the more secure a CGA is against brute-force attacks. However, high number of bits also results in additional computational cost for the generation process, cost that could be deemed excessive in certain environments, such as mobile terminals with low computing power.

As an example, consider a Sec value equals 2, requesting the leftmost 32 bits of a SHA-1 Hash2 to be zero. For assuring this, a system has to generate in mean 2^{32} different modifiers, and perform the Hash2 operation to check the bits required to be 0. An estimation of the CPU power required to do this can be obtained as following: openssl can perform in an Intel Core2-6300 on an Asus p5b-w motherboard close to 0.87 million of SHA-1 operations on 16 byte blocks per second. Since the input data of Hash2 operation is larger than 16 bytes, this value is an upper bound for the number of hash operations that can be performed for generating the modifier. Checking 2^{32} different modifiers requires around 5000 seconds. The high number of required operations can represent a problem for end hosts (i.e. mobile devices)

with much lower computing power than considered in the example, and/or with restrictions in battery resources.

For these cases, a mechanism for delegating the computation of the modifier should be provided. It is also possible that the whole CGA generation procedure is delegated.

4. DHCPv6 Approach

Among the mechanisms in which configuration parameters could be pushed to the end hosts and/or CGA related information sent back to a central administration, we discuss the stateful configuration mechanism based on DHCPv6 in this document. Other mechanisms may also provide similar functions, but out of scope.

DHCPv6 can be extended to:

- propagate to the end hosts the values of the parameters required to configure CGAs,

- receive requests for generating a CGA according to a given security configuration, and returning the result to the end host.

4.1. Node requests CGA-related configuration parameters to the DHCPv6 server

A node may initiate a request for the relevant CGA configuration information needed to the DHCPv6 server. The server responds with the configuration information for the node. The Option Request Option, defined in [Section 22.7 in \[RFC3315\]](#), can be used for node to indicate which options the client requests from the server. To propagate the CHA-related parameters, the Identity Association for Prefix Assignment Option defined in [\[HGID\]](#) and a new CGA-Sec Option defined in [Section 5.1](#) can be used. Of course, a node can also use the sub-prefix received through Router Advertisement message [\[RFC4861\]](#). Future specification may define more options to carry CGA-related configuration parameters.

After receiving the configuration information, the node SHOULD generate a CGA based on its public key and the configuration information. The configuration of the client key pair or certificate is out of scope.

4.2. Node requests CGA generation to the DHCPv6 server

A node may initiate a request for the computation of the modifier or the CGA address for a certain security configuration to the DHCPv6 server. The node includes the values selected for the CGA associated parameters, such as its public key, the value of the Sec parameter, etc. The server either computes by itself, or redirects the computation to other node using a mechanism that is out of the scope of this document. Once the server generates or obtains the CGA, it responds to the node with the resulting address and the CGA Parameters Data Structure using the CGA Generation Request Option defined in [Section 5.2](#).

5. New CGA-related DHCPv6 Options

5.1. DHCPv6 CGA Sec Option

DHCPv6 CGA Sec Option is used to carry a Sec value, the parameters associated with CGA generation on a client. After receiving the CGA Sec Option, the client SHOULD generate a CGA using a Sec value that is not lower than the option indicated.

```

      0             1             2             3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           OPTION_CGA_SEC           |           option-len           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      CGA SEC      |
+---+---+---+---+---+---+

```

option-code OPTION_CGA_SEC (TBA).

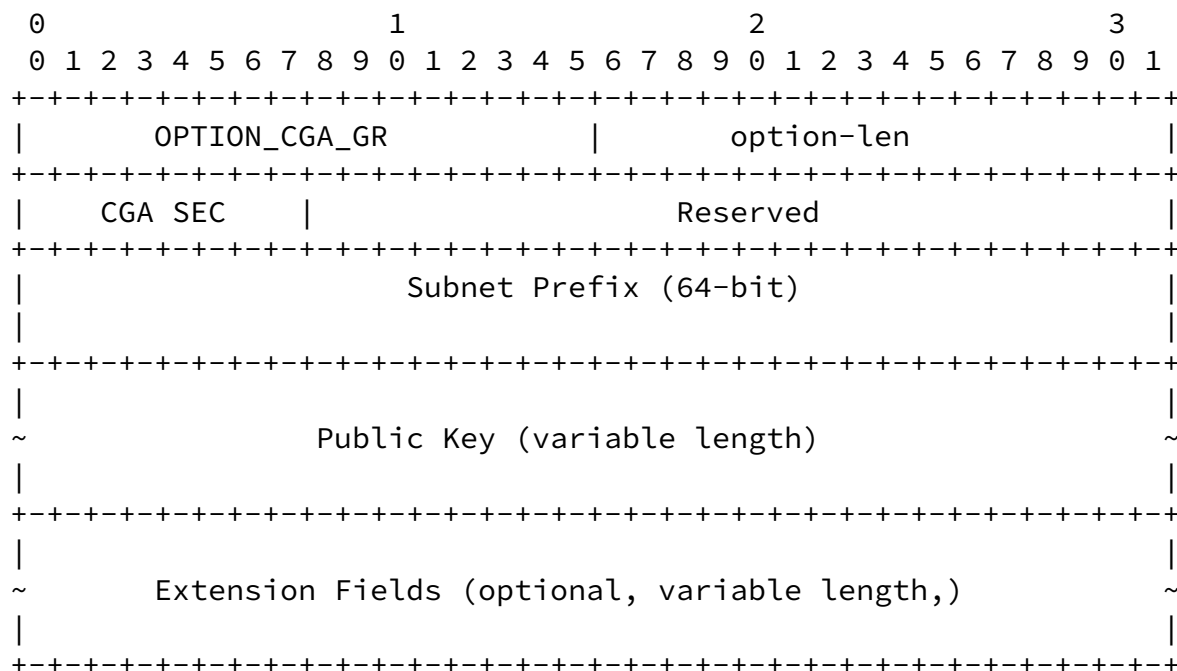
option-len 1.

CGA SEC a digit between 0 and 7, the SEC level.

5.2. DHCPv6 CGA Generation Request Option

DHCPv6 CGA Generation Request Option is sent by a client to request a

DHCPv6 server to generate a CGA address. After a DHCPv6 server receives CGA-relevant parameters sent by the client, it generates a CGA address based on these parameters and its own configuration. It then replies the CGA address and associated CGA Parameters data structure back to the client.



option-code OPTION_CGA_GR (TBA).

option-len 16 + Length of public key field in octets.

CGA SEC	a digit between 0 and 7, the SEC level require by the client.
Reserved	A 24-bit field reserved for future use. The value MUST be initialized to zero by the sender, and MUST be ignored by the receiver.
Subnet Prefix	An IPv6 prefix provided by the client, used for CGA generation. If set all 0, DHCPv6 server will use its own configured IPv6 subnet prefix.
Public Key	This is a variable-length field contain the

public key of the client. This public key will be used for CGA generation.

Extension Fields This is an optional variable-length field that is not used in the current specification. Future versions of this specification may use this field for additional data items that need to be included in the CGA Parameters data structure. Implementations MUST ignore the value of any unrecognized extension fields.

DHCPv6 server MAY use IA-NA or IA TA option with a CGA Parameter Data Structure IA sub-option to return the CGA address and associated CGA Parameters data structure back to the client.

DHCPv6 server MAY generate only a modifier and associated CGA Parameters data structure if it can not perform duplicate address detection, as per [\[RFC3971\]](#).

[6.](#) Security Considerations

The mechanisms based on DHCPv6 are all vulnerable to DOS attacks to the server, such as request for large number of CGA generations. Proper use of DHCPv6 autoconfiguration facilities [\[RFC3315\]](#), such as AUTH option or Secure DHCP [\[SDHCP\]](#) can prevent these threats, provided that a configuration token is known to both the client and the server.

Note that, as expected, it is not possible to provide secure configuration of CGA without a previous configuration of security information at the client (either a trust anchor, a DHCPv6 configuration token...). However, considering that the values of these elements could be shared by the nodes in the network segment, these security elements can be configured more easily in the end nodes than its addresses.

Regarding to the configuration of the Sec parameter, one risk is that a malicious node could propagate a Sec value providing less protection than intended by the network administrator, facilitating a brute force attack against the hash, or the selection of the weakest hash algorithm available for CGA definition. However, even in the worst case, if the hash algorithm cannot be inverted, the expected number of iterations required for a brute force attack is $O(2^{59})$ in

order to find a CGA Parameters data structure that matches a given CGA. Another risk is the use of a Sec, higher than intended by the administrator, which would require a large number of resources of the client to compute the modifier, requiring a long time before the device can communicate. This can be considered a kind of DOS attack. A variation of this attack is the propagation of different Sec values. This kind of attack may be prevented by server authentication.

An attacker could send malicious CGA Generation Requests in order to exhaust the server resources, since the CPU cost for the server can be high, especially considering that the attacker could select a Sec value requiring the highest number of computations for the server. This kind of attack may be prevented by host-based authentication.

[7.](#) 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 DHCPv6 CGA Sec Option (TBA1), described in [Section 5.1](#).

The DHCPv6 CGA Generation Request Option (TBA2), described in [Section 5.2](#).

[8.](#) Acknowledgments

The authors would like to thank Marcelo Bagnulo Braun and Alberto Garcia-Martinez from Universidad Carlos III de Madrid for been involved in the early requirement identification.

[9.](#) References

9.1. Normative References

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

[RFC3315] R. Droms, Ed., "Dynamic Host Configure Protocol for IPv6", [RFC3315](#), July 2003.

[RFC3971] J. Arkko, J. Kempf, B. Zill, P. Nikander, "SEcure Neighbor Discovery (SEND) ", [RFC 3971](#), March 2005.

- [RFC3972] T. Aura, "Cryptographically Generated Address", [RFC3972](#), March 2005.
- [RFC4861] T. Narten, et al., "Neighbor Discovery for IP version 6 (IPv6)", [RFC 4861](#), September 2007.
- [RFC4862] S. Thomson, T. Narten, T. Jinmei, "IPv6 Stateless Address Autoconfiguration", [RFC4862](#), September 2007.
- [RFC4866] J. Arkko, C. Vogt, W. Haddad, "Enhanced Route Optimization for Mobile IPv6", [RFC4866](#), May 2007.
- [RFC4982] M. Bagnulo, "Support for Multiple Hash Algorithms in Cryptographically Generated Addresses (CGAs) ", [RFC4982](#), July 2007.
- [RFC5533] E. Nordmark and M. Bagnulo "Shim6: Level 3 Multihoming Shim Protocol for IPv6" FRC 5533, June 2009

9.2. Informative References

- [PS-DC] S. Jiang, "DHCPv6 and CGA Interaction: Problem Statement", [draft-ietf-csi-dhcpv6-cga-ps-00.txt](#) (work in progress), October, 2009.
- [SDHCP] S. Jiang, "Secure DHCPv6 Using CGAs", [draft-jiang-dhc-secure-dhcpv6-02.txt](#) (work in progress), July 2009.
- [HGID] F. Xia, B. Sarikaya, S. Jiang, "Usage of Host Generating Interface Identifier in DHCPv6", [draft-xia-dhc-host-gen-id-02.txt](#) (work in progress), October 2009.

Author's Addresses

Sheng Jiang
Huawei Technologies Co., Ltd
KuiKe Building, No.9 Xinxu Rd.,
Shang-Di Information Industry Base, Hai-Dian District, Beijing 100085
P.R. China
Email: shengjiang@huawei.com

Sam(Zhongqi) Xia
Huawei Technologies Co., Ltd
KuiKe Building, No.9 Xinxu Rd.,
Shang-Di Information Industry Base, Hai-Dian District, Beijing 100085
P.R. China
Email: xiazhongqi@huawei.com