DHCPv6 and CGA Interaction: Problem Statement

draft-ietf-csi-dhcpv6-cga-ps-06.txt

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 April 25, 2011.

Copyright Notice

Copyright (c) 2010 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 (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.
Abstract

This document describes potential issues in the interaction between DHCPv6 and Cryptographically Generated Addresses (CGAs). Firstly, the scenario of using CGAs in DHCPv6 environments is discussed. Some operations are clarified for the interaction of DHCPv6 servers and CGA-associated hosts. We then also discuss how CGAs and DHCPv6 may have mutual benefits for each other, including using CGAs in DHCPv6 operations to enhance its security features and using DHCPv6 to provide the CGA generation function.

As an informational document, this document aims to analyze the possible interactions between CGAs and DHCPv6 from the functional perspective. This document does NOT propose/define any concrete solutions. Whether these possibilities are going to be defined as solutions or standards in the future is out of scope.

Table of Contents

1. Introduction..................................................3
2. Coexistence of DHCPv6 and CGA...............................3
3. What DHCPv6 can do for CGA................................4
   3.1. Configuring CGA-relevant parameters using DHCPv6......4
   3.2. Computation Delegation of CGA generation using DHCPv6...5
4. What CGA can do for DHCPv6................................6
5. Security Considerations......................................8
6. IANA Considerations..........................................8
7. Acknowledgements...........................................8
8. Change Log [RFC Editor please remove].......................8
9. References................................................9
   9.1. Normative References....................................9
Author’s Addresses.............................................10
1. Introduction

The Dynamic Host Configuration Protocol for IPv6 (DHCPv6) [RFC3315] can assign addresses statefully. Although there are other ways to assign IPv6 addresses [RFC4862, RFC5739], DHCPv6 is still useful when an administrator requires more control over address assignments to hosts. DHCPv6 can also be used to distribute other network configuration information.

Cryptographically Generated Addresses (CGAs) [RFC3972] are IPv6 addresses for which the interface identifiers are generated by computing a cryptographic one-way hash function from a public key and auxiliary parameters. Associated with public & private key pairs, CGAs are used in protocols, such as SEND [RFC3971] or SHIM6 [RFC5533], to provide address validation and integrity protection in message exchanging.

As an informational document, this document aims to analyze the possible interactions between CGAs and DHCPv6 from the functional perspective. This document does NOT propose/define any concrete solution. Whether these possibilities are going to be defined as solutions or standards in the future is out of scope.

This document describes potential issues in the interaction between DHCPv6 and CGAs. Firstly, the scenario of using CGAs in DHCPv6 environments is discussed. Some operations are clarified for the interaction of DHCPv6 servers and CGA-associated hosts. We then also discuss how CGAs and DHCPv6 may have mutual benefits for each other, including using CGAs in DHCPv6 operations to enhance its security features and using DHCPv6 to provide the CGA generation function. This document is designed to generate further discussion on the specifics of if/how the ideas in the document could be realized.

2. Coexistence of DHCPv6 and CGA

CGAs can be used with IPv6 Stateless Address Configuration (SLAAC) [RFC4862]. The CGA-associated public key, which is also transported to the receiver, provides message origin validation and integrity protection without the need for negotiation and transportation of key materials.

CGAs were designed for SeND [RFC3971] and SeND is generally not used in the same environment as a DHCP server. However, after CGA has been defined, as an independent security property, many other CGA usages have been proposed and defined, such as SHIM6 [RFC5533], Enhanced Route Optimization for Mobile IPv6 [RFC4866], also using the CGA for
DHCP security purpose, analyzed in section 4 this document, etc. In these scenarios, CGAs may be used in DHCPv6-managed networks.

In most cases, a CGA address is generated by the associated key pair owner, which normally is also the host that will use the CGA address, although the current CGA specifications do not consider or prohibit generation delegation. However, in a DHCPv6-managed network, hosts should use IPv6 global addresses only from DHCPv6 servers. This difference of roles needs to be carefully considered if there is a requirement to use CGAs in DHCPv6-managed environments.

The current DHCPv6 specification [RFC3315] has a mechanism that could be used to allow a host to self-generate a CGA for use in a DHCPv6-managed environment, i.e. the DHCPv6 server can grant the use of host-generated CGA addresses on request from the client.

Specifically, a node can request that a DHCPv6 server grants the use of a self-generated CGA by sending a DHCPv6 Request message. This DHCPv6 Request message contains an IA option including the CGA address. Depending on whether the CGA satisfies the CGA-related configuration parameters of the network, the DHCPv6 server can then send an acknowledgement to the node to either grant the use of the CGA or to indicate that the node must generate a new CGA with the correct CGA-related configuration parameters of the network. In the meantime the DHCPv6 server may log the requested address/host combination.

3. What DHCPv6 can do for CGA

3.1. Configuring CGA-relevant parameters using DHCPv6

In the current CGA specifications, it is not possible that network management to influence the CGA generation. Administrators may want to be able to configure parameters used to generate CGAs, for example to indicate hosts to use higher Sec-value CGAs. DHCPv6 could be used to assign subnet prefixes or other CGA-relevant parameters to CGA address owners. In some scenarios, the administrator may further want to enforce some parameters, in particular the necessary security-related parameters such as the SEC value.

Depending on the scenario, the configuration information needed to generate CGAs (including a SEC value, a subnet prefix, a modifier, a public key, a Collision Count value and any Extension Fields) may be provided by either hosts or DHCPv6 servers. A DHCPv6 server might receive from hosts the configuration information customized by hosts, generate CGAs by using configuration information provided by both parties and deliver CGAs and their associated CGA Parameters data.
structures to hosts. The details of such potential new methods need to be defined clearly in the solution specifications.

When designing such solutions, the designer should thoroughly consider the impact on DHCPv6 model and the security of CGA usage. In order to be compatible with DHCPv6, the configuring procedure of CGA parameters should be compatible with the current DHCPv6 definition. When a DHCPv6 server configures CGA parameters, integrity protection may be needed to avoid attacks, such as downgrade attack.

3.2. Computation Delegation of CGA generation using DHCPv6

This document only analyzes the functional possibility of computation delegation of CGA generation using DHCPv6. Whether CGA generation could be delegated or whether DHCPv6 is the most suitable tool for computation delegation of CGA generation are primary out of scope.

In the CGA generation procedure, the generation of the Modifier field of a CGA address is computationally intensive. This operation can lead to apparent slow performance and/or battery consumption problems for end hosts with limited computing ability and/or restricted battery power (e.g. mobile devices). As defined in [RFC3972], the modifier is a 128 unsigned integer that is selected so that the 16*SEC leftmost bits of the second hash value, Hash2, are zero. The modifier is used during CGA generation to implement the hash extension and to enhance privacy by adding randomness to the CGA. 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. 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. A practice experimental on a platform with an Intel Duo2 (2.53GHz) workstation showed the results of average CGA generating time as below: when SEC=0, it took 100us; SEC=1, 60ms; SEC=2, 2000s (varies from 100~7000sec). The experiment was unable to be performed for SEC=3 or higher SEC values. Theoretically estimating, about 30000 hours are required to generate a SEC=3 CGA.
Sec was not designed as a way to burden current machines but to enable higher cost searches when machines get faster. That is, if generating a Sec>0 value is a big burden for our computers, host should probably still use Sec=0.

A very low-power host might want to delegate its key and hash generation to a more general purpose computer.

In such cases, a mechanism to delegate the computation of the modifier would be desirable. It is possible that the whole CGA generation procedure could be delegated to the DHCPv6 server. This would be especially useful for large SEC values.

It may be possible to define a delegation operation that allows a client to pass computations to a DHCPv6 server, by introducing new DHCPv6 option(s). A node could thus initiate a DHCPv6 request to the DHCPv6 server requesting the computation of the Modifier or the CGA. The DHCPv6 server could then either compute the Modifier by itself, or redirect the computation requirement to another server. Once the DHCPv6 server generates (or obtains from the redirected computational server) the Modifier or the CGA address, it can respond to the node with the Modifier or the resulting address and the corresponding CGA Parameters data structure.

Generating a key pair, which will be used to generate a CGA, also requires a notable computation, though this may only be issues on a very low-power host occasionally. Generation and distribution of a key pair can also be done by a DHCPv6 server. Of course, when designing these new functions, one should carefully consider the impact on security. However, the security considerations of specific solutions are out of scope of this document.

New DHCPv6 options may be defined to support the interactions that are required when a DHCPv6 server generates a key pair for hosts.

4. What CGA can do for DHCPv6

DHCPv6 is vulnerable to various attacks, e.g. fake address attacks where a 'rogue' DHCPv6 server responds with incorrect address information. A malicious rogue DHCPv6 server can also provide incorrect configuration to the client in order to divert the client to communicate with malicious services, like DNS or NTP. It may also mount a Denial of Service attack through mis-configuration of the client that causes all network communication from the client to fail. A rogue DHCPv6 server may also collect some critical information from
the client. Attackers may be able to gain unauthorized access to some resources, such as network access. See Section 23 [RFC3315].

In the basic DHCPv6 specifications, regular IPv6 addresses are used. However, DHCPv6 servers, relay agents and clients could use host-based CGAs as their own addresses. A DHCPv6 message (from either a server, relay agent or client) with a CGA as source address can carry the CGA Parameters data structure and a digital signature. The receiver can verify both the CGA and signature, then process the payload of the DHCPv6 message only if the validation is successful. A CGA option with an address ownership proof mechanism and a signature option with a corresponding verification mechanism may be introduced into DHCPv6 protocol. With these two new options, the receiver of a DHCPv6 message can verify the sender address of the DHCPv6 message, which improves communication security of DHCPv6 messages. CGAs can be used for all DHCPv6 messages/processes as long as CGAs are available on the sender side.

Using CGAs in DHCPv6 protocol can efficiently improve the security of DHCPv6. The address of a DHCPv6 message sender (which can be a DHCPv6 server, a reply agent or a client) can be verified by a receiver. Also, the integrity of the sent data is provided if they are signed with the private key associated to the public key used to generate the CGA. The usage of CGA with pre-configured authorization, as introduced in next paragraph, can efficiently avoid the abovementioned attacks. It improves the communication security of DHCPv6 interactions. The usage of CGAs can also avoid DHCPv6’s dependence on IPsec [RFC3315] in relay scenarios. This mechanism is applicable in environments where physical security on the link is not assured (such as over certain wireless infrastructures) or where available security mechanisms are not sufficient, and attacks on DHCPv6 are a concern.

A CGA generated from an unauthorized public & private key pair can prove the source address ownership and provide data integrity protection. Furthermore, a CGA generated from a certified public & private key pair can also achieve authorization for DHCPv6 servers or relays, or on another direction, user authorization. The public keys may be pre-configured on both parties of communication or have a third party authority available for users to retrieve public keys. The public keys will be used for users to generate CGAs and verify CGAs and signatures. The pre-configuration can also include configuring more CGA parameters such as SEC value or more depending on policies. The pre-configuration can even be the whole CGA and related parameters, but in this case the address will be fixed. It may increase the vulnerability to, e.g., brute force attacks.
5. Security Considerations

As Section 4 of this document has discussed, CGAs can provide additional security features for DHCPv6. However, in defining solutions using DHCPv6 to configure CGAs, as suggested in Section 3 of this document, careful consideration is required to evaluate whether the new mechanism introduces new security vulnerabilities.

When DHCPv6 is used to manage CGAs, CGA relevant information is stored in a central repository, DHCPv6 server. It does not increase privacy risks. The CGA relevant information is only exposed to the network management plane. The privacy risks are not higher than other network managed entities, like normal IPv6 addresses managed by DHCP. Of course, the privacy risk is higher than using CGA with SLAAC, in which CGAs are fully host based.

Without other pre-configured security mechanism, like PKI, using host-based CGA by DHCPv6 servers could not prevent attacks claiming to be a DHCPv6 server.

This document does not contain a complete security analysis and any further work in this area should include such an analysis. Nobody should implement the techniques described in this document without conducting that more thorough analysis.

6. IANA Considerations

There are no IANA considerations in this document.

7. Acknowledgements

Useful comments were made by Marcelo Bagnulo, Alberto Garcia, Ted Lemon, Stephen Hanna, Russ Housley, Sean Turner, Tim Polk, Jari Arkko, David Harrington and other members of the IETF CSI working group.

8. Change Log [RFC Editor please remove]

draft-jiang-csi-dhcpv6-cga-ps-00, original version, 2008-10-27

draft-jiang-csi-dhcpv6-cga-ps-01, revised after comments at IETF 73, 2009-01-08

draft-jiang-csi-dhcpv6-cga-ps-02, revised after comments at CSI mailing list, 2009-06-17
9. References

9.1. Normative References


Author’s Addresses

Sheng Jiang
Huawei Technologies Co., Ltd
KuiKe Building, No.9 Xinxi Rd.,
Shang-Di Information Industry Base, Hai-Dian District, Beijing 100085
P.R. China
Phone: 86-10-82836081
Email: shengjiang@huawei.com

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

Tim Chown
University of Southampton
Highfield
Southampton, Hampshire SO17 1BJ
United Kingdom
Email: tjc@ecs.soton.ac.uk