

Network Working Group
Internet Draft
Expires: January 2009

Sheng Jiang
Sam(Zhongqi) Xia
Huawei Technologies Co., Ltd
Alberto Garcia-Martinez
UC3M
July 11th, 2008

Requirements for configuring Cryptographically Generated Addresses (CGA)
and overview on RA and DHCPv6-based approaches
[draft-jiang-sendcgaext-cga-config-02.txt](http://www.ietf.org/drafts/requirements/draft-jiang-sendcgaext-cga-config-02.txt)

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of BCP 79](#).

This document may only be posted in an Internet-Draft.

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 January 10, 2009.

Abstract

This document analyzes the requirements for the configuration Cryptographically Generated Addresses and Multi-key CGAs. The applicability of Router Advertisement and DHCPv6 for this configuration is also discussed.

Internet-Draft [draft-jiang-sendcgaext-cga-config-02.txt](#)

July 2008

Table of Contents

1.	Introduction.....	2
2.	Terminology.....	3
3.	Requirements.....	3
	3.1. Configuration of the parameters required for the generation of CGA.....	3
	3.1.1. Offloading the large computational burden.....	4
	3.1.2. Certificate information dissemination.....	5
	3.2. CGA granting and registration.....	5
	3.3. Configuration the parameters in order to enable the CGA proxy	5
4.	Approaches overview.....	6
	4.1. Node requests CGA-related configuration parameters to the DHCPv6 server.....	7
	4.2. Node requests to the DHCPv6 server the computation of the Modifier.....	7
	4.3. Node requests DHCPv6 server to grant the CGA.....	8
	4.4. Node sends MCGA-specific information to the DHCPv6 server	8
5.	Security Considerations.....	8
	5.1. Threat Analysis of the Configuration Requirements.....	8
	5.1.1. Threats faced by the end hosts.....	8
	5.1.2. Threats faced by the configuration servers and proxies	10
	5.2. Threat Analysis of the Approaches Proposed.....	10
	5.2.1. Router Advertisement with SEND support.....	11
	5.2.2. Router Advertisement without SEND support.....	11
	5.2.3. DHCPv6.....	11
6.	IANA Considerations.....	12
7.	Conclusions.....	12
8.	Acknowledgments.....	12
9.	References.....	12
	9.1. Normative References.....	12
	9.2. Informative References.....	13
	Author's Addresses.....	14
	Intellectual Property Statement.....	14
	Disclaimer of Validity.....	15
	Copyright Statement.....	15

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. As an extension to enable SEure Neighbor Discovery (SEND, [[RFC3971](#)]) proxy

Jiang, et al.

Expires January 10, 2009

[Page 2]

Internet-Draft [draft-jiang-sendcgaext-cga-config-02.txt](#)

July 2008

support, multi-key CGAs [[MCGA](#)] have been introduced. The use of both types of addresses has been proposed for allowing identity verification in different protocols, such as SEND, Enhanced Route Optimization for MIPv6 [[RFC4866](#)] or SHIM6 [[SHIM6-proto](#)].

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. Additionally, there is a lack of tools for informing the hosts about the availability of SEND proxies, and exchanging the required information with the proxies. Finally, there are no means to delegate the computation of the Modifier, a CPU intensive operation, to faster or non battery-dependant resources.

This draft analyses the configuration requirements raised by CGA and MCGA generation. Additionally, the applicability of Router Advertisement and Dynamic Host Configuration Protocol for IPv6 (DHCPv6 [[RFC3315](#)]) for performing this configuration is discussed.

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](#), [MCGA](#)] define the procedure to generate a CGA. However, these procedures do not allow the enforcement of a given configuration to a group of hosts, nor address the interactions between end hosts and proxies required for proxy configuration. 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 [[RFC 3972](#)]:

- a Public Key,
- a Subnet Prefix,
- a 3-bit security parameter Sec,

Jiang, et al.

Expires January 10, 2009

[Page 3]

Internet-Draft [draft-jiang-sendcgaext-cga-config-02.txt](#)

July 2008

- 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.

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](#)].

Currently, there are convenient mechanisms for allowing an administrator to configure the subnet prefix for a host. Other parameters used for generating the CGA could also benefit from the possibility of being 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.

[3.1.1](#). 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.

[3.1.2](#). Certificate information dissemination

CGAs enable the verification of the relationship between a public/private key pair (certificate) and an address. However, it does not verify the identity of a sender. In most of scenarios, it is necessary to know which certificates or certificate chains are trustworthy. Mechanisms are required to disseminate such information to CGA receivers.

[3.2](#). CGA granting and registration

The usage of self-generated CGAs may need to be granted by the networking management plate. Only granted CGAs are allowed to be used to access the network. It is also validated whether the CGAs do not use the reserved range of interface identifier [[RIID](#)].

As described in [RFC 3972](#), the modifier can be reused when the prefix of the CGA changes and this is the only change. However, when a mobile node moves from a network to another, not only the prefix changes, but also other CGA relevant parameters may change. Therefore, any CGAs generated by the node itself should also be granted by the networking management plate.

A node that has generated a CGA could register the resulting address so that a central administration could manage this information. The node could be requested to perform this registration.

[3.3](#). Configuration the parameters in order to enable the CGA proxy

In order to preserve location privacy of CGAs, the CGA proxy solutions, such as Multi-Key Cryptographically Generated Addresses (MCGAs), are introduced. These CGA proxy solutions require that certain information/parameters of proxy are configured.

First of all, end hosts should be notified that their SEND validation could be proxied, and therefore that they should generate MCGA addresses. In order to generate the MCGA, and in addition to the Sec

parameter and Extension Fields required for CGA bootstrapping, the node must know the node's own public key and the public key(s) from its proxy(s), which are certified router public keys. [RFC 3971](#) describe a mechanism that allows the node to obtain the public keys of the router(s), although other protocols could be used for this purpose.

Upon reception of this information, the end hosts SHOULD generate MCGAs compliant with the received parameters. If the parameters change, the end hosts SHOULD generate new MCGAs compliant with the parameters propagated.

Additionally, the proxy(s) should be notified the new MCGA and its associated CGA Parameters Data Structure, so that the proxy could securely proxy the MCGA by signing the message with its own private key. Consequently, a mechanism for making proxy(s) aware of the keys used by each end host should be provided.

[4](#). Approaches overview

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 two mechanisms: the stateless address configuration mechanism based in Router Advertisement, and the stateful configuration mechanism based in DHCPv6.

On one hand, Router Advertisement could be extended with an option that could convey parameters related with CGA configuration, such as the value of the Sec or the values of future Extension Fields, etc. In this way, a router could distribute these parameters to all the hosts of the subnet through Router Advertisement, in the same message in which prefix information is conveyed.

On the other hand, DHCPv6 can be extended to:

- propagate to the end hosts the values of the basic parameters required to configure CGAs,
- request the node to propagate to the server the resulting CGA address,
- grant the node to use its self-generated CGA address,
- obtain from the end host CGA information to update any database with the addresses being used,
- inform the end hosts about the convenience for generating MCGA,

- obtain from the end hosts the MCGA information required to configure the proxy(s),
- receive requests for generating a Modifier according to a given security configuration, and returning the result to the end host.

Finally, both Router Advertisement and DHCPv6 could be combined in the following cases:

- when the node is requested by Router Advertisement to register the resulting CGA, DHCPv6 could be used to inform the DHCPv6 server about the resulting address,
- when MCGA address are generated, Router Advertisement could be used to propagate the basic CGA parameters, and a notification that the end host should generate MCGA, and use DHCPv6 to inform the

DHCPv6 server about the public key material used for MCGA generation,

- when the node solicits the computation of the Modifier, after receiving a Router Advertisement with the Sec parameters and Extension Fields, it can issue the request through a DHCPv6 exchange.

We next describe in more detail the interactions foreseen for DHCPv6.

[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 server also sends its known certification information for the node. If registration of the resulting address is required, the server can include such requirement in the message sent. If SEND proxies are available, the server informs the node that an MCGA should be generated. The public keys for the routers, along with their certificates, could be included in the response.

After receiving the configuration information, the node generates a CGA (or a MCGA) based on its public key and the configuration information.

[4.2.](#) Node requests to the DHCPv6 server the computation of the Modifier

A node may initiate a request for the computation of the Modifier 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 the Modifier value, or redirects the computation to other node using a mechanism that is out of the scope of this draft. Once the server obtains the modifier, it computes the CGA or MCGA according to the process described in [RFC 3972](#), and it responds to the node with the resulting address and the CGA Parameters Data Structure.

[4.3.](#) Node requests DHCPv6 server to grant the CGA

A node requests DHCPv6 server to grant a CGA generated by the node itself, listing the CGA addresses in IA options [[RFC3315](#)]. According to whether the CGA matches the CGA-related configuration parameters of the network, the DHCPv6 server sends an acknowledgement to the node, grant the usage of the CGA or indicate the node that it must generate a new CGA with the CGA-related configuration parameters of the network. In the meantime, the DHCPv6 server has had the opportunity to log the address/host combination.

[4.4.](#) Node sends MCGA-specific information to the DHCPv6 server

A node that has generated its MCGA informs the DHCPv6 server about the MCGA and its associated CGA Parameters Data Structure. The DHCPv6 server sends an acknowledgement to the node. The server or the node also needs to notify this information to the routers acting as SEND proxies, in a way that is out of the scope of this document.

[5.](#) Security Considerations

[5.1.](#) Threat Analysis of the Configuration Requirements

[5.1.1.](#) Threats faced by the end hosts

We first discuss the threats that the clients may face as a result of the operations described in this document.

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. 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 with a given node.

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 until 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 could force the nodes to generate different addresses, requiring the

generation of a new modifier, etc. The end host SHOULD store the addresses that were generated in the past according to different Sec values.

The disclosure of the Sec value to any party does not represent any threat.

The analysis of the threats for the configuration of CGA Extension Fields should be performed in a case-by-case basis.

Regarding to the propagation of MCGA-related information, an attacker could generate a key pair, and propagate the public key to the end host, so the MCGA generated were associated with the public key of the attacker, In this way, the attacker would be able to impersonate the end host for all the protocols for which MCGA were used, such as SEND. Note that the privacy features included in the MCGA design prevents correspondent nodes from realizing that the end host identity has been stolen.

In addition, an attacker could propagate different public keys at a high frequency, forcing the end host to generate new MCGAs, resulting if repeated in a DOS attack.

The disclosure of the public keys of the proxy(s) or end host(s) used to build the MCGA does not represent any threat.

Finally, we consider the delegation of the Modifier computation. The configuration at a given end host of a Modifier not compliant with the Sec requirement could break any identity validation performed at other hosts, and consequently, could prevent any communication. However, this event can be easily detected at the end host by a performing the Hash2 computation and certifying that results in the required number of 0 bits. If it were impossible to obtain a valid Modifier, the end host would be forced to compute by itself the modifier, falling back to the current standard procedure.

It is worth to note that the proposed operations do not exchange private keys. An operation requiring such exchange would be the generation of a CGA/MCGA in a different location than the final end host to which it is assigned. The benefits do not outweigh the risks. On one hand, the gain would be small, since a CGA-enabled host is

expected to dynamically sign and validate signatures, and the cost of

generating a key pair is not much higher. On the other hand, there are significant risks, associated to the fact that the compromise of the node generating the keys results in the compromise of the identities of many other systems, and the need for assuring private communications among the parties involved (possibly requiring cryptographic tools, key distribution, etc.)

5.1.2. Threats faced by the configuration servers and proxies

In general, the threats that the configuration servers may face are related with DOS.

An attacker could generate CGA registration requests in order to exhaust the server resources (or to impact on any other operation that depend on the registration of the CGAs). The considerations for MCGAs are similar, although in this case the impact is extended to proxies.

However, the most dangerous attack is bound to malicious requests to compute the Modifier, 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.

We also consider the threats involved in the delivery of the information used to build a MCGA to a SEND proxy. In this case, an attacker could generate fake information in order to exhaust the resources at the proxy. While computing resources are not compromised, since the only check required at the proxy is that its own certified key is included, the state associated to the proxy operation could be exhausted, or proxy operation slowed down.

5.2. Threat Analysis of the Approaches Proposed

Now we discuss the security implications of the use of Router Advertisement and DHCPv6 for performing the proposed operations. To analyze the different scenarios regarding to security in which they can be applied, it is worth to note that the use of CGAs and MCGAs is not bound to SEND enabled networks, since they could be used for identity protection in other protocols such as MIPv6 or SHIM6. Therefore, we can consider different scenarios regarding to security: Router Advertisement with SEND support, Router Advertisement without SEND support, and DHCPv6. For Router Advertisement approaches, only parameter propagation and SEND proxy public key distribution are to be considered.

[5.2.1.](#) Router Advertisement with SEND support

Since the integrity of the RA messages and the identity of their sender are protected by the SEND protocol, protection against malicious nodes generating inappropriate values for the Sec parameter or the Extension Fields is provided. The same protection is provided for the distribution of the public keys of the proxies required for MCGA generation. In this case, a trust anchor must have been configured in the client previously to the reception of the RA messages.

[5.2.2.](#) Router Advertisement without SEND support

In this case there is no protection against the generation of different Sec values, so an attacker could force the generation of CGA with the lowest protection allowed by the standard. It could also force the generation of up to 8 CGA addresses in the end host, wasting resources from the end host. Another attack is related with the association of the public key of an attacker to the MCGA of the end host. DOS attacks based on the request of multiple MCGAs could be issued, although in this case a rate limit set in the client could mitigate the impact.

However, it should be noted that an attacker being able to generate Router Advertisements could also perform Man-In-The-Middle or DOS attacks, by registering itself as a default router for the subnet.

[5.2.3.](#) DHCPv6

All the configuration operations proposed in this document are initiated by the end host. From the point of view of the end host, the difficulty of generating fake responses that were accepted by the end host with the same transaction-id at the precise time is outstanding. However, attacks can be generated by nodes placed in path between the requesting end host and the DHCPv6 server. In particular, non-SEND enabled subnets are more prone to this type of attacks, although SEND does not provide full protection against MITM attacks. In this case, the Sec parameter could be forced to be the lowest, the node could be forced to compute up to 8 CGA addresses, or to compute MCGAs associated with the attacker.

The mechanism based on DHCPv6 is also vulnerable to DOS attacks to the server, such as registration of large number of CGA, or request for Modifier computation.

Internet-Draft [draft-jiang-sendcgaext-cga-config-02.txt](#)

July 2008

Proper use of DHCPv6 autoconfiguration facilities [[RFC3315](#)], such as AUTH option, 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 or MCGA 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.

[6.](#) IANA Considerations

This document defines only the interaction models that involve the Router Advertisement and the DHCPv6 protocol in the CGA generation procedure. The actual DHCPv6 and Router Advertisement extensions are defined in other documents.

[7.](#) Conclusions

This document analyses the requirements for the configuration Cryptographically Generated Addresses (CGA) and Multi-key CGAs. A central administration could configure some parameters such as Sec or Extension Fields to be used by the end hosts in CGA generation. The central administration could notify the availability of CGA proxies, requesting the generation of MCGAs, and propagating the keying material required for MCGAs, and obtaining the end host specific material resulting from this address generation. The computation of the Modifier could also be delegated by an end host to a more appropriate system.

The tools discussed for this performing these interactions are Router Advertisement and the DHCPv6 protocol.

[8.](#) Acknowledgments

The authors would like to thank Marcelo Bagnulo Braun for been involved in the early requirement identification.

[9.](#) References

9.1. Normative References

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

Jiang, et al.

Expires January 10, 2009

[Page 12]

Internet-Draft [draft-jiang-sendcgaext-cga-config-02.txt](#)

July 2008

- [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.
- [RFC4982] M. Bagnulo, "Support for Multiple Hash Algorithms in Cryptographically Generated Addresses (CGAs) ", [RFC4982](#), July 2007.
- [MCGA] J. Kempf, "Secure IPv6 Address Proxying using Multi-Key Cryptographically Generated Address", [draft-kempf-cgaext-ringsig-ndproxy-02](#) (work in progress), August 2007.
- [RIID] S. Krishnan, "Reserved IPv6 Interface Identifiers", [draft-ietf-6man-reserved-ids-00.txt](#) (work in progress), February 2008.

9.2. Informative References

- [RFC2119] S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", [RFC2119](#), March 1997.
- [RFC4866] J. Arkko, C. Vogt, W. Haddad, "Enhanced Route Optimization for Mobile IPv6", [RFC4866](#), May 2007.
- [SHIM6-proto] E. Nordmark, M. Bagnulo, "Shim6: Level 3 Multihoming Shim Protocol for IPv6", [draft-ietf-shim6-proto-10.txt](#) (work in progress), February 2008.

Internet-Draft [draft-jiang-sendcgaext-cga-config-02.txt](#)

July 2008

Author's Addresses

Sheng Jiang
Huawei Technologies Co., Ltd
QuiKe Building, No.9 Xinxu Rd.,
Shang-Di Information Industry Base,
Hai-Dian District, Beijing, P.R. China
100085
Phone: 86-10-82836774
Email: shengjiang@huawei.com

Sam (Zhongqi) Xia
Huawei Technologies Co., Ltd
QuiKe Building, No.9 Xinxu Rd.,
Shang-Di Information Industry Base,
Hai-Dian District, Beijing, P.R. China
100085
Phone: 86-10-82836864
Email: xiazhongqi@huawei.com

Alberto Garcia-Martinez
Universidad Carlos III de Madrid
Av. Universidad 30
Leganes, Madrid 28911
SPAIN
Phone: 34-91-6249500
Email: alberto@it.uc3m.es

Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in

this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

Jiang, et al.

Expires January 10, 2009

[Page 14]

Internet-Draft [draft-jiang-sendcgaext-cga-config-02.txt](#)

July 2008

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Copyright Statement

Copyright (C) The IETF Trust (2008).

This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

