Workgroup: DNSOP Working Group
Internet-Draft:
draft-thomassen-dnsop-dnssec-bootstrapping-01
Published: 23 September 2021
Intended Status: Standards Track
Expires: 27 March 2022
Authors: P. Thomassen
deSEC, Secure Systems Engineering
N. Wisiol
deSEC, Technische Universität Berlin

Automatic Commissioning of New Signers: Solving the DNSSEC Bootstrapping Problem using Authenticated Signals from the Zone's Operator

Abstract

This document describes an authenticated in-band method for automatic signaling of a Child DNS zone's delegation signer information from the zone's DNS operator(s). The zone's registrar or registry may subsequently use this signal for automatic DS record provisioning in the parent zone. The protocol is particularly useful in case of managed DNS providers hosting registrant's domains, where DS provisioning has so far been cumbersome.

The signaling channel is not specific to the DS bootstrapping use case, but equally suitable for announcing other zone-specific information from the DNS Operator in an authenticated fashion. Further potential applications thus include, for example, key exchange between parties in an [RFC8901] multisigner setup.

[Ed note: Text inside square brackets ([]) is additional background information, answers to frequently asked questions, general musings, etc. They will be removed before publication. This document is being collaborated on at https://github.com/desec-io/draft-thomassen-dnsop-dnssec-bootstrapping/. The most recent version of the document, open issues, etc. should all be available there. The authors gratefully accept pull requests.]

Status of This Memo

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 27 March 2022.

Copyright Notice

Table of Contents

1. Introduction

TODO remove: this section is inspired by [RFC7344], Section 1.

The first time a Child DNS Operator signs a zone, they need to communicate the keying material to the Parent. Depending on the desires of the Parent, the Child might send their DNSKEY record, a DS record, or both.

So far, out-of-band methods are typically used to complete the chain of trust. In-band methods exist, in particular based on the CDS and CDNSKEY record types as specified in [RFC7344] and [RFC8078]. However, such communication is only authenticated when performing a rollover of the Child's keys represented in the parent. An authenticated in-band channel for enabling DNSSEC so far has been missing.

How the keying material is conveyed to the Parent during initial DNSSEC bootstrapping depends on the relationship the Child has with the Parent. The communication has to occur between the Child DNS Operator and, depending on the circumstances, the Registry or the Registrar, possibly via the Registrant (for details, see [RFC7344], Appendix A). In many cases, this is a manual process -- and not an easy one. Any manual process is susceptible to mistakes and/or errors. In addition, due to the annoyance factor of the process, involved parties may avoid the process of getting a DS record set published at the Parent.

DNSSEC provides data integrity to information published in DNS; thus, DNS publication can be used to automate maintenance of delegation information. This document describes a method to automate publication of initial DS records for a hitherto insecure delegation.

Readers are expected to be familiar with DNSSEC, including [RFC4033], [RFC4034], [RFC4035], [RFC6781], [RFC7344], and [RFC8078].

1.1. Terminology

The terminology we use is defined in this section. The highlighted roles are as follows:

Child
: The entity on record that has the delegation of the domain from the Parent.

Parent
: The zone that contains the Child's delegation records.

Child DNS Operator
: The entity that maintains and publishes the zone information for the Child DNS.

Parental Agent
: The entity that the Child has a relationship with to change its delegation information.

Signaling Domain(s)
: For any given authoritative nameserver hostname from the Child's NS record set, the hostname prefixed with the label _boot is one of the Signaling Domains for the Child Zone.

Signaling Zone
: The zone which is authoritative for a given Signaling Domain.

Signaling Name
: A name under a Signaling Domain that can be mapped onto the Child zone's name.

Signaling Record
: A DNS record located at a Signaling Name under a Signaling Domain. Signaling Records are used by the Child DNS Operator to publish information about the Child.

CDS/CDNSKEY
: This notation refers to CDS and/or CDNSKEY, i.e., one or both.

Base32hex Encoding
: "Base 32 Encoding with Extended Hex Alphabet" as per [RFC4648].

1.2. Requirements Notation

The key words "**MUST**", "**MUST NOT**", "**REQUIRED**", "**SHALL**", "**SHALL NOT**", "**SHOULD**", "**SHOULD NOT**", "**RECOMMENDED**", "**NOT RECOMMENDED**", "**MAY**", and "**OPTIONAL**" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Signaling

When setting up initial trust, the child generally wants to enable global validation. As long as the child is insecure, DNS answers can be forged. The goal is to promote the child from insecure to secure as soon as reasonably possible by the parent. This means that the period from the child's publication of CDS/CDNSKEY RRset to the parent publishing the synchronized DS RRset should be as short as possible.

This goal is achieved by transferring trust from the Child DNS Operator by publishing an authenticated signal that can be discovered and processed by the Parent. Implementation by Child DNS Operators and Parental Agents is RECOMMENDED.

2.1. Preconditions

If a Child DNS Operator implements the protocol, the following conditions have to be met:

1. Each Signaling Zone MUST be securely delegated, i.e. have a valid DNSSEC chain of trust from the root.

2. The Child DNS Operator MUST publish CDS/CDNSKEY records at the Child's apex, as described in [RFC7344].

[In the course of the bootstrapping protocol, the Parental Agent will fetch the CDS/CDNSKEY records from another source. The second condition ensures that the Parental Agent can validate these records against the customary CDS/CDNSKEY records from the Child. The bootstrapping protocol is thus an extension of the existing CDS/CDNSKEY protocol, and therefore provides strictly stronger guarantees than the traditional model.]

[Requiring presence of CDS/CDNSKEY records in the Child also faciliates simple opt-out by the zone administrator, protects against synchronization errors, and -- if CDS is used, whose value depends on the Child's name -- allows detecting situations of Child name confusion due to hash collisions (see Section 2.2).]

2.1.1. Example

When performing DNSSEC bootstrapping for the Child zone example.co.uk using NS records ns1.example.net and ns2.example.net, the Child DNS Operator

1. needs to ensure that a valid DNSSEC chain of trust exists for the zone(s) that are authoritative for the Signaling Domains _boot.ns1.example.net and _boot.ns2.example.net;

2. publishes CDS/CDNSKEY records at example.co.uk.

2.2. Signaling Names

To publish a piece of information about the Child zone in an authenticated fashion, the Child DNS Operator MUST publish one or more Signaling Records at the Child's Signaling Name under each Signaling Domain.

Signaling Records MUST be accompanied by RRSIG records created with the corresponding Signaling Zone's key(s). The type and contents of these Signaling Records depend on the specific use case as described below.

The Signaling Name MUST consist of the following two labels:

1. the first label of the Child name;

2. a label equal to the SHA-256 hash digest of the fully qualified domain name of the Child's immediate ancestor in the DNS tree (one level up), using wire format for the hash input and "Base 32 Encoding with Extended Hex Alphabet" as specified in [RFC4648] for the output. Trailing padding characters ("=") MUST be dropped.

Note that the "fully qualified domain name of the Child's immediate ancestor in the DNS tree" coincides with the Parent's FQDN only when the delegation is directly (one level) under the Parent's apex. For deeper delegations, it also contains the labels between the Parent and the Child.

[The purpose of the hash function is to avoid the possibility of exceeding the maximum length of a DNS name, and to normalize the number of labels in a Signaling Name. The encoding choice is like in NSEC3, except that SHA-256 is used instead of SHA-1. This is to prevent other tenants in shared hosting environments from creating collisions.]

[Prefixing the first label verbatim minimizes the number of hash calculations that need to be performed by the Child DNS Operator and the Parental Agent, and also facilitates discovery of unprocessed Signaling Records by the Parental Agent by means of NSEC walking the Signaling Domain. (If the first label was part of the hash, the Parental Agent would not be able to infer the Child's name.)]

[**Example code** (Python, with dnspython package):

```
from base64 import b32encode
from hashlib import sha256

import dns.name
from dns.rdtypes.ANY.NSEC3 import b32_normal_to_hex

child = 'example.co.uk.'
prefix, suffix = child.split('.', 1)
suffix_wire_format = dns.name.from_text(suffix).to_wire()
suffix_digest = sha256(suffix_wire_format).digest()
suffix_digest = b32encode(suffix_digest).translate(b32_normal_to_hex).rs
signaling_name = prefix + '.' + suffix_digest.lower().decode()
print(signaling_name)
# >>> 'example.bge2bvlnqt4ei2oq3v9nr8a0lh9nkf6b4lh6c3j51k5kd67helmg'
```

]

3. Bootstrapping a DNSSEC Delegation

3.1. Signaling Intent to Act as the Child's Signer

To announce its willingness to act as the Child's delegated signer, the Child DNS operator co-publishes the Child's CDS/CDNSKEY records at the corresponding Signaling Name under each Signaling Domain as defined in Section 2.2.

Previous use of CDS/CDNSKEY records is specified at the apex only ([RFC7344], Section 4.1). This protocol extends the use of these record types at non-apex owner names for the purpose of DNSSEC bootstrapping. To exclude the possibility of semantic collision, there MUST NOT be a zone cut at a Signaling Name.

Unlike the CDS/CDNSKEY records at the Child's apex, Signaling Records MUST be signed with the corresponding Signaling Zone's key(s). Their contents MUST be identical to the corresponding records published at the Child's apex.

3.1.1. Example

For the purposes of bootstrapping the Child zone example.co.uk with NS records ns1.example.net and ns2.example.net, the required Signaling Domains are _boot.ns1.example.net and _boot.ns2.example.net.

In the zones containing these domains, the Child DNS Operator publishes the Child's CDS/CDNSKEY records at the names

```
example.bge2bvlnqt4ei2oq3v9nr8a0lh9nkf6b4lh6c3j51k5kd67helmg._boot.ns1.e
example.bge2bvlnqt4ei2oq3v9nr8a0lh9nkf6b4lh6c3j51k5kd67helmg._boot.ns2.e
```

where example.bge2bvlnqt4ei2oq3v9nr8a0lh9nkf6b4lh6c3j51k5kd67helmg is derived from the DNS Child Zone's name example.co.uk as described in Section 2.2. The records are accompanied by RRSIG records created using the key(s) of the respective Signaling Zone.

3.2. Steps Taken by the Parental Agent

To complete the bootstrapping process, Parental Agents implementing this protocol can act based upon a number of triggers (see Section 3.4). Once trigger conditions are fulfilled, the Parental Agent, knowing both the Child zone name and its NS hostnames, MUST

1. verify that the Child is not currently securely delegated;

2. query the CDS/CDNSKEY records at the Child zone apex directly from each of the authoritative servers as listed in the NS record set;

3. query the CDS/CDNSKEY records located at each of the Signaling Names using a trusted validating DNS resolver;

4. check (separately by record type) that all record sets retrieved in Steps 2 and 3 have equal contents;

If the above steps succeeded without error, the Parental Agent MUST construct a tentative DS record set either by copying the CDS record contents or by computing DS records from the CDNSKEY record set, or by doing both (i.e. amending the set of records copied from the CDS record set).

The Parental Agent then MUST verify that for each signature algorithm present, (at least) one of the keys referenced in the tentative DS record set signs the Child's DNSKEY record set. [TODO Which other checks are needed to not break anything?]

If this is the case, the Parental Agent SHOULD publish the DS record set in the Parent zone, so as to secure the Child's delegation.

If, however, an error condition occurs, in particular:

* in Step 1: the Child is already securely delegated;

* in Step 2: any failure during the retrieval of the CDS/CDNSKEY records located at the Child apex from any of the authoritative nameservers, with an empty record set qualifying as a failure;

* in Step 3: DNS resolution failure during retrieval of CDS/CDNSKEY records from any Signaling Name, including failure of DNSSEC validation or unauthenticated data (AD bit not set);

*in Step 4: inconsistent responses;

*the tentative DS record set includes a signature algorithm without referencing a key of that algorithm which signs the Child's DNSKEY record set;

the Parental Agent MUST abort the procedure.

[This level of rigor is needed for various reasons, including that it prevents one operator from screwing up the zone in a multi-homed setup (where several operators serve the same zone).]

3.2.1. Example

To bootstrap the Child zone example.co.uk using NS records ns1.example.net and ns2.example.net, the Parental Agent

1. checks that the Child zone is not yet securely delegated;

2. queries CDS/CDNSKEY records for example.co.uk directly from ns1.example.net and ns2.example.net;

3. queries the CDS/CDNSKEY records located at the Signaling Names (see Section 2.2)

example.bge2bvlnqt4ei2oq3v9nr8a0lh9nkf6b4lh6c3j51k5kd67helmg._boot.ns1.e
example.bge2bvlnqt4ei2oq3v9nr8a0lh9nkf6b4lh6c3j51k5kd67helmg._boot.ns2.e

4. checks that the CDS/CDNSKEY record sets retrieved in Steps 2 and 3 agree across responses.

The Parental Agent then publishes a DS record set according to the information retrieved in the previous steps.

3.3. Opt-out

As a special case of Step 2 failure, the Child MAY opt out from DNSSEC bootstrapping by publishing a CDS/CDNSKEY record with algorithm 0 and other fields as specified in [RFC8078], Section 4, at its apex.

This mechanism is workable without regard to whether the Child zone's signatures are managed by the Child DNS Operator or by the zone owner, and without regard to what the Child DNS Operator decides to signal under the Signaling Domain.

3.4. Triggers

[Clarity of this section needs to be improved.]

Parental Agents SHOULD trigger the procedure described in Section 3.2 once one of the following conditions is fulfilled:

* The Parental Agent receives a new or updated NS record set for a Child;

* The Parental Agent encounters Signaling Records for its Children during a scan (e.g. daily) of known Signaling Domains (derived from the NS records used in its delegations).

  To perform such a scan, the Parental Agent iterates over some or all of its delegations and strips the first label off each one to construct the set of immediate ancestors of its children. (For delegations one level below the Parent, such as second-level domain registrations, this will simply be the Parent's name.) The Parental Agent then uses these names to compute the second label of the Signaling Names. The scan is completed by either

  - performing a targeted NSEC walk starting one level below the Signaling Domain, at the label that encodes the Child's ancestor; or

  - by performing a zone transfer of the zone containing the (relevant part of the) Signaling Domain, if the Signaling Zone operator allows it, and iterating over its contents.

  The Child's name is constructed by prepending the first label of the encountered Signaling Names to the ancestor from which the Signaling Name's second label was computed;

* The Parental Agent performs an active (e.g. daily) scan by opportunistically querying the Signaling Records for some or all of its delegations;

* Any other condition as deemed appropriate by local policy.

4. Operational Recommendations

4.1. Child DNS Operator

Signaling Domains SHOULD be delegated as zones of their own, so that the Signaling Zone's apex coincides with the Signaling Domain (such as _boot.ns1.example.net). While it is permissible for the Signaling Domain to be contained in a Signaling Zone of fewer labels (such as example.net), a zone cut ensures that bootstrapping activities do not require modifications of the zone containing the nameserver hostname.

In addition, Signaling Zones SHOULD use NSEC to allow consumers to efficiently discover pending bootstrapping operations by means of

zone walking (see Section 3.4). This is especially useful for bulk processing after a Child DNS Operator has enabled the protocol.

To keep the size of the Signaling Zones minimal, Child DNS Operators SHOULD remove Signaling Records which are found to have been acted upon. This is particularly important when the Child DNS Operator allows Parental Agents to perform scans of the Signaling Zone, either by allowing zone transfers or by permitting zone walks via NSEC, so that bulk processing remains efficient.

4.2. Parental Agent

It is RECOMMENDED to perform queries within Signaling Domains (Section 3.2) with an (initially) cold resolver cache as to retrieve the most current information regardless of TTL. (When a batch job is used to attempt bootstrapping for a large number of delegations, the cache does not need to get cleared in between.)

[It is expected that Signaling Records have few consumers only, so that caching would not normally have a performance benefit. On the other hand, perhaps it is better to RECOMMEND low TTLs instead?]

5. Implementation Status

Note to the RFC Editor: please remove this entire section before publication.

5.1. Child DNS Operator-side

* Knot DNS supports manual creation of non-apex CDS/CDNSKEY/DNSKEY records.

* PowerDNS supports manual creation of non-apex CDS/CDNSKEY/DNSKEY records.

* Proof-of-concept Signaling Domains with several thousand Signaling Names exist at _boot.ns1.desec.io and _boot.ns2.desec.org. Signaling Names can be discovered via NSEC walking.

* A tool to automatically generate signaling records for bootstrapping purposes is under development by the authors.

5.2. Parental Agent-side

* A tool to retrieve and process Signaling Records for bootstrapping purposes, either directly or via zone walking, is available at https://github.com/desec-io/dsbootstrap. The tool implements outputs the validated DS records which then can be added to the parent zone.

6. Security Considerations

Thoughts:

* We use at least one established chain of trust (via the secure delegations of the zones containing the NS hostnames). As a result,
  - communication is authenticated;
  - process is immediate (no need for observing CDS/CDNSKEY records via TCP for several days);
  - an active on-wire attacker cannot tamper with the delegation.
* The security level of the method is strictly higher than the "accept CDS/CDNSKEY after a while"-approach that is already in use at several ccTLD registries ("Accept after Delay", [RFC8078], Section 3.3). This is because the method described here adds stronger guarantees, but removes nothing.
* Actors in the chain(s) of trust of the zone(s) used for bootstrapping (the DNS Operator themselves, plus entities further up in the chain) can undermine the protocol. However,
  - that's also possible in the case of CDS/CDNSKEY (see previous point);
  - if the Child DNS Operator doesn't control the zones in which its NS hostnames live (including their nameservers' A records) because the path from the root is untrusted, you probably don't want to trust that operator as a whole;
  - when bootstrapping is done upon receipt of a new NS record set, the window of opportunity is very small;
  - mitigation exists by diversifying e.g. the nameserver hostname's TLDs, which is advisable anyways;
  - correct bootstrapping is easily monitored by the Child DNS operator.
* Prevention of accidental misprovisioning / enforcing explicit provisioning:
  - In case of a hash collision, two distinct child zones may be associated with the same signaling name so that their keys may get mixed up. While not currently feasible, malicious customers in shared hosting environments may attempt to produce such a collision. Is it worth mitigating this by

introducing a salt, e.g. stored in a TXT record located at the Signaling Domain? (In case of a collision, one can set a new salt.)

7. IANA Considerations

TODO: reserve _boot?

This document has no IANA actions.

8. Acknowledgements

Thanks to Brian Dickson, John R. Levine, and Ondřej Caletka for reviewing draft proposals and offering comments and suggestions.

Thanks also to Steve Crocker, Hugo Salgado, and Ulrich Wisser for early-stage brainstorming.

9. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, DOI 10.17487/RFC4033, March 2005, <https://www.rfc-editor.org/info/rfc4033>.

[RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, DOI 10.17487/RFC4034, March 2005, <https://www.rfc-editor.org/info/rfc4034>.

[RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, DOI 10.17487/RFC4035, March 2005, <https://www.rfc-editor.org/info/rfc4035>.

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006, <https://www.rfc-editor.org/info/rfc4648>.

[RFC6781] Kolkman, O., Mekking, W., and R. Gieben, "DNSSEC Operational Practices, Version 2", RFC 6781, DOI

10.17487/RFC6781, December 2012, <https://www.rfc-editor.org/info/rfc6781>.

[RFC7344] Kumari, W., Gudmundsson, O., and G. Barwood, "Automating DNSSEC Delegation Trust Maintenance", RFC 7344, DOI 10.17487/RFC7344, September 2014, <https://www.rfc-editor.org/info/rfc7344>.

[RFC7477] Hardaker, W., "Child-to-Parent Synchronization in DNS", RFC 7477, DOI 10.17487/RFC7477, March 2015, <https://www.rfc-editor.org/info/rfc7477>.

[RFC8078] Gudmundsson, O. and P. Wouters, "Managing DS Records from the Parent via CDS/CDNSKEY", RFC 8078, DOI 10.17487/RFC8078, March 2017, <https://www.rfc-editor.org/info/rfc8078>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8901] Huque, S., Aras, P., Dickinson, J., Vcelak, J., and D. Blacka, "Multi-Signer DNSSEC Models", RFC 8901, DOI 10.17487/RFC8901, September 2020, <https://www.rfc-editor.org/info/rfc8901>.

Appendix A. Possible Extensions

The mechanism described in Section 2.2 provides a public, authenticated, in-band, unidirectional channel through which the Child DNS Operator can publish information on the zones it serves.

By provisioning other types of Signaling Records, the Child DNS Operator can therefore convey signals that pertain to use cases other than bootstrapping a DNSSEC delegation.

A.1. Multi-Signer Setups: Onboarding a Signing Party

[RFC8901] describes multi-signer models in which several Child DNS Operators serve the same Child zone. In one of these scenarios (Model 2, [RFC8901], Section 2.1.2), each Child DNS Operator holds a unique KSK set and ZSK set to sign the zone.

To ensure smooth resolution of Child zone queries, this scheme demands that participating Child DNS Operators import the ZSK sets of the other providers into their DNSKEY RRset. Further, each operator's KSK(s) need to be included in the DS record set at the delegation point in the Parent zone. When a new Child DNS Operator is joining the scheme, these synchronization processes have to occur

before the new operator's nameserver hostnames are included in the Child's NS record set.

So far, it has been assumed that the KSK and ZSK extraction and provisioning would happen through some proprietary API at each DNS operator ([RFC8901], Section 9). We now describe how a Child DNS Operator can instead use Signaling Records to make its own set of DNSKEY records available for querying by other signing parties, so that they can retrieve, validate, and process them.

A.1.1. Signaling Records

Given a Child zone example.co.uk that is already securely delegated with authoritative nameservers ns1.example.net and ns2.example.net, we consider how a new Child DNS Operator using nameservers ns3.example.org and ns4.example.org can distribute its DNSKEY record set to the existing signing parties, in order to join the multi-signer group.

The Signaling Domains corresponding to the new Child DNS Operator's nameservers are _boot.ns3.example.org and _boot.ns4.example.org.

In the zones containing these domains, the new Child DNS Operator publishes a DNSKEY record set containing the keys used by the operator when operating the Child zone, at the Signaling Names

example.bge2bvlnqt4ei2oq3v9nr8a0lh9nkf6b4lh6c3j51k5kd67helmg._boot.ns3.e
example.bge2bvlnqt4ei2oq3v9nr8a0lh9nkf6b4lh6c3j51k5kd67helmg._boot.ns4.e

where the first label is calculated as described in Section 2.2. The records are accompanied by RRSIG records created using the key(s) of the respective Signaling Zone.

Note that DNSKEY records are not restricted to apex owner names ([RFC4035], Section 2.1). However, only apex DNSKEY records are used for DNSSEC validation ([RFC4035], Section 5). As Signaling Names do not occur on zone cuts (see Section 3.1), the use of DNSKEY records described here does not interfere with existing DNSKEY uses.

A.1.2. Import

With the Signaling Records in place, an algorithm similar to the one given in Section 3.2 can be used to query and validate the joining operator's DNSKEY set. The required steps can either be taken autonomously by each participating operator (query, validate, update local zone copy with imported information), or be subject to central coordination using "Multisigner Controller" tooling which interfaces with each operator.

The new KSKs can then be added to the delegation's DS record set as described in [RFC8901], Section 8 (i.e. via an [RFC7344] rollover using CDS/CDNSKEY records), followed by the inclusion of the new ZSKs in the other operators' DNSKEY record sets. Similarly, the new operator can import the other operators' DNSKEYs into its local copy of the Child zone (either autonomously, or via central coordination).

[Note that the DNSKEY record set in the Child zone contains keys from all operators, whereas the DNSKEY record set published under the Signaling Domain is restricted to keys actively used by the publishing operator.]

After convergence on the served DNSKEY record sets has been achieved, the joining process is completed by amending the Child's NS record set to include the new operator's authoritative nameservers, followed by a corresponding update of the NS delegation records at the Parent (e.g. using CSYNC [RFC7477]).

Appendix B. Change History (to be removed before final publication)

*draft-thomassen-dnsop-dnssec-bootstrapping-01

> Add section on Triggers.
>
> Clarified title.
>
> Improved abstract.
>
> Require CDS/CDNSKEY records at the Child.
>
> Reworked Signaling Name scheme.
>
> Recommend using cold cache for consumption.
>
> Updated terminology (replace "Bootstrapping" by "Signaling").
>
> Added NSEC recommendation for Bootstrapping Zones.
>
> Added multi-signer use case.
>
> Editorial changes.

*draft-thomassen-dnsop-dnssec-bootstrapping-00

> Initial public draft.

Authors' Addresses

Peter Thomassen

deSEC, Secure Systems Engineering
Berlin
Germany

Email: peter@desec.io

Nils Wisiol
deSEC, Technische Universität Berlin
Berlin
Germany

Email: nils@desec.io