DNSOP Working Group
Internet-Draft
Intended status: Best Current Practice
Expires: 23 November 2026

S. Sheng

P. Thomassen
deSEC

22 May 2026

Operational Recommendations for DNSSEC Delegation Signer (DS) Automation
draft-ietf-dnsop-ds-automation-09

Abstract

Enabling support for automatic acceptance of DNSSEC Delegation Signer (DS) parameters from the Child DNS operator (via RFCs 7344, 8078, 9615) requires the parental agent, often a registry or registrar, to make a number of technical decisions around acceptance checks, error and success reporting, and multi-party issues such as concurrent updates. This document describes recommendations about how these points are best addressed in practice.

Discussion Venues

This note is to be removed before publishing as an RFC.

Discussion of this document takes place on the Domain Name System Operations Working Group mailing list (dnsop@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/dnsop/.

Source for this draft and an issue tracker can be found at https://github.com/peterthomassen/draft-shetho-dnsop-ds-automation.

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 https://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 23 November 2026.

Table of Contents

1. Introduction

[RFC7344], [RFC8078], and [RFC9615] automate DNSSEC [RFC9364] delegation trust maintenance by having the child publish Child DS (CDS) and/or Child DNSKEY (CDNSKEY) records which indicate the delegation's desired DNSSEC parameters ("DS automation").

Parental Agents using these protocols have to make a number of technical decisions relating to issues of acceptance checks, timing, error reporting, locks, etc. Additionally, when using the registrant-registrar-registry (RRR) model (as is common amongst top-level domains), both the registrar and the registry can effect parent-side changes to the delegation. In such a situation, additional opportunities for implementation differences arise.

Not all existing DS automation deployments have made the same choices with respect to these questions, leading to somewhat inconsistent behavior. From the perspective of a domain holder with domain names under various TLDs, this may be unexpected and confusing.

In the following sections, operational questions are first raised and answered with the corresponding recommendations. Each section is concluded with an analysis of its recommendations and related considerations. A combined view of the recommendations from all sections is given in Appendix A.

Readers are expected to be familiar with DNSSEC [RFC9364][RFC9615][RFC9859].

The core issues addressed in the document are derived from Section 4.4 of [SAC126]. Readers are referred to this report for additional background.

1.1. 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. Terminology

The term Parental Agent is used as defined in Section 1.1 of [RFC7344]. The document also uses terms defined in [RFC9499], in particular:

* DNS operator
* Registry
* Registrant
* Registrar

In addition, the document makes use of the following terms:

Child zone: DNS zone whose delegation is in the Parent zone.

Child (DNS operator): DNS operator responsible for a Child zone.

Parent zone: DNS zone that holds a delegation for a Child zone.

Parent: The operator responsible for a Parent zone and, thus, involved with the maintenance of the delegation's DNSSEC parameters (in particular, the acceptance of these parameters and the publication of corresponding DS records).

RRR Model: The registrant-registrar-registry (RRR) interaction framework, where registrants interact with a registrar to register and manage domain names, and registrars interact with the domain's registry for the provision and management of domain names on the registrant's behalf. This model is common amongst TLDs.

3. Recommendations for Deployments of DS Automation

The guidelines for deploying DS automation set out in this document are meant to achieve more uniform treatment across suffixes — minimizing user surprise and providing baseline safety and uniformity of behavior. They are also intended to prevent disruption of DNS and DNSSEC functionality. At a minimum, compliance with this RFC requires support for both DNSSEC bootstrapping [RFC9615] and subsequent updates [RFC7344], [RFC8078] under the implementation guidance below.

The recommendations optimize interoperability and safety. In certain cases, local policy may take precedence, such as when a registry is subjected to national cryptographic policy requirements. However, not following any requirements designated with the "SHOULD" key word will generally lead to undesirable effects of ambiguity and interoperability issues. When implementing these recommendations, operators MUST mitigate issues arising from any particular deviation.

Registries with additional requirements on DS update checks MAY implement any additional checks in line with local policy.

4. Acceptance Checks and Safety Measures

This section provides recommendations to address the following operational questions:

* What kind of acceptance checks should be performed on DS parameters?

* Should these checks be performed upon acceptance or also continually when in place?

* How do TTLs and caching impact DS provisioning? How important is timing in a child key change?

* Are parameters for DS automation best conveyed as CDNSKEY or CDS records, or both?

4.1. Recommendations

1. Entities performing automated DS maintenance MUST verify:

    a. the unambiguous intent of each DS bootstrapping or update request as per [I-D.ietf-dnsop-cds-consistency], by checking its consistency both

        * between any published CDS and CDNSKEY records, and

* across all authoritative nameservers in the delegation,

and

b. that the resulting DS record set would allow continued DNSSEC validation if deployed,

and cancel the update if the verifications do not succeed.

2. Parent-side entities (such as registries) SHOULD reduce a DS record set's TTL to a value between 515 minutes when a new set of records is published, and restore the previous (or, if unavailable, default) TTL value at a later occasion (but not before the previous DS RRset's TTL has expired).

3. DNS operators MUST publish both CDNSKEY and CDS records (unless the parent's preference is known), and follow best practice for the choice of hash digest type [DS-IANA].

4.2. Analysis

4.2.1. Continuity of Resolution

To maintain the basic resolution function, it is critical to avoid deployment of flawed DS record sets in the parent zone. It is therefore necessary for the Parent to verify that the DS record set resulting from an automated (or even manual) update does not break DNSSEC validation if deployed, and otherwise cancel the update.

This is best achieved by:

1. verifying that consistent CDS/CDNSKEY responses are served by all of the delegation's nameservers [I-D.ietf-dnsop-cds-consistency];

2. verifying that the resulting DS Resource Record set (RRset) does not break the delegation if applied ([RFC7344], Section 4.1), i.e., it provides at least one valid path for validators to use ([RFC6840], Section 5.11). This is the case if the child's DNSKEY RRset has a valid RRSIG signature from a key that is referenced by at least one DS record, with the digest type and signing algorithm values designated as "RECOMMENDED" or "MUST" in the "Use for DNSSEC Validation" columns of the relevant IANA registries ([DS-IANA] and [DNSKEY-IANA]). Note that these checks need not be enforced when provisioning DS records manually in order to enable the use other digest types or algorithms for potentially non-interoperable purposes.

Even without an update being requested, Parents may occasionally check whether the current DS contents would still be acceptable if they were newly submitted in CDS/CDNSKEY form (see Section 4). Any failures — such as a missing DNSKEY due to improper rollover timing ([RFC6781], Section 4.1), or changed algorithm requirements — can then be communicated in line with Section 5, without altering or removing the existing DS RRset.

4.2.2. TTLs and Caching

To further reduce the impact of any misconfigured DS record set — be it from automated or from manual provisioning — the option to quickly roll back the delegation's DNSSEC parameters is of great importance. This is achieved by setting a comparatively low TTL on the DS record set in the parent domain, at the cost of reduced resiliency against nameserver unreachability due to the earlier expiration of cached records. The availability risk can be mitigated by limiting such TTLs to a brief time period after a change to the DS configuration, during which rollbacks are most likely to occur.

Registries therefore should significantly lower the DS RRset's TTL for some time following bootstrapping or an update. Pragmatic values for the reduced TTL value range between 515 minutes. Using values below 5 minutes risks excessive queries, and using values greater than 15 minutes may impact recovery from operational mistakes.

Note that recent measurements have demonstrated low TTLs like the above to have negligible impact on the overall load of a registry's authoritative nameserver infrastructure [LowTTL].

The reduction should be in effect at least for a couple of days and until the previous DS record set has expired from caches, that is, the period during which the low-TTL is applied typically will significantly exceed the normal TTL value. When using the Extensible Provisioning Protocol (EPP) [RFC5730], the domain <info> command described in Section 2.1.1.2 of [RFC9803] can be used by the registrar to obtain the registry's TTL policy.

While this approach enables quick rollbacks, timing of the desired DS update process itself is largely governed by the previous DS RRset's TTL, and therefore does not generally benefit from an overall speed-up. Note also that nothing is gained from first lowering the TTL of the old DS RRset: such an additional step would, in fact, require another wait period while resolver caches adjust. For the sake of completeness, there likewise is no point to increasing any DS TTL values beyond their normal value.

4.2.3. CDS vs. CDNSKEY

DS records can be generated from information provided either in DS format (CDS) or in DNSKEY format (CDNSKEY). While the format of CDS records is identical to that of DS records (so the record data be taken verbatim), generation of a DS record from CDNSKEY information involves computing a hash.

Whether a Parent processes CDS or CDNSKEY records depends on their preference:

* Processing (and storing) CDNSKEY information allows the Parent to control the choice of hash algorithms. The Parent may then unilaterally regenerate DS records with a different choice of hash algorithm(s) whenever deemed appropriate.

* Processing CDS information allows the Child DNS operator to control the hash digest type used in DS records, enabling the Child DNS operator to deploy (for example) experimental hash digests and removing the need for registry-side changes when additional digest types become available.

The need to make a choice in the face of this dichotomy is not specific to DS automation: even when DNSSEC parameters are relayed to the Parent through conventional channels, the Parent has to make some choice about which format(s) to accept.

As there exists no protocol for Child DNS operators to discover a Parent’s input format preference, interoperability requires publication of both CDNSKEY as well as CDS records, in line with Section 5 of [RFC7344]. The choice of hash digest type should follow current best practice [DS-IANA].

Publishing the same information in two different formats is not ideal. Still, it is much less complex and costly than burdening the Child DNS operator with discovering each Parent’s current policy. Also, it is very easily automated. Operators should ensure that published RRsets are consistent with each other.

If both RRsets are published, Parents are expected to verify consistency between them by verifying that they refer to the same set of keys [I-D.ietf-dnsop-cds-consistency]. By not second-guessing inconsistencies (such as by RRset recency) and instead rejecting them, responsibility to clearly express each update request is placed on the Child DNS operator.

CDS records need only be considered for CDNSKEY consistency when their digest type field is designated as "MUST" in the "Implement for DNSSEC Delegation" column of the "Digest Algorithms" registry [DS-IANA]. Consistency of records with other digest types need not be verified, especially when the digest type is unsupported; such records can be ignored. Note that this does not imply a restriction on the DS hash digest types: if no inconsistencies are found, the parent can publish DS records with whatever digest type(s) it prefers.

5. Reporting and Transparency

This section provides recommendations to address the following operational question:

* Should a failed (or even successful) DS update trigger a notification to anyone?

5.1. Recommendations

1. For certain DS updates (see analysis (Section 5.2)) and for DS deactivation, relevant points of contact known to the parent-side entity (registry or registrar) SHOULD be notified.

2. For error conditions, the child DNS operator and the domain’s technical contact (if applicable) SHOULD be notified first. The registrant SHOULD NOT be notified unless the problem persists for a prolonged amount of time (e.g., three days).

3. Child DNS operators SHOULD be notified of errors using a report query [RFC9567] to the agent domain as described in Section 4 of [RFC9859]. Notifications to humans (domain holder) will be performed in accordance with the communication preferences established with the parent-side entity. The same condition SHOULD NOT be reported unnecessarily frequently to the same recipient.

4. In the RRR model, registries performing DS automation SHOULD inform the registrar of any DS record changes via the EPP Change Poll Extension [RFC8590] or a similar channel.

5. The currently active DS configuration SHOULD be made accessible to the registrant (or their designated party) through the customer portal available for domain management. The DS update history MAY be made available in the same way.

5.2. Analysis

When accepting or rejecting a DS update, it cannot be assumed that relevant parties are aware of what's happening. For example, a registrar may not know when an automatic DS update is performed by the registry. Similarly, a Child DNS operator may not be aware when their CDS/CDNSKEY RRsets are out of sync across nameservers, causing them to be ignored.

To help involved parties act appropriately and in a timely manner, entities performing automated DS maintenance should report on conditions they encounter. The following success situations may be of particular interest:

1. A DS RRset has been provisioned
    a. manually;
    b. due to commencing DS automation (either via DNSSEC bootstrapping, or for the first time after a manual change; see Section 7);
    c. automatically, as an update to an existing DS RRset that had itself been automatically provisioned.
2. The DS RRset has been removed
    a. manually;
    b. automatically, using a delete signal ([RFC8078], Section 4).

In addition, there are error conditions worthy of being reported:

3. A pending DS update cannot be applied due to an error condition. There are various scenarios where an automated DS update might have been requested, but can't be fulfilled. These include:
    a. The new DS record set would break validation/resolution or is not acceptable to the Parent for some other reason (see Section 4).
    b. A lock prevents DS automation (see Section 6).
4. No DS update is due, but it was determined that the Child zone is no longer compatible with the existing DS record set (e.g., DS RRset only references non-existing keys).

In these latter two cases, the entity performing DS automation would be justified to attempt communicating the situation. Potential recipients are:

* Child DNS operator, preferably by making a report query [RFC9567] to the agent domain listed in the EDNS0 Report-Channel option of the DS update notification that triggered the DS update ([RFC9859], Section 4), or else via email to the address contained in the child zone's SOA RNAME field (see Sections 3.3.13 and 8 of [RFC1035]);

* Registrar (if DS automation is performed by the registry);

* Registrant (domain holder; in non-technical language, such as "DNSSEC security for your domain has been enabled and will be maintained automatically") or technical contact, in accordance with the communication preferences established with the parent-side entity.

For manual updates (case 1a), commencing DS automation (case 1b), and deactivating DNSSEC (case 2), it seems worthwhile to notify both the domain's technical contact (if applicable) and the registrant. This will typically lead to one notification during normal operation of a domain. (Case 1c, the regular operation of automation, is not an interesting condition to report to a human.)

For error conditions (cases 3 and 4), the registrant need not always be involved. It seems advisable to first notify the domain's technical contact and the DNS operator serving the affected Child zone, and only if the problem persists for a prolonged amount of time (e.g., three days), notify the registrant.

When the RRR model is used and the registry performs DS automation, the registrar should always stay informed of any DS record changes, e.g., via the EPP Change Poll Extension [RFC8590].

Overly frequent reporting of the same condition to the same recipient is discouraged (e.g., no more than twice in a row). For example, when CDS and CDNSKEY records are inconsistent and prevent DS initialization, the registrant may be notified twice. Additional notifications may be sent with some back-off mechanism (in increasing intervals).

The registrant (or their designated party) should be able to retrieve the current DS configuration through the customer portal available for domain management. Failure to provide the registrant a means to inspect the current configuration after it has been changed may hinder recovery from operational incidents because the registrant may have out-of-date information.

Ideally, the history of DS updates would also be available. However, due to the associated state requirements and the lack of direct operational impact, implementation of this is optional. If supported by the registry, the DS TTL currently in effect can be obtained using the RDAP TTL extension [I-D.ietf-regext-rdap-ttl-extension].

For troubleshooting, dispute resolution, and post-incident analysis, it is instrumental for the Parental Agent to retain structured records of DS automation decisions, including timestamp, triggering CDS/CDNSKEY RRsets, notification channel, authoritative nameservers consulted, verification results, decision outcome, and the applied DS RRset or cancellation reason.

6. Registration Locks

This section provides recommendations to address the following operational question:

* How does DS automation interact with other registration state parameters, such as registration locks?

6.1. Recommendations

1. To secure ongoing operations, automated DS maintenance MUST NOT be suspended based on a registrar update lock alone (such as EPP status clientUpdateProhibited [RFC5731]).

2. When performed by the registry, automated DS maintenance MUST NOT be suspended based on a registry update lock alone (such as EPP status serverUpdateProhibited [RFC5731]).

6.2. Analysis

Registries and registrars can set various types of locks for domain registrations, usually upon the registrant’s request. An overview of standardized locks using EPP, for example, is given in Section 2.3 of [RFC5731]. Some registries may offer additional (or other) types of locks whose meaning and set/unset mechanisms are defined according to a proprietary policy.

While some locks clearly should have no impact on DS automation (such as transfer or deletion locks), other types of locks, in particular "update locks", deserve a closer analysis.

6.2.1. Registrar vs. Registry Lock

A registrar-side update lock (such as clientUpdateProhibited in EPP) protects against various types of accidental or malicious change (like unintended changes through the registrar's customer portal). Its security model does not prevent the registrar's (nor the registry's) actions. This is because a registrar-side lock can be removed by the registrar without an out-of-band interaction.

Under such a security model, no tangible security benefit is gained by preventing automated DS maintenance based on a registrar lock alone, while preventing it would make maintenance needlessly difficult. It is therefore not justified to suspend automation when such a lock is present.

When a registry-side update lock is in place, the registrar cannot apply any changes (for security or delinquency or other reasons). However, it does not protect against changes made by the registry itself. This is exemplified by the serverUpdateProhibited EPP status, which demands only that the registrar's "[r]equests to update the object [...] MUST be rejected" (Section 2.3 of [RFC5731]). This type of lock therefore precludes DS automation by the registrar, while registry-side automation remains unaffected.

DS automation by the registry further is consistent with Section 2.3 of [RFC5731], which explicitly notes that an EPP server (registry) may override status values set by an EPP client (registrar), subject to local server policies. The risk that DS changes from registry-side DS automation might go unnoticed by the registrar is mitigated by sending change notifications to the registrar; see Recommendation 4 of Section 5.

6.2.2. Detailed Rationale

Pre-DNSSEC, it was possible for a registration to be set up once, then locked and left alone (no maintenance required). With DNSSEC comes a change to this operational model: the configuration may have to be maintained in order to remain secure and operational. For example, the Child DNS operator may switch to another signing algorithm if the previous one is no longer deemed appropriate, or roll its Secure Entry Point (SEP) key for other reasons. Such changes entail updating the delegation's DS records.

If authenticated, these operations do not qualify as accidental or malicious change, but as legitimate and normal activity for securing ongoing operation. The CDS/CDNSKEY method provides an automatic, authenticated means to convey DS bootstrapping and update requests [RFC9615][RFC7344]. The resulting operation is subject to the parent's acceptance checks; in particular, it is not applied when it would break the delegation (see Section 4).

Given that registrar locks protect against unintended changes (such as through the customer portal) while not preventing actions done by the registrar (or the registry) itself, such a lock is not suitable for defending against actions performed illegitimately by the registrar or registry (e.g., due to compromise). Any attack on the registration data that is feasible in the presence of a registrar lock is also feasible regardless of whether DS maintenance is done automatically; in other words, DS automation is orthogonal to the attack vector that a registrar lock protects against.

Considering that automated DS bootstrapping and update requests are required to be authenticated and validated for correctness, honoring such requests — while in the registrant's interest — comes with no additional associated risk when compared to other authenticated update methods. Suspending automated DS maintenance therefore is not justified.

Following this line of thought, at the time of document writing some registries (e.g., .ch/.cz/.li) perform automated DS maintenance even when an "update lock" is in place. Registries offering proprietary locks should carefully consider for each lock whether its scope warrants suspension.

In case of a domain not yet secured with DNSSEC, automatic DS initialization is not required to maintain ongoing operation; however, authenticated DNSSEC bootstrapping [RFC9615] might be requested. Besides being in the interest of security, the fact that a Child is requesting DS initialization through an authenticated method expresses the registrant's intent to have the delegation secured.

Further, some domains are equipped with an update lock by default. Not honoring DNSSEC bootstrapping requests then imposes an additional burden on the registrant, who has to unlock and relock the domain in order to facilitate DS provisioning after registration. This is a needless cost especially for large domain portfolios. It is also unexpected, as the registrant already has arranged for the necessary CDS/CDNSKEY records to be published. DS initialization and rollovers therefore should be treated the same way with respect to locks.

7. Multiple Submitting Parties and Suspension of Automation

This section provides recommendations to address the following operational questions:

* How are conflicts resolved when DS parameters are accepted through multiple channels (e.g., via a conventional channel and via automation)?

* In case both the registry and the registrar are automating DS provisioning, how to resolve potential collisions?

7.1. Recommendations

1. Registries and registrars MUST provide another (e.g., manual) channel for DS maintenance in order to enable recovery when the Child has lost access to its signing key(s). This out-of-band channel is also needed when a DNS operator does not support DS automation or refuses to cooperate.

2. DS bootstrapping and update requests MUST be executed at the next publication opportunity after verification of their authenticity, regardless of whether they are received in-band or via an out-of-band channel.

3. When processing a CDS/CDNSKEY "delete" signal to remove the entire DS record set ([RFC8078], Section 4), DS automation MUST NOT be suspended. For all other removal requests (such as when received via EPP or a web form), DS automation SHOULD be suspended until a new DS record set has been provisioned, in order to prevent accidental re-initialization when the registrant intended to disable DNSSEC.

4. Whenever a non-empty DS record set is provisioned, through whichever channel, DS automation SHOULD NOT (or no longer) be suspended (including after an earlier removal).

5. In the RRR model, a registry MUST NOT automatically initialize DS records when it is known that the registrar does not provide a way for the domain holder to later disable DNSSEC. If the registrar has declared that it performs automated DS maintenance, the registry SHOULD publish the registrar’s [RFC9859] notification endpoint (if applicable) and refrain from registry-side DS automation.

7.2. Analysis

In the RRR model, there are multiple channels through which DS parameters can be accepted:

* The registry can retrieve information about an intended DS provisioning request automatically from the Child DNS operator and apply the it directly;

* The registrar can retrieve the same and relay it to the registry;

* The registrar can obtain the information from the registrant through another channel (such as a non-automated "manual update" via webform submission), and relay it to the registry.

There are several considerations in this context, as discussed in the following subsections.

7.2.1. Necessity of Non-automatic Updates

Under special circumstances, it may be necessary to perform a non-automatic DS update. One important example is when the key used for authentication of DS updates is destroyed: in this case, an automatic key rollover is impossible as the Child DNS operator can no longer authenticate the associated information. Another example is when several providers are involved, but one no longer cooperates (e.g., when removing a provider from a multi-provider setup). Disabling all other DS management interfaces therefore poses significant operational risk.

Similarly, when the registrar is known to not support DNSSEC (especially, to not provide a means to remove a DS RRset), registries are cautioned against automatically initializing DS records, in order to prevent situations in which a misconfigured or undesired DS RRset cannot be repaired by the registrant.

7.2.2. Impact of Non-automatic Updates: When to Suspend Automation

When an out-of-band (e.g., manual) DS update is performed while CDS/CDNSKEY records referencing the previous DS RRset’s keys are present, the delegation’s DS records may be reset to their previous state at the next run of the automation process. This section discusses in which situations it is appropriate to suspend DS automation after such a non-automatic update.

One option is to suspend DS automation after a manual DS update, but only until a resumption signal is observed. In the past, it was proposed that seeing an updated SOA serial in the child zone may

serve as a resumption signal. However, as any arbitrary modification of zone contents — including the regular updating of DNSSEC signature validity timestamps — typically causes a change in SOA serial, resumption of DS automation after a serial change comes with a high risk of surprise. Additional issues arise if nameservers have different serial offsets (e.g., in a multi-provider setup). This practice therefore is NOT RECOMMENDED.

Note also that "automatic rollback" due to old CDS/CDNSKEY RRsets can only occur if they are signed with a key authorized by one of new DS records. Acceptance checks described in Section 4 further ensure that updates do not break validation.

Removal of a DS record set is triggered either through a CDS/CDNSKEY "delete" signal observed by the party performing the automation ([RFC8078], Section 4), or by receiving a removal request out-of-band (e.g., via EPP or a web form). In the first case, the registrant can expect automation to be kept active for the delegation to facilitate later DS bootstrapping. In the second case, it is likely that the registrant intends to disable DNSSEC for the domain, and DS automation is best suspended (until a new DS record is provisioned somehow).

One may ask how a registry can know whether a removal request received via EPP was the result of the registrar observing a CDS/CDNSKEY "delete" signal. It turns out that the registry does not need to know that; in fact, the advice works out nicely regardless of who does the automation:

a. Only registry: If the registry performs automation, then the registry will consider any request received from the registrar as out-of-band (in the context of this automation). When such requests demand removal of the entire DS record set, the registry therefore should suspend automation.

b. Only registrar: The registrar can always distinguish between removal requests obtained from a CDS/CDNSKEY "delete" signal and other registrant requests, and suspend automation as appropriate.

c. In the (undesirable) case that both parties automate, there are two cases:

    * If the registrant submits a manual removal request to the registrar, it is out-of-band from the registrar perspective (e.g., web form), and also for the registry (e.g., EPP). As a consequence, both will suspend automation (which is the correct result).

* If a CDS/CDNSKEY "delete" signal causes the registrar to request DS removal from the registry, then the registry will suspend automation (because the removal request is received out-of-band, such as via EPP). This is independent of whether the registry's automation has already seen the signal. The registrar, however, will be aware of the in-band nature of the request and not suspend automation (which is also the correct result).

As a side effect, this works towards avoiding redundant automation at the registry.

All in all:

* It is advisable to generally not suspend in-band DS automation when an out-of-band DS update has occurred.

* An exception to this rule is when the entire DS record set was removed through an out-of-band request, in which case the registrant likely wants to disable DNSSEC for the domain. DS automation should then be suspended so that automatic re-initialization (bootstrapping) does not occur.

* In all other cases, any properly authenticated DS updates received, including through an automated method, are to be considered as the current intent of the domain holder.

7.2.3. Concurrent Automatic Updates

When the RRR model is used, there is a potential for collision if both the registry and the registrar are automating DS provisioning by scanning the child for CDS/CDNSKEY records. No disruptive consequences are expected if both parties perform DS automation. An exception is when during a key rollover, registry and registrar see different versions of the Child's DS update requests, such as when CDS/CDNSKEY records are retrieved from different vantage points. Although unlikely due to Recommendation 1a of Section 4, this may lead to flapping of DS updates. However, it is not expected to be harmful as either DS RRset will allow for the validation function to continue to work, as ensured by Recommendation 1b of Section 4. The effect subsides as the Child's state eventually becomes consistent (roughly, within the child's replication delay); any flapping until then will be a minor nuisance only.

The issue disappears entirely when scanning is replaced by notifications that trigger DS maintenance through one party's designated endpoint [RFC9859], and can otherwise be mitigated if the registry and registrar agree that only one of them will perform scanning.

As a standard aspect of key rollovers [RFC6781], the Child DNS operator is expected to monitor propagation of Child zone updates to all authoritative nameserver instances, and only proceed to the next step once replication has succeeded everywhere and the DS record set was subsequently updated (and in no case before the DS RRset's TTL has passed). Any breakage resulting from improper timing on the Child side is outside of the Parent's sphere of influence, and thus cannot be handled with only parent-side changes.

8. IANA Considerations

This document has no IANA actions.

9. Operational Considerations

The document provides operational recommendations for DNSSEC DS automation. There are no additional operational considerations beyond those listed in Appendix A.

10. Security Considerations

The recommendations in this document are designed to improve the safety and interoperability of DNSSEC delegation maintenance. Relevant security implications and various trade-offs are explained in the analysis subsections above. This section notes additional aspects worth considering.

When inconsistencies between CDS/CDNSKEY RRsets are ignored (contrary to Recommendation 4.1.1.a), a number of security risks result. For example, when a nameserver domain expires and is re-registered maliciously, the adversary may be able to initialize a DS RRset and subsequently redelegate the domain using CSYNC synchronization [RFC7477], resulting in a full hijack of the domain. For details, refer to Appendix A of [I-D.ietf-dnsop-cds-consistency].

Similar risks of total adversarial control exist when the child's SEP key is compromised, as this key can authorize DS update or removal requests if consistently published on all nameservers. This reinforces that loss of key control poses severe risks; utmost care must be taken when managing SEP keys.

When a domain is stripped of its DNSSEC protection by removing the DS RRset — either manually or using an automatic delete signal (Recommendation 7.1.3) —, DNSSEC security guarantees and associated benefits are no longer in effect. For example, an email operator may enforce DANE [RFC7672] for domains previously observed to support it, and as a result experience a service disruption in email delivery. Both child and parent DNS operators MUST take such service disruptions into account when considering removal of the DS RRset for their zone.

11. Acknowledgments

The authors would like to thank the members of ICANN’s Security and Stability Advisory Committee (SSAC) who wrote the [SAC126] report on which this document is based.

Additional thanks are extended to the following individuals (in the order of their first contribution or review): Barbara Jantzen, Matt Pounsett, Matthijs Mekking, Ondej Caletka, Oli Schacher, Kim Davies, Jim Reid, Q Misell, Scott Hollenbeck, Tams Csillag, Philip Homburg, Shumon Huque (Document Shepherd), Libor Peltan, Josh Simpson, Johan Stenstam, Stefan Ubbink, Viktor Dukhovni, Hugo Salgado, Wes Hardaker, Mohamed Boucadair (responsible Area Director), Meir Goldman, Thomas Fossati, Peter van Dijk, Jiankang Yao, Donald Eastlake, James Gannon, Roman Danyliw, Andy Newton, ric Vyncke, Mike Bishop, Mahesh Jethanandani, Deb Cooley, Charles Eckel, Christopher Inacio, Ketan Talaulikar

12. References

12.1. Normative References

[DNSKEY-IANA]
IANA, "DNS Security Algorithm Numbers", n.d., <https://www.iana.org/assignments/dns-sec-alg-numbers/dns-sec-alg-numbers.xml#dns-sec-alg-numbers-1>.

[DS-IANA] IANA, "DNSSEC Delegation Signer (DS) Resource Record (RR) Type Digest Algorithms", n.d., <https://www.iana.org/assignments/ds-rr-types>.

[I-D.ietf-dnsop-cds-consistency]
Thomassen, P., "Clarifications on CDS/CDNSKEY and CSYNC Consistency", Work in Progress, Internet-Draft, draft-ietf-dnsop-cds-consistency-11, 11 December 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-dnsop-cds-consistency-11>.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <https://www.rfc-editor.org/rfc/rfc1035>.

[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/rfc/rfc2119>.

[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/rfc/rfc7344>.

[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/rfc/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/rfc/rfc8174>.

[RFC8590] Gould, J. and K. Feher, "Change Poll Extension for the Extensible Provisioning Protocol (EPP)", RFC 8590, DOI 10.17487/RFC8590, May 2019, <https://www.rfc-editor.org/rfc/rfc8590>.

[RFC9364] Hoffman, P., "DNS Security Extensions (DNSSEC)", BCP 237, RFC 9364, DOI 10.17487/RFC9364, February 2023, <https://www.rfc-editor.org/rfc/rfc9364>.

[RFC9499] Hoffman, P. and K. Fujiwara, "DNS Terminology", BCP 219, RFC 9499, DOI 10.17487/RFC9499, March 2024, <https://www.rfc-editor.org/rfc/rfc9499>.

[RFC9567] Arends, R. and M. Larson, "DNS Error Reporting", RFC 9567, DOI 10.17487/RFC9567, April 2024, <https://www.rfc-editor.org/rfc/rfc9567>.

[RFC9615] Thomassen, P. and N. Wisiol, "Automatic DNSSEC Bootstrapping Using Authenticated Signals from the Zone’s Operator", RFC 9615, DOI 10.17487/RFC9615, July 2024, <https://www.rfc-editor.org/rfc/rfc9615>.

[RFC9859] Stenstam, J., Thomassen, P., and J. Levine, "Generalized DNS Notifications", RFC 9859, DOI 10.17487/RFC9859, September 2025, <https://www.rfc-editor.org/rfc/rfc9859>.

12.2. Informative References

[I-D.ietf-regext-rdap-ttl-extension]
Brown, G., "RDAP Extension for DNS Time-To-Live (TTL Values)", Work in Progress, Internet-Draft, draft-ietf-regext-rdap-ttl-extension-11, 21 May 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-regext-rdap-ttl-extension-11>.

[LowTTL] paek, P., "DS and DNSKEY low TTL experiments", at DNS OARC 41, 6 September 2023, <https://indico.dns-oarc.net/event/47/contributions/1010/attachments/958/1811/DS%20and%20DNSKEY%20TTL%20experiment.pdf>.

[RFC5730] Hollenbeck, S., "Extensible Provisioning Protocol (EPP)", STD 69, RFC 5730, DOI 10.17487/RFC5730, August 2009, <https://www.rfc-editor.org/rfc/rfc5730>.

[RFC5731] Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Domain Name Mapping", STD 69, RFC 5731, DOI 10.17487/RFC5731, August 2009, <https://www.rfc-editor.org/rfc/rfc5731>.

[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/rfc/rfc6781>.

[RFC6840] Weiler, S., Ed. and D. Blacka, Ed., "Clarifications and Implementation Notes for DNS Security (DNSSEC)", RFC 6840, DOI 10.17487/RFC6840, February 2013, <https://www.rfc-editor.org/rfc/rfc6840>.

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

[RFC7672] Dukhovni, V. and W. Hardaker, "SMTP Security via Opportunistic DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS)", RFC 7672, DOI 10.17487/RFC7672, October 2015, <https://www.rfc-editor.org/rfc/rfc7672>.

[RFC9803] Brown, G., "Extensible Provisioning Protocol (EPP) Mapping for DNS Time-to-Live (TTL) Values", RFC 9803, DOI 10.17487/RFC9803, June 2025, <https://www.rfc-editor.org/rfc/rfc9803>.

[SAC126] ICANN Security and Stability Advisory Committee (SSAC), "SAC126: DNSSEC Delegation Signer (DS) Record Automation", 12 August 2024, <https://itp.cdn.icann.org/en/files/security-and-stability-advisory-committee-ssac-reports/sac-126-16-08-2024-en.pdf>.

Appendix A. Recommendations Overview

For ease of review and referencing, the recommendations from this document are reproduced here without further comment. For background and analysis, refer to Sections 47.

A.1. Acceptance Checks and Safety Measures

1. Entities performing automated DS maintenance MUST verify:

    a. the unambiguous intent of each DS bootstrapping or update request as per [I-D.ietf-dnsop-cds-consistency], by checking its consistency both

        * between any published CDS and CDNSKEY records, and

        * across all authoritative nameservers in the delegation,

        and

    b. that the resulting DS record set would allow continued DNSSEC validation if deployed,

    and cancel the update if the verifications do not succeed.

2. Parent-side entities (such as registries) SHOULD reduce a DS record set’s TTL to a value between 515 minutes when a new set of records is published, and restore the previous (or, if unavailable, default) TTL value at a later occasion (but not before the previous DS RRset’s TTL has expired).

3. DNS operators MUST publish both CDNSKEY and CDS records (unless the parent’s preference is known), and follow best practice for the choice of hash digest type [DS-IANA].

A.2. Reporting and Transparency

1. For certain DS updates (see analysis (Section 5.2)) and for DS deactivation, relevant points of contact known to the parent-side entity (registry or registrar) SHOULD be notified.

2. For error conditions, the child DNS operator and the domain's technical contact (if applicable) SHOULD be notified first. The registrant SHOULD NOT be notified unless the problem persists for a prolonged amount of time (e.g., three days).

3. Child DNS operators SHOULD be notified of errors using a report query [RFC9567] to the agent domain as described in Section 4 of [RFC9859]. Notifications to humans (domain holder) will be performed in accordance with the communication preferences established with the parent-side entity. The same condition SHOULD NOT be reported unnecessarily frequently to the same recipient.

4. In the RRR model, registries performing DS automation SHOULD inform the registrar of any DS record changes via the EPP Change Poll Extension [RFC8590] or a similar channel.

5. The currently active DS configuration SHOULD be made accessible to the registrant (or their designated party) through the customer portal available for domain management. The DS update history MAY be made available in the same way.

A.3. Registration Locks

1. To secure ongoing operations, automated DS maintenance MUST NOT be suspended based on a registrar update lock alone (such as EPP status clientUpdateProhibited [RFC5731]).

2. When performed by the registry, automated DS maintenance MUST NOT be suspended based on a registry update lock alone (such as EPP status serverUpdateProhibited [RFC5731]).

A.4. Multiple Submitting Parties and Suspension of Automation

1. Registries and registrars MUST provide another (e.g., manual) channel for DS maintenance in order to enable recovery when the Child has lost access to its signing key(s). This out-of-band channel is also needed when a DNS operator does not support DS automation or refuses to cooperate.

2. DS bootstrapping and update requests MUST be executed at the next publication opportunity after verification of their authenticity, regardless of whether they are received in-band or via an out-of-band channel.

3. When processing a CDS/CDNSKEY "delete" signal to remove the entire DS record set ([RFC8078], Section 4), DS automation MUST NOT be suspended. For all other removal requests (such as when

received via EPP or a web form), DS automation SHOULD be suspended until a new DS record set has been provisioned, in order to prevent accidental re-initialization when the registrant intended to disable DNSSEC.

4. Whenever a non-empty DS record set is provisioned, through whichever channel, DS automation SHOULD NOT (or no longer) be suspended (including after an earlier removal).

5. In the RRR model, a registry MUST NOT automatically initialize DS records when it is known that the registrar does not provide a way for the domain holder to later disable DNSSEC. If the registrar has declared that it performs automated DS maintenance, the registry SHOULD publish the registrar's [RFC9859] notification endpoint (if applicable) and refrain from registry-side DS automation.

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

* draft-ietf-dnsop-ds-automation-09

  Add substance to Security Considerations based on IESG review

  Editorial changes and three more MUSTs from IESG review

* draft-ietf-dnsop-ds-automation-08

  Elevate some defining features of DS automation from SHOULD to MUST

* draft-ietf-dnsop-ds-automation-07

  Editorial changes from proofreading

  Editorial changes (Telechat review, James Gannon)

  Editorial changes (IETF LC, Donald Eastlake)

* draft-ietf-dnsop-ds-automation-06

  Add historic background (IETF LC, Jiankang Yao)

  Editorial changes (IETF LC, Peter van Dijk)

  Point out importance of retaining decision details for troubleshooting (IETF LC, Meir Goldman)

  Editorial changes (IETF LC, Thomas Fossati)

* draft-ietf-dnsop-ds-automation-05

  Editorial changes from AD Review

* draft-ietf-dnsop-ds-automation-04

  Editorial changes

* draft-ietf-dnsop-ds-automation-03

  Editorial changes

* draft-ietf-dnsop-ds-automation-02

  Add Appendix with recommendations overview

  Editorial changes

  Change type to BCP

  Fold CDS/CDNSKEY consistency requirements (Section 6) into Section 2 (on acceptance checks)

  Clarify continuity of validation

  In RRR, clarify that registries should not bootstrap if registrar has no deactivation interface (or if registrar does the automation)

  Remove Appendix C ("Approaches not pursued")

* draft-ietf-dnsop-ds-automation-01

  Remove Recommendation 6.1.2 which had told parents to require publication of both CDS and CDNSKEY

  Clarify Recommendation 5.1.3 (on suspension of automation after DS RRset removal) and provide extra analysis

  Providing access to DS update history is now optional

  Humans (domains holders) should be notified according to preferences established with registry/registrar (not necessarily via email)

  Remove redundant Recommendation 5.1.5 (same as 3.1.4)

  Editorial changes

* draft-ietf-dnsop-ds-automation-00

  Rename after adoption

* draft-shetho-dnsop-ds-automation-02

  Allow DS automation during registry update lock

  Editorial changes

* draft-shetho-dnsop-ds-automation-01

  Incorporated various review feedback (editorial + adding TODOs)

* draft-shetho-dnsop-ds-automation-00

  Initial public draft.

Authors' Addresses

Steve Sheng
Email: steve.sheng@gmail.com

Peter Thomassen
deSEC
Email: peter@desec.io