Domain Name System Operations F. Obser
Internet-Draft M. Pels
Intended status: Informational RIPE NCC
Expires: 14 November 2026 13 May 2026

DNSSEC Key Restore
draft-ietf-dnsop-dnssec-keyrestore-01

Abstract

This document describes the issues surrounding the handling of DNSSEC private keys in a DNSSEC signer. It presents operational guidance in case a DNSSEC private key becomes inoperable.

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/fobser/draft-fobser-dnsop-dnssec-keyrecovery.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

DNSSEC [RFC9364] uses public key cryptography to provide integrity protection of DNS data. The private key used for DNSSEC signing could become inoperable at any point due to hardware failure, natural disaster, operator error, or malicious action. If no backup of the private key exist (due to hardware limitations or operational policies) or if the backup is unusable for some reason, a zone can no longer be changed or re-signed.

This document describes procedures on how to restore the DNSSEC signing functionality without rendering a zone temporarily insecure or bogus. For these procedures, it is assumed a complete copy of the DNSSEC signed zone is still available. If no (usable) backup exists, it may be possible to recover the signed zone from one of the zone's name servers.

2. Conventions and Definitions

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.

This document uses DNS terminology from [RFC9499]. DNSSEC key states and timeline related abbreviations are defined in [RFC7583].

The following additional definitions are used within this document.

Inoperable (private key): The private part of a DNSKEY appearing in the chain of trust of the zone that can no longer be used for signing. Causes include hardware failure, natural disaster, operator error, or malicious action. A compromised key is not an inoperable private key since it can still be used for signing.

Operable (private key): The opposite of an inoperable private key. A key that can be used for signing.

3. Scope

The procedures described in this document pertain to DNSSEC architectures with pre-signed records. Online signing, such as described in [RFC9824], is out of scope since it requires that each server carrying the zone holds a copy of the signing key(s). Thus, the operational challenges are different than described in the introduction.

The root zone is out of scope since the distribution of a new trust anchor takes considerably longer than the RRSIG lifetime [RFC7958].

4. DNSSEC Key Restore

In case of a catastrophe where the DNSSEC private key becomes inoperable and no functioning backups of the private key are available, it is desirable to recover from this situation with DNS resolution continuing to work for the effected zone(s) while performing DNSSEC key restore operations.

This is possible because the moment the DNSSEC private key becomes inoperable, the zone is still correctly signed and served by the authoritative name servers. Signatures typically have a lifetime of many days. That means that the operator has a lot of time to recover from this situation without the zone becoming bogus and no longer validating. Hasty and inappropriate action on the other hand could lead to outages.

While the DNSSEC private key cannot be restored because no functioning backups exist, the function of the zone can be restored.

The restore process uses slightly modified key rollover procedures from [RFC7583].

During the restore process, the signing software operates on a pre-signed zone. That is, the zone already contains a DNSKEY RRset and RRSIG RRsets. The signing software might try to remove these records because the accompanying private key is no longer present. The operator MUST prevent this, otherwise the zone will become bogus.

The signing software MUST NOT remove DNSKEYs until instructed to do so and SHOULD NOT remove old RRSIGs. If a signer implementation does not support keeping the old RRSIG records in place these records, excluding the RRSIG for the old DNSKEY RRset, MUST be manually added back to the zone before publication.

The exact process depends on which key(s) are inoperable and if the zone is signed with a split KSK / ZSK key pair or a Combined Signing Key (CSK).

Performing an Algorithm Rollover as described in [RFC6781] using the procedures defined in this document is NOT RECOMMENDED. If an algorithm rollover is not already in progress, signing using the currently used algorithm should be restored first using the procedures defined in this document. Once this has been completed a regular algorithm rollover can be performed.

4.1. Key Rollover Considerations

If a regular key rollover is in progress, the procedures described in this document can be followed. They effectively cancel the ongoing key rollover and perform a new one.

If an algorithm rollover is in progress the procedures described in this document can be followed, with the exception that a new key MUST be added to the zone per algorithm for which there is an inoperable key.

4.2. SOA considerations

When restoring an inoperable ZSK or CSK, the SOA record of the zone SHOULD NOT be changed when introducing a new key in the DNSKEY RRset, because the SOA cannot be re-signed with the inoperable key. In case the SOA is changed, signed responses for existing records will remain valid, but denial of existence proofs for non-existent record types will become bogus.

To ensure the zone is still propagated, any secondary name servers relying on IXFR/AXFR need to be manually forced to load the new version of the zone.

4.3. CDS/CDNSKEY considerations

For restoring an inoperable KSK or CSK, a new DS record needs to be added to the parent zone. For child zones where this update process is ordinarily handled using CDS/DNSKEY records (see [RFC8078]) the DS update needs to be performed manually if the ZSK or CSK is inoperable. This is because CDS/DNSKEY records added to the child zone cannot be signed with the inoperable key, and thus cannot be cryptographically validated. Additionally, introducing CDS/CDNSKEY records in the zone would change the type bitmap of the NSEC or NSEC3 record in the zone apex, which also cannot be re-signed with the inoperable key.

4.4. KSK / ZSK split, KSK operable, ZSK inoperable

Since the old ZSK is inoperable, it cannot be used to create new RRSIGs. Therefore the zone cannot be changed and only the Pre-Publication method can be used. See [RFC7583] section 2.1.

Section 3.2.1 of [RFC7583] documents the timeline for this method.

The following diagram shows the timeline of the restoration. Time increases along the horizontal scale from left to right and the vertical lines indicate events in the process. Significant times and time intervals are marked.

```
                 |1|      |2|   |3|      |4|
                  |        |     |        |
   Key N         - - ----------->|<-Iret->|
                  |        |     |        |
   Key N+1        |<-Ipub->|<--->|<----- - -
                  |        |     |        |
   Key N                                 Trem
   Key N+1        Tpub    Trdy  Tact

                      ---- Time ---->
```

Event 1: The new ZSK is added to the DNSKEY RRset at its publication time (Tpub).

The inoperable ZSK and all RRSIGs it created MUST remain in the zone.

The SOA record of the zone SHOULD NOT be changed at this point in time, because it cannot be re-signed with the inoperable key. Any secondary name servers relying on IXFR/AXFR need to be manually forced to load the new version of the zone.

The new ZSK must be published long enough to guarantee that any cached DNSKEY RRset contains the new ZSK. This interval is the publication interval (Ipub), given by

Ipub = Dprp + TTLkey

Dprp is the propagation delay, the time it takes for changes to propagate to all authoritative nameserver instances. TTLkey is the TTL of the DNSKEY RRset.

Event 2: The new ZSK can be used when it becomes ready at Trdy.

Trdy = Tpub + Ipub.

At this point the zone can be changed again.

Event 3: At some later time, the zone is signed with the new ZSK. At this point RRSIGs from the inoperable ZSK can be removed. The inoperable ZSK MUST be retained in the DNSKEY RRset.

Event 4: The inoperable ZSK can be removed after the retire interval (Iret).

Iret = Dsgn + Dprp + TTLsig

Dsgn is the delay needed to ensure that all existing RRsets are signed with the new ZSK, Dprp is the propagation delay and TTLsig is the maximum TTL of all RRSIG records.

Theoretically the Double-Signature method could be used as well. In this case records in the zone can only be changed after the retire interval, which is at least as long as the publication interval of the Pre-Publication method. The Double-Signature retire interval is given by:

Iret = Dsgn + Dprp + max(TTLkey, TTLsig)

4.5. KSK / ZSK split, KSK inoperable

Since the old KSK is inoperable, the DNSKEY RRset cannot be changed. Therefore, only the Double-DS method can be used. See [RFC7583] section 2.2.

If the ZSK is inoperable as well, it MUST NOT be restored yet.

Section 3.3.2 of [RFC7583] documents the timeline for this method.

The following diagram shows the timeline of the restoration. The diagram follows the convention described in Section 4.1.

```
            |1|      |2|       |3|  |4|      |5|
             |        |         |    |        |
Key N       - ---------------------->|<-Iret->|
             |        |         |    |        |
Key N+1      |<-Dreg->|<-IpubP->|<-->|<------- -
             |        |         |    |        |
Key N                                        Trem
Key N+1     Tsbm     Tpub      Trdy Tact

                 ---- Time ---->
```

Event 1: A new DS record is added to the DS RRset in the parent zone, this is the submission time, Tsbm.

Event 2: After the registration delay, Dreg, the DS record is published in the parent zone. This is the publication time (Tpub).

Tpub = Tsbm + Dreg.

The DS record must be published long enough to guarantee that any cached DS RRset contains the new DS record. This is the parent publication interval (IpubP).

IpubP = DprpP + TTLds

DprpP is the propagation delay of the parent zone, i.e. the time it takes for changes to propagate to all authoritative servers of the parent zone. TTLds is the TTL of the DS RRset at the parent.

Event 3: The new KSK can be used when it becomes ready at Trdy.

Trdy = Tpub + IpubP

Event 4: At this point, Tact, the new KSK is added to the DNSKEY RRset and used to generate the DNSKEY RRSIG. The old, inoperable KSK can be removed. The ZSK MUST remain in the DNSKEY RRset.

If the ZSK is inoperable, the SOA record of the zone SHOULD NOT be changed at this point in time, because it cannot be re-signed with the inoperable key. Any secondary name servers relying on IXFR/AXFR need to be manually forced to load the new version of the zone. The ZSK signing function can be restored using the procedure in the previous section.

To ensure that no caches have DNSKEY RRset with the old KSK, the old DS record MUST remain in the parent zone for the duration of the retire interval (Iret), given by:

Iret = DprpC + TTLkey

DprpC is the child propagation delay, the time it takes for changes to propagate to all authoritative nameserver instances of the child zone. TTLkey is the TTL of the DNSKEY RRset.

Event 5: The old DS record can be removed from the parent zone at Trem.

Trem = Tact + Iret

4.6. CSK inoperable

Since the old CSK is inoperable, the DNSKEY RRset cannot be changed. Therefore, only the Double-DS method can be used. See [RFC7583] section 2.2.

Section 3.3.2 of [RFC7583] documents the timeline for this method.

Since the CSK is also used to sign the zone, the timing of the Double-DS method needs to be adjusted.

The inoperable CSK and all RRSIGs it created MUST remain in the zone.

The following diagram shows the timeline of the restoration. The diagram follows the convention described in Section 4.1.

```
             |1|      |2|       |3|  |4|      |5|
              |        |         |    |        |
Key N        - ---------------------->|<-Iret->|
              |        |         |    |        |
Key N+1       |<-Dreg->|<-IpubP->|<-->|<------- -
              |        |         |    |        |
Key N                                         Trem
Key N+1      Tsbm     Tpub      Trdy Tact

                  ---- Time ---->
```

Event 1: A new DS record is added to the DS RRset in the parent zone, this is the submission time, Tsbm.

Event 2: After the registration delay, Dreg, the DS record is published in the parent zone. This is the publication time (Tpub).

Tpub = Tsbm + Dreg.

The DS record must be published long enough to guarantee that any cached DS RRset contains the new DS record. This is the parent publication interval (IpubP) given by

IpubP = DprpP + TTLds

DprpP is the propagation delay of the parent zone, i.e. the time it takes for changes to propagate to all authoritative servers of the parent zone. TTLds is the TTL of the DS RRset at the parent.

Event 3: The new CSK can be used when it becomes ready at Trdy.

Trdy = Tpub + IpubP

Event 4: At this point the new CSK is added to the DNSKEY RRset and used to generate the DNSKEY RRSIG.

The old, inoperable CSK MUST remain in the DNSKEY RRset. The RRSIGs generated by the inoperable CSK MUST remain in the zone.

The SOA record of the zone SHOULD NOT be changed at this point in time, because it cannot be re-signed with the inoperable key. Any secondary name servers relying on IXFR/AXFR need to be manually forced to load the new version of the zone.

To ensure that no caches have DNSKEY RRset with the old CSK, the old DS record MUST remain in the parent zone for the duration of the retire interval (Iret), given by:

Iret = Dsgn + DprpC + max(TTLkey, TTLsig)

Dsgn is the delay needed to ensure that all existing RRsets are signed with the new CSK. DprpC is the child propagation delay, the time it takes for changes to propagate to all authoritative nameserver instances of the child zone. TTLkey is the TTL of the DNSKEY RRset and TTLsig is the maximum TTL of all RRSIG records.

Event 5: The old DS record can be removed from the parent zone at Trem.

Trem = Tact + Iret

At the same time the old, inoperable CSK and all its signatures can be removed as well.

At this point the zone can be changed again.

5. Security Considerations

All security considerations of [RFC9364] apply to this document.

6. IANA Considerations

This document has no IANA actions.

7. References

7.1. 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/rfc/rfc2119>.

[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>.

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

7.2. Informative References

[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>.

[RFC7583] Morris, S., Ihren, J., Dickinson, J., and W. Mekking, "DNSSEC Key Rollover Timing Considerations", RFC 7583, DOI 10.17487/RFC7583, October 2015, <https://www.rfc-editor.org/rfc/rfc7583>.

[RFC7958] Abley, J., Schlyter, J., Bailey, G., and P. Hoffman, "DNSSEC Trust Anchor Publication for the Root Zone", RFC 7958, DOI 10.17487/RFC7958, August 2016, <https://www.rfc-editor.org/rfc/rfc7958>.

[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>.

[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>.

[RFC9824] Huque, S., Elmerot, C., and O. Gudmundsson, "Compact Denial of Existence in DNSSEC", RFC 9824, DOI 10.17487/RFC9824, September 2025, <https://www.rfc-editor.org/rfc/rfc9824>.

Acknowledgments

The document draws heavily from the work in [RFC7583] and we thank the authors for their work:

* Stephen Morris

* Johan Ihren

* John Dickinson

* W. (Matthijs) Mekking

Additionally, we thank the following people for contributing ideas and feedback:

* Libor Peltan

* Peter Thomassen

* Anand Buddhdev

* Wes Hardaker

Authors' Addresses

Florian Obser
RIPE NCC
Email: fobser@ripe.net

Martin Pels
RIPE NCC
Email: mpels@ripe.net