Network Working Group
Internet-Draft
Intended status: Best Current Practice
Expires: 9 July 2026

W. Hardaker
Google

5 January 2026

Intentionally Temporarily Degraded or Insecure
draft-hardaker-dnsop-intentionally-temporary-insec-02

Abstract

Performing DNSKEY algorithm transitions with DNSSEC signing is unfortunately challenging to get right in practice without decent tooling support. This document weighs the correct, completely secure way of rolling keys against an alternate, significantly simplified, method that takes a zone through an insecure state first.

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 9 July 2026.

Table of Contents

1. Introduction

Performing DNSKEY [RFC4035] algorithm transitions with DNSSEC [RFC4033] signing is unfortunately challenging to get right in practice without decent tooling support. This document weighs the correct, completely secure way of rolling keys against an alternate, significantly simplified, method that takes a zone through an insecure state.

Section 4.1.4 of [RFC6781] describes the necessary steps required when a new signing key is published for a zone that uses a different signing algorithm than the currently published keys. These are the steps that MUST be followed when zone owners wish to have uninterrupted DNSSEC protection for their zones. The steps in this document are designed to ensure that all DNSKEY records and all DS [RFC4509] records (and the rest of a zone records) are properly validatable by validating resolvers throughout the entire process. Whenever possible, this procedure SHOULD be followed.

Unfortunately, there are a number of these steps that are challenging to accomplish either because the timing is tricky to get right or because current software doesn't support automating the process easily. Some examples:

1. The second step in Section 4.1.4 of [RFC6781] requires that a new key with the new algorithm (which we refer to as K_new) be created, but not yet published. This step requires that both the old key (K_old) and K_new both sign and generate signatures for the zone, but without K_new actually being published in the zone even though its signatures. Put another way, only K_old can exist in the zone even though signatures from both keys must be included. After this odd mix has been published for a sufficient time length, based on the TTL, can K_new be safely introduced and published into the zone as well.

2. Sometimes one of the goals is to transfer zone management to new authoritative server software. But, if the newly desired algorithm isn't supported in the existing (to be replaced) DNSSEC signing software, then the transfer to the new software must be accomplished first. However, if there isn't an overlap between the algorithms available in both software sets, it becomes practically impossible to even transfer the zone since neither software set can use both K_old and K_new.

Although some DNSSEC signing solutions may automate the algorithm rollover steps (making operator involvement unnecessary), many other tools do not yet support automated algorithm updates. In these environments, the most challenging step is requiring that certain RRSIGs be published without the corresponding DNSKEYs that created them. This will likely require operators to use a text editor on the contents of a signed zone to carefully select zone records to extract before publication. This introduces potentially significant operator error(s).

This document proposes an alternate approach that MAY be used to perform algorithm DNSKEY rollovers in these situations, which is potentially more operationally robust but less secure.

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. Transitioning temporarily through insecurity

An alternate approach to properly rolling DNSKEYs to a new algorithm, is to intentionally make the zone become insecure while the DNSKEYs and algorithms are swapped. At a high level, this means removing all DS records from the parent zone first, then remove the old key and introduce the new key with its new algorithm during this period. Zone TTLs can be significantly shortened during this period to minimize the period of insecurity.

Below are the enumerated steps required by this alternate transition mechanism. Note that there are still two critical waiting time requirements (steps 2 and 6) that must be followed carefully.

1. Optional: lower the TTLs of both the zone's DS record, and the TTL of the DNSKEY RRset. Note that in some operational deployments the parent zone may set the TTL of the DS record.

2. Remove all DS records from the parent zone.

3. Ensure the zone is considered unsigned by all validating resolvers by waiting 2 times the maximum TTL length for the DS record, and/or 2 times the largest TTL found in the zone (whichever is larger). This is the most critical timing as all records associated with K_old must be cleared from validating resolver caches. (The author of this document failed to wait the required time once. It was not pretty.)

4. Replace the old DNSKEY(s) using the old algorithm with new DNSKEY(s) using the new algorithm(s) in the zone and publish the zone.

5. Wait 2 times the largest TTL found in the zone to ensure the new DNSKEYs will be found by validating resolvers.

6. Add the DS record(s) for the new DNSKEYs to the parent zone.

7. If the TTLs were modified in the optional step 1, change them back to their preferred values.

3. Operational considerations

The process of replacing a DNSKEY with an older algorithm [RFC9904], such as RSAMD5 [RFC4034] or RSASHA1 [RFC9905] with a more modern one [RFC9904] such as RSASHA512 [RFC5702] or ECDSAP256SHA256 [RFC6605] can be a daunting task if the zone’s current tooling doesn’t provide an easy-to-use solution. For example, this may be the case for zone owners using command line tools integrated into their zone production environment.

This document describes an alternative approach to rolling DNSKEY algorithms that may be significantly less prone to operational mistakes. However, it is paramount that operators understand of the security considerations of using this approach.

The document recommends waiting 2 times TTL values in certain cases for added assurance that the waiting period is long enough for caches to expire. In reality, waiting slightly longer than 1 TTL may be sufficient but requires accepting added risks with propagation timing and clock synchronization.

4. Security considerations

DNSSEC provides data integrity protection for DNS data. This document specifically calls out a reason why a zone owner may desire to deliberately turn off this protection while changing the zone's DNSKEY's cryptographic algorithms. Thus, this technique is potentially harmful if an attacker knows when this will occur and can use that time window to launch DNS modification attacks (for example, cache poisoning attacks) against validating resolvers or other validating DNS infrastructure.

Most importantly, this will deliberately break certain types of DNS records that must be validatable for them to be effective. This includes for example, but not limited to, all DS records for the zone's own children, DANE [RFC6698][RFC7671][RFC7672], PGP key fingerprints [RFC7929], and SSHFP[RFC4255] fingerprints. Zone owners must carefully consider which records within their zone and their zone's children depend on DNSSEC being available before using the procedure outlined in this document.

Given all of this, it leaves the question of: "why would a zone owner want to deliberately turn off security temporarily then?", to which there is one principal answer: if the the complexity of executing an algorithm role the correct way is difficult (or impossible), then the chances of introducing an error that causes an operational outage may be significantly higher than the chances of the zone being attacked during the insecure transition period. Simply put, an invalid zone created by a botched algorithm roll is potentially worse than an unsigned but still available zone.

5. References

5.1. Normative References

[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>.

[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/rfc/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/rfc/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/rfc/rfc4035>.

[RFC4509] Hardaker, W., "Use of SHA-256 in DNSSEC Delegation Signer (DS) Resource Records (RRs)", RFC 4509, DOI 10.17487/RFC4509, May 2006, <https://www.rfc-editor.org/rfc/rfc4509>.

[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>.

[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>.

5.2. Informative References

[RFC4255] Schlyter, J. and W. Griffin, "Using DNS to Securely Publish Secure Shell (SSH) Key Fingerprints", RFC 4255, DOI 10.17487/RFC4255, January 2006, <https://www.rfc-editor.org/rfc/rfc4255>.

[RFC5702] Jansen, J., "Use of SHA-2 Algorithms with RSA in DNSKEY and RRSIG Resource Records for DNSSEC", RFC 5702, DOI 10.17487/RFC5702, October 2009, <https://www.rfc-editor.org/rfc/rfc5702>.

[RFC6605] Hoffman, P. and W.C.A. Wijngaards, "Elliptic Curve Digital Signature Algorithm (DSA) for DNSSEC", RFC 6605, DOI 10.17487/RFC6605, April 2012, <https://www.rfc-editor.org/rfc/rfc6605>.

[RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, DOI 10.17487/RFC6698, August 2012, <https://www.rfc-editor.org/rfc/rfc6698>.

[RFC7671] Dukhovni, V. and W. Hardaker, "The DNS-Based Authentication of Named Entities (DANE) Protocol: Updates and Operational Guidance", RFC 7671, DOI 10.17487/RFC7671, October 2015, <https://www.rfc-editor.org/rfc/rfc7671>.

[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>.

[RFC7929] Wouters, P., "DNS-Based Authentication of Named Entities (DANE) Bindings for OpenPGP", RFC 7929, DOI 10.17487/RFC7929, August 2016, <https://www.rfc-editor.org/rfc/rfc7929>.

[RFC9904] Hardaker, W. and W. Kumari, "DNSSEC Cryptographic Algorithm Recommendation Update Process", RFC 9904, DOI 10.17487/RFC9904, November 2025, <https://www.rfc-editor.org/rfc/rfc9904>.

[RFC9905] Hardaker, W. and W. Kumari, "Deprecating the Use of SHA-1 in DNSSEC Signature Algorithms", RFC 9905, DOI 10.17487/RFC9905, November 2025, <https://www.rfc-editor.org/rfc/rfc9905>.

Appendix A. Acknowledgments

The author has discussed the pros and cons of this approach with multiple people, including:

* Vladimテュr ト蛍nテ。t

* Peter van Dijk

* Viktor Dukhovni

* Warren Kumari.

* Scott Rose

* Tuomo Soini

* Paul Wouters

Appendix B. Github Version of this document

While this document is under development, it can be viewed, tracked, issued, pushed with PRs, ... here:

https://github.com/hardaker/draft-hardaker-dnsop-intentionally-temporarily-insecure

Author's Address

Wes Hardaker
Google
Email: ietf@hardakers.net