dnsop
Internet-Draft
Obsoletes: 6944 (if approved)
Intended status: Standards Track
Expires: August 21, 2019

P. Wouters
Red Hat
O. Sury
Internet Systems Consortium

February 17, 2019

# Algorithm Implementation Requirements and Usage Guidance for DNSSEC
# draft-ietf-dnsop-algorithm-update-06

Abstract

The DNSSEC protocol makes use of various cryptographic algorithms in order to provide authentication of DNS data and proof of non-existence. To ensure interoperability between DNS resolvers and DNS authoritative servers, it is necessary to specify a set of algorithm implementation requirements and usage guidelines to ensure that there is at least one algorithm that all implementations support. This document defines the current algorithm implementation requirements and usage guidance for DNSSEC. This document obsoletes [RFC6944].

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 August 21, 2019.

Table of Contents



## 1. Introduction

The DNSSEC signing algorithms are defined by various RFCs, including [RFC4034], [RFC5155], [RFC5702], [RFC5933], [RFC6605], [RFC8080]. DNSSEC is used to provide authentication of data. To ensure interoperability, a set of "mandatory-to-implement" DNSKEY algorithms are defined. This document obsoletes [RFC6944].

### 1.1. Updating Algorithm Implementation Requirements and Usage Guidance

The field of cryptography evolves continuously. New stronger algorithms appear and existing algorithms are found to be less secure then originally thought. Therefore, algorithm implementation requirements and usage guidance need to be updated from time to time to reflect the new reality. The choices for algorithms must be conservative to minimize the risk of algorithm compromise.

### 1.2. Updating Algorithm Requirement Levels

The mandatory-to-implement algorithm of tomorrow should already be available in most implementations of DNSSEC by the time it is made mandatory. This document attempts to identify and introduce those algorithms for future mandatory-to-implement status. There is no guarantee that algorithms in use today will become mandatory in the future. Published algorithms are continuously subjected to cryptographic attack and may become too weak, or even be completely broken, before this document is updated.

This document only provides recommendations with respect to mandatory-to-implement algorithms or algorithms so weak that recommendation cannot be recommended. Any algorithm listed in the [DNSKEY-IANA] and [DS-IANA] registries, but not mentioned in this document, MAY be implemented. For clarification and consistency, an algorithm will be specified as MAY in this document only when it has been downgraded.

Although this document's primary purpose is to update algorithm recommendations to keep DNSSEC authentication secure over time, it also aims to do so in such a way that DNSSEC implementations remain interoperable. DNSSEC interoperability is addressed by an incremental introduction or deprecation of algorithms.

[RFC2119] considers the term SHOULD equivalent to RECOMMENDED, and SHOULD NOT equivalent to NOT RECOMMENDED. The authors of this document have chosen to use the terms RECOMMENDED and NOT RECOMMENDED, as this more clearly expresses the recommendations to implementers.

It is expected that deprecation of an algorithm will be performed gradually. This provides time for various implementations to update their implemented algorithms while remaining interoperable. Unless there are strong security reasons, an algorithm is expected to be downgraded from MUST to NOT RECOMMENDED or MAY, instead of to MUST NOT. Similarly, an algorithm that has not been mentioned as mandatory-to-implement is expected to be introduced with a RECOMMENDED instead of a MUST.

Since the effect of using an unknown DNSKEY algorithm is that the zone is treated as insecure, it is recommended that algorithms downgraded to NOT RECOMMENDED or lower not be used by authoritative nameservers and DNSSEC signers to create new DNSKEY's. This will allow for deprecated algorithms to become less and less common over time. Once an algorithm has reached a sufficiently low level of deployment, it can be marked as MUST NOT, so that recursive resolvers can remove support for validating it.

Recursive nameservers are encouraged to retain support for all algorithms not marked as MUST NOT.

### 1.3. Document Audience

The recommendations of this document mostly target DNSSEC implementers, as implementations need to meet both high security expectations as well as high interoperability between various vendors and with different versions. Interoperability requires a smooth transition to more secure algorithms. This perspective may differ from from that of a user who wishes to deploy and configure DNSSEC with only the safest algorithm. On the other hand, the comments and recommendations in this document are also expected to be useful for such users.

## 2. Conventions Used in This Document

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.

## 3. Algorithm Selection

### 3.1. DNSKEY Algorithms

Implementation recommendations for DNSKEY algorithms [DNSKEY-IANA].

| Number | Mnemonics | DNSSEC Signing | DNSSEC Validation |
|---|---|---|---|
| 1 | RSAMD5 | MUST NOT | MUST NOT |
| 3 | DSA | MUST NOT | MUST NOT |
| 5 | RSASHA1 | NOT RECOMMENDED | MUST |
| 6 | DSA-NSEC3-SHA1 | MUST NOT | MUST NOT |
| 7 | RSASHA1-NSEC3-SHA1 | NOT RECOMMENDED | MUST |
| 8 | RSASHA256 | MUST | MUST |
| 10 | RSASHA512 | NOT RECOMMENDED | MUST |
| 12 | ECC-GOST | MUST NOT | MAY |
| 13 | ECDSAP256SHA256 | MUST | MUST |
| 14 | ECDSAP384SHA384 | MAY | RECOMMENDED |
| 15 | ED25519 | RECOMMENDED | RECOMMENDED |
| 16 | ED448 | MAY | RECOMMENDED |

RSAMD5 is not widely deployed and there is an industry-wide trend to deprecate MD5 usage.

RSASHA1 and RSASHA1-NSEC3-SHA1 are widely deployed, although zones deploying it are recommended to switch to ECDSAP256SHA256 as there is an industry-wide trend to move to elliptic curve cryptography. RSASHA1 does not support NSEC3. RSASHA1-NSEC3-SHA1 can be used with or without NSEC3.

DSA and DSA-NSEC3-SHA1 are not widely deployed and vulnerable to private key compromise when generating signatures using a weak or compromised random number generator.

RSASHA256 is in wide use and considered strong.

RSASHA512 is NOT RECOMMENDED for DNSSEC Signing because it has not seen wide deployment, but there are some deployments hence DNSSEC Validation MUST implement RSASHA512 to ensure interoperability. There is no significant difference in cryptographics strength between RSASHA512 and RSASHA256, therefore it is discouraged to use RSASHA512, as it will only make deprecation of older algorithms harder. People that wish to use a cryptographically stronger algorithm should switch to elliptic curve cryptography algorithms.

ECC-GOST (GOST R 34.10-2001) has been superseded by GOST R 34.10-2012 in [RFC7091]. The GOST R 34.10-2012 hasn't been standardized for use in DNSSEC.

ECDSAP256SHA256 provides more cryptographic strength with a shorter signature length than either RSASHA256 or RSASHA512. ECDSAP256SHA256 has been widely deployed and therefore it is now at MUST level for both validation and signing. It is RECOMMENDED to use deterministic digital signature generation procedure of the ECDSA ([RFC6979]) when implementing ECDSAP256SHA256 (and ECDSAP384SHA384).

ECDSAP384SHA384 shares the same properties as ECDSAP256SHA256, but offers a modest security advantage over ECDSAP256SHA256 (192 bits of strength versus 128 bits). For most DNSSEC applications, ECDSAP256SHA256 should be satisfactory and robust for the foreseeable future, and is therefore recommended for signing. While it is unlikely for a DNSSEC use case requiring 192-bit security strength to arise, ECDSA384SHA384 is provided for such applications and it MAY be used for signing in these cases.

ED25519 and ED448 use Edwards-curve Digital Security Algorithm (EdDSA). There are three main advantages of the EdDSA algorithm: It does not require the use of a unique random number for each signature, there are no padding or truncation issues as with ECDSA, and it is more resilient to side-channel attacks. Furthermore, EdDSA cryptography is less prone to implementation errors ([RFC8032], [RFC8080]). It is expected that ED25519 will become the future

RECOMMENDED default algorithm once there's enough support for this algorithm in the deployed DNSSEC validators.

### 3.2. DNSKEY Algorithm Recommendation

Operation recommendation for new and existing deployments.

Due to industry-wide trend to move to elliptic curve cryptography, the ECDSAP256SHA256 is RECOMMENDED DNSKEY algorithm for use by new DNSSEC deployments, and users of RSA based algorithms SHOULD upgrade to ECDSAP256SHA256.

### 3.3. DS and CDS Algorithms

Recommendations for Delegation Signer Digest Algorithms [DNSKEY-IANA] These also apply to the CDS RRTYPE as specified in [RFC7344]

| Number | Mnemonics | DNSSEC Delegation | DNSSEC Validation |
|---|---|---|---|
| 0 | NULL (CDS only) | MUST NOT [*] | MUST NOT [*] |
| 1 | SHA-1 | MUST NOT | MUST |
| 2 | SHA-256 | MUST | MUST |
| 3 | GOST R 34.11-94 | MUST NOT | MAY |
| 4 | SHA-384 | MAY | RECOMMENDED |

[*] - This is a special type of CDS record signaling removal of DS at the parent in [RFC8078]

NULL is a special case, see [RFC8078]

SHA-1 is still in wide use for DS records, so validators MUST implement validation, but it MUST NOT be used to generate new DS and CDS records. (See Operational Considerations for caveats when upgrading from SHA-1 to SHA-256 DS Algorithm.)

SHA-256 is in wide use and considered strong.

GOST R 34.11-94 has been superseded by GOST R 34.11-2012 in [RFC6986]. The GOST R 34.11-2012 hasn't been standardized for use in DNSSEC.

SHA-384 shares the same properties as SHA-256, but offers a modest security advantage over SHA-384 (384-bits of strength versus 256-bits). For most applications of DNSSEC, SHA-256 should be satisfactory and robust for the foreseeable future, and is therefore recommended for DS and CDS records. While it is unlikely for a

DNSSEC use case requiring 384-bit security strength to arise, SHA-384 is provided for such applications and it MAY be used for generating DS and CDS records in these cases.

### 3.4. DS and CDS Algorithm Recommendation

Operation recommendation for new and existing deployments.

The SHA-256 is RECOMMENDED DS and CDS algorithm.

## 4. Security Considerations

The security of cryptographic systems depends on both the strength of the cryptographic algorithms chosen and the strength of the keys used with those algorithms. The security also depends on the engineering of the protocol used by the system to ensure that there are no non-cryptographic ways to bypass the security of the overall system.

This document concerns itself with the selection of cryptographic algorithms for the use of DNSSEC, specifically with the selection of "mandatory-to-implement" algorithms. The algorithms identified in this document as MUST or RECOMMENDED to implement are not known to be broken at the current time, and cryptographic research so far leads us to believe that they are likely to remain secure into the foreseeable future. However, this isn't necessarily forever, and it is expected that new revisions of this document will be issued from time to time to reflect the current best practices in this area.

Retiring an algorithm too soon would result in a zone signed with the retired algorithm being downgraded to the equivalent of an unsigned zone. Therefore, algorithm deprecation must be done very slowly and only after careful consideration and measurement of its use.

## 5. Operational Considerations

DNSKEY algorithm rollover in a live zone is a complex process. See [RFC6781] and [RFC7583] for guidelines on how to perform algorithm rollovers.

DS algorithm rollover in a live zone is also a complex process. Upgrading algorithm at the same time as rolling the new KSK key will lead to DNSSEC validation failures, and users MUST upgrade the DS algorithm first before rolling the Key Signing Key.

## 6. Implementation Report

### 6.1. DNSKEY Algorithms

The following table contains the status of support in the open-source DNS signers and validators in the current released versions as of the time writing this document. Usually, the support for specific algorithm has to be also included in the cryptographic libraries that the software use.

| Mnemonics | BIND | Knot DNS | OpenDNS | PowerDNS | Unbound |
|---|---|---|---|---|---|
| RSAMD5 | Y | N | Y | N | N |
| DSA | Y | N | Y | N | Y |
| RSASHA1 | Y | Y | Y | Y | Y |
| DSA-NSEC3-SHA1 | Y | N | Y | N | Y |
| RSASHA1-NSEC3-SHA1 | Y | Y | Y | Y | Y |
| RSASHA256 | Y | Y | Y | Y | Y |
| RSASHA512 | Y | Y | Y | Y | Y |
| ECC-GOST | N | N | Y | Y | Y |
| ECDSAP256SHA256 | Y | Y | Y | Y | Y |
| ECDSAP384SHA384 | Y | Y | Y | Y | Y |
| ED25519 | Y | Y | N | Y | Y |
| ED448 | N | N | N | Y | Y |

## 7. IANA Considerations

This document makes no requests of IANA.

## 8. Acknowledgements

This document borrows text from RFC 4307 by Jeffrey I. Schiller of the Massachusetts Institute of Technology (MIT) and the 4307bis document by Yoav Nir, Tero Kivinen, Paul Wouters and Daniel Migault. Much of the original text has been copied verbatim.

We wish to thank Michael Sinatra, Roland van Rijswijk-Deij, Olafur Gudmundsson, Paul Hoffman and Evan Hunt for their imminent feedback.

Kudos to Roy Arends for bringing the DS rollover issue to the daylight.

# 9. References

## 9.1. 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/info/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/info/rfc8174>.

## 9.2. Informative References

[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>.

[RFC5155] Laurie, B., Sisson, G., Arends, R., and D. Blacka, "DNS Security (DNSSEC) Hashed Authenticated Denial of Existence", RFC 5155, DOI 10.17487/RFC5155, March 2008, <https://www.rfc-editor.org/info/rfc5155>.

[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/info/rfc5702>.

[RFC5933] Dolmatov, V., Ed., Chuprina, A., and I. Ustinov, "Use of GOST Signature Algorithms in DNSKEY and RRSIG Resource Records for DNSSEC", RFC 5933, DOI 10.17487/RFC5933, July 2010, <https://www.rfc-editor.org/info/rfc5933>.

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

[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>.

[RFC6944] Rose, S., "Applicability Statement: DNS Security (DNSSEC) DNSKEY Algorithm Implementation Status", RFC 6944, DOI 10.17487/RFC6944, April 2013, <https://www.rfc-editor.org/info/rfc6944>.

[RFC6979] Pornin, T., "Deterministic Usage of the Digital Signature Algorithm (DSA) and Elliptic Curve Digital Signature Algorithm (ECDSA)", RFC 6979, DOI 10.17487/RFC6979, August 2013, <https://www.rfc-editor.org/info/rfc6979>.

[RFC6986] Dolmatov, V., Ed. and A. Degtyarev, "GOST R 34.11-2012: Hash Function", RFC 6986, DOI 10.17487/RFC6986, August 2013, <https://www.rfc-editor.org/info/rfc6986>.

[RFC7091] Dolmatov, V., Ed. and A. Degtyarev, "GOST R 34.10-2012: Digital Signature Algorithm", RFC 7091, DOI 10.17487/RFC7091, December 2013, <https://www.rfc-editor.org/info/rfc7091>.

[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>.

[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/info/rfc7583>.

[RFC8032] Josefsson, S. and I. Liusvaara, "Edwards-Curve Digital Signature Algorithm (EdDSA)", RFC 8032, DOI 10.17487/RFC8032, January 2017, <https://www.rfc-editor.org/info/rfc8032>.

[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>.

[RFC8080] Sury, O. and R. Edmonds, "Edwards-Curve Digital Security Algorithm (EdDSA) for DNSSEC", RFC 8080, DOI 10.17487/RFC8080, February 2017, <https://www.rfc-editor.org/info/rfc8080>.

[DNSKEY-IANA] "DNSKEY Algorithms", <http://www.iana.org/assignments/dns-sec-alg-numbers/dns-sec-alg-numbers.xhtml>.

[DS-IANA] "Delegation Signer Digest Algorithms", <http://www.iana.org/assignments/ds-rr-types/ds-rr-types.xhtml>.

## Authors' Addresses

Paul Wouters
Red Hat
CA

EMail: pwouters@redhat.com

Ondrej Sury
Internet Systems Consortium
CZ

EMail: ondrej@isc.org