Internet Engineering Task Force
Internet-Draft
Updates: 4035, 6840 (if approved)
Intended status: Standards Track
Expires: 23 April 2026

S. Huque
Salesforce
P. Thomassen
deSEC, SSE
V. Dukhovni
Google LLC
D. Wessels
Verisign
C. Elmerot
Cloudflare

20 October 2025

Multiple Algorithm Rules in DNSSEC
draft-huque-dnsop-multi-alg-rules-07

Abstract

This document restates the requirements on DNSSEC signing and validation and makes small adjustments in order to allow for more flexible handling of configurations that advertise multiple Secure Entry Points (SEP) with different signing algorithms via their DS record or trust anchor set. The adjusted rules allow both for multi-signer operation and for the transfer of signed DNS zones between providers, where the providers support disjoint DNSSEC algorithm sets. In addition, the proposal enables pre-publication of a trust anchor in preparation for an algorithm rollover, such as of the root zone.

This document updates RFCs 4035 and 6840.

Discussion Venues

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

Source for this draft and an issue tracker can be found at https://github.com/shuque/draft-dnsop-multi-alg-rules.

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/.

Copyright Notice

Table of Contents

1. tl;dr: Nutshell Proof of Sanity

It is well known that

1. validator implementations MUST support certain mainstream algorithms ([DNSKEY-IANA]);

2. validators MUST accept any valid path ([RFC6840] Section 5.11).

Thus, when a zone advertises several algorithms which MUST be supported for validation, the zone operator can reasonably expect that validation will work, even when only serving signatures for one of them. (For use cases see below.)

Therefore,

3. if a mainstream algorithm is disabled in a validator (as a matter of local policy), the validator still ought to accept a path using this algorithm, and treat responses as insecure regardless of other algorithms advertised for the zone.

This is because the zone operator has a reasonable expectation that the algorithm is supported in all validators. The zone operator should not have to expect that serving this path would lead to "bogus" security status / SERVFAIL.

Rather, *if a validator _due to local policy_ does not support required mainstream algorithms*, it should *take on responsibility* for that *locally*, and *behave as a non-validating resolver for that zone.*

This document updates validation rules accordingly: primarily as described above, and secondarily to gracefully cover an implicit issue when a mainstream algorithm reaches its end of life. Downgrade protection is preserved.

2. Introduction and Motivation

The Domain Name System Security Extensions (DNSSEC) [RFC4033] [RFC4034] [RFC4035] add data origin authentication and integrity protection to the Domain Name System (DNS), by having DNS zone owners (or their operators) crytographically sign their zone data.

Current specifications [RFC4035][RFC6840] require that a zone be signed with each signing algorithm listed in a zone’s DS RRset or appearing via its trust anchors (TAs). This poses a problem in (at least) the following situations:

* In multi-signer setups (Multi-Signer Extensions [RFC8901] Section 2.1.2), multiple providers using distinct DNSSEC keys can cooperatively serve the same DNS zone. This method does not work however if the providers involved employ different DNSSEC algorithms.

* DNSSEC Automation [DNSSEC-AUTO] further describes how to fully automate multi-signer operations, including how to use a transitional state of a multi-signer configuration to non-disruptively transfer a signed zone from one signer or provider to another. If the old and the new provider do not use the same signing algorithms, the same problem is encountered.

* When performing an algorithm rollover, current specifications mandate that the zone has to be double-signed with both the old and the new algorithm before publishing the new trust anchor or DS record.

  - This implies that it is not possible to independently change the KSK algorithm alone (i.e., without signing the whole zone with it); however, depending on local circumstances, an operator might prefer a SEP-only (KSK) algorithm change over simultaneously duplicating all keys for the new algorithm. For example, a zone could roll the KSK from algorithm 8 to algorithm 13 without changing the ZSK, and later roll the ZSK.

  - For the root zone, the current rules could lead to a potentially rather long phase of double-signing (on the order of a year). As this comes with both financial and operational risks, it seems desirable to find a way for publishing the new trust anchor without introducing the new algorithm into the zone just yet.

* Furthermore, for online signers, producing on the fly signatures for several algorithms imposes a significant computational burden.

The above issues are not just a theoretical problem. Real situations in the field have occurred where the existing requirements have posed an obstacle to DNSSEC deployment and operations.

That said, the existing signing requirements are well motivated: When a zone’s DS RRset or trust anchor set includes multiple DNSKEY algorithms, an attacker who can strip all the supported RRSIGs from a signed response from that zone, leaving just the unsupported signatures, must not be able to cause the response to be considered "insecure" when it otherwise would have been considered "secure". Instead of such a downgrade, the only acceptable effect from attacker interference is to turn a "secure" outcome into a "bogus" one. The

rules therefore ensure the downgrade resistance of DNSSEC when only some, but not all, of a zone's DS RRset or trust anchor set DNSKEY algorithms are supported by a validating resolver.

This document proposes modifications to the IANA signing algorithm registry and minor modifications of the signing and validation rules to accommodate the above (and potentially other) use cases, without compromising the DNSSEC security guarantees and downgrade resistance.

3. Proposed Updates to RFCs

The heart of the issue is that even though any one acceptable signature suffices for validation, the signer cannot, in the general case, know which particular signing algorithm(s) the validator will support; and hence, providing a "large enough set" (read: all of them) is the approach that had been taken so far.

This is set down in Section 2.2 of [RFC4035]:

> There MUST be an RRSIG for each RRset using at least one DNSKEY of each algorithm in the zone apex DNSKEY RRset. The apex DNSKEY RRset itself MUST be signed by each algorithm appearing in the DS RRset located at the delegating parent (if any).

This document advocates that signers adopt a more liberal approach to the requirement of signatures by algorithm sets when zones employ suitably strong and well known algorithms. It precisely defines which algorithms are safe to use in this way, and additionally places some of the burden on validating resolvers to ensure this safety.

The approach establishes a mechanism allowing the signer to determine which RRSIGs can be skipped, without risking validation failures. It does not require all algorithms' RRSIGs to be present, while ensuring that the set of signatures provided is still "large enough" for reliable DNSSEC operation, so that robust multi-signer operation and TA pre-publication are made possible, without risking validation failures.

For the case of a multi-signer setup with two generally supported algorithms (such as 8 and 13, see [DNSKEY-IANA]), the scheme requires only one of the two signatures. Similarly, when pre-publishing a trust anchor, associated signatures don't need to be published immediately, provided that the existing TA's algorithm is generally supported.

3.1. UNIVERSAL and FORMERLY-UNIVERSAL Validation Support

The notion of UNIVERSAL signing algorithms is introduced, as described in Section 6.

Initially, algorithms 8 and 13 are the only algorithms designated to have UNIVERSAL validation support.

As soon as a UNIVERSAL algorithm is known or expected to have declining validation support, it should be moved to FORMERLY-UNIVERSAL. Initially, no algorithms are declared FORMERLY-UNIVERSAL.

Algorithms that are neither UNIVERSAL nor FORMERLY-UNIVERSAL are called NEVER-UNIVERSAL. They have an empty value in the corresponding registry column.

3.2. Signer Requirements

1. Absent any UNIVERSAL algorithms in the DS RRset or trust anchor set, or when any FORMERLY-UNIVERSAL algorithms are present, signers MUST sign with all algorithms listed.

2. Otherwise, signers MUST sign with at least one UNIVERSAL algorithm listed in the DS RRset or trust anchor set. Other signatures are OPTIONAL.

These rules are summarized in Table 1 which indicates the signer requirements depending on the composition of the DS record or trust anchor set.

UNIVERSAL and FORMERLY-UNIVERSAL algorithms SHOULD NOT appear together in a DS RRset or trust anchor set. In fact, FORMERLY-UNIVERSAL algorithms are best avoided: signers SHOULD transition to other algorithms that are UNIVERSAL.

	FORMERLY-UNIVERSAL		
	no	yes	
UNIVERSAL	no	all algorithms	all algorithms
	yes	one UNIVERSAL algorithm	all algorithms

Table 1: Signer requirements, depending on DS RRset / trust anchor composition

Depending on the presence of UNIVERSAL and/or FORMERLY-UNIVERSAL algorithms, signatures may be required for all algorithms, or for just one. The presence of NEVER-UNIVERSAL algorithms is not relevant for determining whether signatures for all algorithms are required (but if so, their signatures MUST be included).

3.3. Validator Requirements

1. When the DS RRset or trust anchor set for a zone includes an unsupported UNIVERSAL or FORMERLY-UNIVERSAL algorithm, validators MUST treat the zone as unsigned, even if the DS RRset or trust anchor set lists another supported algorithm.

2. Otherwise, validators MUST accept any valid path.

These rules allow determining a zone’s security status by inspection of the DS record or TA set alone, independently of which (compliant) subset of signatures is served by a particular nameserver.

Implementing these rules requires validators to keep a record of unsupported algorithms that it is still expected to support (UNIVERSAL) or once was (FORMERLY-UNIVERSAL).

Disabling any UNIVERSAL or FORMERLY-UNIVERSAL algorithm in a validator without implementing these rules will cause the algorithm to be treated like a never supported algorithm (that is, as NEVER-UNIVERSAL). This risks zones turning "bogus", if that algorithm is used as the only signing algorithm by one signer in a multi-signer setup, whereas the correct security status would be "insecure" (as the disabling is a matter of local policy).

4. Discussion

Validators, when configured to disable an algorithm, only need to know whether the disabled algorithm ever was a UNIVERSAL one, which includes currently FORMERLY-UNIVERSAL. Validation rules depend only on this binary distinction; tracking of an algorithm moving from UNIVERSAL to FORMERLY-UNIVERSAL is not required. Implementation therefore can be easily achieved by storing a joint list of algorithm numbers which at any time were UNIVERSAL (regardless of whether the algorithm has moved to FORMERLY-UNIVERSAL).

The new validation requirements enable stable multi-signer setups using UNIVERSAL algorithms as well as robust provider transfers and algorithm upgrades from FORMERLY-UNIVERSAL to UNIVERSAL algorithms, without risking SERVFAIL responses in the event that a validator no longer supports one of the algorithms. For a detailed discussion, see Security Considerations (Section 7).

If no FORMERLY-UNIVERSAL algorithm is in use, but at least one UNIVERSAL one is present, DNS operators are free to limit their responses to serve signatures for one UNIVERSAL algorithm only. This one signature is sufficient to provide a valid path everywhere; other signatures are not required. DNS providers are thus free to introduce additional algorithms without forcing other participating providers to do the same. This includes both additional UNIVERSAL algorithms, as well as other NEVER-UNIVERSAL algorithms (e.g., experimental ones, or algorithms with limited adoption).

When trust anchors are in use for a zone and there is one with a UNIVERSAL algorithm, it is permissible to introduce a new trust anchor for a different algorithm before introducing the corresponding DNSKEY and RRSIGs into the zone. (Of course, they need to be added before the old trust anchor is removed.)

If the added trust anchor is also for a UNIVERSAL algorithm, it is permissible to eventually switch to returning just the RRSIGs for the new algorithm, without an intermediate dual-signing period. If the new trust anchor is not yet UNIVERSAL, a dual signing period is required in order to complete the algorithm rollover.

In typical cases, particularly in the case of the root zone, both algorithms will be UNIVERSAL. In a hypothetical emergency situation where only the new algorithm is UNIVERSAL and the old was just downgraded to FORMERLY-UNIVERSAL, the new signatures would need to be introduced immediately. A short dual signing period would then be required for continuity. Validators would be expected to defer disabling the old algorithm until after the emergency rollover is completed.

5. Example Scenarios

This section elaborates how the signer and validator requirements impact various scenarios in practice. The algorithm combination stated at the beginning of each scenario refers to algorithms advertised in the DS RRset or trust anchor set.

Only one algorithm (potentially several keys): Signers MUST sign with at least one of the keys, and validators MUST accept any valid path. If the validator does not support the algorithm, the zone is insecure.

Several UNIVERSAL algorithms, no other algorithms: Signers MUST sign with at least one of the algorithms, and validators MUST accept any valid path.

At least one UNIVERSAL algorithm and a NEVER-UNIVERSAL algorithms: Signers MUST sign with at least one UNIVERSAL algorithms, and validators MUST accept any valid path.

At least one FORMERLY-UNIVERSAL algorithm: Signers MUST sign with all algorithms. Validators not supporting the FORMERLY-UNIVERSAL algorithm MUST treat the zone as insecure (regardless of their support for other advertised algorithms); other validators MUST accept any valid path.
This applies regardless of the presence of any UNIVERSAL or NEVER-UNIVERSAL algorithms.

6. IANA Considerations

[to be removed by RFC Editor: this section assumes draft-ietf-dnsop-rfc8624-bis is published.]

This document requests that IANA update the "DNS Security Algorithm Numbers" registry ([DNSKEY-IANA]) with the additional column "Validation support status".

Admissible values for this column are "UNIVERSAL", "FORMERLY-UNIVERSAL", and empty. The value "UNIVERSAL" is only acceptable for rows where the value of the "Implement for DNSSEC validation" column is "MUST".

The default value of the new column for existing and new rows is empty. Changing the value of the column requires standards action.

Initially, algorithms 8 and 13 are the only algorithms declared UNIVERSAL. No algorithms are initially declared FORMERLY-UNIVERSAL.

7. Security Considerations

The new validation requirements presume that zones using multiple algorithms are either in a state of transition (e.g., when switching providers) or in a permanent multi-provider configuration. In the first case, if the outgoing algorithm is not supported by the validator, the zone would have been treated as insecure before the transition. For the second case, it is noted that the purpose of multi-provider setups is to provide resilience against any single provider’s failure. Consequently, the zone owner is assumed to consider the security guarantees given by any single provider to be acceptable for the whole zone. By implication, if one of the providers has fallen behind and is signing with an algorithm that is no longer supported by some resolvers (and thus promises no security), there is no guarantee of DNSSEC security for the zone.

In other words, the validation requirements guarantee that a zone in a multi-provider setup has the same security level as if all but one of the involved providers would be unavailable. Consequently, when the configuration involves an algorithm that is no longer universally supported, non-supporting validators treat the zone as insecure. This resolves undue SERVFAIL issues that could occur with certain algorithm combinations under the previous rules.

Example: A zone using only an algorithm that is declared FORMERLY-UNIVERSAL is treated as insecure by validators that do not support this algorithm. (This is as before.) When transferring the domain, via a multi-signer setup, to another provider which uses a currently UNIVERSAL algorithm, however, the zone's security status will now remain "insecure", as the DS RRset still includes the FORMERLY-UNIVERSAL algorithm. The presence of the UNIVERSAL algorithm is inconsequential at this point. Only once the old algorithm is removed, the zone turns secure.

This rule acknowledges the fact that the signer is using a FORMERLY-UNIVERSAL algorithm that SHOULD NOT be used for signing, which might render the zone insecure for validators that lack support. This prevents validation breakage when the validator encounters an unsupported RRSIG from an outdated algorithm, and allows for glitch-free algorithm upgrades with the security status of the zone changing only once the transition is complete.

Validators supporting both algorithms retain security throughtout the transition. In case of a permanent multi-signer setup, the zone maintainer needs to move from the FORMERLY-UNIVERSAL algorithm to a UNIVERSAL one in order to restore universal validation.

7.1. Time Dependency of UNIVERSAL Algorithms

The same situation occurs when an algorithm is removed from the set of UNIVERSAL algorithms. In this case, the algorithm will become FORMERLY-UNIVERSAL. If the zone continues to use the FORMERLY-UNIVERSAL algorithm, it will continue to be accepted by supporting validators, while non-supporting validators will treat the zone as insecure until the algorithm is replaced.

Conversely, when an algorithm is added to the set of UNIVERSAL ones, signers MAY begin to return signatures for just that algorithm. This is, in fact, not a problem, as validators do not need to know the concept of UNIVERSAL; they just need to support that algorithm (or later classify it as FORMERLY-UNIVERSAL). A problem could only occur if the corresponding RRSIG was not supported by a non-negligible population of validators; however, in that case labeling the algorithm as UNIVERSAL would have been premature. Determining

universal support cannot be solved on the protocol level, and it is the community's responsibility to only advance an algorithm to UNIVERSAL when safe enough, i.e. when the population of validators lacking support is deemed negligible.

Validators dropping support for FORMERLY-UNIVERSAL algorithms without implementing this specification will produce SERVFAIL responses for multi-signer setups involving the disabled algorithm. Implementation of the new validation rules is thus advised as soon as support for an algorithm is dropped.

7.2. Variable Key Size Algorithms

Since algorithm 8 supports variable key sizes, multi-signer configurations involving 8 and 13 should take care to employ an RSA keylength that is computationally infeasible to attack.

8. Acknowledgements

In order of first contribution or review: Philip Homburg, Libor Peltan, Stefan Ubbink

9. Normative References

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

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

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

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

[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/info/rfc6840>.

[RFC8624] Wouters, P. and O. Sury, "Algorithm Implementation Requirements and Usage Guidance for DNSSEC", RFC 8624, DOI 10.17487/RFC8624, June 2019, <https://www.rfc-editor.org/info/rfc8624>.

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

10. Informative References

[DNSSEC-AUTO] Wisser, U. and S. Huque, "DNSSEC Automation", <https://www.ietf.org/archive/id/draft-ietf-dnsop-dnssec-automation-01.html>.

Appendix A. Current Multiple Algorithm Rules

This section discusses the multi-algorithm requirements on signers and validators, as specified by the original DNSSEC specification and in effect until updated by this document. It is included for purely informational purposes and context.

A.1. Signing Requirements

In addition to the last paragraph of [RFC4035] Section 2.2 quoted earlier, Section 5.11 of [RFC6840] clarifies:

> A signed zone MUST include a DNSKEY for each algorithm present in the zone's DS RRset and expected trust anchors for the zone.

While it might seem tempting, relaxing this rule without any further adjustments may not be safe depending on the algorithm combination involved. In particular, when using an algorithm that is not universally supported among the resolver population (such as algorithm 7) together with a supported one (such as algorithm 13), resolvers may return SERVFAIL under certain circumstances. Zone owners and signers thus would have to take great care to not leave a validating resolver without a valid supported path in such situations, e.g., when transitioning from algorithm 7 to 13.

More explicitly, when the sole signing algorithm used by a zone is not supported by a given resolver, the resolver will (correctly) treat that zone as unsigned. However, when attempting to transfer the domain to another DNS provider through a multi-signer setup with a supported algorithm, affected resolvers presented with the unsupported signature only will not be able to distinguish this situation from a downgrade-to-insecure attack where the second signature has been stripped, and will return SERVFAIL.

Although unstated in that document, the above rule prevents this kind of downgrade-to-insecure attack by requiring RRSIGs for all advertised algorithms; a validator can thus assume that something is wrong when supported signatures are missing. As a side effect, the rule also protects against downgrade-to-weaker attacks, where an attacker would strip away signatures from signed DNS responses and only attach one for an algorithm that the attacker is able to forge. This property is not a core guarantee of DNSSEC (see below).

A.2. Validator Requirements

In general, when a validating resolver supporting any of the algorithms listed in a given zone’s DS record or TA set responds to a query without the CD flag set, it may not treat that zone as insecure, but must return either authenticated data (AD=1) or SERVFAIL (RCODE=2). For this purpose, any valid path suffices; the validator may not apply a "logical AND" approach to all advertised algorithms.

Accordingly, Section 5.11 of DNSSEC Clarifications [RFC6840] states:

> This requirement applies to servers, not validators. Validators SHOULD accept any single valid path. They SHOULD NOT insist that all algorithms signaled in the DS RRset work, and they MUST NOT insist that all algorithms signaled in the DNSKEY RRset work.

At first glance, the assertions that (1) the signer provide signatures for all advertised algorithms while (2) the resolver shall be content with just one seems somewhat contradictory. However, the role of the RRSIG rules is to ensure that the resolver will find a valid path (using a "logical OR" strategy), regardless of which particular algorithm(s) it supports, and thus be able to distinguish reliably between "all is in order" (validated data) and a downgrade-to-insecure attack (SERVFAIL).

A.3. Incompatible Use Cases

The above rules are incompatible with certain use cases:

* They are impractical to satisfy if DNS providers deployed in a multi-signer configuration are using different signing algorithms. By extension, it also means that multi-signer techniques cannot be employed to non-disruptively transfer a signed zone from one DNS provider to another if the providers use differing algorithms.

* The rules further collide with the conflicting goal of pre-publishing the new trust anchor during a zone's algorithm rollover, while introducing the new algorithm into the zone only later in the process.

* Furthermore, for online signers attempting to deploy multiple algorithms, producing signatures for several algorithms also imposes a significant computational burden, unless a selective algorithm negotiation mechanism is also developed.

As the above rules present a severe limitation for these use cases, this document proposes to relax them in a way so that the set of signatures provided is still "large enough" to ensure reliable DNSSEC operation, while facilitating the above use cases.

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

draft-huque-dnsop-multi-alg-rules-07

* Add tl;dr: Nutshell Proof of Sanity

* Editorial feedback from Stefan Ubbink

* Clarify what a validator needs to know

* Initially don't declare any algorithms FORMERLY-UNIVERSAL

* Clarify new column update requirements for IANA

* No longer updates RFC 8624 (assumes publication of 8624bis)

draft-huque-dnsop-multi-alg-rules-06

* Fix IANA considerations

* Editorial changes (add change log, ...)

* Add overview of cases, and scenario descriptions

* Clarify what to do when both UNIVERSAL and FORMERLY UNIVERSAL algorithms are present

Authors' Addresses

Shumon Huque
Salesforce
Email: shuque@gmail.com

Peter Thomassen
deSEC, SSE
Email: peter@desec.io

Viktor Dukhovni
Google LLC
Email: ietf-dane@dukhovni.org

Duane Wessels
Verisign
Email: dwessels@verisign.com

Christian Elmerot
Cloudflare
Email: elmerot@cloudflare.com