

Multiple Algorithm Rules in DNSSEC

[draft-huque-dnsop-multi-alg-rules](#)

IETF 117 – DNSOP WG

July 24, 2023

Shumon Huque, [Peter Thomassen](#), Viktor Dukhovni

Motivation

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).

RFC 4035 Section 2.2

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

[...] 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.

RFC 6840 Section 5.11

Motivation

- Root zone algorithm rollover (will happen eventually):
 - **Double-signing has risks** (key management, amplification, bandwidth, transport limitations, ...)
 - Upcoming report by ICANN's "Root Zone Algorithm Rollover Study Design Team" finds it would help to **add trust anchors before introducing the new algorithm into the zone**
→ Start double-signing only once TA has some meaningful deployment
- DNSSEC multi-signer setups:
 - Several DNS providers signing and serving the same zone
 - **May not be using the same algorithm(s)**
 - Also needed during **provider change without disruption** (if algorithms are not the same)
- Computational burden for online signers producing several RRSIGs on the fly

 *Do we really need to require presence of all signatures, always?* 

Can't we just relax the rules?

- How about just **relaxing the RRSIG presence requirements, requiring just one?**
 - **Very straightforward**, but risks SERVFAIL when resolver supports only one of the algorithms
 - **Signers beware!** Burdened to *make sure at least one widely supported RRSIG is served*.
 - Murky picture for resolvers, possible incentive to downgrade vs. fail (e.g. 7 + 13 or 13 + 15)
- Existing requirements well-motivated:
 - A zone may advertise multiple algorithms via DS/TA
 - Resolvers may not (yet) support certain algorithms
 - **Attacker may strip all supported RRSIGs** from a signed response (leaving unsupported ones)
 - The stripped response is BOGUS from the resolver's perspective
 - **MUST not result in DNSSEC being disabled** / zone downgraded as "insecure"
→ Resolver MUST return SERVFAIL
- Rules ensure downgrade resistance when advertising multiple algorithms
 - Not too much of a concern when an old algorithm is used which resolver no longer supports
 - **But: New, non-deprecated algorithms must not be a downgrade vector** (else no upgrade path!)

Algorithm Lifecycle / Taxonomy

- **Youth: Insufficient**
 - Algorithms that don't (and might never have) effectively universal adoption
 - Most relevant to signers.
- **Golden years: Sufficient**
 - Algorithms with universal adoption (e.g., unsupported by only a negligible population of resolvers)
 - Most relevant to signers.
- **Sunset years: Insecure**
 - Primarily relevant to resolvers.
 - Algorithms that were once Sufficient, but are **now unsupported** in a non-negligible population of resolvers (administratively disabled or unavailable on current platform).
 - Alternative term: “Deprecated” (but appears to cover algorithms that were never “Sufficient”). Perhaps “Legacy”.
 - **Signers are advised to transition away** from such algorithms once found to be in sunset phase

Proposal

- When a zone advertises **at least one Sufficient signing algorithm** via DS or trust anchors, a signer **MAY skip publishing RRSIGs for all other advertised algorithms**.
- The signer **MUST** otherwise publish RRSIGs for all advertised algorithms.
- Resolvers: any valid path suffices

What happens if a formerly **Sufficient** algorithm is sunsetting?

- Outdated signer may still be publishing RRSIGs for only this algorithm → **no valid path**
 - In such invalid configurations, resolvers **MUST** treat responses with RRSIGs for only “Insecure” algorithms as **“insecure” rather than “bogus”**
 - They **MAY** treat zones advertising such an algorithm as unsigned
 - skipping validation even if supported RRSIGs happen to be returned
- If "Insecure" algorithms are advertised, some resolvers will not support any of them. Owner should expect zone to be effectively unsigned for these resolvers.

Key Benefits

1. Root zone can transition between Sufficient algorithms (8 and 13) without dual-signing
2. Zones signed with Insecure/Deprecated/Legacy algorithm can move to a new provider with Sufficient algorithm, without disruption
 - If neither provider supports the other algorithm
3. Long-term multi-signer deployments can support distinct Sufficient signatures at separate providers (presently one with 8 and the other with 13)
4. Resolver-side implementation straightforward and not time-critical
 - Need only to keep list of Insecure/Deprecated algorithm numbers **once disabled**
 - No need to specifically address any of the use cases explained earlier

Summary

- Only DS/expected TA algorithms matter
 - No requirement to sign with algorithms listed only in apex DNSKEY, but not in DS
 - Multiple “sufficient” (“universal”) algorithms can be used without dual signing
 - Just like with single-algorithm ZSK rollovers
- **Much simpler common-case rollovers, multi-provider support, and switching DNS operators**
- Zones using deprecated algorithms may be downgraded to unsigned (even if also publishing a non-deprecated algorithm)