Comments on “Proposal for Root Zone KSK Algorithm Rollover”

Reviewer: Michael C StJohns, msj@nthpermutation.com, +1-301-633-3956

Date: 12 Feb 2026

Summary: Needs work. Do not proceed as written. The plan is missing most information about the decision process for the critical choices. This will restrict most reviews to a somewhat superficial level. Minor 7 indicates a flaw with publication timing. Major 1 suggests an approach that does not require reducing the signature strength of the root zone for 3 years.

Nits

1. (Editorial) As a general matter, please provide section numbering, page numbers and ideally line numbers on the document to be reviewed. Please also date or otherwise indicate specific versions of this proposal.
2. (Editorial) “slot” is use numerous times in the body of the document in a manner that isn’t explained until the table at the end. Given that HSM’s also have “slots” and “slot” has a normal English meaning, please capitalize this word wherever you’re using it to refer to a specific working time period. E.g. *“Each quarter is divided into nine publication Slots of approximately ten days, with the final Slot extended as needed to complete the quarter.”*
3. (Technical) “Algorithm Selection”, 3rd para, sentence about “Luna G7 HSM” – According to the NIST list of FIPS validated HSMs, the certificate for the Luna G7 HSM expires 2/6/2027. A reference in this document to the operational plans for the HSMs appears to be warranted.
4. (Editorial/Technical) “Prerequisite: Reducing the RSA KSK Sizes” mixes bit-length and byte-lengths and doesn’t actually show the math for a reviewer to conclude these will actually fit. E.g. show the on-the-wire length of a DNSKEY encoded key for each type, and the total length of the response to a DNSKEY query of the root. Also show the length of the RRSIG (DNSKEY) as well.
5. (Editorial) “Operational Plan” The last sentence of the first paragraph along with the “Slot” definition at the end should be merged – either in the table at the end or somewhere before the term “Slot” is used.
6. (Technical/Editorial) “Key Signing Ceremonies” For each “KSR” in each phase, provide a specific list of the keys and signatures for each of the CC-DD, CC-CC, DD-CC and DD-DD grouping and what those keys are signing. The authors might know what these are, but the text is ambiguous.

7. (Technical) “Key Signing Ceremonies”- the phrase “becomes active” is ambiguous. If mean they will be published along with their DNSKEY group, then say so. If it’s something else, explain.
8. (Editorial/Technical) Phases: Include a state transition diagram including loops and backward transitions.

Minor

1. (Technical) “Key Signing Ceremonies” The pre-production of the SKRs may have some security implications if they become public or available outside of ICANN. Please describe the protection domains in place for these prior to publication.
2. (Technical) “Key Signing Ceremonies” If ICANN stops publishing CC-DD and then publishes DD-CC, what is to stop an attacker from using “CC-DD” in an attack along with all the other signed records?
3. (Technical) “Phases” doesn’t appear to include the RSA ZSK size reduction process. From the text, it’s unclear which keys comprise the pre-CC steady state.
4. (Operational/Editorial) “Phases” Phases AA and BB do not involve changes to the published DNS data. They are tasks that need to be completed, but they are prerequisite tasks to phase CC. It’s unclear why they’re listed as phases. Isn’t there a reference document for the HSM management process that could be used as a cross-reference instead of this text? Merely note that this has to happen with any generated key pairs.
5. (Technical or Editorial?) “Phase DD: Signature Publication (ECDSA)” – “Key Ceremony Actions”. There should be no “CC-CC” as you’re in phase DD. If you executed the publication of “DD-CC” signed with the CC phase, you should be back in the CC phase.
6. (Technical) “Phase DD: Signature Publication (ECDSA)”. The description here and other places lacks details on why you’d publish signatures without the DNSKEY public keys. If there’s a good reason, explain it. If the intent is to hide the public keys, note that it is possible to derive the public key related to the ECDSA signing key from a couple of signatures. See https://en.wikipedia.org/wiki/Elliptic_Curve_Digital_Signature_Algorithm - at “Public Key Recovery”
7. (Technical/Operational) – “Phase FF: Revocation of RSA KSK”. “Beginning on the 81st day…the RSA KSK … and their signatures will be removed from the root zone”. I would suggest continuing to publish the RSA KSK with the REVOKED bit and its signature for at least the normal duration of the RRSIG(DNSKEY) by that revoked key. The actual

revocation happens as soon as a client sees the self-signed, revoked key, but that could happen as late as 30 days after the caching time of the KSK being revoked. 70 days publication is probably marginal for best coverage.

Major

1. (Technical/Operational): "Prerequisite: Reducing the RSA ZSK Size". This section contemplates reducing the RSA ZSK so one of the smaller ECDSA ZSKs may also be included in the root DNSKEY RRSET. That has the side effect of reducing the overall security of the root zone down to about 90-95 bits (from 112 bits) for an extended period. My question here is "Why do it this way?" Instead, replacing the current RSA ZSK with an ECDSA ZSK as one step eliminating the interim, 3-year weaker key step. If ECDSA is viable, there is no reason not to use it to sign the non-DNSKEY records.
    a. ZSK Transition
        i. Initial state – one RSA2048 ZSK and one RSA2048 KSK
        ii. Interim state – one RSA2048 ZSK, one ECDSA P256 ZSK and one RSA2048 KSK
        iii. Final ZSK transition state – one ECDSA P256 ZSK and one RSA2048 KSK
    b. KSK Transition
        i. Initial state – one ECDSA P256 ZSK and one RSA2048 KSK
        ii. Interim state (adding) – One ECDSA P256 ZSK, one RSA2048 KSK, and one ECDSA P256 KSK (Pending)
        iii. Interim state (hold down complete) – One ECDSA P256 ZSK, one RSA2048 KSK and one ECDSA P256 (Active) KSK
        iv. Interim state (revoked) – One ECDSA P256 ZSK, one RSA2048 KSK (REVOKED), and one ECDSA P256 KSK
        v. Final state (revoked removed) – One ECDSA P256 ZSK and one ECDSA P256 KSK.
    c. Commentary. The above *MAY* conflict with RFC4035's signing requirements, but the DSNOP WG has this draft https://datatracker.ietf.org/doc/draft-huque-dnsop-multi-alg-rules/ in place and it may be an RFC or on publication track by the time this gets started – it's already twiddling 4035's requirements. From my personal opinion, I would rewrite section 2.2 of RFC 4035 as follows as I think it matches the intent better:

        > For the apex DNSKEY RRSet, there must be at least one RRSIG(DNSKEY) for each signing algorithm represented by a KSK in that RRSet. For the remainder of the data in the zone, all other

> RRSETs MUST have at least one RRSIG per unique ZSK key algorithm represented in the apex DNSKEY RRSet.

I.e., there are/should be separate “MUST sign” rules for DNSKEY KSKs vs DNSKEY ZSKs.