Making Music With DNSSEC: Status Update

The Need to Avoid False Notes

Johan Stenstam Roger Murray

The Swedish Internet Foundation

June 3, 2022

MUSIC: The Multi-Signer Conductor

Since autumn 2021 we have worked on a project that what we call **MUSIC**.

This is an implementation of the processes described in the "multi-signer" Internet-Draft by Shumon Huque and Ulrich Wisser.
For an introduction, please see the presentation from the DNSSEC Workshop during the March 2022 ICANN meeting or the March 2022 CENTR Tech meeting.

MUSIC works.

- But, as always, there were (and are) some things that certainly would benefit from more thought.
- We have identified two such "false notes" that needed additional attention and they are presented here.

This is a status update of where the **MUSIC** project stands right now.

Treatment of Combined Signing Keys (CSKs)

As described in the multi-signer draft, **MUSIC** synchronizes `ZSK`s between multiple "signers".

- The `KSK` is not needed, as that doesn't sign the data in the zone, only some RRsets at the zone apex. I.e. the `KSK`s are only needed to ensure that the correct `CDS` records ar published for subsequent publication of the correct `DS` records in the parent zone.

Problem: However, this logic does not cater to the existence of signers that don't use the `KSK`/`ZSK` split but rather use a single `CSK`, i.e. a combined signing key.

Treatment of CSKs, cont'd

There are basically two ways to deal with this issue.

Alternative 1: Instead of assuming a KSK/ZSK split (and only synchronizing the latter), do a more careful analysis of what DNSKEY is used for what pupose.

- If a DNSKEY signs RRsets in the zone, then sync it with other signers.
- If a DNSKEY has a corresponding DS in the parent zone, then put it into the set of keys that will be the basis for the CDS RRset.

- Pro: this is essentially the "correct" solution.
- Con: this is a bit complex, as we cannot trust the flag bits to describe the use of a DNSKEY.

Treatment of CSKs, cont'd

There are basically two ways to deal with this issue.

Alternative 2: Instead of assuming a KSK/ZSK split (and only synchronizing the latter), just synchronize all DNSKEYs across all signers.

- Pro: this is simple.
- Con: this is crufty and leads to a larger DNSKEY RRset than necessary.

Treatment of CSKs, cont'd

There are basically two ways to deal with this issue.

Alternative 2: Instead of assuming a KSK/ZSK split (and only synchronizing the latter), just synchronize all DNSKEYs across all signers.

- Pro: this is simple.
- Con: this is crufty and leads to a larger DNSKEY RRset than necessary.

Current status: At present **MUSIC** uses the second alternative (synchronize all DNSKEYs) to avoid stalling on this issue.

- We would like to move to the better solution later on.

DNSKEY RRset Size Considerations

As observed already in RFC8901 on Multi-Signer DNSSEC Models the use of any multi-signer process will lead to a DNSKEY RRset with more keys than otherwise.

- This is one of the reasons for only synchronizing ZSKs and not KSKs.

This prompts the question of whether the current implementation of MUSIC (where all DNSKEYs are synchronized) will have issues due to the DNSKEY RRset becoming large enough to be "problematic".

We do not believe this to be the case, because:

- The migration to using elliptic curve DNSKEYs, which are significantly smaller.
- The KSK/ZSK split is essentially an artifact of the DS update in the parent being "complicated". With the emergence of automated DS updates (via CDS/CDNSKEY) it seems likely that over time CSK will replace KSK/ZSK as the primary DNSKEY semantic.

Treatment of Multiple CDS Digest Algorithms

Some signers generate their own CDS records. If a zone uses two signers, A and B, then there are three possible sets of DS digest algorithms:

- The digest algorithms used by signer A, the algorithms used by signer B and the algorithms used by the parent.
- This prompts the question of what digest algorithms to generate CDS records for.

Note that while the parent in the end determines what digest algorithms it will use for published DS records, there may exist additional requirements.

- For example, the .SE and .NU registries run CDS scanning in production and for a CDS RRset to update the DS the scanner requires that the CDS RRset is identical across all authoritative nameservers.

Multiple CDS Digest Algorithms, cont'd

For this reason it is not sufficient to only publish CDS records based on the union of all KSK/CSK from all signers.

- **MUSIC** must also poll all CDS RRsets from all signers.
  - To identify all Digest Algorithms in use for CDS records not created by **MUSIC**.
- This is needed to ensure that the published CDS RRset is consistent across all signers.

Current status: MUSIC does not yet poll all signers to ensure that the CDS RRset is consistent everywhere.

- This is not difficult, we just have not had the time.

Support for the deSEC API

One of the outstanding deliverables from earlier this spring was the addition of support for the deSEC provisioning API.

This is now working.

The deSEC signer implements the `CSK` signing model (see previous slides)

- This prompted additional attention to the issues surrounding different signers with a mix of `KSK`/`ZSK` and `CSK` models.

Code: `https://github.com/DNSSEC-Provisioning/music.git`

Contact: **`music@internetstiftelsen.se`**

Thanks!

Johan and Roger