Multi-signer DNSSEC with NS1 Managed DNS

Jan Včelák
Software Engineer at NS1

ICANN 75 DNSSEC and Security Workshop
DNSSEC Provisioning Automation Panel
2022-09-27

NS1.

RFC 8901 Multi-Signer Models Support

Model 1
Common KSK, unique ZSK per signer

Needs an outside process to collect DNSSEC public keys from the signers and keep DNSKEY+RRSIG at each DNS vendor in sync.

This creates a single point of failure.

It is arguable whether the single KSK improves security.

We used to have proof of concept for this about 3 years ago. Now discontinued.

Model 2
Unique KSK and ZSK per signer

Also needs an outside process to exchange the public keys between signers however the signers now act independently.

No longer a single point of failure. One signer may fail but the other will still work.

More resiliency and easier to operate.

Model 2 is supported by NS1.

NS1.

DNSSEC Signing with NS1

Minimal configuration: Just **enabled or disabled**.

We use **KSK and ZSK** operational model.

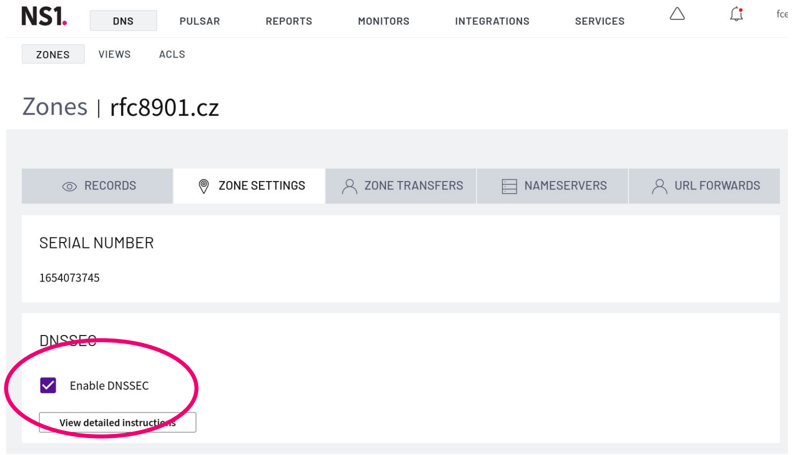
DNSSEC keys are managed by NS1.

DNSKEY records are pre-signed. Everything else is signed **on-the-fly**.

ECDSA-P256-SHA256 (13)

Authenticated denial using **Compact NSEC proofs**

Enabling DNSSEC with NS1



Zones | rfc8901.cz

SERIAL NUMBER

1654073745

DNSSEC

☑ Enable DNSSEC

OR

```
POST /v1/zones/rfc8901.cz
{
  "name": "rfc8901.cz",
  "dnssec": true
}
```

Multi-signer Configuration

All what is needed to support RFC 8901 Model 2 is allow for additional DNSKEY records that are published in the zone but not used for signing.

NS1 has an HTTP API endpoint for this purpose:

```
PUT/GET/POST/DELETE
/v1/zones/{zone}/dnssec/external_keys/{key_set_name}
```

https://help.ns1.com/hc/en-us/articles/7620400771731-Managing-external-DNSSEC-keys

Example: Add external DNSKEYs

```
PUT /v1/zones/rfc8901.cz/dnssec/external_keys/knot
{
  "dnskey": {
    "data": [
      {
        "algorithm": 13,
        "flags": 257,
        "protocol": 3,
        "public_key": "OCgb3x3LNtRIwH2lCOtqy9/znf8ZJSNwatYuEFLUTcp5O6D/sFptMP0w..."
      },
      {
        "algorithm": 13,
        "flags": 256,
        "protocol": 3,
        "public_key": "56HkdxlXMa00cuEsDXPSaviUaXY5bZqYekQEBUY3SdKtxo5nSB/WlFP6..."
      }
    ]
  }
}
```

Example: DNS responses

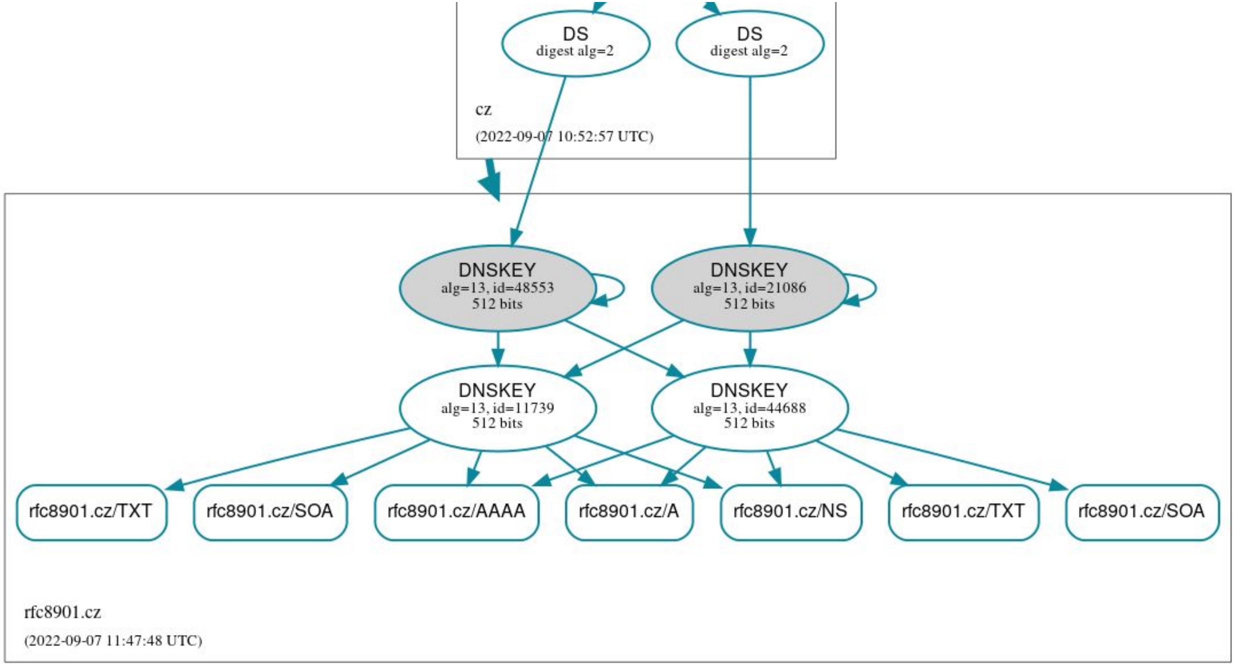
```
$ dig rfc8901.cz. NS +short
a.ns.fcelda.cz.
dns1.p01.nsone.net.

$ dig rfc8901.cz. DNSKEY +short
257 3 13 t+4DPP+MFZ...    ; 48553 (NS1)
256 3 13 pxEUulkf8U...    ; 44688 (NS1)
257 3 13 OCgb3x3LNt...    ; 21086 (external)
256 3 13 56HkdxlXMa...    ; 11739 (external)

$ dig @a.ns.fcelda.cz. rfc8901.cz. TXT +short +dnssec
"Signed and served by Knot DNS."
TXT 13 2 1200 20220921103023 20220907090023 11739 rfc8901.cz. XlLyKfTPUB...

% dig @dns1.p01.nsone.net. rfc8901.cz. TXT +short +dnssec
"Signed and served by NS1."
TXT 13 2 1200 20220908115433 20220906115433 44688 rfc8901.cz. M5ASt0I8LS...
```

Example: DNSViz Diagnostics



```
48553 (NS1)
44688 (NS1)
21086 (external)
11739 (external)
```

https://dnsviz.net/d/rfc8901.cz/YxiE5A/dnssec/

Final Notes

External keys can be currently managed by the **REST API only.** The portal support will be eventually added.

CDS and CDNSKEY are not supported yet but we are working on it.

To support key rollovers, DNSKEY records need to be synchronized between the providers regularly and timely. We are considering implementing integration for "common providers" in the NS1 platform.

We validated the implementation with BIND, Knot DNS, deSEC, and Cloudflare (feature flagged).

Advice for implementers: Focus on Model 2. It it simpler and providers better value.

Thank you!

Jan Včelák
Software Engineer at NS1

jvcelak@ns1.com in/janvcelak @fcelda

NS1.