Multi-Signer Key-Exchange (MSKE)

draft-thomassen-dnsop-mske

IETF 115 – DNSOP WG
November 8, 2022

Peter Thomassen (deSEC, Secure Systems Engineering)

Context: DNS Resolution in a Multi-signer Setup

- **Separate queries** for RRset of interest and for validation DNSKEY
  - RRset of interest (incl. RRSIG) and DNSKEY **may be received from different providers**
  - Resolvers need not to know anything multi-signer (RFC 8901)
    → **DNSKEY responses must contain all keys** a validating resolver may need for validation

- Need to **include other providers' keys**
  - Which? … **depends on qtype**
  - **DNSKEY validation requires *local* DNSKEY only** (RRSIG is always from the same provider)
  - **Everything else** may need **another provider's ZSK** for validation

- **DNSKEY RRset = (local DNSKEYs) U (other provider's ZSK-type DNSKEYs)**

- Also, **CDS/CDNSKEY** = union of **(all provider's KSK-type DNSKEYs)**

Requirements

- Providers need to …
  - … **learn each others' KSKs**, to announce them in their CDS/CDNSKEY RRsets
  - … **learn each others' ZSKs**, to announce them in their DNSKEY RRset
  - High-level description in draft-wisser-dnssec-automation, but key exchange problem left open

- How to **enable providers' exchange of signing keys** (public part)?
  - Which channel?
  - How to decide whether a KSK also serves as a ZSK (CSK)?
  - How to identify obsoleted keys? (cannot be inferred from apex DNSKEY RRsets alone)

- Which properties would be nice?
  - **In-band**
  - **Authentication**
  - **Explicit** is better than implicit

Proposal 1: Authenticated Key Announcement / Retrieval

- Use signaling mechanism from draft-ietf-dnsop-dnssec-bootstrapping using `_multi` prefix (“signaling type”)
  - Each provider signals **only *their* public keys** (for which they have the private key; no union!)
  - Uses nameserver hostnames’ subdomains → **Requires nameserver zones to be secure**

- Signaling Records:
  - KSK-type keys go into CDS/CDNSKEY record set (updates RFC 7344)
  - ZSK-type keys go into DNSKEY record set (updates RFC 4034)
    → Each provider authoritatively **declares each key’s usage type**

- **Key collection** done for all providers, e.g.:
  - KSKs: `CDS/CDNSKEY IN _multi.example.co.uk._signal.ns1.provider.net.`
  - ZSKs: `DNSKEY IN _multi.example.co.uk._signal.ns1.provider.net.`

Proposal 2: Triggering Key Synchronization

- When establishing a multi-signer setup, new provider is **not yet** in NS RRset
  - **How do providers discover each other?**
- Discovery via new record type: CNS
  - Holds *prospective* NS hostnames
  - Analogous to how CDS holds prospective DS records
- Zone owner can put **NS hostnames of all involved parties** into CNS RRset
  - "Source of truth" that **tells each operator where to pull keys** from
  - Also works to **trigger sync for key roll**
- Name inspired by other C* record types
  - Lives on child-side, like CDS/CDNSKEY/CSYNC
  - Used to convey zone configuration (to peers though, not parent)

Multi-Signer Key Exchange: Example Workflow

1. Initial: `example.co.uk` with DNSSEC, Provider A (`ns1.provider-a.net`, …)
2. Domain owner creates zone at Provider B (`ns-a.provider-b.org`, …)
3. Domain owner creates CNS records **at both providers**
   - `@ IN CNS ns1.provider-a.net.`
     `ns2.provider-a.com.`
     `ns-a.provider-b.org.`
     `ns-b.provider-b.io.`
4. Providers A and B observe this, and import (here: Provider B perspective)
   - DNSKEY IN `_multi.example.co.uk._signal.ns1.provider-a.net.` → DNSKEY
   - CDS/CDNSKEY IN `_multi.example.co.uk._signal.ns1.provider-a.net.` → CDS/CDNS…
5. Once DNSKEY and CDS are synchronized: update NS (e.g. EPP or CNS → NS + CSYNC)

What now?

- Proposal solves multi-signer key exchange problem
  - Automated
  - Authenticated
  - In-band
  - Explicit (KSK vs ZSK)
  - Comprehensive: covers onboarding, offboarding, key roll
  - Minimal: (1) signaling mechanism, (2) trigger mechanism
- Not implying this is *the* solution
  - Maybe even no solution is needed
- What does the WG think?
  - Draft: draft-thomassen-dnsop-mske