Migrating an OpenDNSSEC signer (February 2016)

Contributors	Email
David Njuki	david.njuki@afrinic.net
Amreesh Phokeer	amreesh@afrinic.net
Logan Velvindron	logan@afrinic.net
Alain Aina	aalain@trstech.net

Purpose

The goal of this article is to share with the community AFRINIC's experience in migrating a signer from an older version to a newer version of OpenDNSSEC. This, without breaking the validity of zones at any time. Migration was scheduled and was executed over several days while zones were kept secure.

Background

AFRINIC operates reverse DNS for 6 IPv4 zones, and 3 IPv6 zones. In addition to reverse DNS, we also support DNSSEC on those zones, as well as zones such as "afrinic.net". Migrating from an old version of OpenDNSSEC to a newer version which is more stable and secure is not a

straightforward task. There are different strategies that we can adopt to achieve the same result. In this document, we will discuss about different migration options and the choice that we made based on the risk factor each option has. It is important to understand, that the same context may not apply somewhere else.

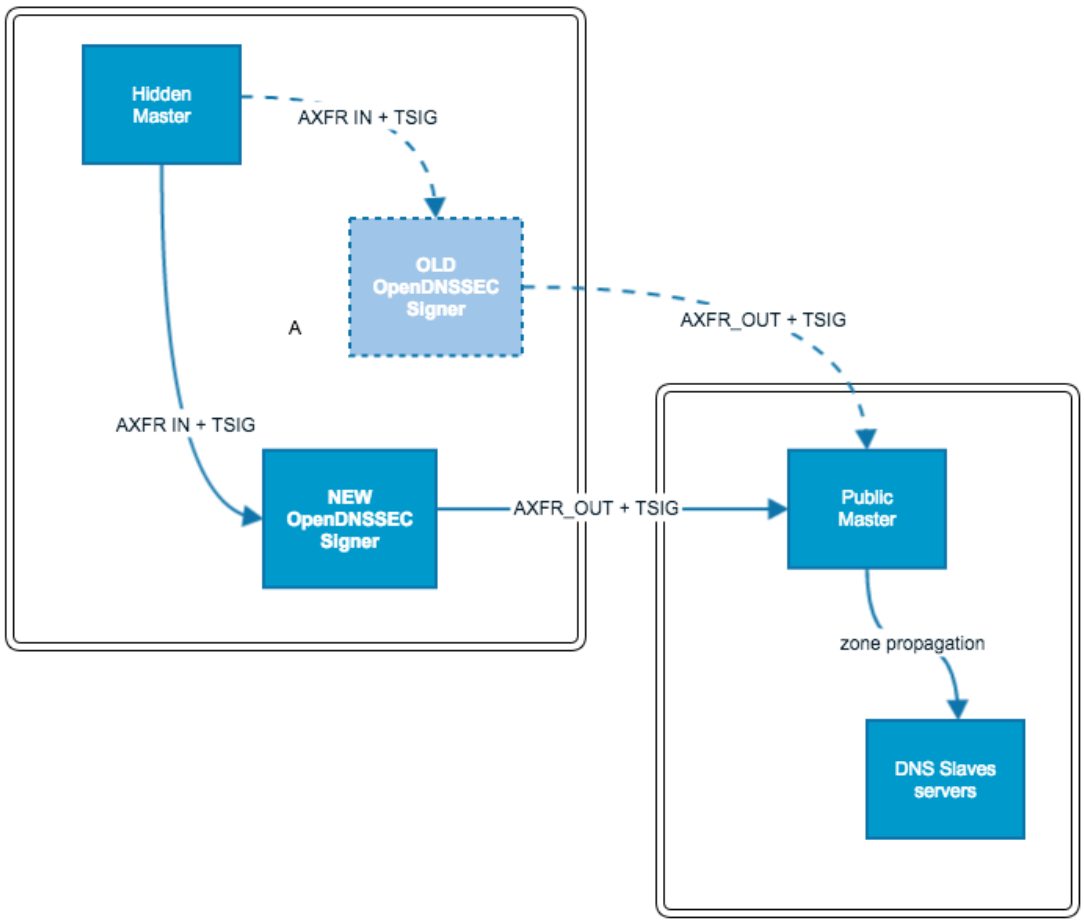
Reasons for migration

Previously, we were using an older version of OpenDNSSEC (v1.3), with sqlite as the backend and we had a few problems with it:

- We were experiencing scalability issues.
- We experienced large delays for signing of zones.
- The old signer was stuck into "flush mode" occasionally, leading to members to complain about time to propagate of their changes.
- Limited support for AXFR IN and OUT

Infrastructure setup

Below is a simplified schema of the DNS provisioning system and servers at AFRINIC. The "Hidden Master" sends unsigned zones to the OpenDNSSEC signer, once signed, the zones are transferred to the "Public Master" which in turn notifies its pool of secondary DNS servers. The "OLD" signer is the current signer which we want to replace by the "NEW" signer.



Migration strategies

There are different migration options that can help achieve the same result, however each of them has an element of risk that should be carefully handled. To help us choose an option we defined the following requirements as guiding principles:

1. DNSSEC validation of all AFRINIC zones should be maintained all the time
2. There should be minimum manual editing of signed zones
3. Migration should be done as quickly as possible
4. Interaction with parent zone should be kept to a mininum
5. Key sizes and algorithms will remain the same on both signers (no change in policy)

Option 1: Keep the same keys

- Export the keys (both ZSK and KSK) on OLD signer to make them available on NEW Signer. Keys can be exported using PKCS#11.
- Then the keys are imported on the NEW signer by specifying the proper key state (active or ready)

Option 2: Key rollover with a double KSK

Consist of having a Double-KSK:

1. On OLD signer, introduce the NEW signer KSK and ZSK public key into the unsigned zone as DNSKEY RR
2. Wait for new KSK and ZSK public keys to get propagated
3. Add new DS record on parent zone
4. On NEW signer, re-sign all the zones
5. On NEW signer add old KSK and ZSK public keys "manually" on the newly unsigned zone as DNSKEY RR
6. On NEW signer, re-sign all the zones
7. Verify serial number, NEW > OLD, before proceeding
8. Do the system rollover, NEW signer is now publishing, OLD signer is stopped
9. Old KSK and ZSK public keys should be maintained until ready to be removed (based on TTL)
10. When signatures have expired and old KSK/ZSK not needed, old DS can be removed
11. Old DNSKEY RRs from OLD Signer can be removed after sufficient time
12. Re-sign

Option 3: New ZSK and KSK

A third solution is to start fresh. Remove any DS records from the parent zone. Stop signing your zone when the DS records are removed from the DNS caches. It is safe to remove the public keys from your zone when the signatures are not present in any DNS caches. Then transfer the zone over to OpenDNSSEC. And let OpenDNSSEC start signing it again.

Option 4: Existing keys followed by Key rollover

This will involve exporting the existing ZSK and KSK to the NEW signer HSM. Sign the zones and immediately proceed with a Key Roll over.

Risk analysis

Criteria (in order of priority):

- Invalid window: will this migration option cause an invalidity period?
- Key manipulation: shall we manipulate the existing private keys e.g. export and transfer?
- Shortest rollover time: what is the ideal rollover time?
- Interactions with parent: number of times we need to interact with the parent zone?
- DNSSEC RRset size: how big is the DNSKEY RRset?

Criteria	**Option 1** **Export existing keys**	**Option 2** **Key rollover with double KSK**	**Option 3** **New Keys**	**Option 4** **Existing keys followed by rollover**
Invalidity window	NO	NO	YES	NO
Key manipulation	YES	NO	NO	YES
Rollover time	None	Wait for old signatures to expire	Wait for caches to pick up new keys.	-
Number of interactions with parents	0	2	1	-
DNSKEY RRset size	Same	Double	Same	Same
Exposure of private keys	YES	NO: only public keys exposed	NO	YES

Preferred option

Option 2 has been selected for the following reasons:

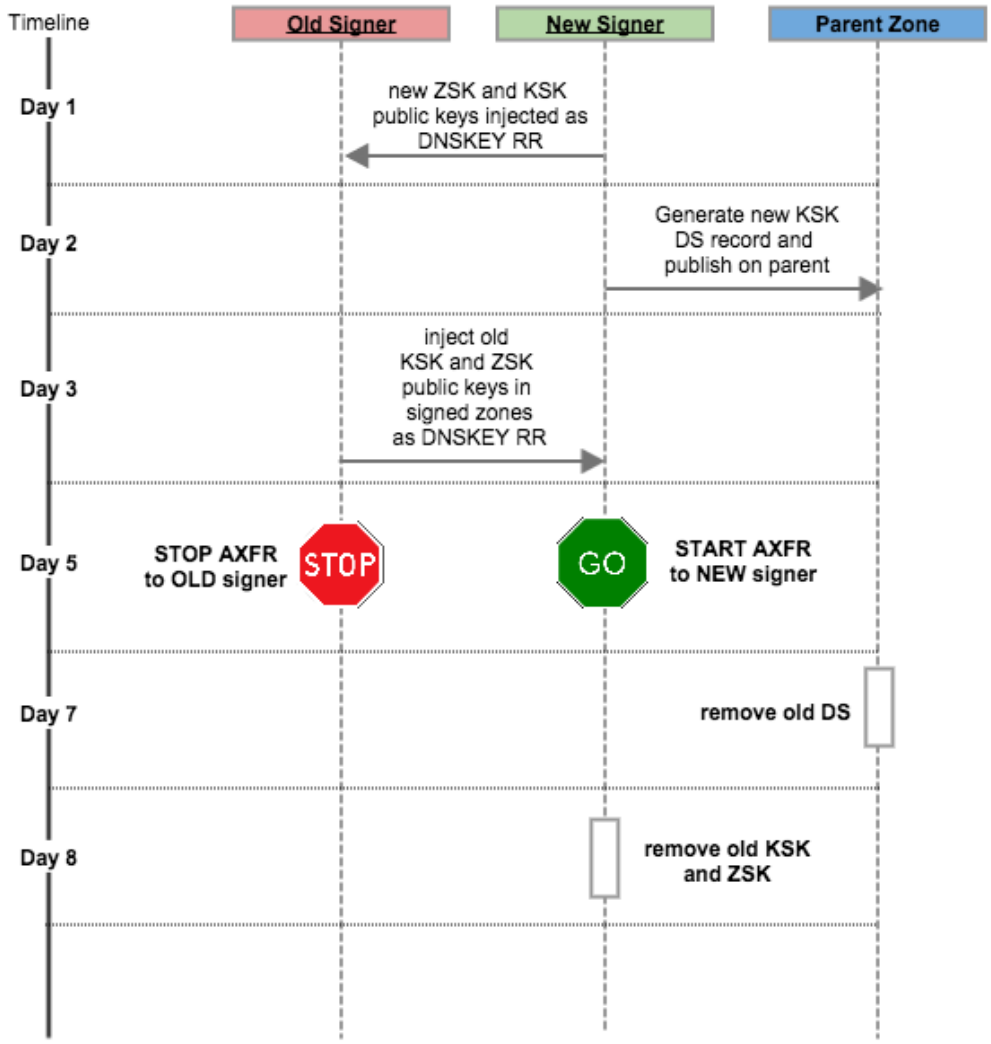
- Zones will remain secure all the time
- No manipulation of private keys
- Private keys will not get exposed

Execution of migration option no. 2 (Key rollover with double KSK)

Important Assumptions
Before migrating the signer we have to make sure of the following:

- **There is no ZSK/KSK rollover in progress in the OLD signer to prevent situation of having multiple DNSKEY RR**
- **The validity of the signatures is much longer that the TTL of the zone (2 or 3 times bigger)**
- **OLD and NEW signers are not name authoritative DNS servers but are hidden primaries.**
- **Both the OLD and NEW signers are being provisioned the same way (with the same zones)**
- **The parent zone in-addr.arpa and ip6.arpa accepts Double-DS records for key rollover procedures.**

We shall take as example a zone that AFRINIC manages "2.4.1.0.0.2.ip6.arpa". Below is the suggested timeline of the migration steps.



Step 1: Export ZSK and KSK public keys from NEW signer to OLD signer

Export the public key ZSK and KSK of the zone by running the following commands. Both ZSK and KSK are not being used for signing yet, they must be in the "ready" state.

```
ods-ksmutil key export --zone 2.4.1.0.0.2.ip6.arpa --keytype ZSK
ods-ksmutil key export --zone 2.4.1.0.0.2.ip6.arpa --keytype KSK
```

- The output of the commands above is in DNSKEY bind format. Note down the outputs for use in the next step.
- **IMPORTANT**. Note down the following zone characteristics on the NEW signer.

  KSK id
  ZSK id
  DNSKEY

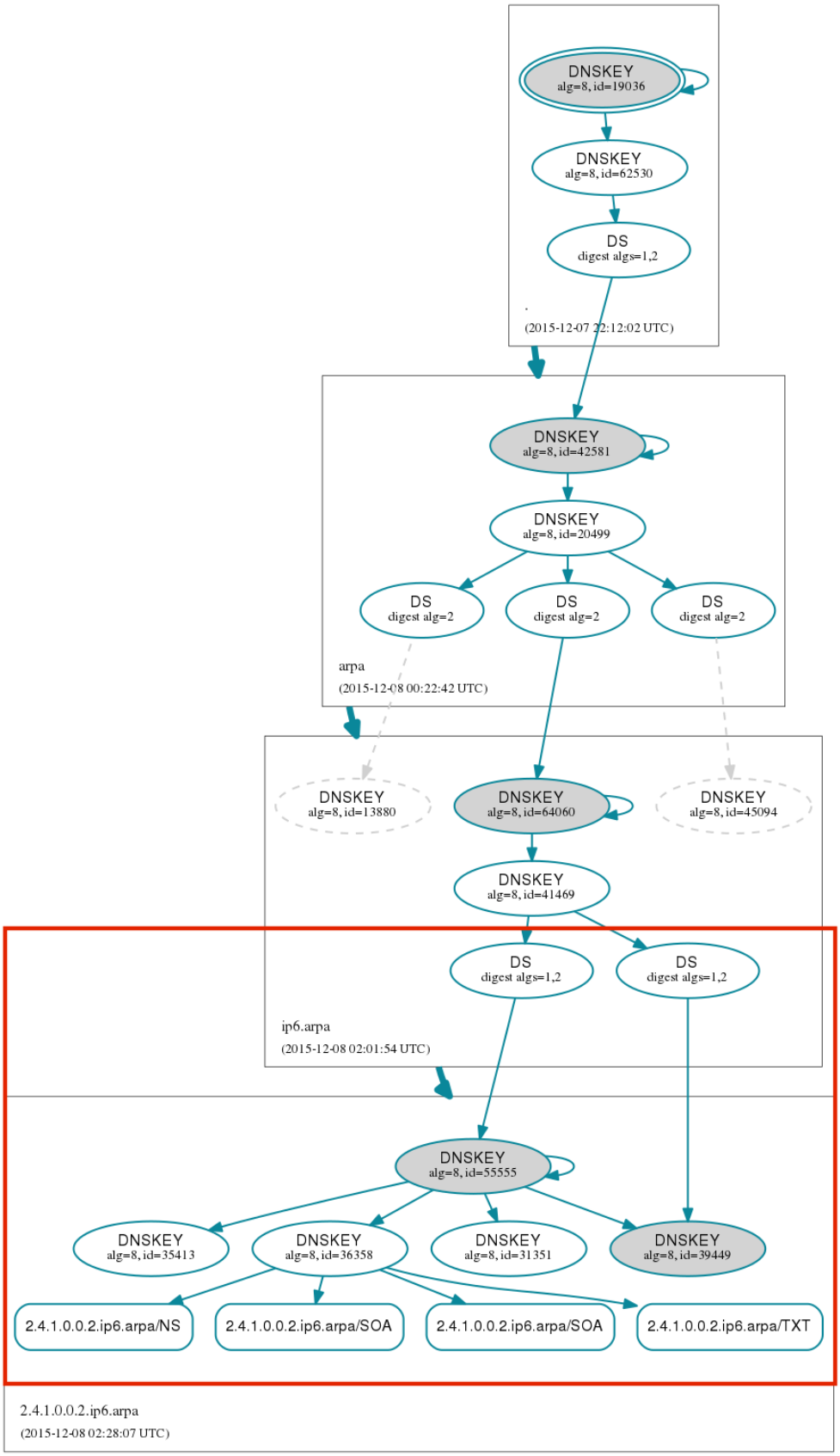
Sub-steps involved:

- On **OLD** Signer, go to the unsigned zones directory which is typically /var/opendnssec/unsigned
- Edit the zones and manually add the exported KSK and ZSK public keys that were exported from the NEW Signer.
- Allow the signer to run after which ensure that the output zone includes a valid signature over the entire DNSKEY set, which should include the active KSK and ZSK from both the **OLD** and the **NEW** signer.

Step 2: Send NEW signer KSK DS records to parent zone

Generate a DS record for each KSK on the NEW signer and send the DS records to the parent zone. Make sure the DS records appear in your dig results before proceeding. You should now have a double DS in your parent zone and two KSKs in your child zone. One KSK is currently being used for active signing.

After upload the double DS record on the parent's zone, your DNS tree should look like this:



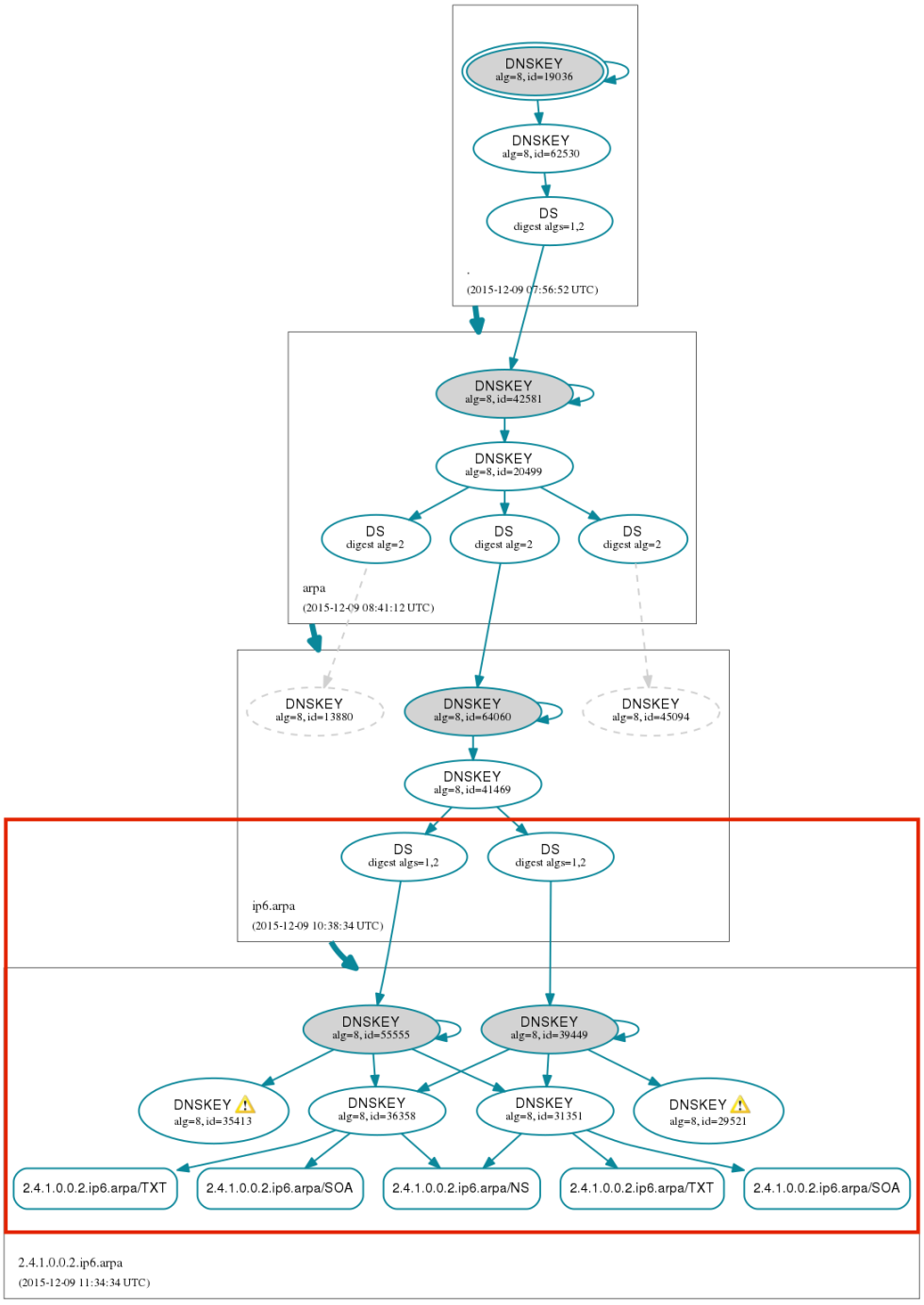
Step 2: Export ZSK and KSK public keys from OLD signer to NEW signer

The purpose of this step is to create a fully cross-signed zone. On the **OLD Signer** under /var/opendnssec/unsigned, take the following sub-steps:

- Manually edit the input unsigned zones on the NEW signer and add the KSK and ZSK public keys from the OLD signer.
- Allow the signer to run on the NEW signer after which ensure that the output zone includes a valid signature over the entire DNSKEY set, which should include the active KSK and ZSK from both the **OLD** and the **NEW** signer.
- The zones will now be signed using both the active KSK and ZSK of the NEW signer, but the OLD KSK will be used to validate signatures done with the OLD KSK.
- This is what will be published to the authoritative DNS servers.
    - **IMPORTANT**. Note down the following zone characteristics of both zones.
      KSK id
      ZSK id

Step 4: Switch signers

Switch over from OLD signer to NEW signer. It means, changing AXFR+TSIG configuration so that the master DNS server stop updating the OLD signer and now updates the NEW signer. After the switch over, your zones should be cross-signed by both the OLD and NEW KSKs/ZSKs, allowing new signatures to be validated but old ones as well.



Step 5: Remove OLD signer DS from parent zone

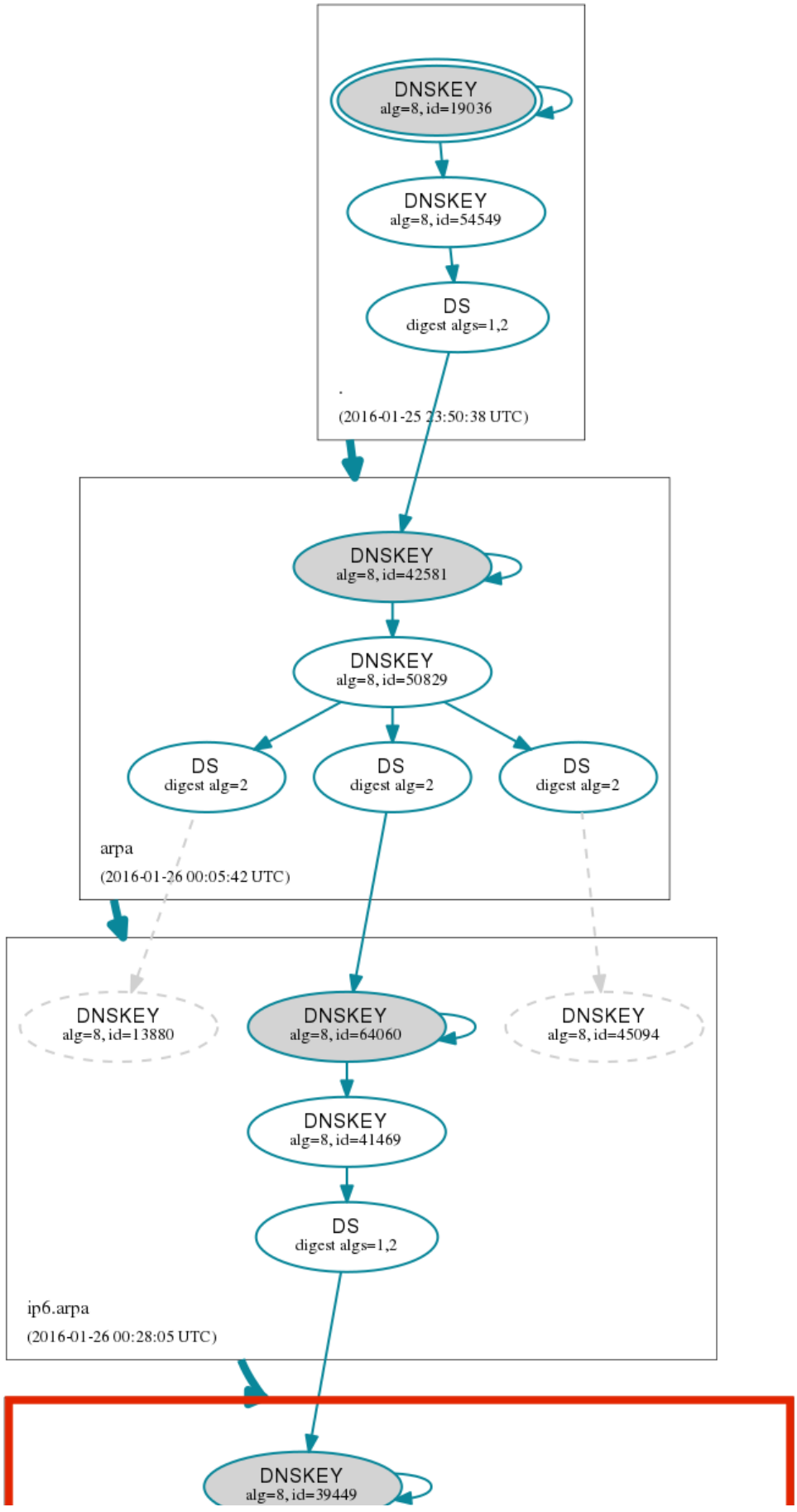
Remove the OLD signer DS from the parent and verify that trust chain still works.

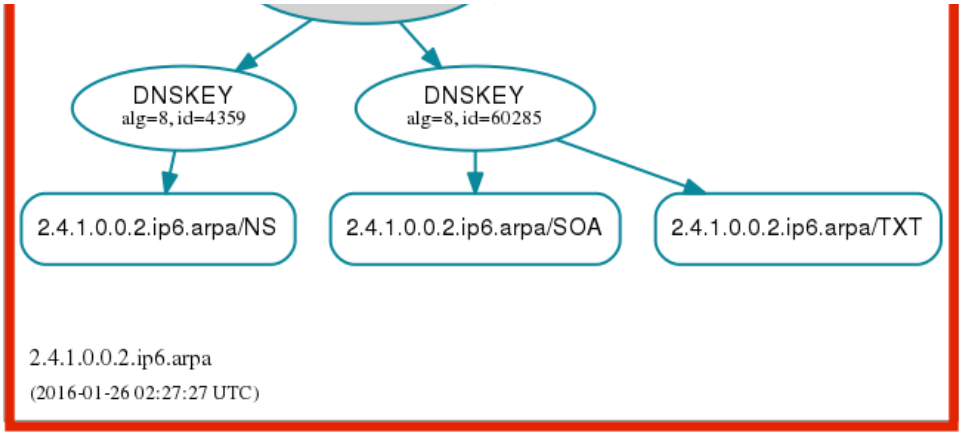
- Wait maxTTL(zone) for all the signatures created with the active ZSK from the OLD signer to disappear from caches and be replaced by signatures created with the active ZSK from the NEW signer

Step 6 - Remove inserted OLD signer KSK and ZSK public keys on NEW signer

Login to the NEW signer and edit the zones by removing the previously inserted OLD signer ZSK and KSK public keys. Dig for results which should only reveal NEW signer DNSKEYs.

After removing the OLD signer KSK and ZSK from on the new signer and the old DS record from the parent zone, your DNS tree should appear like the one below, just as before the migration (one DS and one KSK in your zone).


DNSKEY
alg=8, id=19036
DNSKEY
alg=8, id=54549
DS
digest algs=1,2
.
(2016-01-25 23:50:38 UTC)
DNSKEY
alg=8, id=42581
DNSKEY
alg=8, id=50829
DS
digest alg=2
DS
digest alg=2
DS
digest alg=2
arpa
(2016-01-26 00:05:42 UTC)
DNSKEY
alg=8, id=13880
DNSKEY
alg=8, id=64060
DNSKEY
alg=8, id=45094
DNSKEY
alg=8, id=41469
DS
digest algs=1,2
ip6.arpa
(2016-01-26 00:28:05 UTC)
DNSKEY
alg=8, id=39449



Reducing the TTL on the keys

It is important to reduce the TTL on the KSK and ZSK and to a convenient time span (for example 3600s) during this exercise to speed up the migration process.

Conclusion

OpenDNSSEC migration from one version to another, and from one virtual machine to another is a non-trivial task. Care must be taken so that the keys, and their associated states do not become desynchronised. We have shown that it goes much further than simply copying a set of files from one server to another. We also paid particular attention to the TTL and the sequence of action to follow. At every step, we validated by testing the result. If we had encountered any issue, we were ready to rollback.

References

1. Surfnet.nl - https://dnssec.surfnet.nl/wp-content/uploads/2012/08/DNSSEC-signer-migration-v1.0.pdf
2. Nockso.se - https://nocko.se/2012/03/21/dnssec-quickly-and-correctly/