ZSK versus KSK

Zone Signing Key used to sign entire zone, except DNSKEY RRset.

Key Signing Key used to sign DNSKEY set.

Why are these separate?

- ✘ Have higher security through larger key (breaking KSK will have more severe impact)
- ✘ Limit size of ZSK, so most answers with ZSK based signatures are still small
- ✘ Allow separate rolling of ZSK and KSK, as KSK
- ✘ Keep KSK in more secure and backed up location
    - ○ Switching to new ZSK can be done fast enough such that losing ZSK will not break your zone.

ZSK versus KSK

It is possible to one key as both ZSK and KSK:

A Combined Signing Key: CSK

Signing and validation procedures with CSKs are not affected.

There is no structural difference between ZSK and KSK (only a flag that is a hint).

But changing the key will involve the complexity of KSK and ZSK combined.

Recursing the key lookup problem

The DNS query for the DNSKEY did provide us with an answer that was signed.
…. So it was not forged.
…. And the ZSK can only be the ZSK used to generate
Signatures from encountered earlier in queries

But this does not mean the KSK isn't someone's else KSK.

We need to verify that the KSK is the proper KSK.

Use the same DNS mechanism that directed us to the zone to verify fingerprint.

➥The parent zone indicated which nameservers to use

➥Let the parent zone also provide the fingerprint of the KSK.

DS record in parent zone

The parent zone will contain:

```
example.com.       86400    IN    NS  ns.example.org.
example.com.       86400    IN    DS  43547 8 2
615A64233543F66F44D68933625B17497C89A70E858ED76A2145997E DF96A918
```

The DS record is a Delegation of Signing

DS record in parent zone

The parent zone will contain:

```
example.com.       86400    IN    NS  ns.example.org.
example.com.       86400    IN    DS   43547 8 2
615A64233543F66F44D68933625B17497C89A70E858ED76A2145997E DF96A918
```

The keytag as hint and human inspection.

This is something that is often monitored.

DS record in parent zone

The parent zone will contain:

```
example.com.       86400    IN    NS  ns.example.org.
example.com.       86400    IN    DS   43547 8 2
615A64233543F66F44D68933625B17497C89A70E858ED76A2145997E DF96A918
```

The algorithm 8 as before, must match the algorithm in the DNSKEY for the key.

And the way the fingerprint (hash) was computed:

1. SHA-1
2. SHA-256

DS record in parent zone

The parent zone will contain:

```
example.com.      86400    IN    NS  ns.example.org.
example.com.      86400    IN    DS   43547 8 2
615A64233543F66F44D68933625B17497C89A70E858ED76A2145997E DF96A918
```

And finally the fingerprint itself.

Now as hexadecimal for more convenience, signatures are normally never handled manually, but DS records need to be placed into the parent, often manually.

And spaces may be included for legibility as in base64 encoded signatures.

Chain of Trust

- Result for “dig A dnssec.dnslab.uk @dnslab.uk”
- Signature over result “dnssec.dnslab.uk” using ZSK of dnslab.uk
  - Query “dig DNSKEY dnslab.uk @dnslab.uk”
  - Signature over result using KSK of dnslab.uk
    - Query “dig DS dnslab.uk @uk”
    - Signature over result using ZSK of uk.
      - Query “dig DNSKEY @uk”
      - Signature over result using KSK of uk
        - Query “dig DS @.”
        - Signature over result using ZSK of root zone
          - Query “dig DNSKEY .”
          - Signature over result using root zone KSK

Root

Primary or root key is the trust anchor.

It cannot be discovered this way.

A leap of faith is needed to initially get a root key

Manual retrieved using out of band communication.

Distributed along with recursor or distribution software.

Updated the same way or with future RFC5011 rollover (only for root key).

Changing the root key was a big deal that happened not to long before the disease that shall not be named. It was a success and planned to happen more often.

Disadvantage key in parent zone

Key is pretty large, in already large zone.

- ✘ It is not owned material by parent zone
- ✘ Like NS records are hints, keys should not be there
- ✘ Interaction with parent zone is hard, and changing keys is multi step
- ✘ Use something more shorter

Store fingerprint (= hash) of key in parent zone and key itself in child zone.

Multiple keys

You can have multiple keys signing your zone.

This can give you multiple signatures per RR set.

Or have different RR sets be signed with different keys.

In principle one valid signature path will suffice as long as the KSK and ZSK use same algorithm to prevent downgrade attacks.

Protect your keys paramount.

Solution: Sign keys with better key than other data

Born: Key Signing Key (KSK) and Zone Signing Key (ZSK) twins

Keys, sizes, algorithms

Larger keys:

- ➢ more secure
- ➢ more overhead to sign
- ➢ more compute time on validation
- ➢ more data to transport

New algorithms

- ➢ some for smaller signatures
- ➢ some for more security
- ➢ more compute time overhead

Negative answers

Results from a DNS server

Apart from an answer servers may instead return:

- ✘ SERVFAIL

  internal server error; try another equivalent server
- ✘ NXDOMAIN:

  there is no such domain/label the server
- ✘ NODATA

  there is data for this domain/label, just not for the question.

On NODATA or NXDOMAIN the server will not respond with an answer to the query.

Typical Lookup

Command "dig +dnssec no.dnslab.uk":

```
;; ANSWER SECTION:

;; AUTHORITY SECTION:
dnslab.uk.  60 IN  SOA ns.dnslab.uk. hostmaster.dnslab.uk. 1 28800 14400
604800 60
dnslab.uk.  60 IN  RRSIG    SOA 13 2 300 20211017205753 20210919205753 16480
dnslab.uk. uEVSiHMGOTetJHBdmhZ6/Bdit9rf391+qJAcNpaWNgjZJW2ZrbuKy0H2
rMMBM7U4qIr1GgZ/M1a/daoVWDKyTg==
```

This proves the answer was composed by the domain owner,

but NOT that it was the answer to the question.

There is no answer to sign that contains the question. And it it would, we might be forced to provide answers to any possible question.

Typical Lookup

Command “dig +dnssec no.dnslab.uk”:

```
;; ANSWER SECTION:

;; AUTHORITY SECTION:
dnslab.uk.  60 SOA   ns.dnslab.uk. hostmaster.dnslab.uk. 1 28800 14400 604800
60
dnslab.uk.  60 RRSIG SOA 13 2 300 20211017205753 20210919205753 1648 0
                                  dnslab.uk. uEVSiHMGOTetJHBdmhZ6/Bdit9rf391+
                                  qJAcNpaWNgjZJW2ZrbuKy0H2rMMBM7U4qIr1GgZ/M1a
                                 /daoVWDKyTg==
```

This answer does not prove the non-existence of no.dnslab.uk, it could as well be returned for yes.dnslab.uk.

Typical Lookup

Command “dig +dnssec no.dnslab.uk”:

```
;; ANSWER SECTION:

;; AUTHORITY SECTION:
dnslab.uk.  60 SOA   ns.dnslab.uk. hostmaster.dnslab.uk. 1 28800 14400 604800
60
dnslab.uk.  60 RRSIG SOA 13 2 300 20211017205753 20210919205753 16480
                                  dnslab.uk. uEVSiHMGOTetJHBdmhZ6/Bdit9rf391+
                                  qJAcNpaWNgjZJW2ZrbuKy0H2rMMBM7U4qIr1GgZ/M1a
                                  /daoVWDKyTg==
dnslab.uk.  60 NSEC  ns.dnslab.uk. A NS SOA AAAA RRSIG NSEC DNSKEY
dnslab.uk.  60 RRSIG NSEC 13 2 60 20211017205753 20210919205753 16480
                                  dnslab.uk. T6KmbKLEkML8YWixN1HhxJ5eWwYxkIrU
                                  Vc3OOkttng2c6E2ScH5XETzBXd+ysA79ThAVz1gdMay
                                  C5Lg2fl3kIQ==
```

NSEC records

```
dnslab.uk.  60 NSEC  ns.dnslab.uk. A NS SOA AAAA RRSIG NSEC DNSKEY
dnslab.uk.  60 RRSIG NSEC 13 2 60 20211017205753 20210919205753 16480
                                  dnslab.uk. T6KmbKLEkML8YWixN1HhxJ5eWwYxkIrU
                                  Vc3OOkttng2c6E2ScH5XETzBXd+ysA79ThAVz1gdMay
                                  C5Lg2fl3kIQ==
```

The NSEC record itself is signed just like any other record, so its authenticity and integrity can be validated.

And the NSEC record provides information about the previous and next name that are available.

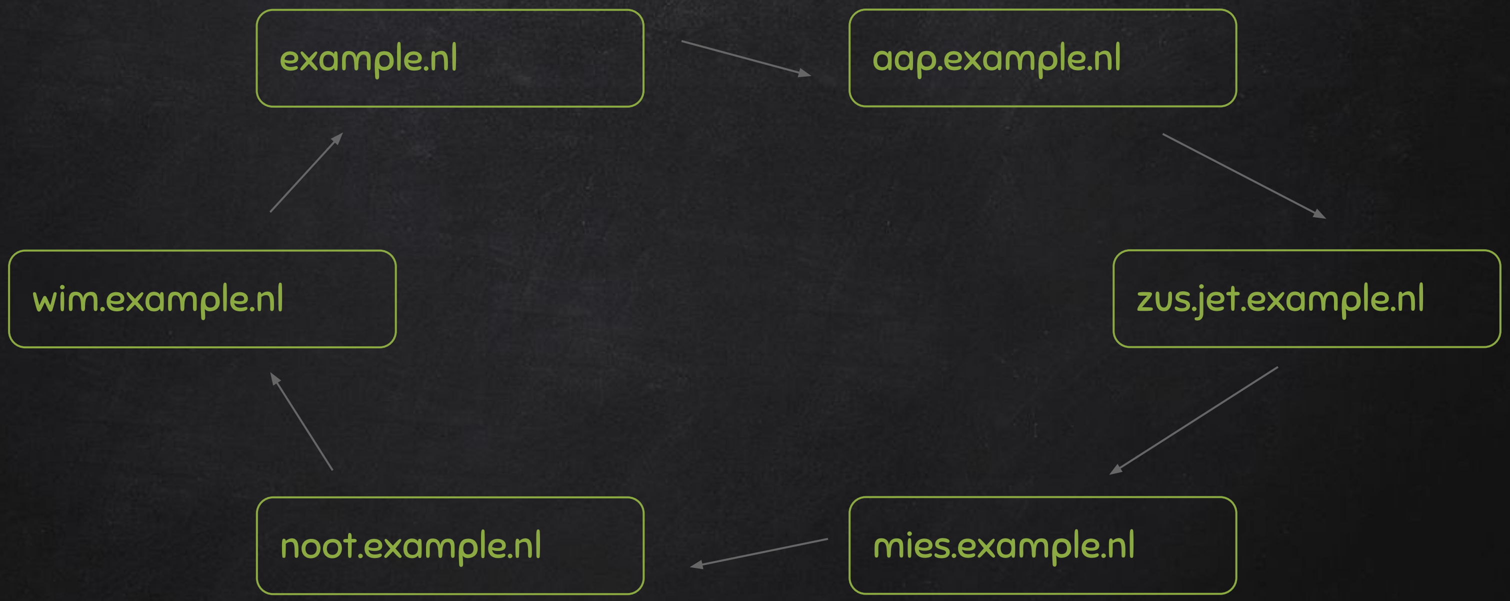
NSEC denial chain

```
example.nl.         60  SOA  ns.dnslab.nl. root.dnslab.uk. (1 8h 20m 1w 20m)
aap.example.nl.     60  A    10.0.0.1
noot.example.nl.    60  A    10.0.0.2
mies.example.nl.    60  A    10.0.0.3
wim.example.nl.     60  A    10.0.0.4
zus.jet.example.nl. 60  A    10.0.0.5
```

Take the labels and lexicographical order them:

```
nl.example
nl.example.aap
nl.example.jet.zus
nl.example.mies
nl.example.noot
nl.example.wim
```

NSEC denial chain



example.nl

aap.example.nl

zus.jet.example.nl

mies.example.nl

noot.example.nl

wim.example.nl

NSEC records

```
dnslab.uk.  60 NSEC  ns.dnslab.uk. A NS SOA AAAA RRSIG NSEC DNSKEY
dnslab.uk.  60 RRSIG NSEC 13 2 60 20211017205753 20210919205753 16480
                                  dnslab.uk. T6KmbKLEkML8YWixN1HhxJ5eWwYxkIrU
                                  Vc3OOkttng2c6E2ScH5XETzBXd+ysA79ThAVz1gdMay
                                  C5Lg2fl3kIQ==
```

There is no label between dnslab.uk and ns.dnslab.uk.

NSEC records

```
dnslab.uk.  60 NSEC  ns.dnslab.uk. A NS SOA AAAA RRSIG NSEC DNSKEY
dnslab.uk.  60 RRSIG NSEC 13 2 60 20211017205753 20210919205753 16480
                                  dnslab.uk. T6KmbKLEkML8YWixN1HhxJ5eWwYxkIrU
                                  Vc3OOkttng2c6E2ScH5XETzBXd+ysA79ThAVz1gdMay
                                  C5Lg2fl3kIQ==
```

There is no label between dnslab.uk and ns.dnslab.uk.

NSEC records

```
dnslab.uk.  60 NSEC  ns.dnslab.uk.  A NS SOA AAAA RRSIG NSEC DNSKEY
dnslab.uk.  60 RRSIG NSEC 13 2 60 20211017205753 20210919205753 16480
                                  dnslab.uk. T6KmbKLEkML8YWixN1HhxJ5eWwYxkIrU
                                  Vc3OOkttng2c6E2ScH5XETzBXd+ysA79ThAVz1gdMay
                                  C5Lg2fl3kIQ==
```

There is no label between dnslab.uk and ns.dnslab.uk.

dnslab.uk has information for DNS RR-types A, NS, SOA, AAA, RRSIG, NSEC and DNSKEY

NSEC records

```
dnslab.uk.  60 NSEC  ns.dnslab.uk. A NS SOA AAAA RRSIG NSEC DNSKEY
dnslab.uk.  60 RRSIG NSEC 13 2 60 20211017205753 20210919205753 16480
                                  dnslab.uk. T6KmbKLEkML8YWixN1HhxJ5eWwYxkIrU
                                  Vc3OOkttng2c6E2ScH5XETzBXd+ysA79ThAVz1gdMay
                                  C5Lg2fl3kIQ==
```

There is no label between dnslab.uk and ns.dnslab.uk.

This solves the NXDOMAIN

dnslab.uk has information for DNS RR-types A, NS, SOA, AAA, RRSIG, NSEC and DNSKEY

This solves the NODATA

Wildcard

```
example.com    SOA ns.example.com root.example.com 1 60 60 60 60
a.example.com. TXT “First”
c.example.com. TXT “Third”
*.example.com. TXT “Wildcard”
```

A query for “b.example.com” should match the wildcard.

In DNS the expansion should already be in the response.

```
b.example.com.  TXT    “Wildcard”
```

There is however no RRSIG available for the name “b.example.com.”

Wildcard

```
example.com    SOA ns.example.com root.example.com 1 60 60 60 60
a.example.com. TXT “First”
c.example.com. TXT “Third”
*.example.com. TXT “Wildcard”
```

A query for “b.example.com” should match the wildcard.

In DNS the expansion should already be in the response.

```
b.example.com.  TXT    “Wildcard”
b.example.com.  RRSIG  TXT 13  2 2021…..
```

Instead return RRSIG for example.com, which a label count of 2 so the resolver knows this was a wildcard expansion.

Wildcard

```
example.com    SOA ns.example.com root.example.com 1 60 60 60 60
a.example.com. TXT “First”
c.example.com. TXT “Third”
*.example.com. TXT “Wildcard”
```

A query for “b.example.com” should match the wildcard.

In DNS the expansion should already be in the response.

```
b.example.com.  TXT    “Wildcard”
b.example.com.  RRSIG  TXT 13  2 2021…..
a.example.com.  NSEC   c TXT
```

And we need prove that there was no record b.example.com
By getting informed that there was nothing between a.example.com and c.example.com

Wildcard

```
example.com    SOA ns.example.com root.example.com 1 60 60 60 60
a.example.com. TXT “First”
c.example.com. TXT “Third”
```

If there is no wildcard, we need two NSEC records:

1. Proving the label does not exist
2. Proving the wildcard does not exist

Sometimes these overlap, in case only one NSEC record is returned.

Hands-on

Deploying signed zone in chain of trust

http://dnslab.uk/