Deploying DNSSEC

Part II DNSSEC Mechanisms and deployment

AFRINIC-17

aalain@afrinic.net

Public Key Crypto (in one slide)

- Key pair: a secret (or private) key and a public key Simplified:
  - If you know the public key, you can decrypt data encrypted with the secret key
    - Usually an encrypted hash value over a published piece of information; the owner is the only person who can construct the secret. Hence this a signature
  - If you know the secret key, you can decrypt data encrypted with the public key
    - Usually an encrypted key for symmetric cipher
- PGP uses both, DNSSEC only uses signatures

DNSSEC Mechanisms

- New Resource Records
- Setting Up a Secure Zone
- Delegating Signing Authority
- DNSSEC Deployment Rollovers

New Resource Records

RRs and RRSets

- Resource Record:
  - name TTL class type rdata

```
www.nlnetlabs.nl.    7200     IN   A    192.168.10.3
```

- RRset: RRs with same name, class and type:

```
www.nlnetlabs.nl.  7200    IN   A  192.168.10.3
                          A 10.0.0.3
                          A    172.25.215.2
```

- RRSets are signed, not the individual RRs

New Resource Records

- Three Public key crypto related RRs
  - RRSIG Signature over RRset made using private key
  - DNSKEY Public key, needed for verifying a RRSIG
  - DS Delegation Signer; 'Pointer' for building chains of authentication
- One RR for internal consistency
  - NSEC Indicates which name is the next one in the zone and which typecodes are available for the current name
    - authenticated non-existence of data

DNSKEY RDATA

- 16 bits: FLAGS
- 8 bits: protocol
- 8 bits: algorithm
- N*32 bits: public key

Example:

nlnetlabs.nl. 3600 IN DNSKEY 256 3 5 (
AQOvhvXXU61Pr8sCwELcqqq1g4JJ
CALG4C9EtraBKVd +vGIF/unwigfLOA
O3nHp/cgGrG6gJYe8OWKYNgq3kDChN)

RRSIG RDATA

- 16 bits - type covered
- 8 bits - algorithm
- 8 bits - nr. labels covered
- 32 bits - original TTL

```
nlnetlabs.nl.  3600 IN      RRSIG   A 5 2 3600 (
        20050611144523 20050511144523  3112 nlnetlabs.nl.
        VJ+8ijXvbrTLeoAiEk/qMrdudRnYZM1VlqhN
        vhYuAcYKe2X/jqYfMfjfSUrmhPo+0/GOZjW
        66DJubZPmNSYXw== )
```

signature field

- 32 bit - signature expiration
- 32 bit - signature inception
- 16 bit - key tag
- signer’s name

Delegation Signer (DS)

- Delegation Signer (DS) RR indicates that:
  - delegated zone is digitally signed
  - indicated key is used for the delegated zone
- Parent is authorative for the DS of the child's zone
  - Not for the NS record delegating the child's zone!
  - DS **should not** be in the child's zone

DS RDATA

- 16 bits: key tag
- 8 bits: algorithm
- 8 bits: digest type
- 20 bytes: SHA-1 Digest

```
$ORIGIN nlnetlabs.nl.
lab.nlnetlabs.nl.   3600 IN   NS  ns.lab.nlnetlabs.nl
lab.nlnetlabs.nl.   3600 IN   DS
```

3112 5 1 (

239af98b923c023371b521
g23b92da12f42162b1a9
)

NSEC RDATA

- Points to the next domain name in the zone
  - also lists what are all the existing RRs for "name"
  - NSEC record for last name "wraps around" to first name in zone
- N*32 bit type bit map
- Used for authenticated denial-of-existence of data
  - authenticated non-existence of TYPEs and labels
- Example:

```
www.nlnetlabs.nl. 3600 IN NSEC     nlnetlabs.nl. A RRSIG NSEC
```

NSEC Records

- NSEC RR provides proof of non-existence
- If the servers response is Name Error (NXDOMAIN):
  - One or more NSEC RRs indicate that the name or a wildcard expansion does not exist
- If the servers response is NOERROR:
  - And empty answer section
  - The NSEC proves that the QTYPE did not exist
- More than one NSEC may be required in response
  - Wildcards
- NSEC records are generated by tools
  - Tools also order the zone

NSEC Walk

- NSEC records allow for zone enumeration
- Providing privacy was not a requirement at the time
- Zone enumeration is a problem for some entities

- NSEC3
  - All RR names hashed
  - Hashed names are ordered
  - “opt-out” for unsecured delegations possibilities

Delegating Signing Authority

Chains of Trust

Using the DNS to Distribute Keys

- Secured islands make key distribution problematic
- Distributing keys through DNS:
  - Use one trusted key to establish authenticity of other keys
  - Building chains of trust from the root down
  - Parents need to sign the keys of their children
- Only the root key needed in ideal world
  - Parents always delegate security to child

Key Problem

- Interaction with parent administratively expensive
  - Should only be done when needed
  - Bigger keys are better

- Signing zones should be fast
  - Memory restrictions
  - Space and time concerns
  - Smaller keys with short lifetimes are better

Key Functions

- Large keys are more secure
  - Can be used longer ☺
  - Large signatures => large zonefiles ☹
  - Signing and verifying computationally expensive ☹
- Small keys are fast
  - Small signatures ☺
  - Signing and verifying less expensive ☺
  - Short lifetime ☹

Key solution: More Than One Key

- RRsets are signed, not RRs
- DS points to specific key
  - Signature from that key over DNSKEY RRset transfers trust to all keys in DNSKEY RRset

- Key that DS points to only signs DNSKEY RRset
  - Key Signing Key (KSK)
- Other keys in DNSKEY RRset sign entire zone
  - Zone Signing Key (ZSK)

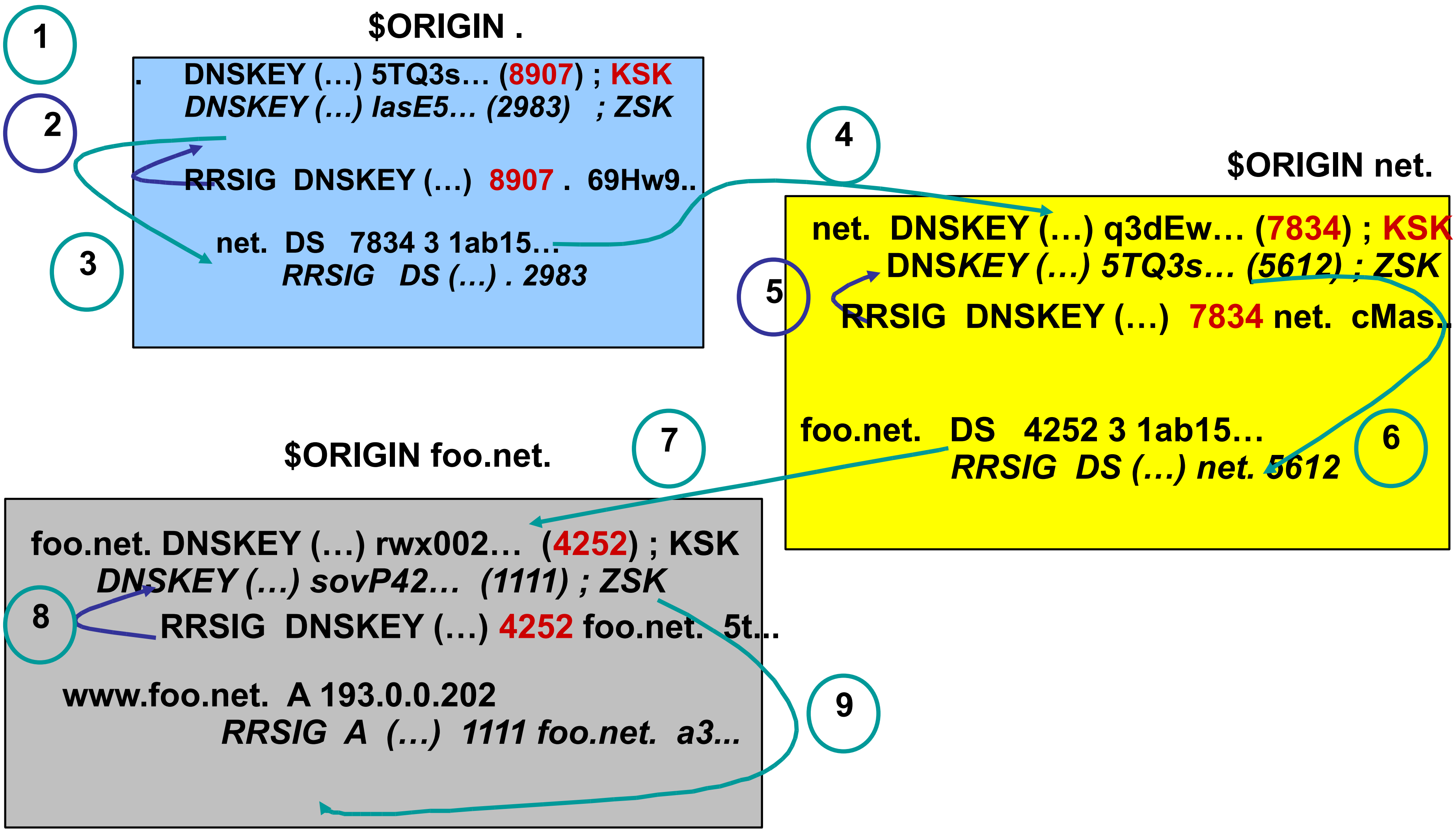
Initial Key Exchange

- Child needs to:
  - Send key signing keyset to parent

- Parent needs to:
  - Check child's zone
    - for DNSKEY & RRSIGs
  - Verify if key can be trusted
  - Generate DS RR

y configured
d key: . 8907

Walking the Chain of Trust



Security Status of Data (RFC4035)

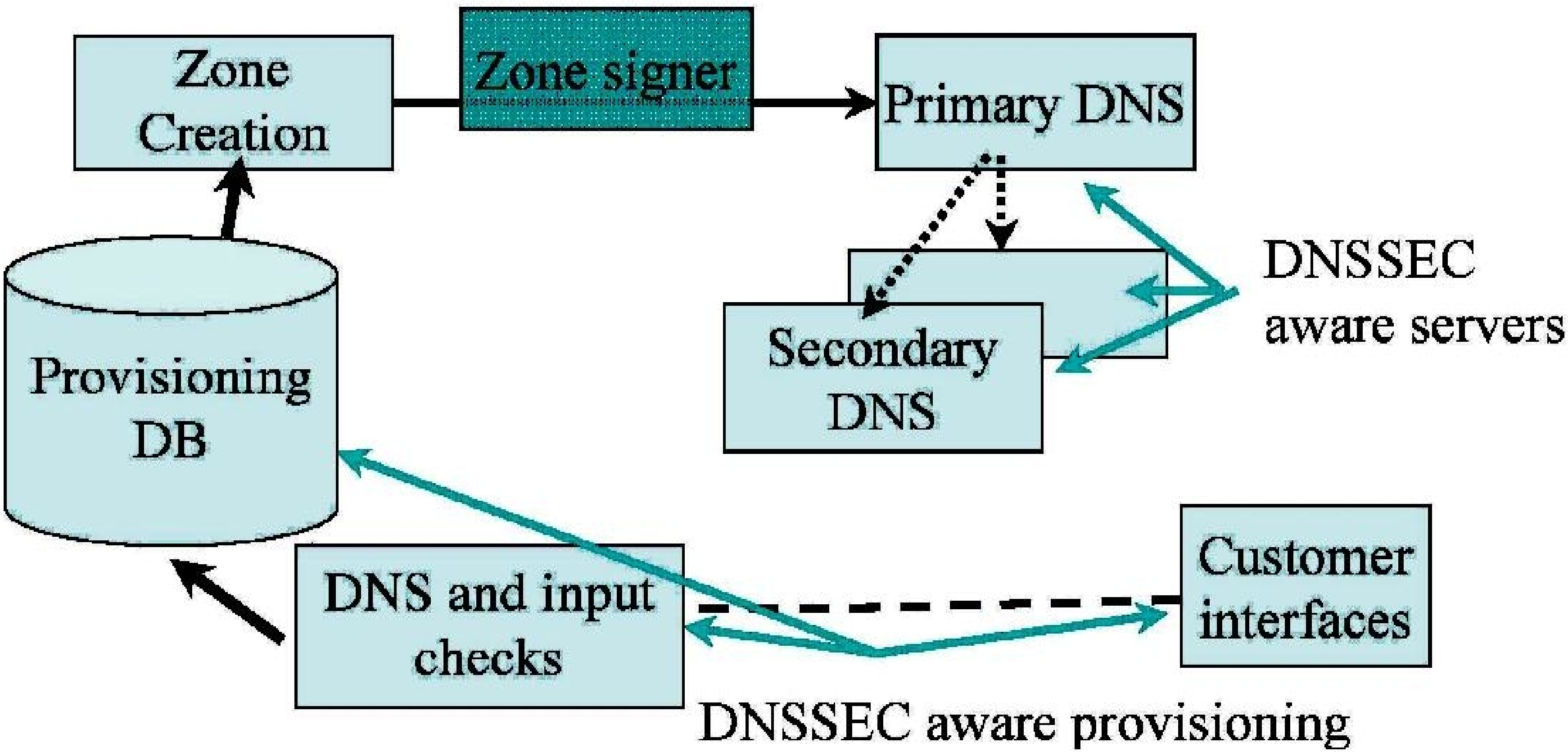
- Secure
  - Resolver is able to build a chain of signed DNSKEY and DS RRs from a trusted security anchor to the RRset
- Insecure
  - Resolver knows that it has no chain of signed DNSKEY and DS RRs from any trusted starting point to the RRset
- Bogus
  - Resolver believes that it ought to be able to establish a chain of trust but for which it is unable to do so
  - May indicate an attack but may also indicate a configuration error or some form of data corruption
- Indeterminate
  - Resolver is not able to determine whether the RRset should be signed

DNSSEC DEPLOYMENT

DNSSEC Deployment Tasks

- Key maintenance policies and tools
  - Private key use and protection
  - Public key distribution
- Zone signing and integration into the provisioning chain
- DNS server infrastructure
- Secure delegation registry changes
  - Interfacing with customers

DNSSEC Architecture modification



Key Maintenance

- DNSSEC is based on public key cryptography
  - Data is signed using a private key
  - It is validated using a public key

Operational problems:

- Dissemination of the public key
- Private key has a '*best before*' date
  - Keys change, and the change has to disseminate

DNSSEC Policy & Practice Statement

- draft-ietf-dnsop-dnssec-dps-framework

```
This document presents a framework to assist writers of DNSSEC Policy
  and Practice Statements such as Domain Managers and Zone Operators on
  both the top-level and secondary level, who is managing and operating
  a DNS zone with Security Extensions (DNSSEC) implemented.

  In particular, the framework provides a comprehensive list of topics
  that should be considered for inclusion into a DNSSEC Policy
  definition and Practice Statement.
```

- ICANN DPS for root zone
  - http://www.root-dnssec.org/wp-content/uploads/2010/06/icann-dps-00.txt

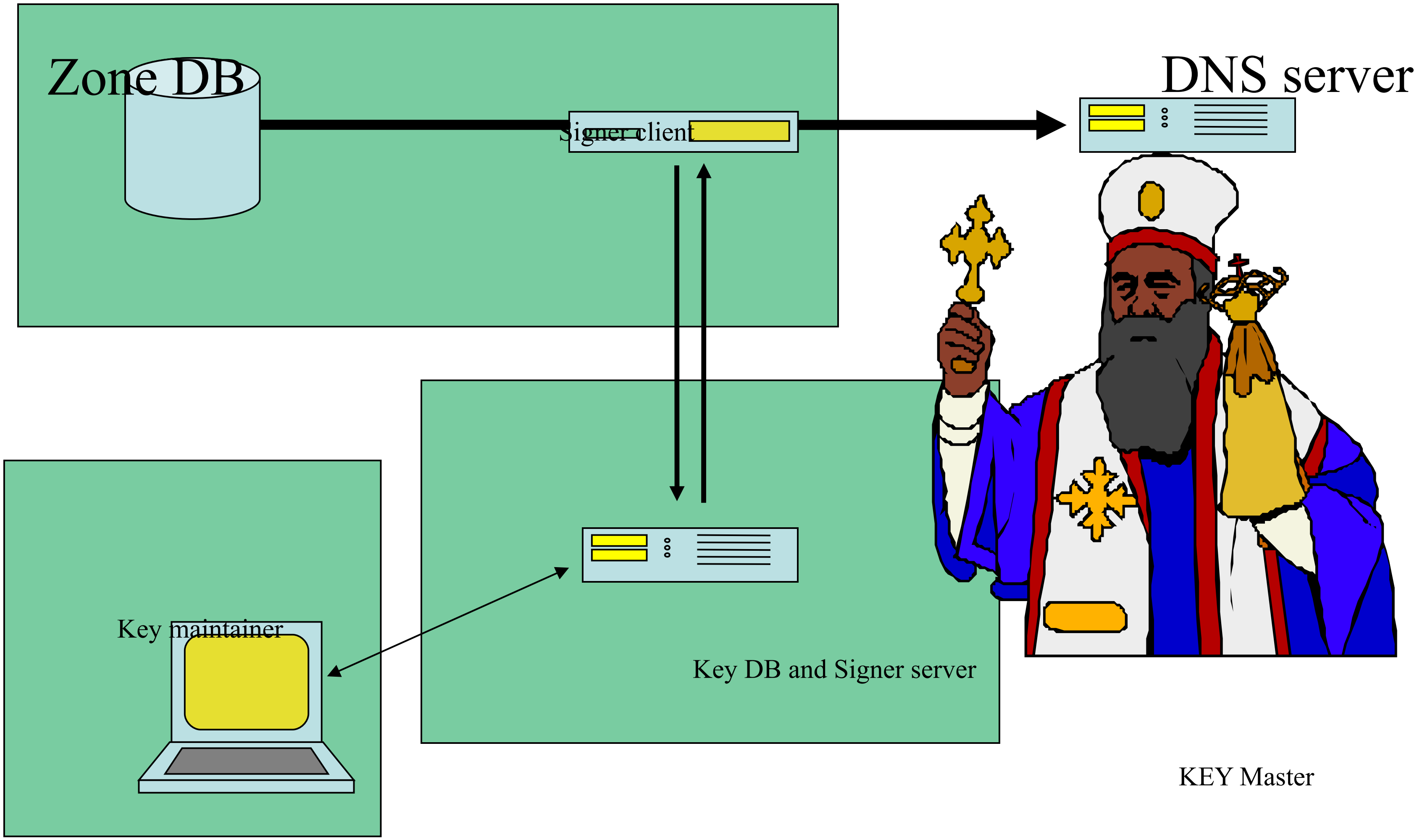
Public Key Dissemination

- In theory only one trust-anchor needed that of the root
  - How does the root key get to the end user?
  - How is it rolled?
- In absence of hierarchy, there will be many trust-anchors
  - How do these get to the end-users?
  - How are these rolled?
- These are open questions,making early deployment difficult.
- DLV registries(https://secure.isc.org/index.pl?/ops/dlv/)

Key Management

- There are many keys to maintain
  - Keys are used on a per zone basis
    - Key Signing Keys and Zone Signing Keys
  - During key rollovers there are multiple keys
    - In order to maintain consistency with cached DNS data
    - RFC4641
- Private keys need shielding

Private Key Maintenance Basic Architecture



Maintaining Keys and Signing Zones

- The KeyDB maintains the private keys
  - It ‘knows’ rollover scenarios
  - UI that can create, delete, roll keys without access to the key material
  - Physically secured
- The signer ties the Key DB to a zone
  - Inserts the appropriate DNSKEYs
  - Signs the the zone with appropriate keys
- Strong authentication

Infrastructure

- One needs primary and secondary servers to be DNSSEC protocol aware
- We have concerns about Firewalls/IDS/IPS on DNS packet size and EDNS0
  - http://www.icann.org/committees/security/sac016.htm
- We had a number of concerns about memory, CPU, network load
  - Research done at RIPE-NCC and published as RIPE 352

Infrastructure

- Bandwidth increase is caused by many factors
  - Hard to predict but fraction of DO bits in the queries is an important factor
- CPU impact is small, Memory impact can be calculated
- Don’t add DNSKEY RR set in additional

Parent-Child Key Exchange

- In the DNS the parent signs the “Delegations Signer” RR
  - A pointer to the next key in the chain of trust

```
$ORIGIN net.

kids NS    ns1.kids
     DS   (…) 1234
     RRSIG DS (…)net.

money NS    ns1.money
      DS    (…)
      RRSIG DS (…)net.
```

```
$ORIGIN kids.net.

@ NS   ns1
 RRSIG NS (…) kids.net.
 DNSKEY (…)  (1234)
 DNSKEY (…)  (3456)
 RRSIG dnskey … 1234 kids.net. …
 RRSIG dnskey … 3456 kids.net. …

beth  A  127.0.10.1
     RRSIG A (…) 3456 kids.net. …
```

- DNSKEY or DS RR needs to be exchanged between parent and child

Underlying Ideas

- The DS exchange is the same process as the NS exchange
  - Same authentication/authorization model
  - Same vulnerabilities
  - More sensitive to mistakes
- Integrate the key exchange into existing interfaces
  - Customers are used to these
- Include checks on configuration errors
  - DNSSEC is picky
- Provide tools
  - To prevent errors and guide customers

Key Rollover

DNSKEY in flavours

- Zone Signin Key (ZSK)
- Key Signing Key (KSK)
  - Functions as secure entry point into the zone
    - Trust-anchor configuration
    - Parental DS points to it
    - Interaction with 3rd party
- DNSKEYs are treated all the same in the protocol
- Operators can make a distinction
  - Look at the flag field: ODD (257 in practice) means SEP

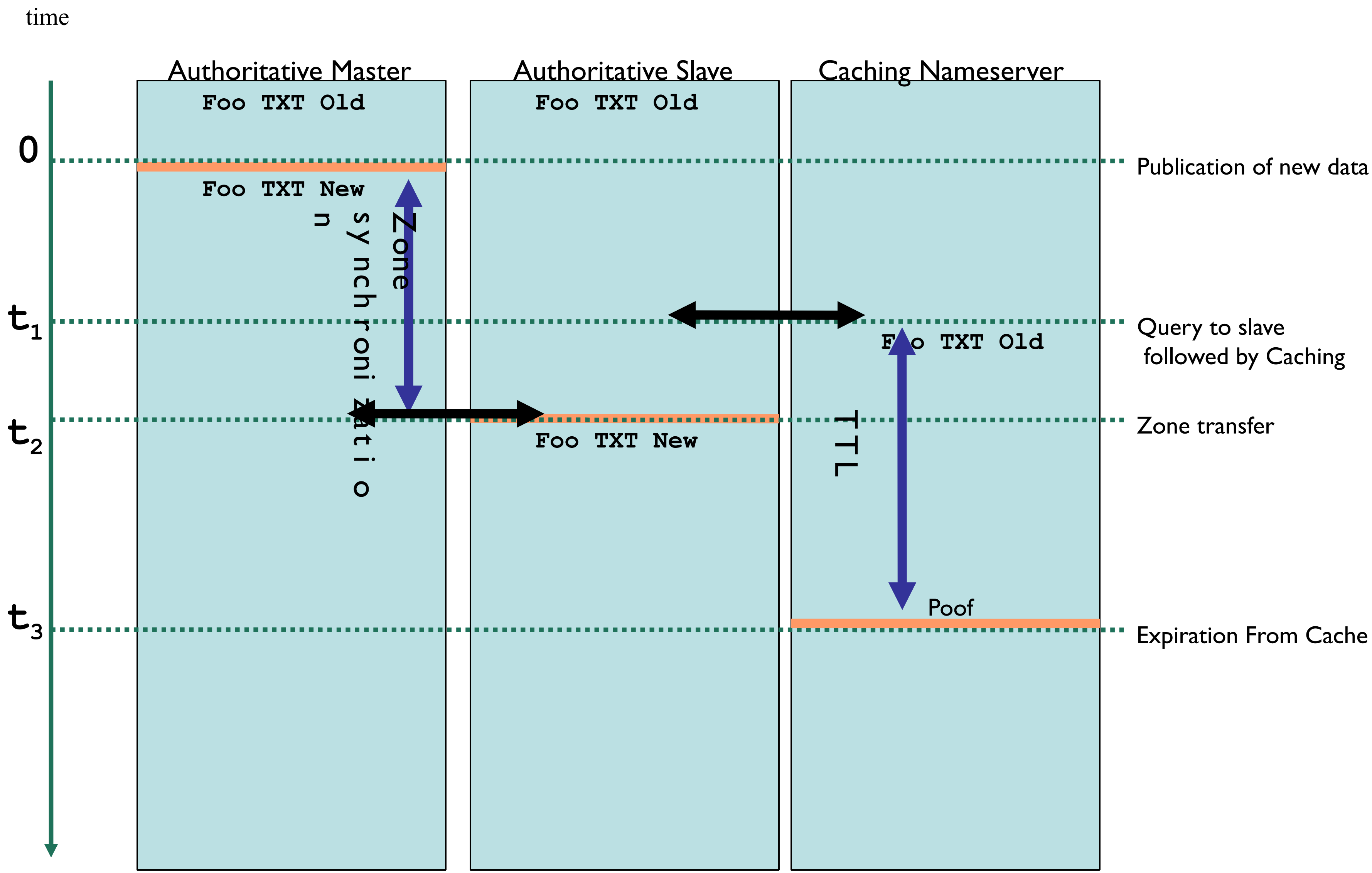
Benefits of using separate keys

- Rolling KSK needs interaction, rolling ZSKs can be done almost instantaneously
- Remember KSK replacement may result in
  - Trust-anchor updates
  - Change of DS record at parent
- Allows different responsibilities
  - ZSKs may be touched day to day by junior staff
  - KSKs may only be touched by senior staff

Rolling keys instantaneously?

- Remember that in the DNS caches are at play.
  - It takes a bit of time to have new information propagate
- When you happen to get new data you would like to be able to use RRSIGs from the cache
- When you happen to get old data from the cache you would like to use new RRSIGs
- Try to make sure both old and new keys are available
- Or, try to make sure both old and new sigs are available

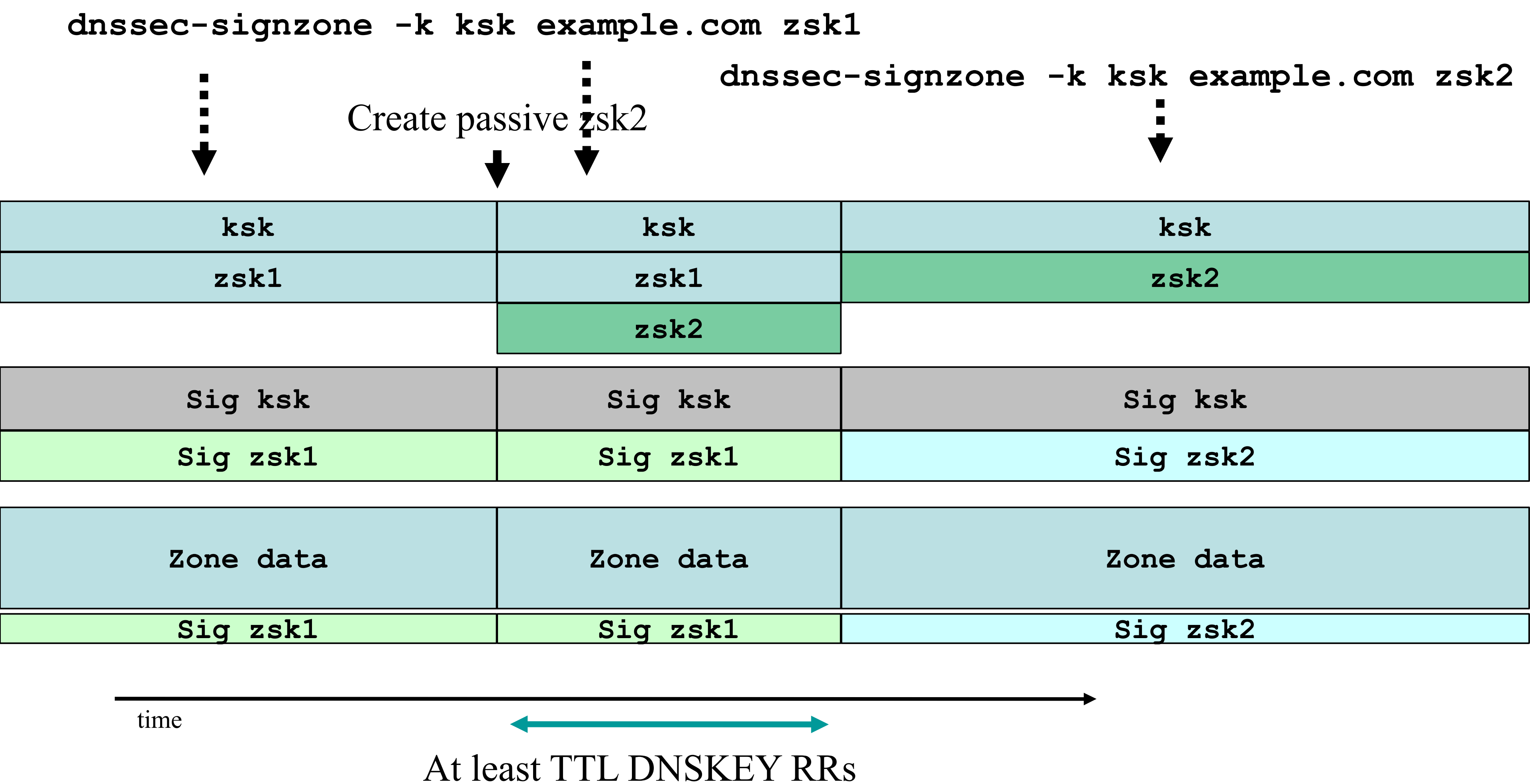
Timing Properties



PRE-publish ZSK rollover

- Introduce the new DNSKEY before you start using it to sign the data.
  - ‘passive and active’ key
  - The passive key is just published, the active key is used for signing
- You could also create two signatures after introducing the key, but that would cause your zone file to grow

ZSK rollover(pre-publish)



ZSK rollover(pre-publish)

Initial	new DNSKEY	New RRSIGs	DNSKEY removal
SOA0	SOA1	SOA2	SOA3
RRSIG10(SOA0)	RRSIG10(SOA1)	RRSIG11(SOA2)	RRSIG11(SOA3)
DNSKEY 1	DNSKEY 1	DNSKEY 1	DNSKEY 1
DNSKEY 10	DNSKEY 10	DNSKEY 10	DNSKEY 11
	DNSKEY 11	DNSKEY 11	
RRSIG1 (DNSKEY)	RRSIG1 (DNSKEY)	RRSIG1 (DNSKEY)	RRSIG1 (DNSKEY)
RRSIG10(DNSKEY)	RRSIG10(DNSKEY)	RRSIG11(DNSKEY)	RRSIG11(DNSKEY)

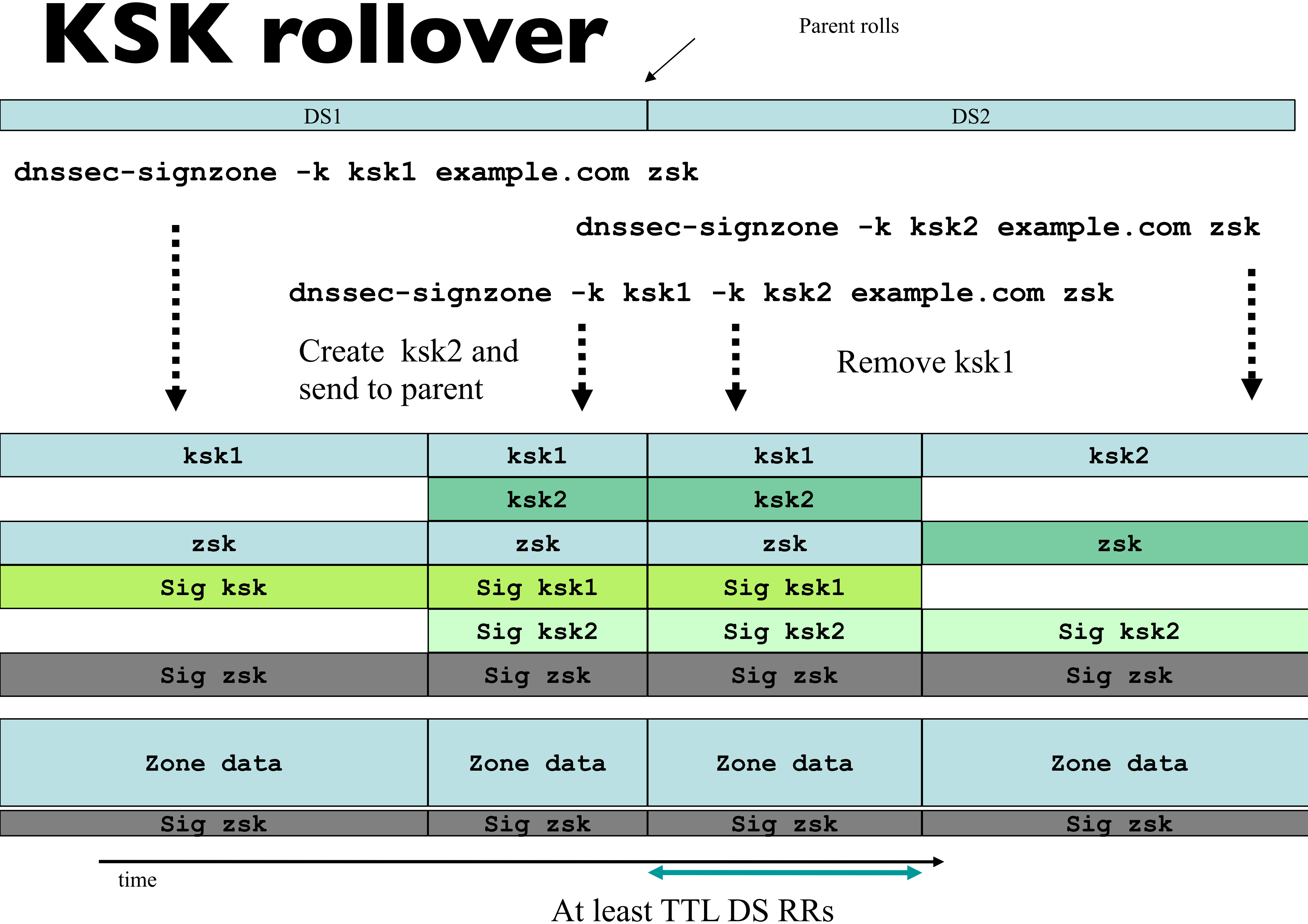
ZSK rollover(double signature)

SOA0	SOA1	SOA2
RRSIG10(SOA0)	RRSIG10(SOA1)	RRSIG11(SOA2)
	RRSIG11(SOA1)	
DNSKEY1	DNSKEY1	DNSKEY1
DNSKEY10	DNSKEY10	DNSKEY11
	DNSKEY11	
RRSIG1(DNSKEY)	RRSIG1(DNSKEY)	RRSIG1(DNSKEY)
RRSIG10(DNSKEY)	RRSIG10(DNSKEY)	RRSIG11(DNSKEY)
	RRSIG11(DNSKEY)	

KSK rollover

- You are dependent on your parent.
  - You cannot control when the parent changes the DS RR
- Use the old KSK until the old DNS had time to expire from caches
- Double signature or pre-publish rollover

KSK rollover



KSK rollover

Initial	New DNSKEY	DS Change	DNSKEY removal
Parent:			
SOA0	SOA0	SOA1	SOA1
RRSIGpar(SOA0)	RRSIGpar(SOA0)	RRSIGpar(SOA1)	RRSIGpar(SOA1)
DS1	DS1	DS2	DS2
RRSIGpar(DS)	RRSIGpar(DS)	RRSIGpar(DS)	RRSIGpar(DS)
Child:			
SOA0	SOA1	SOA1	SOA2
RRSIG10(SOA0)	RRSIG10(SOA1)	RRSIG10(SOA1)	RRSIG10(SOA2)
DNSKEY1	DNSKEY1	DNSKEY1	DNSKEY2
	DNSKEY2	DNSKEY2	
DNSKEY10	DNSKEY10	DNSKEY10	DNSKEY10
RRSIG1(DNSKEY)	RRSIG1(DNSKEY)	RRSIG1(DNSKEY)	RRSIG2(DNSKEY)
	RRSIG2(DNSKEY)	RRSIG2(DNSKEY)	
RRSIG10(DNSKEY)	RRSIG10(DNSKEY)	RRSIG10(DNSKEY)	RRSIG10(DNSKEY)

Planning for emergency Keys rollovers

- A compromised Key can be used as long as a valid trust chain exists
  - As long as a signature over the compromised key in the trust chain is valid
  - As long as a parental DS points to the compromises key
  - As long as the key is anchored in resolvers and used as SEP
- Tradeoff between abuse of the compromised key and cached data validation
- Needs a documented procedure ready

KSK compromising

- The DS pointing to the key or the TA should be replaced as soon as possible
- Keep the chain of trust
  - Introduce a new KSK into the key set, keep the compromised key in the key set
  - Sign the key set with short validity period
    - Signature should expire shortly after the DS appears in parent zone and old DSes has expired from cache
  - Upload the DS for the new key to the parent
  - Follow the procedure for normal KSK rollover
  - Remove the compromised key and re-sign the key set to the normal validity period

KSK compromising

- Breaking the chain of trust
- Two method to break the chain
  - By removing the key in child zone, re-sign key set and send the DS to the parent
    - Zone is bogus and attackers zone valid
  - By removing the DS from Parent zone and the key from child zone
    - Zone insecure
- If A TA is compromised
  - Resolvers should be notify
  - New key distributed and authenticated out-of-band

ZSK compromise

- No child/parent interactions needed
  - Zone should be re-signed with a new ZSK as soon as possible
  - Immediate disappearance of the compromise key can lead to validation problems
  - Until signature expired on the compromised key, the domain may still be at risks.

Questions ???