Practical DNSSEC Debugging

Jaap Akkerhuis

With thanks to Olaf Kolkman & Willem Toorop

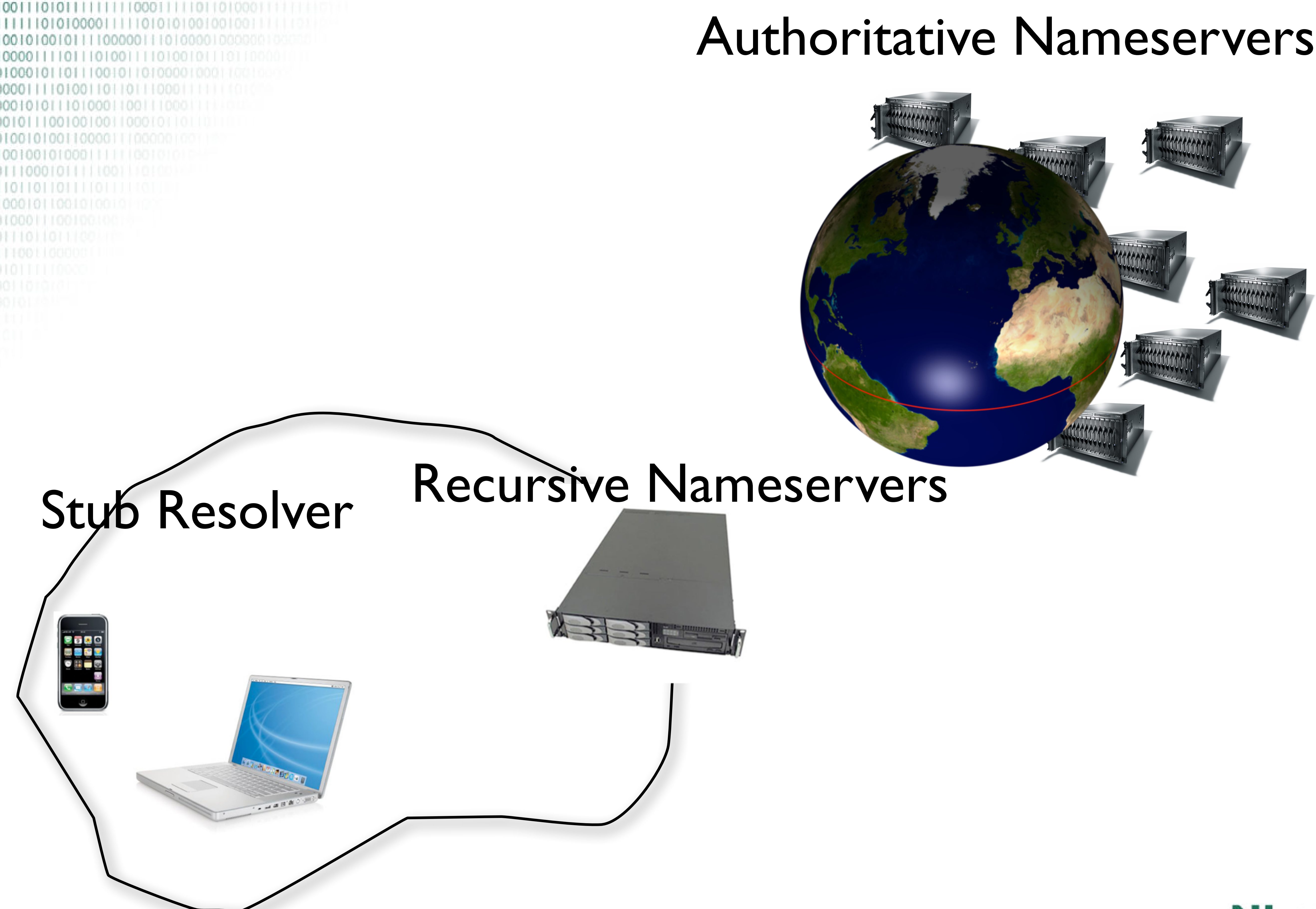
NLnet Labs

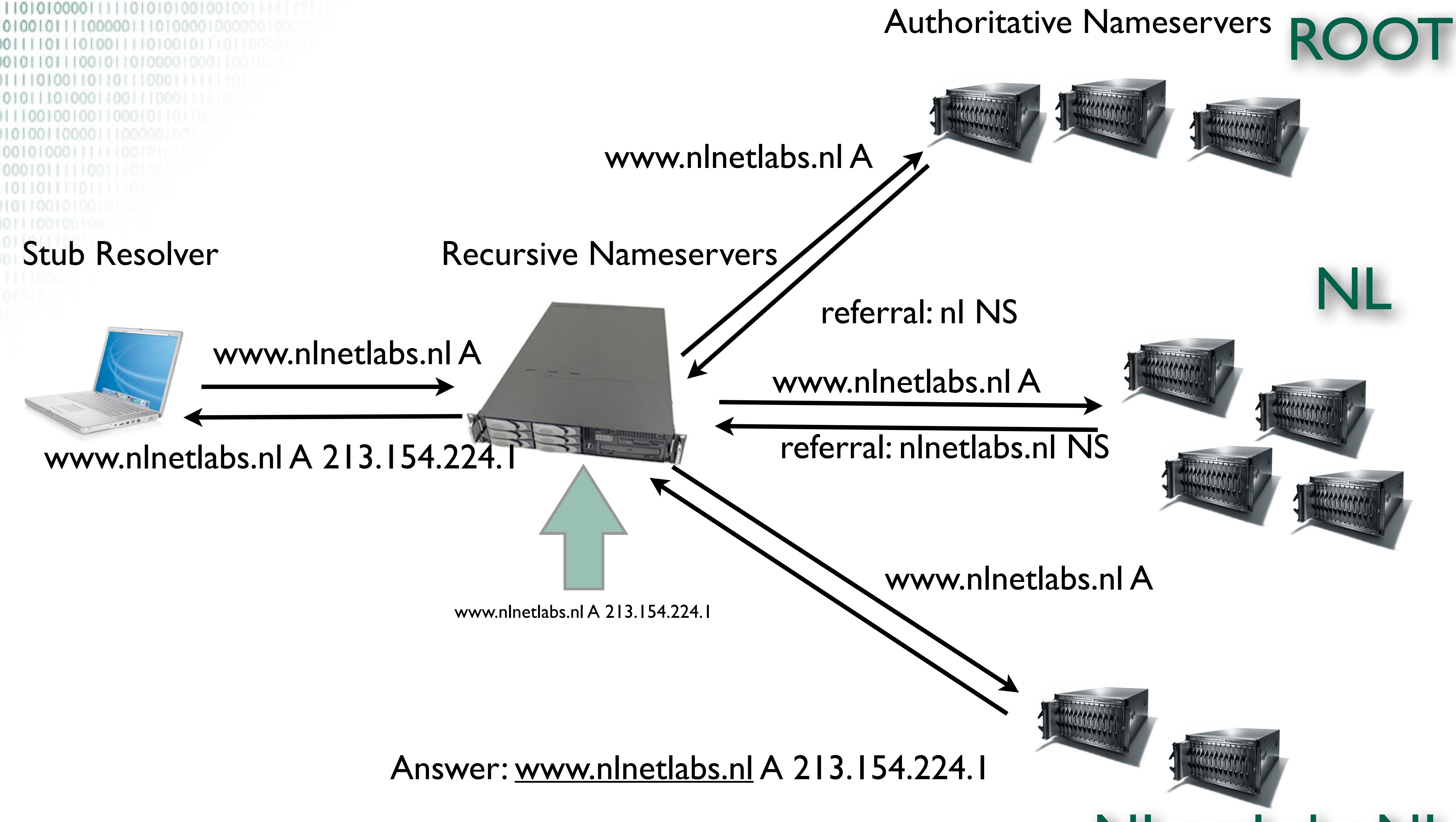
who m i

- Old Timer
- Early DNSSEC Reviewer
- NLnet Labs Guy

Overview

- DNS Overview
- DNSSEC in general
- The RR's
- Chain of Trust
- How to put it together
- Debugging aids


Authoritative Nameservers
Recursive Nameservers
Stub Resolver



root.hints: location of the root servers

```
; <<>> DiG 9.7.0b2 <<>> @k.root-servers.net www.nlnetlabs.nl
; (2 servers found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 41886
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 7, ADDITIONAL: 12
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;www.nlnetlabs.nl.   IN A

;; AUTHORITY SECTION:
nl.             172800 IN NS nl1.dnsnode.net.
nl.             172800 IN NS ns1.nic.nl.
nl.             172800 IN NS ns2.nic.nl.
nl.             172800 IN NS ns3.nic.nl.
nl.             172800 IN NS ns4.nic.nl.
nl.             172800 IN NS ns-nl.nic.fr.
nl.             172800 IN NS sns-pb.isc.org.

;; ADDITIONAL SECTION:
nl1.dnsnode.net.    172800 IN A 194.146.106.42
ns1.nic.nl.         172800 IN A 193.176.144.2
ns2.nic.nl.         172800 IN A 213.154.241.28
ns3.nic.nl.         172800 IN A 194.171.17.2
ns4.nic.nl.         172800 IN A 62.4.86.232
ns-nl.nic.fr.       172800 IN A 192.93.0.4
sns-pb.isc.org.     172800 IN A 192.5.4.1
ns1.nic.nl.         172800 IN AAAA 2a00:d78::102:193:176:144:2
ns2.nic.nl.         172800 IN AAAA 2001:7b8:606::28
ns3.nic.nl.         172800 IN AAAA 2001:610:0:800d::2
ns-nl.nic.fr.       172800 IN AAAA 2001:660:3005:1::1:2
sns-pb.isc.org.     172800 IN AAAA 2001:500:2e::1

;; Query time: 4 msec
;; SERVER: 2001:7fd::1#53(2001:7fd::1)
;; WHEN: Tue Apr  6 14:12:44 2010
;; MSG SIZE  rcvd: 447
```

Question

Referal

Cache and TTL

```
;; ANSWER SECTION:
www.nlnetlabs.nl.  10200 IN A 213.154.224.1
```

- TTL is a parameter that indicates how long data is to remain in a cache
- TTL value is set by the zone owner
- TTL decreases while in the cache

Cache Poison

- Attack is based on ‘predicting’ properties
  - e.g. when asking a question to a female you expect a female voice to answer
- If you ask a question with a specific QID you expect that QID in the answer
  - Cache poisoner will take a wild guess

Isn't Query ID only sufficient?

Chance that *n* people have different birthdays

$$\bar{p}(n) = 1 \times \left(1 - \frac{1}{365}\right) \times \left(1 - \frac{2}{365}\right) \cdots \left(1 - \frac{n-1}{365}\right) = \frac{365 \times 364 \cdots (365 - n + 1)}{365^n} = \frac{365!}{365^n (365 - n)!}$$

Chance that *n* people have the same birthday

$$p(n) = 1 - \bar{p}(n).$$

n	P(n)
10	11,17%
20	41,1%
23	50,7%
30	70,6%
50	97%
57	99%
100	99,99997%

from: http://en.wikipedia.org/wiki/Birthday_problem

NLnet Labs

Bits	50%	5%	Aka
16	10 s	1 s	Unpatched server, random ID
26	2.8 h	17 m	Patched, using only 1024 ports
34	28 days	2.8 days	unbound with defaults
44	28444 days	2844.4 days	unbound with 0x20 and source addresses configured

Kaminsky’s variant

- Classic cache poisoning gave you ‘a few tries’ to get in between the outgoing question and incoming answer
- Kaminsky came with a scheme where the culprit can keep trying
  - Surprisingly simple, a wonder nobody thought of the variety before

DNSSEC

- Prevents cache poisoning for existing labels
- Proves non-existence of labels

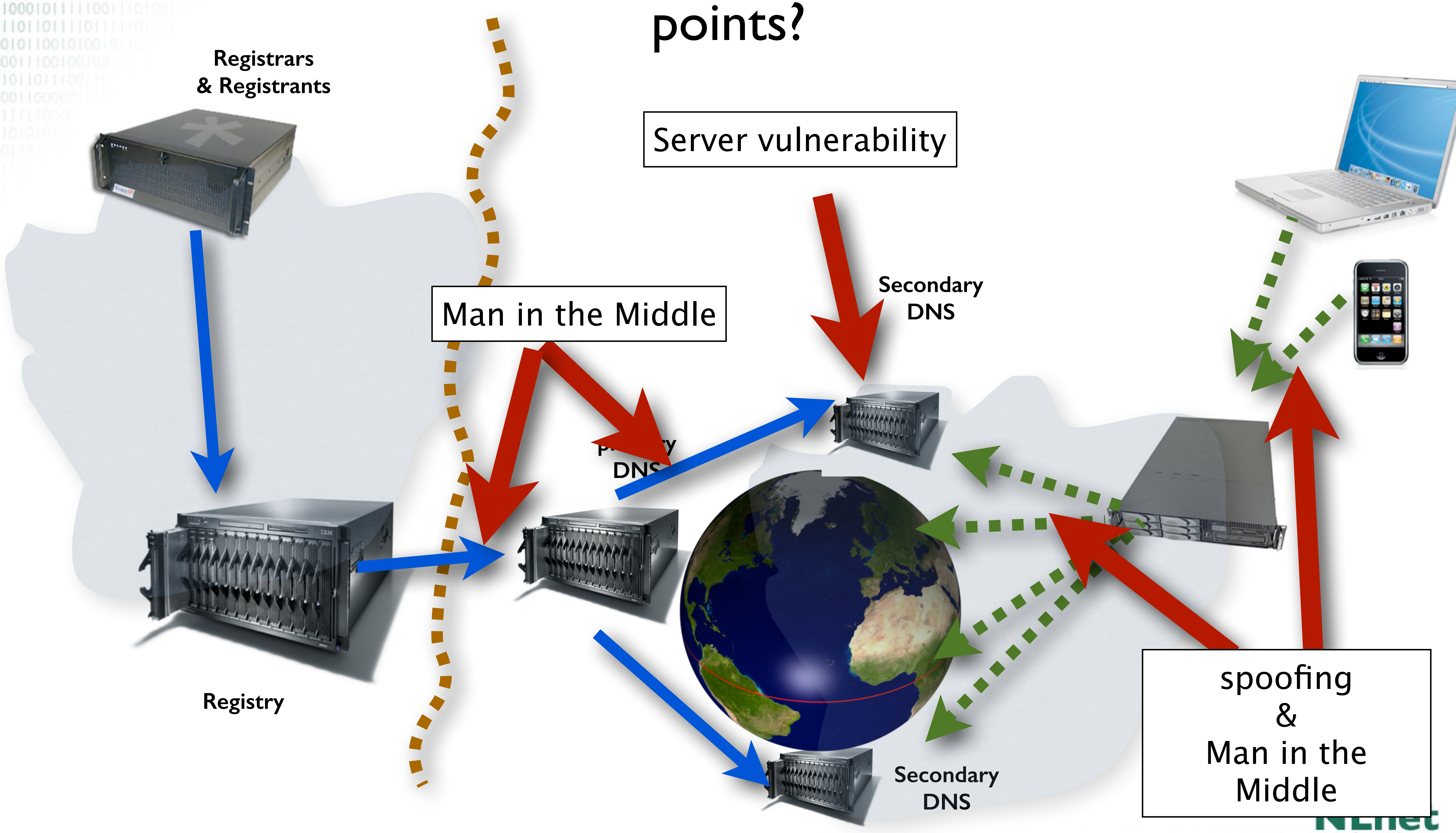
- Attacks à la Kaminsky showed the need

DNSSEC Mechanisms

- New Resource Records
- Secure the Zone
- Delegating Signing Authority

Data flow through the DNS

Where are the vulnerable points?



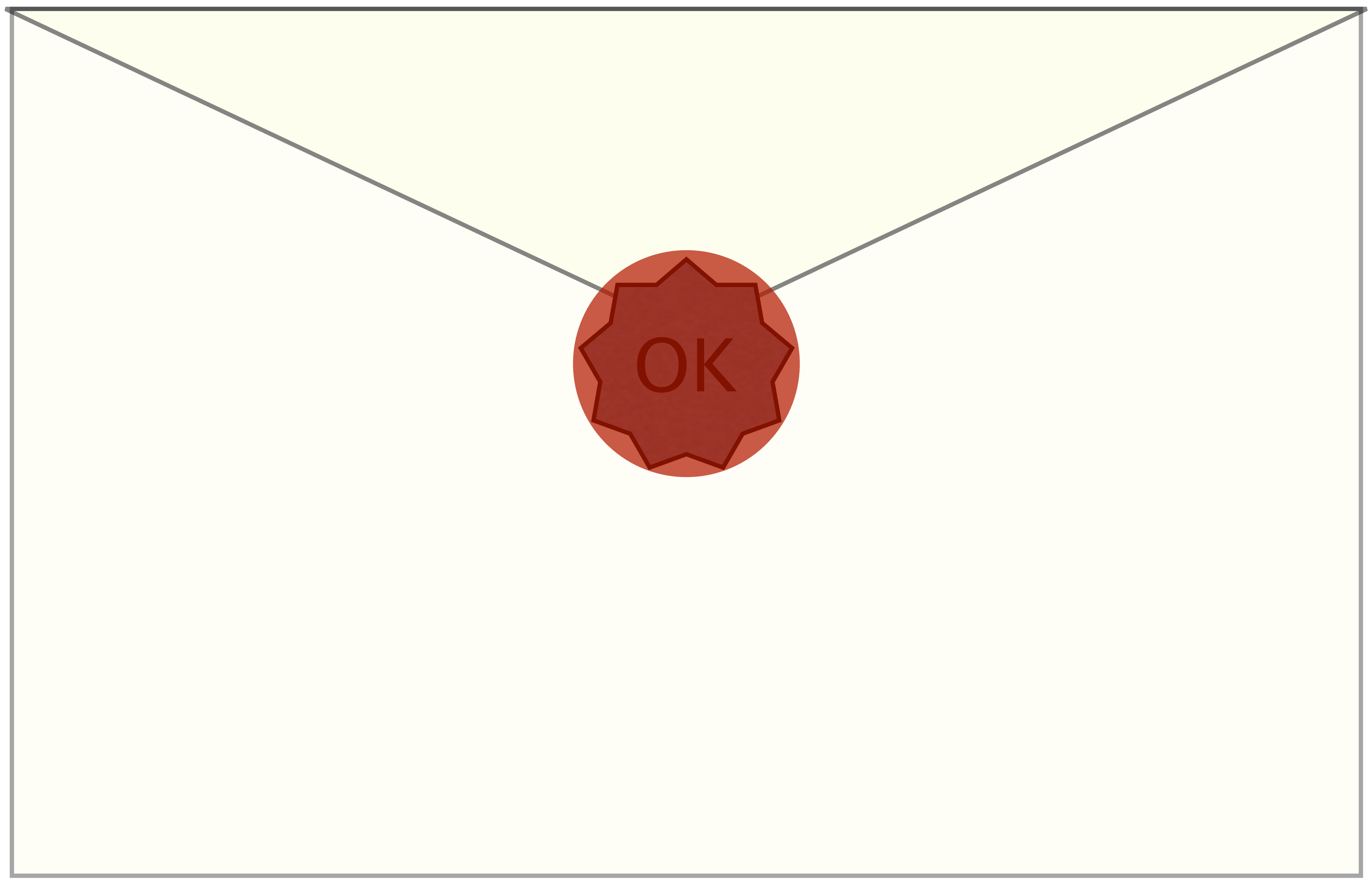
DNSSEC protects all this end-to-end

- As an aside:
  There is a protection mechanism against the man in the middle: TSIG
  - Provides hop-by-hop security
  - TSIG is operationally deployed today
  - Based on shared secret: not scalable
- Used a lot for AXFR transactions

What does DNSSEC provide

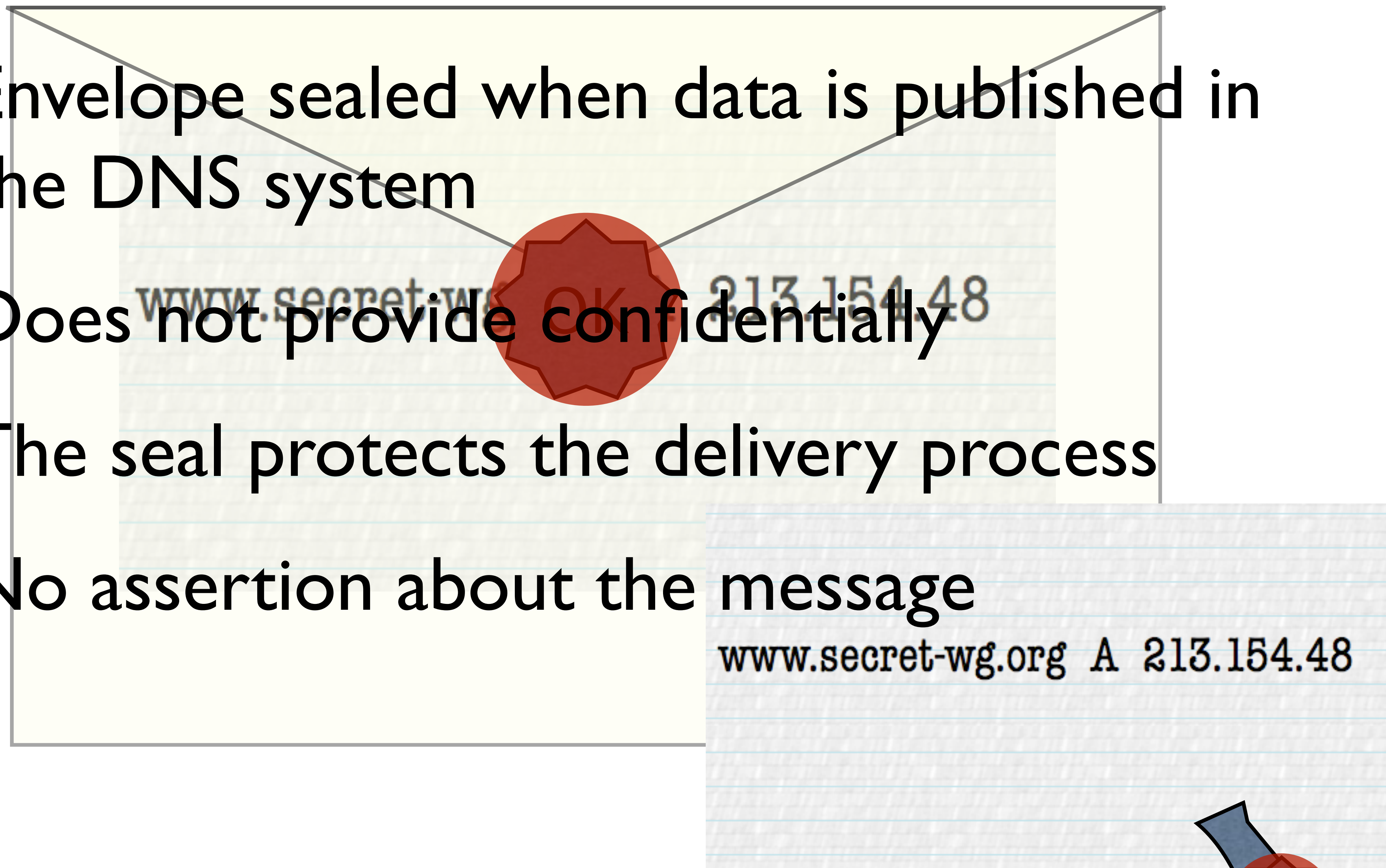
- Provides message authentication and integrity verification through cryptographic signatures
    - You know who provided the signature
    - No modifications between signing and validation
- It does not provide authorization
- It does not provide confidentiality
- It does not provide protection against DDOS

Metaphor



Metaphor

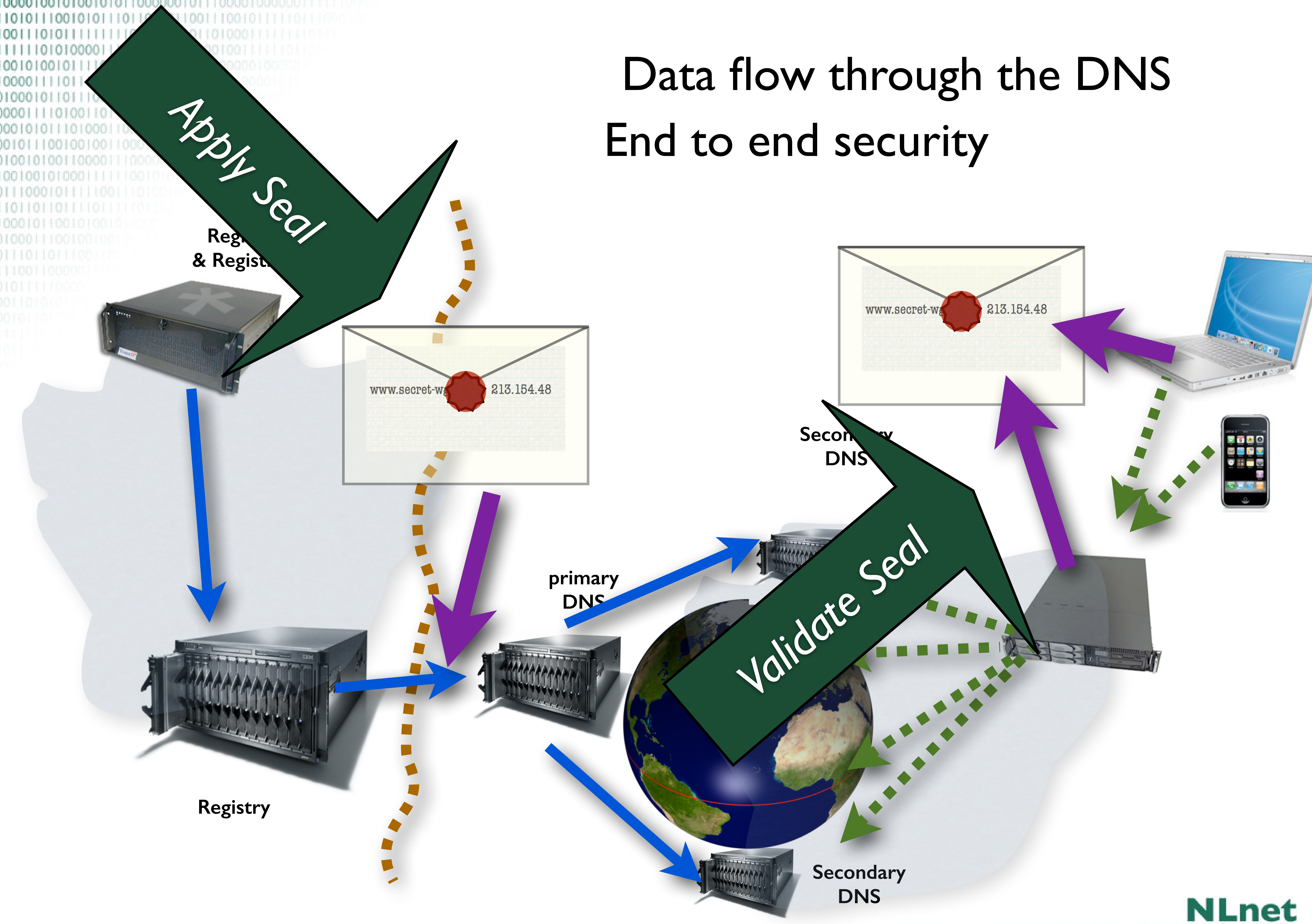
- Envelope sealed when data is published in the DNS system
- Does not provide confidentially
- The seal protects the delivery process
- No assertion about the message



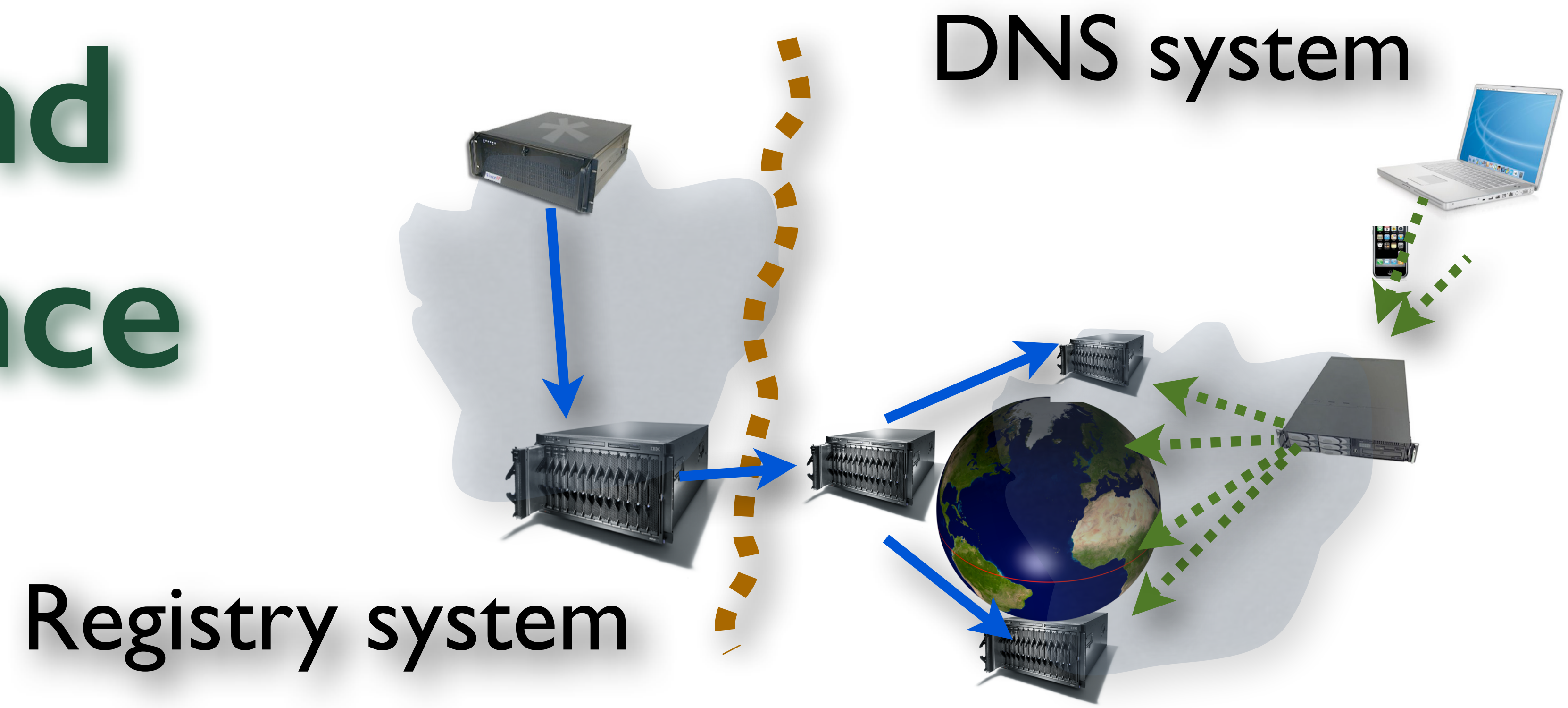
NLnet Labs

Data flow through the DNS

End to end security



Trust and Confidence



- DNSSEC enables confidence in the DNS
- It does not change the trust we put in the Registry/Registrar procedures
  - Although introduction of DNSSEC may improve some of the procedures

NLnet Labs

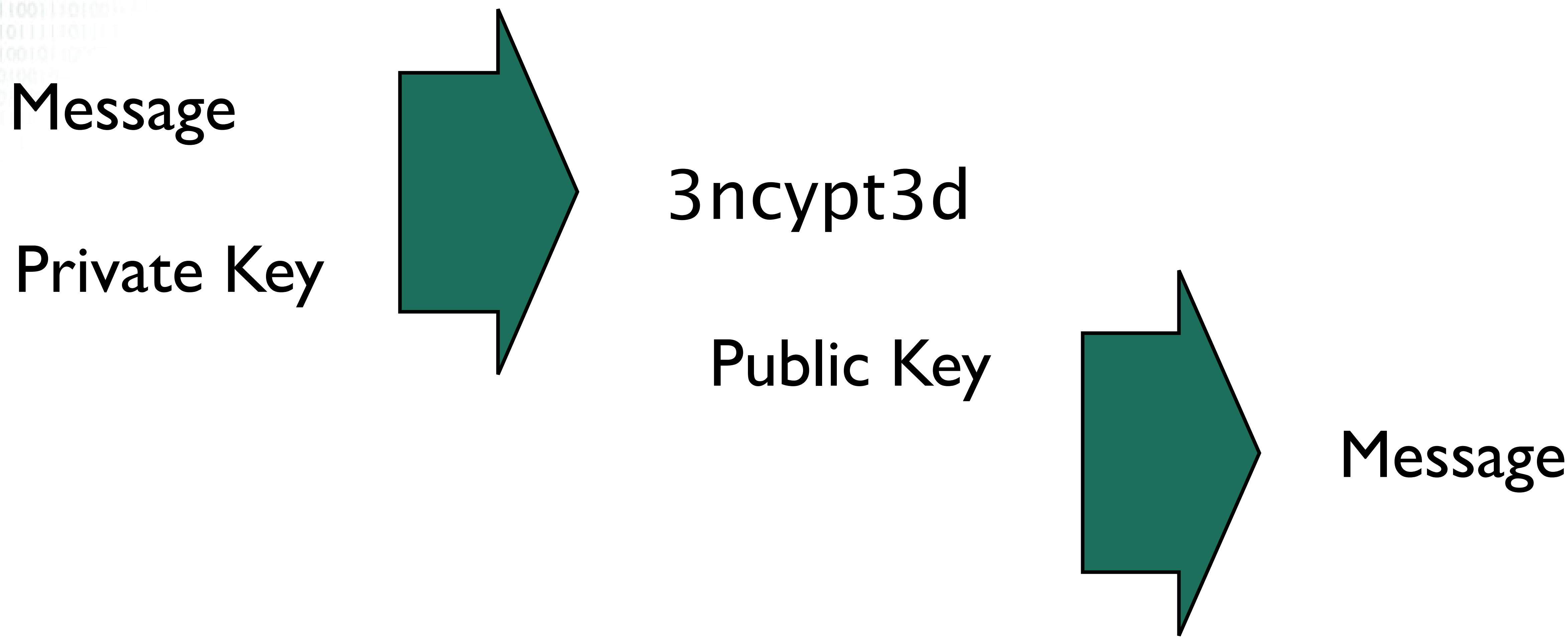
The mechanism used

- Using public key cryptographic algorithms signatures are applied over the DNS data
- By comparing the signatures with public keys the integrity and authenticity of the data can be established.

Public key cryptography in a nutshell

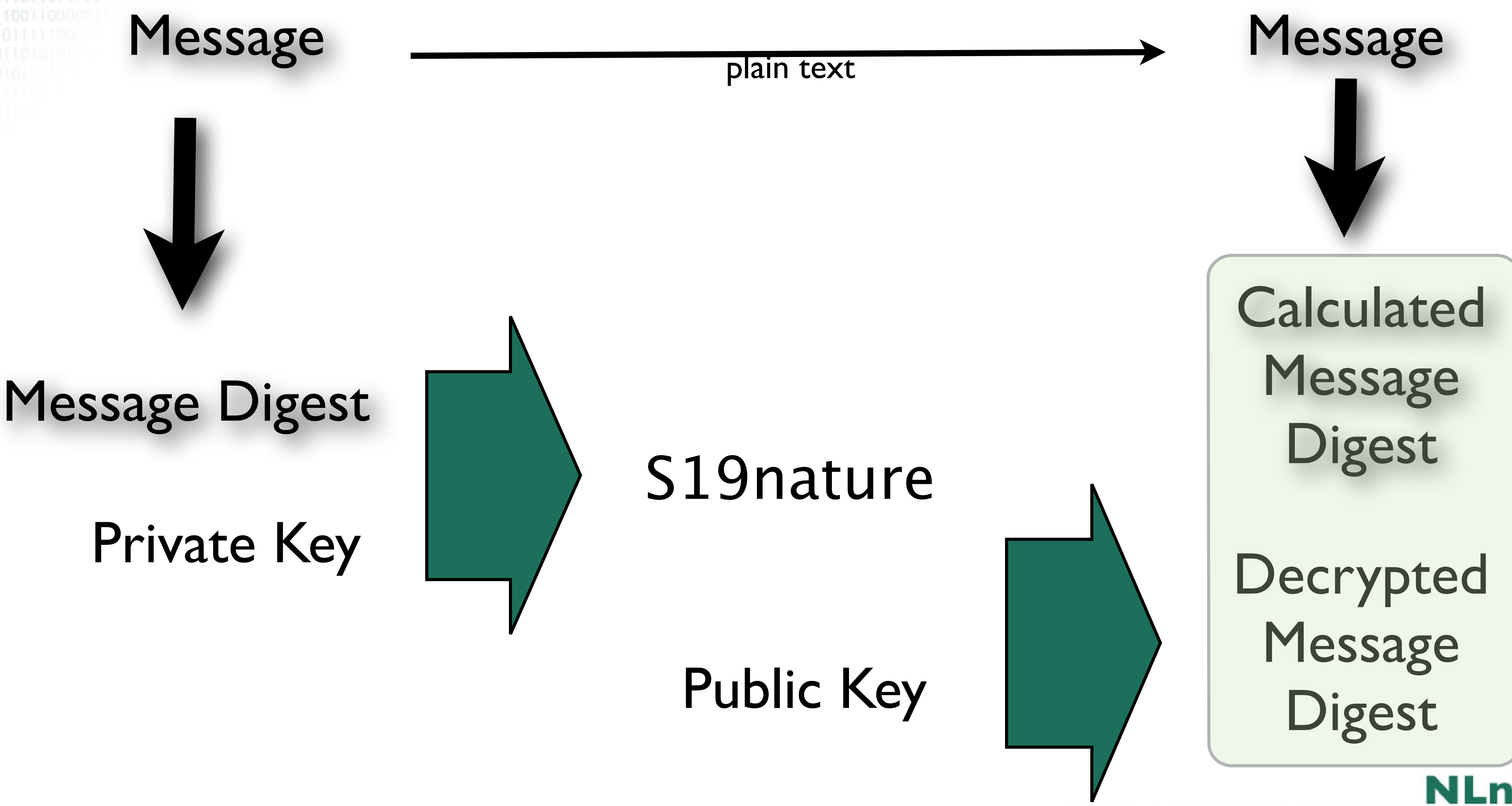
- Two large numbers and an encryption/decryption algorithm
- One of the numbers (the private key) and a message are used for encryption
- The other number (public key) and the decryption algorithm can be used to retrieve the original message

Public Crypto



Works only with matching key:
If you can decrypt with a public key you may assert the message was signed with corresponding private key

Use that method for signatures



In Practice

- Key generation and signing is done by tools
- Validating and signing entity need to communicate which algorithms for hashing and public key cryptography is needed: e.g. RSASHA1, RSASHA256 or DSA

Holy Trinity

- Private Key: kept private and stored locally
- Public Keys: Published in the DNS as a DNSKEY Resource Record
- Signatures: Published in the DNS as a RRSIG Resource Record

NLnet Labs

Signing is done per Zone

- Each zone has one or more key-pairs for signing
- If you have the public keys from a zone you can validate signatures made with the corresponding private keys
- However, signing a complete zone does not scale

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 the atomic data units in the DNS
- RRsets are signed, not the individual RRs

DNSKEY RDATA

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

```
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== )
```

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

Validate Public Keys

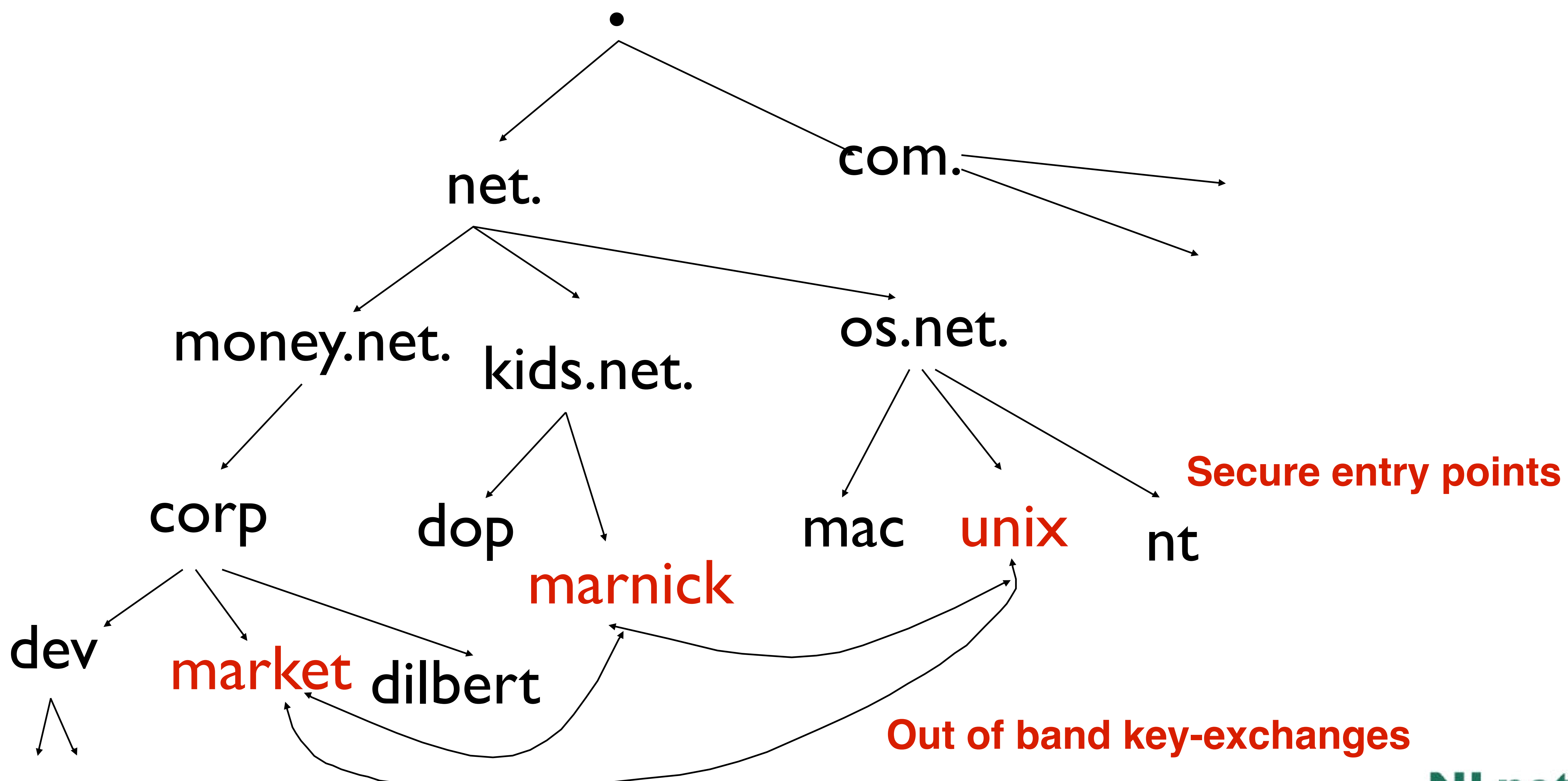
- Make sure you get them from the appropriate entity and configure them as trust-anchors
- If you validate against the wrong public key there is a problem again
- For DNSSEC: key distribution through the DNS
  - Ideally only one key needed: that of the root of the DNS hierarchy (more on that later)

NLnet Labs

Delegating Signing Authority

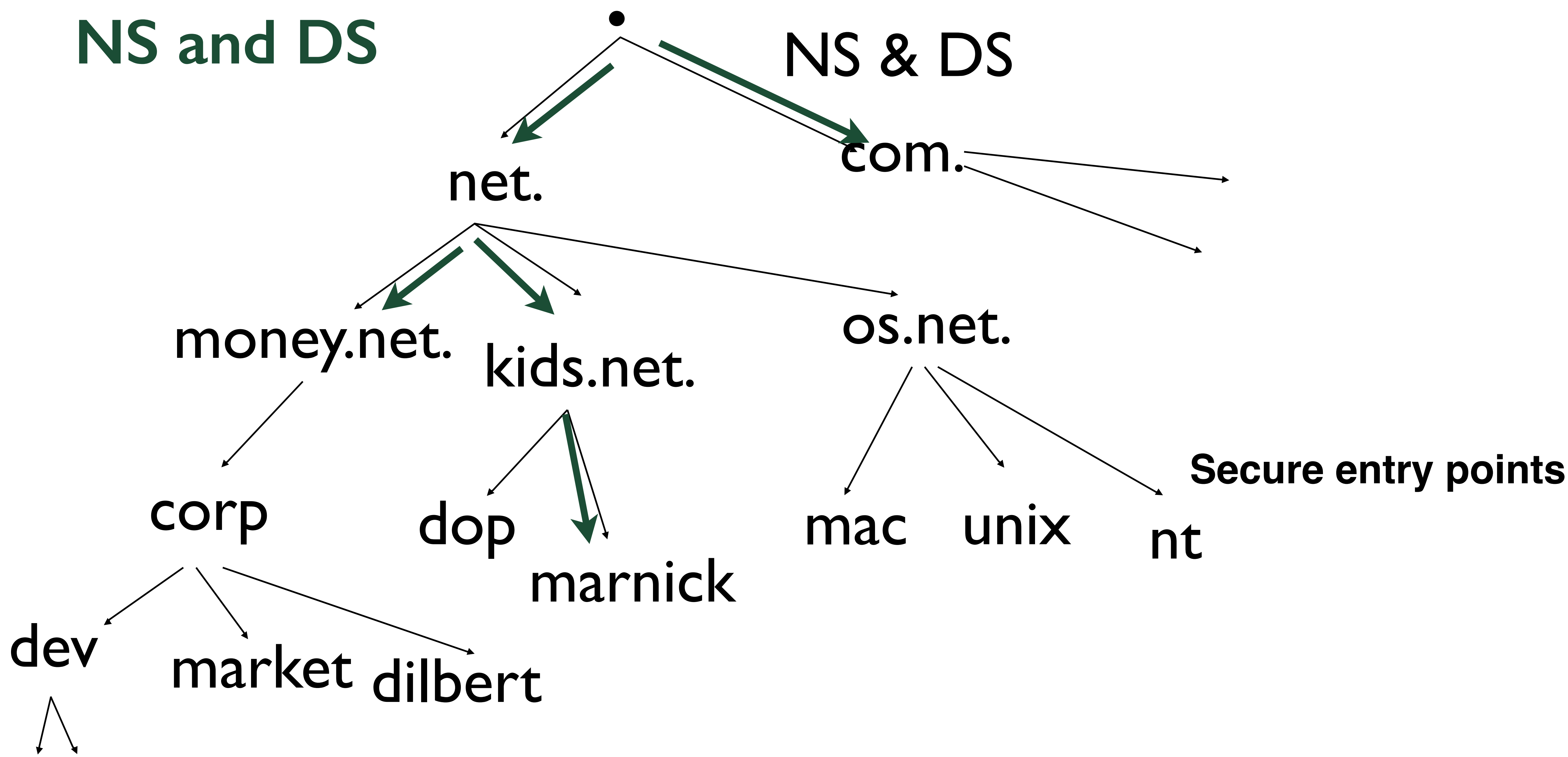
Validating against configured keys

- Key distribution does not scale!



Locally Secured Zones

- Delegate Signing Security



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

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 (
                                   239af98b923c023371b52
                                   1g23b92da12f42162b1a9 )
```

Key Problem

- Interaction with parent administratively expensive
  - Should only be done when needed
  - You might want to lock these in hardware
- Signing zones should be fast
  - Memory restrictions
  - Space and time concerns
  - Operational exposure higher

More Than One Key: KSK and ZSK

- 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)

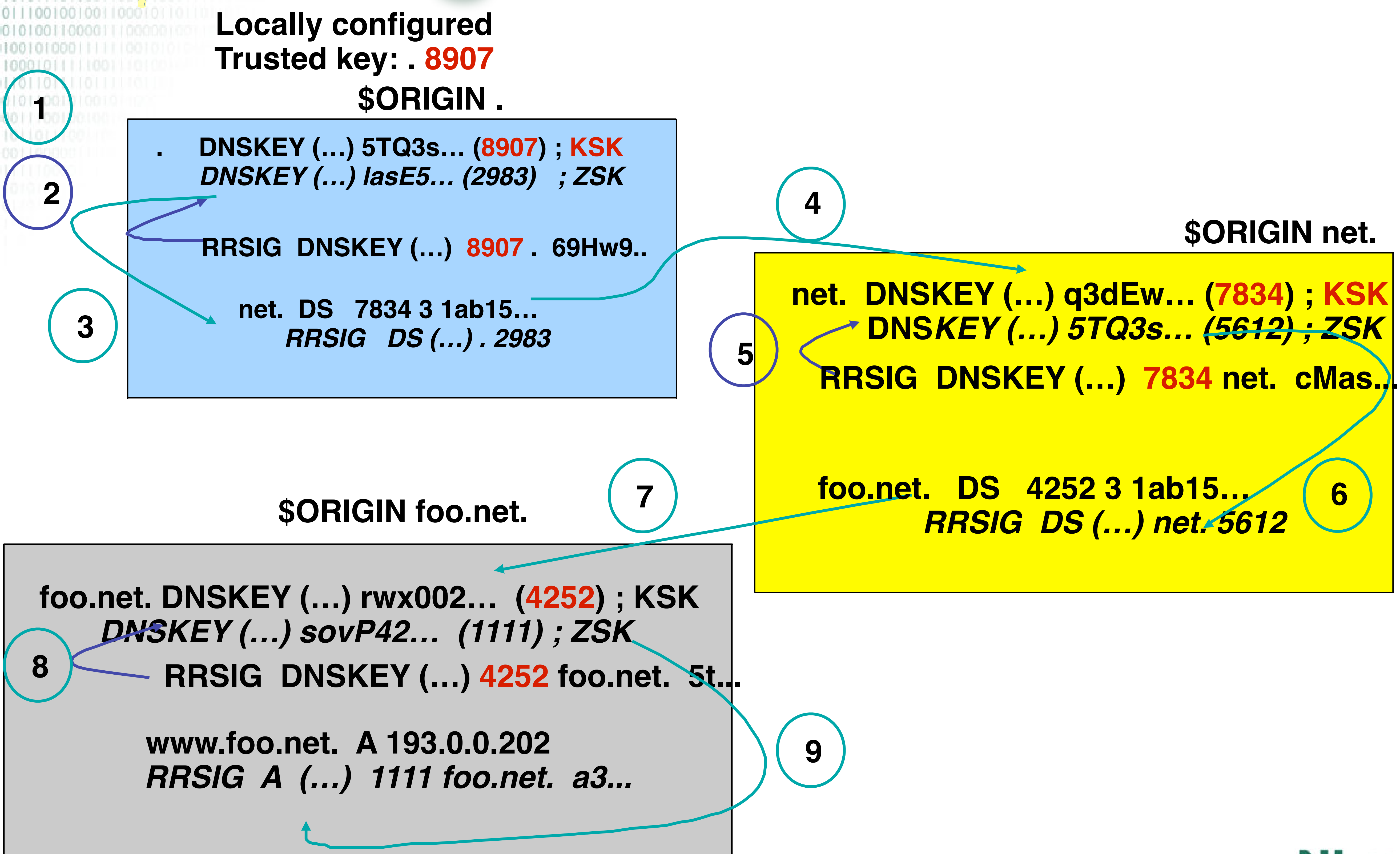
The Important Considerations

- KSK and ZSK have different 'shielding' properties: KSK on smartcard, ZSK on disk
- ZSK needs 'daily' or permanent use.
- KSK less frequent
- ZSK change needs no involvement with 3rd parties
- KSK may need uncontrolled cooperation from 3rd parties

Initial Key Exchange

- Child needs to:
  - Send key signing keyset to parent
- Parent needs to:
  - Check childs zone
    - for DNSKEY & RRSIGs
  - Verify if key can be trusted
  - Generate DS RR

Walking the Chain of Trust



Chain of Trust Verification, Summary

- Data in zone can be trusted if signed by a Zone-Signing-Key
- Zone-Signing-Keys can be trusted if signed by a Key-Signing-Key
- Key-Signing-Key can be trusted if pointed to by trusted DS record
- DS record can be trusted
  - if signed by the parents Zone-Signing-Key
  - or
  - DS or DNSKEY records can be trusted if exchanged out-of-band and locally stored (Secure entry point)

Where are we

- DNSKEY
- RRSIG
- DS

NLnet Labs

Offline Signing and Denial of Existence

- Problems with on-the-fly signing
  - Private key needs to be stored on an Internet facing system
  - Performance, signing is a CPU expensive operation
- How does one provide a proof that the answer to a question does not exist?

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

NLnet Labs

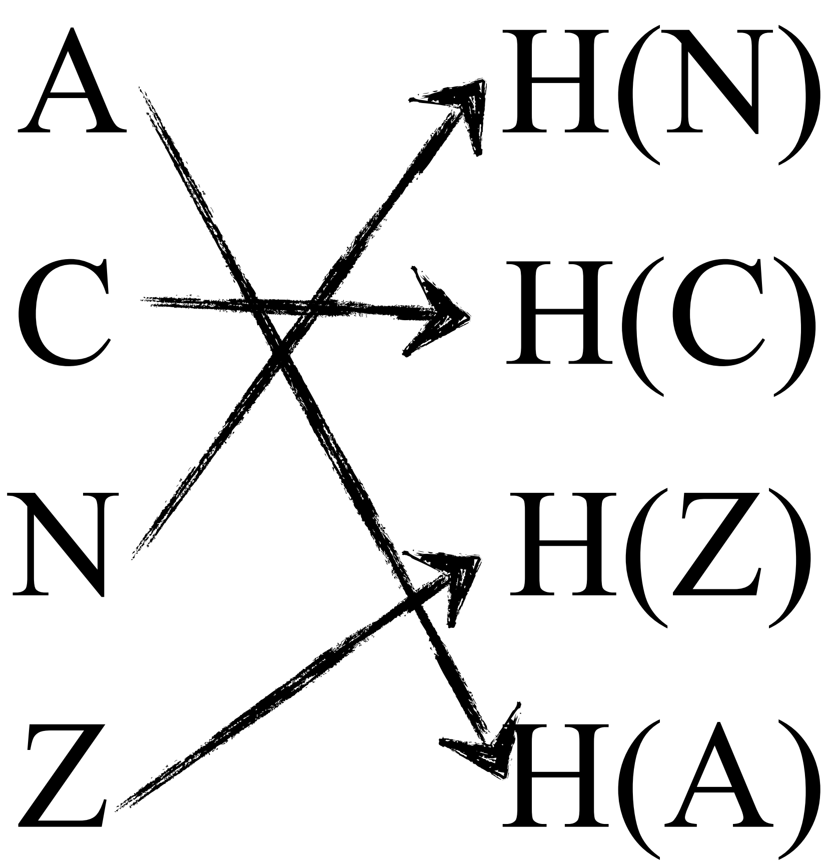
NSEC Walking

- NSEC records allow for zone enumeration
- Providing privacy was not a requirement at the time
- Zone enumeration is a deployment barrier

- Solution has been developed: NSEC3
  - RFC 5155
  - Complicated piece of protocol work
  - Hard to troubleshoot
  - Only to be used over Delegation Centric Zones

NLnet Labs

NSEC3



- Creates a linked list of the hashed names
- Non-existence proof of the hash proofs non-existence of original
- Dictionary attack barriers:
  - Salt
  - Iterations

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 and NSEC3: Indicates which name is the next one in the zone and which typecodes are available for the current name
    - authenticated non-existence of data

Other Keys in the DNS

- DNSKEY RR can only be used for DNSSEC
  - —Keys for other applications need to use other RR types
- CERT
  - —For X.509 certificates
- Application keys under discussion/development
  - —IPSECKEY
    - SSHFP Summary for now
  - DANE!!!

Practicalities

- Create the keys
- Sign the zones
- Keep everything up to date

Key creation

```
$ ldns-keygen -a RSASHA256 -b 1024 1sand0s.nl
K1sand0s.nl.+008+24201
$ ls
K1sand0s.nl.+008+24201.ds K1sand0s.nl.+008+24201.private
K1sand0s.nl.+008+24201.key
$ ldns-keygen -k -a RSASHA256 -b 1024 1sand0s.nl
K1sand0s.nl.+008+24040
$ ls
K1sand0s.nl.+008+24040.ds K1sand0s.nl.+008+24201.ds
K1sand0s.nl.+008+24040.key K1sand0s.nl.+008+24201.key
K1sand0s.nl.+008+24040.private
K1sand0s.nl.+008+24201.private
```

Zone signing

```
$ ldns-signzone 1sand0s.nl.zone K1sand0s.nl.+008+24040 \
     K1sand0s.nl.+008+24201
$ ls
1sand0s.nl.zone K1sand0s.nl.+008+24040.private
1sand0s.nl.zone.signed K1sand0s.nl.+008+24201.ds
K1sand0s.nl.+008+24040.ds k1sand0s.nl.+008+24201.key
```

Notify Parent

Sending the DS to the Parent to attach to the chain of trust

Parent puts this in his/her zone

Signs the zone and publish etc …

Resigning needs

- When zone changes
  - New contents
- When RRSIG expires
  - ldns-signzone 30 days
- New chore for sys admin!
  - Automate
    - (ask for a raise)

Key rollovers

NLnet Labs

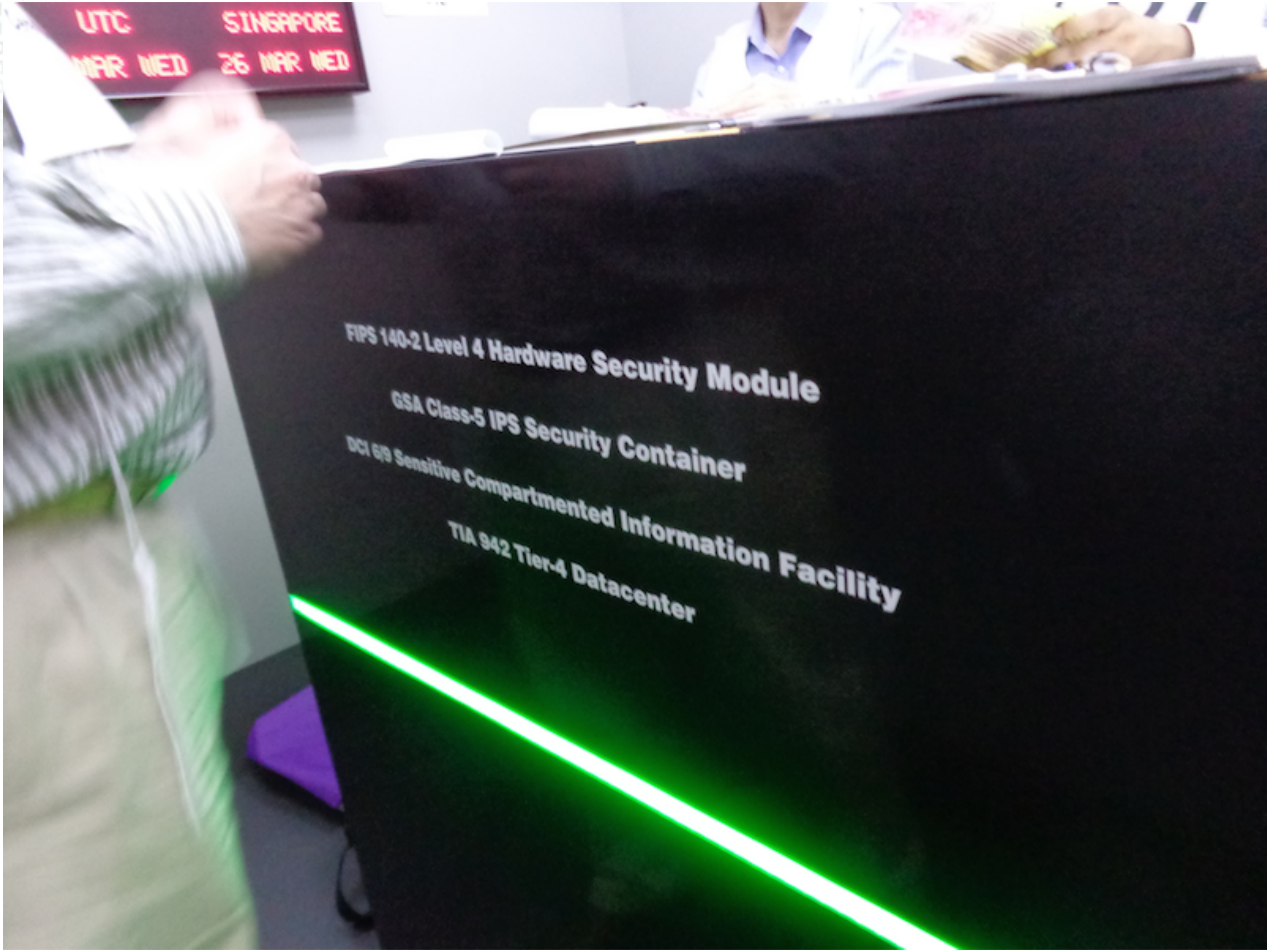
DNSKEY flavours

- Zone Signing 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/parties
- 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

NLnet Labs

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


UTC
SINGAPORE
MAR WED
26 MAR WED
FIPS 140-2 Level 4 Hardware Security Module
GSA Class-5 IPS Security Container
DCI 69 Sensitive Compartmented Information Facility
TIA 942 Tier-4 Datacenter

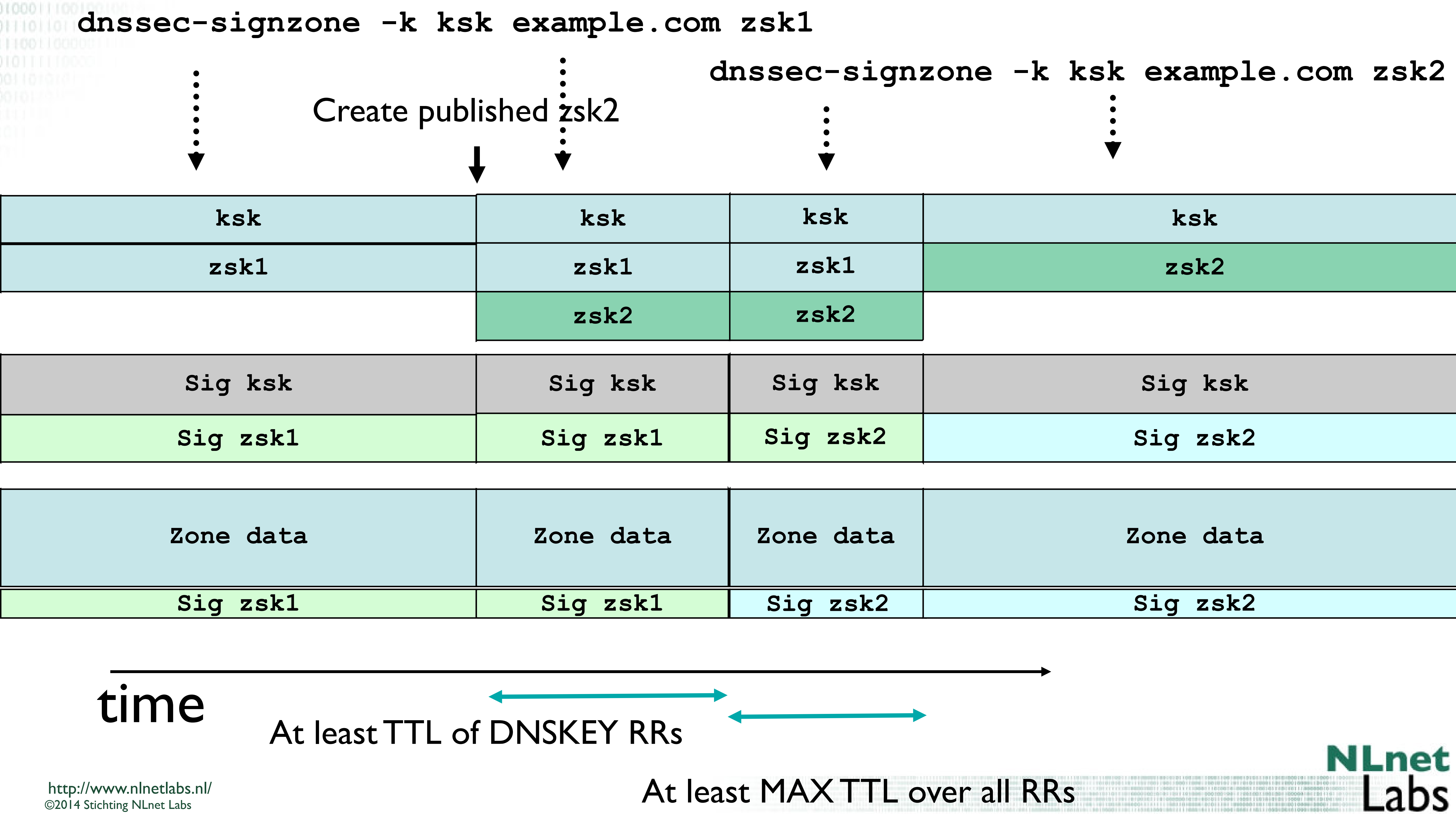

PCH
ccTLD DNSSEC
Key Ceremony
#10
PCH
Packet Clearing House
PCH
ccTLD DNSSEC
Key Ceremony
#10
CA1

NLnet Labs

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 DNSKEYs you would like to be able to use DNSSIGs from the cache
- When you happen to get old DNSKEYs from the cache you would like to use new DNSSIGs
- Try to make sure both old and new keys are available
- Or, try to make sure both old and new sigs are available

ZSK rollover



NLnet Labs

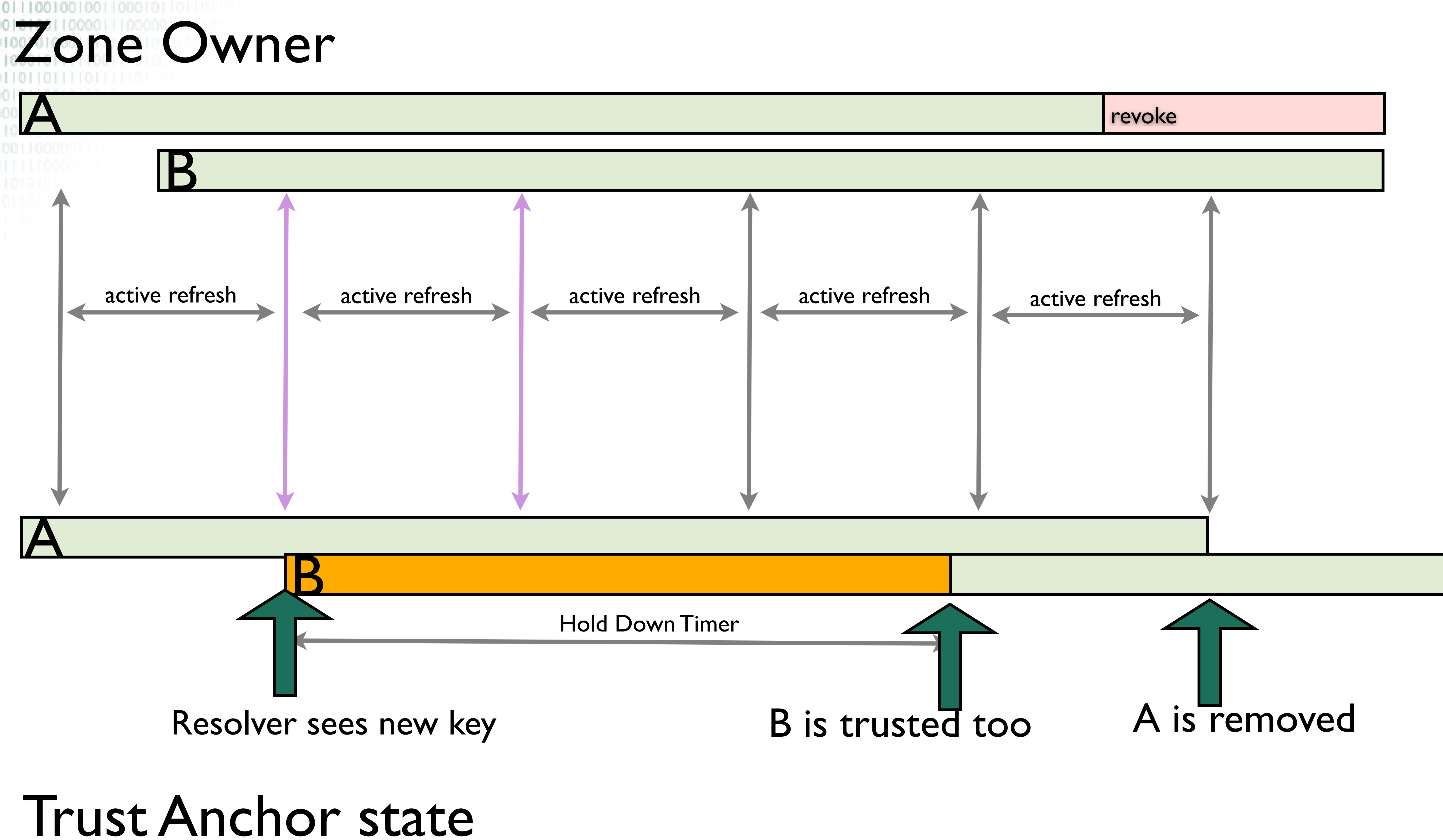
KSK rollover

- 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 propagate from caches

NLnet Labs

RFC 5011 Concepts

- Trust anchor maintenance based on existing trust relation
- New keys only accepted after its been seen for more than 30 days (Hold Down)
- Signaling retirement of the key by setting a ‘revoke’ flag


Zone Owner
A
revoke
B
active refresh
active refresh
active refresh
active refresh
active refresh
A
B
Hold Down Timer
Resolver sees new key
B is trusted too
A is removed
Trust Anchor state

Keeping up to date

By Hand?

- Error prone and complex
- Easy to forget

Use Tools!

- Cronjobs + Scripts
- Tools like OpenDNSSEC
- Build-in tools

NLnet Labs

Problem Shooting

Help, the Internet just stopped working

- DNS is the very resilient
  - Install and forget
    - 90 % works by accident
- DNSSEC makes the (this) system brittle
  - Only three results
    - Secure, insecure, “bogus”
    - Needs periodic maintenance

Most seen problems

- Expired SIGS
- Expired KEYS
- Missing DS (signed but indeterminable state)
- Faulty clocks
- Algorithms mismatch
  - Incomplete rollovers

Debugging tools: dig, date, pencil & paper

- Dig for the Records
- Check the dates
- Draw the chain and see where it brakes

NLnet Labs Tools

- Signing tools etc
- Drill: like dig but with DNSSEC integrated
- Bottom up (-S) or Top down (-TD) DNSSEC chain checking
- Unbound-host -f <key> -d

Online Tools

- http://dnssec-debugger.verisignlabs.com
- http://dnsviz.net
- http://dnscheck.iis.se (http://dnscheck.se)
- http://www.zonecheck.fr/demo/

More Info

http://www.internetsociety.org/deploy360/dnssec/

http://workbench.sidnlabs.nl

https://dnssec.surfnet.nl

Questions

(If you like our work, please consider sponsoring us)