DNSSEC

The details

Presented by

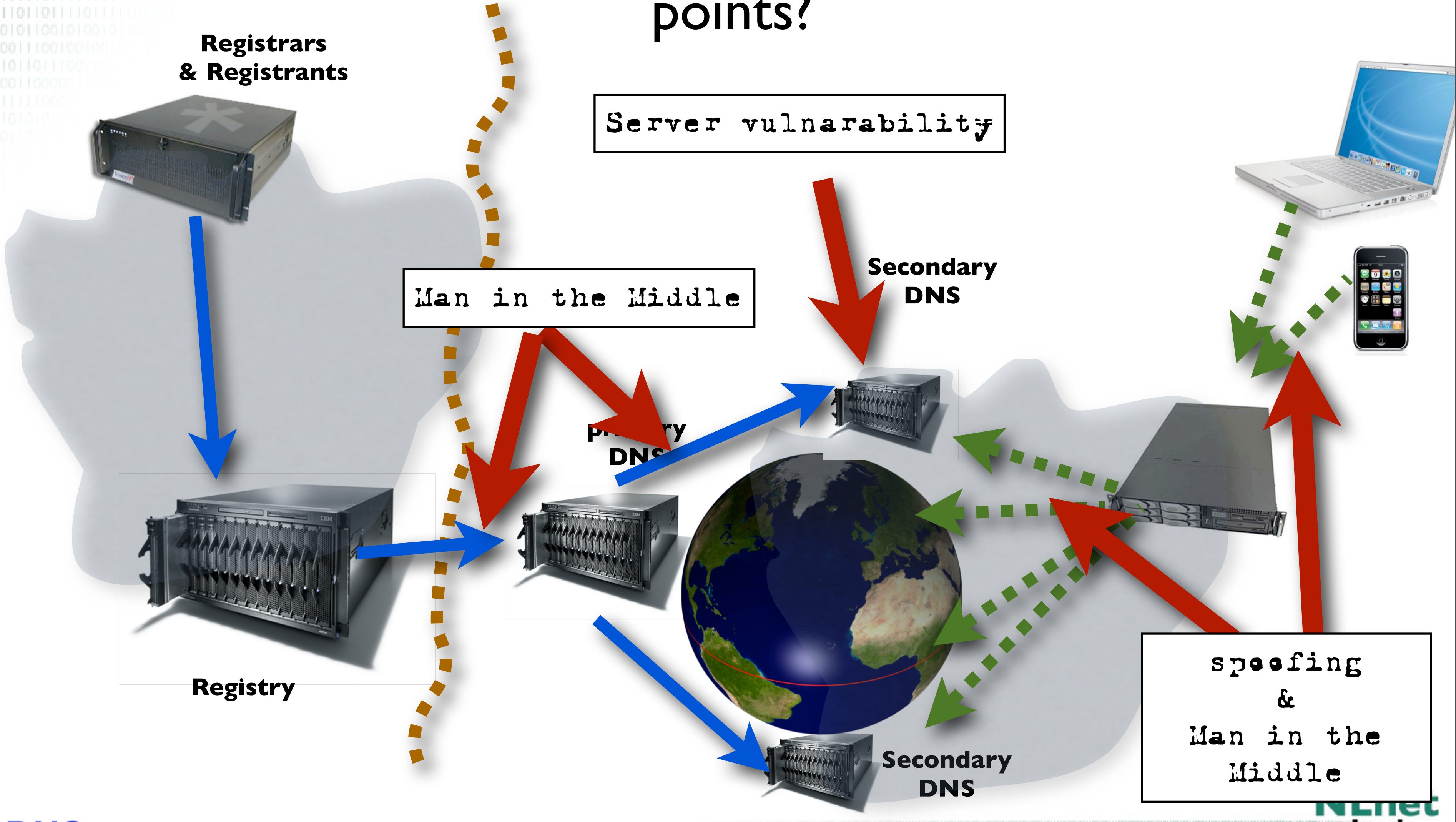
Olaf Kolkman (NLnet Labs)

DNSSEC Mechanisms

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

Data flow through the DNS

Where are the vulnerable points?



Registrars & Registrants

Server vulnerability

Man in the Middle

Secondary DNS

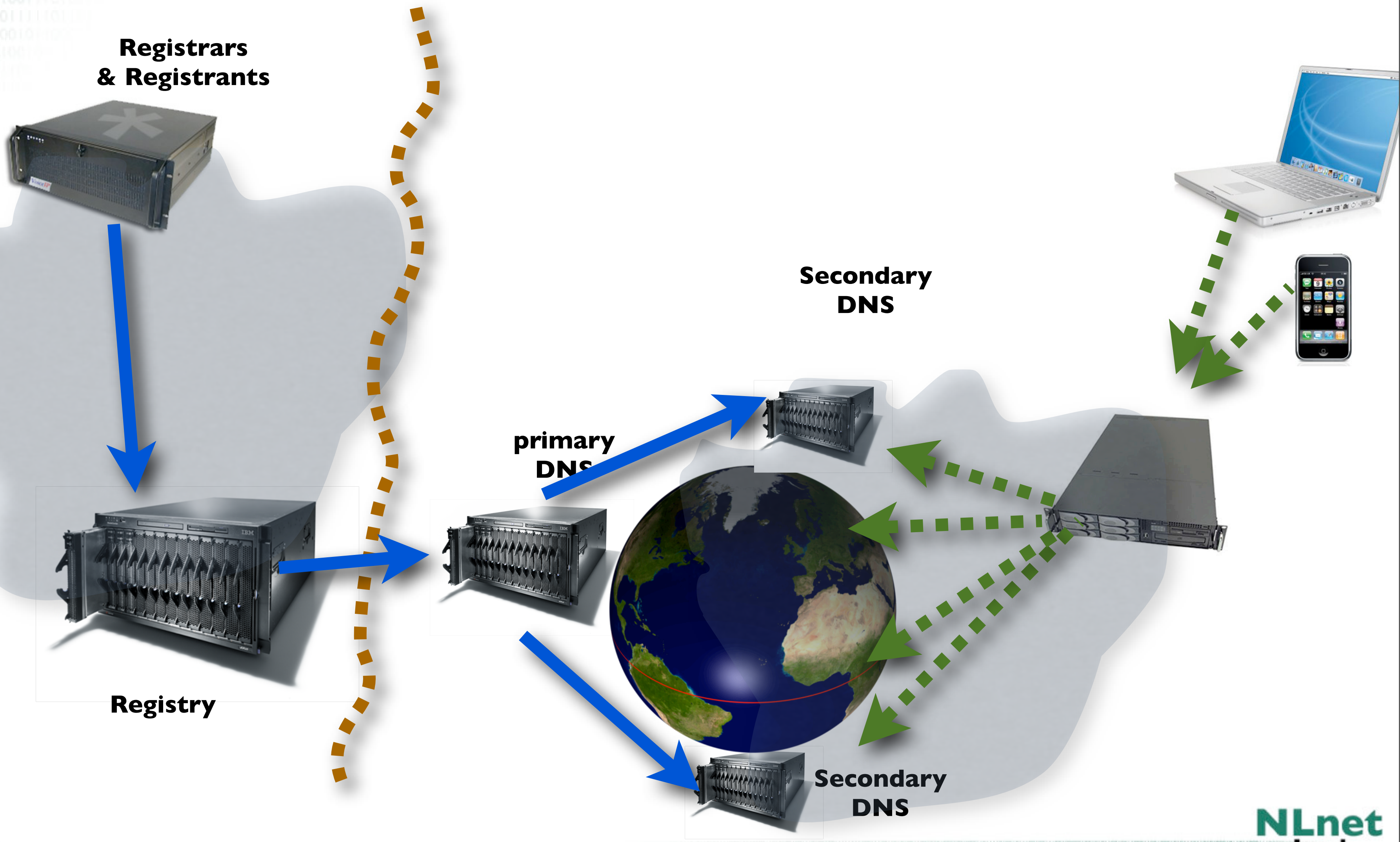
primary DNS

Registry

Secondary DNS

spoofing & Man in the Middle

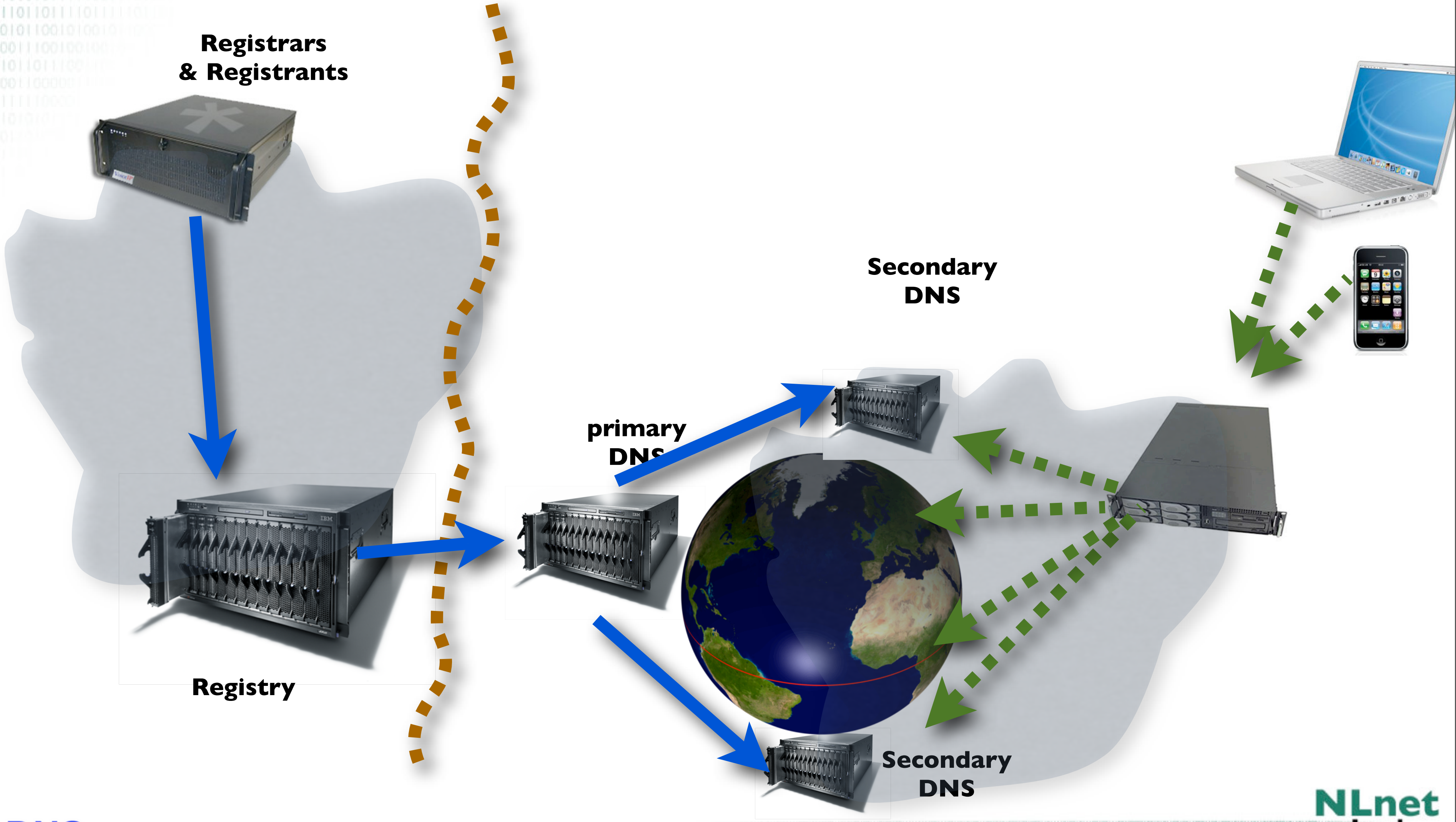
Data flow through the DNS



DNSSEC

NLnet Labs

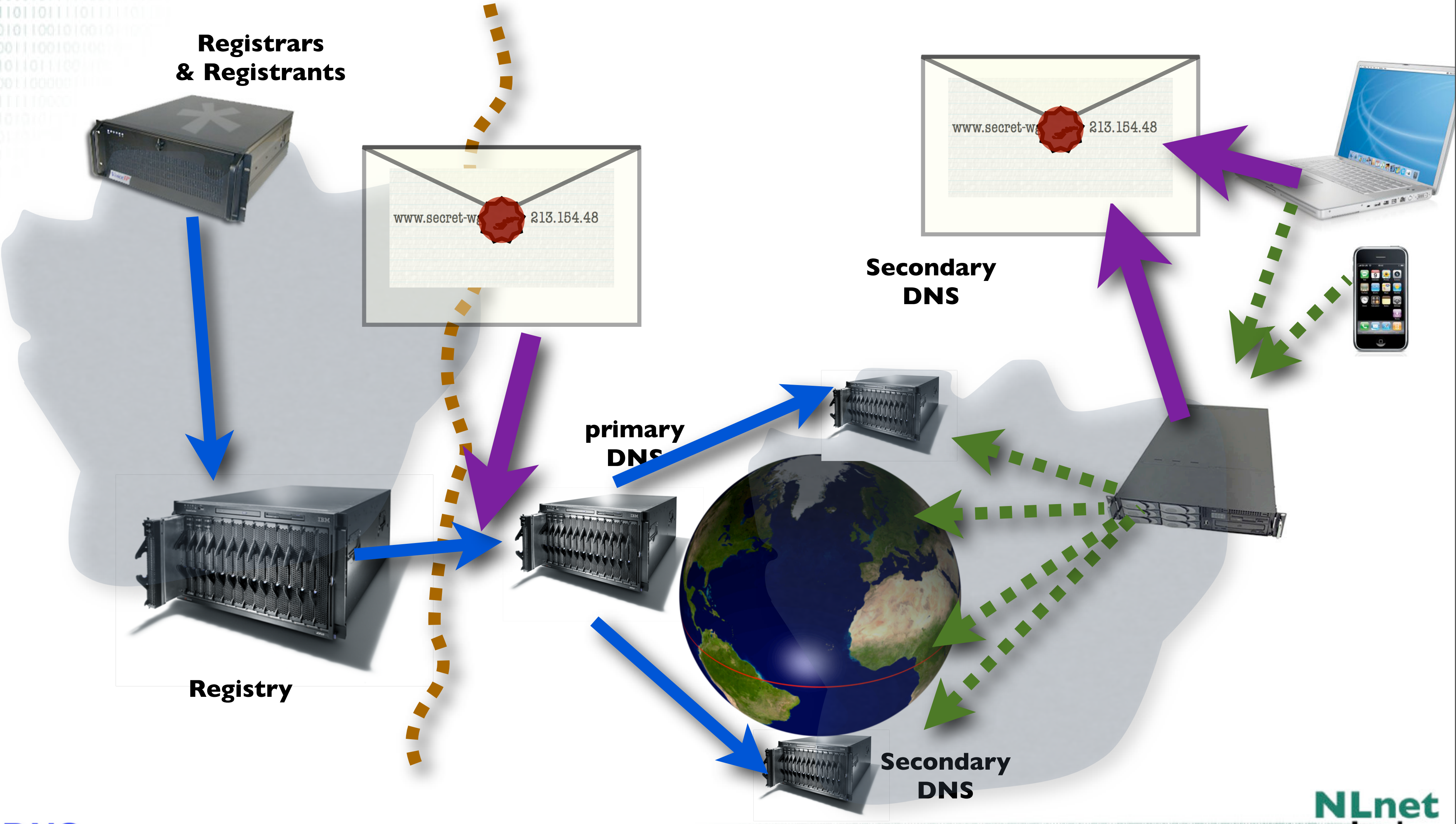
Data flow through the DNS



DNSSEC

NLnet Labs

Data flow through the DNS
End to end security



DNSSEC

NLnet Labs

The DNSSEC RRs

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)
```

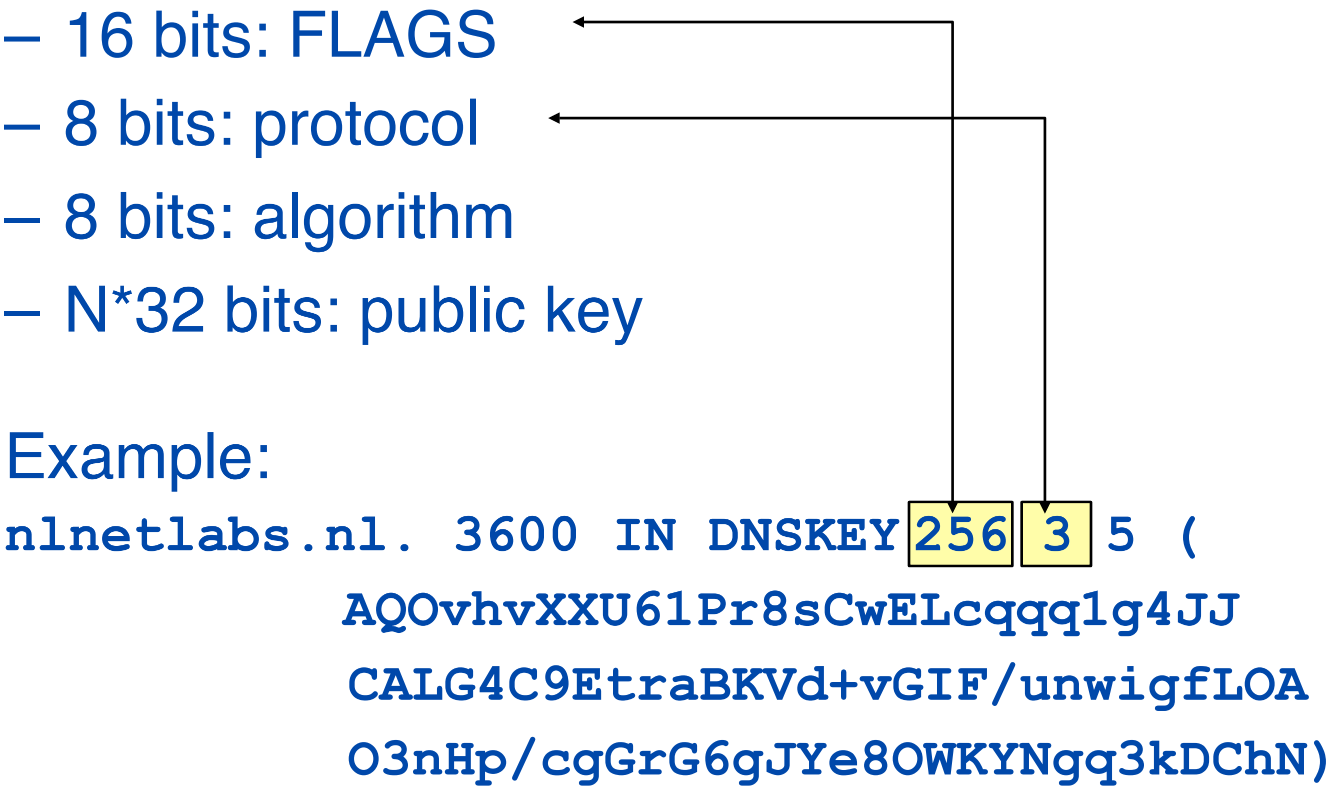
DNSKEY RDATA

- 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)
```

DNSKEY RDATA

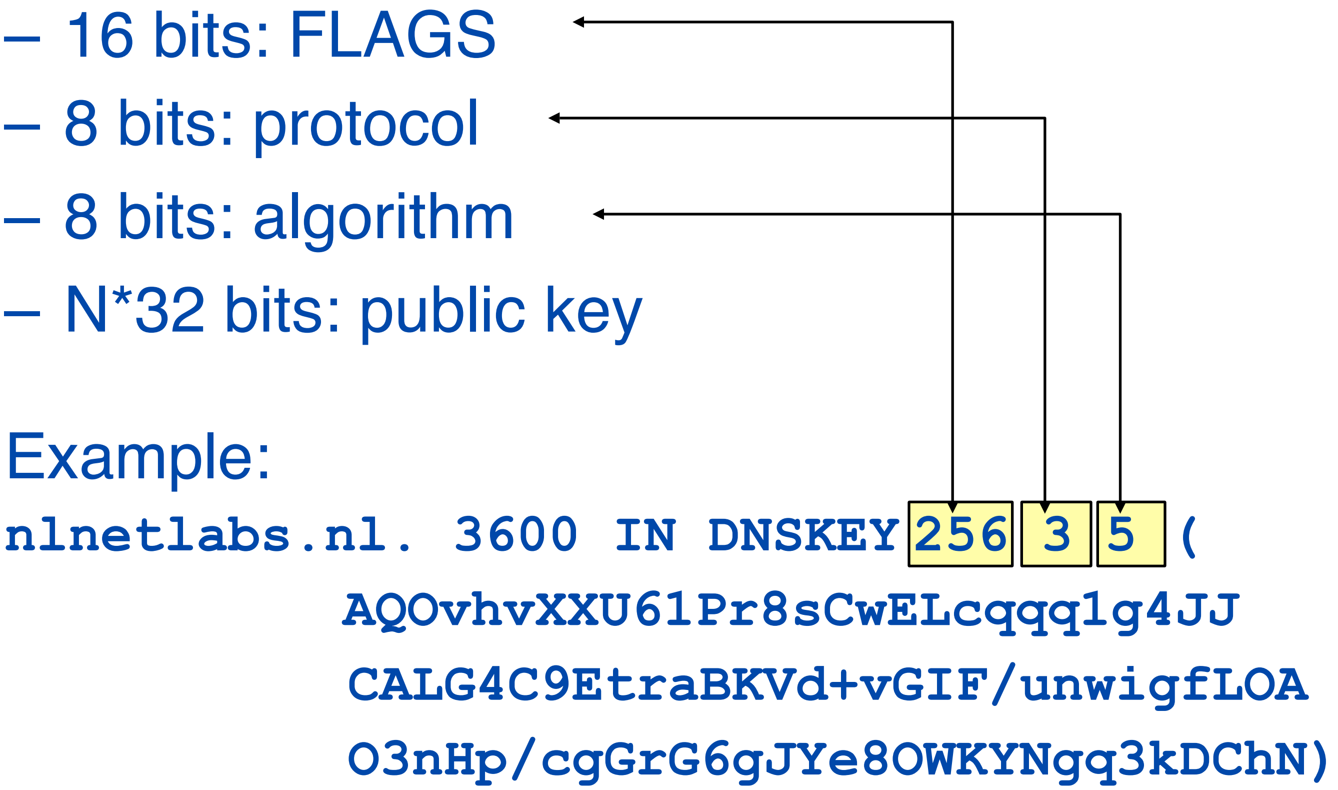


- 16 bits: FLAGS
- 8 bits: protocol
- 8 bits: algorithm
- N*32 bits: public key

Example:

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

DNSKEY RDATA

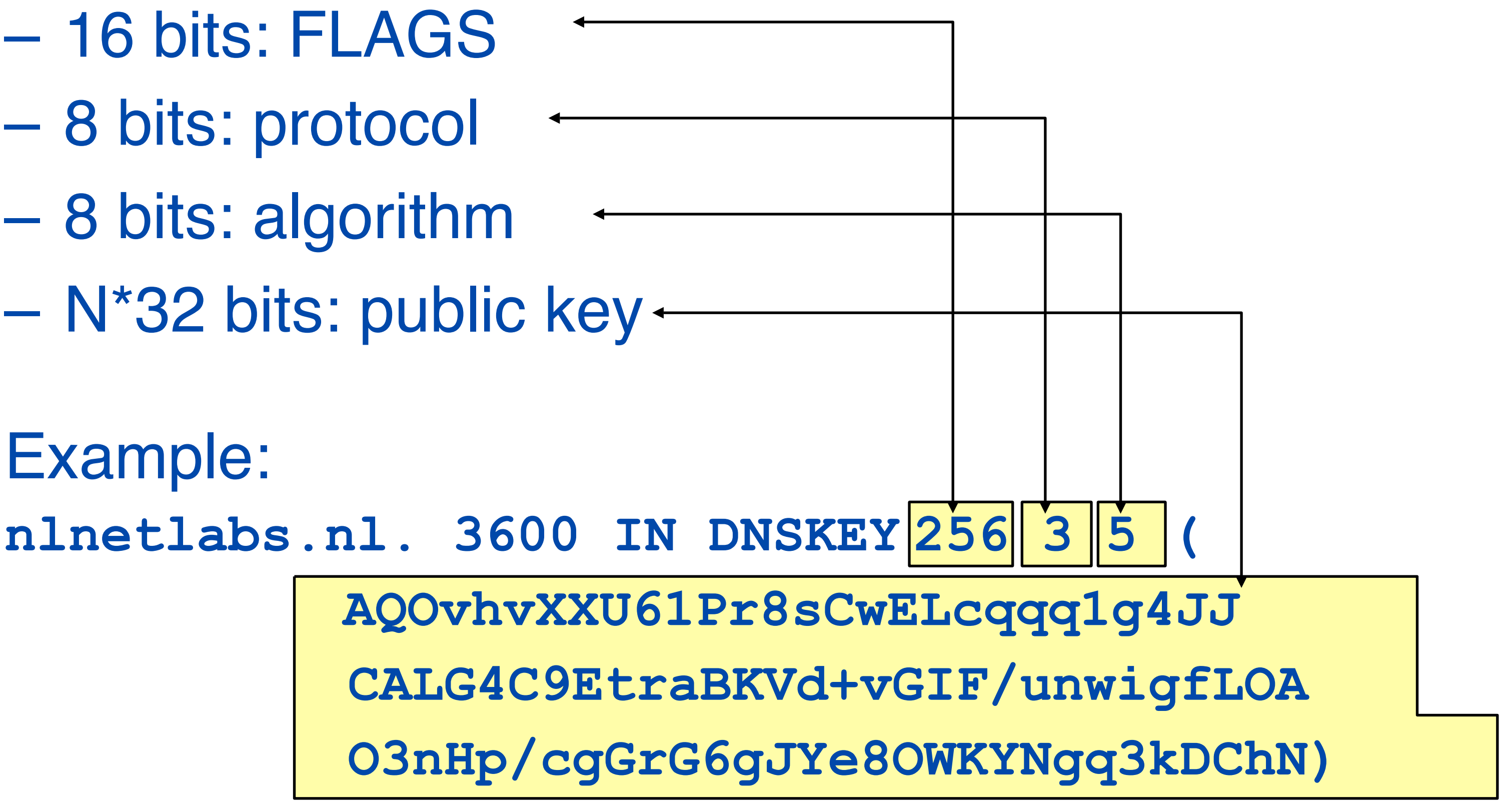


- 16 bits: FLAGS
- 8 bits: protocol
- 8 bits: algorithm
- N*32 bits: public key

Example:

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

DNSKEY RDATA

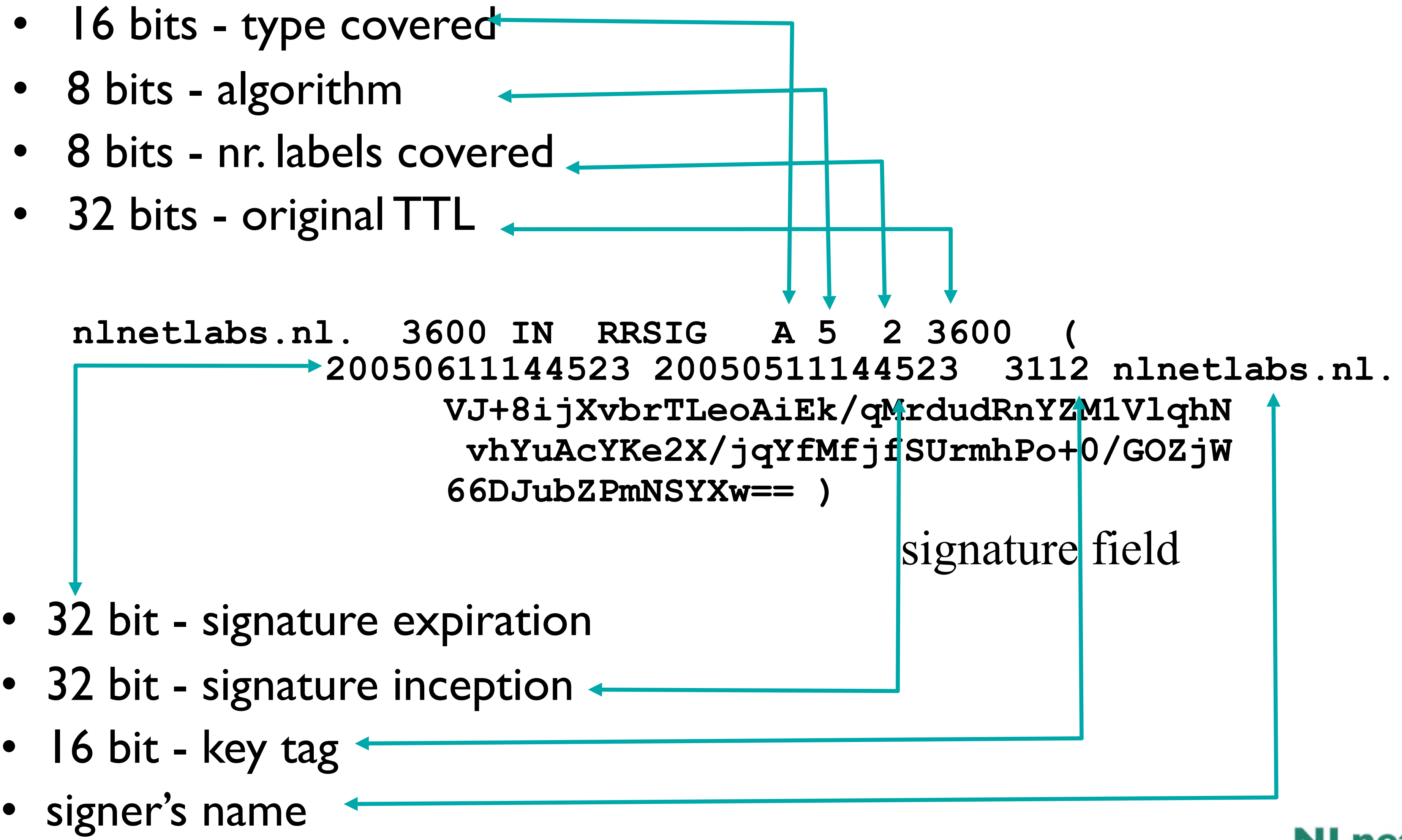


- 16 bits: FLAGS
- 8 bits: protocol
- 8 bits: algorithm
- N*32 bits: public key

Example:

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

RRSIG RDATA



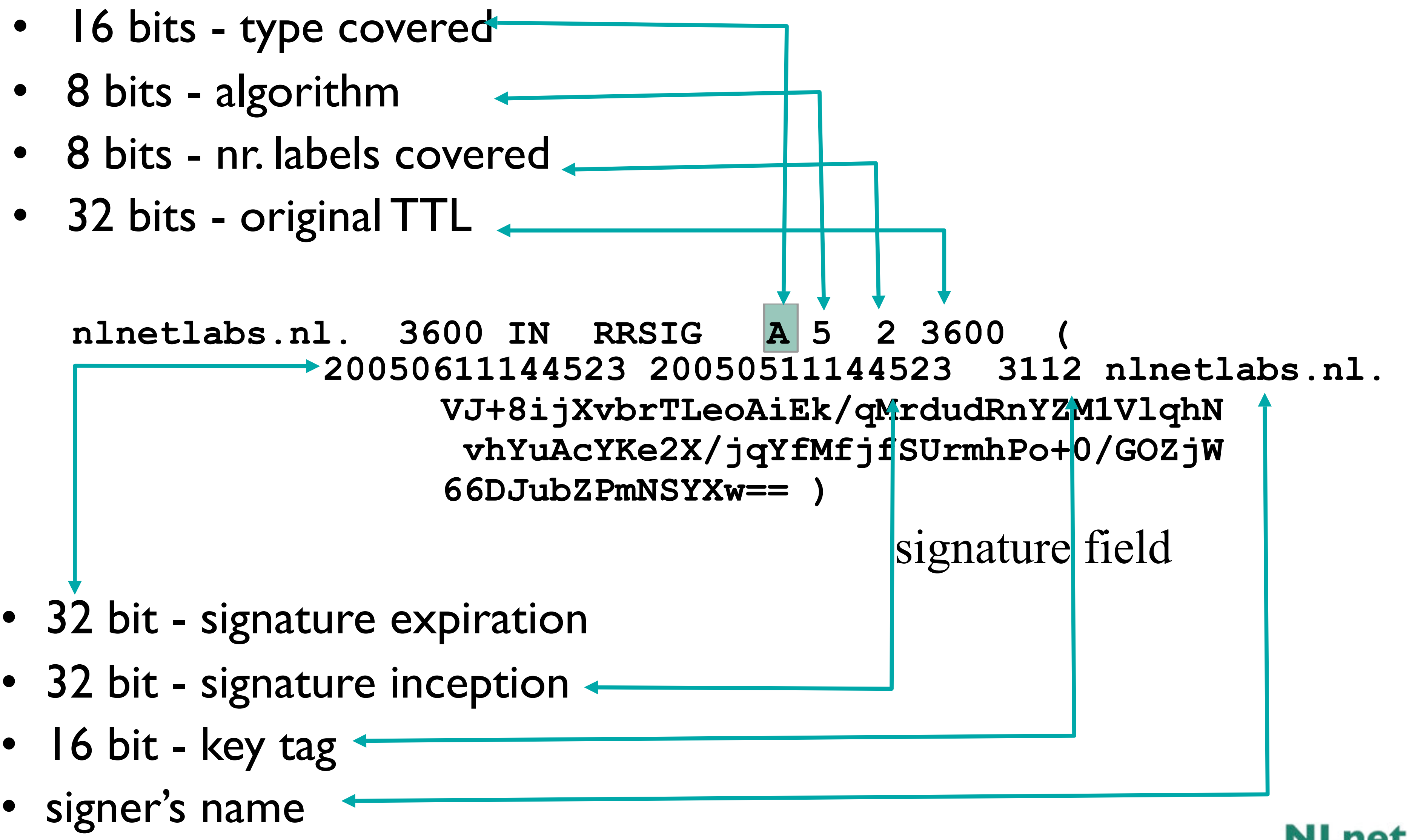
- 16 bits - type covered
- 8 bits - algorithm
- 8 bits - nr. labels covered
- 32 bits - original TTL

```
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

RRSIG RDATA



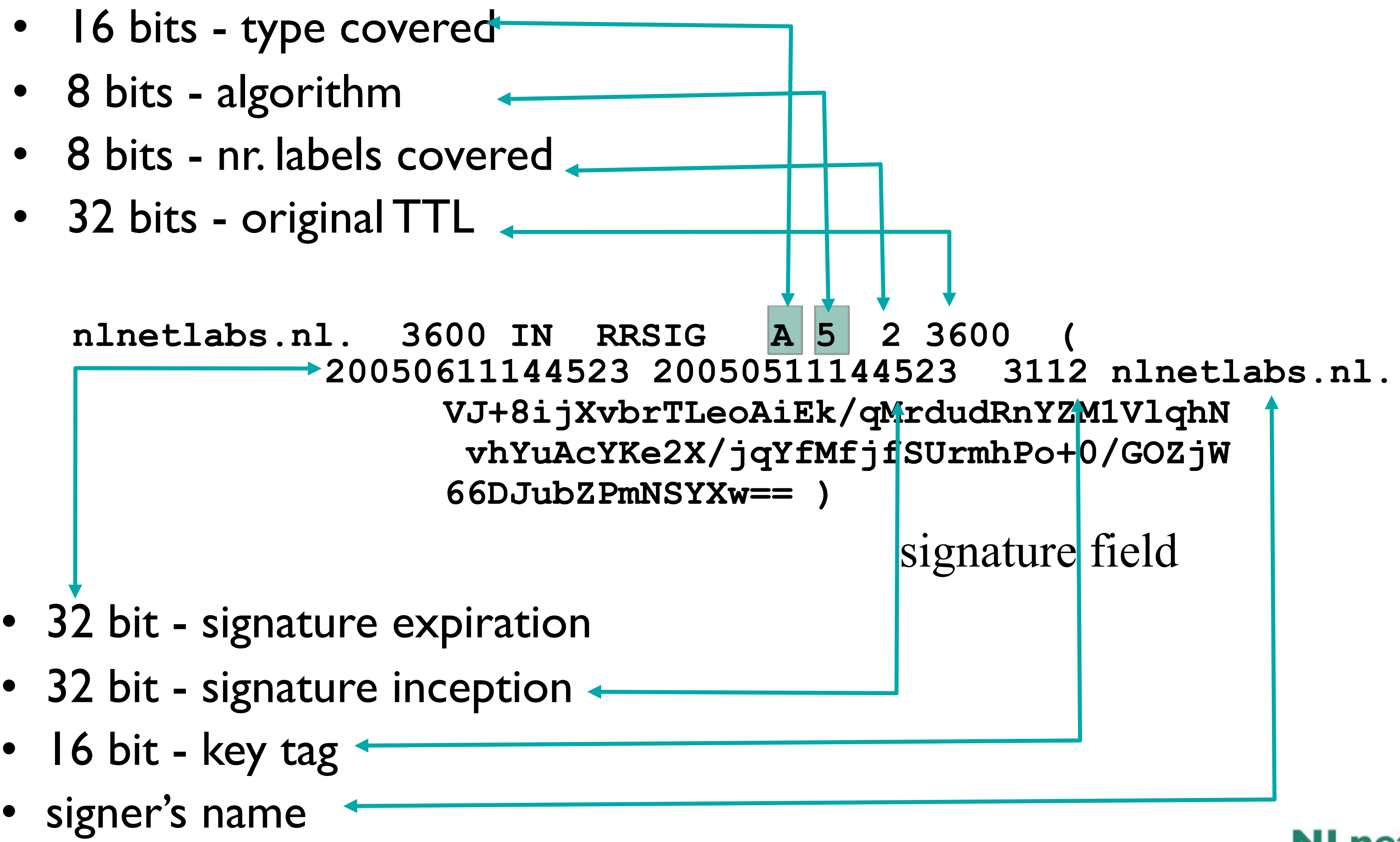
- 16 bits - type covered
- 8 bits - algorithm
- 8 bits - nr. labels covered
- 32 bits - original TTL

```
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

RRSIG RDATA



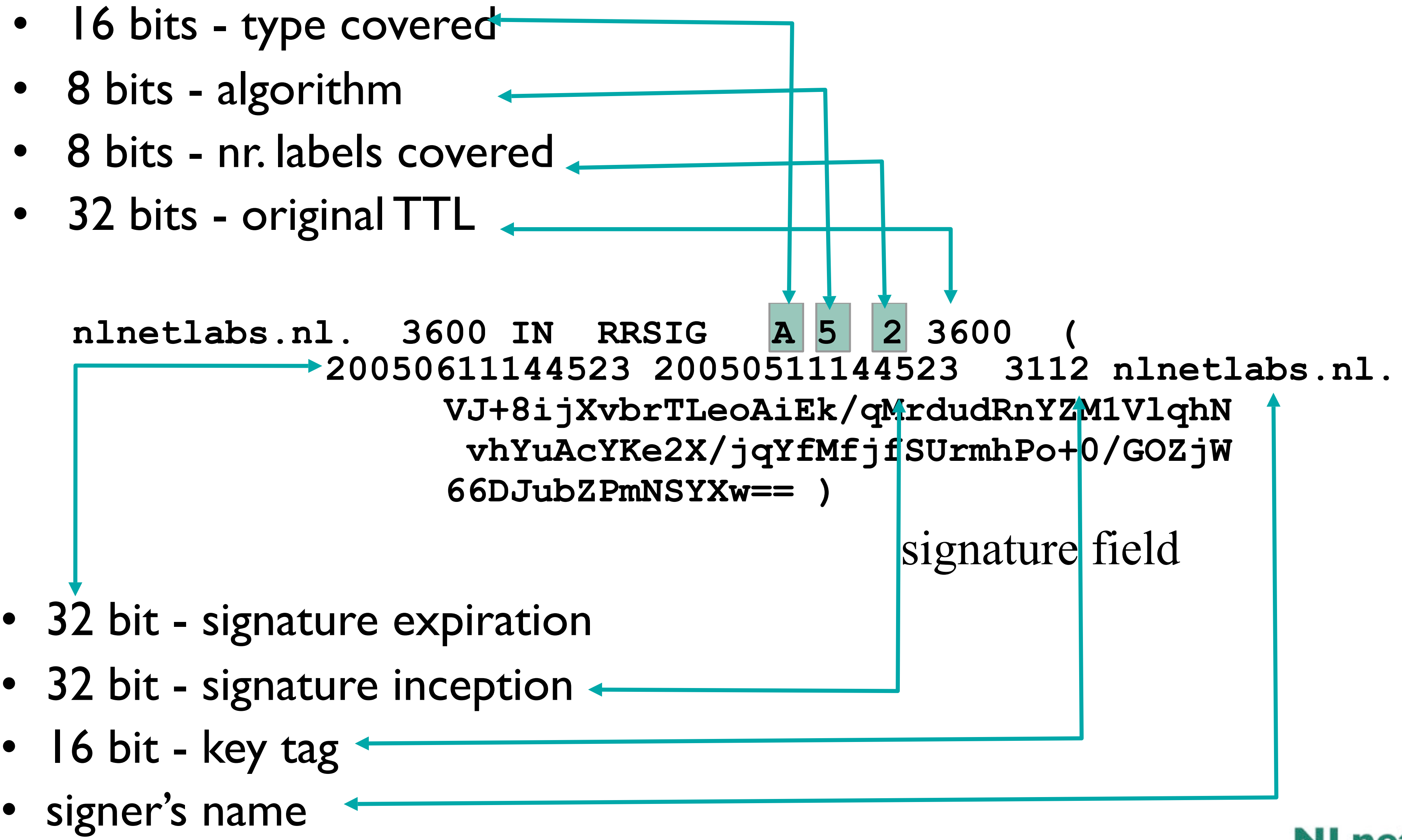
- 16 bits - type covered
- 8 bits - algorithm
- 8 bits - nr. labels covered
- 32 bits - original TTL

```
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

RRSIG RDATA



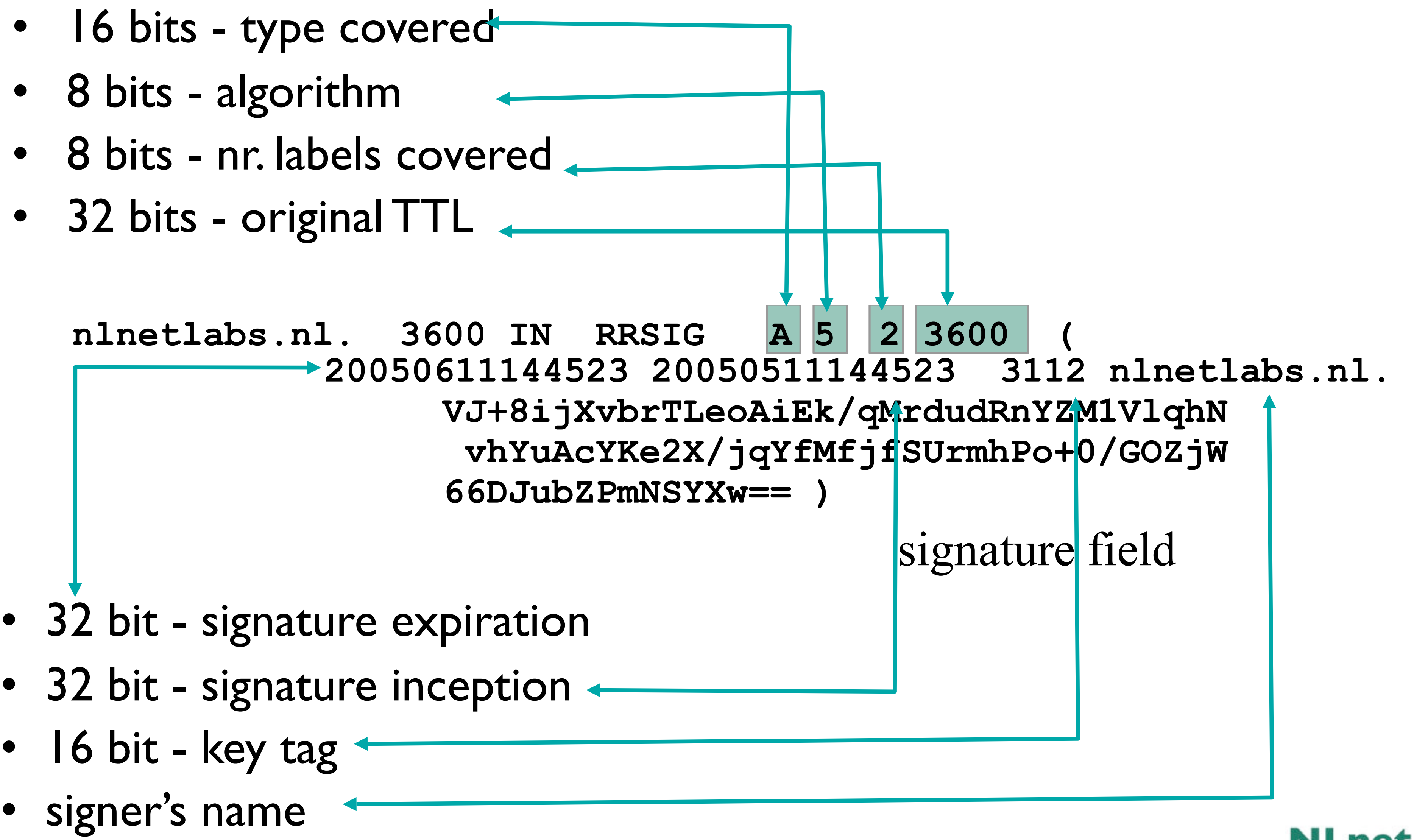
- 16 bits - type covered
- 8 bits - algorithm
- 8 bits - nr. labels covered
- 32 bits - original TTL

```
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

RRSIG RDATA



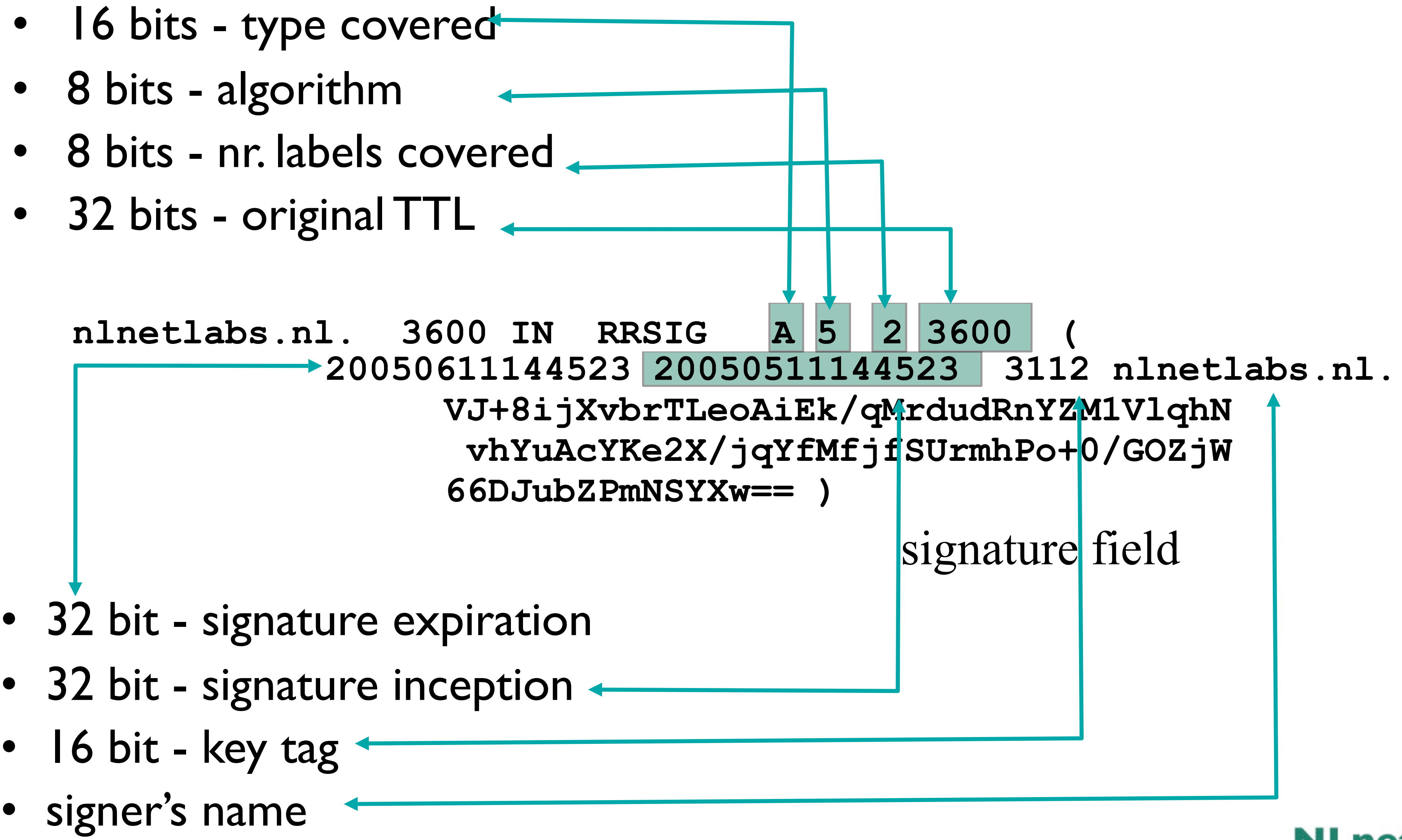
- 16 bits - type covered
- 8 bits - algorithm
- 8 bits - nr. labels covered
- 32 bits - original TTL

```
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

RRSIG RDATA



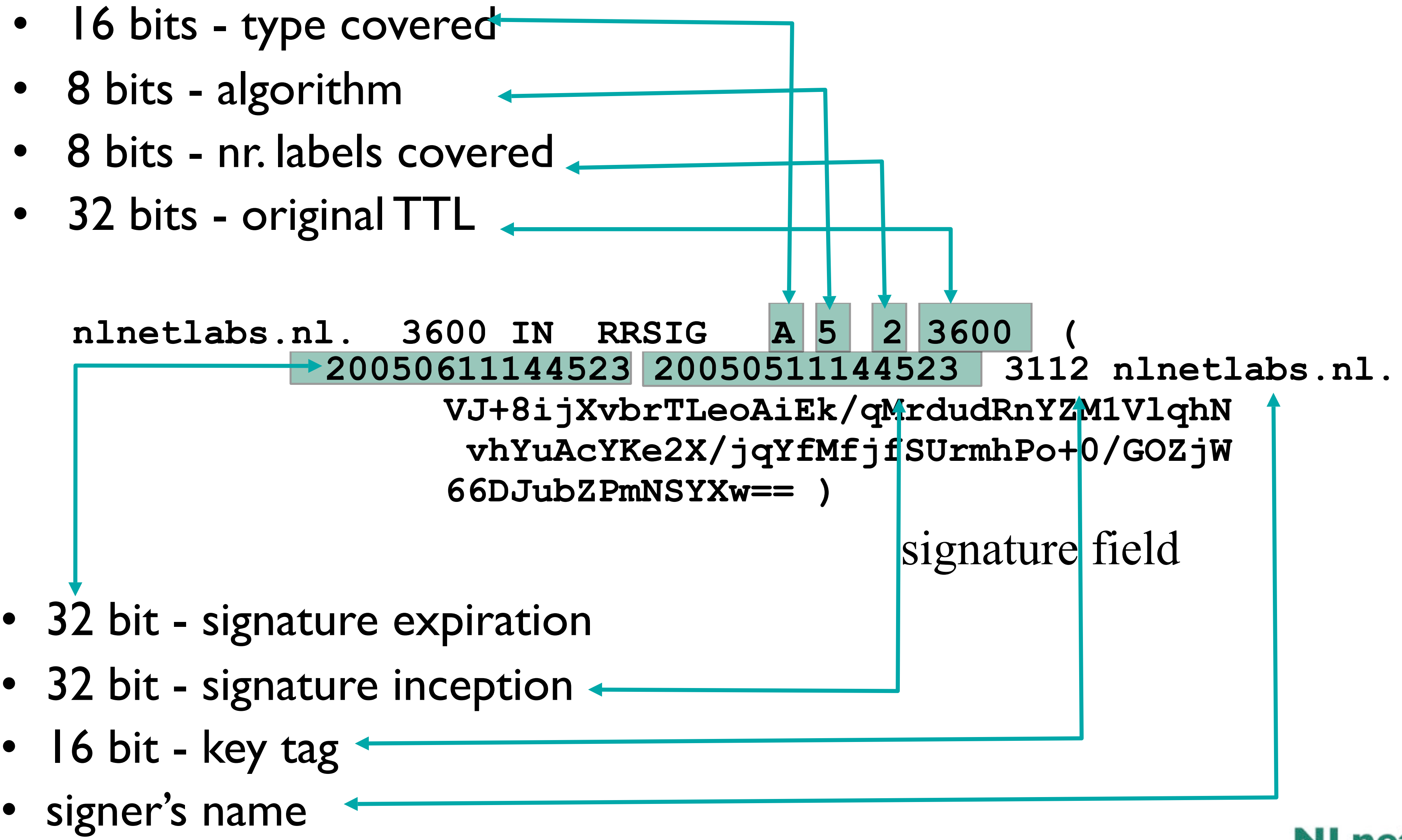
- 16 bits - type covered
- 8 bits - algorithm
- 8 bits - nr. labels covered
- 32 bits - original TTL

```
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

RRSIG RDATA



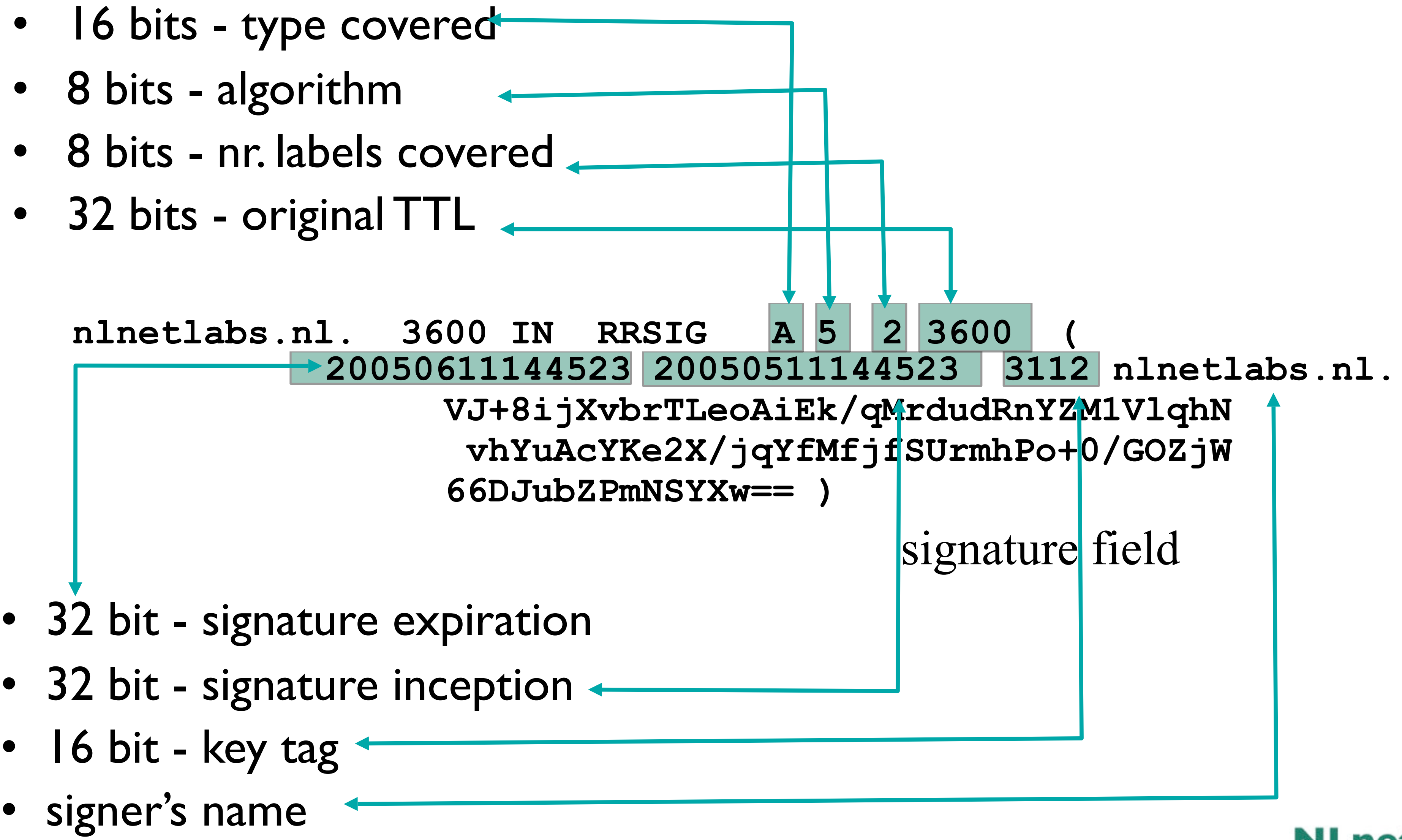
- 16 bits - type covered
- 8 bits - algorithm
- 8 bits - nr. labels covered
- 32 bits - original TTL

```
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

RRSIG RDATA



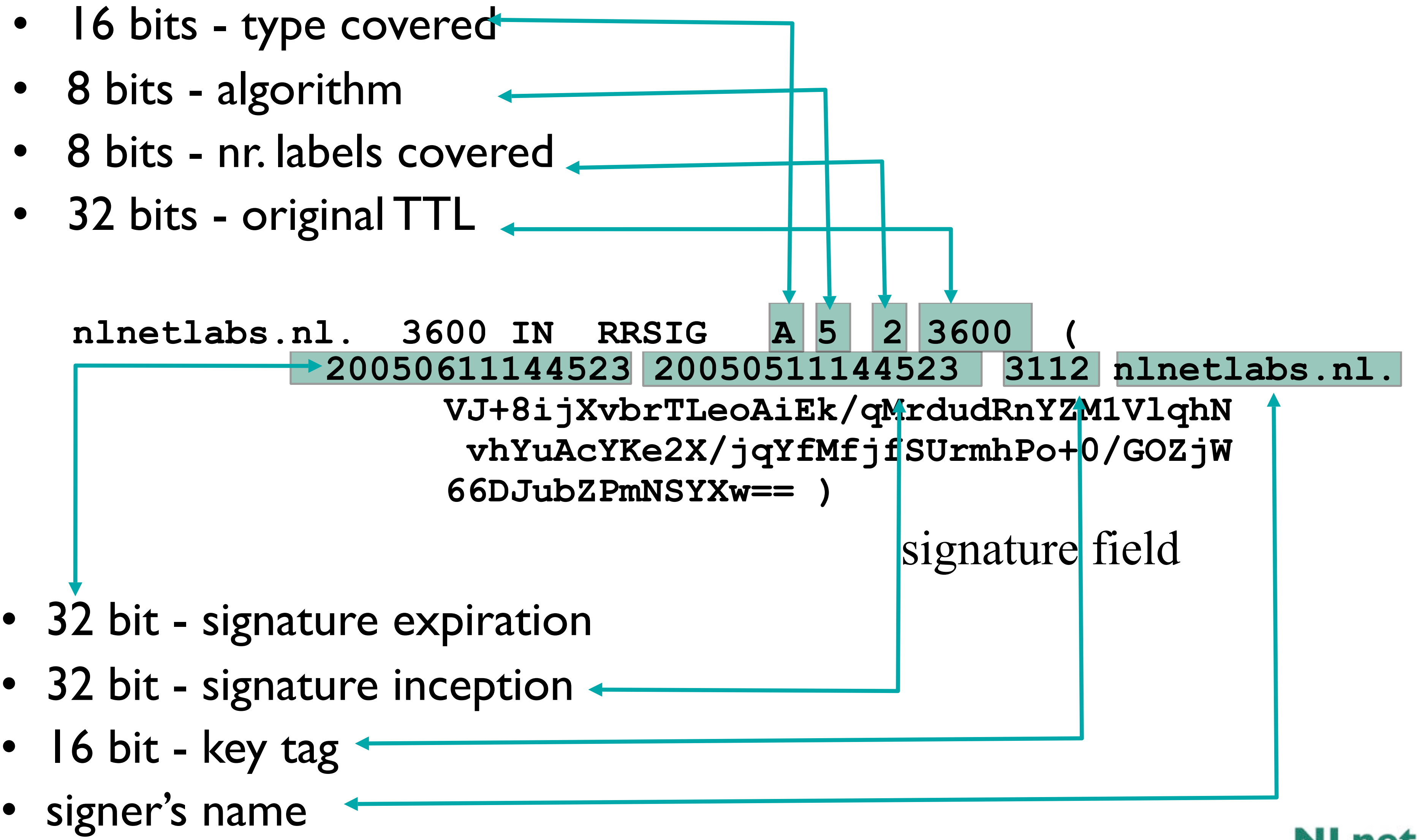
- 16 bits - type covered
- 8 bits - algorithm
- 8 bits - nr. labels covered
- 32 bits - original TTL

```
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

RRSIG RDATA



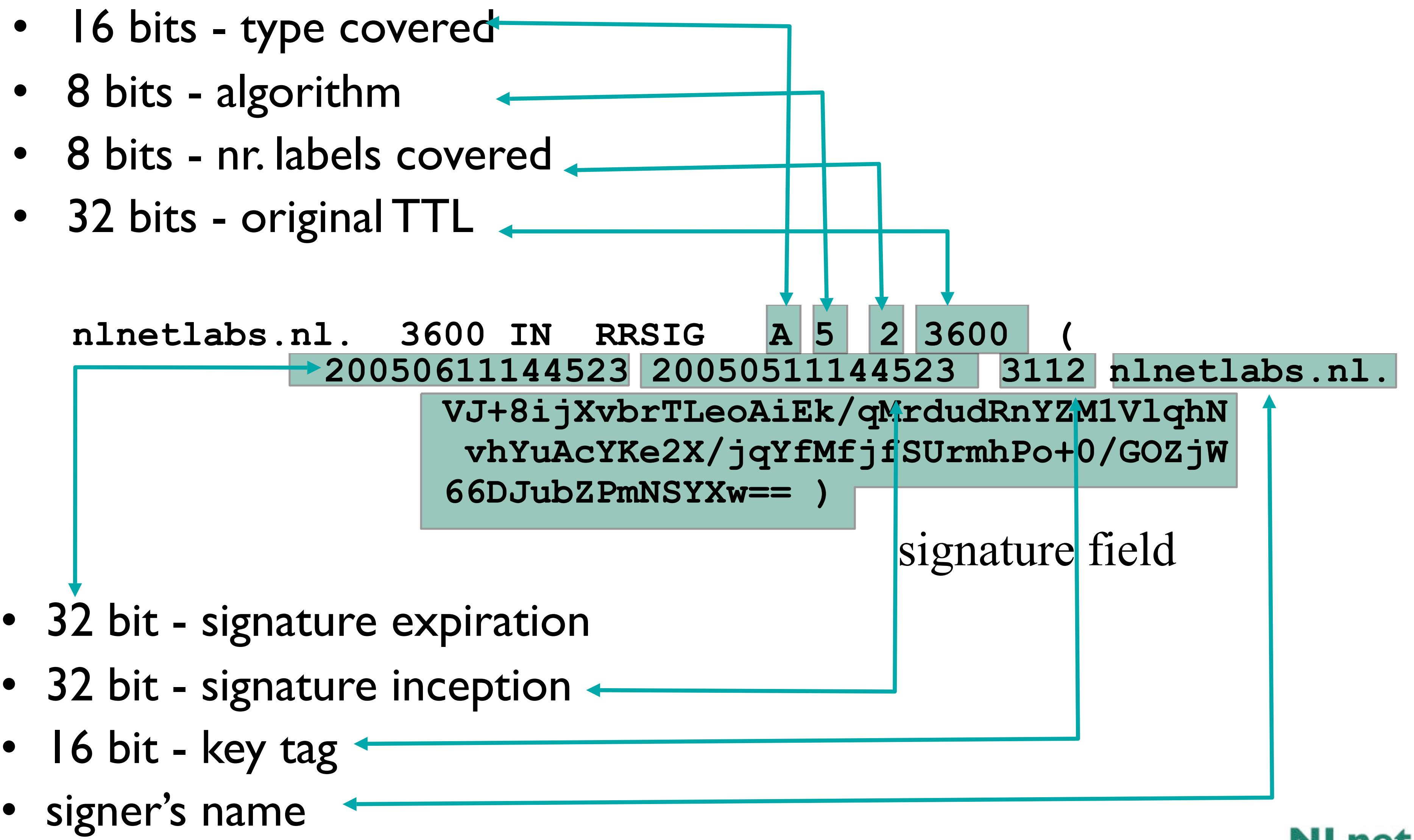
- 16 bits - type covered
- 8 bits - algorithm
- 8 bits - nr. labels covered
- 32 bits - original TTL

```
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

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 (
                                      239af98b923c023371b52
                                      1g23b92da12f42162b1a9
                                    )
```

DS RDATA

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

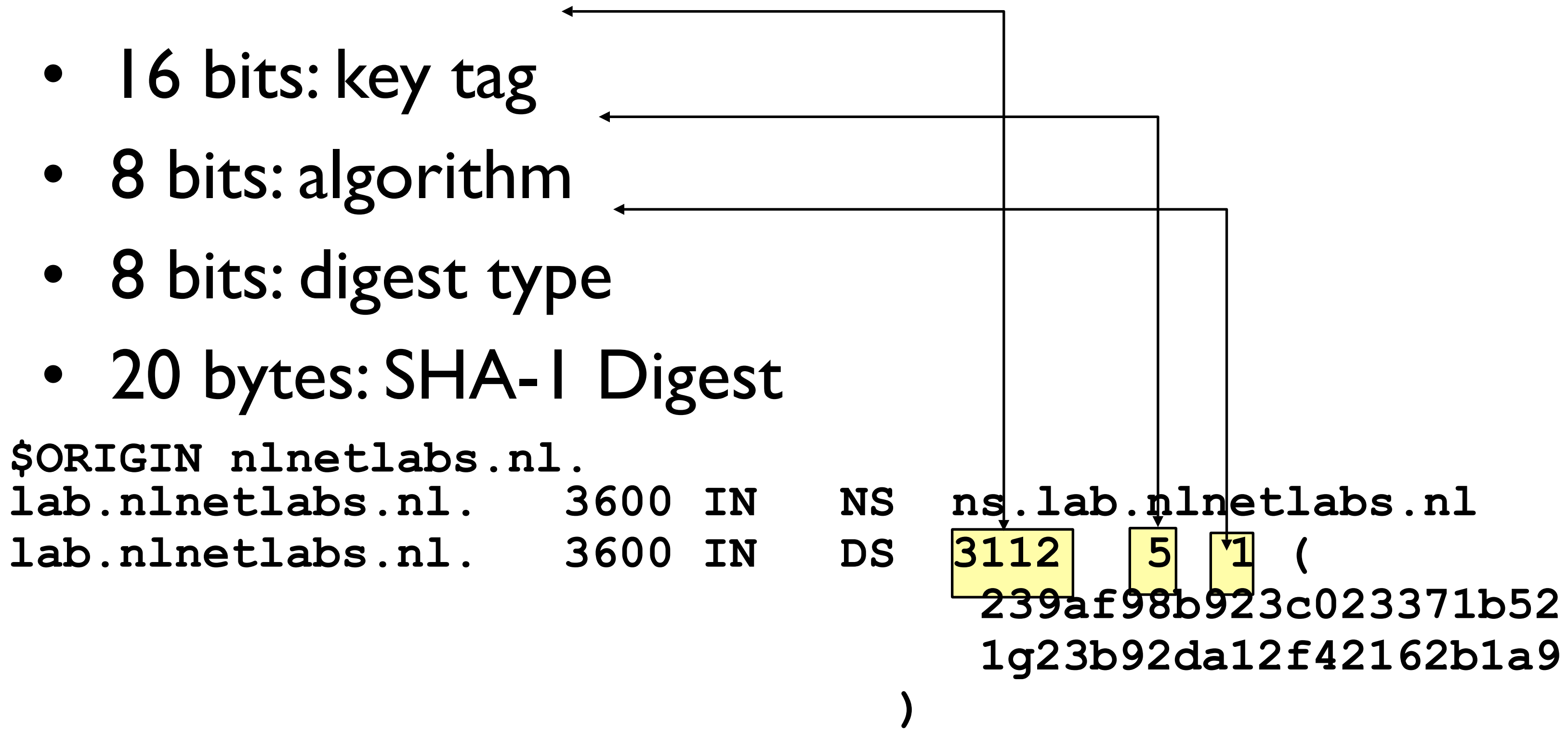
```
$ORIGIN nlnetlabs.nl.
lab.nlnetlabs.nl.   3600 IN   NS  ns.lab.nlnetlabs.nl
lab.nlnetlabs.nl.   3600 IN   DS  3112   5  1 (
                                   239af98b923c023371b52
                                   1g23b92da12f42162b1a9
                                ) 
```

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
                                   )
```

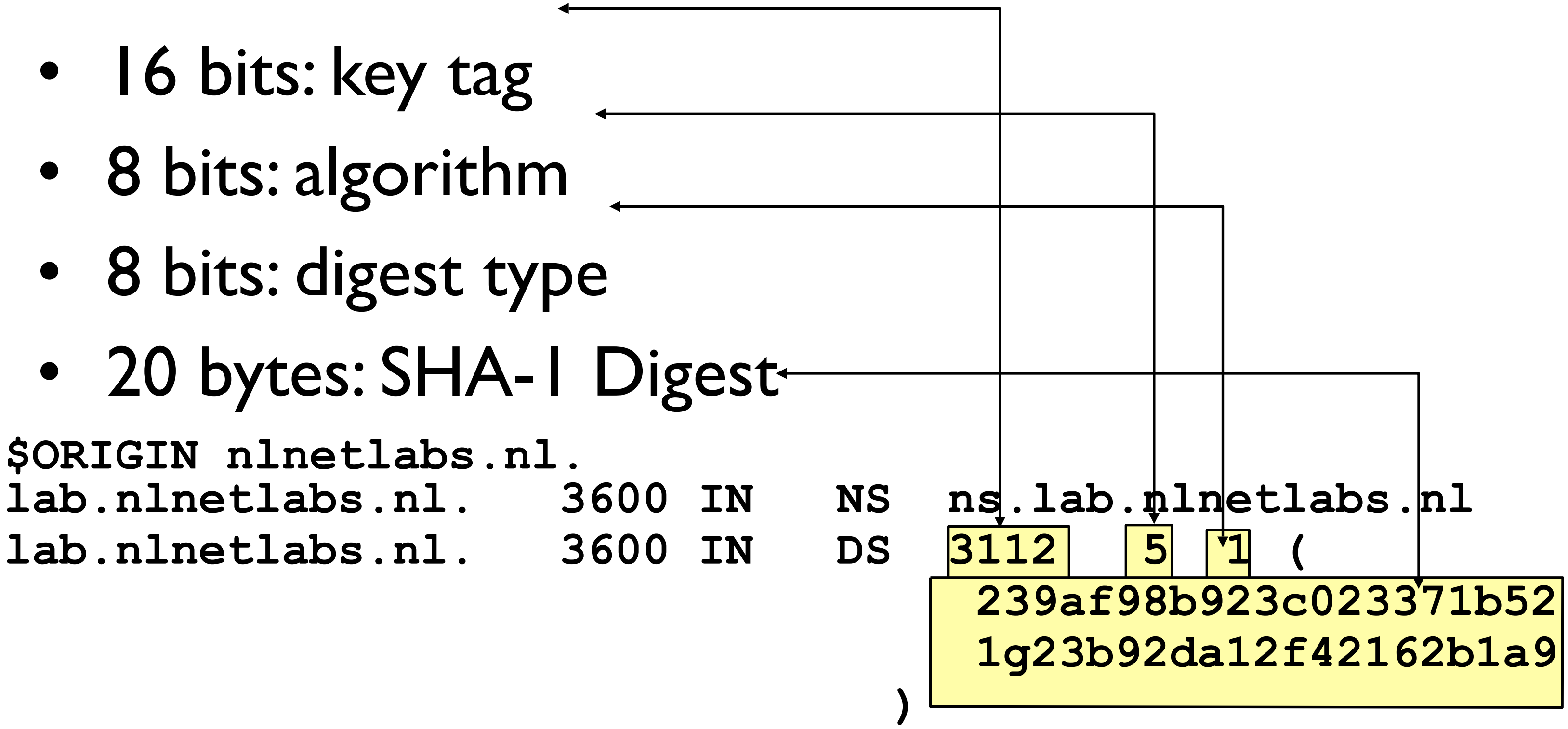
DS RDATA



- 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
                                    )
```

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
                                    )
```

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 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 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 deployment barrier

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

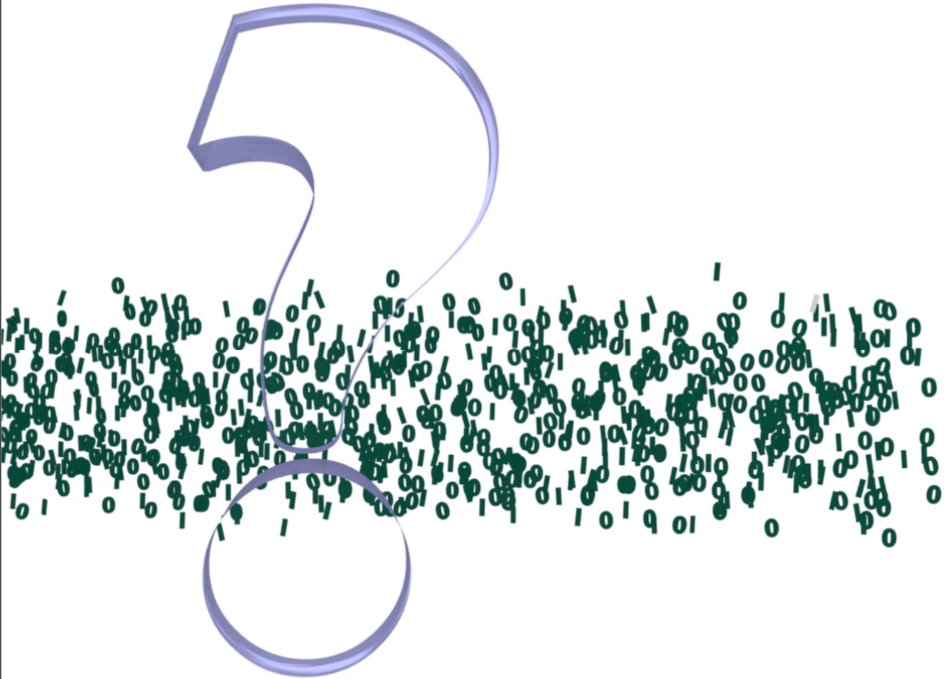
Current Developments

- SHA1 to be deprecated
  - New hash for DS records
  - Overlap, no flag day
- Introduction of SHA256

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
  - SSHFPSummary for now

Summary and questions



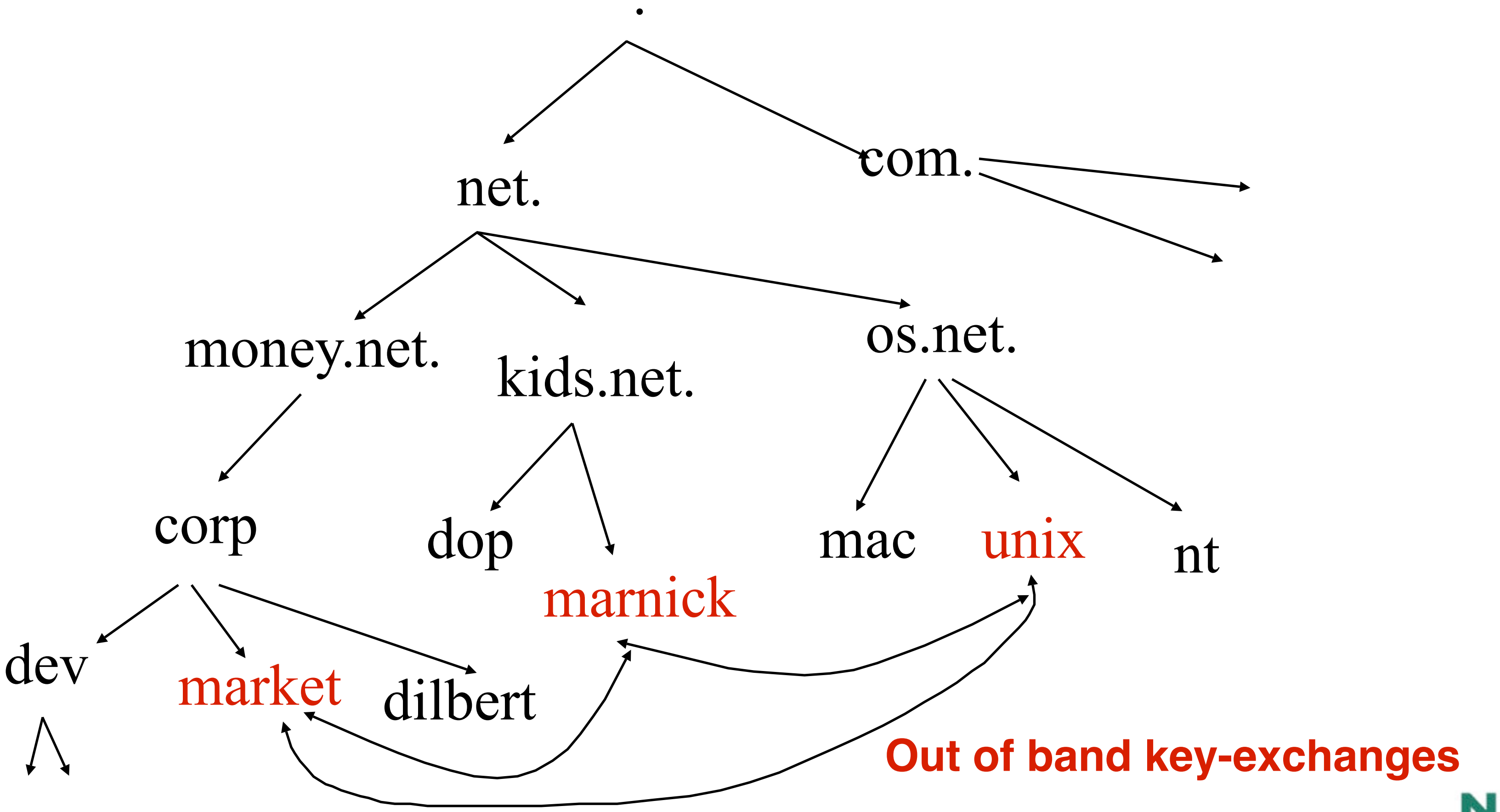
- You have seen the new RRs and learned what is their content

Delegating Signing Authority

Chains of Trust

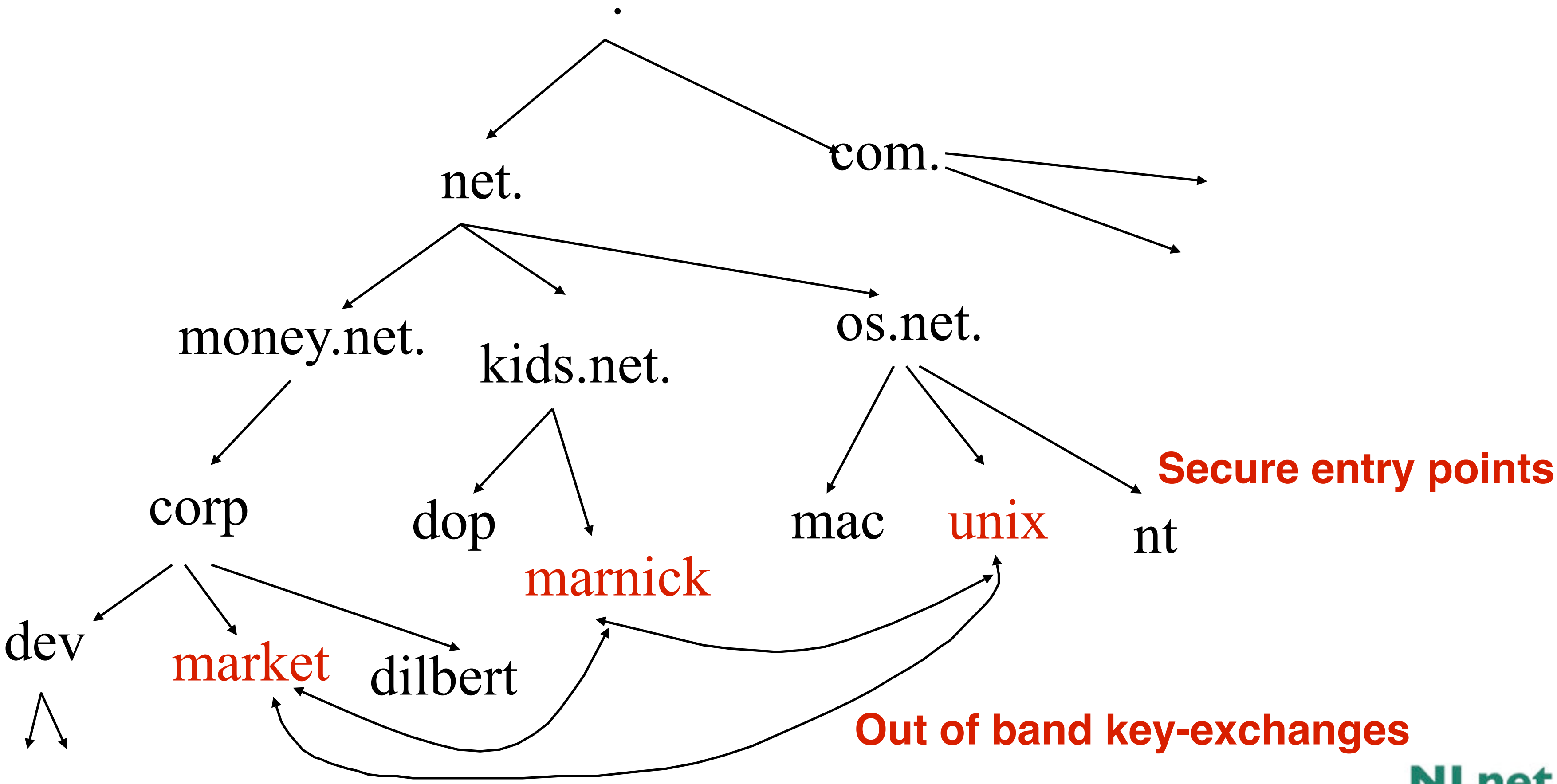
Locally Secured Zones

- Key distribution does not scale!



Locally Secured Zones

- Key distribution does not scale!



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 childs zone
    - for DNSKEY & RRSIGs
  - Verify if key can be trusted
  - Generate DS RR

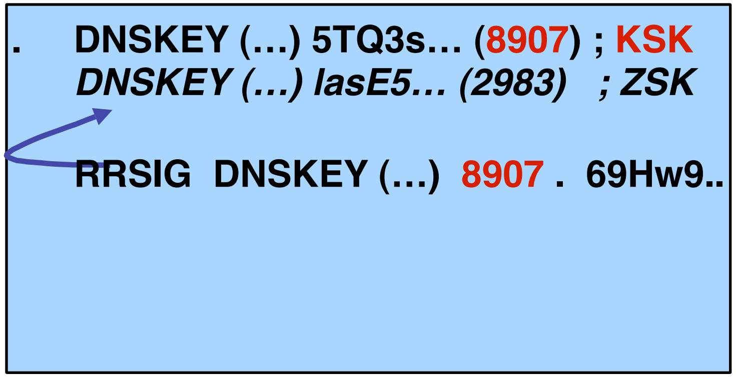
Walking the Chain of Trust

Locally configured
Trusted key: . 8907

1

2

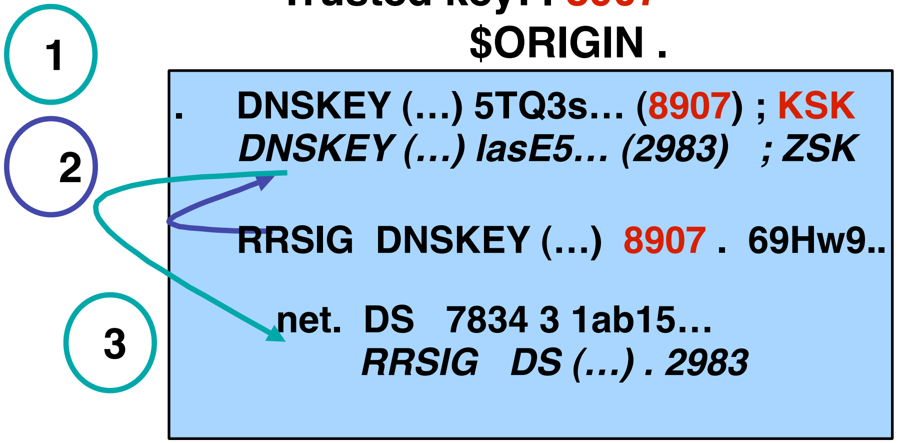
$ORIGIN .



```
.     DNSKEY (…) 5TQ3s… (8907) ; KSK
      DNSKEY (…) lasE5… (2983)   ; ZSK

      RRSIG  DNSKEY (…)  8907 .  69Hw9..
```

Walking the Chain of Trust

Locally configured
Trusted key: . 8907
$ORIGIN .

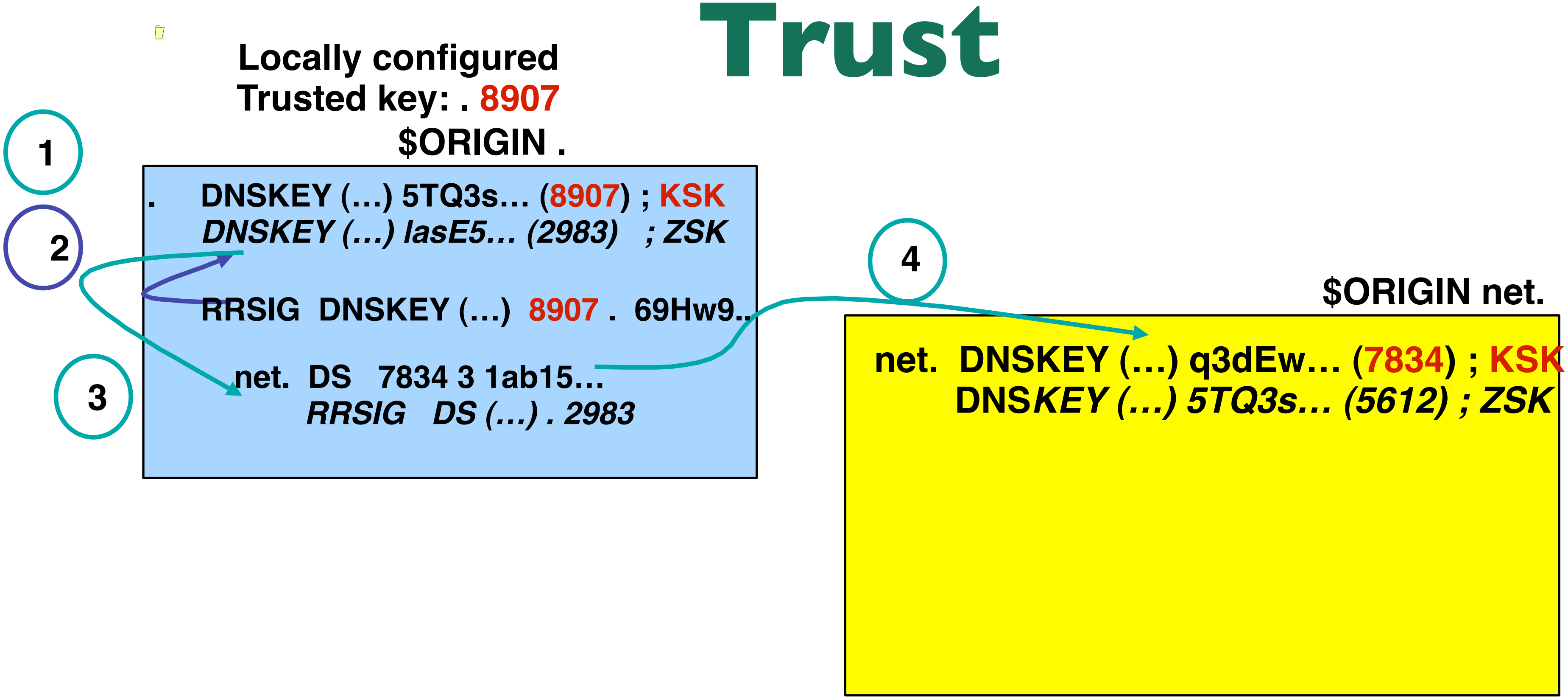


.   DNSKEY (…) 5TQ3s… (8907) ; KSK
DNSKEY (…) lasE5… (2983)   ; ZSK

RRSIG DNSKEY (…) 8907 . 69Hw9..

net. DS   7834 3 1ab15…
RRSIG   DS (…) . 2983

Walking the Chain of Trust



Locally configured
Trusted key: . 8907

$ORIGIN .

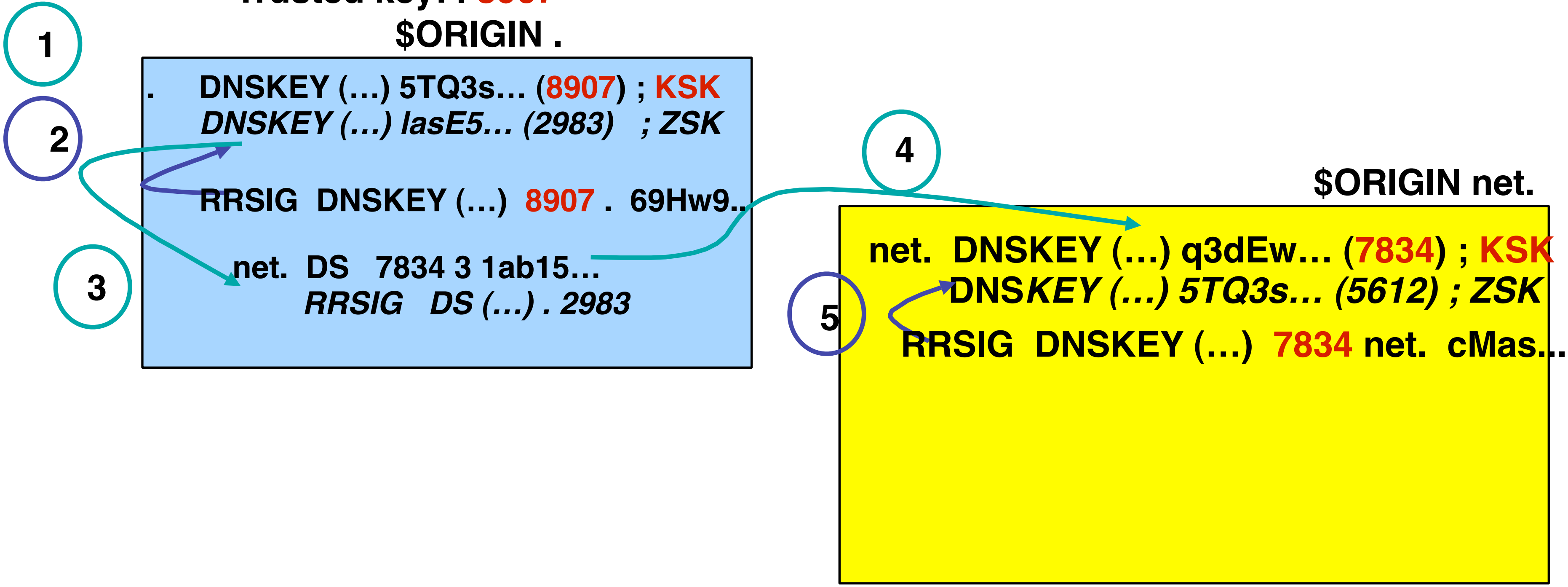
```
.    DNSKEY (…) 5TQ3s… (8907) ; KSK
     DNSKEY (…) lasE5… (2983)   ; ZSK

     RRSIG  DNSKEY (…)  8907 .  69Hw9..

       net.  DS   7834 3 1ab15…
             RRSIG   DS (…) . 2983
```

$ORIGIN net.

```
net.  DNSKEY (…) q3dEw… (7834) ; KSK
      DNSKEY (…) 5TQ3s… (5612) ; ZSK
```

Walking the Chain of Trust

Locally configured
Trusted key: . 8907



$ORIGIN .

. DNSKEY (…) 5TQ3s… (8907) ; KSK
DNSKEY (…) lasE5… (2983) ; ZSK

RRSIG DNSKEY (…) 8907 . 69Hw9..

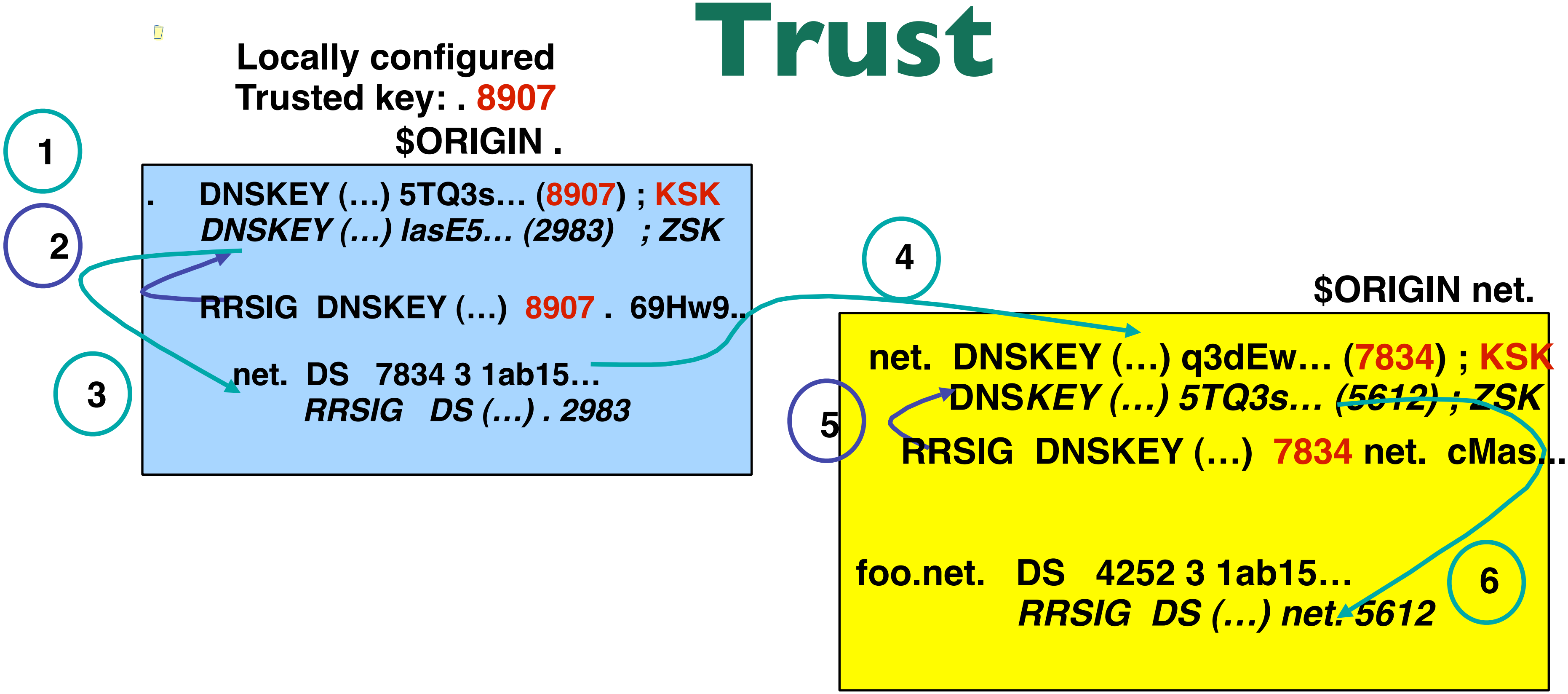
net. DS 7834 3 1ab15…
RRSIG DS (…) . 2983

$ORIGIN net.

net. DNSKEY (…) q3dEw… (7834) ; KSK
DNS*KEY (…) 5TQ3s… (5612) ; ZSK*

RRSIG DNSKEY (…) 7834 net. cMas...

Walking the Chain of Trust



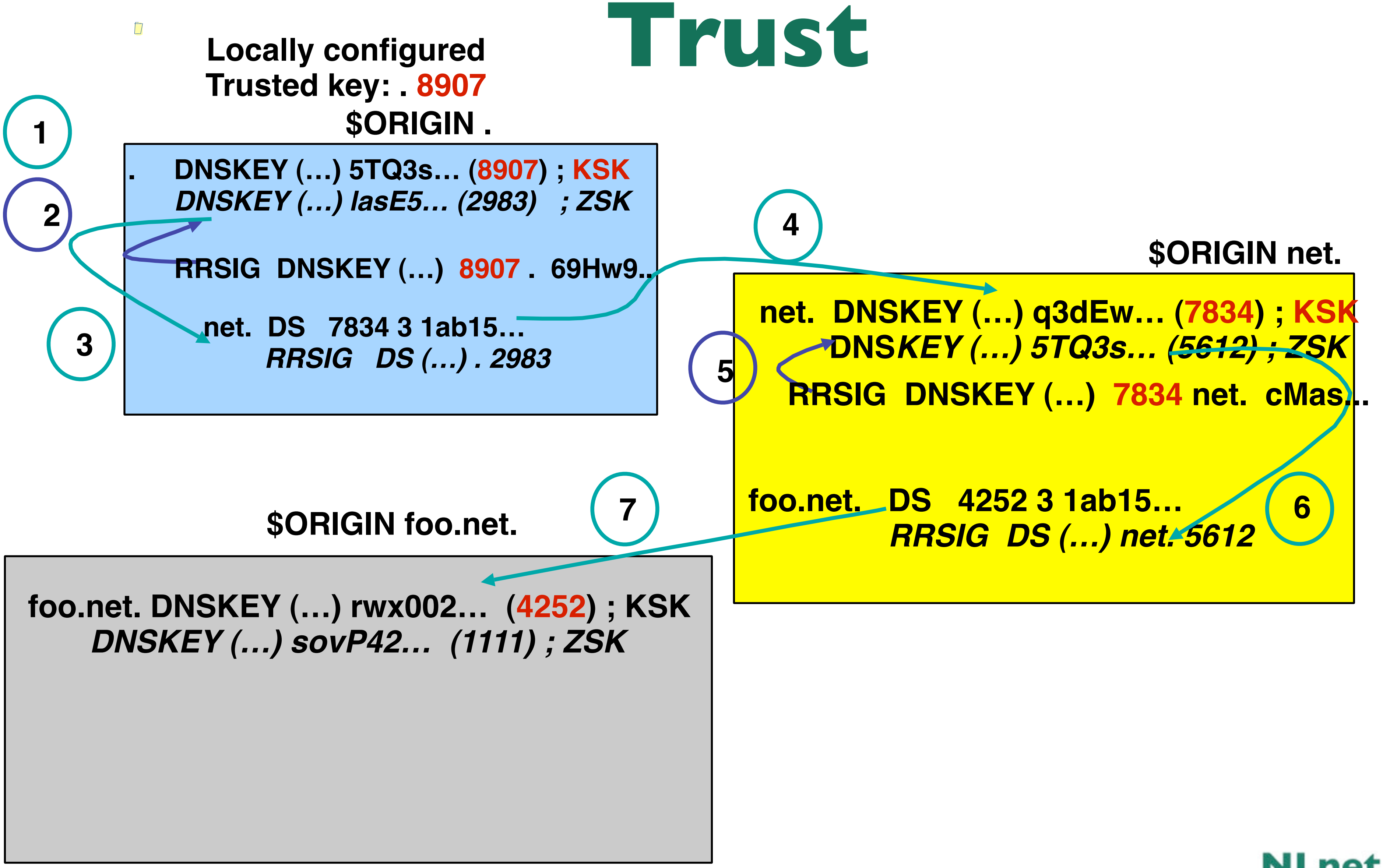
Locally configured
Trusted key: . 8907
$ORIGIN .

```
.    DNSKEY (…) 5TQ3s… (8907) ; KSK
     DNSKEY (…) lasE5… (2983)   ; ZSK

     RRSIG  DNSKEY (…)  8907 .  69Hw9..

       net.  DS   7834 3 1ab15…
             RRSIG   DS (…) . 2983
```

$ORIGIN net.

```
net.  DNSKEY (…) q3dEw… (7834) ; KSK
      DNSKEY (…) 5TQ3s… (5612) ; ZSK
  RRSIG  DNSKEY (…)  7834 net.  cMas...

foo.net.   DS   4252 3 1ab15…
           RRSIG  DS (…) net. 5612
```

Walking the Chain of Trust



Locally configured
Trusted key: . 8907

$ORIGIN .

```
.    DNSKEY (…) 5TQ3s… (8907) ; KSK
     DNSKEY (…) lasE5… (2983)   ; ZSK

     RRSIG  DNSKEY (…)  8907 .  69Hw9..

        net.  DS   7834 3 1ab15…
              RRSIG   DS (…) . 2983
```

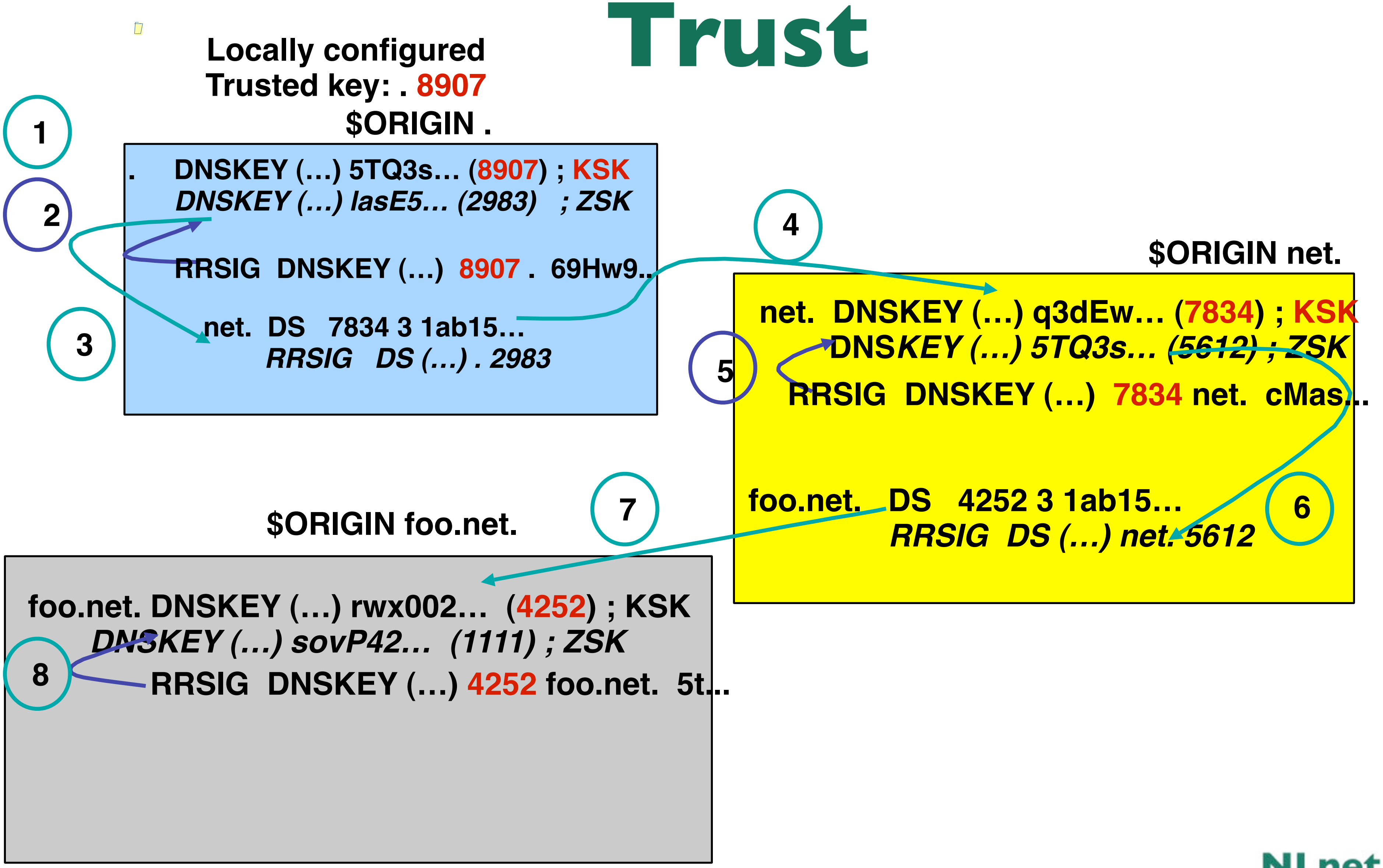
$ORIGIN net.

```
net.  DNSKEY (…) q3dEw… (7834) ; KSK
      DNSKEY (…) 5TQ3s… (5612) ; ZSK
   RRSIG  DNSKEY (…)  7834 net.  cMas...

foo.net.  DS   4252 3 1ab15…
          RRSIG  DS (…) net. 5612
```

$ORIGIN foo.net.

```
foo.net. DNSKEY (…) rwx002…  (4252) ; KSK
         DNSKEY (…) sovP42…  (1111) ; ZSK
```

Walking the Chain of Trust



Locally configured
Trusted key: . 8907

$ORIGIN .

```
.    DNSKEY (…) 5TQ3s… (8907) ; KSK
     DNSKEY (…) lasE5… (2983)   ; ZSK

     RRSIG  DNSKEY (…)  8907 .  69Hw9..

       net.  DS   7834 3 1ab15…
             RRSIG   DS (…) . 2983
```

$ORIGIN net.

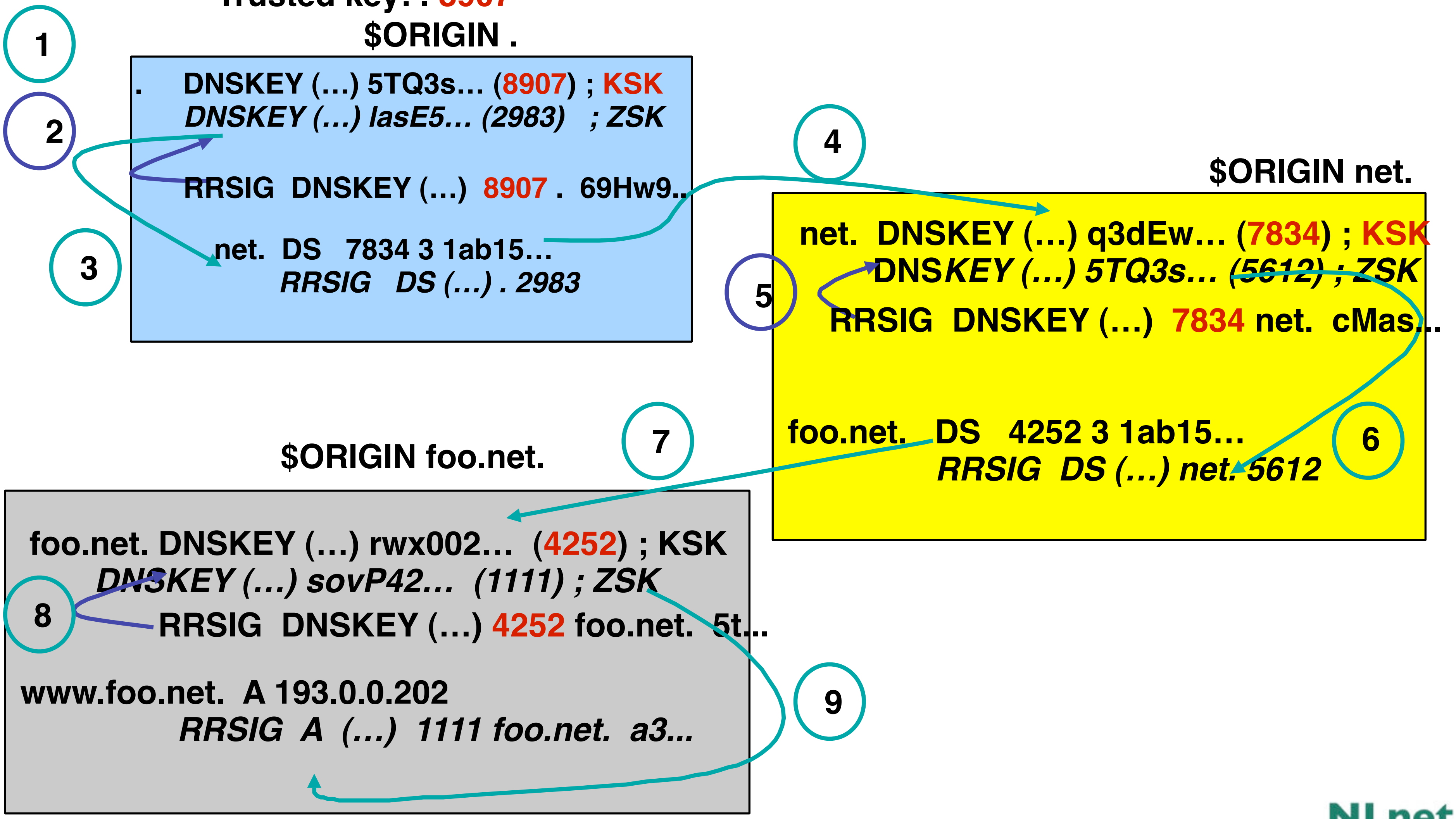
```
net.  DNSKEY (…) q3dEw… (7834) ; KSK
      DNSKEY (…) 5TQ3s… (5612) ; ZSK
    RRSIG  DNSKEY (…)  7834 net.  cMas...

foo.net.  DS   4252 3 1ab15…
          RRSIG  DS (…) net. 5612
```

$ORIGIN foo.net.

```
foo.net. DNSKEY (…) rwx002…  (4252) ; KSK
         DNSKEY (…) sovP42…  (1111) ; ZSK
       RRSIG  DNSKEY (…) 4252 foo.net.  5t...
```

Walking the Chain of Trust

Locally configured
Trusted key: . 8907



$ORIGIN .

```
.    DNSKEY (…) 5TQ3s… (8907) ; KSK
     DNSKEY (…) lasE5… (2983)   ; ZSK

     RRSIG  DNSKEY (…)  8907 .  69Hw9..

        net.  DS   7834 3 1ab15…
              RRSIG   DS (…) . 2983
```

$ORIGIN net.

```
net.  DNSKEY (…) q3dEw… (7834) ; KSK
      DNSKEY (…) 5TQ3s… (5612) ; ZSK
    RRSIG  DNSKEY (…)  7834 net.  cMas...

foo.net.  DS   4252 3 1ab15…
          RRSIG  DS (…) net. 5612
```

$ORIGIN foo.net.

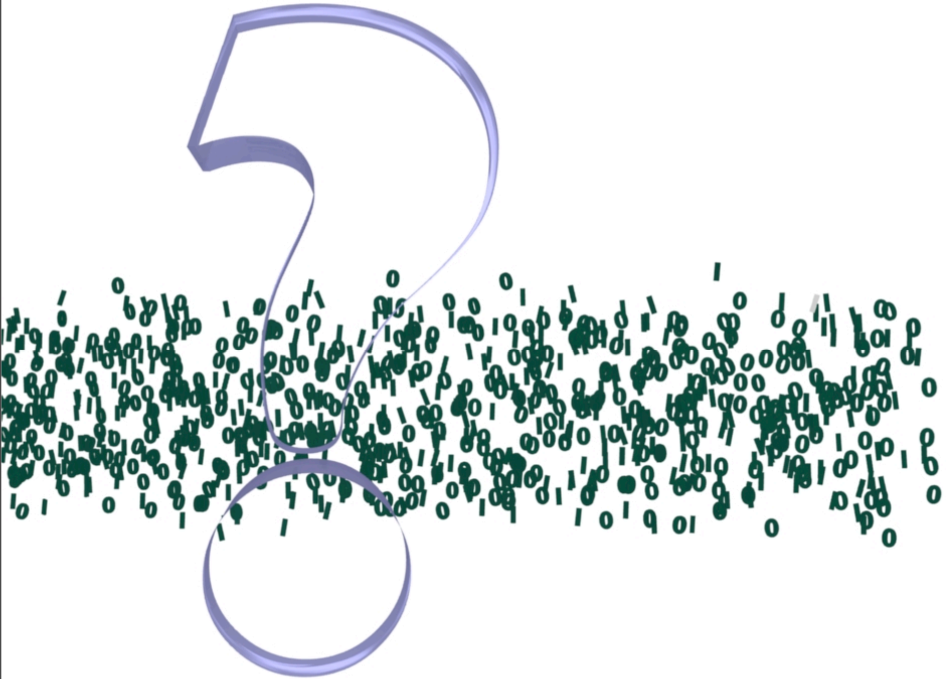
```
foo.net. DNSKEY (…) rwx002…  (4252) ; KSK
         DNSKEY (…) sovP42…  (1111) ; ZSK
         RRSIG  DNSKEY (…) 4252 foo.net.  5t...

www.foo.net.  A 193.0.0.202
              RRSIG  A  (…)  1111 foo.net.  a3...
```

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

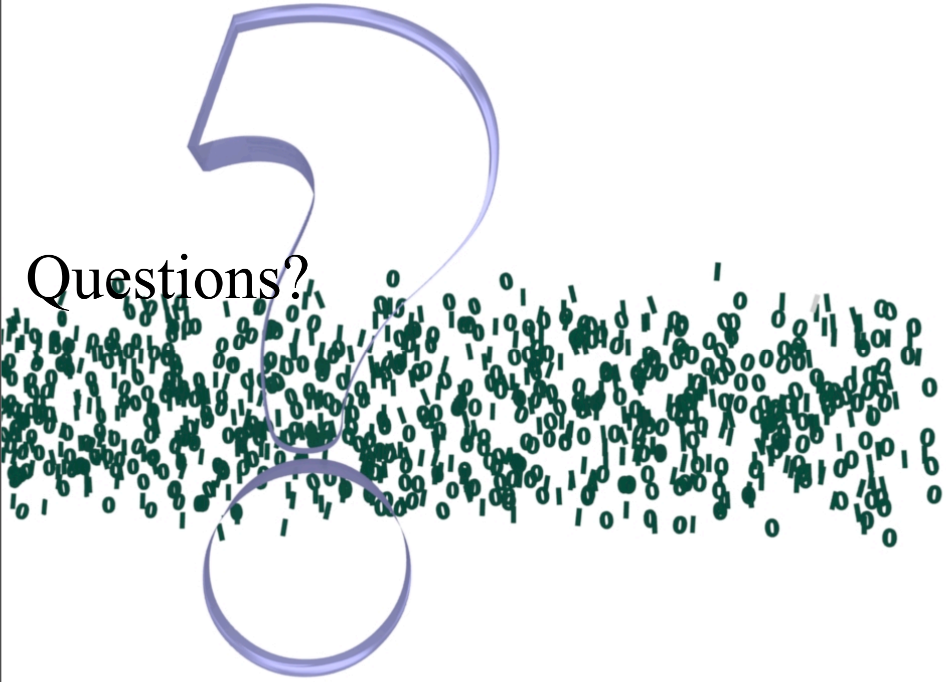
  or
  - DS or DNSKEY records can be trusted if exchanged out-of-band and locally stored (Secure entry point)

Summary



- Scaling problem: secure islands
- Zone signing key, key signing key
- Chain of trust

Summary

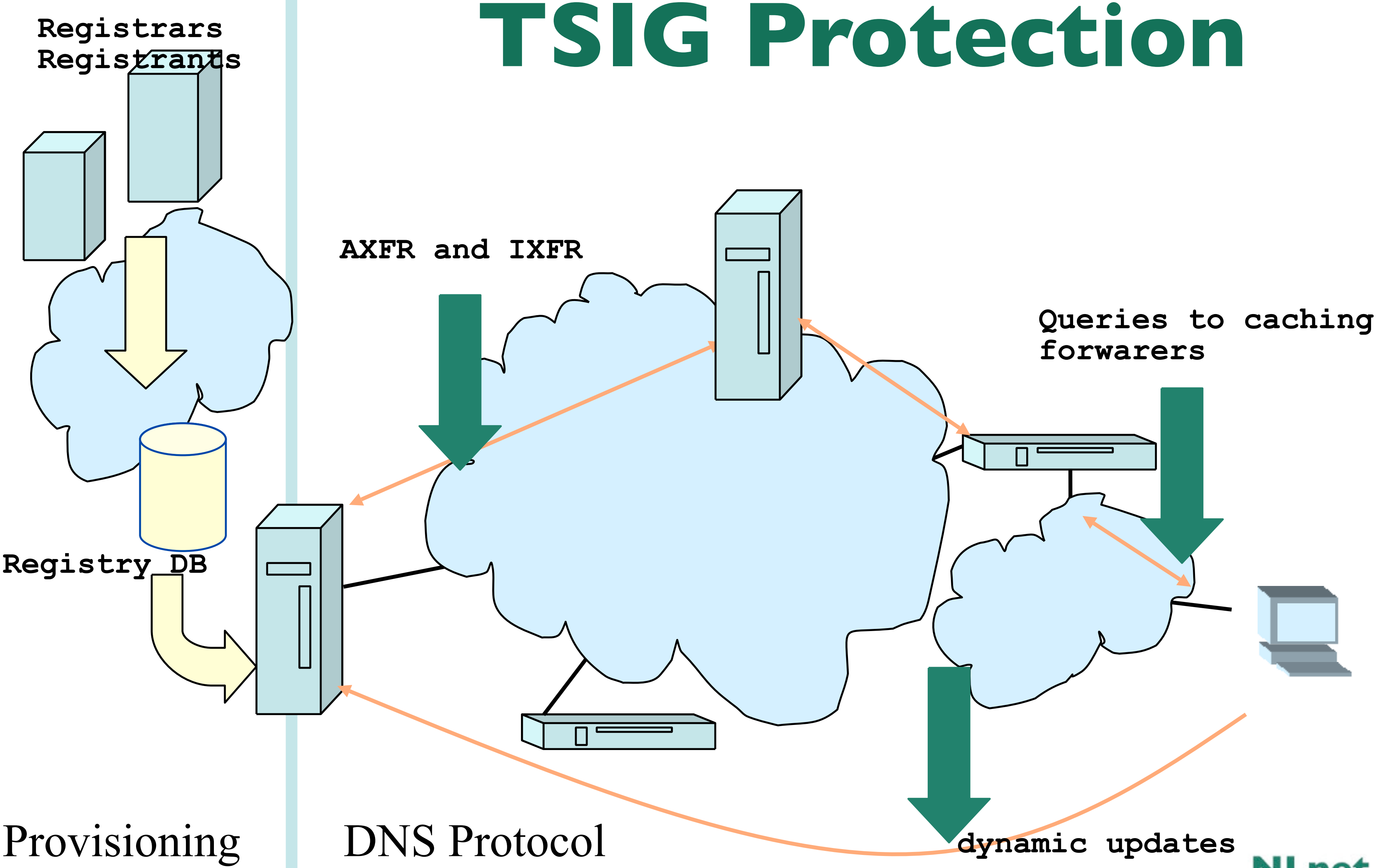


Questions?

- Scaling problem: secure islands
- Zone signing key, key signing key
- Chain of trust

Securing Host-Host Communication

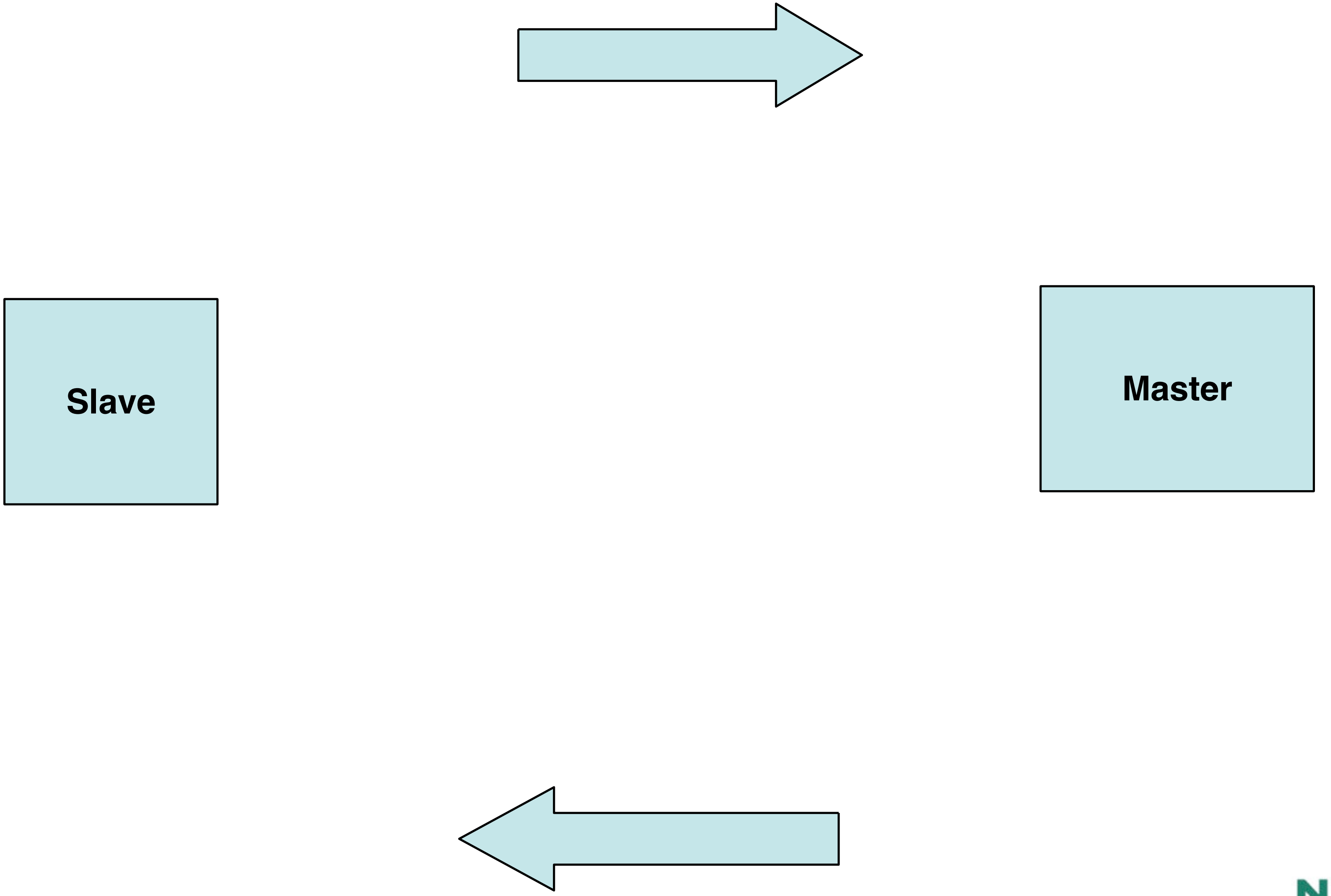
TSIG Protection



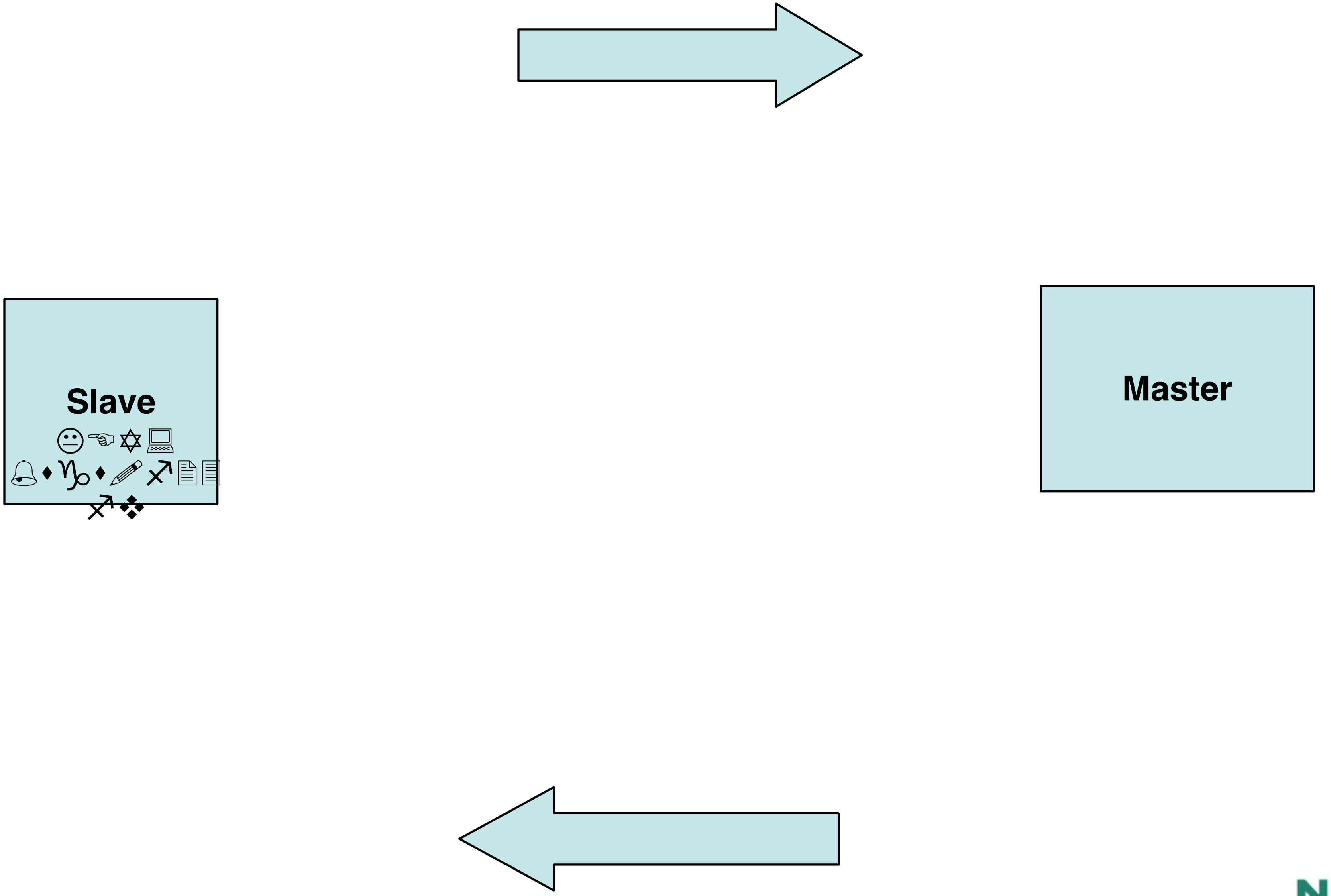
Transaction Signature: TSIG

- TSIG (RFC 2845)
  - Authorising dynamic updates and zone transfers
  - Authentication of caching forwarders
  - Independent from other features of DNSSEC
- One-way hash function
  - DNS question or answer and timestamp
- Traffic signed with "shared secret" key
- Used in configuration, **NOT** in zone file

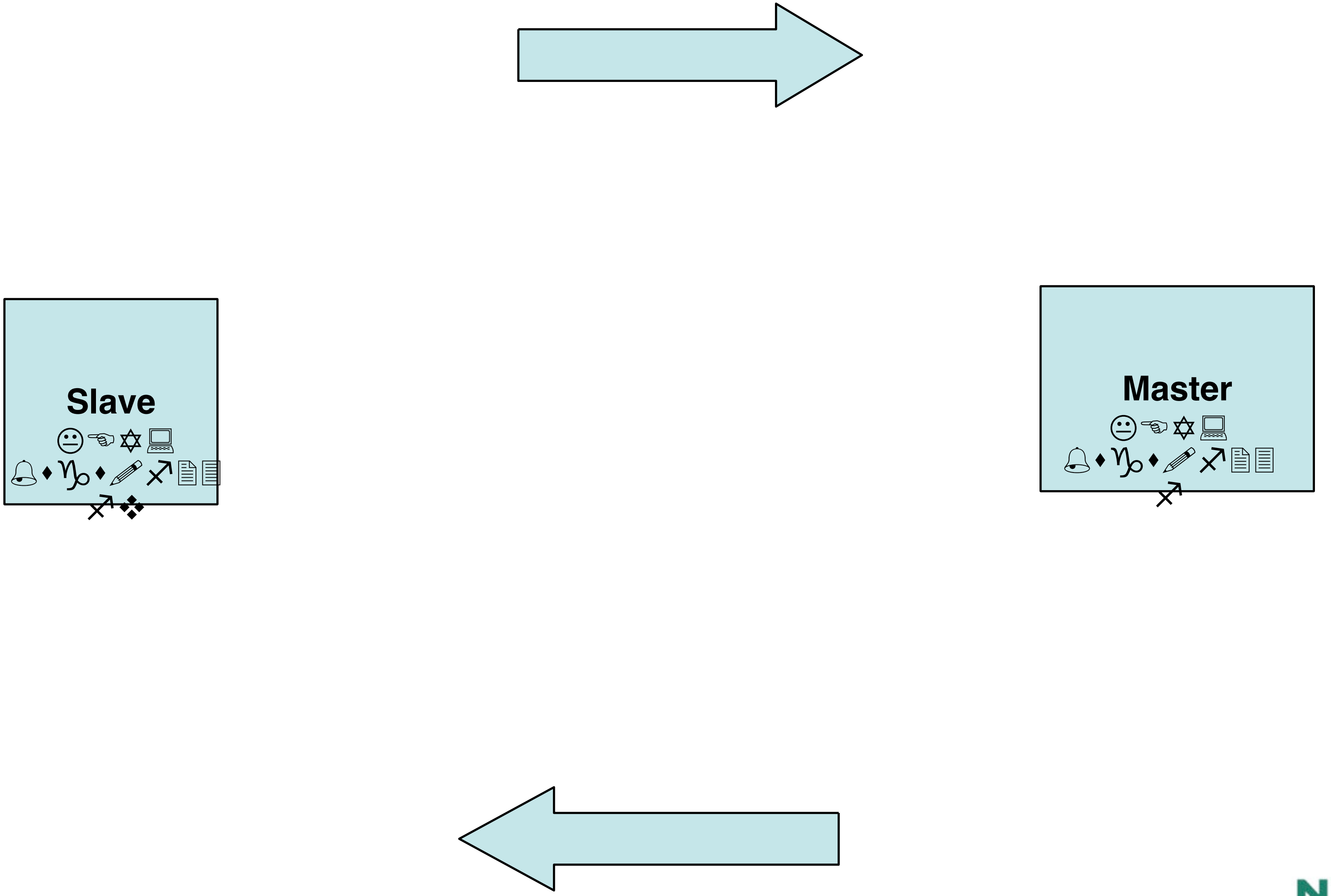
TSIG Example



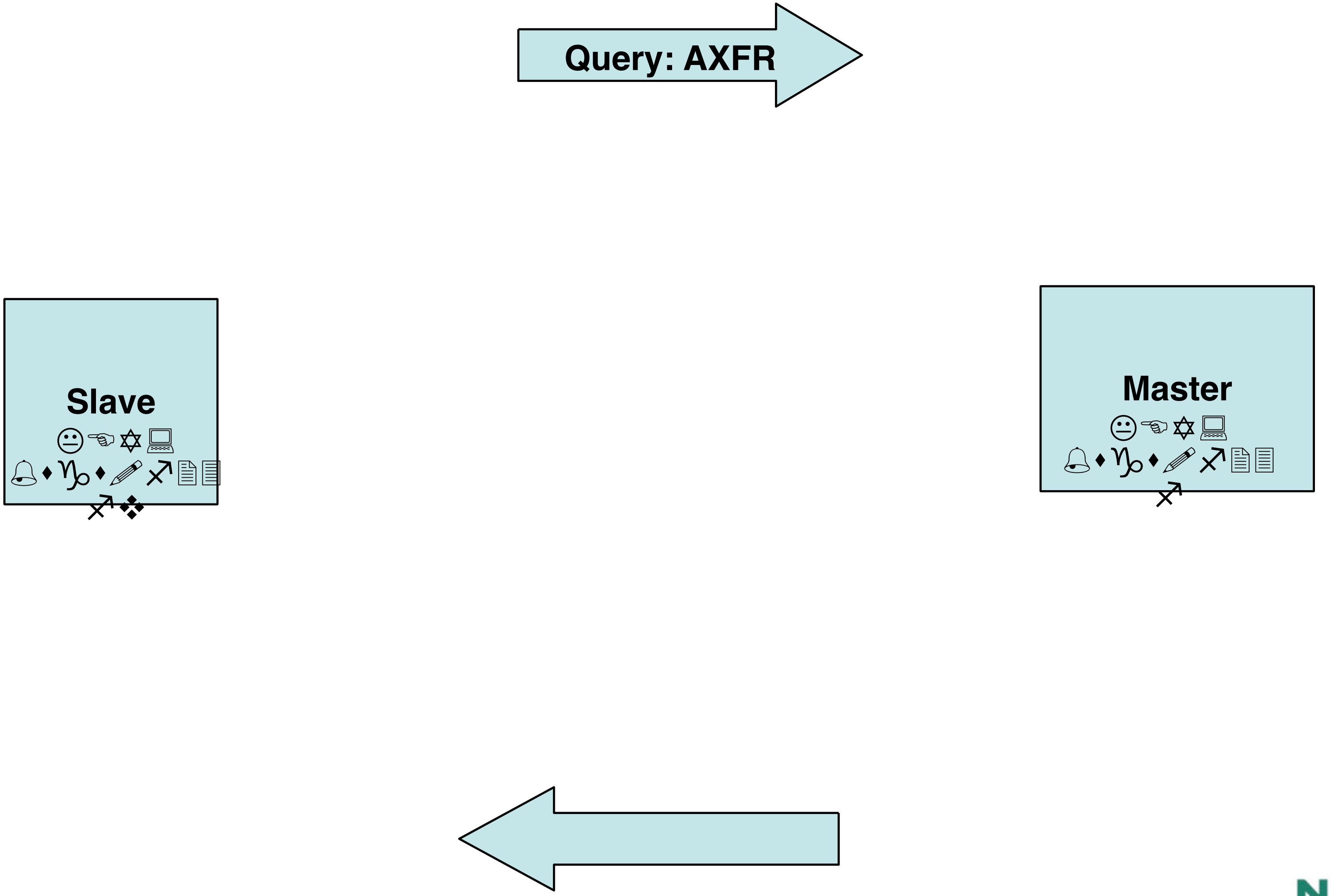
TSIG Example



TSIG Example



TSIG Example



Query: AXFR

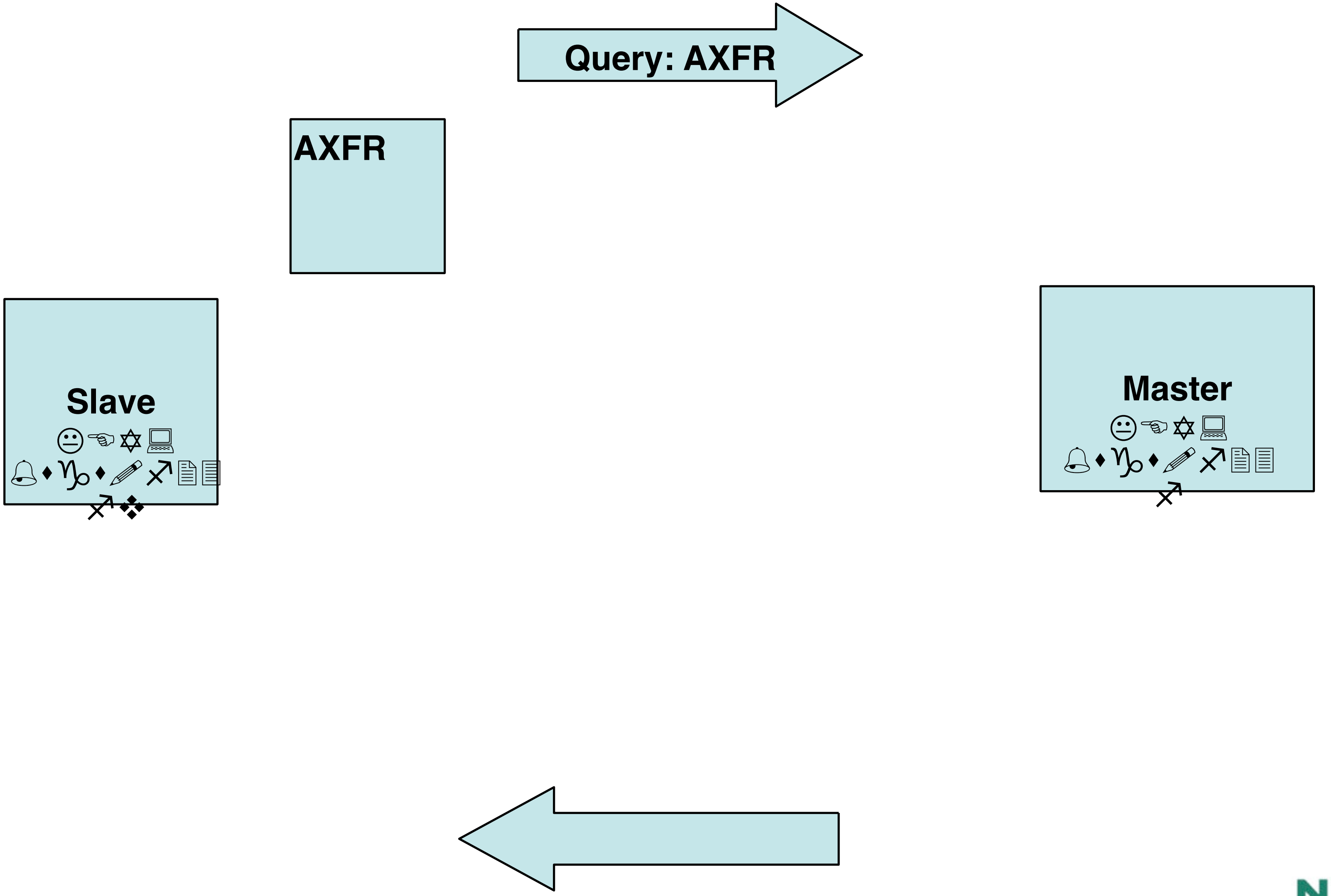
Slave

😐☜✡💻
🔔♦♑♦✏♐🗎🗏
♐❖

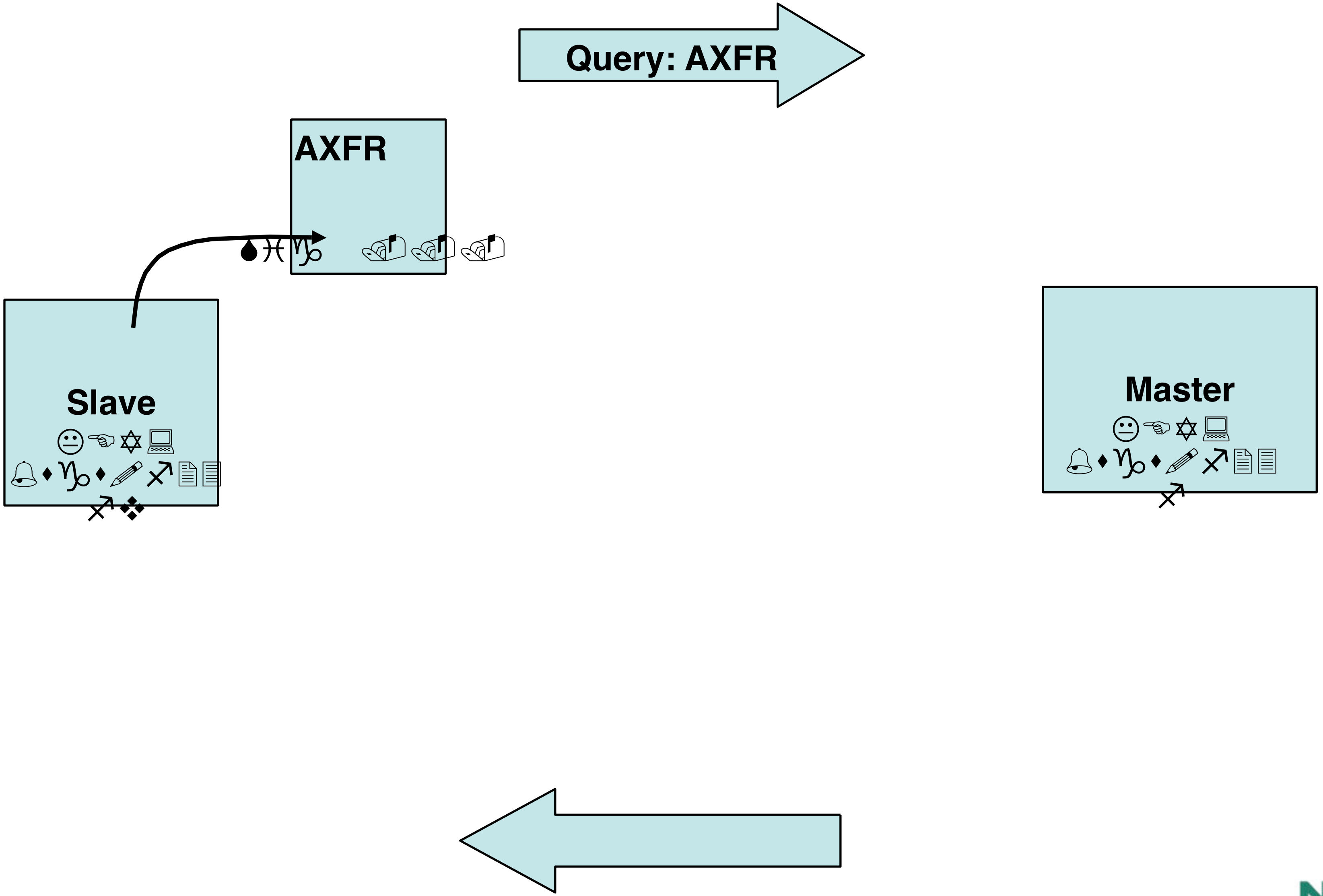
Master

😐☜✡💻
🔔♦♑♦✏♐🗎🗏
♐

TSIG Example



TSIG Example



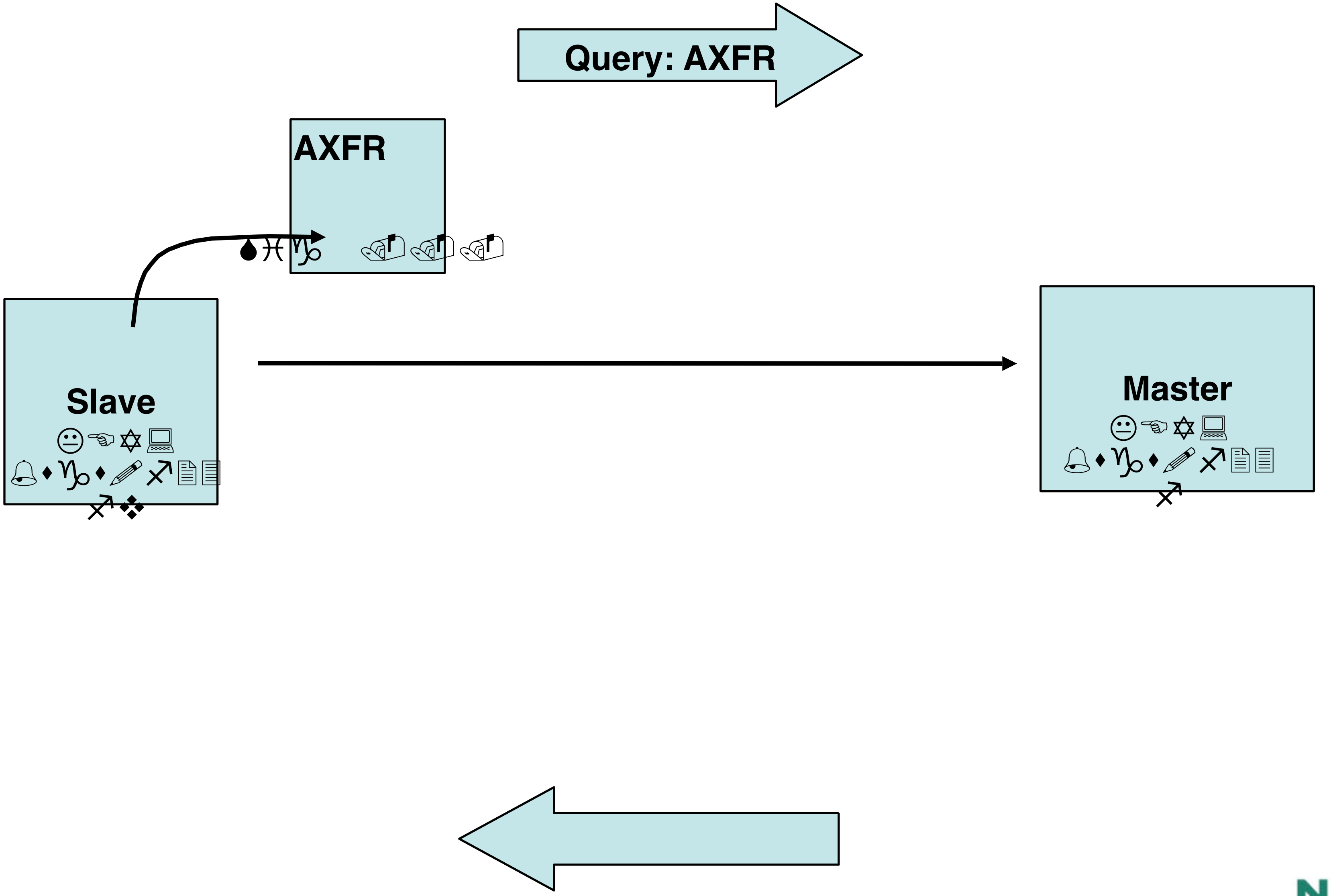
Query: AXFR

AXFR

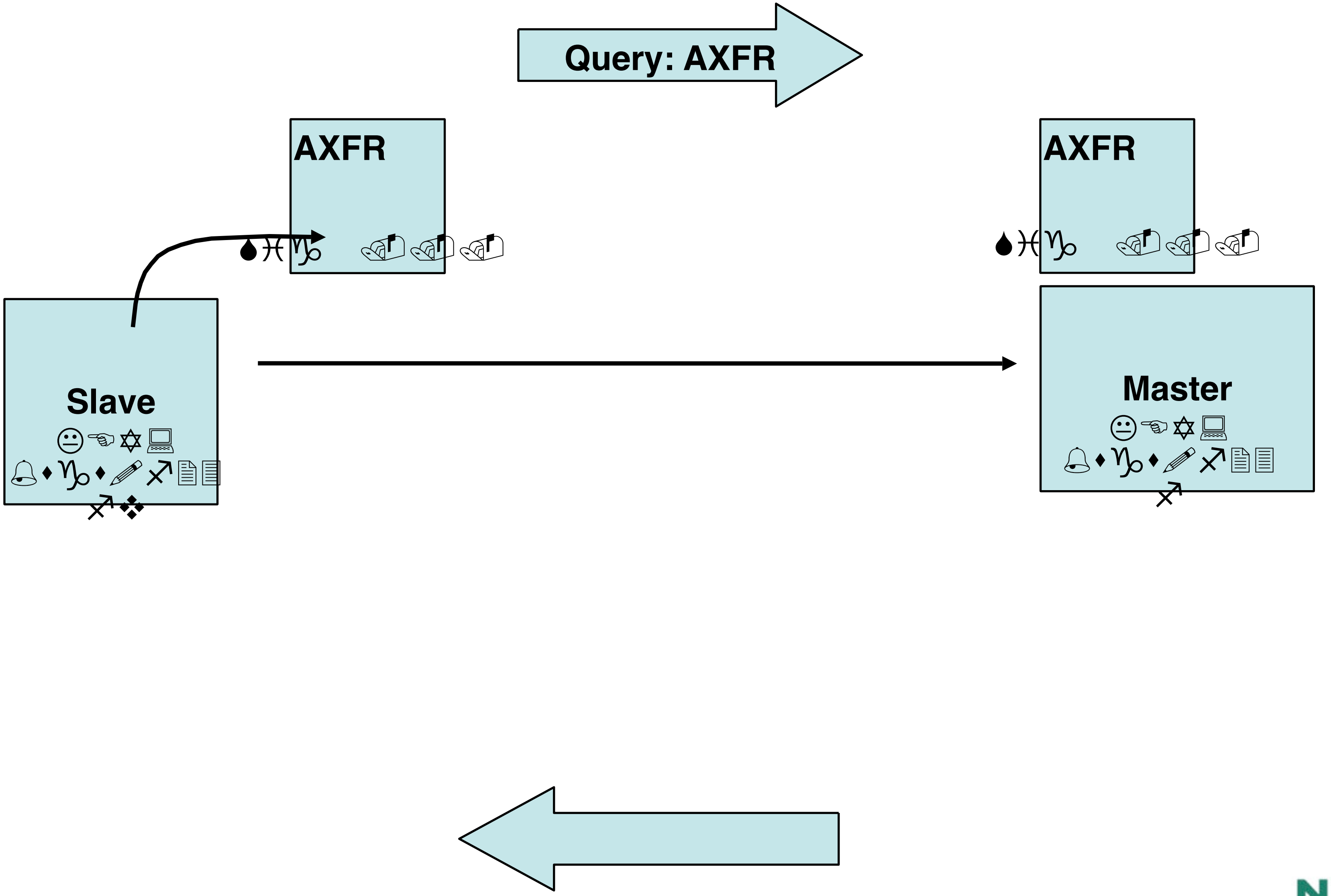
Slave

Master

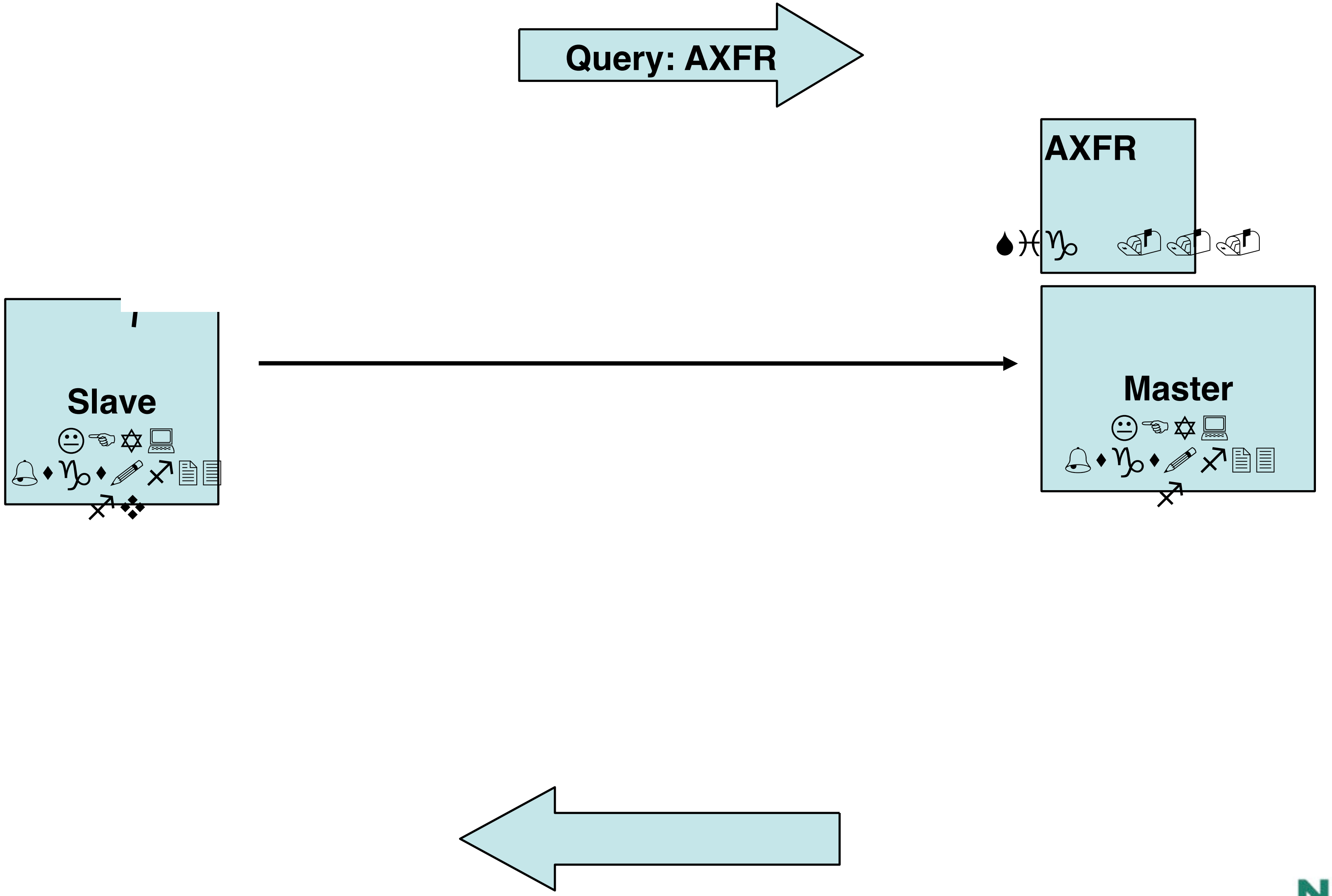
TSIG Example



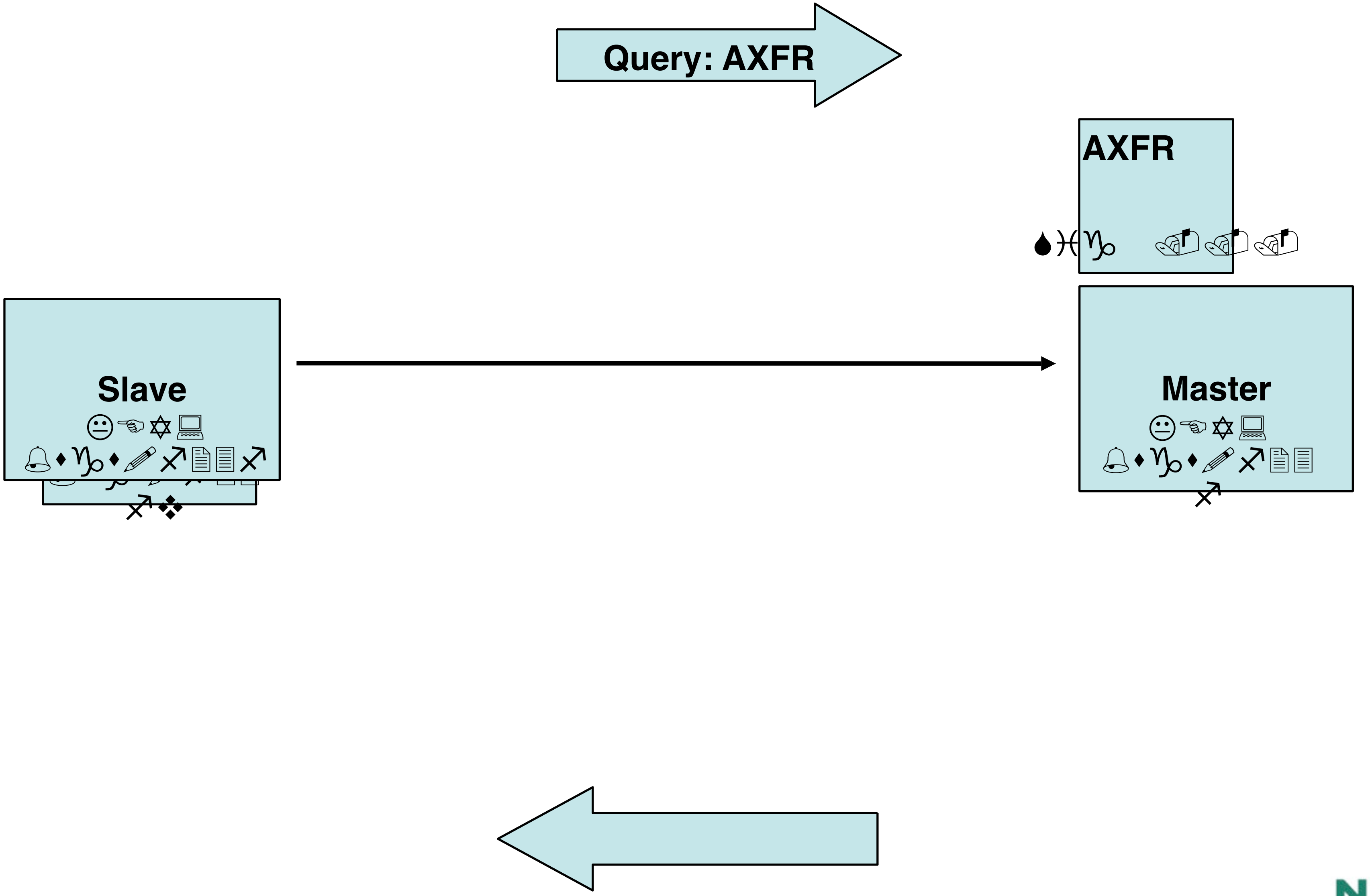
TSIG Example



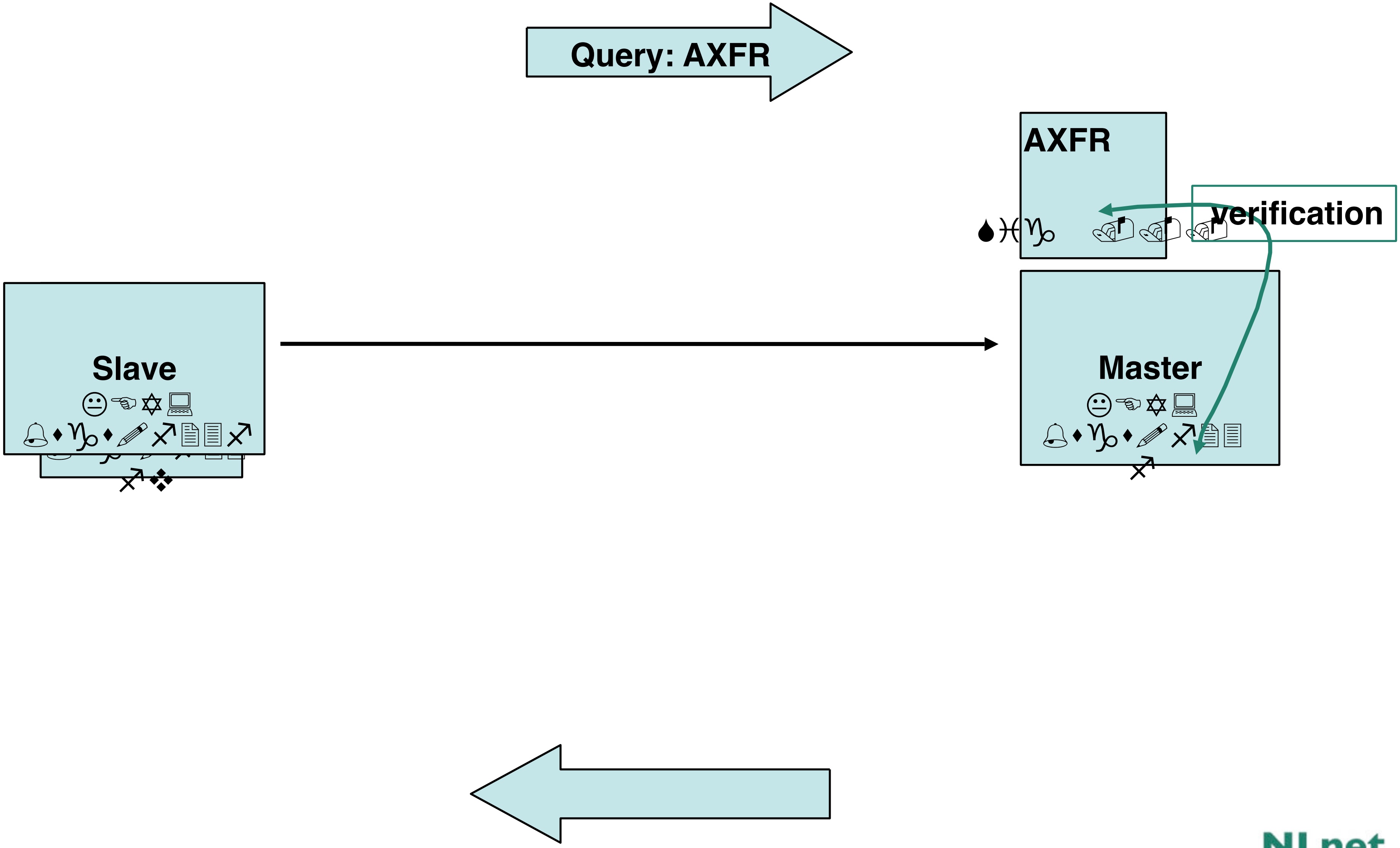
TSIG Example



TSIG Example



TSIG Example



Query: AXFR

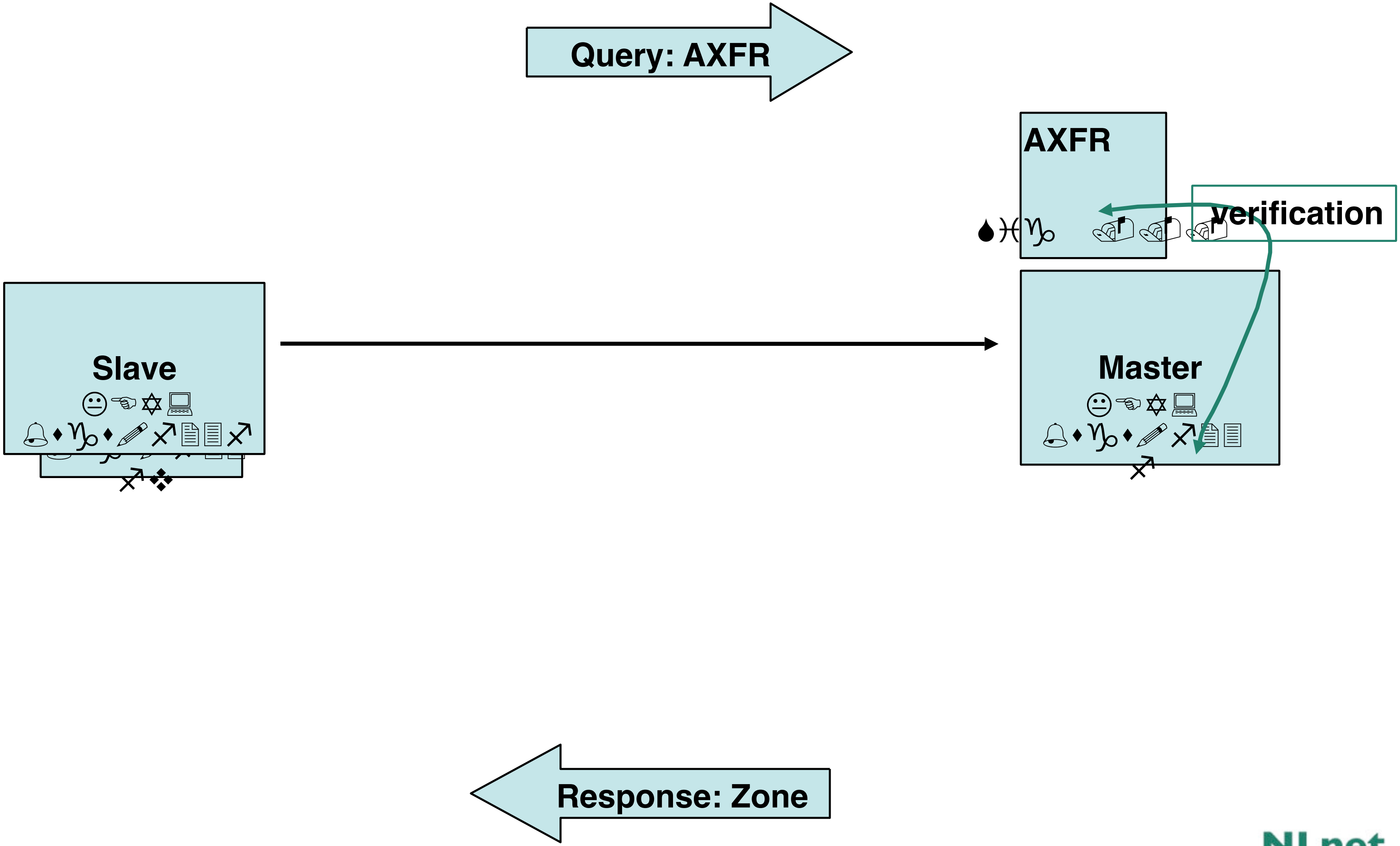
AXFR

verification

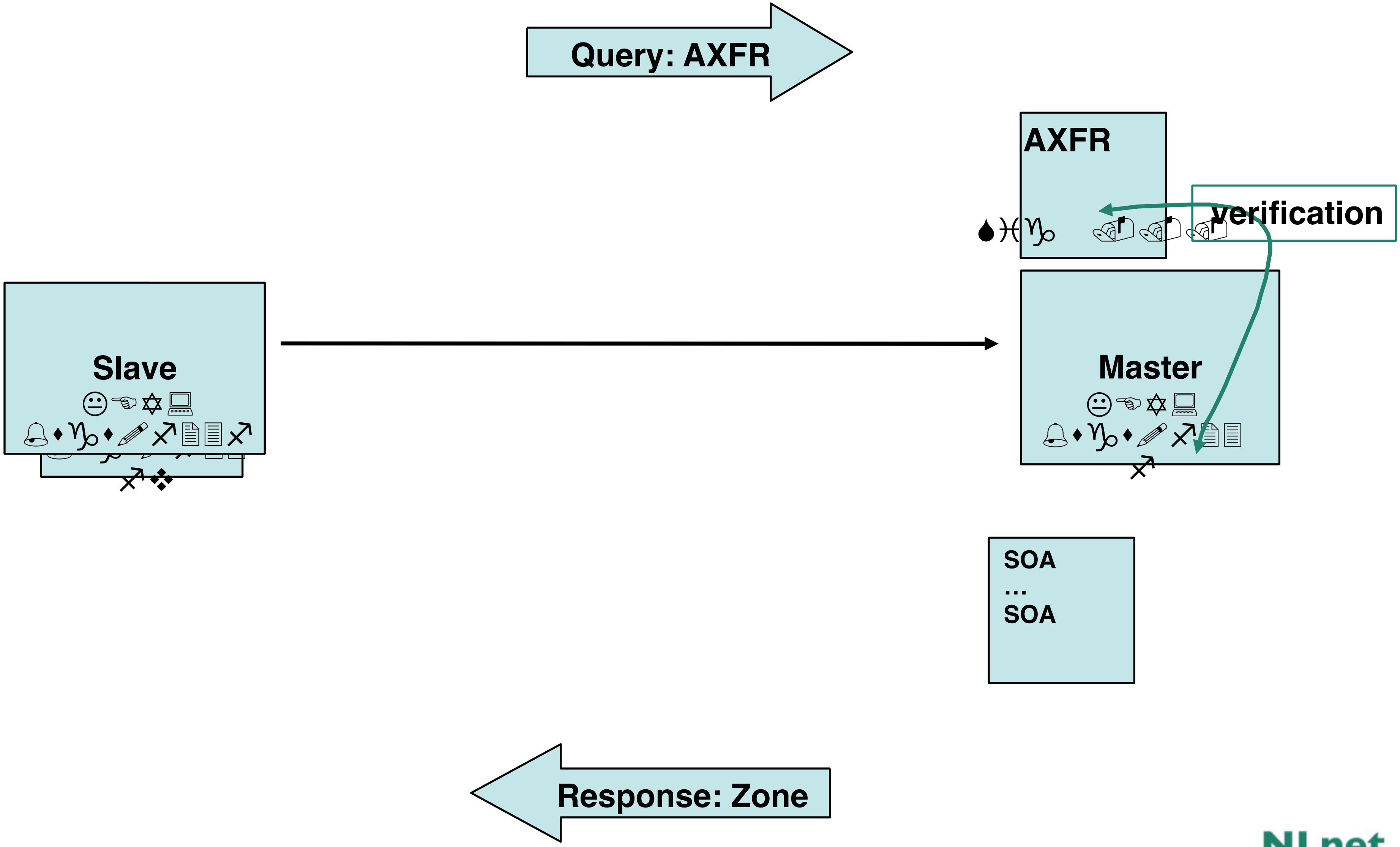
Slave

Master

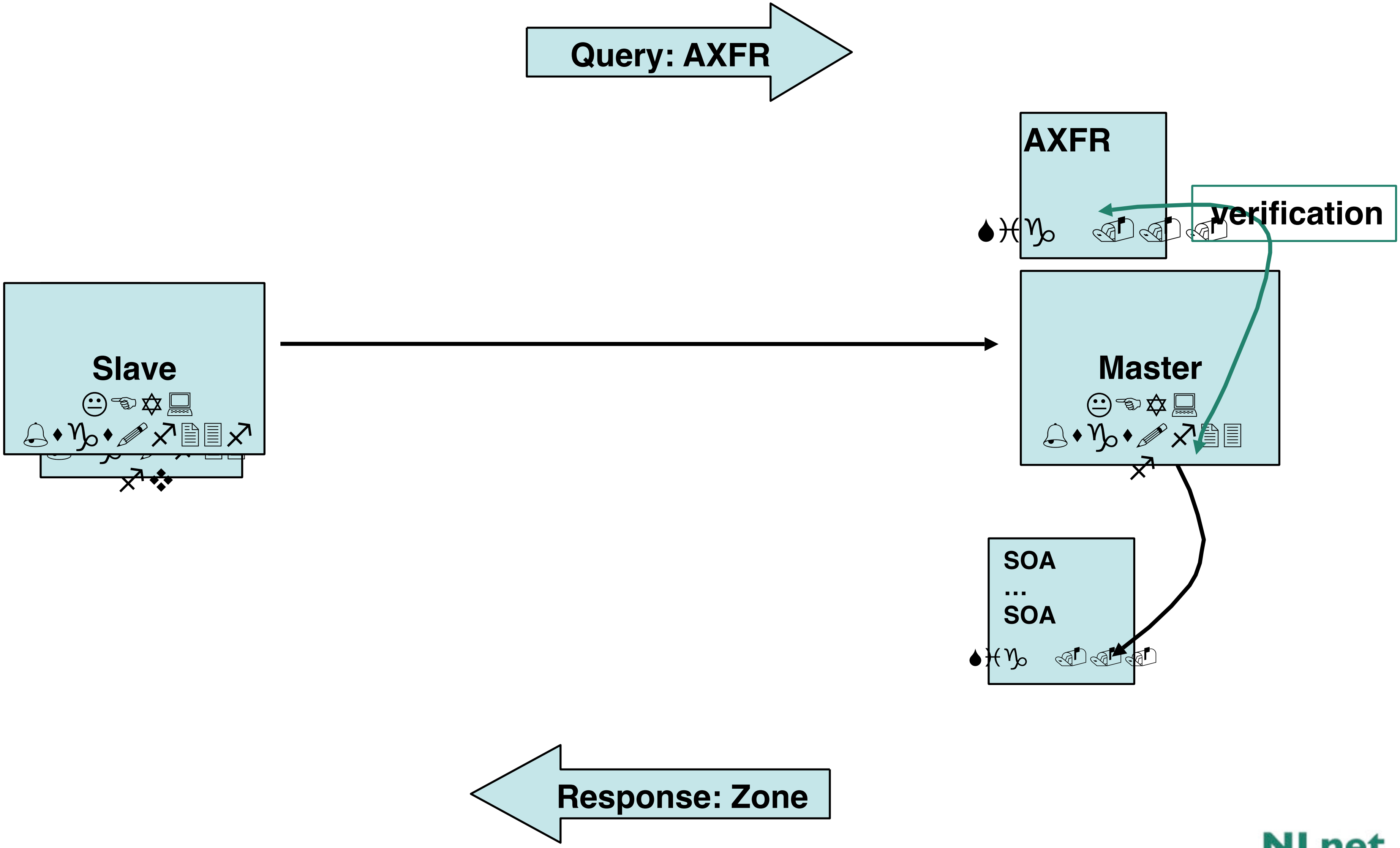
TSIG Example



TSIG Example



TSIG Example



Query: AXFR

AXFR

verification

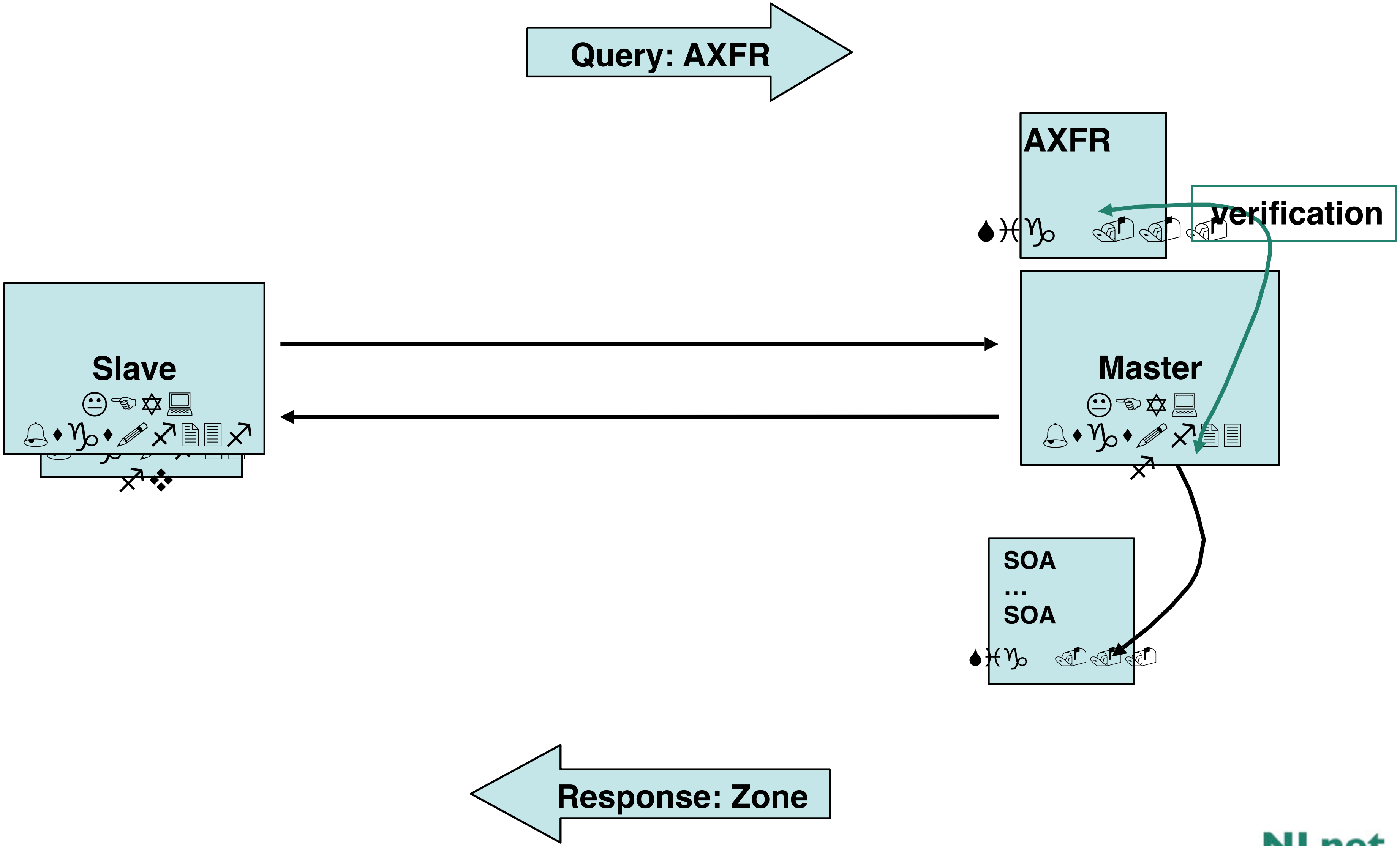
Slave

Master

SOA
...
SOA

Response: Zone

TSIG Example



Query: AXFR

AXFR

verification

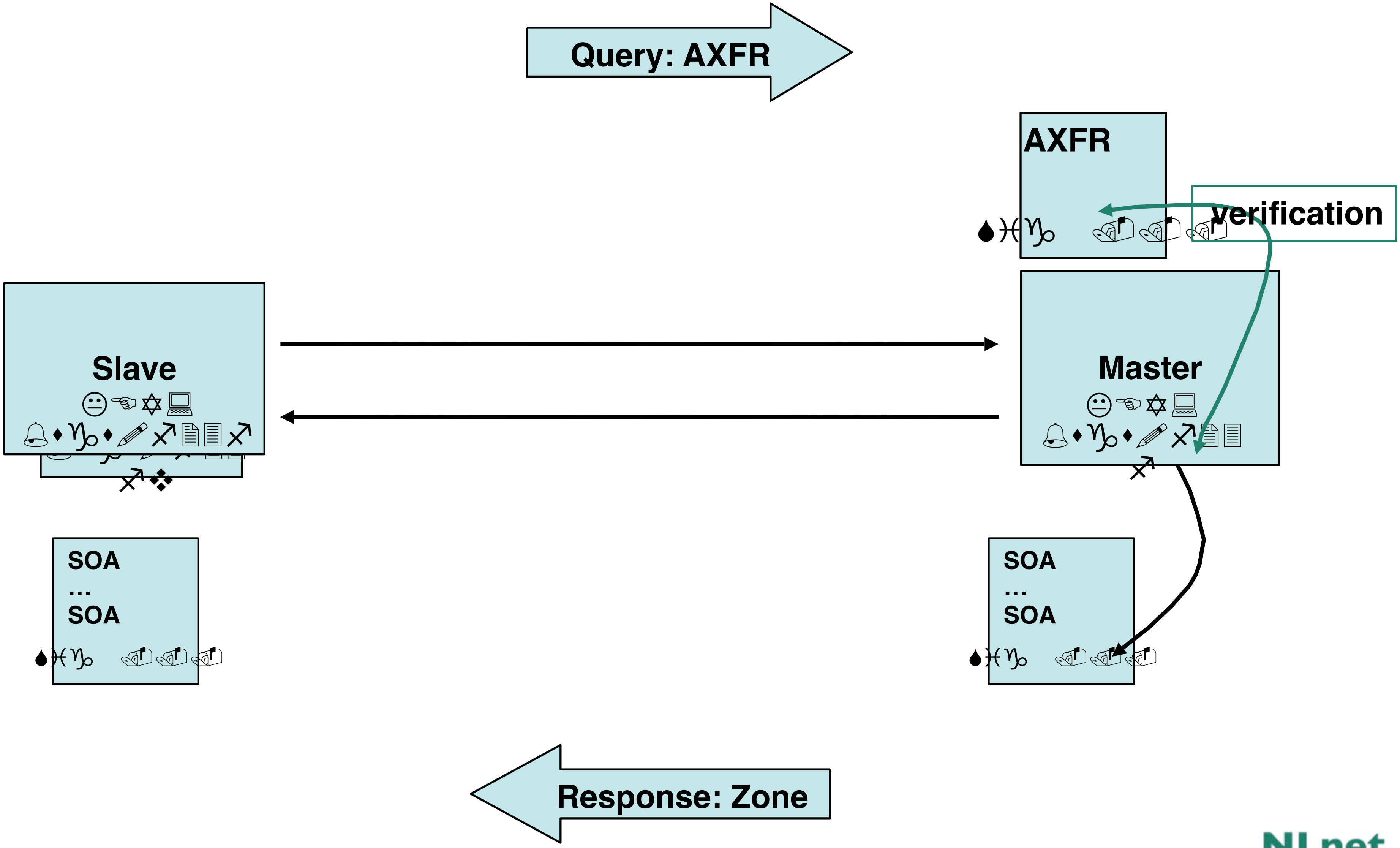
Slave

Master

SOA
…
SOA

Response: Zone

TSIG Example



Query: AXFR

AXFR

verification

Slave

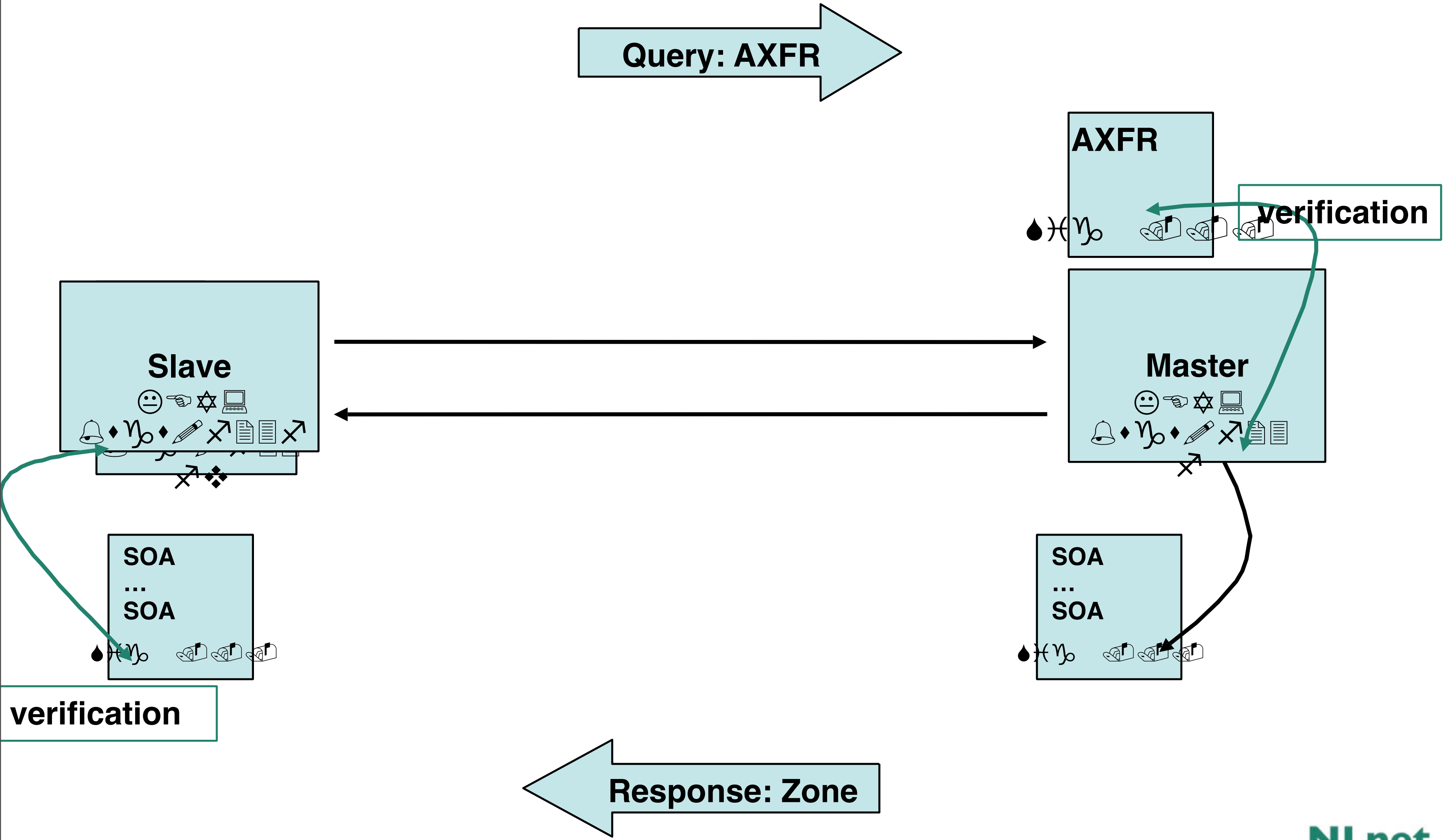
Master

SOA
…
SOA

SOA
…
SOA

Response: Zone

TSIG Example



Query: AXFR

AXFR

verification

Slave

Master

SOA
…
SOA

SOA
…
SOA

verification

Response: Zone

TSIG for Zone Transfers

1. Generate secret
2. Communicate secret
3. Configure servers
4. Test

Importance of the Time Stamp

- TSIG/SIG(0) signs a complete DNS request / response with time stamp
  - To prevent replay attacks
  - Currently hardcoded at five minutes

- Operational problems when comparing times
  - Make sure your local time zone is properly defined
  - `date -u` will give UTC time, easy to compare between the two systems
  - Use NTP synchronisation!

Authenticating Servers Using SIG(0)

- Alternatively, it is possible to use SIG(0)
  - Not yet widely used
  - Works well in dynamic update environment
- Public key algorithm
  - Authentication against a public key published in the DNS
- SIG(0) specified in RFC 2931

Cool Application

- Use TSIG-ed dynamic updates to configure configure your laptops name
- My laptop is know by the name of grover.secret-wg.org
  - http://ops.ietf.org/dns/dynupd/secure-ddns-howto.html
  - Mac OS users: there is a bonjour based tool.
    - www.dns-sd.org

Questions?

ASK

Questions?

ASK