DNS Security

TSIG/DNSSEC

Background

- The original DNS protocol wasn’t designed with security in mind

- It has very few built-in security mechanism

- As the Internet grew wilder & wollier, IETF realized this would be a problem
  - For example DNS spoofing was to easy

- DNSSEC and TSIG were develop to help address this problem

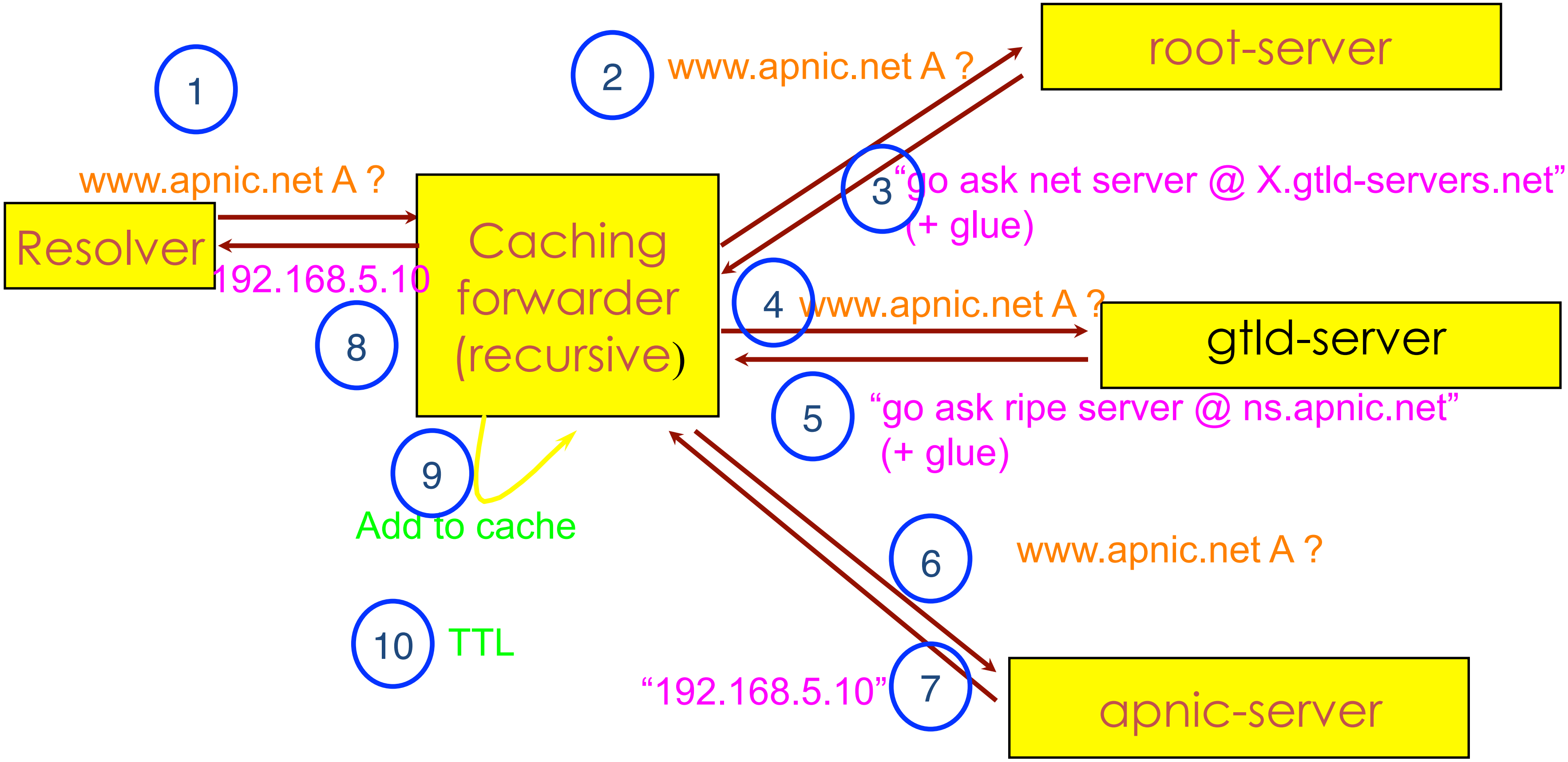
DNS Protocol Vulnerability

- DNS data can be spoofed and corrupted between master server and resolver or forwarder
- The DNS protocol does not allow you to check the validity of DNS data
  - Exploited by bugs in resolver implementation (predictable transaction ID)
  - Polluted caching forwarders can cause harm for quite some time (TTL)
  - Corrupted DNS data might end up in caches and stay there for a long time
- How does a slave (secondary) knows it is talking to the proper master (primary)?

Reminder: DNS Resolving

Question:

www.apnic.net A



Resolver

Caching forwarder (recursive)

root-server

gtld-server

apnic-server

1 www.apnic.net A ?

2 www.apnic.net A ?

3 "go ask net server @ X.gtld-servers.net" (+ glue)

4 www.apnic.net A ?

5 "go ask ripe server @ ns.apnic.net" (+ glue)

6 www.apnic.net A ?

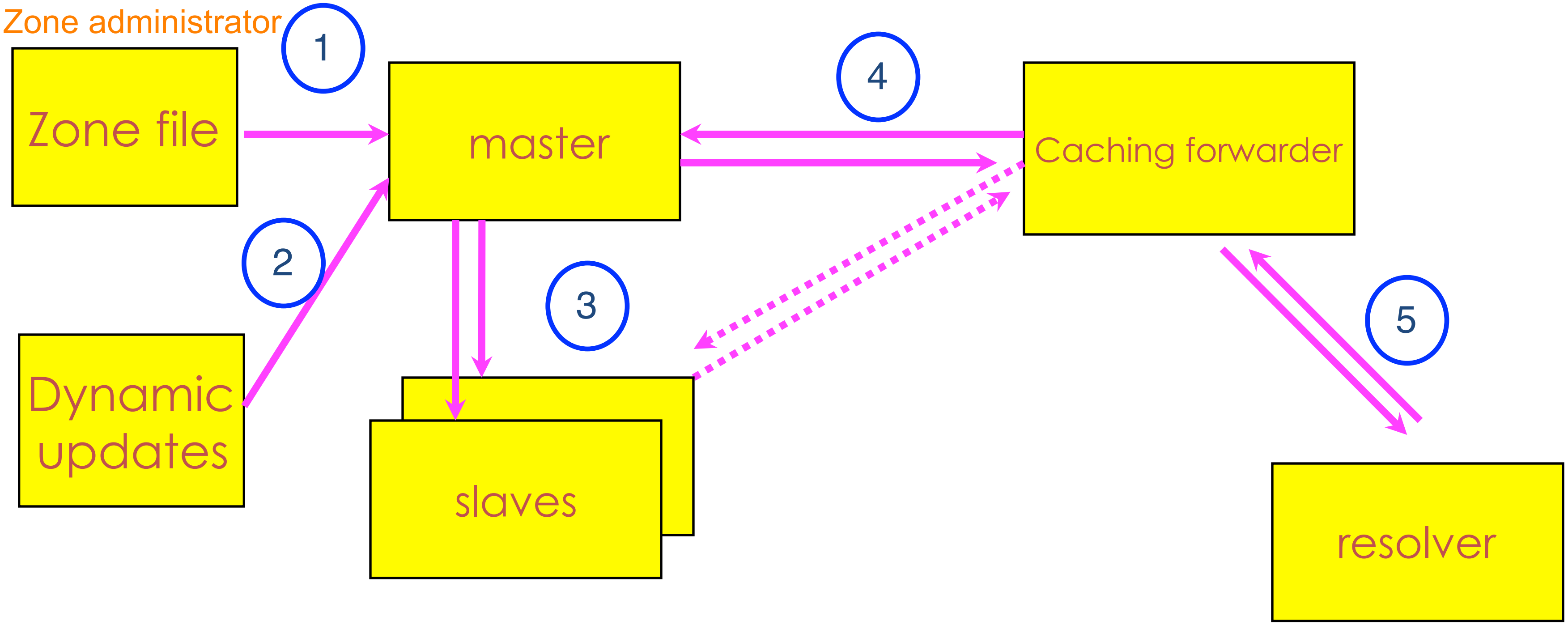
7 "192.168.5.10"

8 192.168.5.10

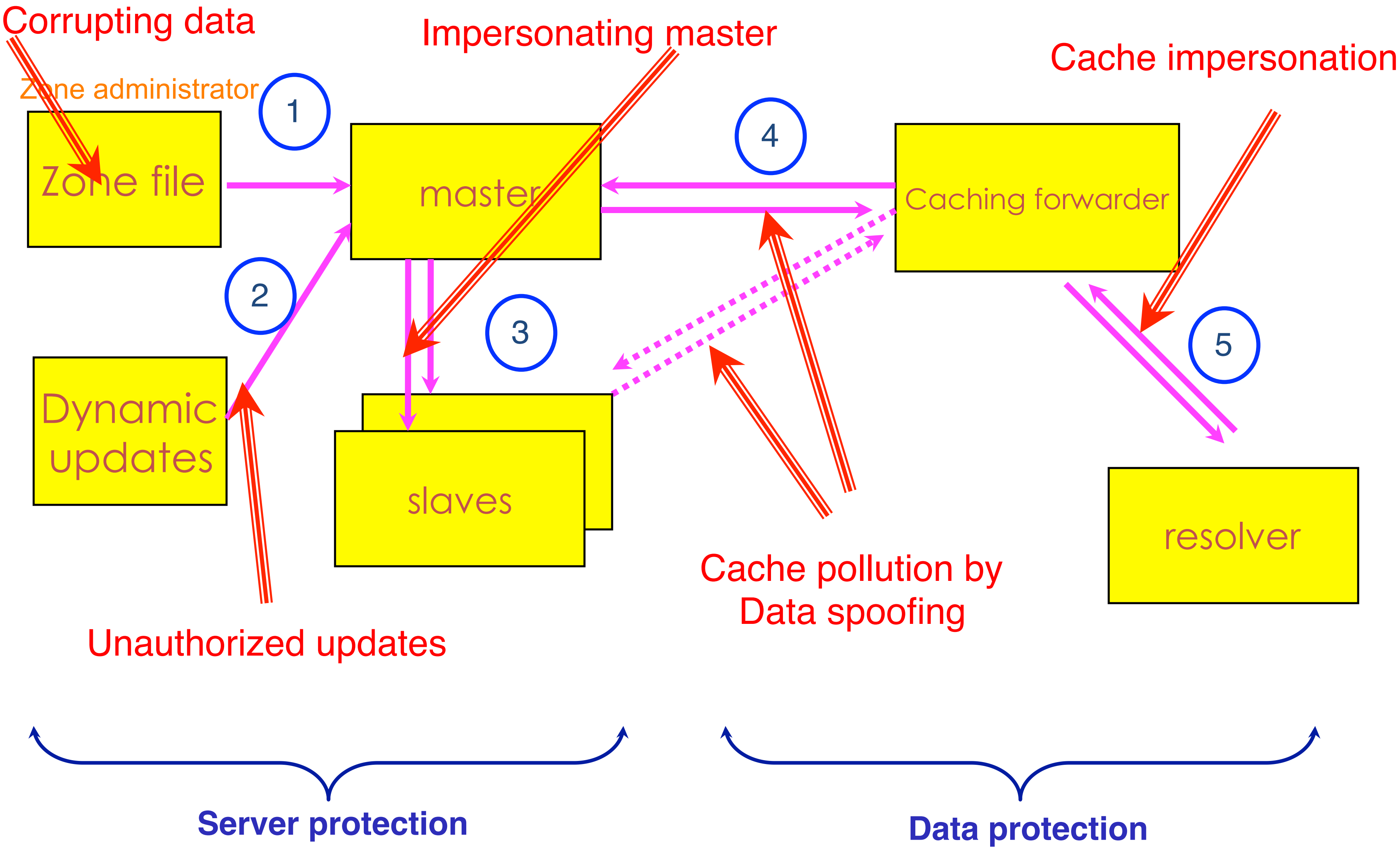
9 Add to cache

10 TTL

DNS: Data Flow



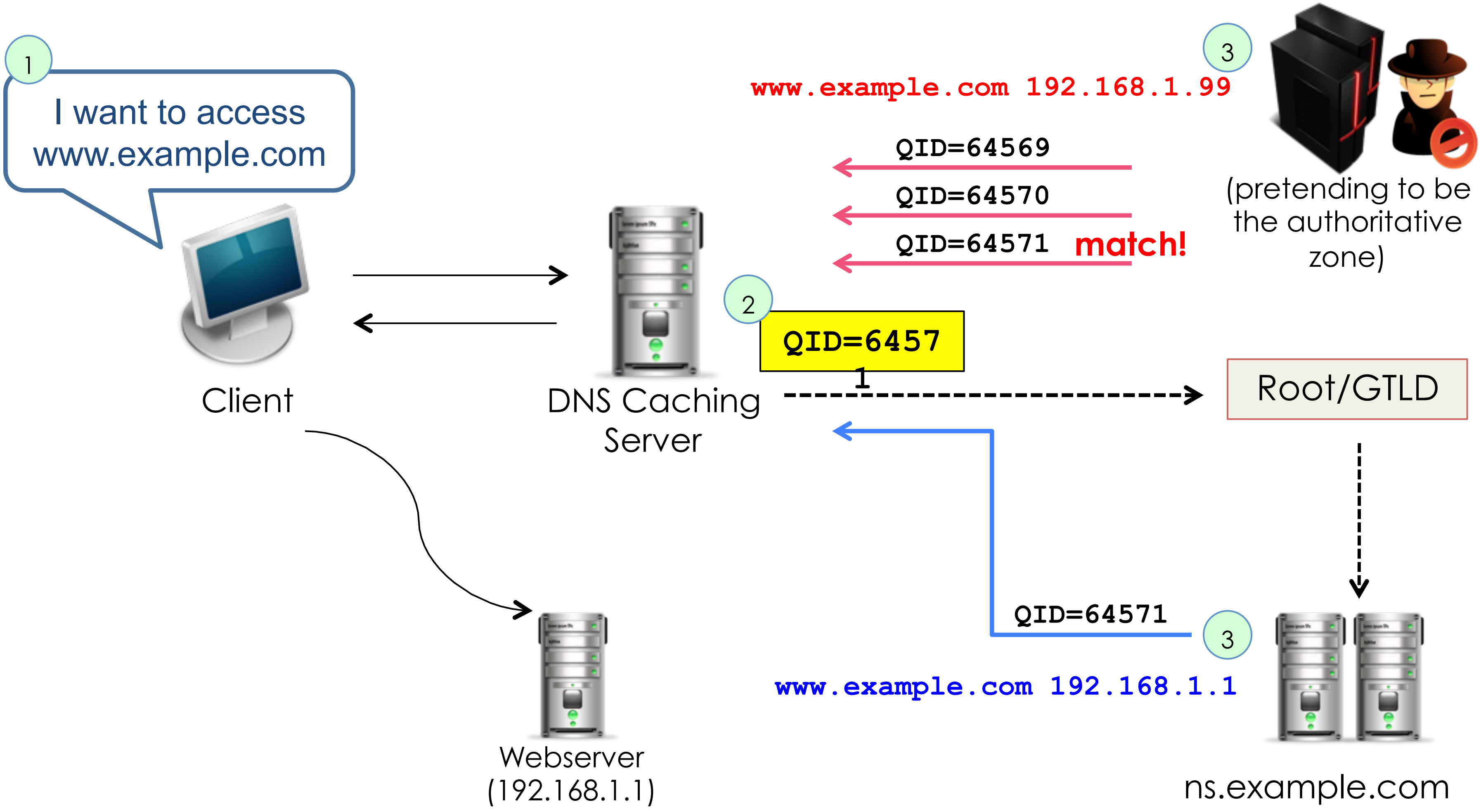
DNS Vulnerabilities



DNS Cache Poisoning

- Caching incorrect resource record that did not originate from authoritative DNS sources.
- Result: connection (web, email, network) is redirected to another target (controlled by the attacker)

DNS Cache Poisoning



1

I want to access www.example.com

Client

DNS Caching Server

2

QID=6457
1

Root/GTLD

3

www.example.com 192.168.1.99

QID=64569

QID=64570

QID=64571 match!

(pretending to be the authoritative zone)

QID=64571

3

www.example.com 192.168.1.1

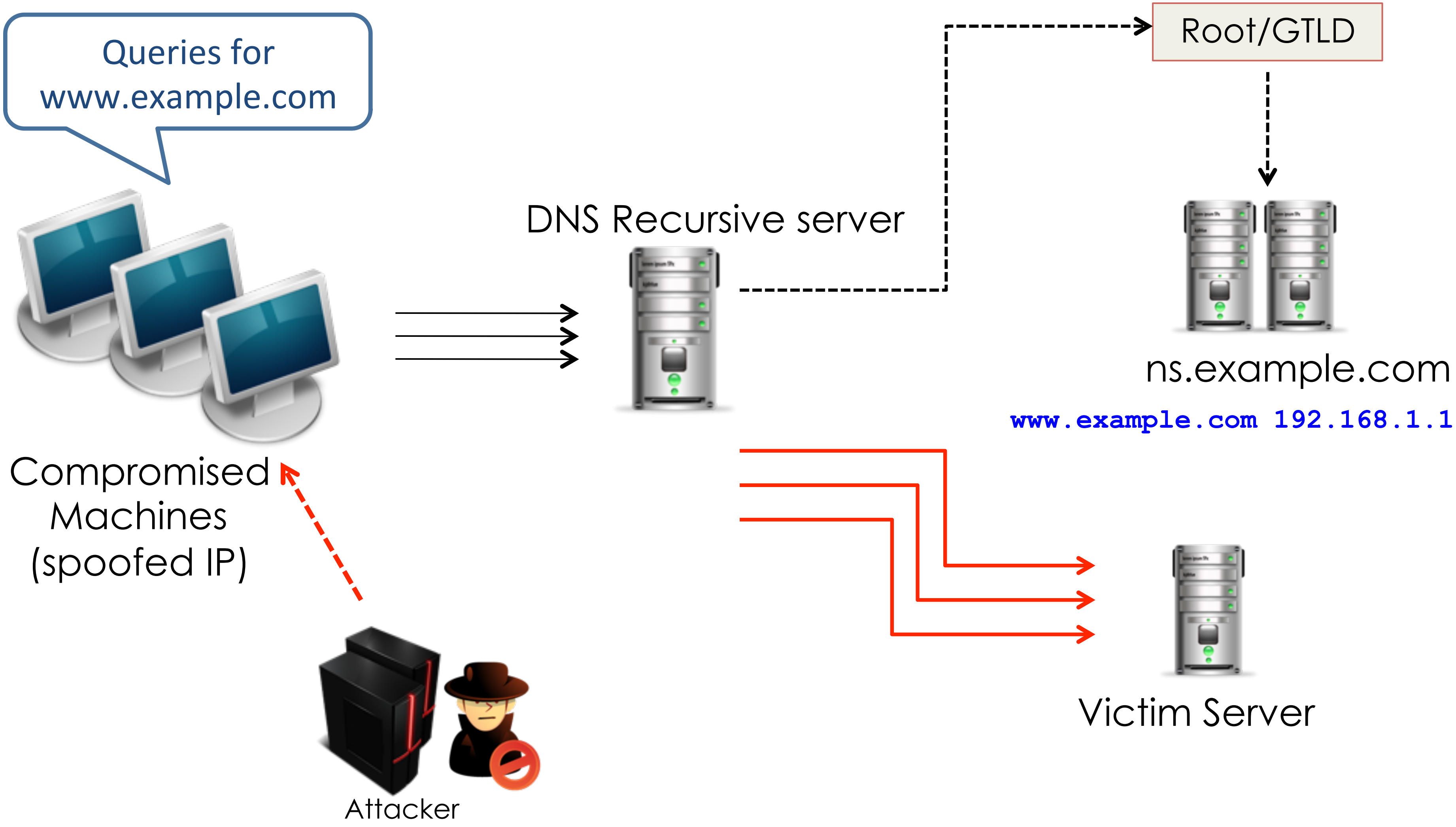
ns.example.com

Webserver
(192.168.1.1)

DNS Amplification

- A type of reflection attack combined with amplification
  - Source of attack is reflected off another machine
  - Traffic received is bigger (amplified) than the traffic sent by the attacker
- UDP packet's source address is spoofed

DNS Amplification Attack



Queries for www.example.com

DNS Recursive server

Root/GTLD

ns.example.com

`www.example.com 192.168.1.1`

Compromised Machines (spoofed IP)

Victim Server

Attacker

What is TSIG - Transaction Signature?

- A mechanism for protecting a message from a primary to secondary and vice versa

- A keyed-hash is applied (like a digital signature) so recipient can verify message
  - DNS question or answer
  - & the timestamp

- Based on a shared secret - both sender and receiver are configured with it

What is TSIG - Transaction Signature?

- TSIG (RFC 2845)
  - authorizing dynamic updates & zone transfers
  - authentication of caching forwarders

- Used in server configuration, not in zone file

TSIG steps

1. Generate secret

2. Communicate secret

3. Configure servers

4. Test

TSIG - Names and Secrets

- TSIG name
  - A name is given to the key, the name is what is transmitted in the message (so receiver knows what key the sender used)
- TSIG secret value
  - A value determined during key generation
  - Usually seen in Base64 encoding

TSIG – Generating a Secret

- dnssec-keygen
  - Simple tool to generate keys
  - Used here to generate TSIG keys

```
> dnssec-keygen -a <algorithm> -b
  <bits> -n host <name of the key>
```

TSIG – Generating a Secret

- **Example**

```
> dnssec-keygen –a HMAC-MD5 –b 128 –n HOST ns1-ns2.pcx.net

This will generate the key
> Kns1-ns2.pcx.net.+157+15921

>ls
➢ Kns1-ns2.pcx.net.+157+15921.key
➢ Kns1-ns2.pcx.net.+157+15921.private
```

TSIG – Generating a Secret

- TSIG should never be put in zone files!!!
  - might be confusing because it looks like RR:

```
ns1-ns2.pcx.net. IN KEY 128 3 157 nEfRX9…bbPn7lyQtE=
```

TSIG – Configuring Servers

- Configuring the key
  - in named.conf file, same syntax as for rndc
  - `key { algorithm ...; secret ...;}`
- Making use of the key
  - in named.conf file
  - `server x { key ...; }`
  - where 'x' is an IP number of the other server

Configuration Example – named.conf

Primary server 10.33.40.46

```
key ns1-ns2.pcx. net {
    algorithm hmac-md5;
    secret "APlaceToBe";
};
server 10.33.50.35 {
    keys {ns1-ns2.pcx.net;};
};
zone "my.zone.test." {
    type master;
    file “db.myzone”;
    allow-transfer {
    key ns1-ns2.pcx.net ;};
};
```

Secondary server 10.33.50.35

```
key ns1-ns2.pcx.net {
    algorithm hmac-md5;
    secret "APlaceToBe";
};
server 10.33.40.46 {
  keys {ns1-ns2.pcx.net;};
};
zone "my.zone.test." {
    type slave;
    file “myzone.backup”;
    masters {10.33.40.46;};
};
```

You can save this in a file and refer to it in the named.conf using ‘include’ statement:

```
include “/var/named/master/tsig-key-ns1-ns2”;
```

TSIG Testing : dig

- You can use dig to check TSIG configuration
  - `dig  @<server> <zone> AXFR -k <TSIG keyfile>`

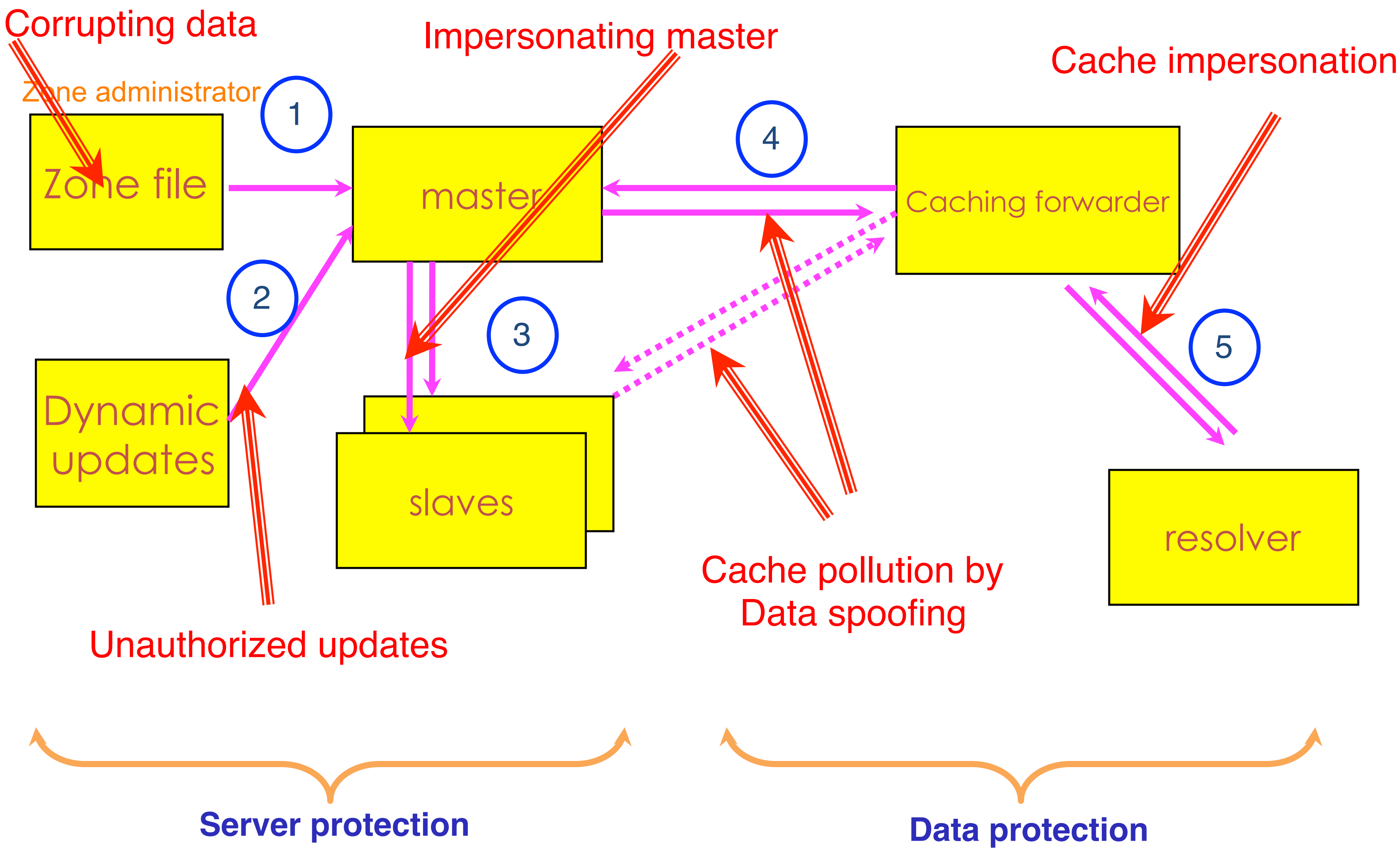
```
$ dig @127.0.0.1 example.net AXFR \
    -k Kns1-ns2.pcx.net.+157+15921.key
```

- Wrong key will give "Transfer failed" and on the server the security-category will log this.

TSIG Testing - TIME!

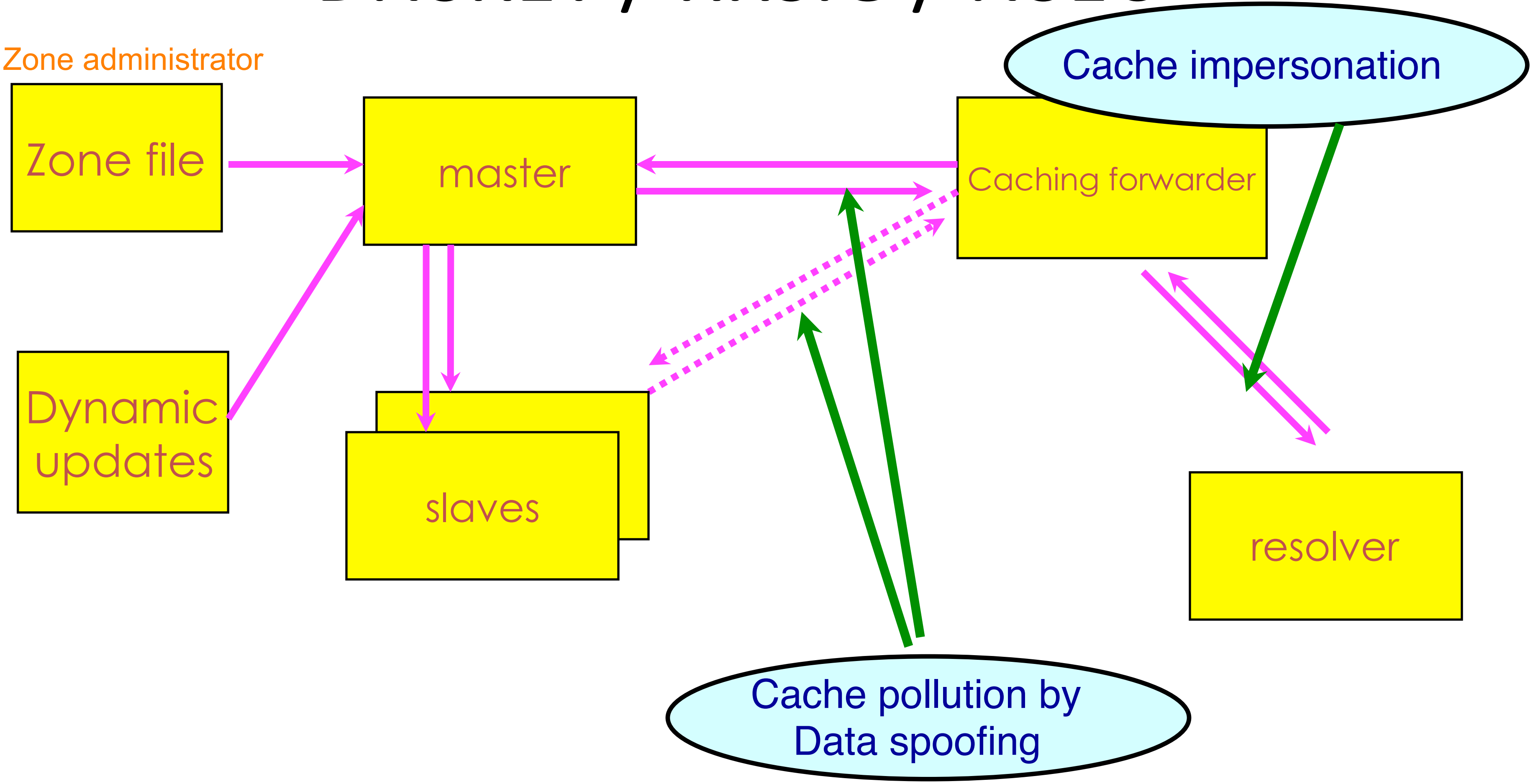
- TSIG is time sensitive - to stop replays
  - Message protection expires in 5 minutes
  - Make sure time is synchronized
  - For testing, set the time
  - In operations, (secure) NTP is needed

DNS Vulnerabilities



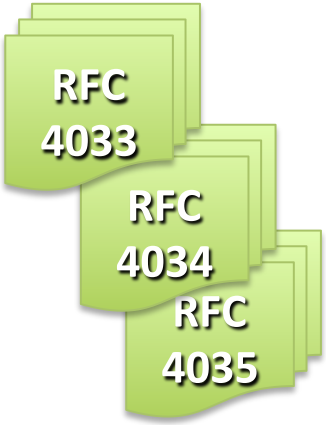
DNSSEC

Vulnerabilities protected by DNSKEY / RRSIG / NSEC



DNS Security Extensions (DNSSEC)

- Protects the integrity of data in the DNS by establishing a chain of trust
- Uses public key cryptography – each link in the chain has a public/private key pair
- A form of digitally signing the data to attest its validity
- Standard is defined in RFC4033, RFC4034, and RFC4035
- Guarantees
  - Authenticity
  - Integrity
  - Non-existence of a domain



RFC 4033

RFC 4034

RFC 4035

DNSSEC Resource Records

RFC 4034

- 3 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** = Next Secure; indicates which name is the next one in the zone and which typecodes are available for the current name
    - authenticated non-existence of data

DNSSEC Resource Records

- DNSKEY, RRSIG, and NSEC records provide mechanisms to establish authenticity and integrity of data
- DS record provides a mechanism to delegate trust to public keys of third parties

DNSSEC RRs

- Data authenticity and integrity by signing the Resource Records Sets with private key

- Public DNSKEY is used to verify the RRSIG

- Children sign their zones with their private key
  - Authenticity of that key established by signature/ checksum by the parent (DS)

- Ideal case: one public DNSKEY distributed

RR’s and RRsets

- Resource Record:

```
Name              TTL      class  type   rdata
www.example.net.  7200        IN  A   192.168.1.1
```

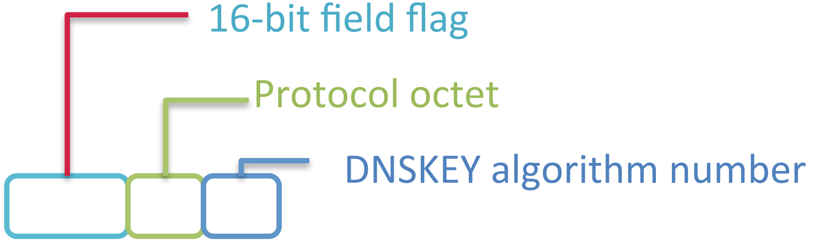
- RRset: RRs with same name, class and type:

```
www.example.net. 7200 IN  A   192.168.1.1
                          A   10.0.0.3
                          A   172.10.1.1
```

- RRsets are signed, not the individual RRs

DNSKEY

- Contains the zone’s public key
- Uses public key cryptography to sign and authenticate DNS resource record sets (RRsets).
- Example:



```
irrashai.net. IN DNSKEY 256 3 5
( AwEAAagrVFd9xyFMQRjO4DlkL0dgUCtogviS+FG9Z6Au3h1ERe4EIi3L
X49Ce1OFahdR2wPZyVeDvH6X4qlLnMQJsd7oFi4S9Ng+hLkgpm/n+otE
kKiXGZzZn4vW0okuC0hHG2XU5zJhkct73FZzbmBvGxpF4svo5PPWZqVb
H48T5Y/9 ) ; key id = 3510
```

Public key (base64)

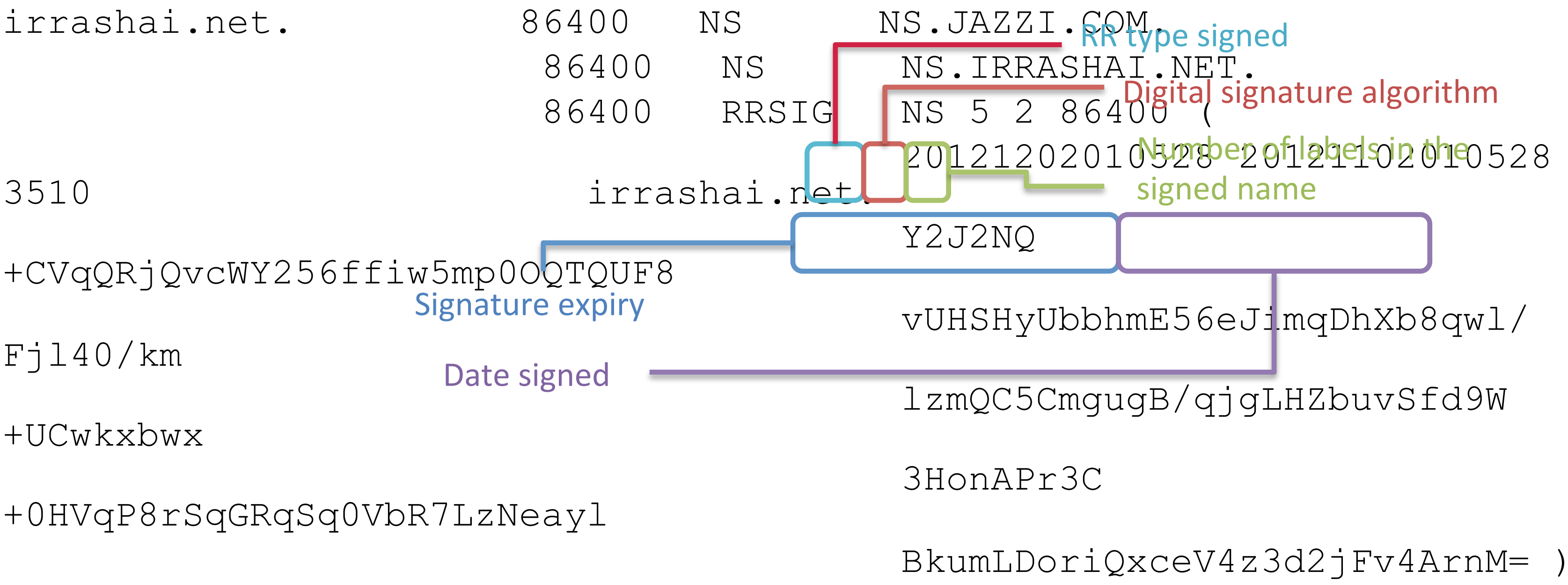
DNSKEY

- Also contains some timing metadata – as a comment in the key file

```
; This is a key-signing key, keyid 19996, for myzone.net.
; Created: 20121102020008 (Fri Nov  2 12:00:08 2012)
; Publish: 20121102020008 (Fri Nov  2 12:00:08 2012)
; Activate: 20121102020008 (Fri Nov  2 12:00:08 2012)
```

RRSIG

- The private part of the key-pair is used to sign the resource record set (RRset) per zone
- The digital signature per RRset is saved in an RRSIG record



```
irrashai.net.              86400    NS       NS.JAZZI.COM.
                            86400    NS       NS.IRRASHAI.NET.
                            86400    RRSIG    NS 5 2 86400 (
                                              20121202010528 20121102010528
3510                           irrashai.net.
                                              Y2J2NQ
+CVqQRjQvcWY256ffiw5mp0OQTQUF8
                                              vUHSHyUbbhmE56eJimqDhXb8qwl/
Fjl40/km
                                              lzmQC5CmgugB/qjgLHZbuvSfd9W
+UCwkxbwx
                                              3HonAPr3C
+0HVqP8rSqGRqSq0VbR7LzNeayl
                                              BkumLDoriQxceV4z3d2jFv4ArnM= )
```

RR type signed
Digital signature algorithm
Number of labels in the signed name
Signature expiry
Date signed

NSEC / NSEC3

- Next Secure
- Forms a chain of authoritative owner names in the zone
- Lists two separate things:
  - Next owner name (canonical ordering)
  - Set of RR types present at the NSEC RR’s owner name
- Also proves the non-existence of a domain

```
irrashai.net.  NSEC    blog.irrashai.net. A NS SOA MX
          RRSIG NSEC DNSKEY
```

NSEC / NSEC3

- “The last NSEC wraps around from the last name in the ordered zone to the first”
- Each NSEC record also has a corresponding RRSIG

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

- Used for authenticated denial-of-existence of data
  - authenticated non-existence of TYPEs and labels

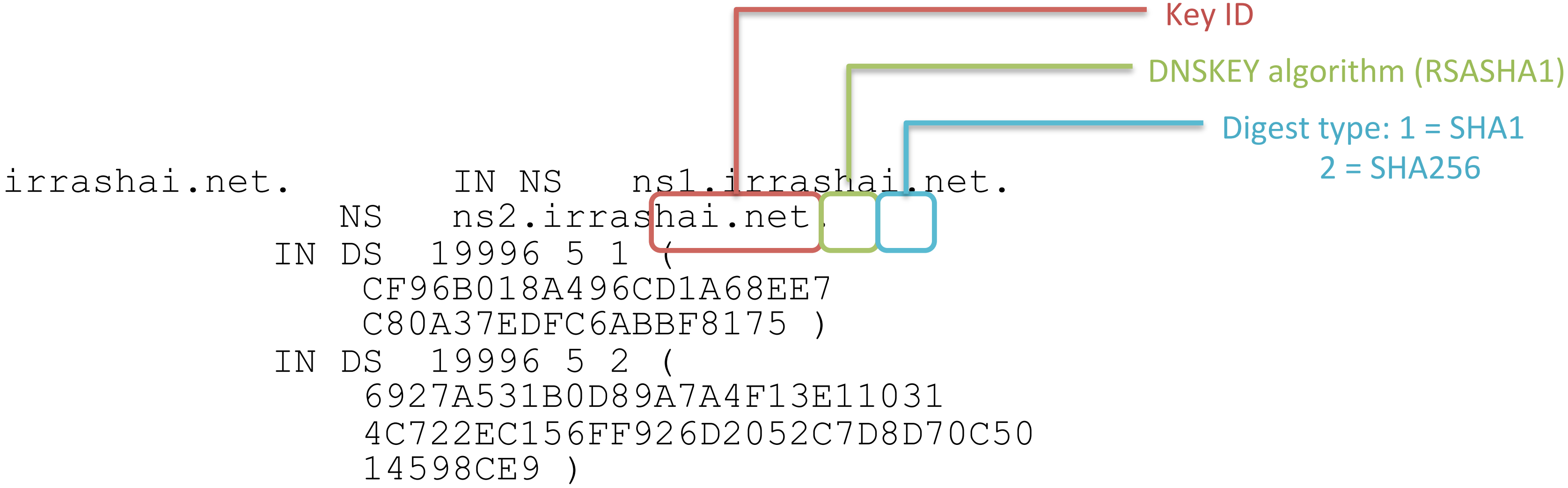
NSEC Record example

```
$ORIGIN example.net.
@  SOA        …
   NS NS.example.net.
   DNSKEY   …
   NSEC   mailbox.example.net. SOA NS NSEC DNSKEY    RRSIG

mailbox  A  192.168.10.2
         NSEC  www.example.net.  A NSEC RRSIG
 WWW     A  192.168.10.3
         TXT   Public webserver
         NSEC  example.net. A NSEC RRSIG TXT
```

Delegation Signer (DS)

- Establishes the chain of trust from parent to child zones
- Found in the parent's zone file
- In this example, irrashai.net has been delegated from .net. This is how it looks like in .net zone file



Key ID

DNSKEY algorithm (RSASHA1)

Digest type: 1 = SHA1
2 = SHA256

```
irrashai.net.          IN NS    ns1.irrashai.net.
                  NS    ns2.irrashai.net.
               IN DS   19996 5 1 (
                   CF96B018A496CD1A68EE7
                   C80A37EDFC6ABBF8175 )
               IN DS   19996 5 2 (
                   6927A531B0D89A7A4F13E11031
                   4C722EC156FF926D2052C7D8D70C50
                   14598CE9 )
```

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 childs zone
  - Not for the NS record delegating the childs zone!
  - DS **should not** be in the childs zone

Types of Keys

- Zone Signing Key (ZSK)
  - Sign the RRsets within the zone
  - Public key of ZSK is defined by a DNSKEY RR
- Key Signing Key (KSK)
  - Signed the keys which includes ZSK and KSK and may also be used outside the zone
- Trusted anchor in a security aware server
- Part of the chain of trust by a parent name server
- Using a single key or both keys is an operational choice (RFC allows both methods)

Creation of keys

- Zones are digitally signed using the private key
- Can use RSA-SHA-1, DSA-SHA-1 and RSA-MD5 digital signatures
- The public key corresponding to the private key used to sign the zone is published using a DNSKEY RR

Chain of Trust

- DNSSEC is based on trust
- Root is on top of the chain of trust.

Implementing DNSSEC

DNSSEC - Setting up a Secure Zone

- Enable DNSSEC in the configuration file (named.conf)

  -
    ```
    dnssec-enable yes; dnssec-validation yes;
    ```
- Create key pairs (KSK and ZSK)

  -
    ```
    dnssec-keygen -a rsasha1 -b 1024 -n zone champika.net
    ```
- Publish your public key
- Signing the zone
- Update the config file
  - Modify the zone statement, replace with the signed zone file
- Test with dig

Updating the DNS Configuration

- Enable DNSSEC in the configuration file (named.conf)

```
options {
      directory "…."
      dnssec-enable yes;
         dnssec-validation yes;
  };
```

- Other options that can be added later
  - `auto-dnssec { off | allow | maintain} ;`
  - These options are used to automate the signing and key rollover

Creating key pairs

- To create ZSK

```
dnssec-keygen -a rsasha1 -b 1024 -n zone
<myzone>
```

- To create KSK

```
dnssec-keygen -a rsasha1 -b 1400 -f KSK -n
zone champika.net
```

Publishing your public key

- Using $INCLUDE you can call the public key (DNSKEY RR) inside the zone file

```
$INCLUDE /path/Kchampika.net.+005+33633.key ; ZSK
$INCLUDE /path/Kchampika.net.+005+00478.key ; KSK
```

- You can also manually enter the DNSKEY RR in the zone file

Signing the zone

```
dnssec-signzone –o champika.net -t -k
  Kchampika.net.+005+00478
  db.champika.net Kchampika.net.
  +005+33633
```

- Once you sign the zone a file with a .signed extension will be created
  - db.champika.net.signed

Signing the Zone

- Sign the zone using the secret keys:

```
dnssec-signzone –o <zonename> -N
<INCREMENT> -f <output-file> -k <KSKfile>
<zonefile> <ZSKfile>
```

```
dnssec-signzone –o champika.net
db.champika.net Kchampika.net.+005+33633
```

- Once you sign the zone a file with a .signed extension will be created
  - `db.champika.net.signed`

Signing the Zone

- Note that only authoritative records are signed NS records for the zone itself are signed
  - NS records for delegations are not signed
  - DS RRs are signed!
  - Glue is not signed
- Difference in the file size
  - db.champika.net vs. db.champika.net.signed

Publishing the Zone

- Reconfigure to load the signed zone. Edit named.conf and point to the signed zone.

```
zone “champika.net” {
 type master;
 # file “db.champika.net”;
 file “db.champika.net.signed”;
};
```

Pushing the DS record

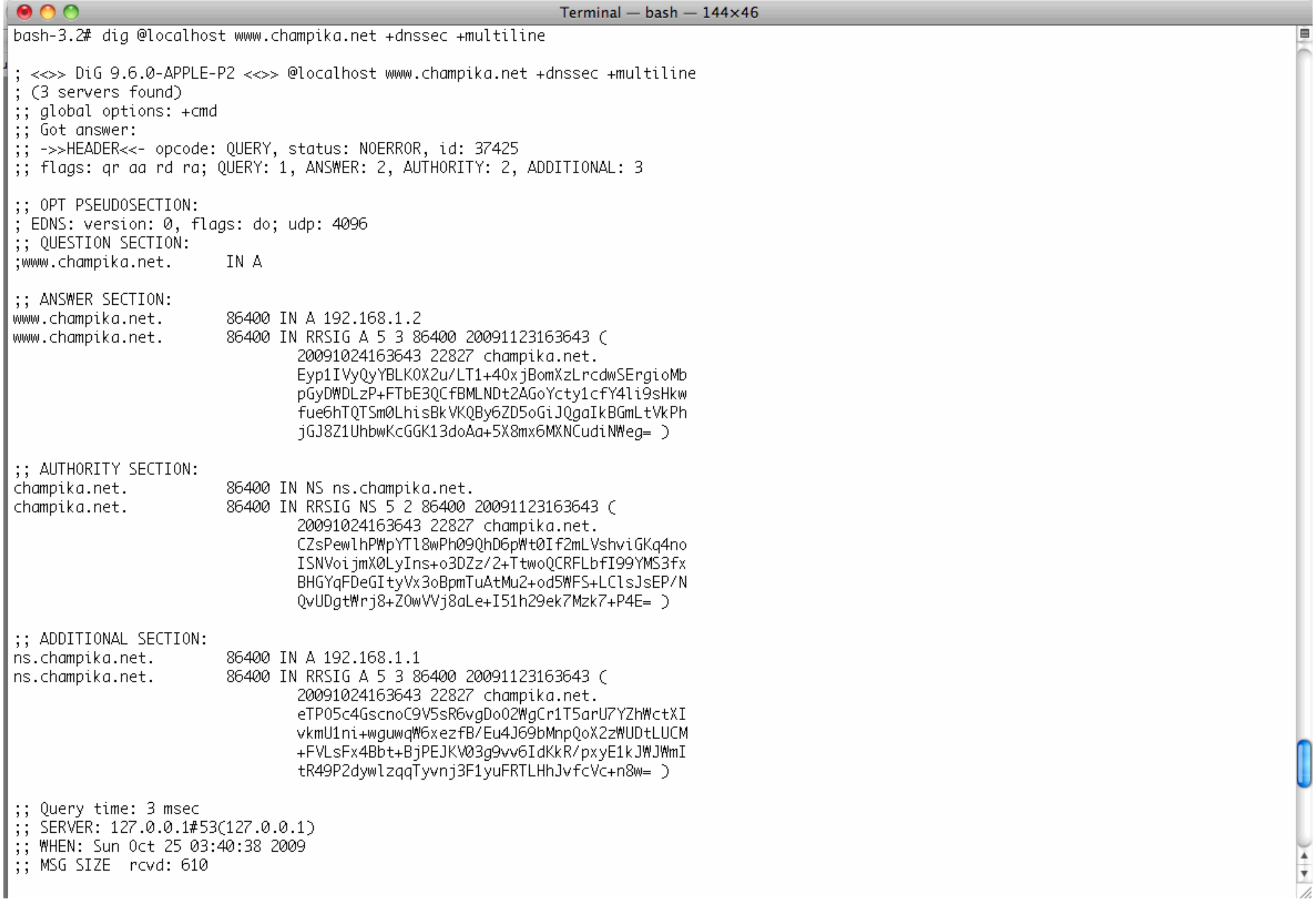
- The DS record must now be published by the parent zone.
- Contact the parent zone to communicate the KSK to them.

Testing the server

- Ask a dnssec enabled question from the server and see whether the answer contains dnssec-enabled data
  - Basically the answers are signed

```
dig @localhost www.champika.net
+dnssec +multiline
```

Testing with dig: an example



```
bash-3.2# dig @localhost www.champika.net +dnssec +multiline

; <<>> DiG 9.6.0-APPLE-P2 <<>> @localhost www.champika.net +dnssec +multiline
; (3 servers found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 37425
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 2, ADDITIONAL: 3

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;www.champika.net.      IN A

;; ANSWER SECTION:
www.champika.net.       86400 IN A 192.168.1.2
www.champika.net.       86400 IN RRSIG A 5 3 86400 20091123163643 (
                                20091024163643 22827 champika.net.
                                Eyp1IVyQyYBLKOX2u/LT1+40xjBomXzLrcdwSErgioMb
                                pGyDWDLzP+FTbE3QCfBMLNDt2AGoYcty1cfY4li9sHkw
                                fue6hTQTSm0LhisBkVKQBy6ZD5oGiJQgaIkBGmLtVkPh
                                jGJ8Z1UhbwKcGGK13doAa+5X8mx6MXNCudiNWeg= )

;; AUTHORITY SECTION:
champika.net.           86400 IN NS ns.champika.net.
champika.net.           86400 IN RRSIG NS 5 2 86400 20091123163643 (
                                20091024163643 22827 champika.net.
                                CZsPewlhPWpYTl8wPh09QhD6pWt0If2mLVshviGKq4no
                                ISNVoijmX0LyIns+o3DZz/2+TtwoQCRFLbfI99YMS3fx
                                BHGYqFDeGItyVx3oBpmTuAtMu2+od5WFS+LClsJsEP/N
                                QvUDgtWrj8+ZOwVVj8aLe+I51h29ek7Mzk7+P4E= )

;; ADDITIONAL SECTION:
ns.champika.net.        86400 IN A 192.168.1.1
ns.champika.net.        86400 IN RRSIG A 5 3 86400 20091123163643 (
                                20091024163643 22827 champika.net.
                                eTPO5c4GscnoC9V5sR6vgDoO2WgCr1T5arU7YZhWctXI
                                vkmU1ni+wguwqW6xezfB/Eu4J69bMnpQoX2zWUDtLUCM
                                +FVLsFx4Bbt+BjPEJKV03g9vv6IdKkR/pxyE1kJWJWmI
                                tR49P2dywlzqqTyvnj3F1yuFRTLHhJvfcVc+n8w= )

;; Query time: 3 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Sun Oct 25 03:40:38 2009
;; MSG SIZE  rcvd: 610
```

Questions