DNS Security

bdNOG 6

Issue Date:

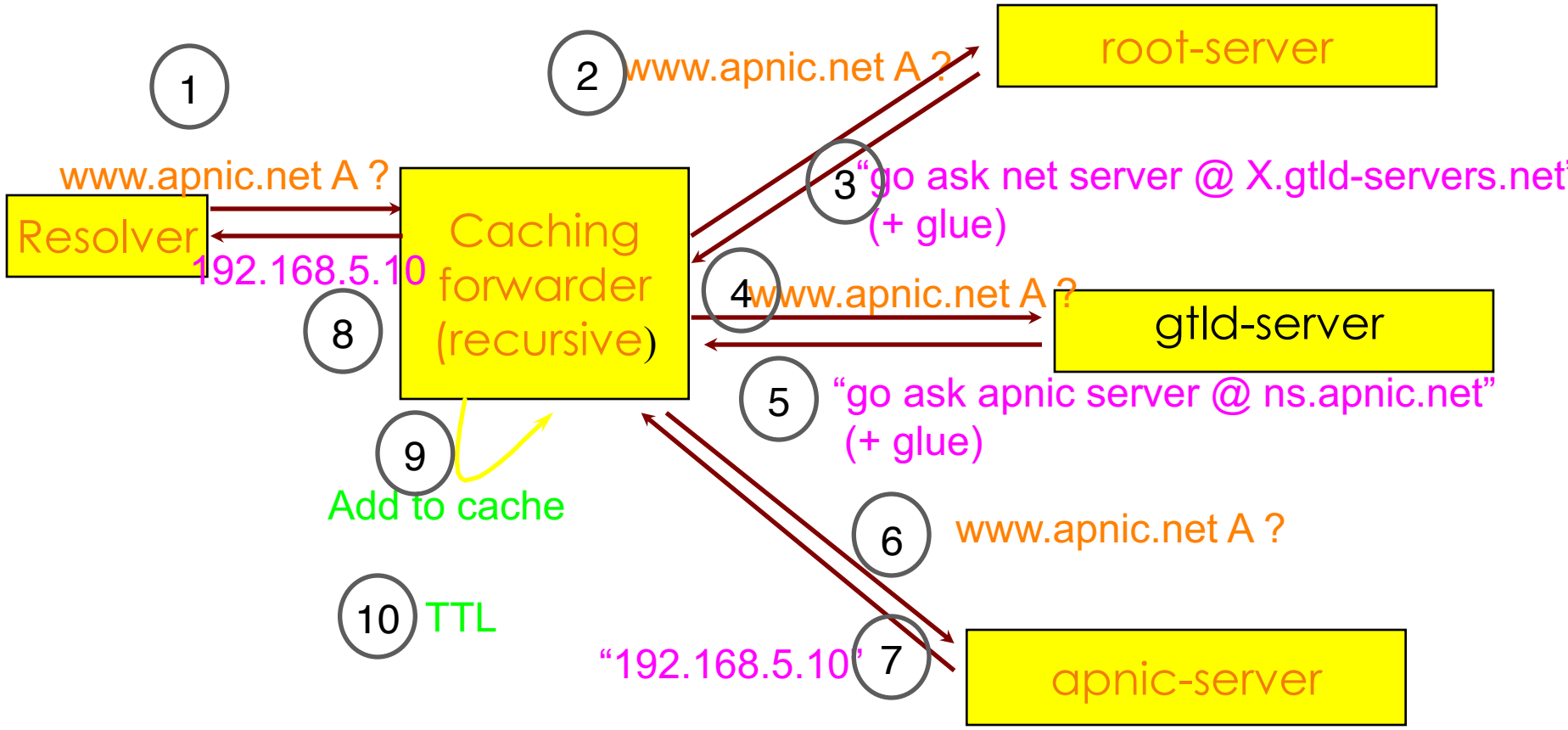
Revision:

APNIC

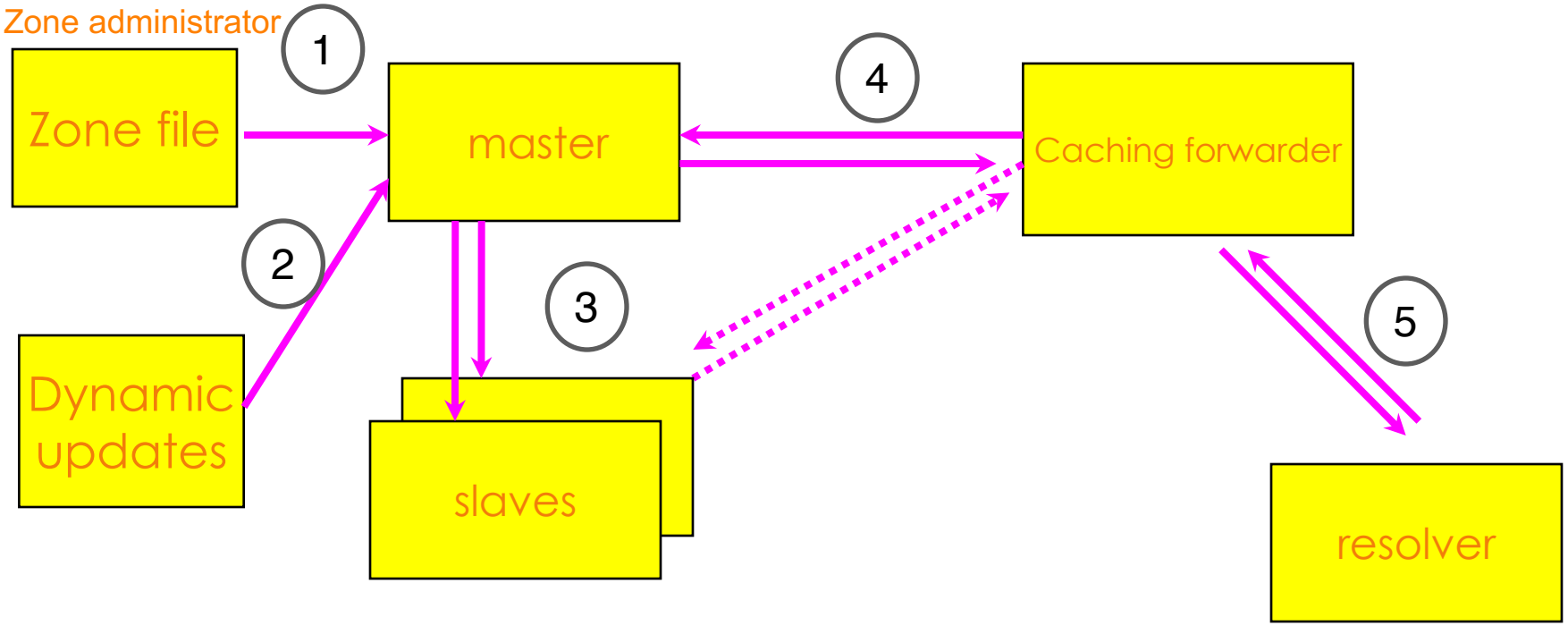
Overview

- How DNS Works
- DNS Vulnerabilities
- Securing the Nameservers
- Transaction Signature (TSIG)
- DNS Security Extensions (DNSSEC)

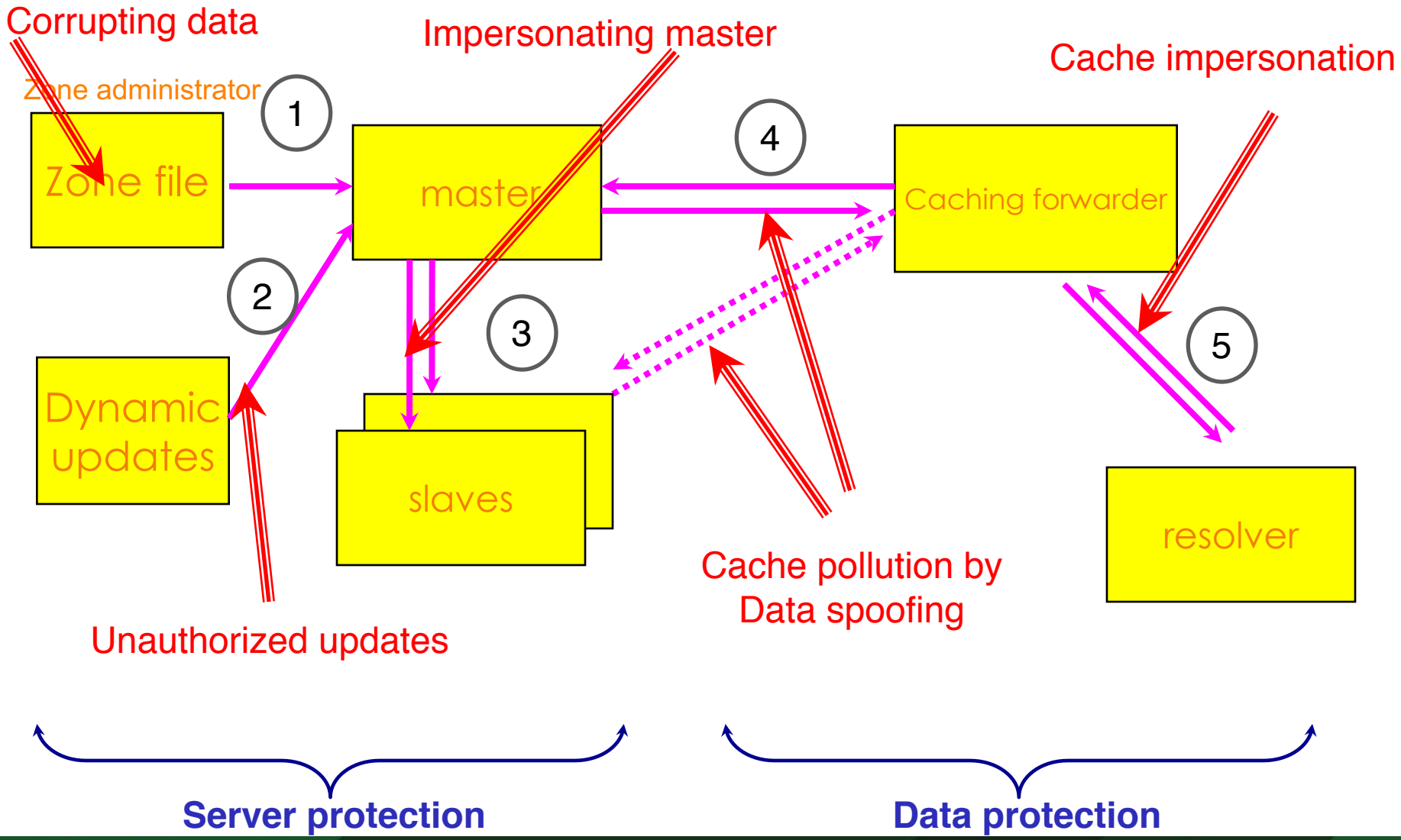
Overview: How DNS Works



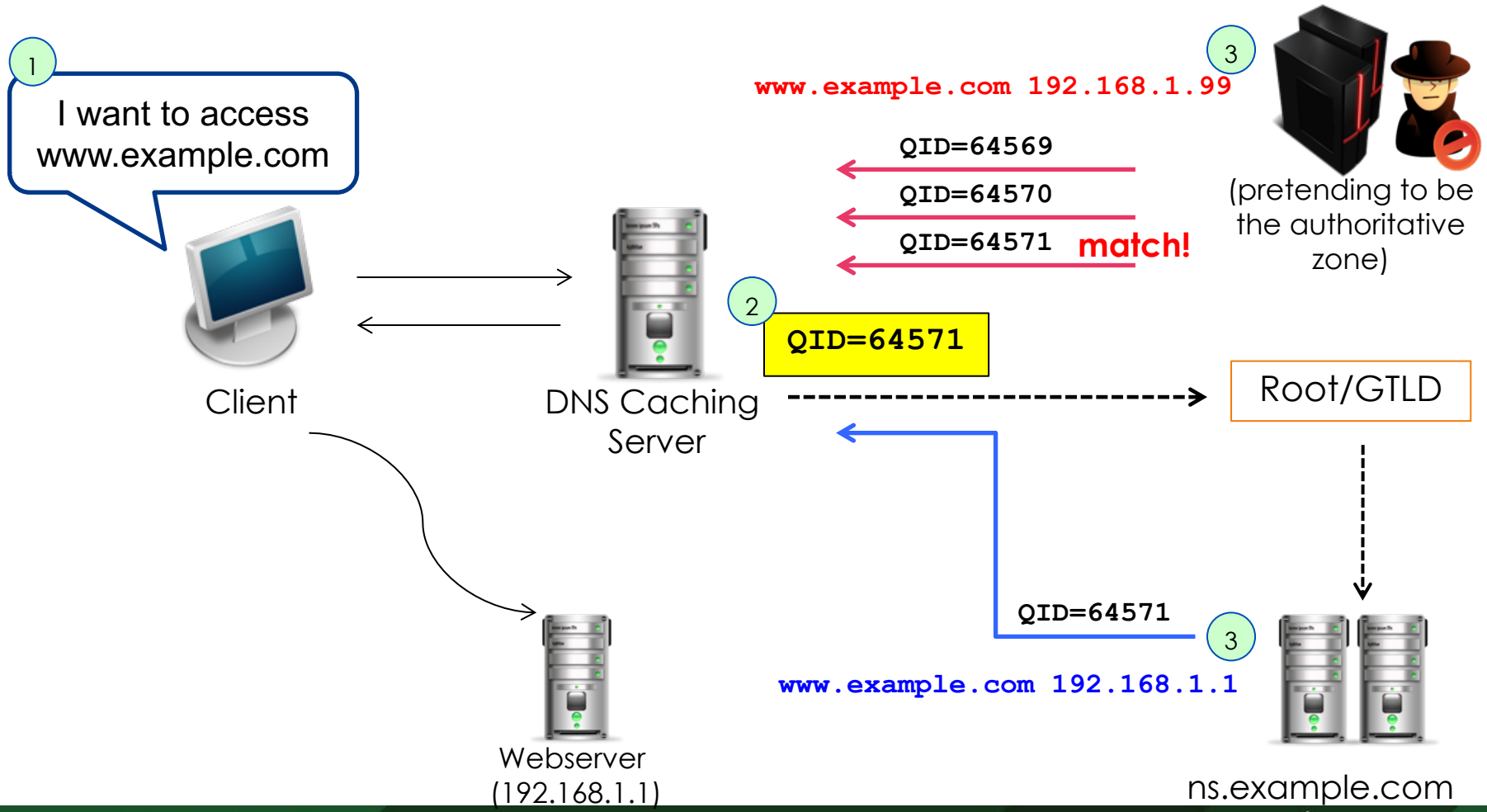
DNS Vulnerabilities



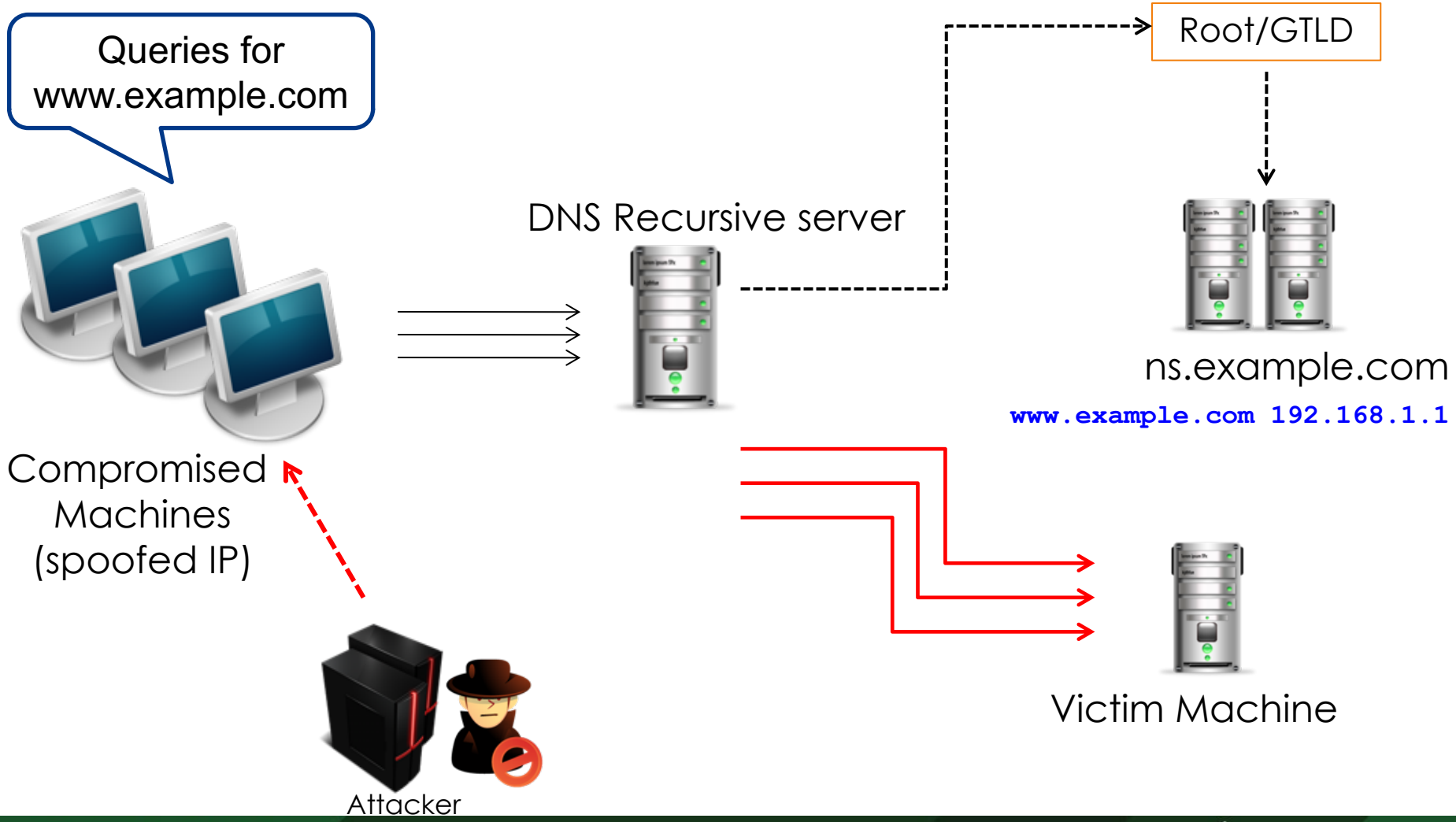
DNS Vulnerabilities



DNS Cache Poisoning



DNS Amplification



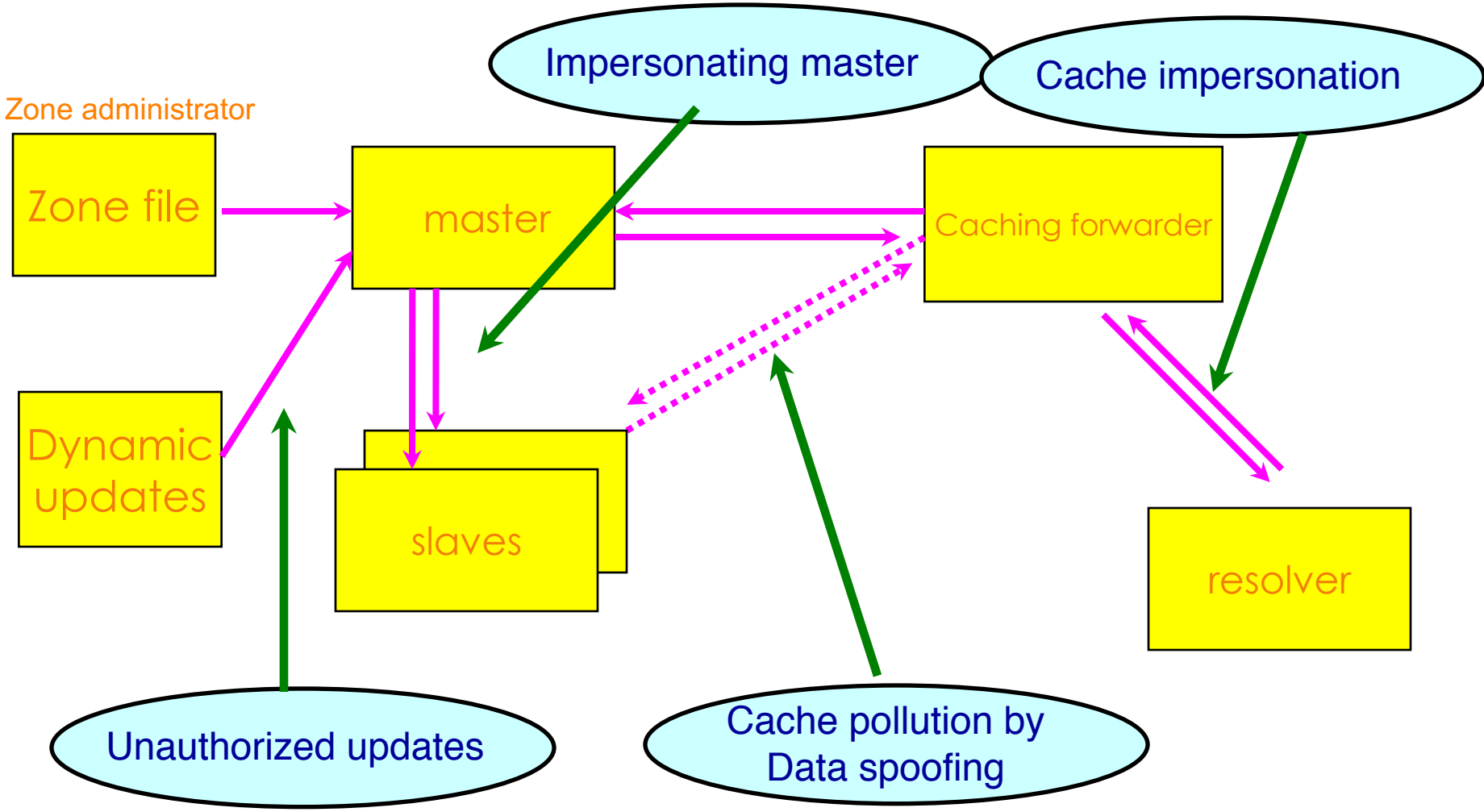
Securing the Nameserver

- Run the most recent version of the DNS software or apply the latest patch
- Restrict queries
- Prevent unauthorized zone transfers
- Run BIND with the least privilege (use `chroot`)
- Randomize source ports
- Secure the box
- Implement TSIG and DNSSEC

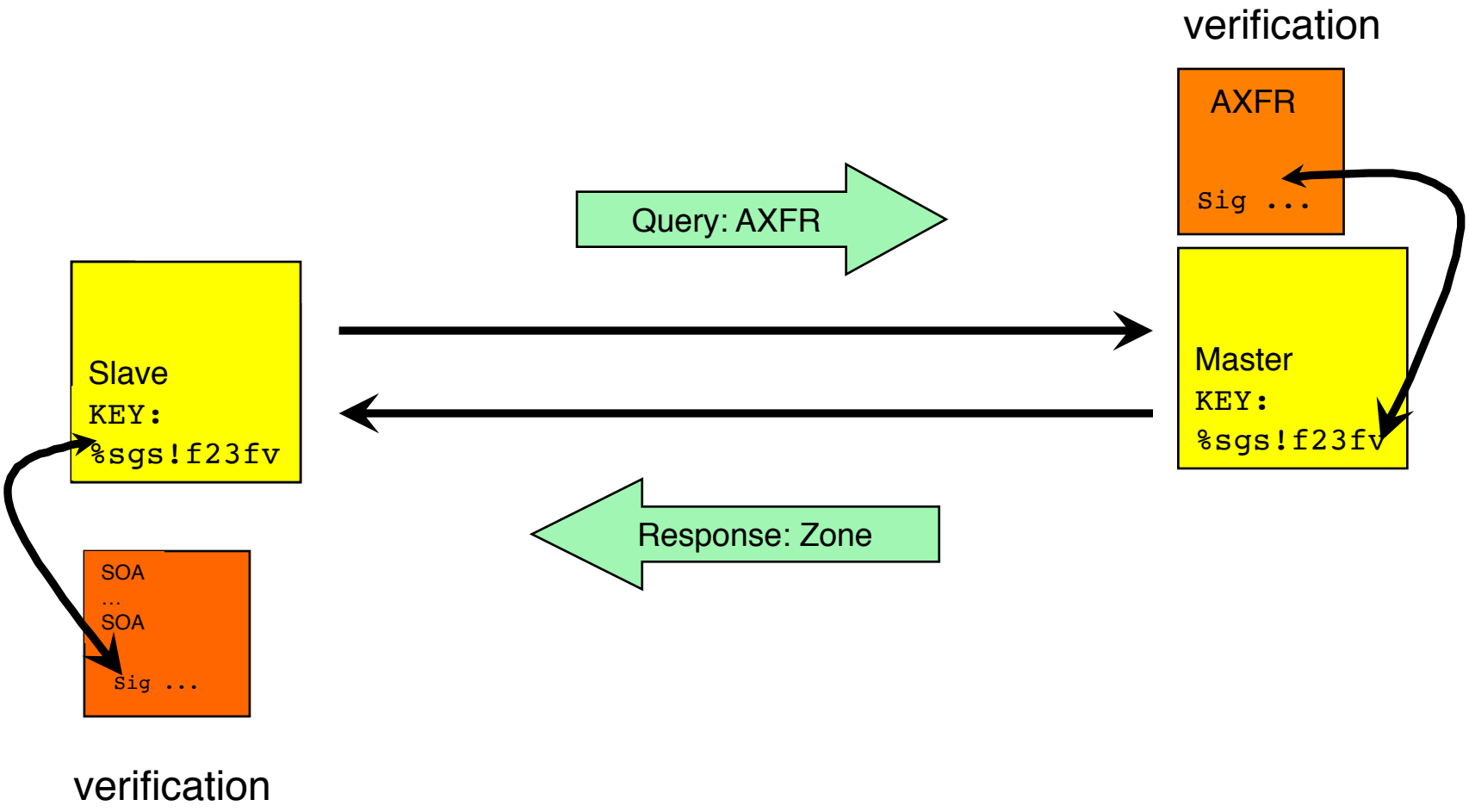
What is Transaction Signature?

- A mechanism for protecting a message from primary to secondary (and vice versa)
- Provides secure communication of queries and responses
  - Also protects zone transfers and dynamic updates
- How?
  - A keyed-hash is applied so recipient can verify the message source
- Based on a shared secret - both sender and receiver are configured with it

TSIG Protected Vulnerabilities



TSIG Example



TSIG Steps

- Generate secret
  - `dnssec-keygen -a <algorithm> -b <bits> -n host <name of the key>`
- Communicate secret
  - Transfer the key securely (ex. SSH/SCP)
- Configure the servers
  - Edit configuration file for primary and secondary
- Test
  - dig @<server> <zone> AXFR -k <TSIG keyfile>

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

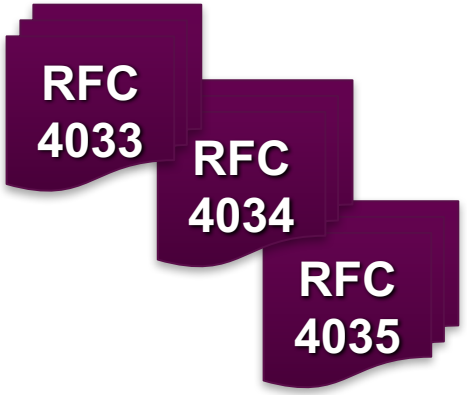
```
dig  @<server> <zone> AXFR -k <TSIG keyfile>
```

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

- A wrong key will give “Transfer failed” and on the server the security-category will log this.
- Note: TSIG is time-sensitive

What is DNSSEC?

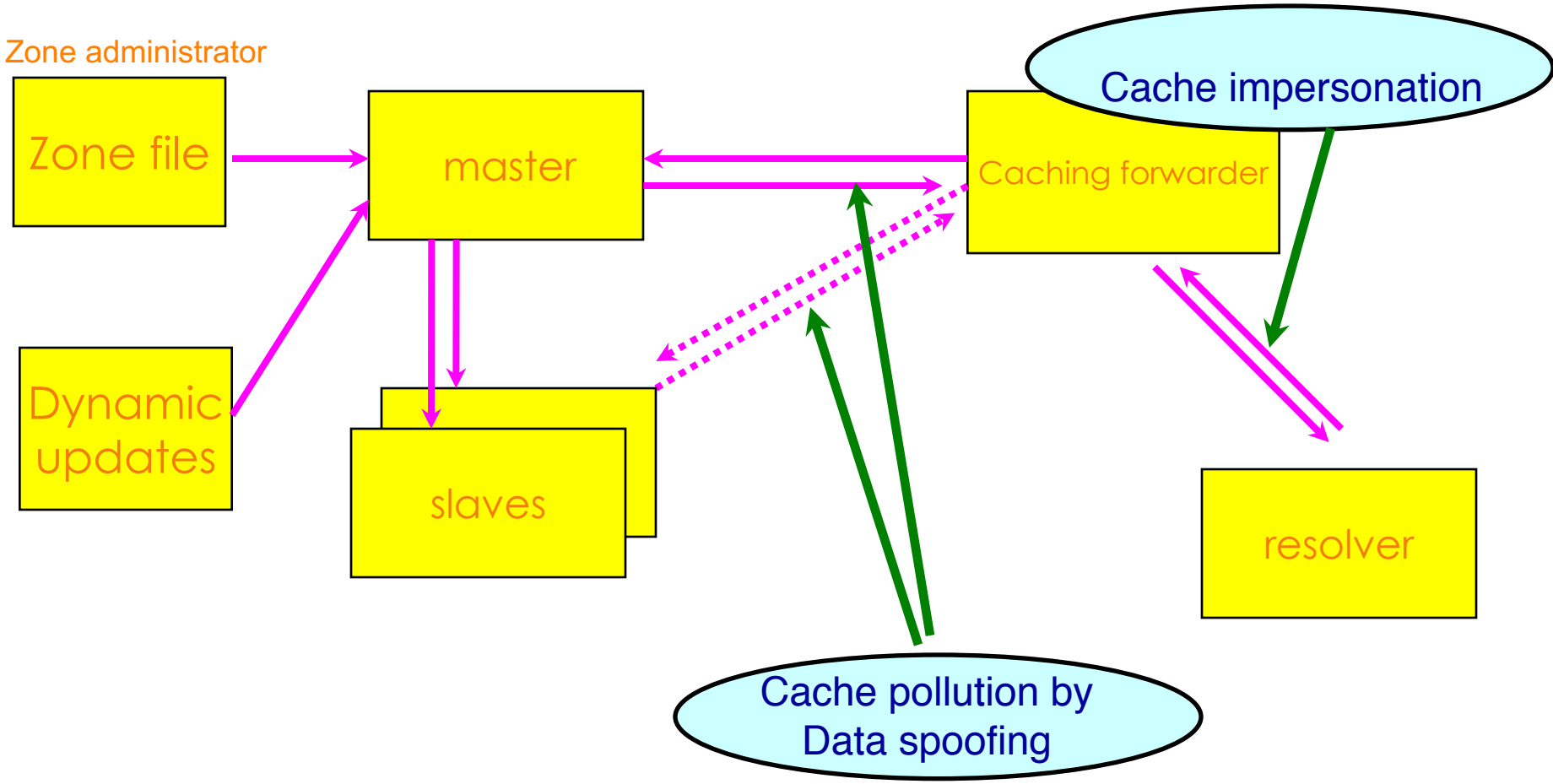
- **DNS Sec**urity Extensions
- Protects the integrity of data in DNS by establishing a chain of trust
- A form of digitally signing the data to attest its validity
- Uses public key cryptography – each link in the chain has a public/private key pair
- Guarantees
  - Authenticity
  - Integrity
  - Non-existence of a domain



How DNSSEC Works

- Records are signed with private key to prove its authenticity and integrity
- The signatures are published in DNS
- Public key is also published so record signatures can be verified
- Child zones also sign their records with their private key
- Parent signs the hash of child zone's public key to prove authenticity

Vulnerabilities protected by DNSSEC



New Resource Records

RFC 4034

Resource Record		Function
RRSIG	Resource Record Signature	Signature over RRset made using private key
DNSKEY	DNS Key	Public key needed for verifying a RRSIG
DS	Delegation Signer	Pointer for building chains of authentication
NSEC / NSEC3	Next Secure	indicates which name is the next one in the zone and which type codes are available for the current name

New Resource Records

- **RRsets** are signed with private key to prove its authenticity and integrity
- The signatures are published in DNS as **RRSIG**
- Public **DNSKEY** is also published so RRSIG can be verified
- Child zones also sign their records with their private key
- Parent signs the child zone's **DS record** to prove authenticity

Chain of Trust

- Establishes a chain of trust from parent to child zone
- How?
  - Parent does not sign child zone
  - Parent only signs a pointer to the child zone (key) – DS RECORD
- The root is on top of the chain

Creation of keys

- In practice, we use two keypairs
  - one to sign the zones, another to sign the other key
- Using a single key or both keys is an operational choice (RFC allows both methods)
- If using a single key-pair:
  - Zones are digitally signed using the private key
  - Public key is published using DNSKEY RR
  - When key is updated, DS record must again be sent to parent zone
- To address this administrative load, two keypairs will be used

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 myzone.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

Signing the Zone

- `dnssec-signzone -o myzone.net db.myzone.net Kmyzone.net.+005+33633`
- Once you sign the zone a file with a .signed extension will be created
  - `db.myzone.net.signed`
- 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.myzone.net vs. db.myzone.net.signed

Testing with Dig

`dig wiki.apnictraining.net +dnssec +multiline`

```
; <<>> DiG 9.8.3-P1 <<>> wiki.apnictraining.net +dnssec +multiline
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 52937
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

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

;; ANSWER SECTION:
wiki.apnictraining.net. 172800 IN A 202.125.96.50
wiki.apnictraining.net. 172800 IN RRSIG A 7 3 172800 20170131003633 (
                               20170103003633 4664 apnictraining.net.
                               pSDo91r7ZzhL4sz/MJQ7fS19ddMqLtb1+wtQ3h9qAVnj
                               YZVzcbS6Zmll9mq/VVPOXoc3FJqClk82e9DsUEz+RVnb
                               Iu5I640TcuN6XdbkokBTg2P7YvHJ3hy5PPjoh2bluqOC
                               FJ29mVt3pFMRyVgUVfMCcu4lrTzSSCzckIAcrbE= )

;; Query time: 41 msec
;; SERVER: 2001:dd8:b:98::123#53(2001:dd8:b:98::123)
;; WHEN: Wed Jan 25 09:19:45 2017
;; MSG SIZE  rcvd: 244
```

Testing with Dig – Reverse

```
dig -x 202.125.96.50 +dnssec +multiline
```

```
; <<>> DiG 9.8.3-P1 <<>> -x 202.125.96.50 +dnssec +multiline
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 27394
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;50.96.125.202.in-addr.arpa. IN PTR

;; ANSWER SECTION:
50.96.125.202.in-addr.arpa. 172800 IN PTR wiki.apnictraining.net.
50.96.125.202.in-addr.arpa. 172800 IN RRSIG PTR 7 6 172800 20161220054845 (
                            20161122054845 27044 96.125.202.in-addr.arpa.
                            uszEAw14s7YKtNsSt/cBJj3jhAZmPM8YHKVasvduQFNC
                            4e8K6RGOUCqEH1ZZlNdOaADHMYJg0eOxx7JnfT+RIOzU
                            qw5qKYK4Np3ArFodUUO6aWP5y1R4f7oNaBlkX/8WIICe
                            p9D8P4rYbkj5zbOITg7Dh/tAeOWFQCIrxNbJwKU= )

;; Query time: 179 msec
;; SERVER: 2001:dd8:b:98::123#53(2001:dd8:b:98::123)
;; WHEN: Wed Jan 25 09:20:54 2017
;; MSG SIZE  rcvd: 274
```