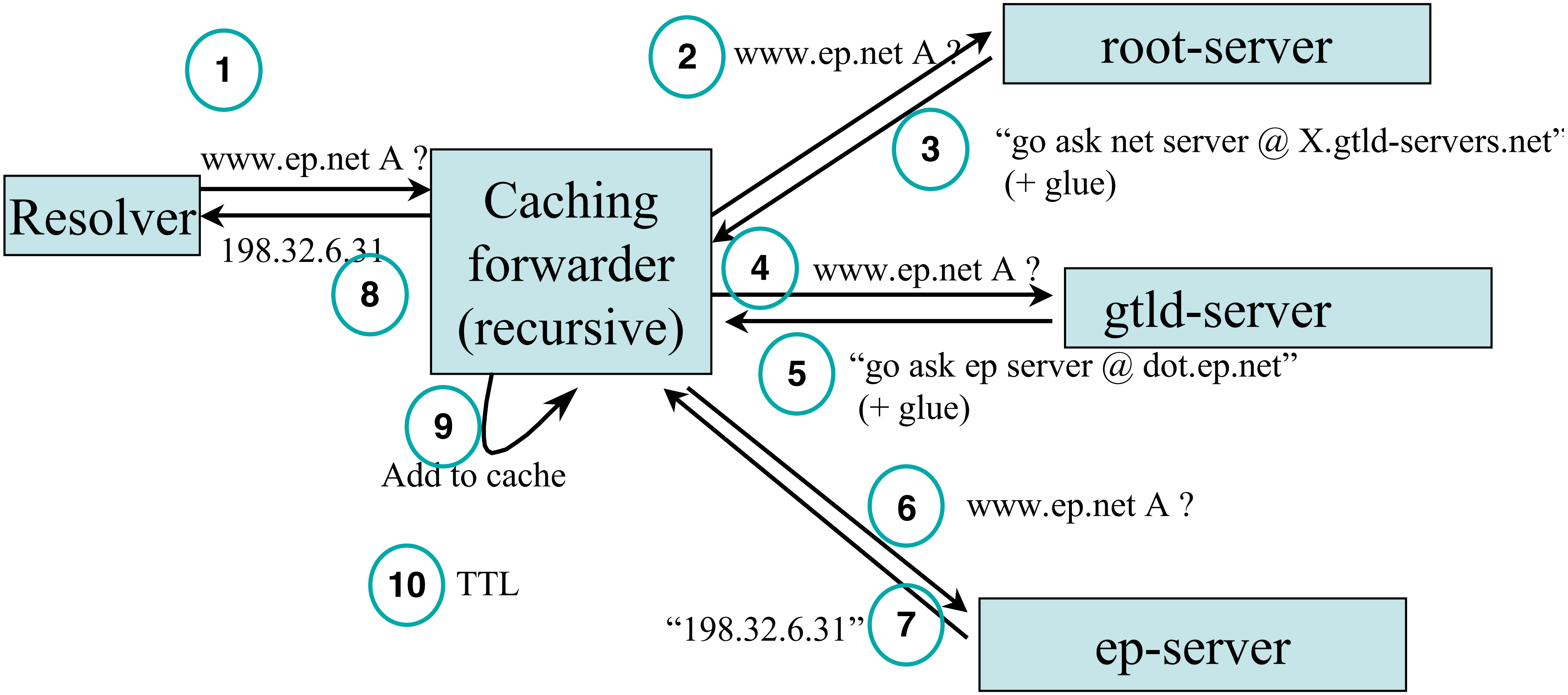
DNSSEC - TLD issues

bill manning
bmanning@ep.net

DNS Resolving

Question:

www.ep.net A



1 www.ep.net A ?

Resolver

Caching forwarder (recursive)

2 www.ep.net A ?

root-server

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

4 www.ep.net A ?

gtld-server

5 “go ask ep server @ dot.ep.net”
(+ glue)

6 www.ep.net A ?

ep-server

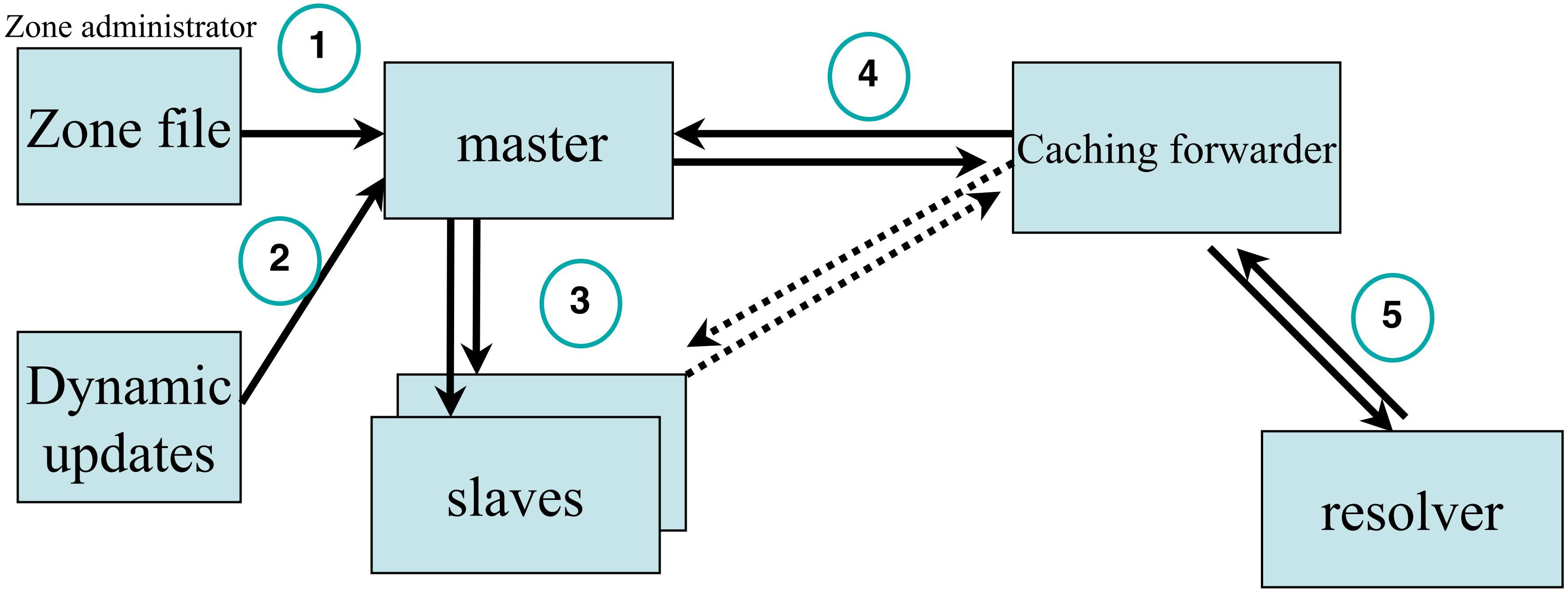
7 “198.32.6.31”

8 198.32.6.31

9 Add to cache

10 TTL

DNS: Data Flow



Zone administrator

Zone file

Dynamic updates

master

slaves

Caching forwarder

resolver

1

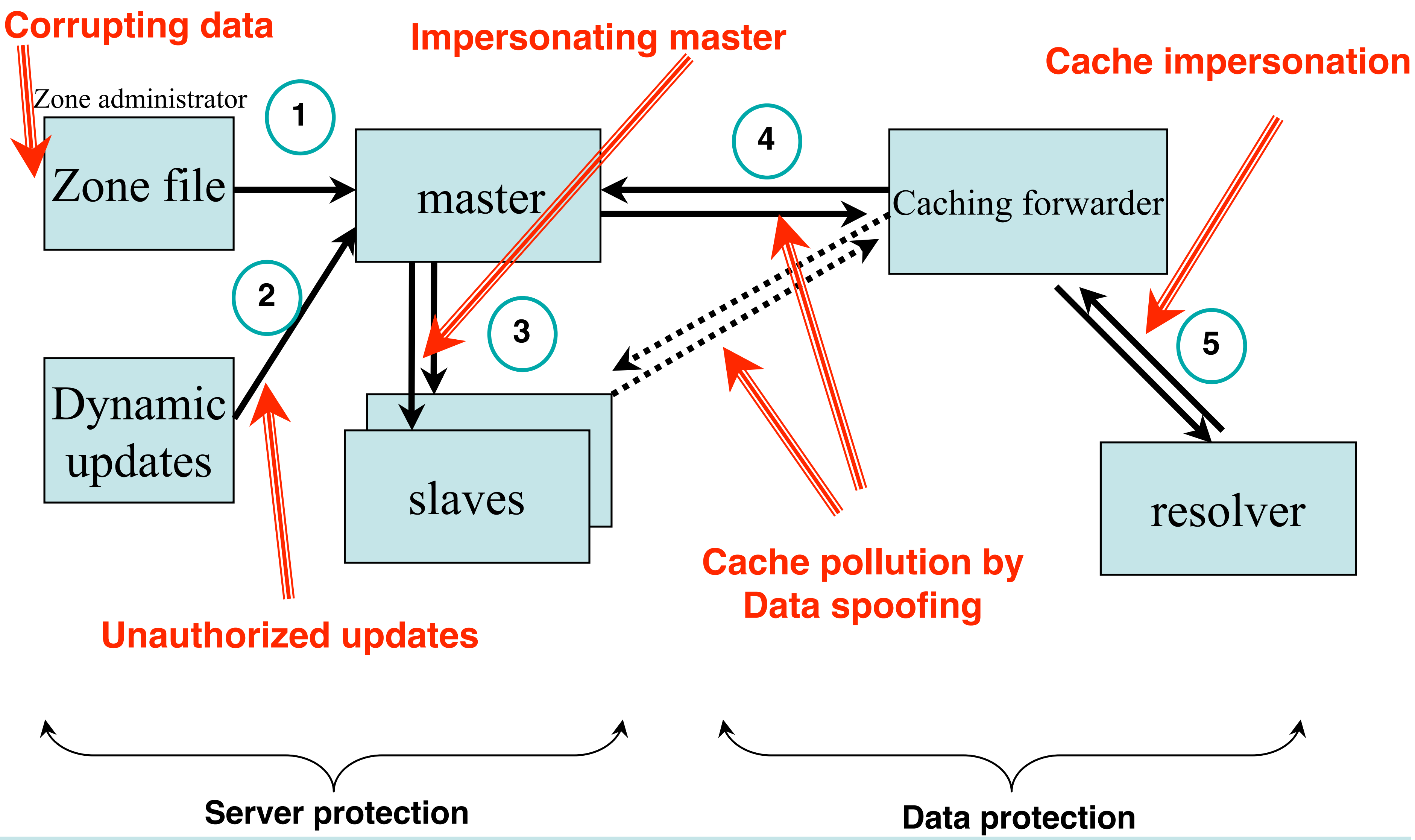
2

3

4

5

DNS Vulnerabilities



DNS Protocol Vulnerability

- DNS data can be spoofed and corrupted on its way between 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)?

Motivation for DNSSEC

DNSSEC protects against data spoofing and corruption

- DNSSEC (TSIG) provides mechanisms to authenticate servers
- DNSSEC (DNSKEY/RRSIG/NSEC) provides mechanisms to establish authenticity and integrity of data

DNSSEC Summary on 1 page

- Data authenticity and integrity by SIGning the resource records
- Public KEYs used to verify the RRSIGs
- Children sign their zones with their private key; The authenticity of their KEY is established by a SIGnature over that key by the parent (DS)
- In the ideal case, only one public KEY needs to be distributed off-band

Authenticity and Integrity of Data

- Authenticity: Is the data published by the entity we think is authoritative?
- Integrity: Is the data received the same as what was published?
- Public Key cryptography helps to answer these questions
  - signatures to check both integrity and authenticity of data
  - verifies the authenticity of signatures

Public Key Crypto Issues

- Public keys need to be distributed
- Secret keys need to be kept secret

DNSSEC Provisioning

- Registrant generates a public/private key pair for a zone
- Registrant signs the zone with the private key
- Registrant sends the zone’s public key to the registrar
- Registrar sends the registrant’s key to the registry
- Registry puts the registrant’s key hash (DS) in the TLD zone
- Registry signs the TLD zone
- Registry publishes the signed TLD zone

One plan - for com/net

- Extensions to EPP supporting DNSSEC provisoning
- Update registry DB to include DNSSEC schema
- Acquire cryptographic hardware
- Define proceses to generate and maintain keys
- Implement incremental signing process
- Update zone file generation process

Whats missing there?

- TLDs are -NOT- the apex of the heirarchy… :)
- TLD operators should become aware of what their key mgmt processes will be
- An additional process step is giving your parent keys.
  - TLDs are registrants in the root zone

Questions that need answers

- without input, ICANN will dictate how it will receive key material from the TLDs
- Are all TLDs running EPP compliant registration code?
- Are there common, sharable technqiues for key generation and storage?
- Key transmittal should not be an issue
  - it’s the public key and its just more data
  - will the key managers be the same as the zone data holders?
- Emergency/Contingency planning - fail secure or insecure?

End of presentation