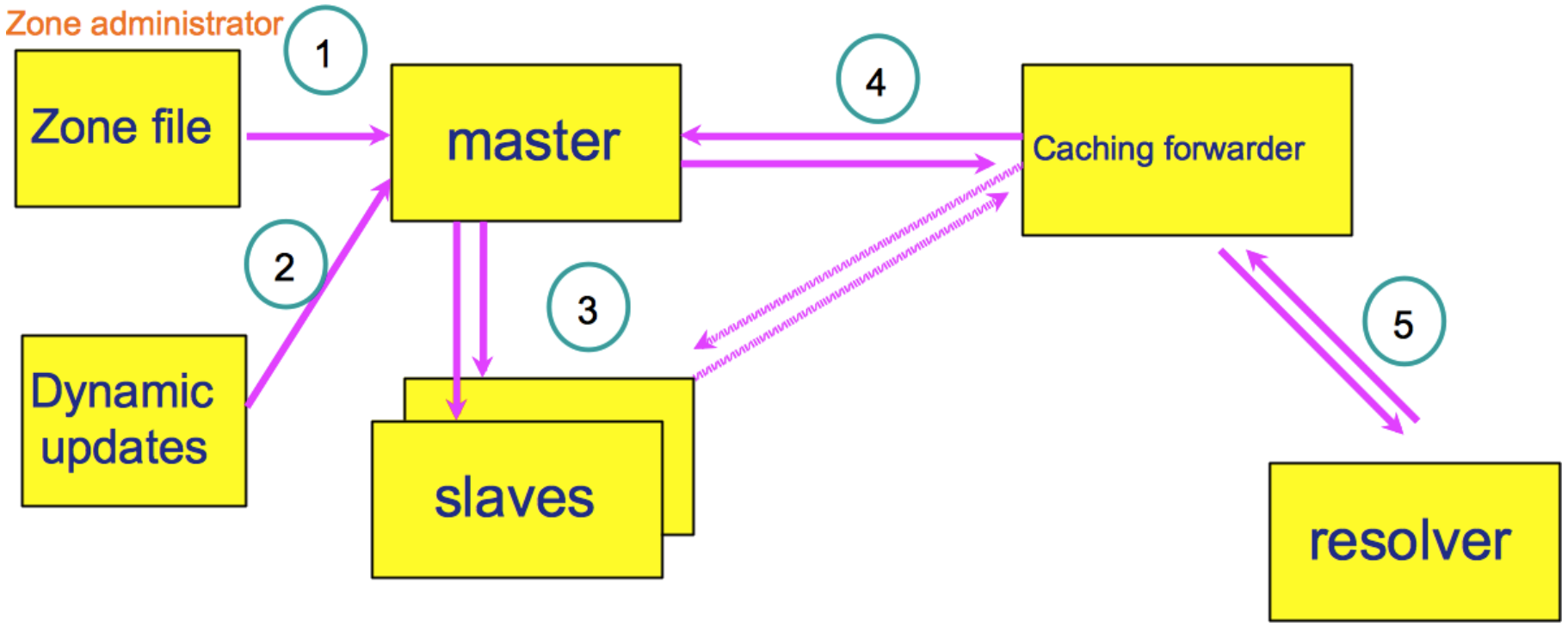
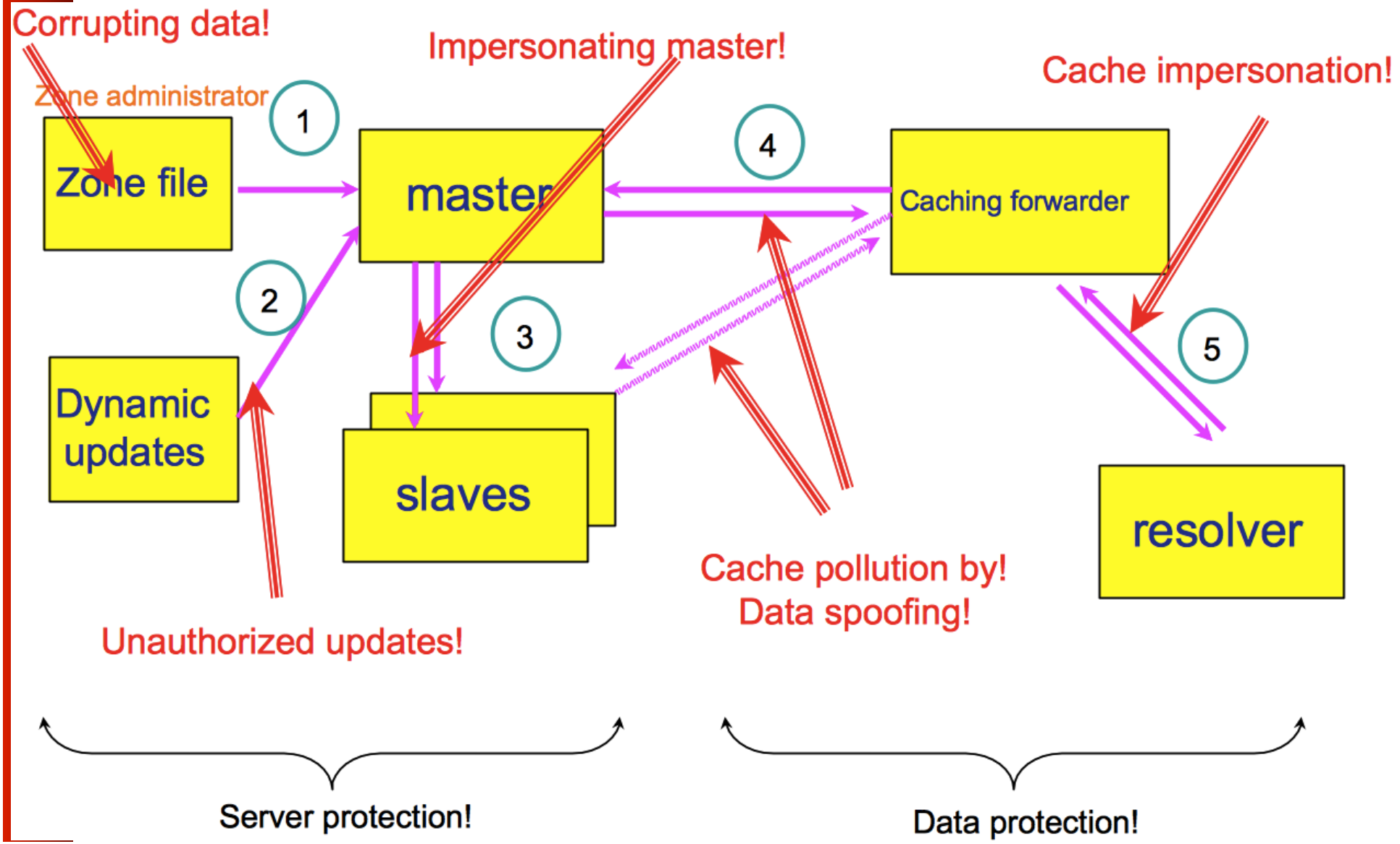
# Fiabilité et Sécurité du DNS

## Sécurité du DNS - TSIG

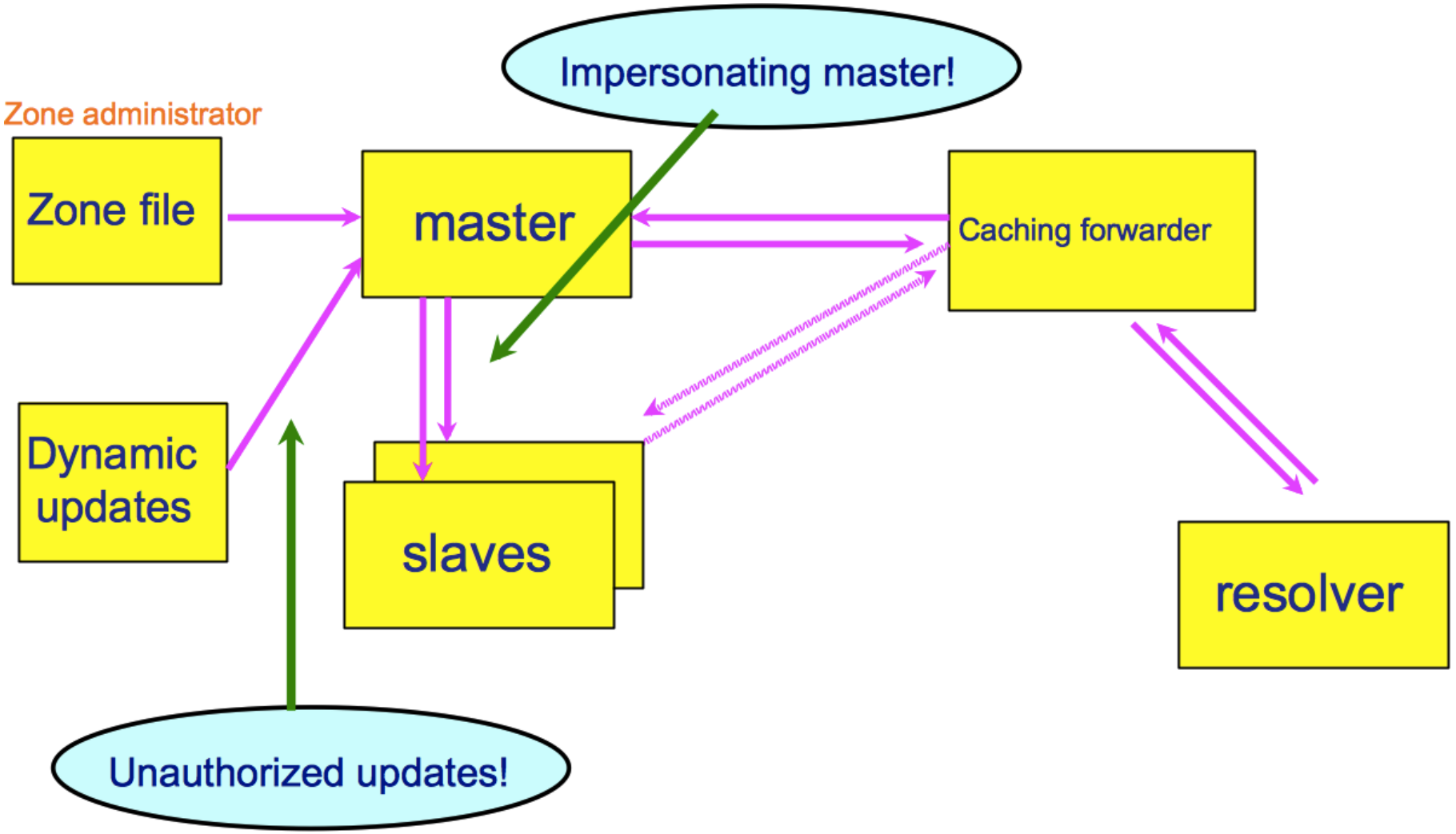
# Flux de données du DNS



# Vulnérabilités du DNS



# Vulnérabilités protégées par TSIG



# Qu’est-ce que TSIG ?

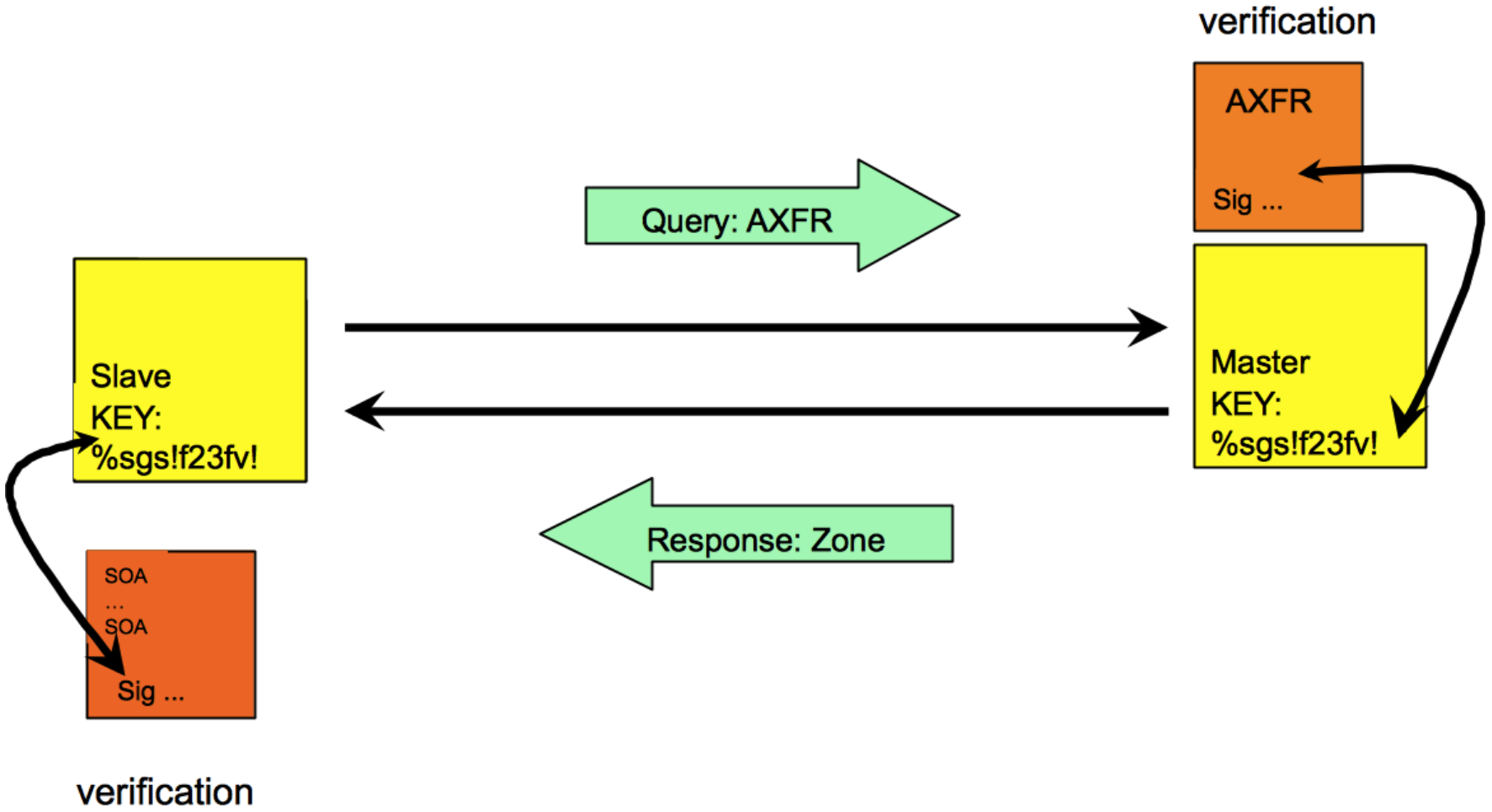
- **Transaction SIGnature**
- Un mécanisme pour protéger les communications entre les serveurs de nom (cache et autoritaire), et entre les résolveurs minimalistes (stub) et les serveurs cache.
- Une somme de contrôle cryptographique – hash (signature électronique) est ajoutée, pour que le destinataire d’un message (requête ou réponse) peut vérifier que qu’il n’a pas été manipulé, couvrant:
  - La requête/réponse DNS
  - Un horodatage
- Construite à partir d’une clé privée (DNS key)
  - L’émetteur et le destinataire doivent tous deux être configurés avec la clé

# Qu’est-ce que TSIG ?

- RFC 2845 – TSIG
- Peut aussi servir à authentifier:
  - les transferts de zone
  - les mises à jour dynamiques
  - l’authentification des serveurs cache

- Utilisé dans la configuration de BIND, pas dans le fichier de zone

# Exemple TSIG



# Étapes TSIG

1. Créer une clé
2. Communiquer la clé
3. Configurer les serveurs
4. Tester!

# TSIG – Noms et Clés

- Nom TSIG

  - Chaque clé reçoit un nom. Le nom est ce qui est transmis dans le message (pour que le destinataire sache quelle clé l'émetteur a utilisé, parmis plusieurs possibles).

- Clé secrète TSIG

  - Valeur déterminée au moment de la création de la clé
  - Souvent représentée sous forme encodée BASE64

# TSIG – Création d'une clé

- dnssec-keygen

  - Outil simple de génération (création) de clé
  - Utilisé ici pour créer une clé TSIG

dnssec-keygen -a <algorithme> -b <bits> -n host <nom-clé>

# TSIG – Création d'une clé

- Exemple

dnssec-keygen -a HMAC-MD5 –b 128 -n host ns1-ns2.grp2.net

- Ceci créera une clé, de manière similaire à celle-ci:

Kns1-ns2.grp2.net.+157+15921

- Fichiers:

  Kns1-ns2.grp2.net.+157+15921.key

  Kns1-ns2.grp2.net.+157+15921.private

# TSIG – Création d'une clé

- Les clés TSIG ne sont jamais incluses dans la zone
- Il peut y avoir confusion car les clés ressemblent à des enregistrements DNS:

ns1-ns2.grp2.net. IN KEY 128 157 nEfRx9…bbPn7lyQtE=

# TSIG – Configuration du serveur

- Configuration de la clé
  - dans named.conf, même syntaxe que pour la déclaration rndc

- Utilisation de la clé:

  - dans named.conf, ajouter:

  server x { key ….; };

… où ‘x’ est l’adresse IP du serveur DISTANT (pas le votre).

# Configuration exemple – named.conf

**Serveur primaire 10.10.0.1**

```
key ns1-ns2.grp2.net {
    algorithm hmac-md5;
    secret “Sup3rS3cr3t”;
};
server 10.10.0.2 {
      keys { ns1-ns2.grp2.net; };
};
zone “my.test.zone” {
      type master;
      file “db.myzone”;
      allow-transfer {
            key ns1-ns2.grp2.net;
      };
};
```

**Serveur secondaire 10.10.0.2**

```
key ns1-ns2.grp2.net {
    algorithm hmac-md5;
    secret “Sup3rS3cr3t”;
};
server 10.10.0.1 {
      keys { ns1-ns2.grp2.net; };
};
zone “my.test.zone” {
      type slave;
      file “db.myzone.slave”;
            masters { 10.10.0.1; };
      };
};
```

# TSIG – Tester avec dig

- Vous pouvez utiliser dig pour tester la configuration

dig @<serveur> <zone> AXFR –k <fichier contenant la clé>

ou

dig @<serveur> <zone> AXFR –y “LaCléSecrète”

- La mauvaise clé donnera un “Transfer failed”, et un message sera journalisé dans la catégorie sécurité du serveur interrogé.

# TSIG – l'horloge!

- TSIG est sensible au temps – pour éviter qu'on rejoue les paquets:
  - protection des messages (expiration après 5 minutes)
  - s'assurer que le temps est synchronisé! (NTP)
  - si on teste, penser à régler l'horloge
  - en production, utiliser NTP!

# Questions

?