IDsA

Sécurisation du DNS : Les extensions DNSsec

Atelier DNSsec
proposé par le
projet IDsA

Plan

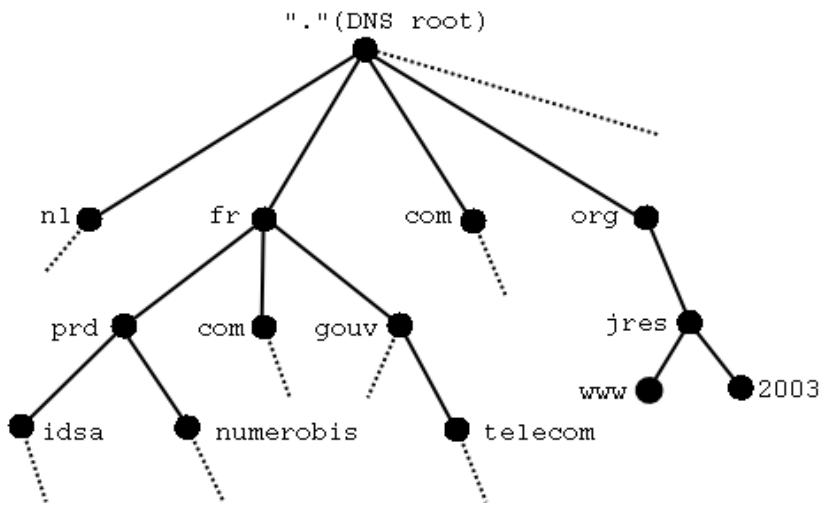
- ***Rappels synthétiques sur le DNS :***
  - Historique
  - Architecture
  - Entités
  - Information
  - Fonctionnement
- Vulnérabilités du protocole DNS
- La sécurisation du DNS : les extensions DNSsec

Historique

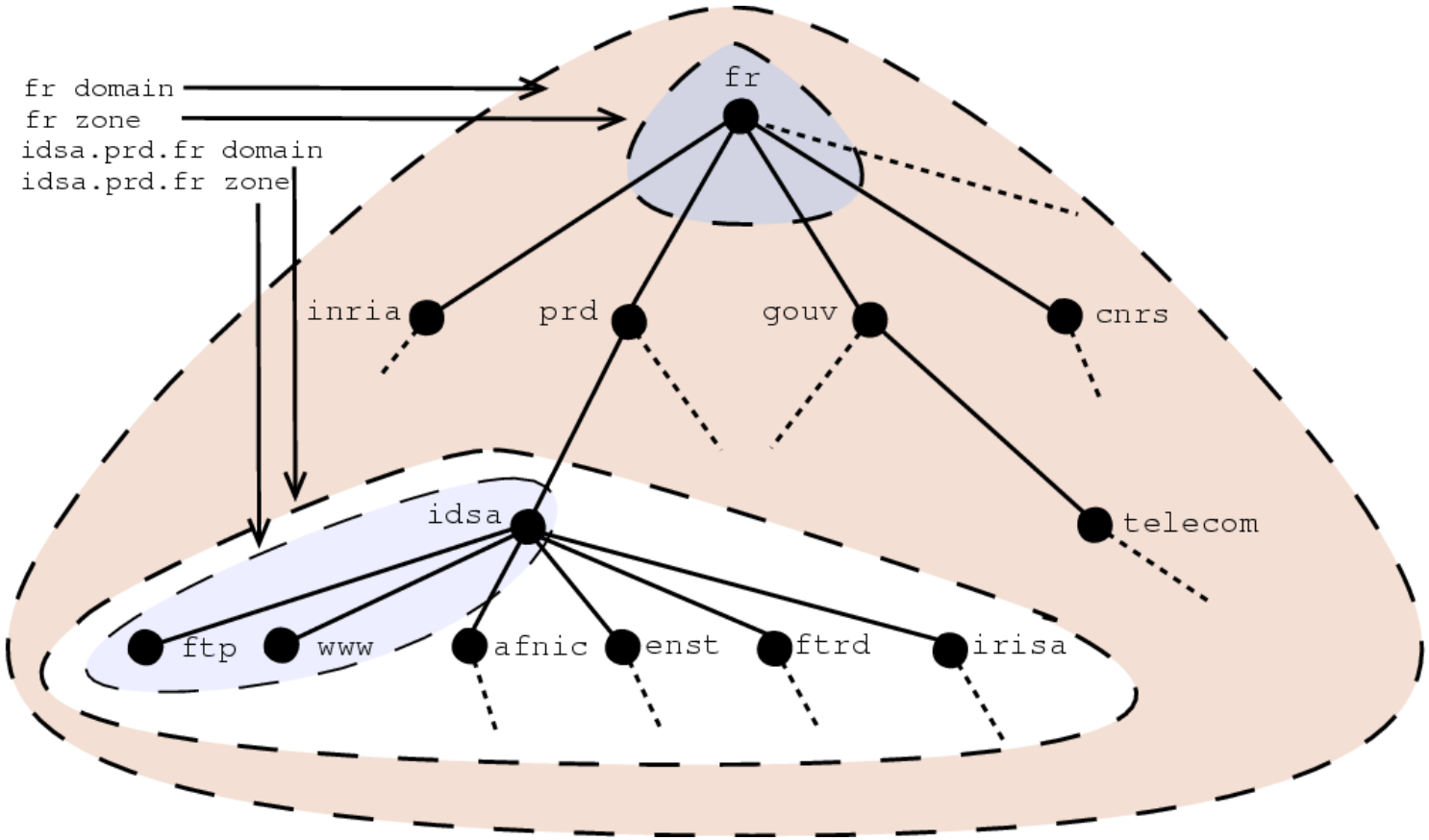
- **Jusqu'en 1984 : réseau restreint militaire/universitaire/recherche**
  - Hôtes de l'ARPAnet/Internet dans un fichier host.txt
  - Mis à jour et diffusé par le SRI-NIC
- **A partir de 1984 : croissance importante du nombre d'hôtes connectés**
  - Limites du modèle précédant
  - Un système de nommage distribué : le DNS (RFC 1034/1035, Paul Mockapetris)
  - Objectifs: performances et robustesse
- **1995 : généralisation du réseau et multiplication des usages**
  - Le DNS : un des piliers du fonctionnement de l'Internet
  - Un nouveau besoin : la sécurité
  - 1999 : Extensions de sécurité au protocole DNS, DNSsec (RFC 2535)
- **2003 :**
  - Premières expérimentations et retours d'expérience
  - Réécriture du protocole en cours (groupe de travail DNSext à l'IETF)

Le modèle DNS

- Architecture client/serveur
- La base de données DNS contient les associations entre les noms de domaine et un certain nombre d'informations (adresses IP, relais mail, serveurs de nom, etc)
  - Hiérarchique
  - Distribuée
  - Redondante

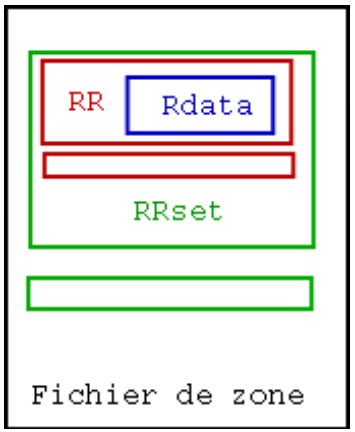


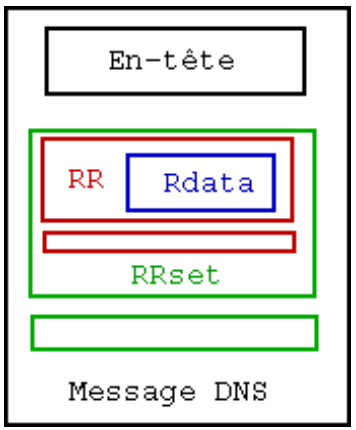
L'arbre DNS (domaines vs zones)



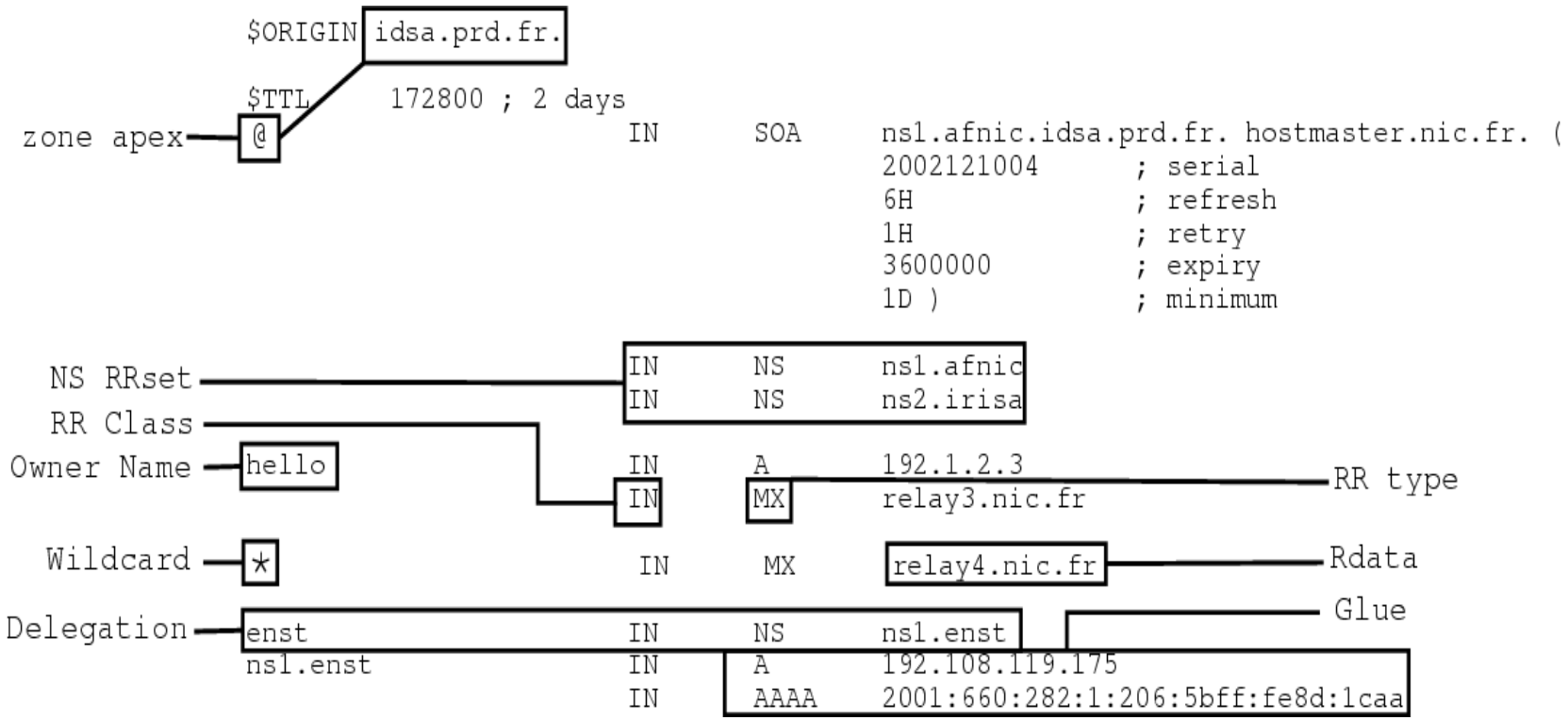
L'information DNS

- Les enregistrements DNS (ressource Records : RRs)
- Les RRsets
- Les fichiers de zone
- Les messages DNS
- La durée de vie de l'information DNS:
  - Sur les serveurs autoritaires : tant que la zone est chargée
  - Sur les serveurs caches: notion de TTL

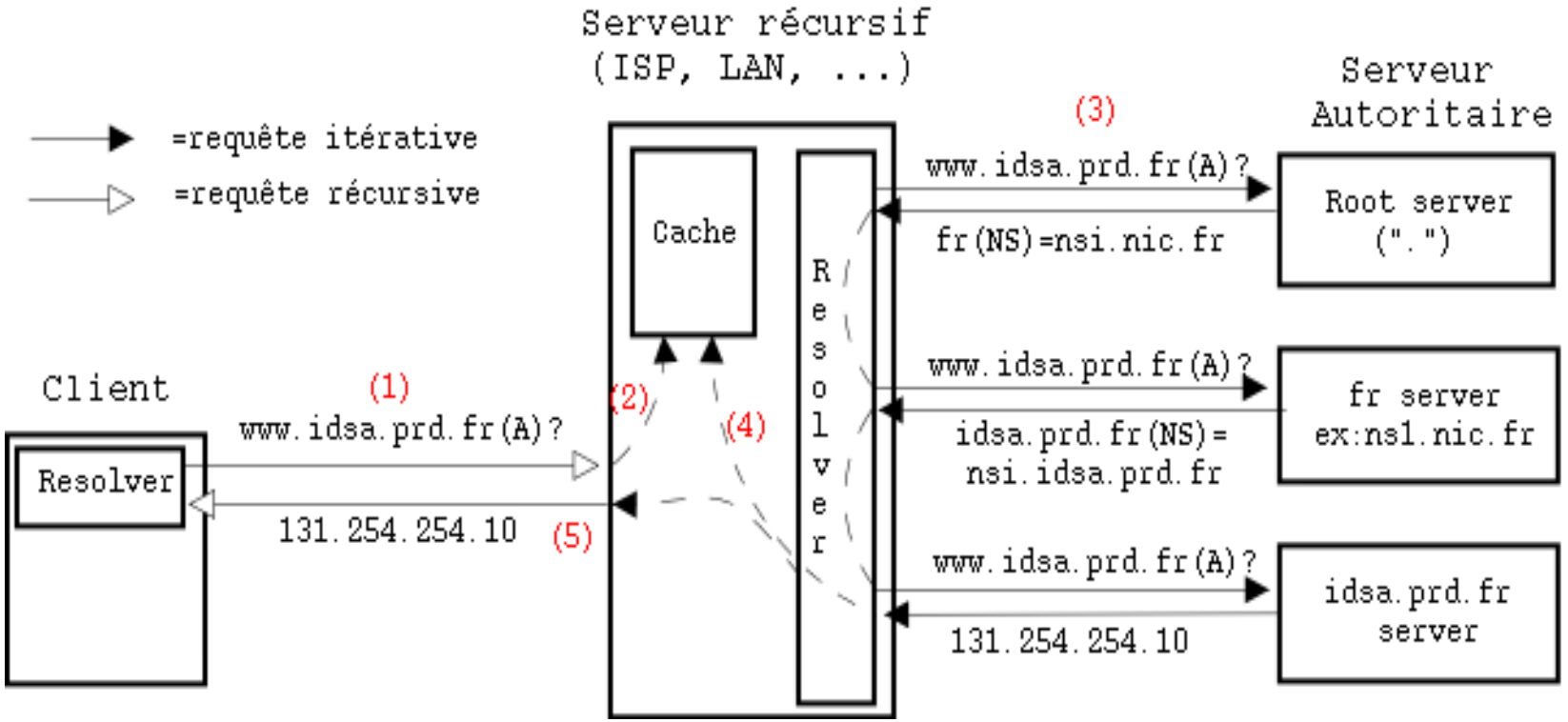




Exemple de fichier de zone



La résolution DNS



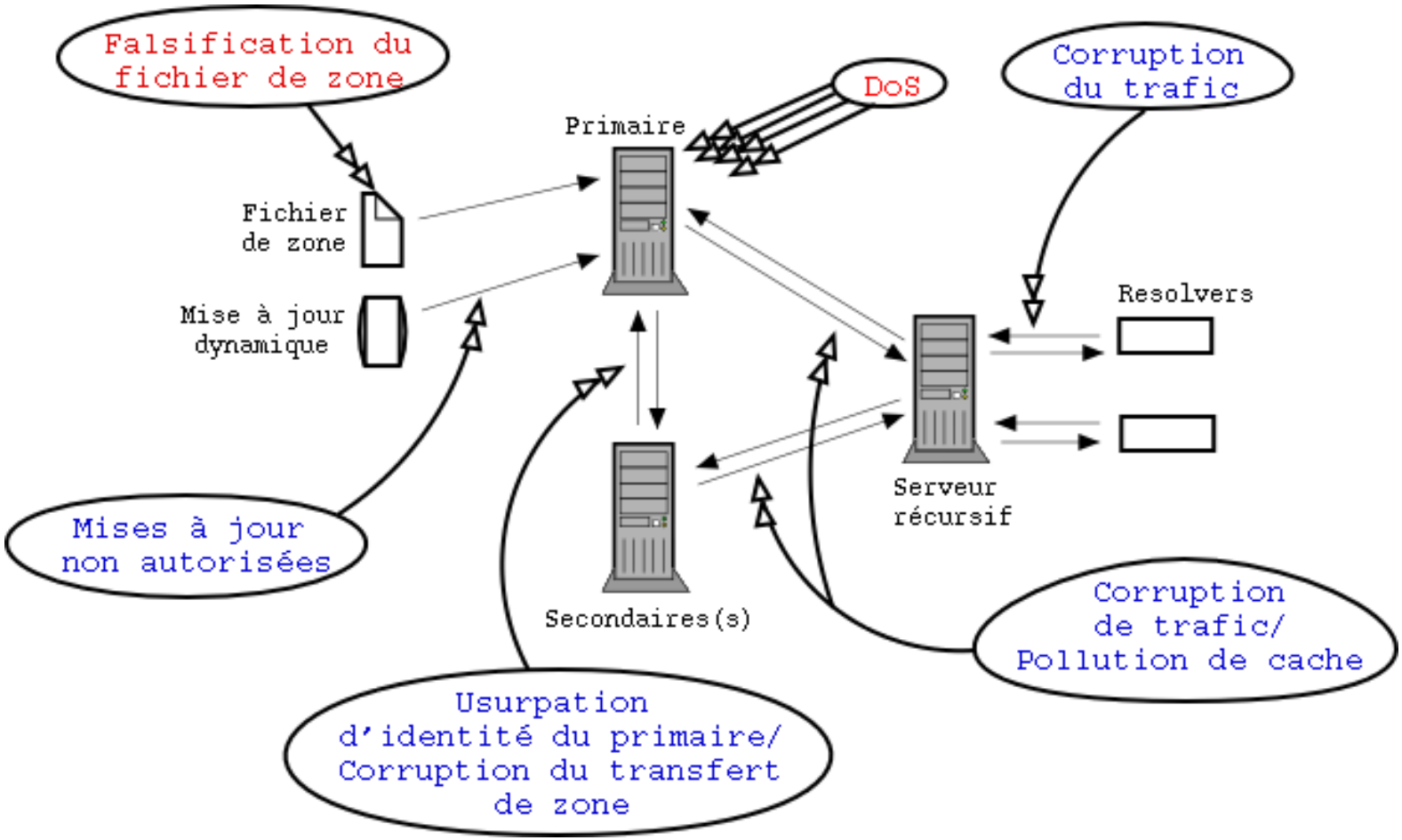
Plan

- Rappels synthétiques sur le DNS
- ***Vulnérabilités du protocole DNS***
  - ***Les vulnérabilités de l'architecture***
  - ***Le but des attaques***
  - ***Un exemple d'attaque***
- La sécurisation du DNS : les extensions DNSsec

Les failles de sécurité

- Nature publique des données/ accès universel : pas de notion de confidentialité à priori
- DNS: omniprésent et invisible lors d'une utilisation « humaine » d'Internet
- Failles spécifiques/ non spécifiques au DNS (ex : DoS)
- Disponibilité des données
- Authenticité et intégrité des données

Vulnérabilités de l'architecture DNS



But des attaques

- Perturber ou bloquer le service DNS
- Empêcher l'accès à certains équipements
  - Raisons économiques, politiques ou malicieuses
- Rediriger les utilisateurs à leur insu : préambule à une attaque plus grave
- Récupérer des informations critiques (mots de passe, emails, ...)

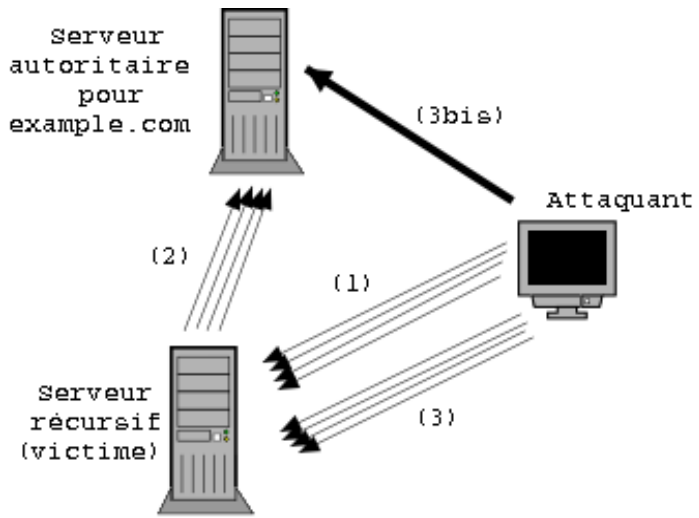
Le « DNS Spoofing »

- Spoofer = usurper l'dentité de quelqu'un
- Principe : l'attaquant répond à la place du serveur autoritaire interrogé pour tromper un utilisateur ou polluer un serveur cache
- Nécessité de connaître la question posée et l'ID associé (2octets), plusieurs modes opératoires:
  - Man in the middle : sniffer les informations sur le réseau local
  - Attaques plus évoluées s'appuyant par exemple sur des failles d'implémentation des serveurs.

    Ex: dans Bind4x, les IDs sont incrémentaux

Un exemple d'attaque

- Attaque de type « Birthday attack »
  - Birthday paradox : sur une classe de 23 élèves ou plus, la probabilité que 2 élèves soient nés le même jour est supérieure à ½



Un exemple d'attaque (2)

- Mode opératoire
  - (1): envoi de N requêtes à un serveur cache portant sur www.exemple.com associés à N IDs différents
  - (2): transfert des N requêtes vers le serveur autoritaire de exemple.com
  - (3): N réponses forgées associées à N IDs différents sont envoyées par l'attaquant
  - (3bis): DoS sur le serveur autoritaire pour le ralentir
- N=300, proba de succès du spoof >1/2

Plan

- Rappels synthétiques sur le DNS
- Vulnérabilités du protocole DNS
- *La sécurisation du DNS : les extensions DNSsec*
  - Rappels de cryptographie
  - Sécurité des transactions
  - Sécurité locale des données
  - Sécurité globale des données
  - DNSsec : le déploiement
  - Les aspects opérationnels
  - Les expérimentations en cours

Les services rendus par DNSsec

- Sécurité des données
- Sécurité des transactions
- Architecture de distribution des clefs
  - Clefs utilisées par DNSsec
  - Clefs stockées dans le DNS sécurisé utilisées pour d'autres applications (IPsec, SSH)
- Outils basés sur la cryptographie

Plan

- Rappels synthétiques sur le DNS
- Vulnérabilités du protocole DNS
- *La sécurisation du DNS : les extensions DNSsec*
  - ***Rappels de cryptographie***
  - Sécurité des transactions
  - Sécurité locale des données
  - Sécurité globale des données
  - DNSsec : le déploiement
  - Les aspects opérationnels
  - Les expérimentations en cours

Rappels de cryptographie

- Deux grandes catégories: symétrique/asymétrique
- Deux services de sécurité possibles :
  - Le chiffrement apporte la confidentialité
  - La signature apporte l'authentification de l'origine et l'intégrité des données

Cryptographie symétrique

- Cryptographie à clé secrète (partage d'un secret)
- Clé de chiffrement = clé de déchiffrement
- Clé utilisée pour signer = Clé utilisée pour vérifier les signatures
- Quelques algorithmes :
  - 3 Data Encryption Standart (DES)
  - Advanced Encryption Standard (AES)

Cryptographie asymétrique

- Basée sur des paires de clés (partie privée/partie publique)
- La partie publique permet de vérifier les signatures générées avec la partie privée
- La partie privée permet de déchiffrer les messages chiffrés avec la partie publique
- Algorithmes
  - Rivest Shamir Adelman (RSA)
  - DSA
- Exemples d'utilisation: PGP (mail), SSH

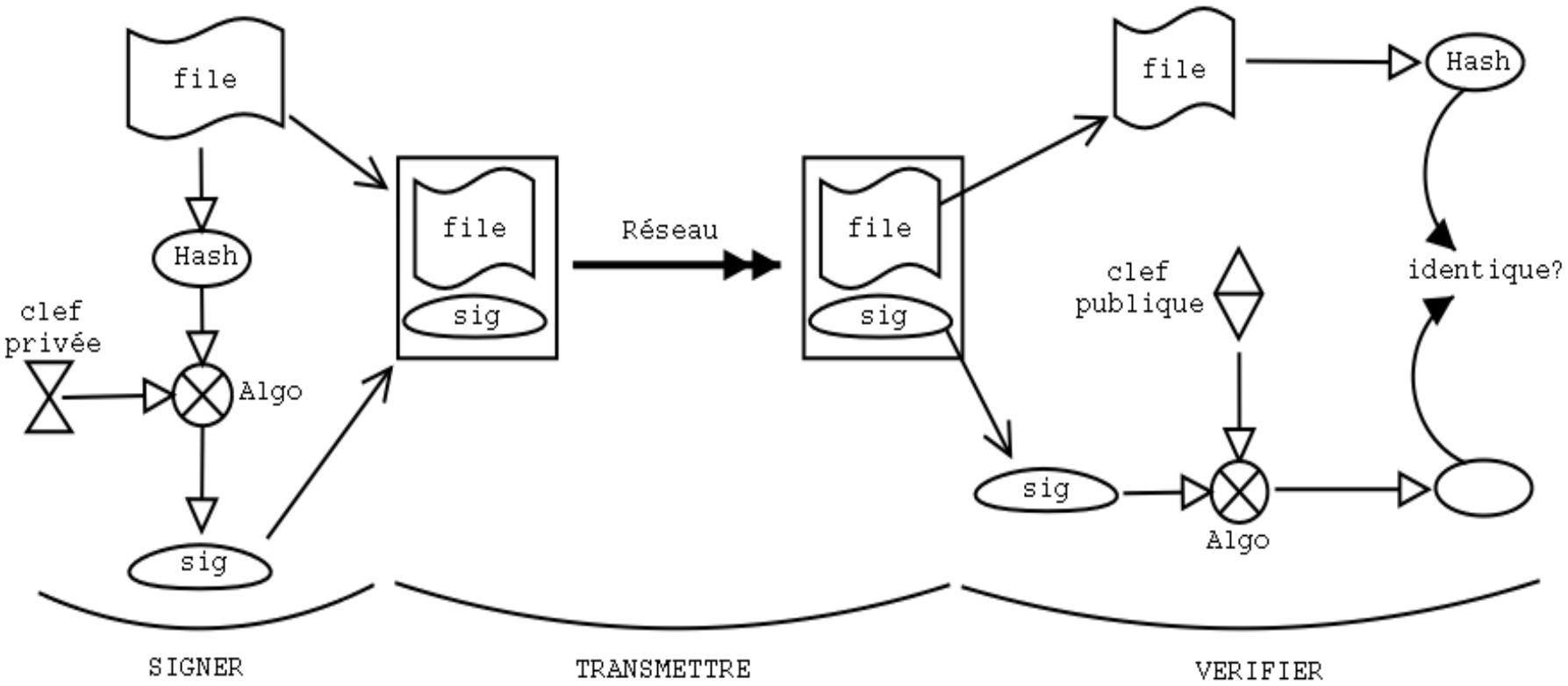
Symétrique *vs* Asymétrique

- Cryptographie symétrique :
  - Rapide
  - Nécessité de connaître son correspondant et partager un secret avec lui
  - Autant de clés distinctes que de couples de correspondants: problème de passage à l'échelle
- Cryptographie asymétrique :
  - Lent pour signer/vérifier et chiffrer/déchiffrer
  - 1 paire de clé par utilisateur
  - La connaissance de la clé publique d'un utilisateur suffit pour communiquer avec lui

Principe du hachage

- Notion d'empreinte (hash) :
  - Passer d'un fichier de taille quelconque à une séquence de taille réduite fixe (ex: 128bits)
  - Transformation irréversible
  - Toute modification du fichier génère une empreinte différente
  - Les signatures cryptographiques sont basées sur la signature des empreintes des fichiers
- Exemples d'algorithmes :
  - MD5 (Message Digest 5)
  - SHA-1 (Secure Hash Algorithm)

Rappels de cryptographie à clés publiques (signatures)



Plan

- Rappels synthétiques sur le DNS
- Vulnérabilités du protocole DNS
- *La sécurisation du DNS : les extensions DNSsec*
  - Rappels de cryptographie
  - ***Sécurité des transactions***
  - Sécurité locale des données
  - Sécurité globale des données
  - DNSsec : le déploiement
  - Les aspects opérationnels
  - Les expérimentations en cours

Sécurité des transactions : motivations

- Besoin de sécurité spécifique pour :
  - Le transfert de zones
  - Les mises à jour dynamiques (DNS Dynamic Updates)
  - Le dernier canal entre serveur récursif et client resolver (ou résolveur)
- Déployable indépendamment de DNSsec

Sécurité des transactions : TSIG

- Transaction SIGnature (RFC 2845) : meta RR (généré à la volée juste avant son utilisation et jamais stocké)
- Secret partagé (cryptographie symétrique)
- Signature d'un hash (algorithme HMAC-MD5)
- Fournit l'authenticité et l'intégrité d'un message
- Protection contre le rejeu par " Timestamp" (synchronisation NTP nécessaire)

TSIG : utilisation pour un transfert de zone

- Le serveur primaire génère une clé (outil `dnssec-keygen`). Actuellement le seul type implémenté est `HMAC-MD5`
- Le serveur primaire transmet cette clé secrète au serveur secondaire (hors-bande, PGP, scp, etc..)
- Les serveurs doivent être configurés de manière adéquate (cf. transparent suivant)

TSIG : configuration des serveurs

Master →

```
key "transfer-key" {
        algorithm hmac-md5;
        secret "sAfRkDLdld56lfD5LvD46DxlFm6flS=";
};
zone confiance.fr {
        type master;
        file "db.confiance.fr";
        allow-transfer { key transfer-key; };
}
```

Slave →

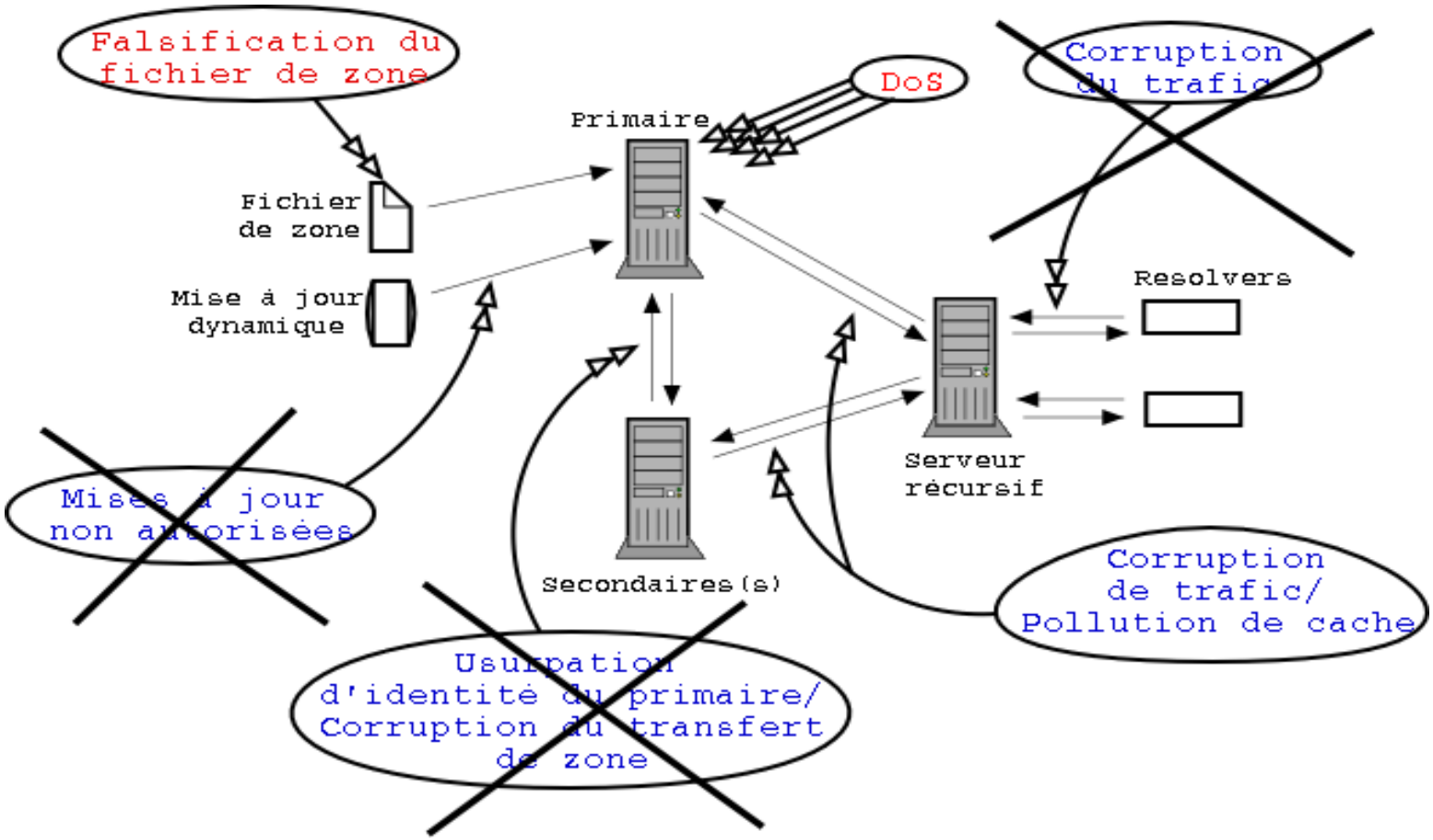
```
key "transfer-key" {
        algorithm hmac-md5;
        secret "sAfRkDLdld56lfD5LvD46DxlFm6flS=";
server 192.249.249.1 {
        keys { transfer-key; };
};
zone confiance.fr {
        type slave;
        file "db.confiance.fr";
        masters { 192.249.249.1; };
};
```

Attention : Secret, algorithme et nom affectés à la clé doivent être identiques sur Master et Slave 1

Une méthode à clés publiques pour sécuriser les transactions : SIG(0)

- RFC 2931
- Très peu implémentée et utilisée
- Principalement pour les mises à jour dynamiques
- Utilise une clef publique stockée dans le DNS

Vulnérabilités résolues par TSIG et SIG(0)



Plan

- Rappels synthétiques sur le DNS
- Vulnérabilités du protocole DNS
- *La sécurisation du DNS : les extensions DNSsec*
  - Rappels de cryptographie
  - Sécurité des transactions
  - ***Sécurité locale des données***
  - Sécurité globale des données
  - DNSsec : le déploiement
  - Les aspects opérationnels
  - Les expérimentations en cours

Les extensions DNSsec

- Historique des extensions DNSsec :
  - Mars 1999 : RFC 2535
  - Septembre 1999 : RFC 2671 (EDNS0)
  - ~2002 : collection d'Internet Drafts pour former à terme une nouvelle version de la RFC 2535 (RFC 2535bis)

  Ex : `draft-ietf-dnsext-dnssec-protocol-03`

  `draft-ietf-dnsext-dnssec-records-05`
  - Juillet 2003 : l'un de ces Internet-Drafts, DS (Delegation Signer) est accepté pour passer en RFC “Proposed Standard” (RFC xxx)
  - 2004? : 2535bis Proposed Standard
- Une seule implémentation partiellement conforme à la 2535bis : BIND9.3snapshots

Les extensions DNSsec (2)

- Sécurité des données
- Distribution de clés
- Besoins
  - Signer les données du DNS (RRsets)
  - Prouver la non existence d'une donnée
  - Vérifier l'authenticité et l'intégrité des données grâce au contrôle des signatures associées

Rappels de sécurité

- **Authentification**: identité de l'emetteur non usurpée
- **Intégrité**: non altération des données garantie
- Confidentialité: garantir le secret des données transmises
- **Protection contre le déni d'existence** : prouver la non-existence d'une donnée

- En souligné, les services apportés par DNSsec

Niveau de sécurité local (côté serveur)

- Chaque zone génère un ensemble de paires de clefs (partie privée/partie publique)
- Les clefs sont associées à la zone et non aux serveurs
- Les parties privées des clefs signent les informations (RRsets) faisant partie intégrante de la zone.
- Certaines informations ne sont pas signées :
  - Les points de délégation
  - Les glues

Nouveaux RRs pour signer les zones

- Nécessité de créer de nouveaux objets pour signer les zones
- Ces objets doivent être au format RR pour rester cohérents avec le DNS originel
- Les signatures sont stockées dans le fichier de zone en compagnie des données qu'elles authentifient : RR SIG
- Les parties publiques des clefs sont publiées dans le fichier de zone et peuvent faire l'objet de requêtes DNS standard : RR KEY
- En revanche seul le signataire d'une zone doit avoir connaissance de la partie privée des clés

Le RR KEY

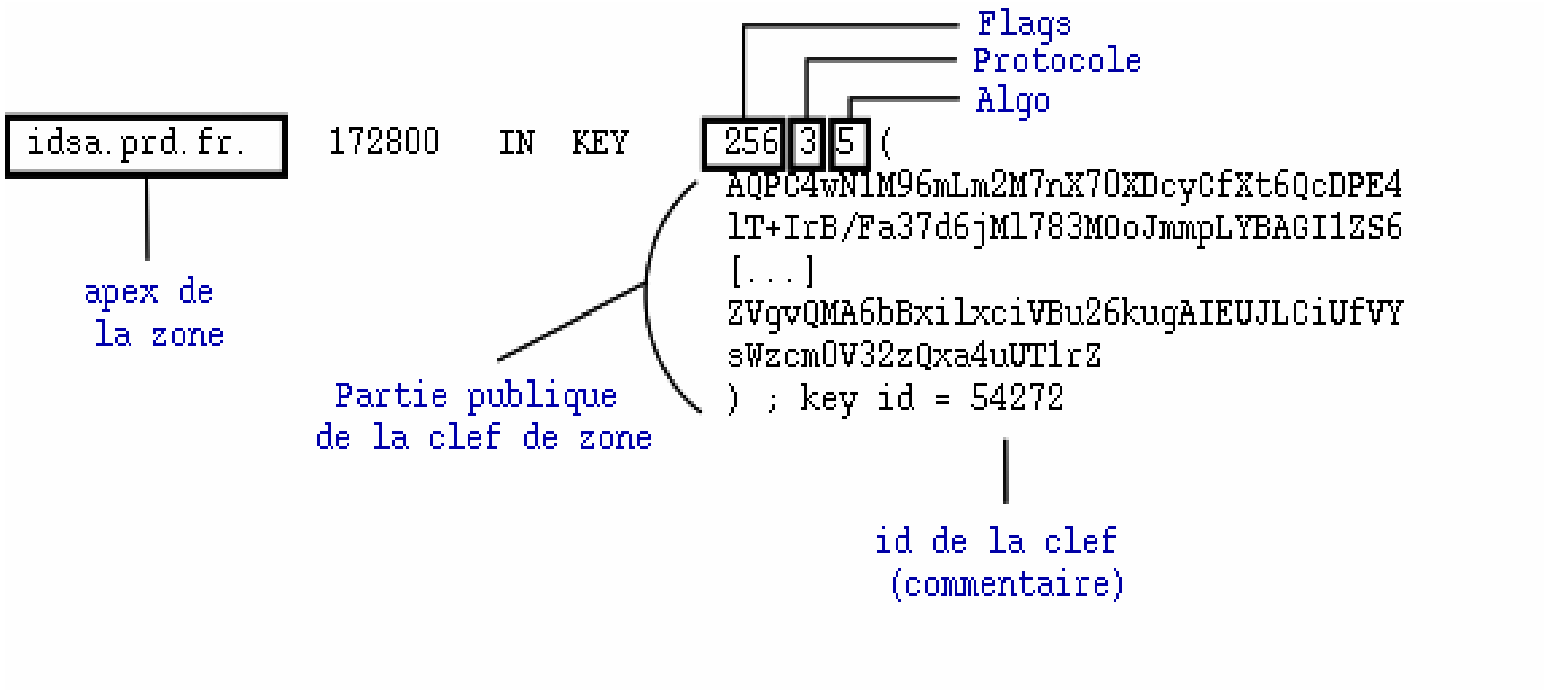
- KEY suit le formatage RR traditionnel :
  - Une partie commune à tous les RRs (nom, TTL, type, classe)
  - Une partie spécifique: son RDATA décrit ci-après

Flags (2 octets)	Protocole (1 octet)	Algorithme (1 octet)
Clé publique		

Description du KEY RDATA

- Flags: permet de distinguer les clés de zone des clés utilisées pour d'autres services DNSsec (ex: SIG(0))
- Protocole : donne la possibilité de stocker des clés utilisées par d'autres protocoles (IPsec par exemple)
- Algorithme : le seul qui est obligatoirement implémenté est RSA-SHA1
- A toute clé est associé un ID
  - Remarque : deux clés distinctes peuvent avoir le même ID

Format KEY : exemple



Le RR SIG

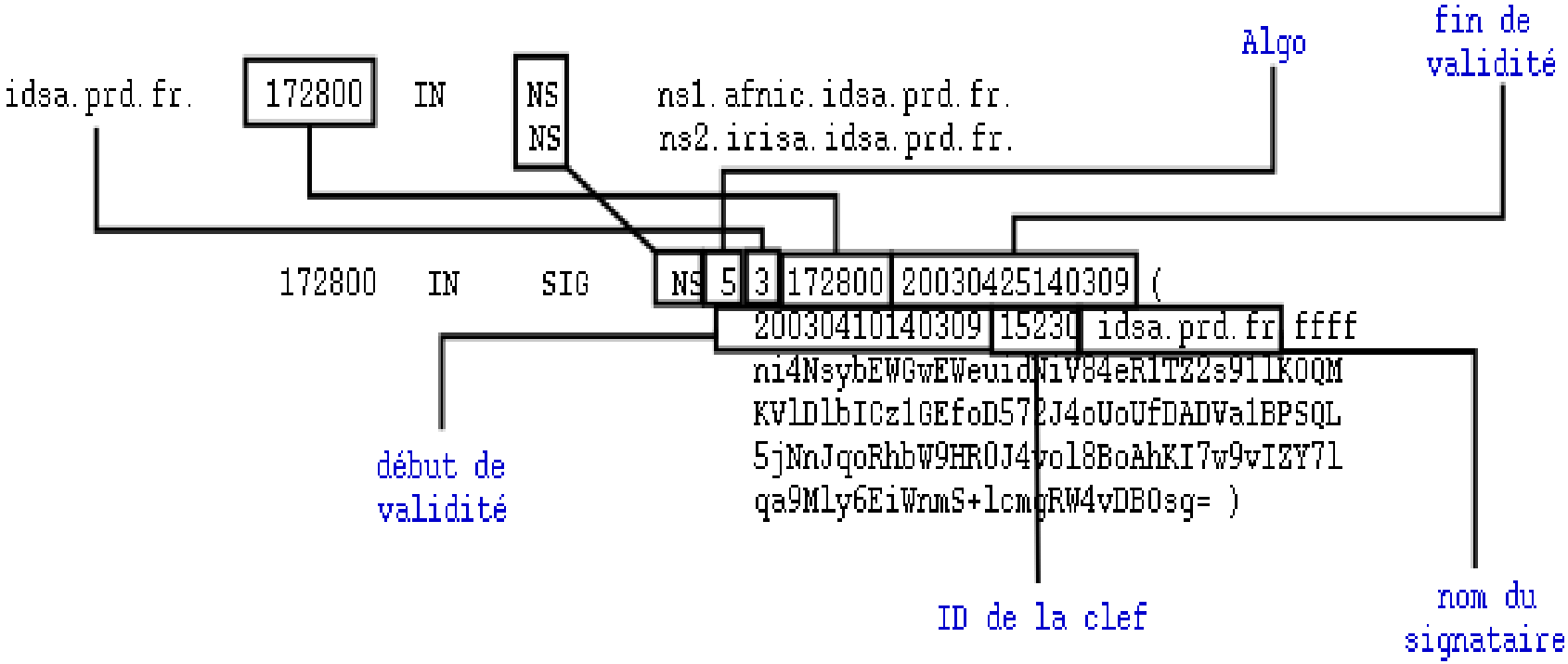
- Le contenu du RDATA est décrit ci-après :

Type Covered (2 octets)	Algorithme (1 octet)	Labels (1 octet)
Original TTL (4 octets)		
Signature Expiration (4 octets)		
Signature Inception (4 octets)		
Key Tag (2 octets)	Nom du signataire (2 octets)	
Nom du signataire (4 octets)		
Signature		

Description du SIG RDATA

- Le RR SIG contient la signature d'un RRset dont le type est indiqué dans le champ « type covered »
- Les champs « signature inception » et « signature expiration » définissent l'intervalle de temps en dehors duquel la signature n'est plus valide
- Les champs « key tag » (ID de la clé) et « signer's name » permettent d'identifier la clé qui a généré la signature

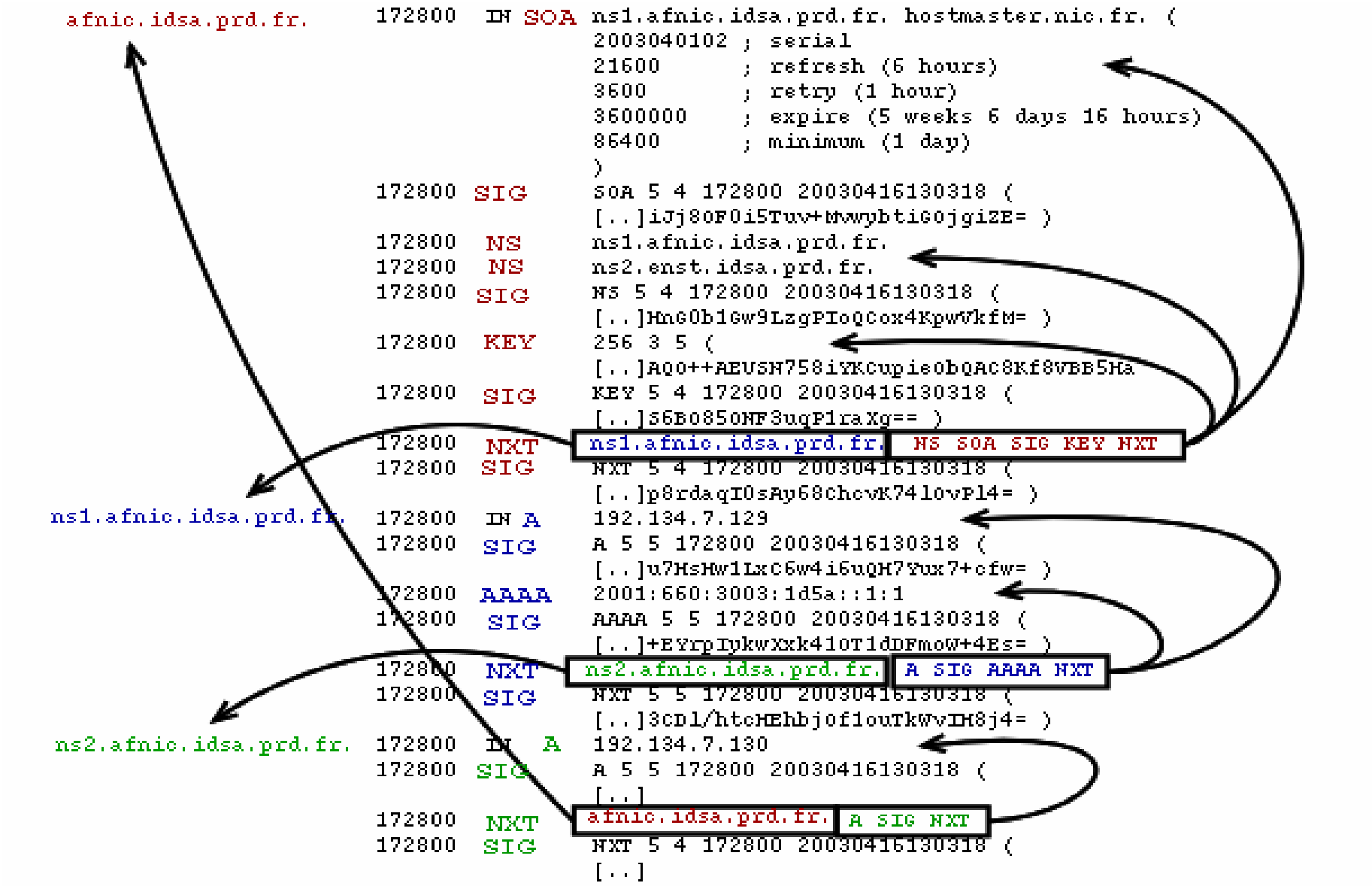
Format SIG : exemple



NXT : nécessité

- Comment signer les réponses négatives (authentification de la non-existence d'un nom ou enregistrement) puisqu'elles ne contiennent pas de RRs
- Ordonnancement de la zone et insertion d'enregistrements NXT entre les noms.
- Le RR NXT d'un nom contient tous les types d'enregistrements associés à ce nom ainsi que le prochain nom présent dans la zone

NXT : fonctionnement



NXT : fonctionnement (2)

- Méthode d'ordonnancement : exemple
  - `idsa.prd.fr`
  - `*.idsa.prd.fr`
  - `afnic.idsa.prd.fr`
  - `*.afnic.idsa.prd.fr`
  - `3.afnic.idsa.prd.fr`
  - `irisa.idsa.prd.fr`
- Remarque: Le NXT du dernier nom pointe sur le premier nom de la zone (vision circulaire de la zone selon le chaînage NXT)

NXT : fonctionnement (3)

- Exemples de fonctionnement (cf. schéma précédent)
  - Une requête portant sur `afnic.idsa.prd.fr, A` (le nom existe mais pas le type) renvoie :

`afnic.idsa.prd.fr NXT ns1.afnic.idsa.prd.fr NS SOA SIG KEY NXT`,

ce qui prouve que le type A n'existe pas pour `afnic.idsa.prd.fr`

  - Une requête portant sur hello.afnic.idsa.prd.fr, `A` (le nom n'existe pas) renvoie :

`afnic.idsa.prd.fr NXT ns1.afnic.idsa.prd.fr NS SOA SIG KEY NXT`,

ce qui prouve qu'il n'existe aucun nom entre `afnic.idsa.prd.fr` et `ns1.afnic.idsa.prd.fr`, donc `hello.afnic.idsa.prd.fr` n'existe pas

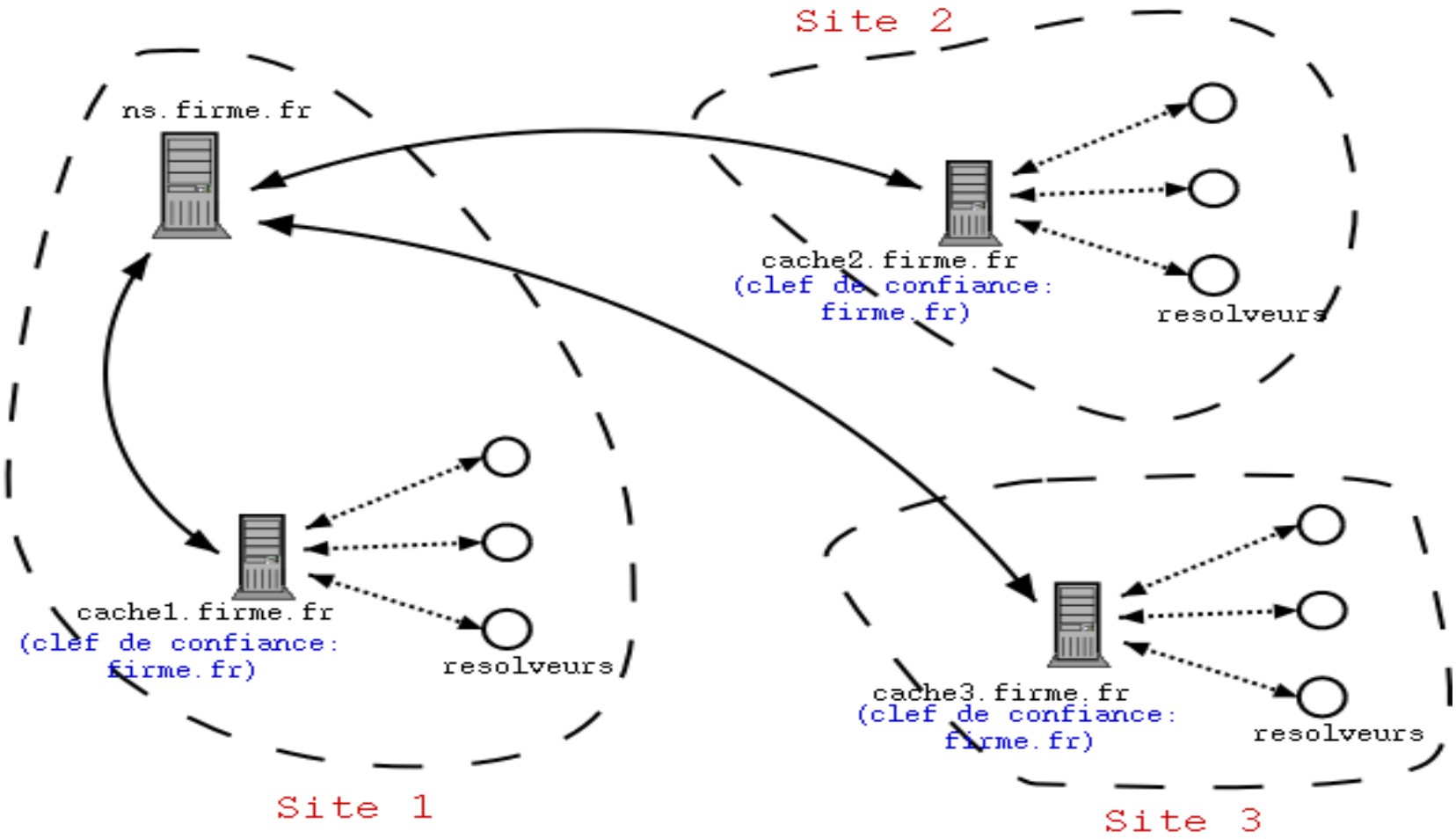
NXT : pour aller plus loin

- Protection contre le rejeu et contre le déni d'existence
- Attention : perte de confidentialité. Possibilité de récupérer tous les noms (ainsi que leurs RRs associés) de la zone (DNS walk)
- Détection des wildcards possible : Si une réponse correspond à une wildcard étendue, le NXT prouvant que le nom demandé n'existe pas explicitement est ajouté dans la réponse

Niveau de sécurité local (côté client)

- La connaissance de la clef publique d'une zone permet de vérifier les signatures et donc l'authenticité et l'intégrité des informations contenues dans la zone
- Concept de clef de confiance
- Limitations : nécessite la connaissance des clefs de toutes les zones avec lesquelles le resolver est susceptible de communiquer

Sécurité locale: exemple



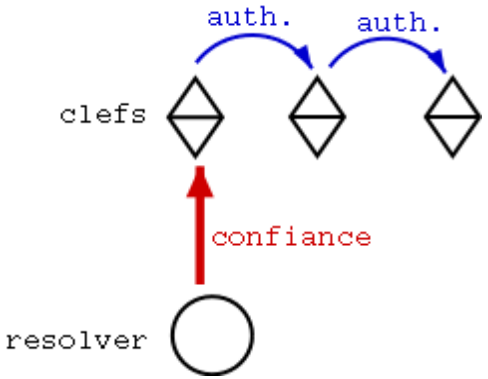
Plan

- Rappels synthétiques sur le DNS
- Vulnérabilités du protocole DNS
- *La sécurisation du DNS : les extensions DNSsec*
  - Rappels de cryptographie
  - Sécurité des transactions
  - Sécurité locale des données
  - ***Sécurité globale des données***
  - DNSsec : le déploiement
  - Les aspects opérationnels
  - Les expérimentations en cours

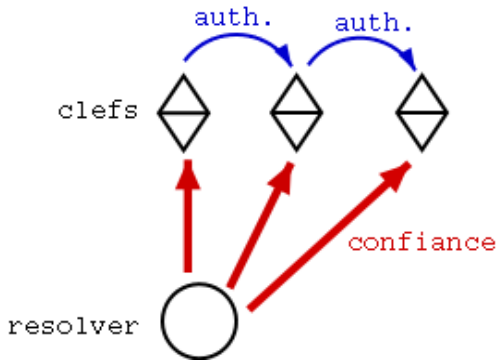
Niveau de sécurité global

- Principe : authentification des clefs en cascade

si



alors



- Structure arborescente du DNS idéale

Notions de délégations sécurisées et chaînes de confiance

- Délégation: présence dans la zone parente d'un point de délégation qui indique le nom des serveurs autoritaires de la zone fille (la zone parente est responsable de la délégation : existence et véracité)
- Délégation sécurisée : d'une manière ou d'une autre la zone parent authentifie la clé utilisée par la zone fille. Avoir confiance en la clé de la zone parent implique la confiance en la clé de la zone fille
- Une chaîne de confiance est un chemin dans l'arbre DNS qui relie un certain nombre de zones séparées par des délégations sécurisées

Mise en place des DS

- Différences modèle RFC2535/modèle DS
  - Le modèle DS ne nécessite qu'un aller au lieu d'un aller-retour dans la communication zone parente/zone fille
- Transmission de la clé publique de la zone fille à la zone parente
- Génération du DS (hash de la clé) et signature de celui-ci dans la zone parente
- Le DS devient le maillon de confiance entre zone parente et zone fille

Délégations sécurisées avec DS

- Dans la zone parente, pour tout point de délégation,
  - La présence d'un DS signé prouve l'existence d'une délégation sécurisée vers la zone fille et authentifie la clef associée au DS
  - L'absence de DS, prouvée par le contenu du NXT, lui même signé prouve qu'aucune délégation sécurisée n'a été établie vers la zone fille.

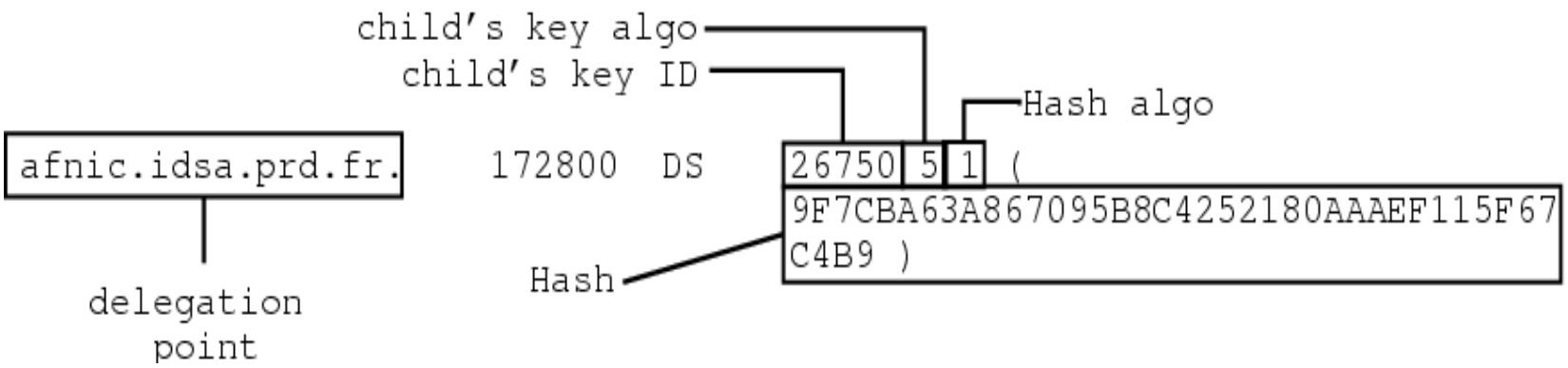
Le RR DS

- Nouveau RR post-RFC 2535 qui propose une nouvelle méthode de sécurisation d'une délégation
- DS = Delegation Signer (en cours de standardisation)
- Format du RDATA :

ID de la clé (2 octets)	Algorithme (1 octet)	Type du hash (1 octet)
Hash		

Description du DS RDATA

- Les champs ID de la clé et algorithme identifient la clé pointée par ce DS
- Le champ «digest type » indique le type d'algorithme utilisé pour réaliser le hash (actuellement on utilise SHA1)



Plan

- Rappels synthétiques sur le DNS
- Vulnérabilités du protocole DNS
- *La sécurisation du DNS : les extensions DNSsec*
  - Rappels de cryptographie
  - Sécurité des transactions
  - Sécurité locale des données
  - Sécurité globale des données
  - ***DNSsec : le déploiement***
  - Les aspects opérationnels
  - Les expérimentations en cours

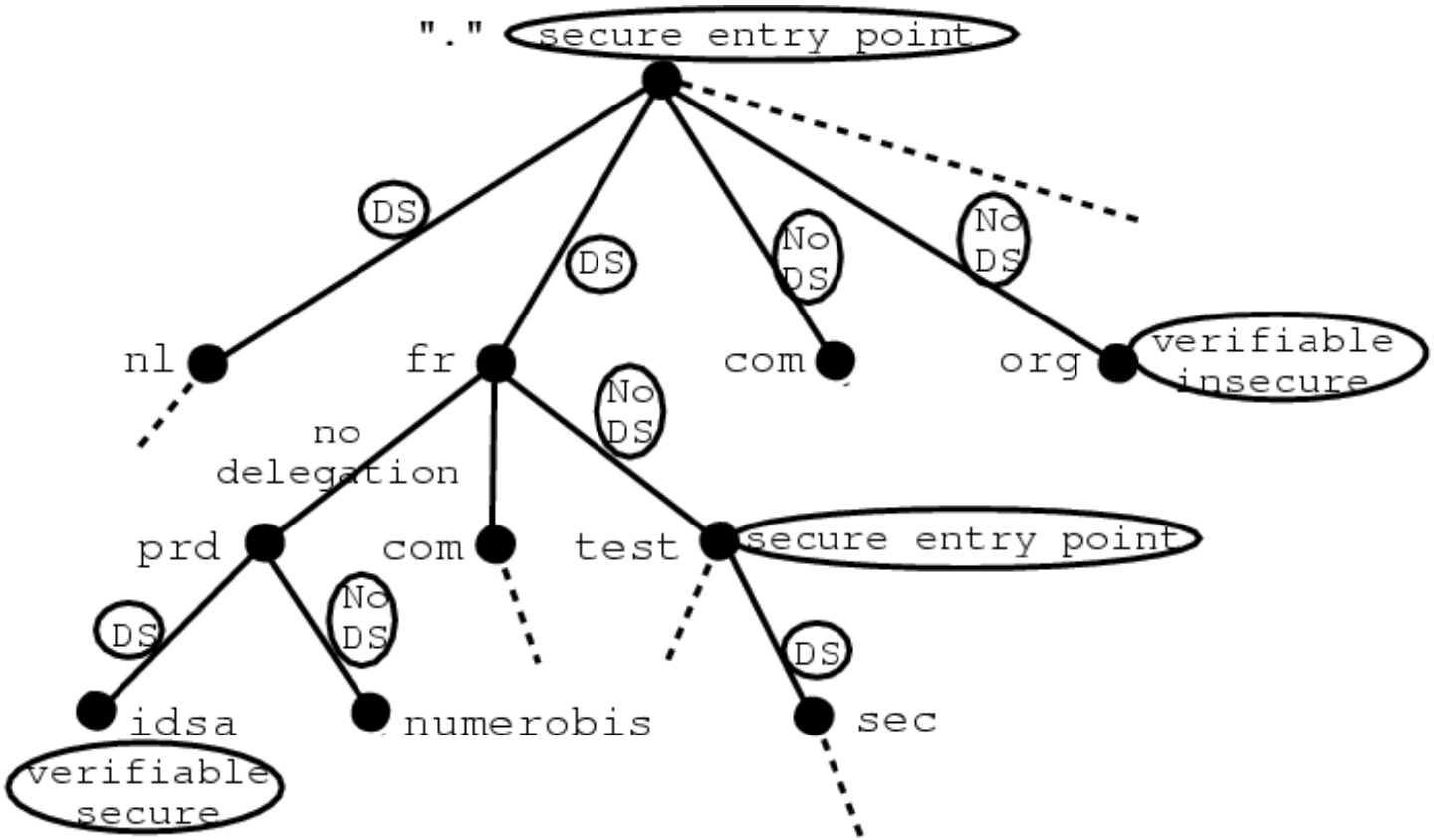
Classification des informations DNS

- Classification objective
  - zone non sécurisée (non signée)
  - sécurisée localement (signée mais la délégation vers sa zone parente n'est pas sécurisée)
  - sécurisée globalement (signée, et on peut l'atteindre en parcourant une chaîne de confiance depuis une zone ancêtre)
- Sécurisation progressive de l'arbre et îlots sécurisés
  - Un îlot sécurisé rassemble toutes les zones accessibles en établissant des chaînes de confiance depuis une zone signée au sommet de l'îlot
  - Le but ultime de DNSsec : la simple connaissance des clés de la racine permettra d'accéder à n'importe quelle zone de manière sécurisée en parcourant les chaînes de confiance

Classification des informations DNS (2)

- Classification subjective :
  - dépendante du resolver en fonctions des clefs de confiance dont il dispose
  - “ verifiable secure”, “ verifiable insecure”, “ wrong”
- Notions de point d’entrée sécurisé et clefs de confiance

Arbre DNS partiellement sécurisé



Caractéristiques des clés en fonction de leur taille

- Clef courte :
  - Plus facile à casser
  - Temps de signature plus court
  - Temps de vérification des signatures par les utilisateurs plus court
  - Taille de zone réduite
- Clef longue :
  - Plus difficle à casser
  - Temps de signature et de vérification des signatures plus important

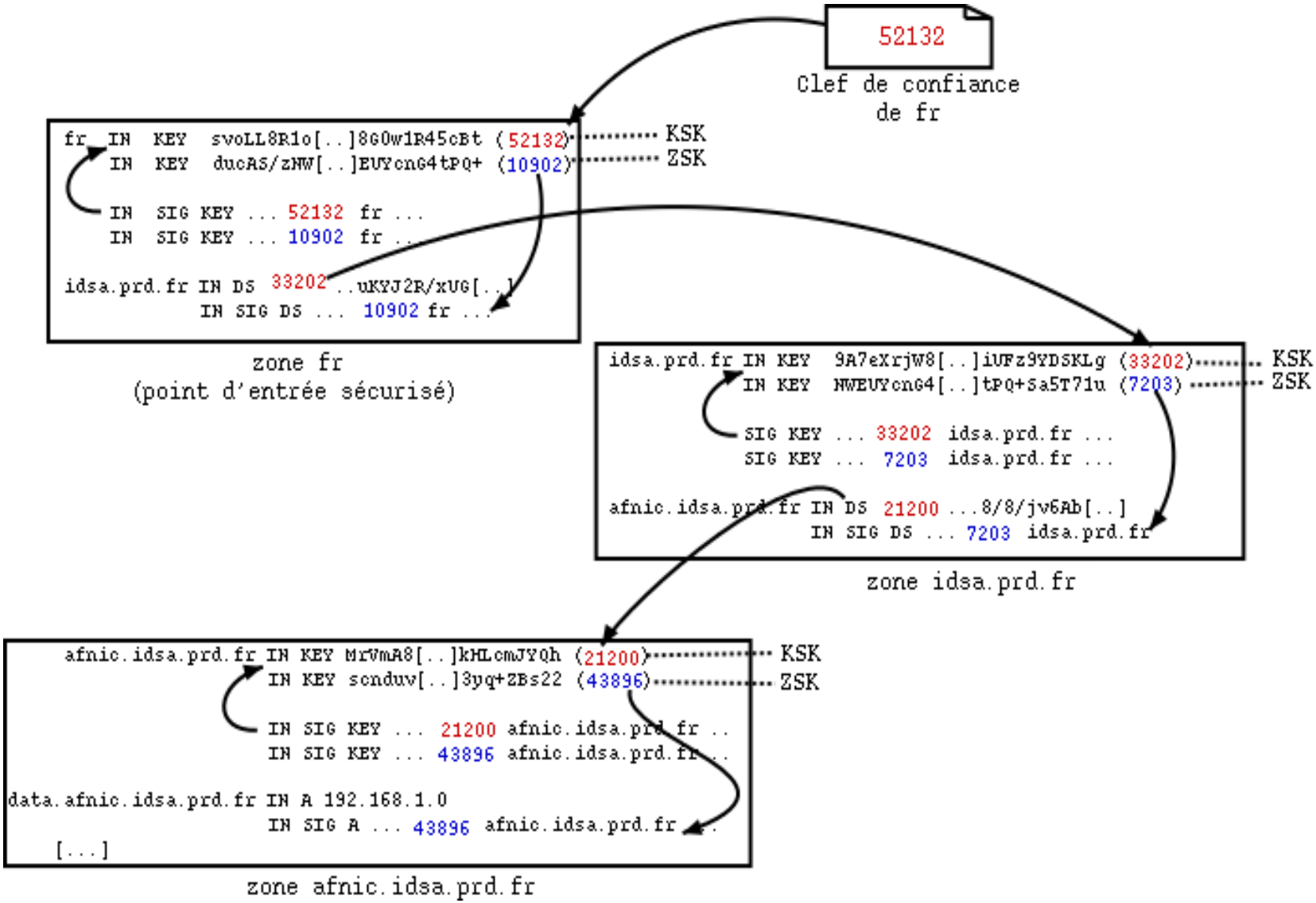
Importance d'un modèle à deux clefs

- Les clefs n'ont pas de durée de vie intrinsèque, elle doivent être changées régulièrement
- La longueur de la clé ne doit pas handicaper la sécurité :
  - Une clé courte devra être associée à une fréquence de roulement élevée
  - Une clé longue pourra être changée moins souvent
- Les besoins DNSsec rendent le compromis difficile entre clé courte et clé longue :
  - Signer les zones et vérifier les signatures devrait être rapide, ce qui implique l'utilisation d'une clé courte
  - On doit limiter au maximum les interactions entre zone fille et zone parente, ce qui implique l'utilisation d'une clé longue (roulement moins fréquent)

Distinction ZSK/KSK

- Utilisation de deux clefs: ZSK (Zone Signing Key) et KSK (Key Signing Key)
- Séparer les rôles :
  - Clef qui signe les enregistrements d'une zone : la ZSK. C'est une clé de taille réduite que l'on changera fréquemment
  - Clef qui fait office de maillon de confiance : la KSK. Elle ne signe que le KEY RRset donc elle peut être de taille plus longue, ce qui permet de limiter sa fréquence de renouvellement
- Une solution pour différencier leur structure est à l'étude: ajout d'un flag dans le RDATA
- Flexibilité accrue dans la relation zone parent/ zone fille

Authentifications en cascade dans une chaîne de confiance



Indication du support DNSsec

- Cohabitation entre entités supportant et ne supportant pas DNSsec
  - Indiquer le support DNSsec
  - Normaliser le comportement envers les données signées et les RRs de sécurité
  - Gérer les requêtes et réponses nécessitant ou pas des RRs relatifs à la sécurité (DNSKEY, RRSIG, NSEC, DS)
- Création de 3 flags (DO, AD, CD) décrits ci-après

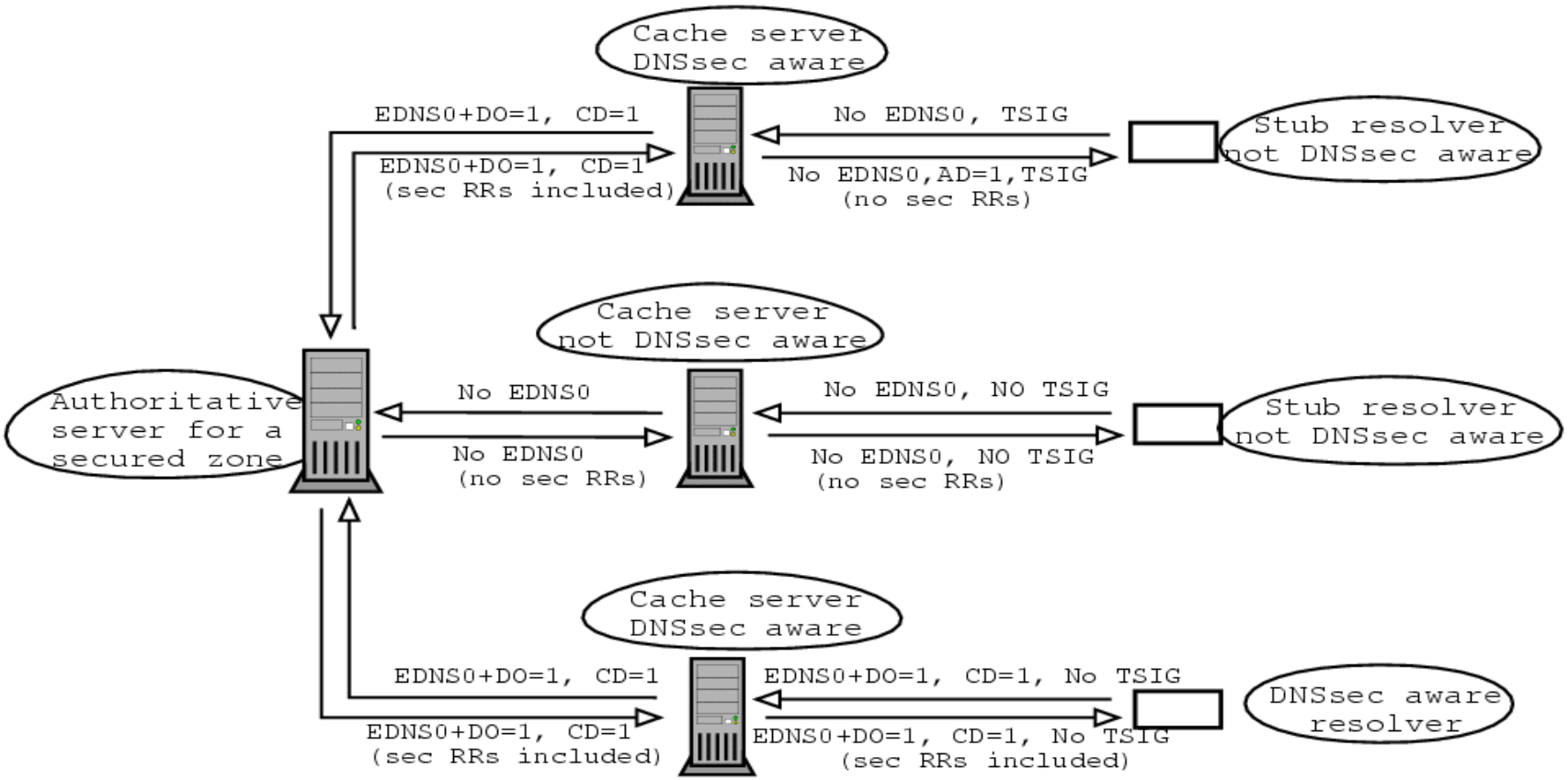
Les extensions EDNS0 (flag DO)

- EDNS0 est un OPT pseudo RR ajouté dans la section additionnelle qui contient un certain nombre d'informations :
  - La longueur maximale supportée pour un paquet UDP (permet d'oublier la limite des 512 octets)
  - Le flag DO (DNSsec OK) positionné indique le support DNSsec

Deux nouveaux flags : AD et CD

- Le flag AD (Authenticated Data) : permet à un serveur récursif de spécifier au résolveur que les données qu'il lui transmet ont été vérifiées
- Si le canal entre le serveur cache et le résolveur n'est pas sécurisé, le résolveur peut choisir de ne pas tenir compte de ce flag
- Le flag CD (Checking Disabled) : permet à un résolveur de spécifier à son serveur récursif qu'il désire faire les vérifications lui-même

Scénarios de cohabitation



Renommage des RRs

- Protocole DNSsec en cours de réécriture
- Nécessité de faire la distinction entre la RFC 2535 et la future version du protocole
- DNSKEY sera utilisé à la place de KEY et est strictement réservé au stockage des clefs DNSsec
- D'autres types seront créés pour stocker les clefs correspondant à d'autre applications (ex: IPSECKEY)
- RRSIG à la place de SIG
- NSEC à la place de NXT
- DS reste DS puisqu'il n'existait pas dans la RFC 2535

Plan

- Rappels synthétiques sur le DNS
- Vulnérabilités du protocole DNS
- *La sécurisation du DNS : les extensions DNSsec*
  - Rappels de cryptographie
  - Sécurité des transactions
  - Sécurité locale des données
  - Sécurité globale des données
  - DNSsec : le déploiement
  - ***Les aspects opérationnels***
  - Les expérimentations en cours

Nouveaux problèmes émergents

- Nécessité d'un niveau de sécurité intrinsèque des serveurs. Le déploiement de DNSsec devrait donc indirectement augmenter le niveau de sécurité des serveurs
- Nouveaux enjeux : maintenance
  - Automatisation des procédures
  - Surveillance
  - Responsabilité dans les chaînes de confiance
  - Précautions pour la gestion des clefs
- Procédure la plus délicate : le roulement des clés

Le roulement des clés

- Key Rollover
- Possibilité de compromission des clés
  - perte ou vol de la partie privée
  - attaques cryptanalytiques
- Roulement planifié/ roulement d'urgence
- Efficacité du modèle ZSK/KSK
- Précautions concernant les temps caractéristiques (validité des SIGs, intervalle de resignature, TTLs)

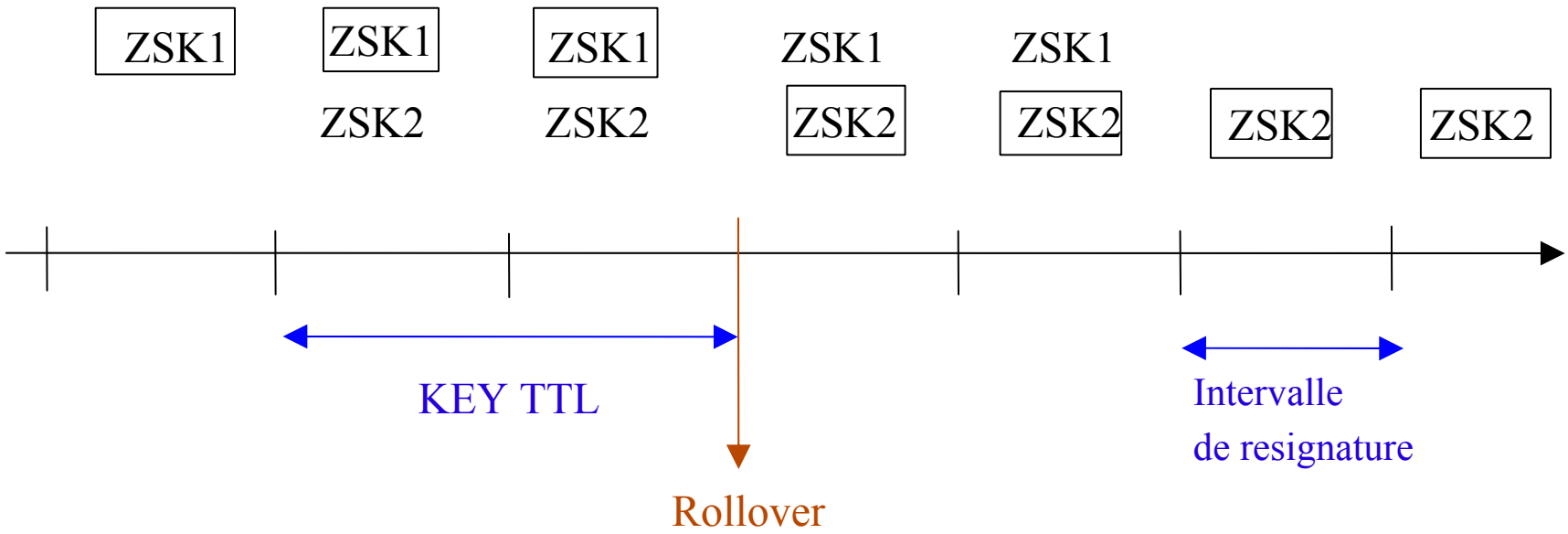
Roulement ZSK planifié

- ZSK de petite taille ➔ Roulement fréquent et régulier
- Ce roulement est local à la zone (pas d'interactions avec la zone parente)
- Contraintes à considérer : les TTLs et la propagation des données dans les caches
- Procédure conseillée : pré-publication de la future clef + post-suppression de l'ancienne clé

Roulement ZSK planifié : schéma

- Exemple : on resigne tous les jours; le TTL est de 2 jours.

Dans ce cas, une clef reste publiée 11 jours (dont 7 jours où elle signe la zone)



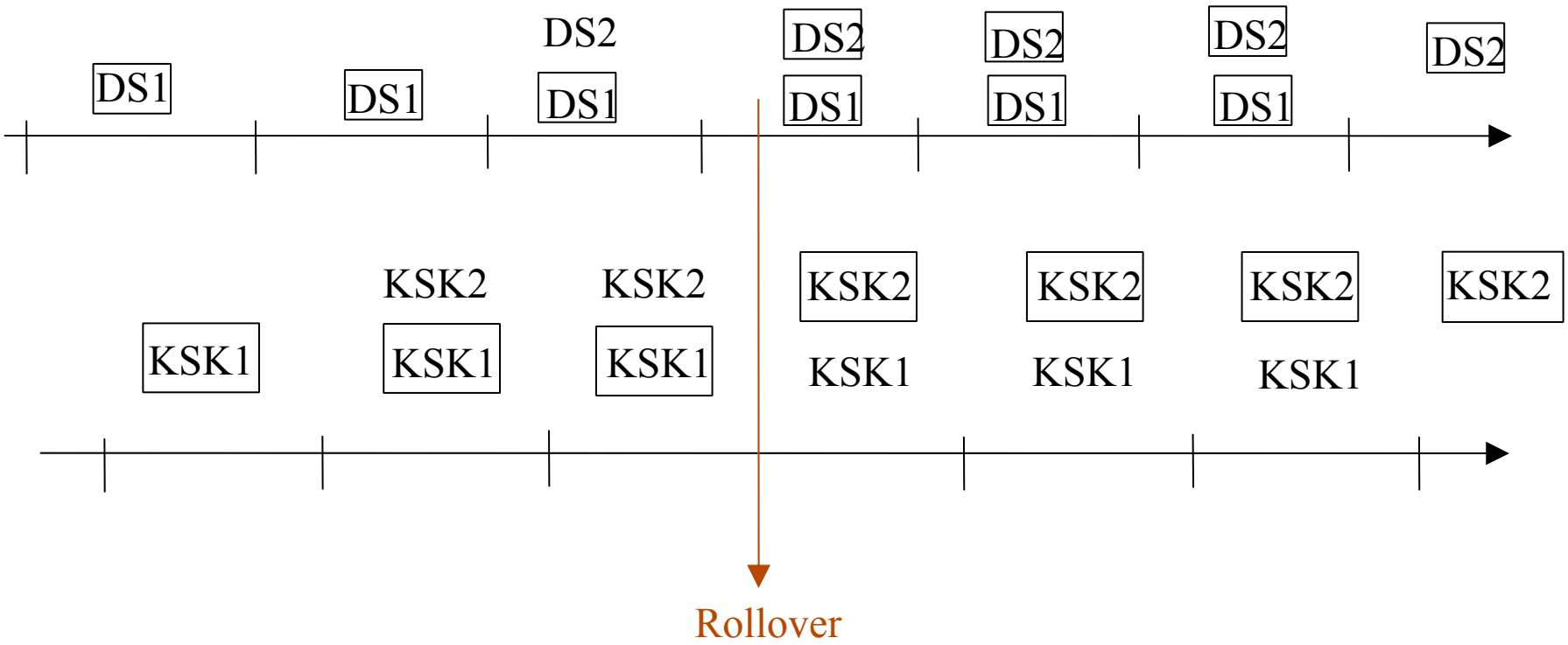
Roulement KSK planifié

- Prépublication de la nouvelle KSK (idem procédure ZSK)
- Transmission de la nouvelle KSK à la zone parente
- Ne pas rompre la chaîne de confiance : le changement de DS doit être propagé dans les caches. Pendant cette durée, il est souhaitable que la zone fille utilise simultanément les deux KSK ou que la zone parente publie 2 DS
- Communiquer sur le changement de clé car certains résolveurs avaient configuré l'ancienne clé comme clé de confiance
- Ce rollover nécessite une bonne synchronisation des zones fille et parente

Roulement KSK planifié : schéma

- Exemple: intervalle de resignature égal à 1 jour pour la zone parente et la zone fille (mais la resignature n'intervient pas en même temps)

KEY TTL = 2 jours, DS TTL = 1 jour



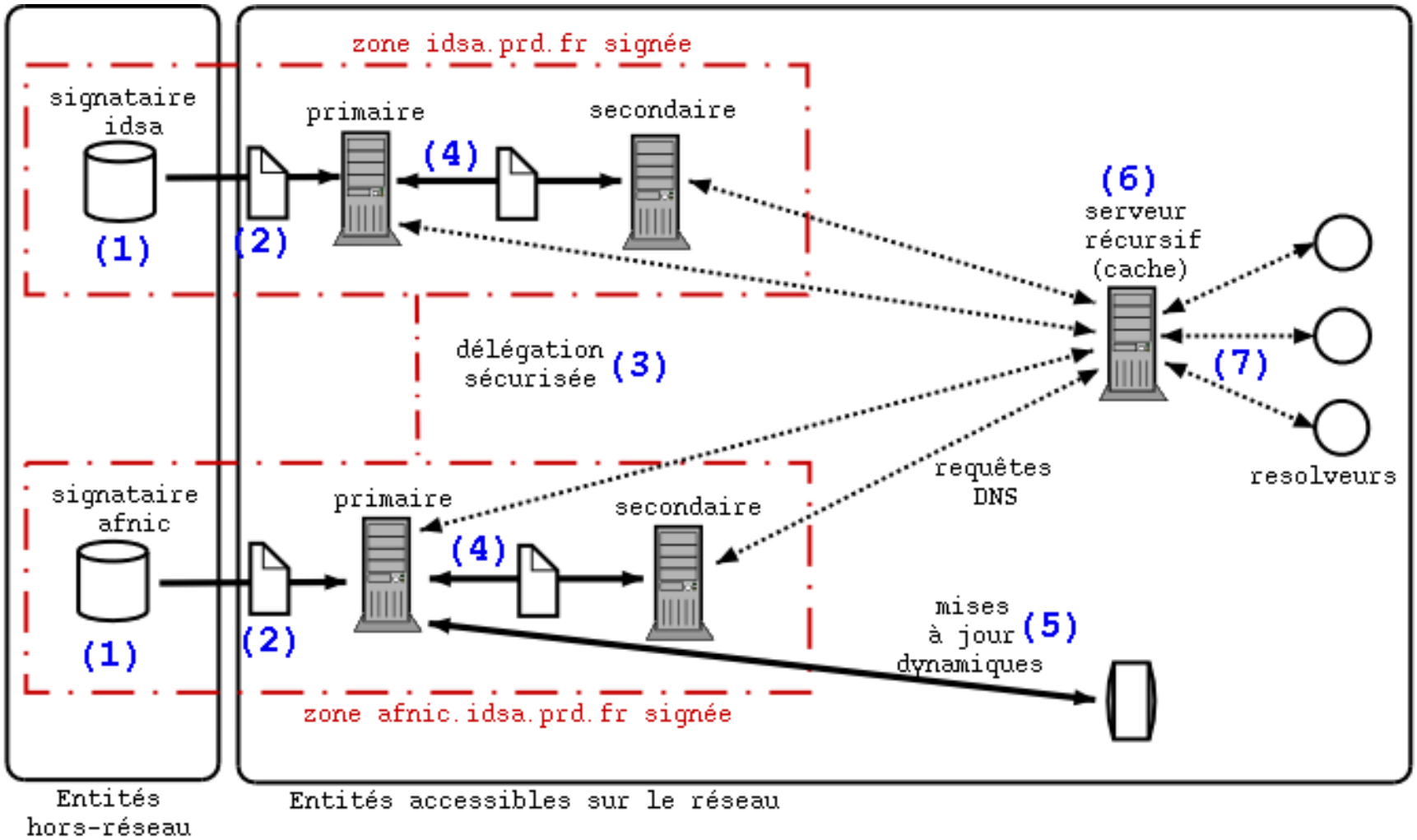
Roulements d'urgence

- Par définition : impossible à préparer, les procédures décrites précedemment ne s'appliquent pas
- Nécessité d'une politique de sécurité locale
- Compromis entre rupture de la chaîne de confiance (si changement de clef immédiat) et risque d'attaques si conservation de la clé corrompue le temps de réaliser une procédure décrite dans les transparents précédants
- Possibilité de publier en permanence dans le KEY RRset une clé qui ne sera utilisée qu'en cas d'urgence

Considérations opérationnelles

- Utilisation de BIND9.3s (snapshots) et ses outils
  - dnssec-keygen
  - dnssec-signzone
- Performances
  - Temps de signature reste raisonnable même pour des zones de grande taille
  - Taille de la zone signée: multipliée par un facteur 6 à 10 par rapport au fichier non signé
  - Taille des réponses: pour une même requête, une réponse DNSsec aura une taille de l'ordre de 5 à 10 fois la taille de la réponse DNS correspondante

Bilan opérationnel



Bilan opérationnel (2)

- Commentaires sur le schéma précédent:
  - (1) : signataire d'une zone, de préférence non accessible sur le réseau, c'est là que les clés sont générées et stockées et la zone est signée
  - (2) : le fichier de zone est transmis au serveur primaire de manière sécurisée
  - (3) : les délégations sécurisées sont mises en place
  - (4) : transfert de zone entre serveur primaire et secondaire(s) (sécurisé par TSIG par exemple)

Bilan opérationnel (3)

- (5) : Les mises à jour dynamiques sont sécurisées grâce à TSIG ou SIG(0)
- (6) : Les serveurs récursifs sont configurés avec des clefs de confiance correspondant à leur(s) point(s) d'entrée(s) sécurisé(s) dans l'arbre DNS
- (7) : Les résolveurs communiquent avec leur serveur récursif par défaut au-dessus d'un canal sécurisé (ex: TSIG)

DNSsec comme PKI

- Nécessité de stocker et distribuer les clefs publiques utilisées par DNSsec (RRset KEY)
- Possibilité de stocker des clefs pour d'autres applications (IPSEC, SSH...)
- Possibilité de stocker des certificats: RR CERT

DNSsec *vs* PKI

- PKI (IGC) : Ensemble des matériels, logiciels, personnes, règles et procédures nécessaires pour créer, gérer, distribuer des certificats X509
- Les plus d'une PKI
  - Importance de l'aspect juridique dans les PKI
  - Les procédures de DNSsec sont basées sur des politiques locales
  - Il n'existe pas de CRL (Certificat Revocation List) dans DNSsec

Interactions IPsec/DNSsec

- IPsec peut être utilisé pour sécuriser
  - les transferts de zone
  - Le canal entre un résolveur light et son serveur récursif (celui qui fait les résolutions DNSsec pour son compte)
- Une architecture DNS sécurisée peut être mise à profit pour stocker des clés utilisées pour établir des communications IPsec entre deux entités ne se connaissant pas à priori (IPSECKEY). C'est le cas d'IPsec Opportunistic Encryption
  - Chacune des entités récupère l'IPSECKEY de l'autre par résolution DNS sécurisée, et on peut ainsi établir le canal IPsec

Plan

- Rappels synthétiques sur le DNS
- Vulnérabilités du protocole DNS
- *La sécurisation du DNS : les extensions DNSsec*
  - Rappels de cryptographie
  - Sécurité des transactions
  - Sécurité locale des données
  - Sécurité globale des données
  - DNSsec : le déploiement
  - Les aspects opérationnels
  - ***Les expérimentations en cours***

Expérimentations DNSsec

- Protocole toujours en évolution
- Expérimentations et retours d'expérience assez limités
- Sécurisation de la zone fr (Autosign-TLD) sur des serveurs non référencés par les serveurs racine
- Projet RS.net : serveurs racine de test (DNSsec, IPv6, IDN, ...) et délégations sécurisées vers les TLDs participants : .fr, .nl, .se, .jp ...

Le projet IDsA

- Projet RNRT IDsA (Infrastructure DNSsec et Applications) : http://www.idsa.prd.fr et ftp://ftp.idsa.prd.fr
- Déploiement d'une plate-forme de tests
- Développement d'outils de vérification des chaînes de confiance et d'un resolver supportant DNSsec
- Développement d'outils d'automatisation des procédures
- Etude des interactions avec IPsec et Mobile IPv6

Conclusions

- DNSsec : sécurité contre les attaques spécifiques au DNS en proposant authentification de la source et intégrité des données
- Déployable dès maintenant et compatible avec le DNS non sécurisé mais protocole non encore finalisé
- Enjeux :
  - automatisation des procédures
  - résolveur supportant DNSsec
- Rôle de pseudo-PKI pour distribuer les clefs d'autres applications

Références / liens utile

- http://www.idsa.prd.fr et ftp://ftp.idsa.prd.fr/local/idsa/
  - idsa-tech@nic.fr
  - dnssec@nic.fr
- http://www.isc.org/
- http://www.ietf.org/html.charters/dnsext-charter.html
- http://www.dnssec.net/

A propos de ce document

- **Auteurs : {Rahim.Djaffar, Bertrand.Leonard, Jean-Philippe.Pick, Mohsen.Souissi}@nic.fr**
- **Copyright IDsA** :

Questions ?