Tutoriel DNSSEC

Présenté par :

Stéphane Bortzmeyer, AFNIC

JRES 2009 - Nantes, 4 décembre 2009

{Stephane.Bortzmeyer,Mohsen.Souissi}@afnic.fr

Plan

Le modèle DNS

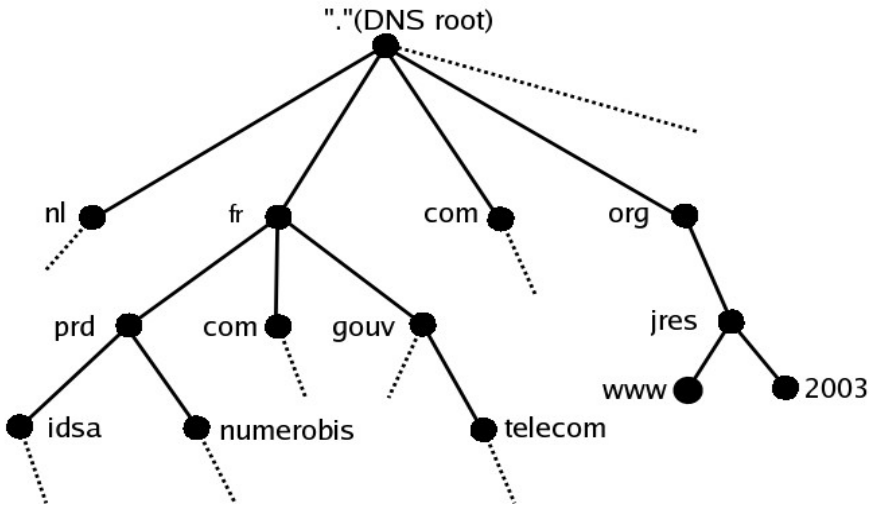
Architecture client/serveur

La base de données DNS contient les associations entre les noms de domaine et un certain nombre d'informations (adresses IP, relais de courrier, serveurs de nom, etc.)

Hiérarchique

Distribuée

Redondante



L'information DNS

Les enregistrements DNS (Resource Records : RRs)

Les RRsets

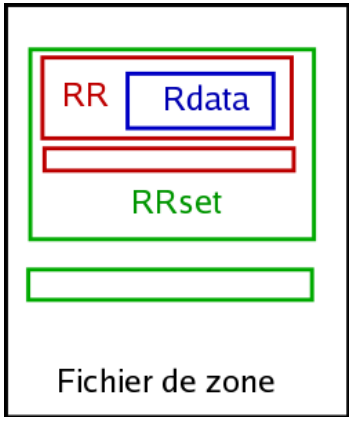
Les fichiers de zone

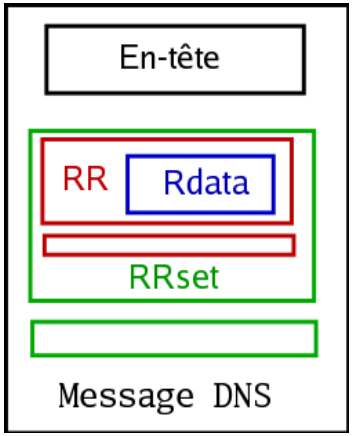
Les messages DNS

La durée de vie de l'information DNS :

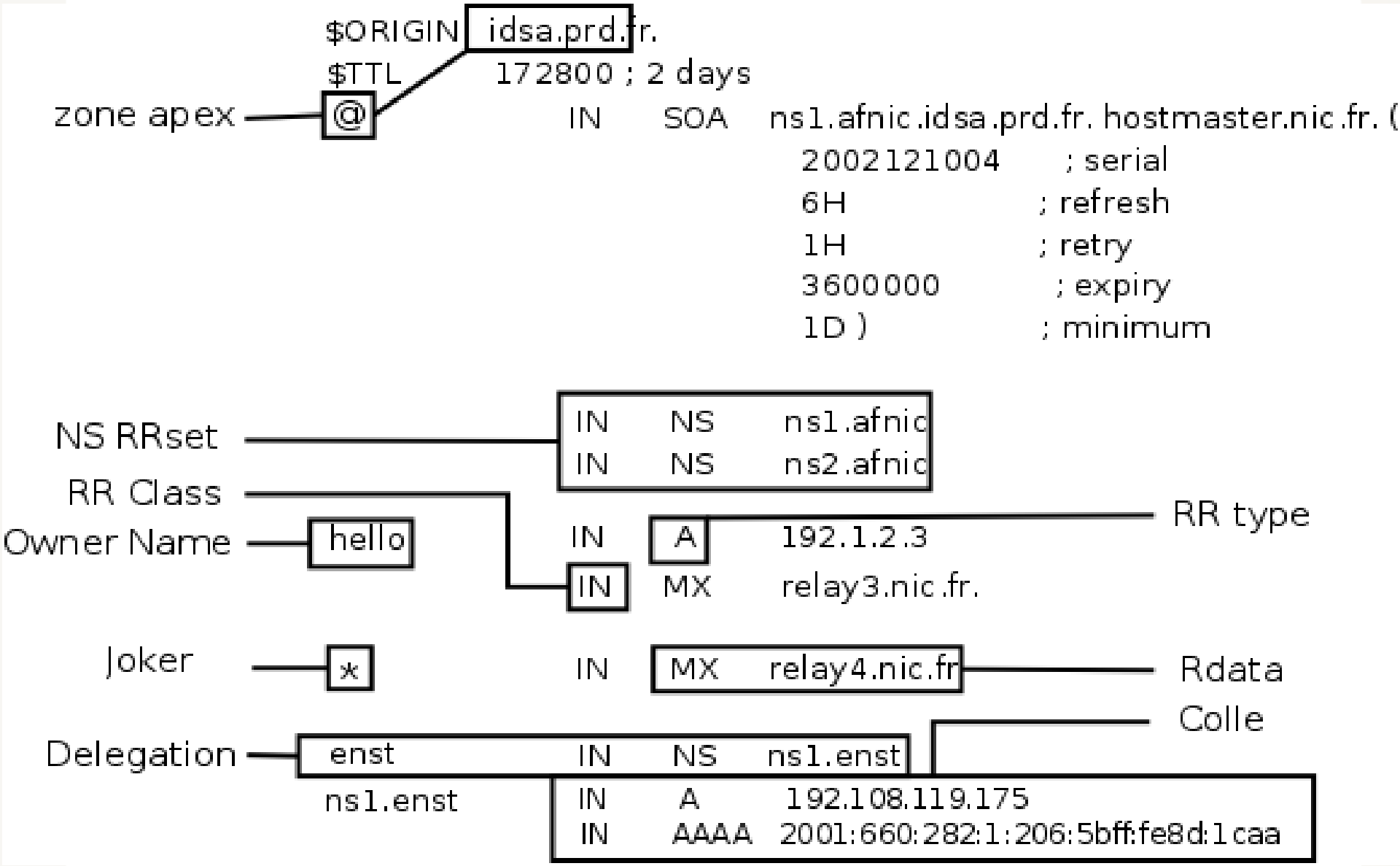
Sur les serveurs faisant autorité : tant que la zone est chargée

Sur les serveurs cache : notion de TTL





Exemple de fichier de zone (sans DNSSEC)



Plan

Rappels synthétiques sur le DNS

Vulnérabilités du 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 a priori

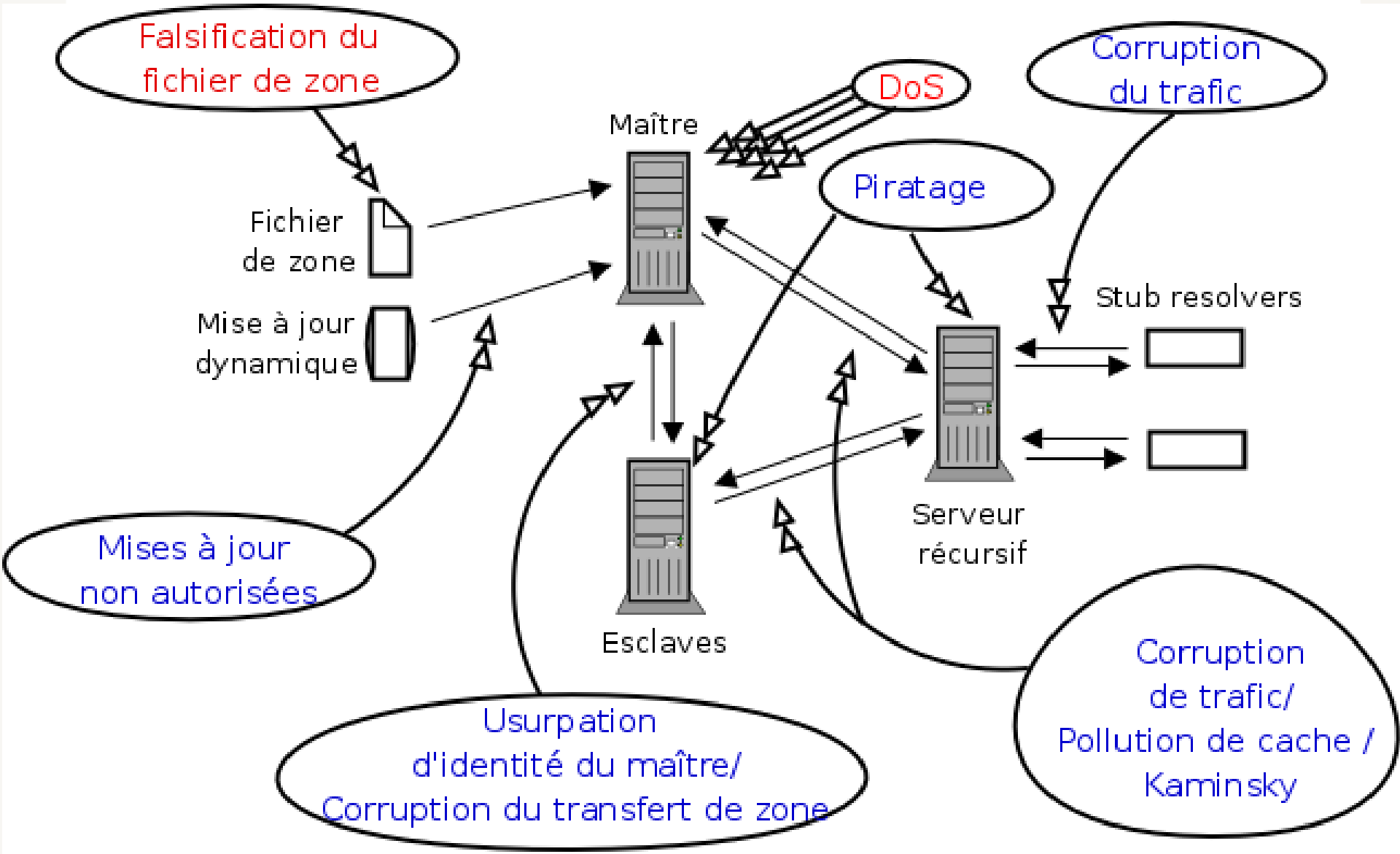
DNS : omniprésent mais invisible lors d'une utilisation « humaine » de l'Internet

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 simple vandalisme

Rediriger les utilisateurs à leur insu : préambule à une attaque plus grave

Récupérer des informations critiques (mots de passe, courriers, ...)

Le « DNS Spoofing »

Spoofer = usurper l'dentité de quelqu'un

Principe : l'attaquant répond à la place du serveur interrogé pour tromper un utilisateur ou polluer un serveur cache

Nécessité de connaître la question posée et l'ID associé (2 octets), 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.

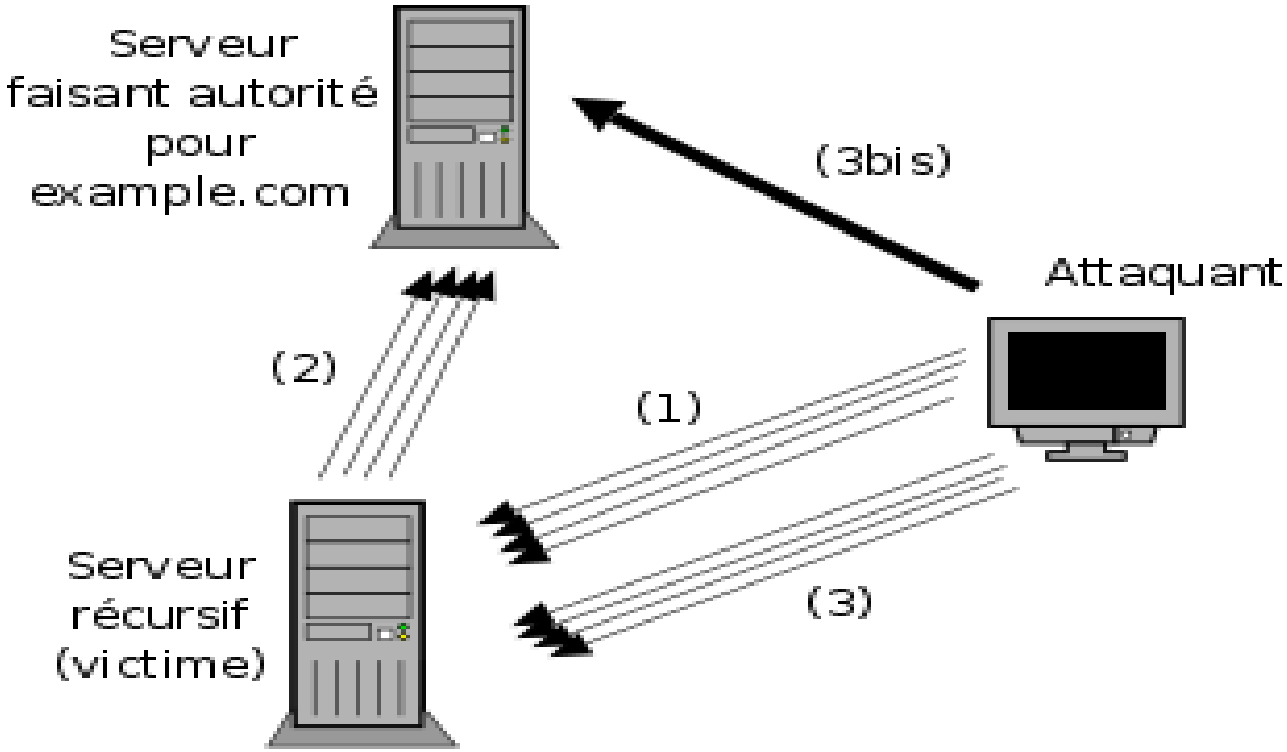
Ex: dans Bind4x, les IDs sont incrémentaux

- Attaque Kaminsky: varier les noms demandés pour pouvoir multiplier les requêtes (cela permet de trouver le bon Query ID)

Un exemple d'attaque

Attaque de type « Birthday attack »

Paradoxe de l'anniversaire : 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.example.com associées à N IDs différents

(2): transfert des N requêtes vers le serveur faisant autorité de example.com

(3): N réponses forgées associées à N IDs différents sont envoyées par l'attaquant

(3bis): DoS sur le serveur faisant autorité pour le ralentir

N=300, probabilité de succès de l'imposture >1/2

RFC 5452 pour les calculs détaillés

Plan

Rappels synthétiques sur le DNS

Vulnérabilités du 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

Architecture de distribution des clés

Clés utilisées par DNSSEC

Clés 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 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 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

RSA (Rivest Shamir Adelman)

DSA (Digital Signature Algorithm, FIPS 186)

Courbes elliptiques

Exemples d'utilisation : PGP (courrier), 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 clés par utilisateur

La connaissance de la clé publique d'un utilisateur suffit pour communiquer avec lui

Le DNS étant public :

Besoin de signature essentiellement

Principe du hachage

Notion d'empreinte / condensat (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 :

Whirlpool

SHA (Secure Hash Algorithm, versions 1, 256, etc)

Plan

Techniques de sécurisation du canal

Pour sécuriser le canal entre deux serveurs, on peut utiliser :

- **TSIG**
- **IPsec (peu déployé en pratique)**
- **DNScurve (non normalisé)**

Plan

Rappels synthétiques sur le DNS

Vulnérabilités du 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

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

DNSSEC authentifie les données, pas le canal. Il peut donc protéger, par exemple, contre un serveur secondaire malhonnête ou piraté.

Rappels de sécurité dans le contexte DNSSEC

Authentification : identité de l'emetteur non usurpée

Intégrité : non altération des données garantie

Protection contre le déni d'existence : prouver la non-existence d'une donnée

Confidentialité : garantir le secret des données transmises (sans objet pour DNSSEC)

Niveau de sécurité local (côté serveur)

Chaque zone génère un ensemble de paires de clés (partie privée/partie publique)

Les clés sont associées à la zone et non aux serveurs

Les parties privées des clés signent les informations (RRsets) faisant partie intégrante de la zone (c'est-à-dire sur lesquelles le serveur fait autorité)

Certaines informations ne sont pas signées :

Les points de délégation (mais les DS, eux, sont signés)

Les colles (adresses IP des serveurs situés dans le domaine qu'ils servent)

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 : RRSIG

Les parties publiques des clés sont publiées dans le fichier de zone et peuvent faire l'objet de requêtes DNS standard : DNSKEY

En revanche seul le signataire d'une zone doit avoir connaissance de la partie privée des clés

Le RR DNSKEY

DNSKEY 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 DNSKEY RDATA

Flags: permet de distinguer les clés de zone des clés utilisées pour d'autres services DNSSEC (ex : SIG(0))

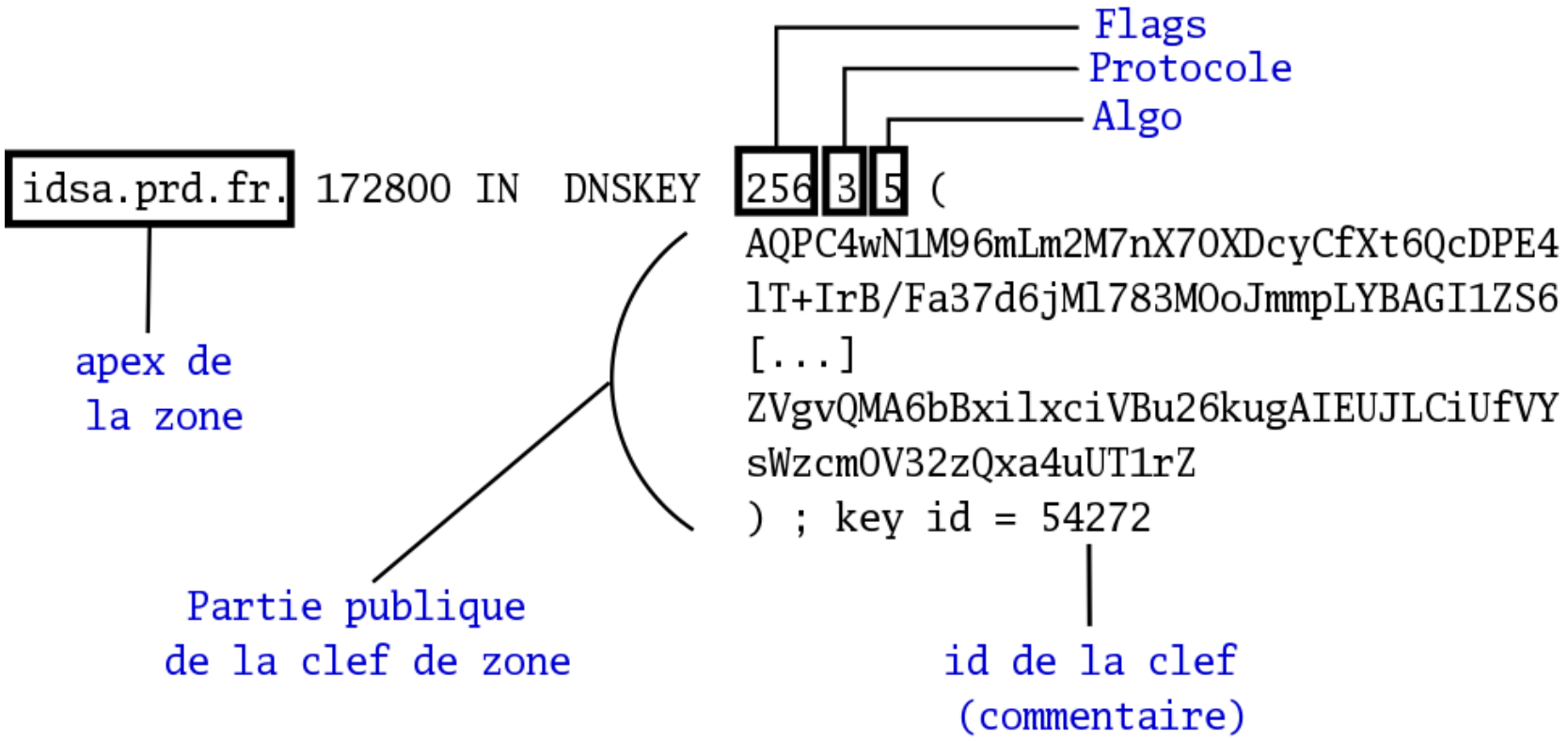
Protocole : donne la possibilité de stocker des clés utilisées par d'autres protocoles (IPsec par exemple)

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 DNSKEY : exemple

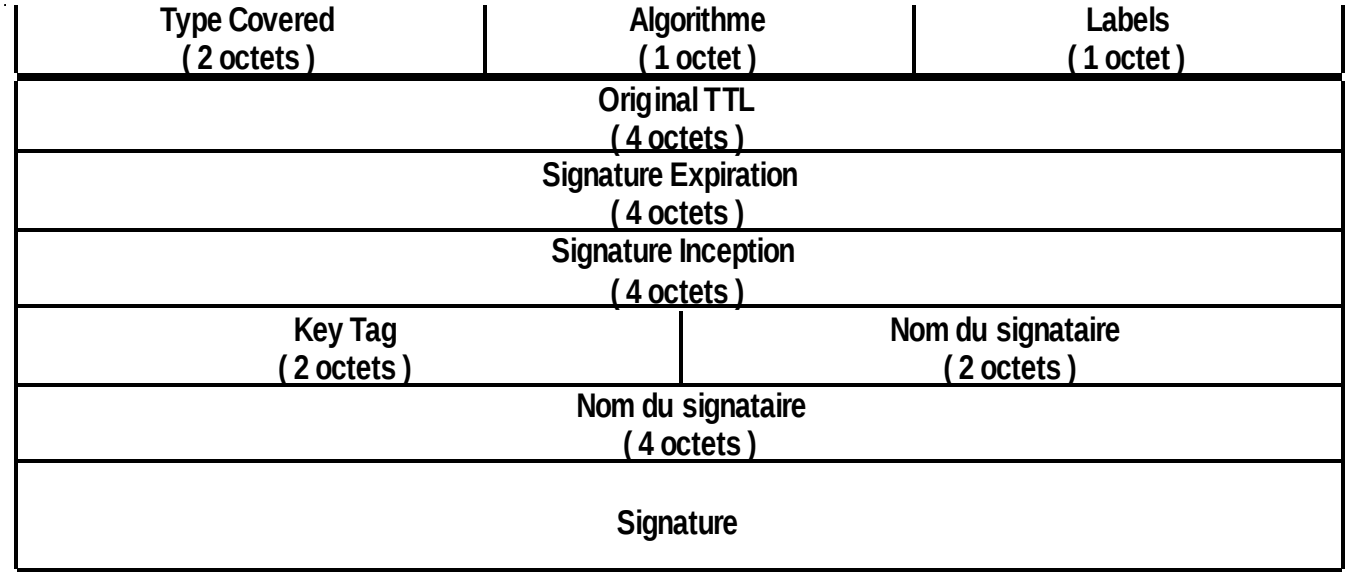


Affichage de DNSKEY avec dig

```
%  dig +multi DNSKEY sources.org
sources.org.      86400 IN DNSKEY 256 3 3 (
CkoOFAptkq5oqs8hlDf1gYDDpRPrjySc+wjBxLBbQtkX
...) ; key id = 14347
```

Le RR RRSIG

Le contenu du RDATA est décrit ci-après :



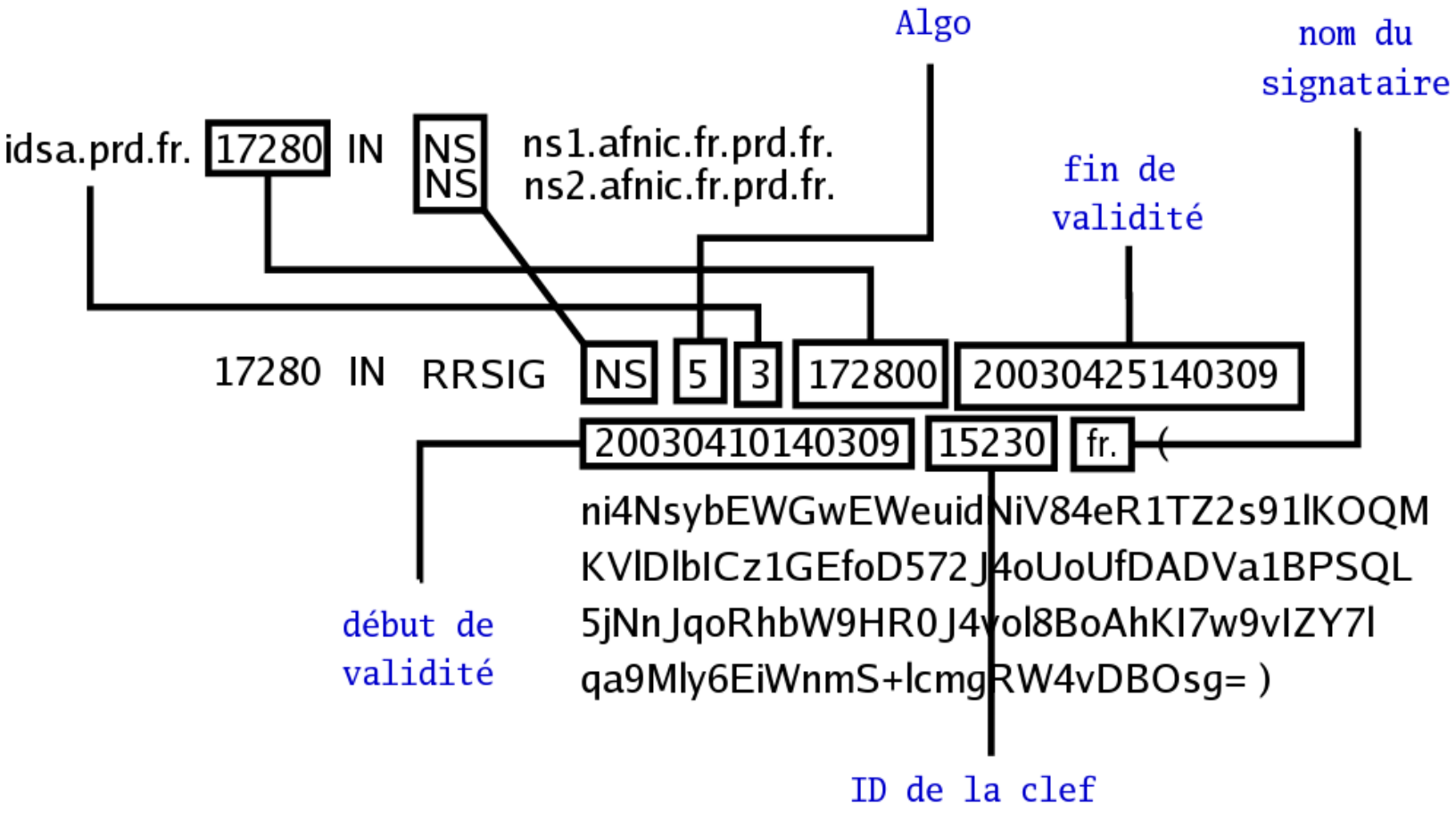
Description du RRSIG RDATA

Le RR RRSIG 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 RRSIG : exemple



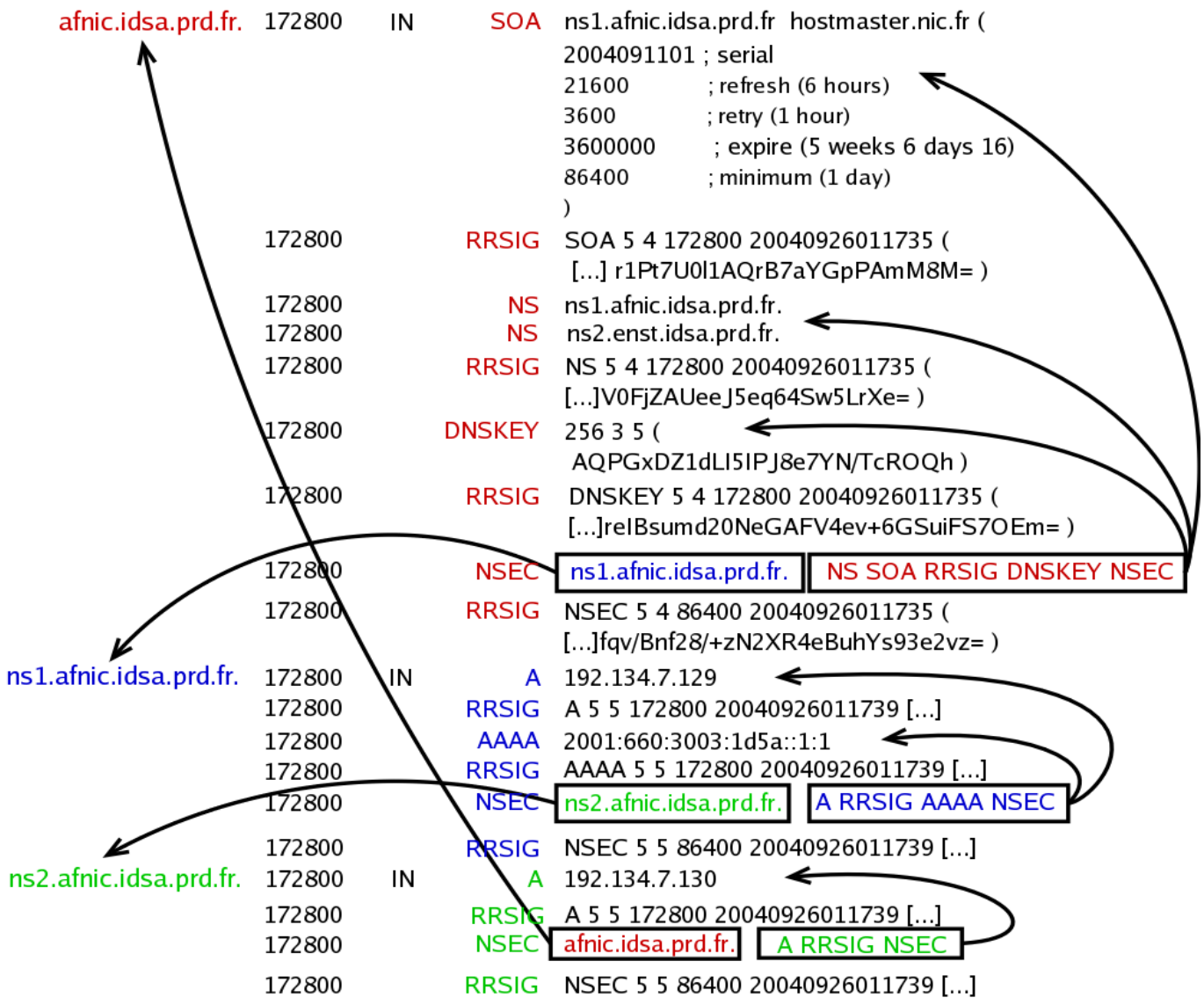
NSEC : nécessité

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 NSEC entre les noms.

Le RR NSEC d'un nom contient tous les types d'enregistrements associés à ce nom ainsi que le prochain nom présent dans la zone

NSEC : fonctionnement



NSEC : 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 NSEC du dernier nom pointe vers le premier nom de la zone (vision circulaire de la zone selon le chaînage NSEC)

NSEC : 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 NSEC ns1.afnic.idsa.prd.fr NS SOA RRSIG DNSKEY NSEC,

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 NSEC ns1.afnic.idsa.prd.fr NS SOA RRSIG DNSKEY NSEC,

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

NSEC : 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 (parcours de zone ou « DNS walking », trivialement mis en oeuvre)

Détection des wildcards possible : Si une réponse correspond à une wildcard étendue, le NSEC prouvant que le nom demandé n'existe pas explicitement est ajouté dans la réponse

Problème d'énumération avec NSEC

Beaucoup de zones DNS ne sont pas distribuées publiquement, seulement accessibles si on connait un nom de domaine.

Autrement, la zone est vendue, avec un contrat, et des conditions d'utilisation.

Exemple : .FR (10 000 euros par an)

Donc, pas question de laisser faire des récupérations gratuites avec « zone walking »

NSEC3

NSEC3 (RFC 5155) résout le problème : le chaînage n'est plus entre noms mais entre hachages des noms.

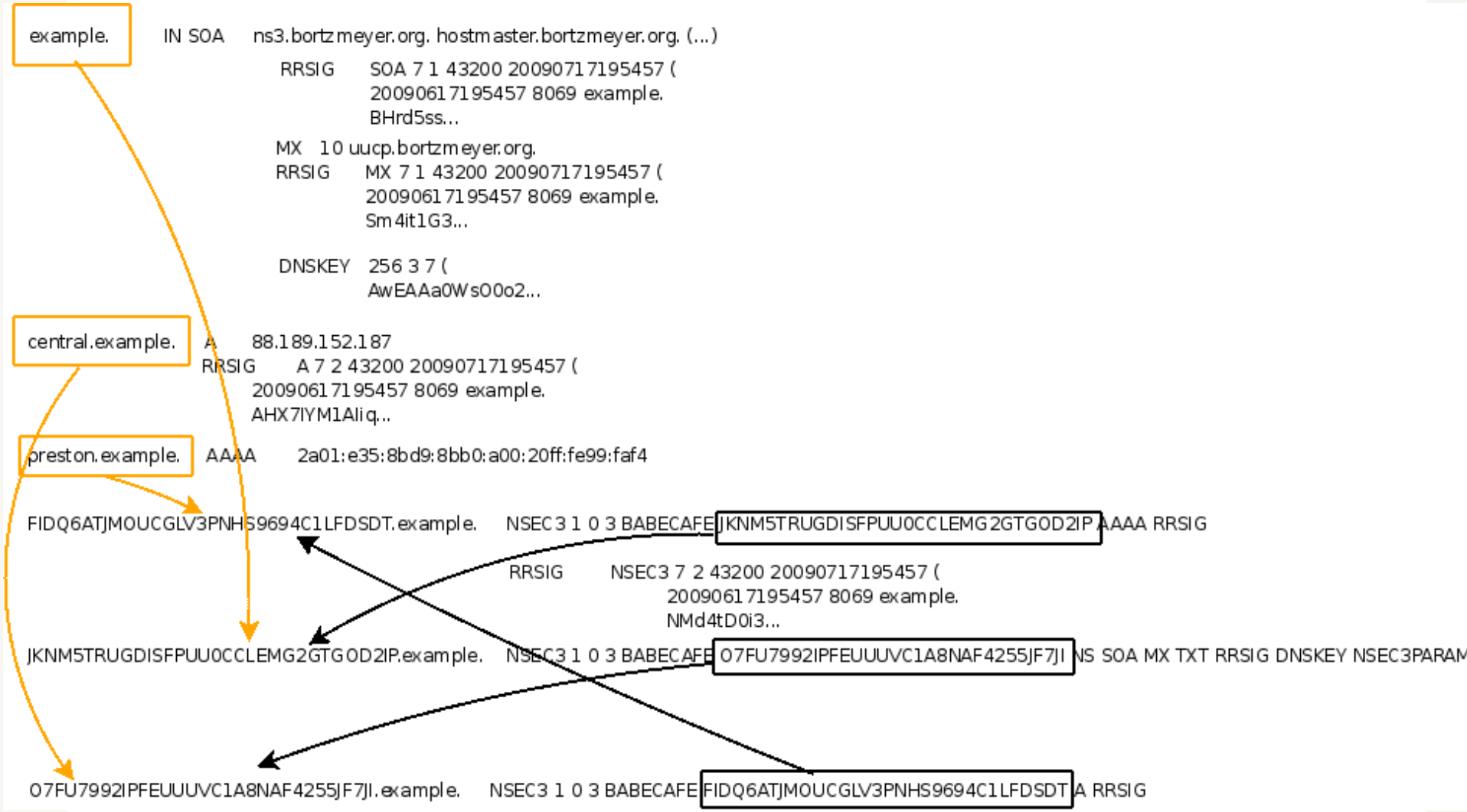
Si la question est le nom D1, et que le résolveur reçoit :
H0 NSEC3 H2 ...
H0 RRSIG ...

Il doit calculer H1, le hachage de D1 et vérifier que H0 < H1 < H2.

Ainsi, on peut prouver la non-existence, mais sans pouvoir suivre la chaîne.

NSEC3 est aujourd'hui mis en oeuvre dans BIND, NSD et Unbound.

Fichier signé avec NSEC3



Requête NSEC3

[Le résumé de foobar.example est RG2LAVDHFJ9DET479MSDUQ5Q0N3DRT99.]

```
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN,
    id: 11884
...
;; QUESTION SECTION:
;foobar.example.                IN    AAAA
...
O7FU7992IPFEUUUVC1A8NAF4255JF7JI.example.
    43200   NSEC3 1 0 3 BABECAFE
    FIDQ6ATJMOUCGLV3PNHS9694C1LFDSDT A RRSIG
```

Niveau de sécurité local (côté client)

La connaissance de la clé publique d'une zone permet de vérifier les signatures et donc l'authenticité et l'intégrité des informations contenues dans la zone

Concept de clé de confiance

Limitations : nécessite la connaissance des clés de toutes les zones avec lesquelles le resolver est susceptible de communiquer

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

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 faisant autorité sur 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 parente 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

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 parent 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 clé associée au DS

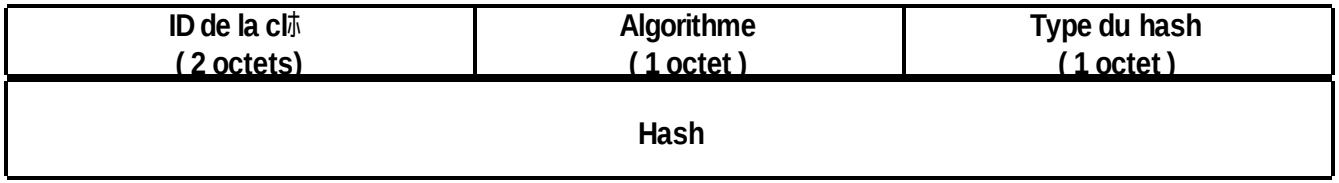
L'absence de DS, prouvée par le contenu du NSEC, lui même signé, prouve qu'aucune délégation sécurisée n'a été établie vers la zone fille.

Le RR DS

RR de sécurisation d'une délégation (les RR NS n'étant pas signés)

DS = Delegation Signer (RFC 4034)

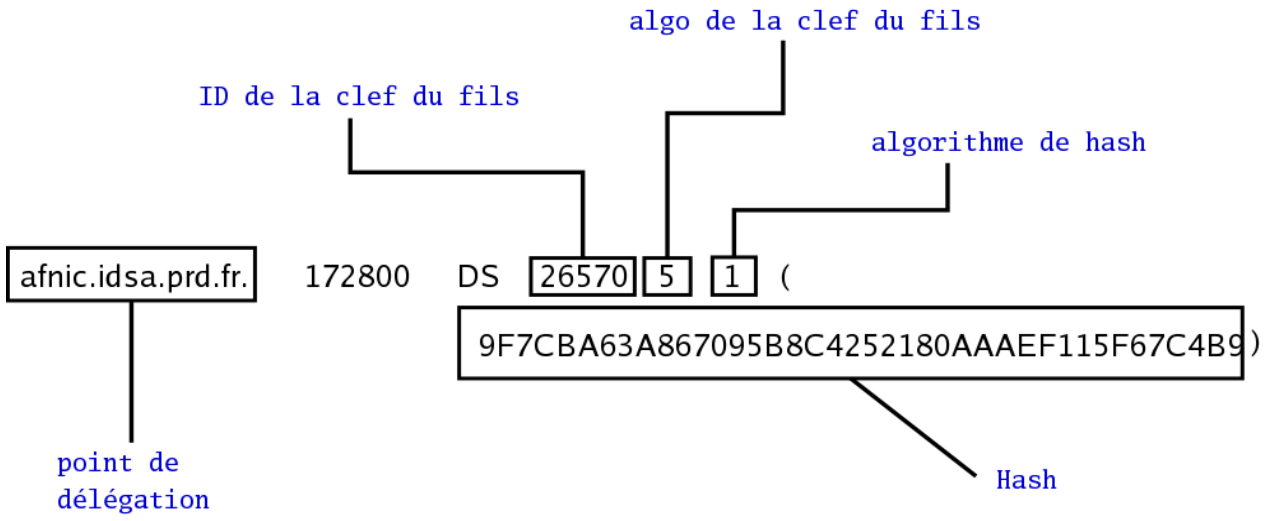
Format des RDATA :



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 depuis la zone parente n'est pas sécurisée)

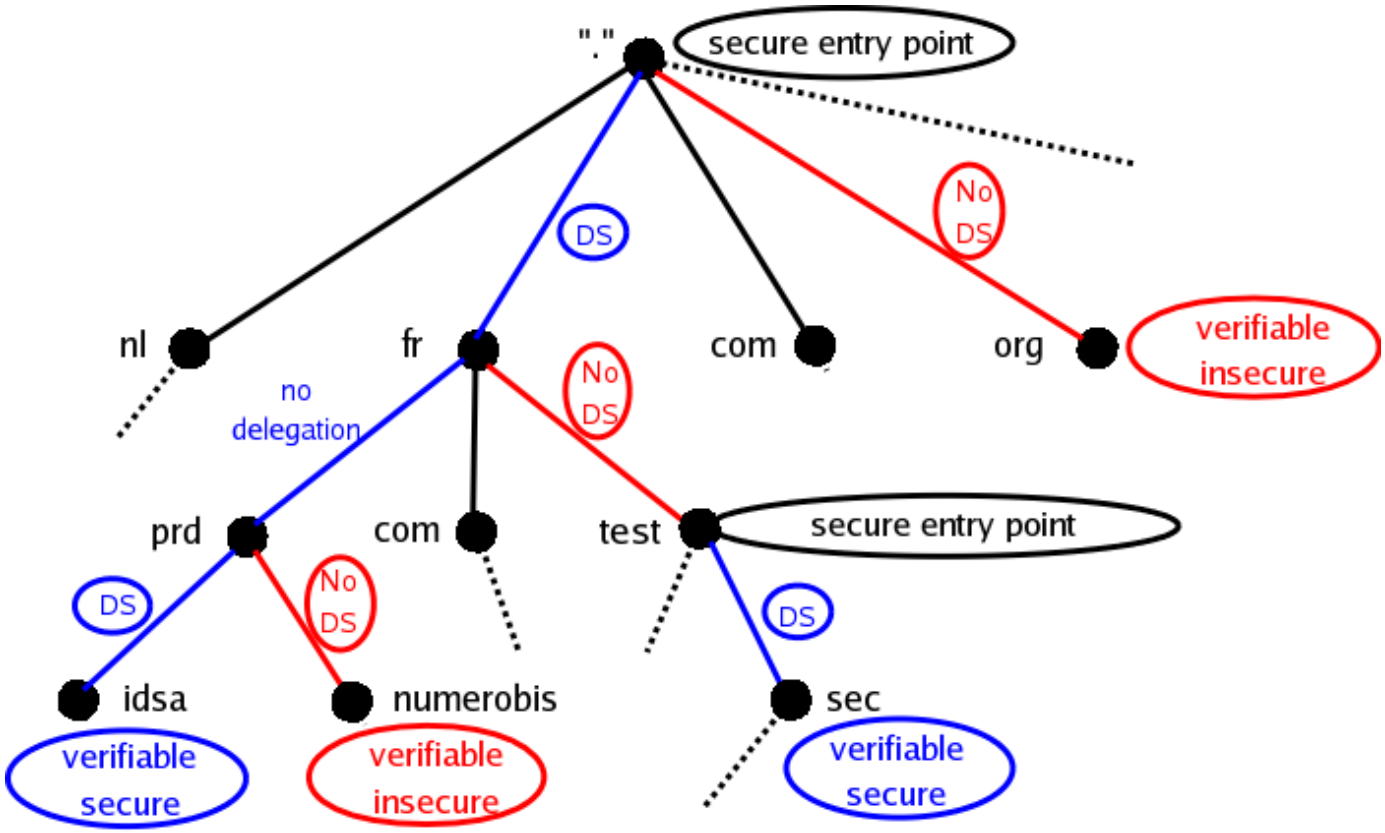
sécurisée globalement (signée, et on peut l'atteindre en parcourant une chaîne de confiance depuis une zone ancêtre)

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 une zone quelconque de manière sécurisée en parcourant les chaînes de confiance.

Arbre DNS partiellement sécurisé



DLV

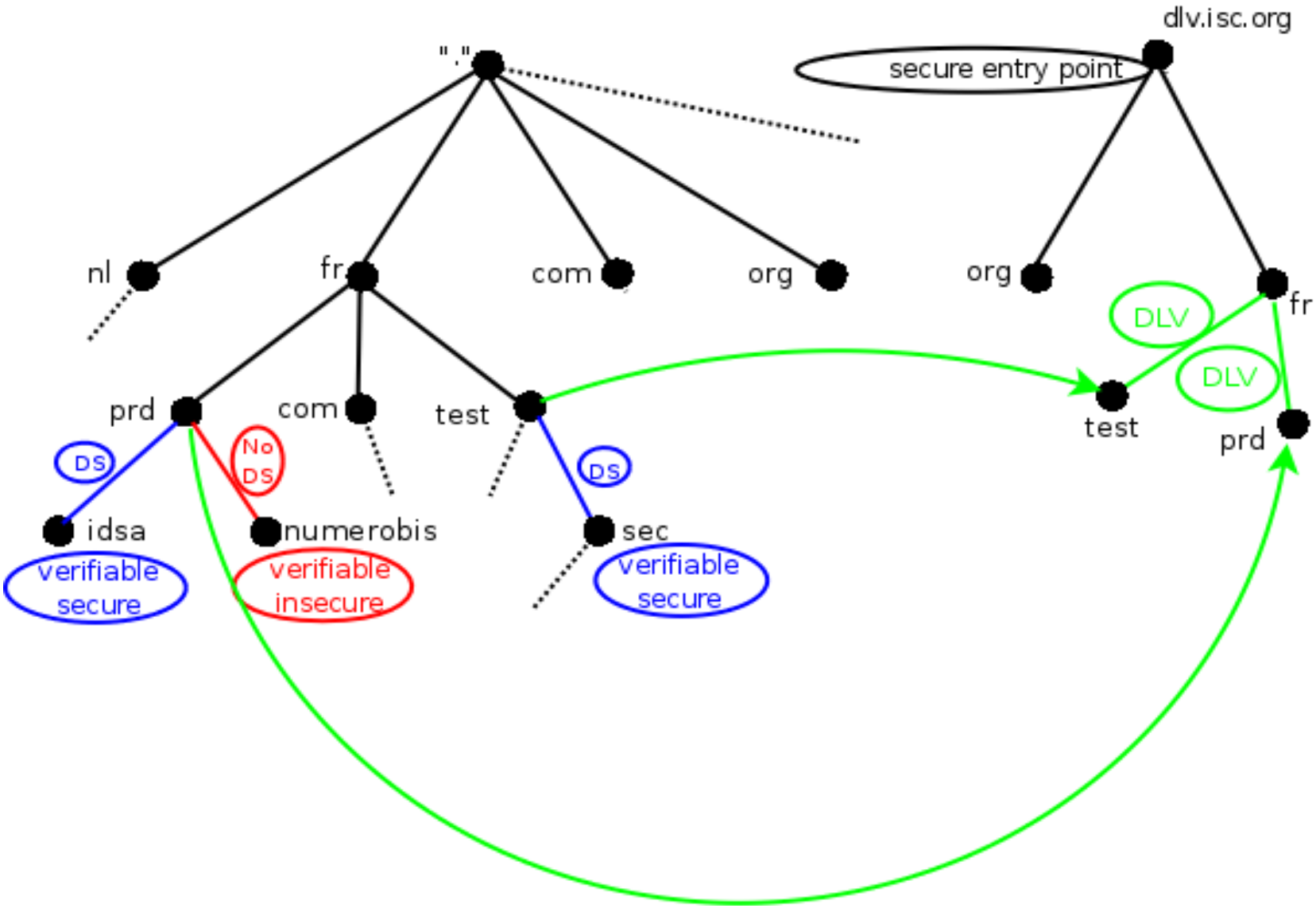
DNSSEC Lookaside Validation permet de séparer la racine de résolution (contrôlée par le gouvernement états-unien, via Verisign) et la racine de validation.

Un résolveur DLV ne demande pas les DS à la racine mais à dlv.example.org. Si sources.org est signé, le DS (en fait, un enregistrement DLV) sera en sources.org.dlv.example.org.

DLV permet donc de démarrer DNSSEC sans gérer à la main des clés et sans attendre la signature de la racine et de tous les TLD.

Aujourd'hui, le plus gros registre DLV est à l'ISC (Internet Systems Consortium, les auteurs de BIND).

Arbre sécurisé avec DLV



Importance d’un modèle à deux clés

Les clés 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 clés: ZSK (Zone Signing Key) et KSK (Key Signing Key)

Séparer les rôles :

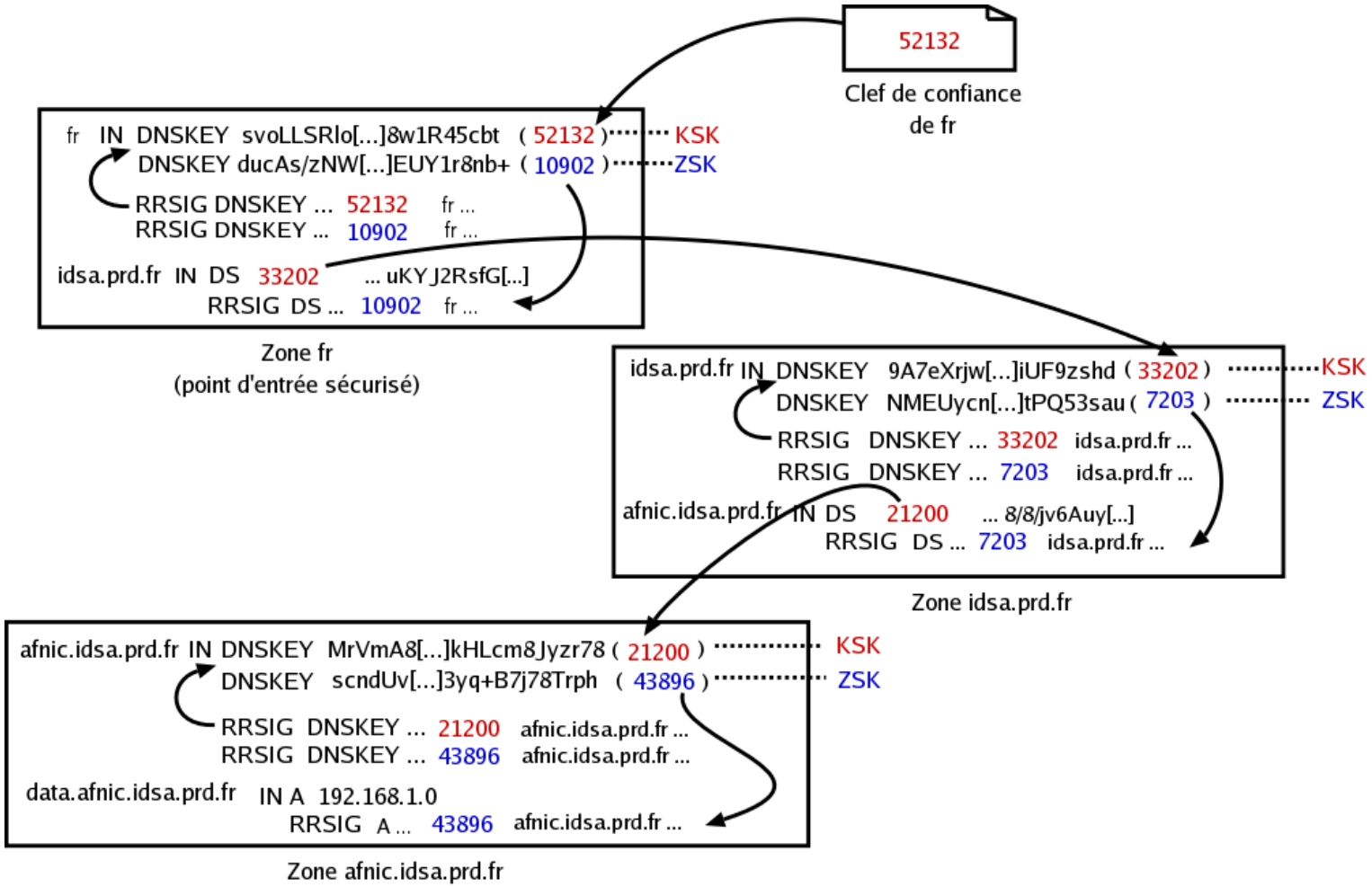
ZSK : Clé qui signe les enregistrements d'une zone. C'est une clé de taille réduite que l'on changera fréquemment.

KSK : Clé qui fait office de maillon de confiance. Elle ne signe que le DNSKEY RRset donc elle peut être de taille plus longue, ce qui permet de limiter sa fréquence de renouvellement (roulement).

Solution pour différencier leur structure : ajout d'un flag dans le RDATA (RFC 3757 : SEP)

Flexibilité accrue dans la relation zone parente / zone fille

Authentifications en cascade dans une chaîne de confiance



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

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 clés

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 RRSIGs, 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 clé + post-suppression de l'ancienne clé

Roulement ZSK planifié : schéma

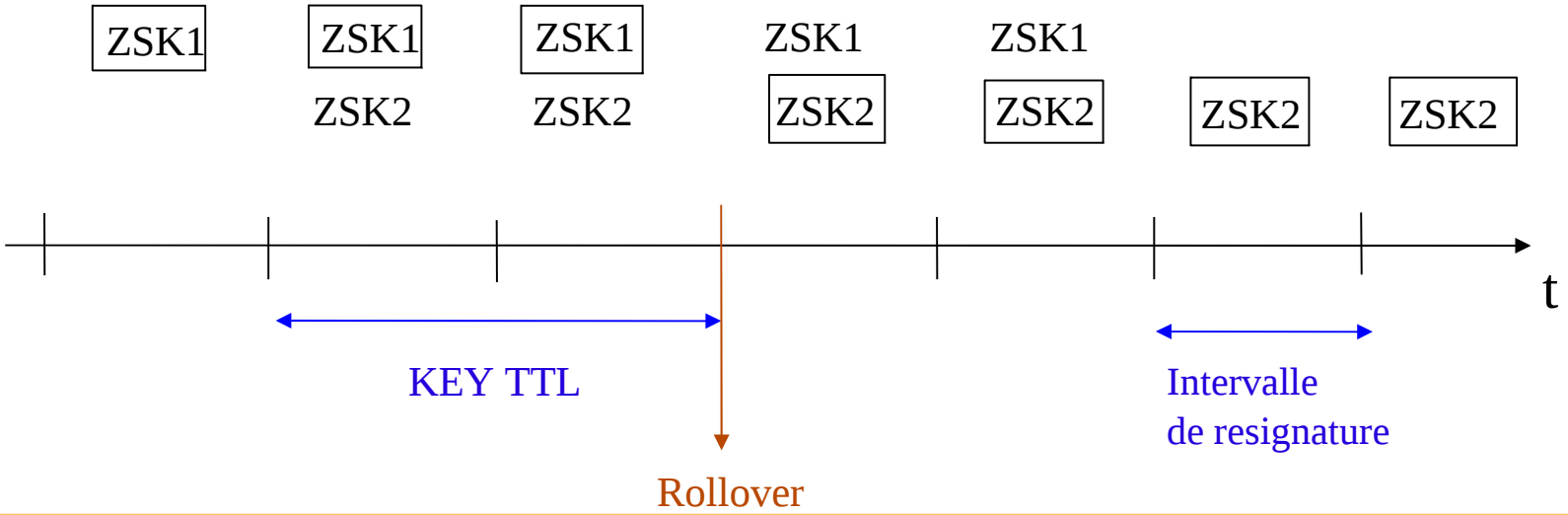
Exemple :

on roule la ZSK toutes les semaines (rappel : les durées optimales font l'objet d'un débat. Leur valeur idéale dépend de beaucoup de choses).

on resigne tous les jours

le TTL est de 2 jours

Dans ce cas, une clé 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

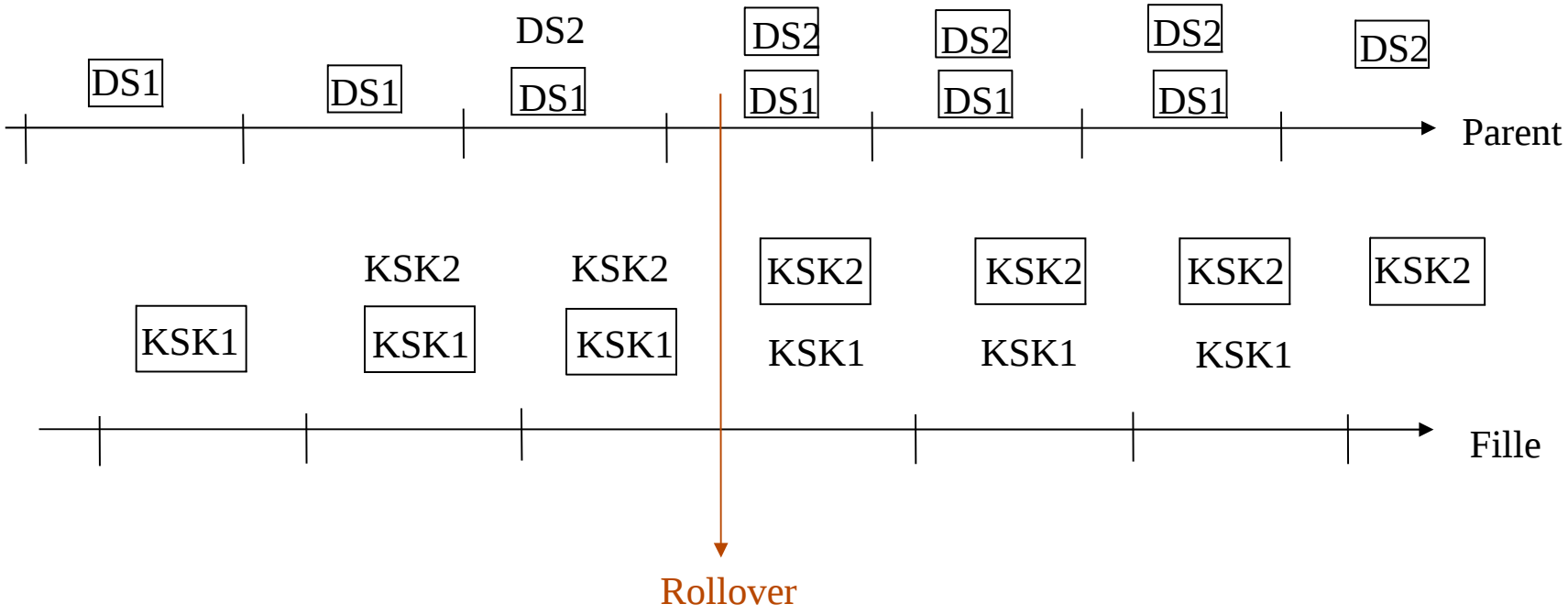
Communiquer sur le changement de clé car certains résolveurs avaient configuré l'ancienne clé comme clé de confiance

Ce roulement nécessite une bonne synchronisation des zones fille et parente

Roulement KSK planifié : schéma

Exemple: on change la KSK une fois par an, 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 clé immédiat) et risque d'attaques si conservation de la clé corrompue le temps de réaliser une procédure décrite dans les transparents précédents

Possibilité de publier en permanence dans le DNSKEY RRset une clé qui ne sera utilisée qu'en cas d'urgence

Considérations opérationnelles

Utilisation de BIND et ses outils

dnssec-keygen

dnssec-signzone

Performances

Temps de signature reste raisonnable même pour des zones de grande taille (vingt minutes pour .FR sur un Opteron)

Taille de la zone signée : multipliée par un facteur 6 à 10 par rapport au fichier non signé (sauf avec NSEC3 et opt-out, où l'augmentation est négligeable)

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

Outils de BIND

```
# Patience, ces outils consomment beaucoup
# d'entropie

# Faire les clés
% dnssec-keygen -a NSEC3RSASHA1 -b 2048  example
Kexample.+007+08069

# Signer la zone (ne pas oublier de resigner
# régulièrement !)
% dnssec-signzone -H 3 -3 babecafe -o example \
      db.example Kexample.+007+08069
```

DNSSEC comme PKI

Nécessité de stocker et distribuer les clés publiques utilisées par DNSSEC (RRset DNSKEY)

Possibilité de stocker des clés pour d'autres applications (IPSEC, SSH...)

Possibilité de stocker des certificats : RR CERT

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 déploiements en cours

Déploiements DNSSEC

Sept ou huit TLD signés sérieusement (.SE, .CZ, .BR, .GOV, .ORG, ...) dont deux en NSEC3

La racine signée en juillet 2010

Registre DLV à l'ISC opérationnel

SHA1 récemment remplacé par SHA2 à l'IETF

Portail DNSSEC : http://www.dnssec.net/

AFNIC et DNSSEC

L'AFNIC suit de près le développement de DNSSEC depuis le début.

L'étude pratique est actuellement activement menée pour permettre un déploiement en 2010.

Mais attention !

- **DNSSEC ne protège pas contre tous les risques (exemple : injection SQL dans un Bureau d'Enregistrement)**
- **DNSSEC vient aussi avec ses propres risques (la cryptographie, c'est difficile)**

A propos de ce document

Auteurs : {Bertrand.Leonard, Jean-Philippe.Pick, Mohsen.Souissi, Stephane.Bortzmeyer}@nic.fr

Questions