Sécurisation du DNS : les extensions DNSsec

Bertrand Léonard
AFNIC/projet IDsA (http ://www.idsa.prd.fr)
Bertrand.Leonard@nic.fr
date :09 octobre 2003

Résumé

Les extensions de sécurité DNS (Domain Name System), regroupées sous le terme DNSsec sont spécifiées à l'IETF[1] *par le biais de la RFC 2535, mais ces spécifications sont actuellement en cours de réécriture par le groupe de travail DNSext*[2]*. Dans ce cadre, une collection d'Internet-Drafts remplacera à terme la RFC2535.*
DNSsec a pour but de combler les failles de sécurité spécifiques au protocole DNS. En effet, malgré son rôle prépondérant dans l'internet actuel, le DNS restait vunérable à des attaques relativement simples à mettre en place mais aux conséquences très dommageables.
DNSsec est basé sur deux niveaux de sécurisation utilisant la cryptographie à clefs publiques. On peut qualifier le premier de local puisqu'il consiste en la signature des enregistrements d'une zone par une(des) clef(s) propre(s) à la zone. Le deuxième niveau de sécurité tire parti de la structure arborescente du DNS : on va mettre en place des chaînes de confiance en créant des liens d'authentification entre une zone mère et ses zones filles. Il va s'agir pour un parent d'authentifier la(les) clef(s) utilisées par ses fils. La connaissance d'un nombre restreint de clefs permet donc grâce à des authentifications en cascade d'accéder à tout enregistrement DNS de manière sécurisé.

Mots clefs

DNS, sécurité, cryptographie à clefs publiques, chaînes de confiance

1 Introduction

Jusqu'en 1984, toutes les associations entre les noms des machines connectées au réseau Internet (successeur d'ARPAnet) et leur(s) adresse(s) étaient contenues dans un simple fichier host.txt présent sur chaque machine. Ce fichier était géré par une autorité centrale, le SRI-NIC (Stanford Reasearch Institute- Network Information Center) chargé de recueillir les mises à jour et de diffuser régulièrement une version actualisée.
Cette manière de procéder est devenue totalement inadaptée devant la croissance rapide du nombre d'équipements connectés au réseau : les collisions entre noms ainsi que les difficultés inhérentes à une conception centralisée (mises à jour et diffusion des informations) ont rendu ce modèle obsolète. Le protocole DNS a donc été conçu en 1984 pour répondre à ces besoins : création d'un espace de nommage quasi infini grâce à un modèle arborescent, robustesse et performance ont été les priorités de conception ; la sécurité n'était à l'époque pas au centre des préoccupations.
Le DNS est donc devenu le deuxième catalyseur de l'expansion de l'internet après l'adoption du protocole TCP/IP quelques temps auparavant, et il est aujourd'hui l'un des piliers de son bon fonctionnement.
Mais parallèlement, les raisons de ce succès (notamment sa simplicité et son efficacité protocolaire) ainsi que son rôle critique dans l'internet moderne ont fait du DNS la cible idéale d'attaques simples mais aux conséquences pouvant être très néfastes. Ceci est d'autant plus problématique que le DNS est à la fois omniprésent et invisible dans l'utilisation grand public de l'internet actuel : un utilisateur qui croit accéder à un domaine en le désignant par son nom peut être redirigé sur la machine d'un pirate sans s'en rendre compte si l'entrée DNS correspondante a été corrompue.

2 Quelques rappels sur le DNS

Le DNS permet d'assurer le stockage et la distribution des informations relatives aux noms de domaines, les plus connues étant les associations entre les noms de domaine et les adresses IP.
Le protocole DNS à été conçu en 1984 par Paul Mockapetris et standardisé à l'IETF par le biais des RFC 1034 [1] et 1035 [2]. Nous donnerons dans cette partie quelques rappels sommaires sur le fonctionnement et les entités du DNS, il est conseillé de se référer aux RFCs pré-citées ou bien à des ouvrages comme [3].

[1] Internet Engineering Task Force. http ://www.ietf.org
[2] DNS Extensions working group. http ://www.ietf.org/html.charters/dnsext-charter.html

Son architecture repose sur un modèle client/serveur classique dans lequel le client, appelé *resolver* ou *résolveur*, est une bibliothèque de fonctions de résolution DNS située sur une machine cliente et le serveur est un serveur de nom.
Les requêtes effectuées par le client au(x) serveur(s) de noms interrogent la base de données qui contient les associations entre les noms de domaine et un certain nombre d'informations qui leur sont propres. Cette base de données, ou arbre de nommage, est la clef de voûte du DNS et a été conçue suivant trois caractéristiques principales :

– elle est hiérarchique, ce qui lui confère l'appellation d'arbre DNS où chaque nom de domaine est représenté par un noeud de l'arbre, la racine étant représentée par ".". Cette structuration en arbre inversé permet de relier la racine à un noeud quelconque de l'arbre par un chemin unique. Un nom de domaine est donc représenté par la succession d'étiquettes (*labels*) rencontrées sur le chemin reliant le noeud à la racine, séparés par des points.
ex :le noeud *enst* sur la figure 1 correspond au nom de domaine *enst.idsa.prd.fr*.
Un domaine est tout simplement un sous-arbre de l'arbre DNS débutant à un noeud spécifique et recouvrant l'arborescence située en dessous de ce point. ex : le domaine *idsa.prd.fr* est inclus dans le domaine *fr* (cf. figure 1).

– elle est distribuée : cette base de données est répartie sur un grand nombre de serveurs, chacun de ces serveurs étant en charge d'un sous-arbre de l'arbre DNS et des informations correspondantes. On évite ainsi la lourdeur d'un système centralisé où toutes les requêtes sont traitées par une base de données unique, même si celle-ci est répliquée sur plusieurs serveurs.
Cette décentralisation permet d'augmenter la flexibilité du système : une administration locale est plus à même de gérer les mises à jour des informations du sous-arbre dont elle est en charge. C'est pour cela qu'au sein d'un domaine, on choisit souvent de transférer la responsabilité de certains sous ensembles de noms, c'est ce qu'on appelle une délégation et cela a pour conséquence la création d'une nouvelle zone dite zone fille de la première (zone parente).
Une zone est la partie descriptive des informations DNS d'un sous-arbre. Ces informations sont contenues dans un fichier de zone stocké sur les serveurs qui sont dits autoritaires pour la zone et qui sont en charge de la mise à jour et de la diffusion de ces informations.
La robustesse du système bénéficie également de cette répartition : l'indisponibilité de certains serveurs n'affectera que les sous-arbres concernés.

– elle est redondante : la décentralisation des responsabilités s'accompagne évidemment d'une redondance des données pour augmenter la robustesse. Chaque zone est en fait prise en charge par plusieurs serveurs répartis géographiquement et topologiquement.
Parmi les serveurs autoritaires pour une zone, l'un d'entre eux (le serveur primaire) est chargé de transmettre une réplique du fichier de zone chaque fois qu'il est modifié aux autres serveur (les secondaires), une telle opération est appelée transfert de zone.

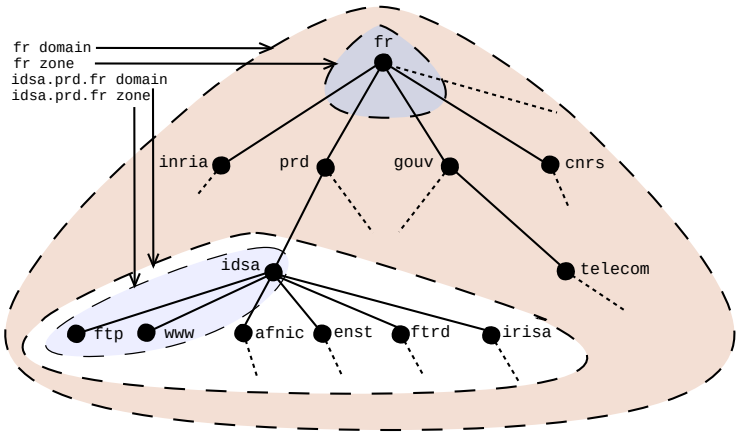

Figure 1 – *domaines,zones et délégations*

Les briques de base de l'information DNS sont les enregistrements, dits RRs (Resource Records). Un RR est une association entre un nom de domaine et une information se rapportant à ce nom (adresse IP, relai de messagerie, ...). Ces RRs peuvent être agencés de différentes manière selon les circonstances :

Ils sont d'une part regroupés en RRsets (ensemble de RRs correspondant au même nom et au même type d'information, par exemple dans le cas d'une zone ayant plusieurs serveurs de nom).
Ces RRsets sont regroupés en fonction de leur appartenance à une même zone pour former un fichier de zone qui sera stocké sur les serveurs autoritaires de cette zone.
Les requêtes DNS portent toujours sur les informations contenues dans ces RRsets, et donc un message DNS (paquet UDP) est composé d'un en-tête, d'une question et d'un certain nombre de RRsets répondant à cette question.
Les serveurs récursifs, quant à eux, stockent dans leur cache les RRsets pour lesquels ils ont déja effectué une recherche pour le compte d'un resolver, et ceci pendant la durée de vie de ces RRsets, le TTL (Time To Live).

3 Les failles de sécurité du DNS

Dans le cas du DNS, les problèmes de sécurité ne touchent pas (ou du moins ne devraient pas toucher) au domaine de la confidentialité des données. Il est utile de rappeler que le protocole DNS a été conçu suivant une philosophie selon laquelle les données DNS sont publiques et l'accès à ces données est universel. Le contrôle d'accès et la confidentialité ne sont donc pas plus à l'ordre du jour dans le DNS sécurisé (au sens protocolaire) que dans le DNS.
Néanmoins ces deux aspects de sécurité sont souvent disponibles dans les implémentations de serveurs de noms (dans BIND, le contrôle d'accès est géré par les acces-list et la confidentialité par les vues DNS). On utilisera donc directement ces options si l'on a ces besoins.

En revanche, un certain nombre de failles de sécurité peuvent perturber le bon fonctionnement du DNS et ainsi aller à l'encontre du cahier des charges du service DNS (qui doit, autant que possible, garantir la disponibilité, l'authenticité et l'intégrité des données DNS).

– La disponibilité des données peut être remise en cause par des attaques de type déni de service (DoS et DDoS) sur les serveurs de nom. On note toutefois que ce type d'attaques est un problème pouvant toucher tout type de serveur.
  La disponibilité des données peut également être affectée par le biais d'erreurs de configuration des serveurs : zones DNS mal chargées, problèmes de transfert de zone, manque de diversité topologique des serveurs répliques.
– L'authenticité et l'intégrité des données DNS peuvent être remises en cause par des attaques relativement simples comme nous allons le voir par la suite. Ce sont ces failles, lorsqu'elles sont imputables à des faiblesses protocolaires du DNS, qui devront être résolues grâce à DNSsec.
  Il est à noter que l'intégrité des données peut également être remise en cause par une intervention directe sur les fichiers de zone présents sur les serveurs si ceux-ci ne sont pas sécurisés de manière adéquate : c'est le cas de la falsification du fichier de zone sur la figure 2.

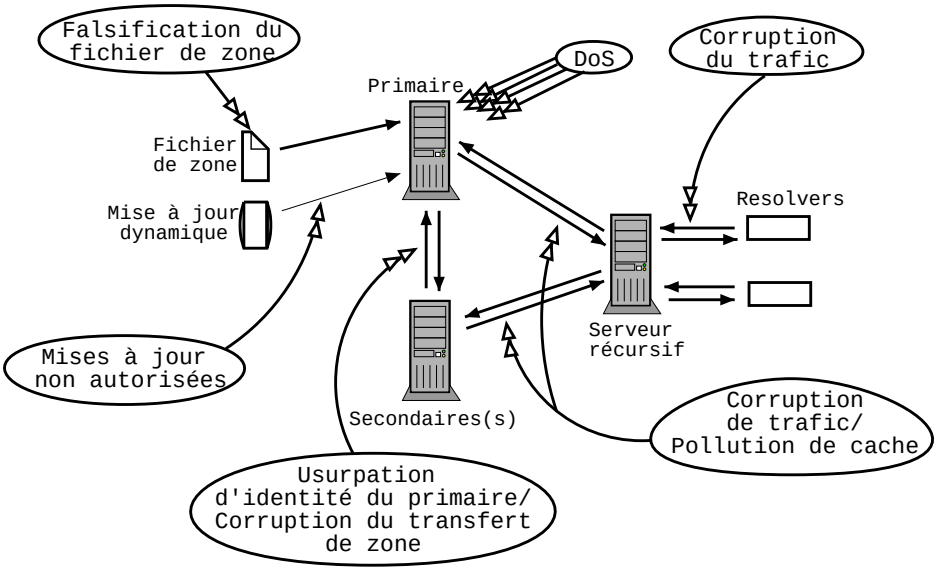

Figure 2 – *vulnérabilités de l'architecture DNS*

Les buts des attaques sur le DNS sont multiples, et les conséquences peuvent aller de l'indisponibilité d'une ressource au vol d'informations critiques (emails, mots de passes etc..). Il est à noter que les attaques sur le DNS ne sont souvent que des prémices à des attaques plus graves ayant pour but de contourner d'autres types de sécurisation par ailleurs mis en place.
On peut classer les objectifs des attaquants comme suit :

– perturber, ralentir ou bloquer le service DNS sur une partie de l'arbre (en général par l'utilisation d'attaques de type DoS ou DDoS) ;
– empêcher l'accès à certains équipements ou services pour des raisons politiques, économiques ou pour le plaisir (déni-de-domaine ou redirection fantaisiste) ;
– rediriger les utilisateurs ou leurs communications (emails) à leur insu vers des hôtes contrôlés par le pirate ;
– récupérer des informations critiques (logins/mots de passe) en se faisant passer pour le serveur (telnet, ftp, etc..) auquel l'utilisateur croit se connecter.

Toutes ces attaques ont généralement une première étape en commun que l'on désigne souvent par le terme de DNS Spoofing (usurpation d'identité) ou DNS Hijacking. Le DNS Spoofing est un terme générique pour désigner une attaque dans laquelle un attaquant répond à une requête DNS à la place du serveur interrogé, pouvant ainsi tromper l'initiateur de cette requête à sa guise et lui fournir des informations erronées. Par ce type de procédé, un attaquant peut même intervenir directement sur les serveurs autoritaires en corrompant les transferts de zone ou les mises à jour dynamiques comme le montre la figure 2.
En outre, ceci est facilité par la simplicité protocolaire du DNS : les messages ont une structure préformatée simple et sont transportés dans un simple paquet UDP, non chiffré, non signé, ce qui rend très facile la génération de faux paquets et leur insertion dans le trafic DNS.
Nous allons maintenant étudier de plus près ces attaques spécifiques au DNS [4] qui correspondent sur la figure 2 à la corruption de trafic et la pollution de cache.

Le but de l'attaquant est donc de répondre à la requête d'un utilisateur avant le serveur supposé répondre, il pourra ainsi fournir à la victime les informations de son choix ; il existe plusieurs possiblités d'action :

– attaque de type *man in the middle* : l'attaquant à la possibilité de sniffer la requête de la victime (écoute sur le réseau par exemple), il lui suffit donc de répondre plus rapidement que le serveur interrogé en utilisant le même DNS ID (référence de la requête) dans l'en-tête du message et en incluant les réponses corrompues désirées.

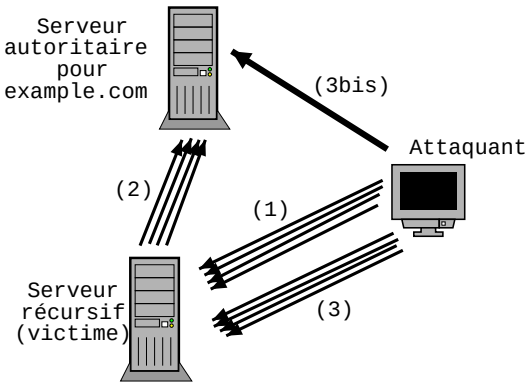

Figure 3 – *birthday attack*

– deviner les requêtes et IDs de la victime : souvent l'attaquant n'est pas présent sur le même réseau physique que la victime, il n'a donc pas directement accès aux requêtes et ID associées de la victime.
  Dans ce cas, le but de l'attaquant va être de forcer la victime à poser une question donnée, et dans le même temps deviner l'ID utilisé.
  Le meilleur moyen de deviner la question est de la provoquer : si la victime est un serveur récursif, l'attaquant posera une question qui doit provoquer l'interrogation du serveur autoritaire correspondant per le serveur récursif.
  Pour deviner l'ID utilisé (en dehors des anciens serveurs qui se contentaient d'incrémenter des IDs à chaque nouvelle requête), c'est plus difficile. Mais il existe une méthode qui donne d'assez bons résultats, elle est basée sur le *birthday*

paradox[3] et est décrite sur la figure 3.
L'attaquant envoie n fois la même requête (www.example.com, A) au serveur récursif victime (1). En tant que serveur récursif, la victime traitera ces n requêtes successivement en envoyant n requêtes au serveur autoritaire adéquat (ns1.example.com), en utilisant n IDs différents (2).
L'attaquant envoie alors n réponses contenant l'information corrompue (www.example.com=1.1.1.1) en utilisant des IDs différents (3), on accompagne souvent cela d'un déni de service sur ns1.example.com (3bis) pour ralentir l'arrivée des vraies réponses. Cette méthode exploite le concept du birthday paradox et permet avec un nombre n modéré d'obtenir de très bons résultats : n=300 donne une probabilité supérieure à 50% de trouver le bon ID.

– Pollution de cache : les serveurs récursifs, qui jouent un rôle central en ce qui concerne les performances du modèle DNS (mutualisation des résultats des requêtes), sont de fait des victimes idéales pour les attaquants puisqu'ils gardent en cache les données pendant la durée de leur TTL. Si un attaquant arrive à polluer un cache, il multiplie donc le nombre d'utilisateurs potentiellement victimes d'informations corrompues.
L'une des méthodes précédemment décrites peut donc être utilisée pour polluer un cache, mais il existe également d'autres méthodes que nous ne détaillerons pas ici (ex : utilisation des sections additionnelles d'un message DNS).

4 Les principes de DNSsec

Il est donc devenu évident qu'un acteur aussi critique et vulnérable que le DNS devait être sécurisé. Les extensions de sécurité au protocole DNS ont été spécifiées à l'IETF en 1999 par le biais de la RFC 2535 : Domain Name System Security Extensions[5]. Cette première version a jeté les bases d'une méthode de sécurisation du DNS, désormais plus connue sous le nom de DNSsec.
Il est à noter que ces extensions sont en cours de réécriture à l'IETF (groupe de travail DNSext) et une collection de documents rendra obsolète la RFC 2535 dans un futur proche ; l'un d'entre eux, consacré à DS (Delegation Signer, en passe de devenir proposed standard, RFC xxx[6] : numéro en cours d'attribution), propose une méthode de sécurisation des délégations bien plus performante que la méthode originelle et dont on parlera plus en détails dans la suite de ce document.

4.1 Services fournis par DNSsec

DNSsec propose des extensions protocolaires s'appuyant sur l'utilisation de signatures cryptographiques (cryptographie à clefs publiques) pour protéger le DNS en fournissant les services suivants :

– sécurisation des transactions DNS ;
– sécurisation des informations contenues dans les messages DNS par le biais de l'authentification de leur origine ainsi que la garantie de leur intégrité durant le transport ;
– stockage et distribution des clefs nécéssaires au bon fonctionnement des deux premiers services cités ci-dessus.

D'autre part, comme on va le voir par la suite, le déploiement de DNSsec ne peut être envisagé que de manière progressive. Ainsi, l'une des contraintes majeures auxquelles étaient soumis les concepteurs de DNSsec fut le respect de la compatibilité ascendante (*backward-compatibility*) de DNSsec avec le protocole DNS. DNSsec ne doit donc pas changer la nature intrinsèque de l'information DNS : tous les nouveaux objets nécéssités par DNSsec suivent le format RR comme on le verra dans la section 4.4, et les messages DNS restent identique ; certains ajustements toutefois nécessaires ont dû être intégrés, ils seront abordés dans la section 7.2.

4.2 Le niveau de sécurité local

Le fonctionnement de DNSsec est basé sur la cryptographie à clefs publiques, dont l'utilisation ne nécessite pas la connaissance préalable de secrets partagés. Chaque zone va générer un ensemble de paires de clefs (privées/publiques). Ces clefs sont associées aux zones et non aux serveurs stockant ces zones.

– Les parties privées sont utilisées pour signer toutes les informations (Resource Records sets ou RRsets) faisant partie intégrante de la zone (les points de délégation et les glues par exemple ne sont pas signés dans la zone mère). Les signatures correspondantes sont stockées dans le fichier de zone à l'aide d'un nouveau type d'enregistrement : le RR SIG.

[3] sur un groupe de 23 personnes et plus, la probabilité que deux personnes soient nées le même jour est supérieure à 50%

La figure 4 illustre un fichier de zone signé.
– Les parties publiques des clefs sont stockées dans le fichier de zone à l'aide d'enregistrements KEY. Elles pourront donc être récupérées par le biais de requêtes DNS classiques et utilisées pour la vérification des signatures.

Ainsi la connaissance des clefs publiques d'une zone permet la vérification de l'intégrité des données signées reçues ainsi que l'authentification de l'origine de celles-ci (puisque la partie privée de la clef n'est connue que du signataire de la zone).
Le cas très spécifique de l'authentification des réponses négatives (réponses à des requêtes portant sur des noms ou des enregistrements qui n'existent pas) est résolu par la création d'un nouveau type d'enregistrement DNS : NXT. Les RR NXT sont insérés dans le fichier de zone entre deux noms consécutifs. Un RR NXT indique tous les types de RRs associés au nom considéré, ainsi que le prochain nom situé dans le fichier de zone, nous en parlerons plus en détails dans la section 4.4.

La seule contrainte pour un resolver désirant effectuer un contrôle des signatures des données reçues reste donc l'obtention des clefs publiques correspondantes de manière sécurisée. Plusieurs possiblités sont envisageables :

– la configuration manuelle des clefs dans le resolver ;
– l'obtention des clefs hors-bande (sans passer par des requêtes DNS classiques) ;
– l'obtention des clefs par résolution DNS classique, à la condition que celles-ci puissent être authentifiées par d'autres clefs en lesquelles le resolver a déjà confiance.

Il est évident qu'un resolver ne peut connaître et avoir confiance en toutes les clefs des zones dans lesquelles il ira chercher des informations. C'est donc ici que l'on va profiter de la nature arborescente du DNS et de l'existence des délégations.

4.3 Le niveau de sécurité global

Le principe de base est l'authentification des clefs en cascade : si un resolver fait confiance à une clef, alors toute clef signée par celle-ci pourra elle-même être considérée comme digne de confiance. Ce procédé pouvant se répéter autant de fois que désiré, on pourra faire confiance à toute clef pouvant être reliée à la première par une suite de clefs telle que la clef n est authentifiée par la clef n-1 et authentifie la clef n+1. On dit dans ce cas que l'on construit une chaîne de confiance reliant une clef que l'on désire authentifier à un point d'entrée sécurisé (une clef en laquelle on a préalablement confiance).

Le DNS, de par sa nature arborescente présente une structure idéale pour la mise en place d'un tel procédé : la(les) clef(s) d'une zone fille sont authentifiées par la(les) clef(s) de sa zone parente créant ainsi une délégation sécurisée. Ainsi la confiance en une clef d'une zone donnée permet à un resolver de faire également confiance à une clef d'une zone située en aval dans l'arbre DNS à condition que les délégations successives reliant ces deux zones soient sécurisées.

A terme, si l'arbre DNS est entièrement sécurisé, la simple connaissance des clefs de la racine permettra à un resolver d'obtenir une information quelconque de l'arbre DNS de manière sécurisée : en construisant une chaîne de confiance entre la clef de la zone racine et la clef de la zone contenant l'information à authentifier, puis en utilisant cette clef pour vérifier la signature de cette information.
On peut donc vérifier les informations DNS d'une zone de deux manières :
– en connaissant (et en ayant confiance en) la clef publique de la zone et en procédant à la vérification des signatures des données ;
– en connaissant la clef publique d'une zone en amont dans l'arbre DNS et en établissant une chaîne de confiance vers la clef de la zone considérée, puis en utilisant cette clef pour vérifier les données.

4.4 Les nouveaux RRs

Comme on vient de le voir, DNSsec a nécéssité la mise en place de nouveaux RRs. Ces RRs ont bien entendu la même structure que les RRs classiques et vont différer par le format de l'information qu'ils contiennent. Ces nouveaux RRs sont décrits en détails dans [8].

– Les RRs KEY stockent les parties publiques des paires de clefs. Ils peuvent donc être récupérés par résolution DNS classique, chaque fois que l'on aura besoin d'effectuer des vérifications de signature.
  Il a été jugé judicieux (mais pas obligatoire) de distinguer les clefs avec lesquelles on va signer les informations d'une zone, des clefs qui vont servir à établir les chaînes de confiance.
  On utilise le terme de ZSK (Zone Signing Key) pour désigner les clefs qui vont signer les RRsets d'une zone. L'autre type

de clef est appelé KSK (Key Signing Key). Les KSK seront donc les maillons intermédiaires entre les zones : la KSK d'une zone est authentifiée par la zone parente, et ne sera utilisée dans la zone fille que pour signer le KEY RRset (qui contient la KSK et la(les) ZSK(s)).

Cela a pour conséquence de cloisonner les niveaux de sécurité, local et global par l'utilisation de clefs distinctes et notamment faciliter les opérations de roulement de clefs : si une zone désire changer de ZSK, elle n'aura pas besoin d'en référer à sa zone mère puisque la KSK restera toujours authentifiée.
Ce procédé n'est pas obligatoire et l'on peut se contenter d'utiliser la même clef en tant que ZSK et KSK mais nous conseillons ici l'utilisation du modèle KSK/ZSK et dans la suite du document les zones considérées seront sécurisées suivant ce modèle.

– Un RR SIG stocke la signature d'un RRset donné par une clef donnée ; chaque RRset d'une zone sera accompagné d'autant de signatures qu'il y a de ZSKs actives dans la zone. Le RRset KEY est quant à lui signé par les ZSKs et la KSK.
Pour éviter les conflits de signatures entre plusieurs zones, on ne signe au sein d'une zone que les RRsets appartenant vraiment à la zone : les points de délégations (RRsets NS d'une zone fille situés dans la zone mère) sont déjà signés dans la zone fille, et les glues sont signées dans leur zone d'appartenance réelle.
Toutes ces signatures ont bien entendu un intervalle de temps de validité en dehors de laquelle elles sont jugées invalides.

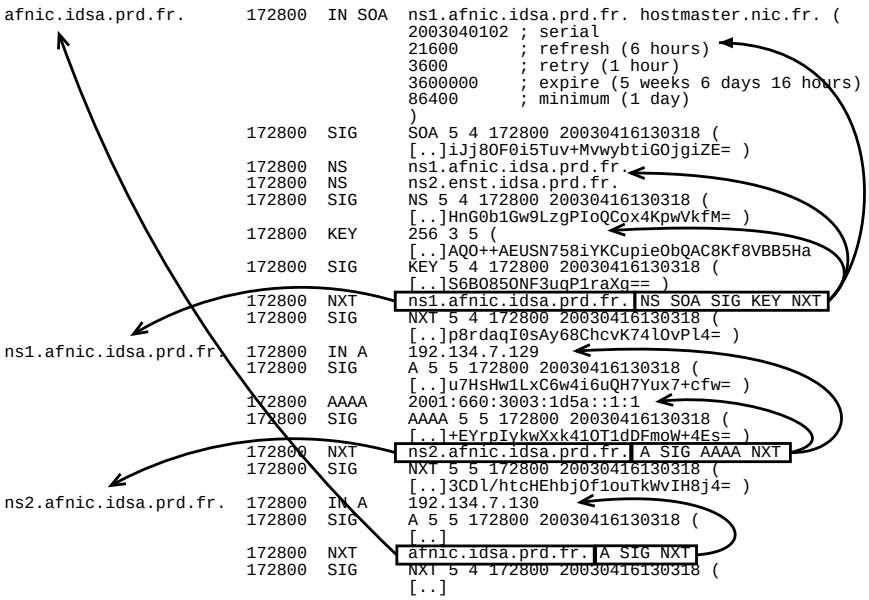

Figure 4 – *fichier de zone signé, mise en évidence des enregistrements NXT*

– Les réponses DNS négatives, correspondant à des requêtes portant sur des enregistrements ou des noms qui n'existent pas doivent être authentifiées au même titre que les réponses positives. Dans le shéma DNSsec évoqué plus haut, on signe les RRsets, ce qui est incompatible avec le fait qu'une réponse négative ne comporte aucun RR à signer (la seule présence d'un flag NXDOMAIN dans l'en-tête caractérise une réponse négative dans le DNS).
On a donc créé un nouveau type de RR, l'enregistrement NXT que l'on insère entre les différents noms contenus dans une zone. Un enregistrement NXT est associé à un nom et indique dans son RDATA la liste des types d'enregistrements associés à ce nom, ainsi que le nom suivant présent dans la zone. Cette notion de nom suivant nécessite un ordonnancement canonique préalable des noms dans la zone.
Les enregistrements NXT sont bien entendu signés dans la zone au même titre que les autres enregistrements.
Ils sont utilisés de la manière suivante pour prouver la non existance d'un nom ou enregistrement :
  – pour une requête portant sur un enregistrement n'existant pas, le serveur renvoie le NXT du nom correspondant, ainsi que la signature associée, ce NXT indique tous les types de RRs associés à ce nom ;
  ex : si on considère la zone de la figure 4, une requête du type "*afnic.idsa.prd.fr, A*" renverrait le NXT de *afnic.idsa.prd.fr*.
  – pour une requête portant sur un nom qui n'existe pas, le serveur renvoie le NXT du nom précédant dans la zone, on sait alors qu'entre ce nom et le nom indiqué dans le NXT, il n'existe aucun autre nom.
  ex : une requête du type "*ns3.afnic.idsa.prd.fr, A*" renverrait le NXT de *ns2.afnic.idsa.prd.fr* qui indique que le prochain

nom de la zone est *idsa.prd.fr*.

– le RR DS sera très largement abordé dans la section suivante.

5 Délégation et chaînes de confiance dans DNSsec

5.1 Le modèle DS

L'établissement de ces chaînes de confiance et la manière de sécuriser les délégations ont récemment été remodelées et la RFC2535 mise à jour par un document (bientôt à l'état de proposed standard, RFC xxx) : Delegation Signer (DS)[6].

Le DS est un enregistrement concernant une zone fille, mais localisé dans la zone parente créant ainsi un lien sécurisé entre ces deux zones. Il contient un hash de la KSK de la zone fille et est signé par la ZSK de la zone mère.
La clef de la zone mère authentifie le DS de la zone fille, qui authentifie la KSK de la zone fille, qui elle-même signe le KEY RRset de la zone fille : la délégation sécurisée est en place.

Le modèle d'une chaîne de confiance sur trois niveaux (zones fille, mère et grand-mère possédant chacune une ZSK et une KSK) est donc le suivant :
(i) les RRsets de la zone fille sont signés par la partie privée de la ZSK de la zone fille ;
(ii) la partie publique de la ZSK de la zone fille est signée par la partie privée de la KSK de la zone fille ;
(iii) la partie publique de la KSK de la zone fille est authentifiée par la zone mère en générant le RR DS correspondant et en l'incluant dans le fichier de zone mère ;
(iv) ce DS est signé dans la zone mère par la partie privée de la ZSK de la zone mère ;
(v) la partie publique de la ZSK de la zone mère est signée par la partie privée de la KSK de la zone mère ;
(vi) la KSK de la zone mère est authentifiée par le DS correspondant dans la zone grand-mère ;
(vii) ce DS est signé par la ZSK de la zone grand-mère ;

Ainsi, si un résolveur est configuré avec la KSK de la grand-mère comme clef de confiance, il pourra vérifier les informations de la zone fille en construisant la chaîne de confiance décrite précédemment. Sur la figure 5, on a un exemple d'une chaîne de confiance reliant la clef de confiance (KSK) de la zone *fr* aux enregistrements (RRsets) de la zone petite-fille *afnic.idsa.prd.fr*.

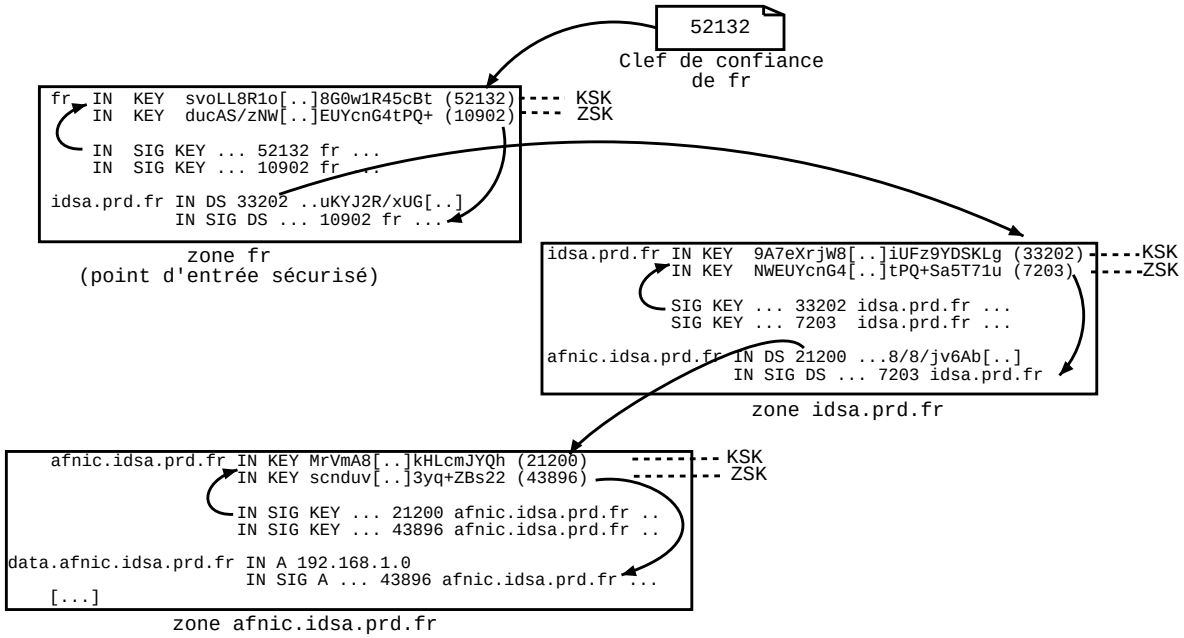

Figure 5 – *authentifications en cascade dans une chaîne de confiance*

Grâce au modèle DS, les délégations sécurisées sont relativement simples à mettre en place : une zone fille se contente d'envoyer sa KSK à sa zone parente qui génère le DS correspondant, le signe et l'inclut dans sa zone. C'est donc une procédure qui ne nécessite qu'un seul échange, alors que dans le modèle RFC2535, la zone fille envoyait sa clef à la zone mère qui la signait puis lui renvoyait signée.
De plus l'existance ou non du DS dans une zone mère définit immédiatement le statut de la zone fille : si pour une délégation vers une zone donnée, le DS est inexistant (cette non-existance étant prouvée par le RR NXT adéquat au point de délégation) on sait immédiatement que la zone fille n'est pas digne de confiance par le biais d'une chaîne de confiance (*verifiable insecure*, cf partie 5.2).
D'autre part, avec le modèle KSK/ZSK, le fait que la zone mère ou la zone fille change de ZSK n'affectera pas cette délégation sécurisée. On se contentera de re-signer les zones, sans avoir besoin de modifier le DS, et donc sans échange entre zones mère et fille, puisque la KSK reste la même.

Les chaînes de confiance sont composées de zones sécurisées localement grâce à leur clefs propres reliées entre elles par les maillons que sont les délégations sécurisées. Ces délégations sécurisées sont à mettre en place avec le plus grand soin par le duo zone mère/zone fille car elles peuvent affecter plus généralement le bon fonctionnement de DNSsec en perturbant par exemple une chaîne de confiance reliant un point d'entrée sécurisé situé en amont de la zone mère avec une zone située en aval de la zone fille.
La transmission de la KSK par la zone fille et la mise en place du DS par la zone mère doivent donc être faits avec beaucoup de précautions.

5.2 Classification des informations DNS

Dans le cadre d'un arbre DNS partiellement sécurisé, on peut classer les zones en trois catégories de manière objective suivant leur statut :

- les zones non sécurisées : elles ne sont pas signées ;

- les zones sécurisées localement : elles sont signées mais non reliées à leur zone parente par une délégation sécurisée. Dans ce cas on ne peut vérifier la véracité des informations que si l'on fait confiance à la KSK de cette zone ;

- les zones sécurisées globalement : elles sont signées et on peut les atteindre par le biais de délégations sécurisées. Il y a plusieurs niveaux de sécurité globale en fonction du plus haut point vers lequel on peut remonter dans l'arbre par le biais de délégations sécurisées (au maximum jusqu'à la racine).

D'autre part si on introduit une subjectivité en considérant les informations DNS du point de vue d'un resolver donné, on peut classer les données DNS en quatre catégories[9]. Cette classification est importante car elle permet à un résolveur effectuant une résolution DNS, couplée d'une vérification DNSsec de choisir l'attitude à adopter envers l'information en fonction du statut de cette dernière (lui faire confiance ou la rejeter notamment). Pour un serveur récursif, une telle classification est indispensable pour le stockage des données dans son cache.

- Les informations dont on sait qu'elles sont sûres (*verifiable secure*) :
  - le RRset est dans une zone signée par une clef en laquelle le resolver a confiance et la signature est valide ;
  - le RRset est dans une zone qu'on peut relier à un point d'entrée sécurisé pour le résolveur par une chaîne de confiance valide ; de plus la signature du RRset est valide ;

- Les informations dont on sait qu'elles ne sont pas sûres (*verifiable insecure*) :
  - la zone parente de la zone contenant le RRset considéré ne contient pas le DS correspondant pour cette zone. L'absence de DS doit être prouvée par la présence d'un RR NXT adéquat et signé dans la zone mère au niveau du point de délégation de la zone fille ; Dans ce cas, une chaîne de confiance ne pourra franchir le maillon manquant entre la zone mère et la zone fille, et ce, même si la zone fille est elle-même signée.
  - entre le point d'entrée sécurisé le plus proche en amont et la zone contenant le RRset, il existe au moins une délégation non sécurisée (absence de DS prouvée comme ci-dessus).

- Les informations erronées (*wrong*) :
  - la signature du RRSet est erronée ou expirée ;
  - il y a un problème au niveau du DS dans la zone parente : il est absent alors que sa présence est indiquée par l'enregistre-

ment NXT, ou il pointe vers une mauvaise clef ;
  – la chaîne de confiance censée relier le RRset à vérifier avec le point d'entrée sécurisé le plus proche contient un maillon corrompu ou qui ne passe pas la vérification.

– Les informations sur lesquelles on ne peut pas se prononcer (*non verifiable*) :
  – on ne dispose pas de point d'entrée sécurisé en amont de la zone considérée, et ceci parce que le resolver ne dispose pas de clef de confiance d'une zone en amont de cette zone ;
  – le chemin reliant la zone à la racine est totalement non sécurisé (y compris la racine).
  Pour ce type d'information, il conviendra donc de définir plusieurs niveaux de sécurité pour choisir l'attitude à adopter (est-on prêt à faire confiance quand même aux informations de cette zone ?).

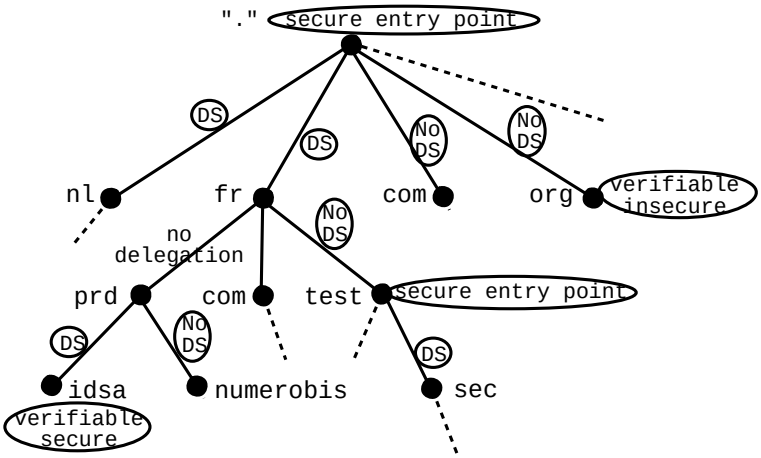

Figure 6 – *arbre DNS partiellement sécurisé*

On notera que cette classification est dépendante du resolver considéré. En effet les notions de point d'entrée sécurisé et de chaîne de confiance sont liées au resolver considéré. Ainsi un même RRset pourra être considéré comme sûr, non sûr ou erroné par trois résolveurs différents en fonction de leur point d'entrée sécurisé respectif.
Par exemple, sur la figure 6, on voit qu'il n'y a pas de DS pour la zone *test.fr*, ainsi un resolver configuré avec la clef de la racine comme clef de confiance considèrera la zone *test.fr* ainsi que toutes les zones en aval comme non sûres, et ce même si par exemple la zone *sec.test.fr* est signée. Au contraire un résolveur configuré avec la clef de *test.fr* considèrera la zone *sec.test.fr* comme sûre puisqu'un DS l'authentifie dans *test.fr*.

D'autre part, la distinction faite entre données non sûres et données erronées est importante car elle permet plus de finesse dans la politique de sécurité choisie par un résolveur. Même si une information est non sûre, on pourra choisir de lui faire confiance quand même en fonction de sa zone d'appartenance par exemple, au contraire, une information qui a été prouvée érronée sera en toute logique systématiquement rejetée.

5.3 Sécurisation progressive du DNS

On l'a introduit plus haut, le but du déploiement DNSsec est d'arriver à un modèle où la simple connaissance des clefs de la racine et leur configuration en tant que clefs de confiance dans les resolvers pourra permettre l'accès de manière sécurisée à toute information de l'arbre.
Néanmoins il est évident que le déploiement de DNSsec ne peut se faire que petit à petit au vu du nombre de zones et délégations à sécuriser. Ainsi dans un premier temps, certaines parties sécurisées de l'arbre côtoieront d'autres non sécurisées. On appelle îlot sécurisé un sous-arbre de l'arbre DNS dans lequel toutes les zones et délégations du sous-arbre sont sécurisées. Dans un premier temps l'arbre DNS sera donc constitué d'îlots sécurisés et de noeuds non sécurisés. Les informations contenues dans ces îlots pourront être considérées sûres par des résolveurs ayant la KSK de la zone sommet de l'îlot configurée comme clef de confiance. Puis petit à petit, ces îlots sécurisés pourront être reliés à la racine par le biais de délégations sécurisées.
Sur la figure 6, le sous-arbre commençant à *test.fr* est un îlot sécurisé ; une fois la délégation sécurisée reliant *test.fr* à *fr* mise en place, il pourra être relié à la racine par une suite de délégations sécurisées et rejoindra donc le modèle DNSsec global.

6 Sécurité des transactions

Nous avons abordé dans les parties précédentes les possibilités offertes par DNSsec pour sécuriser les informations DNS de manière individuelle (authentification de l'origine et intégrité des RRsets). Certains cas de figure nécessitent une sécurisation du message DNS complet, y compris l'en-tête.
Les transactions concernées sont d'une part les transferts de zone et les mises à jour dynamiques du DNS (Dynamic Update) et d'autre part le dernier lien de l'infrastructure DNSsec entre un resolver et son serveur récursif préconfiguré.
Les transferts de zone doivent fournir aux serveurs secondaires une réplique du fichier de zone situé sur le serveur primaire. Etant donné que ces serveurs secondaires font autorité sur les zones au même titre que leurs serveurs primaires, on se doit de garantir l'intégrité du fichier de zone lors de ce transfert en tant qu'entité et non en tant que somme de RRsets. Une fois le transfert de zone effectué, la vérification de l'intégrité des RRsets par le secondaire (contrôle des signatures), associée à l'utilisation des RRs NXTs peut partiellement permettre de vérifier l'exactitude et la complétude du fichier de zone. Néanmoins, les points de délégation ne sont pas signés et les glues ne sont ni signées ni précisées par un enregistrement NXT, ainsi la modification des premiers et la suppression des secondes pourraient passer inaperçues.
Les mises à jour dynamiques souffrent du même problème puisqu'ils contiennent des informations autres que des RRsets qui sont donc non signées.
Enfin, le cas d'une architecture ou les resolvers n'ont pas le support DNSsec mais sont configurés pour interroger un serveur récursif capable de faire les vérifications DNSsec, nous avons le même problème : même si les informations récupérées sur demande du resolver sont authentifiées par le serveur récursif après résolution DNSsec, celui-ci doit transmettre les RRsets non signés au sein d'un message DNS traditionnel (bit AD positionné dans l'en-tête, cf partie 7.1) au resolver. Si ce dernier lien n'est pas sécurisé d'une manière ou d'une autre, les information transmises par le cache ne sont pas dignes de confiance du point de vue du resolver (voir figure 7).

Ces trois scénarios nécessitent une authentification et une intégrité du message DNS en tant qu'objet indivisible. Ils ont également pour point commun que les entités communicantes se connaissent (serveurs primaires/secondaires ou resolver/serveur cache préconfiguré). On peut donc utiliser la cryptographie à clef privée (secret partagé) : c'est le cas de TSIG (Transaction Signature, RFC 2845)[10]. TSIG est un méta-RR, c'est-à-dire un RR qui n'est jamais stocké ni mis en cache dans le DNS, mais généré à la volée quand un message a besoin d'être authentifié. La signature par la clef secrète d'un hash du message DNS complet est intégré dans le RR TSIG, en compagnie de diverses autres informations ; ce RR est ensuite ajouté à la fin du message à transmettre. En réception, ce RR permettra de vérifier l'intégrité du message complet et de valider l'authenticité de la source du message (le seul à pouvoir signer utilisant la clef adéquate).
Il existe également d'autres méthodes pour sécuriser les trois scénarios évoqués ci-dessus, citons d'une part IPsec et d'autre part SIG(0) qui est une méthode spécifique au DNS utilisant les clefs publiques.

7 Nouveaux impératifs pour les acteurs et entités DNSsec

7.1 Indication du support de DNSsec

Nous l'avons évoqué notamment dans la partie 5.3, le déploiement de DNSsec ne peut se faire que de manière progressive, c'est-à-dire que pendant toute une durée transitoire, des entités (resolvers, serveurs) qui supportent DNSsec vont côtoyer des entités qui ne supportent pas DNSsec et communiquer avec elles. Un grand soin a été appliqué à garantir la compatibilité ascendante de DNSsec avec DNS : en dehors du respect du format des RRs et des messages, un certain nombre d'autres mesures ont été prises pour faciliter cette cohabitation :
– indiquer le support DNSsec ;
– normaliser le comportement des entités envers les données signées ou non signées ;
– gérer les requêtes et réponses nécessitant (ou pas) des RRs relatifs à la sécurité (KEY, SIG, NXT, DS).

On a pour ce faire créé trois flags (DO, AD, CD) :
– le flag DO (DNSsec OK) positionné à 1 indique le support DNSsec[11] ;
– le flag AD (Authenticated Data) positionné à 1 indique que les données contenues dans le message ont été authentifiées et sont dignes de confiance, il peut être utilisé par un serveur récursif qui fait les vérifications de signature pour le compte d'un resolveur. Néanmoins pour que ce flag ait une quelconque valeur aux yeux du resolver, il faut que ce dernier fasse

explicitement confiance au serveur et que le lien entre ce serveur et le resolver soit sécurisé par d'autres moyens (TSIG par exemple). Un resolver peut donc choisir de ne pas tenir compte de ce flag ;

– le flag CD (Checking Disabled) positionné à 1 par un resolver par exemple permet d'indiquer au serveur récursif que le resolver désire effectuer la vérification complète des informations lui-même et que le serveur récursif doit donc lui transmettre tous les RRs nécessaires.

7.2 Quelques scénarios pour l'infrastructure DNSsec

Sur la figure 7, nous proposons 3 scénarios possibles lors d'échanges de requêtes DNS et réponses entre des entités qui supportent ou ne supportent pas DNSsec (volontairement ou non). Dans ces 3 scénarios, un resolver envoie une requête DNS à son serveur récursif préconfiguré qui va se charger de récupérer les informations adéquates auprès du serveur autoritaire correspondant.
Dans les trois cas, le serveur autoritaire gère une zone qui est sécurisée.

– dans le premier cas, le resolver ne supporte pas DNSsec. Le serveur récursif qui lui supporte DNSsec va interroger le serveur autoritaire et récupérer tous les RRs lui permettant d'effectuer les vérifications de signature sur les RRSets désirés. Une fois que les enregistrements auront été authentifiés, ils pourront être transmis au resolver de manière sécurisée au resolver en utilisant TSIG (cf partie 6), et en positionnant le bit AD pour indiquer que les informations sont fiables.

– dans le deuxième cas, ni le resolver, ni le serveur récursif ne supportent DNSsec. Ils pourront toutefois continuer à communiquer avec le serveur autoritaire, qui, voyant que le bit DO n'est pas positionné, n'incluera aucun RR relatif à DNSsec. Du point de vue du serveur récursif et du resolver, aucune gêne de fonctionnement ne sera donc constatée.

– dans le troisième cas, le resolver supporte DNSsec et souhaite faire les vérifications de son côté. Grâce au bit CD, il indique au serveur récursif de lui transmettre tous les RRs nécessaires à la vérification des informations. Néanmoins, le serveur récursif fera lui aussi les vérifications et mettra en cache les résultats.

– un quatrième cas, qui n'est pas illustré sur la figure, doit néanmoins être considéré : c'est le cas ou le serveur récursif ne supporte pas DNSsec tandis que le resolver le supporte. Une règle d'usage du bit DO couvre ce cas : quand un serveur ne comprend pas le bit DO (implémentation antérieure à la création de ce flag), il va répondre avec un code d'erreur de type SERVFAIL ou NOTIMP, le resolver doit alors reformuler sa requête sans utilisation de ce drapeau et donc abandonner l'idée d'une vérification DNSsec.

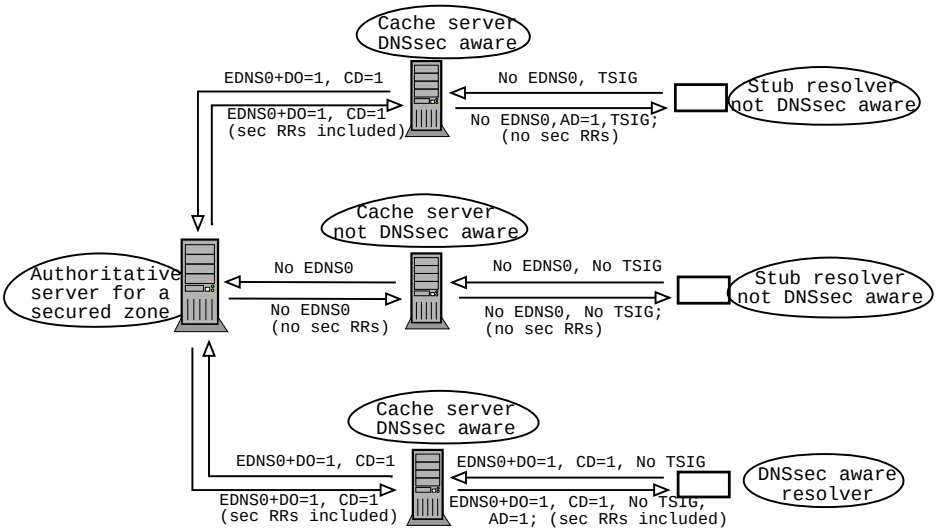

Figure 7 – *trois scénarios de déploiement DNSsec*

7.3 Le roulement des clefs

La sécurité apportée par les clefs utilisées dans DNSsec se base sur le concept fondamental de la cryptographie à clef publique : les parties privées des clefs ne doivent être connues que des signataires des zones, et à partir du moment où ces

clefs ne sont plus secrètes, tout le système s'écroule. Il est bien évident que le bon fonctionnement de DNSsec repose sur la précaution avec laquelle toutes les procédures relatives à la gestion des clefs et signatures sont effectuées (stockage des clefs, mise en place des délégations sécurisées, procédures de signature, etc.)
La compromission d'une clef peut arriver de deux manières, soit la clef privée se retrouve accidentellement dans les mains d'une personne non autorisée, soit la clef est cassée par des méthodes cryptanalytiques. Dans les deux cas, il est nécessaire de changer de clefs, c'est la procédure de roulement de clef (*key rollover*).
D'autre part, le fait de rouler les clefs régulièrement est également un moyen de limiter les attaques cryptographiques puisque les clefs seront utilisées moins longtemps.
Ces procédures de roulement de clefs affectent nécessairement le modèle de sécurité DNSsec et le but est donc de les réaliser sans briser les chaînes de confiance.

– Changer les ZSKs ne nécessite des opérations qu'au niveau de la zone concernée. Le maillon de confiance entre zone mère et zone fille (DS pointant sur la KSK de la fille) n'est pas mis en cause : la zone fille réalisera la procédure de manière transparente et sans aucune interaction nécessaire avec sa zone mère. Seules des précautions concernant les TTLs, les validités de signatures et les intervalles de resignature seront à respecter. Lors d'un roulement de clef, ces trois valeurs temporelles sont corrélées : par exemple, la nouvelle clef doit être diffusée antérieurement à son utilisation pour que les serveurs récursifs n'aient pas en cache que l'ancienne clef lors de l'utilisation effective de la nouvelle.

– Le roulement des KSKs est plus problématique, puisqu'elles sont des maillons dans les chaînes de confiance, et peuvent également être des clefs de confiance de certains resolvers.
  Dans le cas de roulements de KSKs prévus, il conviendra donc d'une part de prévenir à l'avance et aussi largement que possible afin que les résolveurs ayant l'ancienne clef configurée en tant que clef de confiance puissent réagir en temps et en heure. Il est à noter qu'une zone ne peut évidemment pas connaître tous les résolveurs ayant cette clef configurée comme clef de confiance.
  D'autre part, il est nécessaire de ne pas briser la chaîne de confiance durant l'opération de roulement. Pour ce faire, la procédure doit être choisie avec soin et nécéssite une bonne synchronisation des actions entre zone mère et zone fille. Différentes procédures sont décrites dans [12] et [13].
  Le cas du roulement de clef non prévu (*emergency rollover*) qui se pose lors de la compromission d'une clef ne peut par définition pas être automatisé et devra donc être effectué suivant les politiques de sécurité locales ; un article décrivant les *best practices* sur le sujet est toujours attendu.

8 Déploiement DNSsec - le projet IDsA

Les extensions DNSsec, spécifiées pour la première fois en 1999 sont toujours en évolution : durant ces quatres dernières années, la RFC 2535 a subi un certain nombre de mises à jour au travers de quelques RFCs, et est finalement en cours de réécriture par le biais d'une collection d'Internet-Drafts couvrant tous les aspects du protocole. Il semble néanmoins que les nouvelles spécifications aboutissent vers un consensus au sein de la communauté.
Cette relative instabilité protocolaire ainsi que le nombre très restreint d'implémentations supportant DNSsec (seuls les snapshots de BIND9 le supportent et sont vraiment opérationnels) ont retardé son déploiement. Si bien qu'aujourd'hui, il n'est utilisé qu'à titre expérimental au travers de projets de recherche ou d'infrastructures non utilisées en production.
Mais si du côté des serveurs on peut regretter l'existence d'une seule distribution opérationelle, c'est encore pire en ce qui concerne les resolvers : aucune version officielle d'un resolver supportant DNSsec n'existe.
D'autre part, aujourd'hui il existe encore un manque de visibilité et de retour d'expérience sur les aspects opérationnels de la maintenance des zones signées et l'automatisation des procédures, notamment en ce qui concerne la gestion des clefs (des procédures telles que le roulement des clefs).

Un certain nombre de projets sont en cours pour valider les protocoles et mettre en avant les forces et difficultés d'implémentation de DNSsec, on pourra trouver les références de la plupart d'entre eux sur [14].
Citons notamment les projets de sécurisation des domaines *fr* et *nl*, et également le projet RS.NET qui met en place une arborescence DNS parallèle sécurisée depuis la racine.

Le projet RNRT IDsA[4] a quant à lui pour but de mener diverses expérimentations basées sur l'architecture DNSsec à l'aide de la plate-forme de test mise en place sur le domaine *idsa.prd.fr.* :

[4] Projet RNRT IDsA, (Infrastructure DNSsec et Applications). http ://www.idsa.prd.fr et ftp ://ftp.irisa.fr/local/idsa

– développement d'une plate-forme de tests sur laquelle est déployée une infrastructure DNSsec ;
– création d'outils de validation des chaînes de confiance et enrichissement des fonctionnalités des resolvers. Un resolver enrichi supportant DNSsec est déjà disponible sur ftp//ftp.idsa.prd.fr/local/idsa/code ;
– étude et production de documents détaillant les aspects opérationnels de signature et de maintenance des zones sécurisées ;
– développement d'outils facilitant la maintenance des zones sécurisées, la mise en place des délégations sécurisées et le roulement des clefs ;
– étude des interactions entre DNSsec et IPsec ;
– utilisation de DNSsec dans le cadre de l'authentification des noeuds mobiles dans un modèle de réseau utilisant mobile IPv6.

Références

[1] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

[2] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

[3] Albitz, P and Liu, C. DNS and BIND, 4th edition, éditions O'Reilly, April 2001.

[4] Atkins, D. and R. Austein, "Threat Analysis Of The Domain Name System", draft-ietf-dnsext-dns-threats-03 (work in progress), June 2003.

[5] Eastlake, D., "Domain Name System Security Extensions", RFC 2535, March 1999.

[6] Gudmundsson, O., "Delegation Signer Resource Record", draft-ietf-dnsext-delegation-signer-15 (RFC editor), June 2003.

[7] Arends, R., Austein, R., Larson, M., Massey, D. and S. Rose, "Protocol Modifications for the DNS Security Extensions", draft-ietf-dnsext-dnssec-protocol-00 (work in progress), February 2003.

[8] Arends, R., Austein, R., Larson, M., Massey, D. and S. Rose, "Resource Records for DNS Security Extensions", draft-ietf-dnsext-dnssec-records-04 (work in progress), February 2003.

[9] Lewis, E., "DNS Security Extension Clarification on Zone Status", RFC 3090, March 2001.

[10] Vixie, P., Gudmundsson, O., Eastlake, D. and B. Wellington, "Secret Key Transaction Authentication for DNS (TSIG)", RFC 2845, May 2000.

[11] Conrad, D., "Indicating Resolver Support of DNSSEC", RFC 3225, December 2001.

[12] Kolkman, O., Gieben, M, "DNSSEC key operations" draft-kolkman-dnssec-operational-practices-00 (work in progress), August 2003

[13] Leonard, B., "DNSsec : Theoretical and Practical Aspects", May 2003.

[14] DNSsec, securing the domain name system. Collection de liens vers des documents, outils, projets concernant DNSsec disponibles sur internet. http ://www.dnssec.net