

Migration vers DNSSEC avec OpenDNSSEC

Claude GROSS

GIP RENATER
23-25, rue Daviel
75013 Paris

Résumé

Le DNS est un service essentiel sur lequel repose la quasi-totalité des autres services sur Internet. Les menaces sur son bon fonctionnement s'étant développées ces dernières années, sa sécurisation est devenu un enjeu très important.

DNSSEC fait partie des briques qui ont été développées pour consolider la sécurité du DNS et qui semble avoir une chance de se déployer. Mais pour que cela fonctionne complètement, il faut que tout le monde s'y mette. Depuis 2010, la racine a été signée, beaucoup de TLD ont été signés ou ont annoncé leur intention de le faire. Il ne reste donc plus qu'à le faire pour tous les autres domaines...

De plus, la migration vers DNSSEC n'est pas anodine. À quoi servira de signer les réponses aux requêtes des résolveurs, si les données d'origine ou les clés de signatures ont été compromises ? DNSSEC, pour que son utilisation ait un sens, impose donc la mise en place d'une architecture sécurisée adaptée et d'une nouvelle organisation pour la gestion du DNS.

Mots-clefs

Sécurité, DNS, DNSSEC, Chiffrement.

1 Introduction

Comme pour la plupart des protocoles basés sur TCP/IP, le DNS n'a pas été conçu avec une grande considération vis à vis de la sécurité. Or, s'il ne fait pas beaucoup parler de lui quand tout va bien, il est pourtant la base du bon fonctionnement de la plupart des autres services sur Internet.

Les risques concernant le DNS sont principalement de 2 types :

1. Les dénis de services : ces attaques visent à rendre indisponible le service DNS, rendant ainsi également indisponibles pratiquement la totalité des services présents sur Internet ;
2. Les réponses modifiées aux requêtes DNS : ces attaques ont pour but de rediriger un client vers un serveur différent de celui attendu.

DNSSEC essaie de répondre au 2^{ème} type d'attaques en apportant au service DNS l'authentification et l'intégrité des données fournies par les réponses aux requêtes DNS. DNSSEC ne répond donc pas à tous les problèmes de sécurité du DNS. Il est l'une des briques utilisables pour consolider la sécurité du DNS, s'il est correctement utilisé et implémenté.

Cette article propose, après une rapide rappel de ce qu'est DNSSEC, de présenter les différents aspects de la migration vers DNSSEC : architecture, gestion des clés, ... Il présentera également l'outil OpenDNSSEC, qui permet l'automatisation d'une grande partie du processus de gestion des clés de chiffrement.

2 DNSSEC, qu'est ce que c'est ?

Outre les RFCs[1][2][3][4][5][6], de nombreux sites, articles et papiers sont consacrés à DNSSEC [7][8][9][10][11]. Ce chapitre n'est donc qu'un rapide rappel de ses fondamentaux.

DNSSEC est un ensemble d'extensions au protocole DNS dans le but de sécuriser ce service, en permettant l'authentification et l'intégrité des données contenues dans les réponses aux requêtes DNS. Pour cela, ces données sont signées en utilisant la cryptographie à clés asymétriques, ces signatures étant incluses dans les fichiers de zones. DNSSEC ne protège donc pas le canal de communication entre les clients et les serveurs DNS, mais les données qui sont échangées dans ces communications. La plupart des logiciels serveurs DNS supportent aujourd'hui DNSSEC. Les exemples de cet article s'appuieront sur le logiciel BIND¹.

2.1 Les clés dans DNSSEC

Le chiffrement à clés asymétriques² utilise une paire de clés, l'une publique et l'autre privée. La clé privée est utilisée pour signer les données alors que la clé publique sert à vérifier la validation de ces signatures.

Avec DNSSEC, chaque zone DNS possède une paire de clés de signature. Pour chaque RRSet, une empreinte (hash) est calculée et est signée avec la clé privée correspondante à la zone. La clé publique est incluse dans le fichier de zone dans un enregistrement DNSKEY.

```
educ-rech.fr.      3600   IN      DNSKEY  257 3 8 AwEAAcJm9kAnERB2...qDrsNOfdxLA
```

En fait, comme nous le verrons plus loin, on utilise souvent 2 paires de clés par zone avec DNSSEC.

2.2 Les signatures dans DNSSEC

Avec DNSSEC, chaque RRSet est signé avec la clé privée de la zone. Ces signatures sont contenues dans des enregistrements RRSIG qui contiennent également une date de début et de fin de validité, ainsi que l'identifiant de la clé qui a été utilisée pour cette signature.

```
www.educ-rech.fr. 60      IN      A        193.49.159.110
www.educ-rech.fr. 60      IN      RRSIG   A 8 3 5400 20131027033034
                20131019165738 55359 educ-rech.fr. YKnIuqIU...prJqcKlk3HRDLf+Iwz+I=
```

Ces signatures doivent donc être renouvelées :

- à chaque modification de la zone ;
- avant leur date de fin de validité.

Bien que ce soit les RRsets qui soient signés, par commodité on parle de zone signée.

2.3 Etablissement de la confiance

C'est bien de signer les données d'une zone DNS distribuées par des serveurs DNS faisant autorité sur celles-ci. Mais comment être sûr que la clé publique qui va servir pour la validation est bien celle associée à la clé privée avec laquelle la zone a été signée ? Pour établir cette chaîne de confiance, on utilise la hiérarchie de l'architecture du DNS et son système de délégation. Chaque zone signée fournit une empreinte de sa clé publique à sa zone parente. Cette empreinte est signée avec la clé privée de la zone parente et mise dans un enregistrement DS dans le point de délégation de la zone fille.

¹ <http://www.isc.org/downloads/bind/>

² 1999.jres.org/tutoriaux/tutorial4-chiffrement.pdf

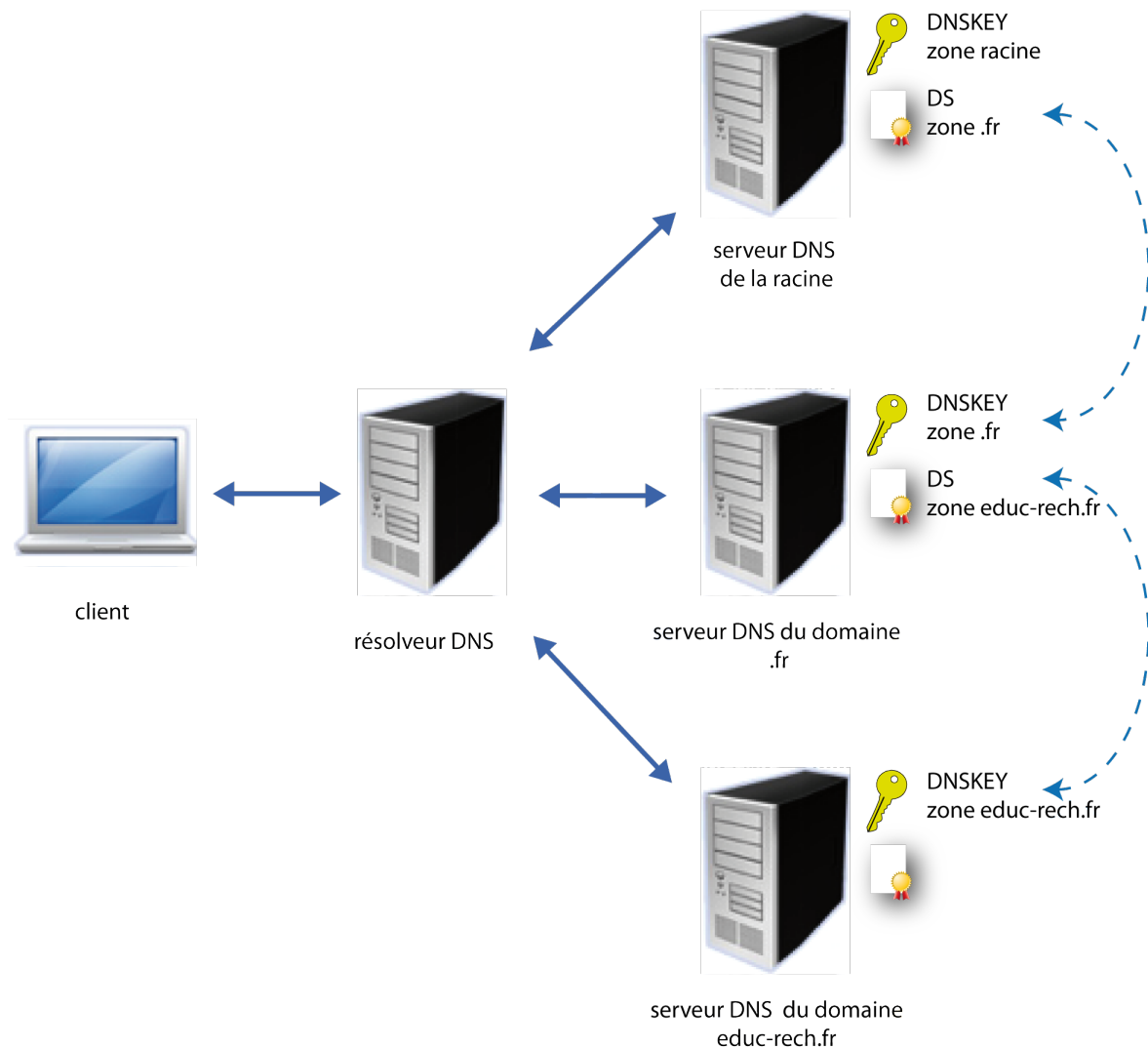


Figure 1 - Validation DNSSEC

Lorsqu'un résolveur DNS doit valider une réponse d'un serveur DNS pour une zone donnée, il récupère la clé publique de ce serveur via un enregistrement DNSKEY. Pour vérifier la validité de cette clé, le résolveur doit récupérer également dans la zone parente un enregistrement DS signé, contenant une empreinte de la clé de la zone fille. Il faut donc que la zone parente elle-même utilise DNSSEC pour pouvoir signer cette empreinte. Ceci doit être répété jusqu'à la racine pour former entièrement la chaîne de confiance. Le fondement de la confiance ici est la confiance que font les résolveurs à la clé de la zone racine (une alternative est *DLV* [12] mais elle ne sera pas traitée ici).

2.4 Utilisation de 2 paires de clés

Pour une zone donnée, pour avoir une chaîne de confiance complète, il faut donc que toutes les zones entre elle et la racine dans l'arbre DNS soient signées et que l'empreinte de la clé de chacune de ces zones soit transmise à sa zone parente. Cette dernière condition, du fait de la forte recommandation de changer périodiquement les clés de signature, apporte une contrainte assez importante. Pour cette raison, bien que DNSSEC ne l'oblige pas, on utilise souvent 2 paires de clés par zone, ceci pour limiter les interactions entre zone fille et zone parente.

L'une des paires de clés, la ZSK (*Zone Signing Key*), va être utilisée pour signer les enregistrements de la zone. Cette clé, utilisée uniquement en interne, pourra être renouvelée assez fréquemment sans problème (un mois par exemple). De ce fait, elle pourra aussi avoir une longueur relativement courte, ce qui rendra plus rapide le processus de validation des signatures.

La seconde, la KSK (*Key Signing Key*), va être utilisée uniquement pour signer la ZSK. C'est l'empreinte de la clé publique de la KSK qui sera transmise à la zone parente pour former l'enregistrement DS. La KSK, dont le

renouvellement devra être signalé à la zone parente, pourra être renouvelée beaucoup moins souvent (un an par exemple). Elle devra donc avoir une longueur plus grande que la ZSK, mais comme elle ne signe que la ZSK, cela aura moins d'importance en terme de performance.

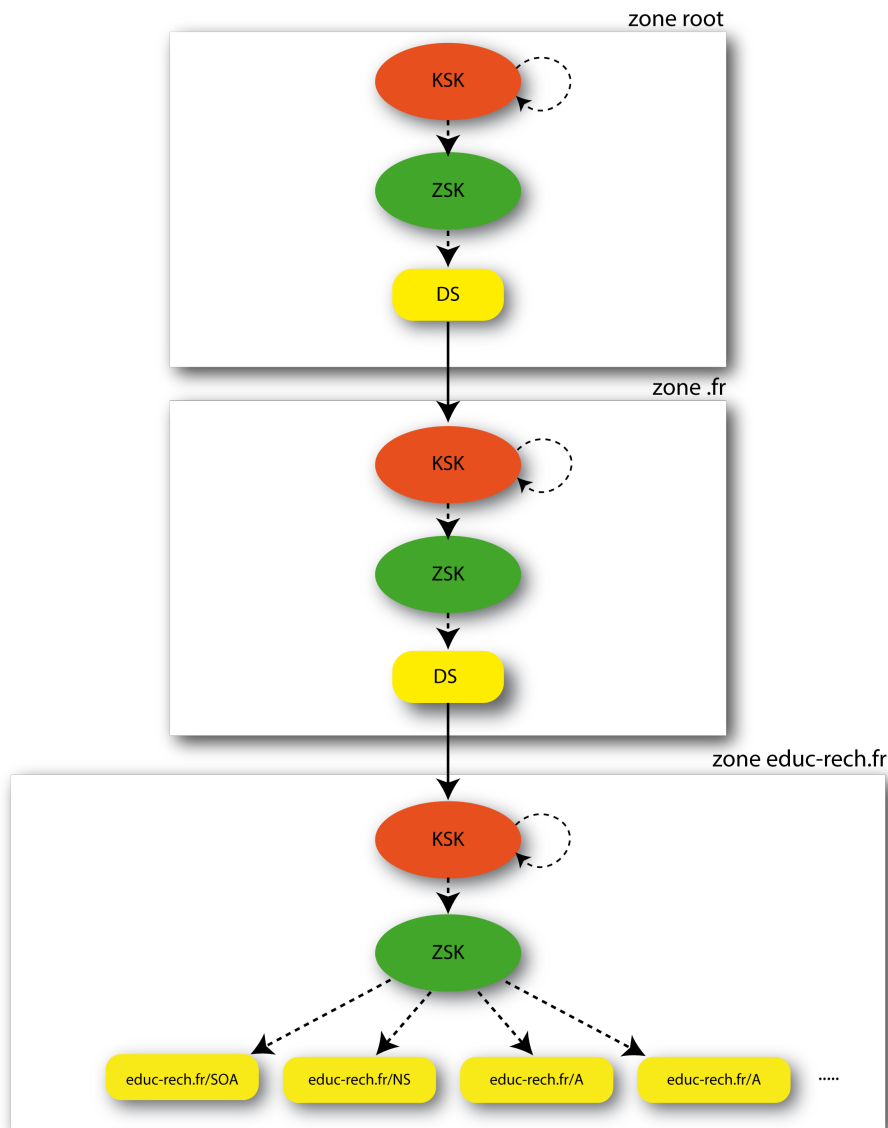


Figure 2 - Exemple de chaîne de validation avec 2 paires de clés.

La figure ci-dessus illustre la chaîne de confiance avec l'utilisation de 2 paires de clés. Les flèches en pointillé indiquent la signature de l'élément qu'elle vise. Le processus de validation est le suivant :

- la zone *educ-rech.fr* utilise sa clé ZSK pour signer les enregistrements de sa zone. Cette clé est signée avec la clé KSK de la zone *educ-rech.fr* ;
- l'empreinte de la clé KSK de la zone *educ-rech.fr* est fournie à la zone parente *.fr* qui la signe avec sa clé ZSK (DS). Cette clé est signée avec la clé KSK de la zone *.fr* ;
- l'empreinte de la clé KSK de la zone *.fr* est fournie à la zone parente *root* qui la signe avec sa clé ZSK (DS). Cette clé est signée avec la clé KSK de la zone *root*.

Pour valider la signature d'un enregistrement DNS, il faut donc que le valideur fasse confiance à la clé KSK de la zone *root* (il existe une alternative [12]) et vérifie toutes les signatures en descendant dans l'arbre DNS jusqu'à l'enregistrement à valider.

2.5 La validation DNSSEC

La question ici est de savoir qui fait la validation DNSSEC : les applications (comme les navigateurs web), les clients DNS sur les postes de travail, les résolveurs DNS ? C'est ce dernier qui apparaît comme étant le plus adapté pour remplir ce rôle.

Normalement, lorsqu'un client DNS fait une requête DNS, il s'adresse à son résolveur qui est un serveur DNS cache récursif. Celui-ci, par défaut, se contente de répondre à la requête DNS sans effectuer de validation DNSSEC. Pour cela, il faut donc utiliser un logiciel qui sache le faire, le configurer et mettre en place la clé de confiance du point de départ de la validation : la clé publique de la racine.

Par ailleurs, il existe des extensions pour les navigateurs, comme par exemple le plugin `DNSSECValidator`³ de Firefox, qui permettent de vérifier si un site utilise DNSSEC ou non.

2.6 Problèmes relatifs à DNSSEC

Comme on peut le voir, le passage à DNSSEC, pour qu'il ait un sens, a des conséquences organisationnelles et techniques sur l'ensemble de la gestion du DNS.

D'abord, la sécurisation des données, qui devrait également être prise en compte dans un service DNS classique, a ici encore plus d'importance.

Ce qui est entièrement nouveau, c'est la gestion nécessaire du renouvellement de la signature des enregistrements, qui doit être périodiquement refaite, et de celui des clés, qui peuvent être nombreuses (2 paires de clés par zone DNS gérée). Les clés privées doivent bien sûr être protégées correctement. Mais le renouvellement périodique des signatures et des clés entraînent aussi la nécessité d'une gestion rigoureuse. En effet, en cas d'erreurs, la conséquence peut être la disparition complète d'une zone de l'espace du DNS.

Par ailleurs, DNSSEC ne résout pas tous les problèmes de sécurité liés au DNS. En particulier, il ne protège pas de la compromission des données avant la signature. Il est donc nécessaire de repenser l'architecture du service DNS pour sécuriser les données, de mettre en place un niveau acceptable de sécurisation des clés privées, d'établir des procédures rigoureuses pour la gestion des signatures et des clés et de mettre en place une supervision du système.

DNSSEC ne protège pas non plus le canal entre les clients DNS et les résolveurs DNSs valideurs, ni celui entre serveurs DNS primaires et secondaires. Ce dernier canal pourra être sécurisé en utilisant *TSIG*[15].

3 Quelle architecture pour DNSSEC ?

Pour que la signature des données apporte un réel gain en terme de sécurité, il faut évidemment que les clés privées, mais aussi les données signées elles-mêmes soient protégées correctement.

3.1 Protection des données

Comme nous l'avons vu, une fois une zone signée, et en s'appuyant sur la chaîne de confiance DNSSEC, celle-ci peut être diffusée sans craindre quant à son intégrité et son authenticité. Ce sont donc les données en amont, qui seront contenues dans les zones DNS signées, qu'il faut protéger. Pour cela, l'ensemble des serveurs, logiciels et données entrant dans le processus de la génération des zones signées doit être cantonné sur un réseau protégé, sur des serveurs sécurisés, et non accessible de l'extérieur.

Une fois la zone signée, elle pourra être transférée vers le ou les serveurs DNS connus de l'extérieur comme faisant autorité pour cette zone.

Dans l'architecture proposée, on peut distinguer, pour un domaine DNS donné :

- les serveurs DNS faisant autorité, non récursifs ;
- les serveurs DNS cache récursifs (résolveurs) qui vont répondre aux requêtes des clients et valider les réponses ;
- les clients DNS ;

³ <https://addons.mozilla.org/fr/firefox/addon/dnssec-validator/>

- le système de gestion des clés et des signatures.

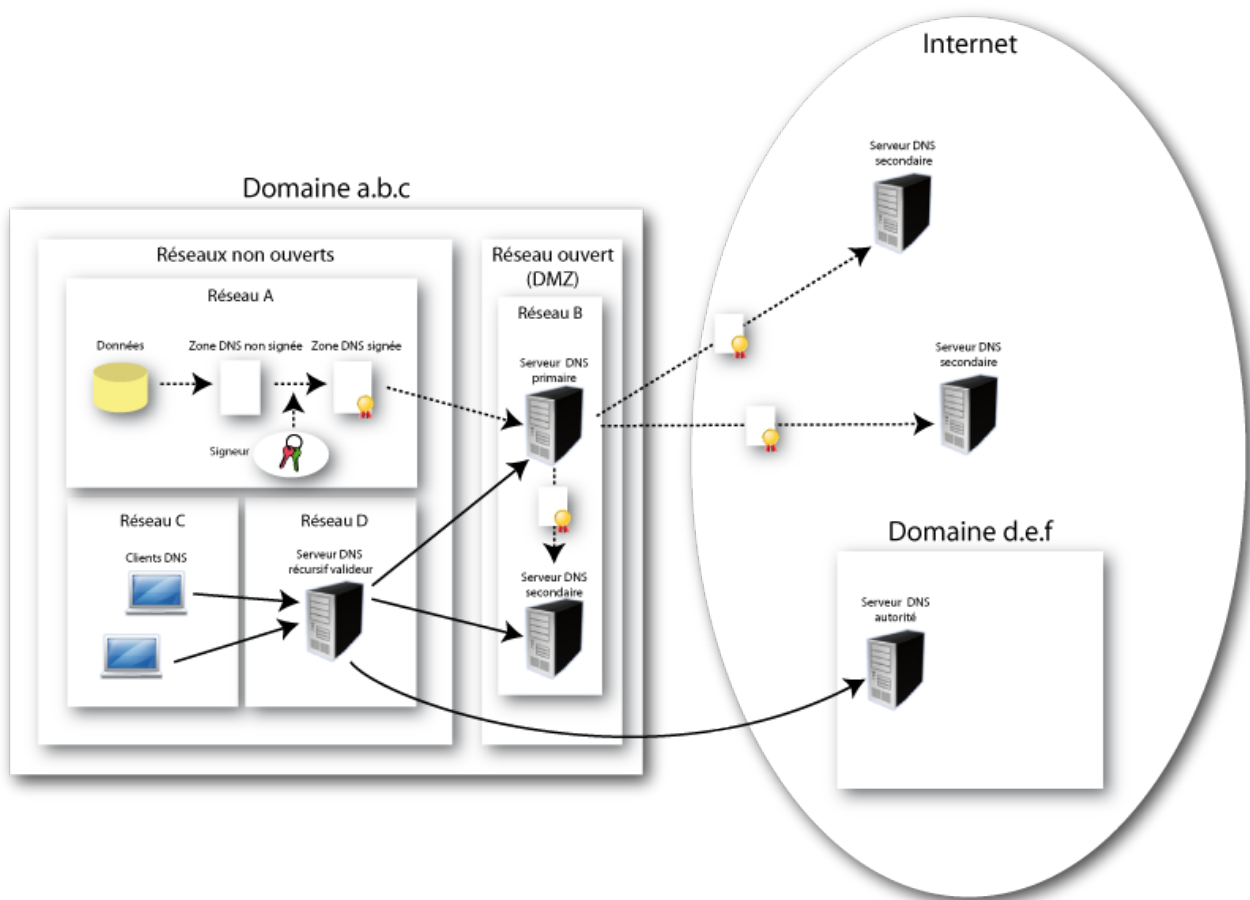


Figure 3 - Architecture DNS sécurisée

Dans le schéma général ci-dessus :

- le réseau A héberge les ressources nécessaires au système de gestion des données internes du DNS, au système de gestion des clés ainsi qu'à la signature des fichiers de zones. Ce réseau n'est ouvert ni vers les autres réseaux du domaine (sinon de manière sécurisée et seulement en fonction des besoins), ni bien sûr vers l'extérieur du domaine. A chaque mise à jours de la zone signée, celle-ci doit être transmise de façon sécurisée au serveur DNS primaire de cette zone sur le réseau B ;
- le réseau B correspond à un réseau ouvert, hébergeant les différents services du domaine accessibles de l'Internet. Le serveur DNS primaire récupère la zone signée à chaque mise à jours. Cette zone peut alors être transmise aux serveurs DNS secondaires concernés, dans le domaine lui-même ou à l'extérieur du domaine ;
- le réseau C héberge typiquement les clients DNS du domaine, par exemple les postes de travail des utilisateurs. Leurs requêtes DNS sont adressées exclusivement à un résolveur interne hébergé sur le réseau D ;
- le réseau D héberge les serveurs internes du domaine, également clients DNS, et en particulier le résolveur DNS interne. Ce résolveur, un serveur DNS cache récursif, se chargera pour les clients du domaine, de répondre à leurs requêtes. C'est lui aussi qui sera chargé également d'effectuer la validation des réponses DNS obtenues à partir des serveurs DNS faisant autorité pour des zones signées via DNSSEC.

Le système de gestion des données DNS situé dans le réseau A peut être un serveur DNS primaire, caché du reste du monde, et le transfert des zones signées entre lui et le serveur DNS du réseau B pourra utiliser les mécanismes classiques de transfert XFR entre serveur maître et serveur esclave.

3.2 Protection des clés privées

Les clés privées utilisées pour signer les données doivent bien sûr être protégées. Cette sécurisation peut être de différents niveaux :

- clés sur un serveur sécurisé, sur un réseau sécurisé, dans un fichier accessible uniquement par l'administrateur DNS. C'est le niveau de protection minimal ;
- clés dans un HSM⁴ protégé par un PIN code. Il en existe plusieurs catégories : Token USB, carte à puce, carte PCI... Leur niveau de sécurité, leur capacité de stockage et leur prix sont très variables.

Le choix sera dicté par différents paramètres : niveau de sécurité nécessaire, ressources disponibles, nombre de clés à gérer, budget...

4 Le renouvellement des clés et des signatures

Mettre en place un service DNS avec DNSSEC apporte clairement un gain en terme de sécurité. Pour autant, il apporte aussi une fragilité du fait des risques d'erreurs dans la gestion des clés et des signatures.

La maîtrise de ce risque passe par des procédures rigoureuses (de préférence automatisées) permettant d'éviter des erreurs dans les processus qu'il faut mettre en œuvre. Ces procédures doivent permettre de gérer les tâches suivantes :

- génération des clés KSK et ZSK et leur renouvellement ;
- génération de la signature des zones et leur renouvellement ;
- propagation de l'empreinte des clés KSK à la zone parente.

Parallèlement à ces procédures de gestion courante, il est hautement recommandé de prévoir une procédure de renouvellement d'urgence en cas, par exemple, de compromission des clés.

La gestion des clés et de leur renouvellement passe par la définition d'une politique définissant différents paramètres :

- longueur des clés ZSK et KSK ;
- algorithmes à utiliser ;
- durée de vie des signatures ;
- durée de vie des clés ZSK et des clés KSK.

Cette gestion des signatures et des clés est d'autant plus complexe du fait de la conservation des données du DNS dans les caches des résolveurs et sur les serveurs DNS secondaires. En particulier, comme nous le verrons plus loin, la durée de vie des clés et des signatures devra tenir compte du TTL utilisé pour les enregistrements du DNS.

De façon générale, une réponse DNS signée doit pouvoir être validée à tout instant et quelque soit la manière dont elle a été obtenue (directement d'un DNS primaire, d'un DNS secondaire ou à partir du cache d'un résolveur). Pour cela, il faut que le valideur obtienne le RRSets lui-même, sa signature et la clé publique de la zone DNS correspondante. Il faut également que :

- le moment de la validation se situe entre la date de début de validité de la signature du RRSets obtenue et sa date d'expiration (cf. paragraphe suivant) ;
- que la signature puisse bien être vérifiée avec la clé publique de la zone.

Ce même processus doit bien sûr être répété tout le long de la chaîne de confiance en utilisant les enregistrements DS successifs, jusqu'à la racine du DNS, pour une validation complète de la réponse DNS.

La première conséquence de tout cela est l'importance des paramètres temporels dans la configuration de DNSSEC et la nécessité d'avoir des serveurs à l'heure.

Le RFC 6781 [13] donne, entre autres, un ensemble de recommandations concernant les paramètres temporels entrants en jeu dans DNSSEC.

⁴ Hardware Security Module (http://fr.wikipedia.org/wiki/Hardware_Security_Module)

4.1 Renouvellement des signatures

Comme nous l'avons vu, les signatures ont un début et une fin de validité. Cette durée de validité doit être au moins égale au *TTL* DNS de l'enregistrement qu'elle signe. Dans le cas contraire, une signature expirée pourrait être diffusée via le cache d'un serveur DNS. Pour autant, cela ne nous dit pas quelle durée choisir. Il faut éviter les durées trop longues, qui posent le problème de la vulnérabilité aux attaques, et une durée trop courte qui risque de poser des problèmes opérationnels, par exemple en cas d'impossibilité de rafraichissement d'une zone sur un DNS secondaire. La recommandation actuelle est de prendre des durées de validité assez longue, par exemple un mois.

Il est à noter qu'on peut très bien avoir 2 signatures différentes pour un même enregistrement, par exemple l'une sur un primaire et l'autre dans un cache, et que les 2 soient valides. Il suffit que le moment de la validation soit compris entre la date de début et de fin des 2 signatures, que les valideurs utilisent la clé publique correspondante à la clé privée qui a signé ces enregistrement et, bien sûr, que l'ensemble de la validation puisse être satisfaite à travers la chaîne de confiance.

En pratique, dans l'opération de signature, en plus de la durée de vie des signatures, d'autres paramètres temporels sont à considérer. Ces paramètres sont illustrés sur la figure ci-dessous.

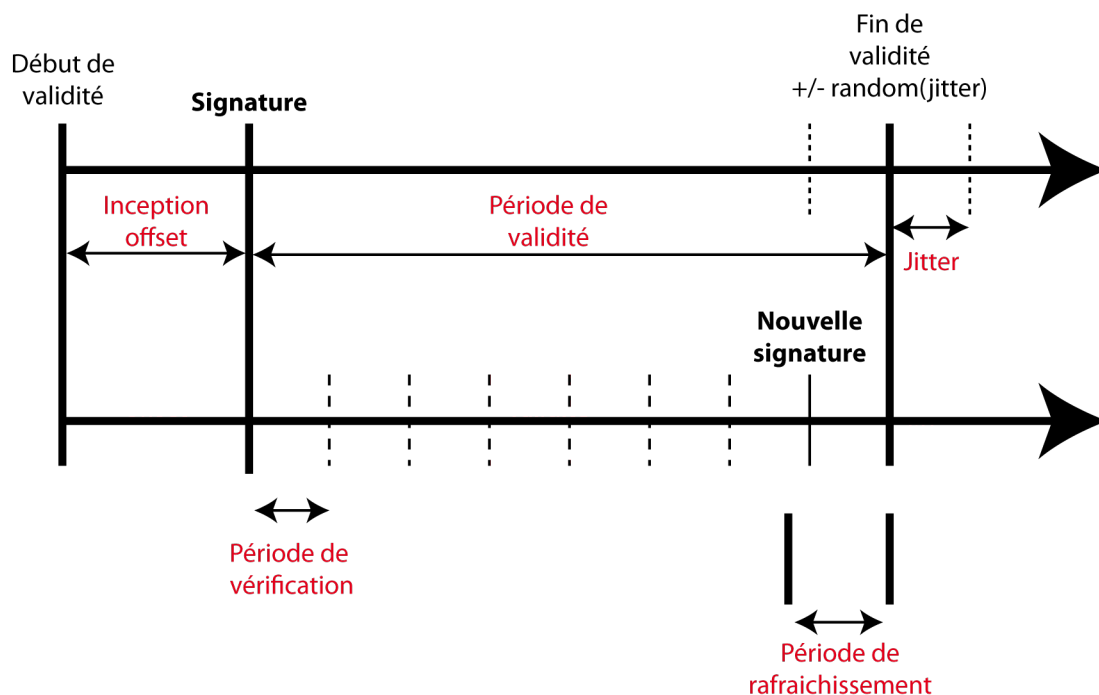


Figure 4 - Gestion des signatures

- **Inception offset** : la date de début de validité est antérieure à la date effective de la signature pour pallier les problèmes de désynchronisation des horloges ;
- **Période de vérification** : périodiquement, le signeur vérifie s'il y a des signatures à refaire. Si ce n'est pas le cas, on continue d'utiliser la même signature ;
- **Période de rafraichissement** : si la différence entre la date d'expiration et le moment de la vérification d'une signature est inférieure à la période de rafraichissement, le RRSets est re-signé ;
- **Jitter** : un nombre aléatoire (entre $-jitter$ et $+jitter$) est ajouté ou retranché à la date normale d'expiration pour éviter que toutes les signatures expirent en même temps.

En plus du paramètre *TTL* des *RRSet*, il faut également tenir compte d'autres paramètres temporels propres au DNS, comme le paramètre *refresh* (durée avant vérification de la zone par un serveur secondaire) dont la valeur doit être plus petite que la durée de vie des signatures pour éviter que des signatures expirées continuent d'exister dans des caches. Dans les faits, les recommandations actuelles vont bien au delà puisque il est préconisé de prendre des durées de validité assez grandes pour les signatures, de l'ordre d'un mois, à comparer aux valeurs usuelles des paramètres *refresh* (quelques heures) et *TTL* (quelques jours).

4.2 Roulement des clés

Le roulement des clés est l'opération qui va consister à remplacer une paire de clés par une autre.

4.2.1 Principe général

Il est conseillé de prévoir 2 types de roulement. Le 1^{er} correspond à la régénération normale des clés en tenant compte des différents paramètres de la politique de roulement : TTL de la zone, durée de vie des clés... Le 2^{ème} est le roulement d'urgence à prévoir pour les cas de compromission des clés ou d'incidents. Ce paragraphe traite du roulement normal, le roulement d'urgence ne pouvant, par nature, se planifier.

Il y a 2 méthodes principales pour le roulement des clés, la prépublication et la double signature.

Dans le cas de la prépublication, le principe général est de créer la nouvelle clé sans l'utiliser, d'attendre qu'elle soit publiée dans tous les caches puis de commencer à l'utiliser pour signer. L'ancienne clé, qui n'est alors plus utilisée, continue d'être publiée un certain temps, jusqu'à ce qu'elle ait disparue de tous les caches DNS.

Dans le cas de la double signature, on commence à signer avec la nouvelle clé dès sa création, tout en continuant à signer avec l'ancienne clé un certain temps.

La double signature pose principalement le problème de la taille du fichier de zone pendant le temps de la double signature. Cette méthode n'est donc pas très adaptée au roulement des clés ZSK pour lequel on préférera la prépublication. Par contre, pour le roulement des clés KSK (qui ne signent que les clés ZSK), la double signature est souvent utilisée.

Du fait du roulement des clés, celles-ci passent dans leur cycle de vie par différents états [14] qui sont illustrés sur la figure suivante.

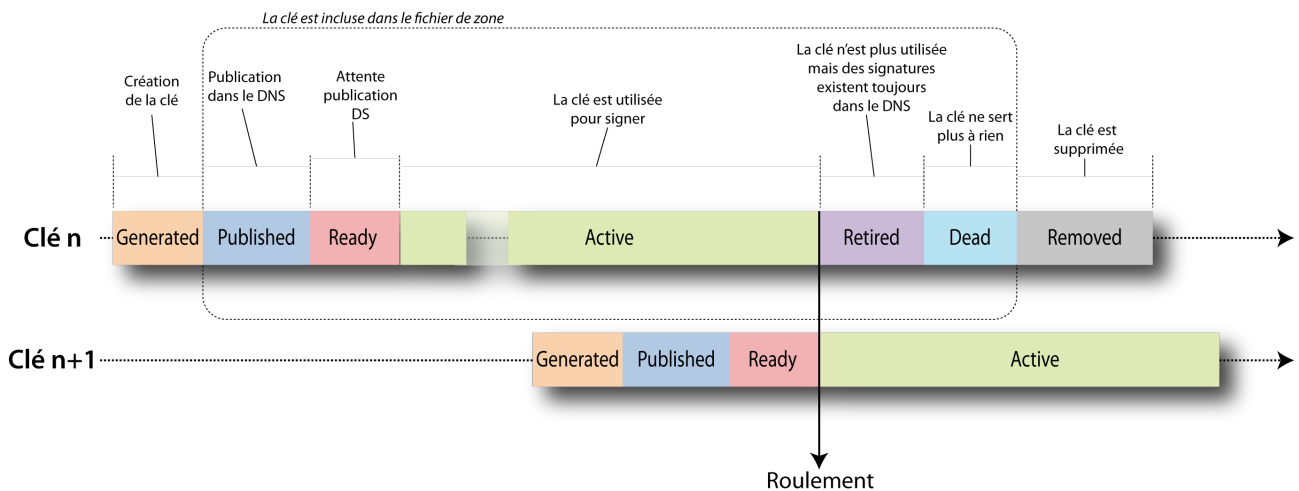


Figure 5 - Etats des clés et roulement par pré-publication

- **Generated** : la clé a été créée ;
- **Published** : la clé est publiée dans le DNS, sous la forme d'un enregistrement DNSKEY mais elle n'a pas encore forcément atteint tous les résolveurs. Cette durée est au moins égale au refresh de la zone + TTL de l'enregistrement DNSKEY ;
- **Ready** : la clé a été diffusée dans tous les caches et est prête à être utilisée ;
- **Active** : la clé est utilisée pour signer ;
- **Retired** : la clé n'est plus utilisée pour signer mais est encore publiée car des signatures faites avec cette clé sont peut-être encore dans des caches ;
- **Dead** : la clé est encore dans le DNS mais ne sert plus à rien ;
- **Removed** : la clé est complètement supprimée ;

Un état **Revoked** est également prévu mais n'est pas implémenté pour l'instant.

Comme on peut le voir, l'automatisation de toutes les procédures est absolument obligatoire pour gérer l'ensemble des tâches à accomplir pour maintenir un service DNS avec DNSSEC.

4.2.2 Quand changer les clés ?

Contrairement aux signatures, les clés n'ont pas de limite de validité dans DNSSEC. Procéder à leur changement est donc un choix du gestionnaire du service DNS en question. Le choix de la périodicité du roulement des clés est donc le résultat d'un compromis entre le risque d'exposition trop longue des clés et les risques induits par les opérations de roulements. Les clés ZSK étant utilisées uniquement en interne, leur changement ne nécessite aucune interaction avec la zone parente. Leur changement peut donc se faire plus souvent que pour les clés KSK. Les recommandations du RFC 6781 pour le roulement des clés KSK est d'un an. Pour celui des clés ZSK, un mois est une durée raisonnable.

En dehors des opérations de roulements périodiques et automatisées (si celles-ci ont été mises en place), le changement de clés d'urgence peut aussi être nécessaire pour d'autres raisons (compromission des clés, ...). Il est donc grandement préconisé de mettre en place, dans tous les cas, les procédures de changement de clés et de les tester.

5 Un outil : OpenDNSSEC

La mise en place de DNSSEC nécessite l'adoption d'outils qui vont permettre de faciliter et d'automatiser les différentes tâches à effectuer : génération et renouvellement des paires de clés, signature des données, ... Il en existe différents types, de différents niveaux, permettant de gérer les différentes tâches induites par DNSSEC dans un service DNS. Du simple utilitaire permettant de signer une zone à l'appliance complet permettant de tout faire, le choix se fera en fonction des besoins, des ressources et des finances.

OpenDNSSEC⁵ est un logiciel libre permettant d'automatiser entièrement les différentes opérations de génération et de renouvellement des signatures et des clés rendues nécessaires par DNSSEC. La seule chose qu'il ne fait pas est la propagation de l'empreinte de la clé publique d'une zone fille vers sa zone mère.

Nous nous baserons ici sur la version 1.4 de OpenDNSSEC.

5.1 Que fait OpenDNSSEC ?

Une fois installé et configuré, OpenDNSSEC s'occupera de toutes les tâches suivantes :

- récupération des données à partir de fichiers de zones ou à travers un transfert XFR depuis un serveur DNS ;
- roulement des clés : génération, publication et suppression des clés dans le HSM en respectant la politique définie ;
- génération et régénération des signatures quand nécessaire en respectant la politique définie ;
- transfert des zones signées soit dans des fichiers de zones, soit par transfert XFR vers un serveur DNS.

Nous n'entrerons pas ici dans tous les détails d'installation et de configuration de OpenDNSSEC, la documentation sur le site de référence étant riche et très bien faite. Il est bien sur fortement conseillé de se faire la main sur une plateforme de tests avant d'envisager de passer à la production.

5.2 Installation

L'installation de OpenDNSSEC se fait soit par package, quand ils existent, soit à partir des sources. La plupart des systèmes Unix sont supportés.

OpenDNSSEC impose l'utilisation d'un *HSM* pour la génération des paires de clés et le stockage des clés privées. On pourra utiliser, au moins dans un premier temps, *SoftHSM* qui est un stockage logiciel des clés simulant un véritable *HSM*, mais évidemment pas avec un niveau de sécurité comparable. *SoftHSM* est développé par les même développeurs que OpenDNSSEC.

⁵ <http://www.opendnssec.org/>

OpenDNSSEC utilise aussi un logiciel de base données, soit *SQLite*, soit *MySQL*. Ce dernier est recommandé dans un service en production.

5.3 Configuration

Le principe général de OpenDNSSEC est de recevoir en entrée des données non signées correspondantes à des zones DNS, de gérer en permanence les clés et les signatures de ces zones en fonction de politiques et de générer en sortie les zones signées correspondantes. Le workflow de OpenDNSSEC est illustré sur la figure suivante.

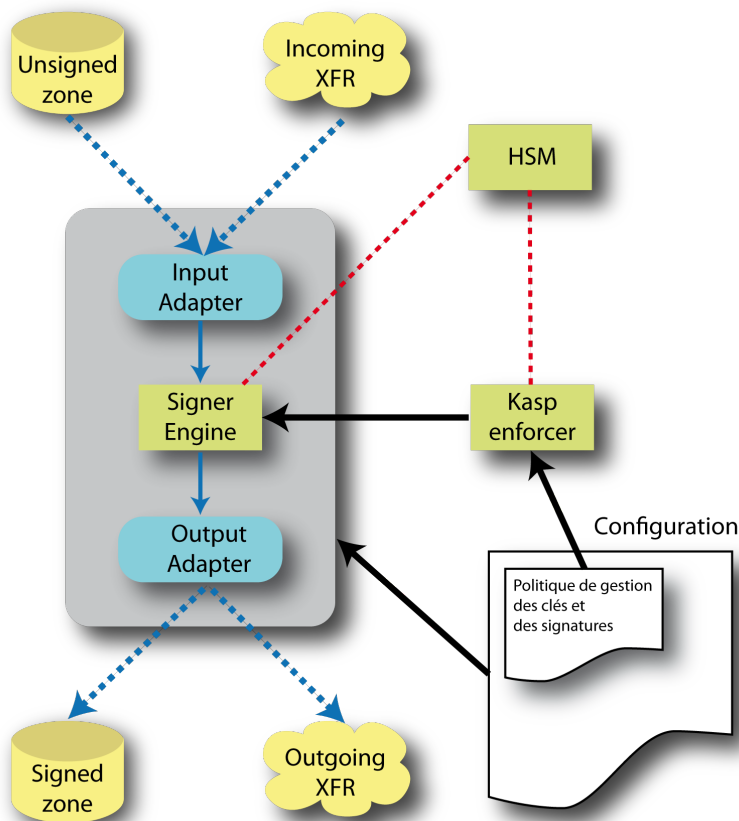


Figure 6 - Workflow de OpenDNSSEC

En entrée, OpenDNSSEC peut recevoir les données soit à partir de fichiers de zones non signées, soit par un transfert XFR depuis un serveur DNS. De la même façon en sortie, les zones signées peuvent être mises dans des fichiers de zones ou transférées directement par XFR à un serveur DNS.

Les fichiers de configuration permettent de spécifier toute la politique de gestion des clés et des signatures, le format des sources de données ainsi que celui de leur destination... Ces fichiers se trouvent par défaut dans `/etc/opendnssec` :

- **conf.xml** : fichier de configuration général ;
- **kasp.xml** : permet de définir une ou plusieurs politiques contenant tous les paramètres DNSSEC à appliquer pour une zone: paramètres temporels, formats et algorithmes pour les clés et les signatures, etc. ;
- **zonelist.xml** : contient la liste des zones DNS à gérer ainsi que la politique à leur appliquer ;
- **addns.xml** : spécifications des paramètres concernant les serveurs DNS pour les transferts XFR en entrée et en sortie.

Ces fichiers sont très bien documentés⁶. Les valeurs proposées par défaut pour la plupart des paramètres temporels sont tout à fait raisonnables pour une plateforme de production. Pour une plateforme de tests, il est conseillé de réduire

⁶ <https://wiki.opendnssec.org/display/DOCS/OpenDNSSEC+Documentation+Home>

fortement les paramètres temporels proposés par défaut afin de provoquer plus rapidement le roulement des clés et la régénération des signatures.

La configuration des paramètres temporels est la partie évidemment la plus critique. Celle-ci doit prendre en compte aussi bien les aspects permettant la bonne validation des enregistrements, mais aussi les problèmes de charge des serveurs DNS.

5.4 Utilisation

5.4.1 Initialisation

Une fois la configuration terminée, y compris le HSM, il faut :

- initialiser la base de données via la commande « `ods-ksmutil setup` » ;
- démarrer OpenDNSSEC pour la 1^{ère} fois via la commande « `ods-control start` » (2 processus sont normalement lancés : `ods-enforcerd` et `ods-signerd`);

A ce stade, les clés KSK et ZSK ont été créées et la zone a été signée. On peut vérifier l'état des clés avec la commande suivante :

```
# ods-ksmutil key list
Keys:
Zone:                Keytype:    State:    Date of next transition:
educ-rech.fr        KSK        publish  2013-11-05 11:15:00
educ-rech.fr        ZSK        active   2013-11-05 13:08:00
```

On voit que 2 paires de clés ont été créées, une clé KSK et une clé ZSK. La clé ZSK est active et est donc tout de suite utilisée pour signer la zone. La clé KSK est dans l'état *publish* ce qui signifie qu'elle n'est pas encore utilisée. Elle restera dans cet état le temps qu'elle soit propagée dans le DNS, et passera alors à l'état *ready*.

```
# ods-ksmutil key list
Keys:
Zone:                Keytype:    State:    Date of next transition:
educ-rech.fr        KSK        ready    waiting for ds-seen
educ-rech.fr        ZSK        active   2013-11-05 13:08:00
```

L'état *ready* indique que l'empreinte de la clé peut-être transmise à la zone parente. Pour cela, la commande ci-dessous permet les informations nécessaires :

```
# ods-ksmutil key export -zone educ-rech.fr --keystate ready --ds

;ready KSK DS record (SHA1):
educ-rech.fr.        60      IN      DS      42607 8 1 e3a09a742a195593dc5dc5041b8ad9b5ace2127f

;ready KSK DS record (SHA256):
educ-rech.fr.        60      IN      DS      42607 8 2
6a1448ecc94b91fcd10d71e8ae2c50579b24e51ee2b3b9adc1d92ed9fbc2e468
```

La procédure pour transmettre ces informations dépendent du bureau d'enregistrement utilisé pour déclarer le domaine dans la zone parente (par exemple le registre AFNIC pour *.fr*) : formulaire web, EPP⁷, ...

Il faut alors attendre que l'enregistrement DS soit effectivement publié. On peut le vérifier avec la commande *dig* par exemple :

```
# dig @a.nic.fr educ-rech.fr DS

;; QUESTION SECTION:
;educ-rech.fr.                IN      DS

;; ANSWER SECTION:
educ-rech.fr.                172800 IN      DS      54 8 2
```

⁷ http://fr.wikipedia.org/wiki/Extensible_Provisioning_Protocol

```
264DF9086290454F3D45275E60F70A1273AAFD614E551BAE5C4E3D6C DC6F3B81
```

```
...
```

D'autres outils⁸ en lignes sont disponibles pour vérifier les zones DNSSEC.

Une fois que c'est fait, on va pouvoir l'indiquer à OpenDNSSEC avec la commande suivante :

```
# ods-ksmutil key ds-seen -zone educ-rech.fr --keytag 42607
Found key with CKA_ID 0bdb802ed3eb751d0e53cde45206782f
Key 0bdb802ed3eb751d0e53cde45206782f made active
Notifying enforcer of new database...
```

Ce qui a pour conséquence de modifier l'état de la clé KSK qui devient *active*.

```
# ods-ksmutil key list
Keys:
Zone:                Keytype:      State:      Date of next transition:
educ-rech.fr         KSK           active     2013-11-05 14:56:38
educ-rech.fr         ZSK           active     2013-11-05 13:08:00
```

L'initialisation est alors terminée et la zone DNS est maintenant signée. On peut le vérifier :

```
# dig educ-rech.fr +dnssec
; ...
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 52878
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 3, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;educ-rech.fr.                IN      A

;; ANSWER SECTION:
educ-rech.fr.                5400    IN      A      193.49.159.110
educ-rech.fr.                5400    IN      RRSIG  A 8 2 5400 20131111012334      20131104072841
34440 educ-rech.fr. Y5X+R9UQ9ft...BuRTAfffHFDCUBvb1Xo3 I8M=

;; AUTHORITY SECTION:
educ-rech.fr.                5400    IN      NS     dnssec2.renater.fr.
educ-rech.fr.                5400    IN      NS     dnssec.renater.fr.
educ-rech.fr.                5400    IN      RRSIG  NS 8 2 5400 201311110140751 20131103112840
34440 educ-rech.fr. Nj0F4B5QJsE/Nl...xmQy6iHKXx h/E=
```

Le flag *ad* indique que la signature de la réponse DNS est valide. Ce flag est positionné uniquement si le résolveur effectue la validation DNSSEC.

5.4.2 Opérations courantes

OpenDNSSEC se charge de toutes les opérations courantes de gestion des signatures et des clés. La bonne marche de ces opérations doit être contrôlée par un système de supervision adéquat, qui devra en particulier vérifier l'état des clés[15].

Le moment venu, OpenDNSSEC se chargera de créer une nouvelle clé KSK pour remplacer la clé KSK active. Il faudra alors lancer la procédure de roulement normale des clés. Les 2 clés KSK existeront donc en même temps, la nouvelle restant dans l'état *ready* jusqu'à publication du nouveau DS dans la zone parente.

Comme les clés n'ont pas de durée de vie au niveau de DNSSEC (contrairement aux signatures), la procédure de roulement peut être faite n'importe quand, le seul critère étant le temps que l'on estime raisonnable de garder une même clé pour la signature de ses zones.

En plus des opérations automatisées, OpenDNSSEC permet d'autres opérations, par exemple forcer la re-signature d'une zone ou le roulement des clés.

⁸ <http://dnssec-debugger.verisignlabs.com/> et <http://dnsviz.net/>

6 Conclusion

La racine du DNS a été signée en 2010 et aujourd'hui pratiquement tous les TLDs le sont également (la zone *.fr* a été signée en septembre 2010). La situation est donc favorable depuis un certain temps déjà pour une propagation de DNSSEC dans les établissements, et pourtant on est loin d'une généralisation de celui-ci. La principale raison est assez classique, on l'a connu pour d'autres évolutions, en particulier dans le domaine de la sécurité. Même si la tâche n'est pas insurmontable comme cet article a essayé de le montrer, passer à DNSSEC demande du travail et des ressources, et n'apporte aucune plus-value au niveau du service. Dans un contexte où les équipes voient leurs ressources diminuer et leur charge de travail augmenter, la migration vers DNSSEC n'est évidemment pas vue comme une priorité. Pour autant, au vu des menaces croissantes qui progressent régulièrement, avons nous vraiment le choix concernant un service aussi crucial que le DNS ?

7 Bibliographie

- [1] R. Arends, R. Austein, M. Larson, D. Massey, S. Rose, RFC 4033 « *DNS Security Introduction and Requirements* », mars 2005
- [2] R. Arends, R. Austein, M. Larson, D. Massey, S. Rose, RFC 4034 « *Resource Records for the DNS Security Extensions* », mars 2005
- [3] R. Arends, R. Austein, M. Larson, D. Massey, S. Rose, RFC 4035 « *Protocol Modifications for the DNS Security Extensions* », mars 2005
- [4] M. StJohns, RFC 5011 « *Automated Updates of DNSSEC Trust Anchors* », septembre 2007
- [5] B. Laurie, G. Sisson, R. Arends, D. Blacka, RFC 5155, « *DNSSEC Hashed Authenticated Denial of Existence* », mars 2008
- [6] J. Jansen, RFC 5702, « *Use of SHA2 Algorithms with RSA in DNSKEY and RRSIG Resource records for DNSSEC* », octobre 2009
- [7] DNSSEC: DNS Security Extensions, <http://www.dnssec.net/>
- [8] Bertrand Léonard, « *Sécurisation du DNS : les extensions DNSsec* », JRES 2003, <http://2003.jres.org/actes/paper.107.pdf>
- [9] Stéphane Bortzmeyer, « *Sécurité du DNS et DNSSEC* », JRES 2009, https://2009.jres.org/planning_files/article/pdf/5.pdf
- [10] Documentation NIST, « *Secure Domain Name System (DNS) Deployment Guide* », <http://csrc.nist.gov/publications/nistpubs/800-81r1/sp-800-81r1.pdf>
- [11] Documentation AFNIC, « *Déployer DNSSEC, comment, quoi, où ?* », <https://www.afnic.fr/medias/documents/DNSSEC/afnic-dnssec-howto-fr-v1.pdf>
- [12] S. Weiler, RFC 5074, « *DNSSEC Lookaside Validation (DLV)* », novembre 2007
- [13] O. Kolkman, W. Mekking, NLnet Labs, R. Gieben, SIDN Labs, RFC 6781, « *DNSSEC Operational Practices, Version 2* », décembre 2012
- [14] S. Morris, J. Ihren, J. Dickinson, « *DNSSEC Key Timing Considerations* », <http://tools.ietf.org/html/draft-ietf-dnsop-dnssec-key-timing-03>
- [15] P. Vixie, O. Gudmundsson, D. Eastlake 3rd, B. Wellington, RFC 2845 « *Secret Key Transaction Authentication for DNS (TSIG)* », <http://www.ietf.org/rfc/rfc2845.txt>
- [16] Stéphane Bortzmeyer, « *Monitoring DNSSEC zones: what, how and when?* », <http://conferences.npl.co.uk/satin/papers/satin2011-Bortzmeyer.pdf>
- [17] Guillaume Valadon, Yves-Alexis Perez, « *Architectures DNS sécurisées* », http://www.sstic.org/media/SSTIC2011/SSTIC-actes/architecture_dns_scurise/SSTIC2011-Slides-architecture_dns_scurise-valadon_perez.pdf