Groupe de travail Réseau
Request for Comments : 4641
RFC rendue obsolète : 2541
Catégorie : Information

O. Kolkman, RIPE NCC
R. Gieben, NLnet Labs
septembre 2006
Traduction Claude Brière de L'Isle

Pratiques de fonctionnement de DNSSEC

Résumé
Le présent document décrit un ensemble de pratiques pour faire fonctionner le DNS avec ses extensions de sécurité (DNSSEC). L'audience cible est les administrateurs de zone qui déploient DNSSEC.

Le document discute les aspects opérationnels de l'utilisation des clés et signatures dans le DNS. Il discute les questions de génération de clés, de mémorisation de clés, de génération de signature, de roulement de clés, et des politiques qui s'y rapportent.

Le présent document rend obsolète la RFC 2541, car il couvre un domaine de fonctionnement plus large et donne des exigences plus à jour à l'égard des tailles de clés et de la nouvelle spécification de DNSSEC.

Table des matières

1. Introduction

Le présent document décrit comment gérer un environnement intégrant la sécurité du DNS (DNSSEC, *DNS Security*). Il est destiné aux opérateurs qui ont connaissance du DNS (voir les [RFC1034] et [RFC1035]) et veulent déployer DNSSEC. Voir la [RFC4033] pour une introduction à DNSSEC, la [RFC4034] pour les nouveaux enregistrements de ressource (RR, *Resource Record*) introduits, et la [RFC4035] pour les changements du protocole.

Durant les ateliers et essais de déploiement opérationnel précoces, les opérateurs et administrateurs de système ont gagné en expérience sur le fonctionnement du DNS avec ses extensions de sécurité (DNSSEC). Le présent document traduit ces expériences en un ensemble de pratiques pour les administrateurs de zone. Au moment de sa rédaction, il existe très peu d'expérience de DNSSEC dans des environnements de production ; le présent document devrait donc ne pas être vu explicitement comme représentant les "Bonnes pratiques courantes".

Les procédures de ce document se concentrent sur la maintenance des zones signées (c'est-à-dire, les zones signées et publiées sur des serveurs d'autorité). Il est prévu que la maintenance de zones comme de resignature ou des roulements de clés soient transparente à tout clients vérificateurs sur l'Internet.

La structure de ce document est la suivante. Dans la Section 2, on discute de l'importance de garder intacte la "chaîne de confiance". Les aspects de génération et de mémorisation des clés privées sont discutés à la Section 3 ; l'objet de cette section est principalement la partie privée de la ou des clés. La Section 4 décrit les considérations concernant la partie publique des clés. Comme ces clés publiques apparaissent dans le DNS, on doit tenir compte de toutes les questions de temps, qui sont discutées dans le paragraphe 4.1. Les paragraphes 4.2 et 4.3 traitent du roulement, ou substitution des clés. Finalement, le paragraphe 4.4 discute des considérations sur la façon dont les parents traitent les clés publiques de leurs enfants afin de maintenir les chaînes de confiance.

Les conventions typographiques utilisées dans ce document sont expliquées dans l'Appendice C.

Comme ce document fait des suggestions de fonctionnement et qu'il n'y a pas de spécifications de protocole, le langage de la [RFC2119] n'est pas appliqué.

Le présent document rend obsolète la [RFC2541] pour refléter l'évolution du protocole DNSSEC sous-jacent depuis lors. Les changements des choix d'algorithmes de chiffrement, des types d'enregistrement et noms de types du DNS, et d'échange de clé parent-enfant et de signature demandaient une réécriture majeure et des informations et explications supplémentaires.

1.1 Utilisation du terme de 'clé'

On suppose que le lecteur est familier du concept de clés asymétriques sur lequel est fondé DNSSEC (cryptographie à clé publique [Schneier]). Donc, le présent document va utiliser le terme de "clé" de façon assez lâche. Lorsque il est écrit que "une clé est utilisée pour signer des données" il est supposé que le lecteur comprend que c'est la partie privée de la paire de clés qui est utilisé pour signer. Il est aussi supposé que le lecteur comprend que la partie publique de la paire de clés est publiée dans l'enregistrement de ressource DNSKEY et que c'est la partie publique qui est utilisée dans les échanges de clé.

1.2 Définitions du temps

Dans ce document, on utilise un certain nombre de termes relatifs au temps. Les définitions suivantes s'appliquent :

- Période de validité de signature : période où une signature est valide. Elle commence à l'instant spécifié dans le champ Début de signature du RR RRSIG et se termine à l'instant spécifié dans le champ Expiration du RR RRSIG.

- Période de publication de signature : temps après lequel une signature (faite avec une clé spécifique) est remplacée par une nouvelle signature (faite avec la même clé). Ce remplacement a lieu par la publication du RRSIG pertinent dans le fichier maître de zone. Après qu'on arrête de publier un RRSIG dans une zone, il peut se passer un certain temps avant que le RRSIG expire dans les antémémoires et soit réellement retiré du DNS.

- Période d'efficacité de clé : période durant laquelle une paire de clés est supposée être efficace. Cette période est définie comme le temps qui s'écoule entre le premier horodatage et la dernière date d'expiration de toute signature faite avec cette clé, sans considération des discontinuités éventuelles de l'utilisation de la clé. La période d'efficacité de clé peut s'étendre sur plusieurs périodes de validité de signature.

o Durée de vie maximum/minimum de zone : valeur maximum ou minimum des durées de vie (TTL, *Time To Live*) de l'ensemble complet des RR d'une zone. Noter que le TTL minimum n'est pas le même que le champ MINIMUM dans le RR SOA. Voir plus d'informations dans la [RFC2308].

2. Conserver intacte la chaîne de confiance

Maintenir une chaîne de confiance valide est important parce que des chaînes de confiance rompues résultent en le marquage de données comme boguées (comme défini à la Section 5 de la [RFC4033]) ce qui peut causer l'invisibilité de sous domaines entiers aux clients vérificateurs. Les administrateurs de zones sécurisées doivent réaliser que leur zone fait, pour les clients vérificateurs, partie d'une chaîne de confiance.

Comme mentionné dans l'introduction, les procédures présentées ici sont destinées à assurer que la maintenance des zones, comme la resignature ou le roulement des clés, va être transparente au clients vérificateurs sur l'Internet.

Les administrateurs de zones sécurisées devront garder à l'esprit que les données publiées sur un serveur principal d'autorité ne vont pas être immédiatement vues par les clients vérificateurs ; il peut s'écouler un certain temps avant que les données soient transférées aux autres serveurs de noms secondaires d'autorité et les clients peuvent aller chercher les données auprès d'antémémoires de serveurs non d'autorité. Sous cet aspect, noter que le temps pour un transfert de zone de maître à esclave est négligeable quand on utilise NOTIFY [RFC1996] et le transfert incrémental (IXFR) [RFC1995]. Il augmente quand des transferts de zone complets (AXFR) sont utilisés en combinaison avec NOTIFY. Il augmente encore plus si on s'appuie sur des transferts de zone complets fondés seulement sur les paramètres de temps de SOA pour le rafraîchissement.

Pour les clients vérificateurs, il est important que les données provenant des zones sécurisées puissent être utilisées pour construire des chaînes de confiance sans considération de si les données viennent directement d'un serveur d'autorité, d'une antémémoire de serveur de noms, ou d'un boîtier de médiation. C'est seulement en utilisant avec prudence les paramètres de temps disponibles qu'un administrateur de zone peut s'assurer que les données nécessaires pour la vérification peuvent être obtenues.

La responsabilité du maintien de la chaîne de confiance est partagée par les administrateurs des zones sécurisées dans la chaîne de confiance. C'est très évident dans le cas d'une "clé compromise" quand il faut arbitrer entre le maintien d'une chaîne de confiance valide et le remplacement aussitôt que possible des clés compromises. Les administrateurs de zone vont alors devoir arbitrer entre garder intacte la chaîne de confiance – permettant par là des attaques avec la clé compromise -- ou rompre délibérément la chaîne de confiance et rendre des sous domaines sécurisés invisibles aux résolveurs soucieux de sécurité. Voir aussi au paragraphe 4.3.

3. Génération et mémorisation des clés

Cette section décrit un certain nombre de considérations à l'égard de la sécurité des clés. Elle traite de la génération, de la période d'efficacité, de la taille, et de la mémorisation, des clés privées.

3.1 Clés de zone et clés de signature

Le protocole de validation DNSSEC ne distingue pas entre les différents types de DNSKEY. Tous les DNSKEY peuvent être utilisés durant la validation. En pratique, les opérateurs utilisent les clés de signature de clé et les clés de signature de zone, et utilisent le fanion de point d'entrée sécurisée (SEP, *Secure Entry Point*) [RFC3757] pour distinguer entre elles durant le fonctionnement. On discute ensuite de la dynamique et des problèmes.

Pour faciliter la mise en œuvre des procédures de resignature de zone et de roulement de clé, il est possible d'utiliser une ou plusieurs clés comme clés de signature de clé (KSK, *Key Signing Key*). Ces clés vont seulement signer le RRSet de DNSKEY sommital d'une zone. D'autres clés peuvent être utilisées pour signer tous les RRSet dans une zone et sont appelées les clés de signature de zone (ZSK, *Zone Signing Key*). Dans le présent document, on suppose que les KSK sont le sous ensemble de clés qui sont utilisées pour les échanges de clés avec la parente et potentiellement pour la configuration comme ancres de confiance – les clés SEP. Dans le présent document, on suppose une transposition biunivoque entre KSK et SEP et on suppose que le fanion SEP est établi sur toutes les KSK.

3.1.1 Motivations de la séparation de KSK et ZSK

Différencier les fonctions de KSK et de ZSK a plusieurs avantages :
- o Aucune interaction parent/enfant n'est requise quand les ZSK sont mises à jour.
- o La KSK peut être rendue plus forte (c'est-à-dire, utiliser plus de bits dans le matériel de clé). Ceci n'a que peu d'impact sur le fonctionnement car elle n'est utilisée que pour signer une petite fraction des données de zone. Aussi, la KSK est seulement utilisée pour vérifier l'ensemble de clés de la zone, pas pour les autres RRSet dans la zone.
- o Comme la KSK est seulement utilisée pour signer un ensemble de clés, qui est très probablement mis à jour moins fréquemment que d'autres données dans la zone, elle peut être mémorisée séparément, et dans des sites plus sûrs, que la ZSK.
- o Une KSK peut avoir une période d'efficacité de clé plus longue.

Pour presque toutes les méthodes de gestion de clé et de signature de zone, la KSK est utilisée moins fréquemment que la ZSK. Une fois qu'un ensemble de clés est signé avec la KSK, toutes les clés dans l'ensemble de clés peuvent être utilisées comme des ZSK. Si une ZSK est compromise, elle peut être simplement éliminée de l'ensemble de clés. Le nouvel ensemble de clés est alors resigné avec la KSK.

Étant donnée l'hypothèse que pour les KSK le fanion SEP est établi, la KSK peut être distinguée d'une ZSK en examinant le champ Fanions dans le RR DNSKEY. Si le champ Fanions est un nombre impair, c'est une KSK. Si c'est un nombre pair, c'est une ZSK.

La clé de signature de zone peut être utilisée pour signer toutes les données dans une zone sur une base régulière. Quand une clé de signature de zone doit être enroulée, aucune interaction avec le parent n'est nécessaire. Cela permet des périodes de validité de signature de l'ordre de plusieurs jours.

La clé de signature de clé n'est utilisée que pour signer les RR DNSKEY dans une zone. Si une clé de signature de clé doit être enroulée, il va y avoir des interactions avec des parties autres que l'administrateur de zone. Cela peut inclure le registre de la zone parente ou les administrateurs des résolveurs vérificateurs qui ont la clé particulière configurée comme des points d'entrée sûrs. Donc, la période d'efficacité de clé de ces clés peut et devrait être plus longue. Cependant,avec une clé assez longue, la période d'efficacité de clé peut être de l'ordre de plusieurs années, on suggère de planifier une efficacité de clé de l'ordre de quelques mois afin que le roulement d'une clé reste une routine de fonctionnement.

3.1.2 KSK pour les zones de haut niveau

Les zones de niveau supérieur sont généralement plus sensibles que celles de niveau inférieur. Quiconque contrôle ou casse la sécurité d'une zone obtient par là l'autorité sur tous ses sous domaines (sauf dans le cas de résolveurs qui ont configuré en local la clé publique d'un sous domaine, et dans ce cas ce sous domaine, et lui seul, ne serait pas affecté par la compromission de la zone parente). Donc, un soin supplémentaire devrait être pris avec les zones de niveau élevé, et des clés fortes devraient être utilisées.

La zone racine est la plus critique de toutes les zones. Quelqu'un qui contrôle ou compromet la sécurité de la zone racine contrôlerait l'espace de noms entier du DNS de tous les résolveurs qui utilisent la zone racine (sauf dans le cas de résolveurs qui ont configuré en local la clé publique d'un sous domaine). Donc, le plus grand soin doit être pris de la sécurisation de la zone racine. Les clés les plus fortes et traitées avec le plus grand soin devraient être utilisées. La clé privée de la zone racine devrait toujours être conservée hors ligne.

De nombreux résolveurs vont commencer comme serveur racine pour leur accès aux données du DNS et leur authentification. Mettre à jour en toute sécurité les ancres de confiance dans une énorme population de résolveurs tout autour du monde va être extrêmement difficile.

3.2 Génération des clés

Une génération attentive de toutes les clés est un élément parfois surestimé mais absolument essentiel dans tout système cryptographiquement sécurisé. Les plus forts algorithmes utilisés avec les plus longues clés ne sont d'aucune utilité si un adversaire peut en deviner assez pour diminuer la taille de l'espace de clés probable afin qu'il puisse faire l'objet d'une recherche exhaustive. Des suggestions techniques pour la génération de clés aléatoires se trouvent dans la [RFC4086]. On devrait vérifier attentivement si le générateur de nombres aléatoires utilisé durant la génération de clé adhère à ces suggestions.

Les clés avec une longue période d'efficacité sont particulièrement sensibles car elles vont représenter une cible de valeur et

être soumises à des attaques pendant plus longtemps que des clés de période courte. Il est fortement recommandé que la génération de clé de long terme se fasse hors ligne d'une manière isolée du réseau via un trou d'air ou, au minimum, un matériel très sécurisé.

3.3 Période d'efficacité de la clé

Pour diverses raisons, les clés dans DNSSEC ont besoin d'être changées une fois de temps en temps. Plus longtemps une clé est utilisée, plus grande est la probabilité qu'elle soit compromise par négligence, accident, espionnage, ou cryptanalyse. De plus, quand les roulements de clés sont un événement trop rare, ils ne font pas partie des habitudes de fonctionnement et il y a un risque que personne sur le site ne se souvienne des procédure du roulement quand le besoin s'en fait sentir.

D'un point de vue purement opérationnel, une période d'efficacité de clé raisonnable pour les clés de signature de clés est de 13 mois, avec l'intention de les remplacer après 12 mois. Une période d'efficacité de clé prévue de un mois est raisonnable pour les clés de signature de zone.

Pour les tailles de clé qui respectent ces périodes d'efficacité, voir au paragraphe 3.5.

Comme expliqué au paragraphe 3.1.2, mettre à jour de façon sécurisée les ancres de confiance va être extrêmement difficile. Par ailleurs, l'argument de "l'habitude de fonctionnement" s'applique aussi à la reconfiguration d'ancre de confiance. Si une période d'efficacité de clé courte est utilisée et si la configuration d'ancre de confiance doit être revisitée de façon régulière, le risque que la configuration tende à être oubliée est plus faible. L'arbitrage est contre un système qui est si dynamique que les administrateurs des clients valideurs ne seront pas capables de suivre les modifications.

Les périodes d'efficacité de clés peuvent être rendues très courtes, comme de quelques minutes. Mais quand on remplace les clés, on doit prendre en compte les considérations des paragraphes 4.1 et 4.2.

3.4 Algorithme de clé

Il y a actuellement trois types différents d'algorithmes qui peuvent être utilisés dans DNSSEC : RSA, DSA, et la cryptographie à courbe elliptique. Cette dernière est très nouvelle et doit d'abord être normalisée pour l'usage de DNSSEC.

RSA a été développé d'une manière ouverte et transparente. Comme le brevet sur RSA a expiré en 2000, son utilisation est aussi libre maintenant.

DSA a été développé par le National Institute of Standards and Technology (NIST). La création de signatures prend en gros le même temps qu'avec RSA, mais est 10 à 40 fois plus lente pour la vérification [Schneier].

On suggère l'utilisation de RSA/SHA-1 comme algorithme préféré pour la clé. Les attaques actuelles connues sur RSA peuvent être déjouées en allongeant la clé. Comme l'algorithme de hachage MD5 montre des faiblesses, on recommande l'usage de SHA-1.

Au moment de la publication, il est connu que le hachage SHA-1 pose des problèmes de cryptanalyse. Des travaux sont en cours sur le traitement de ces problèmes. On recommande l'utilisation d'algorithmes de clé publique fondés sur des hachages plus forts que SHA-1 (par exemple, SHA-256) sitôt que ces algorithmes seront disponibles dans les spécifications de protocole (voir les [RFC5702] et [RFC4509]) et les mises en œuvre.

3.5 Tailles de clé

Quand ils choisissent les tailles de clé, les administrateurs de zone vont devoir prendre en compte le temps pendant lequel une clé va être utilisée, quelle quantité de données vont être signées durant la période de publication de la clé (voir le paragraphe 8.10 de [Schneier]) et, facultativement la taille de clé de la zone parente. Comme la chaîne de confiance est réellement "une chaîne", il n'y a pas grand sens a faire une des clés dans la chaîne plusieurs fois plus grande que les autres. Comme toujours, c'est le maillon le plus faible qui définit la force de la chaîne entière. Voir aussi au paragraphe 3.1.1 la discussion de comment des clés servant à différents rôles (ZSK/KSK) peuvent avoir besoin de tailles de clé différentes.

Générer une clé de la taille correcte est un problème difficile ; la [RFC3766] essaye de le traiter. La première partie de la procédure de choix à la Section 1 de cette RFC déclare :

1. Déterminer la résistance aux attaques nécessaire pour satisfaire aux exigences de sécurité de l'application. Le faire en

estimant le nombre minimum d'opérations informatiques que l'attaquant sera forcé de faire afin de compromettre la sécurité du système et ensuite prendre le logarithme base deux de ce nombre. Appelons cette valeur de logarithme "n". Un rapport de 1996 recommandait 90 bits comme un bon choix tout compris pour la sécurité des systèmes. Le nombre de 90 bits devrait être augmenté d'environ 2/3 bits/an, ou environ 96 bits en 2005.

La [RFC3766] continue en expliquant comment ce nombre "n" peut être utilisé pour calculer les tailles de clé dans la cryptographie à clé publique. Le point culminant est le tableau ci-dessous (légèrement modifié pour nos besoins) :

Exigences système pour la résistance aux attaques (bits)	**Taille de clé symétrique** (bits)	**Taille de module RSA ou DH** (bits)
70	70	947
80	80	1228
90	90	1553
100	100	1926
150	150	4575
200	200	8719
250	250	14596

Les tailles de clé données sont assez grandes. C'est parce que ces clés sont résistantes à l'attaque du trillionnaire. En supposant que ce riche attaquant ne va pas attaquer la clé et que la clé est relevée une fois par an, on en vient aux recommandations suivantes sur les tailles de KSK : 1024 bits pour les domaines de faible valeur, 1300 bits pour les domaines de valeur moyenne, et 2048 bits pour les domaines de grande valeur.

Si la valeur d'un domaine est faible, moyenne, ou élevée dépend seulement des vues du propriétaire de zone. On pourrait, par exemple, voir les nœuds d'extrémité dans le DNS comme de faible valeur, et les domaine de niveau supérieur (TLD, *Top-Level Domain*) ou la zone racine comme de haute valeur. Les tailles de clé suggérées devraient être sûres pour les cinq prochaines années.

Comme les ZSK peuvent être remplacées plus facilement (et donc plus souvent) les tailles de clé peuvent être plus petites. Mais comme on l'a dit dans l'introduction de ce paragraphe, rendre les tailles de clé de ZSK trop petites (en relation avec les tailles de KSK) n'a pas grand sens. Essayer de limiter la différence de taille à environ 100 bits.

Noter que personne ne peut prédire l'avenir et que ces tailles de clé ne sont données que comme indication. On trouvera plus d'informations dans [Sizes] et au paragraphe 7.5 de [Schneier]. On devrait aussi noter que [Sizes] est déjà considéré comme trop optimiste quant aux tailles de clé qui sont considérées comme sûres.

Une note finale concernant les tailles de clé. Les clé plus grandes vont augmenter les tailles des enregistrements RRSIG et DNSKEY et vont donc augmenter les chances de débordement de paquet UDP dans le DNS. Aussi, le temps que prend la validation et la création des RRSIG augmente avec de plus grandes clés, donc il ne faut pas sans nécessité doubler ses tailles de clé.

3.6 Mémorisation des clés privées

Il est recommandé que, lorsque possible, les clés privées de zone et la copie du fichier maître de zone qui sont à signer soient conservées et utilisées hors ligne, non connectées au réseau, et seulement sur des machines physiquement sûres. Périodiquement, une application peut être lancée pour ajouter l'authentification à une zone en ajoutant des RR RRSIG et NSEC. C'est alors que le fichier augmenté peut être transféré.

Quand on s'appuie sur la mise à jour dynamique pour gérer une zone signée [RFC3007], il faut savoir qu'au moins une clé privée de la zone va devoir résider sur le serveur maître. Cette clé n'est sûre qu'autant que la quantité d'exposition du serveur reçue des clients inconnus et de la sécurité de l'hôte. Bien que ce ne soit pas obligatoire, on pourrait administrer le DNS de la façon suivante. Le maître qui traite les mises à jour dynamiques est indisponible à partir des hôtes génériques sur l'Internet, il ne figure pas sur la liste de l'ensemble de RR NS, bien que son nom apparaisse dans le champ MNAME des RR SOA. Les serveurs de noms dans le RRSet NS sont capables de recevoir des mises à jour de zone par des NOTIFY, IXFR, AXFR, ou un mécanisme de distribution hors bande. Cette approche est connue sous le nom de "maître caché".

La situation idéale est d'avoir un flux d'informations unidirectionnel vers le réseau pour éviter la possibilité d'altération à partir du réseau. Garder le fichier maître de zone en ligne sur le réseau et le dérouler simplement à travers un signataire hors ligne ne le fait pas. La version en ligne pourrait encore être altérée si l'hôte sur lequel il réside est compromis. Pour une sécurité maximale, la copie maîtresse du fichier de zone devrait être hors réseau et ne devrait pas être mise à jour sur la

base d'une communication établie par l'intermédiaire d'un réseau non sûr.

En général, garder un fichier de zone hors ligne ne va pas être pratique et les machines sur lesquelles les fichiers de zone sont conservés vont être connectées à un réseau. Il est conseillé aux opérateurs de prendre de mesures de sécurité pour faire un bouclier contre l'accès non autorisé à la copie maîtresse.

Pour les mises à jour dynamiques des zones sécurisées [RFC3007], la copie maîtresse et la clé privée qui est utilisée pour mettre à jour les signatures sur les RR mis à jour devront être en ligne.

4. Génération de signature, roulement de clé, et politiques en rapport

4.1 L'heure dans DNSSEC

Sans DNSSEC, tous les temps dans le DNS sont relatifs. Les champs SOA REFRESH, RETRY, et EXPIRATION sont des temporisateurs utilisés pour déterminer le temps écoulé après qu'un serveur esclave s'est synchronisé avec un serveur maître. La valeur de durée de vie (TTL, *Time to Live*) et le paramètre de RR SOA TTL minimum [RFC2308] sont utilisés pour déterminer pendant combien de temps un transmetteur devrait garder en antémémoire des données après qu'elles ont été prises auprès d'un serveur d'autorité. En utilisant une période de validité de signature, le DNSSEC introduit la notion d'un temps absolu dans le DNS. Les signatures dans DNSSEC ont une date d'expiration après laquelle la signature est marquée invalide et les données signées sont considérées comme boguées.

4.1.1 Considérations d'heure

À cause de l'expiration des signatures, on devrait considérer ce qui suit :

- On suggère que le TTL maximum de zone des données de zone soit une fraction de la période de validité de signature.
  Si le TTL est d'ordre similaire que la période de validité de signature, alors tous les RRSet collectés durant la période de validité vont être mis en antémémoire jusqu'à l'heure d'expiration de la signature. Le paragraphe 7.1 de la [RFC4033] suggère que "le résolveur peut utiliser le temps restant avant l'expiration de la période de validité de la signature d'un RRSet signé comme limite supérieure du TTL". Par suite, la charge d'interrogation sur les serveurs d'autorité peut avoir un pic à l'heure d'expiration d'une signature, car c'est aussi l'heure à laquelle expirent simultanément les RR dans les antémémoires.
  Pour éviter des pics de charge d'interrogation, on suggère que le TTL sur tous les RR d'une zone soit au moins un petit peu moins que la période de validité de signature.

- On suggère que la période de publication de signature se finisse au moins une durée de TTL maximum de zone avant la fin de la période de validité de signature.
  Resigner une zone peu avant la fin de la période de validité de la signature peut causer l'expiration simultanée des données dans les antémémoires. Ceci peut à son tour conduire à des pics dans la charge sur les serveurs d'autorité.

- On suggère que le TTL minimum de zone soit assez long pour aller chercher et vérifier tous les RR dans la chaîne de confiance. Dans des environnements d'essais, il a été démontré [Rose] qu'un TTL bas (moins de 5 à 10 minutes) cause des perturbations à cause des deux problèmes suivants :
  1. Durant la validation, certaines données peuvent expirer avant l'achèvement de la validation. Le valideur devrait être capable de garder toutes les données jusqu'à ce qu'elle soit achevée. Cela s'applique à tous les RR nécessaires pour compléter la chaîne de confiance : les DS, les DNSKEY, les RRSIG, et les réponses finales, c'est-à-dire, le RRSet qui est retourné pour l'interrogation initiale.
  2. De fréquentes vérifications causent une charge sur les serveurs de noms récurrents. Les données aux points de délégation, les DS, les DNSKEY, et les RRSIG bénéficient de la mise en antémémoire. Le TTL sur eux devrait être relativement long.

- Les serveurs esclaves vont devoir être capables d'aller chercher les nouvelles zones signées bien avant que les RRSIG dans la zone desservie par le serveur esclave passe leur heure d'expiration de signature.
  Quand un serveur esclave est désynchronisé de son maître et que les données dans une zone sont signées par des signatures expirées, il peut être mieux que le serveur esclave ne donne aucune réponse.
  Normalement, un serveur esclave qui n'est pas capable de contacter un serveur maître pendant une période étendue va faire expirer une zone. Quand cela se produit, le serveur va répondre différemment aux interrogations sur cette zone. Certains serveurs produisent un SERVFAIL, tandis que d'autres mettent le bit 'AA' à zéro dans les réponses. L'heure d'expiration est réglée dans l'enregistrement SOA et est relative au dernier rafraîchissement réussi entre le maître et les

serveurs esclaves. Il n'existe pas de couplage entre l'expiration de signature des RRSIG dans la zone et le paramètre d'expiration dans le SOA.

Si le serveur dessert une zone DNSSEC, il peut encore se faire que les signatures expirent bien avant que le temporisateur d'expiration du SOA arrive à zéro. Il n'est pas possible d'empêcher complètement cela d'arriver en resserrant les paramètres de SOA. Cependant, les effets peuvent être minimisés lorsque l'heure d'expiration du SOA est égale ou inférieure à la période de validité de signature. La conséquence d'un serveur d'autorité incapable de mettre à jour une zone, tandis que cette zone comporte des signatures expirées, est que des résolveurs non sûrs vont continuer d'être capable de résoudre des données servies par les serveurs esclaves particuliers tandis que les résolveurs qui mettent en œuvre la sécurité vont rencontrer des problèmes parce que les réponses sont marquées comme boguées.

On suggère que le temporisateur d'expiration de SOA soit réglé approximativement à un tiers ou un quart de la période de validité de signature. Cela va permettre que les problèmes de transferts en provenance du serveur maître soient remarqués avant que la signature expire réellement. On suggère aussi que les opérateurs de serveurs de noms qui fournissent des services secondaires développent des "chiens de garde" pour surveiller les expirations de signature à venir dans leurs zones esclaves, et qu'ils prennent les mesures appropriées.

Quand on détermine la valeur du paramètre d'expiration, on doit tenir compte de ce qui suit : quelles sont les chances que tous les secondaires expirent dans la zone ? Dans quel délai peut on joindre un administrateur de serveurs secondaires pour charger une zone valide ? Ces questions ne sont pas spécifiques de DNSSEC mais peuvent influencer le choix des intervalles de validité de signature.

4.2 Roulements de clés

Une clé DNSSEC ne peut pas être utilisée pour toujours (voir le paragraphe 3.3). Donc les roulements de clés -- ou substitutions, comme elles sont parfois appelées -- sont un fait de vie quand on utilise DNSSEC. Les administrateurs de zone qui sont en train de changer leurs clés doivent tenir compte du fait que les données publiées dans les précédentes versions de leur zone continuent de vivre dans les antémémoires. Quand on déploie DNSSEC, cela devient une considération importante ; ignorer les données qui peuvent être dans les antémémoires peut conduire à une perte de service pour les clients.

L'exemple de plus évident se produit quand le matériel de zone signé avec une vieille clé est validé par un résolveur qui n'a pas la vieille clé de zone en antémémoire. Si la vieille clé n'est plus présente dans la zone actuelle, cette validation échoue, marquant les données comme "boguées". Autrement, une tentative pourrait être faite de valider les données signées avec une nouvelle clé contre une vieille clé qui demeure dans une antémémoire locale, résultant aussi en le marquage des données comme "boguées".

4.2.1 Roulement de clé de signature de zone

Pour le "roulement de clé de signature de zone", il y a deux façons de s'assurer que durant le roulement les données encore en antémémoire peuvent être vérifiées avec les nouveaux ensembles de clés ou que les signatures nouvellement générées peuvent être vérifiées avec les clés qui sont encore dans les antémémoires. Une schéma, décrit au paragraphe 4.2.1.2, utilise les doubles signatures ; l'autre utilise la pré-publication de clé (paragraphe 4.2.1.1). Les pour et les contre, et des recommandations sont décrits au paragraphe 4.2.1.3.

4.2.1.1 Roulement de clé pré publiée

Ce paragraphe montre comment effectuer un roulement de ZSK sans avoir besoin de signer deux fois toutes les données dans une zone -- le "roulement de clé pré-publié". Cette méthode a des avantages dans le cas d'une clé compromise. Si la vieille clé est compromise, la nouvelle clé a déjà été distribuée dans le DNS. L'administrateur de zone est alors capable de passer rapidement à la nouvelle clé et retirer la clé compromise de la zone. Un autre avantage majeur est que la taille de la zone ne double pas, comme c'est le cas avec le roulement de ZSK à double signature. On trouvera une explication de la façon de faire cette sorte de roulement à l'Appendice B.

Le roulement de clé pré-publiée implique les quatre étapes suivantes :

initial	nouveau DNSKEY	nouveaux RRSIG	suppression de DNSKEY
SOA0	SOA1	SOA2	SOA3
RRSIG10(SOA0)	RRSIG10(SOA1)	RRSIG11(SOA2)	RRSIG11(SOA3)
DNSKEY1	DNSKEY1	DNSKEY1	DNSKEY1
DNSKEY10	DNSKEY10	DNSKEY10	DNSKEY11
DNSKEY11	DNSKEY11		
RRSIG1 (DNSKEY)	RRSIG1 (DNSKEY)	RRSIG1(DNSKEY)	RRSIG1 (DNSKEY)
RRSIG10(DNSKEY)	RRSIG10(DNSKEY)	RRSIG11(DNSKEY)	RRSIG11(DNSKEY)

Roulement de clé pré publiée

initial : version initiale de la zone : DNSKEY 1 est la clé de signature de clés. DNSKEY 10 est utilisé pour signer toutes les données de la zone, la clé de signature de zone.

nouveau DNSKEY : DNSKEY 11 est introduit dans l'ensemble de clés. Noter qu'aucune signature n'est encore générée avec cette clé, mais cela ne donne aucune sécurité contre des attaques en force brute contre la clé publique. La durée minimum de cette phase de pré roulement est le temps que prend la propagation des données aux serveurs d'autorité plus la valeur de TTL de l'ensemble de clés.

nouveaux RRSIG : à l'étape "nouveaux RRSIG" (SOA série 2) DNSKEY 11 est utilisé pour signer les données dans la zone exclusivement (c'est-à-dire, toutes les signatures provenant de DNSKEY 10 sont retirées de la zone). DNSKEY 10 reste publié dans l'ensemble de clés. De cette façon les données qui ont été chargées dans les antémémoires à partir de la version 1 de la zone peuvent encore être vérifiées avec les ensembles de clés qu'on va chercher de la version 2 de la zone. Le temps minimum pour publier l'ensemble de clés incluant DNSKEY 10 est le temps qu'il faut aux données de zone provenant de la précédente version de la zone pour arriver à expiration dans les vieilles antémémoires, c'est-à-dire, le temps qu'il faut à cette zone pour se propager à tous les serveurs d'autorité plus la valeur de TTL maximum de zone de toutes les données dans la précédente version de la zone.

suppression de DNSKEY : DNSKEY 10 est retiré de la zone. L'ensemble de clés, qui contient maintenant seulement DNSKEY 1 et DNSKEY 11, est resigné avec la DNSKEY 1.

Le schéma ci-dessus peut être simplifié en publiant toujours la "future" clé immédiatement après le roulement. Le schéma ressemblerait à ce qui suit (on montre deux roulements) ; la future clé est introduite dans "nouveau DNSKEY" comme DNSKEY 12 et là encore une encore plus nouvelle, numérotée 13, dans "nouveau DNSKEY (II)" :

initial	nouveaux RRSIG	nouveau DNSKEY
SOA0	SOA1	SOA2
RRSIG10(SOA0)	RRSIG11(SOA1)	RRSIG11(SOA2)
DNSKEY1	DNSKEY1	DNSKEY1
DNSKEY10	DNSKEY10	DNSKEY11
DNSKEY11	DNSKEY11	DNSKEY12
RRSIG1(DNSKEY)	RRSIG1 (DNSKEY)	RRSIG1(DNSKEY)
RRSIG10(DNSKEY)	RRSIG11(DNSKEY)	RRSIG11(DNSKEY)

nouveaux RRSIG (II)	nouveau DNSKEY (II)
SOA3	SOA4
RRSIG12(SOA3)	RRSIG12(SOA4)
DNSKEY1	DNSKEY1
DNSKEY11	DNSKEY12
DNSKEY12	DNSKEY13
RRSIG1(DNSKEY)	RRSIG1(DNSKEY)
RRSIG12(DNSKEY)	RRSIG12(DNSKEY)

Roulement de clé pré publiée, montrant deux roulements

Noter que la clé introduite dans la phase "nouveau DNSKEY" n'est pas encore utilisée pour la production ; la clé privée peut donc être mémorisée de manière physiquement sûre et on n'a pas besoin d'aller la chercher chaque fois qu'une zone a besoin d'être signée.

4.2.1.2 Roulement de double signature de clé de signature de zone

Ce paragraphe montre comment effectuer un roulement de clé ZSK en utilisant le schéma de double signature de données de zone, appelé avec justesse "roulement de double signature".

Durant l'étape "nouveau DNSKEY" la nouvelle version du fichier de zone va devoir être propagée à tous les serveurs d'autorité et les données qui existent dans les antémémoires (distantes) vont devoir expirer, ce qui exige au moins le TTL maximum de zone.

Le roulement de double signature de ZSK implique les trois étapes suivantes :

initiale	**nouveau DNSKEY**	**suppression de DNSKEY**
SOA0	SOA1	SOA2
RRSIG10(SOA0)	RRSIG10(SOA1)	RRSIG11(SOA2)
RRSIG11(SOA1)		
DNSKEY1	DNSKEY1	DNSKEY1
DNSKEY10	DNSKEY10	DNSKEY11
DNSKEY11		
RRSIG1(DNSKEY)	RRSIG1(DNSKEY)	RRSIG1(DNSKEY)
RRSIG10(DNSKEY)	RRSIG10(DNSKEY)	RRSIG11(DNSKEY)
RRSIG11(DNSKEY)		

Roulement de double signature clé de signature de zone

initiale : version initiale de la zone : DNSKEY 1 est la clé de signature de clés. DNSKEY 10 est utilisée pour signer toutes les données de la zone, la clé de signature de zone.

nouveau DNSKEY : à l'étape "nouveau DNSKEY" (SOA série 1) DNSKEY 11 est introduit dans l'ensemble de clés et toutes les données dans la zone sont signées avec DNSKEY 10 et DNSKEY 11. La période de roulement va devoir continuer jusqu'à ce que toutes les données provenant de la version 0 de la zone aient expiré dans les antémémoires distantes. Cela va prendre au moins le TTL maximum de zone de la version 0 de la zone.

suppression de DNSKEY : DNSKEY 10 est supprimé de la zone. Toutes les signatures de DNSKEY 10 sont retirées de la zone. L'ensemble de clés, contenant maintenant seulement DNSKEY 11, est resigné avec DNSKEY 1.

À chaque instance, les RRSIG provenant de la précédente version de la zone peuvent être vérifiés avec le RRSet DNSKEY provenant de la version courante et d'autres façons. Les données provenant de la version courante peuvent être vérifiées avec les données provenant de la précédente version de la zone. La durée de la phase "nouveau DNSKEY" et la période entre les roulements devrait être au moins le TTL maximum de zone.

S'assurer que la phase "nouveau DNSKEY" dure jusqu'à l'heure d'expiration de la signature des données dans la version initiale de la zone est recommandé. De cette façon toutes les antémémoires sont purgées des vieilles signatures. Cependant, cette durée pourrait être considérablement plus longue que le TTL maximum de zone, rendant le roulement une procédure longue.

Noter que dans cet exemple, on a supposé que la zone n'avait pas été modifiée durant le roulement. De nouvelles données peuvent être introduites dans la zone pour autant qu'elles soient signées avec les deux clés.

4.2.1.3 Avantages et inconvénients du schéma

Roulement de clé pré publié : ce roulement n'implique pas de signer deux fois les données de zone. Avant le roulement réel, la nouvelle clé est plutôt publiée dans l'ensemble de clés et est donc disponible pour les attaques de cryptanalyse. Un petit inconvénient est que ce procès exige quatre étapes. Aussi le schéma pré publié implique plus de travail de la zone parente

quand il est utilisé pour les roulements de KSK comme expliqué au paragraphe 4.2.3.

Roulement à double signature de ZSK : l'inconvénient de ce schéma de signature est que durant le roulement, le nombre de signatures double dans la zone ; cela peut être prohibitif si on a de très grosses zones. Un avantage est qu'il n'exige que trois étapes.

4.2.2 Roulements de clé de signature de clé

Pour le roulement d'une clé de signature de clés, les mêmes considérations s'appliquent que pour le roulement d'une clé de signature de zone. Cependant, on peut utiliser un schéma de double signature pour garantir que les vieilles données (seulement l'ensemble de clés sommital) dans les antémémoires peuvent être vérifiées avec un nouvel ensemble de clés et vice versa. Comme seul l'ensemble de clés est signé avec une KSK, les considérations de taille de zone ne s'appliquent pas.

initial	**nouveau DNSKEY**	**changement de DS**	**suppression de DNSKEY**
Parente :			
SOA0	---------->	SOA1	-------->
RRSIGpar(SOA0)	---------->	RRSIGpar(SOA1)	-------->
DS1	---------->	DS2	-------->
RRSIGpar(DS)	---------->	RRSIGpar(DS)	-------->
Fille :			
SOA0	SOA1	-------->	SOA2
RRSIG10(SOA0)	RRSIG10(SOA1)	-------->	RRSIG10(SOA2)
		-------->	
DNSKEY1	DNSKEY1	-------->	DNSKEY2
	DNSKEY2	-------->	
DNSKEY10	DNSKEY10	-------->	DNSKEY10
RRSIG1 (DNSKEY)	RRSIG1 (DNSKEY)	-------->	RRSIG2 (DNSKEY)
	RRSIG2 (DNSKEY)	-------->	
RRSIG10(DNSKEY)	RRSIG10(DNSKEY)	-------->	RRSIG10(DNSKEY)

Stades de déploiement pour un roulement de double signature de clé de signature de clés

initial : version initiale de la zone. Le DS parental pointe sur DNSKEY1. Avant que commence le roulement, la fille devra vérifier quel est le TTL du RR DS qui pointe sur DNSKEY1 -- il est nécessaire durant le roulement et on se réfère à sa valeur comme TTL_DS.

nouveau DNSKEY : Durant la phase "nouveau DNSKEY", l'administrateur de zone génère une seconde KSK, DNSKEY2. La clé est fournie à la parente, et la fille va devoir attendre qu'un nouveau RR DS ait été généré qui pointe sur DNSKEY2. Après la publication de ce RR DS sur tous les serveurs d'autorité pour la zone parente, l'administrateur de zone doit attendre au moins TTL_DS pour s'assurer que le vieux RR DS a expiré dans les antémémoires.

Changement de DS : la parente remplace DS1 par DS2.

suppression de DNSKEY : DNSKEY1 a été supprimée.

Le scénario ci-dessus met la responsabilité de la maintenance d'une chaîne de confiance valide avec la fille. Il est aussi fondé sur la prémisse que la parente a seulement un RR DS (par algorithme) par zone. Un autre mécanisme a été considéré. En utilisant une relation de confiance établie, l'interaction peut être effectuée dans la bande, et la suppression des clés par la fille peut éventuellement être signalée par la parente. Dans ce mécanisme, il y a des périodes où il y a deux RR DS chez la parente. Comme pour le moment, la rédaction du protocole pour cette interaction n'a pas encore été développée, une discussion plus poussée sort du domaine d'application du présent document.

4.2.3 Différence entre roulements de ZSK et de KSK

Noter que les roulements de KSK et de ZSK sont différents en ce sens qu'un roulement de KSK exige une interaction avec la parente (et éventuellement le remplacement des ancres de confiance) et le délai qui s'ensuit pour l'attendre.

Un roulement de clé de zone peut être traité de deux façons différentes : la pré publication (paragraphe 4.2.1.1) et la double signature (paragraphe 4.2.1.2).

Comme la KSK est utilisée pour valider l'ensemble de clés et parce que la KSK n'est pas changée durant un roulement de ZSK, une antémémoire est capable de valider le nouvel ensemble de clés de la zone. La méthode pré publiée fonctionnerait aussi pour un roulement de KSK. Les enregistrements qui sont à pré publier sont les RR DS parents. La méthode pré publiée a quelques inconvénients pour les KSK. On décrit d'abord le schéma de roulement et on indique ensuite ces inconvénients.

initial	**nouveau DS**	**nouveau DNSKEY**	**suppression de DS/DNSKEY**
Parente :			
SOA0	SOA1	---------->	SOA2
RRSIGpar(SOA0)	RRSIGpar(SOA1)	---------->	RRSIGpar(SOA2)
DS1	DS1	---------->	DS2
	DS2	---------->	
RRSIGpar(DS)	RRSIGpar(DS)	---------->	RRSIGpar(DS)
Fille :			
SOA0	---------->	SOA1	SOA1
RRSIG10(SOA0)	---------->	RRSIG10(SOA1)	RRSIG10(SOA1)
	---------->		
DNSKEY1	---------->	DNSKEY2	DNSKEY2
	---------->		
DNSKEY10	---------->	DNSKEY10	DNSKEY10
RRSIG1 (DNSKEY)	---------->	RRSIG2(DNSKEY)	RRSIG2 (DNSKEY)
RRSIG10(DNSKEY)	---------->	RRSIG10(DNSKEY)	RRSIG10(DNSKEY)

Stades de déploiement pour un roulement pré publié de clé de signature de clés

Quand la zone fille veut un roulement, elle le notifie à la zone parente durant la phase "nouveau DS" et soumet la nouvelle clé (ou le DS correspondant) à la parente. La parente publie DS1 et DS2, pointant respectivement sur DNSKEY1 et DNSKEY2. Durant le roulement (phase "nouveau DNSKEY") qui peut avoir lieu aussitôt que le nouvel ensemble de DS est propagé à travers le DNS, la fille remplace DNSKEY1 par DNSKEY2. Immédiatement après cela (phase de "suppression de DS/DNSKEY") elle peut notifier à la parente que le vieil enregistrement DS peut être supprimé.

Les inconvénients de ce schéma sont que durant la phase "nouveau DS", la parente ne peut pas vérifier la correspondance entre le RR DS2 et DNSKEY2 en utilisant le DNS -- car DNSKEY2 n'est pas encore publié. À côté, on introduit une clé "à faible sécurité" (voir le paragraphe 4.4.3). Finalement, l'interaction fille-parente consiste en deux étapes. La méthode de la "double signature" a seulement besoin d'une seule interaction.

4.2.4. Roulements de clé automatisés

Comme les clés doivent être renouvelées périodiquement, il y a des motivations pour automatiser le processus de roulement. Considérons ce qui suit :

- Les roulements de ZSK sont faciles à automatiser car seule la zone fille est impliquée.
- Un roulement de KSK a besoin d'une interaction entre parente et fille. L'échange de données est nécessaire pour fournir les nouvelles clés à la parente ; par conséquent, ces données doivent être authentifiées et leur intégrité doit être garantie afin d'éviter des attaques sur le roulement.

4.3 Prévoir un roulement de clé d'urgence

Ce paragraphe traite de la préparation à une possible compromission de clé. Notre avis est d'avoir une procédure documentée prête pour quand une clé compromise est suspectée ou confirmée.

Quand le matériel privé d'une clé est compromis, il peut être utilisé tant qu'existe une chaîne de confiance valide. Une chaîne de confiance reste intacte

- tant qu'une signature sur la clé compromise dans la chaîne de confiance est valide,
- tant qu'un RR DS parental (et sa signature) pointe sur la clé compromise,

o tant que la clé est ancrée dans un résolveur et est utilisée comme point de départ de validation (ceci est généralement le plus dur à mettre à jour).

Lorsque il existe une chaîne de confiance à la clé compromise, l'espace de noms est vulnérable à un abus par quiconque a obtenu une possession illégitime de la clé. Les opérateurs de zone doivent évaluer si l'abus de la clé compromise est pire que d'avoir des données dans les antémémoires qui ne peuvent pas être validées. Si l'opérateur de zone choisit de casser la chaîne de confiance pour la clé compromise, les données dans les antémémoires signées avec cette clé ne peuvent plus être validées. Cependant, si l'administrateur de zone choisit de prendre le chemin d'un roulement régulier, le détenteur malveillant de la clé peut falsifier des données pour qu'elles apparaissent comme valides.

4.3.1 KSK compromise

Une zone contenant un RRSet DNSKEY avec une KSK compromise est vulnérable tant que la KSK compromise est configurée comme ancre de confiance ou qu'un DS parental pointe sur elle.

Une KSK compromise peut être utilisée pour signer l'ensemble de clés d'une zone d'un attaquant. Cette zone pourrait être utilisée pour empoisonner le DNS.

Donc, quand la KSK a été compromise, l'ancre de confiance ou le DS parental devrait être remplacé aussitôt que possible. Il relève de la politique locale de décider de rompre la chaîne de confiance durant le roulement d'urgence. La chaîne de confiance va être rompue quand la KSK compromise est retirée de la zone fille alors que la parente a encore un DS qui pointe sur la KSK compromise (l'hypothèse est qu'il y a seulement un DS chez la parente. Si il y a plusieurs DS, ceci ne s'applique pas – cependant, la chaîne de confiance de cette clé particulière est rompue).

Noter que la zone d'un attaquant va encore utiliser la KSK compromise et la présence d'un DS parental va faire apparaître les données dans cette zone comme valides. Retirer la clé compromise va faire apparaître la zone de l'attaquant comme valide et la zone fille comme boguée. Donc, on conseille de ne pas supprimer la KSK avant que la parente ait un DS pour une nouvelle KSK en place.

4.3.1.1 Garder intacte la chaîne de confiance

Si on suit ce conseil, le temps de remplacement de la KSK est assez critique. Le but est de supprimer la KSK compromise aussitôt que le nouvel RR DS est disponible chez la parente. Et aussi de s'assurer que la signature faite avec une nouvelle KSK sur l'ensemble de clés avec la KSK compromise dans elle expire juste après que le nouveau DS apparaît chez la parente, supprimant donc les vieux déchets en un seul passage.

La procédure est la suivante :
1. Introduire une nouvelle KSK dans l'ensemble de clés, garder la KSK compromise dans l'ensemble de clés.
2. Signer l'ensemble de clés, avec une courte période de validité. La période de validité devrait expirer peu après le moment où l'apparition du DS est attendue dans la parente et où les vieux DS ont expiré dans les antémémoires.
3. Charger le DS pour cette nouvelle clé chez la parente.
4. Suivre la procédure de roulement régulier de KSK : attendre que le DS apparaisse dans les serveurs d'autorité et ensuite attendre le temps du TTL du vieux RR DS. Si nécessaire, resigner le RRSet DNSKEY et modifier/étendre le temps d'expiration.
5. Supprimer le RR DNSKEY compromis de la zone et resigner l'ensemble de clés en utilisant l'intervalle "normal" de validité.

Un danger supplémentaire d'une clé compromise est que la clé compromise pourrait être utilisée pour faciliter un roulement légitime de DNSKEY/DS et/ou des changements de serveur de noms chez la parente. Quand cela arrive, le domaine peut être en conflit. Un mécanisme de notification authentifié hors bande et sûr pour contacter une zone parente est nécessaire dans ce cas.

Noter que ceci est seulement un problème quand le DNSKEY et ou les enregistrements DS sont utilisés pour l'authentification chez la zone parente.

4.3.1.2 Rupture de la chaîne de confiance

Il y a deux méthodes pour rompre la chaîne de confiance. La première méthode cause l'apparition de la zone fille comme 'boguée' aux résolveurs valideurs. L'autre cause l'apparition de la zone fille comme 'non sûre'. Les deux sont décrites ci-

dessous.

Dans la méthode qui cause l'apparition de la zone fille comme 'boguée' aux résolveurs valideurs, la zone fille remplace la KSK actuelle par une nouvelle et elle resigne l'ensemble de clés. Ensuite elle envoie le DS de la nouvelle clé à la zone parente. C'est seulement après que la parente a placé le nouveau DS dans la zone que la chaîne de confiance de la fille est réparée.

Une autre méthode pour casser la chaîne de confiance est de retirer le RR DS de la zone parente en même temps. Par suite, la zone fille devient non sûre.

4.3.2 ZSK compromise

Principalement parce que il n'y a pas d'interaction parentale requise quand une ZSK est compromise, la situation est moins grave qu'avec une KSK compromise. La zone doit quand même être resignée avec une nouvelle ZSK aussitôt que possible. Comme c'est une opération locale qui n'exige pas de communication entre la parente et la fille, ceci peut être réalisé très rapidement. Cependant, on doit tenir compte que tout comme avec un roulement normal, la disparition immédiate de la vieille clé compromise peut conduire à des problèmes de vérification. On notera aussi que tant que le RRSIG sur la ZSK compromise n'est pas arrivé à expiration, il peut encore y avoir un risque pour la zone.

4.3.3 Compromission des clés ancrées dans les résolveurs

Une clé peut aussi être pré-configurée dans les résolveurs. Par exemple, si DNSSEC est déployé avec succès, la clé racine peut être pré-configurée dans les résolveurs les plus sûrs.

Si des clés d'ancre de confiance sont compromises, les résolveurs qui utilisent ces clés devrait être notifiés de ce fait. Les administrateurs de zone peuvent envisager d'établir une liste de diffusion pour communiquer le fait qu'une clé de SEP est sur le point d'être substituée. Cette communication va bien sûr devoir être authentifiée, par exemple, en utilisant des signatures numériques.

Les utilisateurs finaux en présence de la tâche de mettre à jour une clé ancrée devraient toujours valider la nouvelle clé. Les nouvelles clés devraient être authentifiées hors bande, par exemple, par l'utilisation d'une annonce d'un site de la Toile qui est sécurisé en utilisant des prises sécurisées (TLS) [RFC4366].

4.4. Politiques parentales

4.4.1 Échanges initiaux de clé et considérations de politique parentale

L'échange de clés initial est toujours soumis aux politiques établies par la zone parente. Quand on conçoit une politique d'échange de clés on devrait prendre en compte que les mécanismes d'authentification et d'autorisation utilisés durant un échange de clés devraient être aussi forts que les mécanismes d'authentification et d'autorisation utilisés pour l'échange des informations de délégation entre parente et fille. C'est-à-dire, il n'y a pas de besoin implicite dans DNSSEC de rendre le processus d'authentification plus fort qu'il n'est dans le DNS.

Utiliser le DNS lui-même comme source du matériel DNSKEY réel, avec une vérification hors bande sur la validité de la DNSKEY, a l'avantage de réduire les chances d'une erreur de l'utilisateur. Un outil d'interrogation de DNSKEY peut utiliser le bit SEP [RFC3757] pour choisir la clé appropriée à partir d'un ensemble de clés DNSSEC, réduisant par là les chances que soit envoyé la mauvaise DNSKEY. Il peut valider l'auto signature sur une clé ; vérifiant par là la propriété du matériel de clé privée. Aller chercher la DNSKEY auprès du DNS assure que la chaîne de confiance reste intacte une fois que la zone parente publie le RR DS indiquant que la zone fille est sûre.

Note : la vérification hors bande est encore nécessaire quand le matériel de clé est récupéré via le DNS. La zone parente ne peut jamais être sûre que les RR DNSKEY n'ont pas été falsifiés.

4.4.2 Mémoriser les clés ou les hachages

Quand on conçoit un système de registre, on devrait considérer qui mémoriser, des DNSKEY et/ou des DS correspondants. Comme une zone fille pourrait souhaiter avoir un DS publié en utilisant un algorithme de résumé de messages pas encore compris par le registre, le registre ne peut pas compter être capable de générer l'enregistrement DS à partir d'un DNSKEY

brut. Donc, on recommande que les systèmes de registre prennent au moins en charge la mémorisation des enregistrements de DS.

Il peut aussi être utile de mémoriser les DNSKEY, car les avoir peut aider durant les recherches de problèmes et, pour autant que le résumé de message choisi par la zone fille soit accepté, la surcharge de génération des enregistrements de DS à partir d'eux est minimale. Avoir un mécanisme hors bande, comme un répertoire de registres (par exemple, Whois) pour découvrir quelles clés sont utilisées pour générer les enregistrements de ressource DS pour des propriétaires et/ou zones spécifiques peut aussi aider dans la recherche de problèmes.

Les considérations de mémorisation se rapportent aussi à la conception de l'interface de consommateur et à la méthode par laquelle les données sont transférées entre l'enregistrant et le registre ; l'administrateur de la zone fille sera t-il capable de télécharger le RR DS avec un algorithme de hachage inconnu ou est-ce l'interface seule qui permet les DNSKEY ? Dans le modèle de registre-registraire, on peut utiliser les extensions de DNSSEC au protocole de provisionnement extensible (EPP, *Extensible Provisioning Protocol*) [RFC4310], qui permet le transfert de RR DS et facultativement des RR DNSKE.

4.4.3 Faiblesse de la sécurité

La faiblesse de la sécurité est définie comme ce qui se produit quand une zone parente a un RR DS pointant sur un RR DNSKEY non existant. Quand cela arrive, la zone fille peut être marquée "boguée" par les clients vérificateurs du DNS.

Au titre d'une vérification de délégation complète, la parente pourrait, au moment de l'échange de clés, vérifier que la clé de la fille est en fait configurée dans le DNS. Cependant, si une parente ne comprend pas l'algorithme de hachage utilisé par la fille, les vérifications parentales sont limitées à seulement comparer l'identifiant de la clé.

Les zones filles devraient faire très attention en supprimant le matériel de DNSKEY, en particulier les clés SEP, pour lesquelles il existe un RR DS.

Une fois qu'une zone est "à faible sécurité", un remède (par exemple, la suppression d'un RR DS) va prendre du temps à se propager à travers le DNS.

4.4.4 Période de validité de signature DS

Comme le DS peut être répété tant qu'il a une signature valide, une courte période de validité de signature sur le DS minimise le temps où une fille est vulnérable dans le cas d'une compromission de la ou des KSK de la fille. Une période de validité de signature qui est trop courte introduit la possibilité qu'une zone soit marquée "boguée" dans le cas d'une erreur de configuration chez le signataire. Il peut ne pas y avoir assez de temps pour réparer les problèmes avant l'expiration des signatures. Quelque chose d'aussi courant que l'indisponibilité de l'opérateur durant les fins de semaine montre la nécessité de périodes de validité de signature de DS de plus de deux jours. On recommande un minimum absolu de période de validité de signature de DS de quelques jours.

La période maximum de validité de signature de l'enregistrement DS dépend de pendant combien de temps les zones filles acceptent d'être vulnérables après une clé compromise. Par ailleurs, raccourcir l'intervalle de validité de signature de DS augmente le risque opérationnel pour la parente. Donc, la parente peut avoir une politique d'utilisation d'intervalle de validité de signature considérablement plus long que ce qu'espérerait la fille.

Un compromis entre les contraintes opérationnelles de la parente et la minimisation des dommages pour la fille peut résulter en une période de validité de signature de DS quelque part entre une semaine et quelques mois.

En plus de la période de validité de signature, qui fixe une limite inférieure au nombre de fois qu'un propriétaire de zone va devoir signer les données de zone et qui fixe une limite supérieure à la durée pendant laquelle une fille est vulnérable après une compromission de clé, il y a la valeur de TTL sur le RR DS. Raccourcir le TTL signifie que les serveurs d'autorité vont voir plus d'interrogations. Mais par ailleurs, un court TTL diminue la persistance des RRSet DS dans les antémémoires augmentant par là la vitesse à laquelle les RRSet DS mis à jour se propagent à travers le DNS.

5. Considérations sur la sécurité

DNSSEC ajoute à la protection de l'intégrité des données du DNS. Le présent document essayé de fixer les considérations de fonctionnement pour maintenir un service DNSSEC stable et sûr. Ne pas tenir compte des propriétés de "propagation

des données" dans le DNS causera des échecs de validation et peut rendre des zones sécurisées indisponibles pour les résolveurs soucieux de leur sécurité.

6. Remerciements

La plupart des idées dans ce document résultent d'efforts collectifs durant des ateliers, des discussions, et des simulations.

Au risque d'oublier des personnes qui ont été les contributeurs originaux des idées, nous tenons à remercier les gens qui ont été activement impliqués dans la compilation du présent document. En ordre aléatoire : Rip Loomis, Olafur Gudmundsson, Wesley Griffin, Michael Richardson, Scott Rose, Rick van Rein, Tim McGinnis, Gilles Guette Olivier Courtay, Sam Weiler, Jelte Jansen, Niall O'Reilly, Holger Zuleger, Ed Lewis, Hilarie Orman, Marcos Sanz, et Peter Koch.
Certains matériaux de ce document sont copiés de la [RFC2541].
Mike StJohns a conçu l'échange de clés entre parent et enfant mentionné au dernier alinéa du paragraphe 4.2.2
Le paragraphe 4.2.4 a été fourni par G. Guette et O. Courtay.
Emma Bretherick, Adrian Bedford, et Lindy Foster ont corrigé beaucoup de fautes d'orthographe et de style.
Kolkman et Gieben doivent être blâmés pour toutes les fautes qui restent.
Lorsqu'il travaillait sur ce document, Kolkman était employé par RIPE NCC et Gieben par NLnet Labs.

7. Références

7.1 Références normatives

[RFC1034] P. Mockapetris, "Noms de domaines - Concepts et facilités", STD 13, novembre 1987. (*MàJ par* RFC1101, 1183, 1348, 1876, 1982, 2065, 2181, 2308, 2535, 4033, 4034, 4035, 4343, 4035, 4592, 5936, 8020, 8482, 8767)

[RFC1035] P. Mockapetris, "Noms de domaines – Mise en œuvre et spécification", STD 13, novembre 1987. (*MàJ par* RFC1101, 1183, 1348, 1876, 1982, 1995, 1996, 2065, 2136, 2181, 2137, 2308, 2535, 2673, 2845, 3425, 3658, 4033, 4034, 4035, 4343, 5936, 5966, 6604, 7766, 8482, 8767)

[RFC3757] O. Kolkman, J. Schlyter, E. Lewis, "Fanion de point d'entrée sécurisé (SEP) d'enregistrement de ressource (RR) KEY du système de noms de domaines (DNSKEY)", avril 2004. (*Obsolète, voir* RFC4033, RFC4034, RFC4035) *(P.S.)*

[RFC4033] R. Arends, et autres, "Introduction et exigences pour la sécurité du DNS", mars 2005.

[RFC4034] R. Arends et autres, "Enregistrements de ressources pour les extensions de sécurité au DNS", mars 2005.

[RFC4035] R. Arends et autres, "Modifications du protocole pour les extensions de sécurité du DNS", mars 2005. *(P.S. ; MàJ par* RFC8198

7.2 Références pour information

[RFC1995] M. Ohta, "Transferts de zone par incréments dans le DNS", RFC 1995, août 1996.

[RFC1996] P. Vixie, "Mécanisme de notification rapide des changements de zone (DNS NOTIFY)", août 1996. *(P.S.)*

[RFC2119] S. Bradner, "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997. *(MàJ par* RFC8174)

[RFC2308] M. Andrews, "Mise en antémémoire négative des interrogations du DNS (DNS NCACHE)", mars 1998. (*MàJ par les RFC* 4033, 4034, 4035, 6604, 8020) *(P.S.)*

[RFC2541] D. Eastlake 3rd, "Considérations sur le fonctionnement de la sécurité du DNS", mars 1999. (*Obsolète, voir* RFC4641) *(Information)*

[RFC3007] B. Wellington, "Mise à jour dynamique sécurisée du système des noms de domaine (DNS)", novembre 2000.

[RFC3766] H. Orman, P. Hoffman, "Détermination de la force des clés publiques utilisées pour l'échange de clés symétriques", avril 2004. (BCP0086)

[RFC4086] D. Eastlake 3rd, J. Schiller, S. Crocker, "Exigences d'aléa pour la sécurité", juin 2005. (*Remplace* RFC1750) (BCP0106)

[RFC4310] S. Hollenbeck, "Transpositions d'extensions de sécurité du système de noms de domaines (DNS) dans le protocole d'approvisionnement extensible (EPP)", décembre 2005. *(P.S., Obsolète, voir la* RFC5910)

[RFC4366] S. Blake-Wilson et autres, "Extensions de sécurité de la couche Transport (TLS)", avril 2006. (*Obsolète,* RFC5246) *(P.S.)*

[RFC4509] W. Hardaker, "Utilisation de SHA-256 dans les enregistrements de ressource de signataire de délégation DNSSEC", mai 2006. *(P.S.)*

[RFC5702] J. Jansen, "Utilisation des algorithmes SHA-2 avec RSA dans les enregistrements de ressource DNSKEY et RRSIG pour DNSSEC", octobre 2009. *(P. S.)*

[Rose] Rose, S., "NIST DNSSEC workshop notes", juin 2001.

[Schneier] Schneier, B., "Applied Cryptography: Protocols, Algorithms, and Source Code in C", ISBN (hardcover) 0-471-12845-7, ISBN (paperback) 0-471-59756-2. Publié par John Wiley & Sons Inc., 1996.

[Sizes] Lenstra, A. and E. Verheul, "Selecting Cryptographic Key Sizes", The Journal of Cryptology 14 (255-293), 2001.

Appendice A. Terminologie

Dans le présent document, on utilise un "jargon" qui est défini dans d'autres documents. Dans la plupart des cas, on n'a pas copié le texte des documents qui définissent les termes mais on a donné une explication plus élaborée de sa signification. Noter que ces explications ne devraient pas être vues comme d'autorité.

Clé ancrée : une DNSKEY configurée dans les résolveurs tout autour du monde. Cette clé est difficile à mettre à jour, d'où le terme ancrée.

Bogué : voir aussi la Section 5 de la [RFC4033]. Un RRSet dans DNSSEC est marqué "bogué" quand une signature d'un RRSet ne valide pas contre une DNSKEY.

Clé de signature de clé (KSK, *Key Signing Key*) : c'est une clé qui est utilisée exclusivement pour signer l'ensemble de clés sommital. Le fait qu'une clé soit une KSK n'est pertinent que pour l'outil de signature.

Taille de clé : le terme 'taille de clé' peut être remplacé par 'taille de module' tout au long du document. Il est mathématiquement plus correct d'utiliser taille de module, mais ce document est destiné aux opérateurs qu'on estime plus à l'aise avec le terme taille de clé.

Clés privées et clés publiques : DNSSEC sécurise le DNS par l'utilisation de cryptographie à clé publique. La cryptographie à clé publique est fondée sur l'existence de deux clés (mathématiquement reliées) : une clé publique et une clé privée. Les clés publiques sont publiées dans le DNS par l'utilisation d'enregistrements de ressource DNSKEY (RR DNSKEY). Les clés privées devraient rester confidentielles.

Roulement de clé : un roulement de clé (aussi appelé substitution de clé dans certains environnements) est l'acte de remplacer une paire de clés par une autre à la fin d'une période d'efficacité d'une clé.

Clé de point d'entrée sécurisé (SEP, *Secure Entry Point*) : KSK qui a un enregistrement DS parental pointant sur elle ou qui est configurée comme ancre de confiance. Bien que non exigé par le protocole, on recommande que le fanion SEP [RFC3757] soit établi sur ces clés.

Auto signature : cela ne s'applique qu'aux signatures sur les DNSKEY ; une signature faite avec la DNSKEY x, sur la DNSKEY x est appelée une auto signature. Note : sans autres informations, les auto signatures ne portent aucune confiance. Elles sont utiles pour vérifier l'authenticité de la DNSKEY, c'est-à-dire, elles peuvent être utilisées comme un hachage.

Chanter le fichier de zone : terme utilisé pour l'événement de signature du fichier de zone par son administrateur tout en émettant un schéma mélodique.

Signataire : système qui a accès au matériel de clé privée et signe les enregistrements de ressource établis dans une zone. Un signataire peut être configuré à signer seulement des parties d'une zone, par exemple, seulement les RRSet pour lesquels les signatures existantes sont sur le point d'expirer.

Clé de signature de zone (ZSK, *Zone Signing Key*) : clé utilisée pour signer toutes les données dans une zone. Le fait qu'une clé soit une ZSK n'est pertinent que pour l'outil de signature.

Administrateur de zone : "rôle" responsable de signer une zone et de la publier sur le serveur d'autorité principal.

Appendice B. Comment effectuer le roulement de clé de signature de zone

Utiliser le schéma de signature pré-publiée et la méthode très prudente de s'assurer que les données ne subsistent pas dans les antémémoires, suivent ici le "comment faire".

Étape 0 : préparation : créer deux clés et les publier toutes deux dans l'ensemble de clés. Marquer une des clés "active" et l'autre "publiée". Utiliser la clé "active" pour signer les données de zone. Mémoriser la partie privée de la clé "publiée", de préférence hors ligne. Le protocole ne fournit pas d'attribut pour marquer une clé comme active ou publiée. C'est quelque chose qui doit être fait en dehors du protocole, par l'utilisation d'un ordinateur portable ou d'un outil de gestion de clé.

Étape 1 : déterminer l'expiration : au début du roulement, prendre note du temps d'expiration le plus élevé des signatures dans le fichier de zone créé avec la clé marquée active actuelle. Attendre jusqu'à ce que l'heure d'expiration marquée dans l'étape 1 soit passée.

Étape 2 : commencer alors d'utiliser la clé qui a été marquée "publiée" pour signer les données (c'est-à-dire, la marquer comme "active"). Arrêter d'utiliser la clé qui était marquée "active" ; la marquée comme "roulée".

Étape 3 : il est sain d'engager un nouveau roulement (Étape 1) après au moins une période de validité de signature.

Appendice C. Conventions typographiques

Les conventions typographiques suivantes sont utilisées dans ce document :

Notation de clé : une clé est noté par DNSKEYx, où x est un nombre ou un identifiant ; x pourrait être vu comme l'identifiant de la clé.

Notations de RRSet : les RR sont seulement noté par le type. Toutes les autres informations -- possesseur, classe, rdata, et TT L-- sont ignorées. Donc : "exemple.com 3600 IN A 192.0.2.1" est réduit à "A". Les RRSet sont une liste de RR. Un exemple en serait "A1, A2", spécifiant le RRSet qui contient deux enregistrements "A". Cela pourrait encore être abrégé à juste "A".

Notation de signature : les signatures sont notées par RRSIGx(RRSet), qui signifie que le RRSet est signé avec DNSKEYx.

Représentation de zone : en utilisant la notation ci-dessus on a simplifié la représentation d'une zone signée en laissant de côté tous les détails non nécessaires tels que les noms et en représentant toutes les données par "SOAx"

Représentation de SOA : les SOA sont représentés par SOAx, où x est le numéro de série.

En utilisant cette notation pour la zone signée suivante,

```
exemple.net.      86400  IN SOA  ns.exemple.net. bert.exemple.net. (
                           2006022100                    ; numéro de série
                           86400                         ; rafraîchissement (24 heures)
                           7200                          ; re essai   (2 heures)
                           3600000                       ; expire  (1000 heures)
                           28800 )                       ; minimum (8 heures)
                  86400  RRSIG   SOA 5 2 86400 20130522213204 (
                           20130422213204 14 exemple.net.
                           cmL62SI6iAX46xGNQAdQ... )
                  86400  NS      a.iana-serveurs.net.
                  86400  NS      b.iana-serveurs.net.
                  86400  RRSIG   NS 5 2 86400 20130507213204 (
                           20130407213204 14 exemple.net.
                           SO5epiJei19AjXoUpFnQ ... )
                  86400  DNSKEY  256 3 5 (
                           EtRB9MP5/AvOuVO0I8XDxy0... )          ; id = 14
                  86400  DNSKEY  257 3 5 (
                           gsPW/Yy19GzYIY+Gnr8HABU... )          ; id = 15
                  86400  RRSIG   DNSKEY 5 2 86400 20130522213204 (
                           20130422213204 14 exemple.net.
                           J4zCe8QX4tXVGjV4e1r9... )
                  86400  RRSIG   DNSKEY 5 2 86400 20130522213204 (
                           20130422213204 15 exemple.net.
                           keVDCOpsSeDReyV6O... )
                  86400  RRSIG   NSEC 5 2 86400 20130507213204 (
                           20130407213204 14 exemple.net.
                           obj3HEp1GjnmhRjX... )
  a.exemple.net.    86400  IN TXT  "Une étiquette"
                  86400  RRSIG   TXT 5 3 86400 20130507213204 (
                           20130407213204 14 exemple.net.
                           IkDMlRdYLmXH7QJnuF3v... )
                  86400  NSEC    b.exemple.com. TXT RRSIG NSEC
                  86400  RRSIG   NSEC 5 3 86400 20130507213204 (
                           20130407213204 14 exemple.net.
                           bZMjoZ3bHjnEz0nIsPMM... )
                  ...
```

est réduit à la représentation suivante :

```
SOA2006022100
RRSIG14(SOA2006022100)
DNSKEY14
DNSKEY15

RRSIG14(KEY)
RRSIG15(KEY)
```

Le reste des données de zone a la même signature que l'enregistrement SOA, c'est-à-dire, un RRSIG créé avec DNSKEY 14.

Adresse des auteurs

Olaf M. Kolkman
NLnet Labs
Kruislaan 419
Amsterdam 1098 VA
The Netherlands
mél : olaf@nlnetlabs.nl
URI : http://www.nlnetlabs.nl

R. (Miek) Gieben
mél : miek@miek.nl

Déclaration complète de droits de reproduction

Remerciement
Le financement de la fonction d'édition des RFC est fourni par l'activité de soutien administratif de l'IETF (IASA).