Le roulement automatique des clés de confiance dans les résolveurs DNSSEC

Gilles Guette, Bernard Cousin, David Fort

Le roulement automatique des clés de confiance dans les résolveurs DNSSEC

Gilles Guette, Bernard Cousin et David Fort

IRISA, Campus de Beaulieu, 35042 Rennes CEDEX, FRANCE
{gilles.guette, bernard.cousin, david.fort}@irisa.fr

Les extensions sécurisées du DNS (Domain Name System Security Extensions) sont basées sur la cryptographie à clé publique. Une zone DNS sécurisée possède au moins une paire de clés (publique/privée) utilisée pour générer des signatures numériques des données de la zone. Ceci permet de fournir deux services de sécurité essentiels : l'intégrité des données et l'authentification. Ces deux services protègent les transactions DNS.

Le processus de validation DNSSEC est basé sur l'établissement d'une chaîne de confiance liant les zones sécurisées. Pour construire cette chaîne, un résolveur doit disposer d'un point d'entrée sécurisé dans l'arbre DNS, c'est-à-dire posséder une clé publique d'une zone et lui faire confiance. Cette clé est présente dans le fichier de configuration du résolveur et est appelée clé de confiance.

Dans ce papier, nous étudions le problème de la mise à jour de ces clés de confiance afin de garder la cohérence avec les clés effectivement déployées sur les serveurs de noms. Pour résoudre ce problème, nous définissons un nouvel enregistrement de ressource permettant à un résolveur de mettre à jour dans tous les cas son ensemble de clés de confiance.

Mots-clés: DNSSEC, sécurité réseau, roulement de clés

1 Introduction

Le *Domain Name System* (DNS) [Moc87a, Moc87b] est une base de données hiérarchique et distribuée selon un modèle arborescent. Elle est le plus souvent utilisée pour effectuer les associations entre un nom de machine et l'adresse IP qui lui est associée et inversement. Le concept originel de DNS n'incluait aucun mécanisme de sécurité [Bel95], c'est pourquoi DNSSEC (*Domain Name System Security Extensions*) [Eas99, Gun03, AAL+05a, AAL+05c, AAL+05b] a vu le jour. DNSSEC est basé sur l'utilisation de la cryptographie à clé publique afin de sécuriser l'arbre DNS et de fournir deux services de sécurité : l'intégrité des données et l'authentification [LMMM00].

Chaque nœud de l'arbre DNS, appelé *zone*, possède au moins une paire de clés utilisée pour protéger les enregistrements DNS de la zone avec des signatures numériques. Pour valider des enregistrements DNSSEC, un résolveur construit une chaîne de confiance [Gie01] depuis un point d'entrée sécurisé (typiquement une zone de haut niveau) jusqu'à la ressource demandée. Un point d'entrée sécurisé est une clé publique de zone. Il est configuré statiquement par l'administrateur du résolveur. Cette clé est appelée *clé de confiance*. Un résolveur ne peut valider des enregistrements que s'il est capable de construire une chaîne de confiance en partant d'un point d'entrée sécurisé et en suivant les délégations sécurisées jusqu'à la ressource demandée.

Néanmoins, les clés de zone ont une durée de vie limitée. Pour des raisons de sécurité évidentes (risque de cryptanalyse, *etc*.) les clés sont renouvelées périodiquement sur les serveurs de noms. Ces roulements de clés [GC03] de zone sur les serveurs de noms peuvent amener des incohérences entre les clés de confiance configurées dans les résolveurs et les clés de zone effectivement déployées.

Dans la section 2 nous présentons les notations utilisées dans ce papier ainsi que le processus de validation des données DNSSEC. Puis dans la section 3 nous décrivons le problème du roulement des clés de confiance et finalement dans la section 4 nous définissons un nouvel enregistrement de ressource permettant de résoudre le problème du roulement des clés de confiance.

2 Le processus de validation dans DNSSEC

2.1 Définitions et notations

Dans cette partie sont présentées les définitions et notations utilisées dans la suite de ce papier.

– Un domaine DNS X est le sous arbre complet de l'arbre DNS dont la racine est le nœud X.
– Une zone DNS est un nœud de l'arbre DNS. Le nom d'une zone est obtenu par la concaténation des étiquettes de chaque nœud présent sur le chemin de cette zone à la racine de l'arbre.
– Une zone peut déléguer la responsabilité d'une partie des noms qu'elle contient. Par exemple, la zone `com.` peut déléguer tous les noms terminés par `exemple.com.`. Pour cela, une nouvelle zone est créée (un nouveau nœud dans l'arbre). Cette nouvelle zone est alors responsable de tous les noms se terminant par `exemple.com.` (*cf* fig 1).

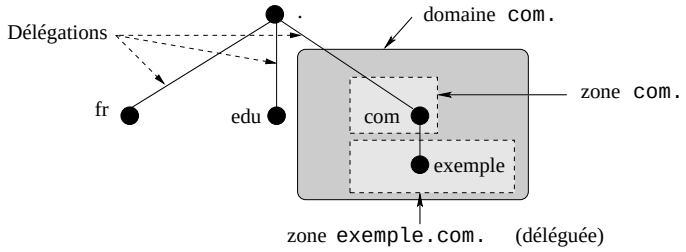

FIG. 1: Domaines et zones DNS.

Chaque zone possède un fichier de zone dans lequel se trouvent les informations dont elle est responsable. Ces informations sont des enregistrements de ressource (RR). Il s'agit de l'unité de base des données du DNS. Les enregistrements possédant le même nom, le même type et la même classe forment un RRset. Par exemple les DNSKEY RR d'une zone forment un DNSKEY RRset. Ils sont signés en groupe ce qui permet de limiter le nombre de signatures et donc la taille du fichier de zone.

Dans ce papier on notera DNSKEY(*key*1) l'enregistrement qui contient la clé nommée *key1* et RRSIG$(X)_y$ la signature de l'enregistrement X générée avec la partie privée de la clé y.

2.2 Les entités présentes dans le DNS

Trois entités possédant des rôles distincts sont présentes dans l'architecture DNS, il s'agit : des serveurs de noms, des résolveurs et des serveurs caches (voir [Moc87a, AL02]).

Le serveur de noms. Le serveur de nom fait autorité sur une zone DNS dont il possède le fichier de zone. Le serveur de noms reçoit des requêtes DNS à propos d'un nom et répond au mieux avec les enregistrements dont il dispose dans son fichier de zone.

Le résolveur. Le résolveur est l'entité cliente. Elle reçoit les demandes d'une application et envoie les requêtes DNS au serveur de noms approprié ou à un serveur cache. Une fois la résolution de nom effectuée, le résolveur retourne la réponse à l'application.

Le serveur cache. Le serveur cache ne fait autorité sur aucune zone. La résistance au facteur d'échelle du DNS est basée sur l'utilisation des serveurs caches. Ces serveurs reçoivent des requêtes des résolveurs et les font suivre aux serveurs de noms. Les serveurs caches conservent en mémoire les réponses reçues.

2.3 La chaîne de confiance DNSSEC

Les extensions de sécurité de DNS définissent de nouveaux enregistrements de ressources afin de conserver les clés et les signatures numériques nécessaires.

Chaque zone sécurisée possède au moins une paire de clés. La partie publique est placée dans un enregistrement DNSKEY. La partie privée est conservée dans un endroit sûr (généralement non connecté au réseau). La partie privée sert à signer le fichier de zone, c'est-à-dire qu'une signature est générée pour chaque RRset (fig. 2). La signature d'un RR consiste en l'application d'une fonction de hachage sur le RR suivi du chiffrement du haché obtenu. Chaque signature ainsi créée est placée dans un enregistrement

RRSIG et est associée au RRset qu'elle référence. La vérification des signatures est présentée sur la figure 2.

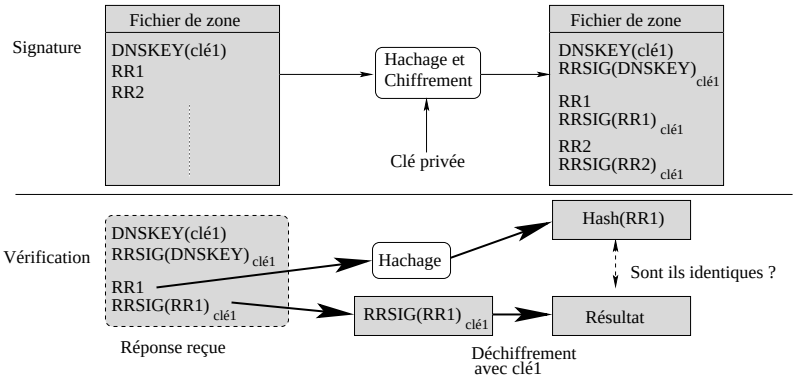

FIG. 2: Le processus de vérification d'une signature.

Lorsqu'un résolveur reçoit un RR et sa signature, il utilise la clé publique de la zone interrogée pour déchiffrer la signature et il compare si le résultat du déchiffrement est identique au haché du RR reçu. Si les deux sont identiques alors l'enregistrement reçu est valide.

Durant le processus de vérification d'une signature, la clé publique de la zone est nécessaire or elle est auto-signée. Pour faire confiance à cette clé lorsqu'elle n'est pas configurée comme point d'entrée sécurisé [KSL04], DNSSEC utilise la structure arborescente de son architecture pour établir une chaîne de confiance [Gie01]. Les maillons de cette chaîne sont représentés par la relation existant entre une zone mère et sa zone fille : il s'agit de l'enregistrement *Delegation Signer* (DS RR) [Gun03]. Cet enregistrement est signé et conservé dans la zone parente. Il contient les informations permettant d'identifier une clé de sa zone fille, comme le montre la figure 3.

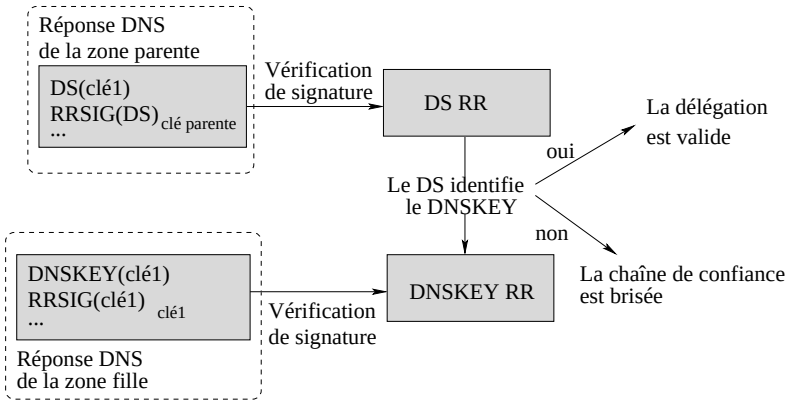

FIG. 3: Le processus de vérification d'une délégation.

Lors d'une résolution de noms sécurisée, le résolveur doit vérifier la signature de la ressource demandée ainsi que les signatures des clés utilisées et des enregistrements DS qui les identifient. Le processus s'arrête lorsque le résolveur reçoit une clé qui est une clé de confiance, il n'est alors plus nécessaire de trouver l'enregistrement DS qui l'identifie.

3 Le problème du roulement des clés de confiance

Si DNSSEC est totalement déployé sur l'arbre DNS, un seul point d'entrée sécurisé est nécessaire, il s'agit d'une clé publique de la zone racine. Mais à cause du déploiement progressif de DNSSEC, le modèle

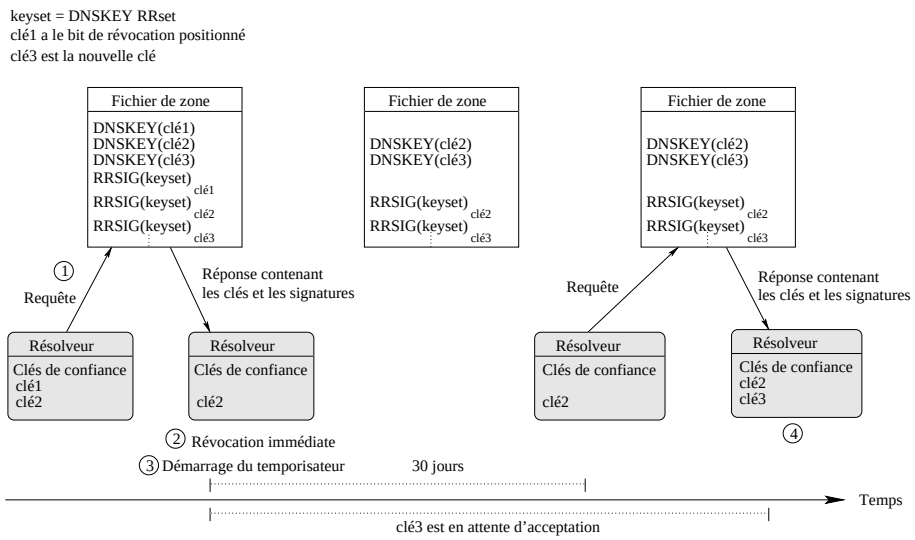

FIG. 4: La méthode de roulement des clés de confiance de M. St Johns.

qui semble émerger est celui d'îlots de sécurité où des zones sécurisées et non sécurisées se côtoient dans l'arbre DNS.

Un îlot de sécurité est un sous-arbre de l'arbre DNS entièrement sécurisé avec DNSSEC. Par conséquent, un résolveur a besoin d'au moins un point d'entrée sécurisé pour la racine de chaque îlot de sécurité afin de pouvoir effectuer des résolutions de noms sécurisées sur n'importe quelle zone de ces îlots.

De plus, pour maintenir un niveau de sécurité convenable, les clés de zone sont renouvelées régulièrement. Par conséquent, les clés de confiances configurées dans les résolveurs doivent être mise à jour afin de garder la cohérence entre les clés de confiance que possède un résolveur et les clés réellement déployées dans les serveurs de noms DNSSEC.

Actuellement, le roulement des clés de confiance dans les résolveurs est effectué manuellement par les administrateurs. Ce qui implique des risques de mauvaise configuration ou d'interruption de service entre le moment où une clé est renouvelée sur le serveur de noms et le moment où l'administrateur d'un résolveur met à jour cette clé.

À ce jour il n'existe pas de méthode standardisée pour effectuer le roulement automatique des clés de confiance dans les résolveurs. Lorsqu'une zone renouvelle une de ses clés, il n'existe pas de moyen automatique de prévenir les résolveurs que cette clé est en cours de changement. Lorsqu'une clé est supprimée de son fichier de zone, les résolutions de noms utilisant cette clé comme point d'entrée sécurisé échoue.

Sans un mécanisme automatique pour prévenir les résolveurs des renouvellements de clés dans les fichiers de zone des serveurs de noms, les résolutions de noms peuvent échouer à tout moment même s'il existe une chaîne de confiance de la racine de l'arbre jusqu'à la zone interrogée. Néanmoins, des propositions commencent à apparaître [StJ04] [IKM04]. Ces deux méthodes font actuellement l'objet de discussions et sont présentées dans les deux sections suivantes, elles souffrent néanmoins toutes les deux du même défaut : il existe des cas dans lesquels ces solutions sont inefficaces. Nous exhibons de tels cas après la présentation de chaque méthode.

3.1 Présentation de la méthode de M. St Johns utilisant un bit de révocation

M. St Johns a publié une méthode de roulement automatique des clés de confiance [StJ04]. Ce document définit une méthode de révocation de clés utilisant un nouveau bit dans le champs *Flags* d'un enregistrement DNSKEY. Lorsque ce bit est positionné, cela signifie que la clé contenue dans l'enregistrement DNSKEY doit être supprimée de l'ensemble des clés de confiance du résolveur. Cette révocation est immédiate. La figure 4 montre le principe de cette méthode.

Tout d'abord, un résolveur envoie une requête DNS ① et reçoit un DNSKEY RRset contenant un enregis-

trement DNSKEY dont le bit de révocation est positionné. Le résolveur vérifie les signatures du DNSKEY RRset. Si le résultat de la vérification est positive alors la révocation est immédiate. Le résolveur met à jour son ensemble de clés de confiance en retirant la clé concernée ②.

Une fois la suppression de la clé effectuée, Mike St Johns introduit un temporisateur. Ce temporisateur a pour but de fortement diminuer le risque d'attaques dues à des clés compromises. En effet, si dans cette méthode l'acceptation était aussi immédiate, une personne malveillante pourrait utiliser une clé compromise pour signer de fausses clés et les introduire dans l'ensemble des clés de confiance du résolveur.

Après révocation, le résolveur démarre donc un temporisateur de trente jours. Durant ces trente jours, si le résolveur interroge la zone, il garde une trace de toutes les nouvelles clés présentent dans le DNSKEY RRset de cette zone et de toutes les clés qui signent le DNSKEY RRset de cette zone ③. Durant ces trente jours, la nouvelle clé (clé3) est en attente d'acceptation et elle n'est pas utilisée.

Si le résolveur reçoit un DNSKEY RRset de la zone correctement signé, avant l'expiration du temporisateur, mais que la nouvelle clé en attente ne s'y trouve pas, alors le résolveur considère cette information comme une preuve de problèmes. En effet, cela peut être dû à une mauvaise configuration du serveur de noms ou alors à une attaque par clé compromise. L'un des DNSKEY RRset a pu être envoyé par un attaquant ayant compromis une clé de la zone et qui essaie alors de placer de fausses clés dans l'ensemble des clés de confiance du résolveur.

Ayant une preuve que quelque chose se passe mal, le résolveur stoppe alors l'acceptation des nouvelles clés et remet le temporisateur à zéro. Pour qu'une attaque par compromission de clé réussisse, il faut que l'attaquant soit capable d'écouter le trafic de sa victime pendant trente jours et de générer de fausses réponses à chaque fois que la victime reçoit un DNSKEY RRset de la zone compromise.

De la même manière, si toutes les clés qui signaient le DNSKEY RRset sont révoquées avant l'expiration du temporisateur, cela pose un problème car la vérification du DNSKEY RRset sera impossible après les trente jours d'attente. Le résolveur stoppe donc l'acceptation des nouvelles clés.

Une fois le temporisateur expiré, lorsque le résolveur reçoit une nouvelle fois le DNSKEY RRset, il vérifie les signatures. Si le DNSKEY RRset vérifie les conditions présentées précédemment : il contient les nouvelles clés, au moins une signature valide, *etc*. Il accepte enfin les nouvelles clés comme clés de confiance pour cette zone ④ (ici clé3).

Le problème majeur de cette solution est qu'elle introduit un délai de 30 jours et requiert quelques précautions dans la gestion des serveurs de noms et dans la fréquence de roulement des clés dans une zone. Cette méthode implique certaines contraintes sur le nombre de clés dont doit disposer une zone, le nombre de clés de confiance et la fréquence de roulement à cause du temporisateur de 30 jours.

En effet, avec cette méthode, lorsqu'un serveur crée une nouvelle clé, celle-ci est *en attente* et ne peut être utilisée pendant 30 jours après sa création. Cela implique que si une zone possède n clés et une période de roulement supérieure à $\frac{30}{n-1}$ jours, alors aucun résolveur ne sera capable de mettre à jour son ensemble de clés de confiance pour cette zone, avec cette méthode.

3.2 Présentation de l'algorithme $M-N$ n'utilisant pas de bit de révocation

Le document [IKM04] présente un algorithme pour le roulement automatique des clés de confiance dans les résolveurs basé sur deux paramètres appelés critère M et critère N, d'où le nom de l'algorithme : l'algorithme $M-N$. Ces paramètres permettent de moduler le niveau de sécurité désiré.

Le critère M représente le nombre minimal de clés de confiance d'un résolveur, devant valider une signature du DNSKEY RRset reçu. Pour un résolveur R donné, le DNSKEY RRset reçu doit donc posséder M signatures valides qui ont été générées par la partie privée de M clés de confiance de R.

Le critère N représente le seuil de différences entre l'ensemble des clés contenues dans le DNSKEY RRset reçu possédant le bit SEP [KSL04] positionné et l'ensemble des clés de confiance configurées dans le résolveur pour cette zone. Le bit SEP indique une clé pouvant être utilisée comme point d'entrée sécurisé (*Secure Entry Point*).

L'algorithme $M-N$ fonctionne de la manière suivante :

1. L'administrateur du résolveur a configuré X clés de confiance pour une zone Z,
2. l'administrateur du résolveur a spécifié les valeurs des paramètres M et N.

3. Si au moins *M* clés de confiance valident les signatures du DNSKEY RRset reçu et
4. si le nombre d'éléments différents entre l'ensemble des clés du DNSKEY RRset possédant le bit SEP positionné et l'ensemble des clés de confiance du résolveur pour cette zone, est plus petit ou égal à *N*,
5. alors toutes les clés de confiance du résolveur pour cette zone sont remplacées par les clés du DNSKEY RRset reçu ayant le bit SEP positionné.

3.3 Présentation des contraintes et limitations des méthodes précédentes

Les deux méthodes décrites précédemment ne fonctionnent pas dans tous les cas. Dans la première méthode, le résolveur doit attendre 30 jours pour faire confiance à une nouvelle clé. Cette contrainte temporelle forte à peu de chances d'être respectée au niveau des politiques locales des serveurs de noms. Si une zone renouvelle toutes ses clés sur une durée de 30 jours, les résolveurs ne peuvent pas mettre à jour leur ensemble de clés de confiance pour cette zone. De plus, les requêtes d'un résolveur sur une zone peuvent être espacées de plus de 30 jours.

La seconde méthode ne fournit pas de contrainte temporelle et n'impose pas non plus une interrogation systématique des zones pour lesquelles des clés de confiance sont configurées. Or, il peut se passer un certain temps entre deux interrogations d'un résolveur sur une zone. Durant cette intervalle la zone concernée peut avoir renouvelé toutes ses clés, le résolveur est alors incapable de mettre à jour son ensemble de clés de confiance.

Il apparaît qu'en ne donnant aucune contrainte sur la période d'interrogation des zones par le résolveur des cas critiques existent et empêche un roulement des clés de confiance totalement automatique. Le roulement des clés est une chose nécessaire à DNSSEC comme pour toutes les applications utilisant la cryptographie. Dans DNSSEC, ce roulement est effectué du côté serveur et ce sont les résolveurs qui doivent être tenus au courant de ces changements. Certaines informations doivent donc être intégrées au fichier de zone afin que les résolveurs puissent connaître la période d'interrogation à satisfaire pour mettre à jour leur ensemble de clés de confiance.

Pour cela, trois informations sont nécessaires aux résolveurs, il faut que chaque zone indique la fréquence à laquelle elle renouvelle ses clés, le nombre de clés qu'elle possède et le nombre maximum de clés qu'elle est susceptible de changer au cours d'un roulement.

Le dysfonctionnement des méthodes de mise à jour automatique des clés de confiance est dû à la configuration actuelle du DNS. Des informations sont partagées entre les serveurs de noms et les résolveurs ou les serveurs caches (les clés de confiance) mais il n'y a aucun mécanisme permettant la cohérence de ces informations dupliquées. Il est nécessaire de contrôler périodiquement que les informations dont dispose un résolveur sont toujours correctes. Si le résolveur n'interroge pas périodiquement les zones pour lesquelles il possède des clés de confiance, ces clés vont devenir caduques. Cela arrive si le résolveur reste inactif un certains temps, ou s'il a besoin d'interroger une zone avec des périodes assez longues, par exemple récupérer des fichiers sur une machine distante tous les deux mois. Ce type de comportement arrive fréquemment sur les ordinateurs personnels lorsque leurs propriétaires désirent effectuer des mises à jours du système par exemple.

Nous montrons dans la section suivante que ces trois informations permettent à un résolveur de connaître la période maximale d'interrogation qu'il ne doit pas dépasser s'il veut utiliser une mise à jour automatique de ses clés de confiances. Ces trois informations permettent d'éviter les cas présentés ci-dessus.

4 Le nouvel enregistrement KRI

Pour pouvoir informer les résolveurs des renouvellements de clés fait au niveau des zones, nous définissons un nouvel enregistrement de ressource appelé KRI (*Key Renewal Information*). Le format de l'enregistrement est défini dans la section 4.1.

4.1 Le format du KRI RR

L'enregistrement KRI (figure 5) contient les informations supplémentaires présentées dans la section précédente et dont les résolveurs ont besoin pour effectuer un roulement automatique des clés de confiances.

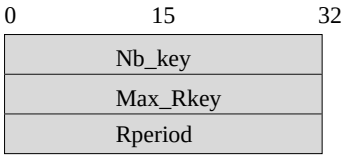

FIG. 5: L'enregistrement de ressource Key Renewal Information.

Le champs *Nb_key* contient le nombre de clés possédées par la zone, le champs *Max_Rkey* contient le nombre maximum de clés que la zone peut changer au cours d'un roulement. Une longueur de 32 bits pour ces champs est nécessaire car une zone peut posséder jusqu'à 2^{16} clés par algorithme cryptographique utilisable par DNSSEC (l'algorithme associé à une clé est spécifié dans le champs *algorithm* de l'enregistrement DNSKEY). Le champ *Rperiod* contient la période minimale de roulement des clés pour cette zone (en secondes). Ces informations sont suffisantes pour qu'un résolveur puisse calculer facilement la période à laquelle il doit interroger la zone pour ne pas tomber dans un cas où la mise à jour de ses clés de confiance devient impossible. La variable *Rperiod* est un entier de 32 bits ce qui permet de pouvoir coder une période de roulement des clés bien supérieure à un siècle, ce qui est largement suffisant. Le document [KG04] contient des conseils sur les durées à utiliser pour toutes opérations concernant DNSSEC (TTL, roulement de clés, *etc.*).

Dans la sous-section suivante, nous définissons la formule de calcul de la période d'interrogation d'une zone qu'un résolveur ne doit pas dépasser. Cette formule fonctionne quelle que soit la méthode de roulement des clés de confiance utilisée par le serveur de noms de la zone (avec ou sans bit de révocation). Cette première formule ne prend pas en compte la présence de serveur cache.

4.2 Calcul de la période d'interrogation sans serveur cache

Soit *Nb_key* le nombre de clés présentes dans une zone donnée, soit *Max_Rkey* le nombre maximal de clés que la zone peut changer au cours d'un roulement de clés et *Rperiod* la période de roulement minimale de cette zone. Soit R un résolveur, soit $S_{(R)}$ le nombre minimal de signatures valides d'un enregistrement DNSKEY pour que R accepte cette nouvelle clé comme clé de confiance et $Iperiod$ la période maximale d'interrogation de cette zone par R, pour que R puisse mettre à jour de manière automatique son ensemble de clés de confiance.

Les paramètres que nous venons de définir sont liés par un certains nombre de conditions :

Propriété 1. $\forall R, S_{(R)} \leq Nb\_key - Max\_Rkey$

Démonstration. Après un renouvellement de clés dans une zone, il reste au pire $Nb\_key - Max\_Rkey$ signatures générées par des *anciennes* clés et donc qu'un résolveur n'ayant pas eu connaissance de ce dernier roulement peut vérifier. Il faut que ce nombre de signatures soit supérieur ou égal au seuil de vérification $S_{(R)}$, pour que R puisse accepter une nouvelle clé de confiance. □

Corollaire 1. $Nb\_key \geq Max\_Rkey + 1$

Démonstration. D'après la propriété 1 et $S_{(R)} \geq 1$. □

Propriété 2. $S_{(R)} \in [1..Nb\_key - 1]$

Démonstration. Une zone qui renouvelle ses clés en renouvelle au moins une, le théorème 1 nous donne $S_{(R)} \leq Nb\_key - 1$. Le RFC 2535 [Eas99] spécifie que pour qu'un enregistrement soit valide il faut qu'au moins une des signatures associées soit valide ce qui donne $1 \leq S_{(R)}$. □

Nous pouvons maintenant donner la formule de calcul de la période d'interrogation à ne pas dépasser ($Iperiod$).

Propriété 3. $\forall R, Iperiod = \lfloor \frac{Nb\_key - S_{(R)}}{Max\_Rkey} \rfloor \times Rperiod$

Démonstration. Soit n le nombre maximal de roulement de clés tel que :
$Nb\_key - n \times Max\_Rkey \geq S_{(R)}$
$\Leftrightarrow -n \times Max\_Rkey \geq S_{(R)} - Nb\_key$
$\Leftrightarrow n \leq \frac{Nb\_key - S_{(R)}}{Max\_Rkey}$ comme $n \in \mathbb{N}$, $n = \lfloor \frac{Nb\_key - S_{(R)}}{Max\_Rkey} \rfloor$
Or $Iperiod = n \times Rperiod$, d'où $Iperiod = \lfloor \frac{Nb\_key - S_{(R)}}{Max\_Rkey} \rfloor \times Rperiod$. □

4.3 Calcul de la période d'interrogation prenant en compte les serveurs caches

Un autre paramètre ayant un impact sur la mise à jour automatique des clés de confiance est le temps (TTL) pendant lequel les enregistrements sont conservés dans les serveurs caches. Les serveurs caches sont un élément essentiel de la résistance au facteur d'échelle du service DNS. Néanmoins, pendant le temps TTL de présence en cache d'un enregistrement, il peut y avoir une incohérence entre des enregistrements présents sur un serveur de noms et sur un serveur cache. C'est pourquoi la présence de serveurs cache dans l'architecture DNS doit être prise en compte pour le calcul de la période d'interrogation d'une zone ($Iperiod$).

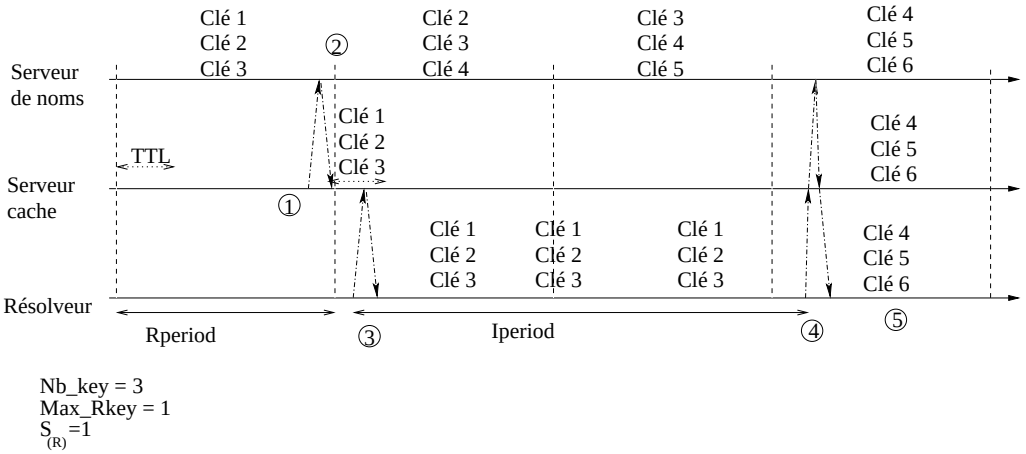

FIG. 6: Présence d'informations incohérentes et problème de mise à jour.

Soit le scénario suivant (figure 6):
- Un serveur cache interroge un serveur de noms et conserve la réponse reçue pendant TTL secondes ①.
- Juste après avoir envoyé la réponse, le serveur de noms renouvelle ses clés. Il y a une incohérence entre les informations du serveur cache et du serveur de noms ②.
- Un résolveur R interroge le serveur cache et reçoit les anciennes clés ③. Puis il attend au maximum $\lfloor \frac{Nb\_key - S_{(R)}}{Max\_Rkey} \rfloor \times Rperiod$ selon la propriété 3, avant sa prochaine interrogation de la zone.
- R interroge de nouveau le serveur cache, celui-ci ne possède pas les clés, il fait suivre la requête au serveur de noms et reçoit la réponse. Il fait suivre cette réponse à R ④.
- R ne peut mettre à jour ses clés de confiances, car |{Clé 1, Clé 2, Clé 3} ∩ {Clé 4, Clé 5, Clé 6}|=0 < $S_{(R)}$. À cause des anciennes clés gardées dans le serveur cache le résolveur a attendu une $Rperiod$ de trop ⑤.

Il faut donc s'assurer que lorsque le résolveur obtient par le serveur cache des informations de la $Rperiod$ précédente, il soit quand même en mesure de mettre à jour ses clés de confiance. La propriété suivante donne la formule de calcul de la période maximale d'interrogation d'une zone en prenant en compte la présence de serveurs cache. On suppose que $Rperiod \geq 2 \times TTL$

Propriété 4. $\forall R, Iperiod = \lfloor \frac{Nb\_key - S_{(R)}}{Max\_Rkey} \rfloor \times Rperiod - TTL$

Démonstration. Notons $n = \lfloor \frac{Nb\_key - S_{(R)}}{Max\_Rkey} \rfloor$. Soit t_1 l'instant où un résolveur R interroge une zone. On a $\exists a$, tel que $a \times Rperiod \leq t_1 < (a+1) \times Rperiod$. Soit $t_2 = t_1 + n \times Rperiod - TTL$. Montrons que t_2 est le dernier instant possible pour l'interrogation suivante.

Si $t_1 \in [a \times Rperiod..a \times Rperiod + TTL[$, les informations en cache au temps t_2 datent de $[(a+n-1) \times Rperiod..(a+n) \times Rperiod[$ car $[(a+n) \times Rperiod - TTL..(a+n) \times Rperiod[\subset [(a+n-1) \times Rperiod..(a+n) \times Rperiod[$:

- Si les informations en cache au temps t_1 datent de l'intervalle $[(a-1) \times Rperiod..a \times Rperiod[$ alors il y a eu au plus n roulements de clés entre les t_1 et t_2, R peut mettre à jour ses clés de confiance.
- Si les informations en cache au temps t_1 datent de l'intervalle $[a \times Rperiod..(a+1) \times Rperiod[$ alors il y a eu au plus $n-1$ roulements de clés entre les t_1 et t_2, R peut mettre à jour ses clés de confiance.

Si $t_1 \in [a \times Rperiod + TTL..(a+1) \times Rperiod[$, les informations en cache au temps t_1 datent de l'intervalle $[a \times Rperiod..(a+1) \times Rperiod[$:

- Si les informations en cache au temps t_2 datent de $[(a+n-1) \times Rperiod..(a+n) \times Rperiod[$ alors il y a eu au plus $n-1$ roulements de clés entre les t_1 et t_2, R peut mettre à jour ses clés de confiance.
- Si les informations en cache au temps t_2 datent de $[(a+n) \times Rperiod..(a+n+1) \times Rperiod[$ alors il y a eu au plus n roulements de clés entre les t_1 et t_2, R peut mettre à jour ses clés de confiance.

Donc $Iperiod = t_2 - t_1 \geq n \times Rperiod - TTL$.

Supposons $t_2 > t_1 + n \times Rperiod - TTL$, posons $t_1 = M$ où $M = \max[a \times Rperiod..a \times Rperiod + TTL[$, M existe car l'intervalle est fini (temps en seconde).

Alors $t_2 > M + n \times Rperiod - TTL \Leftrightarrow t_2 \geq a \times Rperiod + TTL + n \times Rperiod - TTL$

$\Leftrightarrow t_2 \geq (a+n) \times Rperiod$

$\Leftrightarrow t_2 \in [(a+n) \times Rperiod..(a+n+1) \times Rperiod[$

Si les données en cache au temps t_1 datent de l'intervalle $[(a-1) \times Rperiod..a \times Rperiod[$ alors il y a eu au plus $n+1$ roulements de clés entre les t_1 et t_2. Dans le cas où $n+1$ roulements ont eu lieu, R ne peut pas mettre à jour ses clés de confiance.

D'où, $Iperiod \leq n \times Rperiod - TTL$.

Donc, $Iperiod = n \times Rperiod - TTL$. □

La valeur du TTL d'une zone peut être obtenu très facilement. En effet, chaque enregistrement RRSIG contient, pour pouvoir vérifier cette signature, le TTL originel de la zone .

4.4 La gestion du KRI RR

L'enregistrement KRI est présent dans le fichier de zone, ainsi un résolveur ayant configuré des clés de confiance pour une zone donnée doit avoir pris connaissance de l'enregistrement KRI de cette zone. Avec les données contenues dans l'enregistrement KRI et les paramètres de sa politique locale de validation, le résolveur est en mesure de calculer la fréquence à laquelle il doit interroger la zone pour ne pas manquer un roulement de clés. Les informations présentes dans l'enregistrement KRI sont placées par l'administrateur du serveur de noms. Cet administrateur peut vouloir changer sa politique de roulement de clés et modifier les paramètres présents dans l'enregistrement KRI. Afin de ne pas perturber le fonctionnement des résolveurs, l'administrateur voulant changer les paramètres de sa politique de roulement de clés doit mettre à jour les paramètres de l'enregistrement KRI puis attendre un temps égal à $\lfloor \frac{Nb\_key - 1}{Max\_Rkey} \rfloor \times Rperiod - TTL$. Ceci permet à tous les résolveurs d'obtenir les nouveaux paramètres et de calculer la nouvelle fréquence d'interrogation (voir la propriété 4). Après ce délai de $\lfloor \frac{Nb\_key - 1}{Max\_Rkey} \rfloor \times Rperiod - TTL$, la nouvelle politique de roulement de clés du serveur de noms devient effective.

5 Conclusion

Le roulement de clé est nécessaire pour toutes les applications utilisant la cryptographie à clé publique. Néanmoins, le roulement de clés dans DNSSEC pose des problèmes de cohérence entre les clés présentes sur les serveurs de noms et les clés configurées statiquement comme clés de confiance dans les résolveurs. Il n'existe pas actuellement de méthode automatique de roulement de clés de confiance standardisée et implémentée, les changements sont fait manuellement par les administrateurs des résolveurs.

Des méthodes de roulement automatique sont en cours de définition afin de palier aux possibles erreurs de configuration lors des interventions manuelles des administrateurs. Ces méthodes souffrent du même défaut. Dans l'état actuel de DNSSEC, on ne peut pas avoir une solution qui fonctionne dans tous les cas car les résolveurs manquent d'informations sur le roulement pour être sûrs de pouvoir mettre à jour leur ensemble de clés de confiance.

Dans ce papier nous avons défini un nouvel enregistrement de ressource contenant les informations minimales nécessaires à un résolveur pour mettre à jour son ensemble de clés de confiance dans tous les cas. Nous avons aussi fourni la formule qui permet à un résolveur de calculer la période d'interrogation d'une zone donnée afin de pouvoir mettre à jour son ensemble de clés de confiance quelle que que soit la méthode de révocation utilisée. Cette formule prend en compte la présence de serveur cache dans l'architecture DNS et l'incohérence possible entre les informations présentes en cache et les informations déployées sur les serveurs de noms. Dans la dernière section, nous avons fourni la preuve que grâce à cette formule, un résolveur peut automatiquement mettre à jour ses clés de confiance.

Références

[AAL+05a] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. DNS Security Introduction and Requirements. RFC 4033, Mars 2005.

[AAL+05b] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. Protocol Modifications for the DNS Security Extensions. RFC 4035, Mars 2005.

[AAL+05c] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. Resource Records for the DNS Security Extensions. RFC 4034, Mars 2005.

[AL02] P. Albitz and C. Liu. *DNS and BIND*. O'Reilly & Associates, Inc., Sebastopol, CA., fourth edition, Janvier 2002.

[Bel95] S. M. Bellovin. Using the Domain Name System for System Break-Ins. Dans *Proceedings of the fifth Usenix UNIX Security Symposium*, pages 199–208, Salt Lake City, UT, Juin 1995.

[Eas99] D. Eastlake. Domain Name System Security Extensions. RFC 2535, Mars 1999.

[GC03] G. Guette and O. Courtay. KRO: A Key RollOver Algorithm for DNSSEC. Dans *International Conference on Information and Communication (ICICT'03)*, Novembre 2003.

[Gie01] R. Gieben. Chain of Trust. Master's Thesis, NLnet Labs, 2001.

[Gun03] O. Gundmundsson. Delegation Signer Resource Record. RFC 3658, Décembre 2003.

[IKM04] J. Ihren, O. Kolkman, and B. Manning. An In-Band Rollover Algorithm and a Out-Of-Band Priming Method for DNS Trust Anchors. Draft IETF, work in progress, Juillet 2004.

[KG04] O. Kolkman and R. Gieben. DNSSEC operational practices. Draft IETF, work in progress, Avril 2004.

[KSL04] O. Kolkman, J. Schlyter, and E. Lewis. Domain Name System KEY (DNSKEY) Resource Record (RR) Secure Entry Point (Septembre) Flag. RFC 3757, Avril 2004.

[LMMM00] A. Lioy, F. Maino, M. Marsian, and D. Mazzocchi. DNS Security. Dans *Terena Networking Conference*, Mai 2000.

[Moc87a] P. Mockapetris. Domain Names - Concept and Facilities. RFC 1034, Novembre 1987.

[Moc87b] P. Mockapetris. Domain Names - Implementation and Specification. RFC 1035, Novembre 1987.

[StJ04] M. StJohns. Automated Updates of DNSSEC Trust Anchors. Draft IETF, work in progress, Mars 2004.