RFC 4310 : Domain Name System (DNS) Security Extensions Mapping for the Extensible Provisioning Protocol (EPP)

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 25 novembre 2009

Date de publication du RFC : Décembre 2005

https://www.bortzmeyer.org/4310.html

Le protocole EPP d'avitaillement d'un registre (par exemple un registre de noms de domaine), normalisé dans le RFC 5730 [1], manipule des objets qui sont des instances d'une classe (nommée "*mapping*"). Par exemple, il existe une classe (un "*mapping*") pour les noms de domaine, décrite dans le RFC 5731. Notre RFC 4310 (remplacé depuis par le RFC 5910) décrit, lui, une extension EPP à ce "*mapping*" permettant de spécifier les données nécessaires à DNSSEC, notamment la clé publique d'une zone signée.

DNSSEC, normalisé dans le RFC 4033, utilise la même délégation que le DNS. La zone parente d'une zone signée délègue en indiquant la clé publique de sa zone fille. Plus exactement, la zone parente publie un condensat cryptographique de la clé publique de la zone fille, l'enregistrement **DS** (pour "*Delegation Signer*"), normalisé dans la section 5 du RFC 4034 (voir aussi le rappel en section 2.1 de notre RFC 4310).

Lorsqu'un bureau d'enregistrement crée un nom de domaine signé, ou bien informe le registre qu'un domaine est désormais signé, comment indique t-il ce DS ? Il y a plusieurs façons, et notre RFC propose d'utiliser EPP.

L'extension nécessaire est résumée en section 2. Elle fonctionne en ajoutant des éléments à la classe Domaine du RFC 5731. Un élément obligatoire est le condensat cryptographique de la clé (alors qu'il n'est pas forcément unique : cette erreur a été corrigée dans le RFC 5910). Mais le client EPP peut aussi fournir la clé elle-même (le RFC 6781 explique pourquoi le condensat, le futur DS, est obligatoire alors que la clé est facultative). Le RFC prévoit également que le registre de la zone parente peut également récupérer la clé dans le DNS (enregistrement DNSKEY) pour tester si le condensat reçu est correct (et

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc5730.txt

il est donc recommandé que ladite DNSKEY soit publiée **avant** de prévenir le parent par EPP). La clé transmise au registre doit être une clé de confiance, c'est-à-dire avoir le bit SEP à 1 (cf. RFC 3757). En terminologie moderne, cette clé doit être une KSK ("*Key Signing Key*").

Les commandes EPP pour gérer cette information font l'objet de la section 3. Ainsi, les réponses à `<info>` doivent désormais contenir un élément `<secDNS:infData>`, qui contient lui-même des éléments comme `<secDNS:dsData>` qui a son tour contient les champs qu'on trouve dans un enregistrement DS comme `<secDNS:keyTag>` (l'identificateur de la clé), `<secDNS:alg>` (l'algorithme utilisé), etc. L'espace de noms `urn:ietf:params:xml:ns:secDNS-1.0` (ici avec le préfixe `secDNS`) est enregistré dans le registre IANA `<https://www.iana.org/assignments/xml-registry/ns.html>` (voir section 6). (C'est désormais `secDNS-1.1`, depuis le RFC 5910.) Voici un exemple de réponse à `<info>` sur le domaine `example.com` :

```
<resData>
...
<domain:name>example.com</domain:name>
...
<extension>
  <secDNS:infData
   xmlns:secDNS="urn:ietf:params:xml:ns:secDNS-1.0"
   xsi:schemaLocation="urn:ietf:params:xml:ns:secDNS-1.0 secDNS-1.0.xsd">
    <secDNS:dsData>
      <secDNS:keyTag>12345</secDNS:keyTag>
      <secDNS:alg>3</secDNS:alg>
      <secDNS:digestType>1</secDNS:digestType>
      <secDNS:digest>49FD46E6C4B45C55D4AC</secDNS:digest>
    </secDNS:dsData>
  </secDNS:infData>
</extension>
```

Le condensat est de type SHA1 (`<digestType>1</digestType>`), la clé elle-même étant DSA/SHA1 (`<alg>3</alg>`).

L'extension DNSEC permet évidemment de créer un domaine signé, avec `<create>` (section 3.2.1) :

```
<domain:create>
  <domain:name>example.com</domain:name>
  ...
  <extension>
  <secDNS:create xmlns:secDNS="urn:ietf:params:xml:ns:secDNS-1.0"
    xsi:schemaLocation="urn:ietf:params:xml:ns:secDNS-1.0 secDNS-1.0.xsd">
  <secDNS:dsData>
    <secDNS:keyTag>12345</secDNS:keyTag>
    <secDNS:alg>3</secDNS:alg>
    <secDNS:digestType>1</secDNS:digestType>
    <secDNS:digest>49FD46E6C4B45C55D4AC</secDNS:digest>
    <!-- <secDNS:keyData>, la clé elle-même, est *facultatif* -->
  </secDNS:dsData>
</secDNS:create>
 ...
```

Une fois le domaine ainsi créé, le registre publiera typiquement un enregistrement DS comme :

```
example.com.  IN DS 12345 3 1 49FD46E6C4B45C55D4AC
```

Bien sûr, on peut aussi ajouter DNSSEC à un domaine existant, ou bien changer une clé existante. Cela se fait avec `<update>` :

```
<domain:update>
    <domain:name>example.com</domain:name>
...
<extension>
  <secDNS:update
   xmlns:secDNS="urn:ietf:params:xml:ns:secDNS-1.0"
   xsi:schemaLocation="urn:ietf:params:xml:ns:secDNS-1.0 secDNS-1.0.xsd">
    <secDNS:add>
      <secDNS:dsData>
        <secDNS:keyTag>12346</secDNS:keyTag>
        <secDNS:alg>3</secDNS:alg>
        <secDNS:digestType>1</secDNS:digestType>
        <secDNS:digest>38EC35D5B3A34B44C39B</secDNS:digest>
        <!-- <secDNS:keyData>, la clé elle-même, est *facultatif* -->
      </secDNS:dsData>
    </secDNS:add>
  </secDNS:update>
 ...
```

Et, en utilisant `<secDNS:rem>` au lieu de `<secDNS:add>`, on peut retirer une délégation sécurisée (≪ dé-signer ≫ le domaine).

Comme la grande majorité des extensions et "*mappings*" d'EPP, celle-ci est spécifiée en utilisant la syntaxe formelle des W3C schemas, ici en section 4.