

RFC 5702 : Use of SHA-2 algorithms with RSA in DNSKEY and RRSIG Resource Records for DNSSEC

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 28 octobre 2009. Dernière mise à jour le 28 juin 2010

Date de publication du RFC : Octobre 2009

<https://www.bortzmeyer.org/5702.html>

La cryptanalyse progresse sans cesse et, comme aiment le répéter les experts, « Les attaques ne deviennent jamais pires, elles ne font que s'améliorer ». Les menaces contre la fonction de hachage SHA-1 devenant de plus en plus précises, cet algorithme est petit à petit remplacé par la famille SHA-2. Ce RFC 5702¹ spécifie ce futur remplacement pour le protocole DNSSEC.

Normalisé dans le RFC 4033 et suivants, DNSSEC, qui vise à garantir l'authenticité des ressources DNS par le biais d'une signature numérique, n'est pas lié à un algorithme particulier. Aussi bien l'algorithme asymétrique utilisé pour les signatures que celui de hachage qui sert à créer un condensat des données à signer ne sont pas fixés en dur dans le protocole. SHA-1 avait été spécifié dans le RFC 3110. Un registre IANA <<https://www.iana.org/assignments/dns-sec-alg-numbers/dns-sec-alg-numbers.xhtml>> des algorithmes possibles existe et notre RFC 5702 marque l'entrée de la famille SHA-2 dans ce registre.

Il n'est pas évident de savoir si SHA-1 est vraiment vulnérable dès aujourd'hui. Plusieurs attaques ont été publiées <http://www.schneier.com/blog/archives/2005/02/cryptanalysis_o.html> dans la littérature ouverte (et il y en a donc sans doute bien d'autres, gardées secrètes) mais elles ne semblent pas facilement exploitables, surtout dans le cas de l'utilisation particulière qu'en fait DNSSEC. Néanmoins, le remplacement de SHA-1 était une nécessité, compte-tenu du fait que les attaques ne peuvent que s'améliorer (voir la section 8 pour un exposé plus détaillé), et parce qu'un tel remplacement prend du temps (par exemple, l'élaboration de ce RFC a été très longue).

Le remplaçant, SHA-2, décrit dans la norme FIPS 180-3 de 2008, n'est pas vraiment un algorithme unique mais une famille d'algorithmes. DNSSEC n'en retient que deux, SHA-256 (code 8) et SHA-512 (code 10). Voir la section 1 du RFC pour ces rappels.

1. Pour voir le RFC de numéro NNN, <https://www.ietf.org/rfc/rfcNNN.txt>, par exemple <https://www.ietf.org/rfc/rfc5702.txt>

SHA-2 est utilisable dans les enregistrements de type `DNSKEY` (clés cryptographiques distribuées par le domaine) et `RRSIG` (signature cryptographiques). Le RFC 4034 décrit tous ces types d'enregistrement.

La section 2 décrit les enregistrements `DNSKEY`. Comme la clé publique ne dépend pas de l'algorithme de hachage utilisé ultérieurement, on pourrait penser qu'ils ne sont pas modifiés. Mais il est important d'indiquer au résolveur cet algorithme de hachage car tous les résolveurs ne connaissent pas SHA-2 du jour au lendemain. Il faut donc qu'ils sachent tout de suite, en examinant la clé, s'ils pourront valider la zone ou pas. (Sinon, ils considéreront la zone comme étant non signée.) Les `DNSKEY` avec RSA + SHA-256 sont donc stockés avec le même format qu'avant, mais avec le numéro d'algorithme 8. Les `DNSKEY` avec RSA + SHA-512 auront, eux, le numéro d'algorithme 10.

Les signatures (enregistrements `RRSIG`) sont, elles, plus largement modifiées. La section 3 décrit ces modifications. Elle indique la représentation à suivre pour les `RRSIG`, suivant le standard PKCS #1 (RFC 3447).

Les signatures faites avec RSA + SHA-256 seront stockées avec le numéro d'algorithme 8 et leur codage aura un préfixe spécifique, pour les distinguer des signatures faites avec d'autres fonctions (section 3.1). Idem pour celles faites avec RSA + SHA-512 (section 3.2).

Et les problèmes pratiques du déploiement? Pas de panique, ils sont dans la section 4. La taille des clés elle-mêmes n'est pas couverte (section 4.1) mais défermée aux recommandations du NIST. La taille des signatures dépend de celle de la clé, pas de l'algorithme de hachage et ne changera donc pas (section 4.2).

Quels logiciels peuvent aujourd'hui utiliser ces algorithmes? Pour BIND, ce sera à partir de la version 9.7 (actuellement en bêta publique, mais sans SHA-2). Pour `ldns` <<http://www.nlnetlabs.nl/projects/ldns/>>, c'est déjà disponible dans la version actuelle, la 1.6.1 (mais tous les paquetages binaires de `ldns` n'ont pas forcément été compilés avec l'option `--enable-sha2`). La section 5 donne quelques détails pour les programmeurs, notamment sur la raison pour laquelle il n'y a plus de version spécifique à NSEC3 (section 5.2 et RFC 5155).

Voici un exemple d'utilisation du programme `ldns-keygen` (livré avec `ldns`) pour générer des clés utilisant SHA-256 pour la zone `vachement-secure.example` :

```
% ldns-keygen -a RSASHA256 vachement-secure.example
Kvachement-secure.example.+008+23094

% cat Kvachement-secure.example.+008+23094.key
vachement-secure.example.      3600      IN          DNSKEY      256 3 8 \
    AwEAAc87fkhQ3RehZ9AWUtataPhm6Ku+DLKgtUPp/Zi0mwhtDN36oWBhzUt5a82ZeatzsbC6WjIDWWqOx33cWh3ISMkDK0cOul1kM
    ;{id = 23094 (zsk), size = 1024b}
```

Et lorsqu'on la signe :

```
% ldns-signzone vachement-secure.example.zone Kvachement-secure.example.+008+23094

% cat vachement-secure.example.zone.signed
...
.      3600      IN          SOA          localhost. root.localhost. 1 604800 86400 2419200 604800
.      3600      IN          RRSIG        SOA 8 0 3600 \
    20091125150944 20091028150944 23094 . nvIldICQaBpHR/Fn264gL
    7b15RCzzM6h3S5o6k/yq9cFRhYpOO4FQGfozP2q8A7whoHfymCg1D6harOt7b0ffO+TA1iBddpF1DJLQ/DKPhfS2REqXbXjpveU3uQNK
    g0jOdj7G6hrSQ19wXg3Co5yL84jG33oS1DWQM9QhKUyAUk= \
    ;{id = 23094}
```

Comme il y a déjà eu des attaques de cryptanalyse contre SHA-1 publiées, la section 8, qui synthétise les questions de sécurité, recommande donc d'utiliser SHA-2 pour les futurs déploiement de DNSSEC.

Enfin, la section 8.2 explique pourquoi SHA-2 n'est pas vulnérable aux attaques par repli : si une zone a deux clés, utilisant SHA-1 et SHA-2, un méchant ne peut pas forcer l'utilisation de l'algorithme le plus faible car les enregistrements DNSSEC doivent être signés avec **toutes** les clés présentes.

Si on veut voir des signatures en vrai, la racine, et des TLD comme `.pm` sont signés (au 28 juin 2010) avec SHA-256, tandis que `.cat` est signé avec SHA-512. Sinon, on peut trouver des exemples de `DNSKEY` et de `RRSIG` utilisant SHA-2 dans la section 6 du RFC, ou bien les générer soi-même comme dans l'exemple ci-dessus.