DNSCurve, la sécurité pour le DNS ?

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 20 juin 2009. Dernière mise à jour le 31 août 2009

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

Tout le monde sait que le DNS n'est pas sûr. Si quelqu'un n'est pas au courant, il suffit de l'envoyer lire le RFC 3833 [1]. En 2008, la faille Kaminsky <https://www.bortzmeyer.org/comment-fonctionne-la-faille-html> avait encore remis le dossier de la sécurité DNS sur le dessus de la pile. Les risques étant de natures très variées, il n'y a certainement pas de solution miracle à tous les problèmes de sécurité du DNS. Mais comme l'analyse sérieuse des problèmes de sécurité est difficile, comme la « balle d'argent » est une histoire plus vendeuse que la longue liste des mesures à prendre, on voit souvent **une** technologie particulière présentée comme la solution à tout. Vu l'expérience avec d'autres créations de Daniel Bernstein, il est probable que DNSCurve sera ainsi promu.

Avant de regarder DNSCurve et ses différences avec le protocole DNSSEC, revenons en arrière sur la sécurité du DNS. Lorsqu'un serveur DNS **faisant autorité** pour une **zone** envoie une réponse à un **résolveur**, un serveur DNS **récursif**, un méchant peut répondre avant le bon serveur et voir sa réponse acceptée par le résolveur, qui sera alors empoisonné (le RFC 5452 contient une explication plus détaillée et des calculs de la probabilité de réussite). C'est l'une des principales vulnérabilités du DNS.

Pour résoudre ce problème, il y a deux approches <https://www.bortzmeyer.org/securiser-le-dns.html>. Pour reprendre le vocabulaire du RFC 3552, on peut choisir de sécuriser le **canal** (vérifier qu'on parle bien à la machine à qui on voulait parler) ou de sécuriser le **message** (vérifier que le message vient bien de l'autorité et qu'il n'a pas été modifié en route). Les deux méthodes ne sont d'ailleurs pas contradictoires, on peut aussi choisir une **défense en profondeur** en adoptant les deux approches. En général, la sécurisation du message offre le maximum de sécurité (même si des intermédiaires ne sont pas dignes de confiance, on peut s'assurer de l'authenticité du message, ce qui est une propriété essentielle pour le DNS, qui dépend de tels intermédiaires, les serveurs cache) mais la sécurisation du canal est souvent plus facile à déployer.

Le débat existe pour tous les protocoles Internet. Par exemple, pour le courrier électronique, PGP sécurise le message, SMTP sur TLS (RFC 3207) sécurise le canal.

Dans le monde du DNS, la première solution, sécuriser le canal, est possible grâce à de nombreuses technologies :

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc3833.txt

— TSIG (RFC 8945), une signature de la transaction grâce à une clé secrète partagée entre les deux serveurs,
— SIG(0) (RFC 2931), une signature de la transaction utilisant des clés publiques,
— Diverses techniques de résistance à la fraude, comme le choix au hasard du port source (RFC 5452),
— Puisqu'une grande partie de la vulnérabilité du DNS vient de la taille trop petite du "*Query ID*", il y a des propositions de l'étendre comme les "*DNS cookies*" ("*Internet-Draft*" `draft-eastlake-dnsext-cooki` ou "*EDNS ping*" ("*Internet-Draft*" `draft-hubert-ulevitch-edns-ping`),
— IPsec (RFC 3401), si seulement il était déployé,
— Un protocole de transport résistant naturellement aux tricheries, comme TCP ou SCTP,
— Et pourquoi pas des datagrammes protégés par TLS comme avec DTLS (RFC 6347),
— Et enfin DNSCurve.

Toutes ces techniques ont leurs avantages et leurs inconvénients, certaines (comme celles du RFC 5452) ont eu de grands succès (comme de rendre peu exploitable la faille Kaminsky) mais toutes ont en commun de ne sécuriser que le canal entre deux serveurs. Si un serveur esclave d'une zone est contrôlé par un méchant et modifie les données, s'assurer qu'on parle bien à ce serveur ne servira à rien. Idem dans le cas, beaucoup plus fréquent, où un résolveur/cache d'un FAI ment à ses propres clients (cf. RFC 4924, section 2.5.2).

Pour la sécurisation du message, il n'existe à l'heure actuelle qu'une seule méthode, DNSSEC (RFC 4033 et suivants).

Revenons donc à DNSCurve. Cette technique n'est pas normalisée à l'heure actuelle. Il existe une présentation sommaire sur le site officiel <`http://www.dnscurve.org`> et un "*Internet-Draft*", `draft-dempsky-d` Pour la cryptographie sous-jacente, le mieux est de regarder le site Web de NaCl <`http://nacl.cace-project.eu/`>. Le principe de DNSCurve est que le dialogue entre deux serveurs est chiffré (par un algorithme de la famille des courbes elliptiques, Curve25519) et authentifié. La clé publique du serveur auquel on parle est obtenue dans le DNS, c'est le nom du serveur. Si le TLD `.example` est délégué à `uz5xgm1kx1zj8xsh51zp315k0rw7dcsgyxqh2sl7g8tjg25ltcvhyw.nic.example`, le résolveur peut, en examinant ce nom, voir que la délégation est sécurisée par DNSCurve, trouver la clé publique, et interroger le serveur de manière sûre, grâce au chiffrement asymétrique. Seule la communication entre deux serveurs est protégée. Si un des serveurs est une machine pirate, DNSCurve ne sert à rien.

Donc, DNSCurve n'est pas réellement un concurrent de DNSSEC, il traite un problème assez différent. Un exposé de l'auteur de DNSCurve <`http://cr.yp.to/talks/2009.08.10/slides.pdf`> compare DNSCurve avec DNSSEC et est intellectuellement très malhonnête, comme presque toujours avec Daniel Bernstein, avec notamment une utilisation intensive du FUD.

Que peut-on dire encore en comparant ces deux protocoles ? D'abord que DNSSEC a été soigneusement conçu pour que le serveur n'aie aucune opération de cryptographie à faire « par requête » (à l'exception de NSEC3 et encore il ne s'agit que de hachage, pas de chiffrement asymétrique). Toutes les opérations de signature peuvent être faites hors-ligne, une fois pour toute, sur la zone. Au contraire, DNSCurve exige des opérations de signature à chaque requête. Il ne convient donc pas à de gros serveurs.

Il est amusant de noter que la page officielle de DNSCurve reproche à DNSSEC que « "*DNSSEC reduces existing confidentiality by publishing the complete list of "secured" DNS records. This publication is integrated into the DNSSEC protocol ;*" » alors que cette publication est inhérente au fait de signer la zone et pas chaque requête. Si on est prêt à faire des opérations de cryptographie à chaque requête, les solutions des RFC 4470 et RFC 5155 résolvent le problème de la confidentialité des données.

On peut terminer par un point amusant : avec DNSCurve, la clé publique est encodée dans le nom de la machine. C'est astucieux. Mais les clés de la cryptographie asymétrique sont très longues et les composants d'un nom de domaine sont limités à 63 octets. Cela a deux conséquences :

— Pour les zones qui limitent la taille d'une délégation à 512 octets, comme la racine du DNS, on ne peut utiliser qu'un maximum de trois serveurs avec DNSCurve (car il faut aussi laisser de la place pour la question, et pour la colle).
— DNSCurve n'utilise donc pas les courbes elliptiques parce qu'elles sont meilleures que les algorithmes basés sur la factorisation des nombres premiers (comme RSA) mais simplement parce qu'elles permettent des clés plus courtes. Le fait de les mettre dans les noms de domaine va sérieusement contraindre l'évolution ultérieure de DNSCurve. Il sera par exemple impossible d'augmenter la taille des clés.

Il n'existe guère de textes d'analyse de DNScurve, ce système est très marginal. Notons toutefois un article d'Eric Rescorla, qui se concentre sur les questions de performance, « "*Some notes on DNSCurve*" <http://www.educatedguesswork.org/2010/02/some_notes_on_dnscurve.html> » et une très bonne synthèse de Paul Vixie, « "*Whither DNSCurve?*" <http://www.isc.org/community/blog/201002/whither-dnscurve> », qui se focalise sur le fait que DNScurve résout le problème facile que personne n'a, DNSSEC préférant attaquer le problème important.

Merci à Kim Minh Kaplan pour ses nombreuses et pertinentes remarques sur DNSCurve. Un autre bon article sur DNScurve est celui de Sam <http://maradns.blogspot.com/2010/09/eulogy-for-dnscurve.html>.