

RFC 5535 : Hash Based Addresses (HBA)

Stéphane Bortzmeyer

<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 18 Juin 2009

Date de publication du RFC : Juin 2009

<http://www.bortzmeyer.org/5535.html>

Les HBA ("*Hash Based Adresse*") de ce RFC sont des adresses IPv6 fondées sur un résumé cryptographique d'un certain nombre de valeurs, permettant de garantir que toutes les adresses HBA d'un **groupe** donné sont liées. Elles sont notamment utilisées par le protocole SHIM6 (RFC 5533¹). Contrairement aux adresses CGA du RFC 3972, dont elles sont proches, les HBA ne nécessitent pas de cryptographie asymétrique. Par contre, elles ne permettent pas de prouver le lien entre une machine et une adresse IP, juste de prouver que les adresses d'un groupe ont été générées par la même machine.

Quel problème essaient de résoudre les HBA (section 1 du RFC) ? Dans les protocoles conçus pour le "*multihoming*" comme SHIM6 (RFC 5533), une machine annonce à ses pairs qu'elle a plusieurs adresses IP. Qu'est-ce qui l'empêche d'annoncer des adresses qui ne sont pas « à elle », soit pour capter le trafic d'un tiers, soit pour réaliser une DoS contre ce tiers en provoquant l'envoi de nombreux paquets vers lui ? SHIM6 utilise deux mécanismes proches, les CGA et les HBA. Dans les deux cas, les adresses IP annoncées peuvent être vérifiées cryptographiquement. CGA signe les adresses avec une clé asymétrique donc le pair peut vérifier qu'une adresse « appartient » bien au pair. HBA procède différemment : une adresse HBA est composée de deux parties, un préfixe de 64 bits et un identificateur local (également de 64 bits) qui est un résumé de l'ensemble des préfixes IP de la machine (puisque HBA est fait pour les machines "*multihomées*", il y a plus qu'un préfixe) et d'un nombre aléatoire. HBA ne permet donc pas de prouver l'appartenance d'une adresse IP à une machine, uniquement le fait que toutes les adresses HBA d'un même groupe sont liées. Une machine SHIM6 qui utilise HBA ne pourra donc pas ajouter ou retirer des adresses IP au jeu d'adresses annoncé (elle le pourrait avec CGA) et HBA ne convient donc pas à des problèmes comme la renumérotation d'un réseau ou comme la mobilité. Mais le pair pourra vérifier, lors de l'établissement de l'association SHIM6, que ces adresses sont toutes liées à la même machine et sans les calculs longs et compliqués de la cryptographie asymétrique. HBA est donc plus économe. (Voir la section 3.3 pour une discussion plus détaillée du cahier des charges de HBA.)

1. Pour voir le RFC de numéro NNN, <http://www.ietf.org/rfc/rfcNNN.txt>, par exemple <http://www.ietf.org/rfc/rfc5533.txt>

Autrement, HBA et CGA sont très proches, utilisent les mêmes formats et ont également en commun de ne pas nécessiter de PKI.

Les risques de sécurité que les HBA traitent sont résumés en section 3.1, suivant le RFC 4218 (et traités en détail dans la section 11). En gros, les deux attaques possibles sont le détournement (un méchant va essayer d'obtenir des paquets qui n'étaient pas normalement pour lui) et l'inondation (un méchant va essayer de noyer une victime sous un flot de paquets). HBA protège uniquement contre le premier risque, d'autres protocoles doivent gérer le second, par exemple en testant si le pair est joignable (ce que fait SHIM6).

La section 3.2 donne le principe de base de HBA : si une machine a trois préfixes IPv6, mettons A, B et C, elle choisit un nombre M au hasard puis génère trois adresses, une par préfixe. Pour A, elle concatène au préfixe le résumé cryptographique de la concaténation de M, de A et la liste des préfixes (A, B et C). Même principe pour les autres préfixes. Étant donné une liste de préfixes (rappelez-vous que, avec HBA, elle doit être stable), un préfixe et le nombre M, on peut donc vérifier qu'une adresse fait bien partie du groupe, en recalculant le résumé.

Une particularité des HBA est qu'elles sont normalisées comme étant une variante des CGA du RFC 3972. La section 4 de notre RFC est donc dédiée à l'étude de la compatibilité exacte entre HBA et CGA. Cette compatibilité permet d'avoir des adresses qui sont à la fois HBA et CGA. On peut donc les utiliser dans des protocoles comme SEND (RFC 3971), qui impose des CGA. Il y a donc trois types d'adresses :

- Les CGA pures, normalisées dans le RFC 3972, qui n'incluent donc pas la nouvelle extension « multi-préfixe » (section 5),
- Les HBA pures qui n'utilisent pas du tout de clé publique,
- Les mixtes qui sont liées à la fois à une clé publique (CGA) et à un jeu de préfixes.

Quelle est cette extension « multi-préfixe » ? Décrite dans la section 5 de notre RFC, elle permet de représenter l'ensemble des préfixes liés par HBA. C'est cet ensemble qui servira d'entrée au hachage cryptographique.

Le mécanisme exact de génération est en section 6. Il est dérivé de celui de CGA, en section 4 du RFC 3972. La plus grande différence avec CGA est que l'adresse ne peut pas être fabriquée sans connaître les préfixes. Toutes les adresses doivent donc être générées ensemble. L'entrée des fonctions cryptographiques comporte donc l'extension multi-préfixe, qui indique les préfixes, une clé publique (si l'adresse est une mixte CGA/HBA) ou une initialisation aléatoire ("*nonce*") - encodée comme si c'était une clé publique CGA - si l'adresse est une HBA pure et les autres paramètres de CGA.

À ce stade, on a donc un ensemble d'adresses disponible pour la machine. Si une autre machine veut vérifier ces adresses, elle doit utiliser les techniques de la section 7. Par exemple, 7.2 expose comment vérifier qu'une adresse donnée fait bien partie d'un ensemble HBA. (On vérifie que le préfixe de l'adresse est dans l'ensemble indiqué dans l'extension multi-préfixe puis on refait le hachage et on vérifie que l'identificateur local, les 64 bits les plus à droite, sont bien les mêmes.)

La publication des adresses HBA dans le DNS fait l'objet de la section 9. La recommandation est que « ça dépend ». Selon que la machine a uniquement des adresses HBA ou pas, selon que les préfixes soient des ULA ou pas, il peut être raisonnable ou pas de publier ces adresses HBA dans le DNS. Puisqu'une adresse HBA ressemble à n'importe quelle autre adresse IPv6, le client qui les trouve dans le DNS ne sait pas s'il a affaire à une HBA ou pas et il doit donc déduire cela d'un autre mécanisme.

Une mise en œuvre de HBA en C se trouve en <http://ops.ietf.org/multi6/francis-hba.tar.gz>.