Groupe de travail Réseau
Request for Comments : 4982
RFC mise à jour : 3972
Catégorie : sur la voie de la normalisation

M. Bagnulo, UC3M
J. Arkko, Ericsson
juillet 2007
Traduction Claude Brière de L'Isle

Prise en charge de plusieurs algorithmes de hachage dans les adresses générées cryptographiquement (CGA)

Résumé
Le présent document analyse les implications de récentes attaques sur les fonctions de hachage couramment utilisées sur les adresses générées cryptographiquement (CGA, *Cryptographically Generated Addresses*) et met à jour la spécification de CGA pour prendre en charge plusieurs algorithmes de hachage.

Table des Matières

1. Introduction

De récentes attaques contre les fonctions de hachage couramment utilisées ont motivé une quantité considérable de soucis dans la communauté de l'Internet. L'approche recommandée dans la [RFC4270] et [NHA] pour traiter ce problème est d'abord d'analyser l'impact de ces attaques sur les différents protocoles Internet qui utilisent les fonctions de hachage et ensuite de s'assurer que les différents protocoles Internet qui utilisent les fonctions de hachage sont capables de migrer sur une autre fonction de hachage (plus sûre) sans perturbation majeure du fonctionnement de l'Internet.

Le présent document effectue une telle analyse pour les adresses générées cryptographiquement (CGA, *Cryptographically Generated Address*) définies dans la [RFC3972]. La première conclusion de l'analyse est que la sécurité des protocoles qui utilisent des CGA n'est pas affectée par les récentes attaques disponibles contre les fonctions de hachage. La seconde conclusion de l'analyse est que les fonctions de hachage utilisées sont incorporées dans la spécification de CGA. Le présent document met à jour la spécification de CGA [RFC3972] pour permettre la prise en charge d'autres fonctions de hachage. Pour ce faire, le présent document crée un nouveau registre géré par l'IANA pour enregistrer les différents algorithmes de hachage utilisés dans les CGA.

2. Terminologie

Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" en majuscules dans ce document sont à interpréter comme décrit dans le BCP 14, [RFC2119].

3. Impact des attaques de collision dans les CGA

Les progrès récents de la cryptographie ont résulté en une simplification des attaques contre la propriété d'absence de collision de certaines fonctions de hachage couramment utilisées [RFC4270], [NHA], y compris SHA-1 qui est la fonction de hachage utilisée par les CGA [RFC3972]. Le résultat est qu'il est possible d'obtenir deux messages, M1 et M2, qui ont la même valeur de hachage avec beaucoup moins de 2^(L/2) tentatives. Nous allons maintenant analyser l'impact de telles attaques dans les utilisations actuellement proposées des CGA.

D'après ce que nous comprenons, les attaques contre la propriété d'absence de collision d'une fonction de hachage remettent principalement en question l'application de ces fonctions de hachage pour la fourniture de capacités de non répudiation. En effet, un attaquant serait capable de créer deux messages différents qui aboutissent à la même valeur de hachage et il pourrait ensuite présenter n'importe lequel des messages de manière interchangeable (par exemple après que l'un d'entre eux a été signé par l'autre partie impliquée dans la transaction). Cependant, on doit noter que les deux messages doivent être générés par la même partie.

Pour autant qu'on le comprenne, les utilisations actuelles des CGA n'incluent pas la fourniture de capacités de non répudiation, de sorte que les attaques contre la propriété d'absence de collision de la fonction de hachage ne permettent pas d'attaque utile contre les protocoles fondés sur les CGA.

Les utilisations actuelles des CGA visent essentiellement à prouver la propriété d'une CGA et à la lier à d'autres adresses qui peuvent être utilisées pour atteindre la CGA d'origine. Ce type d'application des CGA comprend :

- L'application des CGA pour protéger le protocole shim6 [RFC5533]. Dans ce cas, les CGA sont utilisées comme identifiants pour les communications établies. Les caractéristiques des CGA sont utilisées pour prouver que le propriétaire de l'identifiant est celui qui fournit les adresses de remplacement qui peuvent être utilisées pour atteindre l'identifiant initial. Pour ce faire, la liste des adresses alternatives disponibles dans l'hôte multi rattachements est signée à l'aide de la clé privée de la CGA.

- L'application des CGA pour sécuriser le protocole de prise en charge de la mobilité IPv6 [RFC3775] comme proposé dans [CGA-CBA]. Dans ce cas, les CGA sont utilisées comme adresses de rattachement et servent à prouver que le propriétaire de l'adresse de rattachement est celui qui crée le lien avec la nouvelle adresse d'entretien. Comme dans le cas précédent, ceci est réalisé en signant le message de mise à jour de lien portant l'adresse d'entretien avec la clé privée de la CGA.

- L'application de la CGA à la découverte sécurisée de voisins [RFC3971]. Dans ce cas, les caractéristiques de la CGA sont utilisées pour prouver la propriété de l'adresse, de sorte qu'il est possible de vérifier que le propriétaire de l'adresse IP est celui qui fournit les informations d'adresse de couche 2. Ceci est réalisé en signant les informations d'adresse de couche 2 à l'aide de la clé privée de la CGA.

Essentiellement, toutes les applications courantes des CGA s'appuient sur les CGA pour protéger une communication entre deux homologues contre des attaques de tiers et non pour assurer la protection contre l'homologue lui-même. Les attaques contre la propriété de non collision des fonctions de hachage supposent qu'une des parties génère deux messages avec la même valeur de hachage afin de lancer une attaque contre son homologue dans la communication. Comme les CGA ne sont actuellement pas utilisées pour fournir ce type de protection, il est donc naturel qu'aucune attaque supplémentaire ne soit rendue possible par une plus faible résistance à la collision de la fonction de hachage.

4. Options de prise en charge de plusieurs algorithmes de hachage dans les CGA

Les CGA, telles qu'actuellement définies dans la [RFC3972], sont intrinsèquement liées à l'algorithme de hachage SHA-1 et aucune autre fonction de hachage n'est prise en charge.

Même si les attaques contre la propriété d'absence de collision des fonctions de hachage n'entraînent pas de nouvelles vulnérabilités dans les applications actuelles des CGA, il semble judicieux de permettre la prise en charge de plusieurs fonctions de hachage dans les CGA. Il y a principalement deux raisons à cela : premièrement, les futures applications potentielles de la technologie CGA pourraient être sensibles aux attaques contre la propriété d'absence de collision de SHA-1. La prise en charge d'autres fonctions de hachage permettrait aux applications qui ont des exigences plus strictes en matière de propriété d'absence de collision d'utiliser les CGA. Deuxièmement, une des leçons tirées des récentes attaques contre les fonctions de hachage est qu'il est possible qu'un jour on doive commencer à utiliser d'autres fonctions de hachage en raison d'attaques réussies contre d'autres propriétés des fonctions de hachage couramment utilisées. Il semble donc judicieux de modifier les protocoles en général et les CGA en particulier pour faciliter le plus possible la transition vers d'autres fonctions de hachage.

4.1 Où coder la fonction de hachage ?

La question suivante est de savoir où coder la fonction de hachage utilisée. Plusieurs options peuvent être envisagées :

Une option serait d'inclure la fonction de hachage utilisée en tant qu'entrée de la fonction de hachage. Cela signifie essentiellement qu'il faut créer une extension de la structure de données des paramètres CGA, telle que définie dans la [RFC4581], qui codifie la fonction de hachage utilisée. Le problème est que cette approche est vulnérable aux attaques de minoration ou de dégradation telles que définies dans [NHA]. Cela signifie que même si une fonction de hachage forte est utilisée, un attaquant pourrait trouver une structure de données de paramètres CGA qui utilise une fonction plus faible mais qui donne une valeur de hachage égale. Cela se produit lorsque la fonction de hachage originale H1 et la structure de données des paramètres CGA indiquant H1 aboutissent à la valeur X, et qu'une autre fonction de hachage H2 et la structure de données des paramètres CGA indiquant H2 aboutit également à la même valeur X.

En d'autres termes, l'attaque en dégradation va fonctionner comme suit : supposons qu'Alice génète une CGA CGA_A en utilisant la fonction de hachage forte HashStrong et en utilisant une structure de données de paramètres CGA CGA_PDS_A. La fonction de hachage choisie HashStrong est codée comme un champ d'extension dans CGA_PDS_A. Supposons qu'en utilisant une attaque en force brute, un attaquant X trouve une autre structure de données de paramètres CGA CGA_PDS_X dont la valeur de hachage, en utilisant une plus faible fonction de hachage, soit CGA_A. À ce point, l'attaquant peut prétendre être le propriétaire de CGA_A et la plus forte fonction de hachage n'a pas fourni de protection supplémentaire.

La conclusion de l'analyse précédente est que la fonction de hachage utilisée dans la génération de CGA doit être codée dans l'adresse elle-même.

Comme on veut prendre en charge plusieurs fonctions de hachage, on va probablement avoir besoin d'au moins 2 ou 3 bits pour cela.

Une option serait d'utiliser davantage de bits provenant des bits de hachage de l'identifiant d'interface. Cependant, le problème de cette approche est que la CGA résultante est plus faible car moins d'informations de hachage sont codées dans l'adresse. En outre, comme ces bits sont actuellement utilisés comme bits de hachage, il est impossible de rendre cette approche rétrocompatible avec les mises en œuvre existantes.

Une autre option serait d'utiliser les bits "u" et "g" pour coder cette information, mais ce n'est probablement pas une bonne idée car ces bits ont été respectés jusqu'à présent dans tous les mécanismes de génération d'identifiant d'interface, ce qui leur permet d'être utilisés dans leur but initial (par exemple, on peut toujours créer un registre mondial pour les identifiants d'interface uniques). Enfin, une autre option est de coder la valeur de hachage utilisée dans les bits Sec. Les bits Sec sont utilisés pour introduire artificiellement une difficulté supplémentaire dans le processus de génération des CGA afin de fournir une protection supplémentaire contre les attaques en force brute. Les bits Sec ont été conçus de manière à prolonger la durée de vie des CGA lorsqu'il est possible d'attaquer des valeurs de hachage de 59 bits. Cependant, ce n'est pas le cas aujourd'hui, de sorte qu'en général, les CGA auront une valeur de Sec de 000. La proposition est de coder dans les bits Sec, non seulement des informations sur la protection contre les attaques en force brute, mais aussi la fonction de hachage utilisée pour générer le hachage. Ainsi, par exemple, la valeur de Sec 000 signifierait que la fonction de hachage utilisée est SHA-1 et que les bits 0 de hash2 (tels que définis dans la RFC 3972) doivent être 0. La valeur de Sec 001 pourrait signifier que la fonction de hachage utilisée est SHA-1 et que les 16 bits de hash2 (tels que définis dans la RFC 3972) doivent être zéro. Cependant, les autres valeurs de Sec pourraient signifier qu'une autre fonction de hachage doit être utilisée et qu'un certain nombre de bits de hash2 doivent être à zéro. Il n'est pas proposé de définir une fonction de hachage concrète à utiliser pour les autres valeurs de Sec, car il n'est pas encore clair qu'il faille le faire, ni quelle fonction de hachage devrait être sélectionnée.

Noter que comme il n'y a que 8 valeurs de Sec, il peut être nécessaire de réutiliser des valeurs de Sec quand on a épuisé les valeurs de Sec non utilisées. Le scénario où une telle approche a un sens est lorsque il y a des valeurs de Sec qui ne sont plus utilisées parce que la sécurité résultante est devenue faible. Dans ce cas, lorsque l'usage de la valeur de Sec a été abandonné depuis longtemps, il serait possible de réallouer les valeurs de Sec. Cependant, cela doit être une option de dernier ressort, car cela peut affecter l'interopérabilité. C'est pourquoi deux mises en œuvre qui utilisent des significations différentes d'une certaine valeur de Sec ne seront pas capables d'interopérer correctement (c'est-à-dire, si une ancienne mise en œuvre reçoit une CGA générée avec la nouvelle signification de la valeur de Sec, elle va échouer et la même chose pour une nouvelle mise en œuvre qui reçoit une CGA générée avec l'ancienne signification de la valeur de Sec). Dans le cas de l'approche de la réallocation d'une valeur de Sec, un long délai est nécessaire entre le moment où l'ancienne valeur est déconseillée et la réallocation afin de prévenir une mauvaise interprétation de la valeur par d'anciennes mises en œuvre.

Une interprétation erronée d'une valeur de Sec réutilisée, du côté du possesseur de la CGA comme du côté du vérificateur de la CGA, aurait le résultat suivant : la vérification de la CGA va échouer dans le pire des cas et les deux nœuds vont devoir revenir à des adresses IP non protégées. Cela ne peut arriver qu'avec des ensembles de paramètres de CGA obsolèts, qui seraient de toute façon considérés comme non sûrs. Dans tous les cas, une mise en œuvre ne doit pas prendre en charge simultanément deux significations différentes d'une valeur de Sec.

5. Procédure de génération de CGA

Le registre SEC défini à la Section des considérations relatives à l'IANA du présent document contient des entrées pour les différentes valeurs de Sec. Chacune de ces entrées pointe sur une RFC qui définit les procédures de génération de CGA qui DOIVENT être utilisées lors de la génération des CGA avec la valeur de Sec associée.

On devrait noter que les procédures de génération de CGA peuvent être changées par de nouvelles procédures non seulement en termes de fonction de hachage utilisée mais aussi sous d'autres aspects, par exemple, des valeurs de modificateur plus longues peuvent être nécessaires si le nombre de 0 requis dans hash2 excède la limite actuellement définie de 112 bits. La nouvelle procédure (qui implique potentiellement une valeur de modificateur plus longue) serait décrite dans la RFC sur laquelle pointe l'entrée correspondante du registre Sec.

De plus, la RFC qui définit les procédures de génération de CGA pour une valeur de Sec DOIT explicitement définir la longueur minimum de clé acceptable pour les CGA avec cette valeur de Sec. C'est pour fournir une protection cohérente du hachage et des techniques de clé publique.

6. Considérations relatives à l'IANA

Le présent document définit un nouveau registre intitulé "CGA SEC" pour le champ Sec défini dans la [RFC3972] qui a été créé et est tenu par l'IANA. Les valeurs dans cet espace de noms sont des entiers non signés de 3 bits.

Les valeurs initiales pour le champ Type d'extension de CGA sont données ci-dessous ; les allocations futures sont à faire par action de normalisation de la [RFC2434]. Les allocations consistent en un nom, la valeur, et le numéro de la RFC où la procédure de génération de CGA est définie.

Les valeurs initiales suivantes sont allouées dans le présent document :

Nom	**Valeur**	**RFC**
SHA-1_0hash2bits	000	3972, 4982
SHA-1_16hash2bits	001	3972, 4982
SHA-1_32hash2bits	010	3972, 4982

7. Considérations sur la sécurité

Le présent document est sur les questions de sécurité, et en particulier, sur la protection contre de potentielles attaques contre les fonctions de hachage.

8. Remerciements

Russ Housley, James Kempf, Christian Vogt, Pekka Nikander, et Henrik Levkowetz ont relu le présent document et fourni des commentaires.

Marcelo Bagnulo a travaillé sur ce document tout en visitant le laboratoire de recherche Nomadiclab de Ericsson.

9. Références

9.1 Références normatives

[RFC2119] S. Bradner, "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997. *(MàJ par* RFC8174)

[RFC3971] J. Arkko et autres, "Découverte de voisin sûre (SEND)", mars 2005. (*MàJ par* RFC6494) *(P.S.)*

[RFC3972] T. Aura, "Adresses générées cryptographiquement (CGA)", mars 2005. (*MàJ par* RFC4581, RFC4982) *(P.S.)*

[RFC4581] M. Bagnulo, J. Arkko, "Format de champ d'extension des adresses générées cryptographiquement (CGA)", octobre 2006. (*MàJ* RFC3972) *(P.S.)*

9.2 Références pour information

[CGA-CBA] Arkko, J., "Applying Cryptographically Generated Addresses and Credit-Based Authorization to Mobile IPv6", Travail en cours, juin 2006.

[NHA] Bellovin, S. and E. Rescorla, "Deploying a New Hash Algorithm", NDSS '06, février 2006.

[RFC2434] T. Narten et H. Alvestrand, "Lignes directrices pour la rédaction d'une section Considérations relatives à l'IANA dans les RFC", BCP 26, octobre 1998. *(Rendue obsolète par la* RFC5226)

[RFC3775] D. Johnson, C. Perkins, J. Arkko, "Prise en charge de la mobilité dans IPv6", juin 2004. *(P.S.) (Obs., voir* RFC6275)

[RFC4270] P. Hoffman, B. Schneier, "Attaques contre les hachages cryptographiques dans les protocoles Internet", novembre 2005. *(Info.)*

[RFC5533] E. Nordmark, M. Bagnulo, "Shim6 : Protocole Shim de niveau 3 de multi rattachement pour IPv6", juin 2009. *(P. S.)*

Adresse des auteurs

Marcelo Bagnulo
Universidad Carlos III de Madrid
Av. Universidad 30
Leganes, Madrid 28911
SPAIN
téléphone : 34 91 6249500
mél : marcelo@it.uc3m.es
URI : http://www.it.uc3m.es

Jari Arkko
Ericsson
Jorvas 02420
Finland
mél : jari.arkko@ericsson.com

Déclaration complète de droits de reproduction

Remerciement
Le financement de la fonction d'édition des RFC est actuellement fourni par la Internet Society.