Équipe d'ingénierie de l'Internet (IETF)
Request for Comments : 8205
Catégorie : Sur la voie de la normalisation
ISSN: 2070-1721

M. Lepinski, éditeur, NCF
K. Sriram, éditeur, NIST
septembre 2017
Traduction Claude Brière de L'Isle

Spécification du protocole BGPsec

Résumé
Le présent document décrit BGPsec, extension au protocole de routeur frontière (BGP, *Border Gateway Protocol*) qui assure la sécurité pour le chemin des systèmes autonomes (AS, *Autonomous System*) à travers lequel passe un message BGP UPDATE. BGPsec est mis en œuvre via un attribut de chemin BGP facultatif non transitif qui porte les signatures numériques produites par chaque AS qui propage le message UPDATE. Les signatures numériques fournissent l'assurance que chaque AS sur le chemin des AS énumérés dans le message UPDATE a explicitement autorisé l'annonce de la route.

Statut de ce mémoire
Ceci est un document de l'Internet en cours de normalisation.

Le présent document a été produit par l'équipe d'ingénierie de l'Internet (IETF). Il représente le consensus de la communauté de l'IETF. Il a subi une révision publique et sa publication a été approuvée par le groupe de pilotage de l'ingénierie de l'Internet (IESG). Tous les documents approuvés par l'IESG ne sont pas candidats à devenir une norme de l'Internet ; voir la Section 2 de la RFC5741.

Les informations sur le statut actuel du présent document, tout errata, et comment fournir des réactions sur lui peuvent être obtenues à http://www.rfc-editor.org/info/rfc8205

Table des matières

1. Introduction

Le présent document décrit BGPsec, un mécanisme pour assurer la sécurité de chemin pour les annonces de chemins du protocole de routeur frontière (BGP, *Border Gateway Protocol*) [RFC4271]. C'est à dire qu'un locuteur BGP qui reçoit un message BGPsec UPDATE valide a l'assurance cryptographique que le chemin annoncé a la propriété suivante : chaque système autonome (AS, *Autonomous System*) sur le chemin des AS énumérés dans le message UPDATE a explicitement autorisé l'annonce du chemin à l'AS suivant dans le chemin.

Le présent document spécifie un attribut de chemin BGP facultatif (non transitif) BGPsec_PATH. Il décrit aussi comment un locuteur BGP conforme à BGPsec (appelé à partir d'ici un locuteur BGPsec) peut générer, propager, et valider les messages BGP UPDATE contenant cet attribut pour obtenir l'assurance ci-dessus.

BGPsec est destiné à être utilisé pour augmenter la validation d'origine BGP [RFC6483], [RFC6811], et quand utilisé en conjonction avec la validation d'origine, il est possible d'empêcher une large variété d'attaques de capture de chemin contre BGP.

BGPsec s'appuie sur les certificats de l'infrastructure de clé publique de ressource (RPKI, *Resource Public Key Infrastructure*) qui attestent l'allocation des numéros d'AS et des ressources d'adresse IP. (Voir plus d'informations sur la RPKI, dans la [RFC6480] et les documents qui y sont référencés.) Tout locuteur BGPsec qui souhaite envoyer, à des homologues BGP externes (eBGP) des messages BGP UPDATE contenant le BGPsec_PATH doit posséder une clé privée associée à un certificat de routeur RPKI [RFC8209] qui correspond au numéro d'AS du locuteur BGPsec. Noter, cependant, qu'un locuteur BGPsec n'a pas besoin d'un tel certificat afin de valider les messages UPDATE reçus qui contiennent l'attribut BGPsec_PATH (voir au paragraphe 5.2).

1.1 Langage des exigences

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], [RFC8174] quand, et seulement quand ils apparaissent tout en majuscules, comme montré ci-dessus.

2. Négociation BGPsec

Le présent document définit une capacité BGP [RFC5492] qui permet à un locuteur BGP d'annoncer à un voisin sa capacité d'envoyer ou recevoir des messages BGPsec UPDATE (c'est-à-dire, des messages UPDATE qui contiennent l'attribut BGPsec_PATH).

2.1 Capacité BGPsec

Cette capacité a le code 7.

La longueur de capacité DOIT être réglée à 3.

Les 3 octets du format de capacité sont spécifiés à la Figure 1.

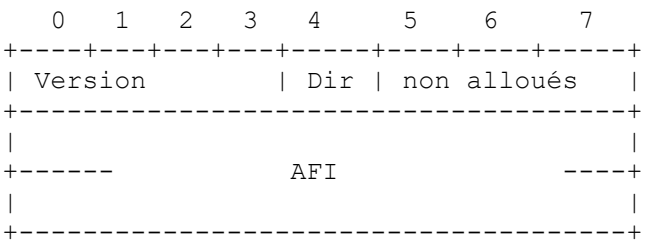

Figure 1 : Format de capacité BGPsec

Les 4 premiers bits du premier octet indiquent la version de BGPsec pour laquelle le locuteur BGP annonce sa prise en charge. Le présent document définit seulement BGPsec version 0 (les 4 bits réglés à 0). D'autres versions de BGPsec pourront peut-être être définies dans de futurs documents. Un locuteur BGPsec PEUT annoncer la prise en charge de plusieurs versions de BGPsec en incluant plusieurs versions de la capacité BGPsec dans son message BGP OPEN.

Le cinquième bit du premier octet est un bit de direction qui indique si le locuteur BGP annonce la capacité d'envoyer des messages BGPsec UPDATE ou de recevoir des messages BGPsec UPDATE. Le locuteur BGP règle ce bit à 0 pour indiquer la capacité de recevoir des messages BGPsec UPDATE. Le locuteur BGP établit ce bit à 1 pour indiquer la capacité d'envoyer des messages BGPsec UPDATE.

Les trois bits restants du premier octet ne sont pas alloués et sont réservés pour une utilisation future. Ces bits sont réglés à 0 par l'envoyeur et ignorés à réception.

Le second et le troisième octet contiennent l'identifiant de famille d'adresse (AFI, *Address Family Identifier*) de 16 bits, qui indique la famille d'adresses pour laquelle le locuteur BGPsec annonce sa prise en charge de BGPsec. Le présent document spécifie seulement BGPsec pour l'utilisation avec deux familles d'adresses, IPv4 et IPv6, avec les valeurs d'AFI respectivement de 1 et de 2 [IANA-AF]. BGPsec utilisé avec d'autres familles d'adresses pourra être spécifié dans des documents futurs

2.2 Négociation de la prise en charge de BGPsec

Afin d'indiquer qu'un locuteur BGP veut envoyer des messages BGPsec UPDATE (pour une famille d'adresses particulière) un locuteur BGP envoie la capacité BGPsec (voir au paragraphe 2.1) avec le bit Direction (cinquième bit du premier octet) établi à 1. Afin d'indiquer que le locuteur veut recevoir les messages BGP UPDATE contenant l'attribut BGPsec_PATH (pour une famille d'adresses particulière) un locuteur BGP envoie la capacité BGPsec avec le bit Direction réglé à 0. Afin d'annoncer la capacité d'envoyer et recevoir les messages BGPsec UPDATE, le locuteur BGP envoie deux copies de la capacité BGPsec (une avec le bit Direction à 0 et une avec le bit Direction établi à 1).

De même, si un locuteur BGP souhaite utiliser BGPsec avec deux familles d'adresses différentes (c'est-à-dire, IPv4 et IPv6) sur la même session BGP, il va inclure deux instances de cette capacité (une pour chaque famille d'adresses) dans le message BGP OPEN. Un locuteur BGP NE DOIT PAS annoncer la capacité BGPsec si il ne prend pas en charge l'extension BGP multi protocoles [RFC4760]. De plus, un locuteur BGP NE DOIT PAS annoncer la capacité de prise en charge de BGPsec pour un AFI particulier à moins qu'il n'ait aussi annoncé la capacité d'extension multi protocoles pour le même AFI [RFC4760].

Dans une session BGPsec d'échange de trafic, il est permis à un homologue d'envoyer des messages UPDATE contenant l'attribut BGPsec_PATH si et seulement si :

- o l'homologue concerné a envoyé la capacité BGPsec pour une version particulière de BGPsec et une famille d'adresses particulière avec le bit Direction établi à 1, et
- o l'autre homologue (receveur) a envoyé la capacité BGPsec pour la même version de BGPsec et la même famille d'adresses avec le bit Direction réglé à 0.

Dans une telle session, on peut dire que l'utilisation de la version particulière de BGPsec a été négociée pour une famille d'adresses particulière. Les messages BGP UPDATE traditionnels (c'est-à-dire, non signés, contenant l'attribut AS_PATH) PEUVENT être envoyés au sein d'une session sans considération de si l'utilisation de BGPsec est ou non négociée avec succès. Cependant, si BGPsec n'est pas négocié avec succès, les messages BGP UPDATE contenant l'attribut BGPsec_PATH NE DOIVENT alors PAS être envoyés.

Le présent document définit le comportement des mises en œuvre dans le cas où BGPsec version 0 est la seule version négociée avec succès. Tout futur document qui spécifiera des versions supplémentaires de BGPsec devra spécifier le comportement en cas de négociation de la prise en charge de plusieurs versions.

BGPsec ne peut pas fournir de garanties de sécurité significatives dans la prise en charge de numéros d'AS de quatre octets. Donc, tout locuteur BGP qui annonce la capacité BGPsec DOIT aussi annoncer la capacité de prise en charge d'AS de quatre octets [RFC6793]. Si un locuteur BGP envoie la capacité BGPsec mais pas la capacité de prise en charge d'AS à quatre octets, BGPsec n'a alors pas été négocié avec succès, et les messages UPDATE contenant l'attribut BGPsec_PATH NE DOIVENT PAS être envoyés dans une telle session.

3. Attribut BGPsec_PATH

L'attribut BGPsec_PATH est un attribut de chemin BGP facultatif non transitif.

Le présent document enregistre un code de type d'attribut pour cet attribut : BGPsec_PATH (voir la Section 9).

L'attribut BGPsec_PATH porte les informations sécurisées concernant le chemin des AS à travers lesquels passe un message UPDATE. Cela inclut les signatures numériques utilisées pour protéger les informations de chemin. Les messages UPDATE qui contiennent l'attribut BGPsec_PATH sont appelés des "messages BGPsec UPDATE". L'attribut BGPsec_PATH remplace l'attribut AS_PATH dans un message BGPsec UPDATE. C'est-à-dire que les messages UPDATE qui contiennent l'attribut BGPsec_PATH NE DOIVENT PAS contenir l'attribut AS_PATH, et vice versa.

L'attribut BGPsec_PATH est constitué de plusieurs parties. Le diagramme général de la Figure 2 fournit une vue d'ensemble de la structure de l'attribut BGPsec_PATH. ((SKI, *Subject Key Identifier*) dans la Figure 2 signifie identifiant de clé sujette.)

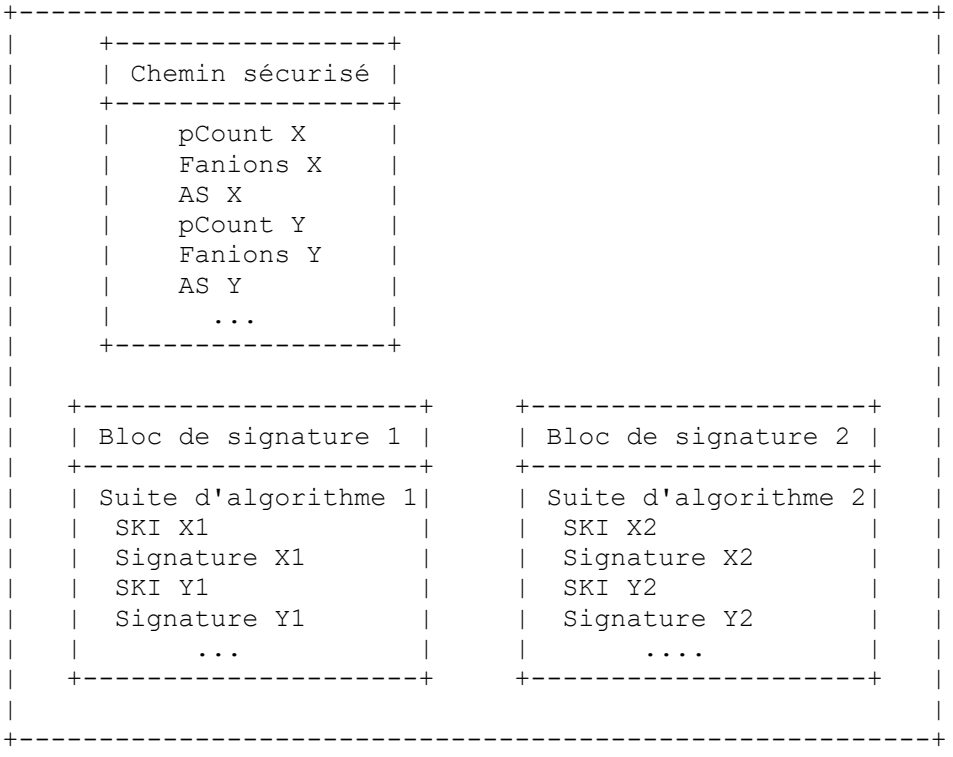

Figure 2 : Diagramme général de l'attribut BGPsec_PATH

La Figure 3 fournit la spécification du format pour l'attribut BGPsec_PATH.

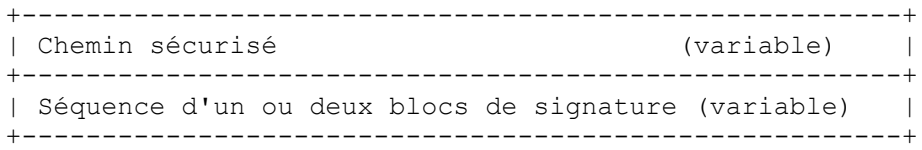

Figure 3 : Format d'attribut BGPsec_PATH

Le chemin sécurisé contient les informations du chemin d'AS pour le message BGPsec UPDATE. Ceci est logiquement équivalent aux informations contenues dans un attribut AS_PATH non BGPsec. Les informations dans le chemin sécurisé sont utilisées par les locuteurs BGPsec de la même façon que les informations provenant de AS_PATH sont utilisées par les locuteurs non BGPsec. Le format du chemin sécurisé est décrit au paragraphe 3.1.

L'attribut BGPsec_PATH va contenir un ou deux blocs de signature, dont chacun correspond à une suite d'algorithme différente. Chaque bloc de signature va contenir un segment de signature pour chaque numéro d'AS (c'est-à-dire, de segment Secure_Path) dans le chemin sécurisé. Dans le cas le plus courant, l'attribut BGPsec_PATH va contenir seulement un seul bloc de signature. Cependant, afin de permettre une transition d'une vieille suite d'algorithmes à une nouvelle (sans jour fanion) il va être nécessaire d'inclure deux blocs de signature (un pour la vieille suite d'algorithmes et un pour la nouvelle) durant la période de transition. (Voir au paragraphe 6.1 une discussion des transitions d'algorithmes.) Le format des blocs de signature est décrit au paragraphe 3.2.

3.1 Secure_Path

On fournit ici une description détaillée des informations de Secure_Path dans l'attribut BGPsec_PATH. La spécification pour le champ Secure_Path est fournie dans les Figures 4 et 5.

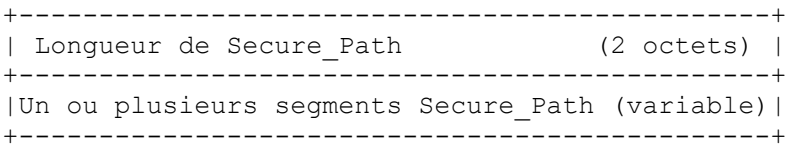

Figure 4 : Format Secure_Path

Le champ Longueur de Secure_Path contient la longueur (en octets) entière de Secure_Path (incluant les 2 octets utilisés pour exprimer ce champ de longueur). Comme expliqué ci-dessous, chaque segment de Secure_Path fait 6 octets. Noter que Longueur de Secure_Path est supérieur de deux à six fois le nombre de segments de Secure_Path (c'est-à-dire, au nombre de numéros d'AS dans le chemin).

Le chemin sécurisé contient un segment Secure_Path (voir la Figure 5) pour chaque AS dans le chemin à l'AS d'origine du préfixe spécifié dans le message UPDATE. (Note : les AS répétés sont "compressés" en utilisant le champ pCount, comme discuté ci-dessous.)

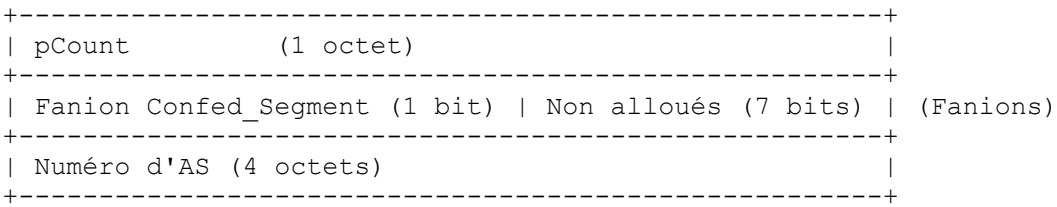

Figure 5 : Format de segment Secure_Path

Le numéro d'AS (dans la Figure 5) est celui du locuteur BGP qui a ajouté ce segment Secure_Path à l'attribut BGPsec_PATH. (Voir à la Section 4 plus d'informations sur le remplissage de ce champ.)

Le champ pCount contient le nombre de répétitions du numéro d'AS associé que couvre la signature. Ce champ permet à un

locuteur BGPsec d'imiter la sémantique de l'ajout de plusieurs copies de leur AS à l'AS_PATH sans exiger que le locuteur génère plusieurs signatures. Noter que le paragraphe 9.1.2.2 ("Départages (phase 2)") de la [RFC4271] mentionne le "nombre de numéros d'AS" dans l'attribut AS_PATH qui est utilisé dans le processus de choix du chemin. Cette métrique (nombre de numéros d'AS) est la même que la longueur de chemin d'AS obtenue dans BGPsec en additionnant les valeurs de pCount dans l'attribut BGPsec_PATH. Le champ pCount est aussi utile pour gérer les serveurs de chemins (voir au paragraphe 4.2), les confédérations d'AS (voir au paragraphe 4.3), et les migrations de numéros d'AS (voir les détails dans la [RFC8206]).

Le bit le plus à gauche (c'est-à-dire, de poids fort) du champ Fanions dans la Figure 5 est le fanion Confed_Segment. Il est établi à 1 pour indiquer que le locuteur BGPsec qui a construit ce segment de Secure_Path envoie le message UPDATE à un AS homologue au sein de la même confédération d'AS [RFC5065]. (C'est à dire, une séquence de fanions consécutifs Confed_Segment est établie dans un message BGPsec UPDATE chaque fois que, dans un message non BGPsec UPDATE, se produit un segment AS_PATH de type AS_CONFED_SEQUENCE.) Dans tous les autres cas, le fanion Confed_Segment est réglé à 0.

Les 7 bits restants du champ Fanions ne sont pas alloués. Ils DOIVENT être réglés à 0 par l'envoyeur et ignorés par le receveur. Noter cependant que la signature est calculée sur tous les 8 bits du champ Fanions.

Comme indiqué au paragraphe 2.2, l'échange de trafic BGPsec exige que les AS communicants DOIVENT chacun prendre en charge les numéros d'AS de 4 octets. Les numéros d'AS de deux octets actuellement alloués sont convertis en numéros d'AS à 4 octets en réglant les deux octets de poids fort du champ de 4 octets à 0 [RFC6793].

3.2 Signature_Block

Les Figures 6 et 7 fournissent une description détaillée des blocs de signature dans l'attribut BGPsec_PATH.

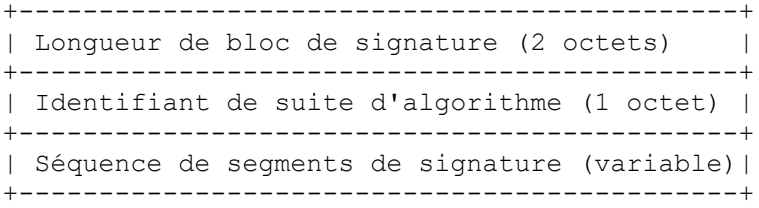

Figure 6 : Format de bloc de signature

La longueur de bloc de signature dans la Figure 6 est le nombre total d'octets dans le bloc de signature (incluant les 2 octets utilisés pour exprimer ce champ de longueur).

Identifiant de suite d'algorithme est un identifiant de 1 octet qui spécifie l'algorithme de résumé et l'algorithme de signature numérique utilisés pour produire la signature numérique dans chaque segment de signature. Un registre IANA des identifiants de suite d'algorithmes à utiliser dans BGPsec est spécifié dans le document des algorithmes BGPsec [RFC8208].

Un bloc de signature dans la Figure 6 a exactement un segment de signature (voir la Figure 7) pour chaque segment de chemin sécurisé dans la portion Chemin sécurisé de l'attribut BGPsec_PATH (c'est-à-dire, un segment de signature pour chaque AS distinct sur le chemin pour le préfixe dans le message UPDATE).

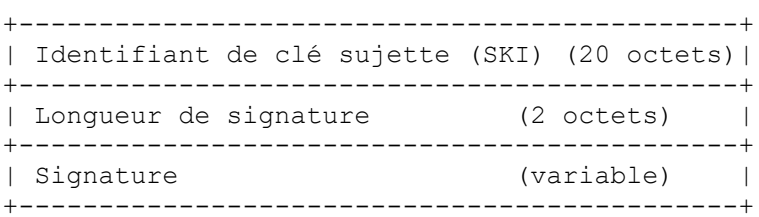

Figure 7 : Format de segment de signature

le champ Identifiant de clé sujette (SKI, *Subject Key Identifier*) de la Figure 7 contient la valeur de l'extension Identifiant de clé sujette du certificat de routeur RPKI [RFC6487] qui est utilisé pour vérifier la signature (voir à la Section 5 les détails

sur la validité des messages BGPsec UPDATE). Le champ SKI a une taille fixe de 20 octets. Voir au paragraphe 6.2 les considérations sur la taille du SKI.

Le champ Longueur de signature contient la taille (en octets) de la valeur du champ Signature du segment de signature.

Le champ Signature de la Figure 7 contient une signature numérique qui protège le préfixe et l'attribut BGPsec_PATH (voir aux Sections 4 et 5 les détails de, respectivement, la génération et la validation de signature).

4. Message BGPsec UPDATE

Le paragraphe 4.1 fournit les lignes directrices générales pour la création des messages BGPsec UPDATE – c'est-à-dire, les messages UPDATE qui contiennent l'attribut BGPsec_PATH.

Le paragraphe 4.2 spécifie comment un locuteur BGPsec génère l'attribut BGPsec_PATH à inclure dans un message BGPsec UPDATE.

Le paragraphe 4.3 contient les instructions de traitement particulières pour les membres d'une confédération d'AS [RFC5065]. Un locuteur BGPsec qui n'est pas membre d'une telle confédération NE DOIT PAS établir le fanion Confed_Segment dans son segment Secure_Path (c'est-à-dire, laisser le fanion Confed_Segment à la valeur par défaut de 0) dans tous les messages BGPsec UPDATE qu'il envoie.

Le paragraphe 4.4 contient des instructions pour reconstruire l'attribut AS_PATH dans les cas où un locuteur BGPsec reçoit un message UPDATE avec un attribut BGPsec_PATH et souhaite propager le message UPDATE à un homologue qui ne prend pas en charge BGPsec.

4.1 Lignes directrices générales

Les informations protégées par la signature sur un message BGPsec UPDATE incluent le numéro d'AS de l'homologue à qui le message UPDATE est envoyé. Donc, si un locuteur BGPsec souhaite envoyer un message BGPsec UPDATE à plusieurs homologues BGP, il DOIT générer un message BGPsec UPDATE séparé pour chaque AS homologue unique à qui le message UPDATE est envoyé.

Un message BGPsec UPDATE DOIT annoncer un chemin à seulement un préfixe. C'est parce que un locuteur BGPsec qui reçoit un message UPDATE avec plusieurs préfixes va être incapable de construire un message BGPsec UPDATE valide (c'est-à-dire, des signatures de chemin valides) contenant un sous ensemble des préfixes dans la mise à jour reçue. Si un locuteur BGPsec souhaite annoncer des chemins à plusieurs préfixes, il DOIT alors générer un message BGPsec UPDATE séparé pour chaque préfixe. De plus, un message BGPsec UPDATE DOIT utiliser l'attribut MP_REACH_NLRI [RFC4760] pour coder le préfixe.

Les attributs BGPsec_PATH et AS_PATH sont mutuellement exclusifs. C'est à dire que tout message UPDATE qui contient l'attribut BGPsec_PATH NE DOIT PAS contenir l'attribut AS_PATH. Les informations qui seraient contenues dans l'attribut AS_PATH sont plutôt portées dans la portion Secure_Path de l'attribut BGPsec_PATH.

Afin de créer ou ajouter une nouvelle signature à un message BGPsec UPDATE avec une certaine suite d'algorithmes, le locuteur BGPsec DOIT posséder une clé privée convenable pour générer des signatures pour cette suite d'algorithmes. De plus, cette clé privée doit correspondre à la clé publique dans un certificat d'entité d'extrémité RPKI valide dont l'extension de ressource de numéro d'AS inclut le numéro d'AS du locuteur BGPsec [RFC8209]. Noter aussi que de nouvelles signatures ne sont ajoutées à un message BGPsec UPDATE que quand un locuteur BGPsec génère un message UPDATE à envoyer à un homologue externe (c'est-à-dire, quand le numéro d'AS de l'homologue n'est pas égal à celui du propre AS du locuteur BGPsec).

La RPKI permet au détenteur légitime du ou des préfixes d'adresse IP de produire un objet signé, appelé une autorisation d'origine de chemin (ROA, *Route Origin Authorization*) qui autorise un certain AS à générer des routes pour un certain ensemble de préfixes (voir la [RFC6482]). On prévoit que la plupart des consommateurs d'assertions (RP, *Relying Party*) vont utiliser BGPsec en tandem avec la validation d'origine (voir la [RFC6483] et la [RFC6811]). Donc, il est RECOMMANDÉ qu'un locuteur BGPsec ne génère un message BGPsec UPDATE annonçant une route pour un certain préfixe que si il existe une ROA valide qui autorise l'AS du locuteur BGPsec a générer des routes pour ce préfixe.

Si un routeur BGPsec a reçu seulement un message non BGPsec UPDATE contenant l'attribut AS_PATH (au lieu de l'attribut BGPsec_PATH) provenant d'un homologue pour un certain préfixe, il NE DOIT alors PAS attacher un attribut BGPsec_PATH quand il propage le message UPDATE. (Noter qu'un routeur BGPsec peut aussi recevoir un message non BGPsec UPDATE provenant d'un homologue interne sans l'attribut AS_PATH, c'est-à-dire, avec juste les informations d'accessibilité de couche réseau (NLRI, *Network Layer Reachability Information*) dedans. Dans ce cas, le préfixe est originaire de cet AS, et si il est choisi pour annonce, le locuteur BGPsec DEVRAIT joindre un attribut BGPsec_PATH et envoyer une route signée (pour ce préfixe) à ses homologues externes locuteurs BGPsec.)

À l'inverse, si un routeur BGPsec a reçu un message BGPsec UPDATE (avec l'attribut BGPsec_PATH) d'un homologue pour un certain préfixe et si il choisit de propager le chemin de cet homologue pour le préfixe, il DEVRAIT alors propager le chemin comme un message BGPsec UPDATE contenant l'attribut BGPsec_PATH.

Noter que retirer les signatures BGPsec (c'est-à-dire, propager une annonce de chemin sans l'attribut BGPsec_PATH) a des ramifications de sécurité significatives. (Voir à la Section 8 la discussion des ramifications pour la sécurité de la suppression des signatures BGPsec.) Donc, quand une annonce de chemin est reçue via un message BGPsec UPDATE, propager l'annonce de chemin sans l'attribut BGPsec_PATH est NON RECOMMANDÉ, sauf si le message est envoyé à un homologue qui n'a pas annoncé la capacité de recevoir les messages BGPsec UPDATE (voir au paragraphe 4.4).

De plus, on note que lorsque un locuteur BGPsec propage une annonce de chemin avec l'attribut BGPsec_PATH, il n'atteste pas l'état de validation du message UPDATE reçu. (Voir à la Section 8 la discussion de la signification pour la sécurité des signatures BGPsec.)

Si le locuteur BGPsec produit un message UPDATE qui contiendrait, en l'absence de BGPsec, un AS_SET (par exemple, le locuteur BGPsec effectue une agrégation de mandataires) alors le locuteur BGPsec NE DOIT PAS inclure l'attribut BGPsec_PATH. Dans un tel cas, le locuteur BGPsec DOIT retirer tout BGPsec_PATH existant dans la ou les annonces reçues pour ce préfixe et produire un message traditionnel (non BGPsec) UPDATE. On devrait noter que le BCP 172 [RFC6472] recommande de ne pas utiliser AS_SET et AS_CONFED_SET dans le AS_PATH des messages BGP UPDATE.

Le cas où le locuteur BGPsec envoie un message BGPsec UPDATE à un homologue iBGP (BGP interne) est assez simple. Quand il génète une nouvelle annonce de chemin et l'envoie à un homologue iBGP à capacité BGPsec, le locuteur BGPsec omet l'attribut BGPsec_PATH. Quand il génète une nouvelle annonce de chemin et l'envoie à un homologue iBGP non BGPsec, le locuteur BGPsec inclut un attribut AS_PATH vide dans le message UPDATE. (Un attribut AS_PATH vide est celui dont le champ Longueur contient la valeur 0 [RFC4271].) Quand un locuteur BGPsec choisit de transmettre un message BGPsec UPDATE à un homologue iBGP, l'attribut BGPsec_PATH NE DEVRAIT PAS être retiré, sauf si l'homologue ne prend pas en charge BGPsec. Dans le cas où un homologue iBGP ne prend pas en charge BGPsec, un message BGP UPDATE avec AS_PATH est alors reconstruit à partir du message BGPsec UPDATE et ensuite transmis (voir au paragraphe 4.4). En particulier, quand il est transmis à un homologue iBGP (ou eBGP) à capacité BGPsec, l'attribut BGPsec_PATH NE DEVRAIT PAS être retiré même dans le cas où le message BGPsec UPDATE n'a pas réussi à la validation. (Voir à la Section 5 plus d'informations sur la validation et à la Section 8 sur les ramifications pour la sécurité de la suppression des signatures BGPsec.)

Tous les messages BGPsec UPDATE DOIVENT se conformer à la taille maximum de message de BGP. Si le message résultant excède la taille maximum de message, les lignes directrices du paragraphe 9.2 de la [RFC4271] DOIVENT être suivies.

4.2 Construction de l'attribut BGPsec_PATH

Quand un locuteur BGPsec reçoit un message BGPsec UPDATE contenant un attribut BGPsec_PATH (avec une ou plusieurs signatures) provenant d'un homologue (interne ou externe) il peut choisir de propager l'annonce de chemin en l'envoyant à ses autres homologues (internes ou externes). Quand il envoie l'annonce de chemin à un locuteur BGPsec homologue interne, l'attribut BGPsec_PATH NE DEVRA PAS être modifié. Quand il envoie l'annonce de chemin à un locuteur BGPsec homologue externe, les procédures suivantes sont utilisées pour former ou mettre à jour l'attribut BGPsec_PATH.

Pour générer l'attribut BGPsec_PATH sur le message UPDATE sortant, le locuteur BGPsec génère d'abord un nouveau segment Secure_Path. Noter que si le locuteur BGPsec n'est pas l'AS d'origine et qu'il y a un attribut BGPsec_PATH existant, alors le locuteur BGPsec ajoute son nouveau segment Secure_Path (le place en première position) devant le Secure_Path existant.

Le numéro d'AS dans ce segment Secure_Path DOIT correspondre au numéro d'AS dans le champ Subject du certificat de routeur RPKI qui va être utilisé pour vérifier la signature numérique construite par ce locuteur BGPsec (voir au paragraphe 3.1.1 dans la [RFC8209] et la [RFC6487]).

Le champ pCount du segment Secure_Path est normalement réglé à la valeur 1. Cependant, un locuteur BGPsec peut régler le champ pCount à une valeur supérieure à 1. Régler le champ pCount à une valeur supérieure à 1a même sémantique que de répéter plusieurs fois un numéro d'AS dans le AS_PATH d'un message non BGPsec UPDATE (par exemple, pour les besoins d'ingénierie du trafic).

Pour prévenir une charge de traitement inutile dans la validation des signatures BGPsec, un locuteur BGPsec NE DEVRAIT PAS produire plusieurs segments consécutifs de Secure_Path avec le même numéro d'AS. Cela signifie que pour réaliser la sémantique de l'ajout du même numéro d'AS k fois, un locuteur BGPsec DEVRAIT produire un seul segment Secure_Path – avec un pCount de k -- et un seul segment Signature correspondant.

Un serveur de chemin qui participe au plan de contrôle BGP mais n'agit pas comme AS de transit dans le plan de données peut choisir de régler pCount à 0. Cette option permet au serveur de chemin de participer à BGPsec et d'obtenir les garanties de sécurité associées sans augmenter la longueur du chemin d'AS. (Noter que les locuteurs BGPsec calculent la longueur du chemin d'AS en additionnant les valeurs de pCount dans l'attribut BGPsec_PATH ; voir la Section 5.) Cependant, quand un serveur de chemin règle la valeur de pCount à 0, il insère quand même son numéro d'AS dans le segment Secure_Path, car cette information est nécessaire pour valider la signature ajoutée par le serveur de chemin. Voir dans la [RFC8206] une discussion sur le réglage de pCount à 0 pour faciliter la migration des numéros d'AS. Aussi, voir au paragraphe 4.3 l'utilisation de pCount=0 dans le contexte d'une confédération d'AS. Voir au paragraphe 7.2 des lignes directrices de fonctionnement pour configurer un routeur BGPsec à régler pCount=0 et/ou accepter pCount=0 provenant d'un homologue.

Ensuite, le locuteur BGPsec génère un ou deux blocs de signature. Normalement, un locuteur BGPsec va utiliser seulement une suite d'algorithmes et donc créer seulement un seul bloc de signature dans l'attribut BGPsec_PATH. Cependant, pour assurer la rétro compatibilité durant une période de transition d'une suite "actuelle" d'algorithmes à une "nouvelle", il va être nécessaire de générer des messages UPDATE qui contiennent un bloc de signature pour les deux suites d'algorithmes "actuelle" et "nouvelle" (voir au paragraphe 6.1).

Si le message BGPsec UPDATE reçu contient deux blocs de signature et si le locuteur BGPsec prend en charge les deux suites d'algorithmes correspondantes, le nouveau message UPDATE généré par le locuteur BGPsec DOIT alors inclure les deux blocs de signature. Si le message BGPsec UPDATE reçu contient deux blocs de signature et si le locuteur BGPsec ne prend en charge qu'une des deux suites d'algorithmes correspondantes, alors le locuteur BGPsec DOIT supprimer le bloc de signature correspondant à la suite d'algorithmes qu'il ne comprend pas. Si le locuteur BGPsec ne prend en charge les suites d'algorithmes dans aucun des blocs de signature contenus dans le message UPDATE reçu, alors le locuteur BGPsec NE DOIT PAS propager l'annonce de chemin avec l'attribut BGPsec_PATH. (C'est-à-dire, si il choisit de propager cette annonce de chemin, il DOIT le faire comme un message BGP UPDATE non signé. Voir au paragraphe 4.4 plus d'informations sur la conversion en un message BGP UPDATE non signé.)

Noter que dans le cas où le BGPsec_PATH a deux blocs de signature (correspondant à des suites d'algorithmes différentes) l'algorithme de validation (voir au paragraphe 5.2) estime qu'un message BGPsec UPDATE est "valide" si il y a au moins une suite d'algorithmes prise en charge (et le bloc de signature correspondant) qui est réputé "valide". Cela signifie qu'un message BGPsec UPDATE "valide" peut contenir un bloc de signature qui n'est pas réputé "valide" (par exemple, il contient des signatures que BGPsec n'a pas reussi à vérifier). Néanmoins, de tels blocs de signature NE DOIVENT PAS être supprimés. (Voir à la Section 8 une discussion des ramifications de sécurité de ce choix de conception.)

Pour chaque bloc de signature correspondant à une suite d'algorithmes que le locuteur BGPsec ne prend pas en charge, le locuteur BGPsec DOIT ajouter un nouveau segment de signature au bloc de signature. Ce segment de signature est ajouté à la liste des segments de signature (placé en première position) afin que la liste des segments de signature apparaisse dans le même ordre que les segments Secure_Path correspondants. Le locuteur BGPsec remplit les champs de ce nouveau segment de signature comme suit :

Le champ Identifiant de clé sujette dans le nouveau segment est rempli avec l'identifiant contenu dans l'extension Identifiant de clé sujette du certificat de routeur RPKI correspondant au locuteur BGPsec [RFC8209]. Cet identifiant de clé sujette va être utilisé par les receveurs de l'annonce de chemin pour identifier le certificat approprié à utiliser pour vérifier la signature.

Le champ Signature dans le nouveau segment contient une signature numérique qui lie le préfixe et l'attribut BGPsec_PATH au certificat de routeur RPKI correspondant au locuteur BGPsec. La signature numérique est calculée

comme suit :

- Pour être clairs, numérotons le chemin sécurisé et les segments de signature correspondants de 1 à N, comme suit. Soit le segment 1 de chemin sécurisé et le segment 1 de signature les segments produits par l'AS d'origine. Soit le segment 2 du chemin sécurisé et le segment 2 de signature les segments ajoutés par l'AS suivant après l'origine. Continuons cette méthode de numérotation, et à la fin, soit le segment N de chemin sécurisé et le segment N de signature ceux qui sont ajoutés par l'AS en cours. L'AS en cours (Nième AS) signe et transmet le message UPDATE au prochain AS (c'est-à-dire, l'AS (N+1)) dans la chaîne des AS qui forment le chemin d'AS.

- Afin de construire la signature numérique pour le segment de signature N (le segment de signature produit par l'AS en cours) construisons d'abord la séquence d'octets à hacher comme montré à la Figure 8. Cette séquence d'octets comporte toutes les données que le Nième AS atteste comme ajoutées à sa signature numérique dans le message UPDATE qui est transmis à un locuteur BGPsec dans l'AS (N+1). (Pour les raisons de la conception du choix de la structure spécifique de la Figure 8, voir dans [Borchert].)

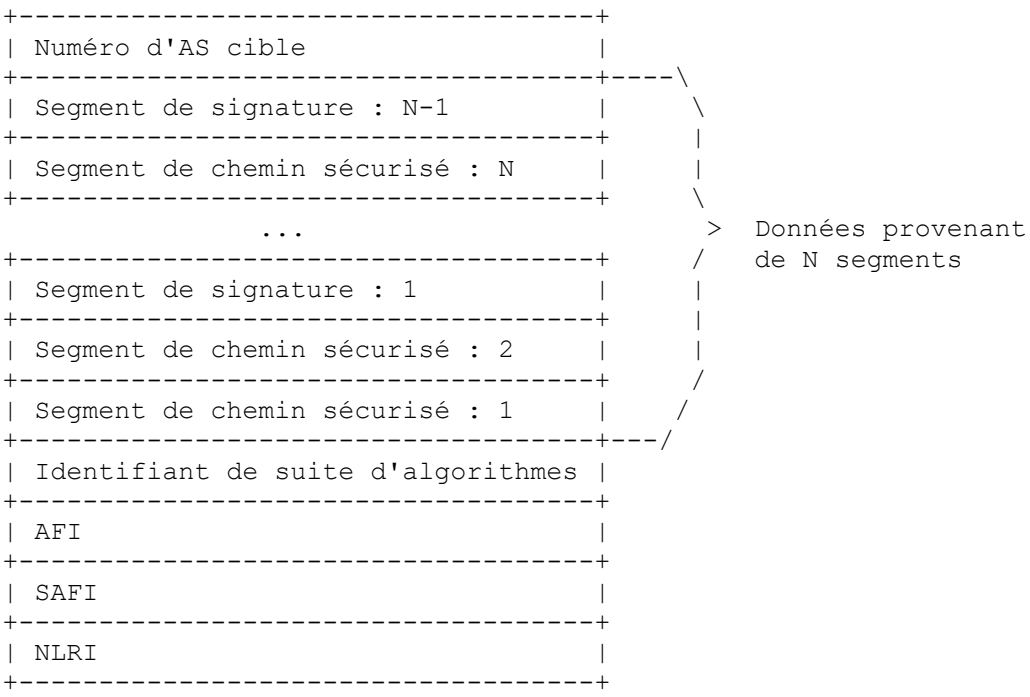

Figure 8 : Séquence d'octets à hacher

Les éléments de cette séquence (Figure 8) DOIVENT être ordonnés exactement comme montrés. Le "Numéro d'AS cible"' est l'AS à qui le locuteur BGPsec entend envoyer le message UPDATE. (Noter que le "Numéro d'AS cible" est le numéro d'AS annoncé par l'homologue dans le message OPEN de la session BGP au sein de laquelle le message UPDATE est envoyé.) Les segments de chemin sécurisé et de signature (1 à N-1) sont obtenus de l'attribut BGPsec_PATH. Finalement, les champs Identifiant de famille d'adresse (AFI), Identifiant suivant de famille d'adresse (SAFI), et NLRI sont obtenus de l'attribut MP_REACH_NLRI [RFC4760]. De plus, dans le champ Préfixe au sein du champ NLRI (voir la Section 5 de la [RFC4760]) tous les bits en queue DOIVENT être réglés à 0 dans la construction de cette séquence.

- Appliquer à cette séquence d'octets (de la Figure 8) l'algorithme de résumé (pour la suite d'algorithmes de ce bloc de signature) pour obtenir une valeur de résumé.

- Appliquer à cette valeur de résumé l'algorithme de signature (pour la suite d'algorithmes de ce bloc de signature) pour obtenir la signature numérique. Remplir ensuite le champ Signature (Figure 7) avec cette signature numérique.

Le champ Longueur de signature (Figure 7) est rempli avec la longueur (en octets) de la valeur dans le champ Signature.

4.3 Instructions de traitement pour membres de confédération

Les membres de confédérations d'AS [RFC5065] DOIVENT de plus suivre les instructions de ce paragraphe pour traiter les messages BGPsec UPDATE.

Quand a locuteur BGPsec dans une confédération d'AS reçoit un message BGPsec UPDATE d'un homologue qui est externe à la confédération et choisit de propager le message UPDATE au sein de la confédération, il ajoute d'abord une signature signée de son propre numéro de membre d'AS (c'est-à-dire, le "Numéro d'AS cible" est le numéro de membre d'AS du locuteur BGPsec). Dans ce message UPDATE modifié en interne, le nouveau segment de chemin sécurisé ajouté contient le numéro d'AS public (c'est-à-dire, l'identifiant de confédération) la valeur de pCount du segment est réglée à 0, et le fanion Confed_Segment est réglé à 1. Régler pCount=0 dans ce cas aide à s'assurer que la longueur du chemin d'AS n'est pas augmentée inutilement. La nouvelle signature ajoutée est générée en utilisant une clé privée correspondant au numéro d'AS public de la confédération. Le locuteur BGPsec propage le message UPDATE modifié à ses homologues au sein de la confédération.

Toutes les modifications de BGPsec_PATH mentionnées ci-dessous dans le contexte de la propagation du message UPDATE au sein de la confédération sont en plus de la modification décrite ci-dessus (c'est-à-dire, avec pCount=0).

Quand un locuteur BGPsec envoie un message BGPsec UPDATE à un homologue qui appartient à son propre AS membre, le membre de la confédération NE DEVRA PAS modifier l'attribut BGPsec_PATH. Quand un locuteur BGPsec envoie un message BGPsec UPDATE à un homologue qui est dans la même confédération mais dans un AS membre différent, le locuteur BGPsec met son numéro de membre d'AS dans le champ Numéro d'AS du segment Secure_Path qu'il ajoute au message BGPsec UPDATE. De plus, dans ce cas, le membre d'AS qui génère le segment Secure_Path règle le fanion Confed_Segment à 1. De plus, la signature est générée avec une clé privée qui correspond au numéro de membre d'AS du locuteur BGPsec. (Note : dans ce document, l'échange de trafic intra membre d'AS est considéré comme iBGP, et l'échange de trafic inter membres d'AS est considéré comme eBGP. Ce dernier est aussi appelé confédération eBGP.)

Au sein d'une confédération, la vérification des signatures BGPsec ajoutées par les autres membres de la confédération est facultative. Noter que si une confédération choisit de ne pas vérifier les signatures numériques au sein de la confédération, BGPsec n'est alors pas capable de fournir d'assurances sur l'intégrité des numéros de membre d'AS placés dans les segments de chemin sécurisé où le fanion Confed_Segment est réglé à 1.

Quand un membre de confédération reçoit un message BGPsec UPDATE d'un homologue au sein de la confédération et le propage à un homologue en dehors de la confédération, il doit supprimer tous les segments de chemin sécurisé ajoutés par les membres de la confédération ainsi que les segments de signature correspondants. Pour ce faire, le membre de la confédération qui propage le chemin en dehors de la confédération fait ce qui suit :

- D'abord, en commençant par le segment de chemin sécurisé dont l'ajout est le plus récent, supprimer tous les segments de chemin sécurisé consécutifs qui ont le fanion Confed_Segment réglé à 1. Arrêter ce processus une fois qu'un segment Secure_Path qui a son fanion Confed_Segment réglé à 0 est atteint. Garder le compte du nombre de segments retirés de cette façon.

- Ensuite, en commençant par le segment de signature ajouté le plus récemment, supprimer un nombre de segments de signature égal au nombre de segments de chemin sécurisé supprimés à l'étape précédente. (C'est-à-dire que, retirer les K segments de signature les plus récemment ajoutés, où K est le nombre de segments Secure_Path supprimés à l'étape précédente.)

- Finalement, ajouter un segment Secure_Path contenant, dans le champ AS, l'identifiant de confédération d'AS (le numéro d'AS public de la confédération) ainsi qu'un segment de signature correspondant. Noter que tous les champs autres que le champ AS sont remplis conformément au paragraphe 4.2.

Finalement, comme expliqué ci-dessus, une confédération d'AS PEUT facultativement décider que ses membres ne vont pas vérifier les signatures numériques ajoutées par les membres. Dans une telle confédération, quand un locuteur BGPsec fait fonctionner l'algorithme du paragraphe 5.2, le locuteur BGPsec, durant le processus de vérification des signatures, vérifie d'abord si le fanion Confed_Segment dans un segment Secure_Path est réglé à 1. Si le fanion est réglé à 1, le locuteur BGPsec saute la vérification pour la signature correspondante et passe immédiatement au segment Secure_Path suivant. Noter que comme spécifé au paragraphe 5.2, c'est une erreur quand un locuteur BGPsec reçoit d'un homologue qui n'est pas dans la même confédération d'AS un message BGPsec UPDATE contenant un fanionConfed_Segment réglé à 1.

4.4 Reconstruction de l'attribut AS_PATH

Les messages BGPsec UPDATE ne contiennent pas l'attribut AS_PATH. Cependant, l'attribut AS_PATH peut être reconstruit à partir de l'attribut BGPsec_PATH. Ceci est nécessaire dans le cas où une annonce de chemin est reçue via un message BGPsec UPDATE et ensuite propagée à un homologue via un message non BGPsec UPDATE (par exemple, parce que le dernier homologue ne prend pas en charge BGPsec). Noter qu'il peut y avoir d'autres cas où une mise en œuvre trouve utile d'effectuer cette reconstruction. Avant de tenter de reconstruire un AS_PATH pour les besoins de la transmission d'un message non signé (non BGPsec) UPDATE à un homologue, un locuteur BGPsec DOIT effectuer les vérifications d'intégrité de base mentionnées au paragraphe 5.2 pour s'assurer que le message BGPsec UPDATE reçu est correctement formé.

L'attribut AS_PATH peut être construit à partir de l'attribut BGPsec_PATH comme suit. En commençant par un attribut AS_PATH blanc, on traite les segments de chemin sécurisé dans l'ordre du plus anciennement ajouté (correspondant à l'origine) au plus récemment ajouté. Pour chaque segment Secure_Path, effectuer les étapes suivantes :

1. Si le segment Secure_Path a pCount=0, ne rien faire (c'est-à-dire, passer au traitement du segment Secure_Path suivant).

2. Si le segment Secure_Path a un pCount supérieur à 0 et si le fanion Confed_Segment est à 1, chercher le segment le plus récemment ajouté dans AS_PATH.
   * Dans le cas où AS_PATH est blanc ou si le segment le plus récemment ajouté est du type AS_SEQUENCE, ajouter (mettre au début de l'AS_PATH) un nouveau segment d'AS_PATH de type AS_CONFED_SEQUENCE. Ce segment de type AS_CONFED_SEQUENCE devra contenir un nombre d'éléments égal au champ pCount dans le segment Secure_Path en cours. Chacun de ces éléments devra être le numéro d'AS contenu dans le segment Secure_Path en cours. (C'est-à-dire que, si le champ pCount est X, alors le segment de type AS_CONFED_SEQUENCE contient X copies du champ Numéro d'AS du segment Secure_Path.)
   * Dans le cas où le segment le plus récemment ajouté dans le AS_PATH est du type AS_CONFED_SEQUENCE, ajouter alors (mettre en tête du segment) un nombre d'éléments égal au champ pCount dans le segment Secure_Path en cours. La valeur de chaque élément devra être le numéro d'AS contenu dans le segment Secure_Path en cours. (C'est-à-dire que, si le champ pCount est X, ajouter alors X copies du champ Numéro d'AS du segment Secure_Path à la AS_CONFED_SEQUENCE existante.)

3. Si le segment Secure_Path a un pCount supérieur à 0 et si le fanion Confed_Segment est à 0, chercher le segment le plus récemment ajouté dans AS_PATH.
   * Si l'AS_PATH est blanc ou si le segment le plus récemment ajouté est du type AS_CONFED_SEQUENCE, ajouter (en tête de l'AS_PATH) un nouveau segment AS_PATH de type AS_SEQUENCE. Ce segment de type AS_SEQUENCE devra contenir un nombre d'éléments égal au champ pCount dans le segment Secure_Path en cours. Chacun de ces éléments devra être le numéro d'AS contenu dans le segment Secure_Path en cours. (C'est-à-dire que, si le champ pCount est X, le segment de type AS_SEQUENCE contient alors X copies du champ Numéro d'AS du segment Secure_Path.)
   * Si le segment le plus récemment ajouté dans l'AS_PATH est du type AS_SEQUENCE, ajouter alors (au début du segment) un nombre d'éléments égal au champ pCount dans le segment Secure_Path en cours. La valeur de chacun de ces éléments devra être le numéro d'AS contenu dans le segment Secure_Path en cours. (C'est-à-dire que, si le champ pCount est X, ajouter alors X copies du champ Numéro d'AS du segment Secure_Path à l'AS_SEQUENCE existante.)

Au titre de la procédure décrite ci-dessus, les actions supplémentaires suivantes sont effectuées afin de ne pas excéder les limitations de taille de AS_SEQUENCE et AS_CONFED_SEQUENCE. Lors de l'ajout du prochain segment Secure_Path (avec ses ajouts en tête, si il en est) au AS_PATH en cours d'assemblage, si cela causerait le dépassement par l'AS_SEQUENCE (ou AS_CONFED_SEQUENCE) en cours de la limite de 255 numéros d'AS par segment [RFC4271] [RFC5065], le locuteur BGPsec devrait alors suivre les recommandations de la [RFC4271] et de la [RFC5065] de créer un autre segment du même type (AS_SEQUENCE ou AS_CONFED_SEQUENCE) et de continuer de le remplir.

Finalement, un cas particulier de reconstruction de AS_PATH est quand l'attribut BGPsec_PATH est absent. Comme expliqué au paragraphe 4.1, quand un locuteur BGPsec génère un préfixe et l'envoie à un homologue iBGP à capacité BGPsec, le BGPsec_PATH n'est pas joint. Donc, quand il est reçu d'un homologue iBGP à capacité BGPsec, pas d'attribut BGPsec_PATH dans un message BGPsec UPDATE est équivalent à un AS_PATH vide [RFC4271].

5. Traitement d'un message BGPsec UPDATE reçu

À réception d'un message BGPsec UPDATE d'un homologue externe (eBGP) un locuteur BGPsec DEVRAIT valider le message pour déterminer l'authenticité des informations de chemin contenues dans l'attribut BGPsec_PATH. Normalement, un locuteur BGPsec va aussi souhaiter effectuer la validation de l'origine (voir la [RFC6483] et la [RFC6811]) sur un message BGPsec UPDATE entrant, mais une telle validation est indépendante de la validation décrite dans cette section.

Le paragraphe 5.1 donne une vue d'ensemble de la validation BGPsec, et le paragraphe 5.2 donne un algorithme spécifique pour effectuer une telle validation. (Noter qu'une mise en œuvre n'a pas besoin de suivre l'algorithme spécifique du paragraphe 5.2 pour autant que le comportement d'entrée/sortie de la validation soit identique à celui de l'algorithme du paragraphe 5.2.) Durant des conditions exceptionnelles (par exemple, le locuteur BGPsec reçoit un nombre incroyable de messages UPDATE en une seule fois) un locuteur BGPsec PEUT temporairement différer la validation des messages BGPsec UPDATE entrants. Le traitement de tels messages BGPsec UPDATE, dont la validation a été différée, est une affaire de politique locale. Cependant, une mise en œuvre DEVRAIT s'assurer que le différé de la validation et l'état des messages différés sont visibles à l'opérateur.

La validité des messages BGPsec UPDATE est fonction de l'état RPKI en cours. Quand un locuteur BGPsec apprend que l'état RPKI a changé (par exemple, à partir d'une antémémoire de validation RPKI via le protocole de routeur RPKI [RFC8210]) le locuteur BGPsec DOIT relancer la validation sur tous les messages UPDATE affectés mémorisés dans son Adj-RIB-In [RFC4271]. Par exemple, quand un certain certificat de routeur RPKI cesse d'être valide (par exemple, il arrive à expiration ou est révoqué) tous les messages UPDATE contenant une signature dont le SKI correspond au SKI dans ce certificat DOIVENT être réévalués pour déterminer si ils sont encore valides. Si cette réévaluation détermine que l'état de validité d'un message UPDATE a changé, alors, selon la politique locale, il peut être nécessaire de relancer le choix du meilleur chemin.

Les messages BGPsec UPDATE ne contiennent pas d'attribut AS_PATH. Le chemin sécurisé contient les informations de chemin d'AS pour le message BGPsec UPDATE. Donc, un locuteur BGPsec DOIT utiliser les informations de chemin d'AS dans Secure_Path dans tous les cas où il aurait autrement utilisé les nformations de chemin d'AS dans l'attribut AS_PATH. La seule exception à cette règle est quand les informations de chemin d'AS doivent être mises à jour afin de propager un chemin vers un homologue (auquel cas le locuteur BGPsec suit les instructions de la Section 4). Le paragraphe 4.4 fournit un algorithme pour construire un attribut AS_PATH à partir d'un attribut BGPsec_PATH. Chaque fois que l'utilisation d'informations de chemin d'AS est invoquée (par exemple, pour la détection de boucles ou l'utilisation de la longueur de chemin d'AS dans le choix du meilleur chemin) le comportement visible extérieurement de la mise en œuvre devra être le même que si la mise en œuvre avait appliqué l'algorithme du paragraphe 4.4 et utilisé l'attribut AS_PATH résultant comme pour un message non BGPsec UPDATE.

5.1 Généralités sur la validation de BGPsec

La validation d'un message BGPsec UPDATE utilise des données provenant des certificats de routeur RPKI. En particulier, il est nécessaire que le receveur ait accès aux données suivantes obtenues d'un certificat valide de routeur RPKI : le numéro d'AS, la clé publique, et l'identifiant de clé sujette provenant de chaque certificat valide de routeur RPKI.

Noter que le locuteur BGPsec pourrait effectuer de lui-même la validation des certificats de routeur RPKI et extraire les données requises, ou qu'il pourrait recevoir les mêmes données d'une antémémoire de confiance qui effectue la validation RPKI au nom des locuteurs BGPsec (ou d'un sous ensemble de ceux-ci). (Par exemple, l'antémémoire de confiance pourrait délivrer les informations de validité nécessaires au locuteur BGPsec en utilisant l'unité de données de protocole pour le protocole de routeur RPKI [RFC8210].)

Pour valider un message BGPsec UPDATE contenant l'attribut BGPsec_PATH, le receveur effectue les étapes de validation spécifiées au paragraphe 5.2. La procédure de validation résulte en un des deux états : "valide" et "non valide".

On s'attend à ce que le résultat de la procédure de validation soit utilisé comme entrée du choix de chemin BGP. Ceci dit, le choix de chemin BGP, et donc, le traitement des états de validation, est une affaire de politique locale et est traité en utilisant des mécanismes de politique locale. Les mises en œuvre DEVRAIENT permettre aux opérateurs de régler cette politique locale session par session. (C'est-à-dire que on s'attend à ce que des opérateurs choisissent de traiter l'état de validation BGPsec différemment pour les messages UPDATE reçus sur des sessions BGP différentes.)

La validation BGPsec a seulement besoin d'être effectuée à la bordure eBGP. L'état de validation d'un message BGP UPDATE signé/non signé PEUT être porté via iBGP à partir d'un routeur bordure d'entrée à un routeur bordure de sortie

via un mécanisme, qui dépend de la politique locale au sein d'un AS. Comme exposé à la Section 4, quand un locuteur BGPsec choisit de transmettre un message BGPsec UPDATE (syntaxiquement correct) il DEVRAIT être transmis avec ses attributs BGPsec_PATH intacts (sans considération de l'état de validation du message UPDATE). Fondé entièrement sur la politique locale, un routeur de sortie qui reçoit un message BGPsec UPDATE provenant de son propre AS PEUT choisir d'effectuer sa propre validation.

5.2 Algorithme de validation

Ce paragraphe spécifie un algorithme pour la validation des messages BGPsec UPDATE. Une mise en œuvre conforme DOIT inclure un algorithme de validation de mise à jour BGPsec qui soit fonctionnellement équivalent au comportement visible extérieurement de cet algorithme.

D'abord, le receveur d'un message BGPsec UPDATE effectue une vérification pour s'assurer que le message est correctement formé. Les erreurs de syntaxe et les violations du protocole sont vérifiées. L'attribut BGPsec_PATH DOIT être présent quand un message BGPsec UPDATE est reçu d'un homologue BGPsec externe (eBGP) et aussi quand un tel message UPDATE est propagé à un homologue BGPsec interne (iBGP) (voir au paragraphe 4.2). Les vérifications d'erreurs spécifiées au paragraphe 6.3 de la [RFC4271] sont effectuées, sauf que pour les messages BGPsec UPDATE, les vérifications sur l'attribut AS_PATH ne s'appliquent pas et qu'à la place on effectue les vérifications suivantes sur l'attribut BGPsec_PATH :

1. Vérifier que l'attribut BGPsec_PATH entier est syntaxiquement correct (conforme aux spécifications de ce document).

2. Vérifier que le numéro d'AS dans le segment Secure_Path le plus récemment ajouté (c'est-à-dire, celui qui correspond à l'homologue eBGP de qui le message UPDATE a été reçu) correspond au numéro d'AS de cet homologue comme spécifié dans le message BGP OPEN. (Note : cette vérification n'est effectuée que à un routeur BGPsec d'entrée lorsque le message UPDATE est d'abord reçu d'un AS homologue.)

3. Vérifier que chaque bloc de signature contient un segment de signature pour chaque segment de chemin sécurisé dans la portion Secure_Path de l'attribut BGPsec_PATH. (Noter que chaque bloc de signature DOIT être vérifié dans son intégralité pour s'assurer qu'il est bien formé, même si le processus de validation peut se terminer avant que toutes les signatures soient cryptographiquement vérifiées.)

4. Vérifier que le message UPDATE ne contient pas d'attribut AS_PATH.

5. Si le message UPDATE a été reçu d'un homologue BGPsec qui n'est pas membre de la confédération d'AS du locuteur BGPsec, s'assurer qu'aucun des segments Secure_Path ne contient un champ Fanions avec le fanion Confed_Segment réglé à 1.

6. Si le message UPDATE a été reçu d'un homologue BGPsec qui est membre de la confédération d'AS du locuteur BGPsec, s'assurer que le segment Secure_Path correspondant à cet homologue contient un champ Fanions avec le fanion Confed_Segment réglé à 1.

7. Si le message UPDATE a été reçu d'un homologue dont il n'est pas prévu qu'il règle pCount=0 (voir les paragraphes 4.2 et 4.3) s'assurer que le champ pCount dans le plus récent segment Secure_Path ajouté n'est pas égal à 0. (Note : voir au paragraphe 7.2 les lignes directrices sur la configuration de routeur relative à ce point.)

8. En utilisant l'équivalent de AS_PATH correspondant au Secure_Path dans le message UPDATE (voir au paragraphe 4.4) vérifier que le numéro d'AS local n'est pas présent dans le chemin d'AS (c'est-à-dire, éliminer toute boucle d'AS).

Si une de ces vérifications échoue, il y a une erreur dans l'attribut BGPsec_PATH. Les locuteurs BGPsec DOIVENT traiter toutes les erreurs de syntaxe ou de protocole dans l'attribut BGPsec_PATH en utilisant l'approche de "traiter comme suppression" définie dans la [RFC7606]. (Note : comme le numéro d'AS d'un serveur de chemin transparent apparaît bien dans le Secure_Path avec pCount=0, le serveur de chemin PEUT vérifier si son AS local figure sur la liste dans le Secure_Path, et cette vérification PEUT être incluse dans la vérification de détection de boucle mentionnée ci-dessus.)

Ensuite, le locuteur BGPsec examine les blocs de signature dans l'attribut BGPsec_PATH. Un bloc de signature correspondant à une suite d'algorithmes que le locuteur BGPsec ne prend pas en charge n'est pas considéré dans le processus de validation. Si il n'y a pas de bloc de signature correspondant à une suite d'algorithmes que le locuteur BGPsec prend en charge, alors afin de considérer le message UPDATE dans le processus de choix de chemin, le locuteur BGPsec DOIT supprimer le ou les blocs de signature, reconstruire le AS_PATH à partir de Secure_Path (voir au paragraphe 4.4), et

traiter le message UPDATE comme si il avait été reçu comme message BGP UPDATE non signé.

Pour chaque bloc de signature restant (correspondant à une suite d'algorithmes prise en charge par le locuteur BGPsec) le locuteur BGPsec itère à travers les segments de signature dans le bloc de signature, en commençant par le dernier segment ajouté (et en concluant par le segment le plus anciennement ajouté). Noter qu'il y a une correspondance biunivoque entre les segments de signature et les segments de chemin sécurisé au sein de l'attribut BGPsec_PATH. Les étapes suivantes utilisent cette correspondance :

- Étape 1 : Supposons qu'il y a K bonds d'AS dans un attribut BGPsec_PATH reçu à valider. Soit AS(1), AS(2), ..., AS(K+1) qui note la séquence des numéros d'AS depuis l'AS d'origine à l'AS de validation. Soient le segment N du chemin sécurisé et le segment N de signature dans l'attribut BGPsec_PATH qui se réfèrent à ceux correspondant à l'AS(N) (où N = 1, 2, ..., K). Le locuteur BGPsec qui traite et valide l'attribut BGPsec_PATH réside dans l'AS(K+1). Soit le segment de signature N le segment de signature qui est actuellement vérifié.

- Étape 2 : Localiser la clé publique nécessaire pour vérifier la signature (dans le segment de signature actuel). Pour ce faire, consulter les données du certificat de routeur RPKI valide et chercher tous les triplets valides (Numéro d'AS, Clé publique, Identifiant de clé sujette) dans lesquels l'AS correspond au numéro d'AS dans le segment Secure_Path correspondant. Parmi ces triplets qui correspondent au numéro d'AS, vérifier si il y a un SKI qui correspond à la valeur dans le champ Identifiant de clé sujette du segment de signature. Si cette recherche ne trouve pas une telle valeur de SKI correspondante, marquer alors le bloc de signature entier comme "non valide" et passer au prochain bloc de signature.

- Étape 3 : Calculer la fonction de résumé (pour la suite d'algorithmes concernée) sur les données appropriées.

Afin de vérifier la signature numérique dans le segment de signature N, construire la séquence d'octets à hacher comme montré à la Figure 9 (en utilisant les notations définie à l'étape 1). (Noter que cette séquence est la même que celle utilisée par l'AS(N) qui a créé le segment de signature N (voir au paragraphe 4.2 et la Figure 8).)

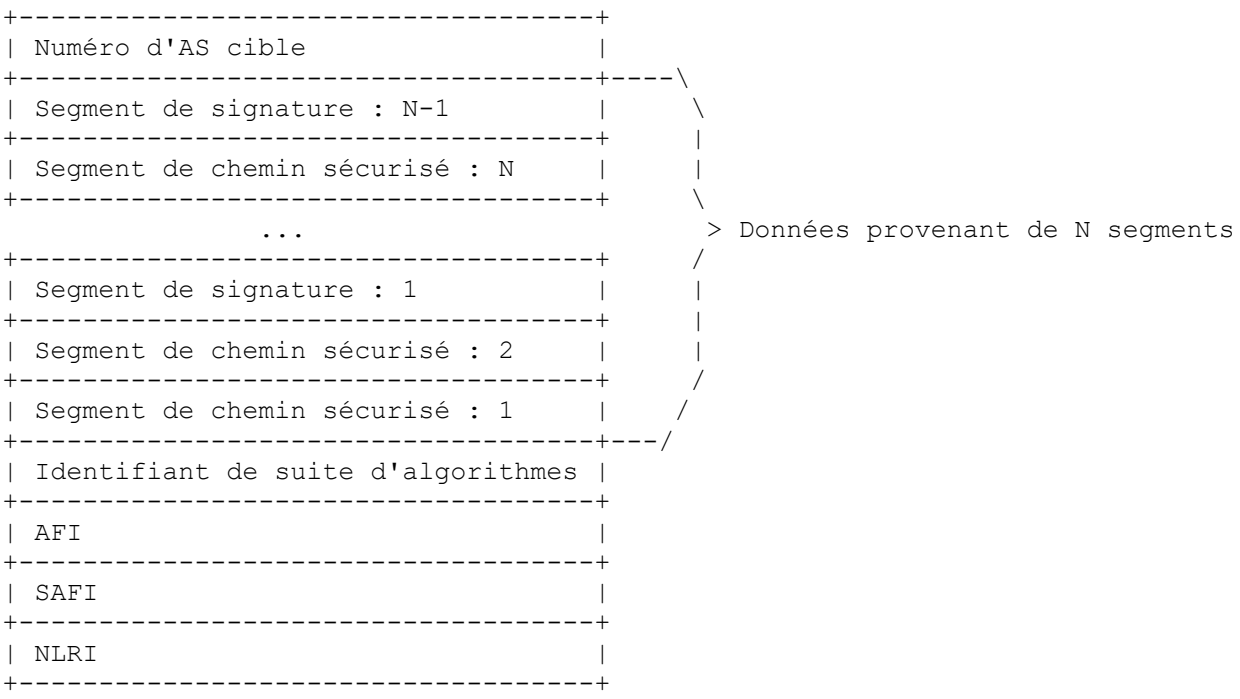

Figure 9 : Séquence d'octets à hacher pour la vérification de signature du segment de signature N ; N = 1,2, ..., K, où K est le numéro de bonds d'AS dans l'attribut BGPsec_PATH

Les éléments de cette séquence (Figure 9) DOIVENT être ordonnés exactement comme montré. Pour le premier segment à traiter (le segment le plus récemment ajouté (c'est-à-dire, N = K) étant donné qu'il y a K bonds dans le chemin sécurisé) le "Numéro d'AS cible" est AS(K+1) le numéro d'AS du locuteur BGPsec qui valide le message UPDATE. Noter que si un locuteur BGPsec utilise plusieurs numéros d'AS (par exemple, le locuteur BGPsec est membre d'une confédération) le numéro d'AS utilisé ici DOIT être le numéro d'AS annoncé dans le message OPEN pour la session BGP sur laquelle le message BGPsec UPDATE a été reçu.

Pour chaque autre segment de signature (N inférieur à K) le "numéro d'AS cible" est AS(N+1), le numéro d'AS dans le segment de chemin sécurisé qui correspond au segment de signature ajouté immédiatement après celui en cours de traitement (c'est-à-dire, dans le segment Secure_Path qui correspond au segment de signature que le valideur a juste fini de traiter).

Le chemin sécurisé et le segment de signature sont obtenus de l'attribut BGPsec_PATH. Les champs AFI, SAFI, et NLRI sont obtenus de l'attribut MP_REACH_NLRI [RFC4760]. De plus, dans le champ Préfixe au sein du champ NLRI (voir la Section 5 de la [RFC4760]) tous les bits en queue DOIVENT être réglés à 0 lors de la construction de cette séquence.

- Étape 4 : utilisation de l'algorithme de validation de signature (pour la suite d'algorithmes concernée) pour vérifier la signature dans le segment actuel. C'est-à-dire qu'on invoque l'algorithme de validation de signature sur les trois entrées suivantes : la valeur du champ Signature dans le segment en cours, la valeur du résumé calculé dans l'étape 3 ci-dessus, et la clé publique obtenue des données valides de RPKI dans l'étape 2. Si l'algorithme de validation de signature détermine que la signature est invalide, marquer alors le bloc de signature entier comme "non valide" et passer au bloc de signature suivant. Si l'algorithme de validation de signature détermine que la signature est valide, continuer le traitement des segments de signature (au sein du bloc de signature en cours.

Si tous les segments de signature au sein d'un bloc de signature réussissent à la validation (c'est-à-dire, si tous les segments sont traités et le bloc de signature n'a pas encore été marqué "non valide") alors le bloc de signature est marqué "valide".

Si au moins un bloc de signature est marqué "valide", alors l'algorithme de validation se termine et le message BGPsec UPDATE est réputé "valide". (C'est-à-dire que, si un message BGPsec UPDATE contient deux blocs de signature, le message UPDATE est réputé "valide" si le premier bloc de signature est marqué "valide" OU si le second bloc de signature est marqué "valide".)

6. Algorithmes et extensibilité

6.1 Considérations sur les suites d'algorithmes

Noter qu'il n'y a actuellement pas de prise en charge de la négociation bilatérale (avec les capacités BGP) entre les homologues BGPsec pour utiliser une suite d'algorithmes (de résumé et de signature) particulière. C'est parce que la suite d'algorithmes utilisée par l'envoyeur d'un message BGPsec UPDATE DOIT être comprise non seulement par l'homologue à qui il envoie directement le message mais aussi par tous les locuteurs BGPsec à qui l'annonce de chemin est éventuellement propagée. Donc, le choix d'une suite d'algorithmes ne peut pas être une affaire locale négociée par les homologues BGP mais plutôt doit être coordonnée à travers l'Internet.

À cette fin, la [RFC8208] spécifie une suite d'algorithmes "courants" d'utilisation obligatoire pour tous les locuteurs BGPsec.

Il est prévu que à l'avenir, la [RFC8208] ou son successeur sera mise à jour pour spécifier une transition de la suite d'algorithmes "actuelle" à une "nouvelle" suite d'algorithmes. Durant la période de transition, tous les messages BGPsec UPDATE DEVRAIENT utiliser simultanément les deux suites d'algorithmes "actuelle" et "nouvelle". (Noter que les Sections 3 et 4 spécifient comment l'attribut BGPsec_PATH peut contenir les signatures, en parallèle, pour deux suites d'algorithmes.) Une fois la transition achevée, l'utilisation du vieil algorithme "actuel" va être déconseillée, l'utilisation du "nouvel'" algorithme va être obligatoire, et une suite d'algorithmes "encore plus nouvelle" suivante pourra être spécifiée comme "de mise en œuvre recommandée". Une fois la transition achevée avec succès de cette manière, les locuteurs BGPsec DEVRAIENT inclure seulement un bloc de signature (correspondant au "nouvel" algorithme).

6.2 Considérations sur la taille de SKI

Selon la méthode de génération des identifiants de clés [RFC7093], la taille du SKI dans un certificat de routeur RPKI peut varier. Le champ SKI dans l'attribut BGPsec_PATH a une taille fixe de 20 octets (voir la Figure 7). Si le SKI fait plus de 20 octets, on utilise alors les 20 octets de gauche du SKI (en excluant l'étiquette et la longueur) [RFC7093]. Si la valeur du SKI fait moins de 20 octets, on bourre le SKI (excluant l'étiquette et la longueur) sur la droite (les octets de moindre poids) avec des octets de valeur "0".

6.3 Considérations d'extensibilité

Ce paragraphe discute des changements potentiels à BGPsec qui exigeraient des changements substantiels du traitement de BGPsec_PATH et nécessiteraient une nouvelle version de BGPsec. Les exemples de tels changements incluent :

- o un nouveau type d'algorithme de signature qui produit des signatures de longueur variable,
- o un nouveau type d'algorithme de signature pour lequel le nombre de signatures dans le bloc de signature n'est pas égal au nombre d'AS dans le chemin sécurisé (par exemple, des signatures agrégées),
- o des changements des données qui sont protégées par les signatures BGPsec (par exemple, des attributs autres que le chemin d'AS)

Dans le cas où un tel changement à BGPsec se révélerait souhaitable, il est prévu qu'une nouvelle version de BGPsec serait créée et que cette version de BGPsec spécifierait un nouvel attribut de chemin BGP – appelons le "BGPsec_PATH_Two" – qui serait conçu pour s'accommoder des changements désirés à BGPsec. Dans ce cas, la [RFC8208] ou son successeur serait mise à jour pour spécifier les suites d'algorithmes appropriées pour la nouvelle version de BGPsec.

À ce point, une transition commencerait, analogue à la transition d'algorithme discutée au paragraphe 6.1. Durant la période de transition, tous les locuteurs BGPsec DEVRAIENT simultanément inclure à la fois l'attribut BGPsec_PATH et le nouvel attribut BGPsec_PATH_Two. Une fois la transition achevée, l'utilisation de BGPsec_PATH serait alors déconseillée, et les locuteurs BGPsec devraient inclure seulement le nouvel atribut BGPsec_PATH_Two. Un tel processus pourrait faciliter une transition à une nouvelle sémantique de BGPsec de façon rétro compatible.

7. Considérations de fonctionnement et de gestion

Certaines questions de fonctionnement et de gestion qui sont étroitement liées à la spécification et au déploiement du protocole BGPsec sont mentionnées ici. Les bonnes pratiques concernant le fonctionnement et le déploiement de BGPsec sont fournies dans la [RFC8207].

7.1 Échec de la négociation de capacités

Le paragraphe 2.2 décrit la négociation requise pour établir une session d'échange de trafic à capacité BGPsec. Non seulement la capacité BGPsec doit être échangée (et faire l'objet d'un accord) mais l'extension BGP multi protocoles [RFC4760] pour le même AFI et la capacité d'AS à quatre octets [RFC6793] DOIT aussi être échangée. L'échec d'une négociation appropriée d'une session BGPsec – à cauxe, par exemple d'une capacité manquante -- peut quand même résulter en l'échange de messages BGP UPDATE (non signés). Il est RECOMMANDÉ qu'une mise en œuvre enregistre l'échec d'une négociation appropriée d'une session BGPsec. Aussi, une mise en œuvre DOIT avoir la capacité d'empêcher l'établissement d'une session BGP si elle est configurée à utiliser seulement BGPsec.

7.2 Empêcher la mauvaise utilisation de pCount=0

Un homologue qui est un point d'échange Internet (IXP, *Internet Exchange Point*) (c'est-à-dire, un serveur de chemin) avec un AS transparent est supposé régler pCount=0 dans son segment Secure_Path lors de la transmission d'un message UPDATE à un homologue (voir au paragraphe 4.2). Il est clair qu'un tel IXP DOIT configurer son routeur BGPsec à régler pCount=0 dans son segment Secure_Path. Cela signifie aussi qu'un locuteur BGPsec DOIT être configuré de façon à permettre pCount=0 provenant d'un homologue IXP. Deux autres cas où pCount est réglé à 0 sont dans les contextes d'une confédération d'AS (voir au paragraphe 4.3) et d'une migration d'AS [RFC8206]. Dans ces deux cas, pCount=0 est établi et accepté au sein du même AS (bien que l'AS ait deux identités différentes). Noter que si un locuteur BGPsec ne s'attend pas à ce qu'un AS homologue règle son pCount=0 et si un message UPDATE reçu de cet homologue viole cela, le message UPDATE DOIT alors être considéré comme une erreur (voir la liste des vérifications du paragraphe 5.2). Voir au paragraphe 8.4 la discussion des considérations de sécurité concernant pCount=0.

7.3 Terminaison précoce de la vérification de signature

Durant la validation d'un message BGPsec UPDATE, l'accélération des performances du traitement de chemin peut être réalisée en incorporant les observations suivantes. Un message UPDATE est réputé "valide" si au moins un des blocs de signature est marqué "valide" (voir au paragraphe 5.2). Donc, si un message UPDATE contient deux blocs de Signature et si le premier vérifié est trouvé "valide", alors le second bloc de signature n'a pas à être vérifié. Et si le message UPDATE est choisi pour meilleur chemin, alors le locuteur BGPsec ajoute sa signature (générée avec les algorithmes respectifs) pour chacun des deux blocs de signature et transmet le message UPDATE. Aussi, un message BGPsec UPDATE est réputé "non

valide" si au moins une signature dans chaque bloc de signature est invalide. Ce principe peut aussi être utilisé pour des économies de charge de travail du traitement de chemin, c'est-à-dire que la vérification pour un bloc de signature se termine plus tôt quand on rencontre la première signature invalide.

7.4 Algorithmes de signature non déterministes

De nombreux algorithmes de signature sont non déterministes. C'est-à-dire que, de nombreux algorithmes de signature vont produire des signatures différentes chaque fos qu'ils fonctionnent (même quand ils signent les mêmes données avec la même clé). Donc, si un routeur BGPsec reçoit un message BGPsec UPDATE d'un homologue et reçoit plus tard un second message BGPsec UPDATE du même homologue pour le même préfixe avec le même Secure_Path et SKI, le second message UPDATE PEUT différer du premier message UPDATE dans les champs de signature (pour un algorithme de signature non déterministe). Cependant, les deux jeux de champs de signature ne vont pas différer si l'envoyeur met en antémémoire et réutilise la signature précédente. Pour un algorithme de signature déterministe, les champs de signature DOIVENT être identiques entre les deux messages UPDATE. Sur la base de ces observations, une mise en œuvre PEUT incorporer des optimisations dans le traitement de validation de mise à jour.

7.5 Numéros d'AS privés

Il est possible qu'un consommateur d'extrémité d'un FAI emploie un numéro d'AS privé. Un tel consommateur d'extrémité ne peut pas publier un ROA dans le RPKI global pour le numéro d'AS privé et les préfixes qu'il utilise. Aussi, le RPKI global ne peut pas prendre en charge les numéros d'AS privés (c'est-à-dire que les locuteurs BGPsec dans les AS privés ne peuvent pas produire des certificats de routeur dans le RPKI global). Pour les interactions entre le consommateur d'extrémité (avec le numéro d'AS privé) et le FAI, les deux scénarios suivants sont possibles :

1. Le consommateur d'extrémité envoie un message BGP UPDATE non signé pour un préfixe à l'AS du FAI. Un locuteur BGPsec de bordure dans l'AS du FAI peut choisir de propager le préfixe à ses homologues non BGPsec et BGPsec. Si il en est ainsi, le locuteur BGPsec du FAI bordure DOIT supprimer le AS_PATH qui a le numéro d'AS privé et ensuite (a) générer à nouveau le préfixe sans aucune signature pour son homologue non BGPsec et (b) générer à nouveau le préfixe en incluant sa propre signature à son homologue BGPsec. Dans les deux cas (c'est-à-dire, (a) et (b)) le préfixe DOIT avoir un ROA dans le RPKI global autorisant l'AS du FAI à le générer.

2. Le FAI et le consommateur d'extrémité peuvent utiliser un répertoire RPKI local (en utilisant un mécanisme tel que décrit dans la [RFC8416]). Ensuite, il peut y avoir un ROA pour le préfixe généré par l'AS d'extrémité, et le locuteur eBGP dans l'AS d'extrémité peut être un locuteur BGPsec ayant un certificat de routeur, bien que le ROA et le certificat de routeur ne soient valides qu'en local. Avec cet arrangement, l'AS d'extrémité envoie un message UPDATE signé pour le préfixe à l'AS du FAI. Un locuteur BGPsec bordure dans l'AS du FAI valide le message UPDATE, en utilisant les données de RPKI sur la base de la vue locale de RPKI. De plus, il peut choisir de propager le préfixe à ses homologues non BGPsec et BGPsec. Si il en est ainsi, le locuteur BGPsec bordure du FAI DOIT supprimer le Secure_Path et le segment de signature reçus de l'AS d'extrémité avec le numéro d'AS privé et ensuite (a) générer le préfixe sans aucune signature à son homologue non BGPsec et (b) générer à nouveau le préfixe en incluant sa propre signature à son homologue BGPsec. Dans les deux cas (c'est-à-dire, (a) et (b)) le préfixe DOIT avoir un ROA dans le RPKI global autorisant l'AS du FAI à le générer.

Il est possible que des numéros d'AS privés soient utilisés dans une confédération d'AS [RFC5065]. Le protocole BGPsec exige que quand un message BGPsec UPDATE se propage à travers une confédération, chaque AS membre qui le transmet à un AS membre homologue DOIT signer le message UPDATE (voir au paragraphe 4.3). Cependant, le RPKI global ne peut pas prendre en charge les numéros d'AS privés. Afin que les locuteurs BGPsec dans les AS membres avec des numéros d'AS privés aient des certificats numériques il DOIT y avoir un mécanisme en place dans la confédération qui permet l'établissement d'une vue locale personnalisée de la RPKI, augmentant les données du répertoire RPKI global comme nécessaire. Comme ce mécanisme (pour augmenter et maintenir une image locale des données de RPKI) opère localement au sein d'un AS ou confédération d'AS, il n'a pas besoin d'être normalisé. Cependant, un mécanisme normalisé peut être utilisé (voir la [RFC8416]). On se rappelle qu'afin d'empêcher l'exposition de l'intérieur des confédérations d'AS, un locuteur BGPsec qui exporte à un non membre retire tous les segments et signatures Secure_Path intra confédération (voir au paragraphe 4.3).

7.6 Considérations de robustesse pour accéder aux données de RPKI

La structure, les technologies, et les bonnes pratiques de déploiement concernant les données de RPKI globales pour joindre les routeurs (via des antémémoires RPKI locales) sont décrites dans les [RFC6810], [RFC8210], [RFC8181],

[RFC7115], [RFC8207], et [RFC8182]. Par exemple, les mécanismes de mise à jour incrémentaire fondés sur le numéro de série sont utilisés pour un transfert efficace de juste les enregistrements de données qui ont changé depuis la dernière mise à jour [RFC6810], [RFC8210]. Le fichier de notification de mise à jour est utilisé par les consommateurs d'assertions (RP, *Relying Parties*) pour découvrir si des changements existent entre l'état du répertoire RPKI global et l'antémémoire du RP [RFC8182]. La notification décrit la localisation (1) des fichiers contenant la photographie et (2) les deltas incrémentaires, qui peuvent être utilisés par le RP pour se synchroniser avec le répertoire. L'utilisation de ces technologies et bonnes pratiques résulte en une robustesse, efficacité, et meilleure sécurité pour les routeurs BGPsec et les antémémoires de RPKI en terme de flux de données de RPKI des répertoires aux antémémoires RPKI aux routeurs. Avec ces mécanismes, il est estimé qu'un attaquant ne sera pas capable de corréler significativement les flux de données de RPKI avec les actions de RP (ou routeur) BGPsec, évitant donc des attaques qui peuvent tenter de déterminer l'ensemble des AS qui interagissent avec un RP via les interactions entre le RP et les serveurs RPKI.

7.7 Redémarrage en douceur

Durant le redémarrage en douceur (GR, *Graceful Restart*) le redémarrage et la réception des locuteurs BGPsec DOIVENT suivre les procédures spécifiées dans la [RFC4724] pour respectivement redémarrer et recevoir les locuteurs BGP. En particulier, le comportement de conservation de l'état de transmission pour les chemins dans le Loc-RIB [RFC4271] et les marquer comme périmés, ainsi que ne pas différencier entre informations d'acheminement périmées et autres informations durant la transmission, va être le même comportement que spécifié dans la [RFC4724].

7.8 Robustesse du numéro aléatoire secret dans ECDSA

L'algorithme de signature numérique à courbe elliptique (ECDSA, *Elliptic Curve Digital Signature Algorithm*) avec la courbe P-256 est utilisé pour signer les messages UPDATE dans BGPsec [RFC8208]. Pour ECDSA, il est déclaré au paragraphe 6.3 de [FIPS186-4] qu'un nouveau nombre aléatoire secret "k" devra être généré avant la génération de chaque signature numérique. Un générateur de bits aléatoires (RBG, *random bit generator*) de forte entropie doit être utilisé pour générer "k", et tout biais potentiel de l'algorithme de génération de "k" doit être atténué (voir les méthodes décrites dans [FIPS186-4] et [SP800-90A]).

7.9 Considérations de déploiement incrémentaire/partiel

À quoi va ressembler la migration de BGP à BGPsec ? Quels sont les avantages pour les premiers qui l'adoptent ? Initialement, de petits groupes d'AS contigus vont constituer BGPsec. Il y aura peut être un ou plusieurs de ces groupes dans différentes régions géographiques de l'Internet mondial. Seuls les chemins générés au sein de chaque groupe et propagés au sein de ses frontières vont tirer les avantages de la protection cryptographique du chemin d'AS. Avec la croissance de l'adoption de BGPsec, chaque groupe va grandir, et finalement ils vont se joindre pour former des groupes à capacité BGPsec encore plus grands d'AS contigus. L'avantage pour les premiers adoptants commence avec la sécurité du chemin d'AS au sein des régions d'AS contigus couverts par leurs groupes respectifs. Avec le temps, ces régions d'AS contigus vont encore grandir.

Durant le déploiement partiel, si un AS dans le chemin ne prend pas en charge BGPsec, BGP revient à son mode traditionnel, c'est-à-dire, les messages BGPsec UPDATE sont convertis en messages UPDATE non signés avant la transmission à cet AS (voir au paragraphe 4.4). À ce point, l'assurance que le message UPDATE est propagé via la séquence des AS de la liste est perdue. En d'autres termes, pour les routeurs BGPsec qui résident dans les AS commençant à partir de l'AS d'origine jusqu'à l'AS avant celui qui ne prend pas en charge BGPsec, l'assurance peut toujours être fournie, mais pas au delà de ce point (pour les messages UPDATE considérés).

8. Considérations sur la sécurité

Pour une discussion du modèle de menaces sur BGPsec et les considérations de sécurité qui s'y rapportent, voir la [RFC7132].

8.1 Garanties de sécurité

Quand il est utilisé en conjonction avec la validation de l'origine (voir la [RFC6483] et la [RFC6811]) un locuteur BGPsec qui reçoit un message BGPsec UPDATE valide qui contient une annonce de chemin pour un certain préfixe reçoit les garanties de sécurité suivantes :

- Le numéro d'AS d'origine correspond à un AS qui a été autorisé, dans le RPKI, par le détenteur de l'espace d'adresses IP pour générer des annonces de chemins pour ce préfixe.
- Pour chaque AS dans le chemin, un locuteur BGPsec autorisé par le détenteur du numéro d'AS choisit intentionnellement (en accord avec la politique locale) de propager l'annonce de chemin à l'AS suivant dans le chemin.

C'est-à-dire que le receveur d'un message BGPsec UPDATE valide est assuré que le message UPDATE propagé via la séquence des AS figure dans la portion Secure_Path de l'attribut BGPsec_PATH. (On devrait noter que BGPsec n'offre aucune garantie que les paquets de données vont s'écouler le long cu chemin indiqué ; il garantit seulement que le message BGP UPDATE qui porte le chemin a bien sûr été propagé le long du chemin indiqué.) De plus, le receveur est assuré que ce chemin se termine dans un AS qui a été autorisé par le détenteur de l'espace d'adresses IP comme destination légitime pour le trafic à ce préfixe.

Noter que bien que BGPsec fournisse un mécanisme pour qu'un AS valide qu'un message UPDATE reçu a certaines propriétés de sécurité, l'utilisation d'un tel mécanisme pour influencer le choix de chemin est entièrement une affaire de politique locale. Donc, un locuteur BGPsec ne peut tirer aucune conclusions sur la validité d'un chemin reçu d'un homologue BGPsec externe (eBGP). C'est-à-dire que un homologue conforme à BGPsec peut (selon la politique locale de l'homologue) envoyer des messages UPDATE qui échouent à l'essai de validité de la Section 5. Donc, un locuteur BGPsec DOIT valider complètement tous les messages BGPsec UPDATE reçus d'homologues externes. (La validation des messages UPDATE reçus d'homologues internes est aussi une affaire de politique locale ; voir la Section 5.)

8.2 Retrait des signatures BGPsec

Il peut y avoir des cas où un locuteur BGPsec estime "valide" (selon l'algorithme de validation du paragraphe 5.2) un message BGPsec UPDATE qui contient à la fois un bloc de signature "valide" et un "non valide". C'est-à-dire que le message UPDATE contient deux ensembles de signatures correspondant aux deux suites d'algorithmes, et un ensemble de signatures se vérifie correctement et l'autre échoue. Dans ce cas, le protocole spécifie qu'un locuteur BGPsec qui choisit de propager l'annonce de chemin dans un tel message UPDATE DOIT ajouter sa signature à chaque bloc de signature (voir au paragraphe 4.2). Donc, le locuteur BGPsec crée une signature en utilisant les deux suites d'algorithmes et crée un nouveau message UPDATE qui contient les deux ensembles de signature, "valide" et "non valide" (d'après son point de vue).

Pour comprendre la raison d'une telle décision de conception, considérons le cas où le locuteur BGPsec reçoit un message UPDATE avec un ensemble d'algorithmes A de signatures qui sont "valides" et un ensemble d'algorithmes B de signatures qui sont "non valides". Dans ce cas, il est possible (peut-être même probable, selon l'état de la transition d'algorithmes) que certains des homologues du locuteur BGPsec (ou autres entités plus en aval de la topologie BGP) ne prennent pas en charge l'algorithme A. Donc, si le locuteur BGPsec devait supprimer le jeu de signatures "non valides" correspondant à l'algorithme B, ces entités traiteraient le message comme si il n'était pas signé. En incluant le jeu de signatures "non valides" lors de la propagation d'une annonce de chemin, le locuteur BGPsec s'assure que les entités en aval ont autant d'informations que possible pour se faire une opinion informée sur l'état de validation d'un message BGPsec UPDATE.

Noter aussi que durant la période de déploiement partiel de BGPsec, une entité aval peut raisonnablement traiter les messages non signés différemment des messages BGPsec UPDATE qui contiennent un seul jeu de signatures "non valides". C'est-à-dire que, en retirant le jeu de signatures "non valides", le locuteur BGPsec peut en fait être cause qu'une entité aval "remonte" le statut d'une annonce de chemin de "non valide" à non signé. Finalement, on notera que dans le scénario ci-dessus, le locuteur BGPsec peut avoir estimé l'algorithme A de signatures "valide" seulement à cause d'un problème avec l'état local de RPKI à son AS (par exemple, son AS peut n'avoir pas encore obtenu une liste de révocation de certificat (CRL, *Certificate Revocation List*) indiquant qu'une clé utilisée pour vérifier un algorithme A de signature appartient à un certificat qui vient d'être révoqué). Dans ce cas, il est très souhaitable qu'une entité aval traite le message UPDATE comme "non valide" (à cause de la révocation) et non comme "non signé" (ce qui arriverait si les blocs de signature "non valides" étaient supprimés en chemin).

Un argument similaire s'applique au cas où un locuteur BGPsec (pour une raison comme le manque d'alternatives viables) choisit comme son meilleur chemin (pour un certain préfixe) un chemin obtenu via un message BGPsec UPDATE "non valide". Dans ce cas, le locuteur BGPsec devrait propager un message BGPsec UPDATE signé, ajoutant sa signature aux signatures "non valides" qui existent déjà. Là encore, c'est pour assurer que les entités en aval sont capables de prendre une décision informée et ne pas traiter par erreur le chemin comme non signé. On devrait aussi noter qu'à cause de possibles différences dans les données de RPKI observées à différents points dans le réseau, un message BGPsec UPDATE réputé "non valide" chez un locuteur BGPsec en amont peut être réputé "valide" par un autre locuteur BGP en aval.

Bien sûr, quand un locuteur BGPsec signe un message UPDATE sortant, il n'atteste pas que toutes les signatures avant sa propre signature sont valides. Il affirme plutôt simplement que :

- o le locuteur BGPsec a reçu cette annonce de chemin avec le préfixe, l'AFI, SAFI, et Secure_Path indiqués, et
- o le locuteur BGPsec a choisi de propager une annonce pour ce chemin à l'homologue (implicitement) indiqué par le "numéro d'AS cible".

8.3 Atténuation des attaques de déni de service

La procédure de validation de mise à jour BGPsec est une cible potentielle pour les attaques de déni de service contre un locuteur BGPsec. L'atténuation des attaques de déni de service qui sont spécifiques du protocole BGPsec est examinée ici.

Pour atténuer l'efficacité de telles attaques de déni de service, les locuteurs BGPsec devraient mettre en œuvre un algorithme de validation de mise à jour qui effectue des vérifications coûteuses (par exemple, vérification de signature) après avoir effectué des vérifications qui sont moins coûteuses (par exemple, vérifications de syntaxe). L'algorithme de validation spécifié au paragraphe 5.2 a été choisi afin d'effectuer les vérifications qui vont probablement être coûteuses après les vérifications qui vont probablement être bon marché. Cependant, le coût relatif de la réalisation des étapes de validation requises peut varier entre les mises en œuvre, et donc l'algorithme spécifié au paragraphe 5.2 peut ne pas fournir la meilleure protection contre le déni de service pour toutes les mises en œuvre.

De plus, l'envoi des messages UPDATE avec de très longs chemins d'AS (et donc un grand nombre de signatures) est un mécanisme potentiel pour conduire des attaques de déni de service. Pour cette raison, il est important qu'une mise en œuvre de l'algorithme de validation arrête de tenter de vérifier les signatures aussitôt qu'une signature invalide est trouvée. (Cela assure que les longues séquences de signatures invalides ne peuvent pas être utilisées pour des attaques de déni de service.) De plus, les mises en œuvre peuvent atténuer de telles attaques en n'effectuant la validation que sur les messages UPDATE qui, si ils sont valides, vont être choisis comme meilleur chemin. C'est-à-dire que, si un message UPDATE contient un chemin qui va perdre pour d'autres raisons dans le choix du meilleur chemin (par exemple, un très long chemin d'AS) il n'est alors pas nécessaire de déterminer l'état de validité BGPsec du chemin.

8.4. Considérations de sécurité supplémentaires

Le mécanisme de réglage du champ pCount à 0 est inclus dans cette spécification pour permettre aux serveurs de chemin sur le chemin de contrôle de participer à BGPsec sans augmenter la longueur du chemin d'AS. Deux autres scénarios où pCount=0 est utilisé sont dans les contextes de confédération d'AS (voir au paragraphe 4.3) et de migration d'AS [RFC8206]. Dans ces deux scénarios, pCount=0 est établi et aussi accepté au sein du même AS (bien que l'AS ait deux identités différentes). Cependant, des entités autres que les serveurs de chemin, les confédérations d'AS, ou les AS migrants pourraient utiliser ce mécanisme (régler pCount à 0) pour attirer illégitimement le trafic (en réduisant la longueur du chemin d'AS). Ce risque est largement atténué si chaque locuteur BGPsec suit les lignes directrices de fonctionnement du paragraphe 7.2 pour la configuration en réglant pCount=0 et/ou en acceptant pCount=0 de la part d'un homologue. Cependant, on note qu'un receveur d'un message BGPsec UPDATE avec lequel une entité deux ou trois bonds en amont a réglé pCount à 0 est incapable de vérifier pour elle-même si pCount a été réglé à 0 à bon droit.

Il y a une possibilité de passer un message BGPsec UPDATE via un tunnelage entre des AS qui s'entendent entre eux. Par exemple, disons que l'AS-X n'échange pas de trafic avec l'AS-Y mais s'entend avec l'AS-Y, et il signe et envoie un message BGPsec UPDATE à l'AS-Y par tunnelage. L'AS-Y peut encore signer et propager le message BGPsec UPDATE à ses homologues. Il sort du domaine d'application du protocole BGPsec de détecter cette forme de comportement malveillant. BGPsec est conçu pour protéger les messages envoyés au sein de BGP (c'est-à-dire, dans le plan de contrôle) – et non quand le plan de contrôle est outrepassé.

Une variante de la collusion par tunnelage mentionnée ci-dessus peut arriver dans le contexte de confédérations d'AS. Quand un routeur BGPsec (en dehors d'une confédération) transmet un message UPDATE à un AS membre de la confédération, il signe le message UPDATE au numéro d'AS public de la confédération et non au numéro d'AS du membre (voir au paragraphe 4.3). L'AS membre peut tunneler le message UPDATE signé à un autre AS membre lorsque il est reçu (c'est-à-dire, sans ajouter de signature). Le message UPDATE peut alors être propagé en utilisant BGPsec aux autres membres de la confédération ou aux voisins BGPsec en dehors de la confédération. Cette sorte d'opération est possible, mais aucune grave compromission de sécurité ou d'accessibilité n'est à craindre pour les raisons suivantes :

- o les membres de la confédération appartiennent à une organisation, et une forte confiance mutuelle est supposée ;
- o on se rappelle que les signatures qui sont internes à la confédération DOIVENT être retirées avant de transmettre le message UPDATE à un routeur BGPsec externe (voir au paragraphe 4.3).

BGPsec ne fournit pas de protection contre les attaques à la couche transport. Comme avec toute session BGP, un adversaire sur le chemin entre un locuteur BGPsec et son homologue est capable d'effectuer des attaques comme la

modification de messages BGPsec UPDATE valides pour causer leur échec à la validation, l'injection de messages BGP UPDATE (non signés) sans attributs BGPsec_PATH, l'injection de messages BGPsec UPDATE avec des attributs BGPsec_PATH qui échouent à la validation, ou de causer la suppression par l'homologue de la session BGP. L'utilisation de BGPsec ne fait rien pour augmenter la puissance d'un adversaire sur le chemin -- en particulier, même un adversaire sur le chemin ne peut pas faire croire à un locuteur BGPsec qu'un chemin BGPsec invalide est valide. Cependant, comme avec toute session BGP, les sessions BGPsec DEVRAIENT être protégées par des mécanismes appropriés de sécurité du transport (voir la Section Considérations sur la sécurité dans la [RFC4271]).

Il y a une possibilité d'attaques en répétition, définie comme suit. Dans le contexte de BGPsec, une attaque en répétition se produit quand un locuteur BGPsec malveillant sur le chemin d'AS supprime un retrait de préfixe (implicite ou explicite). De plus, une attaque en répétition est dite se produire aussi quand un locuteur BGPsec malveillant répète une annonce BGPsec reçue précédemment pour un préfixe qui a été retiré depuis. La stratégie d'atténuation pour les attaques en répétition implique le retour à zéro du certificat de routeur ; voir les détails dans la [RFC8634].

9. Considérations relatives à l'IANA

L'IANA a enregistré une nouvelle capacité BGP décrite au paragraphe 2.1 dans la gamme "revue de l'IETF" [RFC8126] du registre des "Codes de capacités". La description de la nouvelle capacité est "capacité BGPsec". Le présent document est la référence pour la nouvelle capacité.

L'IANA a aussi enregistré un nouvel attribut de chemin décrit à la Section 3 dans le registre"Attributs de chemin". Le code de ce nouvel attribut est "BGPsec_PATH". Le présent document est la référence pour ce nouvel attribut.

L'IANA a défini le registre "Capacités BGPsec" dans le groupe "Infrastructure de clé publique de ressource (RPKI)". Le registre est montré à la Figure 10, avec les valeurs allouées du paragraphe 2.1:

Bits	Champ	Référence
0-3	Version. Valeur = 0x0	[RFC8205]
4	Direction	[RFC8205]
	(les deux valeurs possibles 0 et 1 sont spécifiées par cette RFC)	
5-7	non alloués ; valeur = 0r00 (en binaire)	[RFC8205]

Figure 10 : Registre IANA pour les capacités BGPsec

Le bit Direction (quatrième bit) a une valeur de 0 ou 1, et les deux valeurs sont pleinement spécifiées par le présent document. Les futures valeurs de version et les futures valeurs des bits non alloués seront allouées en utilisant les procédures d'enregistrement de "Action de normalisation" définies dans la [RFC8126].

L'IANA a défini le registre "Fanions BGPsec_PATH" dans le groupe "Infrastructure de clé publique de ressource (RPKI)". Le registre est montré à la Figure 11, avec une valeur allouée du paragraphe 3.1:

Fanion	Description	Référence
0	Confed_Segment	[RFC8205]
	Valeur du bit = 1 signifie fanion établi (indique Confed_Segment)	
	Valeur du bit = 0 par défaut	
1-7	Non allouées, valeur : tous les 7 bits à zéro	[RFC8205] |

Figure 11 : Registre IANA du champ Fanions de BGPsec_PATH

Les valeurs futures des bits non alloués seront allouées par les procédures d'enregistrement de "Action de normalisation" définies dans la [RFC8126].

10. Références

10.1 Références normatives

[IANA-AF] IANA, "Address Family Numbers", <https://www.iana.org/assignments/address-family-numbers>.

[RFC2119] S. Bradner, "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", DOI 10.17487/RFC2119, BCP 14, mars 1997. *(MàJ par* RFC8174)

[RFC4271] Y. Rekhter, T. Li et S. Hares, "Protocole de routeur frontière version 4 (BGP-4)", janvier 2006. DOI 10.17487/RFC4271, *(D.S. MàJ par* RFC6608, RFC8212)

[RFC4724] S. Sangli et autres, "Mécanisme de redémarrage en douceur pour BGP", janvier 2007. DOI 10.17487/RFC4724, *(P.S.)*

[RFC4760] T. Bates, R. Chandra, D. Katz et Y. Rekhter, "Extensions multi protocoles pour BGP-4", janvier 2007. DOI 10.17487/RFC4760,

[RFC5065] P. Traina et autres, "Confédérations de systèmes autonomes pour BGP", août 2007. DOI 10.17487/RFC5065, (*Remplace* RFC3065) *(D.S.)*

[RFC5492] J. Scudder, R. Chandra, "Annonces de capacités avec BGP-4", février 2009. DOI 10.17487/RFC5492, (*Remplace la* RFC3392) *(D.S.)*

[RFC6482] M. Lepinski, S. Kent, D. Kong, "Profil d'autorisations d'origine de chemin (ROA)", février 2012. DOI 10.17487/RFC6482, *(P.S.)*

[RFC6487] G. Huston, G. Michaelson, R. Loomans, "Profil pour les certificats de ressource X.509 PKIX", février 2012. DOI 10.17487/RFC6487, *(P.S. ; MàJ par* RFC8209)

[RFC6793] Q.Vohra et E. Chen, "Prise en charge par BGP d'espace de noms de système autonome (AS) à quatre octets", décembre 2012. DOI 10.17487/RFC6793.

[RFC7606] E. Chen, et autres, "Traitement d'erreur révisé pour le message BGP UPDATE" août 2015. DOI 10.17487/RFC7606, *(P.S. ; MàJ* 1997, 4271, 4360, 4456, 4760, 5543, 5701, 6368*)*

[RFC8126] M. Cotton, B. Leiba, T. Narten, "Lignes directrices pour la rédaction d'une section de considérations relatives à l'IANA dans les RFC", juin 2017. BCP 26. DOI 10.17487/RFC8126, *(Remplace* RFC5226)

[RFC8174] B. Leiba, "Ambiguïté des mots clés en majuscules ou minucules dans la RFC2119", mai 2017. BCP14. DOI 10.17487/RFC8174, *(MàJ* 2119)

[RFC8208] S. Turner, O. Borchert, "Algorithmes, formats de clé et de signature pour BGPsec", septembre 2017. DOI 10.17487/RFC8208, *(P.S. ; MàJ* RFC7935 *; rendue obolète par* RFC8608*)*

[RFC8209] M. Reynolds, S. Turner, S. Kent, "Profil pour les certificats de routeur, les listes de révocation de certificat, et les demandes de certification BGPsec", septembre 2017. DOI 10.17487/RFC8209, *(P.S. ; MàJ* RFC6487*)*

10.2 Références pour information

[Borchert] O. Borchert et M. Baer, message à la liste de diffusion du groupe de travail SIDR de l'IETF du 10 février 2016.

[FIPS186-4] National Institute of Standards and Technology, "Digital Signature Standard (DSS)", NIST FIPS Publication 186-4, DOI 10.6028/NIST.FIPS.186-4, juillet 2013.

[RFC6472] W. Kumari, K. Sriram, "Recommandation de ne pas utiliser AS_SET et AS_CONFED_SET dans BGP", décembre 2011. (BCP0172) DOI 10.17487/RFC6472.

[RFC6480] M. Lepinski, S. Kent, "Infrastructure pour la prise en charge de l'acheminement Internet sécurisé", février 2012. DOI 10.17487/RFC6480, *(Info.)*

[RFC6483] G. Huston, G. Michaelson, "Validation d'origine de chemin en utilisant l'infrastructure de clé publique (PKI) de certificat de ressource et les autorisations d'origine de chemin (ROA)", février 2012. DOI 10.17487/RFC6483, *(Information)*

[RFC6810] R. Bush et R. Austein, "Infrastructure de clé publique de ressource (RPKI) pour protocole de routeur", janvier 2013. DOI 10.17487/RFC6810, *(MàJ par* RFC8210)

[RFC6811] P. Mohapatra, et autres, "Validation d'origine de préfixe BGP", janvier 2013. DOI 10.17487/RFC6811, *(P;S; ; MàJ par* RFC8481

[RFC7093] S. Turner, S. Kent, J. Manger, "Méthodes supplémentaires pour générer des valeurs d'identifiant de clé", décembre 2013. DOI 10.17487/RFC7093, *(Information)*

[RFC7115] R. Bush, "Opération de validation d'origine fondée sur l'infrastructure de clé publique de ressource (RPKI)", janvier 2014, (BCP0185). DOI 10.17487/RFC7115.

[RFC7132] S. Kent, A. Chi, "Modèle de menace pour la sécurité de chemin BGP", février 2014. DOI 10.17487/RFC7132, *(Information)*

[RFC8181] S. Weller, et autres, "Protocole de publication pour RPKI", juillet 2017. DOI 10.17487/RFC8181, *(P.S.)*

[RFC8182] T. Bruijnzeels, et autres, "Protocole de différences de répertoire RPKI (RRDP)", juillet 2017. DOI 10.17487/RFC8182, *(P.S.)*

[RFC8206] W. George, S. Murphy, "Migration des systèmes autonomes sur BGPsec", septembre 2017. DOI 10.17487/RFC8206, *(P.S. ; MàJ* RFC8205*)*

[RFC8207] R. Bush, "Considérations sur le fonctionnement de BGPsec", septembre 2017. BCP211. DOI 10.17487/RFC8207,

[RFC8210] R. Bush, R. Austein, "Infrastructure de clé publique de ressource (RPKI) pour protocole de routeur, version 1", septembre 2017. DOI 10.17487/RFC8210, *(P.S. ; MàJ* RFC6810*)*

[RFC8416] D. Ma, et autres, "Gestion simplifiée de ressource locale de numéros Internet avec RPKI (SLURM)", août 2018. DOI 10.17487/RFC8416, *(P.S.)*

[RFC8634] B. Weis, R. Gagliano, K. Patel, "Changement de clés de certificat de routeur BGPsec", août 2019. DOI 10.17487/RFC8634, *(P.S.)*

[SP800-90A] National Institute of Standards and Technology, "Recommendation for Random Number Generation Using Deterministic Random Bit Generators", NIST SP 800-90A Rev 1, DOI 10.6028/NIST.SP.800-90Ar1, juin 2015.

Remerciements

Les auteurs tiennent à remercier Michael Baer, Oliver Borchert, David Mandelberg, Mehmet Adalier, Sean Turner, Wes George, Jeff Haas, Alvaro Retana, Nevil Brownlee, Matthias Waehlisch, Tim Polk, Russ Mundy, Wes Hardaker, Sharon Goldberg, Ed Kern, Doug Maughan, Pradosh Mohapatra, Mark Reynolds, Heather Schiller, Jason Schiller, Ruediger Volk, et David Ward pour leur relecture, commentaires, et suggestions durant le cours de ce travail. Merci aussi aux nombreux relecteurs de l'IESG dont les commentaires ont largement aidé à améliorer la clarté, la précision, et la présentation du document.
Les auteurs souhaiter remercier particulièrement Oliver Borchert et Michael Baer pour leur relecture et suggestions [Borchert] concernant la séquence d'octets à hacher (Figures 8 et 9 des paragraphes, respectivement 4.2 et 5.2). Cela a été une importante contribution fondée sur leur expérience de mise en œuvre.

Contributeurs

Les personnes suivantes ont fait des contributions significatives au présent document et devraient être considérés comme co-auteurs :

Rob Austein
Dragon Research Labs

Steven Bellovin
Columbia University

Russ Housley
Vigil Security

Stephen Kent
BBN Technologies

mél : sra@hactrn.net mél : smb@cs.columbia.edu mél :housley@vigilsec.com mél :kent@alum.mit.edu

Warren Kumari
Google
mél :warren@kumari.net

Doug Montgomery
NIST
mél : dougm@nist.gov

Chris Morrow
Google, Inc.
mél : morrowc@google.com

Sandy Murphy
SPARTA, Inc.,
mél : sandy@tislabs.com

Keyur Patel
Arrcus
mél : keyur@arrcus.com

John Scudder

mél : jgs@juniper.net

Samuel Weiler
W3C/MIT
mél : weiler@csail.mit.edu

Adresse des auteurs

Matthew Lepinski (editor)
New College of Florida
5800 Bay Shore Road
Sarasota, FL 34243
United States of America
mèl : mlepinski@ncf.edu

Kotikalapudi Sriram (editor)
USA National Institute of Standards et Technology
100 Bureau Drive
Gaithersburg, MD 20899
United States of America
mèl : kotikalapudi.sriram@nist.gov