Groupe de travail Réseau
**Request for Comments : 2726**
Catégorie : En cours de normalisation

J. Zsako, BankNet
décembre 1999
Traduction Claude Brière de L'Isle

# Authentification PGP pour les mises à jour de base de données RIPE

**Statut du présent mémoire**
Le présent document spécifie un protocole en cours de normalisation de l'Internet pour la communauté de l'Internet, et appelle à des discussions et suggestions pour son amélioration. Prière de se référer à l'édition en cours des "Normes officielles des protocoles de l'Internet" (STD 1) pour connaître l'état de la normalisation et le statut de ce protocole. La distribution du présent mémoire n'est soumise à aucune restriction.



**Résumé**
Le présent document présente la proposition d'une méthode d'authentification plus forte pour les mises à jour de la base de données RIPE sur la base de signatures numériques. La proposition essaye d'être aussi générale que possible pour ce qui concerne les méthodes de signature numérique, cependant, il se concentre principalement sur PGP, comme première méthode à mettre en œuvre. La proposition est le résultat des discussions au sein du groupe de travail RIPE DBSEC.

## 1. Motifs

Un besoin croissant a été identifié pour une authentification plus forte de la maintenance de la base de données lors des mises à jour de la base de données (ajout, modification et suppression des objets). Les méthodes d'authentification existantes posent de sérieux problèmes de sécurité : le MAIL-FROM a pour inconvénient qu'un en-tête de message est très facile à falsifier tandis que CRYPT-PW est exposé à l'interception de message, car le mot de passe est envoyé non chiffré par le message de mise à jour.

Le but est de mettre en œuvre un mécanisme de signature numérique fondé sur une technologie largement disponible et déployée. Le premier choix a été PGP ; d'autres méthodes peuvent suivre ultérieurement. PGP est présentement assez largement utilisé au sein de la communauté de l'Internet et est disponible à la fois dans et à l'extérieur des USA.

Le but actuel n'est rien de plus qu'une amélioration de la méthode d'authentification (en particulier, le présent article n'essaye pas de couvrir les problèmes d'autorisation autres que ceux qui se rapportent à l'authentification).

## 2. Changements à la base de données RIPE

Afin de rendre la base de données aussi cohérente que possible, les certificats de clés sont mémorisés dans la base de données RIPE. Pour des raisons d'efficacité un changement de clés local des clés publiques sera aussi effectué ; cependant, le changement de clé en local ne va contenir qu'une copie des certificats de clés présents dans la base de données. La synchronisation de la base de données avec le système de clés local sera faite aussi souvent que possible. Les objets de la base de données ne seront créés que via le mécanisme actuel de messagerie électronique (auto-dbm@ripe.net) en particulier aucun certificat de clé publique ne sera restitué à partir d'un serveur de clés par le logiciel de la base de données.

La présence des certificats de clés dans la base de données permettra aux utilisateurs de la base de données de vérifier "l'identité" de celui qui l'entretient, dans le sens qu'ils peuvent interroger la base de données sur le certificat de la clé que le logiciel de la base de données utilise pour l'authentification de celui qui fait la maintenance. Ce certificat de clé peut alors être vérifié sur les signatures existantes et peut éventuellement être comparé au certificat de clé obtenu par d'autres moyens pour le même utilisateur (par exemple, à partir du propriétaire lui-même par un serveur de clé publique). Bien que les certificats de clés puissent être mémorisés dans la base de données RIPE avec un nombre quelconque de signatures, comme la base de données RIPE ne communique pas directement avec les serveurs de clé publique, il est de bonne pratique d'ajouter au certificat de clé le nombre minimum de signatures possible (de préférence une seule signature : celle de soi-même). Voir aussi les détails à la section 4..

### 2.1 Objet key-cert

Un nouveau type d'objet est défini ci-dessous pour les besoins de la mémorisation des certificats de clé de ceux qui font la maintenance :

key-cert : [obligatoire] [seul] [clé principale/recherche]
method : [générée] [seul] [ ]
owner : [générée] [multiple] [ ]
fingerpr : [générée] [seul] [ ]
certif : [obligatoire] [seul] [ ]
remarks : [facultatif] [multiple] [ ]
notify : [facultatif] [multiple] [clé inverse]
mnt-by : [obligatoire] [multiple] [clé inverse]
changed : [obligatoire] [multiple] [ ]
source : [obligatoire] [seul] [ ]

La syntaxe et la sémantique des différents attributs sont décrits ci-dessous.

key-cert : Est de la forme PGPKEY-hhhhhhhh, où hhhhhhhh est la représentation hexadécimale des quatre octets de l'identifiant de la clé PGP. Le certificat de clé détaillé dans l'attribut certif appartient à la clé PGP avec l'identifiant hhhhhhhh. La raison pour avoir PGPKEY- comme préfixe est de permettre d'autres types de certificats de clé ultérieurement, et d'être en même temps capable de différencier clairement au moment de l'interrogation entre une interrogation de personne et une interrogation de certificat de clé. Au moment de la création/modification de l'objet key-cert, le logiciel de la base de données vérifie si le certificat de clé dans l'attribut certif appartient bien à l'identifiant PGP spécifié ici. La création/modification n'est autorisée que si il y a correspondance de ces deux identifiants.

method : Ligne contenant le nom de la méthode de signature. C'est le nom de la méthode de signature numérique. Le présent certificat appartient à une clé pour les messages avec signature numérique qui utilisent la méthode spécifiée. L'attribut method est généré automatiquement par le logiciel de la base de données à la création de l'objet key-cert. Tout attribut method présent dans l'objet au moment de la soumission de création est ignoré. La méthode doit être cohérente avec le préfixe de l'identifiant contenu dans l'attribut key-cert et avec le certificat contenu dans les attributs certif. Si ces deux derniers (c'est-à-dire le préfixe et le certificat) ne sont pas en cohérence, la création de l'objet key-cert est refusée. Pour la méthode PGP, ce sera la chaîne "PGP" (sans les guillemets).

owner : Ligne contenant une description du propriétaire de la clé. Pour une clé PGP, les propriétaires sont les identifiants d'utilisateur associés à la clé. Pour chaque identifiant d'utilisateur présent dans le certificat de clé, un attribut owner est généré automatiquement par le logiciel de la base de données à la création de l'objet key-cert. Tout attribut owner présent dans l'objet au moment de la soumission de création est ignoré.

fingerpr : Un certain nombre d'octets codés en hexadécimal, séparés par des espaces pour une meilleure lisibilité. Cela représente l'empreinte de la clé associée au certificat présent. C'est aussi un champ généré à la création de l'instance d'objet. Tout attribut fingerpr soumis au robot est ignoré. La raison d'être de cet attribut (et de l'attribut owner) est de permettre une facile vérification du certificat de clé lors d'une interrogation de la base de données. L'interrogateur obtient les informations de owner et fingerprint sans avoir à ajouter le certificat à son propre trousseau de clés publiques. Aussi, comme ces deux attributs sont générés par le logiciel de la base de données à partir du certificat, on peut leur faire confiance (pour autant qu'on puisse faire confiance à la base de données elle-même).

certif : Lignes contenant une ligne du certificat de clé cuirassée en ASCII. Les lignes d'attribut certif contiennent le certificat de clé. Dans le cas de PGP, elles contiennent aussi les lignes de délimitation (BEGIN/END PGP PUBLIC KEY BLOCK). L'ordre des lignes est évidemment essentiel, donc les lignes d'attribut certif sont présentées au moment de l'interrogation dans le même ordre que celui dans lequel elles ont été soumises à leur création. Une application de client de base de données pourrait contenir un scénario qui dépouille les lignes d'attribut certif (retournées par suite d'une interrogation) de la chaîne "certif:" en tête et les espaces blanches suivantes et importe le reste dans le trousseau de clés local.

mnt-by : La syntaxe et la sémantique usuelle de cet attribut sont obligatoires pour cet objet. Donc, l'existence d'un objet mntner est exigée pour la création d'un objet key-cert. Le mntner référencé dans l'attribut mnt-by peut n'avoir pas l'attribut auth réglé à NONE.

remarks : ,
notify : ,
changed : ,
source : La syntaxe et sémantique usuelles.

Dans le cas de PGP, lorsque un objet key-cert est créé, la clé associée est aussi ajoutée à un trousseau local de clés publiques. Lorsque un objet key-cert est supprimé, la clé publique correspondante est aussi supprimée du trousseau de clés local. Chaque fois qu'un objet key-cert est modifié, la clé est supprimée du trousseau de clés local et la clé associée au nouveau certificat est ajoutée au trousseau de clé (ceci n'est évidemment effectué que lorsque la mise à jour de la base de données est autorisée, en particulier si le nouveau certificat de clé appartient à l'identifiant spécifié dans le key-cert d'attribut, voir ci-dessus).

### 2.2 Changements à l'objet mntner

Le seul changement est qu'il y a une nouvelle valeur possible pour l'attribut auth. Cette valeur est de la forme PGPKEY-<id>, où <id> est la représentation hexadécimale des quatre octets de l'identifiant de la clé publique PGP utilisée pour l'authentification.

La sémantique de cette nouvelle valeur est que la clé PGP associée au certificat de clé mémorisé dans l'objet key-cert identifié par le PGPKEY-id est utilisée pour vérifier la signature de tout message de création/modification/suppression envoyé à auto-dbm@ripe.net qui affecte un objet entretenu par ce mntner.

Juste comme avec les autres valeurs, l'attribut auth peut être multiple. Il n'y a donc pas beaucoup de sens à avoir deux attributs auth avec des méthodes différentes (par exemple, PGPKEY-<id> et NONE :)) )parce qu'il n'y en avait pas non plus avant.

Si il y a plusieurs méthodes auth avec une valeur de PGPKEY-<id>, la sémantique est déjà connue, à savoir que l'une et l'autre signatures sont acceptées.

## 3. Création/modification/suppression de message signé PGP

La totalité du message doit être signé. Cela signifie que le logiciel de la base de données vérifie d'abord si le message est un message PGP signé. Si il l'est, il vérifie qu'il y a une signature valide et il associe cette signature aux objets soumis dans le message. Un message ne peut contenir qu'une seule signature PGP.

Si un objet présent dans un message a un attribut mnt-by, et si le mntner correspondant a un ou des attributs auth avec la valeur PGPKEY-<id>, le logiciel de la base de données vérifie si l'objet a une signature qui lui est associée (c'est-à-dire, si le message a traiter a été signé ) et si le type de la signature (PGP dans cette phase de mise en œuvre) et l'identifiant de la clé utilisée pour signer le message sont les mêmes que ceux (un de ceux) du ou des attributs auth. La création/modification/suppression de l'objet n'est effectuée que si cette authentification réussit.

Cette approche permet une simplification du processus d'analyse du message. Une approche différente serait nécessaire si on voulait signer les objets, plutôt que les messages de mise à jour. Dans le cas où les objets seraient signés, l'analyseur devrait identifier quels objets ont été signés, vérifier individuellement la ou les signatures sur chaque objet et permettre/refuser la mise à jour au niveau de l'objet, selon (entre autres) que la signature est ou non valide et qu'elle appartient au ou aux mainteneurs. Cette approche permettrait de mélanger dans le même message électronique les objets signés par différents mainteneurs (qui ne seraient probablement pas typiques) et elle permettrait aussi de mémoriser la signature dans la base de données (afin de permettre la vérification de la signature au moment de l'interrogation). Cette dernière approche (c'est-à-dire de mémoriser les signatures dans la base de données) sort du domaine d'application de la première phase de la mise en œuvre. Elle peut devenir un objectif ultérieur.

Il est recommandé de vérifier que le programme de messagerie ne fait aucune transformations sur le texte du message électronique (et éventuellement qu'il est configuré pour n'en faire aucune). Une de ces transformations courantes est le retour à la ligne après un certain nombre de caractères, la transformation des tabulations en espaces, etc. Il faut aussi vérifier que seuls des caractères ASCII sont utilisés dans le message.

## 4. Exigences pour les certificats de clé PGP

Aucune limitation n'est imposée par rapport à la version de logiciel PGP qui est/a été utilisée pour la création de la clé. Les

clés des deux versions 2.x et 5.0 sont prise en charge, bien que les clés générées avec la version 5.0 soient recommandées.

Les certificats de clé soumis pour créer un objet key-cert doivent contenir une signature de la clé elle-même. Bien que le certificat puisse contenir aussi d'autres signatures, il est recommandé de créer l'objet key-cert avec aussi peu de signatures que possible dans le certificat. Si on a des doutes sur la confiance qu'on peut accorder à la clé, on devrait aller chercher une copie du certificat de clé auprès d'un serveur de clé publique (ou par tout autre moyen approprié) et vérifier les signatures présentes dans ce certificat. Si une telle vérification est effectuée, on devrait veiller à vérifier à la fois l'empreinte de la clé et le type et la longueur de clé afin de s'assurer que les deux certificats (celui récupéré de la base de données RIPE et celui récupéré du serveur de clés publiques ou collecté par d'autres moyens) appartiennent à la même clé.

Bien que ce soit hautement improbable, il peut arriver qu'un key-cert avec un identifiant identique à celui de la clé d'un mainteneur existe déjà dans la base de données RIPE. Au cas où cette dernière clé aurait été utilisée depuis un certain temps et aurait été enregistrée au serveur de clés publiques depuis un certain temps, on devrait contacter le RIPE NCC et en faire rapport à ripe-dbm@ripe.net. De toutes façons on peut avoir à créer une nouvelle clé et enregistrer son certificat dans la base de données RIPE. Une telle procédure, bien qu'il soit très improbable qu'elle arrive, ne devrait pas créer de sérieux problèmes pour le mainteneur concerné.

## 5. Survol des tâches à effectuer afin d'utiliser l'authentification PGP

On doit avoir un objet mntner dans la base de données RIPE avec auth: autre que NONE. L'objet mntner doit être créé de la façon traditionnelle.

On doit obtenir un certificat de sa propre clé et préparer un objet key-cert à partir de lui. L'objet doit faire référence dans le mnt-by au mntner mentionné plus haut.

Pour créer l'objet key-cert dans la base de données RIPE, on envoie l'objet préparé ci-dessus à auto-dbm@ripe.net. On doit évidemment réussir aux vérifications d'authentification requises par l'objet mntner (c'est-à-dire, un message électronique provenant d'une adresse prédéfinie ou envoyer le mot de passe correct).

Changer l'objet mntner pour avoir la valeur d'attribut auth: de PGPKEY-<id>, où <id> est l'identifiant hexadécimal de la clé PGP.

Vérifier tous les objets entretenus par le mntner (de préférence avec la commande). C'est la seule façon de bénéficier de la plus forte méthode d'authentification afin de rendre la base de données plus digne de confiance. Se rappeler que le demandeur est la seule personne qui puisse vérifier et corriger les possibles incohérences.

À partir de ce moment, toujours signer les messages de mise à jour (en totalité) qui se réfèrent aux objets que vous entretenez, avec la clé que vous avez soumise à la base de données RIPE.

## 6. Exemple d'objets qui utilisent le nouveau dispositif

```
mntner:     AS3244-MNT
descr:      BankNet, Budapest HU
descr:      Eastern European Internet Provider via own VSAT network
admin-c:    JZ38
tech-c:     JZ38
tech-c:     IR2-RIPE
upd-to:     ncc@banknet.net
mnt-nfy:    ncc@banknet.net
auth:       PGPKEY-23F5CE35
remarks:    Ceci est le mainteneur de tous les objets qui se rapportent à la BankNet
notify:     ncc@banknet.net
mnt-by:     AS3244-MNT
changed:    zsako@banknet.net 19980525
source:     RIPE

key-cert: PGPKEY-23F5CE35
method:   PGP
owner:    Janos Zsako <zsako@banknet.net>
fingerpr: B5 D0 96 D0 D0 D3 2B B2  B8 C2 5D 22 D4 F5 78 92
```

certif: -----BEGIN PGP PUBLIC KEY BLOCK-----
Version: 2.6.2i
+
mQCNAzCqKdIAAAEEAPMSQtBNFFuTS0duoUiqnPHm05dxrI76rrOGwx+OU5tzGavx
cm2iCInNtikeKjlIMD7FiCH1J8PWdZivpwhzuGeeMimT8ZmNn4z3bb6ELRyiZOvs
4nfxVlh+kKKD9JjBfy8DnuMs5sT0jw4FEt/PYogJinFdndzywXHzGHEj9c41AAUR
tB9KYW5vcyBac2FrbyA8enNha29AYmFua25ldC5uZXQ+iQCVAwUQMjkx2XHzGHEj
9c41AQEuagP/dCIBJP+R16Y70yH75kraRzXY5rnsHmT0Jknrc/ihEEviRYdMV7X1
osP4pmDU8tNGf0OfGrok7KDTCmygIh7/me+PKrDIj0YkAVUhBX3gBtpSkhEmkLqf
xbhYwDn4DV3zF7f5AMsbD0UCBDyf+vpkMzgd1Pbr439iXdgwgwta50qJAHUDBRAy
OSsrO413La462EEBAdIuAv4+Cao1wqBG7+gIm1czIb1M2cAM7Ussx6y+oL1d+HqN
PRhx4upLVg8Eqm1w4BYpOxdZKkxumIrIvrSxUYv4NBnbwQaa0/NmBou44jqeN+y2
xwxAEVd9BCUtT+YJ9iMzZlE=
=w8xL
-----END PGP PUBLIC KEY BLOCK-----
remarks: Ceci est un exemple de certificat de clé PGP
mnt-by: AS3244-MNT
changed: zsako@banknet.net 19980525
source: RIPE

## 7. Considérations pour la sécurité

Le présent document traite de l'authentification de transactions pour faire des ajouts, suppressions, et mises à jour des informations de politique d'acheminement au moyen d'un chiffrement fort. L'autorisation de ces transactions est traitée dans la [RFC2725].

## 8. Remerciements

La présente proposition est le résultat des discussions qui ont eu lieu au sein de l'équipe RIPE DBSEC, qui a été établie par RIPE 27 à Dublin à l'initiative de Joachim Schmitz et Wilfried Woeber. La liste des participants qui ont contribué aux discussions à différentes occasions (réunions de l'équipe et via e-mail) est (par ordre alphabétique) : Cengiz Allaettinoglu, Joao Luis Silva Damas, Havard Eidnes, Chris Fletcher, Daniel Karrenberg, David Kessens, Jake Khuon, Craig Labovitz, Carl Malamud, Dave Meyer, Maldwyn Morris, Sandy Murphy, Mike Norris, Carol Orange, Joachim Schmitz, Tom Spindler, Don Stikvoort, Curtis Villamizar, Gerald Winters, Wilfried Woeber, Janos Zsako.

## 9. Références

[RFC2725] C. Villamizar et autres, "Sécurité du système de politique d'acheminement", décembre 1999. (*MàJ par* RFC4012) *(P.S.)*

## 10. Adresse de l'auteur

Janos Zsako
BankNet
1121 Budapest
Konkoly-Thege ut 29-33.
Hungary

téléphone : +36 1 395 90 28
Fax : +36 1 395 90 32
mél : zsako@banknet.net

## 11. Remarques

PGP est un logiciel commercial.

L'IETF ne prend position sur la validité ou la portée d'aucun droit de propriété intellectuelle ou autres droits qui pourraient être revendiqués à l'égard de la mise en œuvre ou l'utilisation de la technologie décrite dans le présent document ou sur la mesure dans laquelle toute licence sur de tels droits pourrait être ou non disponible ; ni ne prétend avoir fait aucun effort pour identifier de tels droits. Les informations sur les procédures de l'IETF à l'égard des droits dans les documents en cours de normalisation et en rapport avec les normes se trouvent dans le BCP-11. Des copies des revendications de droits rendues disponibles pour la publication et toutes assurances de licences rendues disponibles, ou le résultat d'une tentative faite pour obtenir une licence générale ou permission d'utilisation de tels droits de propriété par ceux qui mettent en œuvre ou par les utilisateurs de la présente spécification, peuvent être obtenues au Secrétariat de l'IETF.

L'IETF invite toute partie intéressée à porter à son attention tous droits de reproductions, brevets ou applications de brevet, ou autres droits de propriété qui pourraient couvrir la technologie qui pourrait être requise pour metttre la présente norme en pratique. Prière d'adresser les informations au Directeur exécutif de l'IETF.

## 12. Déclaration complète de droits de reproduction



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