Groupe de travail Réseau
Request for Comments : 2015
Catégorie : En cours de normalisation

M. Elkins, The Aerospace Corporation
octobre 1996
Traduction Claude Brière de L'Isle

Sécurité de MIME avec Pretty Good Privacy (PGP)

Résumé
Le présent document décrit comment Pretty Good Privacy (PGP) peut être utilisé pour assurer la confidentialité et l'authentification en se servant des types de contenu de sécurité des extensions multi objet de messagerie Internet (MIME, *Multipurpose Internet Mail Extensions*) décrites dans la [RFC1847].

1. Introduction

Des travaux précédents sur l'intégration de PGP dans MIME (incluant le type de contenu application/pgp retiré depuis) ont connu un certain nombre de problèmes, dont le plus significatif était l'incapacité à récupérer les corps de message signés sans analyser les structures de données spécifiques de PGP. Le présent travail utilise la solution élégante proposée dans la [RFC1847], qui définit un format de multiparties de sécurité pour MIME. Les multiparties de sécurité séparent clairement le corps de message signé de la signature, et elles ont un certain nombre d'autres propriétés désirables. Le présent document adopte le style de la [RFC1848], qui définit les services de sécurité d'objet MIME (MOSS, *MIME Object Security Service*) pour assurer la sécurité et l'authentification.

Le présent document définit trois nouveaux types de contenu pour la mise en œuvre de la sécurité et de la confidentialité avec PGP : application/pgp-encrypted, application/pgp-signature et application/pgp-keys.

1.1 Comformité

Pour qu'une mise en œuvre soit conforme à la présente spécification, il est absolument nécessaire qu'elle respecte tous les éléments marqués DOIT ou EXIGÉ.

2. Formats de données PGP

PGP peut générer une cuirasse ASCII (décrite dans la [RFC1991]) ou un résultat binaire à 8 bits lors du chiffrement des données, générant une signature numérique, ou en extrayant des données de clé publique. Le résultat avec la cuirasse ASCII est la méthode EXIGÉE pour le transfert des données. Cela permet aux usagers qui n'ont pas les moyens d'interpréter les formats décrits dans le présent document d'être capables d'extraire et utiliser les informations PGP dans le message.

Lorsque la quantité de données à transmettre exige qu'elles soient envoyées en plusieurs parties, le mécanisme MIME message/partial devrait être utilisé plutôt que le format multiparties PGP de cuirasse ASCII.

3. Restrictions au contenu, transfert, codage

Multipart/signé et multipart/chiffré sont à traiter par les agents comme opaques, ce qui signifie que les données ne doivent être altérées d'aucune façon [RFC1847].

Cependant, de nombreuses passerelles de messagerie existantes vont détecter si le prochain bond ne prend pas en charge MIME ou les données à 8 bits et vont effectuer une conversion en Quoted-Printable ou Base64. Cela cause de sérieux problèmes pour multipart/signé, en particulier, lorsque la signature est invalidée quand survient une telle opération. Pour cette raison, toutes les données signées conformément au présent protocole DOIVENT être contraintes à 7 bits (les données à 8 bits devraient être codées en utilisant soit Quoted-Printable soit Base64). Noter que cela inclut aussi le cas où un objet

signé est aussi chiffré (voir la section 6). Cette restriction augmentera la probabilité de validité de la signature à réception.

Les données qui sont SEULEMENT à chiffrer peuvent contenir des caractères à 8 bits et n'ont donc pas besoin d'être converties au format 7 bits.

Note pour la mise en œuvre : On ne soulignera jamais assez que les applications qui utilisent la présente norme devraient suivre la suggestion de MIME d'être "conservateur dans ce qu'on génère, et libéral dans ce qu'on accepte." Dans ce cas particulier, cela signifie qu'il serait sage qu'une mise en œuvre accepte les messages avec tout contenu-transfert-codage, mais se restreignent à la génération du format de 7 bits exigé par le présent mémoire. Cela permettra la compatibilité future dans le cas où le cadre SMTP de l'Internet passerait à 8 bits.

4. Données chiffrées dans PGP

Avant le chiffrement avec PGP, les données devraient être écrites dans le format canonique de MIME (corps et en-têtes).

Les données chiffrées en PGP sont notées avec le type de contenu "multipart/chiffré" décrit dans la [RFC1991], et DOIVENT avoir une valeur de paramètre "protocole" de "application/pgp-encrypted". Noter que la valeur du paramètre DOIT être incluse entre guillemets.

Le multipart/chiffré DOIT consister en exactement deux parties. La première partie de corps MIME doit avoir un type de contenu de "application/pgp-encrypted". Ce corps contient les informations de contrôle. Un message qui se conforme à la présente norme DOIT contenir un champ "Version: 1" dans ce corps. Comme le format de paquet PGP contient toutes les autres informations nécessaires pour le déchiffrement, aucune autre information n'est requise ici.

La seconde partie de corps MIME DOIT contenir la données chiffrées réelles. Elle doit être étiquetée avec un type de contenu de "application/octet-stream".

Exemple de message :

From: Michael Elkins <elkins@aero.org>
To: Michael Elkins <elkins@aero.org>
Mime-Version: 1.0
Content-Type: multipart/encrypted; boundary=foo;
protocol="application/pgp-encrypted"

--foo
Content-Type: application/pgp-encrypted

Version: 1

--foo
Content-Type: application/octet-stream

-----DÉBUT DU MESSAGE PGP-----
Version: 2.6.2

hIwDY32hYGCE8MkBA/wOu7d45aUxF4Q0RKJprD3v5Z9K1YcRJ2fve87lMlDlx4OjeW4GDdBfLbJE7VUpp13N19GL
8e/AqbyyjHH4aS0YoTk10QQ9nnRvjY8nZL3MPXSZg9VGQxFeGqzykzmykU6A26MSMexR4ApeeON6xzZWfo+0yOq
Aq6lb46wsvldZ96YAAABH78hyX7YX4uT1tNCWEIIBoqqvCeIMpp7UQ2IzBrXg6GtukS8NxbukLeamqVW31yt21DYO
juLzcMNe/JNsD9vDVCvOOG3OCi8=
=zzaA
-----FIN DU MESSAGE PGP-----

--foo--

5. Données signées en PGP

Les messages signés en PGP sont notés par le type de contenu "multipart/signé", décrit dans la [RFC1991], avec un paramètre "protocole" qui DOIT avoir une valeur de "application/pgp-signature" (qui DOIT être entre guillemets). Le paramètre "micalg" DOIT avoir une valeur de "pgp-<hash-symbol>", où <hash-symbol> identifie la vérification d'intégrité du message (MIC, *Message Integrity Check*) utilisée pour générer la signature. Les valeurs actuellement définies pour <hash-symbol> sont "md5" pour la somme de contrôle MD5, et "sha1" pour l'algorithme SHA.1.

Le corps multipart/signé DOIT consister en exactement deux parties. La première partie contient les données signées en format canonique MIME, incluant un ensemble d'en-têtes de contenu appropriés qui décrivent les données.

Le second corps DOIT contenir la signature numérique PGP. Il DOIT être étiqueté avec un type de contenu de "application/pgp-signature".

Lorsque la signature numérique PGP est générée :

(1) Les données à signer doivent d'abord être converties en forme canonique spécifique du type/sous-type. Pour text/plain, cela signifie la conversion en un ensemble approprié de jeu de caractères et une conversion des fins de ligne à la séquence canonique <CR><LF>.

(2) Un Contenu-Transfert-Codage approprié est alors appliqué. Chaque ligne des données codées DOIT se terminer par la séquence canonique <CR><LF>.

(3) Les en-têtes de contenu MIME sont alors ajoutés au corps, chacun se terminant par la séquence canonique <CR><LF>.

(4) Comme décrit dans la [RFC1991], la signature numérique DOIT être calculée à la fois sur les données à signer et l'ensemble des en-têtes de contenu.

(5) La signature DOIT être générée séparément des données signées afin que le processus n'altère en aucune manière les données signées.

Exemple de message :

From: Michael Elkins <elkins@aero.org>
To: Michael Elkins <elkins@aero.org>
Mime-Version: 1.0
Content-Type: multipart/signed; boundary=bar; micalg=pgp-md5;
protocol="application/pgp-signature"

--bar
& Content-Type: text/plain; charset=iso-8859-1
& Content-Transfer-Encoding: quoted-printable
&
& =A1Hola!
&
& Sais que de parler tout seul est un signe de sénilité ?
&
& C'est généralement une bonne idée de coder les lignes qui commencent par
& From=20 parce que certains agents de transport vont insérer un signe plus grand que
& (>), ce qui invalide la signature.
&
& Aussi, dans certains cas, il peut être souhaitable de coder toutes les espaces blanches en queue =20
& qui surviennent sur les lignes afin de s'assurer que =20
& la signature du message n'est pas invalidée lors du passage =20
& d'une passerelle qui modifie de telles espaces (comme BITNET). =20
&
& moi

--bar
Content-Type: application/pgp-signature

-----DÉBUT DU MESSAGE PGP-----

Version: 2.6.2

iQCVAwUBMJrRF2N9oWBghPDJAQE9UQQAtl7LuRVndBjrk4EqYBIb3h5QXIX/LC//jJV5bNvkZIGPIcEmI5iFd9boEg
vpirHtIREEqLQRkYNoBActFBZmh9GC3C041WGquMbrbxc+nIs1TIKlA08rVi9ig/2Yh7LFrK5Ein57U/W72vgSxLhe/zh
dfolT9BrnHOxEa44b+EI=
=ndaj
-----FIN DU MESSAGE PGP-----

--bar--

Les "&" dans l'exemple précédent indiquent la portion des données sur laquelle la signature a été calculée.

Bien que ce ne soit pas exigé, c'est généralement une bonne idée d'utiliser le codage Quoted-Printable dans la première étape (écrire les données à signer en format canonique MIME) si une des lignes des données commence par "From ", et de coder le "F". Cela va éviter qu'un MTA insère un ">" en tête de la ligne, ce qui invaliderait la signature !

À réception d'un message signé, une application DOIT :

(1) Convertir les terminaisons de ligne en la séquence canonique <CR><LF> avant que la signature puisse être vérifiée. Ceci est nécessaire car le MTA local peut les avoir converties selon une convention locale de terminaison de ligne.

(2) Passer les données signées et leurs en-têtes de contenu associés ainsi que la signature PGP au service de vérification de signature.

6. Données chiffrées et signées

Il est parfois désirable à la fois de signer numériquement et de chiffrer un message à envoyer. Le présent protocole permet deux méthodes pour accomplir cette tâche.

6.1 Encapsulation de la RFC1847

Dans la [RFC1847], il est déclaré que les données devraient d'abord être signées comme un corps multipart/signature, puis chiffrées pour former le corps multipart/encrypted final, c'est-à-dire :

Content-Type: multipart/encrypted;
protocol="application/pgp-encrypted"; boundary=foo
--foo
Content-Type: application/pgp-encrypted
Version: 1
--foo
Content-Type: application/octet-stream
-----DÉBUT DU MESSAGE PGP-----
& Content-Type: multipart/signed; micalg=pgp-md5
& protocol="application/pgp-signature"; boundary=bar
&
& --bar
& Content-Type: text/plain; charset=us-ascii
&
& Ce message a été d'abord signé, puis chiffré.
&
& --bar
& Content-Type: application/pgp-signature
&
& -----DÉBUT DU MESSAGE PGP-----
& Version: 2.6.2
&
& iQCVAwUBMJrRF2N9oWBghPDJAQE9UQQAtl7LuRVndBjrk4EqYBIb3h5QXIX/LC//
& jJV5bNvkZIGPIcEmI5iFd9boEgvpirHtIREEqLQRkYNoBActFBZmh9GC3C041WGq
& uMbrbxc+nIs1TIKlA08rVi9ig/2Yh7LFrK5Ein57U/W72vgSxLhe/zhdfolT9Brn
& HOxEa44b+EI=
& =ndaj

```
& -----FIN DU MESSAGE PGP-----
&
& --bar--
-----FIN DU MESSAGE PGP        -----
--foo--
```

(Le texte précédé par '&' indique qu'il est en fait chiffré mais présenté comme du texte pour être clair.)

6.2 Méthode combinée

Les versions 2.x de PGP permettent aussi que les données soient signées et chiffrées en une seule opération. Cette méthode est un raccourci acceptable, et présente l'avantage de réduire la redondance. Les données résultantes devraient être formées comme un objet "multipart/chiffré" comme décrit ci-dessus.

Il est EXIGÉ que les messages qui sont chiffrés et signés de cette façon suivent les mêmes règles de canonisation que pour les objets multipart/signés.

Il est explicitement permis à un agent de déchiffrer un message combiné et de le réécrire comme objet multipart/signé en utilisant les données de signature incorporées dans la version chiffrée.

7. Distribution des clés publiques PGP

Content-Type: application/pgp-keys
Paramètres exigés : aucun
Paramètres facultatifs : aucun

C'est le type de contenu qui devrait être utilisé pour relayer les blocs de clés publiques.

8. Notes

PGP et Pretty Good Privacy sont des marques commerciales de Philip Zimmermann.

9. Considérations pour la sécurité

L'utilisation de ce protocole est sujette aux mêmes considérations de sécurité que PGP, et il n'est pas connu qu'elle augmentent ou diminue la sécurité des messages qui l'utilisent ; voir des détails dans la [RFC1991].

10. Adresse de l'auteur

Michael Elkins
P.O. Box 92957 - M1/102
Los Angeles, CA 90009-2957
USA
téléphone : +1 310 336 8040
fax : +1 310 336 4402

Références

[RFC1847] J. Galvin, S. Murphy, S. Crocker, N. Freed, "Sécurité multiparties pour MIME : multipartie/signée et multipartie/chiffrée", octobre 1995. *(P.S.)*

[RFC1848] S. Crocker, N. Freed, J. Galvin et S. Murphy, "Services de sécurité d'objet MIME", octobre 1995. *(Historique)*

[RFC1991] D. Atkins et autres, "Formats d'échange de message PGP", août 1996. (*Obsolète, voir* RFC4880) *(Information)*