Groupe de travail Réseau
Request for Comments : 3156
RFC mise à jour : 2015
Catégorie : En cours de normalisation
Traduction Claude Brière de L'Isle

M. Elkins, Network Associates, Inc.
D. Del Torto, CryptoRights Foundation
R. Levien, University of California at Berkeley
T. Roessler
août 2001

Sécurité MIME avec OpenPGP

Résumé
Le présent document décrit comment le format de message OpenPGP peut être utilisé pour assurer la confidentialité et l'authentification en utilisant les types de contenu de sécurité des extensions multi objets de la messagerie Internet (MIME, *Multipurpose Internet Mail Extensions*) décrits dans la RFC1847.

Table des Matières

1. Introduction

Le travail sur l'intégration de PGP *(Pretty Good Privacy)* dans MIME [RFC2046] (incluant le type de contenu "application/pgp" retiré depuis) subissait avant la [RFC2015] un certain nombre de problèmes, dont le plus significatif était l'incapacité à récupérer des corps de message signés sans analyser les structures de données spécifiques de PGP. La RFC2015 utilise la solution élégante proposée dans la [RFC1847], qui définit les formats de sécurité multipartie pour MIME. Les multiparties de sécurité séparent clairement le corps de message signé de la signature, et ont un certain nombre d'autre propriétés désirables. Le présent document révise la RFC2015 pour adapter l'intégration de PGP dans MIME aux besoins qui ont émergé durant les travaux sur la spécification OpenPGP.

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

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

2. Formats de données OpenPGP

Les mises en œuvre de OpenPGP peuvent générer un résultat soit en blindage ASCII (décrit dans la [RFC2440]) soit en binaire à 8 bits lors du chiffrement des données, générant une signature numérique, ou extrayant les données de clé publique. Le résultat en blindage ASCII est la méthode EXIGÉE pour les transferts de données. Cela permet aux utilisateurs qui n'ont pas les moyens d'interpréter les formats décrits dans le présent document d'être capables d'extraire et d'utiliser les informations d'OpenPGP 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 OpenPGP multi-part de blindage ASCII.

3. Restrictions au codage de transfert de contenu

Les multipart/signed et multipart/encrypted sont à traiter par les agents comme opaques, ce qui signifie que les données ne doivent être altérées d'aucune façon [RFC1847], [RFC2480]. 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 la conversion en Quoted-Printable ou en Base64. Cela pose de sérieux problèmes pour multipart/signed, en particulier, quand la signature est invalidée lorsque survient une telle opération. Pour cette raison, toutes les données signées conformément à ce protocole DOIVENT être restreintes à 7 bits (les données en 8 bits DOIVENT être codées en utilisant Quoted-Printable ou Base64). Noter que cela inclut aussi le cas où un objet signé est aussi chiffré (voir la section 6). Cette restriction va augmenter la probabilité que la signature soit valide à la réception.

De plus, les mises en œuvre DOIVENT s'assurer qu'aucune espace blanche n'est présente en queue après que le codage MIME a été appliqué.

Note : Dans la plupart des cas, les espaces blanches en queue peuvent être soit retirées, soit protégées en appliquant un codage de transfert de contenu approprié. Cependant, il faut faire particulièrement attention lorsque des lignes d'en-têtes, soit dans les en-têtes d'entité MIME, soit dans les en-têtes incorporés de la RFC 822, sont présentes et qui ne consistent qu'en des espaces : de telles lignes doivent être entièrement retirées, car les remplacer par des lignes vides les transformeraient en délimiteurs d'en-tête, et changerait la sémantique du message. Les restrictions sur les espaces sont nécessaires pour rendre invariant le hachage calculé avec les mécanismes de signature en modes texte et binaire fournis par OpenPGP [RFC2440]. Aussi, elles aident à éviter les problèmes de compatibilité avec les mises en œuvre de PGP qui précèdent la spécification de OpenPGP.

Note : Si une ligne commence par la chaîne "From ", il est fortement conseillé d'appliquer le codage MIME Quoted-Printable ou Base64. Si Quoted-Printable est utilisé, au moins un des caractères de la chaîne devrait être codé en utilisant la règle de codage hexadécimal. Cela parce que de nombreux agents de transfert et de livraison de messagerie traitent "From " (le mot "from" suivi immédiatement par un caractère espace) comme le début d'un nouveau message et insèrent donc un crochet angulaire (>) devant toute ligne commençant par "From " pour distinguer ce cas, ce qui invalide la signature.

Les données qui sont SEULEMENT à chiffrer peuvent contenir des caractères à 8 bits et des espaces en queue et n'ont donc pas besoin de subir la conversion au format à 7 bits, et la suppression des espaces blanches.

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

4. Données chiffrées OpenPGP

Avant le chiffrement OpenPGP, les données sont écrites dans le format canonique MIME (corps et en-têtes).

Les données chiffrées OpenPGP sont notées par le type de contenu "multipart/encrypted", décrit dans la [RFC1847], et DOIVENT avoir une valeur de paramètre "protocol" de "application/pgp-encrypted". Noter que la valeur du paramètre DOIT être enclose entre guillemets.

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

La seconde partie de corps MIME DOIT contenir les 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/wOu7d45aUxF4Q0RKJprD3v5Z9K1YcRJ2fve87lMlDlx4Oj
eW4GDdBfLbJE7VUpp13N19GL8e/AqbyyjHH4aS0YoTk10QQ9nnRvjY8nZL3MPXSZ
g9VGQxFeGqzykzmykU6A26MSMexR4ApeeON6xzZWfo+0yOqAq6lb46wsvldZ96YA
AABH78hyX7YX4uT1tNCWEIIBoqqvCeIMpp7UQ2IzBrXg6GtukS8NxbukLeamqVW3
1yt21DYOjuLzcMNe/JNsD9vDVCvOOG3OCi8=
=zzaA
-----FIN DU MESSAGE PGP-----

--foo--
```

5. Données signées OpenPGP

Les messages OpenPGP signés sont notés par le type de contenu "multipart/signed", décrit dans la [RFC1847], avec un paramètre "protocol" qui DOIT avoir une valeur de "application/pgp-signature" (DOIT être entre guillemets).

Le paramètre "micalg" pour le protocole "application/pgp-signature" DOIT contenir exactement un symbole de hachage du format "pgp-<hash-identifier>", où <hash-identifier> identifie l'algorithme de vérification d'intégrité de message (MIC, *Message Integrity Check*) utilisé pour générer la signature. Les symboles de hachage sont construits à partir des noms textuels enregistrés dans la [RFC2440] ou selon le mécanisme défini dans le présent document en convertissant le nom textuel en minuscules et en le faisant précéder des quatre caractères "pgp-".

Les valeurs actuellement définies sont "pgp-md5", "pgp-sha1", "pgp-ripemd160", "pgp-md2", "pgp-tiger192", et "pgp-haval-5-160".

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

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

Note : Les mises en œuvre peuvent générer des "signatures d'un document en texte canonique" ou des "signatures d'un document binaire", comme défini dans la [RFC2440]. Les restrictions sur le matériel signé avancées à la section 3 et

dans la présente section vont s'assurer que les diverses variantes d'algorithme de MIC spécifiées dans la [RFC2440] et la [RFC1991] vont toutes produire le même résultat.

Lorsque la signature numérique OpenPGP est générée :

(1) Les données à signer DOIVENT d'abord être converties en sa forme canonique spécifique du type de contenu. Pour text/plain *(texte en clair)*, cela signifie la conversion en un jeu de caractères approprié et la conversion des fins de ligne à la séquence canonique <CR><LF>.

(2) Un codage de transfert de contenu approprié est ensuite appliqué ; voir la section 3. En particulier, les terminaisons de ligne dans les données codées DOIVENT utiliser la séquence canonique <CR><LF> lorsque c'est approprié (noter que la terminaison de ligne canonique peut être ou non présente sur la dernière ligne des données codées et elle NE DOIT PAS être incluse dans la signature si elle est absente).

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

(4) Comme décrit à la section 3, toute espace en queue DOIT être alors retirée du matériel signé.

(5) Comme décrit dans la [RFC1847], la signature numérique DOIT être calculée sur les données à signer sur leur ensemble d'en-têtes de contenu.

(6) La signature DOIT être générée séparée des données signées afin que le traitement n'altère en aucune façon les données signées.

Note : La convention OpenPGP acceptée est que les données signées se terminent par une séquence <CR><LF>. Noter que la séquence <CR><LF> qui précède immédiatement une ligne de délimiteur de frontière MIME est considérée comme faisant partie du délimiteur au paragraphe 5.1 de la [RFC2046]. Donc, elle ne fait pas partie des données signées qui précèdent la ligne de délimiteur. Une mise en œuvre qui choisit de suivre la convention OpenPGP doit s'assurer qu'elle insère une paire <CR><LF> à la dernière ligne des données à signer et transmettre (le message signé et le message transmis DOIVENT être identiques).

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-tu que 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 de messagerie vont insérer un signe
& supérieur à (>), ce qui invalide la signature.
&
& Aussi, dans certains cas, il peut être souhaitable de coder toute espace =20
& en queue qui survient sur les lignes afin de s'assurer  =20
& que la signature du message n'est pas invalidée lors du passage =20
& d'une passerelle qui modifie de telles espaces blanches (comme BITNET). =20
&
& me

--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--
```

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

À réception d'un message signé, une application DOIT :
(1) Convertir les terminaisons de ligne en séquences canoniques <CR><LF> avant que la signature puisse être vérifiée. C'est nécessaire car le MTA local peut avoir converti en une convention de fin de ligne locale.
(2) Passer les données signées et leurs en-têtes de contenu associé avec la signature OpenPGP au service de vérification de signature.

6. Données chiffrées et signées

Il est parfois souhaitable de signer numériquement et de chiffrer ensuite un message à envoyer. Ce protocole permet deux méthodes pour accomplir cette tâche.

6.1 Encapsulation selon la RFC1847

Dans la [RFC1847], il est déclaré que les données sont d'abord signées comme un corps multipart/signature, et ensuite chiffrées pour former le corps final multipart/encrypted. Ceci est très utile pour la transmission d'un message standard conforme à MIME.

Exemple :

```
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 d'abord été signé, et ensuite 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 réellement chiffré, mais présenté comme texte dans un souci de clarté.)

6.2 Méthode combinée

Le format de paquet OpenPGP [RFC2440] décrit une méthode pour signet et chiffrer les données dans un seul message OpenPGP. Cette méthode est permise afin de réduire la charge de traitement et augmenter la compatibilité avec les mises en œuvre non MIME de OpenPGP. Les données résultantes sont formatées comme un objet "multipart/encrypted" comme décrit à la Section 4.

Les messages qui sont chiffrés et signés de cette façon combinée DOIVENT suivre les mêmes règles de canonisation que les objets multipart/signed.

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

7. Distribution des clés publiques OpenPGP

Type de contenu : application/pgp-keys
Paramètres exigés : aucun
Paramètres facultatifs : aucun

Une partie de corps MIME du type de contenu "application/pgp-keys" contient des paquets de clé publique transférables à blindage ASCII comme défini au paragraphe 10.1 de la [RFC2440].

8. Considérations pour la sécurité

Les signatures d'un document de texte canonique comme défini dans la [RFC2440] ignorent les espaces blanches en queue dans le matériel signé. Les mises en œuvre qui choisissent d'utiliser les signatures de document de texte canonisé ne seront pas capables de détecter l'ajout d'une espace blanche dans le transit.

Voir les [RFC2046], [RFC1848] pour plus d'informations sur les considérations de sécurité qui concernent les protocoles sous-jacents.

9. Considérations relatives à l'IANA

Le présent document définit trois types de supports : "application/pgp-encrypted", "application/pgp-signature" et "application/pgp-keys". Les paragraphes qui suivent spécifient les enregistrements de l'IANA pour ces types.

9.1 Enregistrement du type de support application/pgp-encrypted

Nom du type de support MIME : application
Nom du sous-type MIME : pgp-encrypted
Paramètres exigés : aucun
Paramètres facultatifs : aucun
Considérations de codage : Actuellement, ce type de support consiste toujours en une seule chaîne de texte en 7 bits.

Considérations de sécurité : Voir la Section 8 et la section 13 de la RFC2440.
Considérations d'interopérabilité : aucune
Spécification publiée : Le présent document.
Informations supplémentaires :
Numéro magique : aucun
Extension de fichier : aucune
Code de type de fichier Macintosh : aucun
Adresse de la personne & de messagerie à contacter pour d'autres informations : Michael Elkins mél : me@cs.hmc.edu
Usage prévu : courant
Auteur/contrôleur des changements : Michael Elkins mél : me@cs.hmc.edu

9.2 Enregistrement du type de support application/pgp-signature

Nom du type de support MIME : application
Nom du sous-type MIME : pgp-signature
Paramètres exigés : aucun
Paramètres facultatifs : aucun
Considérations de codage : Le contenu de ce type de support consiste toujours en texte à 7 bits.
Considérations de sécurité : Voir la Section 8 et la section 13 de la RFC2440.
Considérations d'interopérabilité : aucune
Spécification publiée : La RFC2440 et le présent document.
Informations supplémentaires :
Numéro magique : aucun
Extensions de fichier : asc, sig
Code de type de fichier Macintosh : pgDS
Adresse de la personne & de messagerie à contacter pour d'autres informations : Michael Elkins mél : me@cs.hmc.edu
Usage prévu : courant
Auteur/contrôleur des changements : Michael Elkins mél : me@cs.hmc.edu

9.3 Enregistrement du type de support application/pgp-keys

Nom du type de support MIME : application
Nom du sous-type MIME : pgp-keys
Paramètres exigés : aucun
Paramètres facultatifs : aucun
Considérations de codage : Le contenu de ce type de support consiste toujours en texte à 7 bits.
Considérations de sécurité : Voir la Section 8 et la section 13 de la RFC2440.
Considérations d'interopérabilité : aucune
Spécification publiée : La RFC2440 et le présent document.
Informations supplémentaires :
Numéro magique : aucun
Extensions de fichier : asc
Code de type de fichier Macintosh : aucun
Adresse de la personne & de messagerie à contacter pour d'autres informations : Michael Elkins mél : me@cs.hmc.edu
Usage prévu : courant
Auteur/contrôleur des changements : Michael Elkins mél : me@cs.hmc.edu

10. Notes

"PGP" et "Pretty Good Privacy" sont des marques commerciales enregistrées de Network Associates, Inc.

11. Remerciements

Le présent document s'appuie sur les travaux de définition du format de message OpenPGP du groupe de travail OpenPGP de l'IETF. Le format de message OpenPGP est actuellement décrit par la [RFC2440].

Des remerciements particuliers sont dus à : Philip Zimmermann pour son travail original et ses travaux en cours sur PGP ;

Charles Breed, Jon Callas et Dave Del Torto pour avoir à l'origine proposé la formation du groupe de travail OpenPGP ; et à Steve Schoenfeld pour ses retours utiles durant le processus d'élaboration du projet. Les auteurs tiennent aussi à remercier les ingénieurs de Pretty Good Privacy, Inc (maintenant Network Associates, Inc) incluant Colin Plumb, Hal Finney, Jon Callas, Mark Elrod, Mark Weaver et Lloyd Chambers, pour leurs commentaires techniques.

Des remerciements supplémentaires sont dus à Jeff Schiller et Derek Atkins pour leur soutien continu du chiffrement fort et du logiciel libre PGP au MIT ; à Rodney Thayer de Sable Technology ; à John Noerenberg, Steve Dorner et Laurence Lundblade de l'équipe Eudora à QUALCOMM, Inc ; à Bodo Moeller pour sa proposition de l'approche suivie à l'égard des espaces blanches en queue ; à John Gilmore, Hugh Daniel et Fred Ringel (de Rivertown) et Ian Bell (de Turnpike) pour leurs commentaires critiques au bon moment ; et aux membres internationaux de la liste de diffusion OpenPGP de l'IETF, parmi lesquels William Geiger, Lutz Donnerhacke et Kazu Yamamoto. L'idée d'utiliser multipart/mixed avec multipart/signed a été attribuée à James Galvin. Finalement, notre gratitude es due aux nombreux membres des listes de diffusion "Cypherpunks," "Coderpunks" et "pgp-users" <http://cryptorights.org/pgp-users> et aux nombreux utilisateurs de PGP dans le monde entier pour nous avoir aidé à garder ouvert le chemin de la confidentialité.

12. Adresses des auteurs et du président du groupe de travail OpenPGP

Le groupe de travail OpenPGP peut être contacté via le président actuel :

John W. Noerenberg II
Qualcomm, Inc.
5775 Morehouse Dr.
San Diego, CA 92121 USA
téléphone : +1 619 658 3510
mél : jwn2@qualcomm.com

Les principaux auteurs de ce document sont :

Dave Del Torto
CryptoRights Foundation
80 Alviso Street, Mailstop: CRF
San Francisco, CA 94127 USA
téléphone : +1.415.334.5533, vm: #2
mél : ddt@cryptorights.org, ddt@openpgp.net

Michael Elkins
Network Associates, Inc.
3415 S. Sepulveda Blvd Suite 700
Los Angeles, CA 90034 USA
téléphone : +1.310.737.1663
mél : me@cs.hmc.edu, Michael_Elkins@NAI.com

Raph Levien
University of California at Berkeley
579 Soda Hall
Berkeley, CA 94720 USA
téléphone : +1.510.642.6509
mél : raph@acm.org

Thomas Roessler
Nordstrasse 99
D-53111 Bonn, Germany
téléphone : +49-228-638007
mél : roessler@does-not-exist.org

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.

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

[RFC2015] M. Elkins, "Sécurité de MIME avec Pretty Good Privacy (PGP)", octobre 1996. (*MàJ par* RFC3156) *(P.S.)*

[RFC2046] N. Freed et N. Borenstein, "Extensions de messagerie Internet multi-objets (MIME) Partie 2 : Types de support", novembre 1996. *(D. S., MàJ par* 2646, 3798, 5147, 6657*.)*

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

[RFC2440] J. Callas, L. Donnerhacke, H. Finney et R. Thayer, "Format de message OpenPGP", novembre 1998. *(Obsolète, voir la* RFC4880*)*

[RFC2480] N. Freed, "Les routeurs et le traitement de multiparties de sécurité MIME", janvier 1999. *(P.S.)*

Déclaration complète de droits de reproduction

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