Groupe de travail Réseau
Request for Comments : 3923
Catégorie : En cours de normalisation

P. Saint-Andre, éd., Jabber Software Foundation
octobre 2004
Traduction Claude Brière de L'Isle

Signature de bout en bout et chiffrement d'objet pour le protocole extensible de messagerie et de présence (XMPP)

Statut de ce mémoire

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

Résumé

Le présent mémoire définit les méthodes de signature de bout en bout et de chiffrement d'objet pour le protocole extensible de messagerie instantanée et de présence (XMPP).

Table des Matières

1. Introduction

Le présent mémoire définit les méthodes de signature et de chiffrement de bout en bout d'objet pour le protocole extensible de messagerie instantanée et de présence (XMPP). (Pour des informations sur XMPP, voir les [RFC3920] et [RFC3921].) La méthode spécifiée ici permet à un envoyeur de signer et/ou de chiffrer un message instantané envoyé à un receveur spécifique, de signer et/ou chiffrer les informations de présence qui sont dirigées sur un usager spécifique, et de signer et/ou chiffrer toute strophe XMPP arbitraire dirigée sur un usager spécifique. Le présent mémoire aide par là les spécifications de XMPP à satisfaire aux exigences de la [RFC2779].

1.1 Terminologie

Le présent document hérite de la terminologie définie dans les [RFC2778], [RFC3851], [RFC3852], et [RFC3920].

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].

2. Exigences

Pour les besoins du présent mémoire, on stipule les exigences suivantes :

1. La méthode définie DOIT viser les exigences minimale de signature et de chiffrement sur la messagerie instantanée et présence, comme définies dans la [RFC2779]. En particulier, la méthode DOIT viser les exigences suivantes, qui sont copiées directement de la [RFC2779] :
   * Le protocole DOIT donner les moyens de s'assurer qu'un message reçu (Notification de message ou Message Instantané) n'a pas été corrompu ou altéré (paragraphe 2.5.1).
   * Le protocole DOIT donner les moyens de s'assurer qu'un message reçu (Notification ou Message Instantané) n'a pas été enregistré et répété par un adversaire (paragraphe 2.5.2).
   * Le protocole DOIT fournir des moyens de s'assurer qu'un message envoyé (Notification ou Message Instantan) n'est lisible que par les entités que permet l'envoyeur (paragraphe 2.5.3).
   * Le protocole DOIT permettre à tout client d'utiliser les moyens de s'assurer de la non corruption, non répétition, et confidentialité, mais le protocole NE DOIT PAS exiger que tout client utilise ces moyens tout le temps (paragraphe 2.5.4).
   * Lorsque A établit un abonnement aux informations de présence de B, le protocole DOIT fournir à A le moyen de vérifier la bonne réception du contenu que B choisit de divulguer à A (paragraphe 5.1.4).
   * Le protocole DOIT fournir à A les moyens de vérifier que les informations de présence sont correctes, comme envoyées à B (paragraphe 5.3.1).
   * Le protocole DOIT fournir à A le moyen de s'assurer qu'aucun autre principal C ne peut voir le contenu du message (paragraphe 5.4.6).
   * Le protocole DOIT fournir à A le moyen de s'assurer qu'aucun autre principal C ne peut altérer le message, et à B le moyen de vérifier qu'aucune altération ne s'est produite (paragraphe 5.4.7).

2. La méthode définie DOIT permettre l'interopérabilité avec les systèmes de messagerie non XMPP qui prennent en charge les spécifications de présence commune et de messagerie instantanée (CPIM) publiées par le groupe de travail Messagerie instantanée et présence (IMPP, *Instant Messaging and Presence Protocols*). Deux corollaires de cette exigence sont :
   * Avant de signer et/ou chiffrer, le format d'un message instantané DOIT se conformer au format de message CPIM défini dans la [RFC3862].
   * Avant de signer et/ou chiffrer, le format des informations de présence DOIT se conformer au format des données d'information de présence CPP défini dans la [RFC3863].

3. La méthode DOIT suivre les procédures requises (incluant les algorithmes spécifiques) définis dans la [RFC3860] et la [RFC3859]. En particulier, ces documents spécifient :
   * que signer DOIT utiliser les signatures de la [RFC3851] avec les SignedData de la [RFC3852].
   * que chiffrer DOIT utiliser le chiffrement de la [RFC3851] avec les EnvelopedData de la [RFC3852].

4. Afin de permettre l'interopérabilité des mises en œuvre, les applications envoyeuses et receveuses DOIVENT mettre en œuvre les algorithmes spécifiés au paragraphe 6.10 "Algorithmes de chiffrement de mise en œuvre obligatoire".

On déclare de plus que les fonctionnalités suivantes sortent du domaine d'application du présent mémoire :
o La découverte de la prise en charge de ce protocole. Une entité peut découvrir si une autre entité prend le présent protocole

en charge en (1) tentant d'envoyer des strohes signées ou chiffrées et en recevant une strophe d'erreur (découverte "technique") ou un message textuel en réponse (découverte "sociale") si le protocole n'est pas pris en charge, ou (2) en utilisant un protocole de découverte de service dédié, comme [DISCO] ou [CAPS]. Cependant, la définition d'un protocole de découverte de service sort du domaine d'application du présent mémoire.

- La signature ou le chiffrement des messages d'un groupe de débats XMPP, qui sont mentionnés dans la [RFC3921] mais qui ne sont pas définis ici car ils ne sont pas requis par la [RFC2779] ; de tels messages sont mieux spécifiés dans [MUC].
- La signature ou le chiffrement des messages de présence diffusés comme décrit dans la [RFC3921] (les méthodes définies ici s'appliquent seulement à la présence dirigée).
- La signature ou le chiffrement de communications qui surviennent dans des contextes d'application autres que la messagerie instantanée et la présence comme ceux décrits dans les [RFC2778] et [RFC2779].

3. Sécurisation des messages

3.1 Processus de sécurisation des messages

Dans le but de signer et/ou chiffrer un message, un agent envoyeur DOIT utiliser la procédure suivante :

1. Générer un objet "Message/CPIM" comme défini dans la [RFC3862].
2. Signer et/ou chiffrer les en-têtes et le contenu de l'objet "Message/CPIM" comme spécifié dans l'exigence 3 de la Section 2 ci-dessus.
3. Fournir l'objet résultant signé et/ou chiffré au sein d'une section CDATA XML (voir le paragraphe 2.7 de [XML]) contenu dans un enfant <e2e/> d'une strophe <message/>, où l'élément <e2e/> est qualifié par l'espace de noms 'urn:ietf:params:xml:ns:xmpp-e2e' comme spécifié plus complètement à la Section 9.

3.2 Exemple de message signé

L'exemple suivant illustre les étapes définies pour la signature d'un message .

D'abord, l'agent envoyeur génère un objet "Message/CPIM" conformément aux règles et formats spécifiés dans la [RFC3862].

Exemple 1 : L'envoyeur génère un objet "Message/CPIM" :

```
Content-type: Message/CPIM
From: Juliet Capulet <im:juliet@example.com>
To: Romeo Montague <im:romeo@example.net>
DateTime: 2003-12-09T11:45:36.66Z
Subject: Imploring
Content-type: text/plain; charset=utf-8
Content-ID: <1234567890@example.com>
Wherefore art thou, Romeo?
```

Une fois que l'agent envoyeur a généré l'objet "Message/CPIM", l'agent envoyeur peut le signer. Le résultat est un objet multiparties [RFC3851] (voir la [RFC1847]) qui a un type de contenu de "multipart/signed" et comporte deux parties : une dont le type de contenu est "Message/CPIM" et une autre dont le type de contenu est "application/pkcs7-signature".

Exemple 2 : L'envoyeur génère un objet multipart/signed :

```
Content-Type: multipart/signed; boundary=next;
    micalg=sha1;
    protocol=application/pkcs7-signature

-- ensuite
Content-type: Message/CPIM

From: Juliet Capulet <im:juliet@example.com>
To: Romeo Montague <im:romeo@example.net>
DateTime: 2003-12-09T23:45:36.66Z
Subject: Imploring

Content-type: text/plain; charset=utf-8
Content-ID: <1234567890@example.com>
```

```
Wherefore art thou, Romeo?
--ensuite
Content-Type: application/pkcs7-signature
Content-Disposition: attachment;handling=required;\
      filename=smime.p7s

[partie de corps signée]

--ensuite--
```

L'agent envoyeur enveloppe l'objet "multipart/signed" dans la section CDATA XML, qui est contenue dans un élément <e2e/> qui est inclus dans un élément fils de la strophe de message XMPP et qui est qualifiée par l'espace de noms 'urn:ietf:params:xml:ns:xmpp-e2e'.

Exemple 3 : L'envoyeur génère une strophe de message XMPP :

```
<message to='romeo@example.net/orchard' type='chat'>
<e2e xmlns='urn:ietf:params:xml:ns:xmpp-e2e'>
<![CDATA[
Content-Type: multipart/signed; boundary=next;
      micalg=sha1;
       protocol=application/pkcs7-signature

--ensuite
Content-type: Message/CPIM

From: Juliet Capulet <im:juliet@example.com>
To: Romeo Montague <im:romeo@example.net>
DateTime: 2003-12-09T23:45:36.66Z
Subject: Imploring

Content-type: text/plain; charset=utf-8
Content-ID: <1234567890@example.com>

Wherefore art thou, Romeo?
--ensuite
Content-Type: application/pkcs7-signature
Content-Disposition: attachment;handling=required;\
      filename=smime.p7s

[partie de corps signée]

--ensuite--
]]>
</e2e>
</message>
```

3.3 Exemple de message chiffré

L'exemple suivant illustre les étapes définies pour chiffrer un message.

D'abord, l'agent envoyeur génère un objet "Message/CPIM" conformément aux règles et formats spécifiés dans la [RFC3862].

Exemple 4 : L'envoyeur génère un objet "Message/CPIM" :

```
Content-type: Message/CPIM

From: Juliet Capulet <im:juliet@example.com>
To: Romeo Montague <im:romeo@example.net>
DateTime: 2003-12-09T11:45:36.66Z
Subject: Imploring
```

```
Content-type: text/plain; charset=utf-8
Content-ID: <1234567890@example.com>

Wherefore art thou, Romeo?
```

Une fois que l'agent envoyeur a généré l'objet "Message/CPIM", l'agent envoyeur peut le chiffrer.

Exemple 5 : L'envoyeur génère un objet chiffré :

```
U2FsdGVkX19okeKTlLxa/1n1FE/upwn1D20GhPWqhDWlexKMUKYJInTWzERP+vcQ
/OxFs40uc9Fx81a5/62p/yPb/UWnuG6SR6o3Ed2zwcusDImyyz125HFERdDUMBC9
Pt6Z4cTGKBmJzZBGyuc3Y+TMBTxqFFUAxeWaoxnZrrl+LP72vwbriYc3KCMxDbQL
Igc1Vzs5/5JecegMieNY24SlNyX9HMFRNFpbI64vLxYEk55A+3IYbZsluCFT31+a
+GeAvJkvH64LRV4mPbUhENTQ2wbAwnOTvbLIaQEQrii78xNEh+MK8Bx7TBTvi4yH
Ddzf9Sim6mtWsXaCAvWSyp0X91d7xRJ4JIgKfPzkxNsWJFCLthQS1p734eDxXVd3
i08lEHzyll6htuEr59ZDAw==
```

L'agent envoyeur envoloppe maintenant l'objet chiffré dans une section CDATA XML, qui est contenue dans un élément <e2e/> qui est inclus comme élément fils de la strophe de message XMPP et qui est qualifiée par l'espace de noms 'urn:ietf:params:xml:ns:xmpp-e2e'.

Exemple 6 : L'envoyeur génère une strophe de message XMPP :

```
<message to='romeo@example.net/orchard' type='chat'>
<e2e xmlns='urn:ietf:params:xml:ns:xmpp-e2e'>
<![CDATA[
U2FsdGVkX19okeKTlLxa/1n1FE/upwn1D20GhPWqhDWlexKMUKYJInTWzERP+vcQ
/
OxFs40uc9Fx81a5/62p/yPb/UWnuG6SR6o3Ed2zwcusDImyyz125HFERdDUMBC9Pt6Z4cTGKBmJzZBGyuc3Y+TMBTx
qFFUAxeWaoxnZrrl+LP72vwbriYc3KCMxDbQL
Igc1Vzs5/5JecegMieNY24SlNyX9HMFRNFpbI64vLxYEk55A+3IYbZsluCFT31+a
+GeAvJkvH64LRV4mPbUhENTQ2wbAwnOTvbLIaQEQrii78xNEh+MK8Bx7TBTvi4yH
Ddzf9Sim6mtWsXaCAvWSyp0X91d7xRJ4JIgKfPzkxNsWJFCLthQS1p734eDxXVd3
i08lEHzyll6htuEr59ZDAw==
]]>
</e2e>
</message>
```

4. Sécurisation de présence

4.1 Processus de sécurisation des informations de présence

Afin de signer et/ou chiffrer les informations de présence, un agent envoyeur DOIT utiliser la procédure suivante :

1. Générer un objet "application/pidf+xml" comme défini dans la [RFC3863].

2. Signer et/ou chiffrer l'objet "application/pidf+xml" comme spécifié à l'exigence 3 de la Section 2 ci-dessus.

3. Fournir l'objet résultant signé et/ou chiffré dans une section CDATA XML (voir le paragraphe 2.7 de [XML]) contenue dans un fils <e2e/> d'une strophe <presence/>, où l'élément <e2e/> est qualifié par l'espace de noms 'urn:ietf:params:xml:ns:xmpp-e2e'. La strophe <presence/> DOIT inclure un attribut 'to', c'est-à-dire qu'elle doit être une instance de présence dirigée comme défini dans la [RFC3921].

4.2 Exemple d'informations de présence signées

L'exemple suivant illustre les étapes définies pour signer les informations de présence.

D'abord, l'agent envoyeur génère un objet "application/pidf+xml" conformément aux règles et formats spécifiés dans la [RFC3863].

Exemple 7 : L'envoyeur génère un objet "application/pidf+xml" :

```
<?xml version="1.0" encoding="UTF-8"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
    xmlns:im="urn:ietf:params:xml:ns:pidf:im"
    entity="pres:juliet@example.com">
  <tuple id="hr0zny"
   <status>
   <basic>open</basic>
    <im:im>away</im:im>
   </status>
   <note xml:lang="en">retired to the chamber</note>
   <timestamp>2003-12-09T23:53:11.31</timestamp>
  </tuple>
</presence>
```

Une fois que l'agent envoyeur a généré l'objet "application/pidf+xml", l'agent envoyeur peut le signer. Le résultat est un objet multipart de la [RFC3851] (voir la [RFC1847]) qui a un type de contenu de "multipart/signed" et inclut deux parties : une dont le type de contenu est "application/pidf+xml" et l'autre dont le type de contenu est "application/pkcs7-signature".

Exemple 8 : L'envoyeur génère l'objet multipart/signed :

```
Content-Type: multipart/signed; boundary=next;
     micalg=sha1;
      protocol=application/pkcs7-signature

--ensuite
Content-type: application/pidf+xml
Content-ID: <2345678901@example.com>

<xml version="1.0" encoding="UTF-8"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
     xmlns:im="urn:ietf:params:xml:ns:pidf:im"
     entity="pres:juliet@example.com">
  <tuple id="hr0zny">
   <status&gt;
   <basic>open</basic>
   <im:im>away</im:im>
  </status>
  <note xml:lang="en">retired to the chamber</note>
  <timestamp>2003-12-09T23:53:11.31Z</timestamp>
 </tuple>
</presence>
--ensuite
Content-Type: application/pkcs7-signature
Content-Disposition: attachment;handling=required;\
                     filename=smime.p7s

[partie de corps signé]

--ensuite--
```

L'agent envoyeur enveloppe maintenant l'objet "multipart/signed" dans une section CDATA XML ; elle est contenue dans un élément <e2e/> qui est inclus dans un élément fils de la strophe de message XMPP et qui est qualifié par l'espace de noms 'urn:ietf:params:xml:ns:xmpp-e2e'.

Exemple 9 : L'envoyeur génère une strophe de présence XMPP :

```
<presence to='romeo@example.net/orchard'>
<e2e xmlns='urn:ietf:params:xml:ns:xmpp-e2e'>
<![CDATA[
Content-Type: multipart/signed; boundary=next;
       micalg=sha1;
       protocol=application/pkcs7-signature
```

```
--ensuite
Content-type: application/pidf+xml
Content-ID: <2345678901@example.com>

<xml version="1.0" encoding="UTF-8"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
        xmlns:im="urn:ietf:params:xml:ns:pidf:im"
        entity="pres:juliet@example.com">
  <tuple id="hr0zny">
   <status>
     <basic>open</basic>
     <im:im>away</im:im>
   </status>
   <note xml:lang="en">retired to the chamber</note>
   <timestamp>2003-12-09T23:53:11.31Z</timestamp>
  </tuple>
</presence>
--ensuite
|   Content-Type: application/pkcs7-signature
Content-Disposition: attachment;handling=required;\
                    filename=smime.p7s

[partie de corps signée]

--ensuite--
]]>
</e2e>
</presence>
```

4.3 Exemple d'informations de présence chiffrées

L'exemple suivant illustre les étapes définies poiur le chiffrement des informations de présence.

D'abord, l'agent envoyeur génère un objet "application/pidf+xml" conformément aux règles et formats spécifiés dans la [RFC3863].

Exemple 10 : l'envoyeur génère un objet "application/pidf+xml" :

```
<?xml version="1.0" encoding="UTF-8"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
      xmlns:im="urn:ietf:params:xml:ns:pidf:im"
       entity="pres:juliet@example.com">
  <tuple id="hr0zny"
  <status>
     <basic>open</basic>
    <im:im>away</im:im>
   </status>
    <note xml:lang="en">retired to the chamber</note>
    <timestamp>2003-12-09T23:53:11.31</timestamp>
  </tuple>
</presence>
```

Une fois que l'agent envoyeur a généré l'objet "application/pidf+xml", l'agent envoyeur peut le chiffrer.

Exemple 11 : L'envoyeur génère un objet chiffré :

```
U2FsdGVkX18VJPbx5GMdFPTPZrHLC9QGiVP+ziczu6zWZLFQxae6O5PP6iqpr2No
zOvBVMWvYeRAT0zd18hr6qsqKiGl/GZpAAbTvPtaBxeIykxsd1+CX+U+iw0nEGCr
bjiQrk0qUKJ79bNxwRnqdidjhyTpKSbOJC0XZ8CTe7AE9KDM3Q+uk+O3jrqX4byL
GBlKThbzKidxz32ObojPEEwfFiM/yUeqYUP1OcJpUmeQ8lcXhD6tcx+m2MAyYYLP
boKQxpLknxRnbM8T/voedlnFLbbDu69mOlxDPbr1mHZd3hDsyFudb1fb4rI3Kw0K
```

```
Nq+3udr2IkysviJDgQo+xGIQUG/5sED/mAaPRlj4f/JtTzvT4EaQTawv69ntXfKV
MCr9KdIMMdjdJzOJkYLoAhNVrcZn5tw8WsJGwuKuhYb/SShy7InzOapPaPAl7/Mm
PHj7zj3NZ6EEIweDOuAwWlIG/dT506tci27+EW7JnXwMPnFMkF+6a7tr/0Y+iiej
woJxUIBqCOgX+U7srHpK2NYtNTZ7UQp2V0yEx1JV8+Y=
```

L'agent envoyeur enveloppe maintenant l'objet chiffré dans une section CDATA XML, qui est contenue dans un élément <e2e/> ; celui-ci est inclus dans un élément fils de la strophe de message XMPP qui est qualifié par l'espace de noms 'urn:ietf:params:xml:ns:xmpp-e2e'.

Exemple 12 : L'envoyeur génère une strophe de présence XMPP :

```
<presence to='romeo@example.net/orchard'>
  <e2e xmlns='urn:ietf:params:xml:ns:xmpp-e2e'>
<![CDATA[
U2FsdGVkX18VJPbx5GMdFPTPZrHLC9QGiVP+ziczu6zWZLFQxae6O5PP6iqpr2No
zOvBVMWvYeRAT0zd18hr6qsqKiGl/GZpAAbTvPtaBxeIykxsd1+CX+U+iw0nEGCr
bjiQrk0qUKJ79bNxwRnqdidjhyTpKSbOJC0XZ8CTe7AE9KDM3Q+uk+O3jrqX4byL
GBlKThbzKidxz32ObojPEEwfFiM/yUeqYUP1OcJpUmeQ8lcXhD6tcx+m2MAyYYLP
boKQxpLknxRnbM8T/voedlnFLbbDu69mOlxDPbr1mHZd3hDsyFudb1fb4rI3Kw0K
Nq+3udr2IkysviJDgQo+xGIQUG/5sED/mAaPRlj4f/JtTzvT4EaQTawv69ntXfKV
MCr9KdIMMdjdJzOJkYLoAhNVrcZn5tw8WsJGwuKuhYb/SShy7InzOapPaPAl7/Mm
PHj7zj3NZ6EEIweDOuAwWlIG/dT506tci27+EW7JnXwMPnFMkF+6a7tr/0Y+iiej
woJxUIBqCOgX+U7srHpK2NYtNTZ7UQp2V0yEx1JV8+Y=
]]>
  </e2e>
</presence>
```

5. Sécurisation de données XMPP arbitraires

Les sections précédentes de ce mémoire décrivent comment sécuriser "le plus petit commun dénominateur" des données de messagerie instantanée et de présence de la sorte de celles qui peuvent être directement traduites dans les formats MSGFMT ou PIDF. Cependant, XMPP possède un troisième type de strophe de niveau de base (<iq/>) en plus de <message/> et <presence/>, ainsi que la capacité à inclure des données XML étendues au sein d'éléments fils arbitraires des trois types de strophes centraux. Donc, il serait souhaitable de sécuriser ces données si possible.

Parce que la [RFC3862] spécifie la capacité à encapsuler tout type MIME, l'approche retenue par le présent mémoire est d'inclure des données XMPP arbitraires dans un type de support XML appelé "application/xmpp+xml" comme spécifié plus en détails à la Section 10.

L'exemple suivant illustre la structure du type MIME "application/xmpp+xml". (Noter que l'espace de noms 'http://jabber.org/protocol/evil' utilisé dans ces exemples est associé à un protocole de "April Fool" écrit pour être l'équivalent en messagerie instantanée de la RFC 3514 ; il n'est inclus que comme instance d'informations étendues incluses dans une strophe XML et ne devrait pas être pris sérieusement comme une extension XMPP fonctionnelle.)

Exemple 13 : Strophe de message avec données d'extension dans un type MIME "application/xmpp+xml" :

```
<?xml version='1.0' encoding='UTF-8'?>
<xmpp xmlns='jabber:client'>
  <message
      from='iago@example.com/pda'
      to='emilia@example.com/cell'>
      <body>
      I told him what I thought, and told no more
      Than what he found himself was apt and true.
    </body>
    <evil xmlns='http://jabber.org/protocol/evil'/>
  </message>
</xmpp>
```

Exemple 14 : Strophe de présence avec données d'extension contenues dans le type MIME "application/xmpp+xml" :

```
<?xml version='1.0' encoding='UTF-8'?>
<xmpp xmlns='jabber:client'>
  <presence from='iago@example.com/pda'>
    <show>dnd</show>
    <status>Fomenting dissension</status>
    <evil xmlns='http://jabber.org/protocol/evil'/>
  </presence>
</xmpp>
```

Exemple 15 : Strophe IQ avec données d'extension contenues dans le type MIME "application/ xmpp+xml" :

```
<?xml version='1.0' encoding='UTF-8'?>
<xmpp xmlns='jabber:client'>
  <iq type='result'
      from='iago@example.com/pda'
      to='emilia@example.com/cell'
      id='evil1'>
    <query xmlns='jabber:iq:version'>
      <name>Stabber</name>
      <version>666</version>
      <os>FiendOS</os>
    </query>
    <evil xmlns='http://jabber.org/protocol/evil'/>
  </iq>
</xmpp>
```

Tout comme avec les objets "Message/CPIM" et "application/pidf+xml", l'objet "application/xmpp+xml" serait signé et/ou chiffré, puis encapsulé au sein d'une section CDATA XML (voir le paragraphe 2.7 de [XML]) contenue dans un fils <e2e/> d'une strophe <presence/>, où l'élément <e2e/> est qualifié par l'espace de noms 'urn:ietf:params:xml:ns:xmpp-e2e'.

6. Règles pour la génération et le traitement de S/MIME

6.1 Obtention de certificat

La [RFC3851] ne spécifie pas comment obtenir un certificat d'une autorité de certification, mais rend obligatoire que tout agent envoyeur ait déjà un certificat. Le groupe de travail PKIX a, au moment de la rédaction du présent mémoire, produit deux normes séparées pour l'inscription de certificat : la [RFC2510] et la [RFC2797]. La méthode à utiliser pour l'insciption de certificat sort du domaine d'application du présent mémoire.

6.2 Restitution de certificat

Un agent receveur DOIT fournir un mécanisme de restitution de certificat afin d'obtenir l'accès aux certificats pour les receveurs d'enveloppes numériques. Le présent mémoire ne traite pas de la façon dont les agents S/MIME traitent les certificats, mais seulement de ce qu'ils font après qu'un certificat a été validé ou rejeté. Les questions de certification S/MIME sont traitées dans la [RFC3850].

Cependant, au minimum, pour le déploiement initial de S/MIME, un agent d'utilisateur DEVRAIT automatiquement générer un message à un receveur désigné pour demander le certificat du receveur dans un message de retour signé. Les agents receveurs et envoyeurs DEVRAIENT aussi fournir un mécanisme pour permettre à un utilisateur de "mémoriser et protéger" les certificats pour les correspondants d'une façon telle qu'elle garantisse leur restitution ultérieure.

6.3 Noms des certificats

Les certificats d'entité d'extrémité utilisés par les entités XMPP dans le contexte du présent mémoire DEVRAIENT contenir une adresse valide de messagerie instantanée et de présence. L'adresse DEVRAIT être spécifiée à la fois comme un URI 'im:' (pour la messgerie instantanée, comme défini dans la [RFC3860]) et un URI 'pres:' (pour présence, comme défini dans la [RFC3859]) ; chacun de ces URI DEVRAIT être spécifié dans une entrée séparée de GeneralName de type uniformResourceIdentifier à l'intérieur du subjectAltName (c'est-à-dire, deux entrées séparées). Les informations dans le nom distinctif sujet DEVRAIENT être ignorées.

Chaque URI DOIT être de la forme <im:adresse> ou <pres:adresse>, où la portion "adresse" est une adresse XMPP (aussi appelée un identifiant Jabber ou JID) comme défini dans la [RFC3920], précédé du schéma d'URI 'im:' ou 'pres:'. L'adresse DEVRAIT être de la forme <nœud@domaine> (c'est-à-dire, un "JID nu") mais toute forme de JID valide PEUT être utilisée.

La valeur du JID contenue dans l'attribut XMPP 'from' DOIT correspondre à un JID fourni dans le certificat du signataire, avec l'exception que la portion identifiant de ressource du JID contenu dans l'attribut 'from' DEVRAIT être ignorée pour les besoins de la confrontation.

Les agents receveurs DOIVENT vérifier que le JID envoyeur correspond à un JID fourni dans le certificat du signataire, avec l'exception que la portion identifiant de ressource du JID contenu dans l'attribut 'from' DEVRAIT être ignorée pour les besoins de la confrontation. Un agent receveur DEVRAIT fournir un traitement de remplacement explicite de la strophe si cette comparaison échoue, qui peut être d'afficher un message informant le receveur des adresses qui sont dans le certificat ou d'autres détails du certificat.

L'extension de nom de remplacement du sujet est utilisée dans S/MIME comme moyen préféré pour porter l'adresse de messagerie instantanée et présence qui correspond à l'entité pour ce certificat. Toute adresse XMPP présente dans le certificat DOIT être codée en utilisant l'identifiant d'objet ASN.1 "id-on-xmppAddr" comme spécifié au paragraphe 5.1.1 de la [RFC3920].

6.4 Codage de transfert

Comme il est prévu que les applications XMPP n'aient pas d'interface avec les anciens systèmes à 7 bits, le codage de transfert (comme défini au paragraphe 3.1.2 de la [RFC3851]) DOIT être "binary".

6.5 Ordre de signature et de chiffrement

Si une strophe est à la fois signée et chiffrée, elle DEVRAIT être d'abord signée, puis chiffrée.

6.6 Inclusion des certificats

Si l'envoyeur et le receveur sont impliqués dans une session active de messagerie sur un certaine durée, l'agent envoyeur DEVRAIT inclure le certificat de l'envoyeur avec au moins sa dernière strophe de message chiffrée toutes les cinq minutes. En dehors du contexte d'une session active de messagerie, l'agent envoyeur DEVRAIT inclure le certificat de l'envoyeur avec chaque strophe de message chiffrée. Un agent envoyeur PEUT inclure le certificat de l'envoyeur avec chaque strophe de présence chiffrée. Cependant, un agent envoyeur NE DEVRAIT PAS inclure un certificat plus d'une fois toutes les cinq minutes.

6.7 Rattachement et vérification des signatures

Les agents envoyeurs DEVRAIENT attacher une signature à chaque strophe XML chiffrée. Si une signature est attachée, un champ d'en-tête Content-Disposition (comme défini dans la [RFC2183]) DEVRAIT être inclus pour spécifier comment la signature va être traitée par l'application receveuse.

Si l'agent receveur détermine que la signature attachée à une strophe XML est invalide, il NE DEVRAIT PAS présenter la strophe au receveur désigné (humain ou application) ; il DEVRAIT fournir un traitement de remplacement explicite de la strophe (qui peut être d'afficher un message informant le receveur que la signature attachée est invalide) et PEUT retourner une erreur de strophe à l'envoyeur comme décrit à la Section 7 "Traitement des erreurs chez le receveur".

6.8 Déchiffrement

Si l'agent receveur n'est pas capable de déchiffrer la strophe XML chiffrée, il NE DEVRAIT PAS présenter la strophe au receveur désigné (humain ou application), DEVRAIT fournir un traitement de remplacement explicite de la strophe (qui peut être d'afficher un message informant le receveur qu'il a reçu une strophe qui ne peut pas être déchiffrée), et PEUT retourner une erreur de strophe à l'envoyeur comme décrit à la Section 7 "Traitement des erreurs chez le receveur".

6.9 Inclusion et vérification des horodatages

Des horodatages sont inclus dans les objets "Message/CPIM" et "application/pidf+xml" pour aider à prévenir les attaques en répétition. Tous les horodatages DOIVENT se conformer à la [RFC3339] et être présentés en UTC sans décalage, incluant les

fractions de seconde comme approprié. En l'absence d'un ajustement local à l'heure perçue par l'agent envoyeur ou de l'heure locale sous-jacente, l'agent envoyeur DOIT s'assurer que l'horodatage qu'il envoie au receveur augmente de façon monotone (si nécessaire en incrémentant les fractions de seconde dans l'horodatage si l'horloge retourne la même heure pour plusieurs demandes). Les règles suivantes s'appliquent à l'application receveuse:

- Elle DOIT vérifier que l'horodatage reçu est dans les cinq minutes de l'heure actuelle.

- Elle DEVRAIT vérifier que l'horodatage reçu est supérieur à tout horodatage reçu dans les dix dernières minutes qui ont suivi la dernière vérification.

- Si une des vérifications précédentes échoue, l'horodatage DEVRAIT être présenté à l'entité receveuse (humain ou application) marqué avec "viel horodatage", "futur horodatage", ou "horodatage décroissant", et l'entité receveuse PEUT retourner une erreur de strophe à l'envoyeur comme décrit à la Section 7 "Traitement des erreurs chez le receveur".

6.10 Algorithmes de chiffrement de mise en œuvre obligatoire

Toutes les mises en œuvre DOIVENT prendre en charge les algorithmes suivants. Les mises en œuvre PEUVENT aussi prendre en charge d'autres algorithmes.

Pour les SignedData CMS :
- Le résumé de message SHA-1 comme spécifié au paragraphe 2.1 de la [RFC3370].
- RSA (PKCS n° 1 v1.5) avec l'algorithme de signature SHA-1, comme spécifié au paragraphe 3.2 de la [RFC3370].

Pour les EnvelopedData CMS :
- Le transport de clé RSA (PKCS n° 1 v1.5) comme spécifié au paragraphe 4.2.1 de la [RFC3370].
- L'algorithme de chiffrement AES-128 en mode CBC, comme spécifié dans la [RFC3565].

7. Traitement des erreurs chez le receveur

Lorsque une entité XMPP reçoit une strophe XML contenant des données qui sont signées et/ou chiffrées en utilisant le protocole décrit ici, plusieurs scénarios sont possibles :

Cas n° 1 : L'application receveuse ne comprend pas le protocole.

Cas n° 2 : L'application receveuse comprend le protocole et est capable de déchiffrer la charge utile et de vérifier la signature de l'envoyeur.

Cas n° 3 : L'application receveuse comprend le protocole et est capable de déchiffrer la charge utile et de vérifier la signature de l'envoyeur, mais l'horodatage échoue à la vérification spécifiée ci-dessus au paragraphe 6.9, "vérification des horodatages".

Cas n° 4 : L'application receveuse comprend le protocole et est capable de déchiffrer la charge utile mais est incapable de vérifier la signature de l'envoyeur.

Cas n° 5 : L'application receveuse comprend le protocole mais est incapable de déchiffrer la charge utile.

Dans le cas n° 1, l'application receveuse DOIT faire une seule des choses suivantes : (1) ignorer l'extension <e2e/>, (2) ignorer la strophe entière, ou (3) retourner une erreur <service-unavailable/> à l'envoyeur, comme décrit dans [RFC3920].

Dans le cas n° 2, l'application receveuse NE DOIT PAS retourner une erreur de strophe à l'envoyeur, car c'est le cas de succès.

Dans le cas n° 3, l'application receveuse PEUT retourner une erreur <not-acceptable/> à l'envoyeur (comme décrit dans la [RFC3920]), complétée facultativement par un élément de condition d'erreur spécifique de l'application <bad-timestamp/> comme montré ci-dessous.

Exemple 16 : Le receveur retourne l'erreur <not-acceptable/> :

```
<message from='romeo@example.net/orchard' type='chat'>
  <e2e xmlns='urn:ietf:params:xml:ns:xmpp-e2e'>
  [CDATA section here]
```

```
  </e2e>
  <error type='modify'>
   <not-acceptable xmlns='urn:ietf:params:xml:ns:xmpp-stanzas'/>
   <bad-timestamp xmlns='urn:ietf:params:xml:xmpp-e2e'/>
  </error>
 </message>
```

Dans le cas n° 4, l'application receveuse DEVRAIT retourner une erreur <not-acceptable/> à l'envoyeur (comme décrit dans la [RFC3920]), complétée facultativement par un élément de condition d'erreur spécifique de l'application <unverified-signature/> *(signature non vérifiée)* comme montré ci-dessous :

Exemple 17 : Le receveur retourne l'erreur <not-acceptable/> :

```
<message from='romeo@example.net/orchard' type='chat'>
  <e2e xmlns='urn:ietf:params:xml:ns:xmpp-e2e'>
  [La section CDATA vient ici]
  </e2e>
  <error type='modify'>
   <not-acceptable xmlns='urn:ietf:params:xml:ns:xmpp-stanzas'/>
   <unverified-signature xmlns='urn:ietf:params:xml:xmpp-e2e'/>
  </error>
 </message>
```

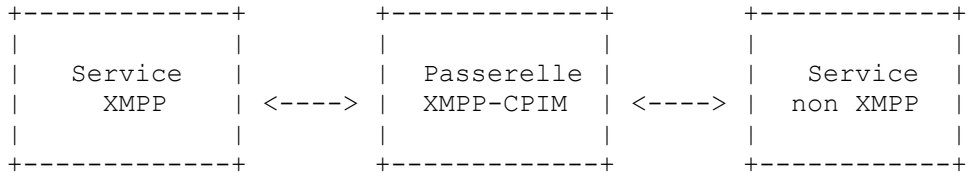
Dans le cas n° 5, l'application receveuse DEVRAIT retourner une erreur <bad-request/> à l'envoyeur (comme décrit dans la [RFC3920]), complétée facultativement par un élément de condition d'erreur spécifique de l'application <decryption-failed/> *(échec du déchiffrement)* comme montré ci-dessous :

Exemple 18 : Le receveur retourne l'erreur <bad-request/> :

```
<message from='romeo@example.net/orchard' type='chat'>
  <e2e xmlns='urn:ietf:params:xml:ns:xmpp-e2e'>
  [La section CDATA vient ici]
  </e2e>
  <error type='modify'>
   <bad-request xmlns='urn:ietf:params:xml:ns:xmpp-stanzas'/>
   <decryption-failed xmlns='urn:ietf:params:xml:xmpp-e2e'/>
  </error>
 </message>
```

8. Communications sûres à travers une passerelle

Une méthode courante pour réaliser l'interopérabilité entre deux services disparates est d'utiliser une "passerelleé qui interprête les protocoles de chaque service et traduit chacun dans le protocole de l'autre. Les spécifications de CPIM (précisément les [RFC3862] et [RFC3863] définissent les profils communs à utiliser pour l'interopérabilité entre les services de messagerie instantanée et de présence qui se conforment à la [RFC2779]. Dans le cas de communications entre un service XMPP et un service non XMPP, on peut visualiser cette relation comme suit :



La méthode de chiffrement de bout en bout définie ici permet l'échange de messages instantanés et de présence chiffrés et/ou signés à travers une passerelle XMPP-CPIM. En particulier :

- lorsque une passerelle reçoit un message ou strophe de présence XMPP sécurisé du service XMPP qui est adressé à un usager sur le service non XMPP, elle DOIT retirer "l'enveloppe" XMPP " (tout jusque aux étiquettes <e2e> et </e2e> incluses) afin de révéler l'objet multipart S/MIME, puis acheminer l'objet au service non XMPP (en l'enveloppant d'abord dans le protocole utilisé par le service non XMPP si nécessaire) ;
- lorsque une passerelle reçoit un message instantané ou document de présence sécurisé non XMPP du service non XMPP

qui est adressé à un usager sur le service XMPP, elle DOIT retirer "l'enveloppe" non XMPP (si il en est) afin de révéler l'objet multipart S/MIME, envelopper l'object dans une "enveloppe" de message ou strophe de présence XMPP (incluant les étiquettes <e2e> et </e2e>) puis acheminer la strophe XMPP au service XMPP.

L'objet S/MIME enveloppé DOIT être immuable et NE DOIT PAS être modifié par une passerelle XMPP-CPIM.

9. Espace de noms urn:ietf:params:xml:xmpp-e2e

L'élément <e2e xmlns='urn:ietf:params:xml:ns:xmpp-e2e'/> est une enveloppe pour une section CDATA XML (voir le paragraphe 2.7 de [XML]) qui contient un objet "Message/CPIM", "application/pidf+xml", ou "application/xmpp+xml". Donc, l'espace de noms 'urn:ietf:params:xml:xmpp-e2e' n'a pas de sémantique inhérente, et la sémantique de l'objet encapsulé est définie par une des spécifications suivantes :

- la [RFC3862] pour "Message/CPIM"
- la [RFC3863] pour "application/pidf+xml"
- la [RFC3920] pour "application/xmpp+xml"

Bien que le type de support "application/xmpp+xml" soit spécifié dans le présent document, l'élément <xmpp/> est simplement une enveloppe pour une strophe <message/>, <presence/>, ou <iq/>, où la sémantique de ces types de strophes est spécifiée dans la [RFC3920].

Étant donné que l'espace de noms 'urn:ietf:params:xml:ns:xmpp-e2e' n'a pas de sémantique inhérente et spécifie seulement un protocole utilisateur, la gestion des versions est de la responsabilité des protocoles qui définissent les objets encapsulés ([RFC3862], [RFC3863], et [RFC3920]).

10. Type de support application/xmpp+xml

Le type de support "application/xmpp+xml" adhère aux lignes directrices spécifiées dans la [RFC3023]. L'élément racine pour ce type MIME est <xmpp/>, et l'élément racine DOIT contenir un et un seul élément fils, correspondant à un des types de strophe XMPP (c'est-à-dire, message, presence, ou iq) si l'espace de noms par défaut est 'jabber:client' ou 'jabber:server' comme défini dans la [RFC3920]. Le codage de caractères pour le type de support XML DOIT être UTF-8, conformément au paragraphe 11.5 de la [RFC3920].

11. Considérations sur la sécurité

Le présent mémoire est entièrement consacré à la sécurité. Des considérations de sécurité détaillées pour les protocoles de messagerie instantanée et de présence sont données aux paragraphes 5.1 à 5.4 de la [RFC2779], et pour XMPP en particulier sont données aux paragraphes 12.1 à 12.6 de la [RFC3920]. De plus, toutes les considérations sur la sécurité spécifiées dans la [RFC3023] s'appliquent au type de support "application/xmpp+xml".

La méthode de sécurité de bout en bout définie ici PEUT résulter en l'échange sécurisé de messages instantanés et d'informations de présence à travers une passerelle qui met en œuvre les spécifications CPIM. Une telle passerellle DOIT se conformer aux exigences minimum de sécurité des protocoles de messagerie instantanée et de présence entre lesquels elle assure l'interface.

12. Considérations relatives à l'IANA

12.1 Nom d'expace de noms XML pour données e2e dans XMPP

Un sous espace de noms d'URN de contenus signés et chiffrés pour le protocole extensible de messagerie instantanée et présence (XMPP) est défini comme suit. (Ce nom d'espace de noms adhère au format défini dans la [RFC3688].)

URI : urn:ietf:params:xml:ns:xmpp-e2e
Spécification : RFC 3923
Description : C'est un nom d'espace de noms XML de contenu signé et chiffré pour le protocole extensible de messagerie instantanée et de présence comme défini par la RFC 3923.
Contact d'enregistrement : IESG, < iesg@ietf.org >

12.2 Enregistrement de type de contenu pour "application/xmpp+xml"

Pour : ietf-types@iana.org
Objet : Enregistrement du type de support MIME application/xmpp+xml
Nom de type de support MIME : application
Nom de sous type MIME : xmpp+xml
Paramètres exigés : aucun
Paramètres facultatifs : (jeu de caractère) même paramètre de jeu de caractère de application/xml que spécifié dans la RFC 3023 ; selon le paragraphe 11.5 de la [RFC3920], le jeu de caractères doit être UTF-8.
Considérations de codage : les mêmes que celles spécifiées pour application/xml dans la RFC 3023 ; selon le paragraphe 11.5 de la [RFC3920], le codage doit être UTF-8.

Considérations de sécurité : toutes les considérations de sécurité spécifiée dans les RFC 3023 et RFC 3920 s'appliquent à ce type de support XML. Voir la Section 11 de la RFC 3923.
Considérations d'interopérabilité : aucune
Spécification : RFC 3923
Applications qui utilisent ce type de support : systèmes de messagerie instanée et de présence cmpatibles XMPP.
Information s suppélemntaires : aucune
Adresse personnelle et de messagerie à contacter pour plus d'informations : IESG, < iesg@ietf.org >
Utilisation prévue : COURANTE
Auteur/contrôleur des changements : IETF, groupe de travail XMPP.

13. Références

13.1 Références normatives

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

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

[RFC2183] R. Troost, S. Dorner, K. Moore, éd., "Communication des informations de présentation dans les messages Internet : le champ d'en-tête Contenu-disposition", août 1997. (*MàJ par* RFC2184, RFC2231) *(P.S.)*

[RFC2778] M. Day, J. Rosenberg et H. Sugano, "Modèle pour Presence et la messagerie instantanée", février 2000.

[RFC2779] M. Day et autres, "Exigences des protocoles Messagerie instantanée / Presence", février 2000. *(Information)*

[RFC3023] M. Murata, S. St.Laurent et D. Kohn, "Types de support XML", janvier 2001. (*Obsolète, voir* RFC7303)

[RFC3339] G. Klyne, C. Newman, "La date et l'heure sur l'Internet : horodatages", juillet 2002. *(P.S.)*

[RFC3370] R. Housley, "Algorithmes de syntaxe de message cryptographique (CMS)", août 2002. *(P.S.)*

[RFC3565] J. Schaad, "Utilisation de l'algorithme de chiffrement de la norme de chiffrement évolué (AES) dans la syntaxe de message cryptographique (CMS)", juillet 2003. *(P.S.)*

[RFC3850] B. Ramsdell, éd., "Traitement de certificat d'extensions multi-objets/sécurisées de messagerie Internet (S/MIME) version 3.1", juillet 2004. *(P.S.) (Remplacée par* RFC5750)

[RFC3851] B. Ramsdell, "Spécification du message d'extensions de messagerie Internet multi-objets/sécurisé (S/MIME) version 3.1", juillet 2004. *(Remplacée par* RFC5751)

[RFC3852] R. Housley, "Syntaxe de message cryptographique (CMS)", juillet 2004. *(Remplacée par la* RFC5652)

[RFC3859] J. Peterson, "Profil commun pour les services de présence (CPP)", août 2004. *(P.S.)*

[RFC3860] J. Peterson, "Profil commun pour la messagerie instantanée (CPIM)", août 2004. *(P.S.)*

[RFC3862] G. Klyne, D. Atkins, "Profil commun pour la messagerie instantanée (CPIM) : format de message ", août 2004. *(P.S.)*

[RFC3863] H. Sugano et autres, "Format des données d'information de présence (PIDF)", août 2004.

[RFC3920] P. Saint-Andre, éd., "Protocole de messagerie et de présence extensibles (XMPP) : éléments centraux", octobre 2004. *(P.S.) (Remplacée par la* RFC6120)

[RFC3921] P. Saint-Andre, éd., "Protocole de messagerie et de présence extensibles (XMPP) : messagerie instantanée et présence", octobre 2004. *(P.S.) (Remplacée par la* RFC6121)

13.2 Références pour information

[CAPS] Hildebrand, J. and P. Saint-Andre, "Entity Capabilities", JSF JEP 0115, août 2004.

[DISCO] Hildebrand, J., Millard, P., Eatmon, R. and P. Saint-Andre, "Service Discovery", JSF JEP 0030, juillet 2004.

[MUC] Saint-Andre, P., "Multi-User Chat", JSF JEP 0045, juin 2004.

[RFC2510] C. Adams, S. Farrell, "Protocoles de gestion de certificat d'infrastructure de clé publique X.509 sur l'Internet", mars 1999. (*Obsolète, voir* RFC4210) *(P.S.)*

[RFC2797] M. Myers, X. Liu, J. Schaad et J. Weinstein, "Messages de gestion de certificat sur CMS", avril 2000. (*Obsolète, voir* RFC5272, *P.S*)

[RFC3688] M. Mealling, "Registre XML de l'IETF", BCP 81, janvier 2004.

[XML] Bray, T., Paoli, J., Sperberg-McQueen, C. and E. Maler, "Extensible Markup Language (XML) 1.0 (3rd ed)", W3C REC-xml, février 2004, < http://pages.videotron.com/fyergeau/w3c/xml10/REC-xml-19980210.fr.html >.

Appendice A. Schéma pour urn:ietf:params:xml:ns:xmpp-e2e

Le schéma XML suivant est descriptif et non normatif.

```
<?xml version='1.0' encoding='UTF-8'?>

<xs:schema
   xmlns:xs='http://www.w3.org/2001/XMLSchema'
   targetNamespace='urn:ietf:params:xml:ns:xmpp-e2e'
   xmlns='urn:ietf:params:xml:ns:xmpp-e2e'
   elementFormDefault='qualified'>

  <xs:element name='e2e' type='xs:string'/>

  <xs:element name='decryption-failed' type='empty'/>
  <xs:element name='signature-unverified' type='empty'/>
  <xs:element name='bad-timestamp' type='empty'/>

  <xs:simpleType name='empty'>
    <xs:restriction base='xs:string'>
      <xs:enumeration value=''/>
    </xs:restriction>
  </xs:simpleType>

</xs:schema>
```

Adresse de l'auteur

Peter Saint-Andre (editor)
Jabber Software Foundation

mél : stpeter@jabber.org

Déclaration complète de droits de reproduction

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