Abstract

This memo defines a method for end-to-end object signing and encryption in the Extensible Messaging and Presence Protocol (XMPP).
Table of Contents

1. Introduction ................................................. 3
2. Requirements ............................................... 3
3. Securing Messages ......................................... 5
4. Securing Presence ......................................... 6
5. Securing Arbitrary XMPP Data .............................. 8
6. Rules for S/MIME Generation and Handling .............. 9
7. Secure Communications Through a Gateway ............... 12
8. Security Considerations ................................. 13
9. IANA Considerations .................................... 13
   Normative References .................................. 14
   Informative References ................................. 16
   Author’s Address ...................................... 16
A. Schema for urn:ietf:params:xml:ns:xmpp-e2e ............ 16
B. Revision History ...................................... 17
   Intellectual Property and Copyright Statements ....... 19
1. Introduction

This memo defines a method for end-to-end signing and encryption in the Extensible Messaging and Presence Protocol (XMPP). (For information about XMPP, see [XMPP-CORE] and [XMPP-IM].) The method defined herein enables a sender to encrypt and/or sign an instant message sent to a specific recipient, encrypt and/or sign presence information that is directed to a specific user, and sign presence information that is broadcasted to a specific user. This memo thereby helps the XMPP specifications meet the requirements defined in [IMP-REQS].

1.1 Terminology

This document inherits terminology defined in [SMIME], [IMP-MODEL], [CMS], and [XMPP-CORE].

The capitalized key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [TERMS].

1.2 Discussion Venue

The authors welcome discussion and comments related to the topics presented in this document. The preferred forum is the <xmppwg@jabber.org> mailing list, for which archives and subscription information are available at <http://www.jabber.org/cgi-bin/mailman/listinfo/xmppwg/>.

1.3 Intellectual Property Notice

This document is in full compliance with all provisions of Section 10 of RFC 2026. Parts of this specification use the term "jabber" for identifying namespaces and other protocol syntax. Jabber[tm] is a registered trademark of Jabber, Inc. Jabber, Inc. grants permission to the IETF for use of the Jabber trademark in association with this specification and its successors, if any.

2. Requirements

For the purposes of this memo, we stipulate the following requirements:

1. The method defined MUST address encryption and signing requirements for minimal instant messaging and presence only, as those are defined in [IMP-REQS]. The method is NOT REQUIRED to support non-IM applications of XMPP, nor to support advanced
instant messaging and presence functionality that is outside the scope of [IMP-REQS]. In particular, the method MUST address the following requirements defined in [IMP-REQS]:

* The protocol MUST provide means to ensure confidence that a received message (NOTIFICATION or INSTANT MESSAGE) has not been corrupted or tampered with. (Section 2.5.1)

* The protocol MUST provide means to ensure confidence that a received message (NOTIFICATION or INSTANT MESSAGE) has not been recorded and played back by an adversary. (Section 2.5.2)

* The protocol MUST provide means to ensure that a sent message (NOTIFICATION or INSTANT MESSAGE) is only readable by ENTITIES that the sender allows. (Section 2.5.3)

* The protocol MUST allow any client to use the means to ensure non-corruption, non-playback, and privacy, but the protocol MUST NOT require that all clients use these means at all times. (Section 2.5.4)

* When A establishes a SUBSCRIPTION to B’s PRESENCE INFORMATION, the protocol MUST provide A means of verifying the accurate receipt of the content B chooses to disclose to A. (Section 5.1.4)

* The protocol MUST provide A means of verifying that the presence information is accurate, as sent by B. (Section 5.3.1)

* The protocol MUST provide A means of ensuring that no other PRINCIPAL C can see the content of M. (Section 5.4.6)

* The protocol MUST provide A means of ensuring that no other PRINCIPAL C can tamper with M, and B means to verify that no tampering has occurred. (Section 5.4.7)

2. The method defined MUST enable interoperability with non-XMPP messaging systems that support the Common Presence and Instant Messaging (CPIM) specifications defined by the Instant Messaging and Presence (IMPP) Working Group. Therefore:

* Prior to encrypting or signing, the format of an instant message MUST conform to the CPIM Message Format defined in [MSGFMT].

* Prior to encrypting or signing, the format of presence
information MUST conform to the CPP Presence Information Data Format defined in [PIDF].

3. The method MUST follow the required procedures (including the specific algorithms) defined in [CPIM] and [CPP]. In particular, these documents specify:


4. In order to enable interoperable implementations, sending and receiving applications MUST implement the algorithms defined under Section 6.9.

3. Securing Messages

In order to encrypt a message, a sending entity MUST use the following procedure:

1. Generate a "Message/CPIM" object as defined in [MSGFMT].

2. Encrypt and/or sign both the headers and content of the "Message/CPIM" object as specified in Requirement 3 of Section 2 above.

3. Provide the resulting multipart [SMIME] object (see [MULTI]) within a CDATA section of an <e2e/> child of a <message/> stanza, where the <e2e/> element is qualified by the 'urn:ietf:params:xml:ns:xmpp-e2e' namespace.

Example 1: Sender generates "Message/CPIM" object:

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?

Example 2: Sender generates signed message (the 'from' address on the XMPP message stanza is stamped by sender’s server):
4. Securing Presence

In order to encrypt presence information, a sending entity MUST use the following procedure:

1. Generate an "application/pidf+xml" object as defined in [PIDF].

2. Encrypt and/or sign the "application/pidf+xml" object as specified in Requirement 3 of Section 2 above.

3. Provide the resulting [SMIME] object within a CDATA section of an <e2e/> child of a <presence/> stanza, where the <e2e/> element is qualified by the ‘urn:ietf:params:xml:ns:xmpp-e2e’ namespace. The <presence/> stanza MUST include a ‘to’ attribute, i.e., it must be an instance of directed presence as defined in [XMPP-IM].
Example 3: Sender generates "application/pidf+xml" object:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<pres xmlns="urn:ietf:params:xml:ns:pidf"
     xmlns:im="urn:ietf:params:xml:ns:pidf:im"
     entity="pres:juliet@example.com">
  <tuple id="h40zny">
    <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>
```

Example 4: Sender generates signed presence (the 'from' address on the XMPP presence stanza is stamped by sender’s server):

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

    <?xml version="1.0" encoding="UTF-8"?>
    <pres 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>
    </pres>
    --next
    Content-Type: application/pkcs7-signature
    Content-Disposition: attachment; handling=required; filename=smime.p7s
    ]]>  
    [signed body part]
```
5. Securing Arbitrary XMPP Data

The foregoing sections of this memo describe how to secure "least common denominator" messaging and presence data of the kind that can be directly translated into the MSGFMT or PIDF formats. However, XMPP possesses a third base-level stanza type (<iq/>) in addition to <message/> and <presence/>, as well as the ability to include extended XML data within arbitrary child elements of the three core stanza types. Therefore it would be desirable to secure such data if possible.

Because [MSGFMT] specifies the ability to encapsulate any MIME type, the approach taken in this memo is to include arbitrary XMPP data in a new MIME type, "application/xmpp+xml". The root element for this MIME type is <xmpp/>, and the root element MUST contain one and only one child element, corresponding to one of the XMPP stanza types (i.e., message, presence, or iq) if the default namespace is 'jabber:client' or 'jabber:server' as defined in [XMPP-CORE].

The following examples illustrate the structure of the "application/xmpp+xml" MIME type.

Example 5: Message stanza with extended data contained in "application/xmpp+xml" MIME type:

```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>
```

Example 6: Presence stanza with extended data contained in "application/xmpp+xml" MIME type:

```xml
<?xml version='1.0' encoding='UTF-8'?>
```
Example 7: IQ stanza with extended data contained in "application/xmpp+xml" MIME type:

```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>
```

6. Rules for S/MIME Generation and Handling

6.1 Certificate Enrollment

[SMIME] does not specify how to obtain a certificate from a certificate authority, but instead mandates that every sending agent must already have a certificate. The PKIX Working Group has, at the time of this writing, produced two separate standards for certificate enrollment: [CMP] and [CMC]. Which method to use for certificate enrollment is outside the scope of this memo.

6.2 Certificate Retrieval

A receiving agent MUST provide some certificate retrieval mechanism in order to gain access to certificates for recipients of digital envelopes. This memo does not cover how S/MIME agents handle certificates, only what they do after a certificate has been validated or rejected. S/MIME certification issues are covered in [CERT].

At a minimum, for initial S/MIME deployment, a user agent could
automatically generate a message to an intended recipient requesting that recipient’s certificate in a signed return message. Receiving and sending agents SHOULD also provide a mechanism to allow a user to "store and protect" certificates for correspondents in such a way so as to guarantee their later retrieval.

6.3 Certificate Names

End-entity certificates used by XMPP entities in the context of this memo SHOULD contain a valid instant messaging and presence address. The address SHOULD be specified as both an ‘im:’ URI (for instant messaging, as defined in [CPIM]) and a ‘pres:’ URI (for presence, as defined in [CPP]); each of these URIs SHOULD be specified in a separate GeneralName entry of type uniformResourceIdentifier inside the subjectAltName (i.e., two separate entries). Information in the subject distinguished name SHOULD be ignored.

Each URI MUST be of the form <im:address> or <pres:address>, where the "address" portion is an XMPP address (also referred to as a Jabber Identifier or JID) as defined in [XMPP-CORE], prepended with the ‘im:’ or ‘pres:’ URI scheme. The address SHOULD be of the form <node@domain> (i.e., a "bare JID"), although any valid JID form MAY be used.

The value of the JID contained in the XMPP ‘from’ attribute SHOULD match the JID provided in the signer’s certificate, with the exception that the resource identifier portion of the JID contained in the ‘from’ attribute SHOULD be ignored for matching purposes.

Receiving agents SHOULD check that sending JID matches a JID provided in the signer’s certificate, with the exception that the resource identifier portion of the JID contained in the ‘from’ attribute MAY be ignored for matching purposes. A receiving agent SHOULD provide some explicit alternate processing of the message if this comparison fails, which may be to display a message that shows the recipient the addresses in the certificate or other certificate details.

The subject alternative name extension is used in S/MIME as the preferred means to convey the instant messaging and presence address that corresponds to the entity for this certificate. Any instant messaging or presence address present in the certificate SHOULD be encoded using the otherName choice of the subjectAltName type along with a type-id of "xmipp" (as these terms are profiled in [X509]).

6.4 Transfer Encoding

Because it is expected that XMPP applications will not interface with older 7-bit systems, the transfer encoding (as defined in Section
3.1.2 of [SMIME]) MUST be "binary".

6.5 Attachment of Signatures

Sending agents SHOULD attach a signature to each encrypted message or presence stanza. If a signature is attached, a Content-Disposition header field (as defined in [DISP]) SHOULD be included to specify how the signature is to be handled by the receiving application.

6.6 Inclusion of Certificates

If the sender and recipient are involved in an active messaging session over a period of time, the sending agent SHOULD include the sender’s certificate along with at least one encrypted message stanza every five minutes. Outside the context of an active messaging session, the sending agent SHOULD include the sender’s certificate along with each encrypted message stanza. A sending agent MAY include the sender’s certificate along with each encrypted presence stanza. However, a sending agent SHOULD NOT include a certificate more than once every five minutes.

6.7 Order of Signing and Encrypting

If a stanza is both signed and encrypted, it SHOULD be signed first, then encrypted.

6.8 Checking of Timestamps

 Timestamps are included in "Message/CPIM" and "application/pidf+xml" objects to help prevent replay attacks. All timestamps MUST conform to [DATETIME] and be presented as UTC with no offset, including fractions of a second as appropriate. Absent a local adjustment to the sending application’s perceived time or the underlying clock time, the sending application MUST ensure that the timestamps it sends to the receiver increase monotonically (if necessary by incrementing the seconds fraction in the timestamp if the clock returns the same time for multiple requests). The following rules apply to the receiving application:

- it MUST verify that the timestamp received is within five minutes of the current time
- it SHOULD verify that the timestamp received is greater than any timestamp received in the last 10 minutes which passed the previous check
- if any of the foregoing checks fails, the timestamp SHOULD be presented to the receiving entity (human or via an API) marked as
"old timestamp", "future timestamp", or "decreasing timestamp"

6.9 Mandatory to Implement Technologies

At a minimum, all implementations MUST support the following CMS algorithms as defined in [CMS-ALG]:

for digest: SHA-1

for signing: RSA

for content encryption: AES

7. Secure Communications Through a Gateway

A common method for achieving interoperability between two disparate services is through the use of a "gateway" that interprets the protocols of each service and translates them into the protocols of the other. The CPIM specifications (specifically [MSGFMT] and [PIDF]) define the common profiles to be used for interoperability between instant messaging and presence services that comply with [IMP-REQS]. In the case of communications between an XMPP service and a non-XMPP service, we can visualize this relationship as follows:

```
+-------------+        +-------------+        +------------+
|             |        |             |        |            |
|    XMPP     |        |  XMPP-CPIM  |        |  Non-XMPP  |
|   Service   | <----> |   Gateway   | <----> |  Service   |
|             |        |             |        |            |
+-------------+        +-------------+        +------------+
```

The end-to-end encryption method defined herein enables the exchange of encrypted and/or signed instant messages and presence through an XMPP-CPIM gateway. In particular:

- When a gateway receives a secured XMPP message or presence stanza from the XMPP service that is addressed to a user on the non-XMPP service, it MUST remove the XMPP "wrapper" (everything down to and including the <e2e> and </e2e> tags) in order to reveal the multipart S/MIME object, then route the object to the non-XMPP service (first wrapping it in the protocol used by the non-XMPP service if necessary).

- When a gateway receives a secured non-XMPP instant message or presence document from the non-XMPP service that is addressed to a user on the XMPP service, it MUST remove the non-XMPP "wrapper"
(if any) in order to reveal the multipart S/MIME object, wrap the object in an XMPP message or presence "wrapper" (including the <e2e> and </e2e> tags), and then route the XMPP stanza to the XMPP service.

The wrapped S/MIME object MUST be immutable and MUST NOT be modified by an XMPP-CPIM gateway.

8. Security Considerations

This entire memo discusses security. Detailed security considerations for instant messaging and presence protocols are given in [IMP-REQS] (Sections 5.1 through 5.4), and for XMPP in particular are given in [XMPP-CORE] (Sections 12.1 through 12.6).

The end-to-end security method defined here MAY result in exchanging secured instant messages and presence information through a gateway that implements the CPIM specifications. Such a gateway MUST be compliant with the minimum security requirements of the instant messaging and presence protocols with which it interfaces.

9. IANA Considerations

9.1 Content-type Registration for "application/xmpp+xml"

To: ietf-types@iana.org

Subject: Registration of MIME media type application/xmpp+xml

MIME media type name: application

MIME subtype name: xmpp+xml

Required parameters: (none)

Optional parameters: charset Indicates the character encoding of the enclosed XML; the default encoding is UTF-8.

Encoding considerations: Contains XML, which can employ 8-bit characters, depending on the character encoding used.

Security considerations: Contains a message, presence information, or IQ (request-response) data in XMPP, which may be considered private. Appropriate precautions should be adopted to limit disclosure of this information.
Interoperability considerations: (none)

Specification: XXXX

Applications which use this media type: XMPP-compliant instant messaging and presence systems.

Additional information: (none)

Person and email address to contact for further information: IETF, XMPP Working Group, <xmppwg@jabber.org>

Intended usage: COMMON

Author/Change controller: IETF, XMPP Working Group

9.2 XML Namespace Name for e2e Data in XMPP

A URN sub-namespace for signed and encrypted content in the Extensible Messaging and Presence Protocol (XMPP) is defined as follows. (This namespace name adheres to the format defined in [XML-REG].)


Specification: XXXX

Description: This is the XML namespace name for signed and encrypted content in the Extensible Messaging and Presence Protocol as defined by XXXX.

Registrant Contact: IETF, XMPP Working Group, <xmppwg@jabber.org>

Normative References


[CMS] Housley, R., "Cryptographic Message Syntax (CMS)", RFC
3369, August 2002.


Informative References


Author’s Address

Peter Saint-Andre
Jabber Software Foundation
EMail: stpeter@jabber.org

Appendix A. Schema for urn:ietf:params:xml:ns:xmpp-e2e

The following XML schema is descriptive, not normative.

<?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:schema>
Appendix B. Revision History

Note to RFC Editor: please remove this entire appendix, and the corresponding entries in the table of contents, prior to publication.

B.1 Changes from draft-ietf-xmpp-e2e-06

- Specified use of SHA-1 for digest and AES for content encryption.
- Specified order of signing then encrypting.
- Specified format and checking of timestamps.
- Clarified use of subjectAltName field, where the GeneralName content is a URI of the form im:user@host and pres:user@host.
- Clarified circumstances under which certificates should be attached.
- Added Content-Disposition header field to examples.

B.2 Changes from draft-ietf-xmpp-e2e-05

- Addressed I-D nits and RFC Editor formatting.

B.3 Changes from draft-ietf-xmpp-e2e-04

- Added text about instant inbox addresses.

B.4 Changes from draft-ietf-xmpp-e2e-03

- Specified that S/MIME multipart objects are enclosed in a CDATA section.
- Changed "text/xml" to "text/plain" for message examples.
- Specified must-implement technologies, transfer encodings, certificate enrollment, certificate retrieval, and certificate names (including subjectAltName for JIDs).
- Specified requirements regarding attachment of signatures and inclusion of certificates.
- Fixed some small terminological errors.
B.5 Changes from draft-ietf-xmpp-e2e-02

- Completely revised to use formats defined in the CPIM specifications, S/MIME only, etc.

B.6 Changes from draft-ietf-xmpp-e2e-01

- Removed old Section 6 (Signalling Support via Presence) -- the ability to sign broadcasted presence made it redundant.
- Made small editorial changes to address RFC Editor requirements.

B.7 Changes from draft-ietf-xmpp-e2e-00

- Added support for all stanza types.
- Specified that the full stanza is encrypted.
- Added support for S/MIME in addition to OpenPGP.
- Specified that encrypted presence must be directed to a specific recipient.
- Specified order of encrypting and signing.
- Added support for signing broadcasted presence.
- Added IANA considerations.
- Changed namespace to 'urn:ietf:params:xml:ns:xmpp-e2e'.
Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF’s procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

Full Copyright Statement

Copyright (C) The Internet Society (2003). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assignees.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION
HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF
MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgment

Funding for the RFC Editor function is currently provided by the
Internet Society.