Network Working Group
Internet-Draft
Expires: November 18, 2003

P. Saint-Andre
Jabber Software Foundation

May 20, 2003

End-to-End Object Encryption in XMPP
draft-ietf-xmpp-e2e-03

Status of this Memo

Copyright Notice

Abstract

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

Table of Contents

1. Introduction

This document define a method for end-to-end signing and encryption in the Extensible Messaging and Presence Protocol (XMPP). (For information about XMPP, see XMPP Core [1] and XMPP IM [2].) 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 document thereby helps the XMPP specifications meet the requirements defined in RFC 2779 [3].

1.1 Terminology

This document inherits terminology defined in XMPP Core [1] and RFC 2778 [4].

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 RFC 2119 [5].

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 document, 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 RFC 2779 [3]. 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 RFC 2799. In particular, the method MUST address the following requirements defined in RFC 2779:

    * 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 Common Presence and Instant

Messaging (CPIM) as 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 [6].

- Prior to encrypting or signing, the format of presence information must conform to the CPIM Presence Information Data Format defined in PIDF [7].

3. The method MUST follow the procedures (including the specific algorithms) defined in Common Profile for Instant Messaging [8] and Common Profile for Presence [9]. In particular, these documents specify:

   - Encryption MUST use S/MIME [10] encryption with CMS [11] EnvelopeData.

   - Signing MUST use S/MIME [10] signatures with CMS [11] SignedData.

   - The S/MIME algorithm SHOULD be AES [12].

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 [6].

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 S/MIME object (see RFC 1847 [13]) as the CDATA of an <e2e/> child of a <message/> stanza, with the <e2e/> element scoped by the 'urn:ietf:params:xml:ns:xmpp-e2e' namespace (note that this namespace name adheres to the format defined in The IANA XML Registry [14]).

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

```
Content-type: Message/CPIM

From: Juliet Capulet <im:juliet@capulet.com>
To: Romeo Montague <im:romeo@montague.net>
DateTime: 2003-05-14T11:45:36Z
Subject: Imploring

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

<body>
Wherefore art thou, Romeo?
</body>
```

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

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

--next
Content-type: Message/CPIM

From: Juliet Capulet <im:juliet@capulet.com>
To: Romeo Montague <im:romeo@montague.net>
DateTime: 2003-05-14T23:45:36Z
Subject: Imploring

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

<body>
Wherefore art thou, Romeo?
</body>
--next
Content-Type: application/pkcs7-signature

[signed body part]

--next--
  </e2e>
</message>
```

4. Securing Presence

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

1. Generate an "application/cpim-pidf+xml" object defined in PIDF [7].

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

3. Provide the resulting S/MIME object as the CDATA of an <e2e/> child of a <presence/> stanza, with the <e2e/> element scoped by the 'urn:ietf:params:xml:ns:xmpp-e2e' namespace (note that this namespace name adheres to the format defined in The IANA XML Registry [14]). The <presence/> stanza MUST include a 'to' attribute, i.e., it must be an instance of directed presence as defined in XMPP IM [2].

Example 3: Sender generates "application/cpim-pidf+xml" object:

```
Content-type: application/cpim-pidf+xml

From: Juliet Capulet <pres:juliet@capulet.com>
To: Romeo Montague <pres:romeo@montague.net>
DateTime: 2003-05-14T23:53:11Z

Content-type: text/xml; charset=utf-8
Content-ID: <2345678901@capulet.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@capulet.com">
  <tuple id="h40zny"
    <status>
      <basic>open</basic>
      <im:im>away</im:im>
    </status>
    <note xml:lang="en">retired to the chamber</note>
    <timestamp>2003-05-14T23:53:11Z</timestamp>
  </tuple>
</presence>
```

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

```
<presence to='romeo@montague.net/orchard'>
```

```
     <e2e xmlns='urn:ietf:params:xml:ns:xmpp-e2e'>
   Content-Type: multipart/signed; boundary=next;
                 micalg=sha1;
                 protocol=application/pkcs7-signature

   --next
   Content-type: application/cpim-pid+xml

   From: Juliet Capulet <pres:juliet@capulet.com>
   To: Romeo Montague <pres:romeo@montague.net>
   DateTime: 2003-05-14T23:53:11Z

   Content-type: text/xml; charset=utf-8
   Content-ID: <2345678901@capulet.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@capulet.com">
     <tuple id="h40zny"
       <status>
         <basic>open</basic>
         <im:im>away</im:im>
       </status>
       <note xml:lang="en">retired to the chamber</note>
       <timestamp>2003-05-14T23:53:11Z</timestamp>
     </tuple>
   </presence>
   --next
   Content-Type: application/pkcs7-signature

   [signed body part]

   --next--
     </e2e>
   </presence>
```

5. 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. CPIM [8] and CPP [9] define the common profiles to be used for interoperability between instant messaging and presence services that comply with RFC 2779 [3]. In the case of communications between an XMPP service and a non-XMPP service, we can visualize this relationship as follows:

```
+-------------+         +-------------+         +-------------+
|             |         |             |         |             |
|    XMPP     |         |  CPIM/CPP   |         |  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 CPIM/CPP gateways. In particular:

- When a gateway receives a secured XMPP message or presence stanza from the XMPP service that 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.

6. IANA Considerations

A URN sub-namespace for signed and encrypted content in the Extensible Messaging and Presence Protocol (XMPP) is defined as follows.

URI: urn:ietf:params:xml:ns:xmpp-e2e

Specification: [RFCXXXX]

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

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

7. Security Considerations

Detailed security considerations for instant messaging and presence protocols are given in RFC 2779 [3], specifically in Sections 5.1 through 5.4.

The end-to-end security method defined here MAY result in exchanging secured instant messages and presence information through a gateway that implements CPIM [8] and CPP [9]. Such a gateway MUST be compliant with the minimum security requirements of the instant messaging and presence protocols with which it interfaces. The introduction of gateways to the security model of instant messaging and presence in RFC 2779 also introduces some new risks. End-to-end security properties (especially confidentiality and integrity) between instant messaging and presence user agents that interface through a CPIM/CPP gateway can be provided only if common formats are supported. The need for end-to-end security is thus met by this specification through the use of common formats, specifically MSGFMT [6] for instant messages and PIDF [7] for presence information. Common formats are further ensured by requiring the use of multipart S/MIME [10] objects, as well as CMS [11] EnvelopeData for encryption and CMS [11] SignedData for signing. Finally, the algorithm used SHOULD be AES [12], since it is expected that AES best suits the capabilities of many platforms. However, an IETF specification for the use of AES is still incomplete at the time of writing.

Normative References

[1] Saint-Andre, P. and J. Miller, "XMPP Core", draft-ietf-xmpp-core-12 (work in progress), May 2003.

[2] Saint-Andre, P. and J. Miller, "XMPP Instant Messaging", draft-ietf-xmpp-im-11 (work in progress), May 2003.

[3] Day, M., Aggarwal, S. and J. Vincent, "Instant Messaging / Presence Protocol Requirements", RFC 2779, February 2000.

[4] Day, M., Rosenberg, J. and H. Sugano, "A Model for Presence and Instant Messaging", RFC 2778, February 2000, <http://www.ietf.org/rfc/rfc2778.txt>.

[5] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[6] Atkins, D. and G. Klyne, "Common Presence and Instant Messaging Message Format", draft-ietf-impp-cpim-msgfmt-08 (work in progress), January 2003.

[7] Fujimoto, S., Sugano, H., Klyne, G., Bateman, A., Carr, W. and J. Peterson, "CPIM Presence Information Data Format", draft-ietf-impp-cpim-pidf-08 (work in progress), May 2003.

[8] Crocker, D. and J. Peterson, "Common Profile for Instant Messaging (CPIM)", draft-ietf-impp-im-02 (work in progress), March 2003.

[9] Crocker, D. and J. Peterson, "Common Profile for Presence (CPP)", draft-ietf-impp-pres-02 (work in progress), March 2003.

[10] Ramsdell, B., "S/MIME Version 3 Message Specification", RFC 2633, June 1999.

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

[12] Housley, R. and J. Schaad, "Use of the AES Encryption Algorithm and RSA-OAEP Key Transport in CMS", draft-ietf-smime-aes-alg-06 (work in progress), January 2003.

[13] Galvin, J., Murphy, S., Crocker, S. and N. Freed, "Security Multiparts for MIME: Multipart/Signed and Multipart/Encrypted", RFC 1847, October 1995.

[14] Mealling, M., "The IANA XML Registry",

draft-mealling-iana-xmlns-registry-04 (work in progress), June 2002.

Author's Address

Peter Saint-Andre
Jabber Software Foundation

EMail: stpeter@jabber.org
URI: http://www.jabber.org/people/stpeter.php

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-02

- Completely revised to use CPIM/CPP.

B.2 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.3 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'.
- Added XML schema.

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

Acknowledgement

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