

Network Working Group
Internet-Draft
Expires: October 6, 2003

P. Saint-Andre
Jabber Software Foundation
J. Hildebrand
Jabber, Inc.
April 7, 2003

End-to-End Object Encryption in XMPP
draft-ietf-xmpp-e2e-01

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on October 6, 2003.

Copyright Notice

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

Abstract

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

Table of Contents

1.	Introduction	3
1.1	Terminology	3
1.2	Discussion Venue	3
1.3	Intellectual Property Notice	3
2.	Requirements	4
3.	Encrypting Stanzas	5
3.1	General Syntax	5
3.2	Encrypting Messages	6
3.3	Encrypting Presence	7
3.4	Encrypting IQs	8
4.	Signing Encrypted Content	9
5.	Signing Broadcasted Presence	10
6.	Signalling Support via Presence	11
7.	IANA Considerations	12
8.	Security Considerations	13
	References	14
	Authors' Addresses	14
A.	XML Schemas	15
A.1	urn:ietf:params:xml:ns:xmpp-e2e	15
A.2	urn:ietf:params:xml:ns:xmpp-e2e#payload	16
B.	Revision History	17
B.1	Changes from draft-ietf-xmpp-e2e-00	17
	Intellectual Property and Copyright Statements	18

1. Introduction

This document describes an end-to-end signing and encryption method for use in the Extensible Messaging and Presence Protocol (XMPP) as defined by XMPP Core [[1](#)] and XMPP IM [[2](#)]. Object signing and encryption enables a sender to encrypt an XML stanza sent to a specific recipient, sign such a stanza, sign broadcasted presence, and signal support for the method defined herein. This document thereby helps the XMPP specifications meet the requirements defined in [RFC 2779](#) [[3](#)].

1.1 Terminology

This document inherits the terminology defined in XMPP Core [[1](#)].

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](#) [[4](#)].

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. Encryption must work with any stanza type (message, presence, or IQ).
2. The full XML stanza must be encrypted.
3. Encryption must be possible using either OpenPGP [[5](#)] or S/MIME [[6](#)].
4. It must be possible to sign encrypted content.
5. It must be possible to sign broadcasted presence.
6. Any namespaces used must conform to The IETF XML Registry [[7](#)].

3. Encrypting Stanzas

3.1 General Syntax

Any stanza may be encrypted. The full stanza must be inserted as a direct child of a <payload/> element scoped by the 'urn:ietf:params:xml:ns:xmpp-e2e#payload' namespace. The stanza data MUST be preceded by another direct child of the <payload/> element, namely an <id/> element. The CDATA of the <id/> element MUST be constructed according to the following algorithm:

1. concatenate the sender's full JID (user@host/resource) with the recipient's full JID
2. concatenate the resulting string with a full ISO-8601 UTC timestamp including year, month, day, hours, minutes, seconds in the following format: yyyy-mm-dd-Thh:mm:ssZ (the timestamp must be UTC, no offsets are allowed)
3. hash the resulting string according to the SHA1 algorithm
4. convert the hexadecimal SHA1 output to all lowercase.

Before encryption, the XML to be encrypted will thus be of the following form:

```
<payload xmlns='urn:ietf:params:xml:ns:xmpp-e2e#payload'>
  <id>someID</id>
  [stanza]
</payload>
```

The full <payload/> element (including all XML tag names and angle brackets) MUST then be encrypted using either OpenPGP or S/MIME. The output MUST be armored US-ASCII with any headers removed. The resulting cipher text MUST then be provided as the CDATA of a <stanza/> child of an <x/> element scoped by the 'urn:ietf:params:xml:ns:xmpp-e2e' namespace, with the value of the 'type' attribute set to either "openpgp" or "smime" depending on which method was used.

The format of the stanza that results from the foregoing procedure is as follows (no 'from' address is included in the stanza to be encrypted, since that is stamped by the sender's server):


```
<[stanza-name]
  to='recipient'
  type='[value if provided]'
  id='[value if provided]'
  xml:lang='[value if provided]'\>
<x type='[openpgp|smime]'
  xmlns='urn:ietf:params:xml:ns:xmpp-e2e'\>
  <stanza>
    [encrypted content]
  </stanza>
</x>
</[stanza-name]>
```

3.2 Encrypting Messages

Message stanzas may be encrypted using the syntax defined above.

Example: Sender generates encrypted message ('from' address is stamped by sender's server):

```
<message to='romeo@montague.net/orchard' type='chat'\>
  <x type='openpgp' xmlns='urn:ietf:params:xml:ns:xmpp-e2e'\>
    <stanza>
hQE0A+fczQLixGb6EAP/SmSRmrzpZQ90Prjbs2HoZ4VkfNEodykB/TiDt86NdtPE
zmeLBduaJZEQqhs1UbBu8355fvy/ykDom1Xe/S1q56ZMEsSXkD04x1xt/30E/Hru
ovLXkTAVNX9pftQb4rC2CC9G+X/ZsRiUf53ug/9PGBDMBYiqWRWUBWipWqxoBbID
/2j83fQTGopp//tKijmhyMK7/xC73p/9TezvIz1ESqJY2NwSoRo0us6mKu4bBQ3G
EtOmMJZZUToNZwgDfL0DzZHGOyiT4tdUL9eCln2a5FAgN75NnCUdHdRw0zpaCVIK
E1389vM18L0ir1mxBMhVYLDyxAwsB8evXkAJeYu0mLuJ0sBZAbyfSlnGr8sAZ7c4
peSUpsBMhA4lA0nUASra2tYNsv0dfiFU2V7k1QEoR4c0HBB+ORX5HElPFdgzYM6Q
yhXSNWxTqBD1CfYSHM2KNzSJnEimSeL6/bh032tAXIK+rigywLyCDAFEpY0jLXhp
9TA5pQw5ADMzmJnYlq3H5q4kn7s7RfzUuWf1QjzhU4u2YFj3lJIRp01szyXAACTG
hJbXpwL0I2Gz4YezWnzIKWU5xTna+V+0heP+lfUfmkP9CtTZZEmxEPKkWtNct7Fk
wUlr9Deqq05dGd+1KT94QY7c1Anb7IRIGP/ZegQpn6A4XRvIDwe3/kMadWLVSR7
aYHSCl6JG9ozHGlwIR3HF8K09je/oQwhXvnzimQ=
=zjBS
    </stanza>
  </x>
</message>
```

The decoded payload is:


```
<payload xmlns='urn:ietf:params:xml:ns:xmpp-e2e#payload'>
  <id>e0ffe42b28561960c6b12b944a092794b9683a38</id>
  <message
    to='romeo@montague.net/orchard'
    type='chat'>
    <subject>Imploring</subject>
    <body>O Romeo, Romeo! Wherefore art thou Romeo?</body>
  </message>
</payload>
```

3.3 Encrypting Presence

Presence stanzas may be encrypted using the syntax defined above. An encrypted presence stanza MUST be an instance of directed presence; i.e., the <presence/> element MUST possess a 'to' attribute specifying a specific intended recipient. Presence information that is intended for broadcasting to all subscribers MUST NOT be encrypted.

Example: Sender generates encrypted presence ('from' address is stamped by sender's server):

```
<presence to='romeo@montague.net/orchard'>
  <x type='openpgp' xmlns='urn:ietf:params:xml:ns:xmpp-e2e'>
    <stanza>
hQE0A+fczQLixGb6EAP/Y0qZS+jgzDrXdqIyuDDJI2oxH2LXZf10LeR6EeBdGqX8
ewI8Nsf3CR4Mou58tRZ1QG5E0sCl6aylUxAiJuSe5f+Lv97dRWGQnrAQ4RNVpJ80
jzPf+UQJ6mBZhGBgrtPB8XML7d0RJqWBR69ra1LcGh0tBr0CsNo7RyoZUWrfl74D
/0yJ7y3ZyHmA1gDRd9f7CzuMwdNF+xCfQtZjtAdc+t7HNsoJSNxGBeQdJbdpIaJo
jvHfiVG6jvrGDzWceyj4SnFkxOfxb+Xu1x7mcmiXW0Jb58wsddttmhqBDdDd4B3H
QKnZCkyMPUclDzCBXUf4JPbC5EcUnN0mT6mth9+Qj0GJ0sAPAW2tZu5L0LVQU5Wo
zMJBZJ0laiyEv74YSYCjGNwKP9Yh+f+rBL1UkmnKqfiZVxSQo50ccPkJ45Syq85j
v8RSvYsU27bTQdCNL/ZS5aILQHryD2iXoLDk9XkzVDTBDNah0k1IWUaJwU5Qy1Lw
oLEYwndAQi0ieXQklw+2HRmq5fZNslItCPJBGWmxAdG06xyKbkbqCfq6ytw9kXjW
wAoBMgWZFFIbBh5EdBd7N08u9bF3oDXxK07c4dkg6WXUjJTZzEIWZCNaFa1PcW+3
/FoQ
=HT9r
    </stanza>
  </x>
</presence>
```

The decoded payload is:


```
<payload xmlns='urn:ietf:params:xml:ns:xmpp-e2e#payload'>
  <id>e0ffe42b28561960c6b12b944a092794b9683a38</id>
  <presence to='romeo@montague.net/orchard'>
    <show>away</show>
    <status>retired to the chamber</status>
  </presence>
</payload>
```

3.4 Encrypting IQs

IQ stanzas may be encrypted using the syntax defined above.

Example: Sender generates encrypted IQ ('from' address is stamped by sender's server):

```
<iq to='romeo@montague.net/orchard' type='get' id='eq7-2521'>
  <x type='openpgp' xmlns='urn:ietf:params:xml:ns:xmpp-e2e'>
    <stanza>
hQE0A+fczQLixGb6EAP/ejh9XMAbiFTA4WRI0yBXiiiAHtKCe/AKcn5I1M+HI/AR
8K3LdWbg4CzuBfDv/Sb9zesVXIZZEvhHqF6ihjxQpW0V0a1lvGDq49Dc0bR4uPsz
sFRr9auTnouZ5062ubwGk3Uic8CChC/JZxlfDRX04ac3jS+uzafC0aJ9hkn0QkoD
/0b9PpTC30Yq5JoMpFSvBHeH0yixqKQh6xhBgJLzr2/6ZId/ax0pgq7ru1GyYmHg
+dg/wuizJLgMaSSLwmEM58JiGKs44RHcQMULnEruQvSbbCCNKIaLCMVQPLXS+oaD
Ly2ZG8BW+lb0j0d2E0dXbM30TvTfCW9w76xv0nX/BLRT0sA6AVmuJLz6+UN55roD
dE7HncBV0J/NmWksTHL/e452109aWSrqTYFsG6Fvvu7In5o0iKHIKLvZosW49zA9
McDna2krEtjWCsx5fEHbxGrBW0puPPHqD+uuSvD7f7RLW0KvW+Jz2/OXB0JUJ2+x
+xX9uaTdP08TlfBa4BrSx5mM+eFhkPC5oDg308Jy612A2Jf8IRQ4lYZDoz6SWoHl
scfHcSWjqont7hUTXtdTEhHcs9UkaxXlrbwLBaEfix0J7ALgjAESfEjG88eHm5oj
49I9rju8kw+HEsSl/moI+icDmuc0mN7bj0cKM3rIeU/roqWD0llWFIyWwMNLg==
=6H0T
    </stanza>
  </x>
</iq>
```

The decoded payload is:

```
<payload xmlns='urn:ietf:params:xml:ns:xmpp-e2e#payload'>
  <id>e0ffe42b28561960c6b12b944a092794b9683a38</id>
  <iq to='romeo@montague.net/orchard' type='get' id='eq7-2521'>
    <query xmlns='jabber:iq:version' />
  </iq>
</payload>
```


4. Signing Encrypted Content

OpenPGP and S/MIME both allow an entity to either encrypt then sign, or sign then encrypt. When signing first, the signatories are obscured by the encryption; when encrypting first, the signatories are exposed but the signatures can be verified without decrypting. Because in XMPP the signatories are exposed by the very act of exchanging a stanza (since the 'from' and 'to' addresses must be exposed for routing purposes), there would be no use in signing first and encrypting second. Therefore, if signing is desired, it SHOULD be performed after encrypting.

5. Signing Broadcasted Presence

An entity may want to sign presence information for broadcasting to all subscribers (i.e., the presence stanza is not directed to a particular recipient, but is sent to all other entities that have subscribed to the sender's presence). Because encrypted presence **MUST** be directed to a particular recipient, signed presence for broadcasting **MUST NOT** be encrypted, only signed. However, there is little to no value in signing the entire stanza; therefore it is enough to sign only the user-provided CDATA of the <status/> element (note that this requires a signed presence broadcast to include some CDATA in the <status/> element). The process is as follows:

1. User provides CDATA for the <status/> element.
2. Client application signs CDATA using OpenPGP or S/MIME.
3. Client application inserts signed ASCII output as CDATA of an <x/> element scoped by the 'urn:ietf:params:xml:ns:xmpp-e2e' namespace (including a 'type' attribute whose value is either "openpgp" or "smime").
4. Client application adds <x/> element to remainder of presence stanza and sends to server with no 'to' attribute.

Example: Sender generates signed presence:

```
<presence>
  <show>away</show>
  <status>retired to the chamber</status>
  <x type='openpgp' xmlns='urn:ietf:params:xml:ns:xmpp-e2e'>
    <signed>
iD8DBQE+kgpNEWF4x4jKHUYRAuthAJ9L1BjML9GIpagVGBEEJr0C7F3k9ACeJRL4
obxiSG72h3ggH0Xr3BmGyjE=
=T4rw
    </signed>
  </x>
</presence>
```


6. Signalling Support via Presence

In order to signal support for this protocol, an entity MAY broadcast its credentials (KeyID and/or certificate subject) in all outgoing presence stanzas. If included, this information MUST be contained in a <support/> child of an <x/> element scoped by the 'urn:ietf:params:xml:ns:xmpp-e2e' namespace. An entity MAY include more than one instance (e.g., one for OpenPGP and one for S/MIME). KeyIDs SHOULD be 64-bit rather than 32-bit.

```
<presence>
  <x type='openpgp' xmlns='urn:ietf:params:xml:ns:xmpp-e2e'>
    <support>[KeyID]</support>
  </x>
  <x type='smime' xmlns='urn:ietf:params:xml:ns:xmpp-e2e'>
    <support>[certificate subject]</support>
  </x>
</presence>
```


7. IANA Considerations

A URN sub-namespace for XMPP encrypted content is defined as follows.

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

Specification: [RFCXXXX]

Description: This is the XML namespace name for XMPP encrypted content as defined by [RFCXXXX].

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

8. Security Considerations

Replay attacks are made more difficult using this method because of the inclusion of a unique ID in the encrypted object. Key exchange may rely on the web of trust model used on the OpenPGP keys network. There is no method to check a fingerprint or ownership of a key other than checking the user IDs on a key. A key or certificate SHOULD have associated with it the Jabber ID of the sender.

References

- [1] Saint-Andre, P. and J. Miller, "XMPP Core ([draft-ietf-xmpp-core-08](#), work in progress)", April 2003.
- [2] Saint-Andre, P. and J. Miller, "XMPP Instant Messaging ([draft-ietf-xmpp-im-08](#), work in progress)", April 2003.
- [3] Day, M., Aggarwal, S., Mohr, G. and J. Vincent, "A Model for Presence and Instant Messaging", [RFC 2779](#), February 2000, <<http://www.ietf.org/rfc/rfc2779.txt>>.
- [4] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [5] Callas, J., Donnerhacke, L., Finney, H. and R. Thayer, "OpenPGP Message Format", [RFC 2440](#), November 1998.
- [6] Ramsdell, B., "S/MIME Version 3 Message Specification", [RFC 2633](#), June 1999.
- [7] Mealling, M., "The IANA XML Registry", [draft-mealling-iana-xmlns-registry-04](#) (work in progress), June 2002.

Authors' Addresses

Peter Saint-Andre
Jabber Software Foundation

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

Joe Hildebrand
Jabber, Inc.

EMail: jhildebrand@jabber.com
URI: <http://www.jabber.org/people/hildjj.php>

[Appendix A](#). XML Schemas

The following XML schemas are descriptive, not normative.

[A.1](#) urn:ietf:params:xml:ns:xmpp-e2e

```
<?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='x'>
    <xs:complexType>
      <xs:choice>
        <xs:element ref='signed' minOccurs='0' maxOccurs='1'/>
        <xs:element ref='stanza' minOccurs='0' maxOccurs='1'/>
        <xs:element ref='support' minOccurs='0' maxOccurs='1'/>
      </xs:choice>
      <xs:attribute name='type' use='required'/>
      <xs:simpleType>
        <xs:restriction base='xs:NCName'>
          <xs:enumeration value='openpgp'/>
          <xs:enumeration value='smime'/>
        </xs:restriction>
      </xs:simpleType>
    </xs:complexType>
  </xs:element>

  <xs:element name='signed' type='xs:string'/>
  <xs:element name='stanza' type='xs:string'/>
  <xs:element name='support' type='xs:string'/>

</xs:schema>
```


[A.2](#) urn:ietf:params:xml:ns:xmpp-e2e#payload

```
<?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#payload'
  xmlns='urn:ietf:params:xml:ns:xmpp-e2e#payload'
  elementFormDefault='qualified'>

  <xs:element name='payload'>
    <xs:complexType>
      <xs:element ref='id' maxOccurs='1'/>
      <xs:any namespace='jabber:client' maxOccurs='1'/>
    </xs:complexType>
  </xs:element>

  <xs:element name='id' 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-00](#)

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

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.

Acknowledgement

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