



# XMPP

## XEP-0027: Current Jabber OpenPGP Usage

Thomas Muldowney  
<mailto:temas@jabber.org>  
<xmpp:temas@jabber.org>

2014-03-14  
Version 1.4

Status	Type	Short Name
Obsolete	Historical	openpgp

This document outlines the current usage of OpenPGP for messaging and presence.

# Legal

## Copyright

This XMPP Extension Protocol is copyright © 1999 - 2014 by the [XMPP Standards Foundation](#) (XSF).

## Permissions

Permission is hereby granted, free of charge, to any person obtaining a copy of this specification (the "Specification"), to make use of the Specification without restriction, including without limitation the rights to implement the Specification in a software program, deploy the Specification in a network service, and copy, modify, merge, publish, translate, distribute, sublicense, or sell copies of the Specification, and to permit persons to whom the Specification is furnished to do so, subject to the condition that the foregoing copyright notice and this permission notice shall be included in all copies or substantial portions of the Specification. Unless separate permission is granted, modified works that are redistributed shall not contain misleading information regarding the authors, title, number, or publisher of the Specification, and shall not claim endorsement of the modified works by the authors, any organization or project to which the authors belong, or the XMPP Standards Foundation.

## Warranty

## NOTE WELL: This Specification is provided on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. ##

## Liability

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall the XMPP Standards Foundation or any author of this Specification be liable for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising from, out of, or in connection with the Specification or the implementation, deployment, or other use of the Specification (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if the XMPP Standards Foundation or such author has been advised of the possibility of such damages.

## Conformance

This XMPP Extension Protocol has been contributed in full conformance with the XSF's Intellectual Property Rights Policy (a copy of which can be found at <http://xmpp.org/about-xmpp/xsf/xsf-ipr-policy/> or obtained by writing to XMPP Standards Foundation, 1899 Wynkoop Street, Suite 600, Denver, CO 80202 USA).

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Signing</b>	<b>1</b>
<b>3</b>	<b>Encrypting</b>	<b>1</b>
<b>4</b>	<b>Security Considerations</b>	<b>2</b>
<b>5</b>	<b>Other Known Issues</b>	<b>3</b>
<b>6</b>	<b>IANA Considerations</b>	<b>3</b>
<b>7</b>	<b>XMPP Registrar Considerations</b>	<b>3</b>
<b>8</b>	<b>XML Schemas</b>	<b>4</b>
8.1	jabber:x:encrypted . . . . .	4
8.2	jabber:x:signed . . . . .	4

## 1 Introduction

The Jabber community has long acknowledged the need for privacy and security features in a well-rounded instant messaging system. Unfortunately, finding a consensus solution to the problem of end-to-end encryption during the community's younger days was not easy. Eventually, early contributors created a quick solution using OpenPGP ([RFC 4880](https://tools.ietf.org/html/rfc4880)<sup>1</sup>). This specification documents the OpenPGP solution as it is used today, so that others may interoperate with clients that support it. This document is not intended to present a standard, because more complete solutions are being investigated.

All operations described here are done with standard OpenPGP software such as [GnuPG](https://gnupg.org/). All program output is US-ASCII armored output with the headers removed. This allows for easy transportation of the program output directly in the XML. All keys are exchanged using OpenPGP key servers, and usually are retrieved when a signed `<presence/>` stanza is received (key retrieval does not happen in-band).

## 2 Signing

Signing enables a sender to verify that they sent a certain block of text. In Jabber, signing uses the 'jabber:x:signed' namespace, and is primarily used with `<presence/>`, but may also be used with `<message/>`. The text that is signed MAY be the empty string. When signing presence, the sender SHOULD sign the XML character data of the `<status>` element. The sender SHOULD sign presence using the private key whose KeyID corresponds to the public key to be used in encrypting messages (see below).

Listing 1: A signed presence stanza

```
<presence from='pgmillard@jabber.org/wj_dev2' to='jer@jabber.org'>
  <status>Online</status>
  <x xmlns='jabber:x:signed'>
    iQA/AwUB0jU5dnol3d88qZ77EQI2JACfRngLJ045brNnaCX78ykKNUZaTioAoPHI
    2uJxPMGR73EBIvEpcv0LRsy+
    =45f8
  </x>
</presence>
```

## 3 Encrypting

Encryption enables the sender to encrypt a message to a specific recipient. This is accomplished using the 'jabber:x:encrypted' namespace in conjunction with `<message/>` stanzas. Because a block of text is necessary in order to have something to encrypt, `<message/>` stanzas intended to be encrypted have the same restrictions as signing (see above). The data

---

<sup>1</sup>RFC 4880: OpenPGP Message Format [http://tools.ietf.org/html/rfc4880](https://tools.ietf.org/html/rfc4880).

encrypted MUST be the XML character data of the <body> element. The sender SHOULD encrypt the message body using the public key whose KeyID corresponds to the private key used in signing presence (see above).

Listing 2: An encrypted message stanza

```
<message to='reatmon@jabber.org/jarl' from='pgmillard@jabber.org/
  wj_dev2'>
  <body>This message is encrypted.</body>
  <x xmlns='jabber:x:encrypted'>
    qANQR1DBwU4DX7jmYZnncmUQB/9KuKBddzQH+tZ1ZywKK0yHKnq57kWq+RFtQdCJ
    WpdWpR0uQsuJe7+vh3NWN59/gTc5MD1X8dS9p0ovStmNcyLhxVgmqS8ZKhsblVeu
    IpQ0JgavABqibJolc3BKrVtVV1igKiX/N7Pi8RtY1K18toaMDhdEfHBRz0/XB0+P
    AQhYlRjNacGcs1khXqNjK5Va4tu0APy2n1Q8UUrHbUd0g+xJ9Bm0G0LZXyvcWyKH
    kuNEHFQiLuCY6Iv0myq6iX6tjuHehZlFSh80b5BVV9tNLwNR5Eqz1klxMhoghJOA
    w7R61cCPt8KSd8Vcl8K+Stq0MZ5wkhosVjUqvEu8uJ9RupdpB/4m9E3g0QZCBsmq
    OsX4/jJhn2wIsfYYWdqkbNKnuYoKcNwrlmn6I+wX72p0R8tTv8peNCwK9bEtL/XS
    mhn4bCxoUkCITv3k8a+Jdvbov9ucduKSFCBq4/l0fpHmPhHQjkFofxmaWJveFF
    619NXyYyCfoLTmWk2AaTHVCjtKdf1WmwcTa0vFfk8BuFHkdah6kJJiJ7w/yNwa/E
    06CMymuZTr/LpcKKWrWCt+SErxmq8ekPI8h7oNwMxZBYAa70J1rXWKNgL9pDtNI
    824Mf0mXj7q5N1eMHvX1QEoKLAda/Ae3TTEevOyeUK1DEgvxfM2KRZ11RzU+XtIE
    My/bJk7EycAw8P/QKyeNl01fxP58VED6Gb8NCPqK0Yn/LKh10+c20ZNVPEPFM4bNV
    XA4hB4UttFF7Ao8kpd1rUqdKyw41EtnmdemYQ6+iIIVPEarWl9PxOMY90KAnZrSAq
    bt9uRY/1rPgelRaWblMKvxgprO8++Y8VjdEyGgMOXx0iE851Ve72ftGzkSxDH8mW
    TgY3pf2aATmBp3lagQ1C0kGS/xupovT5AQA3RzbCxDvc6s6eGYKmVVQVj5vmSj1
    WULad5MB9KT1DzCm6FOSy063nWGBYYMwiejRvGLpo1j4eAnj0q0t7rTWmgv3RkYF
    0in0vD0hW7aC
    =CvnG</x>
  </message>
```

It is considered polite to include an unencrypted message <body/> explaining that the actual message body is encrypted. This helps if the client experiences an error while decrypting the message, or if the user's a client that does not support encryption (although generally this should not happen, since the signed presence can be used to indicate that a client accepts encrypted messages).

## 4 Security Considerations

The method defined herein has the following security issues:

- Key exchange relies on the web of trust model used on the OpenPGP keys network.
- There is no mechanism for checking a fingerprint or ownership of a key other than checking the user IDs on a key.

- When the recipient is not mentioned in the encrypted body, replay attacks are possible on messages.
- Replay of the signed <presence/> status is possible.
- It relies on signing or encryption of XML character data; therefore, it does not support signing or encryption of <iq/> stanzas, and it allows signing of the presence <status/> element and encryption of the message <body/> element only. Thus the method is not acceptable when signing or encryption of full stanzas is required.
- It does not enable both signing and encryption of a stanza, only signing of the presence status and encryption of the message body.

## 5 Other Known Issues

In addition to the security considerations listed above, there are several other known issues with this method:

- It is limited to PGP keys and does not support X.509 certificates, Kerberos, RSA keys, etc.
- It does not include feature negotiation; instead, signed <presence/> is used as an indicator of support. Because of the lack of negotiation it is possible for encrypted <message/> elements to be stored offline and then read by a client that cannot support them.
- It is verbose (the example encrypted <message/> is "Hi").

## 6 IANA Considerations

This document requires no interaction with the [Internet Assigned Numbers Authority \(IANA\)](#)<sup>2</sup>.

## 7 XMPP Registrar Considerations

The [XMPP Registrar](#)<sup>3</sup> shall register the 'jabber:x:encrypted' and 'jabber:x:signed' namespaces as a result of this document.

---

<sup>2</sup>The Internet Assigned Numbers Authority (IANA) is the central coordinator for the assignment of unique parameter values for Internet protocols, such as port numbers and URI schemes. For further information, see <http://www.iana.org/>.

<sup>3</sup>The XMPP Registrar maintains a list of reserved protocol namespaces as well as registries of parameters used in the context of XMPP extension protocols approved by the XMPP Standards Foundation. For further information, see <http://xmpp.org/registrar/>.

## 8 XML Schemas

### 8.1 jabber:x:encrypted

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

<xs:schema
  xmlns:xs='http://www.w3.org/2001/XMLSchema'
  targetNamespace='jabber:x:encrypted'
  xmlns='jabber:x:encrypted'
  elementFormDefault='qualified'>

  <xs:annotation>
    <xs:documentation>
      The protocol documented by this schema is defined in
      XEP-0027: http://www.xmpp.org/extensions/xep-0027.html
    </xs:documentation>
  </xs:annotation>

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

</xs:schema>
```

### 8.2 jabber:x:signed

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

<xs:schema
  xmlns:xs='http://www.w3.org/2001/XMLSchema'
  targetNamespace='jabber:x:signed'
  xmlns='jabber:x:signed'
  elementFormDefault='qualified'>

  <xs:annotation>
    <xs:documentation>
      The protocol documented by this schema is defined in
      XEP-0027: http://www.xmpp.org/extensions/xep-0027.html
    </xs:documentation>
  </xs:annotation>

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

</xs:schema>
```