



XMPP

XEP-0380: Explicit Message Encryption

Emmanuel Gil Peyrot

<mailto:linkmauve@linkmauve.fr>

<xmpp:linkmauve@linkmauve.fr>

2016-10-26

Version 0.1

Status	Type	Short Name
Experimental	Standards Track	EME

This specification provides a way to mark encrypted messages so the recipient can discover how to decrypt it.

Legal

Copyright

This XMPP Extension Protocol is copyright © 1999 – 2017 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 <https://xmpp.org/about/xsf/ipr-policy>) or obtained by writing to XMPP Standards Foundation, P.O. Box 787, Parker, CO 80134 USA).

Contents

1	Introduction	1
2	Requirements	1
3	Use Cases	1
3.1	Basic Flow	1
3.2	Protocols Supported	2
3.3	Determining Support	3
4	Business Rules	4
5	Internationalization Considerations	4
6	Security Considerations	4
7	IANA Considerations	5
8	XMPP Registrar Considerations	5
8.1	Protocol Namespaces	5
9	XML Schema	5
10	Acknowledgements	5

1 Introduction

In the past few years we have seen a strong interest in end to end encryption, with multiple competing mechanisms being defined on top of XMPP (e.g., [Current Jabber OpenPGP Usage \(XEP-0027\)](#)¹, [Current Off-the-Record Messaging Usage \(XEP-0364\)](#)² or [OpenPGP for XMPP \(XEP-0373\)](#)³). This specification addresses the lack of proper discoverability of most of these solutions by adding a machine-readable explanation of how a specific message has been encrypted.

In a federated network where no central entity can mandate a particular encryption mechanism, it becomes important to allow end users to know that a message could not be decrypted (e.g., due to a missing plugin), and to never fail to display that a message has been received due to that.

2 Requirements

This document addresses the following requirements:

1. Enable a client to mark a message as encrypted.
2. Enable a client to determine whether a message was encrypted, no matter the encryption mechanism used.
3. Enable a client to offer the user a possibility to decrypt a received message (depending on the encryption method).
4. Enable a client to offer the user a possibility to decrypt subsequently received messages.

This document DOES NOT address the non-message usecases, encrypted presence and iq have very different requirements than those defined here.

3 Use Cases

3.1 Basic Flow

Romeo, wanting to get Juliet's attention but not wanting to reveal his intentions to the montague.lit nor to the capulet.lit servers, sends an encrypted message tagged as OTR, as follows:

¹XEP-0027: Current Jabber OpenPGP Usage <<https://xmpp.org/extensions/xep-0027.html>>.

²XEP-0364: Current Off-the-Record Messaging Usage <<https://xmpp.org/extensions/xep-0364.html>>.

³XEP-0373: OpenPGP for XMPP <<https://xmpp.org/extensions/xep-0373.html>>.

Listing 1: Example of tagged message encrypted with OTR

```

<message to='juliet@capulet.lit/balcony'
        from='romeo@montague.lit/orchard'
        id='secret1'>
  <body>?OTR?v23?...</body>
  <encryption xmlns='urn:xmpp:eme:0'
             namespace='urn:xmpp:otr:0' />
</message>

```

Juliet's client, noticing it does not have any OTR capability, will display that the message was encrypted but that it is not able to decrypt it instead of displaying the body, for example: This message was encrypted with OTR (urn:xmpp:otr:0) and could not be decrypted.

Juliet may then communicate to Romeo that she was unable to receive his message, through an error, or maybe out of band.

Romeo, standing firm in his belief that they should not communicate without encryption in their world where anyone could be a malicious listener, then discovers that one of Juliet's clients support [OpenPGP for XMPP \(XEP-0373\)](#)⁴ and subsequently starts an encrypted session using that protocol.

Listing 2: Example of tagged message encrypted with OX

```

<message to='juliet@capulet.lit/balcony'
        from='romeo@montague.lit/orchard'
        id='secret2'>
  <openpgp xmlns='urn:xmpp:openpgp:0'>
    ...
  </openpgp>
  <body>This message is encrypted with OpenPGP for XMPP.</body>
  <encryption xmlns='urn:xmpp:eme:0'
             namespace='urn:xmpp:openpgp:0' />
</message>

```

Upon receiving this message, Juliet's current client prompts her to enable a plugin, or even do it on its own, possible representations include:

This message was encrypted with OpenPGP for XMPP (urn:xmpp:openpgp:0), [click here to enable this plugin](#).

3.2 Protocols Supported

Any encryption mechanism using message as a transport is a candidate, and MAY have a 'name' attribute to help the receiving client display it to the user, in case this client doesn't understand its namespace yet. A 'name' attribute SHOULD NOT be included for the protocols listed herein, and SHOULD be ignored by a receiving client:

⁴XEP-0373: OpenPGP for XMPP <<https://xmpp.org/extensions/xep-0373.html>>.

Name	Namespace	Specification
OTR	urn:xmpp:otr:0	Current Off-the-Record Messaging Usage (XEP-0364) XEP-0364: Current Off-the-Record Messaging Usage < https://xmpp.org/extensions/xep-0364.html >.
Legacy OpenPGP	jabber:x:encrypted	Current Jabber OpenPGP Usage (XEP-0027) XEP-0027: Current Jabber OpenPGP Usage < https://xmpp.org/extensions/xep-0027.html >.
OpenPGP for XMPP	urn:xmpp:openpgp:0	OpenPGP for XMPP (XEP-0373) XEP-0373: OpenPGP for XMPP < https://xmpp.org/extensions/xep-0373.html >.

3.3 Determining Support

If an entity supports the Encrypted Message Extension protocol, it MUST report that by including a [Service Discovery \(XEP-0030\)](#)⁵ feature of "urn:xmpp:eme:0" in response to disco#info requests:

Listing 3: Client queries for entity features

```
<iq type='get'
  id='disco1'
  to='juliet@capulet.lit/balcony'
  from='romeo@montague.lit/orchard'>
  <query xmlns='http://jabber.org/protocol/disco#info' />
</iq>
```

Listing 4: Entity responds with features

```
<iq type='result'
  id='disco1'
  to='romeo@montague.lit/orchard'
  from='juliet@capulet.lit/balcony'>
  <query xmlns='http://jabber.org/protocol/disco#info'>
  ...
  <feature var='urn:xmpp:eme:0' />
  ...
  </query>
</iq>
```

⁵XEP-0030: Service Discovery <<https://xmpp.org/extensions/xep-0030.html>>.

Support can also be determined via [Entity Capabilities \(XEP-0115\)](#)⁶, a.k.a. "caps".

4 Business Rules

Entities **MUST** report a failure to the user if they cannot decrypt an incoming message for any reason, and **MAY** prompt the user to install or enable a plugin to decrypt it.

Entities **SHOULD** include a non-encrypted body as possible, since older clients not supporting this protocol might otherwise ignore messages sent with an unknown encryption, making both the sender frustrated that their message did not get an answer, and the recipient frustrated that they never saw any message.

A sender entity **MAY** include the `<encryption/>` element even if the recipient doesn't advertise support for it in their disco, or isn't currently connected, since the recipient may be using multiple clients with different capabilities.

A sender entity **MAY** include a 'name' attribute for any encryption mechanism not listed in this specification, to help the receiving entity present it to the user, but **SHOULD NOT** include one for the ones listed here.

A receiving entity **MUST NOT** use the 'name' attribute if it is present and they already have a name associated with it.

A receiving entity **MAY** not display anything in case an encrypted message has been received, if the user agreed to that behaviour.

5 Internationalization Considerations

When a message is marked with an encryption tag and can not be decrypted, the body can safely be ignored and a localized message displayed instead.

If an entity includes a 'name' attribute, it should attempt to localise it to the best of its abilities for the receiving client.

6 Security Considerations

A malicious entity could try to mimick the style of a client's failure message, maybe including a link to a compromised plugin, so a client should not make those missing plugin messages look like normal messages.

⁶XEP-0115: Entity Capabilities <https://xmpp.org/extensions/xep-0115.html>.

7 IANA Considerations

This document requires no interaction with the Internet Assigned Numbers Authority (IANA).

8 XMPP Registrar Considerations

8.1 Protocol Namespaces

This specification defines the following XML namespace:

- 'urn:xmpp:eme:0'

9 XML Schema

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

<xs:schema attributeFormDefault="unqualified"
  elementFormDefault="qualified"
  targetNamespace="urn:xmpp:eme:0"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">

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

  <xs:element name="encryption">
    <xs:complexType>
      <xs:attribute type="xs:string" use="required" name="namespace"/>
      <xs:attribute type="xs:string" use="optional" name="name"/>
    </xs:complexType>
  </xs:element>

</xs:schema>
```

10 Acknowledgements

Thanks to Mathieu Pasquet for his feedback.