XEP-0434: Trust Messages

Melvin Keskin
mailto:melvo@olomono.de
xmpp:melvo@olomono.de

2020-02-27
Version 0.1.0

Status	Type	Short Name
Experimental	Standards Track	NOT_YET_ASSIGNED

This document specifies a way to communicate the trust in public long-term keys used by end-to-end encryption protocols from one endpoint to another.

Legal

Copyright

Permissions

Warranty

Liability

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

End-to-end encryption without verifying the authenticity of the exchanged public long-term keys only enables the endpoints to protect their communication against passive attacks. This means an attacker cannot read encrypted messages in transit without actively intervening in the key exchange. However, without any other precautions active attacks are still possible. If an attacker replaces the exchanged keys with malicious ones or introduces a new malicious endpoint with an own key, the end-to-end encrypted messages can be read and manipulated by the attacker.
When using end-to-end encryption where public long-term keys are transmitted over a channel which is not protected against active attacks, the authenticity of those keys is not guaranteed. Such a key has to be authenticated by the receiving endpoint over a channel which is protected against active attacks to maintain the confidentiality of sent messages and ensure the authenticity and integrity of received messages.
A trust message is an XMPP message that contains the information of whether the sending endpoint trusts a specific public long-term key. The authenticity and integrity of the message is ensured by a signing mechanism. Trust messages can be used in conjunction with an end-to-end encryption protocol like OpenPGP for XMPP (XEP-0373) [1] or OMEMO Encryption (XEP-0384) [2] e.g. to automatically or semi-automatically establish secure channels protected against active attacks.
Furthermore, the fact that an endpoint trusts a key or not can be kept confidential toward an attacker by encrypting those messages and sending them only to endpoints with authenticated keys. That means particularly that an attacker cannot detect by the content of a trust message whether an authentication of a key took place. An authentication will therefore stay anonymous toward an attacker. The encryption protects against passive attacks since an attacker cannot read the content of the trust message. The restriction to send trust messages only to endpoints with authenticated keys in addition to the encryption protects against active attacks since the attacker will not, after introducing a malicious key, receive a trust message encrypted with that key.

2 Glossary

Endpoint Communication endpoint owning exactly one public long-term key. In most cases that is an XMPP client instance. In the terminology of OMEMO Encryption (XEP-0384) XEP-0384: OMEMO Encryption <https://xmpp.org/extensions/xep-0384.html>., that is a ”device”. To cover also the possibility for using multiple endpoints on the same physical device and via the same client instance, the general term ”endpoint” is used.

Key authentication Verification that a key received over an insecure channel is actually the one of the assumed endpoint

[1] XEP-0373: OpenPGP for XMPP <https://xmpp.org/extensions/xep-0373.html>.
[2] XEP-0384: OMEMO Encryption <https://xmpp.org/extensions/xep-0384.html>.

Key identifier Identifier of a key (e.g., a fingerprint or the key itself)

Trust message XMPP message which indicates that specific keys are trusted or not trusted by the sender. A trust message for an endpoint's key contains the key identifier of the given key.

3 Trust Message Structure

A trust message MUST be signed in a way to ensure its authenticity and integrity.
The part specific for a trust message begins with the <trust-message> element. Its encryption attribute MUST specify the encryption protocol that uses the keys denoted by their identifiers. To send a trust message for keys of OpenPGP for XMPP (XEP-0373) [3] the attribute encryption='urn:xmpp:openpgp:0' or for keys of OMEMO Encryption (XEP-0384) [4] the attribute encryption='eu.siacs.conversations.axolotl' MUST be used. For other values there is an overview of possible encryption protocols[5]. A trust message MUST contain at least one <key-owner> element and each element MUST contain at least one <trust> or <distrust> element. Inside of each <trust> or <distrust> element there MUST be exactly one key identifier. Those elements are used for the following purposes:
In the following example the keys of the later given identifiers are used by the encryption protocol OMEMO Encryption (XEP-0384) [6] specified by eu.siacs.conversations.axolotl.

Listing 1: Specifying the Encryption Protocol of the Keys

```
<trust-message xmlns='urn:xmpp:trust-messages:0' encryption='eu.siacs.
    conversations.axolotl'>
```

In the following example the keys of the later given identifiers belong to alice@example.org.

Listing 2: Specifying the JID Owning the Keys

```
<key-owner jid='alice@example.org'>
```

In the following example the key corresponding to the identifier inside <trust> and </trust> is trusted by the sending endpoint.

Listing 3: Indicating the Trust in a Specific Key

```
<trust>6850019
    d7ed0feb6d3823072498ceb4f616c6025586f8f666dc6b9c81ef7e0a4</trust>
```

[3] XEP-0373: OpenPGP for XMPP <https://xmpp.org/extensions/xep-0373.html>.
[4] XEP-0384: OMEMO Encryption <https://xmpp.org/extensions/xep-0384.html>.
[5] Explicit Message Encryption - Encryption Protocols <https://xmpp.org/extensions/xep-0380.html#table-1>.
[6] XEP-0384: OMEMO Encryption <https://xmpp.org/extensions/xep-0384.html>.

In the following example the key corresponding to the identifier inside the <distrust> and </distrust> is not trusted by the sending endpoint.

Listing 4: Indicating the Distrust in a Specific Key

```
<distrust>
    b423f5088de9a924d51b31581723d850c7cc67d0a4fe6b267c3d301ff56d2413</
    distrust>
```

4 Use Cases

An endpoint of alice@example.org MAY send a trust message to other endpoints of alice@example.org, to contacts like bob@example.com or to a specific resource like carol@example.net/phone.
The usage of Message Carbons (XEP-0280) [7] for trust messages is RECOMMENDED. It minimizes the number of trust messages to be sent while having the same payload because trust messages with the same payload do not have to be sent for each endpoint. In combination with the usage of Message Archive Management (XEP-0313) [8], the delivery of trust messages to temporarily offline endpoints is ensured even if they are available under a different resource after going online than the last known one before going offline. Additionally, using Message Carbons (XEP-0280) [9] for every encrypted trust message will lead to send trust messages which are less distinguishable by analyzing their content from other encrypted messages using Stanza Content Encryption (XEP-0420) [10]. However, it may be possible to distinguish an encrypted trust message from other encrypted messages and therefore detect the fact that a specific authentication took place by analyzing the network traffic over a period of time but that is out of scope for this specification.
TODO: Move this paragraph to Stanza Content Encryption (XEP-0420) [11]. The following message attribute and element are RECOMMENDED because without having <body>, the goals of them would not be achieved. type='chat' is needed to deliver the trust message to all endpoints (see XEP-0280: Message Carbons). <store xmlns='urn:xmpp:hints'/> is needed to deliver the trust message to each offline endpoint after it went online (see XEP-0313: Message Archive Management and XEP-0334: Message Processing Hints).
In the following examples Alice's endpoint sends a trust message for OMEMO Encryption (XEP-0384) [12] (eu.siacs.conversations.axolotl) keys of own endpoints and Bob's endpoints to Carol's resource "phone". Alice's keys corresponding to the identifiers starting with "68" and "22" are trusted by Alice's endpoint connected via resource "laptop". Bob's key corresponding to the identifiers starting with "68" and "22" are trusted by Alice's endpoint connected via resource "laptop". Bob's key corresponding to the identifier starting with "62" is trusted by

[7] XEP-0280: Message Carbons <https://xmpp.org/extensions/xep-0280.html>.
[8] XEP-0313: Message Archive Management <https://xmpp.org/extensions/xep-0313.html>.
[9] XEP-0280: Message Carbons <https://xmpp.org/extensions/xep-0280.html>.
[10] XEP-0420: Stanza Content Encryption <https://xmpp.org/extensions/xep-0420.html>.
[11] XEP-0420: Stanza Content Encryption <https://xmpp.org/extensions/xep-0420.html>.
[12] XEP-0384: OMEMO Encryption <https://xmpp.org/extensions/xep-0384.html>.

Alice's endpoint connected via resource "laptop" but not Bob's keys corresponding to the identifiers starting with "b4" and "d9".

4.1 Unencrypted Trust Message

A trust message before encryption or without any encryption could look like the following example. Keep in mind, like said before, that the authenticity and integrity of the message MUST be ensured by a signing mechanism even if the message is not encrypted. However, the strength of trust messages is the possibility to encrypt them and to choose its recipients.

Listing 5: Alice's endpoint sends an unencrypted trust message to Carol

```
<message from='alice@example.org/laptop' to='carol@example.org' type='
    chat'>
  <store xmlns='urn:xmpp:hints'/>
  <trust-message xmlns='urn:xmpp:trust-messages:0' encryption='eu.
      siacs.conversations.axolotl'>
    <key-owner jid='alice@example.org'>
      <trust>6850019
          d7ed0feb6d3823072498ceb4f616c6025586f8f666dc6b9c81ef7e0a4</
          trust>
      <trust>221
          a4f8e228b72182b006e5ca527d3bddccf8d9e6feaf4ce96e1c451e8648020
          </trust>
    </key-owner>
    <key-owner jid='bob@example.com'>
      <trust>623548
          d3835c6d33ef5cb680f7944ef381cf712bf23a0119dabe5c4f252cd02f</
          trust>
      <distrust>
          b423f5088de9a924d51b31581723d850c7cc67d0a4fe6b267c3d301ff56d2413
          </distrust>
      <distrust>
          d9f849b6b828309c5f2c8df4f38fd891887da5aaa24a22c50d52f69b4a80817e
          </distrust>
    </key-owner>
  </trust-message>
</message>
```

4.2 Encrypted Trust Message

Like described in the introduction, it is possible to encrypt a trust message and send it only to endpoints whose keys have already been authenticated. Both actions are RECOMMENDED, especially for concealing the fact that an endpoint authenticated another endpoint's key. When using an end-to-end encryption like OMEMO Encryption

(XEP-0384) [13] which cannot encrypt arbitrary elements, Stanza Content Encryption (XEP-0420) [14] is needed to encrypt a trust message. The following example shows how such a message could look like. For encrypting with OpenPGP for XMPP (XEP-0373) [15], the element <encrypted xmlns='eu.siacs.conversations.axolotl'> MUST be replaced by <openpgp xmlns='urn:xmpp:openpgp:0'>, the element <envelope xmlns='urn:xmpp:sce:0'> by <signcrypt xmlns='urn:xmpp:openpgp:0'> and <header sid='27183'>...</header> MUST be removed.

Listing 6: Alice's endpoint sends an encrypted trust message to Carol

```
<message from='alice@example.org/laptop' to='carol@example.org' type='
    chat'>
  <store xmlns='urn:xmpp:hints'/>
  <encrypted xmlns='eu.siacs.conversations.axolotl'>
    <header sid='17183'>
    ...
    </header>
    <envelope xmlns='urn:xmpp:sce:0'>
      <rpad>QHqW2arWFewoERL1a43wonBKpTmsrBWnc1d66HSDq85NgMLmjrDJV9lV</
          rpad>
      <time stamp='2020-01-01T00:00:00'/>
      <from jid='alice@example.org/laptop'/>
      <to jid='carol@example.org'/>
      <payload>
        <trust-message xmlns='urn:xmpp:trust-messages:0' encryption='
            eu.siacs.conversations.axolotl'>
          <key-owner jid='alice@example.org'>
            <trust>6850019
                d7ed0feb6d3823072498ceb4f616c6025586f8f666dc6b9c81ef7e0a4
                </trust>
            <trust>221
                a4f8e228b72182b006e5ca527d3bddccf8d9e6feaf4ce96e1c451e8648020
                </trust>
          </key-owner>
          <key-owner jid='bob@example.com'>
            <trust>623548
                d3835c6d33ef5cb680f7944ef381cf712bf23a0119dabe5c4f252cd02f
                </trust>
            <distrust>
                b423f5088de9a924d51b31581723d850c7cc67d0a4fe6b267c3d301ff56d2413
                </distrust>
            <distrust>
                d9f849b6b828309c5f2c8df4f38fd891887da5aaa24a22c50d52f69b4a80817e
                </distrust>
          </key-owner>
        </trust-message>
```

[13] XEP-0384: OMEMO Encryption <https://xmpp.org/extensions/xep-0384.html>.
[14] XEP-0420: Stanza Content Encryption <https://xmpp.org/extensions/xep-0420.html>.
[15] XEP-0373: OpenPGP for XMPP <https://xmpp.org/extensions/xep-0373.html>.

```
      </payload>
    </envelope>
  </encrypted>
</message>
```

5 Implementation Notes

This specification uses Message Carbons (XEP-0280) [16] for sending a trust message to all endpoints of a contact or to all own endpoints at once. By sending a trust message to the contact, each endpoint of the contact and each own endpoint receives the same trust message by the server. Thus, a client needs to send the same trust message only once.
If not all endpoints of the contact should receive the trust message, the trust message MAY be sent to specific endpoints of the contact but for all own endpoints Message Carbons (XEP-0280) [17] MAY be used and vice versa. Even when a client does not yet have a contact, the client MAY use Message Carbons (XEP-0280) [18] for delivering a trust message to all own endpoints by sending it to the own bare JID. If then a client receives a trust message with its own full JID as the sender, it MAY discard that message directly without parsing the content.
Example: Alice's endpoint A1 authenticates the key of her endpoint A2. A1 sends the trust message for A2's key only once to all of Alice's and Bob's endpoints by using Message Carbons (XEP-0280) [19].
Attention: In that context, sending an encrypted trust message to all endpoints of a contact or to all own endpoints does not mean to encrypt it with the keys of all those endpoints. Instead, it only means that all of those endpoints should receive the trust message even if it is not encrypted for some of them and thereby not decryptable by those endpoints. Keep in mind that a trust message SHOULD only be encrypted for endpoints with authenticated keys.

6 IANA Considerations

REQUIRED.

7 XMPP Registrar Considerations

REQUIRED.

[16] XEP-0280: Message Carbons <https://xmpp.org/extensions/xep-0280.html>.
[17] XEP-0280: Message Carbons <https://xmpp.org/extensions/xep-0280.html>.
[18] XEP-0280: Message Carbons <https://xmpp.org/extensions/xep-0280.html>.
[19] XEP-0280: Message Carbons <https://xmpp.org/extensions/xep-0280.html>.

8 XML Schema

REQUIRED for protocol specifications.