XEP-0466: Ephemeral Messages

Maxime Buquet
mailto:pep@bouah.net
xmpp:pep@bouah.net

2022-05-17
Version 0.1.0

Status	Type	Short Name
Experimental	Standards Track	NOT_YET_ASSIGNED

This specification encourages a shift in privacy settings wrt. logging policies.

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

Existing protocols deployed in XMPP networks offer forward secrecy both on the transport (TLS) and message (OMEMO Encryption (XEP-0384) [1]) levels. Forward secrecy prevents recorded communications from being decrypted even if long term encryption keys are compromised by generating ephemeral keys and securely deleting them when they are no longer needed.
However, even though keys are deleted, message contents is retained client archives. While servers generally impose time limits on archives (messages, stored files, etc.), due to privacy laws (e.g., GDPR) and/or disk-space concerns, most XMPP clients still retain message content almost indefinitely even though it may not benefit a majority of their userbase. A device with an installed XMPP client that can be lost or stolen becomes the weakest link.
Unlike ephemeral keys, which have specified lifetimes, message contents cannot be removed immediately after being read. Users have to decide for how long they want to retain conversation contents. Verbally agreeing on the time interval and manually removing messages from all devices is cumbersome and error-prone.
This XEP defines a way to attach a timer value to messages which in order to specify for how long XMPP clients should store message contents. Besides that, it defines a way to synchronize common timer setting across all users of the conversation.
The specification does not depend on any encryption scheme and does not require encryption at all. Plaintext messages will still be readable by servers in between and will depend on trust placed on these server to apply their privacy policy or to respect a Message Processing Hints (XEP-0334) [2] store hint.
Other IM systems, such as Signal, Wickr, Wire and Telegram, already offer ephemeral messages. Signal offers timer synchronization feature for user groups and Telegram offers it for secret chats, which are limited to two users.

2 Requirements

What this specification tries to do:

- Provide a way to specify a timer value after which message contents must be deleted from user devices.
- Clearly define semantics of timer value for XMPP clients.
- Provide a way for a group of users to choose common value for ephemeral message timers and synchronize it across all devices.
- Allow users to vacate to other activities while still being able to keep track of chats, as before.

[1] XEP-0384: OMEMO Encryption <https://xmpp.org/extensions/xep-0384.html>.
[2] XEP-0334: Message Processing Hints <https://xmpp.org/extensions/xep-0334.html>.

What this XEP doesn't try to be:

- A way to securely ensure that logs won't be kept by parties included in chats.

3 Use Cases

3.1 Advertising support

A client implementing this specification MUST advertise the <ephemeral/> disco feature (as per Service Discovery (XEP-0030) [3]). When advertising the feature, a client will honor requests to discard messages after the agreed upon delay.

3.2 Sending an ephemeral message

An ephemeral message is a <message/> including an <ephemeral/> tag in the urn:xmpp:ephemeral:0 namespace, with an attribute timer (xs:unsignedInt) indicating the delay in seconds, after which a message MUST be discarded.
Ephemeral messages SHOULD be sent as usual on the bare JID of the contact, or as is specified for groupchats (e.g., MUC, MIX). If this includes sending to non-supporting clients, and they can be detected, sending clients SHOULD warn the user in some way. Clients MAY warn users anyway if non-supporting clients cannot be detected (e.g., when they don't get a directed presence).
Sending clients MAY include a <no-permanent-store/> Message Processing Hints (XEP-0334) [4] when not doing end-to-end encryption, even though this may break receiver clients' expectations regarding archive management, and cause even supporting devices not to see ephemeral information.

3.3 Negotiating a delay

Sending a message with an ephemeral tag is how a delay is negotiated in a chat. A client receiving a message with an ephemeral tag MUST honor the timer for the received messages, and SHOULD include it in turn in following messages.
To change the timer for the following messages, change the value when sending a new message.

3.4 Implicit timer negociation

A implicit negociation can be done by sending a message with no <body/>, that contains an <ephemeral/> tag and a timer attribute, specified in Sending an ephemeral message.

[3] XEP-0030: Service Discovery <https://xmpp.org/extensions/xep-0030.html>.
[4] XEP-0334: Message Processing Hints <https://xmpp.org/extensions/xep-0334.html>.

The message MUST also contain a <store> hint as described in Message Processing Hints (XEP-0334) [5] so that offline clients see it.

3.5 Opting-out of ephemeral messages

XXX: Help. How do I ensure the receiving client sees what I am going to send, if it's just a single message. Same issue as with negotiating the delay. (That is, if a client doesn't fetch all MAM, it may miss the message). Do I need to send <i-want-out/> forever?
A client that has previously been sending ephemeral messages can choose to stop sending them, and send regular messages instead, in which case it should tell the recipient:

```
<message from='vladimir@example.com/mobile' to='  @example.com' type='
    chat'>
     <body>  </body>
     <i-want-out/>
</message>
```

When the recipients sees the (TODO) <i-want-out/> element, it will stop including <ephemeral/>. The original client seeing no ephemeral tag is being included SHOULD stop sending the opt-out element.
TODO: Negociation within messages is wonky. If a client comes back online and this flag isn't in server archives anymore, it will send ephemeral messages again causing all devices to send them again. This might go on forever.

4 Example scenarios

4.1 Initiating chat

Rosa sends a regular chat message to Peter:

```
<message from='rosa@example.com/mobile' to='peter@example.com' type='
    chat'>
     <body>I have read the book you sent me, it was very insightful.</
        body>
</message>
```

Peter had his client previously configured to send ephemeral messages, before a chat with Rosa was opened. He replies:

```
<message from='peter@example.com/desktop' to='rosa@example.com' type='
    chat'>
```

[5] XEP-0334: Message Processing Hints <https://xmpp.org/extensions/xep-0334.html>.

```
    <body>Something</body>
    <ephemeral xmlns='urn:xmpp:ephemeral:0' timer='604800' />
</message>
```

Rosa's client tells her from now own, messages will disappear in a week (60 * 60 * 24 * 7). Before replying she decides a week is too long and changes her settings so that they now disappear in 5 days. Her client immediately sends an implicit timer negotiation. The message she just received from Peter however will still disappear in 7 days.

```
<message from='rosa@example.com/mobile' to='peter@example.com' type='
    chat'>
    <ephemeral xmlns='urn:xmpp:ephemeral:0' timer='432000' />
    <store xmlns="urn:xmpp:hints" />
</message>
```

Peter's client tells him messages will disappear in 5 days. Peter is fine with this and doesn't change his client settings. His client will continue including the ephemeral tag with the same timer value of 5 days.

```
<message from='peter@example.com/desktop' to='rosa@example.com' type='
    chat'>
    <body>I see you changed the settings slightly. It's␣just␣as␣good␣
        to␣me!</body>
␣␣␣␣<ephemeral␣xmlns='urn:xmpp:ephemeral:0'␣timer='432000'␣/>
</message>
```

5 Business Rules

Timers SHOULD be started once a user has seen/read a message, to give them the possibility to read it – in case the timer was too low, and/or they were taking a holiday from messaging.
XXX: Is "read" and/or "seen" defined anywhere? Should we settle on some definition?
Once it has been sent, the timer on a message cannot be changed.
Discarded messages SHOULD be noted as such in the client (e.g., "This message has disappeared"). Not just removed with no indication of the reason.
When using with encryption mechanisms that include an encrypted wrapper such as OpenPGP for XMPP (XEP-0373) [6] or Stanza Content Encryption (XEP-0420) [7], this element SHOULD be placed in the wrapper.

[6] XEP-0373: OpenPGP for XMPP <https://xmpp.org/extensions/xep-0373.html>.
[7] XEP-0420: Stanza Content Encryption <https://xmpp.org/extensions/xep-0420.html>.

6 Implementation Notes

Discarded messages shall be removed from memory and disk on a best effort basis.
Timers do not have to be exactly exact, the definition of ”seen” or ”read” not being consistent, and clock issues might also be a thing (use NTP?). This is also a best effort basis.
Ephemeral messages can be used with end-to-end encryption mechanisms. Both mechanisms are orthogonal. Messages are decrypted on the client and stored as plaintext in most cases when using end-to-end encryption.

7 Accessibility Considerations

OPTIONAL.

8 Internationalization Considerations

The message that appears once a message is discarded is a suggestion and should be adapted to the environment locale of the user.

9 Security Considerations

Ephemeral messages are not to be considered ”secure” in any way.
Even within well-meaning entities, requiring that messages be discarded and made impossible to retrieve, requires a lot more scrutiny in the specification and in implementations, and even then, is a really technically challenging task, to say the least.
In an adversarial context, requiring that sent messages be deleted from every devices receiving it (thus including to an attacker), requires a lot more control over the infrastructure in place and is not in scope for this specification. The author of this specification has no intention to specify DRM.
This specification doesn’t prevent an attacker to read messages sent to you after they get control of your device (e.g., stolen, confiscated). In this specific case, the situation is improved nonetheless as the spec helps reduce the overall amount of messages that stay on a given device, compared to the current community standards.
Note that if a message hasn’t been fetched by the client yet, using a timestamp instead of a timer doesn’t necessarily protect the user entirely. An attacker obtaining access to a device after a long time would have taken an image of the original device, gain access again at time of obtention of the device, replace the client to handle these particular messages differently. To counter this, a user would have to have go through the gymnastics of getting their server not to send any archive to this device, during the interval necessary to open the device.

10 IANA Considerations

This document requires no interaction with the the Internet Assigned Numbers Authority (IANA) [8]

11 XMPP Registrar Considerations

11.1 Protocol Namespaces

The XMPP Registrar [9] includes urn:xmpp:ephemeral:0 in its registery of protocol namespaces (see <https://xmpp.org/registrar/namespaces.html>).

12 Design Considerations

From the previous ephemeral-messages protoXEP, the requirement that made it incompatible with non-implementing clients has been removed, as well as the one that made clients using e2ee only send only to supporting clients. This is explained by the fact that the goal of this specification is to change privacy defaults in the ecosystem and not to prevent users from getting their messages and break user-experience substantially.
Another use-case mentioned (and alluded to in security considerations) was being able to send time-sensitive messages, or rather, messages that have no purpose after a given time and thus should disappear. This specification doesn't exactly answer it as it might have been necessary to start the timer at the exact same time on both sender and receiver, and as such, a timestamp would have been better. This behaviour can still be observed more or less if sender and receiver are active at the same time, but of course it will differ when the receiver comes back at a later time.
A minimal timer value was originally negociable, but was removed as it complicates the protocol substancially, and can directly be solved between users.
XXX: Do we want to use a per-"contact" model? How? With PEP? How would a client know which node to pick (of the two in a 1:1 chat, easier in MUC). What to do about the access model? This should also not be limited to contacts but whitelist may be annoying to manage. IQ negociation? This requires simultanous online-ness and also not likely with non-contacts as it would require directed presence.

[8] 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/>.

[9] 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 <https://xmpp.org/registrar/>.

13 XML Schema

REQUIRED for protocol specifications.