XEP-0454: OMEMO Media sharing

Daniel Gultsch
mailto:daniel@gultsch.de
xmpp:daniel@gultsch.de

2021-01-26
Version 0.1.0

Status	Type	Short Name
Experimental	Historical	NOT_YET_ASSIGNED

An informal way of sharing media files despite limitations in the OMEMO encryption

Legal

Copyright

Permissions

Warranty

Liability

Conformance

Contents

1 Introduction

OMEMO Encryption (XEP-0384) [1], despite already being deployed in multiple clients, currently suffers from the limitation of only being able to encrypt the message body. The current strategy for a mid term solution is to gather experience on stanza content encryption by implementing OpenPGP for XMPP (XEP-0373) [2] and then later apply the gathered knowledge to OMEMO. However end users are demanding working, end-to-end encrypted media sharing right now. For that reason client developers came up with a temporary work around that that utilizes HTTP File Upload (XEP-0363) [3] and puts the resulting URL and a symmetric key in the body of an OMEMO message. This XEP describes the technical details of the work around.

2 Requirements

- OMEMO Media Sharing should be relatively easy to implement in clients that already support OMEMO and HTTP File Upload
- Reutilize the same encryption mode, namely AES-256 in GCM mode, that is already used by OMEMO.
- Use a relatively strict syntax to communicate the URL, the key and an optional thumbnail so receiving clients can easily differentiate it from regular messages.

3 The AESGCM URI scheme

An entity wishing to share an end-to-end encrypted file first generates a 32 byte random key and a 12 byte random IV. After successfully requesting a slot for HTTP upload the file can be encrypted with AES-256 in Galois/Counter Mode (GCM) on the fly while uploading it via HTTP. The authentication tag MUST be appended to the end of the file.
To share the file the entity converts the HTTPS URL, the key and the IV to an aesgcm:// URL. Both IV and key are converted to their hex representation of 24 characters and 64 characters respectively and concatenated for a total of 88 characters (44 bytes). The IV comes first followed by the key. The resulting string is put in the anchor part of the aesgcm URL.

```
GET URL: https://download.montague.tld/4a771ac1-f0b2-4a4a-9700-
    f2a26fa2bb67/tr%C3%A8s%20cool.jpg
IV: 8c3d050e9386ec173861778f
Key: 68e9af38a97aaf82faa4063b4d0878a61261534410c8a84331eaac851759f587
Resulting URL: aesgcm://download.montague.tld/4a771ac1-f0b2-4a4a-9700-
    f2a26fa2bb67/tr%C3%A8s%20cool.jpg#8
```

[1] XEP-0384: OMEMO Encryption <https://xmpp.org/extensions/xep-0384.html>.
[2] XEP-0373: OpenPGP for XMPP <https://xmpp.org/extensions/xep-0373.html>.
[3] XEP-0363: HTTP File Upload <https://xmpp.org/extensions/xep-0363.html>.

```
    c3d050e9386ec173861778f68e9af38a97aaf82faa4063b4d0878a61261534410c8a84331eaac85175
```

Note: HTTP Upload has transport encryption as a MUST. Non HTTPS URLs MUST not be converted to the aesgcm URL scheme.
The resulting aesgcm URL is encrypted as an OMEMO message and send to the recipient(s).

4 Embedded Thumbnails

The sending entity MAY also generate a thumbnail as a JPEG data uri and include that in the same message. The aesgcm:// and the data:image/jpep, are seperated by a new line character. The message SHOULD NOT include anything else. The JPEG thumbnail SHOULD be kept small (approximately 5KiB) to not run into into stanza size limitations. As a result the resulting thumbnail is considered to only be a very blury, very rough representation of the image.

```
aesgcm://download.montague.tld/4a771ac1-f0b2-4a4a-9700-f2a26fa2bb67/tr
    %C3%A8s%20cool.jpg#8
    c3d050e9386ec173861778f68e9af38a97aaf82faa4063b4d0878a61261534410c8a84331eaac85175

data:image/jpeg,/9j/4…AAQSkZJRgABAQEBLAEsAAD
```

5 Business Rules

The parser on the receiving end should be very strict and only display OMEMO message as shared media that contain a valid aesgcm URL or a valid aesgcm URL followed by a valid data uri seperated by a single new line character.
Traditional media sharing with HTTP Upload uses Out-of-Band Data (XEP-0066) [4] to repeat the URL from the body and thereby communicating that the URL is in fact meant as media attchment as opposed a clickable link. For the aesgcm URL scheme no such annotation is necessary as aesgcm URLs are considered unique enough and are never supposed to stand alone in a message.

6 Implementation Notes

When requesting the HTTP Upload slot and attempting on the fly encryption the requesting entity MUST take into account that the encrypted file size is larger then the original file due to the block mode of AES and the appended authentication tag. Most crypto libraries should have a method to calculate the size of the resulting file.

[4] XEP-0066: Out of Band Data <https://xmpp.org/extensions/xep-0066.html>.

7 Security Considerations

A aesgcm URL MUST never be linkified and clients MUST NOT offer another direct way for users to open them in a browser as this could leak the anchor with the encryption key to the server operator. This is also the reason the aesgcm URL was choosen in the first place to prevent users from accidentally opening a HTTP URL in the browser.

8 IANA Considerations

The aesgcm scheme is not registered at the IANA, and it probably shouldn't be as it is mostly a hack in order to work around the limitations of OMEMO Encryption (XEP-0384) [5] version 0.3.0 with regards to the extensibility, as this extension could only encrypt the <body/> of a message.

[5] XEP-0384: OMEMO Encryption <https://xmpp.org/extensions/xep-0384.html>.