Workgroup: Network Working Group
Internet-Draft:
draft-ietf-jmap-smime-sender-extensions-03
Published: 13 March 2023
Intended Status: Informational
Expires: 14 September 2023
Authors: A. Melnikov
Isode Ltd

JMAP extension for S/MIME signing and encryption

Abstract

This document specifies an extension to JMAP for sending S/MIME signed and S/MIME encrypted messages.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 14 September 2023.

Table of Contents

1. Introduction

[RFC8621] is a JSON based application protocol for synchronising email data between a client and a server.

This document describes an extension to JMAP for sending S/MIME signed and encrypted messages . It allows JMAP server to sign/ encrypt messages on user's behalf.

2. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Addition to the capabilities object

The capabilities object is returned as part of the standard JMAP Session object; see the JMAP spec. Servers supporting _this_ specification MUST add a property called "urn:ietf:params:jmap:smime-advanced" to the capabilities object.

The value of this property is an empty object in both the JMAP session _capabilities_ property and an account's _accountCapabilities_ property.

4. Extension to Email/set for S/MIME signing and/or encryption

[RFC8621] defines Email/set method for creating new email messages. This document defines the following additional request arguments that can be used to create S/MIME signed and/or encrypted messages:

*smimeSign: "Boolean" (default: false). If included and has the value "true", this requests the JMAP server to create an S/MIME

signed message from the message constructed according to other specified arguments (the "original message"). This is done by encapsulating the original message either inside application/pkcs7-mime [RFC8551] or multipart/signed [RFC1847] container. (smimeSignOpaque argument (see below) controls which of the two mechanisms is used.) The signature's private key/certificate is associated with the email address in the Sender header field, if present; otherwise, it is associated with the email address in the From header field, if present.

If multiple addresses are present in one of these header fields, or there is more than one Sender/From header field, the server SHOULD reject the Email/set as invalid with the "invalidEmail" error code; otherwise, it MUST take the first address in the last Sender/From header field. If JMAP account is not authorized to sign message as the selected sender (as above), it SHOULD return "signedSenderNotAllowed" error code.

* smimeEncrypt: "Boolean" (default: false). If included and has the value "true", this requests the JMAP server to create an S/MIME encrypted message from the constructed message. This is done by encapsulating the message inside application/pkcs7-mime [RFC8551] media type. The message MUST be encrypted to the sender and all To/Cc/Bcc recipients. This extension assumes that there is some kind of per user or organizational addressbook, that can be used to lookup public keys of recipients. If lookup of a particular public key fails, or results in an expired or revoked certificate, the Email/set operation MUST fail with the "validEncryptionKeyNotFound" error code.

* smimeHeaderProtect: "Boolean" (default: true). If has the value "true", this requests the JMAP server to use S/MIME header protection as specified in [draft-ietf-lamps-header-protection] when at least one of smimeEncrypt/smimeSign is true.

* smimeSignOpaque: "Boolean" (default: true). If has the value "true", this requests the JMAP server to use application/pkcs7-mime media type for S/MIME signing, otherwise multipart/signed media type.

If both "smimeSign" and "smimeEncrypt" are set to true, the message is first signed and then the signed version is encrypted (in that order).

(Note that this extension doesn't allow management of private keys/certificates. How private keys are managed or configured for a particular user is out of scope for this document.)

```
      [[ "Email/set", {
        "accountId": "ue150411c",
        "create": {
          "k192": {
            "mailboxIds": {
              "2ea1ca41b38e": true
            },
            "keywords": {
              "$seen": true,
              "$draft": true
            },
            "from": [{
              "name": "Joe Bloggs",
              "email": "joe@example.com"
            }],
            "subject": "World domination",
            "receivedAt": "2021-07-07T01:03:11Z",
            "sentAt": "2021-07-10T11:03:11+10:00",
            "smimeSign": true,
            "smimeEncrypt": true,
            "bodyStructure": {
              "type": "text/plain",
              "partId": "bd48",
              "header:Content-Language": "en"
            },
            "bodyValues": {
              "bd48": {
                "value": "I have the most brilliant plan.  Let me tell
                  you all about it.",
                "isTruncated": false
              }
            }
          }
        }
      }, "0" ]]
```

This will result in the following response:

```
      [[ "Email/set", {
        "accountId": "ue150411c",
        "oldState": "780823",
        "newState": "780839",
        "created": {
          "k192": {
            "id": "Mf40b5f831efa7233b9eb1c7f",
            "blobId": "Gf40b5f831efa7233b9eb1c7f8f97d84eeeee64f7",
            "threadId": "Td957e72e89f516dc",
            "size": 5096
```

```
    }
  },
  ...
}, "0" ]]
```

Figure 1: Example 1:

5. Extension to Email/get for S/MIME decryption

[RFC8621] defines Email/get method for retrieving information about email messages. This document defines the following additional request arguments of the "bodyProperties" object that can be used to facilitate decryption of S/MIME encrypted messages:

* smimeBlobId: "Id|null". The id representing the raw octets of the decrypted contents of the part. When processing this request argument, the server first removes the Content-Transfer-Encoding (see then "blobId" request argument in RFC 8621). If the Content-Transfer-Encoding is supported by the server, the content is then S/MIME decrypted. The resulting Id can be used in Email/parse method to import the decrypted message. If the body part is not encrypted, the returned smimeBlobId attribute has the same value as the "blobId" attribute.

```
    [[ "Email/get", {
        "ids": [ "f123u986" ],
        "properties": [ "threadId", "mailboxIds", "from", "subject",
          "receivedAt", "htmlBody" ],
        "bodyProperties": [ "partId", "smimeBlobId", "size", "type" ]
      }, "#1" ]]
```

This will result in the following response:

```
    [[ "Email/get", {
     "accountId": "abc",
     "state": "41234123231",
     "list": [
       {
         "id": "f123u986",
         "threadId": "cb1314a",
         "mailboxIds": { "da123c": true },
         "from": [{ "name": "Mice Ace", "email": "mike@example.com" }],
         "subject": "Dinner on Saturday?",
         "receivedAt": "2023-03-09T14:12:00Z",

         "htmlBody": [{
           "partId": "1",
           "smimeBlobId": "B841623871",
           "size": 120537,
           "type": "text/html"
         }]
       }
     ]
   }, "#1" ]]
```

Figure 2: Example 2:

6. IANA Considerations

6.1. JMAP capability registration for "smime-advanced"

IANA is requested to register the "smime" JMAP Capability as follows:

Capability Name: "urn:ietf:params:jmap:smime-advanced"

Specification document: this document

Intended use: common

Change Controller: IETF

Security and privacy considerations: this document, Section 7

6.2. JMAP Error Codes Registry Updates

6.2.1. signedSenderNotAllowed error code

JMAP Error Code: signedSenderNotAllowed

Intended use: common

Change controller: IETF

Reference: This document, Section 4

Description: JMAP account is not authorized to S/MIME sign message as the specified sender.

6.2.2. validEncryptionKeyNotFound error code

JMAP Error Code: validEncryptionKeyNotFound

Intended use: common

Change controller: IETF

Reference: This document, Section 4

Description: S/MIME encrypted message can't be generated because no valid certificate (non expired and non revoked) can be found for one of recipients.

7. Security Considerations

This JMAP extension assumes trust between the user and the JMAP server for purposes of signing and encrypting messages on user's behalf.

This JMAP extension also relies on access to user's (or organization's) addressbook which contain up-to-date certificates for recipients.

This JMAP extension doesn't support management of user's private keys and corresponding certificates.

8. Normative References

[RFC1847] Galvin, J., Murphy, S., Crocker, S., and N. Freed, "Security Multiparts for MIME: Multipart/Signed and Multipart/Encrypted", RFC 1847, DOI 10.17487/RFC1847, October 1995, <https://www.rfc-editor.org/info/rfc1847>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8550] Schaad, J., Ramsdell, B., and S. Turner, "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 4.0 Certificate Handling", RFC 8550, DOI 10.17487/RFC8550, April 2019, <https://www.rfc-editor.org/info/rfc8550>.

[RFC8551] Schaad, J., Ramsdell, B., and S. Turner, "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 4.0 Message Specification", RFC 8551, DOI 10.17487/RFC8551, April 2019, <https://www.rfc-editor.org/info/rfc8551>.

[RFC8620] Jenkins, N. and C. Newman, "The JSON Meta Application Protocol (JMAP)", RFC 8620, DOI 10.17487/RFC8620, July 2019, <https://www.rfc-editor.org/info/rfc8620>.

[RFC8621] Jenkins, N. and C. Newman, "The JSON Meta Application Protocol (JMAP) for Mail", RFC 8621, DOI 10.17487/RFC8621, August 2019, <https://www.rfc-editor.org/info/rfc8621>.

[draft-ietf-lamps-header-protection] Gillmor, D. K., Hoeneisen, B., and A. Melnikov, "Header Protection for Cryptographically Protected E-mail", Work in Progress, Internet-Draft, draft-ietf-lamps-header-protection-13, 10 March 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-lamps-header-protection-13>.

Author's Address

Alexey Melnikov
Isode Ltd
14 Castle Mews
Hampton
TW12 2NP
United Kingdom

Email: Alexey.Melnikov@isode.com