JMAP extensions for S/MIME signing and encryption

draft-ietf-jmap-smime-sender-extensions-04
and
draft-melnikov-jmap-smime-sender-extensions-alt-06

Alexey Melnikov <alexey.melnikov@isode.com>

Summary of WG-04 draft (1 of 2)

- Extra parameters to "Email/set" command (both optionals) to control S/MIME signing and/or encryption:
  - "smimeSign": true
  - "smimeEncrypt": true
  - If both "smimeSign" and "smimeEncrypt" are set to true, the message is first signed and then the signed version is encrypted
  - smimeSignOpaque (boolean) parameter controls how S/MIME signing is done. The default value is “true” (use application/pkcs7-mime container). “false” means use of multipart/signed media type.

Summary of WG-04 draft (2 of 2)

- Extra parameters to "Email/query" command to be able to find “all encrypted” (or all non encrypted) or “all signed” (or all non signed) messages

Recent Changes to the alt draft, as compared to the WG draft

- Use terminology from draft-ietf-lamps-e2e-mail-guidance for referencing what is an encrypted message and what is a signed message

- Use smimeOperations array in Email/set, which allows more precise (and extensible) way to describe all transformations on the message. For example this would work for triple wrap (sign, encrypt, sign again).

- Added support for the LAMPS Header Protection specification, using 2 new arguments “headersToObscure” and “headersToProtect”. These are no longer a part of smimeOperations.

- Added new Email/smimeDecrypt operation, that can create a new decrypted blob from an encrypted blob or a sequence of blobs.

Email/smimeDecrypt

- Similar to Blob/upload command.
- Results of Email/smimeDecrypt can be fed into Email/parse later on
- Should this be named Blob/smimeDecrypt instead?

Differences in Email/set between two proposals

```
"Email/set", {
        "accountId": "ue150411c",
        "create": {

[...]

          "smimeSign": true,
          "smimeEncrypt": true,

    [...]
```

```
"smimeOperations": [{
   "operation": "sign",
   "options": {
    "opaque": false,
   }
  },
  {
    "operation": "encrypt"
  }
],
```

Comparison of 2 proposals

- draft-ietf-jmap-smime-sender-extensions-04:
  - pros: simple
  - cons: not extensible

- draft-melnikov-jmap-smime-sender-extensions-alt-06:
  - pros: flexible and extensible
  - cons: less simple

Next step

- Adopt draft-melnikov-jmap-smime-sender-extensions-alt-06 as a replacement for draft-ietf-jmap-smime-sender-extensions-04? Or carry on with simpler design in draft-ietf-jmap-smime-sender-extensions-04?