S/MIME related extensions to JMAP Mail

draft-ietf-jmap-smime-00

Alexey Melnikov <alexey.melnikov@isode.com>

Reporting status of signature verification

The "smimeStatus" response property for Email/get:

smimeStatus: "String|null". null signifies that the message doesn't contain any signature. Possible string values of the property are listed below. Servers MAY return other values not defined below.

Reporting status of signature verification

Possible values (Client MUST treat unrecognized values as "unknown"):

- unknown: S/MIME message, but it is neither signed, nor encrypted. This can also be returned for a multipart/signed message which contains unrecognized signing protocol (for example OpenPGP).
- signed: S/MIME signed message, but the signature was not yet verified. Some servers might not attempt to verify signature until a particular message is requested by the client.
- signed/verified: S/MIME signed message and the sender's signature was successfully verified, sender matches the From header field and the sender's certificate (and the certificate chain) is trusted for signing.
- signed/failed: S/MIME signed message, but the signature failed to verify. This might be a policy related decision (message signer doesn't match the From header field), message was modified, the signer's certificate has expired or was revoked, etc.

Further extensions for automatic Decryption/Encryption/Signing: credentials

The *Identity* object (jmap-mail, Section 6) is extended to support Signing/Decryption/Encryption:

credentials: "SmimeCredentials[]". SmimeCredentials object contains the following attributes:

- keyId: "String"
- certificate: "String" (DER representation of the certificate encoded in URL-safe base64 representation as defined in [RFC4648].
- forSigning: Boolean
- forEncryption: Boolean

-- These 2 attributes allow to separately control whether a certificate+private key is only used for signing, for encryption or both.

– Disabled keys can have both set to false, they can be used for decryption only

– Private key is deliberately not accessible through this interface. Should there be a way?

Further extensions for automatic Decryption/Verification

The "Email/parse" method takes the following extra argument:

- smimeDecode: "Boolean" (default: false) If "true", the specified Blobs are first decrypted (or if they represent signed only messages, the signed content is extracted) before each decoded content is parsed as an [RFC5322] message.

Further extensions for automatic Encryption/Signing: operations

The EmailSubmission object is extended to include the following extra attributes:

- smimeEncrypt: "SmimeEncrypt|null" SmimeEncrypt object contains the following attributes:

  keyId: "String". keyId of the private key that has forEncryption: true, that will be used as the originator of this message.

- smimeSign: "SmimeSign|null" SmimeSign object contains the following attributes:

  keyId: "String". keyId of the private key that has forSigning: true, that will be used as the originator of this message.