

LAMPS
Internet-Draft
Updates: 5480 (if approved)
Intended status: Standards Track
Expires: September 30, 2020

T. Ito
SECOM CO., LTD.
S. Turner
sn3rd
March 31, 2020

Clarifications for Elliptic Curve Cryptography Subject Public Key
Information
draft-ietf-lamps-5480-ku-clarifications-03

Abstract

This document updates RFC 5480 to specify semantics for the keyEncipherment and dataEncipherment key usage bits when used in certificates that support Elliptic Curve Cryptography.

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 <http://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 September 30, 2020.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Terminology	2
3. Updates to Section 3	2
4. Security Considerations	3
5. IANA Considerations	3
6. Normative References	3
Authors' Addresses	3

1. Introduction

[RFC5480] specifies the syntax and semantics for the Subject Public Key Information field in certificates that support Elliptic Curve Cryptography. As part of these semantics, it defines what combinations are permissible for the values of the key usage extension [RFC5280]. [RFC5480] specifies 7 of the 9 values; it makes no mention of keyEncipherment and dataEncipherment key usage bits. This document corrects this omission, by updating Section 3 of [RFC5480] to make it clear that neither keyEncipherment nor the dataEncipherment key usage bits are set for key agreement algorithms defined therein. The additions are to be made to the end of Section 3.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Updates to Section 3

If the keyUsage extension is present in a certificate that indicates id-ecPublicKey in SubjectPublicKeyInfo, then following values MUST NOT be present:

keyEncipherment; and
dataEncipherment.

If the keyUsage extension is present in a certificate that indicates id-ecDH or id-ecMQV in SubjectPublicKeyInfo, then the following values also MUST NOT be present:

keyEncipherment; and
dataEncipherment.

4. Security Considerations

This document introduces no new security considerations beyond those found in [RFC5480].

5. IANA Considerations

This document makes no request of IANA.

6. Normative References

- [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>>.
- [RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <<https://www.rfc-editor.org/info/rfc5280>>.
- [RFC5480] Turner, S., Brown, D., Yiu, K., Housley, R., and T. Polk, "Elliptic Curve Cryptography Subject Public Key Information", RFC 5480, DOI 10.17487/RFC5480, March 2009, <<https://www.rfc-editor.org/info/rfc5480>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

Authors' Addresses

Tadahiko Ito
SECOM CO., LTD.

Email: tadahiko.ito.public@gmail.com

Sean Turner
sn3rd

Email: sean@sn3rd.com

LAMPS Working Group
Internet-Draft
Updates: 4210, 5912, 6712 (if approved)
Intended status: Standards Track
Expires: 31 December 2022

H. Brockhaus, Ed.
D. von Oheimb
Siemens
J. Gray
Entrust
29 June 2022

Certificate Management Protocol (CMP) Updates
draft-ietf-lamps-cmp-updates-23

Abstract

This document contains a set of updates to the syntax and transfer of Certificate Management Protocol (CMP) version 2. This document updates RFC 4210, RFC 5912, and RFC 6712.

The aspects of CMP updated in this document are using EnvelopedData instead of EncryptedValue, clarifying the handling of p10cr messages, improving the crypto agility, as well as adding new general message types, extended key usages to identify certificates for use with CMP, and well-known URI path segments.

CMP version 3 is introduced to enable signaling support of EnvelopedData instead of EncryptedValue and signaling the use of an explicit hash AlgorithmIdentifier in certConf messages, as far as needed.

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 31 December 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	3
1.1. Convention and Terminology	4
2. Updates to RFC 4210 - Certificate Management Protocol (CMP)	5
2.1. New Section 1.1. - Changes Since RFC 4210	5
2.2. New Section 4.5 - Extended Key Usage	6
2.3. Update Section 5.1.1. - PKI Message Header	7
2.4. New Section 5.1.1.3. - CertProfile	8
2.5. Update Section 5.1.3.1. - Shared Secret Information	9
2.6. Replace Section 5.1.3.4 - Multiple Protection	9
2.7. Replace Section 5.2.2. - Encrypted Values	10
2.8. New Section 5.2.9 - GeneralizedTime	12
2.9. Update Section 5.3.4. - Certification Response	12
2.10. Update Section 5.3.18. - Certificate Confirmation Content	13
2.11. Update Section 5.3.19.2. - Signing Key Pair Types	14
2.12. Update Section 5.3.19.3. - Encryption/Key Agreement Key Pair Types	14
2.13. Replace Section 5.3.19.9. - Revocation Passphrase	15
2.14. New Section 5.3.19.14 - CA Certificates	15
2.15. New Section 5.3.19.15 - Root CA Certificate Update	15
2.16. New Section 5.3.19.16 - Certificate Request Template	16
2.17. New Section 5.3.19.17 - CRL Update Retrieval	18
2.18. Update Section 5.3.21 - Error Message Content	18
2.19. Replace Section 5.3.22 - Polling Request and Response	19
2.20. Update Section 7 - Version Negotiation	24
2.21. Update Section 7.1.1. - Clients Talking to RFC 2510 Servers	25
2.22. Add Section 8.4 - Private Keys for Certificate Signing and CMP Message Protection	25
2.23. Add Section 8.5 - Entropy of Random Numbers, Key Pairs, and Shared Secret Information	25

2.24. Add Section 8.6 - Trust Anchor Provisioning Using CMP Messages 26

2.25. Add Section 8.7 - Authorizing requests for certificates with specific EKUs 27

2.26. Update Appendix B - The Use of Revocation Passphrase . . 27

2.27. Update Appendix C - Request Message Behavioral Clarifications 28

2.28. Update Appendix D.1. - General Rules for Interpretation of These Profiles 29

2.29. Update Appendix D.2. - Algorithm Use Profile 30

2.30. Update Appendix D.4. - Initial Registration/Certification (Basic Authenticated Scheme) 30

3. Updates to RFC 6712 - HTTP Transfer for the Certificate Management Protocol (CMP) 30

3.1. Update Section 1. - Introduction 30

3.2. New Section 1.1. - Changes Since RFC 6712 31

3.3. Replace Section 3.6. - HTTP Request-URI 31

4. IANA Considerations 32

5. Security Considerations 34

6. Acknowledgements 34

7. References 34

7.1. Normative References 34

7.2. Informative References 36

Appendix A. ASN.1 Modules 38

A.1. Update to RFC4210 - 1988 ASN.1 Module 38

A.2. Update to RFC5912 - 2002 ASN.1 Module 52

Appendix B. History of Changes 65

Authors' Addresses 72

1. Introduction

[RFC Editor:

Please perform the following substitution.

* RFCXXXX --> the assigned numerical RFC value for this draft

Please update the following references to associated drafts in progress to reflect their final RFC assignments, if possible:

* I-D.ietf-lamps-cmp-algorithms

* I-D.ietf-lamps-lightweight-cmp-profile

* I-D.ietf-ace-cmpv2-coap-transport

]

While using CMP [RFC4210] in industrial and IoT environments and developing the Lightweight CMP Profile [I-D.ietf-lamps-lightweight-cmp-profile] some limitations were identified in the original CMP specification. This document updates RFC 4210 [RFC4210] and RFC 6712 [RFC6712] to overcome these limitations.

Among others, this document improves the crypto agility of CMP, which means to be flexible to react on future advances in cryptography.

This document also introduces new extended key usages to identify CMP endpoints on registration and certification authorities.

As the main content of RFC 4210 [RFC4210] and RFC 6712 [RFC6712] stays unchanged, this document lists all sections that are updated, replaced, or added to the current text of the respective RFCs.

The authors acknowledge that the style of the document is hard to read because the original RFCs must be read along with this document to get the complete content. The working group decided to use this approach in order to keep the changes to RFC 4210 [RFC4210] and RFC 6712 [RFC6712] to the required minimum. This was meant to speed up the editorial process and to minimize the effort spent on reviewing the whole text of the original documents.

1.1. Convention and Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Technical terminology is used in conformance with RFC 4210 [RFC4210], RFC 4211 [RFC4211], and RFC 5280 [RFC5280]. The following key words are used:

CA: Certification authority, which issues certificates.

RA: Registration authority, an optional system component to which a CA delegates certificate management functions such as authorization checks.

KGA: Key generation authority, which generates key pairs on behalf of an EE. The KGA could be co-located with an RA or a CA.

EE: End entity, a user, device, or service that holds a PKI

certificate. An identifier for the EE is given as its subject of the certificate.

2. Updates to RFC 4210 - Certificate Management Protocol (CMP)

2.1. New Section 1.1. - Changes Since RFC 4210

The following subsection describes feature updates to RFC 4210 [RFC4210]. They are always related to the base specification. Hence, references to the original sections in RFC 4210 [RFC4210] are used whenever possible.

Insert this section at the end of the current Section 1:

1.1. Changes Since RFC 4210

The following updates are made in this document:

- * Add new extended key usages for various CMP server types, e.g., registration authority and certification authority, to express the authorization of the entity identified in the certificate containing the respective extended key usage extension to act as the indicated PKI management entity.
- * Extend the description of multiple protection to cover additional use cases, e.g., batch processing of messages.
- * Offering EnvelopedData as the preferred choice next to EncryptedValue to better support crypto agility in CMP. Note that according to RFC 4211 [RFC4211] section 2.1. point 9 the use of the EncryptedValue structure has been deprecated in favor of the EnvelopedData structure. RFC 4211 [RFC4211] offers the EncryptedKey structure, a choice of EncryptedValue and EnvelopedData for migration to EnvelopedData. For reasons of completeness and consistency the type EncryptedValue has been exchanged in all occurrences in RFC 4210 [RFC4210]. This includes the protection of centrally generated private keys, encryption of certificates, and protection of revocation passphrases. To properly differentiate the support of EnvelopedData instead of EncryptedValue, the CMP version 3 is introduced in case a transaction is supposed to use EnvelopedData.
- * Offering an optional hashAlg field in CertStatus supporting confirmation of certificates signed with signature algorithms, e.g., EdDSA, not directly indicating a specific hash algorithm to use to compute the certHash.

- * Adding new general message types to request CA certificates, a root CA update, a certificate request template, or a CRL update.
- * Extend the usage of polling to p10cr, certConf, rr, genm, and error messages.
- * Delete the mandatory algorithm profile in RFC 4210 Appendix D.2 [RFC4210] and refer to CMP Algorithms Section 7 [I-D.ietf-lamps-cmp-algorithms].

2.2. New Section 4.5 - Extended Key Usage

The following subsection introduces a new extended key usage for CMP servers authorized to centrally generate key pairs on behalf of end entities.

Insert this section at the end of the current Section 4:

4.5. Extended Key Usage

The Extended Key Usage (EKU) extension indicates the purposes for which the certified key pair may be used. It therefore restricts the use of a certificate to specific applications.

A CA may want to delegate parts of its duties to other PKI management entities. This section provides a mechanism to both prove this delegation and enable an automated means for checking the authorization of this delegation. Such delegation may also be expressed by other means, e.g., explicit configuration.

To offer automatic validation for the delegation of a role by a CA to another entity, the certificates used for CMP message protection or signed data for central key generation MUST be issued by the delegating CA and MUST contain the respective EKUs. This proves the authorization of this entity by the delegating CA to act in the given role as described below.

The OIDs to be used for these EKUs are:

```
id-kp-cmCCA OBJECT IDENTIFIER ::= {
    iso(1) identified-organization(3) dod(6) internet(1)
    security(5) mechanisms(5) pkix(7) kp(3) 27 }

id-kp-cmCRA OBJECT IDENTIFIER ::= {
    iso(1) identified-organization(3) dod(6) internet(1)
    security(5) mechanisms(5) pkix(7) kp(3) 28 }

id-kp-cmKGA OBJECT IDENTIFIER ::= {
    iso(1) identified-organization(3) dod(6) internet(1)
    security(5) mechanisms(5) pkix(7) kp(3) 32 }
```

Note: RFC 6402 section 2.10 [RFC6402] specifies OIDs for a CMC CA and a CMC RA. As the functionality of a CA and RA is not specific to using CMC or CMP as the certificate management protocol, these EKUs are re-used by CMP.

The meaning of the id-kp-cmKGA EKU is as follows:

CMP KGA: CMP Key Generation Authorities are CAs or are identified by the id-kp-cmKGA extended key usage. The CMP KGA knows the private key it generated on behalf of the end entity. This is a very sensitive service and needs specific authorization, which by default is with the CA certificate itself. The CA may delegate its authorization by placing the id-kp-cmKGA extended key usage in the certificate used to authenticate the origin of the generated private key. The authorization may also be determined through local configuration of the end entity.

2.3. Update Section 5.1.1. - PKI Message Header

Section 5.1.1 of RFC 4210 [RFC4210] describes the PKI message header. This document introduces the new version 3 indicating support of EnvelopedData as specified in Section 2.7.

Replace the ASN.1 Syntax of PKIHeader and the subsequent description of pvno with the following text:

```

PKIHeader ::= SEQUENCE {
    pvno                INTEGER          { cmp1999(1), cmp2000(2),
                                         cmp2021(3) },
    sender              GeneralName,
    recipient           GeneralName,
    messageTime        [0] GeneralizedTime    OPTIONAL,
    protectionAlg      [1] AlgorithmIdentifier{ALGORITHM, {...}}
                       OPTIONAL,
    senderKID          [2] KeyIdentifier      OPTIONAL,
    recipKID           [3] KeyIdentifier      OPTIONAL,
    transactionID      [4] OCTET STRING      OPTIONAL,
    senderNonce        [5] OCTET STRING      OPTIONAL,
    recipNonce         [6] OCTET STRING      OPTIONAL,
    freeText           [7] PKIFreeText       OPTIONAL,
    generalInfo        [8] SEQUENCE SIZE (1..MAX) OF
                       InfoTypeAndValue     OPTIONAL
}

PKIFreeText ::= SEQUENCE SIZE (1..MAX) OF UTF8String

```

The usage of pvno values is described in Section 7.

2.4. New Section 5.1.1.3. - CertProfile

Section 5.1.1 of RFC 4210 [RFC4210] defines the PKIHeader and id-it OIDs to be used in the generalInfo field. This section introduces id-it-certProfile.

Insert this section after Section 5.1.1.2:

5.1.1.3. CertProfile

This is used by the EE to indicate specific certificate profiles, e.g., when requesting a new certificate or a certificate request template, see Section 5.3.19.16.

```

id-it-certProfile OBJECT IDENTIFIER ::= {id-it 21}
CertProfileValue ::= SEQUENCE SIZE (1..MAX) OF UTF8String

```

When used in an ir/cr/kur/genm, the value MUST NOT contain more elements than the number of CertReqMsg or InfoTypeAndValue elements and the certificate profile names refer to the elements in the given order.

When used in a p10cr, the value MUST NOT contain multiple certificate profile names.

2.5. Update Section 5.1.3.1. - Shared Secret Information

Section 5.1.3.1 of RFC 4210 [RFC4210] describes the MAC based protection of a PKIMessage using the algorithm id-PasswordBasedMac.

Replace the first paragraph with the following text:

In this case, the sender and recipient share secret information with sufficient entropy (established via out-of-band means or from a previous PKI management operation). PKIProtection will contain a MAC value and the protectionAlg MAY be one of the options described in CMP Algorithms [I-D.ietf-lamps-cmp-algorithms]. The PasswordBasedMac is specified as follows (see also [RFC4211] and [RFC9045]):

Replace the last paragraph with the following text (Note: This fixes Errata ID 2616):

Note: It is RECOMMENDED that the fields of PBMPParameter remain constant throughout the messages of a single transaction (e.g., ir/ip/certConf/pkiConf) to reduce the overhead associated with PasswordBasedMac computation.

2.6. Replace Section 5.1.3.4 - Multiple Protection

Section 5.1.3.4 of RFC 4210 [RFC4210] describes the nested message. This document enables using nested messages also for batch-delivery transport of PKI messages between PKI management entities and with mixed body types.

Replace the text of the section with the following text:

5.1.3.4. Multiple Protection

When receiving a protected PKI message, a PKI management entity such as an RA MAY forward that message adding its own protection (which is a MAC or a signature, depending on the information and certificates shared between the RA and the CA). Additionally, multiple PKI messages MAY be aggregated. There are several use cases for such messages.

- * The RA confirms having validated and authorized a message and forwards the original message unchanged.
- * The RA modifies the message(s) in some way (e.g., adds or modifies particular field values or adds new extensions) before forwarding them, then it MAY create its own desired PKIBody. If the changes made by the RA to PKIMessage break the POP of a certificate request, the RA MUST set the popo field to RAVerified. It MAY

include the original PKIMessage from the EE in the generalInfo field of PKIHeader of a nested message (to accommodate, for example, cases in which the CA wishes to check POP or other information on the original EE message). The infoType to be used in this situation is {id-it 15} (see Section 5.3.19 for the value of id-it) and the infoValue is PKIMessages (contents MUST be in the same order as the message in PKIBody).

- * A PKI management entity collects several messages that are to be forwarded in the same direction and forwards them in a batch. Request messages can be transferred as batch upstream (towards the CA); response or announce messages can be transferred as batch downstream (towards an RA, but not to the EE). This can for instance be used when bridging an off-line connection between two PKI management entities.

These use cases are accomplished by nesting the messages within a new PKI message. The structure used is as follows:

```
NestedMessageContent ::= PKIMessages
```

2.7. Replace Section 5.2.2. - Encrypted Values

Section 5.2.2 of RFC 4210 [RFC4210] describes the use of EncryptedValue to transport encrypted data. This document extends the encryption of data to preferably use EnvelopedData.

Replace the text of the section with the following text:

5.2.2. Encrypted Values

Where encrypted data (in this specification, private keys, certificates, or revocation passphrase) are sent in PKI messages, the EncryptedKey data structure is used.

```
EncryptedKey ::= CHOICE {  
    encryptedValue      EncryptedValue, -- deprecated  
    envelopedData       [0] EnvelopedData }
```

See CRMF [RFC4211] for EncryptedKey and EncryptedValue syntax and CMS [RFC5652] for EnvelopedData syntax. Using the EncryptedKey data structure offers the choice to either use EncryptedValue (for backward compatibility only) or EnvelopedData. The use of the EncryptedValue structure has been deprecated in favor of the EnvelopedData structure. Therefore, it is RECOMMENDED to use EnvelopedData.

Note: The EncryptedKey structure defined in CRMF [RFC4211] is reused here, which makes the update backward compatible. Using the new syntax with the untagged default choice EncryptedValue is bits-on-the-wire compatible with the old syntax.

To indicate support for EnvelopedData the pvno cmp2021 has been introduced. Details on the usage of pvno values is described in Section 7.

The EncryptedKey data structure is used in CMP to transport a private key, certificate, or revocation passphrase in encrypted form.

EnvelopedData is used as follows:

- * It contains only one RecipientInfo structure because the content is encrypted only for one recipient.
- * It may contain a private key in the AsymmetricKeyPackage structure as defined in RFC 5958 [RFC5958] wrapped in a SignedData structure as specified in CMS section 5 [RFC5652] and [RFC8933] signed by the Key Generation Authority.
- * It may contain a certificate or revocation passphrase directly in the encryptedContent field.

The content of the EnvelopedData structure, as specified in CMS section 6 [RFC5652], MUST be encrypted using a newly generated symmetric content-encryption key. This content-encryption key MUST be securely provided to the recipient using one of three key management techniques.

The choice of the key management technique to be used by the sender depends on the credential available at the recipient:

- * Recipient's certificate that contains a key usage extension asserting keyAgreement: The content-encryption key will be protected using the key agreement key management technique, as specified in CMS section 6.2.2 [RFC5652]. This is the preferred technique.
- * Recipient's certificate that contains a key usage extension asserting keyEncipherment: The content-encryption key will be protected using the key transport key management technique, as specified in CMS section 6.2.1 [RFC5652].
- * A password or shared secret: The content-encryption key will be protected using the password-based key management technique, as specified in CMS section 6.2.4 [RFC5652].

2.8. New Section 5.2.9 - GeneralizedTime

The following subsection point implementers to [RFC5280] regarding usage of GeneralizedTime.

Insert this section after Section 5.2.8.4:

5.2.9 GeneralizedTime

GeneralizedTime is a standard ASN.1 type and SHALL be used as specified in RFC 5280 Section 4.1.2.5.2 [RFC5280].

2.9. Update Section 5.3.4. - Certification Response

Section 5.3.4 of RFC 4210 [RFC4210] describes the Certification Response. This document updates the syntax by using the parent structure EncryptedKey instead of EncryptedValue as described in Section 2.7 above. Additionally, it clarifies the certReqId to be used in response to a p10cr message.

Replace the ASN.1 syntax with the following text (Note: This also fixes Errata ID 3949 and 4078):

```

CertRepMessage ::= SEQUENCE {
    caPubs          [1] SEQUENCE SIZE (1..MAX) OF CMPCertificate
                   OPTIONAL,
    response        SEQUENCE OF CertResponse
}

CertResponse ::= SEQUENCE {
    certReqId       INTEGER,
    status          PKIStatusInfo,
    certifiedKeyPair CertifiedKeyPair OPTIONAL,
    rspInfo         OCTET STRING OPTIONAL
    -- analogous to the id-regInfo-utf8Pairs string defined
    -- for regInfo in CertReqMsg [RFC4211]
}

CertifiedKeyPair ::= SEQUENCE {
    certOrEncCert   CertOrEncCert,
    privateKey      [0] EncryptedKey OPTIONAL,
    -- see [RFC4211] for comment on encoding
    publicationInfo [1] PKIPublicationInfo OPTIONAL
}

CertOrEncCert ::= CHOICE {
    certificate      [0] CMPCertificate,
    encryptedCert    [1] EncryptedKey
}

```

Add the following as a new paragraph right after the ASN.1 syntax:

A p10cr message contains exactly one CertificationRequestInfo data structure as specified in PKCS#10 [RFC2986] but no certReqId. Therefore, the certReqId in the corresponding certification response (cp) message MUST be set to -1.

Add the following as new paragraphs to the end of the section:

The use of EncryptedKey is described in Section 5.2.2.

Note: To indicate support for EnvelopedData the pvno cmp2021 has been introduced. Details on the usage of different pvno values are described in Section 7.

2.10. Update Section 5.3.18. – Certificate Confirmation Content

This section introduces an optional hashAlg field to the CertStatus type used in certConf messages to explicitly specify the hash algorithm for those certificates where no hash algorithm is specified in the signatureAlgorithm field.

Replace the ASN.1 Syntax of CertStatus with the following text:

```
CertStatus ::= SEQUENCE {
    certHash      OCTET STRING,
    certReqId     INTEGER,
    statusInfo    PKIStatusInfo OPTIONAL,
    hashAlg [0] AlgorithmIdentifier{DIGEST-ALGORITHM, {...}}
                OPTIONAL
}
```

The hashAlg field SHOULD be used only in exceptional cases where the signatureAlgorithm of the certificate to be confirmed does not specify a hash algorithm in the OID or in the parameters. In such cases, e.g., for EdDSA, the hashAlg MUST be used to specify the hash algorithm to be used for calculating the certHash value. Otherwise, the certHash value SHALL be computed using the same hash algorithm as used to create and verify the certificate signature. If hashAlg is used, the CMP version indicated by the certConf message header must be cmp2021(3).

2.11. Update Section 5.3.19.2. – Signing Key Pair Types

The following section clarifies the usage of the Signing Key Pair Types on referencing EC curves.

Insert this note at the end of Section 5.3.19.2:

Note: In case several EC curves are supported, several id-ecPublicKey elements as defined in RFC 5480 [RFC5480] need to be given, one per named curve.

2.12. Update Section 5.3.19.3. – Encryption/Key Agreement Key Pair Types

The following section clarifies the use of the Encryption/Key Agreement Key Pair Types on referencing EC curves.

Insert this note at the end of Section 5.3.19.3:

Note: In case several EC curves are supported, several id-ecPublicKey elements as defined in RFC 5480 [RFC5480] need to be given, one per named curve.

2.13. Replace Section 5.3.19.9. - Revocation Passphrase

Section 5.3.19.9 of RFC 4210 [RFC4210] describes the provisioning of a revocation passphrase for authenticating a later revocation request. This document updates the handling by using the parent structure EncryptedKey instead of EncryptedValue to transport this information as described in Section 2.7 above.

Replace the text of the section with the following text:

5.3.19.9. Revocation Passphrase

This MAY be used by the EE to send a passphrase to a CA/RA for the purpose of authenticating a later revocation request (in the case that the appropriate signing private key is no longer available to authenticate the request). See Appendix B for further details on the use of this mechanism.

```
GenMsg:    {id-it 12}, EncryptedKey
GenRep:    {id-it 12}, < absent >
```

The use of EncryptedKey is described in Section 5.2.2.

2.14. New Section 5.3.19.14 - CA Certificates

The following subsection describes PKI general messages using id-it-caCerts. The intended use is specified in Lightweight CMP Profile Section 4.3 [I-D.ietf-lamps-lightweight-cmp-profile].

Insert this section after Section 5.3.19.13:

2.3.19.14 CA Certificates

This MAY be used by the client to get CA certificates.

```
GenMsg:    {id-it 17}, < absent >
GenRep:    {id-it 17}, SEQUENCE SIZE (1..MAX) OF
           CMPCertificate | < absent >
```

2.15. New Section 5.3.19.15 - Root CA Certificate Update

The following subsection describes PKI general messages using id-it-rootCaCert and id-it-rootCaKeyUpdate. The use is specified in Lightweight CMP Profile Section 4.3 [I-D.ietf-lamps-lightweight-cmp-profile].

Insert this section after new Section 5.3.19.14:

5.3.19.15. Root CA Certificate Update

This MAY be used by the client to get an update of a root CA certificate, which is provided in the body of the request message. In contrast to the ckuann message this approach follows the request/response model.

The EE SHOULD reference its current trust anchor in a TrustAnchor structure in the request body, giving the root CA certificate if available, otherwise the public key value of the trust anchor.

```
GenMsg:    {id-it 20}, RootCaCertValue | < absent >
GenRep:    {id-it 18}, RootCaKeyUpdateContent | < absent >
```

```
RootCaCertValue ::= CMPCertificate
```

```
RootCaKeyUpdateValue ::= RootCaKeyUpdateContent
```

```
RootCaKeyUpdateContent ::= SEQUENCE {
    newWithNew      CMPCertificate,
    newWithOld      [0] CMPCertificate OPTIONAL,
    oldWithNew      [1] CMPCertificate OPTIONAL
}
```

Note: In contrast to CAKeyUpdAnnContent, this type offers omitting newWithOld and oldWithNew in the GenRep message, depending on the needs of the EE.

2.16. New Section 5.3.19.16 - Certificate Request Template

The following subsection introduces the PKI general message using id-it-certReqTemplate. Details are specified in the Lightweight CMP Profile Section 4.3 [I-D.ietf-lamps-lightweight-cmp-profile].

Insert this section after new Section 5.3.19.15:

5.3.19.16. Certificate Request Template

This MAY be used by the client to get a template containing requirements for certificate request attributes and extensions. The controls id-regCtrl-algId and id-regCtrl-rsaKeyLen MAY contain details on the types of subject public keys the CA is willing to certify.

The `id-regCtrl-algId` control MAY be used to identify a cryptographic algorithm, see RFC 5280 Section 4.1.2.7 [RFC5280], other than `rsaEncryption`. The algorithm field SHALL identify a cryptographic algorithm. The contents of the optional parameters field will vary according to the algorithm identified. For example, when the algorithm is set to `id-ecPublicKey`, the parameters identify the elliptic curve to be used, see [RFC5480].

The `id-regCtrl-rsaKeyLen` control SHALL be used for algorithm `rsaEncryption` and SHALL contain the intended modulus bit length of the RSA key.

```

GenMsg:      {id-it 19}, < absent >
GenRep:      {id-it 19}, CertReqTemplateContent | < absent >

CertReqTemplateValue ::= CertReqTemplateContent

CertReqTemplateContent ::= SEQUENCE {
    certTemplate      CertTemplate,
    keySpec           Controls OPTIONAL }

Controls ::= SEQUENCE SIZE (1..MAX) OF AttributeTypeAndValue

id-regCtrl-algId OBJECT IDENTIFIER ::= { iso(1)
    identified-organization(3) dod(6) internet(1) security(5)
    mechanisms(5) pkix(7) pkix(5) regCtrl(1) 11 }

AlgIdCtrl ::= AlgorithmIdentifier{ALGORITHM, {...}}

id-regCtrl-rsaKeyLen OBJECT IDENTIFIER ::= { iso(1)
    identified-organization(3) dod(6) internet(1) security(5)
    mechanisms(5) pkix(7) pkix(5) regCtrl(1) 12 }

RsaKeyLenCtrl ::= INTEGER (1..MAX)

```

The `CertReqTemplateValue` contains the prefilled `certTemplate` to be used for a future certificate request. The `publicKey` field in the `certTemplate` MUST NOT be used. In case the PKI management entity wishes to specify supported public-key algorithms, the `keySpec` field MUST be used. One `AttributeTypeAndValue` per supported algorithm or RSA key length MUST be used.

Note: The `Controls` ASN.1 type is defined in CRMF Section 6 [RFC4211]

2.17. New Section 5.3.19.17 - CRL Update Retrieval

The following subsection introduces the PKI general message using `id-it-crlStatusList` and `id-it-crls`. Details are specified in the Lightweight CMP Profile Section 4.3 [I-D.ietf-lamps-lightweight-cmp-profile]. Insert this section after new Section 5.3.19.16:

5.3.19.17. CRL Update Retrieval

This MAY be used by the client to get new CRLs, specifying the source of the CRLs and the `thisUpdate` value of the latest CRL it already has, if available. A CRL source is given either by a `DistributionPointName` or the `GeneralNames` of the issuing CA. The `DistributionPointName` should be treated as an internal pointer to identify a CRL that the server already has and not as a way to ask the server to fetch CRLs from external locations. The server shall provide only those CRLs that are more recent than the ones indicated by the client.

```
GenMsg:      {id-it 22}, SEQUENCE SIZE (1..MAX) OF CRLStatus
GenRep:      {id-it 23}, SEQUENCE SIZE (1..MAX) OF
              CertificateList | < absent >
```

```
CRLSource ::= CHOICE {
  dpn          [0] DistributionPointName,
  issuer       [1] GeneralNames }
```

```
CRLStatus ::= SEQUENCE {
  source       CRLSource,
  thisUpdate   Time OPTIONAL }
```

2.18. Update Section 5.3.21 - Error Message Content

Section 5.3.21 of RFC 4210 [RFC4210] describes the regular use of error messages. This document adds a use by a PKI management entity to initiate delayed delivery in response to `certConf`, `rr`, and `genm` requests and to error messages.

Replace the first sentence of the first paragraph with the following one:

This data structure MAY be used by EE, CA, or RA to convey error info and by a PKI management entity to initiate delayed delivery of responses.

Replace the second paragraph with the following text:

This message MAY be generated at any time during a PKI transaction. If the client sends this request, the server MUST respond with a PKIConfirm response, or another ErrorMsg if any part of the header is not valid. In case a PKI management entity sends an error message to the EE with the pKIStatusInfo field containing the status "waiting", the EE will initiate polling as described in Section 5.3.22. Otherwise, both sides MUST treat this message as the end of the transaction (if a transaction is in progress).

2.19. Replace Section 5.3.22 - Polling Request and Response

Section 5.3.22 of RFC 4210 [RFC4210] describes when and how polling messages are used for ir, cr, and kur messages. This document extends the polling mechanism for outstanding responses to any kind of request message. This update also fixes the inconsistent use of the terms 'rReq' vs. 'pollReq' and 'pRep' vs. 'pollRep'.

Replace Section 5.3.22 with following text:

This pair of messages is intended to handle scenarios in which the client needs to poll the server to determine the status of an outstanding response (i.e., when the "waiting" PKIStatus has been received).

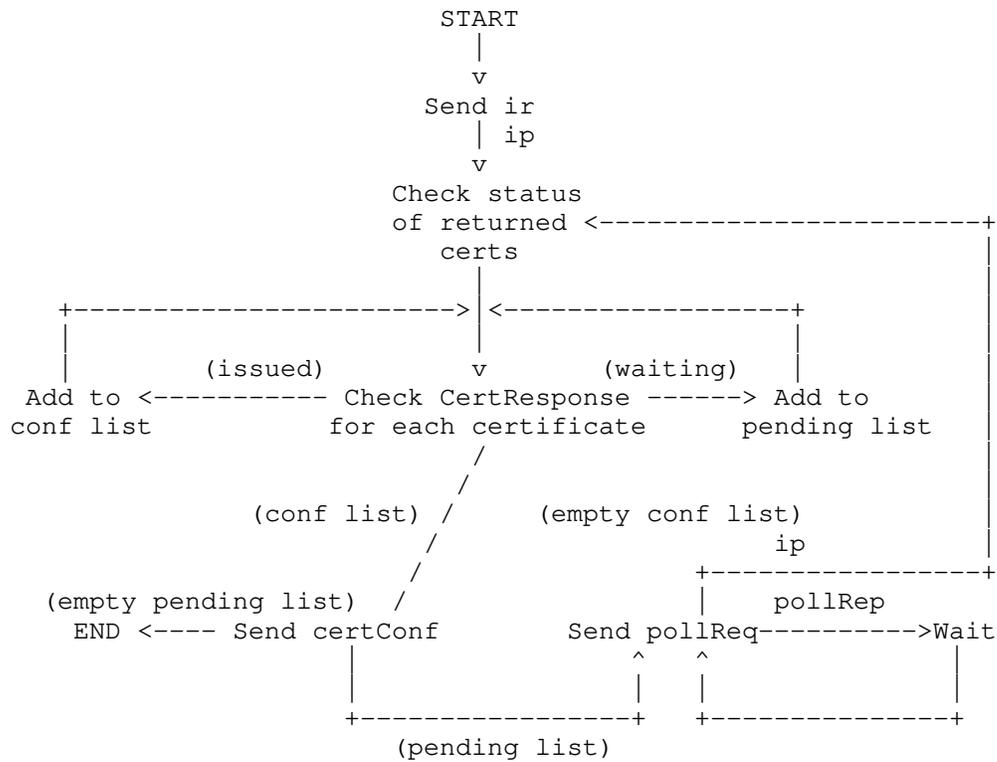
```
PollReqContent ::= SEQUENCE OF SEQUENCE {
    certReqId    INTEGER }

PollRepContent ::= SEQUENCE OF SEQUENCE {
    certReqId    INTEGER,
    checkAfter   INTEGER, -- time in seconds
    reason       PKIFreeText OPTIONAL }
```

In response to an ir, cr, p10cr, or kur request message, polling is initiated with an ip, cp, or kup response message containing status "waiting". For any type of request message, polling can be initiated with an error response messages with status "waiting". The following clauses describe how polling messages are used. It is assumed that multiple certConf messages can be sent during transactions. There will be one sent in response to each ip, cp, or kup that contains a CertStatus for an issued certificate.

- 1 In response to an ip, cp, or kup message, an EE will send a certConf for all issued certificates and expect a PKIConf for each certConf. An EE will send a pollReq message in response to each CertResponse element of an ip, cp, or kup message with status "waiting" and in response to an error message with status "waiting". Its certReqId MUST be either the index of a CertResponse data structure with status "waiting" or -1 referring to the complete response.
- 2 In response to a pollReq, a CA/RA will return an ip, cp, or kup if one or more of still pending requested certificates are ready or the final response to some other type of request is available; otherwise, it will return a pollRep.
- 3 If the EE receives a pollRep, it will wait for at least the number of seconds given in the checkAfter field before sending another pollReq.
- 4 If the EE receives an ip, cp, or kup, then it will be treated in the same way as the initial response; if it receives any other response, then this will be treated as the final response to the original request.

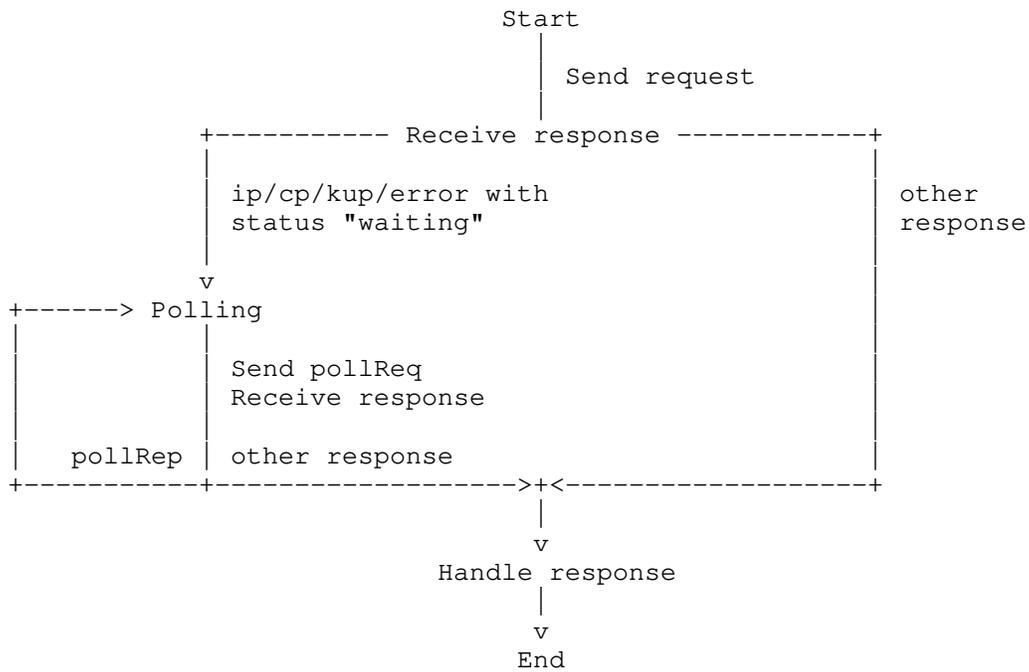
The following client-side state machine describes polling for individual CertResponse elements.



In the following exchange, the end entity is enrolling for two certificates in one request.

Step	End Entity			PKI
1	Format ir			
2		-> ir	->	
3				Handle ir
4				Manual intervention is required for both certs.
5		<- ip	<-	
6	Process ip			
7	Format pollReq			
8		-> pollReq	->	
9				Check status of cert requests
10				Certificates not ready
11				Format pollRep
12		<- pollRep	<-	
13	Wait			
14	Format pollReq			
15		-> pollReq	->	
16				Check status of cert requests
17				One certificate is ready
18				Format ip
19		<- ip	<-	
20	Handle ip			
21	Format certConf			
22		-> certConf	->	
23				Handle certConf
24				Format ack
25		<- pkiConf	<-	
26	Format pollReq			
27		-> pollReq	->	
28				Check status of certificate
29				Certificate is ready
30				Format ip
31		<- ip	<-	
31	Handle ip			
32	Format certConf			
33		-> certConf	->	
34				Handle certConf
35				Format ack
36		<- pkiConf	<-	

The following client-side state machine describes polling for a complete response message.



In the following exchange, the end-entity is sending a general message request, and the response is delayed by the server.

Step	End Entity	PKI
1	Format genm	
2		-> genm ->
3		Handle genm
4		delay in response is necessary
5		Format error message "waiting" with certReqId set to -1
6		<- error <-
7	Process error	
8	Format pollReq	
9		-> pollReq ->
10		Check status of original request general message response not ready
11		Format pollRep
12		<- pollRep <-
13	Wait	
14	Format pollReq	
15		-> pollReq ->
16		Check status of original request general message response is ready
17		Format genp
18		<- genp <-
19	Handle genp	

2.20. Update Section 7 - Version Negotiation

Section 7 of RFC 4210 [RFC4210] describes the use of CMP protocol versions. This document describes the handling of the additional CMP version `cmp2021` introduced to indicate support of `EnvelopedData` and `hashAlg`.

Replace the text of the second paragraph with the following text:

If a client knows the protocol version(s) supported by the server (e.g., from a previous `PKIMessage` exchange or via some out-of-band means), then it MUST send a `PKIMessage` with the highest version supported by both it and the server. If a client does not know what version(s) the server supports, then it MUST send a `PKIMessage` using the highest version it supports, with the following exception. Version `cmp2021` SHOULD only be used if `cmp2021` syntax is needed for the request being sent or for the expected response.

Note: Using `cmp2000` as the default `pvno` is done to avoid extra message exchanges for version negotiation and to foster compatibility with `cmp2000` implementations. Version `cmp2021` syntax is only needed if a message exchange uses `hashAlg` (in `CertStatus`) or `EnvelopedData`.

2.21. Update Section 7.1.1. - Clients Talking to RFC 2510 Servers

Section 7.1.1 of RFC 4210 [RFC4210] describes the behavior of a client sending a cmp2000 message talking to a cmp1999 server as specified in RFC 2510 [RFC2510]. This document extends the section to clients with any higher version than cmp1999.

Replace the first sentence of Section 7.1.1 with the following text:

If, after sending a message with a protocol version number higher than cmp1999, a client receives an ErrorMessageContent with a version of cmp1999, then it MUST abort the current transaction.

2.22. Add Section 8.4 - Private Keys for Certificate Signing and CMP Message Protection

The following subsection addresses the risk arising from reusing the CA private key for CMP message protection.

Insert this section after Section 8.3 (Note: This fixes Errata ID 5731):

8.4. Private Keys for Certificate Signing and CMP Message Protection

A CA should not reuse its certificate signing key for other purposes such as protecting CMP responses and TLS connections. This way, exposure to other parts of the system and the number of uses of this particularly critical key is reduced to a minimum.

2.23. Add Section 8.5 - Entropy of Random Numbers, Key Pairs, and Shared Secret Information

The following subsection addresses the risk arising from low entropy of random numbers, asymmetric keys, and shared secret information.

Insert this section after Section 8.4:

8.5. Entropy of Random Numbers, Key Pairs, and Shared Secret Information

Implementations must generate nonces and private keys from random input. The use of inadequate pseudo-random number generators (PRNGs) to generate cryptographic keys can result in little or no security. An attacker may find it much easier to reproduce the PRNG environment that produced the keys and to search the resulting small set of possibilities than brute-force searching the whole key space. As an example of predictable random numbers see [CVE-2008-0166]; consequences of low-entropy random numbers are discussed in Mining

Your Ps and Qs [MiningPsQs]. The generation of quality random numbers is difficult. ISO/IEC 20543:2019 [ISO.20543-2019], NIST SP 800-90A Rev.1 [NIST.SP.800-90Ar1], BSI AIS 31 V2.0 [AIS31], and others offer valuable guidance in this area.

If shared secret information is generated by a cryptographically secure random-number generator (CSRNG) it is safe to assume that the entropy of the shared secret information equals its bit length. If no CSRNG is used, the entropy of a shared secret information depends on the details of the generation process and cannot be measured securely after it has been generated. If user-generated passwords are used as shared secret information, their entropy cannot be measured and are typically insufficient for protected delivery of centrally generated keys or trust anchors.

If the entropy of a shared secret information protecting the delivery of a centrally generated key pair is known, it should not be less than the security strength of that key pair; if the shared secret information is re-used for different key pairs, the security of the shared secret information should exceed the security strength of each individual key pair.

For the case of a PKI management operation that delivers a new trust anchor (e.g., a root CA certificate) using caPubs or genm (a) that is not concluded in a timely manner or (b) where the shared secret information is re-used for several key management operations, the entropy of the shared secret information, if known, should not be less than the security strength of the trust anchor being managed by the operation. The shared secret information should have an entropy that at least matches the security strength of the key material being managed by the operation. Certain use cases may require shared secret information that may be of a low security strength, e.g., a human generated password. It is RECOMMENDED that such secret information be limited to a single PKI management operation.

2.24. Add Section 8.6 - Trust Anchor Provisioning Using CMP Messages

The following subsection addresses the risk arising from in-band provisioning of new trust anchors in a PKI management operation.

Insert this section after new Section 8.5:

8.6. Trust Anchor Provisioning Using CMP Messages

A provider of trust anchors, which may be an RA involved in configuration management of its clients, MUST NOT include to-be-trusted CA certificates in a CMP message unless the specific deployment scenario can ensure that it is adequate that the receiving EE trusts these certificates, e.g., by loading them into its trust store.

Whenever an EE receives in a CMP message, e.g., in the caPubs field of a certificate response or in a general response (genp), a CA certificate for use as a trust anchor, it MUST properly authenticate the message sender with existing trust anchors without requiring new trust anchors included in the message.

Additionally, the EE MUST verify that the sender is an authorized source of trust anchors. This authorization is governed by local policy and typically indicated using shared secret information or with a signature-based message protection using a certificate issued by a PKI that is explicitly authorized for this purpose.

2.25. Add Section 8.7 - Authorizing requests for certificates with specific EKUs

The following subsection addresses the security considerations to follow when authorizing requests for certificates containing specific EKUs.

Insert this section after new Section 8.6:

8.7. Authorizing requests for certificates with specific EKUs

When a CA issues a certificate containing extended key usage extensions as defined in Section 4.5, this expresses delegation of an authorization that originally is only with the CA certificate itself. Such delegation is a very sensitive action in a PKI and therefore special care must be taken when approving such certificate requests to ensure that only legitimate entities receive a certificate containing such an EKU.

2.26. Update Appendix B - The Use of Revocation Passphrase

Appendix B of RFC 4210 [RFC4210] describes the use of the revocation passphrase. As this document updates RFC 4210 [RFC4210] to utilize the parent structure EncryptedKey instead of EncryptedValue as described in Section 2.7 above, the description is updated accordingly.

Replace the first bullet point of this section with the following text:

- * The OID and value specified in Section 5.3.19.9 MAY be sent in a GenMsg message at any time, or MAY be sent in the generalInfo field of the PKIHeader of any PKIMessage at any time. (In particular, the EncryptedKey structure as described in Section 5.2.2 may be sent in the header of the certConf message that confirms acceptance of certificates requested in an initialization request or certificate request message.) This conveys a revocation passphrase chosen by the entity to the relevant CA/RA. When EnvelopedData is used, this is in the decrypted bytes of encryptedContent field. When EncryptedValue is used, this is in the decrypted bytes of the encValue field. Furthermore, the transfer is accomplished with appropriate confidentiality characteristics.

Replace the third bullet point of this section with the following text:

- * Either the localKeyId attribute of EnvelopedData as specified in RFC 2985 [RFC2985] or the valueHint field of EncryptedValue MAY contain a key identifier (chosen by the entity, along with the passphrase itself) to assist in later retrieval of the correct passphrase (e.g., when the revocation request is constructed by the entity and received by the CA/RA).

2.27. Update Appendix C - Request Message Behavioral Clarifications

Appendix C of RFC 4210 [RFC4210] provides clarifications to the request message behavior. As this document updates RFC 4210 [RFC4210] to utilize the parent structure EncryptedKey instead of EncryptedValue as described in Section 2.7 above, the description is updated accordingly.

Replace the comment within the ASN.1 syntax coming after the definition of POPOSigningKey with the following text (Note: This fixes Errata ID 2615):

```

-- *****
-- * For the purposes of this specification, the ASN.1 comment
-- * given in [RFC4211] pertains not only to certTemplate, but
-- * also to the altCertTemplate control.
-- *****
-- * The signature (using "algorithmIdentifier") is on the
-- * DER-encoded value of poposkInput (i.e., the "value" OCTETs
-- * of the POPOSigningKeyInput DER). NOTE: If CertReqMsg
-- * certReq certTemplate (or the altCertTemplate control)
-- * contains the subject and publicKey values, then poposkInput
-- * MUST be omitted and the signature MUST be computed on the
-- * DER-encoded value of CertReqMsg certReq (or the DER-
-- * encoded value of AltCertTemplate). If
-- * certTemplate/altCertTemplate does not contain both the
-- * subject and public key values (i.e., if it contains only
-- * one of these, or neither), then poposkInput MUST be present
-- * and MUST be signed.
-- *****

```

Replace the comment within the ASN.1 syntax coming after the definition of POPOPrivKey with the following text:

```

-- *****
-- * the type of "thisMessage" is given as BIT STRING in RFC 4211
-- * [RFC4211]; it should be "EncryptedKey" (in accordance with
-- * Section 5.2.2 of this specification). Therefore, this
-- * document makes the behavioral clarification of specifying
-- * that the contents of "thisMessage" MUST be encoded either as
-- * "EnvelopedData" or "EncryptedValue" (only for backward
-- * compatibility) and then wrapped in a BIT STRING. This
-- * allows the necessary conveyance and protection of the
-- * private key while maintaining bits-on-the-wire compatibility
-- * with RFC4210 and [RFCXXXX].
-- *****

```

2.28. Update Appendix D.1. - General Rules for Interpretation of These Profiles

Appendix D.1 of RFC 4210 [RFC4210] provides general rules for interpretation of the PKI management messages profiles specified in Appendix D and Appendix E of RFC 4210 [RFC4210]. This document updates a sentence regarding the new protocol version cmp2021.

Replace the last sentence of the first paragraph of the section with the following text:

Mandatory fields are not mentioned if they have an obvious value (e.g., in this version of these profiles, pvno is always cmp2000).

2.29. Update Appendix D.2. - Algorithm Use Profile

Appendix D.2 of RFC 4210 [RFC4210] provides a list of algorithms that implementations must support when claiming conformance with PKI Management Message Profiles as specified in CMP Appendix D.2 [RFC4210]. This document redirects to the new algorithm profile as specified in Section 7.1 of CMP Algorithms [I-D.ietf-lamps-cmp-algorithms].

Replace the text of the section with the following text:

D.2. Algorithm Use Profile

For specifications of algorithm identifiers and respective conventions for conforming implementations, please refer to CMP Algorithms Section 7.1 [I-D.ietf-lamps-cmp-algorithms].

2.30. Update Appendix D.4. - Initial Registration/Certification (Basic Authenticated Scheme)

Appendix D.4 of RFC 4210 [RFC4210] provides the initial registration/certification scheme. This scheme shall continue using EncryptedValue for backward compatibility reasons.

Replace the line specifying protectionAlg of the Initialization Response message with the following text (Note: This fixes Errata ID 5201):

```
protectionAlg          MSG_MAC_ALG
```

Replace the comment after the privateKey field of crc[1].certifiedKeyPair in the syntax of the Initialization Response message with the following text:

```
-- see Appendix C, Request Message Behavioral Clarifications  
-- for backward compatibility reasons, use EncryptedValue
```

3. Updates to RFC 6712 - HTTP Transfer for the Certificate Management Protocol (CMP)

3.1. Update Section 1. - Introduction

To indicate and explain why delayed delivery of all kinds of PKIMessages may be handled at transfer level and/or at CMP level, the introduction of RFC 6712 [RFC6712] is updated.

Replace the third paragraph of this section with the following text:

In addition to reliable transport, CMP requires connection and error handling from the transfer protocol, which is all covered by HTTP. Additionally, delayed delivery of CMP response messages may be handled at transfer level regardless of the message contents. Since this document extends the polling mechanism specified in the second version of CMP [RFC4210] to cover all types of PKI management transactions, delays detected at application level may also be handled within CMP, using pollReq and pollRep messages.

3.2. New Section 1.1. - Changes Since RFC 6712

The following subsection describes feature updates to RFC 6712 [RFC6712]. They are related to the base specification. Hence, references to the original sections in RFC 6712 [RFC6712] are used whenever possible.

Insert this section at the end of the current Section 1:

1.1 Changes Since RFC 6712

The following updates are made in this document:

- * Introduce the HTTP path `'/.well-known/cmp'`.
- * Extend the URI structure.

3.3. Replace Section 3.6. - HTTP Request-URI

Section 3.6 of RFC 6712 [RFC6712] specifies the used HTTP URIs. This document introduces the HTTP path `'/.well-known/cmp'` and extends the URIs.

Replace the text of the section with the following text:

3.6. HTTP Request-URI

Each CMP server on a PKI management entity supporting HTTP or HTTPS transfer MUST support the use of the path prefix `'/.well-known/'` as defined in RFC 8615 [RFC8615] and the registered name `'cmp'` to ease interworking in a multi-vendor environment.

The CMP client needs to be configured with sufficient information to form the CMP server URI. This is at least the authority portion of the URI, e.g., `'www.example.com:80'`, or the full operation path segment of the PKI management entity. Additionally, OPTIONAL path segments MAY be added after the registered application name as part of the full operation path to provide further distinction. The path segment `'p'` followed by an arbitraryLabel `<name>` could for example

support the differentiation of specific CAs or certificate profiles. Further path segments, e.g., as specified in the Lightweight CMP Profile [I-D.ietf-lamps-lightweight-cmp-profile], could indicate PKI management operations using an operationLabel <operation>. A valid full CMP URI can look like this:

```
http://www.example.com/.well-known/cmp
http://www.example.com/.well-known/cmp/<operation>
http://www.example.com/.well-known/cmp/p/<name>
http://www.example.com/.well-known/cmp/p/<name>/<operation>
```

4. IANA Considerations

This document updates the ASN.1 modules of RFC 4210 Appendix F [RFC4210] and RFC 5912 Section 9 [RFC5912]. The OIDs 99 (id-mod-cmp2021-88) and 100 (id-mod-cmp2021-02) were registered in the SMI Security for PKIX Module Identifier registry to identify the updated ASN.1 modules.

This document contains an update to the IANA Consideration sections of [RFC4210] adding this content.

In the SMI-numbers registry "SMI Security for PKIX Extended Key Purpose Identifiers (1.3.6.1.5.5.7.3)" (see <https://www.iana.org/assignments/smi-numbers/smi-numbers.xhtml#smi-numbers-1.3.6.1.5.5.7.3>) as defined in RFC 7299 [RFC7299] one addition has been performed.

One new entry has been added:

Decimal	Description	References
32	id-kp-cmKGA	[RFCXXXX]

Table 1: Addition to the PKIX Extended Key Purpose Identifiers Registry

In the SMI-numbers registry "SMI Security for PKIX CMP Information Types (1.3.6.1.5.5.7.4)" (see <https://www.iana.org/assignments/smi-numbers/smi-numbers.xhtml#smi-numbers-1.3.6.1.5.5.7.4>) as defined in RFC 7299 [RFC7299] seven additions have been performed.

Seven new entries have been added:

```
+=====+=====+=====+
```

Decimal	Description	References
17	id-it-caCerts	[RFCXXXX]
18	id-it-rootCaKeyUpdate	[RFCXXXX]
19	id-it-certReqTemplate	[RFCXXXX]
20	id-it-rootCaCert	[RFCXXXX]
21	id-it-certProfile	[RFCXXXX]
22	id-it-crlStatusList	[RFCXXXX]
23	id-it-crls	[RFCXXXX]

Table 2: Addition to the PKIX CMP
Information Types Registry

In the SMI-numbers registry "SMI Security for PKIX CRMF Registration Controls (1.3.6.1.5.5.7.5.1)" (see <https://www.iana.org/assignments/smi-numbers/smi-numbers.xhtml#smi-numbers-1.3.6.1.5.5.7.5.1>) as defined in RFC 7299 [RFC7299] two additions have been performed.

Two new entries have been added:

Decimal	Description	References
11	id-regCtrl-algId	[RFCXXXX]
12	id-regCtrl-rsaKeyLen	[RFCXXXX]

Table 3: Addition to the PKIX CRMF
Registration Controls Registry

This document contains an update to the IANA Consideration sections of [RFC6712] adding this content.

This document defines a new entry with the following content in the "Well-Known URIs" registry (see <https://www.iana.org/assignments/well-known-uris/>) as defined in RFC 8615 [RFC8615].

URI Suffix: cmp
Change Controller: IETF
References: [RFCXXXX] [I-D.ietf-ace-cmpv2-coap-transport]

Related Information: CMP has a sub-registry at
[<https://www.iana.org/assignments/cmp/>]

This document defines a new protocol registry group entitled "Certificate Management Protocol (CMP)" (at <https://www.iana.org/assignments/cmp/>) with a new registry "CMP Well-Known URI Path Segments" containing three columns: Path Segment, Description, and Reference. New items can be added using the Specification Required RFC 8615 [RFC8615] process. The initial contents of this registry is:

Path Segment: p
Description: Indicates that the next path segment specifies, e.g., a CA or certificate profile name
References: [RFCXXXX] [I-D.ietf-ace-cmpv2-coap-transport]

5. Security Considerations

The security considerations of RFC 4210 [RFC4210] are extended in Section 2.22 to Section 2.24. No security considerations updates of RFC 6712 [RFC6712] were required.

6. Acknowledgements

Special thank goes to Jim Schaad for his guidance and the inspiration on structuring and writing this document we got from [RFC6402] which updates CMC. Special thank also goes to Russ Housley, Lijun Liao, Martin Peylo, and Tomas Gustavsson for reviewing and providing valuable suggestions on improving this document.

We also thank all reviewers of this document for their valuable feedback.

7. References

7.1. Normative References

[I-D.ietf-ace-cmpv2-coap-transport]
Sahni, M. and S. Tripathi, "CoAP Transfer for the Certificate Management Protocol", Work in Progress, Internet-Draft, draft-ietf-ace-cmpv2-coap-transport-04, 8 November 2021, <<https://datatracker.ietf.org/doc/html/draft-ietf-ace-cmpv2-coap-transport-04>>.

[I-D.ietf-lamps-cmp-algorithms]
Brockhaus, H., Aschauer, H., Ounsworth, M., and J. Gray, "Certificate Management Protocol (CMP) Algorithms", Work in Progress, Internet-Draft, draft-ietf-lamps-cmp-

- algorithms-15, 2 June 2022,
<<https://datatracker.ietf.org/doc/html/draft-ietf-lamps-cmp-algorithms-15>>.
- [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>>.
- [RFC2510] Adams, C. and S. Farrell, "Internet X.509 Public Key Infrastructure Certificate Management Protocols", RFC 2510, DOI 10.17487/RFC2510, March 1999, <<https://www.rfc-editor.org/info/rfc2510>>.
- [RFC2985] Nystrom, M. and B. Kaliski, "PKCS #9: Selected Object Classes and Attribute Types Version 2.0", RFC 2985, DOI 10.17487/RFC2985, November 2000, <<https://www.rfc-editor.org/info/rfc2985>>.
- [RFC2986] Nystrom, M. and B. Kaliski, "PKCS #10: Certification Request Syntax Specification Version 1.7", RFC 2986, DOI 10.17487/RFC2986, November 2000, <<https://www.rfc-editor.org/info/rfc2986>>.
- [RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, DOI 10.17487/RFC3629, November 2003, <<https://www.rfc-editor.org/info/rfc3629>>.
- [RFC4210] Adams, C., Farrell, S., Kaese, T., and T. Mononen, "Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP)", RFC 4210, DOI 10.17487/RFC4210, September 2005, <<https://www.rfc-editor.org/info/rfc4210>>.
- [RFC4211] Schaad, J., "Internet X.509 Public Key Infrastructure Certificate Request Message Format (CRMF)", RFC 4211, DOI 10.17487/RFC4211, September 2005, <<https://www.rfc-editor.org/info/rfc4211>>.
- [RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <<https://www.rfc-editor.org/info/rfc5280>>.

- [RFC5480] Turner, S., Brown, D., Yiu, K., Housley, R., and T. Polk, "Elliptic Curve Cryptography Subject Public Key Information", RFC 5480, DOI 10.17487/RFC5480, March 2009, <<https://www.rfc-editor.org/info/rfc5480>>.
- [RFC5652] Housley, R., "Cryptographic Message Syntax (CMS)", STD 70, RFC 5652, DOI 10.17487/RFC5652, September 2009, <<https://www.rfc-editor.org/info/rfc5652>>.
- [RFC5958] Turner, S., "Asymmetric Key Packages", RFC 5958, DOI 10.17487/RFC5958, August 2010, <<https://www.rfc-editor.org/info/rfc5958>>.
- [RFC6402] Schaad, J., "Certificate Management over CMS (CMC) Updates", RFC 6402, DOI 10.17487/RFC6402, November 2011, <<https://www.rfc-editor.org/info/rfc6402>>.
- [RFC6712] Kause, T. and M. Peylo, "Internet X.509 Public Key Infrastructure -- HTTP Transfer for the Certificate Management Protocol (CMP)", RFC 6712, DOI 10.17487/RFC6712, September 2012, <<https://www.rfc-editor.org/info/rfc6712>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8615] Nottingham, M., "Well-Known Uniform Resource Identifiers (URIs)", RFC 8615, DOI 10.17487/RFC8615, May 2019, <<https://www.rfc-editor.org/info/rfc8615>>.
- [RFC8933] Housley, R., "Update to the Cryptographic Message Syntax (CMS) for Algorithm Identifier Protection", RFC 8933, DOI 10.17487/RFC8933, October 2020, <<https://www.rfc-editor.org/info/rfc8933>>.
- [RFC9045] Housley, R., "Algorithm Requirements Update to the Internet X.509 Public Key Infrastructure Certificate Request Message Format (CRMF)", RFC 9045, DOI 10.17487/RFC9045, June 2021, <<https://www.rfc-editor.org/info/rfc9045>>.

7.2. Informative References

- [AIS31] Bundesamt fuer Sicherheit in der Informationstechnik (BSI), Killmann, W., and W. Schindler, "A proposal for: Functionality classes for random number generators, version 2.0", 18 September 2011,

<https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/Interpretationen/AIS_31_Functionality_classes_for_random_number_generators_e.pdf>.

[CVE-2008-0166]

National Institute of Science and Technology (NIST),
"National Vulnerability Database - CVE-2008-0166", 13 May 2008, <<https://nvd.nist.gov/vuln/detail/CVE-2008-0166>>.

[I-D.ietf-lamps-lightweight-cmp-profile]

Brockhaus, H., Oheimb, D. V., and S. Fries, "Lightweight Certificate Management Protocol (CMP) Profile", Work in Progress, Internet-Draft, draft-ietf-lamps-lightweight-cmp-profile-12, 13 May 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-lamps-lightweight-cmp-profile-12>>.

[IEEE.802.1AR_2018]

IEEE, "IEEE Standard for Local and metropolitan area networks - Secure Device Identity", IEEE 802.1AR-2018, DOI 10.1109/IEEESTD.2018.8423794, 2 August 2018, <<https://ieeexplore.ieee.org/document/8423794>>.

[ISO.20543-2019]

International Organization for Standardization (ISO),
"Information technology -- Security techniques -- Test and analysis methods for random bit generators within ISO/IEC 19790 and ISO/IEC 15408", ISO Draft Standard 20543-2019, October 2019.

[MiningPsQs]

Security'12: Proceedings of the 21st USENIX conference on Security symposium, Heninger, N., Durumeric, Z., Wustrow, E., and J. A. Halderman, "Mining Your Ps and Qs: Detection of Widespread Weak Keys in Network Devices", August 2012, <<https://www.usenix.org/conference/usenixsecurity12/technical-sessions/presentation/heninger>>.

[NIST.SP.800-90Ar1]

Barker, Elaine B. and John M. Kelsey, "Recommendation for Random Number Generation Using Deterministic Random Bit Generators", NIST NIST SP 800-90Ar1, DOI 10.6028/NIST.SP.800-90Ar1, June 2015, <<https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90Ar1.pdf>>.

- [PKCS11] RSA Laboratories, "The Public-Key Cryptography Standards - Cryptographic Token Interface Standard. Version 2.10", December 1999, <<https://www.cryptsoft.com/pkcs11doc/STANDARD/pkcs11v2-10.pdf>>.
- [RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, DOI 10.17487/RFC2104, February 1997, <<https://www.rfc-editor.org/info/rfc2104>>.
- [RFC2202] Cheng, P. and R. Glenn, "Test Cases for HMAC-MD5 and HMAC-SHA-1", RFC 2202, DOI 10.17487/RFC2202, September 1997, <<https://www.rfc-editor.org/info/rfc2202>>.
- [RFC5912] Hoffman, P. and J. Schaad, "New ASN.1 Modules for the Public Key Infrastructure Using X.509 (PKIX)", RFC 5912, DOI 10.17487/RFC5912, June 2010, <<https://www.rfc-editor.org/info/rfc5912>>.
- [RFC7299] Housley, R., "Object Identifier Registry for the PKIX Working Group", RFC 7299, DOI 10.17487/RFC7299, July 2014, <<https://www.rfc-editor.org/info/rfc7299>>.

Appendix A. ASN.1 Modules

A.1. Update to RFC4210 - 1988 ASN.1 Module

This section contains the updated ASN.1 module for [RFC4210]. This module replaces the module in Appendix F of that document. Although a 2002 ASN.1 module is provided, this 1988 ASN.1 module remains the normative module as per the policy of the PKIX working group.

```
PKIXCMP {iso(1) identified-organization(3)
  dod(6) internet(1) security(5) mechanisms(5) pkix(7)
  id-mod(0) id-mod-cmp2021-88(99)}

DEFINITIONS EXPLICIT TAGS ::=
BEGIN

-- EXPORTS ALL --

IMPORTS

  Certificate, CertificateList, Extensions, Name, Time,
  AlgorithmIdentifier, id-kp
  --, UTF8String -- -- if required; otherwise, comment out
```

```
        FROM PKIX1Explicit88 {iso(1) identified-organization(3)
        dod(6) internet(1) security(5) mechanisms(5) pkix(7)
        id-mod(0) id-pkix1-explicit-88(18)}
-- The import of Name is added to define CertificationRequest
-- instead of importing it from PKCS#10 [RFC2986]

DistributionPointName, GeneralNames, GeneralName, KeyIdentifier
        FROM PKIX1Implicit88 {iso(1) identified-organization(3)
        dod(6) internet(1) security(5) mechanisms(5) pkix(7)
        id-mod(0) id-pkix1-implicit-88(19)}

CertTemplate, PKIPublicationInfo, EncryptedKey, CertId,
CertReqMessages, Controls, AttributeTypeAndValue, id-regCtrl
        FROM PKIXCRMF-2005 {iso(1) identified-organization(3)
        dod(6) internet(1) security(5) mechanisms(5) pkix(7)
        id-mod(0) id-mod-crmf2005(36)}
-- The import of EncryptedKey is added due to the updates made
-- in CMP Updates [RFCXXXX]. EncryptedValue does not need to
-- be imported anymore and is therefore removed here.

-- see also the behavioral clarifications to CRMF codified in
-- Appendix C of this specification

EnvelopedData, SignedData, Attribute
        FROM CryptographicMessageSyntax2004 { iso(1)
        member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-9(9)
        smime(16) modules(0) cms-2004(24) }
-- The import of EnvelopedData and SignedData is added due to
-- the updates made in CMP Updates [RFCXXXX]
-- The import of Attribute is added to define
-- CertificationRequest instead of importing it from
-- PKCS#10 [RFC2986]

;

-- the rest of the module contains locally-defined OIDs and
-- constructs

CMPCertificate ::= CHOICE {
    x509v3PKCert      Certificate
}
-- This syntax, while bits-on-the-wire compatible with the
-- standard X.509 definition of "Certificate", allows the
-- possibility of future certificate types (such as X.509
-- attribute certificates, WAP WTLS certificates, or other kinds
-- of certificates) within this certificate management protocol,
-- should a need ever arise to support such generality. Those
-- implementations that do not foresee a need to ever support
```

```
-- other certificate types MAY, if they wish, comment out the
-- above structure and "un-comment" the following one prior to
-- compiling this ASN.1 module. (Note that interoperability
-- with implementations that don't do this will be unaffected by
-- this change.)

-- CMPCertificate ::= Certificate

PKIMessage ::= SEQUENCE {
    header          PKIHeader,
    body            PKIBody,
    protection      [0] PKIProtection OPTIONAL,
    extraCerts      [1] SEQUENCE SIZE (1..MAX) OF CMPCertificate
                    OPTIONAL
}

PKIMessages ::= SEQUENCE SIZE (1..MAX) OF PKIMessage

PKIHeader ::= SEQUENCE {
    pvno            INTEGER          { cmp1999(1), cmp2000(2),
                                    cmp2021(3) },
    sender          GeneralName,
    -- identifies the sender
    recipient       GeneralName,
    -- identifies the intended recipient
    messageTime     [0] GeneralizedTime      OPTIONAL,
    -- time of production of this message (used when sender
    -- believes that the transport will be "suitable"; i.e.,
    -- that the time will still be meaningful upon receipt)
    protectionAlg   [1] AlgorithmIdentifier  OPTIONAL,
    -- algorithm used for calculation of protection bits
    senderKID       [2] KeyIdentifier        OPTIONAL,
    recipKID        [3] KeyIdentifier        OPTIONAL,
    -- to identify specific keys used for protection
    transactionID   [4] OCTET STRING        OPTIONAL,
    -- identifies the transaction; i.e., this will be the same in
    -- corresponding request, response, certConf, and PKIConf
    -- messages
    senderNonce     [5] OCTET STRING        OPTIONAL,
    recipNonce      [6] OCTET STRING        OPTIONAL,
    -- nonces used to provide replay protection, senderNonce
    -- is inserted by the creator of this message; recipNonce
    -- is a nonce previously inserted in a related message by
    -- the intended recipient of this message
    freeText        [7] PKIFreeText         OPTIONAL,
    -- this may be used to indicate context-specific instructions
    -- (this field is intended for human consumption)
    generalInfo     [8] SEQUENCE SIZE (1..MAX) OF
```

```

                                InfoTypeAndValue      OPTIONAL
    -- this may be used to convey context-specific information
    -- (this field not primarily intended for human consumption)
}

PKIFreeText ::= SEQUENCE SIZE (1..MAX) OF UTF8String
    -- text encoded as UTF-8 String [RFC3629]

PKIBody ::= CHOICE {
    -- message-specific body elements
    ir      [0] CertReqMessages,      --Initialization Request
    ip      [1] CertRepMessage,      --Initialization Response
    cr      [2] CertReqMessages,      --Certification Request
    cp      [3] CertRepMessage,      --Certification Response
    p10cr   [4] CertificationRequest, --imported from [RFC2986]
    popdecc [5] POPODecKeyChallContent, --pop Challenge
    popdecr [6] POPODecKeyRespContent, --pop Response
    kur     [7] CertReqMessages,      --Key Update Request
    kup     [8] CertRepMessage,      --Key Update Response
    krr     [9] CertReqMessages,      --Key Recovery Request
    krp     [10] KeyRecRepContent,     --Key Recovery Response
    rr      [11] RevReqContent,       --Revocation Request
    rp      [12] RevRepContent,       --Revocation Response
    ccr     [13] CertReqMessages,     --Cross-Cert. Request
    ccp     [14] CertRepMessage,      --Cross-Cert. Response
    ckuann  [15] CAKeyUpdAnnContent,   --CA Key Update Ann.
    cann    [16] CertAnnContent,      --Certificate Ann.
    rann    [17] RevAnnContent,       --Revocation Ann.
    crlann  [18] CRLAnnContent,       --CRL Announcement
    pkiconf [19] PKIConfirmContent,    --Confirmation
    nested  [20] NestedMessageContent, --Nested Message
    genm    [21] GenMsgContent,       --General Message
    genp    [22] GenRepContent,       --General Response
    error   [23] ErrorMsgContent,     --Error Message
    certConf [24] CertConfirmContent, --Certificate confirm
    pollReq [25] PollReqContent,      --Polling request
    pollRep [26] PollRepContent,      --Polling response
}

PKIProtection ::= BIT STRING

ProtectedPart ::= SEQUENCE {
    header    PKIHeader,
    body      PKIBody
}

id-PasswordBasedMac OBJECT IDENTIFIER ::= {1 2 840 113533 7 66 13}
PBMPParameter ::= SEQUENCE {
    salt      OCTET STRING,

```

```
-- note: implementations MAY wish to limit acceptable sizes
-- of this string to values appropriate for their environment
-- in order to reduce the risk of denial-of-service attacks
owf          AlgorithmIdentifier,
-- AlgId for a One-Way Function
iterationCount  INTEGER,
-- number of times the OWF is applied
-- note: implementations MAY wish to limit acceptable sizes
-- of this integer to values appropriate for their environment
-- in order to reduce the risk of denial-of-service attacks
mac          AlgorithmIdentifier
-- the MAC AlgId (e.g., DES-MAC, Triple-DES-MAC [PKCS11],
} -- or HMAC [RFC2104, RFC2202])

id-DHBasedMac OBJECT IDENTIFIER ::= {1 2 840 113533 7 66 30}
DHBMParameter ::= SEQUENCE {
  owf          AlgorithmIdentifier,
  -- AlgId for a One-Way Function
  mac          AlgorithmIdentifier
  -- the MAC AlgId (e.g., DES-MAC, Triple-DES-MAC [PKCS11],
} -- or HMAC [RFC2104, RFC2202])

NestedMessageContent ::= PKIMessages

PKIStatus ::= INTEGER {
  accepted          (0),
  -- you got exactly what you asked for
  grantedWithMods  (1),
  -- you got something like what you asked for; the
  -- requester is responsible for ascertaining the differences
  rejection        (2),
  -- you don't get it, more information elsewhere in the message
  waiting          (3),
  -- the request body part has not yet been processed; expect to
  -- hear more later (note: proper handling of this status
  -- response MAY use the polling req/rep PKIMessages specified
  -- in Section 5.3.22 of [RFC4210]; alternatively, polling in the
  -- underlying transport layer MAY have some utility in this
  -- regard)
  revocationWarning (4),
  -- this message contains a warning that a revocation is
  -- imminent
  revocationNotification (5),
  -- notification that a revocation has occurred
  keyUpdateWarning  (6)
  -- update already done for the oldCertId specified in
  -- CertReqMsg
```

```
}

PKIFailureInfo ::= BIT STRING {
-- since we can fail in more than one way!
-- More codes may be added in the future if/when required.
  badAlg          (0),
  -- unrecognized or unsupported Algorithm Identifier
  badMessageCheck (1),
  -- integrity check failed (e.g., signature did not verify)
  badRequest      (2),
  -- transaction not permitted or supported
  badTime         (3),
  -- messageTime was not sufficiently close to the system time,
  -- as defined by local policy
  badCertId       (4),
  -- no certificate could be found matching the provided criteria
  badDataFormat   (5),
  -- the data submitted has the wrong format
  wrongAuthority  (6),
  -- the authority indicated in the request is different from the
  -- one creating the response token
  incorrectData   (7),
  -- the requester's data is incorrect (for notary services)
  missingTimeStamp (8),
  -- when the timestamp is missing but should be there
  -- (by policy)
  badPOP          (9),
  -- the proof-of-possession failed
  certRevoked     (10),
  -- the certificate has already been revoked
  certConfirmed   (11),
  -- the certificate has already been confirmed
  wrongIntegrity  (12),
  -- not valid integrity, password based instead of signature or
  -- vice versa
  badRecipientNonce (13),
  -- not valid recipient nonce, either missing or wrong value
  timeNotAvailable (14),
  -- the TSA's time source is not available
  unacceptedPolicy (15),
  -- the requested TSA policy is not supported by the TSA.
  unacceptedExtension (16),
  -- the requested extension is not supported by the TSA.
  addInfoNotAvailable (17),
  -- the additional information requested could not be
  -- understood or is not available
  badSenderNonce  (18),
  -- not valid sender nonce, either missing or wrong size
```

```
badCertTemplate      (19),
  -- not valid cert. template or missing mandatory information
signerNotTrusted    (20),
  -- signer of the message unknown or not trusted
transactionIdInUse  (21),
  -- the transaction identifier is already in use
unsupportedVersion   (22),
  -- the version of the message is not supported
notAuthorized       (23),
  -- the sender was not authorized to make the preceding
  -- request or perform the preceding action
systemUnavail       (24),
  -- the request cannot be handled due to system unavailability
systemFailure       (25),
  -- the request cannot be handled due to system failure
duplicateCertReq    (26)
  -- certificate cannot be issued because a duplicate
  -- certificate already exists
}

PKIStatusInfo ::= SEQUENCE {
  status          PKIStatus,
  statusString    PKIFreeText    OPTIONAL,
  failInfo        PKIFailureInfo OPTIONAL
}

OOBCert ::= CMPCertificate

OOBCertHash ::= SEQUENCE {
  hashAlg    [0] AlgorithmIdentifier    OPTIONAL,
  certId     [1] CertId                  OPTIONAL,
  hashVal    BIT STRING
  -- hashVal is calculated over the DER encoding of the
  -- self-signed certificate with the identifier certID.
}

POPODecKeyChallContent ::= SEQUENCE OF Challenge
-- One Challenge per encryption key certification request (in the
-- same order as these requests appear in CertReqMessages).

Challenge ::= SEQUENCE {
  owf          AlgorithmIdentifier    OPTIONAL,
  -- MUST be present in the first Challenge; MAY be omitted in
  -- any subsequent Challenge in POPODecKeyChallContent (if
  -- omitted, then the owf used in the immediately preceding
  -- Challenge is to be used).
  witness      OCTET STRING,
  -- the result of applying the one-way function (owf) to a
```

```

-- randomly-generated INTEGER, A. [Note that a different
-- INTEGER MUST be used for each Challenge.]
challenge          OCTET STRING
-- the encryption (under the public key for which the cert.
-- request is being made) of Rand.
}

-- Added in CMP Updates [RFCXXXX]

Rand ::= SEQUENCE {
-- Rand is encrypted under the public key to form the challenge
-- in POPODecKeyChallContent
  int              INTEGER,
  -- the randomly-generated INTEGER A (above)
  sender           GeneralName
  -- the sender's name (as included in PKIHeader)
}

POPODecKeyRespContent ::= SEQUENCE OF INTEGER
-- One INTEGER per encryption key certification request (in the
-- same order as these requests appear in CertReqMessages). The
-- retrieved INTEGER A (above) is returned to the sender of the
-- corresponding Challenge.

CertRepMessage ::= SEQUENCE {
  caPubs           [1] SEQUENCE SIZE (1..MAX) OF CMPCertificate
                  OPTIONAL,
  response         SEQUENCE OF CertResponse
}

CertificationRequest ::= SEQUENCE {
  certificationRequestInfo SEQUENCE {
    version          INTEGER,
    subject          Name,
    subjectPublicKeyInfo SEQUENCE {
      algorithm      AlgorithmIdentifier,
      subjectPublicKey BIT STRING },
    attributes       [0] IMPLICIT SET OF Attribute },
  signatureAlgorithm AlgorithmIdentifier,
  signature          BIT STRING
}

CertResponse ::= SEQUENCE {
  certReqId        INTEGER,
  -- to match this response with corresponding request (a value
  -- of -1 is to be used if certReqId is not specified in the
  -- corresponding request, which can only be a p10cr)
  status           PKIStatusInfo,

```

```
certifiedKeyPair    CertifiedKeyPair    OPTIONAL,
rspInfo             OCTET STRING        OPTIONAL
-- analogous to the id-regInfo-utf8Pairs string defined
-- for regInfo in CertReqMsg [RFC4211]
}

CertifiedKeyPair ::= SEQUENCE {
  certOrEncCert      CertOrEncCert,
  privateKey         [0] EncryptedKey    OPTIONAL,
  -- see [RFC4211] for comment on encoding
  -- Changed from Encrypted Value to EncryptedKey as a CHOICE of
  -- EncryptedValue and EnvelopedData due to the changes made in
  -- CMP Updates [RFCXXXX]
  -- Using the choice EncryptedValue is bit-compatible to the
  -- syntax without this change
  publicationInfo   [1] PKIPublicationInfo OPTIONAL
}

CertOrEncCert ::= CHOICE {
  certificate        [0] CMPCertificate,
  encryptedCert     [1] EncryptedKey
  -- Changed from Encrypted Value to EncryptedKey as a CHOICE of
  -- EncryptedValue and EnvelopedData due to the changes made in
  -- CMP Updates [RFCXXXX]
  -- Using the choice EncryptedValue is bit-compatible to the
  -- syntax without this change
}

KeyRecRepContent ::= SEQUENCE {
  status             PKIStatusInfo,
  newSigCert         [0] CMPCertificate OPTIONAL,
  caCerts            [1] SEQUENCE SIZE (1..MAX) OF
                    CMPCertificate OPTIONAL,
  keyPairHist       [2] SEQUENCE SIZE (1..MAX) OF
                    CertifiedKeyPair OPTIONAL
}

RevReqContent ::= SEQUENCE OF RevDetails

RevDetails ::= SEQUENCE {
  certDetails        CertTemplate,
  -- allows requester to specify as much as they can about
  -- the cert. for which revocation is requested
  -- (e.g., for cases in which serialNumber is not available)
  crlEntryDetails   Extensions        OPTIONAL
  -- requested crlEntryExtensions
}
```

```
RevRepContent ::= SEQUENCE {
    status          SEQUENCE SIZE (1..MAX) OF PKIStatusInfo,
    -- in same order as was sent in RevReqContent
    revCerts [0] SEQUENCE SIZE (1..MAX) OF CertId
                    OPTIONAL,
    -- IDs for which revocation was requested
    -- (same order as status)
    crls           [1] SEQUENCE SIZE (1..MAX) OF CertificateList
                    OPTIONAL
    -- the resulting CRLs (there may be more than one)
}

CAKeyUpdAnnContent ::= SEQUENCE {
    oldWithNew    CMPCertificate, -- old pub signed with new priv
    newWithOld    CMPCertificate, -- new pub signed with old priv
    newWithNew    CMPCertificate -- new pub signed with new priv
}

CertAnnContent ::= CMPCertificate

RevAnnContent ::= SEQUENCE {
    status          PKIStatus,
    certId          CertId,
    willBeRevokedAt GeneralizedTime,
    badSinceDate    GeneralizedTime,
    crlDetails      Extensions OPTIONAL
    -- extra CRL details (e.g., crl number, reason, location, etc.)
}

CRLAnnContent ::= SEQUENCE OF CertificateList

CertConfirmContent ::= SEQUENCE OF CertStatus

CertStatus ::= SEQUENCE {
    certHash      OCTET STRING,
    -- the hash of the certificate, using the same hash algorithm
    -- as is used to create and verify the certificate signature
    certReqId     INTEGER,
    -- to match this confirmation with the corresponding req/rep
    statusInfo    PKIStatusInfo OPTIONAL,
    hashAlg [0] AlgorithmIdentifier OPTIONAL
    -- the hash algorithm to use for calculating certHash
    -- SHOULD NOT be used in all cases where the AlgorithmIdentifier
    -- of the certificate signature specifies a hash algorithm
}

PKIConfirmContent ::= NULL
```

```
-- CertReqTemplateContent, id-regCtrl-algId, id-regCtrl-algId, and
-- id-regCtrl-rsaKeyLen were added in CMP Updates [RFCXXXX]
```

```
CertReqTemplateContent ::= SEQUENCE {
    certTemplate          CertTemplate,
    -- prefilled certTemplate structure elements
    -- The SubjectPublicKeyInfo field in the certTemplate MUST NOT
    -- be used.
    keySpec              Controls OPTIONAL
    -- MAY be used to specify supported algorithms.
    -- Controls ::= SEQUENCE SIZE (1..MAX) OF AttributeTypeAndValue
    -- as specified in CRMF (RFC4211)
}
```

```
id-regCtrl-altCertTemplate OBJECT IDENTIFIER ::= { id-regCtrl 7 }
AltCertTemplate ::= AttributeTypeAndValue
-- specifies a template for a certificate other than an X.509v3
-- public-key certificate
```

```
id-regCtrl-algId OBJECT IDENTIFIER ::= { id-regCtrl 11 }
AlgIdCtrl ::= AlgorithmIdentifier
-- SHALL be used to specify supported algorithms other than RSA
```

```
id-regCtrl-rsaKeyLen OBJECT IDENTIFIER ::= { id-regCtrl 12 }
RsaKeyLenCtrl ::= INTEGER (1..MAX)
-- SHALL be used to specify supported RSA key lengths
```

```
-- RootCaKeyUpdateContent, CRLSource, and CRLStatus were added in
-- CMP Updates [RFCXXXX]
```

```
RootCaKeyUpdateContent ::= SEQUENCE {
    newWithNew          CMPCertificate,
    -- new root CA certificate
    newWithOld          [0] CMPCertificate OPTIONAL,
    -- X.509 certificate containing the new public root CA key
    -- signed with the old private root CA key
    oldWithNew          [1] CMPCertificate OPTIONAL
    -- X.509 certificate containing the old public root CA key
    -- signed with the new private root CA key
}
```

```
CRLSource ::= CHOICE {
    dpn                [0] DistributionPointName,
    issuer              [1] GeneralNames }
```

```
CRLStatus ::= SEQUENCE {
    source              CRLSource,
    thisUpdate          Time OPTIONAL }
```

```

InfoTypeAndValue ::= SEQUENCE {
    infoType          OBJECT IDENTIFIER,
    infoValue         ANY DEFINED BY infoType OPTIONAL
}
-- Example InfoTypeAndValue contents include, but are not limited
-- to, the following (un-comment in this ASN.1 module and use as
-- appropriate for a given environment):
--
-- id-it-caProtEncCert      OBJECT IDENTIFIER ::= {id-it 1}
--   CAProtEncCertValue    ::= CMPCertificate
-- id-it-signKeyPairTypes  OBJECT IDENTIFIER ::= {id-it 2}
--   SignKeyPairTypesValue ::= SEQUENCE SIZE (1..MAX) OF
--                               AlgorithmIdentifier
-- id-it-encKeyPairTypes   OBJECT IDENTIFIER ::= {id-it 3}
--   EncKeyPairTypesValue  ::= SEQUENCE SIZE (1..MAX) OF
--                               AlgorithmIdentifier
-- id-it-preferredSymmAlg  OBJECT IDENTIFIER ::= {id-it 4}
--   PreferredSymmAlgValue ::= AlgorithmIdentifier
-- id-it-caKeyUpdateInfo   OBJECT IDENTIFIER ::= {id-it 5}
--   CAKeyUpdateInfoValue  ::= CAKeyUpdAnnContent
-- id-it-currentCRL        OBJECT IDENTIFIER ::= {id-it 6}
--   CurrentCRLValue       ::= CertificateList
-- id-it-unsupportedOIDs   OBJECT IDENTIFIER ::= {id-it 7}
--   UnsupportedOIDsValue  ::= SEQUENCE SIZE (1..MAX) OF
--                               OBJECT IDENTIFIER
-- id-it-keyPairParamReq   OBJECT IDENTIFIER ::= {id-it 10}
--   KeyPairParamReqValue  ::= OBJECT IDENTIFIER
-- id-it-keyPairParamRep   OBJECT IDENTIFIER ::= {id-it 11}
--   KeyPairParamRepValue  ::= AlgorithmIdentifier
-- id-it-revPassphrase     OBJECT IDENTIFIER ::= {id-it 12}
--   RevPassphraseValue    ::= EncryptedKey
--   - Changed from Encrypted Value to EncryptedKey as a CHOICE
--   - of EncryptedValue and EnvelopedData due to the changes
--   - made in CMP Updates [RFCXXXX]
--   - Using the choice EncryptedValue is bit-compatible to the
--   - syntax without this change
-- id-it-implicitConfirm  OBJECT IDENTIFIER ::= {id-it 13}
--   ImplicitConfirmValue  ::= NULL
-- id-it-confirmWaitTime  OBJECT IDENTIFIER ::= {id-it 14}
--   ConfirmWaitTimeValue  ::= GeneralizedTime
-- id-it-origPKIMessage    OBJECT IDENTIFIER ::= {id-it 15}
--   OrigPKIMessageValue   ::= PKIMessages
-- id-it-supplLangTags     OBJECT IDENTIFIER ::= {id-it 16}
--   SupplLangTagsValue    ::= SEQUENCE OF UTF8String
-- id-it-caCerts           OBJECT IDENTIFIER ::= {id-it 17}
--   CaCertsValue          ::= SEQUENCE SIZE (1..MAX) OF
--                               CMPCertificate
--   - id-it-caCerts added in CMP Updates [RFCXXXX]

```

```
-- id-it-rootCaKeyUpdate OBJECT IDENTIFIER ::= {id-it 18}
--   RootCaKeyUpdateValue ::= RootCaKeyUpdateContent
--   - id-it-rootCaKeyUpdate added in CMP Updates [RFCXXXX]
-- id-it-certReqTemplate OBJECT IDENTIFIER ::= {id-it 19}
--   CertReqTemplateValue ::= CertReqTemplateContent
--   - id-it-certReqTemplate added in CMP Updates [RFCXXXX]
-- id-it-rootCaCert OBJECT IDENTIFIER ::= {id-it 20}
--   RootCaCertValue ::= CMPCertificate
--   - id-it-rootCaCert added in CMP Updates [RFCXXXX]
-- id-it-certProfile OBJECT IDENTIFIER ::= {id-it 21}
--   CertProfileValue ::= SEQUENCE SIZE (1..MAX) OF
--                       UTF8String
--   - id-it-certProfile added in CMP Updates [RFCXXXX]
-- id-it-crlStatusList OBJECT IDENTIFIER ::= {id-it 22}
--   CRLStatusListValue ::= SEQUENCE SIZE (1..MAX) OF
--                           CRLStatus
--   - id-it-crlStatusList added in CMP Updates [RFCXXXX]
-- id-it-crls OBJECT IDENTIFIER ::= {id-it 23}
--   CRLsValue ::= SEQUENCE SIZE (1..MAX) OF
--                   CertificateList
--   - id-it-crls added in CMP Updates [RFCXXXX]
--
-- where
--
--   id-pkix OBJECT IDENTIFIER ::= {
--     iso(1) identified-organization(3)
--     dod(6) internet(1) security(5) mechanisms(5) pkix(7) }
-- and
--   id-it OBJECT IDENTIFIER ::= {id-pkix 4}
--
-- This construct MAY also be used to define new PKIX Certificate
-- Management Protocol request and response messages, or general-
-- purpose (e.g., announcement) messages for future needs or for
-- specific environments.
GenMsgContent ::= SEQUENCE OF InfoTypeAndValue
--
-- May be sent by EE, RA, or CA (depending on message content).
-- The OPTIONAL infoValue parameter of InfoTypeAndValue will
-- typically be omitted for some of the examples given above.
-- The receiver is free to ignore any contained OBJ. IDs that it
-- does not recognize. If sent from EE to CA, the empty set
-- indicates that the CA may send
-- any/all information that it wishes.
GenRepContent ::= SEQUENCE OF InfoTypeAndValue
-- Receiver MAY ignore any contained OIDs that it does not
```

```
-- recognize.

ErrorMsgContent ::= SEQUENCE {
    pKIStatusInfo          PKIStatusInfo,
    errorCode              INTEGER          OPTIONAL,
    -- implementation-specific error codes
    errorDetails           PKIFreeText     OPTIONAL
    -- implementation-specific error details
}

PollReqContent ::= SEQUENCE OF SEQUENCE {
    certReqId              INTEGER
}

PollRepContent ::= SEQUENCE OF SEQUENCE {
    certReqId              INTEGER,
    checkAfter             INTEGER, -- time in seconds
    reason                 PKIFreeText OPTIONAL
}

--
-- Extended Key Usage extension for PKI entities used in CMP
-- operations, added due to the changes made in
-- CMP Updates [RFCXXXX]
-- The EKUs for the CA and RA are reused from CMC as defined in
-- [RFC6402]
--

-- id-kp-cmcCA OBJECT IDENTIFIER ::= { id-kp 27 }
-- id-kp-cmcRA OBJECT IDENTIFIER ::= { id-kp 28 }
id-kp-cmKGA OBJECT IDENTIFIER ::= { id-kp 32 }

-- There is no 1988 ASN.1 module of PKCS#9 available to import the
-- syntax of the localKeyId attribute type and value from. Therefore,
-- the syntax is added here as needed for the updates made in
-- CMP Updates [RFCXXXX]

pkcs-9 OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
                                rsadsi(113549) pkcs(1) 9}

pkcs-9-at-localKeyId OBJECT IDENTIFIER ::= {pkcs-9 21}

LocalKeyIdValue ::= OCTET STRING

END -- of CMP module
```

A.2. Update to RFC5912 - 2002 ASN.1 Module

This section contains the updated 2002 ASN.1 module for [RFC5912]. This module replaces the module in Section 9 of [RFC5912]. The module contains those changes to the normative ASN.1 module from RFC4210 Appendix F [RFC4210] that were to update to 2002 ASN.1 standard done in [RFC5912] as well as changes made in this document.

```
PKIXCMP-2021
  { iso(1) identified-organization(3) dod(6) internet(1)
    security(5) mechanisms(5) pkix(7) id-mod(0)
    id-mod-cmp2021-02(100) }
DEFINITIONS EXPLICIT TAGS ::=
BEGIN
IMPORTS

AttributeSet{}, SingleAttribute{}, Extensions{}, EXTENSION, ATTRIBUTE
FROM PKIX-CommonTypes-2009
  { iso(1) identified-organization(3) dod(6) internet(1) security(5)
    mechanisms(5) pkix(7) id-mod(0) id-mod-pkixCommon-02(57) }

AlgorithmIdentifier{}, SIGNATURE-ALGORITHM, ALGORITHM,
DIGEST-ALGORITHM, MAC-ALGORITHM
FROM AlgorithmInformation-2009
  { iso(1) identified-organization(3) dod(6) internet(1) security(5)
    mechanisms(5) pkix(7) id-mod(0)
    id-mod-algorithmInformation-02(58) }

Certificate, CertificateList, Time, id-kp
FROM PKIX1Explicit-2009
  { iso(1) identified-organization(3) dod(6) internet(1) security(5)
    mechanisms(5) pkix(7) id-mod(0) id-mod-pkix1-explicit-02(51) }

DistributionPointName, GeneralNames, GeneralName, KeyIdentifier
FROM PKIX1Implicit-2009
  { iso(1) identified-organization(3) dod(6) internet(1) security(5)
    mechanisms(5) pkix(7) id-mod(0) id-mod-pkix1-implicit-02(59) }

CertTemplate, PKIPublicationInfo, EncryptedKey, CertId,
CertReqMessages, Controls, RegControlSet, id-regCtrl
FROM PKIXCRMF-2009
  { iso(1) identified-organization(3) dod(6) internet(1)
    security(5) mechanisms(5) pkix(7) id-mod(0)
    id-mod-crmf2005-02(55) }
  -- The import of EncryptedKey is added due to the updates made
  -- in CMP Updates [RFCXXXX]. EncryptedValue does not need to
  -- be imported anymore and is therefore removed here.
```

```
-- see also the behavioral clarifications to CRMF codified in
-- Appendix C of this specification
```

```
CertificationRequest
```

```
FROM PKCS-10
```

```
    {iso(1) identified-organization(3) dod(6) internet(1) security(5)
      mechanisms(5) pkix(7) id-mod(0) id-mod-pkcs10-2009(69)}
-- (specified in RFC 2986 with 1993 ASN.1 syntax and IMPLICIT
-- tags). Alternatively, implementers may directly include
-- the [RFC2986] syntax in this module
```

```
localKeyId
```

```
FROM PKCS-9
```

```
    {iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-9(9)
      modules(0) pkcs-9(1)}
-- The import of localKeyId is added due to the updates made in
-- CMP Updates [RFCXXXX]
```

```
EnvelopedData, SignedData
```

```
FROM CryptographicMessageSyntax-2009
```

```
    {iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-9(9)
      smime(16) modules(0) id-mod-cms-2004-02(41)}
-- The import of EnvelopedData and SignedData is added due to
-- the updates made in CMP Updates [RFCXXXX]
```

```
;
```

```
-- the rest of the module contains locally defined OIDs and
-- constructs
```

```
CMPCertificate ::= CHOICE { x509v3PKCert Certificate, ... }
```

```
-- This syntax, while bits-on-the-wire compatible with the
-- standard X.509 definition of "Certificate", allows the
-- possibility of future certificate types (such as X.509
-- attribute certificates, WAP WTLS certificates, or other kinds
-- of certificates) within this certificate management protocol,
-- should a need ever arise to support such generality. Those
-- implementations that do not foresee a need to ever support
-- other certificate types MAY, if they wish, comment out the
-- above structure and "uncomment" the following one prior to
-- compiling this ASN.1 module. (Note that interoperability
-- with implementations that don't do this will be unaffected by
-- this change.)
```

```
-- CMPCertificate ::= Certificate
```

```
PKIMessage ::= SEQUENCE {
```

```
    header          PKIHeader,
    body            PKIBody,
```

```

protection    [0] PKIProtection OPTIONAL,
extraCerts   [1] SEQUENCE SIZE (1..MAX) OF CMPCertificate
              OPTIONAL }

```

```
PKIMessages ::= SEQUENCE SIZE (1..MAX) OF PKIMessage
```

```

PKIHeader ::= SEQUENCE {
  pvno          INTEGER          { cmp1999(1), cmp2000(2),
                                cmp2012(3) },
  sender        GeneralName,
  -- identifies the sender
  recipient     GeneralName,
  -- identifies the intended recipient
  messageTime   [0] GeneralizedTime          OPTIONAL,
  -- time of production of this message (used when sender
  -- believes that the transport will be "suitable"; i.e.,
  -- that the time will still be meaningful upon receipt)
  protectionAlg [1] AlgorithmIdentifier{ALGORITHM, {...}}
              OPTIONAL,
  -- algorithm used for calculation of protection bits
  senderKID     [2] KeyIdentifier            OPTIONAL,
  recipKID      [3] KeyIdentifier            OPTIONAL,
  -- to identify specific keys used for protection
  transactionID [4] OCTET STRING            OPTIONAL,
  -- identifies the transaction; i.e., this will be the same in
  -- corresponding request, response, certConf, and PKIConf
  -- messages
  senderNonce   [5] OCTET STRING            OPTIONAL,
  recipNonce    [6] OCTET STRING            OPTIONAL,
  -- nonces used to provide replay protection, senderNonce
  -- is inserted by the creator of this message; recipNonce
  -- is a nonce previously inserted in a related message by
  -- the intended recipient of this message
  freeText      [7] PKIFreeText             OPTIONAL,
  -- this may be used to indicate context-specific instructions
  -- (this field is intended for human consumption)
  generalInfo   [8] SEQUENCE SIZE (1..MAX) OF
              InfoTypeAndValue             OPTIONAL
  -- this may be used to convey context-specific information
  -- (this field not primarily intended for human consumption)
}

```

```

PKIFreeText ::= SEQUENCE SIZE (1..MAX) OF UTF8String
  -- text encoded as UTF-8 String [RFC3629]

```

```

PKIBody ::= CHOICE {
  -- message-specific body elements
  ir    [0] CertReqMessages,      --Initialization Request
  ip    [1] CertRepMessage,       --Initialization Response

```

```

cr      [2]  CertReqMessages,      --Certification Request
cp      [3]  CertRepMessage,      --Certification Response
p10cr   [4]  CertificationRequest, --imported from [RFC2986]
popdecc [5]  POPODecKeyChallContent, --pop Challenge
popdecr [6]  POPODecKeyRespContent, --pop Response
kur     [7]  CertReqMessages,      --Key Update Request
kup     [8]  CertRepMessage,      --Key Update Response
krr     [9]  CertReqMessages,      --Key Recovery Request
krp    [10]  KeyRecRepContent,      --Key Recovery Response
rr      [11]  RevReqContent,        --Revocation Request
rp      [12]  RevRepContent,        --Revocation Response
ccr     [13]  CertReqMessages,      --Cross-Cert. Request
ccp     [14]  CertRepMessage,      --Cross-Cert. Response
ckuann  [15]  CAKeyUpdAnnContent,    --CA Key Update Ann.
cann    [16]  CertAnnContent,       --Certificate Ann.
rann    [17]  RevAnnContent,        --Revocation Ann.
crlann  [18]  CRLAnnContent,        --CRL Announcement
pkiconf [19]  PKIConfirmContent,     --Confirmation
nested  [20]  NestedMessageContent, --Nested Message
genm    [21]  GenMsgContent,        --General Message
genp    [22]  GenRepContent,        --General Response
error   [23]  ErrorMsgContent,      --Error Message
certConf [24] CertConfirmContent,    --Certificate confirm
pollReq [25]  PollReqContent,       --Polling request
pollRep [26]  PollRepContent,       --Polling response
}

```

```
PKIProtection ::= BIT STRING
```

```
ProtectedPart ::= SEQUENCE {
```

```
  header      PKIHeader,
```

```
  body        PKIBody }
```

```
id-PasswordBasedMac OBJECT IDENTIFIER ::= { iso(1) member-body(2)
```

```
  usa(840) nt(113533) nsn(7) algorithms(66) 13 }
```

```
PBMPParameter ::= SEQUENCE {
```

```
  salt          OCTET STRING,
```

```
  -- note: implementations MAY wish to limit acceptable sizes
```

```
  -- of this string to values appropriate for their environment
```

```
  -- in order to reduce the risk of denial-of-service attacks
```

```
  owf           AlgorithmIdentifier{DIGEST-ALGORITHM, {...}},
```

```
  -- AlgId for a One-Way Function
```

```
  iterationCount  INTEGER,
```

```
  -- number of times the OWF is applied
```

```
  -- note: implementations MAY wish to limit acceptable sizes
```

```
  -- of this integer to values appropriate for their environment
```

```
  -- in order to reduce the risk of denial-of-service attacks
```

```
  mac           AlgorithmIdentifier{MAC-ALGORITHM, {...}}
```

```
-- the MAC AlgId (e.g., DES-MAC, Triple-DES-MAC [PKCS11],
-- or HMAC [RFC2104, RFC2202])
}

id-DHBasedMac OBJECT IDENTIFIER ::= { iso(1) member-body(2)
  usa(840) nt(113533) nsn(7) algorithms(66) 30 }
DHBMPParameter ::= SEQUENCE {
  owf          AlgorithmIdentifier{DIGEST-ALGORITHM, {...}},
  -- AlgId for a One-Way Function
  mac         AlgorithmIdentifier{MAC-ALGORITHM, {...}}
  -- the MAC AlgId (e.g., DES-MAC, Triple-DES-MAC [PKCS11],
  -- or HMAC [RFC2104, RFC2202])
}

PKIStatus ::= INTEGER {
  accepted          (0),
  -- you got exactly what you asked for
  grantedWithMods  (1),
  -- you got something like what you asked for; the
  -- requester is responsible for ascertaining the differences
  rejection        (2),
  -- you don't get it, more information elsewhere in the message
  waiting          (3),
  -- the request body part has not yet been processed; expect to
  -- hear more later (note: proper handling of this status
  -- response MAY use the polling req/rep PKIMessages specified
  -- in Section 5.3.22 of [RFC4210]; alternatively, polling in the
  -- underlying transport layer MAY have some utility in this
  -- regard)
  revocationWarning (4),
  -- this message contains a warning that a revocation is
  -- imminent
  revocationNotification (5),
  -- notification that a revocation has occurred
  keyUpdateWarning  (6)
  -- update already done for the oldCertId specified in
  -- CertReqMsg
}

PKIFailureInfo ::= BIT STRING {
  -- since we can fail in more than one way!
  -- More codes may be added in the future if/when required.
  badAlg          (0),
  -- unrecognized or unsupported Algorithm Identifier
  badMessageCheck (1),
  -- integrity check failed (e.g., signature did not verify)
  badRequest      (2),
  -- transaction not permitted or supported
}
```

```
badTime          (3),
-- messageTime was not sufficiently close to the system time,
-- as defined by local policy
badCertId        (4),
-- no certificate could be found matching the provided criteria
badDataFormat    (5),
-- the data submitted has the wrong format
wrongAuthority   (6),
-- the authority indicated in the request is different from the
-- one creating the response token
incorrectData     (7),
-- the requester's data is incorrect (for notary services)
missingTimeStamp (8),
-- when the timestamp is missing but should be there
-- (by policy)
badPOP           (9),
-- the proof-of-possession failed
certRevoked      (10),
-- the certificate has already been revoked
certConfirmed    (11),
-- the certificate has already been confirmed
wrongIntegrity   (12),
-- not valid integrity, password based instead of signature or
-- vice versa
badRecipientNonce (13),
-- not valid recipient nonce, either missing or wrong value
timeNotAvailable (14),
-- the TSA's time source is not available
unacceptedPolicy (15),
-- the requested TSA policy is not supported by the TSA
unacceptedExtension (16),
-- the requested extension is not supported by the TSA
addInfoNotAvailable (17),
-- the additional information requested could not be
-- understood or is not available
badSenderNonce   (18),
-- not valid sender nonce, either missing or wrong size
badCertTemplate  (19),
-- not valid cert. template or missing mandatory information
signerNotTrusted (20),
-- signer of the message unknown or not trusted
transactionIdInUse (21),
-- the transaction identifier is already in use
unsupportedVersion (22),
-- the version of the message is not supported
notAuthorized    (23),
-- the sender was not authorized to make the preceding
-- request or perform the preceding action
```

```
    systemUnavail      (24),
    -- the request cannot be handled due to system unavailability
    systemFailure      (25),
    -- the request cannot be handled due to system failure
    duplicateCertReq   (26)
    -- certificate cannot be issued because a duplicate
    -- certificate already exists
}

PKIStatusInfo ::= SEQUENCE {
    status          PKIStatus,
    statusString    PKIFreeText      OPTIONAL,
    failInfo        PKIFailureInfo   OPTIONAL }

OOBCert ::= CMPCertificate

OOBCertHash ::= SEQUENCE {
    hashAlg         [0] AlgorithmIdentifier{DIGEST-ALGORITHM, {...}}
                    OPTIONAL,
    certId          [1] CertId          OPTIONAL,
    hashVal         BIT STRING
    -- hashVal is calculated over the DER encoding of the
    -- self-signed certificate with the identifier certID.
}

POPODecKeyChallContent ::= SEQUENCE OF Challenge
-- One Challenge per encryption key certification request (in the
-- same order as these requests appear in CertReqMessages).

Challenge ::= SEQUENCE {
    owf             AlgorithmIdentifier{DIGEST-ALGORITHM, {...}}
                    OPTIONAL,
    -- MUST be present in the first Challenge; MAY be omitted in
    -- any subsequent Challenge in POPODecKeyChallContent (if
    -- omitted, then the owf used in the immediately preceding
    -- Challenge is to be used).
    witness         OCTET STRING,
    -- the result of applying the one-way function (owf) to a
    -- randomly-generated INTEGER, A. [Note that a different
    -- INTEGER MUST be used for each Challenge.]
    challenge       OCTET STRING
    -- the encryption (under the public key for which the cert.
    -- request is being made) of Rand.
}

-- Added in CMP Updates [RFCXXXX]

Rand ::= SEQUENCE {
```

```

-- Rand is encrypted under the public key to form the challenge
-- in POPODecKeyChallContent
  int                INTEGER,
  -- the randomly-generated INTEGER A (above)
  sender             GeneralName
  -- the sender's name (as included in PKIHeader)
}

POPODecKeyRespContent ::= SEQUENCE OF INTEGER
-- One INTEGER per encryption key certification request (in the
-- same order as these requests appear in CertReqMessages). The
-- retrieved INTEGER A (above) is returned to the sender of the
-- corresponding Challenge.

CertRepMessage ::= SEQUENCE {
  caPubs             [1] SEQUENCE SIZE (1..MAX) OF CMPCertificate
                    OPTIONAL,
  response           SEQUENCE OF CertResponse }

CertResponse ::= SEQUENCE {
  certReqId          INTEGER,
  -- to match this response with the corresponding request (a value
  -- of -1 is to be used if certReqId is not specified in the
  -- corresponding request, which can only be a p10cr)
  status             PKIStatusInfo,
  certifiedKeyPair   CertifiedKeyPair   OPTIONAL,
  rspInfo            OCTET STRING       OPTIONAL
  -- analogous to the id-regInfo-utf8Pairs string defined
  -- for regInfo in CertReqMsg [RFC4211]
}

CertifiedKeyPair ::= SEQUENCE {
  certOrEncCert      CertOrEncCert,
  privateKey         [0] EncryptedKey   OPTIONAL,
  -- see [RFC4211] for comment on encoding
  -- Changed from Encrypted Value to EncryptedKey as a CHOICE of
  -- EncryptedValue and EnvelopedData due to the changes made in
  -- CMP Updates [RFCXXXX]
  -- Using the choice EncryptedValue is bit-compatible to the
  -- syntax without this change
  publicationInfo   [1] PKIPublicationInfo OPTIONAL }

CertOrEncCert ::= CHOICE {
  certificate        [0] CMPCertificate,
  encryptedCert      [1] EncryptedKey
  -- Changed from Encrypted Value to EncryptedKey as a CHOICE of
  -- EncryptedValue and EnvelopedData due to the changes made in
  -- CMP Updates [RFCXXXX]
}

```

```
-- Using the choice EncryptedValue is bit-compatible to the
-- syntax without this change
}

KeyRecRepContent ::= SEQUENCE {
    status                PKIStatusInfo,
    newSigCert            [0] CMPCertificate OPTIONAL,
    caCerts               [1] SEQUENCE SIZE (1..MAX) OF
                        CMPCertificate OPTIONAL,
    keyPairHist          [2] SEQUENCE SIZE (1..MAX) OF
                        CertifiedKeyPair OPTIONAL }

RevReqContent ::= SEQUENCE OF RevDetails

RevDetails ::= SEQUENCE {
    certDetails          CertTemplate,
    -- allows requester to specify as much as they can about
    -- the cert. for which revocation is requested
    -- (e.g., for cases in which serialNumber is not available)
    crlEntryDetails     Extensions{{...}} OPTIONAL
    -- requested crlEntryExtensions
}

RevRepContent ::= SEQUENCE {
    status              SEQUENCE SIZE (1..MAX) OF PKIStatusInfo,
    -- in same order as was sent in RevReqContent
    revCerts [0] SEQUENCE SIZE (1..MAX) OF CertId OPTIONAL,
    -- IDs for which revocation was requested
    -- (same order as status)
    crls [1] SEQUENCE SIZE (1..MAX) OF CertificateList OPTIONAL
    -- the resulting CRLs (there may be more than one)
}

CAKeyUpdAnnContent ::= SEQUENCE {
    oldWithNew    CMPCertificate, -- old pub signed with new priv
    newWithOld    CMPCertificate, -- new pub signed with old priv
    newWithNew    CMPCertificate -- new pub signed with new priv
}

CertAnnContent ::= CMPCertificate

RevAnnContent ::= SEQUENCE {
    status                PKIStatus,
    certId               CertId,
    willBeRevokedAt     GeneralizedTime,
    badSinceDate        GeneralizedTime,
    crlDetails          Extensions{{...}} OPTIONAL
    -- extra CRL details (e.g., crl number, reason, location, etc.)
}
```

```
}

CRLAnnContent ::= SEQUENCE OF CertificateList
PKIConfirmContent ::= NULL

NestedMessageContent ::= PKIMessages

-- CertReqTemplateContent, AttributeTypeAndValue,
-- ExpandedRegControlSet, id-regCtrl-altCertTemplate,
-- AltCertTemplate, regCtrl-algId, id-regCtrl-algId, AlgIdCtrl,
-- regCtrl-rsaKeyLen, id-regCtrl-rsaKeyLen, and RsaKeyLenCtrl
-- were added in CMP Updates [RFCXXXX]

CertReqTemplateContent ::= SEQUENCE {
    certTemplate          CertTemplate,
    -- prefilled certTemplate structure elements
    -- The SubjectPublicKeyInfo field in the certTemplate MUST NOT
    -- be used.
    keySpec              Controls OPTIONAL
    -- MAY be used to specify supported algorithms.
    -- Controls ::= SEQUENCE SIZE (1..MAX) OF AttributeTypeAndValue
    -- as specified in CRMF (RFC4211)
}

AttributeTypeAndValue ::= SingleAttribute({ ... })

ExpandedRegControlSet ATTRIBUTE ::= { RegControlSet |
    regCtrl-altCertTemplate | regCtrl-algId | regCtrl-rsaKeyLen, ... }

regCtrl-altCertTemplate ATTRIBUTE ::=
    { TYPE AltCertTemplate IDENTIFIED BY id-regCtrl-altCertTemplate }

id-regCtrl-altCertTemplate OBJECT IDENTIFIER ::= { id-regCtrl 7 }

AltCertTemplate ::= AttributeTypeAndValue
    -- specifies a template for a certificate other than an X.509v3
    -- public-key certificate

regCtrl-algId ATTRIBUTE ::=
    { TYPE AlgIdCtrl IDENTIFIED BY id-regCtrl-algId }

id-regCtrl-algId OBJECT IDENTIFIER ::= { id-regCtrl 11 }

AlgIdCtrl ::= AlgorithmIdentifier(ALGORITHM, {...})
    -- SHALL be used to specify supported algorithms other than RSA

regCtrl-rsaKeyLen ATTRIBUTE ::=
    { TYPE RsaKeyLenCtrl IDENTIFIED BY id-regCtrl-rsaKeyLen }
```

```
id-regCtrl-rsaKeyLen OBJECT IDENTIFIER ::= { id-regCtrl 12 }

RsaKeyLenCtrl ::= INTEGER (1..MAX)
  -- SHALL be used to specify supported RSA key lengths

-- RootCaKeyUpdateContent, CRLSource, and CRLStatus were added in
-- CMP Updates [RFCXXXX]

RootCaKeyUpdateContent ::= SEQUENCE {
  newWithNew      CMPCertificate,
  -- new root CA certificate
  newWithOld      [0] CMPCertificate OPTIONAL,
  -- X.509 certificate containing the new public root CA key
  -- signed with the old private root CA key
  oldWithNew      [1] CMPCertificate OPTIONAL
  -- X.509 certificate containing the old public root CA key
  -- signed with the new private root CA key
}

CRLSource ::= CHOICE {
  dpn             [0] DistributionPointName,
  issuer          [1] GeneralNames }

CRLStatus ::= SEQUENCE {
  source          CRLSource,
  thisUpdate      Time OPTIONAL }

INFO-TYPE-AND-VALUE ::= TYPE-IDENTIFIER

InfoTypeAndValue ::= SEQUENCE {
  infoType        INFO-TYPE-AND-VALUE.
                  &id({SupportedInfoSet}),
  infoValue       INFO-TYPE-AND-VALUE.
                  &Type({SupportedInfoSet}{@infoType}) }

SupportedInfoSet INFO-TYPE-AND-VALUE ::= { ... }

-- Example InfoTypeAndValue contents include, but are not limited
-- to, the following (uncomment in this ASN.1 module and use as
-- appropriate for a given environment):
--
-- id-it-caProtEncCert      OBJECT IDENTIFIER ::= {id-it 1}
--   CAProtEncCertValue     ::= CMPCertificate
-- id-it-signKeyPairTypes  OBJECT IDENTIFIER ::= {id-it 2}
--   SignKeyPairTypesValue  ::= SEQUENCE SIZE (1..MAX) OF
--                               AlgorithmIdentifier{...}
-- id-it-encKeyPairTypes   OBJECT IDENTIFIER ::= {id-it 3}
--   EncKeyPairTypesValue   ::= SEQUENCE SIZE (1..MAX) OF
```

```

--                                     AlgorithmIdentifier{...}}
-- id-it-preferredSymmAlg OBJECT IDENTIFIER ::= {id-it 4}
--   PreferredSymmAlgValue ::= AlgorithmIdentifier{...}}
-- id-it-caKeyUpdateInfo OBJECT IDENTIFIER ::= {id-it 5}
--   CAKeyUpdateInfoValue ::= CAKeyUpdAnnContent
-- id-it-currentCRL OBJECT IDENTIFIER ::= {id-it 6}
--   CurrentCRLValue ::= CertificateList
-- id-it-unsupportedOIDs OBJECT IDENTIFIER ::= {id-it 7}
--   UnsupportedOIDsValue ::= SEQUENCE SIZE (1..MAX) OF
--                               OBJECT IDENTIFIER
-- id-it-keyPairParamReq OBJECT IDENTIFIER ::= {id-it 10}
--   KeyPairParamReqValue ::= OBJECT IDENTIFIER
-- id-it-keyPairParamRep OBJECT IDENTIFIER ::= {id-it 11}
--   KeyPairParamRepValue ::= AlgorithmIdentifier{...}}
-- id-it-revPassphrase OBJECT IDENTIFIER ::= {id-it 12}
--   RevPassphraseValue ::= EncryptedKey
--   - Changed from Encrypted Value to EncryptedKey as a CHOICE
--   - of EncryptedValue and EnvelopedData due to the changes
--   - made in CMP Updates [RFCXXXX]
--   - Using the choice EncryptedValue is bit-compatible to
--   - the syntax without this change
-- id-it-implicitConfirm OBJECT IDENTIFIER ::= {id-it 13}
--   ImplicitConfirmValue ::= NULL
-- id-it-confirmWaitTime OBJECT IDENTIFIER ::= {id-it 14}
--   ConfirmWaitTimeValue ::= GeneralizedTime
-- id-it-origPKIMessage OBJECT IDENTIFIER ::= {id-it 15}
--   OrigPKIMessageValue ::= PKIMessages
-- id-it-supplLangTags OBJECT IDENTIFIER ::= {id-it 16}
--   SupplLangTagsValue ::= SEQUENCE OF UTF8String
-- id-it-caCerts OBJECT IDENTIFIER ::= {id-it 17}
--   CaCertsValue ::= SEQUENCE SIZE (1..MAX) OF
--                               CMPCertificate
--   - id-it-caCerts added in CMP Updates [RFCXXXX]
-- id-it-rootCaKeyUpdate OBJECT IDENTIFIER ::= {id-it 18}
--   RootCaKeyUpdateValue ::= RootCaKeyUpdateContent
--   - id-it-rootCaKeyUpdate added in CMP Updates [RFCXXXX]
-- id-it-certReqTemplate OBJECT IDENTIFIER ::= {id-it 19}
--   CertReqTemplateValue ::= CertReqTemplateContent
--   - id-it-certReqTemplate added in CMP Updates [RFCXXXX]
-- id-it-rootCaCert OBJECT IDENTIFIER ::= {id-it 20}
--   RootCaCertValue ::= CMPCertificate
--   - id-it-rootCaCert added in CMP Updates [RFCXXXX]
-- id-it-certProfile OBJECT IDENTIFIER ::= {id-it 21}
--   CertProfileValue ::= SEQUENCE SIZE (1..MAX) OF
--                               UTF8String
--   - id-it-certProfile added in CMP Updates [RFCXXXX]
-- id-it-crlStatusList OBJECT IDENTIFIER ::= {id-it 22}
--   CRLStatusListValue ::= SEQUENCE SIZE (1..MAX) OF

```

```
--
--                                     CRLStatus
--   - id-it-crlStatusList added in CMP Updates [RFCXXXX]
--   id-it-crls          OBJECT IDENTIFIER ::= {id-it 23}
--   CRLsValue          ::= SEQUENCE SIZE (1..MAX) OF
--                                     CertificateList
--   - id-it-crls added in CMP Updates [RFCXXXX]
--
-- where
--
--   id-pkix OBJECT IDENTIFIER ::= {
--     iso(1) identified-organization(3)
--     dod(6) internet(1) security(5) mechanisms(5) pkix(7) }
-- and
--   id-it  OBJECT IDENTIFIER ::= {id-pkix 4}
--
-- This construct MAY also be used to define new PKIX Certificate
-- Management Protocol request and response messages, or general-
-- purpose (e.g., announcement) messages for future needs or for
-- specific environments.

GenMsgContent ::= SEQUENCE OF InfoTypeAndValue

-- May be sent by EE, RA, or CA (depending on message content).
-- The OPTIONAL infoValue parameter of InfoTypeAndValue will
-- typically be omitted for some of the examples given above.
-- The receiver is free to ignore any contained OBJECT IDs that it
-- does not recognize.  If sent from EE to CA, the empty set
-- indicates that the CA may send
-- any/all information that it wishes.

GenRepContent ::= SEQUENCE OF InfoTypeAndValue
-- Receiver MAY ignore any contained OIDs that it does not
-- recognize.

ErrorMsgContent ::= SEQUENCE {
  pkiStatusInfo      PKIStatusInfo,
  errorCode          INTEGER          OPTIONAL,
  -- implementation-specific error codes
  errorDetails       PKIFreeText      OPTIONAL
  -- implementation-specific error details
}

CertConfirmContent ::= SEQUENCE OF CertStatus

CertStatus ::= SEQUENCE {
  certHash  OCTET STRING,
  -- the hash of the certificate, using the same hash algorithm
```

```

-- as is used to create and verify the certificate signature
certReqId  INTEGER,
-- to match this confirmation with the corresponding req/rep
statusInfo PKIStatusInfo OPTIONAL,
hashAlg [0] AlgorithmIdentifier{DIGEST-ALGORITHM, {...}} OPTIONAL
-- the hash algorithm to use for calculating certHash
-- SHOULD NOT be used in all cases where the AlgorithmIdentifier
-- of the certificate signature specifies a hash algorithm
}

PollReqContent ::= SEQUENCE OF SEQUENCE {
    certReqId          INTEGER }

PollRepContent ::= SEQUENCE OF SEQUENCE {
    certReqId          INTEGER,
    checkAfter         INTEGER, -- time in seconds
    reason             PKIFreeText OPTIONAL }

--
-- Extended Key Usage extension for PKI entities used in CMP
-- operations, added due to the changes made in
-- CMP Updates [RFCXXXX]
-- The EKUs for the CA and RA are reused from CMC as defined in
-- [RFC6402]
--
-- id-kp-cmcCA OBJECT IDENTIFIER ::= { id-kp 27 }
-- id-kp-cmcRA OBJECT IDENTIFIER ::= { id-kp 28 }
id-kp-cmKGA OBJECT IDENTIFIER ::= { id-kp 32 }

END

```

Appendix B. History of Changes

[RFC Editor: This appendix must be deleted in the final version of the document.]

From version 22 -> 23:

- * Addressed comments from IESG discussion (see thread "Francesca Palombini's No Objection on draft-ietf-lamps-cmp-updates-22: (with COMMENT)")
- * Addressed comment from Carl (see thread "Paul Wouters' Discuss on draft-ietf-lamps-cmp-updates-21: (with DISCUSS and COMMENT)")

From version 21 -> 22:

- * Addressed comments from IESG discussion (see thread " Paul Wouters' Discuss on draft-ietf-lamps-cmp-updates-21: (with DISCUSS and COMMENT) ")

From version 20 -> 21:

- * Extended Section 1 based on feedback from the IESG telechat
- * Removed a redundant paragraph from the Abstract

From version 19 -> 20:

- * Addressed comments reported after GEN AD review

From version 18 -> 19:

- * Deleted the Comments on IANA Todos and changed the decimals TBD1 -> 22 and TBD2 -> 23
- * Updated Section 3.4 regarding Todos updating the well-known URI registration

From version 17 -> 18:

- * Addressed comments from AD Evaluation (see thread "AD Review of draft-ietf-lamps-cmp-updates-17")
- * Added Section 2.8 to clarify on the usage of GeneralizedTime (see thread "draft-ietf-lamps-cmp-updates: fractional seconds")
- * Updated Section 3.4 introducing the path segment 'p' to indicate the following arbitrary label according to the discussion during IETF 113 (see thread "/.well-known/brski reference to brski-registry")
- * Capitalized all headlines

From version 16 -> 17:

- * Removed the pre-RFC5378 work disclaimer after the RFC 4210 authors granted BCP78 rights to the IETF Trust
- * Removed note on usage of language tags in UTF8String due to reference to references to outdated/historic RFCs
- * Resolved some nits reported by I-D nit checker tool

From version 15 -> 16:

- * Updated IPR disclaimer

From version 14 -> 15:

- * Updated Section 2.16 clarifying the usage of CRLSource (see thread "CRL update retrieval - WG Last Call for draft-ietf-lamps-cmp-updates-14 and draft-ietf-lamps-lightweight-cmp-profile-08")
- * Updated Section 2.22 adding further references regarding random number generation (see thread "CMP draft WGLC: measuring entropy, CA certificates")
- * Fixed some nits

From version 13 -> 14:

- * Extended id-it-caCerts support message to allow transporting to-be-trusted root CA certificates; added respective security consideration (see thread "Generalizing the CMP "Get CA certificates" use case")
- * Rolled back changes made in previous version regarding root CA update to avoid registration of new OIDs. Yet we stucked to using id-it-rootCaCert in the genm body instead its headers' generalInfo field and removed the ToDos and TBDs on re-arranging id-it OIDs (see thread "Allocation of OIDs for CRL update retrieval (draft-ietf-lamps-cmp-updates-13)")

From version 12 -> 13:

- * Added John Gray to the list of authors due to fruitful discussion and important proposals
- * Fixed errata no. 2615, 2616, 3949, 4078, and 5201 on RFC 4210
- * Added reference on RFC 8933 regarding CMS signedAttrs to Section 2.7
- * Updated Section 2.9 and the ASN.1 modules moving the position of the hashAlg field (see thread "[CMP Updates] position of hashAlg in certStatus")
- * Changed "rootCaCert" from generalInfo to genm body and generalized to "oldTrustAnchor", renaming "rootCaKeyUpdate" to "trustAnchorUpdate" in Sections 2.14, A.1, and A.2, removing former Section 2.4
- * Added genm use case "CRL update retrieval" in Section 2.16, A.1, and A.2. (see thread "[CMP Updates] Requesting a current CRL")
- * Updated Section 2.18 and 2.17 to support polling for all kinds of CMP request messages initiated by an error message with status "waiting" as initially discussed at IETF 111
- * Updated Sections 2.19 and 2.20 regarding version handling
- * Added further OIDs and a TBD regarding reordering of the OIDs
- * Added Sections 2.21 to 2.23 with new security considerations and updated Section 5 accordingly
- * Added a ToDo regarding OID registration, renaming, and re-ordering
- * Added Section 3.1 updating the introduction of RFC 6712

- * Fixed some nits in the ASN.1 modules (see thread "draft-ietf-lamps-cmp-updates-12: Comments on A.1. 1988 ASN.1 Module" and "draft-ietf-lamps-cmp-updates-12: Comments on A.2. 2002 ASN.1 Module")
- * Replaced the term "transport" by "transfer" where appropriate to prevent confusion
- * Minor editorial changes

From version 11 -> 12:

- * Extended Section 2.5 and the ASN.1 modules in Appendix A to allow a sequence of certificate profiles in CertProfileValue (see thread "id-it-CertProfile in draft-ietf-lamps-cmp-updates")

From version 10 -> 11:

- * Add Section 2.10 to add an additional hashAlg field to the CertStatus type to support certificates signed with a signature algorithm not explicitly indicating a hash algorithm in the AlgorithmIdentifier (see thread "Hash algorithm to us for calculating certHash")
- * Added newly registered OIDs and temporarily registered URI suffix
- * Exchanged the import of CertificationRequest from RFC 2986 to the definition from RFC 6402 Appendix A.1 (see thread "CMP Update of CertificationRequest")
- * Corrected the definition of LocalKeyIdValue in Appendix A.1
- * Updated new RFC numbers for draft-lamps-crmf-update-algs

From version 9 -> 10:

- * Added 1988 ASN.1 syntax for localKeyId attribute to Appendix A.1

From version 08 -> 09:

- * Deleted specific definition of CMP CA and CMP RA in Section 2.2 and only reference RFC 6402 for definition of id-kp-cmCCA and id-kp-cmCRA to resolve the ToDo below based on feedback of Tomas Gustavsson
- * Added Section 2.4. and 2.5 to define id-it-rootCaCert and id-it-certProfile to be used in Section 2.14 and 2.15
- * Added reference to CMP Algorithms in Section 2.8
- * Extended Section 2.14 to explicitly indicate the root CA an update is requested for by using id-it-rootCaCert and changing the ASN.1 syntax to require providing the newWithOld certificate in the response message
- * Extended Section 2.15 to explicitly indicate the certificate request template by using id-it-certProfile and on further details of the newly introduced controls

- * Deleted the table on id-kp-cmcCA and id-kp-cmcRA and adding id-it-rootCaCert and id-it-certProfile in Section 2.19
- * Adding the definition of id-it-rootCaCert and id-it-certProfile in both ASN.1 modules in Appendix A
- * Minor editorial changes reflecting the above changes

From version 07 -> 08:

- * Added a ToDo to Section 2.2 to reflect a current discussion on the need of an additional CMP-CA role and ECU and differentiation from CMP-RA
- * Added ToDos to Section 2.12 and 2.13

From version 06 -> 07:

- * Added David von Oheimb as co-author
- * Changed to XML V3
- * Added Section 2.3 to enable a CMP protocol version number 3 in the PKIHeader for cases where EnvelopedData is to be used (see thread "Mail regarding draft-ietf-lamps-cmp-updates").
- * Added Section 2.4 to refer to draft-ietf-lamps-crmf-update-algs for the update of id-PasswordBasedMac for PKI message protection using passwords or shared secrets.
- * Updated Section 2.6 to introduce the protocol version number 3 to properly indicate support of EnvelopedData instead of EncryptedValue in case a transaction requires use of EnvelopedData (see thread "Mail regarding draft-ietf-lamps-cmp-updates").
- * Update Section 2.14 to make the minimal changes to the respective section in CMP more explicit.
- * Added Sections 2.15 and 2.16 to address the new cmp2021 protocol version in Section 7 Version Negotiation.
- * Updated Section 2.17 to add new OIDs for id-regCtrl-algId and id-regCtrl-rsaKeyLen for registration at IANA.
- * Added Section 2.20 to update the general rules of interpretation in Appendix D.1 regarding the new cmp2021 version.
- * Added Section 2.21 to update the Algorithm Use Profile in Appendix D.2 with the reference to the new CMP Algorithms document as decided at IETF 108.
- * Updates Section 3.1 to delete the description of a discovery mechanism as decided at IETF 108.
- * Various changes and corrections in wording.

From version 05 -> 06:

- * Added the update of Appendix D.2 with the reference to the new CMP Algorithms document as decided in IETF 108
- * Updated the IANA considerations to register new OIDs for id-regCtrl-algId and d-regCtrl-rsaKeyLen.

- * Minor changes and corrections

From version 04 -> 05:

- * Added Section 2.11 and Section 2.12 to clarify the usage of these general messages types with EC curves (see thread "AlgorithmIdentifier parameters NULL value - Re: InfoTypeAndValue in CMP headers")
- * Split former section 2.7 on adding 'CA Certificates', 'Root CA Certificates Update', and 'Certificate Request Template' in three separate sections for easier readability
- * Changed in Section 2.15 the ASN.1 syntax of CertReqTemplateValue from using rsaKeyLen to usage of controls as specified in CRMF Section 6 [RFC4211] (see thread "draft-ietf-lamps-cmp-updates and rsaKeyLen")
- * Updated the IANA considerations in Section 4 to introduce new OID for id-regCtrl-algId and id-regCtrl-rsaKeyLen (see thread "draft-ietf-lamps-cmp-updates and rsaKeyLen")
- * Updated the IANA Considerations in and the Appendixes to introduce new OID for the updates ASN.1 modules (see thread "I-D Action: draft-ietf-lamps-cmp-updates-04.txt")
- * Removed EncryptedValue from and added Controls to the list of types imported from CRMF [RFC4211] in ASN.1 modules (see thread "draft-ietf-lamps-cmp-updates and the ASN.1 modules")
- * Moved declaration of Rand out of the comment in ASN.1 modules (see thread "draft-ietf-lamps-cmp-updates and the ASN.1 modules")
- * Minor changes and corrections

From version 03 -> 04:

- * Added Section 2.7 to introduce three new id-it IDs for uses in general messages as discussed (see thread "draft-ietf-lamps-cmp-updates add section to introduce id-it-caCerts, id-it-rootCaKeyUpdate, and id-it-certReqTemplate")
- * Added the new id-it IDs and the /.well-known/cmp to the IANA Considerations of [RFC4210] in Section 2.9
- * Updated the IANA Considerations of [RFC4210] in Section 2.26
- * Some changes in wording on Section 3 due to review comments from Martin Peylo

From version 02 -> 03:

- * Added a ToDo on aligning with the CMP Algorithms draft that will be set up as decided in IETF 108
- * Updated section on Encrypted Values in Section 2.7 to add the AsymmetricKey Package structure to transport a newly generated private key as decided in IETF 108
- * Updated the IANA Considerations of [RFC4210] in Section 2.26

- * Added the pre-registered OID in Section 2.26 and the ASN.1 module
- * Added Section 3 to document the changes to RFC 6712 [RFC6712] regarding URI discovery and using the path-prefix of '/.well-known/' as discussed in IETF 108
- * Updated the IANA Considerations section
- * Added a complete updated ASN.1 module in 1988 syntax to update Appendix F of [RFC4210] and a complete updated ASN.1 module in 2002 syntax to update Section 9 of [RFC5912]
- * Minor changes in wording

From version 01 -> 02:

- * Updated section on EKU OIDs in Section 2.2 as decided in IETF 107
- * Changed from symmetric key-encryption to password-based key management technique in Section 2.7 as discussed with Russ and Jim on the mailing list
- * Defined the attribute containing the key identifier for the revocation passphrase in Section 2.26
- * Moved the change history to the Appendix

From version 00 -> 01:

- * Minor changes in wording

From draft-brockhaus-lamps-cmp-updates-03 -> draft-ietf-lamps-cmp-updates-00:

- * Changes required to reflect WG adoption

From version 02 -> 03:

- * Added some clarification in Section 2.1

From version 01 -> 02:

- * Added clarification to section on multiple protection
- * Added clarification on new EKUs after some exchange with Tomas Gustavsson
- * Reused OIDs from RFC 6402 [RFC6402] as suggested by Sean Turner at IETF 106
- * Added clarification on the field containing the key identifier for a revocation passphrase
- * Minor changes in wording

From version 00 -> 01:

- * Added a section describing the new extended key usages

- * Completed the section on changes to the specification of encrypted values
- * Added a section on clarification to Appendix D.4
- * Minor generalization in RFC 4210 [RFC4210] Sections 5.1.3.4 and 5.3.22
- * Minor changes in wording

Authors' Addresses

Hendrik Brockhaus (editor)
Siemens
Werner-von-Siemens-Strasse 1
80333 Munich
Germany
Email: hendrik.brockhaus@siemens.com
URI: <https://www.siemens.com>

David von Oheimb
Siemens
Werner-von-Siemens-Strasse 1
80333 Munich
Germany
Email: david.von.oheimb@siemens.com
URI: <https://www.siemens.com>

John Gray
Entrust
1187 Park Place
Minneapolis, MN 55379
United States of America
Email: john.gray@entrust.com
URI: <https://www.entrust.com>

Network Working Group
Internet-Draft
Updates: 5652 (if approved)
Intended status: Standards Track
Expires: February 28, 2021

R. Housley
Vigil Security
August 27, 2020

Update to the Cryptographic Message Syntax (CMS) for Algorithm
Identifier Protection
draft-ietf-lamps-cms-update-alg-id-protect-05

Abstract

This document updates the Cryptographic Message Syntax (CMS) specified in RFC 5652 to ensure that algorithm identifiers in signed-data and authenticated-data content types are adequately protected.

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 February 28, 2021.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Terminology	3
3. Required use the same hash algorithm	3
3.1. RFC 5652, Section 5.3	3
3.2. RFC 5652, Section 5.4	4
3.3. RFC 5652, Section 5.6	4
3.4. Backward Compatibility Considerations	5
3.5. Timestamp Compatibility Considerations	5
4. Recommended inclusion of the CMSAlgorithmProtection attribute	5
4.1. RFC 5652, Section 14	6
5. IANA Considerations	6
6. Security Considerations	6
7. Acknowledgements	7
8. References	7
8.1. Normative References	7
8.2. Informative References	8
Author's Address	8

1. Introduction

This document updates the Cryptographic Message Syntax (CMS) [RFC5652] to ensure that algorithm identifiers in signed-data and authenticated-data content types are adequately protected.

The CMS signed-data Content Type [RFC5652], unlike X.509 certificates [RFC5280], can be vulnerable to algorithm substitution attacks. In an algorithm substitution attack, the attacker changes either the algorithm identifier or the parameters associated with the algorithm identifier to change the verification process used by the recipient. The X.509 certificate structure protects the algorithm identifier and the associated parameters by signing them.

In an algorithm substitution attack, the attacker looks for a different algorithm that produces the same result as the algorithm used by the originator. As an example, if the signer of a message used SHA-256 [SHS] as the digest algorithm to hash the message content, then the attacker looks for a weaker hash algorithm that produces a result that is of the same length. The attacker's goal is to find a different message that results in the same hash value, which is called a cross-algorithm collision. Today, there are many hash functions that produce 256-bit results. One of them may be found to be weak in the future.

Further, when a digest algorithm produces a larger result than is needed by a digital signature algorithm, the digest value is reduced to the size needed by the signature algorithm. This can be done both

by truncation and modulo operations, with the simplest being straightforward truncation. In this situation, the attacker needs to find a collision with the reduced digest value. As an example, if the message signer uses SHA-512 [SHS] as the digest algorithm and ECDSA with the P-256 curve [DSS] as the signature algorithm, then the attacker needs to find a collision with the first half of the digest.

Similar attacks can be mounted against parameterized algorithm identifiers. When looking at randomized hash functions, such as the example in [RFC6210], the algorithm identifier parameter includes a random value that can be manipulated by an attacker looking for collisions. Some other algorithm identifiers include complex parameter structures, and each value provides another opportunity for manipulation by an attacker.

This document makes two updates to CMS to provide protection for the algorithm identifier. First, it mandates a convention followed by many implementations by requiring the originator to use the same hash algorithm to compute the digest of the message content and the digest of signed attributes. Second, it recommends that the originator include the CMSAlgorithmProtection attribute [RFC6211].

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Required use the same hash algorithm

This section updates [RFC5652] to require the originator to use the same hash algorithm to compute the digest of the message content and the digest of signed attributes.

3.1. RFC 5652, Section 5.3

Change the paragraph describing the digestAlgorithm as follows:

OLD:

digestAlgorithm identifies the message digest algorithm, and any associated parameters, used by the signer. The message digest is computed on either the content being signed or the content together with the signed attributes using the process described in Section 5.4. The message digest algorithm SHOULD be among those listed in the digestAlgorithms field of the associated SignerData.

Implementations MAY fail to validate signatures that use a digest algorithm that is not included in the SignedData digestAlgorithms set.

NEW:

digestAlgorithm identifies the message digest algorithm, and any associated parameters, used by the signer. The message digest is computed on either the content being signed or the content together with the signedAttrs using the process described in Section 5.4. The message digest algorithm SHOULD be among those listed in the digestAlgorithms field of the associated SignerData. If the signedAttrs field is present in the SignerInfo, then the same digest algorithm MUST be used to compute both the digest of the SignedData encapContentInfo eContent, which is carried in the message-digest attribute, and the digest of the DER-encoded signedAttrs, which is passed to the signature algorithm. Implementations MAY fail to validate signatures that use a digest algorithm that is not included in the SignedData digestAlgorithms set.

3.2. RFC 5652, Section 5.4

Add the following paragraph as the second paragraph in Section 5.4:

ADD:

When the signedAttrs field is present, the same digest algorithm MUST be used to compute the digest of the encapContentInfo eContent OCTET STRING, which is carried in the message-digest attribute, and the digest of the collection of attributes that are signed.

3.3. RFC 5652, Section 5.6

Change the paragraph discussing the signed attributes as follows:

OLD:

The recipient MUST NOT rely on any message digest values computed by the originator. If the SignedData signerInfo includes signedAttributes, then the content message digest MUST be calculated as described in Section 5.4. For the signature to be valid, the message digest value calculated by the recipient MUST be the same as the value of the messageDigest attribute included in the signedAttributes of the SignedData signerInfo.

NEW:

The recipient MUST NOT rely on any message digest values computed by the originator. If the SignedData signerInfo includes the signedAttrs field, then the content message digest MUST be calculated as described in Section 5.4, using the same digest algorithm to compute the digest of the encapContentInfo eContent OCTET STRING and the message-digest attribute. For the signature to be valid, the message digest value calculated by the recipient MUST be the same as the value of the messageDigest attribute included in the signedAttrs field of the SignedData signerInfo.

3.4. Backward Compatibility Considerations

The new requirement introduced above might lead to incompatibility with an implementation that allowed different digest algorithms to be used to compute the digest of the message content and the digest of signed attributes. The signatures produced by such an implementation when two different digest algorithms are used will be considered invalid by an implementation that follows this specification. However, most, if not all, implementations already require the originator to use the same digest algorithm for both operations.

3.5. Timestamp Compatibility Considerations

The new requirement introduced above might lead to compatibility issues for timestamping systems when the originator does not wish to share the message content with the Time Stamp Authority (TSA) [RFC3161]. In this situation, the originator sends a TimeStampReq to the TSA that includes a MessageImprint, which consists of a digest algorithm identifier and a digest value, then the TSA uses the originator-provided digest in the MessageImprint.

When producing the TimeStampToken, the TSA MUST use the same digest algorithm to compute the digest of the encapContentInfo eContent, which is an OCTET STRING that contains the TSTInfo, and the message-digest attribute within the SignerInfo.

To ensure that TimeStampToken values that were generated before this update remain valid, no requirement is placed on a TSA to ensure that the digest algorithm for the TimeStampToken matches the digest algorithm for the MessageImprint embedded within the TSTInfo.

4. Recommended inclusion of the CMSAlgorithmProtection attribute

This section updates [RFC5652] to recommend that the originator include the CMSAlgorithmProtection attribute [RFC6211] whenever signed attributes or authenticated attributes are present.

4.1. RFC 5652, Section 14

Add the following paragraph as the eighth paragraph in Section 14:

ADD:

While there are no known algorithm substitution attacks today, the inclusion of the algorithm identifiers used by the originator as a signed attribute or an authenticated attribute makes such an attack significantly more difficult. Therefore, the originator of a signed-data content type that includes signed attributes SHOULD include the CMSAlgorithmProtection attribute [RFC6211] as one of the signed attributes. Likewise, the originator of an authenticated-data content type that includes authenticated attributes SHOULD include the CMSAlgorithmProtection attribute [RFC6211] as one of the authenticated attributes.

5. IANA Considerations

This document makes no requests of the IANA.

6. Security Considerations

The security properties of the CMS [RFC5652] signed-data and authenticated-data content types are updated to offer protection for algorithm identifiers, which makes algorithm substitution attacks significantly more difficult.

For the signed-data content type, the improvements specified in this document force an attacker to mount a hash algorithm substitution attack on the overall signature, not just on the message digest of the encapContentInfo eContent.

Some digital signature algorithms have prevented hash function substitutions by including a digest algorithm identifier as an input to the signature algorithm. As discussed in [HASHID], such a "firewall" may not be effective or even possible with newer signature algorithms. For example, RSASSA-PKCS1-v1_5 [RFC8017] protects the digest algorithm identifier, but RSASSA-PSS [RFC8017] does not. Therefore, it remains important that a signer have a way to signal to a recipient which digest algorithms are allowed to be used in conjunction with the verification of an overall signature. This signaling can be done as part of the specification of the signature algorithm, in an X.509v3 certificate extension [RFC5280], or some other means. The Digital Signature Standard (DSS) [DSS] takes the first approach by requiring the use of an "approved" one-way hash algorithm.

For the authenticated-data content type, the improvements specified in this document force an attacker to mount a MAC algorithm substitution attack, which is difficult because the attacker does not know the authentication key.

The CMSAlgorithmProtection attribute [RFC6211] offers protection for the algorithm identifiers used in the signed-data and authenticated-data content types. However, no protection is provided for the algorithm identifiers in the enveloped-data, digested-data, or encrypted-data content types. Likewise, The CMSAlgorithmProtection attribute provides no protection for the algorithm identifiers used in the authenticated-enveloped-data content type defined in [RFC5083]. A mechanism for algorithm identifier protection for these content types is work for the future.

7. Acknowledgements

Many thanks to Jim Schaad and Peter Gutmann; without knowing it, they motivated me to write this document. Thanks to Roman Danyliw, Ben Kaduk, and Peter Yee for their careful review and editorial suggestions.

8. References

8.1. Normative References

- [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>>.
- [RFC3161] Adams, C., Cain, P., Pinkas, D., and R. Zuccherato, "Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)", RFC 3161, DOI 10.17487/RFC3161, August 2001, <<https://www.rfc-editor.org/info/rfc3161>>.
- [RFC5652] Housley, R., "Cryptographic Message Syntax (CMS)", STD 70, RFC 5652, DOI 10.17487/RFC5652, September 2009, <<https://www.rfc-editor.org/info/rfc5652>>.
- [RFC6211] Schaad, J., "Cryptographic Message Syntax (CMS) Algorithm Identifier Protection Attribute", RFC 6211, DOI 10.17487/RFC6211, April 2011, <<https://www.rfc-editor.org/info/rfc6211>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

8.2. Informative References

- [DSS] National Institute of Standards and Technology (NIST), "Digital Signature Standard (DSS)", FIPS Publication 186-4, July 2013.
- [HASHID] Kaliski, B., "On Hash Function Firewalls in Signature Schemes", Lecture Notes in Computer Science, Volume 2271, DOI 10.1007/3-540-45760-7_1, February 2002.
- [RFC5083] Housley, R., "Cryptographic Message Syntax (CMS) Authenticated-Enveloped-Data Content Type", RFC 5083, DOI 10.17487/RFC5083, November 2007, <<https://www.rfc-editor.org/info/rfc5083>>.
- [RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <<https://www.rfc-editor.org/info/rfc5280>>.
- [RFC6210] Schaad, J., "Experiment: Hash Functions with Parameters in the Cryptographic Message Syntax (CMS) and S/MIME", RFC 6210, DOI 10.17487/RFC6210, April 2011, <<https://www.rfc-editor.org/info/rfc6210>>.
- [RFC8017] Moriarty, K., Ed., Kaliski, B., Jonsson, J., and A. Rusch, "PKCS #1: RSA Cryptography Specifications Version 2.2", RFC 8017, DOI 10.17487/RFC8017, November 2016, <<https://www.rfc-editor.org/info/rfc8017>>.
- [SHS] National Institute of Standards and Technology (NIST), "Secure Hash Standard", FIPS Publication 180-4, August 2015.

Author's Address

Russ Housley
Vigil Security, LLC
516 Dranesville Road
Herndon, VA 20170
US

Email: housley@vigilsec.com

LAMPS Working Group
Internet-Draft
Updates: 8551 (if approved)
Intended status: Standards Track
Expires: 2 September 2024

D. K. Gillmor
American Civil Liberties Union
B. Hoeneisen
pEp Project
A. Melnikov
Isode Ltd
1 March 2024

Header Protection for Cryptographically Protected E-mail
draft-ietf-lamps-header-protection-20

Abstract

S/MIME version 3.1 introduced a mechanism to provide end-to-end cryptographic protection of e-mail message headers. However, few implementations generate messages using this mechanism, and several legacy implementations have revealed rendering or security issues when handling such a message.

This document updates the S/MIME specification ([RFC8551]) to offer a different mechanism that provides the same cryptographic protections but with fewer downsides when handled by legacy clients. The Header Protection schemes described here are also applicable to messages with PGP/MIME cryptographic protections. Furthermore, this document offers more explicit guidance for clients when generating or handling e-mail messages with cryptographic protection of message headers.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at <https://dkg.gitlab.io/lamps-header-protection/>. Status information for this document may be found at <https://datatracker.ietf.org/doc/draft-ietf-lamps-header-protection/>.

Discussion of this document takes place on the LAMPS Working Group mailing list (<mailto:spasm@ietf.org>), which is archived at <https://mailarchive.ietf.org/arch/browse/spasm/>. Subscribe at <https://www.ietf.org/mailman/listinfo/spasm/>.

Source for this draft and an issue tracker can be found at <https://gitlab.com/dkg/lamps-header-protection>.

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 2 September 2024.

Copyright Notice

Copyright (c) 2024 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1.	Introduction	6
1.1.	Two Schemes of Header Protection	7
1.2.	Problems with Wrapped Messages	7
1.3.	Problems with Injected Headers	8
1.4.	Motivation	8
1.4.1.	Backward Compatibility	8
1.4.2.	Deliverability	9
1.5.	Other Protocols to Protect E-Mail Header Fields	9
1.6.	Applicability to PGP/MIME	10
1.7.	Requirements Language	10
1.8.	Terms	11
1.9.	Document Scope	12
1.9.1.	In Scope	12
1.9.2.	Out of Scope	13
2.	Specification	13

2.1.	Injected Headers Scheme	14
2.2.	Wrapped Message Scheme	14
2.3.	Sending Side	15
2.3.1.	Composing a Cryptographically-Protected Message Without Header Protection	15
2.3.2.	Header Confidentiality Policy	16
2.3.3.	Definition of HP-Removed and HP-Obscured Header Fields	17
2.3.4.	Composing with "Injected Headers" Header Protection	18
2.3.5.	Composing with "Wrapped Message" Header Protection .	24
2.3.6.	Choosing Between Wrapped Message and Injected Headers	26
2.4.	Default Header Confidentiality Policy	26
2.4.1.	Minimal Header Confidentiality Policy	26
2.4.2.	Strong Header Confidentiality Policy	27
2.4.3.	Null Header Confidentiality Policy	27
2.4.4.	Offering Stronger Header Confidentiality	27
2.5.	Receiving Side	28
2.5.1.	Identifying that a Message has Header Protection . .	29
2.5.2.	Updating the Cryptographic Summary	29
2.5.3.	Rendering a Message with Injected Headers	30
2.5.4.	Rendering a Wrapped Message	33
2.5.5.	Guidance for Automated Message Handling	35
2.5.6.	Affordances for Debugging and Troubleshooting	36
2.5.7.	Rendering Other Schemes	37
2.5.8.	Composing a Reply to an Encrypted Message with Header Protection	37
2.5.9.	Implicitly-rendered Header Fields	38
2.5.10.	Unprotected Header Fields Added in Transit	39
2.5.11.	Handling Undecryptable Messages	40
3.	E-mail Ecosystem Evolution	41
3.1.	Dropping Legacy Display Elements	42
3.2.	Stronger Default Header Confidentiality Policy	42
3.3.	Deprecation of Messages Without Header Protection	43
4.	Usability Considerations	44
4.1.	Mixed Protections Within a Message Are Hard To Understand	44
4.2.	Users Should Not Have To Choose a Header Confidentiality Policy	45
4.3.	Users Should Not Have To Choose a Header Protection Scheme	45
5.	Security Considerations	46
5.1.	Caution about Composing with Legacy Display Elements . .	46
6.	Privacy Considerations	47
6.1.	Some Encrypted Header Fields Are Not Always Private . . .	47
6.2.	Header Fields Can Leak Unwanted Information to the Recipient	48

6.2.1.	Encrypted Header Fields Can Be Inferred From External or Internal Metadata	49
6.2.2.	HCP May Not Mask All Data in an Encrypted Header Field	49
6.2.3.	A Naive Recipient May Overestimate the Cryptographic Status of a Header Field in an Encrypted Message	49
6.2.4.	Summary and Implementation Guidance	50
6.3.	Privacy and Deliverability Risks with Bcc and Encrypted Messages	51
7.	IANA Considerations	51
8.	Acknowledgments	54
9.	References	54
9.1.	Normative References	54
9.2.	Informative References	55
Appendix A.	Possible Problems with some Legacy Clients	58
A.1.	Problems Reviewing signed-and-encrypted Messages in List View	58
A.2.	Problems when Rendering a signed-and-encrypted Message	58
A.3.	Problems when Replying to a signed-and-encrypted Message	59
A.4.	Problems Reviewing signed-only Messages in List View	60
A.5.	Problems when Rendering a signed-only Message	60
A.6.	Problems when Replying to a signed-only Message	60
Appendix B.	Test Vectors	61
B.1.	Baseline Messages	61
B.1.1.	No Cryptographic Protections Over a Simple Message	61
B.1.2.	S/MIME Signed-only signedData Over a Simple Message, No Header Protection	62
B.1.3.	S/MIME Signed-only multipart/signed Over a Simple Message, No Header Protection	64
B.1.4.	S/MIME Encrypted and Signed Over a Simple Message, No Header Protection	66
B.1.5.	No Cryptographic Protections Over a Complex Message	69
B.1.6.	S/MIME Signed-only signedData Over a Complex Message, No Header Protection	70
B.1.7.	S/MIME Signed-only multipart/signed Over a Complex Message, No Header Protection	72
B.1.8.	S/MIME Encrypted and Signed Over a Complex Message, No Header Protection	75
B.2.	Signed-only Messages	79
B.2.1.	S/MIME Signed-only signedData Over a Simple Message, Wrapped Message	79
B.2.2.	S/MIME Signed-only multipart/signed Over a Simple Message, Wrapped Message	81
B.2.3.	S/MIME Signed-only signedData Over a Simple Message, Injected Headers	83

B.2.4.	S/MIME Signed-only multipart/signed Over a Simple Message, Injected Headers	85
B.2.5.	S/MIME Signed-only signedData Over a Complex Message, Wrapped Message	88
B.2.6.	S/MIME Signed-only multipart/signed Over a Complex Message, Wrapped Message	90
B.2.7.	S/MIME Signed-only signedData Over a Complex Message, Injected Headers	93
B.2.8.	S/MIME Signed-only multipart/signed Over a Complex Message, Injected Headers	96
B.3.	Encrypted-and-signed Messages	99
B.3.1.	S/MIME Encrypted and Signed Over a Simple Message, Wrapped Message With hcp_minimal	99
B.3.2.	S/MIME Encrypted and Signed Over a Simple Message, Injected Headers With hcp_minimal	102
B.3.3.	S/MIME Encrypted and Signed Over a Simple Message, Injected Headers With hcp_minimal (+ Legacy Display)	105
B.3.4.	S/MIME Encrypted and Signed Over a Simple Message, Wrapped Message With hcp_strong	108
B.3.5.	S/MIME Encrypted and Signed Over a Simple Message, Injected Headers With hcp_strong	112
B.3.6.	S/MIME Encrypted and Signed Over a Simple Message, Injected Headers With hcp_strong (+ Legacy Display)	115
B.3.7.	S/MIME Encrypted and Signed Reply Over a Simple Message, Wrapped Message With hcp_minimal	118
B.3.8.	S/MIME Encrypted and Signed Reply Over a Simple Message, Injected Headers With hcp_minimal	121
B.3.9.	S/MIME Encrypted and Signed Reply Over a Simple Message, Injected Headers With hcp_minimal (+ Legacy Display)	124
B.3.10.	S/MIME Encrypted and Signed Reply Over a Simple Message, Wrapped Message With hcp_strong	127
B.3.11.	S/MIME Encrypted and Signed Reply Over a Simple Message, Injected Headers With hcp_strong	131
B.3.12.	S/MIME Encrypted and Signed Reply Over a Simple Message, Injected Headers With hcp_strong (+ Legacy Display)	134
B.3.13.	S/MIME Encrypted and Signed Over a Complex Message, Wrapped Message With hcp_minimal	137
B.3.14.	S/MIME Encrypted and Signed Over a Complex Message, Injected Headers With hcp_minimal	141
B.3.15.	S/MIME Encrypted and Signed Over a Complex Message, Injected Headers With hcp_minimal (+ Legacy Display)	145
B.3.16.	S/MIME Encrypted and Signed Over a Complex Message, Wrapped Message With hcp_strong	149
B.3.17.	S/MIME Encrypted and Signed Over a Complex Message, Injected Headers With hcp_strong	153

- B.3.18. S/MIME Encrypted and Signed Over a Complex Message, Injected Headers With hcp_strong (+ Legacy Display) . 156
- B.3.19. S/MIME Encrypted and Signed Reply Over a Complex Message, Wrapped Message With hcp_minimal 160
- B.3.20. S/MIME Encrypted and Signed Reply Over a Complex Message, Injected Headers With hcp_minimal 165
- B.3.21. S/MIME Encrypted and Signed Reply Over a Complex Message, Injected Headers With hcp_minimal (+ Legacy Display) 169
- B.3.22. S/MIME Encrypted and Signed Reply Over a Complex Message, Wrapped Message With hcp_strong 173
- B.3.23. S/MIME Encrypted and Signed Reply Over a Complex Message, Injected Headers With hcp_strong 177
- B.3.24. S/MIME Encrypted and Signed Reply Over a Complex Message, Injected Headers With hcp_strong (+ Legacy Display) 181
- Appendix C. Composition Examples 185
 - C.1. New message composition 185
 - C.1.1. Unprotected message 186
 - C.1.2. Encrypted with hcp_minimal and Legacy Display 186
 - C.2. Composing a Reply 188
 - C.2.1. Unprotected message 189
 - C.2.2. Encrypted with hcp_null and Legacy Display 190
- Appendix D. Rendering Examples 192
 - D.1. Example text/plain Cryptographic Payload with Legacy Display Elements 193
 - D.2. Example text/html Cryptographic Payload with Legacy Display Elements 193
- Appendix E. Other Header Protection Schemes 194
 - E.1. Original RFC 8551 Header Protection 195
 - E.2. Pretty Easy Privacy (pEp) 195
 - E.3. "draft-autocrypt" Protected Headers 195
- Appendix F. Document Changelog 195
- Authors' Addresses 200

1. Introduction

Privacy and security issues regarding e-mail Header Protection in S/MIME and PGP/MIME have been identified for some time. Most current implementations of cryptographically-protected electronic mail protect only the body of the message, which leaves significant room for attacks against otherwise-protected messages. For example, lack of Header Protection allows an attacker to substitute the message subject and/or author.

This document describes two different schemes for how message headers can be cryptographically protected, and provides guidance for implementers of MUAs that generate and interpret such messages. It

uses the term "Legacy MUA" to refer to an MUA that does not implement either scheme. This document takes particular care to ensure that messages interact reasonably well with Legacy MUAs.

1.1. Two Schemes of Header Protection

This document addresses two different schemes for cryptographically protecting e-mail Header Sections or fields and provides guidance to implementers. One scheme ("Injected Headers") is more interoperable with Legacy MUAs, and is mandatory to implement and interpret. The other, older scheme ("Wrapped Message") is described here to enable interpretation of archived messages.

The older scheme was first specified in S/MIME 3.1 ([RFC8551]), and involves wrapping a message/rfc822 or message/global MIME object with a Cryptographic Envelope around the message to protect. This document calls this scheme "Wrapped Message", and it updates the scheme described in that document, effectively replacing the final two paragraphs of Section 3.1 of [RFC8551]. However, experience has shown that even the updated "Wrapped Message" form does not interact well with some Legacy MUAs (see Section 1.2).

The more interoperable "Injected Headers" scheme of Header Protection is introduced in this document, and is preferred over the "Wrapped Message" scheme. In the "Injected Headers" scheme, the protected Header Fields are placed directly on the Cryptographic Payload, without using an intervening message/* MIME object. See Section 2.3.4 and Section 2.5.3 for more details.

1.2. Problems with Wrapped Messages

Several Legacy MUAs have revealed rendering issues when dealing with a message that uses the Wrapped Message Header Protection scheme.

In some cases, some mail user agents cannot render message/rfc822 message subparts at all, in violation of baseline MIME requirements as described on page 5 of [RFC2049]. This leaves all Wrapped Messages unreadable by any recipient using such an MUA.

In other cases, the user sees an attachment suggesting a forwarded e-mail message, which -- in fact -- contains the protected e-mail message that should be rendered directly. In most of these cases, the user can click on the attachment to view the protected message.

However, viewing the protected message as an attachment in isolation may strip it of any security indications, leaving the user unable to assess the cryptographic properties of the message. Worse, for encrypted messages, interacting with the protected message in isolation may leak contents of the cleartext, for example, if the reply is not also encrypted.

1.3. Problems with Injected Headers

A Legacy MUA dealing with an encrypted message that has some Header Fields obscured using the Injected Headers scheme will not render the obscured Header Fields to the user at all. A workaround "Legacy Display" mechanism is provided in this document, which most Legacy MUAs should render to the user, albeit not in the same location that the Header Fields would normally be rendered.

1.4. Motivation

Users generally do not understand the distinction between message body and message header. When an e-mail message has cryptographic protections that cover the message body, but not the Header Fields, several attacks become possible.

For example, a Legacy Signed Message has a signature that covers the body but not the Header Fields. An attacker can therefore modify the Header Fields (including the Subject header) without invalidating the signature. Since most readers consider a message body in the context of the message's Subject header, the meaning of the message itself could change drastically (under the attacker's control) while still retaining the same cryptographic indicator of authenticity.

In another example, a Legacy Encrypted Message has its body effectively hidden from an adversary that snoops on the message. But if the Header Fields are not also encrypted, significant information about the message (such as the message Subject) will leak to the inspecting adversary.

However, if the sending and receiving MUAs ensure that cryptographic protections cover the message Header Section as well as the message body, these attacks are defeated.

1.4.1. Backward Compatibility

If the sending MUA is unwilling to generate such a fully-protected message due to the potential for rendering, usability, deliverability, or security issues, these defenses cannot be realized.

The sender cannot know what MUA (or MUAs) the recipient will use to handle the message. Thus, an outbound message format that is backward-compatible with as many legacy implementations as possible is a more effective vehicle for providing the whole-message cryptographic protections described above.

This document aims for backward compatibility with Legacy MUAs to the extent possible. In some cases, like when a user-visible header like the Subject is cryptographically hidden, the message cannot behave entirely identically to a Legacy MUA. But accommodations are described here that ensure a rough semantic equivalence for Legacy MUA even in these cases.

1.4.2. Deliverability

A message with perfect cryptographic protections that cannot be delivered is less useful than a message with imperfect cryptographic protections that can be delivered. Senders want their messages to reach the intended recipients.

Given the current state of the Internet mail ecosystem, encrypted messages in particular cannot shield all of their Header Fields from visibility and still be guaranteed delivery to their intended recipient.

This document accounts for this concern by providing a mechanism (Section 2.3.2) that prioritizes initial deliverability (at the cost of some header leakage) while facilitating future message variants that shield more header metadata from casual inspection.

1.5. Other Protocols to Protect E-Mail Header Fields

A separate pair of protocols also provides some cryptographic protection for the e-mail message header integrity: DomainKeys Identified Mail (DKIM) [RFC6376], as used in combination with Domain-based Message Authentication, Reporting, and Conformance (DMARC) [RFC7489]. This pair of protocols provides a domain-based reputation mechanism that can be used to mitigate some forms of unsolicited e-mail (spam).

However, the DKIM+DMARC suite provides cryptographic protection at a different scope than the mechanisms described here. In particular, the message integrity and authentication signals provided by DKIM+DMARC correspond to the domain name of the sending e-mail address, not the sending address itself, so the DKIM+DMARC suite does not provide end-to-end protection. DKIM and DMARC are typically applied to messages by (and interpreted by) mail transfer agents, not mail user agents. The mechanisms in this document are typically applied to messages by (and interpreted by) mail user agents.

Furthermore, the DKIM+DMARC suite only provides cryptographic integrity and authentication, not encryption. So cryptographic confidentiality is not available from that suite.

The DKIM+DMARC suite can be used on any message, including messages formed as described in this document. There should be no conflict between these schemes.

Though not strictly e-mail, similar protections have been in use on Usenet for signing and verification of message headers for years. See ([PGPCONTROL] and [PGPVERIFY-FORMAT] for more details. Like DKIM, these Usenet control protections offer only integrity and authentication, not encryption.

1.6. Applicability to PGP/MIME

This document describes end-to-end cryptographic protections for e-mail messages in reference to S/MIME ([RFC8551]).

Comparable end-to-end cryptographic protections can also be provided by PGP/MIME ([RFC3156]).

The mechanisms in this document should be applicable in the PGP/MIME protections as well as S/MIME protections, but analysis and implementation in this document focuses on S/MIME.

To the extent that any divergence from the mechanism described here is necessary for PGP/MIME, that divergence is out of scope for this document.

1.7. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

The key words "SPECIFICATION REQUIRED" and "IETF REVIEW" that appear in this document when used to describe namespace allocation are to be interpreted as described in [RFC8126].

1.8. Terms

The following terms are defined for the scope of this document:

- * S/MIME: Secure/Multipurpose Internet Mail Extensions (see [RFC8551])
- * PGP/MIME: MIME Security with OpenPGP (see [RFC3156])
- * Message: An E-Mail Message consisting of Header Fields (collectively called "the Header Section of the message") followed, optionally, by a Body; see [RFC5322].

Note: To avoid ambiguity, this document avoids using the terms "Header" or "Headers" in isolation, but instead always uses "Header Field" to refer to the individual field and "Header Section" to refer to the entire collection.

- * Header Field: A Header Field includes a field name, followed by a colon (":"), followed by a field body (value), and terminated by CRLF; see Section 2.2 of [RFC5322] for more details.
- * Header Section: The Header Section is a sequence of lines of characters with special syntax as defined in [RFC5322]. The Header Section of a Message contains the Header Fields associated with the Message itself. The Header Section of a MIME part (that is, a subpart of a message) typically contains Header Fields associated with that particular MIME part.
- * Body: The Body is the part of a Message that follows the Header Section and is separated from the Header Section by an empty line (i.e., a line with nothing preceding the CRLF); see [RFC5322]. It is the (bottom) section of Message containing the payload of a Message. Typically, the Body consists of a (possibly multipart) MIME [RFC2045] construct.
- * Header Protection (HP): cryptographic protection of e-mail Header Sections (or parts of it) for signatures and/or encryption
- * Cryptographic Layer, Cryptographic Payload, Cryptographic Envelope, Cryptographic Summary, Structural Header Fields, Main Body Part, User-Facing Header Fields, and MUA are all used as defined in [I-D.ietf-lamps-e2e-mail-guidance]

- * Legacy MUA: an MUA that does not understand Header Protection as described in this document. A Legacy Non-Crypto MUA is incapable of doing any end-to-end cryptographic operations. A Legacy Crypto MUA is capable of doing cryptographic operations, but does not understand or generate messages with Header Protection.
- * Legacy Signed Message: an e-mail message that was signed by a Legacy MUA (and therefore has no cryptographic authenticity or integrity protections on its Header Fields).
- * Wrapped Message: The Header Protection scheme that uses the mechanism described in [RFC8551], where the Cryptographic Payload is a message/rfc822 or message/global MIME object, augmented with a Content-Type parameter to indicate that this is the explicit intent. (see Section 2.2).
- * Injected Headers: The Header Protection scheme that uses the mechanism described in this document (see Section 2.1), where the protected Header Fields are inserted on the Cryptographic Payload directly.
- * Header Confidentiality Policy (HCP): a functional specification of which Header Fields should be obscured when composing an encrypted message with Header Protection. See Section 2.3.2.
- * Ordinary User: a user of an MUA who follows a simple and minimal experience, focused on sending and receiving e-mails. A user who opts into advanced configuration, expert mode, or the like is not an "Ordinary User".

1.9. Document Scope

This document describes sensible, simple behavior for a program that generates an e-mail message with standard end-to-end cryptographic protections, following the guidance in [I-D.ietf-lamps-e2e-mail-guidance]. An implementation conformant to this draft will produce messages that have cryptographic protection that covers the message's Header Fields as well as its body.

1.9.1. In Scope

This document also describes sensible, simple behavior for a program that interprets such a message, in a way that can take advantage of these protections covering the Header Fields as well as the body.

The message generation guidance aims to minimize negative interactions with any Legacy receiving MUA while providing actionable cryptographic properties for modern receiving clients.

In particular, this document focuses on two standard types of cryptographic protection that cover the entire message:

- * A cleartext message with a single signature, and
- * An encrypted message that contains a single cryptographic signature.

1.9.2. Out of Scope

The message composition guidance in this document (in Section 2.3.4) aims to provide minimal disruption for any Legacy MUA that receives such a message. However, a Legacy MUA by definition does not implement any of the guidance here. Therefore, the document does not attempt to provide guidance for Legacy MUAs directly.

Furthermore, this document does not explicitly contemplate other variants of cryptographic message protections, including any of these:

- * Encrypted-only message (without a cryptographic signature)
- * Triple-wrapped message
- * Signed message with multiple signatures
- * Encrypted message with a cryptographic signature outside the encryption.

All such messages are out of scope of this document.

2. Specification

As mentioned in Section 1.1, this document describes two ways to provide end-to-end cryptographic protection for an e-mail message that includes all Header Fields known to the sender at message composition time.

A receiving MUA MUST be able to handle both Header Protection schemes, as described in Section 2.5.

A sending MUA MUST be able to generate the Injected Headers scheme (Section 2.3.4), and MAY generate the Wrapped Message scheme (Section 2.3.5).

2.1. Injected Headers Scheme

A message that uses the Injected Headers scheme has protected Header Fields in the Header Section of the Cryptographic Payload.

For an encrypted message that has at least one user-visible Header Field omitted or obscured outside of the Cryptographic Payload, those Header Fields MAY also be duplicated into decorative copies in the Main Body MIME part of the Cryptographic Payload itself. These decorative copies within the message are known as "Legacy Display Elements".

Such a Legacy Display Element can be useful for a Legacy receiving MUA that doesn't yet understand how to interpret or display a cryptographically-protected confidential header. See Section 3.1 for more details about how the ecosystem could shift so that a sending MUA could avoid the need to generate any Legacy Display Element.

Composing a message with the Injected Headers scheme is described in Section 2.3.4. Rendering such a message is described in Section 2.5.3.

2.2. Wrapped Message Scheme

A message that uses the Wrapped Message scheme has a Cryptographic Payload of a single message/rfc822 (or message/global) MIME object, which itself contains the original message (including the protected Header Section).

The Wrapped Message Header Protection scheme is very similar to that described in Section 3.1 of [RFC8551]. The main augmentations this document provides to that scheme are:

- * an explicit discussion of how to obscure or remove Header Fields,
- * an additional protected-headers=wrapped parameter to the Content-Type Header Field of the Cryptographic Payload to indicate the explicit intent, and
- * a recommendation to mark such a Wrapped Message as Content-Disposition: inline to encourage Legacy MUAs to render the inner message directly rather than treating it as an attachment.

Composing a message with the Wrapped Message scheme is described in Section 2.3.5. Rendering such a message is described in Section 2.5.4.

2.3. Sending Side

This section describes the process an MUA should use to apply cryptographic protection to an e-mail message with Header Protection. We start by describing the legacy message composition process as a baseline.

2.3.1. Composing a Cryptographically-Protected Message Without Header Protection

Section 5.1 of [I-D.ietf-lamps-e2e-mail-guidance] describes the typical process for a Legacy Crypto MUA to apply cryptographic protections to an e-mail message. That guidance and terminology is replicated here for reference:

- * **origbody**: the traditional unprotected message body as a well-formed MIME tree (possibly just a single MIME leaf part). As a well-formed MIME tree, origbody already has structural Header Fields (Content-*) present.
- * **origheaders**: the intended non-structural Header Fields for the message, represented here as a list of (h,v) pairs, where h is a Header Field name and v is the associated value. Note that these are Header Fields that the MUA intends to be visible to the recipient of the message. In particular, if the MUA uses the Bcc header during composition, but plans to omit it from the message (see Section 3.6.3 of [RFC5322]), it will not be in origheaders.
- * **crypto**: The series of cryptographic protections to apply (for example, "sign with the secret key corresponding to X.509 certificate X, then encrypt to X.509 certificates X and Y"). This is a routine that accepts a MIME tree as input (the Cryptographic Payload), wraps the input in the appropriate Cryptographic Envelope, and returns the resultant MIME tree as output.

The algorithm returns a MIME object that is ready to be injected into the mail system:

- * Apply crypto to MIME part origbody, producing MIME tree output
- * For each Header Field name and value (h,v) in origheaders:
 - Add Header Field h to output with value v
- * Return output

2.3.2. Header Confidentiality Policy

When composing an encrypted message with Header Protection, the composing MUA needs a Header Confidentiality Policy (HCP). In this document, we represent that Header Confidentiality Policy as a function `hcp`:

```
* hcp(name, val_in) val_out: this function takes a non-structural
  Header Field identified by name with initial value val_in as
  arguments, and returns a replacement header value val_out. If
  val_out is the special value null, it means that the Header Field
  in question should be omitted from the set of Header Fields
  visible outside the Cryptographic Envelope.
```

Note that `hcp` is only applied to non-structural Header Fields. When composing a message, Structural Header Fields are dealt with separately, as described in Section 2.3.4 and Section 2.3.5.

As an example, an MUA that obscures the Subject Header Field by replacing it with the literal string "[...]", hides all Cc'ed recipients, and does not offer confidentiality to any other Header Fields would be represented as (in pseudocode):

```
hcp_hide_cc(name, val_in) val_out:
  if name is 'Subject':
    return '['...']'
  else if name is 'Cc':
    return null
  else:
    return val_in
```

Note that such a policy is only needed when the end-to-end protections include encryption (confidentiality). No comparable policy is needed for other end-to-end cryptographic protections (integrity and authenticity), as they are simply uniformly applied so that all Header Fields known by the sender have these protections.

This asymmetry is an unfortunate consequence of complexities in message delivery systems, some of which may reject, drop, or delay messages where all Header Fields are removed from the top-level MIME object.

This document does not mandate any particular Header Confidentiality Policy, though it offers guidance for MUA implementers in selecting one in Section 2.4. Future documents may recommend or mandate such a policy for an MUA with specific needs. Such a recommendation might be motivated by descriptions of metadata-derived attacks, or stem from research about message deliverability, or describe new signalling mechanisms, but these topics are out of scope for this document.

For alignment with common practice as well as the ABNF in Section 2.3.3 for HP-Obscured, val_out MUST be one of the following:

- * identical to val_in, or
- * the special value null, or
- * a sequence of printable and whitespace (that is, space or tab) 7-bit clean US-ASCII characters (of course, non-ASCII text can be encoded as US-ASCII using the encoded-word construct from [RFC2047])

The HCP can compute val_out using any technique describable in pseudocode, such as copying a fixed string or invocations of other pseudocode functions. If it alters the value, it MUST NOT include control or NUL characters in val_out.

2.3.3. Definition of HP-Removed and HP-Obscured Header Fields

This document defines 2 new Header Fields used for conveying the effect of sender's Header Confidentiality Policy: HP-Removed and HP-Obscured. These Header Fields enable the MUA receiving an encrypted message to reliably identify whether the sending MUA intended to make a Header Field confidential (see Section 6.2.3).

An implementation that composes encrypted e-mail and hides any of the Header Fields as described in this document (for example, due to a non-null HCP) MUST include the appropriate HP-Removed or HP-Obscured Header Fields in the Cryptographic Payload. These two MIME Header Fields should only ever appear directly within the Header Section of the Cryptographic Payload of a Cryptographic Envelope offering confidentiality. They MUST be ignored if they appear in other places.

HP-Removed includes a comma separated list of Header Field names that were omitted from the outer header when the message with Header Protection was generated. The HP-Removed Header Field can appear at most once in the Header Section of a Cryptographic Payload.

Each instance of HP-Obscured contains a Header Field name and the value that this Header Field was modified to in the outer header. The HP-Obscured Header Field can appear multiple times in the Header Section of a Cryptographic Payload.

If a Header Field name A doesn't appear in an HP-Obscured Header Field value, then the Header Field A was either removed (and thus would appear in the HP-Removed Header Field) or it was copied without any modifications to the outer header.

Syntax of these new Header Fields is defined using the following ABNF [RFC5234], where field-name, WSP, VCHAR, and FWS are defined in [RFC5322]:

```
hp-removed      = "HP-Removed:" field-name-list CRLF
field-name-list = [FWS] field-name
                 *(([FWS] "," [FWS] field-name) [FWS])
hp-observed     = "HP-Obscured:" [FWS] field-name ": "
                 replacement-value CRLF
replacement-value =  (*([FWS] VCHAR) *WSP)
```

Note that replacement-value is the same as unstructured from [RFC5322], but without the obsolete obs-unstructured option.

2.3.4. Composing with "Injected Headers" Header Protection

The "Injected Headers" Header Protection scheme places the Header Fields to be protected directly on the Cryptographic Payload. Unlike in the "Wrapped Scheme" (see compose-wrapped-message), there is no wrapping of the message body in any additional message/* MIME part. This section describes how to generate such a message.

To compose a message using "Injected Headers" Header Protection, the composing MUA uses the following inputs:

- * All the inputs described in Section 2.3.1
- * hcp: a Header Confidentiality Policy, as defined in Section 2.3.2
- * legacy: a boolean value, indicating whether any recipient of the message is believed to have a Legacy MUA. If all recipients are known to implement this draft, legacy should be set to false. (How an MUA determines the value of legacy is out of scope for this document; an initial implementation can simply set it to true)

Enabling visibility of obscured Header Fields for decryption-capable legacy clients requires transforming a header list into a readable form and including it as a decorative Legacy Display Element in specially-marked parts of the message. This document recommends two different mechanisms for such a decorative adjustment: one for a text/html Main Body Part of the e-mail message, and one for a text/plain Main Body Part. This document does not recommend adding a Legacy Display Element to any other part.

Please see Section 7.1 of [I-D.ietf-lamps-e2e-mail-guidance] for guidance on identifying the parts of a message that are a Main Body Part.

To build such a message, we replace the algorithm described in Section 2.3.1 with a more sophisticated approach. The algorithm for applying "Injected Headers" cryptographic protection to a message is as follows:

- * Let newbody be a copy of origbody
- * If crypto contains encryption, and legacy is true:
 - Create ldlist, an empty list of (header, value) pairs
 - For each Header Field name and value (h,v) in origheaders:
 - o If h is user-facing (see Section 1.1.2 of [I-D.ietf-lamps-e2e-mail-guidance]):
 - + If hcp(h,v) is not v:
 - * Add (h,v) to ldlist
 - If ldlist is not empty:
 - o Identify each leaf MIME part of newbody that represents the "main body" of the message.
 - o For each "Main Body Part" bodypart of type text/plain or text/html:
 - + Adjust bodypart by inserting a Legacy Display Element header list ldlist into its content, and adding a Content-Type parameter hp-legacy-display with value 1 (see Section 2.3.4.1 for text/plain and Section 2.3.4.2 for text/html)
- * For each Header Field name and value (h,v) in origheaders:

- Add Header Field `h` to MIME part `newbody` with value `v`
- * Set the `protected-headers` parameter on the Content-Type of MIME part `newbody` to `v1`
- * If `crypto` does not contain encryption:
 - Let `newheaders` be a copy of `origheaders`
- * Else (if `crypto` contains encryption):
 - Create new empty list of Header Field names and values `newheaders`
 - Let `hpr` be an empty comma-separated list of Header Field names
 - For each Header Field name and value `(h,v)` in `origheaders`:
 - o Let `newval` be `hcp(h,v)`
 - o If `newval` is null:
 - + Add the value `h` to `hpr`
 - o Else (if `newval` is not null):
 - + Add `(h,newval)` to `newheaders`
 - + If `newval` is not `v`:
 - * Let string `record` be the concatenation of `h`, a literal `": "` (ASCII colon (0x3A) followed by ASCII space (0x20)), and `newval`
 - * Add Header Field "HP-Obscured" to MIME part `newbody` with value `record`
 - If `hpr` is not empty:
 - o Add Header Field "HP-Removed" to MIME part `newbody` with value `hpr`
 - * Apply `crypto` to MIME part `newbody`, producing MIME tree output
 - * For each Header Field name and value `(h,v)` in `newheaders`:
 - Add Header Field `h` to output with value `v`

* Return output

Note that both new parameters (hcp and legacy) are effectively ignored if crypto does not contain encryption. This is by design, because they are irrelevant for signed-only cryptographic protections.

2.3.4.1. Adding a Legacy Display Element to a text/plain Part

For a list of obscured Header Fields represented as (header, value) pairs, concatenate them as a set of lines, with one newline at the end of each pair. Add an additional trailing newline after the resultant text, and prepend the entire list to the body of the text/plain part.

The MUA MUST also add a Content-Type parameter of hp-legacy-display with value 1 to the MIME part to indicate that a Legacy Display Element was added.

For example, if the list of obscured Header Fields was [{"Cc", "alice@example.net"}, {"Subject", "Thursday's meeting"}], then a text/plain Main Body Part that originally looked like this:

```
Content-Type: text/plain; charset=UTF-8
```

```
I think we should skip the meeting.
```

Would become:

```
Content-Type: text/plain; charset=UTF-8; hp-legacy-display=1
```

```
Subject: Thursday's meeting  
Cc: alice@example.net
```

```
I think we should skip the meeting.
```

Note that the Legacy Display Element (the lines beginning with Subject: and Cc:) are part of the body of the MIME part in question.

This example assumes that the Main Body Part in question is not the root of the Cryptographic Payload. For instance, it could be a leaf of a multipart/alternative Cryptographic Payload. This is why no additional Header Fields have been injected into the MIME part in this example.

2.3.4.2. Adding a Legacy Display Element to a text/html Part

Adding a Legacy Display Element to a text/html part is similar to how it is added to a text/plain part (see Section 2.3.4.1). Instead of adding the obscured or removed User-Facing Header Fields to a block of text delimited by a blank line, the composing MUA injects them in an HTML <div> element annotated with a class attribute of header-protection-legacy-display.

The content and formatting of this decorative <div> have no strict requirements, but they MUST represent all the obscured and removed User-Facing Header Fields in a readable fashion. A simple approach is to assemble the text in the same way as Section 2.3.4.1, wrap it in a verbatim <pre> element, and put that element in the annotated <div>.

The annotated <div> should be placed as close to the start of the <body> as possible, where it will be visible when viewed with a standard HTML renderer.

The MUA MUST also add a Content-Type parameter of hp-legacy-display with value 1 to the MIME part to indicate that a Legacy Display Element was added.

For example, if the list of obscured Header Fields was [{"Cc", "alice@example.net"}, {"Subject", "Thursday's meeting"}], then a text/html Main Body Part that originally looked like this:

```
Content-Type: text/html; charset=UTF-8
```

```
<html><head><title></title></head><body>
<p>I think we should skip the meeting.</p>
</body></html>
```

Would become:

```
Content-Type: text/html; charset=UTF-8; hp-legacy-display=1
```

```
<html><head><title></title></head><body>
<div class="header-protection-legacy-display">
<pre>Subject: Thursday's meeting
Cc: alice@example.net</pre></div>
<p>I think we should skip the meeting.</p>
</body></html>
```

This example assumes that the Main Body Part in question is not the root of the Cryptographic Payload. For instance, it could be a leaf of a multipart/alternative Cryptographic Payload. This is why no additional Header Fields have been injected into the MIME part in this example.

2.3.4.2.1. Step-by-step Example for Inserting Legacy Display Element to text/html

A composing MUA MAY insert the Legacy Display Element anywhere reasonable within the message as long as it prioritizes visibility for the reader using a Legacy decryption-capable MUA. This decision may take into account special message-specific HTML formatting expectations if the MUA is aware of them. However, some MUAs may not have any special insight into the user's preferred HTML formatting, and still want to insert a Legacy Display Element. This section offers a non-normative, simple, and minimal step-by-step approach for a composing MUA that has no other information or preferences to fall back on.

The process below assumes that the MUA already has the full HTML object that it intends to send, including all of the text supplied by the user.

- * Assemble the text exactly as specified for text/plain (see Section 2.3.4.1).
- * Wrap that text in a verbatim `<pre>` element.
- * Wrap that `<pre>` element in a `<div>` element annotated with the class header-protection-legacy-display.
- * Find the `<body>` element of the full HTML object.
- * Insert the `<div>` element as the first child of the `<body>` element.

2.3.4.3. Only Add a Legacy Display Element to Main Body Parts

Some messages may contain a text/plain or text/html subpart that is not a Main Body Part. For example, an e-mail message might contain an attached text file or a downloaded webpage. Attached documents need to be preserved as intended in the transmission, without modification.

The composing MUA MUST NOT add a Legacy Display Element to any part of the message that is not a Main Body Part. In particular, if a part is annotated with Content-Disposition: attachment, or if it does not descend via the first child of any of its multipart/mixed or multipart/related ancestors, it is not a Main Body Part, and MUST NOT be modified.

See Section 7.1 of [I-D.ietf-lamps-e2e-mail-guidance] for more guidance about common ways to distinguish Main Body Parts from other MIME parts in a message.

2.3.4.4. Do Not Add a Legacy Display Element to Other Content-Types

The purpose of injecting a Legacy Display Element into each Main Body MIME part is to enable rendering of otherwise obscured Header Fields in Legacy MUAs that are capable of message decryption, but don't know how to follow the rest of the guidance in this document.

The authors are unaware of any Legacy MUA that would render any MIME part type other than text/plain and text/html as the Main Body. A generating MUA SHOULD NOT add a Legacy Display Element to any MIME part with any other Content-Type.

2.3.5. Composing with "Wrapped Message" Header Protection

The Wrapped Message Header Protection scheme is very similar to that described in Section 3.1 of [RFC8551]. The differences are outlined in Section 2.2.

To compose a message using "Wrapped Message" Header Protection, the composing MUA uses the following inputs:

- * All the inputs described in Section 2.3.1
- * hcp: a Header Confidentiality Policy, as defined in Section 2.3.2

To build such a message, we replace the algorithm described in Section 2.3.1 with a more sophisticated approach. The algorithm for applying "Wrapped Message" cryptographic protection to a message is as follows:

- * Let newbody be a copy of origbody
- * For each Header Field name and value (h,v) in origheaders:
 - Add Header Field h to MIME part newbody with value v
- * If crypto does not contain encryption:

- Let newheaders be a copy of origheaders
- * Else (if crypto contains encryption):
 - Create new empty list of Header Field names and values newheaders
 - Let hpr be an empty comma-separated list of Header Field names
 - For each Header Field name and value (h,v) in origheaders:
 - o Let newval be hcp(h,v)
 - o If newval is null:
 - + Add the value h to hpr
 - o Else (if newval is not null):
 - + Add (h,newval) to newheaders
 - + If newval is not v:
 - * Let string record be the concatenation of h, a literal ":", " (ASCII colon (0x3A) followed by ASCII space (0x20)), and newval
 - * Add Header Field "HP-Obscured" to MIME part newbody with value record
 - If hpr is not empty:
 - o Add Header Field "HP-Removed" to MIME part newbody with value hpr
- * If any of the Header Fields in MIME part newbody, including Header Fields in the nested internal MIME structure, contain any 8-bit UTF-8 characters (see Section 3.7 of [RFC6532]):
 - Let payload be a new MIME part with one Header Field: Content-Type: message/global; protected-headers=wrapped, and whose body is newbody.
- * Else:
 - Let payload be a new MIME part with one Header Field: Content-Type: message/rfc822; protected-headers=wrapped, and whose body is newbody.

- * Add a Content-Disposition Header Field to MIME part payload with value inline
- * Apply crypto to MIME part payload, producing MIME tree output
- * For each Header Field name and value (h,v) in newheaders:
 - Add Header Field h to output with value v
- * Return output

Note that the Header Confidentiality Policy hcp parameter is effectively ignored if crypto does not contain encryption. This is by design, because it is irrelevant for signed-only cryptographic protections.

2.3.6. Choosing Between Wrapped Message and Injected Headers

When composing a message with end-to-end cryptographic protections, an MUA SHOULD protect the Header Fields of that message as well as the body, using one of the formats described here.

A compatible MUA MUST be capable of generating a message with Header Protection using the Injected Headers Section 2.3.4 format.

2.4. Default Header Confidentiality Policy

An MUA MUST have a default Header Confidentiality Policy that offers at least the protections provided by hcp_minimal as described in Section 2.4.1. Local policy and configuration may alter this default, but the MUA SHOULD NOT require the user to select an HCP.

hcp_minimal provides confidentiality for the Subject Header Field by replacing it with the literal string "[...]". This is a sensible minimal default because most users treat the Subject of a message the same way that they treat the body, and they are surprised to find that the Subject of an encrypted message is visible.

2.4.1. Minimal Header Confidentiality Policy

The most conservative recommended Header Confidentiality Policy only protects the Subject Header Field:

```
hcp_minimal(name, val_in)  val_out:
  if name is 'Subject':
    return '[...]'
  else:
    return val_in
```

hcp_minimal is the recommended default HCP for a new implementation, as it provides meaningful confidentiality protections, and is unlikely to cause deliverability or usability problems.

2.4.2. Strong Header Confidentiality Policy

Alternately, a more aggressive (and therefore more privacy-preserving) Header Confidentiality Policy only leaks a handful of fields whose absence is known to increase rates of delivery failure, and simultaneously obscures the Message-ID behind a random new one:

```
hcp_strong(name, val_in) val_out:
  if name in ['From', 'To', 'Cc', 'Date']:
    return val_in
  else if name is 'Subject':
    return ' [...]'
  else if name is 'Message-ID':
    return generate_new_message_id()
  else:
    return null
```

The function generate_new_message_id() represents whatever process the MUA typically uses to generate a Message-ID for a new outbound message.

hcp_strong is known to cause usability problems with message threading for many Legacy MUAs, and is not recommended as a default HCP for new implementations.

2.4.3. Null Header Confidentiality Policy

Legacy MUAs can be conceptualized as offering a null Header Confidentiality Policy, which offers no confidentiality protection to any Header Field:

```
hcp_null(name, val_in) val_out:
  return val_in
```

A conformant MUA that is not modified by local policy or configuration MUST NOT use hcp_null by default.

2.4.4. Offering Stronger Header Confidentiality

An MUA MAY offer even stronger confidentiality for Header Fields of an encrypted message than described in Section 2.4.2. For example, it might implement an HCP that obfuscates the From field, or omits the Cc field, or ensures Date is represented in UTC (obscuring the local timezone).

The authors of this document hope that implementers with deployment experience will document their chosen Header Confidentiality Policy and the rationale behind their choice.

This document defines `hcp_null`, `hcp_minimal`, `hcp_hide_cc`, and `hcp_strong` as a way to compare and contrast different possible behavioral choices for a composing MUA. While the HCP is not strictly a protocol element, this document creates a registry of named Header Confidentiality Policies for ease of communication.

2.4.4.1. Expert Guidance for Registering Header Confidentiality Policies

There is no formal syntax specified for the Header Confidentiality Policy, but any attempt to specify an HCP for inclusion in the registry needs to provide:

- * a stable reference document clearly indicating the distinct name for the proposed HCP
- * pseudocode that other implementers can clearly and unambiguously interpret
- * a clear explanation of why this HCP is different from all other registered HCPs
- * any relevant considerations related to deployment of the HCP (for example, known or expected deliverability, rendering, or privacy challenges and possible mitigations)

An entry should not be marked as "Recommended" unless it has been shown to offer confidentiality or privacy improvements over the status quo and have minimal or mitigatable negative impact on messages to which it is applied, considering factors such as message deliverability and security. Only one entry in the table (`hcp_minimal`) is initially marked as "Recommended". In the future, more than one entry may be marked as "Recommended".

2.5. Receiving Side

An MUA that receives a cryptographically-protected e-mail will render it for the user.

The receiving MUA will render the message body, a selected subset of Header Fields, and (as described in Section 3 of [I-D.ietf-lamps-e2e-mail-guidance]) provide a summary of the cryptographic properties of the message.

Most MUAs only render a subset of Header Fields by default. For example, few MUAs typically render Message-Id or Received Header Fields for the user, but most do render From, To, Cc, Date, and Subject.

An MUA that knows how to handle a message with Header Protection makes the following two changes to its behavior when rendering a message:

- * If it detects that an incoming message had protected Header Fields, it renders Header Fields for the message from the protected Header Fields, ignoring the external (unprotected) Header Fields.
- * It includes information in the message's Cryptographic Summary to indicate the types of protection that applied to each rendered Header Field (if any).

An MUA that handles a message with Header Protection does not need to render any new Header Fields that it did not render before.

2.5.1. Identifying that a Message has Header Protection

An incoming message can be identified as having Header Protection based on one of two signals:

- * The Cryptographic Payload has Content-Type: message/rfc822 or Content-Type: message/global and the parameter protected-headers has a value of wrapped. See Section 2.5.4 for rendering guidance.
- * The Cryptographic Payload has some other Content-Type and it has parameter protected-headers set to v1. See Section 2.5.3 for rendering guidance.

Messages of both types exist in the wild, and a compliant MUA MUST be able to handle them both. They provide the same semantics and the same meaning.

2.5.2. Updating the Cryptographic Summary

Regardless of whether a cryptographically-protected message has protected Header Fields, the Cryptographic Summary of the message should be modified to indicate what protections the Header Fields have. This field-by-field status is complex and isn't necessarily intended to be presented in full to the user. Rather, it represents the state of the message internally within the MUA, and may be used to influence behavior like replying to the message (see Section 2.5.8.1).

Each Header Field individually has exactly one the following protections:

- * unprotected (this is the case for all Header Fields in messages that have no Header Protection)
- * signed-only (bound into the same validated signature as the enclosing message, but also visible in transit)
- * encrypted-only (only appears within the Cryptographic Payload; the corresponding external Header Field was either omitted or obfuscated)
- * signed-and-encrypted (same as encrypted-only, but additionally is under a validated signature)

Note that while the message itself may be signed-and-encrypted, some Header Fields may be replicated on the outside of the message (e.g. Date). Those Header Fields would be signed-only, despite the message itself being signed-and-encrypted. Additionally, the data from some encrypted or signed-and-encrypted Header Fields may not be fully private (see Section 6.1 for more details).

Rendering the cryptographic status of each Header Field is likely to be complex and messy --- users may not understand it. It is beyond the scope of this document to suggest any specific graphical affordances or user experience. Future work should include examples of successful rendering of this information.

2.5.3. Rendering a Message with Injected Headers

When the Cryptographic Payload does not have a Content-Type of message/rfc822 or message/global, and the parameter protected-headers is set to v1, the values of the protected Header Fields are drawn from the Header Fields of the Cryptographic Payload, and the body that is rendered is the Cryptographic Payload itself.

2.5.3.1. Example Signed-only Message with Injected Headers

```
A application/pkcs7-mime; smime-type="signed-data"  
  (unwraps to)  
B multipart/alternative [Cryptographic Payload + Rendered Body]  
C text/plain  
D text/html
```

The message body should be rendered the same way as this message:

```
B multipart/alternative
C text/plain
D text/html
```

It should render Header Fields taken from part B.

Its Cryptographic Summary should indicate that the message was signed and all rendered Header Fields were included in the signature.

The MUA should ignore Header Fields from part A for the purposes of rendering.

Because this message is signed-only, none of its parts will have a Legacy Display Element.

2.5.3.2. Example Signed-and-Encrypted Message with Injected Headers

Consider a message with this structure, where the MUA is able to validate the cryptographic signature:

```
E application/pkcs7-mime; smime-type="enveloped-data"
  (decrypts to)
F application/pkcs7-mime; smime-type="signed-data"
  (unwraps to)
G multipart/alternative [Cryptographic Payload + Rendered Body]
H text/plain
I text/html
```

The message body should be rendered the same way as this message:

```
G multipart/alternative
H text/plain
I text/html
```

It should render Header Fields taken from part G.

Its Cryptographic Summary should indicate that the message was signed and encrypted. Each rendered Header Field found in G should be considered against any HP-Removed Header Field found in G and all HP-Obscured Header Fields found in G. If the field's name is found in the list of Header Field names in HP-Removed, or if one of the HP-Obscured fields refers to the field name, then the Header Field should be marked as signed-and-encrypted. Otherwise, the Header Field should be marked as signed-only.

If any of the User-Facing Header Fields are removed or obscured, the composer of this message MAY place Legacy Display Elements in parts H and I.

The MUA should ignore Header Fields from part E for the purposes of rendering.

2.5.3.3. Do Not Render Legacy Display Elements

As described in Section 2.1, a message with cryptographic confidentiality protection MAY include Legacy Display Elements for backward-compatibility with Legacy MUAs. These Legacy Display Elements are strictly decorative, unambiguously identifiable, and will be discarded by compliant implementations.

The receiving MUA SHOULD avoid rendering the identified Legacy Display Elements to the user at all, since it is aware of Header Protection and can render the actual protected Header Fields.

If a text/html or text/plain part within the Cryptographic Envelope is identified as containing Legacy Display Elements, those elements SHOULD be hidden when rendering and SHOULD be dropped when generating a draft reply or inline forwarded message. Whenever a Message or MIME subtree is exported, downloaded or otherwise further processed, implementers should consider whether or not to drop the Legacy Display Elements.

2.5.3.3.1. Identifying a Part with Legacy Display Elements

A receiving MUA acting on a message that contains an encrypting Cryptographic Layer identifies a MIME subpart within the Cryptographic Payload as containing Legacy Display Elements based on the Content-Type of the subpart.

- * The subpart's Content-Type contains a parameter hp-legacy-display with value set to 1
- * The subpart's Content-Type is either text/html (see Section 2.5.3.3.3) or text/plain (see Section 2.5.3.3.2)

Note that the term "subpart" above is used in the general sense: if the Cryptographic Payload is a single part, that part itself may contain a Legacy Display Element if it is marked with the hp-legacy-display=1 parameter.

2.5.3.3.2. Omitting Legacy Display Elements from text/plain

If a text/plain part within the Cryptographic Payload has the Content-Type parameter hp-legacy-display="1", it should be processed before rendering in the following fashion:

- * Discard the leading lines of the body of the part up to and including the first entirely blank line.

Note that implementing this strategy is dependent on the charset used by the MIME part.

See Appendix D.1 for an example.

2.5.3.3.3. Omitting Legacy Display Elements from text/html

If a text/html part within the Cryptographic Payload has the Content-Type parameter `hp-legacy-display="1"`, it should be processed before rendering in the following fashion:

- * If any element of the HTML `<body>` is a `<div>` with class attribute `header-protection-legacy-display`, that entire element should be omitted.

This cleanup could be done, for example, as a custom rule in the MUA's HTML sanitizer, if one exists. Another implementation strategy for an HTML-capable MUA would be to add an entry to the [CSS] stylesheet for such a part:

```
body div.header-protection-legacy-display { display: none; }
```

2.5.4. Rendering a Wrapped Message

Some MUAs may compose and send a message with end-to-end cryptographic protections that offer Header Protection using the Wrapped Message scheme described in Section 3.1 of [RFC8551] as augmented by this document. This section describes how a receiving MUA should identify and render such a message.

When the Cryptographic Payload has Content-Type of `message/rfc822` or `message/global`, and the parameter `protected-headers` is set to `wrapped`, the values of the protected Header Fields are drawn from the Header Fields of the Cryptographic Payload, and the body that is rendered is the body of the Cryptographic Payload.

2.5.4.1. Example Signed-Only Wrapped Message

Consider a message with this structure, where the MUA is able to validate the cryptographic signature:

```
J application/pkcs7-mime; smime-type="signed-data"
  (unwraps to)
K message/rfc822 [Cryptographic Payload]
L multipart/alternative [Rendered Body]
M text/plain
N text/html
```

The message body should be rendered the same way as this message:

```
L multipart/alternative
M text/plain
N text/html
```

It should render Header Fields taken from part K.

Its Cryptographic Summary should indicate that the message was signed and all rendered Header Fields were included in the signature.

The MUA SHOULD ignore Header Fields from part J for the purposes of rendering, unless it is rendering debugging information.

2.5.4.2. Example Signed-and-Encrypted Wrapped Message

Consider a message with this structure, where the MUA is able to validate the cryptographic signature:

```
O application/pkcs7-mime; smime-type="enveloped-data"
  (decrypts to)
P application/pkcs7-mime; smime-type="signed-data"
  (unwraps to)
Q message/rfc822 [Cryptographic Payload]
R multipart/alternative [Rendered Body]
S text/plain
T text/html
```

The message body should be rendered the same way as this message:

```
R multipart/alternative
S text/plain
T text/html
```

It should render Header Fields taken from part Q.

Its Cryptographic Summary should indicate that the message was signed and encrypted. As in Section 2.5.3.2, each rendered Header Field found in Q should be considered against any HP-Removed Header Field found in Q and all HP-Obscured Header Fields found in Q. If the field's name is found in the list of Header Field names in HP-

Removed, or if one of the HP-Obscured fields refers to the field name, then the Header Field should be marked as signed-and-encrypted. Otherwise, the Header Field should be marked as signed-only.

2.5.5. Guidance for Automated Message Handling

Some automated systems have a control channel that is operated by e-mail. For example, an incoming e-mail message could subscribe someone to a mailing list, initiate the purchase of a specific product, approve another message for redistribution, or adjust the state of some shared object.

To the extent that such a system depends on end-to-end cryptographic guarantees about the e-mail control message, Header Protection as described in this document should improve the system's security. This section provides some specific guidance for systems that use e-mail messages as a control channel that want to benefit from these security improvements.

2.5.5.1. Interpret Only Protected Header Fields

Consider the situation where an e-mail-based control channel depends on the message's cryptographic signature and the action taken depends on some Header Field of the message.

In this case, the automated system **MUST** rely on information from the Header Field that is protected by the mechanism described in this document. It **MUST NOT** rely on any Header Field found outside the Cryptographic Payload.

For example, consider an administrative interface for a mailing list manager that only accepts control messages that are signed by one of its administrators. When an inbound message for the list arrives, it is queued (waiting for administrative approval) and the system generates and listens for two distinct e-mail addresses related to the queued message -- one that approves the message, and one that rejects it. If an administrator sends a signed control message to the approval address, the mailing list verifies that the protected To: Header Field of the signed control message contains the approval address before approving the queued message for redistribution. If the protected To: Header Field does not contain that address, or there is no protected To: Header Field, then the mailing list logs or reports the error, and does not act on that control message.

2.5.5.2. Ignore Legacy Display Elements

Consider the situation where an e-mail based control channel expects to receive an end-to-end encrypted message -- for example, where the control messages need confidentiality guarantees -- and where the action taken depends on the contents of some MIME part within message body.

In this case, the automated system that decrypts the incoming messages and scans the relevant MIME part **MUST** identify when the MIME part contains a Legacy Display Element (see Section 2.5.3.3.1), and it **MUST** parse the relevant MIME part with the Legacy Display Element removed.

For example, consider an administrative interface of a confidential issue tracking software. An authorized user can confidentially adjust the status of a tracked issue by a specially-formatted first line of the message body (for example, severity #183 serious). When the user's MUA encrypts a plain text control message to this issue tracker, depending on the MUA's HCP and its choice of legacy value, it may add a Legacy Display Element. If it does so, then the first line of the message body will contain a decorative copy of the confidential Subject: Header Field. The issue tracking software decrypts the incoming control message, identifies that there is a Legacy Display Element in the part (see Section 2.5.3.3.1), strips the lines comprising the Legacy Display Element (including the first blank line), and only then parses the remaining top line to look for the expected special formatting.

2.5.6. Affordances for Debugging and Troubleshooting

Note that advanced users of an MUA may need access to the original message, for example to troubleshoot problems with the rendering MUA itself, or problems with the SMTP transport path taken by the message.

An MUA that applies these rendering guidelines **SHOULD** ensure that the full original source of the message as it was received remains available to such a user for debugging and troubleshooting.

If a troubleshooting scenario demands information about the cryptographically-protected values of Header Fields, and the message is encrypted, the debugging interface **SHOULD** also provide a "source" view of the Cryptographic Payload itself, alongside the full original source of the message as received.

2.5.7. Rendering Other Schemes

Other MUAs may have generated different structures of messages that aim to offer end-to-end cryptographic protections that include Header Protection. This document is not normative for those schemes, and it is NOT RECOMMENDED to generate these other schemes, as they can either have structural flaws or simply render poorly on Legacy MUAs. A conformant MUA MAY attempt to infer Header Protection when rendering an existing message that appears to use some other scheme not documented here. Pointers to some known other schemes can be found in Appendix E.

2.5.8. Composing a Reply to an Encrypted Message with Header Protection

When composing a reply to an encrypted message with Header Protection, the MUA is acting both as a receiving MUA and as a sending MUA. Special guidance applies here, as things can go wrong in at least two ways: leaking previously-confidential information, and replying to the wrong party.

2.5.8.1. Avoid Leaking Encrypted Header Fields in Reply

As noted in Section 5.4 of [I-D.ietf-lamps-e2e-mail-guidance], an MUA in this position MUST NOT leak previously-encrypted content in the clear in a follow-up message. The same is true for protected Header Fields.

Values from any Header Field that was identified as either encrypted-only or signed-and-encrypted based on the steps outlined above MUST NOT be placed in cleartext output when generating a message.

In particular, if Subject was encrypted, and it is copied into the draft encrypted reply, the replying MUA MUST obfuscate the unprotected (cleartext) Subject Header Field as described above.

When crafting the Header Fields for a reply message, the composing MUA can make use of the HP-Removed and HP-Obscured Header Fields from within the Cryptographic Envelope of the reference message to ensure that Header Fields derived from the reference message do not leak in the reply.

Consider a Header Field in a reply message that is generated by derivation from a Header Field in the reference message. For example, the To Header Field is typically derived from the reference message's Reply-To or From Header Fields. When generating the outer copy of the Header Field, the composing MUA first applies its own Header Confidentiality Policy. If the Header Field's value is changed by the HCP, then it is applied to the outside header and

noted in the protected Header Section using HP-Removed or HP-Obscured as appropriate, as described in Section 2.3.3. Otherwise, if the Header Field's value is unchanged, the composing MUA re-generates the Header Field using the source Header Fields from the values within the Cryptographic Payload of the reference message, as modified by the HP-Obscured or HP-Removed Header Fields. If that value is itself different than the protected value, then it is applied to the outside header and noted in the protected Header Section using HP-Obscured. If the value is the same as the protected value, then it is simply copied to the outside header directly.

See Appendix C.2 for a simple worked example of this process.

2.5.8.2. Avoid Misdirected Replies to Encrypted Messages with Header Protection

When replying to a message, the Composing MUA typically decides who to send the reply to based on:

- * the Reply-To, Mail-Followup-To, or From Header Fields
- * optionally, the other To or Cc Header Fields (if the user chose to "reply all")

When a message has Header Protection, the replying MUA MUST populate the destination fields of the draft message using the protected Header Fields, and ignore any unprotected Header Fields.

This mitigates against an attack where Mallory gets a copy of an encrypted message from Alice to Bob, and then replays the message to Bob with an additional Cc to Mallory's own e-mail address in the message's outer (unprotected) Header Section.

If Bob knows Mallory's certificate already, and he replies to such a message without following the guidance in this section, it's likely that his MUA will encrypt the cleartext of the message directly to Mallory.

2.5.9. Implicitly-rendered Header Fields

While From and To and Cc and Subject and Date are often explicitly rendered to the user, some Header Fields do affect message display, without being explicitly rendered.

For example, Message-Id, References, and In-Reply-To Header Fields may collectively be used to place a message in a "thread" or series of messages.

In another example, Section 2.5.8.2 observes that the value of the Reply-To field can influence the draft reply message. So while the user may never see the Reply-To Header Field directly, it is implicitly "rendered" when the user interacts with the message by replying to it.

An MUA that depends on any implicitly-rendered Header Field in a message with Header Protection MUST use the value from the protected Header Field, and SHOULD NOT use any value found outside the cryptographic protection unless it is known to be a Header Field added in transit, as specified in Section 2.5.10.

2.5.10. Unprotected Header Fields Added in Transit

Some Header Fields are legitimately added in transit, and could not have been known to the sender at message composition time.

The most common of these Header Fields are Received and DKIM-Signature, neither of which are typically rendered, either explicitly or implicitly.

If a receiving MUA has specific knowledge about a given Header Field, including that:

- * the Header Field would not have been known to the original sender, and
- * the Header Field might be rendered explicitly or implicitly,

then the MUA MAY decide to operate on the value of that Header Field from the unprotected Header Section, even though the message has Header Protection.

The MUA MAY prefer to verify that the Header Fields in question have additional transit-derived cryptographic protections before rendering or acting on them. For example, the MUA could verify whether these Header Fields are covered by an appropriate and valid ARC-Authentication-Results (see [RFC8617]) or DKIM-Signature (see [RFC6376]) Header Field.

Specific examples of user-meaningful Header Fields commonly added by transport agents appear below.

2.5.10.1. Mailing list Header Fields: List-* and Archived-At

If the message arrives through a mailing list, the list manager itself may inject Header Fields (most of which start with List-) in the message:

- * List-Archive
- * List-Subscribe
- * List-Unsubscribe
- * List-Id
- * List-Help
- * List-Post
- * Archived-At

For some MUAs, these Header Fields are implicitly rendered, by providing buttons for actions like "Subscribe", "View Archived Version", "Reply List", "List Info", etc.

An MUA that receives a message with Header Protection that contains these Header Fields in the unprotected section, and that has reason to believe the message is coming through a mailing list MAY decide to render them to the user (explicitly or implicitly) even though they are not protected.

2.5.11. Handling Undecryptable Messages

An MUA might receive an apparently encrypted message that it cannot currently decrypt. For example, when an MUA does not have regular access to the secret key material needed for decryption, it cannot know the cryptographically protected Header Fields, or even whether the message has any cryptographically protected Header Fields.

Such an undecrypted message will be rendered by the MUA as a message without any Header Protection. This means that the message summary may well change how it is rendered when the user is finally able to supply the secret key.

For example, the rendering of the Subject Header Field in a mailbox summary might change from [...] to the real message subject when the message is decrypted. Or the message's placement in a message thread might change if, say, References or In-Reply-To have been removed or obscured (see Section 2.5.9).

Additionally, if the MUA does not retain access to the decrypting secret key, and it drops the decrypted form of a message, the message's rendering may revert to the encrypted form. For example, if a MUA follows this behavior, the Subject Header Field in a mailbox summary might change from the real message subject back to [...]. Or, the message might be yanked out of its current thread if the MUA loses access to a removed References or In-Reply-To header.

These behaviors are likely to surprise the user. However, an MUA has several possible ways of reducing or avoiding all of these surprises, including:

- * Ensuring that the MUA always has access to decryption-capable secret key material.
- * Rendering undecrypted messages in a special quarantine view until the decryption-capable secret key material is available.

To reduce or avoid the surprises associated with a decrypted message with removed or obscured Header Fields becoming undecryptable, the MUA could also:

- * Securely cache metadata from a decrypted message's protected Header Fields so that its rendering doesn't change after the first decryption.
- * Securely store the session key associated with a decrypted message, so that attempts to read the message when the long-term secret key are unavailable can proceed using only the session key itself. See, for example, the discussion about stashing session keys in Section 9.1 of [I-D.ietf-lamps-e2e-mail-guidance].

3. E-mail Ecosystem Evolution

This document is intended to offer tooling needed to improve the state of the e-mail ecosystem in a way that can be deployed without significant disruption. Some elements of this specification are present for transitional purposes, but would not exist if the system were designed from scratch.

This section describes these transitional mechanisms, as well as some suggestions for how they might eventually be phased out.

3.1. Dropping Legacy Display Elements

Any decorative Legacy Display Element added to an encrypted message that uses the Injected Header scheme is present strictly for enabling Header Field visibility (most importantly, the Subject Header Field) when the message is viewed with a decryption-capable Legacy MUA.

Eventually, the hope is that most decryption-capable MUAs will conform to this specification, and there will be no need for injection of Legacy Display Elements in the message body. A survey of widely-used decryption-capable MUAs might be able to establish when most of them do support this specification.

At that point, a composing MUA could make the legacy parameter described in Section 2.3.4 to false by default, or could even hard-code it to false, yielding a much simpler message construction set.

Until that point, an end user might want to signal that their receiving MUAs are conformant to this draft so that a peer composing a message to them can set legacy to false. A signal indicating capability of handling messages with Header Protection might be placed in the user's cryptographic certificate, or in outbound messages.

This draft doesn't attempt to define the syntax or semantics of such a signal.

3.2. Stronger Default Header Confidentiality Policy

This draft defines two different forms of Header Confidentiality Policy. An MUA implementing an HCP for the first time SHOULD deploy `hcp_minimal` as recommended in Section 2.4. This HCP offers the most commonly-expected protection (obscuring the Subject Header Field) without risking deliverability or rendering issues.

The HCPs proposed in this draft are relatively conservative and still leak a significant amount of metadata for encrypted messages. This is largely done to ensure deliverability (see Section 1.4.2) and usability, as messages without some critical Header Fields are more likely to not reach their intended recipient.

In the future, some mail transport systems may accept and deliver messages with even less publicly-visible metadata. Many MTA operators today would ask for additional guarantees about such a message to limit the risks associated with abusive or spammy mail.

This specification offers the HCP formalism itself as a way for MUA developers and MTA operators to describe their expectations around message deliverability. MUA developers can propose a stronger default HCP, and ask MTA operators (or simply test) whether their MTAs would be likely to deliver or reject encrypted mail with that HCP applied. Proponents of a stronger HCP should explicitly document the HCP, and name it clearly and unambiguously to facilitate this kind of interoperability discussion.

Reaching widespread consensus around a stronger global default HCP is a challenging problem of coordinating many different actors. A piecemeal approach might be more feasible, where some signalling mechanism allows a message recipient, MTA operator, or third-party clearinghouse to announce what kinds of HCPs are likely to be deliverable for a given recipient. In such a situation, the default HCP for an MUA might involve consulting the signalled acceptable HCPs for all recipients, and combining them (along with a default for when no signal is present) in some way.

If such a signal were to reach widespread use, it could also be used to guide reasonable statistical default HCP choices for recipients with no signal.

This draft doesn't attempt to define the syntax or semantics of such a signal.

3.3. Deprecation of Messages Without Header Protection

At some point, when the majority of MUA clients that can generate cryptographically protected messages with Header Protection, it should be possible to deprecate any cryptographically protected message that does not have Header Protection.

For example, as noted in Section 4.1, it's possible for an MUA to decline to render a signed-only message that has no Header Protection the same as an unsigned message. And a signed-and-encrypted message without Header Protection could likewise be marked as not fully protected.

These stricter rules could be adopted immediately for all messages. Or an MUA developer could roll them out immediately for any new message, but still treat an old message (based on the Date Header Field and cryptographic signature timestamp) more leniently.

A decision like this by any popular receiving MUA could drive adoption of this standard for sending MUAs.

4. Usability Considerations

This section describes concerns for MUAs that are interested in easy adoption of Header Protection by normal users.

While they are not protocol-level artifacts, these concerns motivate the protocol features described in this document.

See also the Usability commentary in Section 2 of [I-D.ietf-lamps-e2e-mail-guidance].

4.1. Mixed Protections Within a Message Are Hard To Understand

When rendering a message to the user, the ideal circumstance is to present a single cryptographic status for any given message. However, when message Header Fields are present, some message Header Fields do not have the same cryptographic protections as the main message.

Representing such a mixed set of protection statuses is very difficult to do in a way that a normal user can understand without training. There are at least three scenarios that are likely to be common, and poorly understood:

- * A signed message with no Header Protection.
- * A signed-and-encrypted message with no Header Protection.
- * An signed-and-encrypted message with Header Protection as described in this document, where some User-Facing Header Fields have confidentiality but some do not.

An MUA should have a reasonable strategy for clearly communicating each of these scenarios to the user. For example, an MUA operating in an environment where it expects most cryptographically-protected messages to have Header Protection could use the following rendering strategy:

- * When rendering a message with signed-only cryptographic status but no Header Protection, an MUA may decline to indicate a positive security status overall, and only indicate the cryptographic status to a user in a message properties or diagnostic view. That is, the message may appear identical to an unsigned message except if a user verifies the properties through a menu option.
- * When rendering a message with signed-and-encrypted or encrypted-only cryptographic status but no Header Protection, overlay a warning flag on the typical cryptographic status indicator. That

is, if a typical signed-and-encrypted message displays a lock icon, display a lock icon with a warning sign (e.g., an exclamation point in a triangle) overlaid. See, for example, the graphics in [chrome-indicators].

- * When rendering a message with signed-and-encrypted or encrypted-only cryptographic status, with Header Protection, but where the Subject Header Field has not been removed or obscured, place a warning sign on the on the Subject line.

Other simple rendering strategies could also be reasonable.

4.2. Users Should Not Have To Choose a Header Confidentiality Policy

This document defines the abstraction of a Header Confidentiality Policy object for the sake of communication between implementers and deployments.

Most e-mail users are unlikely to understand the tradeoffs between different policies. In particular, the potential negative side effects (e.g. poor deliverability) may not be easily attributable by a normal user to a particular HCP.

Therefore, MUA implementers should be conservative in their choice of default HCP, and should not require the Ordinary User to make an incomprehensible choice that could cause unfixable, undiagnosable problems. The safest option is for the MUA developer to select a known, stable HCP (this document recommends `hcp_minimal` in Section 2.4) on the user's behalf. An MUA should not expose the Ordinary User to a configuration option where they are expected to manually select (let alone define) an HCP.

4.3. Users Should Not Have To Choose a Header Protection Scheme

This document also describes two different Header Protection schemes: Wrapped Messages in Section 2.2 and Injected Headers in Section 2.1.

These distinct schemes are described for the sake of implementers who may have to deal with messages found in the wild, but their intended semantics are identical. They represent different tradeoffs in terms of rendering and user experience on the recipient's side, things that a given user writing a message is not prepared to select.

When composing a message with cryptographic protections, the Ordinary User should not be confronted with any choices about which Header Protection scheme to use. Rather, the MUA developer should use a single scheme for all outbound cryptographically-protected messages.

This document recommends the Injected Headers scheme for generating messages with cryptographic protections, as described in Section 2. An MUA should not expose the Ordinary User to any configuration option where they are expected to manually select, enable, or disable Header Protections for new cryptographically-protected messages.

5. Security Considerations

This document describes a mechanism for improving the security of cryptographically-protected e-mail messages. Following the guidance in this document should improve security for users of these technologies by more directly aligning the underlying messages with user expectations about confidentiality, authenticity, and integrity.

However, many existing messages with cryptographic protections will not have these protections, and MUAs encountering these messages will need to handle older forms (without Header Protection) for quite some time. An implementation that deals with legacy message archives will need to deal with all the various formats forever. Helping the user distinguish between cryptographic protections of various messages is a difficult job for message renderers.

However, on the message generation side, the situation is much clearer: there is a standard form that a protected message can take, and an implementer can always generate the standard form. Generating the standard form also makes it more likely that any receiving implementation will be able to handle the generated message appropriately.

The security considerations from Section 6 of [RFC8551] continue to apply for any MUA that offers S/MIME cryptographic protections, as well as Section 3 of [RFC5083] (Authenticated-Enveloped-Data in CMS) and Section 14 of [RFC5652] (CMS more broadly). Likewise, the security considerations from Section 8 of [RFC3156] continue to apply for any MUA that offers PGP/MIME cryptographic protections, as well as Section 13 of [I-D.ietf-openpgp-crypto-refresh-13] (OpenPGP itself). In addition, these underlying security considerations are now also applicable to the contents of the message header, not just the message body.

5.1. Caution about Composing with Legacy Display Elements

When composing a message, it's possible for a Legacy Display Element to contain risky data that could trigger errors in a rendering client.

For example, if the value for a Header Field to be included in a Legacy Display Element within a given body part contains folding whitespace, it should be "unfolded" before generating the Legacy Display Element: all contiguous folding whitespace should be replaced with a single space character. Likewise, if the header value was originally encoded with [RFC2047], it should be decoded first to a standard string and re-encoded using the charset appropriate to the target part.

When including a Legacy Display Element in a text/plain part (see Section 2.3.4.1), if the decoded Subject Header Field contains a pair of newlines (e.g., if it is broken across multiple lines by encoded newlines), any newline MUST be stripped from the Legacy Display Element. If the pair of newlines is not stripped, a receiving MUA that follows the guidance in Section 2.5.3.3.2 might leave the later part of the Legacy Display Element in the rendered message.

When including a Legacy Display Element in a text/html part (see Section 2.3.4.2), any material in the header values should be explicitly HTML escaped to avoid being rendered as part of the HTML. At a minimum, the characters <, >, and & should be escaped to <;, >;, and &;, respectively (see for example [HTML-ESCAPES]). If unescaped characters from removed or obscured header values end up in the Legacy Display Element, a receiving MUA that follows the guidance in Section 2.5.3.3.3 might fail to identify the boundaries of the Legacy Display Element, cutting out more than it should, or leaving remnants visible. And a Legacy MUA parsing such a message might misrender the entire HTML stream, depending on the content of the removed or obscured header values.

The Legacy Display Element is a decorative addition solely to enable visibility of obscured or removed Header Fields in decryption-capable Legacy MUAs. When it is produced, it should be generated conservatively and narrowly, as described above, to avoid damaging the rest of the message.

6. Privacy Considerations

6.1. Some Encrypted Header Fields Are Not Always Private

For encrypted messages, depending on the sender's HCP, some Header Fields may appear both within the Cryptographic Envelope and on the outside of the message (e.g. Date might exist identically in both places). Section 2.5.2 identifies such a Header Field as signed-only. These Header Fields are clearly not private at all, despite a copy being inside the Cryptographic Envelope.

A Header Field whose name can be found in the HP-Removed or in any HP-Obscured Header Field from the same part will have encrypted-only or signed-and-encrypted status. But even Header Fields with these stronger levels of cryptographic confidentiality protection might not be as private as the user would like.

For example, even if the Date Header Field has been obscured, for example by normalizing the timezone to UTC or rounding to the most recent minute or hour (so that Header Field is formally signed-and-encrypted), the MTAs which handle the message can of course record the time that they first encountered it, which is likely to be identical or very close to the original value of the field.

6.2. Header Fields Can Leak Unwanted Information to the Recipient

For encrypted messages, even with an aggressive HCP that successfully obscures most Header Fields from all transport agents, Header Fields will be ultimately visible to all intended recipients. This can be especially problematic for Header Fields that are not user-facing, which the sender may not expect to be injected by their MUA. Consider the three following examples:

- * The MUA may inject a User-Agent Header Field that describes itself to every recipient, even though the sender may not want the recipient to know the exact version of their OS, hardware platform, or MUA.
- * The MUA may have an idiosyncratic way of generating a Message-ID header, which could embed the choice of MUA, a timezone, a hostname, or other subtle information to a knowledgeable recipient.
- * The MUA may erroneously include a Bcc Header Field in the origheaders of a copy of a message sent to the named recipient, defeating the purpose of using Bcc instead of Cc (see Section 6.3 for more details about risks related to Bcc).

Clearly, no end-to-end cryptographic protection of any Header Field as described in this document will hide such a sensitive field from the intended recipient. Instead, the composing MUA MUST populate the origheaders list for any outbound message with only information recipient should have access to. This is true for messages without any cryptographic protection as well, of course, and it is even worse there: such a leak is exposed to the transport agents as well as the recipient. An encrypted message with Header Protection and a strong Header Confidentiality Policy avoid these leaks exposing information to the transport agents, but cannot defend against such a leak to the recipient.

6.2.1. Encrypted Header Fields Can Be Inferred From External or Internal Metadata

For example, if the To: and Cc: Header Fields are omitted from the unprotected Header Section, the values in those fields might still be inferred with high probability by an adversary who looks at the message either in transit or at rest. If the message is found in, or being delivered to a mailbox for bob@example.org, it's likely that Bob was in either To: or Cc:. Furthermore, encrypted message ciphertext may hint at the recipients: for S/MIME messages, the RecipientInfo, and for PGP/MIME messages the key ID in the Public Key Encrypted Session Key (PKESK) packets will all hint at a specific set of recipients. Additionally, an MTA that handles the message may add a Received: Header Field (or some other custom Header Field) that leaks some information about the nature of the delivery.

6.2.2. HCP May Not Mask All Data in an Encrypted Header Field

In another example, if the HCP modifies the Date: header to mask out high-resolution time stamps (e.g. rounding to the most recent hour) and to convert the local timezone to UTC, some information about the date of delivery will still be attached to the e-mail. At the very least, the low resolution, global version of the date will be present on the message. Additionally, Header Fields like Received that are added during message delivery might include higher-resolution timestamps. And if the message lands in a mailbox that is ordered by time of receipt, even its placement in the mailbox and the non-obscured Date: Header Fields of the surrounding messages could leak this information.

Some fields like From: may be impossible to fully obscure, as many modern message delivery systems depend on at least domain information in the From: field for determining whether a message is coming from a domain with "good reputation" (that is, from a domain that is not known for leaking spam). So even if an aggressive HCP opts to remove the human-readable part from any From: Header Field, and to standardize/genericize the local part of the From: address, the domain will still leak.

6.2.3. A Naive Recipient May Overestimate the Cryptographic Status of a Header Field in an Encrypted Message

When an encrypted (or signed-and-encrypted) message is in transit, an active intermediary can strip or tamper with any Header Field that appears outside the Cryptographic Envelope. A receiving MUA that naively infers cryptographic status from differences between the external Header Fields and those found in the Cryptographic Envelope could be tricked into overestimating the protections afforded to some

Header Fields.

For example, if the original sender's HCP passes through the Cc: Header Field unchanged, a cleanly-delivered message would indicate that the Cc: Header Field has a cryptographic status of signed. But if an intermediary attacker simply removes the Header Field from the unprotected Header Section before forwarding the message, then the naive recipient might believe that the field has a cryptographic status of signed-and-encrypted.

This draft offers protection against such an attack by way of the HP-Obscured and HP-Removed Header Fields that can be found on the Cryptographic Payload. If a Header Field appears to have been obscured, but no HP-Obscured header matches it; or if the Header Field appears to have been removed, but the HP-Removed header does not include its field name, the receiving MUA can indicate to the user that the Header Field in question may not have been confidential.

In such a case, a conservative MUA may render the Header Field in question as signed (because the sender did not hide it), but still treat it as signed-and-encrypted during reply, to avoid accidental leakage of the cleartext value in the reply message, as described in Section 2.5.8.1.

6.2.4. Summary and Implementation Guidance

In the abstract sense, the above concerns are of course also true for any encrypted data, including the body of the message: if the sender isn't careful, the message contents or session keys could leak in many different ways that are beyond the scope of this draft. The message recipient has no way in principle to tell whether the apparent confidentiality of any given piece of encrypted content has been broken via channels that they cannot perceive. And an active intermediary aware of the recipient's public key can always encrypt a cleartext message in transit to give the recipient a false sense of security.

Despite the external inferrability of some encrypted or signed-and-encrypted Header Fields, the MUA should still strive to avoid additional leakage of these Header Fields, as described in Section 2.5.8.1.

6.3. Privacy and Deliverability Risks with Bcc and Encrypted Messages

As noted in Section 9.3 of [I-D.ietf-lamps-e2e-mail-guidance], handling Bcc when generating an encrypted e-mail message can be particularly tricky. With Header Protection, there is an additional wrinkle. When an encrypted e-mail message with Header Protection has a Bcc'ed recipient, and the composing MUA explicitly includes the Bcc'ed recipient's address in their copy of the message (see the "second method" in Section 3.6.3 of [RFC5322]), that Bcc Header Field will always be visible to the Bcc'ed recipient.

In this scenario, though, the composing MUA has one additional choice: whether to hide the Bcc Header Field from intervening message transport agents, by returning null when the HCP is invoked for Bcc. If the composing MUA's rationale for including an explicit Bcc in the copy of the message sent to the Bcc recipient is to ensure deliverability via a message transport agent that inspects message Header Fields, then stripping the Bcc field during encryption may cause the intervening transport agent to drop the message entirely. This is why Bcc is not explicitly stripped in `hcp_minimal`.

If, on the other hand, deliverability to a Bcc'ed recipient is not a concern, the most privacy-preserving option is to simply omit the Bcc Header Field from the protected Header Section in the first place. An MUA that is capable of receiving and processing such a message can infer that since their user's address was not mentioned in any To or Cc Header Field, they were likely a Bcc recipient.

Please also see Section 9.3 of [I-D.ietf-lamps-e2e-mail-guidance] for more discussion about Bcc and encrypted messages.

7. IANA Considerations

This document requests IANA to register the following two Header Fields in the "Permanent Message Header Field Names" registry within "Message Headers" in accordance with [RFC3864].

Header Field Name	Template	Protocol	Status	Reference
HP-Removed		mail	standard	Section 2.3.3 of RFCXXXX
HP-Obscured		mail	standard	Section 2.3.3 of RFCXXXX

Table 1: Additions to 'Permanent Message Header Field Names' registry

The Author/Change Controller of these two entries (Section 4.5 of [RFC3864]) should be the IETF itself.

This document also defines the Content-Type parameter known as protected-headers. Consequently, the Content-Type row in the "Permanent Message Header Field Names" registry should add a reference to this RFC to its "References" column.

That is, the current row:

Header Field Name	Template	Protocol	Status	Reference
Content-Type		MIME		[RFC4021]

Table 2: Existing row in 'Permanent Message Header Field Names' registry

Should be updated to have the following values:

Header Field Name	Template	Protocol	Status	Reference
Content-Type		MIME		[RFC4021] [RFCXXXX]

Table 3: Replacement row in 'Permanent Message Header Field Names' registry

This document also requests IANA to create a new registry in the "Mail Parameters" protocol group (<https://www.iana.org/assignments/mail-parameters/>) titled Mail Header Confidentiality Policies with the following content:

Header Confidentiality Policy Name	Description	Reference	Recommended
hcp_null	No header confidentiality	RFCXXX (this document)	N
hcp_minimal	Subject Header Field is obscured	RFCXXX (this document)	Y
hcp_strong	Remove or obscure everything but From, Date, To, and Cc	RFCXXX (this document)	N
hcp_hide_cc	Obscure Subject, remove Cc	RFCXXX (this document)	N

Table 4: Mail Header Confidentiality Policies registry

Please add the following textual note to this registry:

The Header Confidentiality Policy Name never appears on the wire. This registry merely tracks stable references to implementable descriptions of distinct policies. Any addition to this registry should be governed by guidance in Section 2.4.4.1 of RFC XXX (this document).

Adding an entry to this registry with an N in the "Recommended" column follows the registration policy of SPECIFICATION REQUIRED. Adding an entry to this registry with a Y in the "Recommended" column or changing the "Recommended" column in an existing entry (from N to Y or vice versa) requires IETF REVIEW. During IETF REVIEW, the designated expert must also be consulted. Guidance for the designated expert can be found in Section 2.4.4.1.

8. Acknowledgments

The authors would like to thank the following people who have provided helpful comments and suggestions for this document: Berna Alp, Bernhard E. Reiter, Carl Wallace, Claudio Luck, David Wilson, Hernani Marques, juga, Krista Bennett, Kelly Bristol, Lars Rohwedder, Michael StJohns, Nicolas Lidzborski, Phillip Tao, Robert Williams, Roman Danyliw, Russ Housley, Sofia Balicka, Steve Kille, Volker Birk, and Wei Chuang.

9. References

9.1. Normative References

- [I-D.ietf-lamps-e2e-mail-guidance]
Gillmor, D. K., Hoeneisen, B., and A. Melnikov, "Guidance on End-to-End E-mail Security", Work in Progress, Internet-Draft, draft-ietf-lamps-e2e-mail-guidance-15, 1 March 2024, <<https://datatracker.ietf.org/doc/html/draft-ietf-lamps-e2e-mail-guidance-15>>.
- [I-D.ietf-lamps-header-protection-requirements]
Melnikov, A. and B. Hoeneisen, "Problem Statement and Requirements for Header Protection", Work in Progress, Internet-Draft, draft-ietf-lamps-header-protection-requirements-01, 29 October 2019, <<https://datatracker.ietf.org/doc/html/draft-ietf-lamps-header-protection-requirements-01>>.
- [I-D.ietf-openpgp-crypto-refresh-13]
Wouters, P., Huigens, D., Winter, J., and N. Yutaka, "OpenPGP", Work in Progress, Internet-Draft, draft-ietf-openpgp-crypto-refresh-13, 4 January 2024, <<https://datatracker.ietf.org/doc/html/draft-ietf-openpgp-crypto-refresh-13>>.
- [RFC2045] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, DOI 10.17487/RFC2045, November 1996, <<https://www.rfc-editor.org/rfc/rfc2045>>.
- [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/rfc/rfc2119>>.

- [RFC3864] Klyne, G., Nottingham, M., and J. Mogul, "Registration Procedures for Message Header Fields", BCP 90, RFC 3864, DOI 10.17487/RFC3864, September 2004, <<https://www.rfc-editor.org/rfc/rfc3864>>.
- [RFC5083] Housley, R., "Cryptographic Message Syntax (CMS) Authenticated-Enveloped-Data Content Type", RFC 5083, DOI 10.17487/RFC5083, November 2007, <<https://www.rfc-editor.org/rfc/rfc5083>>.
- [RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <<https://www.rfc-editor.org/rfc/rfc5234>>.
- [RFC5322] Resnick, P., Ed., "Internet Message Format", RFC 5322, DOI 10.17487/RFC5322, October 2008, <<https://www.rfc-editor.org/rfc/rfc5322>>.
- [RFC5652] Housley, R., "Cryptographic Message Syntax (CMS)", STD 70, RFC 5652, DOI 10.17487/RFC5652, September 2009, <<https://www.rfc-editor.org/rfc/rfc5652>>.
- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/rfc/rfc8126>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/rfc/rfc8174>>.
- [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/rfc/rfc8551>>.

9.2. Informative References

- [chrome-indicators] Schechter, E., "Evolving Chrome's security indicators", May 2018, <<https://blog.chromium.org/2018/05/evolving-chromes-security-indicators.html>>.
- [CSS] World Wide Web Consortium, "Cascading Style Sheets Level 2 Revision 2 (CSS 2.2) Specification", 12 April 2016, <<https://www.w3.org/TR/2016/WD-CSS22-20160412/>>.

[HTML-ESCAPES]

W3C, "Using character escapes in markup and CSS", n.d.,
<<https://www.w3.org/International/questions/qa-escapes#use>>.

[I-D.autocrypt-lamps-protected-headers]

Einarsson, B. R., "juga", and D. K. Gillmor, "Protected Headers for Cryptographic E-mail", Work in Progress, Internet-Draft, draft-autocrypt-lamps-protected-headers-02, 20 December 2019,
<<https://datatracker.ietf.org/doc/html/draft-autocrypt-lamps-protected-headers-02>>.

[I-D.ietf-lamps-samples]

Gillmor, D. K., "S/MIME Example Keys and Certificates", Work in Progress, Internet-Draft, draft-ietf-lamps-samples-08, 2 February 2022,
<<https://datatracker.ietf.org/doc/html/draft-ietf-lamps-samples-08>>.

[I-D.pep-email]

Marques, H. and B. Hoeneisen, "pretty Easy privacy (pEp): Email Formats and Protocols", Work in Progress, Internet-Draft, draft-pep-email-02, 16 December 2022,
<<https://datatracker.ietf.org/doc/html/draft-pep-email-02>>.

[I-D.pep-general]

Birk, V., Marques, H., and B. Hoeneisen, "pretty Easy privacy (pEp): Privacy by Default", Work in Progress, Internet-Draft, draft-pep-general-02, 16 December 2022,
<<https://datatracker.ietf.org/doc/html/draft-pep-general-02>>.

[PGPCONTROL]

UUNET Technologies, Inc., "Authentication of Usenet Group Changes", 27 October 2016,
<<https://ftp.isc.org/pub/pgpcontrol/>>.

[PGPVERIFY-FORMAT]

Lawrence, D. C., "Signing Control Messages, Verifying Control Messages", n.d.,
<<https://www.eyrie.org/~eagle/usefor/other/pgpverify>>.

[RFC2047]

Moore, K., "MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text", RFC 2047, DOI 10.17487/RFC2047, November 1996,
<<https://www.rfc-editor.org/rfc/rfc2047>>.

- [RFC2049] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Five: Conformance Criteria and Examples", RFC 2049, DOI 10.17487/RFC2049, November 1996, <<https://www.rfc-editor.org/rfc/rfc2049>>.
- [RFC3156] Elkins, M., Del Torto, D., Levien, R., and T. Roessler, "MIME Security with OpenPGP", RFC 3156, DOI 10.17487/RFC3156, August 2001, <<https://www.rfc-editor.org/rfc/rfc3156>>.
- [RFC3851] Ramsdell, B., Ed., "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification", RFC 3851, DOI 10.17487/RFC3851, July 2004, <<https://www.rfc-editor.org/rfc/rfc3851>>.
- [RFC4021] Klyne, G. and J. Palme, "Registration of Mail and MIME Header Fields", RFC 4021, DOI 10.17487/RFC4021, March 2005, <<https://www.rfc-editor.org/rfc/rfc4021>>.
- [RFC5751] Ramsdell, B. and S. Turner, "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification", RFC 5751, DOI 10.17487/RFC5751, January 2010, <<https://www.rfc-editor.org/rfc/rfc5751>>.
- [RFC6376] Crocker, D., Ed., Hansen, T., Ed., and M. Kucherawy, Ed., "DomainKeys Identified Mail (DKIM) Signatures", STD 76, RFC 6376, DOI 10.17487/RFC6376, September 2011, <<https://www.rfc-editor.org/rfc/rfc6376>>.
- [RFC6532] Yang, A., Steele, S., and N. Freed, "Internationalized Email Headers", RFC 6532, DOI 10.17487/RFC6532, February 2012, <<https://www.rfc-editor.org/rfc/rfc6532>>.
- [RFC7489] Kucherawy, M., Ed. and E. Zwicky, Ed., "Domain-based Message Authentication, Reporting, and Conformance (DMARC)", RFC 7489, DOI 10.17487/RFC7489, March 2015, <<https://www.rfc-editor.org/rfc/rfc7489>>.
- [RFC8617] Andersen, K., Long, B., Ed., Blank, S., Ed., and M. Kucherawy, Ed., "The Authenticated Received Chain (ARC) Protocol", RFC 8617, DOI 10.17487/RFC8617, July 2019, <<https://www.rfc-editor.org/rfc/rfc8617>>.

Appendix A. Possible Problems with some Legacy Clients

When an e-mail message with end-to-end cryptographic protection is received by a mail user agent, the user might experience many different possible problematic interactions. A message with Header Protection may introduce new forms of user experience failure.

In this section, the authors enumerate different kinds of failures we have observed when reviewing, rendering, and replying to messages with different forms of Header Protection in different Legacy MUAs. Different Legacy MUAs demonstrate different subsets of these problems.

A conformant MUA would not exhibit any of these problems. An implementer updating their Legacy MUA to be compliant with this specification should consider these concerns and try to avoid them.

A.1. Problems Reviewing signed-and-encrypted Messages in List View

- * Unprotected Subject, Date, From, To are visible
- * Threading is not visible

A.2. Problems when Rendering a signed-and-encrypted Message

- * Unprotected Subject is visible
- * Protected subject (on its own) is visible in the body
- * Protected subject, date, from, to visible in the body
- * User interaction needed to view whole message
- * User interaction needed to view message body
- * User interaction needed to view protected subject
- * Impossible to view protected subject
- * Nuisance alarms during user interaction
- * Impossible to view message body
- * Appears as a forwarded message
- * Appears as an attachment
- * Security indicators not visible

- * User has multiple different methods to Reply: (e.g. reply to outer, reply to inner)
- * User sees English "Subject:" in body despite message itself being in non-English
- * Security indicators do not identify protection status of Header Fields
- * Header Fields in body render with local Header Field names (e.g. showing "Betreff" instead of "Subject") and dates (TZ, locale)

A.3. Problems when Replying to a signed-and-encrypted Message

Note that the use case here is:

- * User views message, to the point where they can read it.
- * User then replies to message, and they are shown a message composition window, which has some UI elements
- * If the MUA has multiple different methods to Reply: to a message, each way may need to be evaluated separately

This section also uses the shorthand UI:x to mean "the UI element that the user can edit that they think of as x."

- * protected subject is in UI:subject (and will leak)
- * protected subject is quoted in UI:body
- * protected subject is not anywhere in UI
- * message body is not visible/quoted in UI:body
- * user cannot reply while viewing protected message
- * reply is not encrypted by default (but is for normal S/MIME sign+enc messages)
- * unprotected From: is in UI:To
- * User's locale (lang, TZ) leaks in quoted body
- * Header Fields not protected (and in particular, Subject is not obscured) by default

A.4. Problems Reviewing signed-only Messages in List View

- * Unprotected Subject, Date, From, To are visible
- * Threading is not visible

A.5. Problems when Rendering a signed-only Message

- * Unprotected Subject is visible
- * Protected subject (on its own) is visible in the body
- * Protected subject, date, from, to visible in the body
- * User interaction needed to view whole message
- * User interaction needed to view message body
- * User interaction needed to view protected subject
- * Impossible to view protected subject
- * Nuisance alarms during user interaction
- * Impossible to view message body
- * Appears as a forwarded message
- * Appears as an attachment
- * Security indicators not visible
- * Security indicators do not identify protection status of Header Fields
- * User has multiple different methods to Reply: (e.g. reply to outer, reply to inner)
- * Header Fields in body render with local Header Fields (e.g. showing "Betreff" instead of "Subject") and dates (TZ, locale)

A.6. Problems when Replying to a signed-only Message

This uses the same use case(s) and shorthand as Appendix A.3.

- * Unprotected Subject: is in UI:subject
- * Protected Subject: is quoted in UI:body

- * Protected Subject: is not anywhere in UI
- * Message body is not visible/quoted in UI:body
- * User cannot reply while viewing protected message
- * Unprotected From: is in UI:To
- * User's locale (lang, TZ) leaks in quoted body

Appendix B. Test Vectors

This section contains sample messages using the different schemes described in this document. Each sample contains a MIME object, a textual and diagrammatic view of its structure, and examples of how an MUA might render it.

The cryptographic protections used in this document use the S/MIME standard, and keying material and certificates come from [I-D.ietf-lamps-samples].

These messages should be accessible to any IMAP client at `imap://bob@header-protection.cmrg.net/` (any password should authenticate to this read-only IMAP mailbox).

You can also download copies of these test vectors separately at <https://header-protection.cmrg.net>.

If any of the messages downloaded differ from those offered here, this document is the canonical source.

B.1. Baseline Messages

These messages offer no header protection at all, and can be used as a baseline. They are provided in this document as a counterexample. An MUA implementer can use these messages to verify that the reported cryptographic summary of the message indicates no header protection.

B.1.1. No Cryptographic Protections Over a Simple Message

This message uses no cryptographic protection at all. Its body is a text/plain message.

It has the following structure:

text/plain 152 bytes

Its contents are:

IFBLQ1MjNyBzaWduZWREYXRhLiAgVGh1DQpwYX1sb2FkIGlzIGEgdGV4dC9wbGFp
biBtZXNzYWdlLiBjZCB1c2VzIG5vIGh1YWR1ciBwcm90ZWN0aW9uLg0KDQotLSAN
CkFsaWN1DQphbGljZUBzbWltZS5leGFtcGxlDQqgggemMIIDzCCAreAwIBAgIT
Dy0lvRE5l0rOQ1SHoe49NAaKtDANBgkqhkiG9w0BAQ0FADBVMQ0wCwYDVQQKEwRJR
RVGRMREwYDVQQLLEwhMQU1QUyBXRzExMC8GA1UEAxMoU2FtcGx1IEExBTBTIFJT
QSBDDXJ0aWZpY2F0aW9uIEF1dGhvcml0eTAGFw0xOTExMjAwNjU0MThaGA8yMDUy
MDkyNzA2NTQxOFowOzENMASGA1UEChMESUVURjERMA8GA1UECXMITEFNUFMgV0cx
FzAVBgNVBAMTDkFsaWN1IEExvdmVsYWN1MlIBIjANBgkqhkiG9w0BAQEFAAOCAQ8A
MIIBCgKCAQEAAmpUp+ovBouOP6AFQJ+RppwODxxzY60n1lJ53pTeNSiJlWkwtw/cx
Qq0t4uD2vWYB8gOUH/CVt2Zp1c+auzPKJ2Zu5mY6kHm+hVB+IthjLeI7Htg6rNeu
Xq50/TuTSxX5R1I1EXGt8p6hAQVeA5oZ2afHg4b97enV8gozR0/Nkug4AkXmbk7T
HNc8vvjMUJanZ/VmS4TgDqXjWShplcI3lcvvBZMswt41/0HJvmswqps6oQcAx3We
ag0yCNj1V9V9yu/3DjcYbwW2lJf5NbMHbM1LY4X5chWfNEbkN6hQury/zxnlsukg
n+fHbcqvwDhJLAqFpW/jA/EB/WI+whUpqtQIDAQABo4GvMIGsMAwGA1UdEwEB/wQC
MAAwFwYDVR0gBBADjAMBgpghkgBZQMCAATABMB4GA1UdEQQXMBWBE2FsaWN1QHNT
aW1lLmV4YW1wbGUwEwYDVR0lBAwwCgYIKwYBBQUHAWQwDgYDVR0PAQH/BAQDAgUg
MB0GA1UdDgQWBBSiU0HVRDyAKRV8ASPw546vzfN3DzAfBgNVHSMEGDAWgBSRMI58
BxcMp/EJKGU2GmccaHb0WTANBgkqhkiG9w0BAQ0FAAOCAQEAgU14oJyxMpwWpAYl
OvK6NEbM1lgD5H14EC4Muxqlu0q2XgXOSBHI6DFX/4LDsfx7fSIus8gWVY3WqMeu
OA7IizkBD+GDEu8uKveERRXZncxGwy2MfbH1Ib3U8QzTjqB8+dz2AwYeMxODWq9o
pwtA/lTOkRg8uuivZfg/m5fFo/QshlHNaatDVEXsU4Ps98Hm/3gznbvhdjFbZbi4
oZ3tAadRLE5K9JiQaJYOnUmGpFB8PPwDR6chMZeegSQAW++OIKqHrg/WEh4yiuPf
qmAvX2hzkPpivNJYdTPUXTSO7K459CyqbqG+sNo02kc1nTX185RHNrVKQK+LOYWY
1Q+hWDCCA88wggK3oAMCAQICEzdBBXntdX9CqaJcOvT4as6aqdcwDQYJKoZIhvcN
AQENBQAwVTENMASGA1UEChMESUVURjERMA8GA1UECXMITEFNUFMgV0cxMTAvBgNV
BAMTKFNhbXBzS2BMQU1QUyBSU0EgQ2Vydg1maWNhdGlvbiBBdXR0b3JpdHkwIBcN
MTkxMTIwMDY1NDE4W4hgPMjA1MjA5MjcwNjU0MThaMDsxDTALBgNVBAoTBELFVEYx
ETAPBgNVBAsTCEExBTBTIFdHMRcwFQYDVQQDEw5BbGljZSBMbz3ZlbGFjZTCCASIw
DQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBALTOiehYOBY+TZp/T5K2KNI05Hwr
+E3wP6XTvvi6WwYtgBK9LCOWI2juwdRrjFBSXkk7pWpjXwsA3A5G0tz0FpfgyC70
xsVcF7q4WHWZwleYXFK1QHJD73nQwXP968+A/3rBX7Ph00DBbZnfitOLPpEwjTt
dg0VQQ6Wz+CRQ/YbHPKaw7aRphZO63dKvIKp4cQVtkWQH16syTjGsgkLcLNau5LZ
DQUdsGV+SAo3nBdWCRYV+I65x8Kf4hCxxqmjV3d/2NKRu0BXnDe/N+iDz3X0zEoj
0fqXgq4SWc0nsG1llyXt1TL270I6ATKRGJWiQVCCpDtc0NT6vdJ45bCSzsCAwEA
AaOBrzCBrdAMBGNVHRMBAf8EAJAAMBcGA1UdIAQOMA4wDAYKYIZIAWUDAgEwATAe
BgNVHREEFzAVGRNhbGljZUBzbWltZS5leGFtcGxlMBMGAlUdJQQMMAoGCCsGAQUF
BwMEMA4GA1UdDwEB/wQEAwIGwDAdBgNVHQ4EFgQUu/bMsi0dBhIc164papAQ0yBm
ZnMwHwYDVR0jBBGwFoAUKTCOfAcXDKfxcSh1NhpNHGh29FkwDQYJKoZIhvcNAQEN
BQADggEBAHOJoJanzqmgasN3/gqSQ4cbbmdj/R40BEP+r+gXT+xiidfZ2iLNwYyTn
euK6AChwKfnNvOFb8lV1iffRtF/KtmVEDMR/sYeqAH83KM5p3e12lVh4OHhyI0qN
uz5oShNaACsioQ23WxHGvy9vsdVfnbhsplrWg9NQ2WbpCmK+2oMh2oYl0Z/wvXmt
9cG6jbMvcdH4z0IOvg6mrYkKTM/RCGnumghxwYToj1OyD5Gs4D2IJCw+fx5ODxh5
2MbNRYXTus2ZPRPM8JXNQC4Gwv4km3M4rKnJDd6hnoQ9rNeozIcBVyybQYjfrgg4
DRvw9Ksk22OH4ConlB8f7R7s1LM2cSYxggIAMIIB/AIBATBsMFUxDALBgNVBAoT
BELFVEYxETAPBgNVBAsTCEExBTBTIFdHMTewLwYDVQQDEyhTYW1wbGUgTEFNUFMg
U1NBIEEN1cnRpZmljYXRpb24gQXV0aG9yaXR5AhM3QQV57XV/QqmiXDr0+GrOmqnX
MASGCWCGSAF1AwQCAaBpMBGCSqGSIB3DQEJAzELBgkqhkiG9w0BBwEwHAYJKoZI
hvcNAQkFMQ8XDTIxMDIyMDE1MDEwLwYJKoZIhvcNAQkEMSIEIESMi+9/LULD

```
fGjj+6U50VNLfxbzvvyVJ0wzwnTS114DyMAOGCSqGSib3DQEBAQUABIIBACJHeayB
U1lC4GdcgdojTUjoeLy6UIbrSg/aKZgAkCB8Dwq0hdU10qiun6WKI/TxM5izpRvL
UsNBGmqknPBMFhvWx6KCrwFk0p0j5Y5DZqX30deiQiGTUv3NiwZGTrKJ3JkyymFO
HGbe5Thrq3inRLVfileuIZewaJsnJhKfnEq9fS09icTJ5o1PDAH6mZbW6hpYmU3F
KBk2qJNqJX6bo60rCogu3wXDj0wxnqEXmeNDH5/+L9UVZur+EWzviUc8Ldd/kP3L
DOO7ivs10bAWe8Tbw7NjuP8Z1Vzcvj3nXWzZzxh2ymDIOvyJA+t0LHQvsN/fbdW
fC6Pm51fEkabbmw=
```

B.1.3. S/MIME Signed-only multipart/signed Over a Simple Message, No Header Protection

This is a signed-only S/MIME message via PKCS#7 detached signature (multipart/signed). The payload is a text/plain message. It uses no header protection.

It has the following structure:

```
multipart/signed 4191 bytes
  text/plain 224 bytes
  application/pkcs7-signature [smime.p7s] 3429 bytes
```

Its contents are:

```
MIME-Version: 1.0
Content-Type: multipart/signed;
  protocol="application/pkcs7-signature"; boundary="052";
  micalg="sha-256"
Subject: smime-multipart
Message-ID: <smime-multipart@lhp.example>
From: Alice <alice@smime.example>
To: Bob <bob@smime.example>
Date: Sat, 20 Feb 2021 10:02:02 -0500
User-Agent: Sample MUA Version 1.0
```

--052

```
MIME-Version: 1.0
Content-Type: text/plain; charset="utf-8"
Content-Transfer-Encoding: 7bit
```

This is the smime-multipart message.

This is a signed-only S/MIME message via PKCS#7 detached signature (multipart/signed). The payload is a text/plain message. It uses no header protection.

--

```
Alice
alice@smime.example
```

--052

Content-Transfer-Encoding: base64

Content-Type: application/pkcs7-signature; name="smime.p7s"

MIIJ4AYJKoZIhvcNAQcCoIIJ0TCCc0CAQEExDTALBg1ghkgBZQMEAgEwCwYJKoZIhvcNAQcBoIIHpjCCA88wggK3oAMCAQICEw8tJb0ROZdKzkJU6HuPTQGirQwDQYJKoZIhvcNAQENBQAwVTENMAsGA1UEChMESUVURjERMA8GA1UECXMITEFNUFNgV0cxMTAvBgNVBAMTKFNhbXBsZSBMQU1QUyBSU0EgQ2VydG1maWNhdG1vbiBBdXR0b3JpdHkwIBcNMtKxMTIwMDY1NDE4WhgPMjA1MjA5MjcwNjU0MThaMDsxDTALBgNVBAoTBELFVEYxETAPBgNVBAsTCEExBTBVTIFdHMRcwFQYDVoQDEw5BbG1jZSBMbz3ZlbGFjZTCCASIdDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAJqVKfLwaLjj+gBUCfkacKTg8cc20tJ9Zsed6U3jUoiZVpMLcP3MUKtLeLg9r1mAfID1B/w1bdmadXPmrszyidmbuZmOpB5voVQfiLYYy3iOx7YOqzXr16udP07k0sV+UdSNRfxrfKeoQEFXgOaGdmnx4OG/e3p1fIKM0dPzZLoOAJF5m500xzXPL74zFCWp2f1ZkuE4A6141koaZXCN5XL7wWTLMLenF9Byb5ksKqUuqEHAMd1nmoNmgy9VfVfcrv9w43GG8FtpSX+TWzB2zNS2OF+XIVnzRG5DeoULq8v88Z5bLpIJ/nx26r8A4SSwIBaVv4wPxAf1iPsIVK arUCAwEAAaOBrzCBrdAMBgNVHRMBAf8EAjAAMBcGA1UdIAQQMA4wDAYKYIZIAWUDAgEwATAeBgNVHREEFzAVgRNhbG1jZUBzbW1tZS5leGFtcGx1MBMGA1UdJQQMMAoGCCsGAQUFBwMEMA4GA1UdDwEB/wQEAWIFIDAdBgNVHQ4EFgQUo1NB1UQ8gCkVfAEj8OeOr83zdW8wHwYDVR0jBBGwFoAUkTCOFacXDKfXCSh1NhpHGh29FkwDQYJKoZIhvcNAQENBQADggEBAIFJeKCCsTKcFqQMpTryuJRgzJdYA+R9eBAuDLsatbtKt14FzkgRyOg3l/+Cw7H8e30iLrPIf1WN1qjHrjgOyIs5AQ/hgxLvLir3hEUV2Z3MRsMth2x9SG91PEM046gfpnc9gMGHjMTg1qvaKcLQP5UzpeYPLror2X4P5uXxaP0LIZRzWmkw1RF7FD7PFB5v94M5274Xyxw2W4uKGD7QGnUZROSvSYkGiWdp1JhqXwfdz8A0enITGXnoEkaFvVjiCqh64PlhIeMorj36pgL19oWZD6YrzSWHuz1F00juyuOfQsqm6hvrDTqNpHNZ015fOURza1SkCvi9GFmNUPoVgwgPpMIct6ADAgECAhM3QQV57XV/QqmiXDr0+GrOmgnXMA0GCSqGSIb3DQEBDQUAMFUxDTALBgNVBAoTBELFVEYxETAPBgNVBAsTCEExBTBVTIFdHMTEwLWYDVoQDEyYTYW1wbGUgTEFNUFNgV1NB1EN1cnRpZmljYXRpb24gQXV0aG9yaXR5MCAxdTE5MTEyMDA2NTQxOFoYDzIwNTIwOTI3MDY1NDE4WjA7MQ0wCwYDVoQKEwRJRVRGMREwDwYDVoQLEwhMQU1QUyBXRzEXMBUGA1UEAxMOQWxpY2UgTG92ZWxhY2UwgGEMa0GCSqGSIb3DQEBQUAA4IBDwAwggEK AoIBAQC09InoWDgWPk2af0+StijSNOR8K/hN8D+1078oullsk4ASvSwjsCNo7sHU a4xQU15J06VqY18LANwORjrc9BaX4MguzsbFXBe6uFh1mVpXmFxSpUByQ+950MFz /evPgP96wV+z4TtAwW2Z34rTiz4DxMI07XYNFUEOlS/gkUP2GxzymsO2kaYWTut3 SryCqeHEfBzFk4urMk4xrIJC3CzWruS2Q0FHbBlfkgKN5wXVgkWFfiOucfCn+IQ saqp0ld3f9jSkbtAV5w3vzfog8919MxKI9H614KuElnAtJ7BtZcs17duY9u9COgE ykRiVokFQgqQ7XNDU+r3SeOWwks7AgMBAAGjga8wgawwDAYDVR0TAQH/BAIwADAX BgNVHSAEEDAOMAwwGCMGSAF1AwIBMAEwHgYDVR0RBBCwFYETYWxpY2VAc21pbWUu ZXhhbXBsZTATBgNVHSUEDDAKBggrBgEFBQcDBDAOBgNVHQ8BAf8EBAMCBsAwHQYD VR0OBByEFLv2zLItHQYSHJeuKWqQENMgZmZzMB8GA1UdIwQYMBaAFJEWjnwHFwyn 8QkoZTYaZxxodvRZMA0GCSqGSIb3DQEBDQUAA4IBAQBziaI2p86poGkjd/4KkkOH G25nY/0eNARD6/of0/sYonX2doizcGmK53riugAocCn5zbzhW/JVdYn30UxfyrZl RAzEf7GHqgB/NyjOad3pdpVYeDh4ciNKjbs+aEoTWgAkoqEnt1sRx1cvb7HVX524 bKZa1oPTUNlm6QpivtqDIdqGJdGf8L1zLFXBu02zL3HR+M9CDr40pq2JcKzP0Qhp 7poIccGE6I9Tsg+Rr0A9iCQsPn1+Tg8YedjGzUWF07rNmT0TzPCVzUAuBlr+JJtz OKypyQ3eoZ6EPazXqMyHAVcsm0GI364IOA0b8PSrJNtjh+AqJ5QfH+0e7NSzNnEm MYICADCCafwCAQEwbDBVMQ0wCwYDVoQKEwRJRVRGMREwDwYDVoQLEwhMQU1QUyBX RzExMC8GA1UEAxMoU2FtcGx1IEExBTBVTIFJTSQSDZlJ0aWZpY2F0aW9uIEF1dGhv

```

cm10eQITN0EFee1lf0Kpolw69PhqzpqplzALBglghkgBZQMEAgGgaTAYBqkqhkiG
9w0BCQMxCwYJKoZIhvcNAQcBMBwGCSqGSIb3DQEJBTEPFw0yMTAyMjAxNTAyMDJa
MC8GCSqGSIb3DQEJBDEiBCDAkJYhqvAHhprkzEWP6PweksoYhj5ULTLbcfQ9Tu3C
zDANBgkqhkiG9w0BAQEFAASCAQcJe818STb4M4utvQsdcQEH0CZR7I38uL5TSZF3
1lKmD9PuCDuV3GIkfdmZISKRuffBle1xANc2av/0Qogr7OaFF485DAONVAEIQ7ah
t94pwgAE4yvXXWkMFQkKidltnMXbnHADKWU0YC+BQkgd/5J3zg4ESeMwOUm0+b3C
GDaUBTIJhHfu9sqt7jXa7PbzQEfemYZORPI14/uZSs86SLkPvNGUpWb4mN6o1C0
2h/U4SCpq80y390oNM0VNpaa+nsTu5yOFc34pMlvjwCJyIOYPaDnvw9FYgr2oOp7
cdOgFcSJ8q7I+Tx2yg60VW8tAT7UBkifc37UuuVbnOsqeVB3

```

--052--

B.1.4. S/MIME Encrypted and Signed Over a Simple Message, No Header Protection

This is a encrypted and signed S/MIME message using PKCS#7 envelopedData around signedData. The payload is a text/plain message. It uses no header protection.

It has the following structure:

```

application/pkcs7-mime [smime.p7m] 6720 bytes
  (decrypts to)
  application/pkcs7-mime [smime.p7m] 3960 bytes
    (unwraps to)
    text/plain 239 bytes

```

Its contents are:

```

Content-Transfer-Encoding: base64
Content-Type: application/pkcs7-mime; name="smime.p7m";
  smime-type="enveloped-data"
Subject: smime-enc-signed
Message-ID: <smime-enc-signed@lhp.example>
From: Alice <alice@smime.example>
To: Bob <bob@smime.example>
Date: Sat, 20 Feb 2021 10:03:02 -0500
User-Agent: Sample MUA Version 1.0

```

```

MIITXAYJKoZIhvcNAQcDoIITTCCE0kCAQAxggMQMIIBhAIBADBbsMFUxDTALBgNV
BAoTBELFVEYxETAPBgNVBAsTCExBTvBTIFdHMTEwLwYDVQQDEyhTYW1wbGUgTEFN
UFMgU1NBIENlcnRpb24gQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIeh7j00
Boq0MA0GCSqGSIb3DQEBAQUABIIBAHmnSO2IdHZqhpStR4KWdgv3WQtCaxYUhXTJ
AmWV0NBvy5u7gilyKnpY7CcJ4T5bA68lWNos4i4D2bsiLDGtMAuEynCKejeKp+r
rS6BU+iI3QAruW8v4xxFHmYtOdge1tVluws7atc8fXnU1gcfpnOD+IvLOdwrJBs
o0AePTxqKmi3pUkSoz4FVkfXJNkM3KK1Xsqf5VFJV21r/AY+3w5V5sFkengnXv6e
kAZWUVMZ5GiiLzCk5412rGO3Wi5oC1cYqkbnmKndm2MvcwEosO48N6XTvW9geENp
y9stPxxv9pAp9HD4miuwWA2K1UPBVLh717XwjDwA08MGsRCzHP64wggGEAgEAMGww

```

VTENMAsGA1UEChMESUVURjERMA8GA1UECmIETFNUFMgV0cxMTAvBgnVBAMTKFNh
bXBsZSBMQU1QUyBSU0EgQ2VydGhmaWNhdG1vbiBBdXR0b3JpdHkCEzB8R0APhiY6
HGLS64Mv1sDXhpQwDQYJKoZIhvcNAQEBBQAEggEAdOnjKorhe+/7PA3sZPAMGBA6
bQ1RDw3HF8/5y4ld+ZCHw02YeGKvc4OT1TO4SsY8zdOhNBhJRaQqRkK+5HKOOPqV
ADA6a90U36FAyNI0Zn8veG4rH1b/vWHVdxWbOW69Liymia3fBz65o/6E1yX/GAb8
m+KPtKx9cvSFCazv95M4C3Girn8LkAswtmWR+deEp7tYPdjHky7TOkdXpV/z0Ee9
HtjillLeqUD+mvV3CJkIbywsUBRsZ0iLA8B9WoIsvcpYDU1biAxMko0rWlUfH2VSD
j6+Tj1W90dSZM7xUF1YefRDd9XnF+HcRNbO58ucu8iIMxVJq+LNBey4N70XmFjCC
EC4GCSqGSib3DQEHATAdbglghkgBZQMEAAIEEMyuzbDBN6Tv2WSNq2aSZ5WAghAA
nq1HK1EGkfDdd9BKbpZgRqgsSUEEBdGSgAC4v0Ugu6eD+ukLBk+TzZGuLHFj1vB3
/Nk6mjv4xakp/x23yGk7zc6bzmHduR27avvu9zZf8fdeNMkwBeuB47WIXENqKmlt
y6I7vVEJJE4CEVF0VDIFH7B2wpo5pogs0N3vJt/Gr2vAO2NjRosgGuHTRDXybQlY
KZKOCw2G0+vB1CYCP9YeM5gG7vQNirjQdVPJOK+4NOEHY8JZHQZvu7dR2P02/QiS
5p8wcYPSRLsWRdaPaBDnfkDTWaaQYUcm909iydoYUI5Xg33LzjGh0UMDg0vouQ/1
Aqj7zwHXfHJVJK381SQc8fL88/TaCkouGMAw/dHCUQYOB5v4JlsSaYBo8ojaPIIk
T6PYuFu01ghi56h21sKNsuhnYSR8c8rZMq3jIKDKmdjOpNpn6kevu1BHeNnH1wK
WPBiMx4CAapizFjeVmbgnFbjNBdw2kO55bPqXrHMOG5/hHC85JV/IgCF0uvQgOY/
kG2eT180pJ3dF3/iJnHsn6wB50UDPYAqXt9bpAgtNNd0iCyd5Gd3guQOCAfvpBOO
IkMPH2K8xsvqk6cUncEtrbSColr1dePnQhiTiwyAmJevan++mvjUuBRPN1grXH4v
AeCR28K+htOxC/5SaONcLX6FhppXOMR09j4n1wLWvVXfmm0Bo3eyaYqLAatmId1/
ig17gk0JQBw2zzZHqEm1URQh50r/6DvStMj2ASjGgtSPPhBQKO+CaITceLhuRNyw
cH3tSLeGmhMj01DT6gmB/d3PFcLjUx8DwCwYsshDY3Z15GrzIq1jgZvmzjBxaCuA
VPGA3jWMOwBdJtXhAP7uYCe5qjbtL9L6EqIo8RQl7zrXxp7etwSjbaFbTUKBxxik
AZKPAGoTFsO3cVhUBmSzoMupgiUAieTOOS43iP9JeXLFHOnN+cAlo7iJx/gEcL68
1ENpSaWRV00NBtF6vjpNIEh7eNOMCA/fTipRR7Pz+g2oKQLUZPNkVxUTi7PjoSPb
bFKpK0xbHqao40mJdNvX61ng73PsQnJGadYu6DnMvVG7oTibcsA3aoh3jreb1vLO
mzpATxg4b1QFC0Cjxqd8FKRxQZ1ync5cO5E3EhY1VXW0pi17wW/a2Ca7S8iT3+Rw
bVNd2A01JgS6r+NsvgIXQTjxA6RNzP3K1Iorkuhg6nNbqgJffskHz5uD72AXQc9J
OfxGIFAgNlbnr9u+pvj3WVqJLZTHFdDvvXPgza5/D3tnoWb83j8Z9T8px1TGK3m2
GVFm4CyJxdzDrOcfXznRO31YkNeTA01SySF0yhTHAZIOU81YaUT/2P4y28Fc/79w
ofFZSqvz+J2QcObZfbWsJ8RbrcaPYzPj0cBwTuxPyCni0Mf/4if+GxLv1F8a7DI
onHVJg5w+Lo1RKcvPpRIrq/w7rwfOhEehyQr6a/8WbiAOSMMRsQj3+9atQViPFb
QChAtGHq1TMWysVVGod4S3OhkiOsp1s6tOfCJb8QIL2DY1DSbg/wtnNbWA0Bxytf
tR1bhQRI0ytm7mhN01kfw+dWXPqzofRG/zvaKIGoufnmqJpbk4RR4r+KHUZ3xDP
2URkSh5Qrf9yZ7wE791QKomGSZygvX1Tp8TzicUWpeTQB0IHxsCg2JBTykU3q3m/
SV1NY16oP6oC1vVAzRNxQgs6TQ8PEgGqPse323VDCpgAnqsA5zq5zeZjjEK8p+Zy
HWjcaWf1top6+19Tt/5chnAmCk4wS120Lkisu7fOzB9M8UzQC0yVrJ4L1A/MD73Q
KE1zP92o87ZfJnnNjpbB4A/EcBTmhVxbjS1C4cT6UR08pv0cfhSqFni9eMhImQmS
0XST/0NkVeqBmC6b72fATGQb09Iv02pyV/2w5W04gCNCvWBN8kmQLEEHkDaOmZD
OYxGkgfbT00RxsC2fa8VnRuc8FyRjWFO9qWn8OTNhnVHbd3DPfsoTHN15v7dsGDz
0aOnVMmwSmAFfzQStA9qC+OPeBPXBCKNXd1Y7/7ru00GpUW9hSHKkOc227QtbtTAH
LdUAW1bBIPA3gNJQDKmGQaeFVfJDV8xn9v/1RuVxegh4N8QIK1U9IPz7+wec81S/
4cXz/JT01u/oGpcSE86jzarGMh/ik3ovcckGLvH7q7TdT5BdOYyZza6PcinfkT1Tj
rj/SMsHH3a1XNipnSnb+50dEIQUJksSgQYE1nFgv2M9PBONy3YA07Z2ArF/f0sEf
hRKQw9YH9grv0beRAOC5182tvvKrZ5j0q6gTtYZ8PacoD9DnaXJjNGKJ01jwNsmV
v1Px7G8yOuxx2qUuTBbqr8jHg7XR9/UaYEuvmDs1QZpnuDMOrxuRPufI1nWVZVd7
wxWd588fI3XOXmE9ZA2/kq5uq57xpoRL1Ph/sVqVysj9ruYTU7uHz629jFeq5mF4
iIpa80hPVJyC4gDtKLqF8Jb8VVKb4kdbTph6+pcRwnqIj6pEZq4G8FvquntzNn0o

8ydpnyZVV/bu+Py7MYq8YtkcEVvIk70b9gBI3UhKEL1PfRj/t/q0XM2C63a+c93j
 YpMSCnb/w0lpy9Ws5VMCISKsDYQLdKwNjj/aYWiHfgyghXGSY8/KDL18Yyzfqz2n
 zaOUaFMS7TMvHSjTe6Cv0zIYvht8P6gQmXVvEOLJ1VWUu+q3ccXnW5EHg4CgIbCI
 dm5iN3a+OlIejFQSZvFW4kB/RWNsOiyBextmOxxyAmu7xGayLZul/bzBFT5XrQwv
 sb524bGOYs6zcKA5zjnkQY215aGztAXFuMkI2nRiUsve5ARm/KQhbl2NGthQu++2
 r807AnZGdjhGlz4h5XfR/VvmjuMF/LxdgIJG31VC37u/343lgNbIOWybUorzFaeg
 rVnSDvMrfzMdZ/KRLTBhVUC9KFj1hn4L7FdfpWz3LbcW5Kn+uIU6EsRkbdOwdRPN
 mEPHgjT/+PD+msMoxTC0kaPtgRgB39I5jnIgpBAO8iKtObHttmZoZeqD5+N2uTyK
 WB+tC1CctNGGyFCR+YAUMTojhoulFSwiJIBTTE7QmSueuLmrEuCYvxUdEuA7RtTd
 LO1Abt0S05WURWu0pNDFroYbYPEjX5vEoFbU5jHhzEZF5WQ3cy+/EqMkxk7/47dh
 ux/J9UXXJTyT4Sh8KNZOPh381cVliqIO/Ms4Nn859zwaFCAKBZxn6ZqFQbBmxZWu
 D8ejB8KFxUIUp9H6wSPWvxJ2XW8By01UuZFI6vzvunm55eYvotkhjQFIag6CzOH
 CaUZfwJ6bEWreih41WFghnRL1zhRptnfQhnsKKVUqJW0jiaGZNZC+4jVCO+36bo
 W9e6LYfKemtKEMer/nrdgvW9LXo2CaL4BNgReK+T4ZkQbyob/2/ADN3mYe+ETBF8
 m71bfeIx73e87xNY2mWhvNMA1/hz041IJQdPySNwi5V9YE2/cS+6UuLfOVIyxING
 DpixonTJroJ6GeKotBn/K5eCqxKoF3gKiH98DnH9NV1otBej74998NG6ATN5jpaZ
 C46LiTJpMZpTx91EyasuT6eDW+1EGa6EWy1C7x7zjjjwan1qD2mM1NpnSm8L1oB3
 vvcwP60GoLgyu50+MOC+hYxrNuyCG2aoX6bvzdFrh9DyLl8LEErVdOPj9r/hOMtB
 PJzmiDqHIYaZv6+uyarrjFRG6d0+kCZDtzuAy/HEU+UXCuv27i99gkEyeMcasQSp
 DkRjvnVJQ101fMx/ttIGyyUbTH/jlBmLQ0cc+hrBeGGTYyKM5N6eB5WCukYSkfva
 6p7zGiKUER1py0Zmc04BN3UqPR6P9pJbJ0cNhpCTx7/pKa9OgDpT8+Ma1RxaNOLK
 msKkQpqnJf+2ays9Rv0oYtbnFvZJJPRT8iVg1D3aFwmCop0M1/kW5sYfDpPFgsH
 byzTzq3Fjw0AQ5UOG5Qq8EpsAlAJ3hy/5Vv4OaVizAoJz2fZxNq9Bw00lud/outL
 ZbRUEC72vJewbIAS11zdzJ7RLlpSMvB48/ca2dgeXqqfnvnAsMzgOIlaf1VID9H4m
 /KtMJfKPKagrka91wFwLECu207zihtHmRbkkWlrsqwA4SyumWfR5AEGW/sZ8g9LA
 rugrt/se6SpyYi5zzYL9/vNT61kQVY7UhuQcasQU+1CLVuaplAK4uvRs088wXYKn
 SSQXesmy5m6eYOIevOmyUMQzzfwKswT49j/7hrHsEctzpyCOP0/8zBgGH8f/wglr
 /sZ/O+sZnu819qUaJhHSFIEx/CQKuHYv5ez6aT3BAtmPn0iWrFVzna3Ogo8XAL68
 eDwN69Qm82ikDO2LFkKZrBzn/1dyZs/dT61QYpsmhxJzoluZzW/sYFeOCX6fWs7n
 fcrz9yMIDkvj70JrZp5jPRghFKHmqo5xh39TmeTsQFp2B8U1GD9YK6YfgSEAgbyL
 3BpUjZn/713jmwYHzGvEQfx7vP3SaZBMZ4GSCoeBT2grQoUDE575H7UDJsmRVJ04
 bo7iTWPZ1LdIC+oifedAhGhCoom+tApUYj+3BH1xIAZJMCGARqgyKcnvjw5WVu3
 fDna+4xJdNs0YK1uBkr6N9FBDfmQIuneIsQHAM71ZfucdlFenZhy1zNreqgls9QO
 NncRN1ltqmT2qmERXw8/HwcwNjR8FWRwbCCApsMgAZ0xWaRxpEct5lnGNbBpp1En
 BrMafVecU1QgwaljchA5ZiOuaZxizilPr9/eaX93aa2u+6OpsyPqdadxwDeV1Do
 4dg2NrDqQMFo3I1IcAdEzEcEqP8PV0tYjEeFZYsE0k3Qmcti+RuRj/rNTaxQ2Xw
 VkgL1BG8P0kxw0pVIKvyevcPtUD5tSlTxfp4qBF1EY/yrGCHy36q2mboBcRyYQry
 oBnsvoEfrIE8FEz1rOJVM+HN2udrKVJZzEPySf1ZvbDzxINcqDu09r3UO+L+ywW5
 9/ncHCMyoa0KbQ08q9i8VsGchL2FF5Q66g7I8U9u7R7V4Fz8RvLOzs6bB/Oh7+Z9
 0dTWreRYp9/82pQ0VSuvkWYiSPwiy37spaE8uALD5MvZOS3CqOwGI+o45uLBP/a6
 dgalPv1kThe8/a25+FqiQP6boCsN9wgA+T3v3kRFibzFEtyqX8C6Vu795PpycZ14
 /RGFTm2Df/U38DN/mlNhGgM6gMQr1YuSPieFJ+0/ctzGpSaS835d+DkQVvS3zT3/
 5EpybkOZrqr6erhNTVa8Onr3ZNdt9QyNUCmwxpYVvV2exwoVfcIjQgCxwehySLW5
 UprvrRNgho0OBMH+UmSggBfT7/omejxHgAJz5WCL/P+DiQ/dZcBK1OCRh1ZkocLB
 WvpunKTMuLyqSqNG87nzXAgFCLYQRWeCQNCItSbJ4aed+sJIYxmEm2UzyKAK9eXI
 dCZ/5fHOTmMD1645r/v9eSjeZd7Ed6MhGladuV1Nm9D129sIzKcUu3zfZAqBlzFK
 1RzPS3IUeM2VEJbK9AowEQ==

B.1.5. No Cryptographic Protections Over a Complex Message

This message uses no cryptographic protection at all. Its body is a multipart/alternative message with an inline image/png attachment.

It has the following structure:

```
multipart/mixed 1406 bytes
  multipart/alternative 794 bytes
    text/plain 206 bytes
    text/html 304 bytes
    image/png inline 232 bytes
```

Its contents are:

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="c39"
Subject: no-crypto-complex
Message-ID: <no-crypto-complex@lhp.example>
From: Alice <alice@smime.example>
To: Bob <bob@smime.example>
Date: Sat, 20 Feb 2021 12:00:02 -0500
User-Agent: Sample MUA Version 1.0
```

--c39

```
MIME-Version: 1.0
Content-Type: multipart/alternative; boundary="05a"
```

--05a

```
Content-Type: text/plain; charset="us-ascii"
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit
```

This is the no-crypto-complex message.

This message uses no cryptographic protection at all. Its body is a multipart/alternative message with an inline image/png attachment.

--

```
Alice
alice@smime.example
--05a
Content-Type: text/html; charset="us-ascii"
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit
```

```
<html><head><title></title></head><body>
```

```

<p>This is the <b>no-crypto-complex</b> message.</p>
<p>This message uses no cryptographic protection at all. Its body
is a multipart/alternative message with an inline image/png
attachment.</p>
<p><tt>-- <br/>Alice<br/>alice@smime.example</tt></p></body></html>
--05a--

```

```

--c39
Content-Type: image/png
Content-Transfer-Encoding: base64
Content-Disposition: inline

```

```

iVBORw0KGgoAAAANSUUhEUgAAABQAAAAUCAYAAACNiR0NAAAACe1EQVR42uVTOxbA
MAgS739nO3TpRw20dqpbFARQEjOywiwYnCtkDKnbcLk66sqlT+zt9cidkE+6KwkZ
sgrzfcqVMpL2jo0447gYDpeArk+OnJHkIhAfTPRicihAf5YJrw7vjv0ZWRWM/uli
vdPf1QZ2kDD9xppd8wAAAAABJRU5ErkJggg==

```

```

--c39--

```

B.1.6. S/MIME Signed-only signedData Over a Complex Message, No Header Protection

This is a signed-only S/MIME message via PKCS#7 signedData. The payload is a multipart/alternative message with an inline image/png attachment. It uses no header protection.

It has the following structure:

```

application/pkcs7-mime [smime.p7m] 5249 bytes
  (unwraps to)
  multipart/mixed 1288 bytes
    multipart/alternative 882 bytes
      text/plain 258 bytes
      text/html 353 bytes
      image/png inline 236 bytes

```

Its contents are:

```

Content-Transfer-Encoding: base64
Content-Type: application/pkcs7-mime; name="smime.p7m";
  smime-type="signed-data"
Subject: smime-one-part-complex
Message-ID: <smime-one-part-complex@lhp.example>
From: Alice <alice@smime.example>
To: Bob <bob@smime.example>
Date: Sat, 20 Feb 2021 12:01:02 -0500
User-Agent: Sample MUA Version 1.0

```

MIIPHwYJKoZiIhvcNAQcCoIIPEDCCDwwCAQExDTALBglghkgBZQMEAgEwgGVIbGkq
hkiG9w0BBwGgggU5BIIFNU1JTUUtVmVyc2l1vbjogMS4wDQpDb250ZW50LVR5cGU6
IG11bHRpcGFydC9taXhlZDsgYm91bmRhcnc9IjMzZSINCgOKLS0zM2UNck1JTUUt
VmVyc2l1vbjogMS4wDQpDb250ZW50LVR5cGU6IG11bHRpcGFydC9hbHRlcm5hdG12
ZTsgYm91bmRhcnc9ImUwYiINCgOKLS1lMGINckNvbnRlbnQtVH1wZTogdGV4dC9w
bGFpbjsgY2hhcnNldD0idXMtYXNjaWkiDQpNSU1FLVZlcnNpb246IDEuMAOKQ29u
dGVudC1UcmFuc2Zlci1FbmnVzGluZzZogN2JpdAOKDQpUaGlzIGlzIHRoZSBzbWlt
ZS1vbWUtCgFydC1jb21wbGV4IG11c3NhZ2UuDQoNClRoXMGaXMGYSBzaWduZWQt
b25seSBTL01JTUUgbWVzc2FnZSB2aWEgUeTdUyM3IHNpZ25lZERhdGEuICBUaGUN
CnBheWxvYWQgaXMGYSBtdWx0aXBhcnQvYWx0ZXJvYXRpdmUgbWVzc2FnZSB3aXR0
IGFuIGlubGluZQ0KaWlhZ2UvcG5nIGF0dGFjaG11bnQuIE10IHVzZXMGbmg8gaGVh
ZGVyIHByb3RlY3Rpb24uDQoNci0tIAOKQWxpY2UNCmFsaWNlQHntaW1lLmV4YW1w
bGUNCi0tZTBiDQpDb250ZW50LVR5cGU6IHRleHQuaHRtbdsgY2hhcnNldD0idXMt
YXNjaWkiDQpNSU1FLVZlcnNpb246IDEuMAOKQ29udGVudC1UcmFuc2Zlci1FbmnV
ZGluZzZogN2JpdAOKDQo8aHRtbd48aGVhZD48dG10bGU+PC90aXRzZT48L2hlYWQ+
PGJvZk+DQo8cD5UaGlzIGlzIHRoZSA8Yj5zbWltZS1vbWUtCgFydC1jb21wbGV4
PC9iPiBtZXNzYWdlLjwvcD4NCjxwP1RoXMGaXMGYSBzaWduZWQtb25seSBTL01J
TUUgbWVzc2FnZSB2aWEgUeTdUyM3IHNpZ25lZERhdGEuICBUaGUNCnBheWxvYWQg
aXMGYSBtdWx0aXBhcnQvYWx0ZXJvYXRpdmUgbWVzc2FnZSB3aXR0IGFuIGlubGlu
ZQ0KaWlhZ2UvcG5nIGF0dGFjaG11bnQuIE10IHVzZXMGbmg8gaGVhZGVyIHByb3Rl
Y3Rpb24uPC9wPgOKPHA+PHR0Pi0tIDxici8+QWxpY2U8YnIvPmFsaWNlQHntaW1l
LmV4YW1wbGU8L3R0PjwvcD48L2JvZk+PC9odG1sPgOKLS1lMGItdQ0KQotLTMz
ZQOKQ29udGVudC1UeXB1OiBpbWFnZS9wbmcNckNvbnRlbnQtVHJhbnNmZXItRW5j
b2Rpbmc6IGJhc2U2NAOKQ29udGVudC1EaXNwb3NpdG1vbjogaW5saW5lDQoNcm1W
Qk9SdzBLR2dvQUFBQU5TVWhFVWdBUFCUUFBUFBQ0FzQUFBQ05pUjBOUFBQ0NF
bEVRV1I0MnVWVE94YkENCk1BZlM3MzluTzNUcFJ3MjBkZXBiZkF5UUVqT3l3aXZ
bkN0a0RLbMjJGts2NnNxbFQrenQ5Y2lka0UrNkt3a1oNcnNncnmpY3FwTXBMmpv
MDQ0N2dzRHBlQXJrK09uSkhrSWhBZlRQUmljaWhBZjVZSnJ3N3ZqdjBaV1JXTS9l
bGkNcnZkUGYxUVoya0REOXhwcGQ4d0FBQUFCSlJvNUVya0pnZ2c9PQOKDQotLTMz
ZS0tDQqgggemMIIDzzCCAregAwIBAgITDy01vRE5l0rOQ1SHoe49NAaKtDANBgkq
hkiG9w0BAQ0FADBVMQ0wCwYDVQQKEWRJRVRGMREwDwYDVQQLLEwhMQU1QUyBXRzEx
MC8GA1UEAxMoU2FtcGx1IEExBTBVTIFJTQSBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0
eTAqFw0xOTExMjAwNjU0MThaGA8yMDUyMDkyNzA2NTQxOFowOzENMA8GA1UEChME
SUVURjERMA8GA1UECxMITEFNUFMgV0cxZzAVBGNVBAITDkFsaWNlIEExvdmVsYWNl
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAAmpUp+ovBouOP6AFQJ+Rp
wpODxxxY60n1lJ53pTeNSiJlWkwtw/cxQq0t4uD2vWYB8gOUH/CVt2Zp1c+auzPK
J2Zu5mY6kHm+hVB+IthjLeI7Htg6rNeuXq50/TuTSxX5R1I1EXGt8p6hAQVeA5oZ
2afHg4b97enV8gozR0/Nkug4AkXmbk7THNc8vvjMUJanZ/VmS4TgDqXjWShp1cI3
lcvvBZMswt41/0HJvmswqps6oQcAx3Weag0yCNj1V9V9yu/3DjcYbwW2lJf5NbMH
bm1LY4X5chWfNEbkN6hQury/zxn1sukgn+fHbqvwDhJLAgFpW/jA/EB/WI+whUpq
tQIDAQABo4GvMIGsMAwGA1UdEwEB/wQCMAAFwYDVR0gBBAwDjAMBgpghkgBZQMC
ATABMB4GA1UdEQQXMBWBE2FsaWNlQHntaW1lLmV4YW1wbGUwEwYDVR0lBAwwCgYI
KwYBBQUHAWQwDgYDVR0PAQH/BAQDAgUgMB0GA1UdDgQWBBSiU0HVRDyAKRV8ASPw
546vzfN3DzAfBgNVHSMEGDAWgBSRMI58BxcMp/EJKGU2GmccaHb0WTANBgkqhkiG
9w0BAQ0FAAOCAQEAgU14oJyxMpwWpAy10vK6NEbM1lgD5H14EC4Muxqlu0q2XgXO
SBHI6DFx/4LDsfx7fSIus8gWVY3WqMeuOA7IizkBD+GDEu8uKveERRXZncxGwy2M
fbH1Ib3U8QzTjqB8+dz2AwYeMxODWq9opwA/1TOKRg8uuivZfg/m5ffo/QshlHN
aaTDVEXsU4Ps98Hm/3gznbvhdjFbZbi4oZ3tAadR1E5K9JiQaJYOnUmGpFb8PPwD

R6chMZeegSQAW++OIKqHrg/WEh4yiuPfqmAvX2hZkPpivNJYdTPUXTS07K459Cyg
 bqG+sNo2kc1nTX185RHNrVKQK+L0YWY1Q+hWDCCA88wggK3oAMCAQICEzdBBXnt
 dx9CqaJcOvT4as6aqdcwDQYJKoZIHvcNAQENBQAwVTENMAsgA1UEChMESUVURjER
 MA8GA1UECxMITEFNUFMgV0cxMTAvBgNVBAMTKFNhbXBsZSBMQU1QUyBSU0EgQ2Vy
 dG1maWNhdG1vbiBBdXRob3JpdHkwIBcNMTkxMTIwMDY1NDE4WhgPMjA1MjA5Mjcw
 NjU0MThaMDsxDTALBgNVBAoTBELFVEYxETAPBgNVBAsTCExBTvBTIFdHMRcwFQYD
 VQQDEw5BbG1jZSBMb3ZlbGFjZTCCASIdDQYJKoZIHvcNAQEBBQADggEPADCCAQoC
 ggEBALTOiehY0BY+TZp/T5K2KNI05Hwr+E3wP6XTvyi6WWyTgBK9LcOWI2juwdRr
 jFBSXkk7pWpjXwsA3A5G0tz0FpfgyC70xsVcF7q4WHWZwLeYXFK1QHJD73nQwXP9
 68+A/3rBX7Ph00DBBznfitOLPgPEwjTtdg0VQQ6Wz+CRQ/YbHPKaw7aRphZO63dK
 vIKp4cQvtkQWHi6syTjGsgkLcLNau5LZDQudsGV+SAo3nBdWCRYV+I65x8Kf4hCx
 qqmjv3d/2NKRu0BXnDe/N+iDz3X0zEoJ0fQXgq4SWcC0nsG1lyyXt1TL270I6ATK
 RGJwiQVCCpDtC0NT6vdJ45bCSzsCAwEAAaOBrzCBrdAMBgnVHRMBAf8EAjAAMBcG
 A1UdIAQQMA4wDAYKYIZIAWUDAgEwATAeBgNVHREEFzAVgRNhbG1jZUBzbWltZS51
 eGftcGx1MBMGA1UdJQQMMAoGCCsGAQUFBwMEMA4GA1UdDwEB/wQEAWIGwDAdBgNV
 HQ4EFgQUu/bMsi0dBhIc164papAQ0yBmZnMwHwYDVR0jBBgwFoAUKTCOfAcXDKfx
 CSh1NhpHGh29FkwDQYJKoZIHvcNAQENBQADggEBAHOJojanzqmgasN3/gqSQ4cb
 bmdj/R40BEP+r+XT+xiidfZ2iLNwYyTneuK6AChwKfnNvOFb81V1iffRTE/KtmVE
 DMR/sYeqAH83KM5p3e121Vh4OHhyI0qNuz5oShNaACSioQ23WxHGvy9vsdVfnbhs
 plrWg9NQ2WbpCmK+2oMh2oY10Z/wvXmt9cG6jbMvcdH4z0IOvg6mrYkKTM/RCGnu
 mgxwYToj10yD5Gs4D2IJCw+fX50Dxh52MbNRYXTus2ZPRPM8JXNQC4Gwv4km3M4
 rKnJdD6hnoQ9rNeozIcBVyybQYjfrgg4DRvw9Ksk22OH4Con1B8f7R7s1LM2cSYx
 ggIAMIIB/AIBATBsMFUxDTALBgNVBAoTBELFVEYxETAPBgNVBAsTCExBTvBTIFdH
 MTEwLwYDQDEyYTYW1wbGUgTEFNUFMgU1NBIEN1cnRpZmljYXRpb24gQXV0aG9y
 aXR5AhM3QQV57XV/QqmiXDr0+GrOmqnXMASGCWCGSAFlAwQCAaBpMBGCSqGSIB3
 DQEJAzELBgkqhkiG9w0BBwEwHAYJKoZIHvcNAQkFMQ8XDITxMDIyMDE3MDEwM1ow
 LwYJKoZIHvcNAQkEMSIEMhGVzAx/S4dUwqko0cb+oa+gXfmEqw2Iz+svSKpWzC+
 MA0GCSqGSIB3DQEBQUABIIBAGtNM3MMhWZVJdn1n1fSk3mhNk6E+LFoOqG4aiHz
 e+HEQjN6bKft5zulMCqh7NKRpRmDcEE9RXDGKGYQ9BKBf6Od/041o1BY/xpPu9G5
 XnUTHN3MmqubrTSP3xxU5AozL8i7XmkB68VxKBQ2YpfcXBFgbuvlc6FXkbbh2QtRX
 UgBZEp+GSxG7o0UVJRa97t6wblUdMwaQ1ONrtBsmrO46bThv4cgrlGBvz8tGfHwR
 4HbS/Rp+6jNAS0K9fZ0PQxy2b4M4braYg3f1n4q3dDH8N0XiUcwG8FiB9XQo18+d
 fdkZwTVUoDHWjSVdIREobdPI2wdpnGxS/AB1VuiYpcebi4o=

B.1.7. S/MIME Signed-only multipart/signed Over a Complex Message, No Header Protection

This is a signed-only S/MIME message via PKCS#7 detached signature (multipart/signed). The payload is a multipart/alternative message with an inline image/png attachment. It uses no header protection.

It has the following structure:

```
multipart/signed 5234 bytes
multipart/mixed 1344 bytes
multipart/alternative 938 bytes
text/plain 278 bytes
text/html 376 bytes
image/png inline 232 bytes
application/pkcs7-signature [smime.p7s] 3429 bytes
```

Its contents are:

```
MIME-Version: 1.0
Content-Type: multipart/signed;
  protocol="application/pkcs7-signature"; boundary="452";
  micalg="sha-256"
Subject: smime-multipart-complex
Message-ID: <smime-multipart-complex@lhp.example>
From: Alice <alice@smime.example>
To: Bob <bob@smime.example>
Date: Sat, 20 Feb 2021 12:02:02 -0500
User-Agent: Sample MUA Version 1.0
```

--452

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="ac5"
```

--ac5

```
MIME-Version: 1.0
Content-Type: multipart/alternative; boundary="813"
```

--813

```
Content-Type: text/plain; charset="us-ascii"
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit
```

This is the smime-multipart-complex message.

This is a signed-only S/MIME message via PKCS#7 detached signature (multipart/signed). The payload is a multipart/alternative message with an inline image/png attachment. It uses no header protection.

--

```
Alice
alice@smime.example
--813
Content-Type: text/html; charset="us-ascii"
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit
```

```

<html><head><title></title></head><body>
<p>This is the <b>smime-multipart-complex</b> message.</p>
<p>This is a signed-only S/MIME message via PKCS#7 detached
signature (multipart/signed). The payload is a
multipart/alternative message with an inline image/png
attachment. It uses no header protection.</p>
<p><tt>-- <br/>Alice<br/>alice@smime.example</tt></p></body></html>
--813--

```

```

--ac5
Content-Type: image/png
Content-Transfer-Encoding: base64
Content-Disposition: inline

```

```

iVBORw0KGgoAAAANSUUhEUgAAABQAAAAUCAyAAACNiR0NAAAAcELEQVR42uVTOxbA
MAGs739nO3TpRw20dqpbfARQEjOywiwYnCtkDKnbcLk66sqlT+zt9cidkE+6KwkZ
sgrzfcqVMpL2jo0447gYDpeArk+OnJHkIhAfTPRicihAf5YJrw7vjv0ZWRWM/uli
vdPflQZ2kDD9xppd8wAAAAABJRU5ErkJggg==

```

--ac5--

```

--452
Content-Transfer-Encoding: base64
Content-Type: application/pkcs7-signature; name="smime.p7s"

```

```

MIIJ4AYJKoZIhvcNAQcCoIIJ0TCCcC0CAQExDTALBg1ghkgBZQMEAgEwCwYJKoZI
hvcNAQcBoIIHpjCCA88wggK3oAMCAQICEw8tJb0ROZdKzkJU6HuPTQGirQwDQYJ
KoZIhvcNAQENBQAwVTENMAsGA1UEChMESUVURjERMA8GA1UECXMITEFNUFMgV0cx
MTAvBgNVBAMTKFNhbXBsZSBMQUU1QUYBSU0EgQ2VydG1maWNhdGlvbiBBdXR0b3Jp
dHkwIBcNMkxMTIwMDY1NDE4WhgPMjA1MjA5MjcwNjU0MThaMDsxDTALBgNVBAoT
BELFVEYxETAPBgNVBAsTCExBTvBTIFdHMRcwFQYDVQQDEw5BbG1jZSBMb3ZlbGFj
ZTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAlJqVKfLwLjj+gBUCfk
acKTg8cc20tJ9ZSed6U3jUoiZVpMLcP3MUKtLeLg9r1mAfID1B/wlbdmadXPmrsz
yidmbuZmOpB5voVQfilyYy3iOx7YOqzXr16udP07k0sV+UdSNRFxrfKeoQEFXgOa
Gdmnx4OG/e3plfIKM0dPzZLoOAJF5m500xzXPL74zFCWp2flZkuE4A6141koaZXC
N5XL7wWTLMLenF9Byb5ksKqUuqEHAMdlnmoNMgjY9VfVfcrv9w43GG8FtpSX+TWz
B2zNS2OF+XIVnzRG5DeoULq8v88Z5bLpIJ/nx26r8A4SSwIBaVv4wPxAfliPsIVK
arUCAwEAAaOBrzCBrdAMBgNVHRMBAf8EAjAAMBcGA1UdIAQQMA4wDAYKYIZIAWUD
AgEwATAeBgNVHREEFzAVgRNhbG1jZUBzbWltZS5leGFtcGxlMBMGAlUdJQQMMAoG
CCsGAQUFBwMEMA4GA1UdDwEB/wQEAWIFIDAdBgNVHQ4EFgQUo1NB1UQ8gCkVfAEj
8OeOr83zdw8wHwYDVR0jBBGwFoAUKTCOfAcXDKfxCSH1NhpNHGh29FkwDQYJKoZI
hvcNAQENBQADggEBAIFJeKCsTKcFqQMpTryujRGzJdYA+R9eBAuDLsatbtKt14F
zkgRyOg3l/+Cw7H8e30iLrPIfLWN1qjHrjgOyIs5AQ/hgxLvLir3hEUV2Z3MRsMt
jH2x9SG91PEM046gfPnc9gMGHjMTg1qvaKcLQP5UzpEYPLror2X4P5uXxaP0LIZR
zWmkw1RF7FOD7PFB5v94M5274XYxW2W4uKgd7QGnUZROSvSYkGiWdp1JhQxwfdZ8
A0enITGXnoEkAFvviCqh64P1hIeMorj36pgL19oWZD6YrzSWHUz1F00juyuOfQs
qm6hvrDTqNpHNZ015fOURza1SkCvi9GFmNUPoVgwggPPMIICt6ADAgECAhM3QQV5
7XV/QqmiXDr0+GrOmqnXMA0GCSqGSIB3DQEBDQUAMFUxDTALBgNVBAoTBELFVEYx

```

```

ETAPBgNVBAsTCExBTVBTIFdHMTEwLwYDVQQDEyhTYW1wbGUgTEFNUFMgU1NBIEN1
cnRpZmljYXRpb24gQXV0aG9yaXR5MCAXDTE5MTEyMDA2NTQxOFoYDzIwNTIwOTI3
MDY1NDE4WjA7MQ0wCwYDVQQKEwRJRVRGMREwDwYDVQQLEwhMQU1QUyBXRzEXMBUG
A1UEAxMOQWxpY2UgTG92ZWxhY2UwgGEMAA0GCSqGSIb3DQEBAQUAA4IBDwAwggEK
AoIBAQC09InoWDgWPK2af0+StijSNOR8K/hN8D+1078oullsk4ASvSwjsCNo7sHU
a4xQU15JO6VqY18LANwORjrc9BaX4MguzsbFXBe6uFhlmVpXmFxSpUByQ+950MFz
/evPgP96wV+z4TtAwW2Z34rTiz4DxMI07XYNFUE0ls/gkUP2GxzymsO2kaYWTut3
SryCqeHEFbZFk4urMk4xrIJC3CzWruS2QOFHbBlfkgKN5wXVgkWFfiOucfCn+IQ
saqpo1d3f9jSkbtAV5w3vzfog8919MxKI9H614KuElNAtJ7BtZcsl7dUy9u9COgE
yKRiVokFQgqQ7XNDU+r3SeOWwks7AgMBAAGjga8wgawwDAYDVR0TAAQH/BAIwADAX
BgNVHSAEEDAOMAAGCmCGSAFlAwIBMAEwHgYDVR0RBBCwFYETYWxpY2VAc2lpbWUu
ZXXhhbXBsZTATBGNVHSUEDDAKBGgrBgEFBQcDBDAOBGNVHQ8BAf8EBAMCBsAwHQYD
VR0OBBYEFV2zLlTtHQYSHJeuKWqQENMgZmZzMB8GA1UdIwQYMBAAFJEWjnwHFWyn
8QkoZTYaZxxodvRZMA0GCSqGSIb3DQEBAQUAA4IBAQBziaI2p86poGkjD/4KkkOH
G25nY/0eNARD6/oF0/sYonX2doizcGMk53riugAocCn5zbzhW/JVdYn30UxfyrZl
RAzEf7GHqgB/NyjOad3pdpVYeDh4ciNKjbs+aEoTWgAkoqENT1sRx1cvb7HVX524
bKZa1oPTUN1m6QpivtqDIdqGJdGf8L1zLfXBuo2zL3HR+M9CDr4Opq2JckzP0Qhp
7poIccGE6I9Tsg+RrOA9iCQsPn1+Tg8YedjGzUWF07rNmT0TzPCVzUAUblr+JJtz
OKypyQ3eoZ6EPazXqMyHAVcsm0GI364IOA0b8PSrJntjh+AqJ5QfH+0e7NSzNnEm
MYICADCCAfwCAQEwbDBVMQ0wCwYDVQQKEwRJRVRGMREwDwYDVQQLEwhMQU1QUyBX
RzExMC8GA1UEAxMoU2FtcGx1IExBTVBTIFJTQSBDZXJ0aWZpY2F0aW9uIEF1dGhv
cm10eQITN0EFee1lf0Kpolw69Phqzpp1zALBglghkgBZQMEAgGgATAYBgkqhkiG
9w0BCQMxwYJKoZIhvcNAQcBMBwGCSqGSIb3DQEJBTBTEPFw0yMTAyMjAxNzAyMDJa
MC8GCSqGSIb3DQEJBTDEiBCBwnBPnNMORN+JxFvMbZIJ5PtqEBkyDbOtU1Ar5RuGl
LjANBgkqhkiG9w0BAQEFAASCAQBRpxYXiiCEQ/lshkbhph566H65wAf9rZbGn+r+
o8vLTFSS84ER/EAHGhePmVDiObJS+nXIC7Sa5Y+tUe8JitKPXBQ2oDq2+3tN7tY5
G398yv+LnmYMMf91dlnlyPnQujsEfPSLXyNtoa0qBqp1DThm/pfn6RbbOqpZjYr9
fdCNdErDq15+CKaf8R/JDW+hiLyvD0KCpXucWLHb1okt1Jpld4kkaA4wu9Idh9fK
G1N20s+dBXoytH/G6K8NhOh3Qaf3lMP1R60gkvJVJ3j9jIs3/ZG4qh5qWQJHLvi2
WLSxDhkYmZ+dYScyfIauNkq7a0wauSpZj82e1FA7HdyZmNp0

```

--452--

B.1.8. S/MIME Encrypted and Signed Over a Complex Message, No Header Protection

This is a encrypted and signed S/MIME message using PKCS#7 envelopedData around signedData. The payload is a multipart/alternative message with an inline image/png attachment. It uses no header protection.

It has the following structure:

```

application/pkcs7-mime [smime.p7m] 8690 bytes
  (decrypts to)
application/pkcs7-mime [smime.p7m] 5426 bytes
  (unwraps to)
multipart/mixed 1356 bytes
  multipart/alternative 950 bytes
  text/plain 293 bytes
  text/html 388 bytes
  image/png inline 236 bytes

```

Its contents are:

```

Content-Transfer-Encoding: base64
Content-Type: application/pkcs7-mime; name="smime.p7m";
  smime-type="enveloped-data"
Subject: smime-enc-signed-complex
Message-ID: <smime-enc-signed-complex@lhp.example>
From: Alice <alice@smime.example>
To: Bob <bob@smime.example>
Date: Sat, 20 Feb 2021 12:03:02 -0500
User-Agent: Sample MUA Version 1.0

```

```

MIIZDAYJKoZIhvcNAQcDoIIY/TCCGPkCAQAxggMQMIIBhAIBADBsMFUxDTALBgNV
BAoTBELFVEYxETAPBgNVBAsTCExBTVBTIFdHMTEwLWYDVQQDEyhTYW1wbGUgTEFN
UFMgU1NBIENlcnRpb24gQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIeh7j00
BoqMA0GCSqGS1b3DQEBAQUABIIBAB5TXoiCIIILxehywh5/tdFM72iw946N6OzE
mkIj1x+ShPweKrmTgPxaZbNgZpMdyNetqSXTn5H1ZwUAXOkE+EPp301kveWwxBAM
/Umr/ODGiYlHWORWh+cPwjo00IHo8IJzmf9FWMr7CKYhvbSZn3AFuERRfEccwH9
xsbB+X5og5bu0Mn3y8KdX7XOFVbgAgFuqqWpj6mK2AsyWS0zRKngNd72rELjEzCv
RZqBFAecaxdJd2RXKKwLmJg5EL/VmKuyN6TgtmtwvzGCKc5YywdhVrP2IvQTye10
+paJ8dFQb3W9AGOUcdw8r5CoawAZdYmVz/v0ixYIkQid7fsOE+AwggGEAgEAMGww
VTENMA8GA1UEChMESUVURjERMA8GA1UECXMITEFNUFMgV0cxMTAvBgNVBAMTKFNh
bXBsZSBMQUU1QUYBSU0EgQ2Vydg1maWNhdGlvbiBBdXRob3JpdHkCEzB8R0APhiY6
HGLS64Mv1sDXhpQwDQYJKoZIhvcNAQEBBQAEggEAFVnVvKTKKAXPN6g5xLlW/7bO
5NQs0DVNxjuCAOXWm9zsyhH8tYGdNVvzktXkn0JV4g19TEu4MisuhcIhqJyrSsh
4epi0ZxbyM/YTnhHvi4wttazq07tNVF6eafyuecDKLV8/WF+AGSVWe0xPumEni3w
GADvkwmcO2mDZO/ad/u7Jv14jF//Id/IG/A0y/yBgrWq4pH7BPwp1W/rXbnw1EEm
8an56+5f/m8teqqXaiRMVQgMaKGCmXHd3Ud21Rqc4jwsN0VCpzabK9DSDPcxwV1
H+PPUtza/Ux7yNgJ1gm816e85lu0jvvpf+H1iioHpNKCQ+eh6mH0BqLJKJkjetjCC
Fd4GCSqGS1b3DQEHAATAdbglghkgBZQMEAAQIEEGaUNdPZs2K03fcnaJXCvfaAghWw
qf0kEcGM1xiJegJu4TKQPvtUKje4+xRba0xUUS1TzhXrDk9tk2J8zdmnQglvRW54
r/xH0TLM7ny0unGI3ow8lpyUV7g/LFmW9kia0TnhNcEe6qqSk09dEH1rEqIpQoQ9
1GjuFw1c7uf9vMzb72TDAehFIOVwboM5hmLtoAdvQeH4AsDKfbPvkrFPPNDB7Rj4
QCh2Pile1LH+3+78XvJ0NdcZw5KyVFIAa29v1SYTjPNriFn+rKPzXjk/cQc/zHSH
DLZ0CRHvJZKX6z+oIVAq/DYUJfhm4zz4LRSReQfdyChRHDVv4V1dFT2uaqtBPP5C
6c8Ad/SQUfk84wns2/+pKocqa69tVTIoK3Y4+1nDcvg8jzkdPD0cednWdyjh0vQz
0qXaJYFflyVuQV8A3IUFV7uX7JCuo6m/PIQIiH23dE7fkGXCPiIwA19BZj002bo10
ZbQgka+Csxk/S/1BiJ5hfdS6tOk7JZwSNQXHgGrCp7lboSljxOefiKQVjo+ynye

```

LmUM0zoZz2eUdsUILLQtMfS9r0AvBrW8PcF79IIOQR+X4QEJ6Ztz3zAgj409q0Fmm
fCrhJTUMcVZyeqLUTpyLWDBKqV+ jm5dA7WR8CL5NqEsmtYQRTAbkPv8a0DNpgMl7
fCN3bIs6VdsiQXdhwwH8U8pcdzSINvNb2nNbUrFWlU6Z0x160DGQKm5KxuUd0Uzi
xKe2v0DMl5TyjRekBPPhoZC3Mwqf7Ud6vDoBk4Evh1xjv8MAKA5LOghtfvv4xP/eP
L5i4V3EnZtRy4hnW060tcDOodWW2PXPYFPxN0z7UEdKk1hjomBGF1Wt1QrPzMO5x
0/m4NezYVWJNwkqPmCUyz+bDzQgIdWXGXGAejNBjssEvS8eHlRs7V27UfOQ9c2k
/KqDn/Wf15RfoIiTlRfoU4FjBoiq6IXkerP1Km+SzHHnZozF15M684ulz/PPpo29
pziu9WRjDPsWYmS8RK/XzutHp1r7vDInwCdrManEI811C7z/3/FgwA7RJIJ6GNhn
GVD+PUBULWxEIPHQU58y7KwBeXtNX/o9rPul3Nt2HOINyYhhLNqX5AyTpG1ONrFJ
TzP3rrqvGLslmq644pBLfJagaxCAJENyoZ7GT9YgWrT6WzVM6t92VpfCo0Wy0SRy
uy+16De9bJWDvWPy+RciW5UyN7YuCWxe/vYcAiL55Lv2ZO0m3zmE101bJ7/ZgwT
A7yABCQqUQPRBc2EnchLv8JdYw1ACX9JIIG/dTmyI1OLNAGb20UGX0d76mGajwT+
a5OF6z+HYxd2KehL1+W7wYrUxfZ1UtK6rACIVD5b+36nElmqTTnSow9z9mAzo+8+
hRcBQ9I0JOB3YMAi4LepbcGGvEAFh9kOSY+9bYy7Lri0HoQEAADZ0aQxf1/12UEAj
P83AjqaswVVKBJNvFpqJnJeh6Y/sTr9eAYE2+Y1PGGH9Z8fzbD7+CqL78sbpaMCP
7cgM9UHRjLY8yOIEl3fME/JF1pR3NMG3LQ9dohsgv18Z11JABY8+Zz8103g5ZjBy
xJXkWAXBdTYx210bdaIyoTQnWcN1OPaCatCv4P4P8L0SoCj3DrEb1rK9pCUIJl0M
pElAoPDJIGYrEPo2d3TunL2qJwAJEY0asaONMvvA3eSdC8kzM+NP5gYHl6gRFvDQ
WbU2LRsCKwu4TtHRR92OqKW1r9x4ZgyZH7UvVnIZVGz2buta7ssQ+PLDwIXemtFh
3LaYmNYrSSJ71nd3WwXvS5MxWa/OBwPpDS20IRwOOGmAYKWpQzGFJb/gWF4/rSiK
KSeC0qIb9UXL31AX7eA++TR9mb1zEoIrlBebF+MwX8EzQbYRtbvezL3xhXeu1TsS
JUUBS0Z7qF/2AljMgrTjkIQGNuVLhLxexaQJr0GLAwlK2ijOxXK6bGh+JUW12HcT
Ms71ef811J1fHrS7mTzqAAreAsUrUs30WBBYmWsvRyMMqNuWRJR4Ax1jF/5HBNPI
bdx9X6Dz51azBBQb78S2hxLwrGLfbbheyYJO6CwMeM1epsV/VvCuKfakVGINS4yg
i7DHBQRHXekU6XzCgCRARC288zwdPrSxqubQYgchpewg9ZBK/SyulFRw/AjQowNS
ONatIkKD5N8UZAaf/iLznBzG+bXF4esrMpUm8MY1acow7A6IyQBioGEaAh6U05Ww
sQz+6K06RNneu5+PvGt18rGGmVjdevtTZSTT//dlJyREItmsyHkY5cHMugzz8FAh
Yy2ez/q7sbl12P7YFY6TXRc4FIIEVooK6LbsHggzwcibhc80Ue7bq+T6ouFYECBW
1hNwzGLbtjkO1ui/1jBqBRAYkbbqciWj337ZRjzbea8NeaoYYQo2ZHM9HKMK7mqS
z6E0XGz++vz83pdsh/ZHF/i815OgvGZjG99KvpDy6zZ3PxSdASBOxx403wpUEd4B
+8RB9N4I+9xPKmqBFQx2/gLY3jqLc81WGp8oP1jZHDCYv4rMPnFzK4k+gpYu65r+
Iwy8HIYDzsUNJPxZwHo1GX9BQkt+/X4p0aqLE04G5gP10TrnsL5CM4WGyphPz0U
3b69yGFwP/LFj2NZ3LxD6b+fFsVccoqrEz70WPpgfB4NAVvVXLTjI4GkMCHApLhr
466UrQvoEGLVzAPbxVo/2qVal+cTc8XvIY3s/kKLcHnsOvC6oICvKmlfNPQLv42s
K+qg2NZpM3RHyeplbHe+rPzUeIOmCSUluVQxp6HghEivLX9D4WU1Asaut747uMy
fugR11vaTmqVhpc06Bdc/104TiyAXvZyYh+Uv9U8YZPckNZCH15y4sJTVxQGQhLN
KzQzFNX3mcqFYBW7xZr0fLSGaQxClqQ6SPaOcUKp2jShAInPMB13i16MzOSOo9BW
9SgnXDcqaigQWem4VY1gHuuQmKj4WitLU0Ue5AizZDTPMN0JvUnh99brfVETjien
gNHRtdvrXwt+N2baVRn0GFtj66ebu/rAzqTNZsA5p/F+APdUzxUDrPfh1WYrzzSQ
8Dx1RmCTLLRzafCVXLV3xNbWnrFFPX4ilkT+roGTRjYqPv0yDUtvrIt7HKFnZoLl
mLkk4auI/TQgJ72Ne3+wYYSMvOwrHbF8NLmsgyAJSEgWl+FUUBx653i9H6CiABOF
8YVvz7ShqSwhxG1lroER11wJLdXclLWgR65rvkCYvCH7bIHU7kvQoyIZXaLs1Anh
rBNh185OH8RmBfNXNPbt6Hh+2KknmaPCKMxEWkNrLmGseoTJ1/okRunut+DW3FXI
ashoguanB05zVngb+r+jzAwFRGVY30Cgepb0gBwQDyeZBCCWD3Mr/1wXnB7S4Oh
/zMURX7NtwZUOh2qcJ3Xlpi0S12mNvLSmlyxzZv2dYDolmPwJHptP7tBiKnsZoHM
wbCEUA21JSHRLDXXyC82AtttZv2auF1p06Ne2H/en8Y+z8MRDG7gBI48IDGKq3Ej
E0h1VdxVhWvEuavw83TVpvdKo0Q7rVRC1hHSttat1z8TxnKRxIRvxC+fJ2xGx1Pv
onlaYRq5tL/jIuJIGVHHeSeqB81yiwJ2dFfydlI3VaCSobVBwVbDKvRli1HskeSB

WGT7hyhS0SDnh9MVHw0z30JWnxxxXfg4dB0C0vQWLsTqZm0bncxxBZBR060kSY8RL
S9mYpaSeHL129h30IKecjiXhhsA3UI60yIS7VS9dzLE9W53ttU5MLiHhXnYANy5U
eqar+811luxtB90CjunOqtgkH0u4Ch+lnAUjdmz7cUPxLwgPgwr/WqJxORTnpGL10
hEumGFYF3h/XIUW3bNCqjAutco8B38s0kGBipd0XCg+Rr60S31US2//mnrq1E05K
VtKVK+NxfCWkpzcZLFOIxGLwHsSqq3He2QgGovkRRkCZE0/bBqhvbvAeZYZ10i2/
clB4eYdplZZJ7s3hKPwq678LBRXT3Fs4a9BpqEnvUot6WfgOsP/zsszS247EjWra
w+OAKgdhSOILeuaXfpHRR2FEDYVU+yBdwJjHYzp3knXDDsEALaUmAbOIhZ3A79hY
tCSmzEhXFHdOdpwOwqV0L8VpvumZna/GZE84U8uPEHbE5eeX/6BLNJx3606FXkB8
waoUUNuiHpPMQbz3cLxZzXN2TGrrmmUbpId9+CPfymRGQ9sqBTShxg+tZ7Fz03vSM
WB7Vv+uxhCfBoY45MPX05vVAaxIENdQRabGPty7WqZepGXNdjwC5PaKDPuG699WD
22BOPA8sJ7TLqGj/yJ8Azkl0p15DUr+Kr5gDSwf+j8jt3hhzeFUpQ+9aFmxb1IVf
W01Kq5AXLVscZZ13J7hpbG62BmnlEMPY7pV6B+PkbxWkXaT8b+GW8OVsZw2uuOc1
Fedl9AGzjYp1FFFRATzKHqpMfqbt60SNkGx/9mjs5oYR90RCmy2PCKiMh3tPYCj
iQnyJymV6x58UB1tRNbjaUD+rCiuea5HEUv04xdKB37XJ10EcNT/Z8A+DQGLpLby
u7GHTCTMzNLOWmibhfc2FRFC2q/MaZC4N/IrB0EWAXDIm7GDHlkUoaHL9ADc9vyg
xz44m/CTcf5ETE4d/rEm7FEFnzVtBPbd1Ghi3EXhQ7WCRy1oJRPoktdKNvePXSQ1
fVemwRsBA9jflTwIzS/ASUTQohDpYaaqV97aUNn9psRuFblwgGUx0I/XUCUdbFxa
zuM9a7jxDBYOvYtn43GINFOlnK+/R3zX1cYm0CvF4+QUNZIOuEP0NvE9Cjb68SfH
qAeV4HIRBg3/jU+8PRHTyUz1Qf7vRXKiDM1nrTlbelccJTWxUtybEKECersUX+zv
Ybv2/w339RJRy0+Bc2VJt9uB6DX7p2HTQyFvaZTgN80ZLAKBJ/xk4WC6Vc+h7Em/
y5cqIjJj0SES2VoyP0cu/rJ06+gg7v+OHHehmhkehuQNsLnXldAgGJyIFKcvw4C
+NrQ2II8uJ54Q+ytRAmr8GDV7F6cHb9BuyTT1ubQEP1L5EwcEFWUESEv3A4quit9
t1r3jEuPbc3fqyIcmDNKP58qs0ZPO3m/fJEW1LX6yR0IEkrSxZD6PbUYgNT+qZD+
Rh1NUJ6dIpd+xxA837NxUOnkrJQ3uvOvURBKVv20oOXzDVkRtAIEy8aVic6ZAxIX
ZHqkikEiFxxgNcMxi04agsE7qwCKvpq61lM+xxXF5Puqoj7vL1ihzCjoABqne5SE
yBkYqU2OU7uo1vWSwVdtwqX1Ih/adN5t01n1HWcMHBBooh04nfpMrhci80i/XYTA0
new3jLMwZXEBzh1kZ62Zzt1PA68K9f6XkStAJ+bx+s3iV0K4RmLt7VC88+1Kspns
/pnDEBfBCQhGD07YeKUJbBj3RPdRi6rsj54PRsZkOAi2MoQZJ6PnzfI6EHsQXNad
PnyFB6ZGrse1ayA9QqibkRFMKGRSakKB+fq12M36RB8CeO766iMoc5qc8n5qz0oH
BB1fTiAHTGU+6AhEGU5kiFlZaehBcp5yDl2I5I5lc0X786Zjdm4oGbGq4q6Ieyul
OLx8vkb9L3ZvkLgZAvn1r2dZKOxyNewjQwFG05ErbK7ppqD6TC5VZCiTLJKs1N+B
L3/UjwSwc0Lt3P7dep8oDySMgxKYDQJ0qNBFa6kwdZzTlaXRfQUFHukwn6fn10kX
1p/2K+oYUsA40E9qL0cWEMWcNmYRQyk0qpgWWIykrM14efXkQxSddTqP0WfW/uxs
pQB4rVeZStpz09cie1E0tVcoipItPnvvQTENdc/p4Eg2bw2dW+Vd6NB/HwobsPY3
YRox1LGrfj0LH7Rg0qg3pIOD2u9qo3A7ZZ95vkGUtTtF0BYkIf9/SFoEwNSJARNp
BOBA51Mrq3S9qwJEOYoA4KuFqLmpbmQg1K3bdi9M9aDK3hgQgLqW5GB4TF00WuaG
1kKQSPvZH0dZGtYxCjndth5Bp1MhVmS05mlr/uRKdVjdSq3MKj/20/Nm7P28dRt
O+w7rvRINTp5fWbstkwtBnheOkyX9usXU1qigTIUsA1Xq1aG5g5qrDpG9Ijgya1i
ShQJ7cLOtGF1J1kZgG/fT+jbJNSNke5uvMLF9/chmmR2SZEhou1tahe8J2/97H+H
L6epMyb4QYeH9JTLDEbyz8bvouA8ydhOHbMj6Vr8Ox9af+Uu1FhDtJs57goehgS
/SB1jJGQMw10kHhLpK8qOk9i+NZO05N+GiB1VgusHDyjsUHnxk3mM8hoRqqpkxAW
7mqZagME09qk7PEct1loAgrwdTSIB9WHIudg9cV1yFilkI2ktjEZPD/i8uZq05n
pd6v4w/XJuPopVn5nwJxOwQy1RKDNSOUaWRasZc31+16D4eywDgDesSLaBmXU1Ui
dbbtKOi4OnAEwQ1iyE+Q7JABttILJ8aDSejBvP5gUvKPB1iDLwAXMR98ruJeMdbE
/6qCA6YAc5v/UxREKcZBqSYsOaEqD1YKZEIMhn64NDqpdicX4gwe/sCawTcX1E5r
XLgnSSpflbIexggQ46Ma1BLGp9ChiGO2bw1IzmlGGOXqpmKN6FP00sSnwq9D2J
nquParO4ILWbL9aWbCA6EIkcer/COfWGidtazmTj5MXkD831Y3cozRuC9dYLO+4R
FXsWzvqQeXiauLz8iQsgxKUj2DcPT2k6j/qzSXz/M5xapj13Bk6VH9KoR194/sMT

```

ggGJvWOnYdZjv5J3i3oQOwCL9T/ZgdqIFW82jfmGvoe2zu/00XnV9FP4Lbr4rtv6
if54Hr/h8jqJoRnBGAh3doQIGdGLiZZDPt+GWMxreYAk16mbXpuqn49bP8G75ZKq
5Azp5xgNcm/rPGYE+9iQJSggoz+dqGiQ0u371K+i0/A0OzJ845NW82hoUye0C+X
DB6OkbbYCgGmPou7bBVaUJNQqdRUTnGd/Yr1EaOQVScMZ09FN2hJx6V1zjdMUvTe
XXpJ2C5R14kxHY6pw8mInAg9ja7jmY2e7xaNA4cwRNTjbH7J5uZFNEC2kSf4ZO7V
k7MOX+zDe285FFVBS2+97yAL3xaljlE4DZVFOW+3dKD+W2bg4r0Yhds/wxYH+M5
GU9zLrHEbw0GsPwUr50w9isSu+o9SKeOCfWrzHz1fJnH26woPOObWy+kkG2cunPN
T5e+OPw9K3MgBkNZ9YG6Ce9ULqhO65f4LISdwDSsMGL3eNhgZMPLtCJZAP8K7dEt
80c3POY0NSB8lqloyxDwHKJz0S/HMwrancUO5V9abkZuYhsOGW+1Kjswd+cPh5Y8
HoL3GF+OAopbYYesvIWgzh0/MtYYUoI3kPvUd4vdWNHEbtHlfsALDs5pukAE9ny8
0GhNtdoH04cVlvDmpyfbLcDTwi+UJ5tT1VQMGLuFo/CxDV9vWjXhJd7kSt+7+K1L
YPzrT6ggMFRlA0kYRIa5K/n99wp2aYab7/DkwfpejZI=

```

B.2. Signed-only Messages

These messages are signed-only, using different schemes of header protection and different S/MIME structure. The use no Header Confidentiality Policy because the hcp is only relevant when a message is encrypted.

B.2.1. S/MIME Signed-only signedData Over a Simple Message, Wrapped Message

This is a signed-only S/MIME message via PKCS#7 signedData. The payload is a text/plain message. It uses the Wrapped Message header protection scheme.

It has the following structure:

```

application/pkcs7-mime [smime.p7m] 4319 bytes
  (unwraps to)
  message/rfc822 inline 642 bytes
    text/plain 228 bytes

```

Its contents are:

```

Content-Transfer-Encoding: base64
Content-Type: application/pkcs7-mime; name="smime.p7m";
  smime-type="signed-data"
Subject: smime-one-part-wrapped
Message-ID: <smime-one-part-wrapped@lhp.example>
From: Alice <alice@smime.example>
To: Bob <bob@smime.example>
Date: Sat, 20 Feb 2021 10:04:02 -0500
User-Agent: Sample MUA Version 1.0

```

```

MIIMcAYJKoZIhvcNAQcCoIIMYTCCDF0CAQExDTALBglghkgBZQMEAgEwggKZBgkq
hkiG9w0BBWGGggKKBIIChk1JTUUtVmVyc2l1vbjogMS4wDQpDb250ZW50LVR5cGU6

```

IG1lc3NhZ2UvcMzJODIyOyBwcm90ZWNOZWQtAGVhZGVycz0id3JhcHBlZCINCKNv
bnR1bnQtRG1zcG9zaXRpb246IGlubGluZQ0KDQpNSU1FLVZ1cnNpb246IDEuMApD
b250ZW50LVR5cGU6IHRleHQvcGxhaW47IGNoYXJzZXQ9InV0Zi04IgpDb250ZW50
LVRyYW5zZmVyLUVuY29kaW5nOiA3Yml0C1N1YmplY3Q6IHNTaW1lLW9uZS1wYXJ0
LXdyYXBwZWQKTWVzc2FnZS1JRDogPHNtaW1lLW9uZS1wYXJ0LXdyYXBwZWRAbGhw
LmV4YW1wbGU+CkZyb206IEFsaWNlIDxhbGljZUBzbWltZS5leGFtcGx1PgpUbzog
Qm9iIDxib2JAc2lpbWUuZShhbXBsZT4KRGF0ZTogU2F0LCAyMCGZWIgMjAyMSAx
MDowNDowMiAtMDUwMApVc2VyLUFnZW50OiBTYW1wbGUgTVVBIkZ1cnNpb24gMS4w
CgpUaGlzIGlzIHRoZSBzbWltZS1vbWUtCgFydC13cmFwcGVkIG1lc3NhZ2UuCgpU
aGlzIGlzIGEgc2lnbmVklW9ubHkgUy9NSU1FIG1lc3NhZ2Ugdm1hIFBLQ1MjNyBz
aWduZWREYXRhLiAgVGhlCnBheWxvYWQgaXMgYSB0ZXh0L3BsYW1uIG1lc3NhZ2Uu
IEl0IHVzZXMGdGh1IFdyYXBwZWQgTWVzc2FnZQpoZWFKZXIgcHJvdGVjdGlvbiBz
Y2h1bWUuZCgotLSAKQWxpY2UKYXpY2VAc2lpbWUuZShhbXBsZQqgggemMIIDzCC
AregAwIBAgITDy01vRE510rOQ1SHoe49NAaKtDANBgkqhkiG9w0BAQ0FADBVMQow
CwYDVQQKEwRJRVRGMREwDwYDVQQLEWhMQU1QUyBXRzExMC8GA1UEAxMoU2FtcGx1
IEExBTBTIFJTSBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eTAqFw0xOTExMjAyMSAx
MThaGA8yMDUyMDkyNzA2NTQxOFowOzENMA8GA1UEChMESAUVURjERMA8GA1UECXM
IETFNUFMgV0cxZAVBgNVBAMTDkFsaWNlIEExvdmVsYWNlMIIBIjANBgkqhkiG9w0B
AQEFAAOCAQ8AMIIBCgKCAQEAAmpUp+ovBouOP6AFQJ+RpwP0DxxxZy60n1lJ53pTeN
SiJlWkwtw/cxQq0t4uD2vWYB8gOUH/CVt2Zp1c+auzPKJ2Zu5mY6kHm+hVB+ItHj
LeI7Htg6rNeuXq50/TuTSxX5R1I1EXGt8p6hAQVeA5oZ2afHg4b97enV8gozR0/N
kug4AkXmbk7THNc8vvjMUJanZ/VmS4TgDqXjWShp1cI3lcvvBZMswt41/0HJvmsw
qpS6oQcAx3Weag0yCNj1V9V9yu/3DjcyBwW2lJf5NbMHbM1LY4X5chWfNEbkN6hQ
ury/zxn1sukgn+fHbqvwDhJLAgFpW/jA/EB/WI+whUpqtQIDAQAB04GvMIGsMAwG
A1UdEwEB/wQMAAwFwYDVR0gBBAwDjAMBgpghkgBZQMCAATABMB4GA1UdEQQXMBWB
E2FsaWNlQHNTaW1lLmV4YW1wbGUwEwYDVR01BAwwCgYIKwYBBQUHAwQwDgYDVR0P
AQH/BAQDAgUgMB0GA1UdDgQWBBSiU0HVRDyAKRV8ASPw546vzfN3DzAfBgNVHSME
GDAWgBSRMI58BxcMp/EJKGU2GmccaHb0WTANBgkqhkiG9w0BAQ0FAAOCAQEAgU14
oJyxMpwWpAy1OvK6NEbM1lgD5H14EC4Muxq1u0q2XgXOSBHI6DfX/4LDsfx7fSIu
s8gWVY3WqMeuOA7IizkBD+GDEu8uKveERRXZncxGwy2MfbH1Ib3U8QzTjqB8+dz2
AwYeMxODWq9opwtA/lTokRg8uuiVZfg/m5fFo/QshlHNaaTDVEXsU4Ps98Hm/3gz
nbvhdjFbZbi4oz3tAadR1E5K9JiQaJYOnUmGpF8PPwDR6chMZeeegSQAW++OIKqH
rg/WEh4yiuPfqmAvX2hZkPpivNJYdTPUXTSO7K459CyqbqG+sNoo2kc1nTX185RH
NrVKQK+L0YWY1Q+hWDCCA88wggK3oAMCAQICEzdBBXntdX9CqaJcOvT4as6aqdcw
DQYJKoZIhvcNAQENBQAwVTENMA8GA1UEChMESAUVURjERMA8GA1UECXMITEFNUFMg
V0cxMTAvBgNVBAMTKFNhbXBsZSBMQU1QUyBSU0EgQ2VydG1maWNhdGlvbiBBdXR0
b3JpdHkwIBcNMTkxMTIwMDY1NDE4WhgPMjA1MjA5MjcwNjU0MThaMDsxDTALBgNV
BAoTBE1FVEYxETAPBgNVBAStCExBTBTIFdHMRcwFQYDVQQDEw5BbGljZSBMbz3Zl
bGFjZTCCASlWdQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBALTOiehYOBY+TZp/
T5K2KNI05Hwr+E3wP6XTvvi6WWyTgBK9LCOWI2juwdRrjFBSXkk7pWpjXwsA3A5G
Otz0FpfgyC7OxsVcF7q4WHWZwleYXFK1QHJD73nQwXP968+A/3rBX7Ph00DBbZnf
itOLPpEwjtTdg0VQQ6Wz+CRQ/YbHPKaw7aRphZO63dKvIKp4cQVtkWQHih6syTjG
sgkLcLNau5LzDQUdsGV+SAo3nBdWCRYV+I65x8Kf4hCxqqmjV3d/2NKRu0BXnDe/
N+iDz3X0zEoj0fQXgq4SWcC0nsG1lyyXt1TL270I6ATKRGJWiQVCCpDtC0NT6vdJ
45bCSzsCAwEAAaOBrcCBrdAMBgNVHRMBAf8EAJAAMBcGA1UdIAQQMA4wDAYKYIZI
AWUDAgEwATAeBgNVHREEFzAVgRNhbGljZUBzbWltZS5leGFtcGx1MjAyMSAxMDUw
MAoGCCsGAQUFBwMEMA4GA1UdDwEB/wQEAwIGwDAdBgNVHQ4EFgQUu/bMsi0dBhIc
164papAQ0yBmZnMwHwYDVR0jBBgwFoAUKTCOfAcXDKfxCSHlNhpNHGh29FkwDQYJ

```

KoZIhvcNAQENBQADggEBAHOJojanzqmgasN3/gqSQ4cbbmdj/R40BEPr+gXT+xiidfZ2iLNwYyTneuK6AChwKfnNvOFb81V1iffRtF/KtmVEDMR/sYeqAH83KM5p3e121Vh40HhyI0qNuz5oShNaACSioQ23WxHGvy9vsdVfnbhsp1rWg9NQ2WbpCmK+2oMh2oY10Z/wvXmt9cG6jbMvcdH4z0IOvg6mrYkKTM/RCGnumghxwYToj1OyD5Gs4D2IJCw+fx5ODxh52MbNRYXTus2ZPRPM8JXNQc4GWv4km3M4rKnJDd6hnoQ9rNeozIcBVyybQYjfrgg4DRvW9Ksk22OH4ConlB8f7R7s1LM2cSYxggIAMIIB/AIBATBsMFUxDTALBgNVBAoTBE1FVEYxETAPBgNVBAsTCExBTvBTIFdHMTEwLwYDVQDEyhTYW1wbGUgTEFNUFMgU1NBIEEN1cnRpZmljYXRpb24gQXV0aG9yaXR5AhM3QQV57XV/QqmiXDr0+GrOmqnXMAsgCWCgsAF1AwQCAaBpMBGcSgGSib3DQEJAzELBgkqhkiG9w0BwEwHAYJKoZIhvcNAQkFMQ8XDTIxMDIyMDE1MDQwMlowLwYJKoZIhvcNAQkEMSIIPno+5X5nFLPT0q5vegHgVP4OV2/uzd4xPnLWkqhqYIvMA0GCSqGSib3DQEBAQUA
BIIBAKG7Nq53TFMHU6ciIcQ9Tqq987YPEVAIJJ23U+60DXrXsrrmcZCqd2ZYhJnf5Wc8vBoC9tzRBoQpl0WMS3WYQQkkWYY+ovPyDqcEt3iixC0aVRWIZoDiq5SiWR81B9CUcsKueu0IG1xmdvCmI/wrODkDEg1SV0Z+d2cs/I+OS1F5NVosffsd4JhkTx12dD5BMCfa0zaS96GPadv47p3oizmsO9u2TIBCceD94k6iIhG0j19rdeUmOunTK1bOdz6Y1TlVrb+s+nYGQUtOWWGulO854oCYjWuTi2Twz1BI9NrrMM6xR+T8JAXIkXxvKwjA1EtT2Nvp0OqVR9izIeei00=

```

B.2.2. S/MIME Signed-only multipart/signed Over a Simple Message, Wrapped Message

This is a signed-only S/MIME message via PKCS#7 detached signature (multipart/signed). The payload is a text/plain message. It uses the Wrapped Message header protection scheme.

It has the following structure:

```

multipart/signed 4562 bytes
  message/rfc822 inline 672 bytes
  text/plain 256 bytes
  application/pkcs7-signature [smime.p7s] 3429 bytes

```

Its contents are:

```

MIME-Version: 1.0
Content-Type: multipart/signed;
  protocol="application/pkcs7-signature"; boundary="8a8";
  micalg="sha-256"
Subject: smime-multipart-wrapped
Message-ID: <smime-multipart-wrapped@lhp.example>
From: Alice <alice@smime.example>
To: Bob <bob@smime.example>
Date: Sat, 20 Feb 2021 10:05:02 -0500
User-Agent: Sample MUA Version 1.0

```

--8a8

```

MIME-Version: 1.0
Content-Type: message/rfc822; protected-headers="wrapped"

```

Content-Disposition: inline

MIME-Version: 1.0
Content-Type: text/plain; charset="utf-8"
Content-Transfer-Encoding: 7bit
Subject: smime-multipart-wrapped
Message-ID: <smime-multipart-wrapped@lhp.example>
From: Alice <alice@smime.example>
To: Bob <bob@smime.example>
Date: Sat, 20 Feb 2021 10:05:02 -0500
User-Agent: Sample MUA Version 1.0

This is the smime-multipart-wrapped message.

This is a signed-only S/MIME message via PKCS#7 detached signature (multipart/signed). The payload is a text/plain message. It uses the Wrapped Message header protection scheme.

--

Alice
alice@smime.example

--8a8

Content-Transfer-Encoding: base64
Content-Type: application/pkcs7-signature; name="smime.p7s"

MIIJ4AYJKoZIhvcNAQcCoIIJ0TCCc0CAQExDTALBg1ghkgBZQMEAgEwCwYJKoZI
hvcNAQcBoIIHpjCCA88wggK3oAMCAQICEw8tJb0ROZdKzkJU6HuPTQGirQwDQYJ
KoZIhvcNAQENBQAwVTENMA8GA1UEChMESUVURjERMA8GA1UECzMITEFNMFgV0cx
MTAvBgNVBAMTKFNhbXBsZSBMQUU1QUYBSU0EgQ2VydG1maWNhdG1vbiBBdXRob3Jp
dHkwIBcNMTEwMDY1NDE4WhgPMjA1MjA5MjcwNjU0MThaMDsxDTALBgNVBAoT
BE1FVEYxETAPBgNVBAsTCExBTVBTIFdHMRcwFQYDVoQDEw5BbGljZSBMb3Z1bGFj
ZTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBBAJqVKfqlwaLjj+gBUCfk
acKTg8cc20tJ9ZSed6U3jUoiZVpMLcP3MUKtLeLg9r1mAfID1B/wlbdmadXPmrsz
yidmbuZmOpB5voVQfiLYy3iOx7YOqzXr16udP07k0sV+UdSNRFxrfKeoQEFXgOa
Gdmnx4OG/e3p1fIKM0dPzzLoOAJF5m500xzXPL74zFCWp2f1ZkuE4A6141koaZXC
N5XL7wWTLMLenF9Byb5ksKqUuqEHAMd1nmoNMgjY9VfVfcrv9w43GG8FtpSX+TWz
B2zNS2OF+XIVnzRG5DeoULq8v88Z5bLpIJ/nx26r8A4SSwIBaVv4wPxAf1iPsIVK
arUCAwEAAaOBrzCBrdAMBgNVHRMBAf8EAjAAMBcGA1UdIAQQMA4wDAYKYIZIAWUD
AgEwATAeBgNVHREEFzAVgRNhbGljZUBzbW1tZS5leGFtcGx1MBMGA1UdJQQMMAoG
CCsGAQUFBwMEMA4GA1UdDwEB/wQEAWIFIDAdBgNVHQ4EFgQUo1NB1UQ8gCkVfAEj
8OeOr83zdW8wHwYDVR0jBBgwFoAUKTCOfAcXDKfxCSH1NhpNHGh29FkwDQYJKoZI
hvcNAQENBQADggEBAIFJeKcCsTKcFqQMpTryuJRgzJdYA+R9eBAuDLsatbtKt14F
zkgRyOg31/+Cw7H8e30iLrPIF1WN1qjHrjgOyIs5AQ/hgxLvLir3hEUV2Z3MRsMt
jH2x9SG91PEM046gfPnc9gMGHjMTg1qvaKcLQP5UzpEYPLror2X4P5uXxaP0LIZR
zWmkw1RF7FOD7PFB5v94M5274XYxW2W4uKgd7QGnUZROSvSYkGiWdp1JhqXwfdZ8
A0enITGXnoEkAFvviCqh64P1hIeMorj36pgL19oWZD6YrzSWHUz1F00juyuOfQs
qm6hvrDTqNpHNZ015FOURza1SkCvi9GFmNUPoVgwggPPMIICt6ADAgECAhM3QQV5

```

7XV/QqmiXDr0+GrOmqnXMA0GCSqGS Ib3DQEBDQUAMFUxD TALBgNVBAoTBELFVEYx
ETAPBgNVBA sTCExBTVBTIFdHMTEwLwYDVQQDEyhTYW1wbGUgTEFNUFMgU1NBIEN1
cnRpZmljYXRpb24gQXV0aG9yaXR5MCAXDTE5MTEyMDA2NTQxOFoYDzIwNTIwOTI3
MDY1NDE4WjA7MQ0wCwYDVQQKEwRJRVRGMREwDwYDVQQLEWhMQU1QUyBXRzEXMBUG
A1UEAxMOQWxpY2UgTG92ZWxhY2UwgG EiMA0GCSqGS Ib3DQEBAQUAA4IBDwAwggEK
AoIBAQC09InoWDgWPk2af0+Sti jSNOR8K/hN8D+1078oullsk4ASvSwjsCNo7sHU
a4xQUl5JO6VqY18LANwORjrc9BaX4MguzsbFXBe6uFh1mVpXmFxSpUByQ+950MFz
/evPgP96wV+z4TtAwW2Z34rTiz4DxMI07XYNFUE0ls/gkUP2GxzYms02kaYWTut3
SryCqeHEFBzFk4urMk4xrIJC3CzWrus2Q0FHbBlfkGKN5wXVgkWFfiOucfCn+IQ
saqpo1d3f9jSkbtAV5w3vzfog8919MxKI9H614KuElnAtJ7BtZcs17dUy9u9COgE
yKriVokFQgqQ7XNDU+r3SeOWwks7AgMBAAGjga8wgawwDAYDVR0TAQH/BAIwADAX
BgNVHSAEEDAOMA wGCMCGSAFlAwIBMAEwHgYDVR0RBBCwFYETYWxpY2VAC21pbWUu
ZXhhbXBsZTATBgNVHSEDDAKBggrBgEFBQcDBDAOBgNVHQ8BAf8EBAMCBsAwHQYD
VR0OBBYEFLv2zLlTHQYSHJeuKWgQENMgZmZzMB8GA1UdLwQYMBaAFJEWjnwHFwyn
8QkoZTYaZxxodvRZMA0GCSqGS Ib3DQEBDQUAA4IBAQBziaI2p86poGkjd/4KkkOH
G25nY/0eNARD6/0F0/sYonX2doizcGMk53riugAocCn5zbzhW/JVdYn30UxfyrZl
RAzEf7GHqgB/NyjOad3pdpVYeDh4ciNKjbs+aEoTWgAkoqENT1sRx1cvb7HVX524
bKZal0PTUN1m6QpivtqDIdqGJdGf8L1zLfxBuo2zL3HR+M9CDr4Opq2JckzP0Qhp
7poIccGE6I9Tsg+RrOA9iCqsPn1+Tg8YedjGzUWF07rNmT0TzPCVzUAuBlr+JJtz
OKypyQ3eoZ6EPazXqMyHAVcsmGI364IOA0b8PSrJNtjh+AgJ5QfH+0e7NsZnEm
MYICADCCAfwCAQEwbDBVMQ0wCwYDVQQKEwRJRVRGMREwDwYDVQQLEWhMQU1QUyBX
RzExMC8GA1UEAxMoU2FtcGx1IEExBTvBTIFJTQSBDZXJ0aWZpY2F0aW9uIEF1dGhv
cm10eQITN0EFee11f0Kpolw69Phqzpp1zALBg1ghkgBZQMEAgGgaTAYBgkqhkiG
9w0BCQMx CWYJKoZIhvcNAQcBMBwGCSqGS Ib3DQEBJTEPFw0yMTAyMjAxNTA1MDJa
MC8GCSqGS Ib3DQEBJDEiBCALOMrQogvVsAh7w8dZ49veRaAFhTQ49VmGVz+1eTbz
tjANBgkqhkiG9w0BAQEFAASCAQA/IjhMNkM+NpI3wGfQyDC1EMkiUG5SQ88JC0zc
Xaz46K2nncQh+PW9TChvi9V9VR9EvKx7sh0dBnjhogrMTH3V1mZPgyL2HdsfLvXa
WHmHQmbTnsZH8+kqQLdOZG/zbQMgR3sSv992f6ShxZNdazwGSf5s7Hs6+an6yy24
VtJqhT5xHHvMfDLUVW4sXwRugWKohiW+cjZ16SQ5zP14KJBpriMWv8A/4sJv5aC2
ImraEATJ1gIse53X6XPdt/+9BsXOrvbIvXRibgMJBK8gIz6aO72n/dvmlfHjdBXv
9t75zqN+O821RiUiSbBoaB3FP0sl3prsz4QRr3Yv7vpv/HoR

```

--8a8--

B.2.3. S/MIME Signed-only signedData Over a Simple Message, Injected Headers

This is a signed-only S/MIME message via PKCS#7 signedData. The payload is a text/plain message. It uses the Injected Headers header protection scheme.

It has the following structure:

```

application/pkcs7-mime [smime.p7m] 4234 bytes
(unwraps to)
text/plain 239 bytes

```

Its contents are:

VQQLewhMQU1QUyBXRzEXMBUGA1UEAxMOQWxpY2UgTG92ZWxhY2UwgGElMA0GCSqG
 SIb3DQEBAQUAA4IBDwAwggEKAoIBAQC09InoWDgWPk2af0+StijSNOR8K/hN8D+l
 078oullsk4ASvSwjsCNo7sHUa4xQU15JO6VqY18LANwORjrc9BaX4MguzsbFXBe6
 uFhlmVpXmFxpUByQ+950MFz/evPgP96wV+z4TtAwW2Z34rTiz4DxMI07XYNFUEO
 ls/gkUP2GxzYmsO2kaYWTut3SryCqeHEFbZfKb4urMk4xrIJC3CzWruS2Q0FHbBl
 fkgKN5wXVgkWFfiOucfCn+IQsaqppold3f9jSkbtAV5w3vzfog8919MxKI9H6l4Ku
 ElnAtJ7BtZcsl7dUy9u9COgEykRiVokFQgqQ7XNDU+r3SeOWwks7AgMBAAGjga8w
 gawwDAYDVR0TAQH/BAIwADAXBgNVHSAEEDAOMAwwGCMCGSAFLAwIBMAEwHgYDVROR
 BBcwFYETWxpY2VAc2lpbWUuZXhhbXBsZTATBgNVHSUEDDAKBggrBgEFBQcDBDAO
 BgNVHQ8BAf8EBAMCBsAwHQYDVR0OBBYEFv2zLItHQYSHJeuKWqQENMgZmZzMB8G
 A1UdIwQYMBAAFEJEWjnWHFwyn8QkoZTYaZxxodvRZMA0GCSqGSIB3DQEBDQUAA4IB
 AQBziaI2p86poGkjd/4KkkOHG25nY/0eNARD6/0F0/sYonX2doizcGMk53riugAo
 cN5zbzhW/JVdYn30UxfyrZlRAzEf7GHqgB/NyjOad3pdpVYeDh4ciNKjbs+aEoT
 WgAkoqENt1sRx1cvb7HVX524bKZa1oPTUNlm6QpivtqDIdqGJdGf8L1zLFXBuoz
 L3HR+M9CDr40pQzP0Qhp7poIccGE6I9Tsg+RrOA9iCQsPn1+Tg8YedjGzUWF
 07rNmT0TzPCVzUAuBlr+JJtzOKypyQ3eoz6EPazXqMyHAVcsm0GI364IOA0b8PSr
 JNtjh+AqJ5QfH+0e7NSzNnEmMYICADCCAfwCAQEwbDBVMQ0wCwYDVQKKEwRJRVRG
 MREwDwYDVQQLewhMQU1QUyBXRzExMC8GA1UEAxMoU2FtCGxlIEExBTVBtIFJTQSBD
 ZXJ0aWZpY2F0aW9uIEFldGhvcml0eQITN0EFee1lf0Kpolw69PhqzppplzALBglg
 hkgBZQMEAgGgaTAYBgkqhkiG9w0BCQMxCwYJKoZIhvcNAQcBMwGCsGSIb3DQEJ
 BTEPFw0yMTAyMjAxNTA2MDJAMC8GCSqGSIB3DQEJBDEiBCBwJ1HsKaiXvrMR26xS
 /wrb+5CS85FLWuHRuKm85dkUFTANBgkqhkiG9w0BAQEFAASCAQBE/g/trAYogNeF
 9oD6esBshX+oPQP8AhmTNR5mdEi+YChauio4z941PIGHwPGGI220c1y1C68bMsjT
 HPlaumv6zhotJym5OtJH1nD0cOxeqMSP+/htEgb/YmOTs1tGL5W6MRDE2Qpk+ZT+
 skuKKBt98a/VQGEmyIZSTJV9SmiapvYDb9BA+KpuFZ0Yd/vMtTjqlDRBzadE9byX
 010GDNMBiqOeDeVcfU2j/rb3UELfJqSpiTqEST/JIq1PvZhr+En2ZOPfMA7BKjTm
 sl/sczGLBObDAJztOOG7oU83zowcKn0JNse2cKU2eQMAENTuahfaXzVrmbfsW665
 Mrfom9Z/

B.2.4. S/MIME Signed-only multipart/signed Over a Simple Message,
 Injected Headers

This is a signed-only S/MIME message via PKCS#7 detached signature (multipart/signed). The payload is a text/plain message. It uses the Injected Headers header protection scheme.

It has the following structure:

```
multipart/signed 4487 bytes
  text/plain 258 bytes
  application/pkcs7-signature [smime.p7s] 3429 bytes
```

Its contents are:

arUCAwEAAaOBrzCBrdAMBgNVHRMBAf8EAJAAMBcGA1UdIAQOMA4wDAYKYIZIAWUD
 AgEwATAeBgNVHREEFzAVgRNhbG1jZUBzbW1tZS5leGFtcGx1MBMGA1UdJQOMMAoG
 CCsGAQUFBwMEMA4GA1UdDwEB/wQEAWIFIDAdBgNVHQ4EFgQUo1NB1UQ8gCkVfAEj
 8OeOr83zdw8wHwYDVR0jBBgwFoAUkTCOfAcXDKfxCSh1NhpHGh29FkwDQYJKoZI
 hvCNAQENBQADggEBAIFJeKCCsTKcFqQMpTryujRGzJdYA+R9eBAuDLsatbtKt14F
 zkgRyOg31/+Cw7H8e30iLrP IF1WN1qjHrjgOyIs5AQ/hgxLvLir3hEUV2Z3MRsMt
 jH2x9SG91PEM046gfPnc9gMGHjMTg1qvaKcLQP5UzpeYPLror2X4P5uXxaP0LIZR
 zWmkw1RF7FOD7PFB5v94M5274XYxW2W4uKGD7QGnUZROsvSYkG1Wdp1JhqXwFDz8
 A0enITGXnoEkAFvvjiCqh64P1hIeMorj36pgL19oWZD6YrzSWHUz1F00juyuOfQs
 qm6hvrDTqNpHNZ015fOURza1SkCvi9GFmNUPoVgwgqPPMIICt6ADAgECAhM3QQV5
 7XV/QqmiXDr0+GrOmgnXMA0GCSqGS Ib3DQEBDQUAMFUxDTALBgNVBAoTBELFVEYx
 ETAPBgNVBAS TCEXBTvBTIFdHMTEwLwYDVQDEyhTYW1wbGUgTEFNuFMgU1NBIEN1
 cnRpZmljYXRpb24gQXV0aG9yaXR5MCAxZDTE5MTEyMDA2NTQxOFoYDzIwNTIwOTI3
 MDY1NDE4WjA7MQ0wCwYDVQQKEwRJRVRGMREwDwYDVQQLEWhMQU1QUyBXRzEXMBUG
 A1UEAxMOQWxpY2UgTG92ZWxhY2UwgGElMA0GCSqGS Ib3DQEBAQUAA4IBDwAwggEK
 AoIBAQC09InoWDgWPK2af0+StijsNOR8K/hN8D+1078oullsk4ASvSwjsCNo7sHU
 a4xQU15JO6VqY18LANwORjrc9BaX4MguzsbFXBe6uFh1mVpXmFxSpUByQ+950MFz
 /evPgP96wV+z4TtAwW2Z34rTiz4DxMI07XYNFUEOlS/gkUP2GxzYmsO2kaYWTut3
 SryCqeHEFbZfKb4urMk4xrIJC3CzWruS2Q0FhbBlfkgKN5wXVgkWFfiOucfCn+IQ
 saqp01d3f9jSkbtAV5w3vzfog8919MxKI9H614KuELnAtJ7BtZcs17dUy9u9CogE
 ykRiVokFQgqQ7XNDU+r3SeOWwks7AgMBAAAgja8wgawwDAYDVR0TAQH/BAIwADAX
 BgNVHSAEEDAOMAAGCmCGSAF1AwIBMAEwHgYDVR0RBBCwFYETYWxpY2VAc21pbWUu
 ZXhhbXBsZTATBgNVHSUEDDAKBggrBgEFBQcDBDAOBgNVHQ8BAf8EBAMCBsAwHQYD
 VR0OBBYEFV2zLzLitHQYSHJeuKWqQENMgZmZzMB8GA1UdIwQYMBAAFEJewjnwHFwyn
 8QkoZTYaZxxodvRZMA0GCSqGS Ib3DQEBDQUAA4IBAQBziaI2p86poGkjd/4KkkOH
 G25nY/0eNARD6/0F0/sYonX2doizcGMk53riugAocCn5zbzhW/JVdYn30UxfyrZ1
 RAzEf7GHqgB/NyjOad3pdpVYeDh4ciNKjbs+aEoTWgAkoqENT1sRx1cvb7HVX524
 bKZa1oPTUN1m6QpivtqDIdqGJdGf8L1zLFXBuo2zL3HR+M9CDr40ppq2JCKzP0Qhp
 7poIccGE6I9Tsg+RrOA9iCQsPn1+Tg8YedjGzUWF07rNmT0TzPCVzUAuBlr+JJtz
 OKypyQ3eoZ6EPazXqMyHAVcsm0GI364IOA0b8PSrJNtjh+AqJ5QfH+0e7NsZnNEm
 MYICADCCAfwCAQEwbDBVMQ0wCwYDVQQKEwRJRVRGMREwDwYDVQQLEWhMQU1QUyBX
 RzExMC8GA1UEAxMoU2FtcGx1IEExBTvBTIFJTQSBDZXJ0aWZpY2F0aW9uIEF1dGhv
 cm10eQITN0EFee11f0Kpolw69Phqzppp1zALBg1ghkgBZQMEAgGgaTAYBgkqhkiG
 9w0BCQMxCwYJKoZIhvcNAQcBMBwGCSqGS Ib3DQEBTEPFw0yMTAyMjAxNTA3MDJa
 MC8GCSqGS Ib3DQEBJDEiBCA6Rhu8s2iPcyWQk+TNKhP9ZHJ9+wulWjsMpAF1NXCE
 jDANBgkqhkiG9w0BAQEFAASCAQB4QMAYf42dnAelBRb2NotiixNgdjdSpVK75af6
 oND3UjdcWcd4bPbrfTzMQKp0FBP0ft91w2fWNgXwKbhd1cL3RWUmUq0zcnbI3XI
 86vWp79p+KwM/+SyDdfgudIRGjbs/tmKaBvaH89a8SvuxhNxxq/pxgDzpy/JWC8Er
 AUDTbKrNVsYD+MfzMy9B0T1K2YlKoQ6rV0N1n2nXbW0e+Ztv0a/getNKAEAP+5hE
 OQkq50RxUP9pI5kQ1NdU6zqCNhRjmd1wnMxn45K+hfy8cxwwemFn94PgDGpPG4mB
 yRXQPj+5oyduWiHRMLXG1+fs4tqxHZXN+WaUHvSIDqNXK3rj

--file--


```

DQUdsGV+SAo3nBdWCRYV+I65x8Kf4hCxqqmjV3d/2NKRu0BXnDe/N+iDz3X0zEoj
0fqXgq4SWcC0nsG1lyyXt1TL270I6ATKRGJWiQVCCpDtc0NT6vdJ45bCSzsCAwEA
AaOBrzCBrdAMBgNVHRMBAf8EAjAAMBcGA1UdIAQQMA4wDAYKYIZIAWUDAgEwATAe
BgNVHREEFzAVgRNhbG1jZUBzbWltZS5leGFtcGxlMBMGAlUdJQQMMAoGCCsGAQUF
BwMEMA4GA1UdDwEB/wQEAwIGwDAdBgNVHQ4EFgQUu/bMsi0dBhIc164papAQ0yBm
ZnMwHwYDVROjBBgwFoAUkTCOfAcXDKfxCSHlNhpnHGh29FkwDQYJKoZIhvcNAQEN
BQADggEBAHOJoJanzqmgaSN3/gqSQ4cbbmdj/R40BEP+r+gXT+xiidfZ2iLnwYyTn
euK6AChwKfnNvOFb8lV1iffRtF/KtmVEDMR/sYeqAH83KM5p3e12lVh4OHhyI0qN
uz5oShNaACSioQ23WxHGvy9vsdVfnbhsp1rWg9NQ2WbpCmK+2oMh2oYl0Z/wvXmt
9cG6jbMvcdH4z0IOvg6mrYkKTM/RCGnumghxwYToj1OyD5Gs4D2IJCw+fx50Dxh5
2MbNRYXTus2ZPRPM8JXNQc4Gwv4km3M4rKnJDD6hnoQ9rNeozIcBVyybQYjfrgg4
DRvW9Ksk22OH4ConlB8f7R7s1LM2cSYxggIAMIIB/AIBATBsMFUxDALBgNVBAoT
BE1FVEYxETAPBgNVBAsTCExBTVBTIFdHMTEwLwYDVQQDEyhTYW1wbGUgTEFNUFMg
U1NBIEENlcnRpZmljYXRpb24gQXV0aG9yaXR5AHM3QQV57XV/QqmiXDr0+GrOmqnX
MAsgCWCgsAF1AwQCAaBpMBGCSqGSIb3DQEJAzELBgkqhkiG9w0BBwEwHAYJKoZI
hvcNAQkFMQ8XDTIxMDIyMDE3MDQwMlowLwYJKoZIhvcNAQkEMSIEICsRogMUJrtS
GAERSFiPMhqWk+9misjv48XcSNJBKUj5MA0GCSqGSIb3DQEBQUABIIBALJCpFEK
FQ+M1YQIuTcVEHr/K/w/8ht4pOy4BmEE+q3yZUBATHt37DxdZUXRZjUB52FdsWed
agkt3DjtFzJwRiDSteChrjrA/0jbFVOuV/9VBm0VGGfodRTovS+6wH+yJNAXHSW9
p1GXmPcDFAtn5wr69zBNCX5mKU6bwcaVX41S7/fmcD1BNSQ45Ex+RrXRhMX/vG2A
tgu01LuRSCvGgz719968R5D3obEtZwUi8uSOpv13XqThZC5Q4NMg68UNgNb//OT
PuaqlMOVhWhSkTNKjbtv2P/MifHWXj9TYHkRc915k707LqWj3yWNFR7tpVO07n0+
hTEzoJRFKuxJlQ4=

```

B.2.6. S/MIME Signed-only multipart/signed Over a Complex Message, Wrapped Message

This is a signed-only S/MIME message via PKCS#7 detached signature (multipart/signed). The payload is a multipart/alternative message with an inline image/png attachment. It uses the Wrapped Message header protection scheme.

It has the following structure:

```

multipart/signed 5653 bytes
  message/rfc822 inline 1747 bytes
  multipart/mixed 1642 bytes
    multipart/alternative 1002 bytes
      text/plain 310 bytes
      text/html 408 bytes
      image/png inline 232 bytes
      application/pkcs7-signature [smime.p7s] 3429 bytes

```

Its contents are:

```
MIME-Version: 1.0
Content-Type: multipart/signed;
  protocol="application/pkcs7-signature"; boundary="aa9";
  micalg="sha-256"
Subject: smime-multipart-complex-wrapped
Message-ID: <smime-multipart-complex-wrapped@lhp.example>
From: Alice <alice@smime.example>
To: Bob <bob@smime.example>
Date: Sat, 20 Feb 2021 12:05:02 -0500
User-Agent: Sample MUA Version 1.0
```

--aa9

```
MIME-Version: 1.0
Content-Type: message/rfc822; protected-headers="wrapped"
Content-Disposition: inline
```

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="a30"
Subject: smime-multipart-complex-wrapped
Message-ID: <smime-multipart-complex-wrapped@lhp.example>
From: Alice <alice@smime.example>
To: Bob <bob@smime.example>
Date: Sat, 20 Feb 2021 12:05:02 -0500
User-Agent: Sample MUA Version 1.0
```

--a30

```
MIME-Version: 1.0
Content-Type: multipart/alternative; boundary="844"
```

--844

```
Content-Type: text/plain; charset="us-ascii"
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit
```

This is the smime-multipart-complex-wrapped message.

This is a signed-only S/MIME message via PKCS#7 detached signature (multipart/signed). The payload is a multipart/alternative message with an inline image/png attachment. It uses the Wrapped Message header protection scheme.

--

```
Alice
alice@smime.example
--844
Content-Type: text/html; charset="us-ascii"
MIME-Version: 1.0
```



```

A0enITGXnoEkAFvvjiCqh64P1hIeMorj36pgL19oWZD6YrzSWHUz1F00juyuOfQs
qm6hvrDTqNpHNZ015fOURza1SkCvi9GFmNUPoVgwgPpMIICt6ADAgECAhM3QQV5
7XV/QqmiXDr0+GrOmgnXMA0GCSqGS Ib3DQEBDQUAMFUxDTALBgNVBAoTBELFVEYx
ETAPBgNVBAsTCEExBTBVTIFdHMTEwLwYDVQQDEyhTYW1wbGUgTEFNUFMgU1NBIEN1
cnRpZmljYXRpb24gQXV0aG9yaXR5MCAXDTE5MTEyMDA2NTQxOFoYDzIwNTIwOTI3
MDY1NDE4WjA7MQ0wCwYDVQQKEwRJRVRGMREwDwYDVQQLEwhMQU1QUyBXRzEXMBUG
A1UEAxMOQWxpY2UgTG92ZWxhY2UwgGEMAI0GCSqGS Ib3DQEBAQUAA4IBDwAwggEK
AoIBAQC09InoWDgWPk2af0+StijSNOR8K/hN8D+1078oullsk4ASvSwjsCNo7sHU
a4xQU15J06VqY18LANwORjrc9BaX4MguzsbFXBe6uFh1mVpXmFfxSpUByQ+950MFz
/evPgP96wV+z4TtAwWZ234rTiz4DxMI07XYNFUEOlS/gkUP2GxzymsO2kaYWTut3
SryCqeHEFbZfKb4urMk4xrIJC3CzWruS2Q0FHbBlfkgKN5wXVgkWFfiOucfCn+IQ
saqpold3f9jSkbtAV5w3vzfog8919MxKI9H614KuElnAtJ7BtZcs17dUy9u9COgE
yKriVokFQgqQ7XNDU+r3SeOWwks7AgMBAAGjga8wgawwDAYDVR0TAQH/BAIwADAX
BgNVHSAEEDAOMAAGCmCGSAFlawIBMAEwHgyYDVRORBBCwFYETYWxpY2VAc21pbWUu
ZXhhbXBsZTATBgNVHSUEDDAKBggrBgEFBQcDBDAOBgNVHQ8BAf8EBAMCBsAwHQYD
VR0OBBYEFv2zLlItHQYSHJeuKWqQENMgZmZzMB8GA1UdIwQYMBaAFJEWjnwHFwyn
8QkoZTYaZxxodvRZMA0GCSqGS Ib3DQEBDQUAA4IBAQBziaI2p86poGkjd/4KkkOH
G25nY/0eNARD6/oF0/sYonX2doizcGmk53riugAocCn5zbzhW/JVdYn30UxfyRz1
RAzEf7GHqgB/NyjOad3pdpVYeDh4ciNKjbs+aEoTWgAkoqENTlsRx1cvb7HVX524
bKZa1oPTUNlm6QpivtqDIdqGJdGF8L1zLFXBuozL3HR+M9CDr4Opq2JckzP0Qhp
7poIccGE6I9Tsq+RrOA9iCqsPn1+Tg8YedjGzUWF07rNmT0TzPCVzUAuBlr+JJtz
OKypyQ3eoz6EPazXqMyHAVcsm0GI364IOA0b8PSrJNtjh+AqJ5QfH+0e7NSzNnEm
MYICADCCafwCAQEwbDBVMQ0wCwYDVQQKEwRJRVRGMREwDwYDVQQLEwhMQU1QUyBX
RzExMC8GA1UEAxMoU2FtcGx1IEExBTBVTIFJTSQBDZXJ0aWZpY2F0aW9uIEF1dGhv
cm10eQITN0EFee1lf0Kpolw69PhqzppplzALBglghkgBZQMEAgGgATAYBgkqhkiG
9w0BCQMxCwYJKoZIhvcNAQcBMBwGCSqGS Ib3DQEBTEPFw0yMTAyMjAxNzA1MDJa
MC8GCSqGS Ib3DQEBTEPDEiBCDvCBOZJKngosmsBz3B3if2ErlyiRyR1KnTpWbe6AN0
fzANBgkqhkiG9w0BAQEFAASCAQB6Xc+YUIEUCqF3vq1ZTP41u/jEG330+bc5jw7D
VLUBkQ+AI6c6602LAgMwX17VuBdbgHecf59trY2F47Wr8N1cbTcAq0jN54tqrhri
8cL4YzS8YGH0vLrDdwilChjs0N1+t5nQ8Rya+rdGqseE0TK38P/K28cnU3udgTjb
6E/QcopIlnLaaji+x5qjRHq10Yt9tbA5F1L9vgqgu7Zf9w55tZie9cESnVZpud/1
+zqsKdfj4ndnMDFzrUtXztY2e1f/Y8EVjSIVtY+ZeYuldtGhPpvk/N3koxZ1yL2Z
mrPQemZ0C2bIet7T1vv7lFCUtU0bdyHoHBvXI70hbCmGmak3

```

--aa9--

B.2.7. S/MIME Signed-only signedData Over a Complex Message, Injected Headers

This is a signed-only S/MIME message via PKCS#7 signedData. The payload is a multipart/alternative message with an inline image/png attachment. It uses the Injected Headers header protection scheme.

It has the following structure:

eGftcGx1PC90dD48L3A+PC9ib2R5PjwvaHRtbD4NCi0tOTA3LS0NCg0KLS0zOTUN
 CkNvbnRlbnQtVHlwZTogaW1hZ2UvcG5nDQpDb250ZW50LVRyYW5zZmVyLUVuY29k
 aW5nOiBiYXN1bnJONCkNvbnRlbnQtRG1zcG9zaXRpb246IGlubGluZQ0KDQppVkJP
 UncwS0dnb0FBQUFOU1VoRVVnQUFBQ1FBQUFBVUNBWUFBUUNOaVIwTkFBQUFjRWxY
 UVZSNDJ1V1RPeGJBDQpNQWdTNzM5bk8zVHBSdzIwZHFwYmZBU1FFak95d2l3WW5D
 dGtES25iY0xrNjZzcWxUK3p0OWNpZGtFKzZLd2taDQpzZ3J6ZmNxVk1wTDJqbzA0
 NDdnWURwZUFyaytPbkpIa0loQWZUUFJpY2loQWY1WUpydzd2anYwWldSV00vdWxp
 DQp2ZFBNMVFAMmtERDl4cHBkOHdBQUFBQkpSVTVFcmTKZ2dnPT0NCg0KLS0zOTUt
 LQ0KoIiHjPjCCA88wggK3oAMCAQICEw8tJb0ROZdKzkJU6HuPTQGirQwDQYJKoZI
 hvCNBQENBQAwVTENMAsgA1UEChMESUVURjERMA8GA1UECXMITEFNUFMgV0cxMTAv
 BgNVBAMTKFNhbXBsZSBMQU1QUyBSU0EgQ2VydG1maWNhdGlvbiBBdXR0b3JpdHkw
 IBcNMTkxMTIwMDY1NDE4WhgPMjA1MjA5MjcwNjU0MThaMDsxDTALBgNVBAoTBELF
 VEYxETAPBgNVBAsTCExBTBVTIFdHMRcwFQYDVQQDEw5BbG1jZSBMbz3ZlBGFjZTCC
 ASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBBAJqVKfLwaLjj+gBUCfkacKT
 g8cc20tJ9ZSed6U3jUoiZVpMLcP3MUKtLeLg9r1mAfID1B/wlbdmadXPmrszyidm
 buZmOpB5voVQfiLYy3iOx7YQzXr16udP07k0sV+UdSNRFxrfKeoQEFXgOaGdmn
 x4OG/e3p1fIKM0dPzZLoAJF5m500xzXPL74zFCWp2f1ZkuE4A6141koaZXC5XL
 7wWTLMLenF9Byb5ksKqUuqEHAMd1nmoNMgjY9VfVfcrv9w43GG8FtpSX+TWzB2zN
 S2OF+XIVnzRG5DeoULq8v88Z5bLpIJ/nx26r8A4SSwIBaVv4wPxAfliPsIVKarUC
 AwEAAaOBrzCBrdAMBgNVHRMBAf8EAJAAMBcGA1UdIAQQMA4wDAYKYIZIAWUDAgEw
 ATAeBgNVHREEFzAVGjRNhbG1jZUBzbWltZS5leGftcGx1MBMGA1UdJQQMMAoGCCsG
 AQUFBwMEMA4GA1UdDwEB/wQEAwIFIDAdBgNVHQ4EFgQUo1NB1UQ8gCkVfAEj80eO
 r83zdw8wHwYDVR0jBBGwFoAUKTCOfAcXDKfxCSH1NhpHGh29FkwDQYJKoZIhvcN
 AQENBQADggEBAIFJeKCsTKcFqQMPtryuJRgzJdYA+R9eBAuDLsatbtKt14FzkgR
 yOg31/+Cw7H8e30iLrPIFLWN1qjHrjGyOIs5AQ/hgxLvLir3hEUV2Z3MRsMtjH2x
 9SG91PEM046gfPnc9gMGHjMTg1qvaKcLQP5UzpeYPLror2X4P5uXxaP0LIZRzWmk
 w1RF7FOD7Pfb5v94M5274XYxW2W4uKgd7QgnUZROsvSYkGiWdp1JhqXwfdz8A0en
 ITGXnoEKAFvvjiCqh64P1hIeMorj36pgL19oWZD6YrzSWHUz1F00juyuOfQsqm6h
 vrDTqNpHNZ015fOURza1SkCvi9GFmNUPoVgwgqPPMIICt6ADAgECAhM3QQV57XV/
 QqmiXDr0+GrOmqnXMA0GCSqGS Ib3DQEBAQUAMFUXDTALBgNVBAoTBELFVEYxETAP
 BgNVBAsTCExBTBVTIFdHMTEwLWYDVQQDEyhTYW1wbGUgTEFNUFMgU1NB1EN1cnRp
 ZmljYXRpb24gQXV0aG9yaXR5MCAXDTE5MTEyMDA2NTQxOFoYDzIwNTIwOTI3MDY1
 NDE4WjA7MQ0wCwYDVQQKEWRJRVRGMREwDwYDVQQLEWhMQU1QUyBXRzEXMBUGA1UE
 AxMOQWxpY2UgTG92ZWxhY2UwggEiMA0GCSqGS Ib3DQEBAQUAA4IBDwAwggEKAoIB
 AQc09InoWDgWPK2af0+StijSNOR8K/hN8D+1078oullsk4ASvSwjsCNo7sHUa4xQ
 U15JO6VqY18LANwORjrc9BaX4MguzsbfXBe6uFh1mVpXmFxpSpUByQ+950MFz/evP
 gP96wV+z4TtAwW2Z34rTiz4DxMI07XYNFUEOls/gkUP2GxzymsO2kaYWTut3SryC
 qeHEFbZfK4urMk4xrlJC3CzWruS2Q0FHbBlfkqKN5wXVgkWFfiOucfCn+IQsaqp
 o1d3f9jSkbtAV5w3vzfog8919MxKI9H614KuElNAtJ7BtZcs17dUy9u9C0gEykRi
 VokFQgqQ7XNDU+r3SeOWwks7AgMBAAAgjga8wgawwDAYDVR0TAQH/BAIwADAXBgNV
 HSAEEDAOMAAGCmCGSAFlAwIBMAEwHgYDVR0RBBCwFYETyWxpY2VAc2lpbWUuZUxhh
 bXBsZSBMQU1QUyBSU0EgQ2VydG1maWNhdGlvbiBBdXR0b3JpdHkwIBcNMTkxMTIw
 MDY1NDE4WhgPMjA1MjA5MjcwNjU0MThaMDsxDTALBgNVBAoTBELFVEYxETAPBgNV
 BAsTCExBTBVTIFdHMTEwLWYDVQQDEyhTYW1wbGUgTEFNUFMgU1NB1EN1cnRpZmlj
 YXRpb24gQXV0aG9yaXR5MCAXDTE5MTEyMDA2NTQxOFoYDzIwNTIwOTI3MDY1NDE4
 WjA7MQ0wCwYDVQQKEWRJRVRGMREwDwYDVQQLEWhMQU1QUyBXRzEXMBUGA1UEAxMO
 QWxpY2UgTG92ZWxhY2UwggEiMA0GCSqGS Ib3DQEBAQUAA4IBAQBziaI2p86poGkj/
 4KkkOHG25nY/0eNARD6/0F0/sYonX2doizcGmk53riugAocCn5zbzhW/JVdYn30Uxfyr
 Z1RAzEf7GHqgB/NyjOad3pdpVYeDh4ciNKjbs+aEoTWgAkoqENT1sRx1cvb7HVX524b
 KZa1oPTUN1m6QpivtqDIdqGJdGf8L1zLFXBuo2zL3HR+M9CDr4Opq2JckzP0Qhp7poI
 ccGE6I9Tsg+RrOA9iCQsPnl+Tg8YedjGzUWF07rNmT0TzPCVzUAUblr+JJtZOKyp

```

yQ3eoZ6EPazXqMyHAVcsm0GI364IOA0b8PSrJNtjh+AqJ5QfH+0e7NSzNnEmMYIC
ADCCAfWCAQEwbDBVMQ0wCwYDVQKKEwRJRVRGMREwDwYDVQLEwhMQU1QUyBXRzEx
MC8GA1UEAxMoU2FtcGx1IEExBTBVTIFJTQSBDZXXJ0aWZpY2F0aW9uIEF1dGhvcml0
eQITN0EFee11f0Kpolw69Phqzpp1zALBglghkgBZQMEAgGgaTAYBgkqhkiG9w0B
CQMxCwYJKoZIhvcNAQcBMBwGCSqGSIb3DQEJBTEPFw0yMTAyMjAxNzA2MDJAMC8G
CSqGSIb3DQEJBDEiBCC84gf/+no5va6ErXhHIk1xELMQNWg9BUh8E1M78W5u5TAN
BgkqhkiG9w0BAQEFAASCAQB+q8buLwucKfPrBoXxKP7ZaJ/ifg8Y4Axf84AhNJXC
+NwzThUSgq12Fn9cdSVO858oDrWDSndd/zwgab0TgQZ+64atwiQ7bVTDkG8qgeT+
I/R1I8jGOCUTpkKcK34tOYbmhkc7/2BLITc3qOAxuN+1rsWVL2NF8LFGH9RbfzRu
WfVqAMyfa09DRr1PeFD0DQnjAGti37M8/WvftXixxOAevVmFUWbpnFiwvSwdrt0
CKquQ1NYbFAvxOawxLU0jFqhIgw10+fU4jqQDUkUVSKFiw1/dK+7j1ZC6sCXf3Ys
oHRhxqY/bSsgXn1DUWSDjhae3Hn1ZuoVXLJDHGcd6oSR

```

B.2.8. S/MIME Signed-only multipart/signed Over a Complex Message, Injected Headers

This is a signed-only S/MIME message via PKCS#7 detached signature (multipart/signed). The payload is a multipart/alternative message with an inline image/png attachment. It uses the Injected Headers header protection scheme.

It has the following structure:

```

multipart/signed 5580 bytes
  multipart/mixed 1672 bytes
    multipart/alternative 1006 bytes
      text/plain 312 bytes
      text/html 410 bytes
      image/png inline 232 bytes
      application/pkcs7-signature [smime.p7s] 3429 bytes

```

Its contents are:

```

MIME-Version: 1.0
Content-Type: multipart/signed;
  protocol="application/pkcs7-signature"; boundary="f91";
  micalg="sha-256"
Subject: smime-multipart-complex-injected
Message-ID: <smime-multipart-complex-injected@lhp.example>
From: Alice <alice@smime.example>
To: Bob <bob@smime.example>
Date: Sat, 20 Feb 2021 12:07:02 -0500
User-Agent: Sample MUA Version 1.0

```

```
--f91
```

```

MIME-Version: 1.0
Subject: smime-multipart-complex-injected
Message-ID: <smime-multipart-complex-injected@lhp.example>

```

From: Alice <alice@smime.example>
To: Bob <bob@smime.example>
Date: Sat, 20 Feb 2021 12:07:02 -0500
User-Agent: Sample MUA Version 1.0
Content-Type: multipart/mixed; boundary="099"; protected-headers="v1"

--099

MIME-Version: 1.0
Content-Type: multipart/alternative; boundary="9a5"

--9a5

Content-Type: text/plain; charset="us-ascii"
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit

This is the smime-multipart-complex-injected message.

This is a signed-only S/MIME message via PKCS#7 detached signature (multipart/signed). The payload is a multipart/alternative message with an inline image/png attachment. It uses the Injected Headers header protection scheme.

--

Alice
alice@smime.example
--9a5
Content-Type: text/html; charset="us-ascii"
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit

```
<html><head><title></title></head><body>  
<p>This is the <b>smime-multipart-complex-injected</b> message.</p>  
<p>This is a signed-only S/MIME message via PKCS#7 detached  
signature (multipart/signed). The payload is a  
multipart/alternative message with an inline image/png  
attachment. It uses the Injected Headers header protection  
scheme.</p>  
<p><tt>-- <br/>Alice<br/>alice@smime.example</tt></p></body></html>  
--9a5--
```

--099

Content-Type: image/png
Content-Transfer-Encoding: base64
Content-Disposition: inline

iVBORw0KGGoAAAANSUhEUgAAABQAAAAUCAyAAACNiR0NAAAACeLEQVR42uVTOxbA
MAGS739nO3TpRw20dqpbfARQEjOywiwYnCtkDKnbcLk66sqlT+zt9cidkE+6KwkZ


```

bKZa1oPTUN1m6QpivtqDIdqGJdGf8L1zLFXBuo2zL3HR+M9CDr40Ppq2JckzP0Qhp
7poIccGE6I9Tsg+RrOA9iCQsPn1+Tg8YedjGzUWF07rNmT0TzPCVzUAuBlr+JJtz
OKypyQ3eoZ6EPazXqMyHAVcsm0GI364IOA0b8PSrJNtjh+AqJ5QfH+0e7NSzNnEm
MYICADCCAfwCAQEwbDBVMQ0wCwYDVQQKEwRJRVRGMREwDwYDVQQLEwhMQU1QUyBX
RzExMC8GA1UEAxMoU2FtcGx1IEExBTBVTIFJTQSBDZXJ0aWZpY2F0aW9uIEF1dGhv
cml0eQITN0EFee1lf0Kpolw69PhqzppplzALBglghkgBZQMEAgGgaTAYBgkqhkiG
9w0BCQMxCwYJKoZIhvcNAQcBMBwGCSqGSIb3DQEJBTEPFw0yMTAyMjAxNzA3MDJa
MC8GCSqGSIb3DQEJBDEiBCDzzjU9zkYamvSgC05wewF4LgTekLa4P8khUZ1HRNkO
GzANBgkqhkiG9w0BAQEFAASCACFAaiW0MvY2tnagCpthNu6sAL22/BBu2BS5XY0
vTH4/MtLzU4lSokfcs8lgpXmE852prfBZfyoBiOtKZF6TkW59XPiEx4TfBZ+pFwb
MaJbZ5Kil2GpqKib2sEKbaNHaUY0H+vixz3NP6lo2Izras33cw4Z7FE24qs3zTAA
1WYTF8rtPhXVW9rFLumBOF8LgGKPTh4mjWrAEcaqmqmscixTJ5yp5DjHMF9Xv
/HVi9lOJJ5BlYOQOL/jWPxQorYJAP62HwEEzz7/GE24hm43pK8uHT5DPHiG+gZZL
35qcfe8j50JVLtG2wcRH/aKhat12MnPFMqnJGwugLv4rww5

```

--f91--

B.3. Encrypted-and-signed Messages

These messages are encrypted and signed. They use PKCS#7 signedData inside envelopedData, with different header protection schemes and different Header Confidentiality Policies.

B.3.1. S/MIME Encrypted and Signed Over a Simple Message, Wrapped Message With hcp_minimal

This is a encrypted and signed S/MIME message using PKCS#7 envelopedData around signedData. The payload is a text/plain message. It uses the Wrapped Message header protection scheme with the hcp_minimal Header Confidentiality Policy.

It has the following structure:

```

application/pkcs7-mime [smime.p7m] 7540 bytes
  (decrypts to)
  application/pkcs7-mime [smime.p7m] 4580 bytes
    (unwraps to)
    message/rfc822 inline 783 bytes
      text/plain 321 bytes

```

Its contents are:

```

Content-Transfer-Encoding: base64
Content-Type: application/pkcs7-mime; name="smime.p7m";
  smime-type="enveloped-data"
Subject: [...]
Message-ID: <smime-enc-signed-wrapped-minimal@lhp.example>
From: Alice <alice@smime.example>

```

To: Bob <bob@smime.example>
Date: Sat, 20 Feb 2021 10:08:02 -0500
User-Agent: Sample MUA Version 1.0

MIIVvAYJKoZIhvcNAQcDoIIIVrTCCFakCAQAxggMQMIIBhAIBADBsMFUxDTALBgNV
BAoTBELFVEYxETAPBgNVBAsTCExBTvBTIFdHMTEwLwYDVQQDEyhTYW1wbGUgTEFN
UFMgU1NBIEN1cnRpb24gQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIeh7j00
Boq0MA0GCSqGSIsb3DQEBAQUABIIBAH7NZ5T5anffqtWAgt0oMtA/krAJvMnVsgHb
3dWk15izranm5qH2EdFCxvdagu4bsboapU7GH2o8sZ+Hr7ExuiAFRSOQMS/wgOgW
Vt fwjKSoKYqQb0/jxCKMtDGqfz1p5qBgNAz7GLEkC/P+PqYNHJrWx2ddr1HJ100G
6ut7Qjgsv03UIxSO9IZ+KwsnxuPko5AuveAifbOyN5zNA/yNGWrdVsLFboz5sD1Q
uyI/cWctTDCLvoyVtBRkIWRUJlHmgB8AlFoT2pBRmFCExx1NK0IG2x1Dc/K8K2g3
LTFEoderXpcOYlS9WuXuEGWpYFu//Pqt0kmAacfbp8DbF/KL0k0wggGEAgEAMGww
VTENMAsGAlUEChMESUVURjERMA8GAlUECxMI TEFNUFMgV0cxMTAvBgNVBAMTKFNh
bXBsZSBMQUU1QUYBSU0EgQ2VydG1maWNhdG1vbiBBdXRob3JpdHkCEzB8R0APhiY6
HGLS64Mv1sDXhpQwDQYJKoZIhvcNAQEBBQAEggEAPLsdF0Kyueyd/ofoyTKriNDH
mh/Nr7KhbiqQDRZpJ40SL2QR5Tkt95RZ2FchOmP8QVRoCmPDIY7tXXVxdaCewju
qBEW8TrDCSLsBa0NZ0hFvMUed1VgMLZuyj9RFumYCFg6MXjvS2yLskPCvdZJ6urd
n7P1Q+Izs8yKSzZkYuxY3Zu94pA7uedClTP8hS3LB6JeZWSQIVA4ZLZ2/9JD+0Tn
0EX6Zx8fySJCZwcIoWewcn6KSmSekQ7XRvkoXj7FWvJ4UB1Qeo/trWa25Y/oj4Q
BoBvnOSiMm+64zARzVjmqIHTRmZ/HCzdeEcM6Ci/+OxRs7a05pPEKCRtRtPQ5zCC
Eo4GCSqGSIsb3DQEHATAAdBg1ghkgBZQMEAEIEEIdKwwRA9368qAMYmmSuOLqAghJg
05DvwW0FJ4IGaliquIe+Cxt+Bh0UMV7FaAia3k+cV581Iq3yTmhX8bZpRLBqM2Hz
yb65FD0CFqzmi1BH2rirDi/ewj0y0rXunHq4WvNx11a0a5meWec2kdG3vUir8BzX
b9qVNGn2NNkOUWkPtdOhalGjRVAfF+hgzdU3GTmBRsEtzaOqRKg0Bfxa8Fa8Q8n
1IjYA6HV4bgZWTg6Pd+nsjZHLv1LcoY5fHh6Z7ZFWJ/oxXRTXBCYurOqFz+YPtod
p/0h7yiBEbOTFPCAavzQ+9d1K/SK43somDj66P1BwNm8gi5K6M1MxpXqXvJkGMYu2
X1sfp2NH3pzHe6raO//jdBoSnHN/qPeeyJeGpPe311/FJmBEWX+ZW1Ob2Q9/hyvS
sSkfEHkypV539+WK43C1MA8FCLC1z1Zxv/oSBJS3CWz9OtpsXk1yXKJo8QZV96Gd
pn0pzdDuEzx/xLrBDDkWMs3UW13xf/1gHznnU6Sv14VF/Q8Rmbx5wsveQunECnaT
J7Ay+p3RuywANEFfBjz1MwW0zk1/zH2f5vdGyIjjUhJoHDDTs2xNe2KpCpc2ZvIw
rgLXVb+lep+Qc53Un99tKCAAb0H3ApCa81XpWVBR1zfpike6Jc5T8EYpeEjLyDr
w3jQcR4jAg/5dwiSXX88GzfwJQQg28CCTWX9moVevQAH/y8ZbALaiChzfoGEXvNb
I3r/e6ebWYf1JJKUEPGQeUU3IBUT4ZQY+S/ZPvPkhDUBho/2Gk5zIZiAS+YRRyXO
IUOYkjpObtnd+sKnqQYE1wCItzG9hOVcuJdU5uJjkXTSquf1DwIt5GYR+4EqW8nN
vnrbeRvCUgiy6G0kPFfEvFbFVvLD14ldVAJyJPosP1G3QGTEhBtAilRzEQU8jvtk4
IHm2aqYKntIFcC/wq9KGXjiKBfwhg9mFvyESYFa j8pJcIbgPzyez/+WSRTV6LdMd
sbwiCXbeJVeZAF1b5yd0aBjHCDE2q7KR4ccTksf0n4Z6Kt2WXir5yd2StKcJ4sLb
5P2MchRjPSPDM6091/5sUOItLje1NgeVYUzvn494kV3s7rCNfuyyw0gRoM9HGw1j1
rvIdVGKZ0vJhaV/WjxznFKsZuOUG+zQz1ka3LGriTQH1R6cVrSi7Xm1CLrKDR70M
mN5S1Fq9Uw0TZ5K56IJQ6MIjUezIwONSFDwynw86LVLM33cvV21Fy7/4X1MkIRYU
vSgwDSmVxLTrdauNNThIamtC3LtTwZ91XjnjgmIXHdQ7JS3cX9cIsNTBGOYCT6V8
taxyzv33pjwL2GU+3C6GFtZGnPGOBYOnAWpiKxbECz3fuUmG1EwyBYeyLcta7ZE3
y70fhpvFggdVt2Q2fpMFemnc6d5hdi3KBrTb2YpyFRgpE91HjtGoB/iB+StrTclS
W2MeGn+p9EkJmPmq+JubFN3Vx0mLFvZLLFQFRv9emZmtWYJLkQz3c3PSge9QOrZP
hEv4WgV1U3jz2L10xMMAqlv065tqZIAuDVUOoVLE5busbp7/kk/boNimArS2pY1F
1IWIk7GR2c3i6QDrVf2FGLFJxmitYscrPyiyFS36dI+iWu5B+tbvEfp8zjJgWA93
BueKKNalKOG5JbOAbBrErm00l8/g7auxPthuRwo3hax+Y7ESVNTf5tniEmluPj1B

/A2wfORTi41sE4CQpMVqWuFaOUZ+syc0Ow6Xu/JINvYGxpU2X9mMbSVzv3ZJ4pZ0
AV1CvEVLp7bt5XZR2ko1Pa3PXU21jCh1iWshgt1XxtdQSZFPYxItjC0VIJ7X608S
ByN06PYQ/pioG6RaJ7DGK95xtz8sxbYKW5oDliU6IF31EuVPCXfWKG2tks1aLfKL
dXDIdw1P5ZmDmHlnyzsMzrjcdrlvj3hOE2rGb03o1/cfmAD7LWsetaXnSTOUS18Y
ksvaKwIffgFbu98nxLMbWxjLBAX/FTagi+3NJ881KbnI+2ayPwPFqEQZsI5W5N/w
IlrjckDTxfZ/nvICwDdKnB813pJWoMk0/SM0NYEMANMmFexG2NfjRwhUAXLpy2Ma
nTr2fRycolz2VyoUmI5909NNDHrOtEtsBR/LcPOENy9tR1N7Wbpoktj1Z1s9uYxJ
ng5QDXtIN077yCdhzbPpdx5eEQEx8jUC4eqLuFiINusUILLf+jzErtAS4Dr3P+HjH
1ZXU/klxwxngMgG9FdePn00807JoYVYjpaZVarZWwaKjkypcmehYxXq5fx9UIYxm
gbTImF7u8uK4SR0i28fEigIvInts2xEYfO9WFq1A2TgpTh7q/I3JyuW52KYCtFok
40xiQuxiC+/58aZycbjLfp6e+pYsB1BQiBamlfJceZuCTW2vn5sjuVCdSqi5k02q
hgZUSnlduCb1T7QqZ9KjDZ1EIN2fgmA2RVxeaFZ5EXVxVjA6C1dL70yW/G1A4Pjm
hojv+slDVfXxHoaFC0LutvBFxMf9I6efheihKbGM3mCXWplzMKcqWgl9KIJT9raf
N9SrtHuhC2JwEqRvvn/XQN29Ncr2GbhSctmayGkmqD/c4vgN4noukUo1vuNVF4Wh
1GPWju657zAHJT6qRe1p7BqE81Cpf8aNeWwK1xBu/H1AryRMKKwRXm8x2baOs/L2
CokiV3GM7ip9Uf8hF5aML8fL0yNMMpHyk3h+rhsntjK0A/0sF4gysk8WYhBnD8Qw
lhJdkVoc81NkGNxIrlgVFjQ79fScPfe3oIveHHrs7BpEABdcZDf4NSrCZVStw0AX
YdQ9RjYbTiDhPrs7s07D9DV3VEVin1Ng3rMtOPqxb7HKv+Sa72+11QJeu+1zVQax
Uy9EapOeTELAwOqsSSMxgn65VMaLgd3E7ThUr0Kp8RjWp+mEcJ0c6AzdYLnfpYD6
ic12ENgtL/Q8FG/0tahkp0Th7TkVcjpJzuVNpijvkIxhuA7d1xIkKaLHxjKxCiUw
oVtfdW8Yt+R89SnkPcx81+Ar1pkCaC8V4K9U6C2FIz9W40dHFUFajTnycgUWMybf
A7D1UNAeJBNjRFEliSqPu1Yr2mooi4+hV2LIMjLxs/aHWKTMah3K3mTVyo1tAwVt
+2kMIaqtWKQi5xr3Aw1P8GKEo9F1osza4B1kWK3eDovCy2HGL7R3HJGgWnDxt02I
KM/HTywrU13qMwxdkEjYgV/4RWQeOI5FBBAemnwJNdquKrOOQiuHFxGx140Y1jja
l/sRUtS3pecm5x+CHCqYHSTlmAk+1kWL4ELwdAd4atsyrKn7SiVuZCgZ3/pi0kEd
ZBkxh7WmzAn49FMg21S1S68skCN14LH+315uxs2PiTtxtm+h8D+Fsc9g+Wnjp593
CyPHQxQo8xSqCrVupdxeuimn1I+Onn1JUph906VRS/Ld7A27xW0a9hkGx5V0ACQ
J14i+gpcsw5jP3JV21plpgXqktR0gMbgUOU7Qvst3ZRQueiLJb9Ujdvhx4KcJS1
q6jreldOXTHkz4N+RZyMn7JJAlwBB/gag5biD1HjvFYKWnrpLL+fBj5KPrfaDK8I
AvKMhm3PdbSAw6qieAntacTzE/ivFsORPUv1Zr9JFJ3C+E6ScztrMvBCCqK94Zst
WVjCwVvKmd1ARSmPElQ/SO9OzfHBTKMaFNXA916yUfQ1b1E8TNDHIDO+CS+6U2Pt
oiPay22qExWsnkuU0mCUDkrzKUR01MQ1YPTf+zD1qHPZOBCHHFscNxcE3YKpK4s4
y4HdE8oCVwo3II/rpOHAqIb3qEM91AH41jtX0Z6FfIhoi0nltPJCIEml0ElmVjpZ
fiOYsXjTw4QgDiQF2w88sIV20ov/bvCydBTwd3Q0YgDLLmGfo99XJREaPhXeKKNf
noNSNV/xR30PwOnWoWpTSPZnYioxFOY1knpUIRVEbqW48B9KMUoXrawIZPGSWO+U
Ib3H1Dxw1cWEpkC4GB/G7UYeZS0Z4XKcQStEdn5QSSkX0v7DwoqI7etmUhuspNGn
Po/HL1PR4q9JF6jPtYqscKm0EjF4H4C6QR3FrDz8FQeIT0Mz+9/6rAgYjtCbaQN1
I2zn8qkKQfmbKC9jYTRgg/T+IGbSvZPuWVrkOmMrv6K8uQCYSuDpfPS9KmIT/0Ln
iGtUtycME+riNw4Tc4SjOP2VVoFEX4rfiGaybVy05BUcZVahbmL2CebxLyOT8uE/
D3/w196tyWywNADDgYXdh6jSdws9FJvTNT6I60Z6fAiDspAlPO/wr/S/yTiFHDJw
h3jzSj2GQtWGiDFmLuLXztFG6BTDDVdyqBhAg9AghLuPLHZctNvyFmIVNUxDjvzG
1ViFJVfkuoj3YLMeLwrD6vtATct5GUQfKK4sagGwZ80egMMSxb0yViB1SglEsRrd
nQP5vA+1INUQR2n/L4mG5ZdJL1Eh/dRpBbRn8szKMXTGIuLx0LIYV15rnFCbBMN
H1U4fbHFihdX41FToiurCxvya6dnBoLwm/2qQY64dzbj5kTQpxz/UmBN/8AwdvOf
NAkb97d3/CsEli/soZowZMghezjWUKs/hhL7/KBIcXiTG+2aXKs3etryNJRiyCOW
ehkEpOvhHA6IX4y9VmorT2v9vee7hlGaOWekbl62EpukuD+dCCay+FRlP1jU6wqD
Q0Cqv/7kybANL4jcZI4Rf3joE/yB/mr8Ygd+5ATFHNmOVhdm+RKR0Qchuy+1hrre
1mjLtoeQs4d8bUT6T/WcX+xGG1Z7krfiYwJQ90qHclVqAUsYFi3eQOtSdHliyoLm

AW8Mr/aZSkSWgygqL7dd0KGC/aO07GcryqAQeQtSFBIXvb3xR1S0HgowngCTdZs
 IqWrM8BkESGpywMrSi3bsfkuKnTX01Fuso0q7Kn3VQE0kTCfSRUunOT81NYLA+MV
 jsWgB7uYX8AXFhWM+MANGIuOfk+IeLwtCfWfk01YCLn47NUahQsMPo5/4N0CeiWa
 SFmwu8CY5UCLPCW5tD+zP/mRtLM9Xd9joS8LXF2gRUAKEzOCJpy+qy9YkCuMgPd
 PNx1cq3rcLz1qMopCmrDO5xR/LkUuY3I017kf29Hb4HZ/nXil/p9tKlOJ+qOiQI9
 zFRxqQoxLQsN5QxA7D/w/5mBSDuRda6am2yifmdvwjsARsZiSSY9CY8Q5yEc5C+H
 BhK6qMC0u82Y158VjqrJRqvQaluJMN8+CS8+4KiK3giZU6PE4mqoBMmNy9Mg4zQ7
 zOjg0m/DYvPz5/AMk8Z/jRF8PQEffb0JcfE40ksKQyja81NLTJsqs1vYQdITz1f
 ghmVxuDFcXURzz7vQLGcezLOe6cKbPtt6S7OoAvvtJjpJOrdwphSmJN94BG/9DYn
 fQoQz9hUbboUgFRVeUWfStMER++fciSexJVyAj+kgObAaJrhstvjM871PFLlFY19
 EZFMrV3ymygWYc/pLKWW7VFXKxmHjMAG2tm69LCpPWxsw/rmUaVBVe2jycb2FLHi
 8sw3ecNwOfsCd9fucBGtmqPEiWr9nrIVj6I4mPd7tCXZQEhaN7sLz9hX61Td9Ybg
 3W01YSWzaBZyxJDuxXbz4Zd2t4T43PRJov6W1FAcatQO21xzOIqlu0oY1s0eMXHO
 FF554eor4J7SceENG0c0v1IarFDPYzPmNoMMthvb9+7N4qmgJTBjH/SwBa1beDBI
 7yN/SZwHb8juuX00lfmuBDofTWWS4nkPi3Z+vUMUVVlqP2Th4mlmP6f4H2aknSj
 OrMFPM3C97UY6Azyvb7RYb/VrUcnM7kiYjYm1irfRSYjD/vVYwxfgj1ruSFYw7Sb
 +iaVQ0+g9XDTVytovy4xr7302goBJcUK35kD1z/2E2CLeFBxEQu/PmYjOoSvvp3f
 YQvWsQsCqBFZIGAlYbQjPeHJIISVsJg8pa/BkKCCu1VgvnuyQoCABawv81tMB6sh
 L66GdRK9zc8G4dcr1tjaxAp6/LW+taetP04yRNhBlXAjd10/6ldyaEkyLRk23dWN
 VMr38oup6w4rhFwWt8Py+b48djfqRzqlcdqrx4B+qLsecEaojx3SgBriytofYhT
 a1zNXHz1tqSPV5202s2DPGkjQy9ZCIjX85WRW6KZ1e6aT9TXE3jzDjDtsAnp/jf7
 OS0DZMAx0hh7ELKqrG0xP92IYh1sf+OhpubGIjuBAPo8L0JaQ0SmSWKUwfF8XrzX
 HCzu+MtnQ+6Lf7ctJ15XQJNEsPEWshPFpXGL2IRfdl/EgvIk75OC4JQ1kW3D1/s
 R93ikylznWBF7PDqWREq9Buo53ENUx/lBdsXxJ/AxF5hz8tFe5QnK5fZ+iYHbhPV

B.3.2. S/MIME Encrypted and Signed Over a Simple Message, Injected Headers With hcp_minimal

This is a encrypted and signed S/MIME message using PKCS#7 envelopedData around signedData. The payload is a text/plain message. It uses the Injected Headers header protection scheme with the hcp_minimal Header Confidentiality Policy.

It has the following structure:

```
application/pkcs7-mime [smime.p7m] 7435 bytes
  (decrypts to)
  application/pkcs7-mime [smime.p7m] 4498 bytes
    (unwraps to)
    text/plain 333 bytes
```

Its contents are:

Content-Transfer-Encoding: base64
 Content-Type: application/pkcs7-mime; name="smime.p7m";
 smime-type="enveloped-data"
 Subject: [...]
 Message-ID: <smime-enc-signed-injected-minimal@lhp.example>
 From: Alice <alice@smime.example>
 To: Bob <bob@smime.example>
 Date: Sat, 20 Feb 2021 10:09:02 -0500
 User-Agent: Sample MUA Version 1.0

MIIVbAYJKoZIhvcNAQcDoIIIVXTCCFVvCAQAxggMQMIIBhAIBADBsMFUxDTALBgNV
 BAoTBELFVEYxETAPBgNVBAsTCExBTVBTIFdHMTEwLwYDVQQDEyhTYW1wbGUUjTEFN
 UFMgU1NBIENlcnRpZmljYXRpb24gQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIeh7j00
 Boq0MA0GCSqGSIsb3DQEBAQUABIIBAE4jHFjgjcL+vJbAAHC/TgYkD0lhFkLlWZh
 gSxqqLgjf4wiewJudnfk5t9F091LxUqqrqFC0oR7MTdQMjhgmscb9G8ncJoWsNsO
 EZ5Fdt/rrxHgtjXJodVbrk0BOJ7L9GVfzQBPFdwKEg49vP6+sVp+CGmByXvd1A54
 ueZCKs6SK2QMzodp1nJws4IXm7BIaJsvGu6huNEI51Ne+NS19qAGej+oJn0i5vsa
 S/2H/0fxS81sIBfY/QYRr8AAb4lbFltWRWFQgix+kORhltIPP4A7Jo5a+fA92ZCT
 HpFER/cZBLpalp2M+HVBa jOUgASwsA/Y30Y7Sj3kXqE37RvaO1IwggGEAgEAMGww
 VTENMA8GA1UEChMESUVURjERMA8GA1UECXMITEFNuFMgV0cxMTAvBgNVBAMTKFNh
 bXBsZSBMQUU1QUYBSU0EgQ2VydG1maWNhdGlvbiBBdXRob3JpdHkCEzB8R0APhiY6
 HGLS64Mv1sDXhpQwDQYJKoZIhvcNAQEBBQAEggEAEBy7Zg8b9DsTrdlACEAgIB5r
 w6FQ6Bugd6UDLrGOMyCSZ1KoCmPUxpb3veBdbYTrjSIuhkMYq0/ZUQ7JVS4jgFMe
 4dHUshBT3CKj63FQj/FT4G7xFKuRnyfk7fpeaGBR/1UsvQ+OyViHQgf4JA6OGEk0
 R7oyMOROCznSFT/Em585/5Iq2dxsq2X+fQUPeHW9sSRRnDZQMmIhQGwo0tDI1vv
 00LAGv2FP0p9iYQSzJ7VgJAViKHYoXDZTrGJnL9uygiIJeA0gvw6f2jWlK4j04c1
 1DNnQ4KYhWgIaPp5njGCKEiqssMGIj+TkkiYludeGy6dEK6f+Noqc7Lotfz7YDCC
 Ej4GCSqGSIsb3DQEHATAAdBg1ghkgBZQMEAQIEEPLJkiAiTok6hJMM2eSXOzyAghIQ
 VzdGI800ZwU7vWIZenIr6HSnsw6yJDWd6K6bteA6qxZ4LMCFXNpNxH5VF1owK5
 PqneUhxG3FUR88453uLLUD11Y1ynMwvcbH0GGPOIn+tcP0VQHkFpmJk7qbmclf5Y
 jOsWMVvdYDGqgIDMgBAPp2YdqNv6o3h+RYItALY6rebm/OfbQq1nSRduwh8oBlnX
 BOhV/LwC4CsqvRo8SisgWxGOMMhrJeV2127uuqEmOIA6fNpQ7yGiKJHxz+eaVfDmy
 bhz9jPZCVH5gL+7cBE2LVTjDrF8H+JDpTC+uQ5YJzGCaxubDbHay2R66Y+qfSy1o
 EDXvli1/aX2yqXviRyxhkPteHBct5Mtwqnfqai0krk76mx1JBeBQ7KrwPi8US0Hv
 LXnQxj8tVVts4btT9bNRh8WPAdnhc/elcokASMaEZIB/Oix6hvhR2/AxIIXEOn+5
 HOHzJ96UhfBstBf71mIwMzwW/127zyIzNGK9r90kUhK5psMMkR5U16evSDPMO3rT
 gKJJwfLH9nKvm12kp+Knn8QDoiHqAmjytzrBwgZrpk1qgFFTG8Zz633BpPLwqb3s
 j3tSaGrNv0dfFG1HgGsgahfXtvvFpNFj4zR9zx7UNQASXTRXZ51NVt69CnKkvuYp
 45toocAZkYQhTGENU9s+GD82vFfxKYN6PL6oRyef3fvAZ9F9tY0w5x1yf8TZxoMIY
 GGM4Unaqsty6YmFqqMO4do+bF2G1bFXyI/2MXa34jz0tnExGogZ6bsfi/5KYZIia
 +w26I000yv58j0Jy+CQ6Mfx57+9W0whx2tOcYeyv1SM2ER6edH0j2bMgztGO9+UJ
 APUN6Hq/NUJ1uiBNq7e7nnDHFS9gyiHabq7GI2yilnEebZe32jw9OSyu0v/SyAsx
 47m7OKZAukwI3h/9W4iS8L9cEShGUJtSKf5Bnp/m2iix9B61SdqT6nwVWEJ+671v
 6wonwAn2CDGDosvXNoMTktDt4dBnbl1gLC/CgtupTXSospovX8vgpb4VdzK2arCL
 ec8EiaJmGVRW0xyI/w+EkcyIzBAoUDIt+fAIHLz5OXKPPFs2rGHRmneOsWtToCcJ
 L3oqppz2QXV8/teUQ5vxf+11nF95vIBDeiZrEY2eAIPZwhdaCvc/Ekzdxee2Tx+cq
 JIoVTA/anwMuxmgIRPKdIEMevgiUe/te4pIm+aXhy3VTN1Dk+AnGAHvJnh705Zx2
 zmmhRUj2OL1sOLxHkC/bMz6E0vjMiE1WsIhxds3EW9boon06wCjz6GUKnSvOj8S1

ac5kAAomzErAUisWkbsQ+lNCysqNGEowSWqOG4703CzjcCMDofwCv/K7JvpHxvv
zosGC0LXLQHI TM9qT2PMN4D5HPavNCGAxKQz5mJsovndj6BMJ7HqvhtPixWrL NK0
N4yQMc6NUUDn1J7h+PNquTtzRMqSURk/L/baNf5txyv5m6TgIHBfslnMrfRBEvuI
3sgpW+9aers/0vMh1LOLAW009kCf5+nkqQ/I8ZFaLIFvdRM+AkvbVaQN8li+Ew2z
lef/Aeyo4X1ofNkmFTqxyP+F+ZrB3ZF3/Z2m0d27379QyCXviiNrBvOE1BXzadwd
TqcyILwqQaQflgEx2d4R/sdYoZLu95R9iLezeZmzYi2KLXmm/WGTzB2gzW0WINqE
k0+b7Jqg4qVJJBEOUrRFBZvVwVDQ+cXfWzt3ij6jo8h0iHG+LXH1Q/sIKSmCZKK
XV3U5Zz4iioCCWEenuA69XN60VJON15QRBIiWtr5vjNUJ8AAg01qCygz5VkQzxi
fh4YIBk00Y0nzVibKKvei4mNDYNdv2rWwUSFSUp3MfqPfl1Wt35sSapBXPgUNLujs
7J12ZGPeiV6iB7xibbLsIQQTjroktQrP7qgGvKpSu2Q6yQOsJd5zqrQmyVzzhKEo
V1lwAMYDE004vxNHSHpz6m4B0+ey7lth8MpeXHk5cyQYAh+dn0u5uR96FWRjM6Fp
G3gPC/0mS2PytJG7KfQOkOKElwlzt/ypg/iAKsuaMBx70HLuVR+BiQYfTd3YO/72
y6c4u7BarWgn1FVLjnNQ4aodZyodqh/DluEdkF5AkJb0jNjP8DQAp+78E+Zs04OK
C65HWQdfag2gNtTvm9ORMtQjK7K4vXneBvWLaHP74vouNNaZSS9mAAQQ/1YEDIdk
rJxa5hnjgB4+m63U0IqZh06Yzuv4AlkVtp+BdYcCjur10hvWyq4k2FwFslarOh0d
id0lMirNC/rSnXcVagVonmS28Ykg61SE95r7CHtbUIKIGcsOe+AcSGX+mpJwLYqr
1qNV5PZZ/mFX69QwcDVRrzmDBLi0MW4iGQOup0f/S6RXTjW1nTvoJomcm9J7/Bgn
nRhkYcd8C/4g//H3XndKdxyojr7KV3UY7iL/KPHI6pIVI7h/HgPJTAuecdXIXWt9
Yr/Srk7R48cpqLxdFvaaDWe3Q30LtNeiL5czscnLubAT6LBstJPTeQE6vnag6N0J
BU0Z0kiCLLIE6We1CUzwQjBzUAWVwH12uTuFJZdPyVt94VpWeBEP3daeCwnJaOgF
krqkYLC3qySMLK240y6X8wESNuJjTEPn30t6/D5CzLIF0SugIwd7GeswWfJvbjl6
4Z7JiTcVpZ+M65LFmLn+2oPB4xh/hyzNe0qs+9Z1zd94M02TxZdk6LRaNwI2yne1
2Wv0Eg+JEjqilnIP1jd5KhJLou9BwBKciZTGu6OgCeIWY8pKsflFvMdxkUs41xvN
o3FRhQ1UZPslVzMabkP/NRb8D0pEedyPiY7v1PlefnU4jX6jP++Ejwbr8vT8K5NK
zB3tC+1MfZa8Ytb4zuEiZ4ept++/At6oUaZ29D0zhPzckILtsHxoqdbudSpC/RQ/
djKYTYu3XM1EYCUf9FRDaowYjPTHjrNgFzqF/Gv7tAr/1EOT/5SeMnrKaDCnqh27
BzE92JTTjgkIjyQKo39JT0DNbcxViUX41EIH17E7tzY7Kaaphousqdjo/mBm4SCu
ncHK+mEBQ+2IGm8EaRlZTHqUqPXwwY5hsv4QMFezLQCFAlsghlvA1/IpPpESV+n
EvIgzCr+RLFWnX4m9mEOKHjK+yTds+Gspc1BWBby3pQUqWFQa36zSfA6Lkm0vuFv
0C8YKHKDZdtIrhPTD7e1Gooz4yGZc9//xiU018HruLHiCnsbQjSHaln+EFk9qzxj
hRSI/4iyfn6mDqwfFqIt39GGA4Jk1eeb871bwTBhATbBkGwGhKVkeRT8xp+dRlCj
S4IsUDbU30rs50SbJ/fRYpVB68nQQNCC8pE2Hg9TlopAnRY9kKiJ1pnMNWRMoRV7
axH3BppdTvAcqa00XFAtTUJR11SrJ2XzYQ4GzoaA6Y4VjEu21V1apjg8Zd2ehtVf
Xfjyc9vQsrV5AUuCRlQRdt26s5VveM0c9wODONxLgL5pimKUmPC1p/0oD6vWdSEn
uGgx1XF/Y0qk92o0AIFjey7xiQELwIP0bl7ukxi6TBayeZMttq4y/OrVgMZMoM/p
PWYnTHfoq+c6iuHc9HBcBlkUpK9crvliKaNo9UgHvfIg87FkGkLRvol/c49VnRLb
Vm2IImWCOS4TyQxWrdo+iBENltYA09vpCHw4wrz9qzCGEblfvHhFHSMn0V0TJA6r
Rv3W7KrYhIYrLRouWtm6pR0yvXtsGK2b7w1Cn9afowBsQOyx1AFfSwMpp1XIA4rJ
6gbR0FIKgcA6XVGQQroYtdUihp+Ie9EmQuoesyZg30f2T/ehNil9aZqmeh9rNuSM
PkGIfa/qMaXYiX1pECSNgRaPeUkt655B424KedP4A1p5eDkKKAwHoAsPM5nZ3LIp
WwK6pBzy4wy9ivoTR8WQUtyqf36yEOJLdVf8r5h+UjR2RGg2e0S/sbSyU95KWshp
2agwKQnzGBO08K8IP1ELlNP45stzpxYfCxxqezUwwRzyWqC+hK5RPNjP4CXjAd8j
z0ex0sEoe+5laknet+MPWkQ1wGRqzkrqgbiWb15SFpbM1Qtfv56YUTE25h1gmu8ik
cRBVoPVIi5As0Jpgc8cw/q/1mmC7ha73V22W5s97y2B2aSn457eXZjJ6tR0p6WPF
q5PDDjjlvDliZP4NgM/uyllfbyi0gvW+Tziha7YQIWATAg3EF+0QTzBuHJADH+M3
4RfT92fv7Euya0+/nNxCh47H1ex6v8fxvN46aAuYLv+GVVKC5Sa/qqX3IwBqXBwa
Vb/57a8+dqonQpvr6q8FjdymapGR4kCDVzXNdCgAuoqMRcu06wJI+ZjgmvNHTwIx
03ASdCVgk8FZaR8hA0MKSDexsliIvzEzWnckwVdGsuIszxlLmnhTiAxJZygh5GJ9

SYEV5exBe9E4tpAV2fKtzLk3b439Zk25JVCE4ZDY7M/4kPBQ9caFQzx5AiE5PuSB
URZbMFLK4wldwmfM3B3lRsRlgHxr3D7X7fp7/92+fkcM7F6kGwoR3YZ+cXbVrdYP
IJBepUoDIzoLXwC0/5KjivVlt/VVGrL5SKcQ+QEob9DlhP6l4jevV6KYq0QXEw7R
r79EnzkKGqgb4lHjP902ylagv8+RqLQnna4cpiySi4SX3de0objntyet06Rq2EDY
062yLHGAYRrOs+qxV3DPAWKnMbXa+Ae0C8D+MzJcK9ZJZnNTRzeJ+bVByPVC5w0
0E4ouXA3i1tCgrjQqr3yg69l/aj9sPoT5ybe90+pdYccH0VO3beXOS+xZUUpcyqq
VlizINAOf4y+P7FgPh7+gvrfrKYIh+SJMck1Dxs04zA4M/aE7QhxjivEpi7ngr9
+0v/VV6X+pCFPmFxia9TpEiiUG81LsdGCHSzedABgWFG0M7rsPuX/5gNN0s2rdti
7tZu39pRWZ4+HXwXgKnMPk3Kx6i5PMLEW0P1M7NV+pLLRiwS5C/8w0RWnzBlth6g
nqX4mN3euezQmTrZAoFD0SEymljLhOoMLIMEuDBp9k/4pQTE74VMW7ZwjYxz9cDS
sAWa3+sk4c28sAmTdV8hNltSey+NqA5hRj/bvVEiKNLvuilkbwlseIzqq40Hnrqq
6OgAaZ0bNxZ5PYbY5T2hVA2+qtja9FGJLcVFr0Dq7w76VeAui9hqmpQVmw9YrHmz
TqYYYvCZRtn7leHmIT07j7MGRnyfqcZNM/olJya8vss8tiusS4DkGNiqq3J2Vk
KjueAqBo+3uYmzqm5gKSbnfXrkDTZJcxj41ZYztoldCCHUADSQ0vQ8QoZ7ICW7yN
4sMnoqQGL63m7oaqc4983iHk9sK1ZoB9rrkBg1QVNN+ZWE1SgE2ASfen+tnvFKeJ
72WWtgQtK7NhYVPfWF0pzOlMoBEwJaLoMVokYW3I1Cp2joriszqu2ALAmgGTUbc/
dafVABuvHuOErPhHmlp0yVciff6496mspG2pRxEb3hhHkOmqlJwrVkk37qMUuMTJ
Npr2r0galtYT+Hzmsw4ZMG42O9fUEyAvsNfF2VeanmBJRdxHs1BwMHDEyxrkYvce
R+FMtAivKNqyDTQZOWkdy3knwDgfz2TJ3M5guMPO9zdQLN1ckEDA7nn83lCtjJmw
lujtT5NORYIpkt0Xb9ZZKAsnxvn5L1Sfz2dC9VFeoIn/amkVAVaZXZ9vWY8V5Ae4
UD6f19EhvZ2SbDck4uRWCf/i5LcjkOyGwLotTY2HCfqjmfpaHDfNJKwikIx1Yly
I5421BKwMlaQuVPYzBUgN3Abd5CaRn1etDax+i1N2jyg+dj+x5NQDBsWJ9IJUOXT
nMDScnH1YW3CeuL+WbcBozVltZaO2RKSDCpm1z4TGTAGHYMoek8PGW8/ZBTIMqCh
7Y1gq54IRMIhO5JS+MTbp4MwAr570XxKrc/09PyDD1EzhIpixAOHQdf4LI97i8Pt
M33AKEIwZjg7lmmCnURdu5YNA9Q1hBgjshd7tHAZI57I8UwdX/GrH/jGm3Zd0L38
xPfZpa9Qsr2Fs/f54Zje/G+9vK543k5PY26PckeSxVFrAc1eLNRRXuPODHVc5xxX
pwj+ARVUo23qb2bn2j3Rk8u41Z+mtOq4YmLc5Q6a0M034HTqrc4jiHU0Hy2nekJV
pBbOU/BFByUFHn+M1h6yRtgQjVKmC880/aBb5u7MqrOsQ6cvNqHfs3A12HgjBxga
+vBLwEHtHYgBoeZRdIeQwA==

B.3.3. S/MIME Encrypted and Signed Over a Simple Message, Injected Headers With hcp_minimal (+ Legacy Display)

This is a encrypted and signed S/MIME message using PKCS#7 envelopedData around signedData. The payload is a text/plain message. It uses the Injected Headers header protection scheme with the hcp_minimal Header Confidentiality Policy with a "Legacy Display" part.

It has the following structure:

```
application/pkcs7-mime [smime.p7m] 7670 bytes
  (decrypts to)
  application/pkcs7-mime [smime.p7m] 4674 bytes
    (unwraps to)
    text/plain 423 bytes
```

Its contents are:

Content-Transfer-Encoding: base64
 Content-Type: application/pkcs7-mime; name="smime.p7m";
 smime-type="enveloped-data"
 Subject: [...]
 Message-ID:
 <smime-enc-signed-injected-minimal-legacy@lhp.example>
 From: Alice <alice@smime.example>
 To: Bob <bob@smime.example>
 Date: Sat, 20 Feb 2021 10:10:02 -0500
 User-Agent: Sample MUA Version 1.0

MI IWHAYJKoZiIhvcNAQcDoI IWDTC CFgkCAQAxggMQMI I BhaI BADB sMFUxDTALBgNV
 BAoTBElFVEYxETAPBgNVBAsTCExBTvBTIFdHMTEwLwYDVQQDEyhTYW1wbGUgTEFN
 UFMgU1NBIEEN1cnRpZmljYXRpb24gQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIeh7j00
 Boq0MA0GCSqGS Ib3DQEBAQUABI I BAA7foZVL0cKGxTAGMEqr24xmXk+R9+1tBvx0
 vVC0FR62j6F3bEqRPggJoL/HYhvhbCluNzS46201GUESTn6dU0sFnAtHvpm/aggs
 ywFJsWc/fzzIyEN9wQ5X+2BWM9SoFTEikdGaUUz/fub8KpV3ZHmpO+boNOMRWys5
 gOR9GFt+iv5LEdqhvaymsdFs/qKAZBZo28ffe4DsanZEVmYufMriwoyRtyqndHD4A
 hmihNTH5ZCdeUUSZxb0w/UP9TW1Q9C3m663fywaSlzUNao14gEpTcto76D/FohGk
 s9mZ4vFcBgWzH7GJWJFWE4VRCQoNiWC4H8y+wIqfIDE9d4isEMwggGEAgEAMGww
 VTENMA sGA1UEChMESUVURjERMA8GA1UEC xMI TEFNUFMgV0cxMTAvBgNVBAMTKFNh
 bXBsZSBMQUU1QUYBSU0EgQ2VydG1maWNhdGlvb iBBdXRob3JpdHkCEzB8R0APhiY6
 HGLS64Mv1sDXhpQwDQYJKoZiIhvcNAQEBBQAEggEASY3CY6TZFO/11DvncjzRwpZ
 S+1JJ7S/t7cPtxZxd8ZVVAmNmVEvYkcxScNbvUrTy2BlVFWYKuPOovfXQVHhK4PP
 Yq23OYseIXVnsP7q1DMS/ZS+ptGBIXV2ZzqBt7I9jgMLC7f5i2NqWdNs0720Slz1
 MOIztq+Ccy8131WLF5k4OPLI6oy6PLv5RqM7v5CGr4RmGBZBiv2rQPylfSSGvAQ+
 Xn16CHji/70f9tEXfXGREJRzx/1IKFjz+JdROE4gptu/wXNjw6bTVTPx6FmfOhnD
 8XUZA6oBjN14Hi31LHzY1rhKQG+9owD4tsTcOcdIh7B8ZsMy2G8Mg0mWWHTWgTCC
 Eu4GCSqGS Ib3DQEHAAdBg1ghkgBZQMEAI EEFmrLeGX3dF7SOczv6nMLxWAghLA
 C3wQFKe2rnY/Rc4LgupEDEmQ9p39XhdQIEYeYvcNiPuRC0ietAnYPfAegOQ1hsZh
 Nd87LpWctj736OkRwUXhQyoVEDr8YJRIBBgOYC56WYHutkdWAFsCXrHhJAeHdq4y
 5XAdOPX9McvqKmdcDCfanXMWNs28G/sVlFwA1o6Tg4COW7g1DXVJhYqyZnX2tvDH
 u5XM4EMVezY3F1dh7rs+NTNQ3ziFs/48dzIVPLmOZj/OX9o2pcnhzU2gyE2ciPtR
 t8p/hWw2bdllp5+ZH4Ma/Cmaz+48GrRn3TgQzSw1/QtI+x6h6RBGSVTRo/nTEvWQ
 t9SaoC1c+SxmEtHCPWtWLDnf979+I9ZGkqsrrjasoTKZAieq6KeTBB9Fya6eyyGj
 VdDEx7jmKtpJpGvb0pBv18xxWKD7hjX2f3qbgFKrwuToayXLXCw1hYnX3UQ4L7ch
 t7h5T5m3pIehG8+HyNFOGvt1QaFTLzibQ1fgU8hdDQqkVhPdkPyCbLI3nFZ8HH9D
 V5dxxd602t6oNeBJQUKMAzOxnfsygBhw89fobdskQnOPOBv17PCSLrzGMvvE1WUq
 wamSi94s7V6gFfUmbe7YYdZEL/VEEawzaw/eZ+wHbjVxQkpEZ388cFHMdHOnkhUG
 SFobdwBYQj8vV4hxRTuoM9V7ZaV58S1MuS4Z86MUKCf2V9Z/9+XgkvmZMu/G+G7A
 td574PqjRaFrOuLuEQHRRZgCGUP+5troXLRgcJjTYdZB6JKdfNg1ikBF6Bs12Fv2
 XxXU5o89L53X87Q6oyycSuZUUwhaVQbx4voWjnoR/Wcgo5be9+moXhXHkFYoaJds
 UAORrQGVC+NaaVwpQMQujvZUOYQJJaRiZc5kALd8TZ8c2W9s3j4L4pDDmQcrgBp7
 BVdBnNDh7rNwFqrsP5Tt91JkaA7A3JcMhTnxvWQCbe3e8jbgj5oFFR0vIZju7md/
 NwT5rZrSVOAHpgUGEa1M6u+zN5YU1Am9aisFYy9s0dlj6uzGTP++UfSI6mJPX5HZ
 7HHVExQmVCjVgQwpiFt8sOA8GBWUj9w5i3+BXhJroFJkgELcna6RHIRasndr5fdK
 ssqW/DRjs+O5W1NfHhnXW6fBpXGeG7tUuaOj26Va00VWkpudP9jiH8qw/tc/ff4v
 5aNwO81MZ5XjKdNRN1KS41SFTUsYDhxpniIS4PRpbpr3GLKlm9d+vWsq+MV0xz

u4UM375UCi4ngrtagq/pgrQKdMW7zd6S019eRSm0QbGV97o3Cipr4+6uC+Hv/MHQ
GTCxM/6+uTqESnsngQu5N85Zt/zs7HagRGS0ozZwfakUuxpqyGQ3C5W7JMsbQ2HM
KFwQuYh//C1mSX9AZU7Fp0i7sKPP6C211ErUeWUgyViDrwFRi6F2f0nDHNr5bmXy
QCcsJdvRIZxCQpWtz7/iFPLEP68dNsGZsz1nXjgwXycst36IRdPks4A3Wfx1H2Ifn
BrLKqg1FUhJhE7dqo4KrvJ3zWIhMoyeQf6roKdxmCECrzigftrVRP5C7++3Jqj5U
VFDeof6JedRUP1rXv0TYjzwt+a+PUhyGFbDIU6CskSQo8+Rj3U5uYGSUPsbv2bE6u
luOPZYpYnKgBylPsoHZZIRbzmeTit41DlehWANRFjCwsGjdmUTd8yca9zWr9l5sX
qhWA3ViJz8CHW3DQMSO2obmvDwGnOmnHoxvjWWcexoTuT6AFTBw04XIIh5UAgexI
e5FS/2RzsqbY9la2WhGerXdrB2EIWsO2xaQvExyuo6JJEyk+8IsBqmgRr7mS11Id
H35SzbjwXkPK36si16vgsbDs/p0NIvrWE9bLCj9YZTagqyyUSkXNZssfQQdHGssE
kX/pWS+811dXcbQxamf1XENYHuovkX96nTq7a8jxP62FR0fbz3CfcNSAmu3bdGGR
CsQfW996D18+xtbHuks801cJW4Lnnavjq+Scb7mZroFuDSeS96poK+g84uXPdMj
1TAPgXxHDuvf880nUeuwdnM4j5nA1nHGSn1B0U8ZPQTRn+MVGKHgMyc1+Rh1K1km
DjwrzflGBkMbtipqKLA1nsyHw9TnYfBqQ5Mh1Y+jnH4MaT8t3Fm3hzmki2m4u+3W
AXeS5uznfU8p6Sbm5UvXJITRQbBowGD3/6cz04ymkjGwwAEyUyjUNOt1bjLa+8Lu
g/zvJ8EWud3a0az7hFFVY0ZQTR4CympFQUNtT71szCRL93lCa2RLD+LZst8wCoJ6
vdrHmCsuuXoNnoDE+Ox0CNGRZI9t6SleqzENwLpY//X3Gna/iLEdWzgo9V91DZQj
WVxuPB6YLr1WYocLG4ZB8LANalt3iGnLGSdzmWDY5ajrAEiaPDe/6ApPbHkuhB7/
fl6S11je2MiJlHJn8x3NLamw7qGJeYdq9lnsr+5UbhU+3+xtIUScT+7ncvWgf0aZ
Dib+Xv5ss/GIh3AwYdgx48mqd8/ERfgA9dbr1SiHk3KD/OR5t9cU8Vfo449vbODy
1E9s2tyRU95zkArMudoHKvoIB5qBazMPnTEE3AKNbr6HDZwP9EAkpSkdc1ZXq5pW
SvELQVvdVLtK7Ujwr0GfDDasCmK/g2EFAROVDPDhcPuAivHx9Q2BMCX0ZePjeKc
xOy/iTWnwCwtv1badizD8McGqQRkFnIezjKnsGDKJkuTxuigBitDNM9m7hKR2N7r
nbYcfPEJ+PorfaaeLIFThejzpbW38NqjPJay+APZ/r3fWNqb40Z/5pB4viBttLx4
ZHEqf/82CA/hNKOYDucEx9lJwB4CBniJDPE9j//Ncr20M0DJYziFgpb6g4+9KNsn
Zz2HlkYvy2DLlgxzyCxcZsmZIBahX2ID0zsgo8hZ524yyubAG82OCwKF6q1OCFv
ONVGNtH4/GGzQ6PEjeaJiibzVYJJPBeaqpitJMiVdWu8Ar+yS7a01p8RS5iXxBjV
L770yo2DGgwU3J6BquWeuiO5BK+4AsKVSMhsQgZ5q1krKZpMOUviGb03lCx+SsNd
pLev1ECSzqkhjC/XaiHeoHRAuGd8Vo9LcntNjcfJKRXBE/gQ7H9nB1C7qIf2FngI
y23th7XSRUA8R8xHi+AwWyHS8g+WeTx3w5yDh5ey411qOR5SpNvuYOGBgZhWx1sV
agmPUcoULPsxeIyQYKQq42fcb60hJrtw+gYB4x7RPDQkX2bEA9TgaXIOYPnQnxen
mkAlIE7VSHKhPdDpQ6NBueQDmMwby3UbgjttiHXtffUmgZPTFE7G98Nfpq/8Stg
RNPuncj0SUIbIrdMTUbyHOKLMq6kch9EXu9NqdY7lBLDMo8da0edY28n/sdgrzDI
03GESEjBV6KYjs9gOzPGhUMNXM5t+pst2LbzFpVOA+rONMzyO2lbED8Vc0skQtGz
H40liksszm1Cy2zFUXt2Y4kzm08FCD+vfeTD/2QestE9geJOL3P0YQdGQntB/Wff
2T2J/ERLNLgwZzB+WQcBmH9rIgoEJ+LaWzHF7cJRqkH7b4wui4Wsxpd1B1Tj3Xsv
jVifXsRSUrvCT7QBxcBHCEncPo1ETMv6/owEysVPYEnym7zc6L5e9krLDoJCYOWR
wENraaPluDzy7PA6NIiKknhAR/MxnpQE1XF5Bhi111+1hW0KNooHjiJgHQrxxA38
oSrQRciYbzVsBSjiUEqZ+ksD0IeCQq4MzkwV+3WhQ2Y38pKeTNIDsr1we05UsXXb
c8c0nFaWoSsAP15G5TSqiywqOMEZ/K4sqb4H+FBrcqXtAzxzRJMCKW0su2WsM6o+
YEqxZ5xBL/GmTLVCMR+DIOV9Bd9fnKdjk1qvTbOWK/RFleAyMvWO1W79B+Z1Ro36
0m5xGBns9m5Q6doBefeSjXmCBo3krhxznDD/RG85psnlxOugVJuAl8cWXnz8t8pZ
uuyNzc59S67IQj1lvJlS5Ta90LcroATUGB5AFRkjqzAkVDF+9LaWeIaIkxfocqF
UPCDVaxdupakvrw4+pLukG2C6e+GUODqv76Bnx8xfPrPSafG9whqi3wrzq3dWIah
kUFnkhAE4tZH5ek1fOJYBneStouSN8Yf6M6qe0TsgFWo9EI0iUWASB9HhS6bfTCu
Erg1bH0+JOKrf07HoKCSBx2cn1QJU06NET23bnUg4Zg2DDMdox/278ocQ8qmqu
4cpayWMHvTMgfZ1Inl++n13n8EVB1KJE0NpNFs1YnRHYRk1z2x6jB1iYXbFPJxje
pSx1qAL2w+hr/qi3NXnkKnz57h08weIgwFjf+cvF84sMThqf4Kr7r3iRdlXtY63C

```

mm1YKZ3iJVZEULsRnCGXsOla6x9DVqP5a/EurYPWqlzvxXp5sCvqIxdfIc0IGIjg
ncOXHSK4V0ezr0vRzL3rINxh8W0uvkcfqthJf1S9aeYS5S/8YEYTDdXF9BK/PcHt
tN6SX8EPYpHDtPatkS5vHQG4cfdGQG57Z644Do1SNs+bKsmjb2KFPMaEyoDCW5pN
ue86Wkzk7ArN3HK6tq/HSqrSU4tUBObViI4trOxbNsPDFmcbJ3RIfcKKIVGKEjGD
t0eh3ox4vdNkiW+5La75VAPGD7Ox40zqHT+6K2oNHfrAgRcecBBAbw9dCRuPPH8u
+m5kNdTo8cvF3BR6pVOx4rYn2T2uZaZPZ6JhMsRRwHbYDsoMEWBmrhGcHMnrVXKa
hnygPpI10z5REF1WSliNMpX/35RG7dODm6TeK+Wtp16qdSLOso3Kd0BgcjEUbM1B
DMefqY+0dE3Pts7J3UXPw8pn0H6ARrZn01euFeHVxMPJU3GPss/1B5Y+xtT2zrVh
j+ouAdHOTxX7VnOwpYi5P91UEdlBOG4ez6eBc3BMVi5Mol1Qgp5Jr6eHrOU11DEg
+G2HD2jrl/C1hWcPUJSEZqqH3hkhQ25iJxBd0o16F5W7NQ2MLaDeE2/xGZ5OBBPB
stf0dFsoohdVtIM6laOIVeZ+TviAh4I1JJoHZrmjMRjpZ7vGN1Idjg7z6xM4YYtCl
piJ10n2/rr66+GS7pQcoVOuFAyBnblEg1HrJTfDBY6BAG46Fe03npuCYpiBGoFR
4I791+nN85fE+JzuEuny182ui/qtR+PQWeNV/oiV8wmhCez8g2zDmuLwfNcajJtI
xQSOvH5PNt2XA40jaJWv8YzHdnEHdSmV0gxm7g7TVeT8Ez866jn93FwOKo17shfZ
9Y7TyDCRiCg8hAi/kEM8eRLOG2/Lgb1jMH1HHTZuguE3DYf+LhGXkcvmmwzpa1Z0
vLSKYRWObJBU7ag95fr4LptxD0nVfzXytesyTYRyyjceeqcPNieg4c46mYxalmU9U
BZ1p+2eM3AVLiW9+J/UmWE1M+oAjKiJ7C2OjNda2ap/eCLQUsvoHUNQKLz8uawn9
zVJiD40xcpah1F99YhzGTdkUf4vSSaoG7J2g1y12kto6eWS3SawEnm93qJAVDQFK
I91T7QKqJ305eN9WVuv9+uQBgzHBUfMgbaeGtlycTfasOD5P4y52hp536f7+jS9f
bjyLRnXj2Pzpj+fr5XfksMU2teCChJsqoED7EhTeymOg00Ot252dORqQxb47Woy
xRHi40jusIM+HWXCMMPRPYsHESSG2+Mu1IM11ZN5ofSEUuswoFaboO/ssZaL/Xf+
5rhPpG09YC+I9ZWYyotI18HQbf1C6hylXTuWQo8bU2IsuXCNH6GdlMJiUtKhLGk+
+RAhVnCc9A1abcvuAYCDFnngY/b78DIENgq5cmSnC+1740SV3TdxVIVEmz8oCgrt
2UMbnsxrgmTW6qDLZdF0bda4854AI3SQ0G3UUUTTqk8+/E2HOVXKBsPKPKIMi9md
mlRE/xKUVsb/RtW2AoYjDEyciwi4jCc+nyv6ACbhW017v9FpcHAb8QRD8BxTo2S9
bB5J72cU1BLec3z6p7iYxn9G9GzyHb0R8kbTcwUnFsP8/LGhN9Lx911/2Y66t/2
7GtZkv6xcttKPN4xDfSdu6Ymvjh/2EjvyvitWTXCMmbVTrkLu4DXeBW3SUYawjxi
8UvT441E6oOK669K33yNnj9q+YtuUWm/vx9oIICcv8njy44W/tLS74wXasF6T9nB
OdZB0NVb1cA5gCgkMyY961BkTe0h0P5gQjU2cxuEsVc9FhEUsR6j5IGpPJAsmr66
HqUKznyG28I+Khru69SZnyewyvKMSnlCrMSMTsIDn7vfZmB7nDbwhSITm7t3ksfP
/weh7b31c9dqlm6Pi89ZZ1hCCSA/VcjpLT0SwbjvG6s7Z0JX10en7Yxr+09RxghB
sfFSWHHhwXjuVC3uQyRMtF5PN4HGo5FI4tSqfWnK4ScVVEKX1SxKTIrJOkkyZTgn
4jyvntoOV6/ViCIEeub6qd/rU7H6I/01Sio60W+hjgqh09CcHz98fH01CoWK9+0a

```

B.3.4. S/MIME Encrypted and Signed Over a Simple Message, Wrapped Message With hcp_strong

This is a encrypted and signed S/MIME message using PKCS#7 envelopedData around signedData. The payload is a text/plain message. It uses the Wrapped Message header protection scheme with the hcp_strong Header Confidentiality Policy.

It has the following structure:

```

application/pkcs7-mime [smime.p7m] 7735 bytes
  (decrypts to)
application/pkcs7-mime [smime.p7m] 4712 bytes
  (unwraps to)
message/rfc822 inline 878 bytes
  text/plain 319 bytes

```

Its contents are:

```

Content-Transfer-Encoding: base64
Content-Type: application/pkcs7-mime; name="smime.p7m";
  smime-type="enveloped-data"
Subject: [...]
Message-ID: <73a42f8e-8f5a-5c62-b982-82ace766fd32@lhp.example>
From: Alice <alice@smime.example>
To: Bob <bob@smime.example>
Date: Sat, 20 Feb 2021 10:11:02 -0500

```

```

MI IWTAYJKoZiIhvcNAQcDoI IWP TCCF jkCAQAxggMQMI IBhAIBADBsMFUxD TALBgNV
BAoTBElFVEYxETAPBgNVBAsTCEExBTBVTIFdHMTEwLWYDVQQDEyhTYW1wbGUgTEFN
UFMgU1NBIEIcnRpb24gQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIeh7j00
Boq0MA0GCSqGS Ib3DQEBAQUABI IBAIYa3OenGvm2fxVDHCD1/mOK+G0pkvIp9vgH
9ie1Xt9FsGcfZkoi6msDh/Td2ZLZXWyp3RCOcqvwu3e0M6IEbbWhFVAdgkfJ4k1a
wlfIpe+ECDsja7I4rP2Fle1lPelgQ0yw+pmG/epN9Ga9FVvfKhDTHm0Zr1lMnJIO
FRuTtU+G6A+hQJrCz+DVh/3ub7P1DBomlG+bL8P IcgSzVwigtC0Hh905uZwB8ypd
CE7R4SzfX6u2/I/9K7FgZ9pSp8zZpi5WvcBuJvSqeLgTL08mm+7AMAYHEld005y
B5GFc9fTTV8ByIleLzvFK4x18EnFeQNVtCpoIuJ+BxAihm3OahwwggGEAgEAMGww
VTENMASGA1UEChMESUVURjERMA8GA1UECXMITEFNUFMgV0cxMTAvBgNVBAMTKFNh
bXBsZSBMQUU1QUYBSU0EgQ2VydG1maWNhdG1vbiBBdXR0b3JpdHkCEzB8R0APhiY6
HGLS64Mv1sDXhpQwDQYJKoZiIhvcNAQEBBQAEggEAhCWApYit+JqyC6p1+Y2mE0rR
LziSUECz72cLwSS2GXyl4YE86WTYQPgF5IHUymyTwtngy jKZB2DUP4jOCqOouHJQ
cEVy+u007cYIp/K1bZY3mKy5EQkdlo6qpOYJmIs03zoQfzYb/5FxBBIhudMqB5U0
t2kPtnlgFsLbo5c4FTn CzVBezJRyA1Gw/tQeZU2Rfe8xySkKEU00vUkIVI96X1RR
UNPGVg072/V4w/Yr0oF0ZT36RZdW54hhccAS1t7VZoiV8z09xsgS05xvs5dleRzz
DcaFCz+bvtACJs jt/UIf4PP1 jar9bL9BYoKzI8ypqzxfsmJSYiQziKpEwoaJSDCC
Ex4GCSqGS Ib3DQEHATA dBglghkgBZQMEAI EED3mqLx7mUQ1V1YW1LnecdmAgHlw
+jeehq0xxQt5o5VAsKJcy0+00gqRre1nhO/2cQRsFmJHkOhTtWzY7H6P/0Ayw6iG
KvS1Atb7J+ttV21T4UJEzr9abvMIGwZ2wDNZAHuyWv7hKVuriVh/NLsDDFeJXGJVP
XJ01saqeGsyx4UJmjV3a1s jtqeEzcU8Dz0TA5133v1FNXR+HB44Sejg3zHWLPw+2
MMc7WiNZeIcovrOKR8RAuBER74EawkBsNoAG+itMP Ir+iTjXD1AJNOADfz2SBi7p
zPMS5ypb70F0xnLwme3MS6QMSkV5Qg21lDVzDR4vfcqgLWkjN/fUOei/90ERrY6Cx
Dwt6xloy+cIi6DmMKBiVnblM1UdWhGsgmaA6LV9ZKm4BFXPxZ9HJRq8JXgRwBXR0
ih6xjdjkVzyPnB0 jeGInCRBz5vPp4GFUhXPulwJzuOjVdvMecqrciyF/sN/RfqGo
KmZ2YO6iKAt0aijTPWeDprUeE3BgEQ0DwyjySWTsnAdqPBCT3XPpUV62nhb9Iu5/
P459Trn6R0LapKmeKdTSj6QC+pnDLe7dMlynjzirX+EfkFJVSiy/PgsnQ1A8vRut
1CtmYTF3GAtBd4K58whmTBLBzyuJlXKNmmZ/OvfalVZ/+Zsz+vNdgvrE+Gev2kO
PGn+OBtx35F7joWW/HVgzhySOztE9/erD/1mAc5Gi+YH5pv1i7Qltow3x4srGHv
TbugWvLVdIkzufB8k7IlDyMGYrAP70BK2ogKd2J4QqDot85YmwPephof+R9SszU2j

```

PyahZr1xwglLbuw8Qhv7pad040Y+Af55ZVktcqV62T4PaYy3Qc+gTOSfcNf7BoR2
aIsaoX+OQVuL7SQo11tzETE1b1iyZj5Z4DUWxyqmrz4fJHKm99YubT6qe4n1CTFs
NrRcris570kqf2EjIs4VHzpN3bsbMG1Qwr51b1KXT4EjBO7LFeNppze7Az9Vq3aM
ZoLQ5YMG/OFDYOVIOHqjq9zgoRw2X5KaXC8Fzm/hiSqRVNtnQTXtQaVbSWUo3voP
BX+0zL7U9EGyg3/ZwSLHsteGIoDGA59cFYaG75GTFER618r97ETk jxmxsYbMTyRN
8HfSx3kQpm4ODyvWqaXZuWM+uzSQuTMXro84RtndNGUryVsQItzw8cCTzw1ejwj4
9MC92mTKgXkc5ShYU+TiKTchBUznGj27hklFmss4YC/V2Q2X5TzdFj601cuyP9QH
zBLYkAgx1+wXuyr4Q8iYy2JN4eC+LQitnzH1EANrnQ06quwQPtdt4qyrF11u7VN5
wF84SvB7KsJa j6ft5FvsPjafdp5z8Yq585ytPwLQ8+os0fJC3GosIzngpJPx/13d
+4MV39BoENEB3AJe3UHtafueBqmwsZG1ps1gcX/Cnrrkrncywi8t fKEVXRaERzKw0
D41TD4R3R1w5duqTfVJ8c8gSDR84UW+XZ8e9aXRKPkKQGSvfquuTDZ77ed+0Y5+g
2hse1k2svSQFnkH+OWAcGZy4RarI6CoovVbqaByGnwB3G6R3rztT6g6b9kv+qps
sOnBanfi4yEoYUVW9eo3cqlnjo63eT61aOcl6DqDRo95D5VLZPCBt2xBh9D1KV+b
6kvyLo08/HJDIQHPnsKwoGaQMxkg15kAx0aDxKp00IFxaUU42cxKMyEmrfzF7Dj
cXH/++jrgD51788PaAfS1L73WA1QafBExqQe4t i i 8gPrjCyVo3/XsIcciz1TJDW2
OofINjUrCW53bLkxn5xA40FX7zOBGzwnyGbfhnU1PDEthU61q4MU25UY/tnFPqK
2GjtgQrOVpFlitYCjxWcFoEFWYCy774wU6juHT4nDuKpCLXuJcnyzRLbmRnQpO81
skocHI5mRHtDYfeyzioG1qGG8wC0c8JX3wrXHX8LSnjkQYf4mPiClzbzWXSAS8Pe+
xwxV7EvU4maXQBIWUshvL85WdfXABKw+cvG/dt0OdCt8yz2vz44qf6Bnt6z5jMJ3
SW7Bc/4FFH5W+uZV8uuXChFs7aWVW/rWAcB6saT5KoOm3EhrxXxdGJeeuOP05xJO
UR8hsj95Icad4yP2mtnh7kKrTXtv7MsRsJKqLWRhaeSdf2XBvluo1V31F/mFYmaZ
gdvAyxbL6rY5dljH2moP4TxjvaA6V331FYCecnv/e5UZQBjLE4WEZYAAuTwgn2p
/B85JP1/yB5BP5pzm4zM3ye64BKsmjN4xwsFkHuKg5whYiKQ+/BhL2x+Jsu8iY2
7y1oAMh1BFADf5DjFsuFeouIj8P3wpPMF3FVsv4hgkQH17zZiNtyaga9q+zS7Q
omgiUOJMcv3LSXnjkf8GhyqwEuliZDmnlHBMMy90ASC4bTIFHa9bBG5TJzDnNtUiC
FHncdERJw28fod0FPvZQaQuvN+sLJ0tmaycsOnIkGUjqxu/GYRR2hBPo6QEuwXAG
paGbRSmSLoMg65AEk4XKCsTrEQUWvejYnIi9G8J8fu5pLoHZ7HQUBttthmah3S/A
s/yRcqqUz/83XOJuf/OybwGDGRGS4Y03Mnq9H6owu3F2h3BwASjK//nflxm7AvpO
RxsKZ/s6dMLJjWA9+g/uacJozJa5d5EY6yY2TKR8/T143/b31aJfj1rfRcXLPBJ/
AJQK60RoDNFmmJKIF66xB6g8wF2pumwve8XW/BK+c7baEInlSnMqHeqpoACrk9BZ
Y+hM+2Pyq8kK2hvwr4eG5C14z1JSwrT55SNbSY28iJUEJNE5dAQzgy1f3vg03Key
pTCAyPZ1nZa/1ttJokiHwdSq5ZdxxRWC5WZKv+9bHdgQqqmEyNgasTAlkdjeriZN
pQMxHCvUBUggzpf90c/GOIx5F2P9f9cRVE5eHACIGn9noZgCrLsJ0VMtRWBy/dCa
3eS1++nDtO//2DHkUHLjdIZ1fcbqE/4BG9z071HZhOP/Lu9thTQomut1W8s1r1XQ
LYe9hz9oPTVDsVxNF28k8YKuiVkBic+9tw12H4pFyBhJyJ8+mhXm/dbkq2ivrRYJ
0tP8vrLftBt5kocdUYzpaUQR5K7PAM+pNfo6vpOLN+ODgVk903fOYoqk7GmfN0YU
wXC3tPldjOSQhNCiH6YdMljREzY01Yf4u5hpBimeyS+WauFcNu2Misbo3e+4x3zA
3DyNYJKgj591NEbZA0Nd8pQ22qvBpsKyZTIav91dItntSORY8XTvXmYj+wYt/0wB
/1/G8jrnXetyASXo0gARxWK02+qpn+1waz4ml98luqy13fwp6F5X3vimLQTSgIy0
RCgxOzZZBQxX7fFrT+erTGjtoVMMXaLwepMqrB6aGAF611Ku6bQ0c9/RpYK+tegt
R4ZqLYg4Acmam3X4mRmX9XZC7WJzYIFuJRQw7/vFPBz3dsrtt/F+j8ggLF+NLM2B
yFZMUYJmrf+jU8PxL4mHI5UxLjKvhZ/LyNG3jbTXT9jQeK3AQ9HCUpXkFhbmrvRm
LQMXiFq4gGwM9PgAweg3fY06TEy11aIjyDNNeIld9vWCiNG/tgH5NocZSUSbOPSw
11WCNMJRiJHAjWGHquAN/seBE3gCFftDU6UgZVAgH0wE3z6nVzAzrmfR/Lwe1kU4
T+WwUYBi0CMzn44ecVwA4n3GL1aWgGcKo+g66jUfTtng2IIn8dm84QtW7RDM4Lku
iofBom42+RzL7IYOiZPqzAccfAOiFb/yOekTLaktqrksv5P8PkNkgGFBFDrmc5br
VTOZVVDiZcvQZ6kvW1Hd0yHfOGSqM/YPchvUGjof4khiq9XXzwFamWw1knjNR/Lx
NwCDMKtZXEbiuGya/NZP6dKaZhSCHMmElY12TVgS9+q45eY3J0hHiKnjMrEQP/j/

txu2pqqDedURvXNst6R+R/MIlcsUoRjag3zVxzTaJghdZdsw88WO/0IzTIIzmQwd
v3nDIYOZwVwL5QnjKgeMDK8Tr5BHBjDdV8QZOpVtki7/EgJ28ddySuaxqtzzYMX
sb1eBNjSIjgx0a3k148jnf6V7PqVvy5m/2OPcQmly/3qt178b8N2cNBzBavyAKnUM
68dCfQ35iCnVUIfIwfnUVhNKiiKAGM1/6GBAN4aUgqdlLq4BBgJMU9aYRObiepXc
YVKXWJOjsKOaHKTWWTjaWi2DEn3h6PkLidZm2ZMm5RJSwX5H5Qj4Sh7NcATBZNnS
bBOgTrBj4ygnhnPWStTQOygKTVZ9beT+GLsJGD9xI6vejiro4j/Vw3sYTYuigmCM
ufMS8n8P2IB/DjVU/GE2+dZ5mL33sUbjHIHJ6J6+1XISEI2F6YILoCK4x7gBp0Vr
5BacDYcAwfgbI45ZurXWaxY2ij7zHg9mupavujjwv6y9MuLfkHR163xEkFX67ZOz
u4aCFQZ/8u4WiAVcyQKTypzfNxxz117azpUwT7E2IEpPF/zDVpeo7K2W4fHgrG+lp
lNc5f7flrrbr109/V7dMTmqocFjjjaOmHOvpVv1kpKoscVEoEeSx41nMmyPyJkEDP
INDak1B9tt/t3q+vEQkJKPKojfQ1Yzchs41+z4aJ+4ccU0+3K5tfrungA9LCevnY
+R/RH+TIGxGMW9WwWjqmKIPlhoD8JmUK9tYC0JHwB0KL7hxf13sIqI/BpNGRZ1oG
40HdzmxYZW6HQvWQtUYFxDOa20ztBp2rRxJmHuB2gK+Wd0t2/HXxQe1JjaW0YQaF
nNmee7PTMk1bCBYr4cJzmOCfTtHAdHN1jrzY55BCHntWekYhk5GpzaMttu+4BsW2
lSrurp4xY1zrZkUYGNXLgU0/hmVCasYJSShypp/y8ZGpFI6uEzHY0gok0akWFLe7
7SN0PdxP3abKrR1ROInFV5YC1hvjsNEStZZxk2Jv14j5q3d00CWROB/y6+P16954
jSp+il/Fop41IpAt22NZgwC1jMg89aTnK79THy+SSj4S5J/2h7QaS3v9XdGKmJ0J
msVwgavzK2amj4InTp5/dT5nMAA+GgvvF/8+W/NNc3yTSG/D3M5re7p2Jof7Ueo1
Kja5SytmeF5+Ot6fhwQhiI7nUZC0dgCXg4ZnKR7T4Cha9WB1YUotNrGr+Xi2Y7F5
nJ16NC+K2jcyxf027VTNA3xaOhtwg9pioeYaZmqErIRhm/8R26ganjVK8Zx9AmxK
sn25U99AmTeiMNxwMRRFFQC363YrcdX6kz/YV81DvEv9SeJ7psPYlCkTVJ+OUn9T+
PauBE+VH+Df/CAjF2yJyEMr+M+QziXqxBI5pGC9lDRQv1EzkOKwbs0dlG1Qroafw
KbJh6WiJufkF0nInX8FFCIUKF7f3WoqrbGGXm+rgdGChmxv1T/vEuPSEhJyzX4
pA52Y6LUOg23VlibFqWZvtZ/SYG7gZ4mT4iYak7ba/g5NGLBi4DCstHKKWRB48OY
bd/v/ix9e1l8Pno0ximW9AI9vHbZqAmCpMjKMumYiSh3Uuaxfn3Wv5dU6eUbQzo
W7yrSBHi4Ik8tbe1XjdKHg0Q90NHbxBMIZY07NC8gTM/VRUNOR0wZkjo9yzmu/xN
CDdNA2mBeFwoa6gkhUOahSLAgfCcHYKN0yv0JHTYULBkfgG7Dvp5N1j5M0oDhJo5
0CP7VXRzUyYDUbGzZWS/JTH+VroILLUH6exoyHIJzrwTRGqEZmEcaOv5/r6fGYQdx
UMWRAAuh2/IEDketRdcfnRZLv9jmqJj24wFjcuaiqGzFlYj8VWvfjn1hZDUDpZOS
a00SEBBTr7OisiryKiT+fvaoo/SMm9fu+Rqat147j04FcZYHaDj0GE7KBEQe6FR8
S6jqCH+/IbFDXj/scyQAyE/PxCW2BjsihktnMXqz7D7+8C0JYiQpXw8VegGob60V
R0fYbKp9R55mRpxI9th+PhEhggRqvM7sf1Byaw5K15s/+M43RPzL3hDdlgDRGFz5
jKEYDNArBSzxUCrRfGU8q/OrapWmIjAFdMcH9MSh73X6SmLMTsHjniSCQ1nmbZUQ
uWND/WSArv0cT19TpVRWgPYZwQQFPE88x0DwcIaqz0DFpWgE/4ccx9uyQwfZeSb4
K2cp5yDrYxdTciH30Ha9+w+7/2XK/AfEgSBMtoYtkdN5yNggR7NLhjc3MzaHLhQx
WGDSSxoEctMarW2aXUTpzIvJwAM0z3Z/aF29DnihMhTWC88s+rizq5abnNNTodQ/
1RUfCGKmV48N5QrtR6UstwDqEFyMqLGNqR3WNTQYZM+4EiAVEVecZyjoayQj7hF8
4vGvhj7am2+BDuVCY4r9wLu9n0VCnic2wOafjm9ET7RmuhoebdVxm5Dz0g007bAr
lxhMmax1jhzQkS9T+wygwTBVedEJPb4H0EMa+E38Xj0610Xhh/F3Dp1yhW+RD3oU
jwrH8KIx1e+RNOR3zmkr8I5RFaIWWY8lQk6YgJvbsKjgCSPg+/hQ4cL6uLaGxkJT
gk/BqMWgKsJFrcQst2zUg46wBjHJF+k11cfcvkvp7dMQn/CbmiAZ1kVTGMgHkrzBz46

B.3.5. S/MIME Encrypted and Signed Over a Simple Message, Injected Headers With hcp_strong

This is a encrypted and signed S/MIME message using PKCS#7 envelopedData around signedData. The payload is a text/plain message. It uses the Injected Headers header protection scheme with the hcp_strong Header Confidentiality Policy.

It has the following structure:

```
application/pkcs7-mime [smime.p7m] 7605 bytes
  (decrypts to)
  application/pkcs7-mime [smime.p7m] 4630 bytes
    (unwraps to)
    text/plain 331 bytes
```

Its contents are:

```
Content-Transfer-Encoding: base64
Content-Type: application/pkcs7-mime; name="smime.p7m";
  smime-type="enveloped-data"
Subject: [...]
Message-ID: <27139e00-e05f-581d-a339-d2bd43bd0f42@lhp.example>
From: Alice <alice@smime.example>
To: Bob <bob@smime.example>
Date: Sat, 20 Feb 2021 10:12:02 -0500
```

```
MIIIV7AYJKoZIhvcNAQcDoIIIV3TCCFdkCAQAxggMQMIIBhAIBADBsmFUxDTALBgNV
BAoTBELFVEYxETAPBgNVBAsTCExBTVBTIFdHMTEwLWYDVQQDEyhTYW1wbGUgTEFN
UFMgU1NBIENlcnRpZmljYXRpb24gQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIeh7j00
Boq0MA0GCSqGSIB3DQEBAQUABIIBAHDjF6b8nYADPvM7jm6fi20/h20vJSvpXabk
JPChxwLJxY3a33r0vWwEanKZo/k1fbkxXa7w+FqMEEM/3EsktY3EgsTBDC+vN2Dx
1/hX9wBNi2D3emJnmwEv8vOmNxGeg+P+vZN7WjM6kqVrUgEYfyRkzMo1o8YNaFgB
F/b9ss3PjYUEKN+k+oi1Pyi3GIxPw1KoYyO4LXX4QQhTFOIje7b9UOZk6zeoz1qZ
sBQjrOnh2bKeSEnWgaS+61RvS1FKweluIyE1OuUUvx46WQXVJ4czZmdnSORW0+nD
XbSo3Um6fzW07Aqqbw82qHcg7sGhQWhbA4F2Ud2aM8p+zviUEn8wggGEAgEAMGww
VTENMASGA1UEChMESUVURjERMA8GA1UECxMITEFNUFMgV0cxMTAvBgNVBAMTKFh
bXBsZSBEMQU1QUYBSU0EgQ2Vydg1maWNhdG1vb1BBdXR0b3JpdHkCEzB8R0APhiY6
HGLS64MvlsDXhpQwDQYJKoZIhvcNAQEBBQAEggEARXtsiPCj7mfzszkuZ4d+30YF
Q2pPbJbLfhl6xEI171WNKwmLMtWS10oQ40jmxw+W2/yJCMtUbr1gXW01kW07ln0
ATq9WCN99ipuScfQ7mfB1AsCelAoxbEzGtrNX3IInAk59oN21SK1tH4hd3UCULlo
So5A8AEJODYnzb/Wq16ln1wOvAIIOUSVa335bEoAMco4rS4TitZKYdFnD4PS6tB/
8hU1vet84cSYqoFT7Bxz7TfnP+JksrSGrUK6dqWiFPJbbQhtNKmzpSM25Vfm1gHV
hPX7Z3HJiYpkGaYVmu89MbX52WeBrHj0BqMAk3ufG2exN0VxUI7j0burMpZ+tzCC
Er4GCSqGSIB3DQEHATAAdBg1ghkgBZQMEAAIEEPvMKX3d5Gy0duoR8bPf3G6AghKQ
LqYpvTxH6buu+cekW2Pe2RA6jN+IBCcBJ+6cxCKvOPPnwwCJ69Zx1tM1cIVpUkuT
2TbdnTeSqCD68rvmVVJuwagJxQKiTOvRpxNTj+jUssmuMMiA0WIff/M5FFQAGJhq
d0JadL7CjuJaHYu/4aw4Xk3Mmw26Ptp2DYCzr316UksQwHW+OnDPX+BEfsc41Qjj
```

eup790jAXl+111wilpoPQrsB6TtxuIr2z8J99L6t4ZUT7WHm1UH6ukEeYmOjWIpD
9UD0VD7jZCAK5LE+YbDuoYuQ9vFjMnDmvZoyH5WAvSYsPSQS1M0oyVxEhKugQUIF
aKcp/fgnqcmtn9ko8QmVCvZpR7Jju84Dhc3Bpf/Y0ma0Qzqpu5IYcPmrnany2x+k
hDQaRsrJzke/d0UJ7djUHuyeSucC9qj9Y7ch4RtWUjCKhsQs1BpFmAyCjd287CXh
a03YYg1/Z3o8D8ZrghJ3xmmHt1hCH+1SOBQnPZrPCOSrDkU4+BAw/oGPVypqYUaI
WJbk4xP4qi9EtLOHz8jOhMrgFlgMthbicK+kkHti3bA/xWM5I25N9mvDYjHHRhuJ
0RcPBngxro6ZGrxvBYhXtSOGEn5RbxCeS7lZtAK5XcrAL5DV7mur+Ehp3Nu1TDj2
2GSNuneGWqYMMT8dvFG/UKmt50dmOockk/x7UBMJ3TX0DQUxrJDFsUVU1r1gbZwuE
5K16iZsNxoaZUi/cUaEv1ZHxN9GkM0wXNATMcbcbHbbxxhd5+Zd/PJWmTbWK7Tde2
Bir54zdAo90jo/0AfT06nuQsVdM2LDr3PNEQ4aRLJzIDSA8IrQVZWB5wQBwS7Msf
2+CKkYwaWunIJ0DVUQVSg72bQ8xzT1NhuwFXIlekPECI2B8yaaZeLT31fctGzvMq
jodeOtgynMwWQBmHVw1yHlagnIJDjEXVL9Rc7jOwv1q1rHE4Qm03EObEmQwmUHS
DA5W2ODPAuH373jS32Mq13lXastFG407kZU6vHZ3HBtrBUmYHcK7Madx7/FYYEdi
tUa6an1BEYunHF's6srHl1t1f3v1iIX9UCqorruSGyUfw1NSXpLxEElbQMigNu9Vgmx
8nAq51UoxS7TALdy+xn9uG0JH4JbTptWJqhnaDJfUQfHWBKTnQmZLElIy+Dz+BvA
GXT+V6Ay7dq0Z034+NNVsnDa9rMqW/C8uDoCgADb4+JVQ2pwZgmki8FPHpXDKM7V
HX19hK8WGNYPmFot2aNuJTRIB9VWFJhUCNpJgc0xhzbTv3V5DTOMCuXkrQHe7JjB
eR9BGvUs5KSjB2KegldfeFJSIz5zAEZYNeTkQVhAGd6r/OmYj0YJN63eXQCub0Lf
q920ok8k65cBl71HZ96fJqTDAfjA3LkanV5RUAWaTQUcG00jGgNc1E4pCXa2G3au
VN9iES00s9wbsE2ZR8Hk7ys11qfQk1o2drqeKFH5pKI4bhmkMjeLIa2tzR6AxxrnK
jKxIO6fpCAAvWXf9mRpuJ5YrQBqChe97AqfaNwE2CeSOk04FYIXeOm6iK4iCFLOG
xG1R8uNt+OD/Z+6ODUGiogzH8GYjbljDbLHn4q00hZaNiB9eCbP2Hx5ighKA72Wg
nGaUCQTTwS2N+Xty/u9HGXXK9jDWBrcTo5YhT7TQ0MiwjRAZSqRii4VBjyAQAnTi
ECS7wsAuljtrPChIW/JIaXlqDCTBg8hirdDXSgk1p9ZmC1NmOtP707c8zXRhRii
NcgZowC3DG7asJZGXQWoje/df9XT0gI1ucxmndRwmID6y3BQ7qCge8TubbnDHbUl
cAinpK16k327c3IOD54w1C+BNhwtRgtGTC3CXMSmEBqQ4fJdyvM+sLSNS+rxMhYR
K0Wr0lGT7I3oNHSTjUQ+T/vuOm+blur0ziYTnNlcWYtPYwV812hdKuHSkeEXb7p+
HBJE87wtIHoYpM5QZSqWBTFAuMTPD/+3n/w3UqFq+xsq58QyNFU4007+u33ttYda
+UXtPu4iReGCHS4Ay151t57xZFZMsXHVaTz7bQ3pBe6sEiXWP9uu2J1GZ7b0N7gWN
HEWmJkp511T/OuhmilfxwdCQFM2oJvftqt8h3ex5qAn4hHgipUFI43AaSAj68LaD
wQb67bSkJL2pMbGwTOFNkKb/Rf506ytV16S9CZJ+62Es002vE9b6c7uJQkqmdZW6
GQkKzxPR0ghsComJ0h81djW9BUg3qit1Oz36GKDHU3PkmjyPlrTFamByceF1Bk7q
FLASGnS22UQzPS0iPpNJrsHxr80e9LqVMB+ehs74gDeQiULdow1cn1LwRb1JJTkc
Qy3Cpoi5Vev/MTV+O2Kh3R5L86U/RSfBLXqby8dQISbEGUxIME387kI2BjgDKV1R
ypOOGUBTneqpeBkzh7WZ01713a6BC4sunMqkd6GmrD12V0/AWcNDBU7S17W4IQyj
sSNzMIeCE0gCVAQ4cJ5AnyqSFkqwbzCECBr7Ojbx3zsJOSXqSNvuZKzj4iQnbmVN
rUhVnU0a1gDozNXT+jsRUctKu/OYwp+MnporZrdMkt4KZ/E3LPWnLY0tUBcWgen
KY5ea9X7rPuons1LqMEMrLsn0GWQ3sDRw42vIPN+tmJUoeDTqfaW6knY9xvT7238
r1HcX6bqLLyQdBl5H9XOPEDiwH8dwYuHMLexpUw/oJ2q+qD1a4Mmboi7UYmaBTWn
t1sFSUAmwKt+H5kT1ivROq72KwY9Katrj5WBcfZWdcPaPlogsF1sb41UzVc6Nwrp
MVjU+f4i6I1N13UXtAKcgwzUPR/QCQ1WfPC4oInSceXnmUFg/R7aM1uPVJOR370
5yoIy4T5p0H2I0jiuO1Nk7g5Mt4GulRXVx+mfDf8xytnh/QcQDmGER7HkFGPrHnf
Ye0fjorSCNfoaJkzWRe+S8I5MjT0KdsEJlHXhe5HYMv1OoYG3bbvp6181FFhIqz
EzkJKm6QSF4ucQFQBHhxoyPO0ck1o/XO2YSmEty76cNdgM0XDqE5RY0dv6xx/Q7
oiN17uNs+sCX/B5GXek4cSX4o4DpETAerxHSQ+RTt0uBGXdMal1MzXYzvVSwFu0w
DZ5gk3U6o13u4d2ybyXb6FdJE6Xa2HECqY+8rjtAp6kH2DWT3+ZI+c8nRd6c6qN9
pDQU1+IkkgGB110TW+Y112fvOqdFar6K3sNHRby4dNG2o8KuEYT/8ugX0vubsioz
puXFdGMGTtYdwlkDDH2jNot0LivJ83jCSHYHKh02tepBY03k50/c5+/OUAeDDLeJ

BKj18gSxmSuxbdoNd6bOcr+8MavjblUj+FUJX5rfeYaam2hGe0EUzy7xUTFqIsFb
3FWos6oUdW6Je2nBEqitj9JmtpOK1pQV/+HtGQb52VZ+VrfEhQAhb7AaeHTo3s1E
i9m6p+61WTomxSefzXVKHayZ/M0VedRHba7aHrZoy6wq+QkpWGwzGmtr6RFoXJtg
PloLaAGIMqRhpDJ1ltJMrbyY53Nn2GIQJKz5pzyx5Q3Q6RzHvYBb0g2n8wYsmV0H
GzT/PMBUc6QcfWdNTIpQooRuIDL0p3iuNO+1CU4cDi5hiJ6MWKEyabsMqw9MWEX8
9YWG8j0fHOcyDaHh8L/kcv8kS7ZkeyULPrJg0LYGfCWUjhBuHRmjiAGlw/6XddgY
MKeSedaqY1k6aWbfgz6P3R8q8tnooRT+SSgafEf6FTLl0RqhcpEbR3Cxac0f27GY
6s5WyMCMpqs56o5eUExkDHlgcY7en+SvrgJ53FbgGiEtX06F0/OXSTn+ZqiCmJAA
nF1hCWkY7mqfFh1RfWba8acf2zC6H8KVpohS4ysfpildCSvn144YkXt0q4b1A1RF
Dv8/7nY1Exe8PbAve8ZMWco/ymkk63Jd566Xc+wNT0KRYWPA20twc2DADreIiK6H
Lh7rkX6ECjN3BouQjsSszZvK1UH3aUfGpXzR3QAPiLG7FBAI9VnRv6+xp1u7eSbV
xPQvaCQL4wpvqlesafxBe382ortk1jN8QkPVBHaUWbg/MGr1B9AzaW4MoxdyTiSH
401Xvci3ed7dMcdOpQo4yXiYzRGqUCABxbQt9uFjATfWNQfpYkEJ/Deqg6mEGmdC
OYt8WhZ11YQGzOAtjbdjJtVbc8sqe9C4c9q040UMQbBKEXCfN6BM60ZDam6AAUM4
gUf1zf/yuIuU13g4GJE8tQH4Apj3W/b6VDBcbqPkoSoq8Yeqq3qU/DVuKMDxUa4n
Mus4uc41tp5oDHUOr+/850IURWJrW8Kg5+uEduYAmw88k6s8EHdmEcr08mla9ayR
m3pRnjAmN5vqrhq3Q403qXFv0ykwHjC2WOjmiK8cAmUG18H0JAbAcLyD5zHNIG
PbQB45HCp0OGvPhD9psTA6eRkpGgtxhdzwwFwZqYOYrDTIURWwhyf01V9M4ic4wD
coosKQViJ0G1pavFtNg5gD4sEbgfSfdwWr/91w+wewdfV9Jj2iOb12FcUSf2sqpv
cB6m06b3ZyR1cWABdtI1YL6f/VVY1omR60muzBhIP2jZgVq19DNh4ybyqAHkjhHex
Z9EqQiKt1HmleD1sxtNKvWdKLMAIRmnnxfrXkgWESVw4kNSvx71kcjOd6nYUt5ye9
IiYIHxemsnbu7hEdWoaOba7pTmQy6I91CO65PcLvU1C8aTP5m7IY7Uq+RU1reVS
1KcXieD/dXZ1k+TsC5UnCr4YjvCKLKhZSFJxEBDo72BrCHemHONC8gqvT68iOgny
GwsFYI6H4m1ZDUvJvMq2AGNgK8P1p8gcvjBhZ5rTlci2PugR+MTkV+F8X55sCtHi
NVZ8IcbctOf2OUd6hC29sKwc2T4mL2L1+aBxa+K69q00ovkcoEeuQhp7Qq4GU7fF
v1jG18AQn3MgdjK1gz2EoRfpV/ldPutJj9AE/6HNJLJ+EA53GttHHmTITkaMpFR
RPRihuaXChirqsUj1o0/7/xSch/N3YZqpfQjqsxVIUtY0aVvWXRRLKkZUByc5dgr
z20xjjkZaZKEfvwfffsI1/bjUeROKAPPRrRDN90kOuRSa6jMqwEp2rUtqbJLiNrE
Bd+WT9deckx1CA7KayNNnV4iaesg03rfB+D+vZq6NSvG64fBQR+Z3acg+EH/F349
2gqq5FU4XpaCtCP6u8/dDRKdXyhXy828ccNWJ376U3MGp0f2yv69hQxHZP0HH2Yf
MnpzSL+rvM3W7lmdCCBe5R0H9EhU5cA3Igc9CqWnW9i1UJ1hJ3YUaceTAU7maqAx
AFeykeFBrIXuFtPOJlpCF0hiKiv+ErAe18JsjbR4UF2aQC7t7of30555N577Kj5k
e8ACBNxpQe1tSYgxPtFmCHZpvSoca9cls3dBXU1GhhMtIqW0EfzMIb3Yal/J6Ex
NS2hKchqPCdXTUbrG5N14Oyf3QLMaTFCNUj4F9QiKJF6GkYpbH7WWuiGAKZQ3Sfk
VZYAovx4Jpu2p1ETEeqkpP/y+ZCfYEj87aBCffr6KMZV5Dph2Prgk6lGWQGPxh1H
b9yF43oTnrNhHvICmxveNRhRVNWNmGpCnqgTmzZYCsxEKauBaz6wE7RVC3/zrrjD
1F970wVw1JZzKXDWidcNhfzhRA0fYA4PwribzPsPQLOR3CjLoguFBg/O+rdjs55d
405UFNk7h2C1npA8IN1dnmJtLCTd6o0QWLC91S3lonmdYoBIClqbrRDW+1GiS4Ss
pWHB9IgpnieX1+wbEGqtdPPE3+ePW/gOZTgnRvGvZeZbvHqrCUoGsqldBjwLBvD0
BAHwRFavH2mj9QTxr2bZMNt035pFh0TnQ+cYnvtX60GuZFJM6LRydzWVurZXB1Lo
v1Q8PvIjPUEpAZx1k2qSRkreV97NQULQknjdcXXxVQCeF6J4g5Y86Cv1DPzRE8Ou
1xfNL1pFhQQyOQ7xjM2LCDkM2/o6HHjmppyiH0F6sg/Fk1AYysK20loKgFQdi3dC
lO9V8L/2Z0jZcA5gr0GwC0/Hu2T7cMeK8MNvOsRpI9dUQY5P1nQ2o3Ea/vj2qvPy
Zlow1vZxNcYml7+3AcSWG+W6Z70DJw1aOz2HAHiwPk1H/U4VJtFqJ+Q000FmWeE
tZkFcKcbivE2E/sBQ2fGnmf0ZF7fAx9D2CMXmoq38hJeoBasdfLCjIU30+S1on1B
IdVeWlnxpigFuyF198kJDuWcRxEIFJk5Bt8yG4KWYD+4R04NK/CPS56AyPoB/2CD
lmLZUeWYYGrqFER375gyRnCGPDaircopx0XiEh5ZGox3ml7/QdkHXvV8kx55NLGz
dnVerNDadBm/1oIBkWpeQ2CMnuJHsIGDlFYtC6N4k9cBBIHfh8dItE6BYuDCzcas

B.3.6. S/MIME Encrypted and Signed Over a Simple Message, Injected Headers With hcp_strong (+ Legacy Display)

This is a encrypted and signed S/MIME message using PKCS#7 envelopedData around signedData. The payload is a text/plain message. It uses the Injected Headers header protection scheme with the hcp_strong Header Confidentiality Policy with a "Legacy Display" part.

It has the following structure:

```
application/pkcs7-mime [smime.p7m] 7845 bytes
  (decrypts to)
  application/pkcs7-mime [smime.p7m] 4806 bytes
    (unwraps to)
    text/plain 420 bytes
```

Its contents are:

```
Content-Transfer-Encoding: base64
Content-Type: application/pkcs7-mime; name="smime.p7m";
  smime-type="enveloped-data"
Subject: [...]
Message-ID: <fdccb76a-49ed-50c5-9030-e4aeb83d7f04@lhp.example>
From: Alice <alice@smime.example>
To: Bob <bob@smime.example>
Date: Sat, 20 Feb 2021 10:13:02 -0500
```

```
MIIWnAYJKoZIhvcNAQcDoIIWjTCCFokCAQAxggMQMIIBhAIBADBsMFUxDTALBgNV
BAoTBELFVEYxETAPBgNVBASCExBTVBTIFdHMTEwLwYDVQDEyhTYW1wbGUgTEFN
UFMgU1NBIENlcnRpZmljYXRpb24gQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIeh7j00
Boq0MA0GCSqGSIB3DQEBAQUABIIBAGXrH1WNm/k3nn8sEvr1Nxi6vN9dWkgNKBk
uyHpuWbmQxgdsC4i0rQBk0W4X0aDdu5yYwt4uzqqfblIlgJQRnFfNt5Dj0tx+Wqxs
/uK0Fp8oCFZ4pJQVyX4idsfWvbq6J3iTIA0cPHBogIE4y8mMuByXh97VK5IGkvXc
RDYnE9vsYJY0Hpm//5ZUvUcNa7PeIJmrv/eJ0k jxAW7pa/64ni9T5qP8BKHgvcJm
YFYS6zy4UMjRNEftjLGNZa6QE1sy207BIZI3Vp3I1nvBCZI/Y6IHYN/Z3dKLG+Yp
eRhvtvF+PO+YeOLjm+o76hCIkXj8qqg3EYLV8dbbthK1aDgNO2swggGEAgEAMGww
VTENMASGA1UEChMESUVURjERMA8GA1UECXMITEFNUFMgV0cxMTAvBgNVBAMTKFh
bXBsZSBMQU1QUyBSU0EgQ2VydG1maWNhdG1vbiBBdXR0b3JpdHkCEzB8R0APhiY6
HGLS64MvlsDXhpQwDQYJKoZIhvcNAQEBBQAEggEAbVIWhJ9+bVLKFMdyq+QNi1mn
qFxmMKKidulH5s3NmRCYn9/nu82R8k+r4+FeVv+mrIIH90rG6v8pJZDFUDkG270Vj
v+ZmqSjLTuV1xsS8p6sOi/1sdoHC/GBLUffalroOJhRJ90aoSYnM5b9h4hWxYFi2
ai+WG6mgK7A5/LN1OW5em+aWzWNjoDNDzLAcPapv7ZjeKA5loyIutbb11Lgkta8t
b+hBmyREyCb/Qh0xS5ikztPqgDO2n39erubT09E0YzvGo7RTmb1DwnH1kW44Sdlj
wqVIwRlX4oIDLKmvPd717j7wEplmgAHCWVRMTs6E1c jNm+CezS3o9S+6CjkQsZCC
E24GCSqGSIB3DQEHATAAdBg1ghkgBZQMEAIIEECNM/iph7panVLTQtINOBe6AghNA
Qo2zwm6jSmU3io3mCT1Oe8vTtf9fspgytooplI5ZqNb1lqqiz4jdvAbqYVo5nnw2
arDhE1C1ZaLGxTnRC0XQbC/bltBmQepeQsOYizTIj+LdcZLN+M3AymhRPXWc0H5n
```

wibbdCxS9+OPP0B+QGfH95bSynkzdmD5vIiNuVGFm4FQOPnN3ZuPxID/OpVTZLAF
E9JN1SRdxiyZFDpNA1tduk3GVRuSt4Upb3X/jfTe6FhqDjFKCcx4D1WypmTR9lvba
B/+DiM9xrEry0mv+5eauxR6Swoclp5NR1jSWHCbD7g8viF2dVA01qefOm4+WwMbu
YbhjIDZtfWbNcYKtRnGOB33qSS6+K4Z0aPy0q/ACNzi/8srbxY+jRgIqimXxMCjW
y2hmPPct8YYx8333wLs/psld/zLowShPRg7Fsnj0HxDp1AKYbW6ja72ER8sDONS1
KpZ9JaHclqk9FWCBSZjqM+cChupgq74LYakwM+leXncSuNs8uMcaZYqrqM/nrigE
gIQ0jYOPBVnm2SAGOdLslexex9K9B86w8zNJKYuZ17C7I0iicM1kEGl7UO+Wu4V
XmYqLW1E9QmF+LFqXFQlhBbpfkRUu73us8VCyLN8aaM8Tkqean5cvvC02LFCySfp
hhQIPKqN97ccULAuatdK5si0RW1Hg94dzW5FBn87RqXKLoUYuck/NZs9r2tSkKcd
VuE9o90GEQwhWvcZYiZz9OsPY4NkhMHQ2Mz8FeVPmqEmzRlnPJRBgt9tild8UfMa
xzMfNZru7RRMP46WvpOy2iGvCUIDwaoz9tY2+Q/o6BYrOn3Fdd+HRik3PxrQERjnW
UGCztWCuaR3/ughENO7RkhD78sLGXe9Fzqj+CT6XxPREctmd4SDSE9SpZTKB7rnr
fk7+Y2wUf8Za0fZjiqtmfoLcJg+58fPGXlcrpqDbI+iLmXo/YaiEeLr40Ifa6R8Z
pgx7Qni6iVHLFHV2xUurrYwQMgtysG1ZV73kGmKIdfZuljygg2aytx5JvrKk4X1Y
nnS5+N7yX91I7pDj+k7kFJpDG6zEdiHyMtOLgEARvi8a68+6+oI0/QTi/t4aE8jz
xfQNWBDxOqkJtv7437P7D8RKJ5fKpoarCD9haO/WkZhi0zoCE09Ii33x/Ww2n+
qbpfqivl0FarBUwm2Ch2zCqF9n3xYHceJJz05UDqjn0i/obgYVYw0LHikI1Lg764
hy7xiuD8v24dOPpLbbVBqZnVTn++QsIy7UgkDOzC13IexTSXIwCj73Jp21Nkyho0
NIZRo/SJ/otAG/qMzA60/hip6tklqo94Ku0/y2XtdTc2NyKGxwWN31B8XnBIKDOo
V3d/eCDvmDFm1oOkHBtaa5Zq7c4uy6b1V6tYbqOoQSS9DECTKMncQh1aG3V/kyCy
ddK8cKLDVpNXzjzVYysL7/0Ata+iwjr6UpHzLEpU6BQWpPzkc6rgi6ornd5iYN1T
9DyxilBWz41NVg4XyY+C8iFmiTcS1/+wocjrV4/rReyDX8/f7IMubpwtxC5Joe92
bwrkHg0imSBZv1oiZBVjORnv/QKD4jZhffjMDTbGTuM1lowR6Qiyw0vgWXN1jbjP2
R/HcWFEej8HwYWTR9RUEB9GFnxPsdmv2EZEf944hp7Ic2JtI1M/eKc0r3VGnPIr7
q8L/4kMA1bE0bbyCKaSTskVD9+81+dNxBwPICArautROOammgmwBQmjsyfdCraCX
Cu+P3HoDV11s+Nu6PfoZGjEBBoaphvAkvrBboH5Zi6i5uw98EXbX/LsuBj+xpBeS
4ChlG/He6/z/et6zfnewQ7VvcmUwi3q5xIFMCMr6/w3u031UgPuq6CqMzt7wSid4
78SPQ03EmUeAtottuaKyW38pUiyfzZ7ZpBVuPCE9MXR9H1B3ccuGNJdtUcPu6UBi
ZZrkkG0ahBF25NNuTtTzx7reEt+LfQXQl1jxYO7qfoNa1VkJy6ZUJ570ITorgoW
sU3/W9sIujanCYHtJVHjnhFYFasbzkS7Xri/mrPx/P3R2f0FQW/LBJ2CMcaFxmK
JkpfzVBxHgHmv+g4UFnyECawrxDXoRuasd2F9AvB+YqkDLLxdHsbBiCnpjjetZyV
DSv5Dlpr19jrfbgqb9OaQVigeCZxt1WXV2nx6UvU8ZVfmJPb202eBiPK16GYyGni
cSdQYzy1KNR4Ge0sN1iCYOipwAYrwcDmct9S3A9EaTqy5qh9DeCuaHhMpQmRdeI
X7Kks5Q/8kSeLG2e3Fqk+tX0HBDvJoxpV56NdwHwTuyS6p8I6HAMQZLG5e25MLQ
UWkStji9ot11X57ZbKxwyb5FLXR3dsGORD2oODQxIqxulErogz6QSGk413I3c4Lc
YnE/ni5a0FjbSk/GoZWoTfe/11FRKJEtL25KwMSO7x6jZSnOQVKFR/z/gNdV0zsi
MEpeExLkPt8PYVCLHJ3RNiLEGZBnWyYptf2+SAZjNZ16G1UIbOX1CZxdJfQYPOM7
LNmRF0eBtydwhnyyjm3e/ub+BgtCDJctYZZLntmZLbFIfFDoTUELX2Yz8uwRvkKo
tZy2yd42PYbiP3ShlxmIWrYllzlnmFRq8ack/ooosUxwmu3QOAC1k7Uzn00qdC95
X0KZ5C2UMD90/+2v/bFohGg7FZH/kfJRuOJHgzG03dYS5fsr7sUQ2n4i8qmNwKf
cQhNIsaCEYrXQcIaUOUYjn34GN2UcStjCxEEN1N0LWvXc+ri72fTVfbO/oHEPdOLE
gJUNg+HrBGZvDdjLvXh8+XaGYXhWd8sJR3ZnIjmL1N5ExrUztL6lY8Pxxv0TnntP
AEXrJjxMX41WzZ4dGQiGko4GDmcQxz6XS9qRe6V06szDcD1WMO8K6XZYFSeogUvk
Frw5z1occx5dw1GNI81ju6EjlfzJKyyEVBKSGFKh+KoSP24u+rNDR5pTXvgrZcTc
8iBC1dbgQrOfppXVfV8/PSjEM12J3a5BFK1WtHHqF4uvhUaYSc8/i26bW2Oe78Co
bFqTaclus602iHkyd8a5rnA9TOzN/2lMh3KxtlDDglbIAPvrHrAfMEp4LtBQqD6p
ztbsFjaFJ38ErhFfyUNIFm8RpcLWFS51MTKHkhdq4hFgfYa3oD0QAHeTmtMydniy
sB1VaSFiihGpdz0Jc5DH3ctkW5z5PoKcJGO+zInaT4ZQbIxQeXfofn2wOD6bEbk/

REar3MAKFvpYGVHrtRLnVhBtvzF5YBl6DDm7CA5uwdOuUlq3WZixz5T1N1IxQEwc
9giATZqkns5KMzd6HUzCrSCxRLK5pyDI+0wDg1kNEl/Zj5esdBV70XtBE/PwxM5/
WQJEhHmlBDokH4wFQ2P/MUG017DEZju64u8ecXqMMYV2IdLZUp4YqoStSik8j7/6
hYBrI5LmC/Ix7h0UZzbJF68i7NgV15jrlraF12SEk/CCuAu28xtT8r74egOAwNbo
zi/FeWIvtXL3Yhf8JH/ixOq7VIDucmaeFNta67z3AZnLvpBluzevUU1n2/oHmgAD
c9nGegB6z5oqYxUqQuSQgMbwwtcYw9aT3vu9Kp+gkxqDPfeegVTFPWSodXD+WBWg
+wQD/alscbM9OET2jjYen1kbiwGbr1wYqPaLslhm/PaSDCE7bAVjNhtm3m/PeThT
C7OomaXsSiQGJYU3JcRGP1jHAA9WQMflsCimBFMfrv93VkJm0LdbeFeCunPeV/jA
Jmv10Cp0jBZMbfFrng3P+kCJgqVMO5tOZzc1vTFQu7FhgFOxAdC2S2RWyf4F7uYjD
SfIize9a56bglabgNitpEQDnLMDcPEdPXUNve3aWTZxm/b6GsqDjw3xdXF7fHwHy
0H1HB5iZnKrIWEKEQ39v7kDdLxKN1S2QjOq67dK0BsJlfsqeXndO+aiVfX+Ba8V9
79w6+pbA3icZxmE4NX7wwBDSH38ApMr1Xxyi5RNSCT7IYa4cLxmHVHyWwM8TcTA
N/vyBGrMGWZWavUUVdwk+LdU3PiuYoxR4KzegQan9N4FQk5UJt17hyVfL8RSocm
3gqxb6kp1TSLVVi6jEBiMvaV3i1l+2L0MgLSoyfm1WD3RYkvh5+IMLXsotqyHRVf
U4ba+gCxZl6vURbj13x14JM00isTCXBKp9INr3eu0Q0PQ6rNbqx7Hp8GjJx4sXJK
IgtRP7k1960vtSqMb8b8P3l/mwqvB78UlawDr7CPgxeEII5liB1zcXIULstXNjvK
X4P073MAonSLwx7mNY9xKDRuPtDWULdgi5pXgs25MY0ihsN6STfIOB+TTC1WLQvT
/5UUVL3MitLxttN2Xx2m13K1M+hmeOihrqBKzhgZIRrxMSde5auXU1RqlcN9VOBrI
kQDKJN7ep0p8012R8Yqa6jeOvohm+GU0V/GjCxoilt9oCfhkAAB4xPpFCYEtPGyf
9JAE/NoKotGE4LbzBvGERqBa058QXgQ0Bdt4tEVsZMdCdFWyBqjdic3smHV7TCNp
2UFw3fgFKGb1QetyuQkF1gdLCXf0U5P1KpA0G2jh7cerGQZsXZxnW47wf1Ndgw5s
9GR/NPdZgU0VZbJUN2mcFz4G9ZH529P6fDCpBdHNjytwEkk5PF5FGKiThyufN9d4
rnNnswfum0xd+iDDVcw62233XsiABn7cTdiNAMgVFka5nyjer5rahKb1LbpTfoc
M7Udiic+v6jCeKAZ0LLeFcDzup+MiVz42Ej7KELseu7DgSoz6H+D0irGKJYRFoy8
Kk005aNSKw4MzJFKnFH+k6jbr7e2QBR1Ez5vZi1s1l7VE8OfK/dig42iEe0QjCQ
a2cq32gUJk9vx1XiGKb1uXtnLrtgygNsmuTlwHaRZrJETIVUn/v+luj0Ork7eLSH
ROuUdaYravWkRYwMbVsxP/Nien3DXvzaxH0Yg7cdWaFP9RTXsIe2N3SO6TzKgKgP
cVZ3qwiFS6gt3o04tXqkZYmnj1kpoxHRYCj/dtBywX+0V0oZznm/Sib3ldnHBnGR
ucCCw37DDKxad8H5c2NSDOQ5s4s1Tuzaf/N1x4d1UoKzTCX5WecUJGIEAduYjdTm
ZBrkx+qPy6DvnzWVL8CaI3zfgBLoLuqPY5WRufCp9j9raLTg5XWFgabXFzQFR3Q
a61HhRCp/PihuQjnzB9ptTYaAT8Jd03rNDM8Dp7gHC/KFkbZLvnrhZUBLWuP/YPD
T1cKQst74Emxtqvkw81G3h/NZZ7POMRyL76Uq258RNkibjDhwGQKGWvHL/KhJXZq
70Z8bdceHcz3uFYbV5gfpAbYWRgYtctF6Yg/OeMQBI7g0XTLzn9famG80poiLGLV
pfWUsjkiX5xP6tz6zyvS4d4QpT9e5/fB/PCp2XHEwEuIZLQz2uiqwuwnDnOmi8G7
I5cxhgPBZA7v73VBmLP5oJ71P5SmOWfAPB5xPXwmDkxhpg51s4OxDOqvEakQTU20
udBZsy4GSJyusTkeEy+GqXCcspEuJ8nEcJ7Q1Ut11sShzfiVaXa12+U5CB3kPen
Tv44U5XkQpOB7Qny6VkmSy9C9FxSagQfsqhvS98xB+zZ+JFvSwpffQ/1Z1wCkCvS
FjkUBep2DtIQWBS0FW+UoQfo/hqYqEtYSyh+nmOJrozT1wfBdxLkSvH3QsC7p+Ia
OaPsIpTl+8fwngzxE4CBOLHEuyQt8BrUrb5mv1uTjAticxSe39A6sDqPK9HXjYbb
5eJfY2TT7PvH0S21hEdUK6KX2TPFgfam/KETn1wFZxFxf82jCd0PM5WQn+COYkFQ
KbQgsiyDhd6zqS4o3gOF9gFyRAA6TtaTygaR64kTFsqWWFDA+V21fz85U5Wy0KA4
/s5Q11MJfrYHWIn2MsBYMi52Ac9JqK3Fm3uVltxRwtnCmOZCuoJoGeP1VNUFA5/3
wK4Zs5XERUmVKEh1w8DMduuRbZfVvBmE4/8aCjDCVfbvxNz7s+Sm6mvTmDh3RYUF
ycMXmp47b078qgAj9hzCcYtJKzbYc0d60vLKjesGXycWY8irkjwzbdxVcPghoYGZ
xgverdc1W38h52/Cb9jXtYFek/6ZTkG4tmzJdwXjqcvMsoZnmpNIYVRRb5bTLmRL
JI3VBioAc8D5YsgaSmd97GnASRCaS2sR2zUfSElmvXiJr94LrcDyfk86P/aHN5Ly
9VhHlyhjtILy3Bot+uArWFjnIEJ7LxHd7DknIYQ8JWnxYQyEJ+4zpIkS8weBs9bP
BDxwfiN/gUVj+PbTueLVR8VgYzta/yc0PobG91iStSiQZdXoCzihjbctN7WbYb9a

```
70+E5GosuFO3VpWxchFXWSUziMnI3Rn9bjzK/xEHMgMe87ptvIp/J7dNwdHCYU2z
dOi3aTvuK+9EcqUK14k75wY+sysg/ljl+YrwZ6AFCOJ0q1R4Xpsu0GszFGAh/Pgc
HR9+sS2JY1U32Pw6b3c+6PMohOZzb0i80GUOphN0SDH+bbKWejwca7Tqee6oKHRC
w/zoutXWDDK8Wmd1JTScfF/z0DjHa771J+7ypwu+JcDhAhjqWWMYJ8G89fq9CkIL
v53RWDv4IhiylEv0KDaVOKDVJ8OpOIc0I7SCiZDcn5c=
```

B.3.7. S/MIME Encrypted and Signed Reply Over a Simple Message, Wrapped Message With hcp_minimal

This is a encrypted and signed S/MIME message using PKCS#7 envelopedData around signedData. The payload is a text/plain message. It uses the Wrapped Message header protection scheme with the hcp_minimal Header Confidentiality Policy.

It has the following structure:

```
application/pkcs7-mime [smime.p7m] 7800 bytes
  (decrypts to)
  application/pkcs7-mime [smime.p7m] 4770 bytes
    (unwraps to)
    message/rfc822 inline 920 bytes
      text/plain 327 bytes
```

Its contents are:

```
Content-Transfer-Encoding: base64
Content-Type: application/pkcs7-mime; name="smime.p7m";
  smime-type="enveloped-data"
Subject: [...]
Message-ID: <smime-enc-signed-wrapped-minimal-reply@lhp.example>
From: Alice <alice@smime.example>
To: Bob <bob@smime.example>
Date: Sat, 20 Feb 2021 10:14:02 -0500
User-Agent: Sample MUA Version 1.0
In-Reply-To: <smime-enc-signed-wrapped-minimal@lhp.example>
References: <smime-enc-signed-wrapped-minimal@lhp.example>
```

```
MIIfAYJKoZiHvcNAQcDoIIWbTCCFmkCAQAxggMQMIIBhAIBADBsMFUxDALBgNV
BAoTBELFVEYxETAPBgNVBAsTCExBTVBTIFdHMTEwLWYDVQQDEyhTYW1wbGUgTEFN
UFMgU1NBIENlcnRpZmljYXRpb24gQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIeh7j00
Boq0MA0GCSqGSIB3DQEBAQUABIIBADEgUQKlrg2+/XSX0UPb/Ei3BGHV9bPdcd6
9Kb5AzgrFjXl62h75z9kr7n7laGQIEvqVHr/93cOMCfOrmF4Q1jiQC2HazguNuQW
x9frIxOQqKnSf6rkeN/HeMp/z+ySEn2rAD/zJxQkqcX6vOLCR102whuQzkCkWSun
vgWYeyOHcf7tbf3u/FTAZkBW4lfpA6vBgNXG9ntspArT10IyI8sworBZho3nldHi
Y7A/02cARB7jVoueV8YhcAs4QPGxNtpseWHfQn1ISTT+SYc+sBmmdznvWo3w9a+0
HrXHwYaayfJ9iH9gFLeiBGNC6yahQXMbgzxXHfFw6w10LvGe2NQwggGEAgEAMGww
VTENMASGA1UEChMESUVURjERMA8GA1UECxMITEFNUFUMgV0cxMTAvBgNVBAMTKFNh
bXBsZSBMQUU1QyBSU0EgQ2VydGlmaWNhdGlvbiBBdXRob3JpdHkCEzB8R0APhiY6
```

HGLS64MvlsDXhpQwDQYJKoZlIhvcNAQEBBQAEggEAjtCbyUK4xtTg8t0Bw1Ll6j24
DjRCQoOLLhszshjDrp9pnOh6s1QJv8VbzDevogdYjAqPWdrDmk1tuWch2OBIVjtv
rUEXGs9+sBmZglM+6JKfXsvwXM38Yf8i4RRapMT1V8yY7j7QJCXZNh692flbhxUx
yaxznpBTqRwT4x53QrceqgkW5YWpDvAd0PjUT1PHJ1+4ydqKvVxccndbagHi2Kr3N
Eg7zWLgJJS/Qdmo7J/ABG2iMGJy7BkfSI3Lb3sXtvzo34W92xyrQ15djXr4sdgn
6pAnDdadewJPjkKOCJyEMVdAIU9Esrr24u+3+M/JmBwK7n6GWJdZ24BU9OnIXzCC
E04GCSqGSIb3DQEhATAdbG1ghkgBZQMEAsx8ZPSgJzn8vj4hZoLZmuAghMg
iS17y/PtsB4ir0/csayKDXSY+QJi2gtR2Pj1BXvcd5798sNFVd8v1gAXrRD7gEiG
rc7epVre+xFxcPxpPmve1rINh7rKqqewi9tkfjHDs1SuuMdpk8fxrzmmfBRP3Gf
YaX68MIhEUPoP3IyaKSeGxmttqimF9r/5px/QHMu2F1jqMR2vTJvUs9Rdjg5C2Mf
CGMf7Vq+nr3sfMAZCLHRJV7DpakDc1luYHGAQR2v4Hy1eKpN17MDWQtAB7/9+e7o
HHw/wlfeulbduI9yZbQrHhVzRGzdVYS1OcQfqtE4QP+wTr//Zos0046bjxqFH6qx
Sy9WZmLI65f2KJntJ8WLH+6Fwh0q0+n9vXSJZRKPIRE9Im3m+WYJgE21ON1aAr4t
Xh3Vetqpf2RpxduF+h3Rwu9eNGI2WK+7/mkXiZaCgo7MGh3xZo8vPW5RHOp0+qo
FQRFY133SI1EhfgUJy1qZITam1C5ofheuGFaZusbIiqiwJzHAjGuXjEPNw19ATas
Fwx1syLlYybYnxexaPQ47mo+oHY22+5uy7o9/w9V+TmJeTzPoG+zjep3Ly11TJIG
zVGFDT1boc2XbF4i3KtuouPppZ0Jwc9vL5VW6BWKDMpelhdFTSuEJb/OPF/9cCNF
m4ie91Ke26vk85D11rFKAiDXqTeoybboIQkxqJ9VX+9d+zKRIwZr6nDAFuiCYZFS
LSYZahUU7ynGzi9NCK0Hrun33G2dPkMsCWAhdI8/EbFZQW4r0UJup3/DfIBpS2rV
Z+aZuBHUDR/VsdlJ/rWF3MvpfNBxPfkbtS7Vir3innwzEpw7LrVLU69pQQCq5mz
VICUCuHYi+S6x7fcFQqDK059D1L4k1j1bRiBdzWAE4iEFMjX3w1v9Gy5TGbWCXoX
JV8OQsyVKq4renCtVCZz8+EsmuNsND4sQu7aiy1nBa0RCjerYltoXQI7Mb7Q+JKF
tQbx140lx+C1Y28HcUyLr4a15o3fc7Em2Ymvq8rfrGiI8RPAC+ILPSY5BjD/wUEn
aUDG9H2IRvujr1jKScWRhSFF7kC4cZTIVf2pSSb9HounYaL06fhn+ORkxqWsh/uw
oeC8WAfRb0wHciuPzXTCDxc1Nse2Bb0Ora+Py9fQfGd8JmXj3JvBLVOb2rgtawk
z5j09zUqpXNudjsni0JgF9+gSIX0Bauh2Lvh9y/gQoEegrp1xk9JYSMQRFLLPnfd
jnp8V19NiAalG+Nb2JSMxtmiqCKOQn1Oyi+zOXpPt1TVOpfF9z1dyIrJ9V71/Fry
xdak6KdpWa8OhZ1TCvFHg9qjRvqkMu5tsLo152pFxcFa5SVogNgTyiAtlKae8Ndx
+2AMb2tEHmOEzKXsf4F92q3qI66KdMaUQjuwX7PjR7VvVbv6lC1NM5ipgQd9OuMw
eXeqwR8I52bpsdwWPcChf4Yz+hcgGil9n0XpTbHEjSewRQ9Sbpmgv7eSDpLSdStD
YKvIx1Q2ryp74wCMJvN4XfYpdi2wbRM7Gwth/65UADR2YFr4SvghWbMjRZoQTGM
r1ldQv1qL5GPz1XV+xegItPjCWNiy1JmCK1/YEnntd9ZJTJpJ4PE0f6yWwLwm0S0
yEWkZe/GluJp6G53HGJomLlJPYNGv85+wEmHkHF0au6K2LmNtdk1D8b2PoiCI75w
mlBF0CjknNt6ThDsNvr7nc1UW7HOnCn5WmI02MTzEVPUCuZguB+txXNTtWuf6wRP
eomizA+yRjQDjqBAVIEHJ103iTcmhrcrcuBYFX6zA120KHVsVadLS6KGcAxU9iYec
3TYoapsk/UQfBaJAABRL+JEroUv1n4rUFkCjAoxHIgtwdeU1HvzGPAQPA9nZOKeW
DIEyqfSvMiuiQ21BO6jncosYYMFAMQShPAposPh+sSOEsm4qdyiKj57aukzHRwK3
Rvk0HKAPc86zgjVxycwmHqfZJQeQ+Wtn/F57FB4BgcGDG1j1kPBZjKSG6LwuWOG
wIb+INJGPjtGupHsxniigLnF3mpjS6TgRgXKxzXQExhdJP4LAWfGtY11k1b39Q/4
V6vDp3orU6LBaqPCKfSzm2RH3rFk9uWoKpfcE2gyTQn+Z/jNmjh1XXQ3AXi6205N
9GCa+cLN/Wzb4OyL1UgburhpV98dWoNFxwAUsRQDYk1W73E1+7BG43xj2BR2Joiy
WI5OqND9q5Ar4NdlqSNXKimbPKUs7rsmkX/4ZhGj6q7f2Ab3IOw1ppriTiLNfuKh
+0/pEL7ylcCYpfoweDGOHYkQyH5I//E2tZi0IPVVsB2XZYf89/Kx9kcpV3GTjOtq
jDyG8cYBUIBEhk5iXvzN9qPQtKcYc21phk52AG2fKX4LJRc008i6Bt0AuDCOY1yo
CmCou2RNUM6CeAvD1ByfJF10t5z1Z5vILqNwM5P0ceRYvr75IYOsliwD9niVw4vz
nrZF3p3st1AAqn0xU3+DGGkSCTG++Y6i+tUI+XG663dDgebHfG4hQO0uRjfwHkk1
mgxwU1LvaKSnlu8RPPynmyBopwJzp42hlDsK/353KzW/fbNHBEVKdAdB6BWiqsSR
mynNSEB40BiEt1Hz+emLHAbXA2dQR0VzMErSu+pL7qGAMI0uYV0yGMDfWnkNz9F1

IMMzfn56MDXbvLupC4qOYV80o3JmFt9HP1Aym3gEcoTBqU9pywqtJ/ZKLQniP35o
EGr83kjqrZEWG9tkPHZ/goYnv7jkPny/Xl3VTzoeO+OYlWjFTLkVCeaZYZRjymO
cRbwkDqoEceY3r+EzOu2EOczNwOpMmNGwEwTKjYQ3kCdfZOhOne/s5e07vwfttCv
/Isj2aruUyik5t3pVverelLUjri/f7I3t6lyBvIZ097k8oRfrzxtxGJCrfIL93hG
xMwwYmBLEpZSZJITPrjijQ3JeHwn1VjN8OY0hRvhCJDLK99ZjVEHY0qj2HHPQ3lX
+xlyJM7Jyz3Npo73kfpBAjVxNRvXlDnMwz/ArIxdr04NDncJlKrKAFDGl2vhXvKp
Iyg9tTWHs6pNeeZg7cNTdtHLIOTrCnlxYSp+A8iQGB2D8I8fVOX7XkqEO+WikEvs
LXiTDGI59+tqZOxeP+/i747NjJdkPfxC+1tDXxADkFBcm6+ANoo1o3DrmBCmn9O1
ckf6Mz0uzjT8l8E45Gfn8UBTqB+bzjqPbzfk/Adl9P9RhpK0j5mcUtZ9qSPglTY
wrwDro9fBLq0ZlfhJnHx3DGV7SZaMP6Uo60u2MO6NDVsnQXo+ocpU45CDsxgMvH7
elmYKhs1SDSGRKAUvXv1VRtLB48q14Dgy93ElRmr491BmQTcgJlrlxJVmA/knZ57
4qY/jGQUarug3lCFcbiiEJjWmNoHO6Pe3JeYq5snveiH3tUwuHiJQ7awt8KJGQGP
NrwldQuk2jhrYSsK9CTQJQ+7/pf9DP3EuB5S5lPLs6sgQ9ycVdJyZDkbUYXvQ9r1
IjiiLKNMar/QswzUQSRGJmzrEPKP01UFJY5YTVXunr1HaxQ9sFrt0VDbcTJL5Kty
Y646gRnQbKXXMdS6EdLmvpCAS7idGSa jo95kUHWLML19YhOI2Nf5i1dNxx4EwIoQN
2dPTPDuBO+D528sXnKs8COg7Q/g8Jzr72dBWxk5SxcG4L8E+aX/XBIKe1eUB66mR
bfrmdAuYy75WkrZjA88bzYg9hmVsJ8C5O37y6vSBOPpQfeus+IjiiL5N37DjkUv+a
Jbm1/hotI4RNSAUmt rqqbI/Jklf4wwk4/dH9+Tz0gfI7Y4UqExlV49zuAtSROv2t
GjyuNonI1pCzhvD5sIwfbUM3NxCNVwgQ/sHBnd42HDOLDlJwHrDTUH3pmIX1XYOt
+HDxecBZB5r+vGmbpt4656gFqdmMikyJSNLfn/KSg+SccTFDoPVs2p7lONDDFXVD
Nc2QR24d3Yn7oXTBXso/K4f0sFI3L+G3DVhmf4DyvkNERfYw5RBPgzq1SXqFSOCA
ZCVIhoWYU5WtVKPp+tX2uy9Jq3Sv8XvV+ABvcimu9K105kcMMsB6EvACBj3yZKE
I3HyVjL/xsrBrMXhqH0lizt8XV87B0vzvGOACMrCzKamWLQgMjnAYPuS9In886N
c/AtcB6sd9MsIc7+eBx0ZrL0VLqc/OVSDmx2xZIHlRpm4xRKLkDpnQOBz7eeVtXd
a0nqWLTlObFj79cqK3qzRTxBmsyTlU+EJpRhkRsTw2z3aGsTBgsORRORRHnXoXr+
zMNlRpxeFl1xFCEncKYlESBXh809hpNvaucqQEe5cSGAzxVBY5kJG12NGtQ05emc
JopRgzkk6NGhMomod7nSbc/Xm59uglu4kYFagCmcdx77hkQUYENdvc6VqzFuGSH+s
9VNLqk6XHsF7JMD5zX3lNIJEGL/1J6Xje0wHU3503wxWpi3eQDvqMe6liwJmMWZI
AkeoRYTe/AbWI8v6oRDOryP70UL4oVbUj9u2XOZC3ileXlTJO8WNudDYCFHhJxq2
9d8xiN0uKrSetmkXSIsWlkCK9WXIpu1XiWlkfalG7lQPe7UzFuAMf5NfPValfiha
pFFKZF9+8FiV0ITWlwl7zRHppxxDlaAk3RAM/PtOJL879ZVEwMC/ojqcfMGHlHtG
NzpGDh8/IJWfK4EP8TGJ+BRcg1402cEZNUGL1NINkPTT3+gPb9xqKa14vmyPxMxV
QihLWp46rN+h09pWdfBUTcCu2i9pPduVaFQlBPPhQnQqpldYGP3doV/0dAHijPMin
6z1Z4pH7rJ5lvNhbwsF6FgQJcX8nvl1CVDHgaEKSnzffaLbNFe/Ino2Kcn8FyxQ9
bg1q52Tc/fg9OSqL7w0mtsCY1lXyP6Pe5JM84ZS05qbodmBiFJEuhXBWwbG70cGs
YQXbas3elkZU+wXkiAhMZ6CE8tWKg4jg7cK0pFEquFdJywhyvcIB3ZcpF6YoYVMk
8rbp3tFrSPiZRysvYBaVWMwvmtfh3Hm54j5l2HtQESa/1D5Qj0I1W58kCvYKbOB
wHdchLAjfcuSyf3kRt f9fS/3YX4SyWKzqhw8Obwkh3bL76dI5AebhR4HWQCvW6r8
tggt4/qewNm2fg3MeKQ+Cf9AG6MWhu5NpZ7RPjIE9Vo+5NUwulIh1bFOnrbMTKWD
dW0PveLdWdOVNVT6hnFTYYvmsmrhaDoVa0+Li8RuLhYsgVGahqOPxnmukSTTHcuR
uz5apKQHlgdcNZtNzynv+ruAVoSsf/b+TXGoLQ8ylbEY9tNki62w1+ZgZmUSbMxZ
reoiBSlXiZLvNtoBpsbrB3hvp0v1+IzldXuEwOjwvEwwfq8+az/g1VA2iQRcdFzW
uedZed6vGX9q89IBerou2y2Z7a+f2tILq6vUSWKR3ThY3dPBDlCmVgcBvqC7u9l
PMXh1SG7eGSLXvNabDwZ3QU0Ztruzefin7488j0qyv2Y8e8AjbXspxl1Pgjn2d
sTDTlm9TQ0N1Q2Z7JwmT/v5cVQeRqmmXHDYk6U56I2JRdLHavyNJe8G0pPmQX9f9
YeSL+2Zxfx+VJ7N4ia6xv8HOfMxhJxRVOcHEaAGBS8wSaWniyZTMq/CdD2/gLhI8
WF3HsSrZvJL4WzjrarXOGWrZEgn2H2y0mK3b52Flpvunm+TACpIhzfP6MkdvFLJS
prCQH0fplNH/taeEMpcehv5qd+VlQHdAtx0Rt0Vx+j+gVYwtlA/bG8LitVDUX9kYr

```

ngwyUZS1wKDz95Dz2I6KufzIftSxSJPWl2IoegVu7Eb7A/xWWdDojUv2cS/QxHiP
NBp1M6VCUQ9rXnhuM6wZQnUFboecqtxSBNmLgN9443vnRw+9xOUbdCQPVDGS3MB
2t4X+TLBfJPadxtnD9YN+xpF2UZzbhTLBfw2gIlz5eg939BJ43WATFsrBxmvhVNm
+5HB5zKZDqdydAy4fiGeF+xmQ83xA6x+bYBzdEYqDcNMgIjkoG5fit1dVkykgt1s
Iy58ittUjba9wxVJVSazh/HTYpJ5qMLAFsq8zdcV9xVsB0SVuMRs4TtThScej2lC
rb89/BFQX/BHcvTEWGsUqjH1rjGxteE0kUPpbCWW4bFyY53ayBT/0p66TA36DNTB
ddfbL26ptulZxKU4Gdlk1wR+GTaITVqEu7C+ZJWGUrf3BZyOqVJChr2ZwyKqUK9M
8wrvDU4eoDVqzT1z5Ttj/g0SGX8LjFv+Qznd3xt518MWiuguL/1FSSZeZPNGhFPJ
nISe8wWDh9MLBcV8xy3ZHAVLj11+cYvIHhn85T7To058X8YFL6ki7k5UPm4PYQsF
HuSEWnQ2KZLPVUJw0ckbZAYWgzkwoR0SltIeaGvJ7nM/10WpLdxGQr3tnHk8e7PR
r4rsLVQJvEfj8FLgki651UrcnKTEPt1TChLZbhegBSSGkgOokLpDsR99hGdQtQT
TCet+30l00tq5uCRkncOGVDbrJO2yqONU4Sq0oksMt6ZQIEZM8150kh+bVxu/ixt
86+BxkTFfKo+yOL5/K0Qo0J2WK1ADN16IKZbrr3kQFuVOnHmKcZrt/kwt108iFj

```

B.3.8. S/MIME Encrypted and Signed Reply Over a Simple Message, Injected Headers With hcp_minimal

This is a encrypted and signed S/MIME message using PKCS#7 envelopedData around signedData. The payload is a text/plain message. It uses the Injected Headers header protection scheme with the hcp_minimal Header Confidentiality Policy.

It has the following structure:

```

application/pkcs7-mime [smime.p7m] 7695 bytes
  (decrypts to)
  application/pkcs7-mime [smime.p7m] 4692 bytes
    (unwraps to)
    text/plain 339 bytes

```

Its contents are:

```

Content-Transfer-Encoding: base64
Content-Type: application/pkcs7-mime; name="smime.p7m";
  smime-type="enveloped-data"
Subject: [...]
Message-ID:
  <smime-enc-signed-injected-minimal-reply@lhp.example>
From: Alice <alice@smime.example>
To: Bob <bob@smime.example>
Date: Sat, 20 Feb 2021 10:15:02 -0500
User-Agent: Sample MUA Version 1.0
In-Reply-To: <smime-enc-signed-injected-minimal@lhp.example>
References: <smime-enc-signed-injected-minimal@lhp.example>

```

```

MIIWLAYJKoZIhvcNAQcDoIIWHTCCFhkCAQAxggMQMIIBhAIBADBBSMFUxDTALBgNV
BAoTBELFVEYxETAPBgNVBAsTCEExBTBVTIFdHMTEwLWYDQVQDEyhTYW1wbGUgTEFN
UFMgU1NBIENlcnRpZmljYXRpb24gQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIeh7j00

```


1E1VP/Eg4PFpFmMkOl62rPNzXQnm2iEixa7S2Rbzpcj0Lgu/h3PCccZnw9G12k4c
DJoWmPdaOvOODW845ophWQCWNCDoeY9KJyJTz/vqC3Gyf0EYXH2SGNhL3tpZtgnO
O1LfQJ2gu4dzBAMMGfXvfmza1se1xE+uhBeP+Fjpcfq7PNp4rc7fJu5JoVBCMI0
EkchC9Q5fRNnyCwunYFGd6N71sVtdDHDLSykeEzSoGH32ZzbjkUXKyMkEcm5DDx
k1FQSusYCMdFhS09n1+Q+A7gjj3Nxs1rEPVrdkKW01aUgg4OxFuN4nV77NBE28qv7
hJ0dl0jvZes+tggl8nXgtqJ2cWaM3cspKT78fpwnqbg3rGkgQrgcpuU1VXO+sEk5
CDEQ9RAsCLW+A5VRXHMnggzobOmVnXAZLQ+M40LnyQTxn80NvFr5hC0uthnRAF4a
1Fu1CIaw2MMcrPHPRXR776hQGmMk11+1Qbr/XFG+D40vAVWu1OLMw9vccahQqBjY
G0Hv6whQPJEx66ubMBa8uRNdCTOJ9dJ1xYd/ETrswLw2OULJYtZtek8gWwQXgFNn
X4WnSQSCbhN4hbaCmcnmXiCxQVHNruc5cR2YzGQkgSD9u0CPiVMHHVcJrXFjBKM+
//OmFwCteJaVwJS0fVZb+BeHibR48NZmAL1614z8vGGAX7MTvtWd2KQSnKkDz7f6
/ktj8R1p7qLOMaGgUTX6zjTEY4mY/SkCuWeH3wrHHcvE5RBz9PbPU8QySOBEZTrN
oCwBAivsGUEBORbjLWuXoB0bx4Yzx0vRf69Aysweg75gAni6UXBOzp2hXMPZiCxS
1JhNiWJrGwY/q8Z6ATTMODnFhKbN1JiwhKveTni9Dfsje6z4C1QR9p1fqwb4qGpw
m6tVhn2G4cbOUThfELe/o2hv0WXqMj5ev7D48QznR17Kp0tHvQqMYZ27n+e/haui
405F5HBuc8HCW/VwPRtprxK1ACi7jyfsQP9iQ/XOKyz0JpiyFzJmSLlmFm3q6a7
JXkTdUPOsyihmaOQMZUaggBSX91HMjL1i7A8mCEK+wIEzLbQmsoH1aJ8SANoP268
6j8eCT+/DAXWWSGnqIsfB7c97m3zkDZIFR66KUsvoebVWgVIuQSVDe5o+Oq16006
3zBlxqC8z7LFmrX3P/IItA7R1DYMDaZdVh6Vgpg1epfHDzy9hdvGV6Jzc6vAi8m
TPS5xRdipf0OqwiHo9ohbOB6bFDCF9pKBHxzZkg2C4Ncjewa2wu/Kd2Y1DhuVy2M
6xz8KrTPGd9TEBHL4Vus07xYgsdCIkdWUrHSAu0MdJAP42502bILxq20FVLMjFDU
/71qHRYZl19Q7yv63A+91Sqnrdrb9MLzqX4cCcQryi0GKzKx2d2IZacSUViUoP09u
ngg4T8DvUz51lGL1kbPSPnZJY2LEkUjemb9S2zqGJmcguAqc91t2BAKZIoENUX66x
IJpr8RprrolgomTGBAbX0rAqX1vyGp4T2iStwnNEtHmocetfGN5IdtmCEY3Xv+5a
YJvFq4q49Nagz1mLXpskg2krz64Y5k/z7cYnsnsgWlLec9hcvSEyhF3wnt0j2ABe
TK6dDOIcvy2JtucgyModsFTQSAxOvd0hmKG2/0zn/08j1d14yBZ16osCUzZTaH6t
IYCAuPi8HfiYa9UbmX4V9zoMN9c1kUqcwvFnu/6mUsMNJjvNukgH2bXTteckFM3S
IfDi9yr3WohnQzt1vITL8c1g9iRxn1Avwh4C3X/CTpCnTawTTQ1D7ZWIJm7slgOy
m0dk0coKGO87sYf0BECv4I705iyV20ILpsFC28RsFBJY/cxXFOCX5siu3HM9E5Z4
H+FaZJ5ToyAwhjvY9FWv4Ti6RSxz5OEDcQ3KJnNIynHKWihSg2Q7YpCXP1H1NgS8
T58rUJyJd0ny1RUDrxDOcNCx9KCsZS7K9k8O9BtPax6rUC1qnPEX00sKeNUzpbH9
vJhBq9ROFuVTACgHPJ9g8vFOAkDubhtKfUGHTFPkaGvS1V9ZrQ7j1jS6MT+Q+jQO
DBjddj0VGTbdRxdkeK69fuUTP7rnnqfE41TzLCSFi5krqDAT6rJxKy77LwKi+qEZ
o8YuPHciXH/gIoGnGgcOlKoEXMILHxWDFuuKNU771gvbbDoUqrRqsxUTxKeuSvHw
Cc9cIvsoBHS1pK+wxmIOEBBSdfdeyvh8dpAtmrQHMH20aYmc456+H+2TCTBpfcg
g509ov7/W26AyC/OP7nIYV9Ar7sHgS6s78jHnfwv7weH9FB4iXXgoTkm5dT/vjsR
uqgRxgFm84cAXmxgOcr4UrafMV5+PAXCzrZY+0xtCFDOr//Y/k67qTPZc0pm05jE
Ix1PjxTkWvXe3oz3bOspcHjQwrIF0UpeQ7WL/uQskIzHkwkcu0zHnTKkZCQke80w
xczH/bjD27nHOFzUWZkeUwjNd2MF7VXKwQtAPgj0T0f9TxGiyNQgKT1IdvSRS+s9
iiffpaOtdS1MiOiLRDL4CzQDy7Bz50DwzhrA1xJ65SIYL43R1vk4QIkSP5n9KkbV
/AgJahlpkEdfq1hSa0i2BQW3VMYHsaLbnEtgcrnmNKcDDBS6XmM/KBuS/C1EsUBi
4k9+KQzY1CJcQH1Wy4fuz2su3P5uiHmbK2pm7td3GxAeqkzsqKFYgdCRMSLSOMLb
jDUBmKWUOE8oqjilaswkk3DBxAKGh+uFNMsEGjK5uWGuJ5GzUZ480PBiyng0WdC0
VgihPWbHWDqvZcCspn13ctcLeQNfnk1JbWdyYmVh5sIeYCjD6c8FZhgtAK37g8qV
yWmXUVrflTnHMDVect+w1aJoAkCvDUcIjvqI/82xaC6uQHkixVsKu+etn7/FChpW
02+7TNMRKypX2uzpoXe7ac5mGaf63tUiRyMSSKb01KRn/3yHCY4seFso3t+Qoo2w
830YlB5Zxhfb/Y5n3NQGVwWDjgyAmm9gNy0EJHDVKyxT9OH/1eNVOQSJ91pUSiw6
DCKnvxqQ27LBB8DEBC2jIZNc5Hc+ZWSHR38WCDj5EheuHZk1kbrkqWwGhzBfr2+F

```

qQgLn9l7zVPX+UgQfntjz9Ob7SNGx+LJevZqEXLlk2kCmGy8lOdlwyaIOXMFcWlu
d8xX3Yn4WL3rHiLHk2TvJ5cd4vtmjf+hymGlgUs+dX6HOapOyxUcS/Uy4CmabJ/O
GlsWS2A1RBR6ZqloqmWrHPPrZl7ueDHLJMFh4EW0of5/hALa+8oz4JqvqQVhxaIQZ
f2/NanRIIbg/Gk8mS+XhmojHvBVWovqFxDj7pXKr5/WQnDFdp4Dn/cKGeO/uwwhL
TKBwaGuxOf1+Wt1rliL6lccrFd5ig/WBCGUKHTOy5kXzNHZjf5LRj9V+R5A jWy1t
FJDar2UKU/zYl3BKmesrL3CIqMfEiM6DBvj0vyI4E2eWceH6VCQGCEleHCGR7WO5
SluhPIAvBbSFra/LCSqirWwh+NYrWq29672fA00zm7so6xAIS0zPJquC/wI3VFM8
T19KG7zDj+O6iiY/kNyLqHLdGRcCerXNreYF5ECVDPvv24wDNYNEdHz5VViQp4p9
1RT5fozXiecbkaLZUAJFZlXMHuU6xjFwsCKvnYlVNUvePDXsiYE0WXGj2EwTXRcN
zUvFNX0a8nB4bEwiQ/YfTKXD0ddCNX5jwEhDdf2fe4cyvmuUJfXC+F8ZdydupSrH
Qu/0XTCLEA+iJEDmc/7GXAQ3+P4lVn4RvdbwnO6Kn8aUPge5yzSk/XNjQ3G/eHP3
twEYCIhcWH1TWHx+yU5l292CCb6nBvO+mNNlTmTNEwmYMJPttkVAmMRIoxcCOOK3
tdQtdnVty8ffhA15B06PwNuQ+EUSbvZxLZXrbDA9X2RMgfUqEJfyIWTIa9M57rsD
83EVdafKSbP++/EpKMImsvPVGmawSSxYOR6Xbz80ER0OvghegfR6Q6dv5NT9r8CW
zmFtg0kmjYfcUR8/mt+EIF02524dzqprmi/sfIW80fOH6AJwSOGqFxxzUmlKoLkXc
bEr0mv5Sr89W1FdRxsH3zSLnPHacHx4GYO0tNh71eeu28Z6VeJdLIVOf2wy0Mu2e
DsJxExn8Jsp4SKVY6USRe8mWcr1HADibmFNjvv97DA9+3sRp20x1rk/FGL504nvL
ArvivC1f0t3LkTDhnXI+/Ae2jOdIolpJjNMOU9XXVnzs2A6v+Zke0ZfsS/SoPq+v
vME37CehB9IHyjYq7pikz7vLFdRn7JyIbPqExItB861lsXkKvJPsmekJE6kzvJD
KWZrv4qEgfgQMjHavYX2TQ==

```

B.3.9. S/MIME Encrypted and Signed Reply Over a Simple Message, Injected Headers With hcp_minimal (+ Legacy Display)

This is an encrypted and signed S/MIME message using PKCS#7 envelopedData around signedData. The payload is a text/plain message. It uses the Injected Headers header protection scheme with the hcp_minimal Header Confidentiality Policy with a "Legacy Display" part.

It has the following structure:

```

application/pkcs7-mime [smime.p7m] 7975 bytes
  (decrypts to)
application/pkcs7-mime [smime.p7m] 4898 bytes
  (unwraps to)
text/plain 435 bytes

```

Its contents are:

```

Content-Transfer-Encoding: base64
Content-Type: application/pkcs7-mime; name="smime.p7m";
  smime-type="enveloped-data"
Subject: [...]
Message-ID:
  <smime-enc-signed-injected-minimal-legacy-reply@lhp.example>
From: Alice <alice@smime.example>
To: Bob <bob@smime.example>

```

Date: Sat, 20 Feb 2021 10:16:02 -0500

User-Agent: Sample MUA Version 1.0

In-Reply-To:

<smime-enc-signed-injected-minimal-legacy@lhp.example>

References:

<smime-enc-signed-injected-minimal-legacy@lhp.example>

MIIW/AYJKoZiHvcNAQcDoIIW7TCCFukCAQAxggMQMIIBhAIBADBsMFUxDTALBgNV
BAoTBELFVEYxETAPBgNVBAsTCEExBTvBTIFdHMTEwLwYDVQQDEyhTYWlwbGUGgTEFN
UFMgU1NBIENlcnRpZmljYXRpb24gQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIEh7j00
Boq0MA0GCSqGSIb3DQEBAQUABIIBAJDxg4GjNIaOr9Kf4xVYzLZ9okfUMbBaiZn
ecCbmpTZbaEOu7Lsxuw+Mi rounSBPZIEG3keg/u00HHo9r+kHDt2wq97StpAQRTe
Hb9sdS0xHiGYiH2vpgtIInNztCQQduzOHBzbGtQWAlKG+DoaGp3jzqLp3yaP+o4f
BxcCLcNjLxn7I+H04wSWHE9jQpaguk/2SiGzUZxr+KMP+0HFuYT41+72cOVcAAXY
p73P8kiMMj27mf28SB3naBDB75+fwsgtcrfqOPHCIXwyKnGpJ6vmKvFvEzAP9kM
oFQGsI7dBTzi+MQBtg6EfxgHhJfGtchFE25F1AJJj3o9SbGVEV0wggGEAgEAMGww
VTENMASGA1UEChMESUVURjERMA8GA1UECXMITEFNUFMgV0cxMTAvBgNVBAMTKFNh
bXBsZSBMQUU1QUYBSU0EgQ2VydG1maWNhdG1vbiBBdXRob3JpdHkCEzB8R0APhiY6
HGLS64MvlsDXhpQwDQYJKoZiHvcNAQEBAEggEAVQdgmLjOaxQWmpnLHXA3Y2Zk
ZWxNCpmIwmiVM5jvuIjRsU07QcEkLYXVM1Jx6UbJ5A5o1BUM9719poHGSPTP+bv+
E3U4Nx1u3D7tgJ6hyZNhn2mGfZmrHahQ3ZzVazhBOPxjIyXo8NmxHIOql8I+1loG
WZIZ41ICZl/nR3Wb+2t8WGWOwPbhqN5GJdngzvYcRzna36ug4UV+cdp23qceR33Z
nD11PDV0Ss1cGjTH8qpL/45/wOjuLWb+8dOnsQZww1PiIA4XxJgsIjcwD+/Z6g4v
ql9le8oFFZxa6QwoZKrX9x2mbzkZoIugF6sL2TQS87WiDd2SElT8xaqfgYhLDTC
E84GCSqGSIb3DQEHATAdBglghkgBZQMEAAIEEGR32whnqKtvXU4g4YzKmm+AghOg
2lFfb921csggkptJ/+rpubdQoPqjuugHIPlnXT85kvzfdldnun9BqrSYaT7KGeVqT
+h+/4hHCwH0HVE3d/cnxFzXrr5a4KoQ98mwnSeDgPcRXUg/AX/ujo6ISFgLPMAbl
XX30BDL3B5CamNf99TxPgTl74qeHHXpW32j3whi4kZ3069YvwItHKFdfpgwa+gg7
/gQJeJUJ5PXF+RLOCA38aA1ttNNj8VJ1shQTarg5EcmTABp56sq7xtFFnBnTab9P
ZEAvaUwYUyJaJkmsydsMfNLHFrtOhs17KC/VIgRP9OB4jiYs6FGUsxJJFUSXLH7U
H1DM2i/L/AXCVKdVl9UKdfnroVfnYUN2B6IHplttX5McGsc88N1/rms5T8Meu5t4
JNBrl0MRzGAzdT4RcsJehRBHZHcBdVFM+ia2LQWNHqCAGjCqMxw48Yh4YwPFf+jL
oOkQO/iOdju2oUkgbHORifXs9NDSPA4Pt378dD/8UDbyYnNystzbf4w8dCyP6Bzz
2tFeRao9Pmj58lIBvsD7KMHSeoQWVZSv9cz12tZ3S/44BLL7J/o3vQFfSCpsImsP
Lxf8pGzBlyoxTGlVlRucmIX8WqwjYx0ks1rTCLs8hd82kSTD42D4MCOC9Q5FP1lB
t2P/mwryGVBH8nrd3AKqoGV0fs66nKow7PptKKF1rZt+6/GUe6w9tSmGY78Ttedh
/NwhsA2gKoxMRefM84UTbV5bVdf3xEeS5spdUU/tgJkGULHutCJTuf3e/y7bXaeo
79y7TmuAImqltF3q/Ca/RCa+Dt8fjqNoeFW8PjB2e7+JniDtCzRFIHBtgc1eEGh7
IGGX17p7C31HY2uc5dfZGiMgIeh1lnbMUELM1FXa7poslg4lxhv1Zjp4D2ik4YvA
VMITUelZwn51gIs/ehhyfBKmSFML/X4Q9ORFUcogCi3kDjd5J5D1SMZGzLIuWLGy
tUuLHOXGdiqxIoMoe+aANm27mcmHAJNN861Keg6Uee4pAIQpOUIM7NL/qH8tZnbn
lrZfXYTKcot6xU0bDaW5pnsKjRtmlsrHJ4ptM+10GeyMai+YGJE1bgdsHv36Pj
9yzxUoEY7LXDo7AQbpe/PLZoqXnTMQi24/+7jjOMRdxWtTNjBQRH7vYeh3mwDvNN
gy7AXcYC+EM9zo604ZJ9Ui+b8yzoI8oWJhi47dUA9RNxB4JOU6RV285d8pCR/pyR
kKuTNojs5j3uHfCRwyrriuVRGMsJnoUKbbQ9wJ8Jx0xamrXJBQfqp8yi2KLpXeYni
cyMlkErYBCBNuLX/q2xc2tNO4dUvE7kVt+bDoozx1Ql4bRqZJAHptUWxEuz1a9Mb
Gs3M6j1d0fuXmJbc338aAdkcAkqWYJ3kOJuV5xwmsY60KpXaQ/glvL41gGA01bdb
UFJD1Nr5mftCfrDf1COBU+Gkcf72xtGnlaF6QNSdreznJj1hOpHK/4IrCDChWsbI

JRO9kz4gBp0L4T20vsAjTq166fhrVZNU18mh4b//LIGHwp6pITvfA/7PsP/NBewS
 1/OAagmiSYDKONByLYeSND4iMPv/XC07RR7+LqjZhEzOQDxGiA4Qxaz3D0wIBX//
 SQI1r7PA5xcLt03AWMbfOUX1IDpgoCL0joJqKQsRSCRvKS7tMqTq8R4jq3Bepx1h
 wy7c1FUh875araXYrFP/Qodw3v1weVou7gkIMt3xYLJdPukMzovZuUYtFyUbug/k
 KdzjZMs1V7z/5zebF6vXde2T26rJX/x2lnl+/6Cnd5ouzYjVtYUD6keay5McDeWm/
 jd4L1SWKIXIaP7g63Z7PfoESg3LfZSSQqEyoBQCjsIzovw44nji7g7hhnt1RUYfW
 ansgLFyQjIoytcp7jStdkUpDF7D5gVrzfR13Y38ICQ2K/s6kUQshwg8+EOCIJgDw
 01uW60Q3mK9m5KkGkb4gTHKhQ6EUEKiYzh0N8Lz6EuYh2U1FxFVVojsCPXSOkUm0
 MGGouE3Emh5oqvM7RZ1UdZqCgZ8GEsXyVd6Btw6e244ScNa0PawcHxN1Y0NL8x1Z
 ZGjainNwNhIm7+Oh6310xmWAGQDHaaxuLq/IgCmjzykv/7EiclsAGx36HtroVBY2
 hn2AvFBTd5jxgwRsQZB11ULfzFbJI4DN+3F7EgZJpHhW3FDU53zGIMB8/PyDN4n
 w42R0kaoGxm1FMMfdLEt9FVvraaA9cLc0lcpa3mUoyOUeaHnKzHnj6BuZ4XChjF
 v6PHxHopLPh481OdSKvbrj4E1wxw00F6+cHqNf01aLkDopyk/WrklnFftJOEAOHC
 wJ/JFHBWputTFsxxqKbcX9sTij098Ev/RoBUaGRZUgNFNQoZ/UpOhvu8OQeW3M7T
 6qb+WbGsxS2yP/MHK/ndvJD1I+/1NxfBEve7A8uwMLTGvbpawNaOU91m0H7tQhix
 Zs4yW8RSs8GtcvfkC5f+mvwTHKGAZGqR3RF0wSeqT5PrGHBjtPQYVoSbyj2PL0+C
 o03+/TPoxqt3GiqhPquawBCi9B2QfJS/G8H8naocVhCcxINMx9bhIZUIVbz+0Lo0
 NSHpeok9++dHNMFigsIpeHrXubh+829CI52WXZOp1tZXza9XVGgcBD3rH1FT2mYD
 f2dtO43MDcp0WYQtItFHV/CpmlK8ro2o1+G+ONhkNgRD7h9+2EU2ZVgSjQM6U6Ec
 Y90MH2zi5UWzR2z/JPGRCif20pyzHz1WWv50W2t4LxU3CVfLbMTLe7LW5GULGk6
 7RgazcpPHMCokgUxOggyIA/PAi/pYe7NOvrBbUuqK7a86V5vMAZkQuKXhH1hv1jc
 DFv68Xwt5AIazMGhmWx+sn3ZFN17NU/ymWKXeDXEvngxuJjP6ZoFOXmm+TCcnOUel
 +TxQaF0VG7oVHnQTqJCRCjrP0Sg6IQ+m1gS6Tb0bDS5jeGM1uP4DDQHV3+1wk4x5
 zhjPpc8VJuj/h4e/v6Ixv1vnuBri+g8B9RwjaJqIYnMgTtrYKz2gRJuU5Vz4KEj9
 ocO1dUyQRGF/uadBBnt0yQLlojLMkcZB+WzTmM9ie2N1HmIK+RmhJtOHCmN0h0Tm
 DKVVeatwpVcOV4aGsoeNrcmx8b/8t1T0ZHpSdmWCiNoKKLX88wZAINbI6W7ZRm61
 yx5iXaxQu0PqtVqjoDUiObfBVn2/ndoZ8hZXnd4L7P0KnkakNuBzCRSXxdRXu9m/
 OJruF0wtJjDyhnk6wP9zk/x86zt7/yVNGMrK1A7YjxTSzSi6hPow70atzw3TTnm6
 MDJ6N1IvWVd071G9F2tQaH/315wflbzIBQQW6q3wKLF4nakWiBv4R5wZQDIUHsQH
 z0OnJT1cdZVPQhfHI/mgKdZWOW+4E4PNnsDgzhdCsjeVJfAb0WxysyGBpxPs8DF3
 0/aLzMoFTnoysbR6XjmeZE+fZr51GxljessNjSC/64JBznZICv7cNn8N2BhdKMxx
 y1hgRBBVqSRRUdJfWeYQ/70s9MKQMr0pFaIG9SOqnjTwRobuNSsVP1TeNvYSuC5j
 SKC4+UsqX+Yn9x6q82oCO0s3vDVF2FfmTE41i/TyAMUaWaKUm7GCLkJD3NP5DBso
 MG6X0eyUVnw00kNryFDRrkzZC1M3emVBSb9AJZdtVd36QiA1pC2k1vZymbVBaQu1
 oRZiy3zXY0PRKXylj1PIXX/u5tExzIKy4aufl06ijj9B1LrQ3SAI/PYEisYWTZTG
 jPdqJb3yXpawXufjYVeQHcNIjT63dl0tk7z9Jn581d6/T9sTraD+O6Y4CingybdU
 LmQ4LS3vEbJwIQiS2siCVG/NLkZK0UMie7NxDbFr0jIBu7SrbIamNU1fLPr3w1JJ
 fi5i6664AdPxp8myP6AGRiN9eP6UkTr7K1w7V6KVbYQ9dhSpssT9uxW9dYODE960
 4pTI9xXtk8pAfRuZzIhZWMIVgBz9u2GByz6+sze7PDfjP0MXZd3ByPSFPgBctU6t
 EIyEtZ9rYe3Jwm5ySdIeTzZ2S2fSEBq2BxoR/aTj/2H2cd9+BD+DKoDrCAZTV3aL
 8JEGkic+h5HbI5bhye9vRxDY6zywDexbG9PSB3QAZSszYqJDye+21Gog3zStMpXEX
 UzrpfFfzOhr2hOZkAMFmMapnuzw3rvLVsiu3qCiUnG7r9/eJQ8MwNDy8nqT6TCLw
 870KN72CRyuiKaXdm8VfPRdthwzBzBvUwex2DkX8F/0vSAYUc5ZH1WM3xPulHPRM
 7naUuSsv735oWv1N31HWbj2wHg44tXKmhEU0Y16MfeEEkd0IkGypUNkGVysHVaPx
 AaVYrPtbSqmHyCpDeA6Xolu0rEUzPnc2SYTt1GRbPHDv0YmXVmDwEo/mOwDGj6C5
 RGWSRcIDn3gp/ySu07C2JX8E4xredCAPq9Nb+bsjXvqQ1Q0MchAEQK08eP18QLK1
 InQ6+T2938iliBg8ixbipkWsV+Ep7YBSicowe+rJJocVzLafdqyjl5qOSkJcHLsu
 MBNU3LcJn+BA0QB7+BjX4f9dNNMvive0FT49o32XzN/pEdntoDQKsZW5ZPW76kUe

```

ctCGV2moGavodZVD9Ur/HWdHwYhRyrAeRWXy14YCeYD+K0S4GiaGYKuA3rMU+r1/
X91wYcdaC00G1i7JGP0ka+7HmoW6iDMHTbg024Iv4S4ot/iQM7L47OFraAJ05zId
i68W8HRnZSMfbwC6r36mT1hLNZ1/PTYKEZntZszM57dK2qEmdbI/BW530wwxQ7TQ
JAzVES1+EVN1jJw6EIIVXK6q7uM0woFCBYLhrwzy7kJ8jsL+5ugyEYKPsZJrcOCN
f2aznRRq5m7qRACNhlppSv8ByS6OGAbG964j4fbUYtdcXQTKA6OZ61wBd/2jprt5
OudG5QjqtSH404RYZS3F2KSbC2jXvhhhJh++/vCP I rhleP7xcdMLB7Vhffq0Sadf
pSWqz2mavJqA4J2qTixNbZuef0Rc2zNBpYWTFaw2F9AIwYLABzjQTbJw4Bodquze
OWsY//12b1TUESK+Tw/8Lu4tEq6qqUzPwgrFW8FfTSX3DrVOWFIgJBdlqfvss5ta
vDNin2vh3f3Rb15p8bqw5w1QhEFYEB0YdZOM0IUFKsTrtC8+iAnuM6ngoXW+ldYu
F106Z9kLacsMTZSBzC8SVjOvHEFTysH9uttHvNtBLF1HyRCN1aND531Nc3J39Ftq
yiHm7xWQaCZSFcvoIgoaFtkt78H1PJAoQVTGWA6Frj0oTxPtQufSaqs58aHWZJ6G
jjskZbSZP9g+gsa8tDiIxepfiG/c0FG+bfDsvM0hHgtkfy1vEi1v7fAghkZmT7d
kiBII9WtYxfkppjyF4eSjYoLFskRIys+v4Ki41Ys1SrbDmeBBdoYEnD8D70qVdGoV
Gg1n1w+PBF9g3EgtwKxV66LvFACArHYzpyPzuzT0ICL6sjVmRFgNTU64Dra4uaaj
nK7iUyHKxPPXMD5oTXE0aBKbW6H+fySrYcjiUKW6N5hk1aGzkui5tke9L5Gn1ZkS
J3sVaJduS1L4fdejTFitqStbyr0YDp/iaaYUH6TA03YS6TxMk5uCgiLjZOohoeef
9pm9SCTWKHIXiX9/vP14ZqU8rCwt0520U6qK+hx2RVENYOY1LUQRUYucULc9FFdW
wnD6bi3OMmMMPMvVbtbMKp1N9gsBtDa9yBjRwv17L0iV9OLc45pJpde6Xd3A2P/D
6mxX194H+4FbvTmRn01JHHpgmJ5q4faFcj9o5XCUMRvX8rKp6uxGX3U+wDJSq9Bx
12CSArU2cJ8D5yBvns8eOHPFb6V1cJw8FFMR3g1qezR9pg0z+K+ZSJTFeTQf2Tm
4HhFY001ZEGBGHHO7NiQP26Mj4EzbSSfUSEIgi0t6+w75uH6+dbiEyPm5tAwpk5C
DLy9p8eVkiXz8H2GWQjULBYzO21dK46b79Sa1pudQ8bHyt/eVT/aMcs3nNwn9xO7
ZpddAqveyjwMf4CE+gt8zmAGls6Waz74LTNJIdc+KNkLg2VpAID6U1CrpjzqPZv/
oDa2DbKyDHLU9T2AiTcGBkmGYXmoVLVfuHf1XDeVSDyOPtpOdcEkzBqy/qRf34MI
Kx/X42u/uOX8Eh9ivApezUoAp0J1FeB32wPtmmfN/Lm1e3IGtMJsNkperFjVq78
rKQF5uf9w3CKdAqWfOQBPKmjP5WI5q99TzMTvQcNiKW3f9p1HbmVaEIVor2Btws
B6rHqBxcvN3mTy27BDYzvJEGe7QK12kfeNGIRmWTGo/DT6xxmwYmVdHTboZmUDKI
z129E2C4ITu4A7xvT1C0CScD3fvjDg7D2SVfcYSHzA/K3b0jkOYMG0/OiU1HOI//
iYFURenOu70sXJXtT1ttz4cQEEkRgKN9SiIloI/TdbwDcz9Sg3+NnLkeEG1U1Ez3
eFUbAsBCwJBVZQACGtAtyLGEElMEdNz2za+G6Mpb4MA0XTI3gENKu8SAKLzAU/DC
Cns8/koY5tSTFlPbwA3cxrrFXVvWLRbqCfEpa8/L/peuj870nOsjtr485s4+Gca
t5Yde9k76pIC/JLFBa5GpTjY79wevaWEmsmKTry97cn+C73zzT4YxVFjpVeRuCBH
4ScqlsR5315HRzoP4mCkIe7hm7pbYSd9tk+uJJULCu0h0ZiUelbNtnZQiSp/zGqM
MdCfVk66rAsqEdIY6iwhMos4tJHbn5xWrugyfjc2jKk=

```

B.3.10. S/MIME Encrypted and Signed Reply Over a Simple Message,
 Wrapped Message With hcp_strong

This is a encrypted and signed S/MIME message using PKCS#7 envelopedData around signedData. The payload is a text/plain message. It uses the Wrapped Message header protection scheme with the hcp_strong Header Confidentiality Policy.

It has the following structure:

```

application/pkcs7-mime [smime.p7m] 8020 bytes
  (decrypts to)
application/pkcs7-mime [smime.p7m] 4930 bytes
  (unwraps to)
message/rfc822 inline 1038 bytes
  text/plain 325 bytes

```

Its contents are:

```

Content-Transfer-Encoding: base64
Content-Type: application/pkcs7-mime; name="smime.p7m";
  smime-type="enveloped-data"
Subject: [...]
Message-ID: <0e210732-9184-5855-9a95-2a635560d3a6@lhp.example>
From: Alice <alice@smime.example>
To: Bob <bob@smime.example>
Date: Sat, 20 Feb 2021 10:17:02 -0500

```

```

MIIXHAYJKoZIhvcNAQcDoIIXDTCCFwkCAQAxggMQMIIBhAIBADBsMFUxDTALBgNV
BAoTBElFVEYxETAPBgNVBAsTCEExBTBVTIFdHMTEwLWYDVQQDEyhTYW1wbGUgTEFN
UFMgU1NBIENlcnRpZmljYXRpb24gQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIeh7j00
Boq0MA0GCSqGSIB3DQEBAQUABIIBADm9A/Lp5jfk4RN5/fhwF4iuuVOef9Jr8ed4
zR65esdHuxyfoM+gBpdSnrLZQ/0uWwXFKh2ORkofXO+K1qm+UaYbOq5byHwddP+6
iNP86nopcRjpeUNqsbWCSWr+niLbjxfZyJX3brf3+ckwjgo5+gik4wePBK1c58Ks
DmRWb1l6bqYeCREIFAHJNXd9dpGcUkyI1NunHi9720uyDqOvmM1lXarP0QalZ/Au
/D24dDevouD04V6YGvbQ0Xy1rJ7DeIowrlqAq3t5+NbuZZPgDDQ/NdmLdrQ01sEi
0v2M820B0uM9L/6nO3BrFw66CWOx+PSAwrTNRnWLP68+XVJaHBiWggGEAgEAMGww
VTENMASGA1UEChMESUVURjERMA8GA1UECXMITEFNUFMgV0cxMTAvBgNVBAMTKFNh
bXBsZSBMQUU1QUYBSU0EgQ2VydG1maWNhdG1vbiBBdXR0b3JpdHkCEzB8R0APhiY6
HGLS64Mv1sDXhpQwDQYJKoZIhvcNAQEBBQAEggEAm/DOeCI+Z5umxSECDJc8oKbW
cicXtQzeI2drFZVX0d0QNvwKLXKMM0Jt7MzeJnYG7J+LKI/VbVJJ1kGJ0rDEYNZe
6cb8HDEW3TJxhB4BMf/offnCpOgw1E6+wlp0h8vgAZsPW/dFSMQKpJU+p1VabchR
Gu45855mlRhL+m1FR/ihLARYrecR8JcRmFr4dFCXcodVIHDjwGuKtk2yWYRPzHcu
3SwOW4QGCKyB7SiWzFfuNjoAmBnZA7qhI2CYuZH823xiDMuZ7c1uDypXokDvq9Kv
MPSKR22uK245maFCYuznTJ9Ytsx0ZD4k9u5R5vuQ/TW6NSEfOpXhBO4BXWR47TCC
E+4GCSqGSIB3DQEHATAAdBg1ghkgBZQMEAIIEEGtJ4Z+U4xbXftzqmsfU/U6AghPA
p9ayt17k6BlaYmjgIihLk/8MjagX8PWTBf8EyjvtPgSVHQtFagEUSz6qxqD8e+EF
kgYXoxwqQ1jG2SgUkMiD6Cnvo5LTABgkStQe48hUtZGTHiNTzdjy20e20eZSbtr5
M3+nwb+z9ED5UChCqS75dfCpJwvtOFcoTn8EbARZuK0xChaOf6QjaAcVjmZEbChA
hjPWg79eIYvpiNUqmtDd/Fic4SyqlI1WPcks8EHhJBdQyIEucsSxs34WMf434K3R
wQ0Uj7CFc6NEptG1aYodQ+ln9nbo4LMz3fa7ZlYMYsSkGn7zmGtsVbFCr4K0ZwSn
M/NK6bAI5TFYi+gky3myxxV3S9nW9uFOMpRN16kDKoUdoScK3KWoNOVn4CxU101k
hNzGhIhkbw3jqkqxtlQAcTvtIhb0nB4hMFIgT00Ei7Fd2UwsVPVD1VSG9P7D+OXg
a6G4CzOCV9zLPY2/Sjb+J9saq91T00NyDVc2tyttc/HpxezZMDeD1A5f9yj6HK6
kg3c3E6q5u6z2+eyC9tkgAsOPUT7NFYHqfkmClfbbHPJqEm9nxRToqIEEgfNiy8
jx2iFOZN5kbYBzI8eJEjHXkRXEldgx1rH0koUEgOns/D/a6sNk6x8TeXOK/elh7B
oESRWMKWlG9XsVvbIyTUELuE4NeQE/Dvh4ihOykkM0FqmMHPNYUcYvFQEtnnqCXX
+L2sEMl5LR8P1Pkhi2nvVtQiNTNxcGh3JlZIEFaGaltC5qmujuTsCpt+nirfTGTu

```

RNe7iYjYoiC1XnaMgJsMPDcwHsHdHot8T7ygUpKIPCGXCYfg8LKHv8k+m1/MvWMjg
SkO86BNro1DutdQzXNHdkH6t5deKcbRuPe/bVOKNBavG+WYdAjHJkHTyWB4YdBey
rvadVNQtI1qqpNhIanX6I/rJoyabAjjTs+pBAunzs5HUbrmYmIGN7INf5FDdQbDb
z1btZs4L42dwvJXGV8X8OHVzFEfskSSoSq1r6qf19T/uaX5OEXNWuNrI2k4i5Wy6
jDrleDBctOI/QDFtCaPuqfDRt6I5DuP32JebT8J5M7jX8kqjPUfB/ufyiZCNDCRd
Q/S/HXRNF846LdRYfg254fwTfaN2LBKIFMS5veiDnWkHtYmlXKPka8GPi56FZFvG
M4zStmbWUyd9AkeWirtGwL+d6hqk5fHwWSWT2z39iY1DhwVjoE/ne4JkytM6O4ug
1AnIL2e8uz7PXI03gWSCrcafWZfYK1iHh8AMcYThwjNqN71MBIU26m9am4GvcXhR
jhvBI1hkMU6JrgoDT5JqRC6gCI8AC8pIJX1C0uMSS60Vbc+7XSVi+oWtuZCzUBqa
5k1MzAH2NRRKXzs7mo7YuECRopaH01OKQ5eCTMAzHA3VxJG2no8x+PTCNGNOC2Vn
lzPMVnCJgDdpGpm84+KA4OSUSdIPmxuSkSfuEbdH06vBuOeu9NEjx1JiFobolGGG
7Dqv6O18ZRBlieCLXEk/JuL0yM2KZ4oEFx+iOPdiSNFuuupwSYHWw9HxGtTMZ81P
+XE6KsoTR51aoTwbUbnI9GiAHRcOVRKaB5aPFJJ/1hkculb0vKZQQXgQrWc19Tbd
lKwzF1bp7j9bBFoPtyP23KwTwlae10ACagVbEc7+2ZWDcOMs3ypB5NwtZT5BCiYe
xj/t0ZYD1KWN99XV1bRmmx7r0osHT200+cjmgW9RbX9UPADtzGmlq2Pymj240hn2
ly7iM2td64JjWMSvW16HJJ6USOt1/6oZOUSow+3RoX13K3Bz0UmB8ZjTF3WQJI8a
8TJHckFf5RR5IBQiNTU3mM4dsuFXhr/mPQ+O5jNKyEaW4FWgH7z0rn6ksbhv6A2W
U/ohnuKWOMj4is9yAxVnFMJMqAb7q1mSNA2IKi41FyZuQnV+TxaxYQpppFx2x48e
11Zy1tPKuV1xg+1pwW3DBmawDOAAYIpU4bw6s1COIRMNup6hXLxULVegKIpiRu6d
xLccRPyrhHhsUfmsaPOqyZ71oTUPJ90OpVK2LuY018aTR9EWW/Xk2bompBjfh41
Fzdpvxvtd2mYKN8g0V68uF1crnQCq3yvJk/21DFNL9fAtk4ey33vQ5Jv1peCe3hP/
rR2oBpyjdhsvIKv/gjdKgzneTpEGxfus/HlvcIgrC3/umwVeHB45jbGh68/dxcEU
UtA6MTbEskhGDSV6uhdvQQ5jCeZyINKye1fWuna/wyVpDJ2b4N3uwj2bbPxVsPPo
yd6wNdAkDxYc8IWY5I80t2U+Ncz2+DrrwFE9ZyMMYkJJyDhp1RDQ1n66K8X1Hfa3
N/Q/uFqyxTBhMvvPZ1GThSuJubC04KoaI/5XR3D7zmQFuINwZm1GHQEdH8r6tFuG
hNjWno/X5BK1A5/+VC7300ucPmPhsm/nEJ+y8+eIMEG3+yBQHL4nvEAbinj6+aPV
zTHqYoRnaZAlsbSHZ1KjGvZu7kuN8wrr+W1bWnzo89YXRmwa2UUCf8968i9fKP6n
26uM4WCJhUC1oxGEio+6urXBq/htCSgdE5OPpxKxH2HDvmcNQzzxPjOQPkdbi1f1R
IdiObHbA1MPOgTc8Cz1qAKZFmoDjWoimkbWJOLU+Ft9Ft9ru0qrCeZXO3wt1xGTz
20omYEdvK3yUChTA71BjzjMnfF+eoX4bHVGFcfmgqPufojZ9bFNSTpuVv85um1Nh
gxzAuMp0gqzoHzUaCjorSNag0d5N8HJSC3iY+OJaI8fNfVcOugb+afnxjqRTxDnK
dGMu9YyBnZB4iLzG0uIoT4zKmQDdxEJazCg++3qBW8b3P1KRyLrI62xXhPqi9cgM
2n/UreP5L4giCtwVM+u/nbV+jw4TbcMdhETom7PC1M0fpc+1L2v7SaqThBZNAfa/
dQH1bct+zC/sJZIrZCzJ7gHCc2P8Ssx9Ro26E/1L4pOzTBkSeW3v/4mwrnrnd/b/
3sheutKzEBSmJCBMioV4EGLu6m7iQNGs3dZoWgWyvQVJ6nrKQ0gOjRlI8x8yGZt7
7X87m16KEHTz1AEkTI96QDw9kesWZ1Tmc35zW2ce34ks0//uqPUk6fQbUIXIiu76
YdXns6VZ7VZj2NE/CqI4zRbLlhygeyyMGeU661Rh1flzb2qJWXW/Nh2yc6pIm5O6
XEc2KqG/rXw4K3oTdz/y0CpgW0zBMOa8UA1do58EH101p181m0EmBjAg18yk0NGN
lkoHNmWhDkOgwwY1Fncx1VqGB46io9oWhk1DRGiJqL02MmYmdq/Xnu0HHQciIvH2
3T+/JQ3mYHrbHaqb5zKX8ZU8QASMTXtE/382cWfAQ1xB015t61JA5z04IMZ90Whk
NiMZVb9ExHFjwz0EGZLzG47mdR4APxSUjNEY8Z0f7Jdf/cQ6LLM0HphKocXU9v2u
oFYV1XfL3uWq/EtU3PEX97NkQjdoSQ390BDWxWade/PgOVsyboYHC7y/njkjj/Nu
3hSyhH77/j7iafINbyNudIKYftjzmibVKV+OKb+/ET/r0sYtPIoA//ydD5YjANsD
8Z8/WSd7ynve15OSagRnC+b/FuXyKBXLXQgnf4MhbpzyVOMzuhgWCK4u5e5iMGjG
Nnn8LveQ11SuvGK1Wx0BCymk3OWWCs1kPYxPxU+m5XQ7M1XTyUFAXv4MSsk18+O+
RXCTjMfOUE1rmKR75KaRgn8NpEpD8/PuzBF0EAB1dRX6AfrhMxjvxi+HzOvhVgR5
FjcCDmu34X1XdT3hWnVgp7iRjpkbMmF07K3ocKWBjtGhGFXL7nViJaY6z/58dCt6
6IFX4NDdUY4RTn5LQZK/ikoBIJV81ndz+iGq8H0KCynuuZEOr1AUuXDC0luQtZhm

zJgLYZy97SzapoSD51I/P5/e3J4cwU/3IS1+IZon7vWBVu/k5sBazKp1zmc0VrhAb
i6qLPm0PT2hNA9zuzzOg+RVX8QvthJXeF3+wFRQKXqWf+7ksZWz+3w20qxaXIq2k
lCiJ96p1pY7URCexuT8o jzu2NWjb2DHCr/zD00oYYwtFQGN488p4W5y2GdZevC5F
c4ulz4nYifn3P0lYeHboCnVNhnCG9gfvWMeoeFG1brRqXXQJFdDqW0js4/c2sUHI
nJu57c8QE99N9Ff+V4LwcDcOsPM27InX1BntFSyaIF7WVkcXSN3TG8 jJ9HdtIo8Z
tnTE3tgs1jrWzh8f/93XHf22e5ONRXaFCMFx8Ytd420k92b0hiZUf100f4iq7W/W
YVzuBuBub6Qc0pH8bkQ9uPNY+LwsWwDXoWwQxTq2m9kQVyyZ86Kl5mgoNpO95Bjk
qyIUogq4sd6v8hsmesRbodZWLPdE+L4Cqk5VBj9IEqd4ysmc5MrSn21hegQKlRnT
UsCgWwEs2Tk7MYyH/suoXw jYBXf0hXWdEWsxHZE6LFDEFnMJJXRgdg jduEah00Tl
Aplo+In9D6hbH7imH+aDERpfoiwpS8lJtmHc9JZi jH0zJpWzulxpoowdJ2gu6uiR
CkWRP7Cx+x0MMtMXWZmxNZi93FwGUE3VkTMeV+Vrn jOhrAH jrN4S je9POQCW5ez+
ankMWvnqjJySSJEKKF2r1MY/bSrd3nadrm+DYQgKYORkkZ7adqbhQBYW+y1qpyld
XtZ9R5RPozSNhuMRluTdPgu/GLfTTcWLF j+hTpFFUwibzcpu/uOmnP8vOaxx+kAp
NEZbxxla+OnidzH+DJ/atOeJGfiF3/c2W1apsSRDXFF3f0bIVTKX8nF90nICNhOj
3MU63gN2ZitDhpRdIe jMeL++Ew6fJIVnBIJsQhCZNTXW8MvA0xkDdaTDah5RrkGV
cd50F3IO5/vretTC+29bSEE2DTkBoa8MgYgqo1XikHfUpQ2MmshlK5w7dtDre34b
zEBu009M4EBD59wxKnzpfbnwI/7i19GjqdJJs/kHcFsZ8ySsKldW6idfrkKCeUF8
MhzEEJtmrwrveeTgrWHqB9gQXUiZoS/OkzCb0Ks2qQmf3ilQxtXS60Hsj5xy291S
/jL9aQ101hDEpgeIyqE3tSkDKfCAdlS03nmd0HEeLpz2ehxUiT2pfsvcHF39CoZ9
bQXFfOziZmJGRtXv1ZE653IALcZaJJAQd jQOTaR3+MnBZ0BJ57zw6MtggbPnMHQ
CnCS4EJ0OgHwZcNGC5DU/QqELmiCyuQwUtwdqLgJFFs3Fm/KnFZmuzc9eRkREwea
hOzRdqFUysLWPC79PO3T3abokv+YB9fHh1WIiRlqYhUTV5Pgc0DwZ+ra7rSi656E
JhpFQFe4XmYtiMEYm3+TRV4NxbqoA0x/Apz3L7xCaHkCnszgv9RfmGtCNTb+J8BU
Ivh+ENByU6lAkeCmud/aYIRsOZqVYNOTITXnJspOg4Eo7etLX+dng6RCqtYV/dzC
+C+zL7iL52/WxFp9Hm7bzGaNQIQrP+TayH69yhJ5aVRoM+YDDaqwARv6AcxL06RS
OTGeobPVtu1UFQOLSsWKuLmw+E6YsuX7Pq3AN6dGeFamn7Aw/HVXoFyZ2pK9DKRt
CFNh6q/kdY2nVzXQ+mCoWO6qysw5WNk+BmcUd+GVjq6 jm+eOSS0U+VHcxFofz3l0
DptR/hDz jBhLn9wfclooN99hxKZhXH+aKZk9/AUjau3GU7yZGBNda7NOJodjtXe3
j+SZ+nVcenPKuVewTHEOzDp1U8k0KwGW1+EW+Lk/z6OxyAh55d3cwRpHxYsuJSUU
C2eNrgv+iKAlKY4KBmDH2T/U58k5+qXxeHpBPdRnk8yxvDTihIgf1jLR37zhM4Td
M7F30MyDGT44OBAHTEbPBhG0B5gZoj0mIyoBhoxPi/257AfLXXY72bzKmswqmok
PwMhH4J9/MfLnJ9uDN03dIgcJ1kOkCxx8XF/BSs4Fda2mfwmawTMRtk3BB0qfIYkB
eAW2DADrliZEkL+SsapvFsn+9HmnVICsIB6gkOtZLCKyVwkIThwidhNBknqj93EW
yReer8xcaoIdRfJNluA5ck0A+f4hZxP9lo5fqMs8xa+sdc6fhpLUoj4RUafmo1Ss
P+4DP jHkpTGetlTf4t4cQe05ZQesVRt3Bis3nmKpVPV6jv22EumjmsEbRESsidDQ
3wnIADljTzyOXvAESQm/SiRQ7HyrHzzSwyOkO6MyuYYQZJVbQ3kBW2EmuBXP1WjP
I62JN4S2vvyMPuKIWxSUXoraWWI67iK4rmK8PhiO2I6bfBlGayDw924X+xTUw9d/
nWO9+xuSQHZIk2ykPb4c jvPKxV5Z1zmfI+b5WmdTF32SKR0tPci8hcYsBgrfTv3+
UME/HraCoC0eHV3mzRff0puWyEu3v5Vrbip7Nz8QbYGkm2JRDfIip4ZD4ZBUizJY
qyAJHhkpX0rDgAnzV8kkfjdEAF3Ji6+RKNgrGHcKq6gyE6G1797Vzof7MgzJy6en
3ertjNhZGjms0qAUSIsx1jQVF62XoLVyO1uZwU7PqxOgfJSe4JyE8a+ddcY8x90
Dy2R1536+eeRmguf0XC1G9wd82w/OadV2yWMOmpxAB8Ase+iU1Wyz7YtVWlKGye
LWfbtQqVSlFzQr+MWOMi1BT9+TPj+8EIQodap1PjmU8RLebZs0EcNaPv37djsIFn
SycK9UB1Eai7T0/1Yr5h3f2/04XsLqtjGwq553nnnk56WpIc5Muo1SS1jiz5OX5F
lpIdOuLXNQlG/+emf1GTbcsPta38GX5VAwe9kf3vVjsWryw1SNPXYoOKAJVkBq+
C0nuJO5Lu+dbA+wkaMCEBw==

B.3.11. S/MIME Encrypted and Signed Reply Over a Simple Message, Injected Headers With hcp_strong

This is a encrypted and signed S/MIME message using PKCS#7 envelopedData around signedData. The payload is a text/plain message. It uses the Injected Headers header protection scheme with the hcp_strong Header Confidentiality Policy.

It has the following structure:

```
application/pkcs7-mime [smime.p7m] 7930 bytes
  (decrypts to)
  application/pkcs7-mime [smime.p7m] 4856 bytes
    (unwraps to)
    text/plain 337 bytes
```

Its contents are:

```
Content-Transfer-Encoding: base64
Content-Type: application/pkcs7-mime; name="smime.p7m";
  smime-type="enveloped-data"
Subject: [...]
Message-ID: <0b3ea6dd-0e91-5a91-9bc0-3d553f892983@lhp.example>
From: Alice <alice@smime.example>
To: Bob <bob@smime.example>
Date: Sat, 20 Feb 2021 10:18:02 -0500
```

```
MIIW3AYJKoZIhvcNAQcDoIIWzTCCFskCAQAxggMQMIIBhAIBADBsmFUxDTALBgNV
BAoTBELFVEYxETAPBgNVBAsTCExBTVBTIFdHMTEwLWYDVQQDEyhTYW1wbGUgTEFN
UFMgU1NBIENlcnRpZmljYXRpb24gQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIeh7j00
Boq0MA0GCSqGSIB3DQEBAQUABIIBAJcdIoUSpo1n7vGPkIbII5F90QJDgjFBWqN8
mrP3eorKCd/HmE614/YrIqI4MD0rcJBkd6xNbUeB12z3wU9w0tyThZKAxZH8XkNw
ZZu1aA3MRM+wqwCnxfJTSaZjkIMhsMe8U9ROY7InwRXqH200QRqRU4iJpIe5/DUH
dn/70Yq05g0HOGjzWS+6IoQdiHf3eSU40AlqNyg0QQT5CP1OM7aRXxt006GWvqLW
Lq52uimRL8AanDUkrEsOh1DggpFwsn/kTkOq9eBrjgNA8wHDA1BYfoLBHJQvn9yd
ivkXnsjIqoaBcx/61TLrP97dn2v4STbiZd3LDe/8yBCdnOv08qkwggGEAgEAMGww
VTENMASGA1UEChMESUVURjERMA8GA1UECxMI TEFNUFMgV0cxMTAvBgNVBAMTKFh
bXBsZSBMQU1QUyBSU0EgQ2Vydg1maWNhdG1vb1BBdXR0b3JpdHkCEzB8R0APhiY6
HGLS64Mv1sDXhpQwDQYJKoZIhvcNAQEBBQAEggEAAZ09H8ZLLO2dMDjR2ysuzrqi
j3KqVh8Rq7uzjB+IKkzFfun3FxFVZ1UAvIwb1Pwrt3lFx20ekpF4PzC7x9sdbxWJ4
ZJKftmD6sMZ7DVeV5GABH3C1o+aY1MWs10Lq82S1TBzwcJZpKf5srR0QCuxaQq76
47owb3Dd9Ecn03AIPeJDy05EMNGLRjFqc8md08ykQEJwHFxeZOotDWDm3lBAmqn+
An3leGbsWMcYYwAXoz65me1W788tJWcht708gsiVzGdY4Nd5gQAysf0/iCFhQQzg
X+vrFmPwm8EJUwHPEX6I0V8ylyDXBt5qplJgku+51eH1BJtF7WWMVvI/1RSE+zCC
E64GCSqGSIB3DQEHATAAdBg1ghkgBZQMEAAQIEEPmUUHGHYj8IXE0zUFiAKOAgHOA
iBj501lKw12s28QbuFG9Tc0EjhF/AcHUB2kdSeTBAKLACOsXNXc0/eDFttc+v3sJ
eVAZIO3p97Xjwzi7q4YWIInBEFjVrZilMfnkmWymEEuhpAx8eYBZ176CRtq4F48F
DHekBraJgyEOPiuU/k0K0rsPu0/2W2vsvy00QsefIBI/LN2+CxgPqcJ9+s3Veru6L
```

VbHR0Ih2NoLj2RPi1czHvU7B/dQGIWw5e0ZWJFAiEuJz416Vp/9I0CN/Wwj5BO8Z
 B1cm/rrD6uM7VDJS5jPenm008JHd8TuJh2XBdscrw9sePmwyubHG11fViRotOw9r
 ux552Pq+8Vvx70+nZVvST8LzYfKT/GmRt5qP1cyg81UgiNZyb1wScDff9BIXNKnA
 +8tZE7yi2VA3vaUMGPb1CWbpTm46mbhsfzRacyyB0pEK1mphHbPdU5pFYULJUHWu
 t8O5q7gWy3SeJxtmkTsSbMfRHxaWQyzfOuH8iWW9IPdPd8bcj1Z1pk1De/cy48zT
 TWUpSy1hnEOhuB/NLwycjT0pREGH007G04SvR46bH1t587B9Uy3q004tn3NK8H5U
 +hi0SpMxO5CjP25sOrNT0E++zO7dUe6ZjnBDyO20o+a7ASKG1Ec1bJfa260+TxoJ
 kW+R8eMHPF29QefDz8LIDozyu1Z/te1KPX/Y8pIaOPbnaqcWmO611k6kCWLeZAKd
 VCpdT3DHbLDdGswODw6iXAMpe31811knTKebGJyX8XTnPTu3HOWirdtoMvmh01Mm
 DqqSKR8+uxg3c0++pVGrXwCurOyYArWvkl1i7ZGy7Ve4ToAOq36hImlr+Z1+G8+Vw
 Jj26za6H82M8w3kmuZRwQwqH6YzAivjVxutOnwU8DTSp2b+eIzuyb8dMRqp/X4aQ
 CVGBDC5Uqrtccealbs8pV9Q5t6KDh7jEgISAoTdhmrnwXWnXqcWDX4RCz1dWJiI1
 DeMJ1DOQcxuMdlvO15qGRI8PLXa3FPrAABKQWg0zFV2UHqzP7kybbYCOQy++TDA/
 dYdJ6SUccFfuYIfuUI+tJP7tYihnjZmD1JWPuDoIymjpnRk6t0J1ZRnka9UaXWYD
 wC6sLn3yDvKsP7ZdiQDUt40fqW8tY5NmaxTSVdwOMjkkB9JN4KxWKsox15ntar8e
 So/vSDBPzSLAjzBqOnP0RppwUAhm2eLiEPNmATCk08jx/F/bqhainHUZFmq/2D4m
 WsoQogJ01DtaVBvtzc+6GqknFATY2wZVpbNe0RY5T0vnFfD1g/S8BHKrGNX36tBA
 +2S/Lj6qm5B7Lk+BOqrJqhfquyUaNsmvyxftGbzcdjmAYtOLEpaZ+QtDPBjaZGdd
 J1v5hErhQW9At52gT35iZP1kKMhMWFkC09VBQw0QHf9hv8p1t3ugvYpmLn4fzKhe
 xytNcyLbaWooIDi0Tbpm2QZw9IJ2VyW71Qtqke0q1LA5m4tkB9/PpIXu5xeC0OgPm
 orNz6IrEGtbbg1C7OFesJTvRLE6zzmyT7KnR1CjTLXQ6cODPVduoFF/qISe1wKhO
 hsBHjdGdK4rcJbVzX5QkJbaqc09IsVnkpJfATyDBN4IGM6xzLbBXxHboK08928ZP
 PnDUYjzyArjKhWguH10f82ioF/Y9miD+iohw1TpHci2aHo1Tf1XgLTWtOLCOXxV19
 V49w28dLoz4jPi9P1OpdsVK2q8gag2vndmDPnIK3AXbiIKAYz7GHXnCIKChgBWZ
 TuGfWhVbmQ4yySlizCmwdQ8ws0qCFethd0HaJUBMIKMYesmbwhw91QEg8T/cTpxW
 IsRfC0bVo6MScz4QG+mtUOHnutzA2R6LmFBoaf+25nWOA4bCosrlw9qHRok1AVCz
 f20uzQ5sTKU8rTXyWH7+9qzc+LDb47Y31s1xLNxGcGfOc15HXmx8EOUQrYeIyKle
 lupWdhk5woEwRc1jgnyqsMG55NbLSjPKnmaRyISZBM9MFZRUwtNKt42DgCHXLdM7
 oBEMJX1Uz1zoSmFKZ/eZ1yjrZSZAhrvr2H2u1XLXukLfpSWmRX/TY4e5x707ygg
 2WVGxTYLCwDRGVR3/ugq3hfm15jFUhux+/0So8NVwgWfL74GYJT4Zx8821fvu9n2
 gy2dXVFzQwGI4hYUx/SjztSGpheuAvUGf6tJLBGVQz7Z/2LiNWge9giNioE9M8e7
 8Vwb1cWx6fdvG1aUvwqR7tt7y6kot9giYVgEtzDBv+owhubFa5LFx/U47smDvV1J
 A6fElBOMSJZMU2Obp3ycYL/CEXtillbvX9nK3InWn5/ldG3JtUhWDSUGMxUonvcw4
 BJkxZwAQUQieYYLUF5Q60qF8k5AIWHyp3dtQ+Yt1qwfBpm9ijgfJnxqmtVeUZyA2
 dQh1cDhSd9UCodHm85pLfdIdqUcd06uqBR9TQh45Hpgoo9LM8HjeRXeHAYCiJqgs
 qAnRgvyQCnqUibh1sE5e1rdQGHG0n8zzVBYoB2knm8AwfrlbrVD+nTvmgsGNm1lw
 9KU/VMYfHzVQr1fkxUh5E5ILVNMHPp/4RtQ/10NLPfYfrzrlxFaxbQyVR0N4gTWF
 o8OfmoNviIAAxp3Kbu5sc2k0hZHyw+ASxNB5y8s0gwwFZkIiow5VoGT9LWP7BKbe
 1VdPq5M8/0ouuzwV+2L/KpqYNbOIUryuJJ/YJenJ3g/xmlqcWmCybKIwXWqfawBB
 1BPquzSz41/rrjmkrrqw8Jcf0MrC51275U2RN0FEOcBCFrNquHH3OzBQrUxHIEZMw
 mmzqsM4vW+7qz2ezpa7nPWGFahzqxtsJCs0DnZveLaIfiidQ1x9ePxuraXB8d07T
 OpayZXMmrNyaUkxA042EcB7w5IrIW9Gypkcm8AyA1NgLYbh9hiXy7MMbKOV6sTdR
 cC2cMoC1GMvH+NywpGWhc0WH0yZTbVH6ldT+wXz8C0lpXCmpl10cJv8f7kwFVJLB
 MjQUZCsRnWFRyo84vHTEhkviLEM1DLooTVdvqd6m3XkhkHfZLKFKH1KH5B1SskQ
 UPJszpZB2I1+OYuTPfTnbTieQToxA2BB/HhXbj5eRx1LEQ56ZL6QGVQp6f14zGu0
 ZjNQb81wumE5uUQrW4aye+1v8ObWe/OnNe0swGqhXXPOT51vjbXTbXIZ4j9mMnig
 9fIMVSHkNWgA5KUHx1c0XRypWWm9iwsTFIOW8LssH5gtYhVjShUGxXM4W1erQwz8
 EmGefrRxv112w0IIV4Lc0F8kSgM/yxBE6yW0PRhorcsbMU7wHPj51yRISntcHG3Y

MHm28iiL+ztiEwlowne4R4xYGMT3aThMXCXEYUI77jpocMP3rWLAjt19lsPAds1o
I7PzN/3g+0EFPPh7pJng3C7JZwYhDJ8p15y3sUB8Or+gcm+4pk2aHYz7d2P1RMMy0/
fPaAeoIOwi4Rv4YoaqxNMYf81DuLcY7rJ146PbNPcqHNpbGBaq8ZH0b6Fp7hvNp8
dCyC439vM1bEA9ttQaOcyDi4bGSB7Mg9NvLfcGjSEFvbwbl8sLYsNrvAetKXUDy7
AK5qGHaRykDTkERofBdCACt ruRkvBAg6EXGxtQQtHstDBr6J5J7Mc4jdsBcYaLU5
ojWxPYnDo32c6+Z0qWfV6rKgS1epva4jxSe7TiK7rkgyf5JzF8rE0ZAOEox2UYER
3HDuERoK158ln0FnyD2khZNai003/SJfyvnlx1FYhpsS/8z0TFDq4UmSz+eS5UF
vqLVeJ5yJmcmXy1gSR29EDjemi1fwzTPXF826D4WjZwGGecNt6KdaSP+PBqsTOIM
mReUZsSlu6Dg6MiIIQptSczH/6Xslwz1HbK55ElqYTs8KY/pVY4sjYrxNlobfWpL
MJReUehfNwg3Ki39HUh9q7zYHxufqn+JmKKwnJNp4AOhIW1GoGCMuX3ncr4Yj2C
pRHixXB6/pOxf/UQmpxnBC7fmPeYq0hxMxc2M3VjUWnWxN61jXbPIGgcpzulbej
T5bs/C601iqgRWAV1Pr27DCW1VFPJARsuPb7Pg+USOHF+Vzjom6+TelVKgbQYQRd
xo5M881NCPxyXWsbE9nmYhZpXBFU9wmHhOCfOVfyDDFIdS+X4if8JxQfcbHan+4Y
+OdeWcCVnEzccKgc4K3sKhrhn68L4KEsLKgVSm4bRWFwU5Wf4vahdOfGcczf40gS
NtW2fBTR7EQ91csLXE3VJrETcy1QcdLrIykLbrg0F7qzbvi7RVXpUDrvQGfIsCpv
68b9h45msj5nGLh9f5onwo/DUpU87fkuUNgjh5r4FkNAjdgQI04aYbDQ+KK+1c9G
bsIcRhkIZ8fLQ4Wtccq1M+CVH7hkZwtMJXHXESB+n+iXn60tnHao2St9dtDwY1NV
lUIeqHTqXluMEL8yKs872P81rnPAyVZKg50TW7ie7aLlXTD2TPOfx/pATDVyHLb
VzGaoYr8NHKwrGECzr1Mo0zb9nuhc3NHqDLj3gtwnT4LbVsgdIXwaQ9gEL7E+eR
Y2YFrtz9AXeuEWpvm/DOZgmYXIQeHv4VPv/CSped5JZMMQ2ZnXrG0ptqNgI78Tdi
xuHJDKVfsmLsHRDX0Q/DadNMccjF05i5pQjKqRwVI7BF3vIajtMB1QQA/fYxK3ib
94PceJK1xDb430CZgzgW5+e2Gbo431P4f4HDIZk7lbfTfxIZWdqBlgYHf9ZFXF6S
9kIqQS0plZUxv/4fqLFQ4gs/caAufbwtfeqfFODwecdvZwiAGfThrOLhowxJGhMf
NIU3UiHcv+onKvNi0XODU2YQe9ONr/rK19W54EhpIDA6z/dkTTGCw6cRtvRN22cI
KZEKfU61f1llaDv0ea3BOVY0mIrsTTQtk47vH/HYRXAubYgEmD4WXGFP20tdG01I
OAZ1h9w5La4058urEk002ZJUeMxEZ1BzjeTljb8rZoeFeivEJ7Ns0gitHesLJv81
mWrqhx56HHzLIJ6RxW2ChEkZyMsjzYK9eXQ3duSAd7Ye12/dVQEKQVqmK06UdQJB
76kbQum/jgmOIi2mHiFwChEW76kzfnIqzxd0Wu8nwQj2OR9wHO7KoiI+/T6ur4s6
FP1VBvzfUxt0Qa9EaI9wMUyAVoZ2xNyZSzpLkQh9Yec1FycEjzkW9cjyBYkJSVwc
WDVDFDdjZUulonv0rmlz9i9fsK0tsDYcS4TDkimaDOKrGctnxbxBzzUhEm8jN3W
qoVoAWCnE3TgIbo4Vw1gkFMP37obVrw9ocSMk1X3+Lrp1B+Rod2Ps1n6LbuyFXr5
1ZsfFJr6eT1DFQ3JB1hm47uGURZrKAucCK63kh3Y1zjLlL4mVDrARMnHYZw+2hIA
lFpuTp7Cu9DNSAsMTIykM0UGNU1XsOGRPo1HkMfxFLChb8G9N9SAwGggAT4ygOn4
TZ9TbG98508vyMfRYSLODZ+63bvunv+RUtMH40WQE/teOwNiykDJeQ5igkeLO1N9
SIsUXGsnZG/8UAZSvGxMgPrjg+7dF2afmE4IHRKFBhElp2TkIaKzkbYgRftnnSC
JYSueC9y9IwDEH01R2ZR8keYLGrg9cxJBWb0Ow2R04XmbarLyvFih6AZ8WnPdGPS
mn84uHqyOupRaIDwv065LDs07v/ArqkUZcy/ADw6F/2No9nju7zehWcnOYOx4k2x
x00JPKi8h7nQc0GH+qtIAwt4pAXorqTbGqyWKXgW/TBm7uwdg+ciIaUL1hStw8XV
3RWW2cmL1ew4DzG4auZ0OpAPxkOkPq9g0j6Nj1PbAz3g67v820bv/YOzLwxa69jU
MofBs5itg8XQf23gUVN8tC2zbJL8letTIKnKGvxe1QHm96R83PxT4gUjfnKR63rs
cyrtlqfU2+PKa4SByfb9Ngas/v4h2R95j6JGGtSW1Ua9rp3aFLVf1fACHiMz9EJP
pbPFxUnT5GWxORbP5Y0vVU8RFgR0ArKRZhn1Mmyk9vRaJSrT+6K1c3igKdpDvcZJ
AF8NHDUL65szSSWvc0b50w1wBfAIW5MgI55uqDrhTleip41bbWNwxc3a6yba9qv
lu0ZAD6E+drFKgzU5B86BRnvcCYGaK90WaHA72ptEQcSKbAAe90x3IJ5C15aCr1m
M+2nh0x5JbSuCP76n4PEJgrwYJU1SsHy2ga2xMc4wIvi/hkgvthWNLi3unev6A7C
zF2AMR1vxdJYJV833JkA7oLEojGM9ykjmdBkV0QfD2WPYLAfRLR70BmVo2JB1Utx
rb+g5Zav7wI/yusXsFMjeJ9rEVhBvhNvpmselh2ZnvOk6jUrldNksxH0CdT5hXFP
4fEeZuIxxv0mzkAbWntTAYy7HAhBp7i34Pe7c19c97UnP1ZYB8xCWu1lty9kydQQD

```

9Ve8V2DvgTdgLrc3SHZn1BgtWwISf1jLrX3IWmB6kIRTKoqUND+Mh/bgblfnKy4o
OTPmg2hFLvY64mJEnWC5ATZUx8IN71dsKa18CyDCVWjaq99H+DMbBB+DWk15nbke
ZPwTyUM7CiHIlnpOMBu5Xc9H/2EtLsESNZ90tNbyQHleCU/OaBM/5ivEZWE3VCnT
7VRke7s3JYbcBAkWMO1oRGj/s0HrPFR6ju7LHjZvWIjeZap1Zf4ldJpTyC6yRcs9
DjJIu9BUU1QE/t4uLOCPsCLlcmTzXtZpD+jV7+9wH8s+LZ0AE1GH+3FZyL9p3UA7

```

B.3.12. S/MIME Encrypted and Signed Reply Over a Simple Message, Injected Headers With hcp_strong (+ Legacy Display)

This is a encrypted and signed S/MIME message using PKCS#7 envelopedData around signedData. The payload is a text/plain message. It uses the Injected Headers header protection scheme with the hcp_strong Header Confidentiality Policy with a "Legacy Display" part.

It has the following structure:

```

application/pkcs7-mime [smime.p7m] 8190 bytes
  (decrypts to)
  application/pkcs7-mime [smime.p7m] 5058 bytes
    (unwraps to)
    text/plain 432 bytes

```

Its contents are:

```

Content-Transfer-Encoding: base64
Content-Type: application/pkcs7-mime; name="smime.p7m";
  smime-type="enveloped-data"
Subject: [...]
Message-ID: <b10dcc75-cf43-5fd7-9e48-f932a9d68fb5@lhp.example>
From: Alice <alice@smime.example>
To: Bob <bob@smime.example>
Date: Sat, 20 Feb 2021 10:19:02 -0500

```

```

MIIXnAYJKoZIhvcNAQcDoIIXjTCCF4kCAQAxggMQMIIBhAIBADBsMFUxDTALBgNV
BAoTBELFVEYxETAPBgNVBAsTCExBTVBTIFdHMTEwLWYDVQQDEyhTYW1wbGUgTEFN
UFMgU1NBIEUENlcncpZmljYXRpb24gQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIeh7j00
Boq0MA0GCSqGSIb3DQEBAQUABIIBACIQq5gYVGjxS7N/umioYgQaBDzYuvtrP0wn
3/iHOuNThZd19MbrcaWCYkYZfrFFpAucpVCPZ8mtxHrijYN47vAQUV6uOSDoZYft
drJZYTnB3yuCJmfeS9zonrI+CYksfA9NwkFJdy19b0ILw7tVf2QFEqX/5tU+6o6b
NEoxlwp8I2+tICsm2oXq7rLZq9Wxw72pyV9OzNAwajoQML1nvPFyV7P1nB3EY6K6
3Mcx5TMplYEYEQ0sDzftTXfsau2fbQ756q1myA6aa344Y6j/oeUMeOuuUx/dQJMy
Bbvzma6bLmr1mBkuSJRher3NNZkY5BlYpziXXlZrdkZcClYAtcwggGEAgEAMGww
VTENMA8GA1UEChMESUVURjERMA8GA1UECxMI TEFNUFMgV0cxMTAvBgNVBAMTKFVh
bXBsZSBMQUU1QUYBSU0EgQ2VydG1maWNhdGlvb1BBdXRob3JpdHkCEzB8R0APhiY6
HGLS64MvlsDXhpQwDQYJKoZIhvcNAQEBBQAEggEADTrdvyR85I7Vq+f9+ow8LIO6
6OgElCs5YeDyYgdYW5xpKbd/WKj8IbrUqN7ucVaFEyGjP9Iruf74Zw+MA9CO/iZ
SDn+UiblU1WTqtIwX/4m8ZIkEkh0CgcBNHJT/ZqIF5WclQKqvxJGGUB1BQBhJSD5

```

snC6cKkTedQBfJ8lGZT2ZmoX0dRLABvo/bu5k1h/5FtQibRcd/XGzIeeSSTsiCS4
8BsQKkx+mBDsEAocallzHA1Kmm2fDwPwDBDDcGAV4P0nnzZWK5Zdo17pJRpg9yLy
OfUh/w7EqPopX8bHRQuyLIoFs9lZNgMTcGmIq7SL86SfkClkJ831EXg4zX6D1DCC
FG4GCSqGSIb3DQEhATAdbglghkgBZQMEAQIEEMJrzgXD5KffUisHbSEv+TWAghRA
IybXhr1TywIGr1n5kLYPD1+FGUjGaKjKFAIK6MYGZur6Ba9G9y467ayUbv5tpU+G
EF8VfYFZG5o3NL809/9vII0FGlujgRN+t72UvIBuZTDMdP50+imi0G0La93BNdd/
bz/9eWFM/RGEIo+jkrdearRJ9xeb+Y755WcrvlyQBhgDwlTeEAdDbTj/3GFGjIYO
jypfQPFUNoFYhFLFi7QwrctHzP7qeLe64/i4ouHWk1ETw9vzgfXp3CuZVdmImuQR
PSXHpBwwfMnqQXAV1keUPqCifrNWkVgEvWGPBgLoC1jhOK/lbJUBpvhEiFtbcywd
gR7wg0LEsXe4zsEW6GJQy2wF8+L2nXAA3X1SCwpHPjluAvL3spgOTKd4tSlQ9f9e
5DeRJ60FO4KIq7xG30bwBAGuprf/8KzSl7xVntSslz6sp0YLk2OHcD8EC9ZkbcKJ
w9qH487wyqWr10gAMrxWyac4OsXJV/SfFvYjBMGpBrZXYPYO3Ay0ekLT6xFf31Bb
OaA0hi/TNh0jwBpeEah+zgAyUYlaofsYrQGrJBzUmQxcSWATGfBtAvHGc3EVFSh8
gyP6B0k8tm8vz6Gkp74S/3BhudhO818GBMLRQ2JNwhdXyBmwiTUwYKjCrvoWcSp6
CgTC6c1bSn9u3zwoenBs0pCarBGHMbL6TytfeUb5l1Dmtebv92C1F9i7x+nuOML
ZVKjjGTISnWJX86zj3bJRNQHNOj/dQMrGhnJmvIpdp2iayt2vR3yYTHIWMmI/H3d
yeBNVb5pU/RWt9AfxkSNZjrhEh/tiMXhawYChB7aHpGI8diS8N5mBGjvXMFQMtGqK
8oFwocldvtjpaLqmlYPRaPbLyaZNQRjrcFRyLA94WyuflPT6EWwIycB4spSWTriv
uN7aVVZwxis97frZ+qeaVt5lIRSUwmkliIM9bwq1NVVjNXHweN5IBVhVikl/sjdl
FtmCF9eKcspXsEKsIRvN+AFTVnFP7VQB9xY90MuBrqgwzDoknaZKzJs5BvD6VDqvK
4N5eFGhmQqWZaaN/Jgyk/Bg2Hf25146/wsPsOTdRmlurQ19/G5QiAggZ90dCPKJX
qdmu09Jg6DYckWE+MG83Q0gkoi5s+z9mZDtZPCIMU6wh8zRXwAFVNCi2oNE6TMz
WnLgYyYj/YioeKeYjgAXQeq1MOy47wXn84Za5XBOaNmYV5cr5MBD6heVcQauCHTM
ofotaxXaRsrqH77iEVsG3i0BaIagj6KwnlaCmy+xCMDR/WEIX/t1UQE6KOhNGHyFP
z1GaxlIHlfnjWweHoyFntunZrY1MbTnmaSCA+xx/ii9lg5urxqhrPEtbUv49p0Bo
CeSQOYCTp/Yla4j80bPDB1eno5riUPDzR4UNSLpQ8Fms+qvJJK5e5rsA38Z6pOOi
KZFlCOQqRw/loZgoiPEhYjnymM6wZyLeZHrz2NJINyYB0ODP8AG71xbU0IFEB0c
orxAjpAS1giNwHPKdoSdCAwOainwCNvDuc9XSHH//sL9tHQK2o6h/USpxeYK8weC
cmkQD06rqzZEXes2ahfuQo5hq19cSBodUqq48KBGLCF10oIIQkVw5X/PkKX/MrYk
u6rk1NT19Eg3+HUXfp56X+qQvx/KS1ClqRzIZrq4x7p3ANQNSUM/C5h1stMD7Q9L
WNj19BnTAJRJLnASVzBjn/Tvd1D9ersXGjwpzPe8fAcXJWfPP3D6gsLdNP3imiac
Etg6Vt6PjtvW01jf2Gq81Zu4GX3SH15n4jkDOWQtJO9hEG16PCx4zT/5TqdVpYxO
q8QA0QTXjL+zONDwCCgL395n9zW1VGvJ3HXUCHo3vLrRwEazmy1lJf7z9nsGyW3
O12kMeLE9ddPYavLm8FCQSDq0g9W1w2mZDtRahx66kV9WtOXJdCKU1LPYRr1/gVH
VKpC0NR5f/WNB1RcbCyFb0TqGVirR4tletjdUIbdY2nRov7PIV5hNH68WNS4pJrs
ZNP1iYiohIvy59OyBzsz2mQR/ETCquOf82fJCXRzZ0wphAdXO2oy2o9Vky/njGFf
Fz3Est1H7Z7EoyLkj5d5F+74a+1hWzShS4mw3aX3LmsNq9f5MWC0TuwzxDvSGPU3
PeVLog/vsCNT0fhrWold4Eazc9FmTsyVKtWgNopnXrDO/neQpy8ipcRzn+k1pPmY
5g0R+BohkWzBP1aIWhF+b56ZL3Afkpqw5q5LkXmHCuS1YA6yMhR7govC1uFoGJ2c
dP56jqn9y84MqKUMw1fhizhxTjvUKfltpk5398zwQTx2yKRH4bThluK82EFtnNC5
B6E7uTGHX4/x3nz6Q9hLf5zmhUdFJzo3bh0KZx17YFBEZMHFqdcv9jXMBQzy6aUp
qVav9IzRx7h8uYGUwo2agvCoUCuBbujuJrm1tGy0Z3IMxy1w0KMktkL4Q2uunLzm
MI9KratRPCpqZ2yY0HoGoOUJUmu9CGxrmYSUCWZVdyMdGoUMPUuc+7hqqSvChgq
LfJSqyYKk8TQXVycB+Zq8Q6GI4PogaorIjxqenAMQwqsNziX4/X/YRWSzaHf8PNq
uHUGjv57I36gU110ZKbsWrVTP0C2/DcilAdhHyJmynoYDpFkMMvmgP08A66Z46//
XTAtEipnx3Mp7KX2D2M8UyYq6h8c6y10dPvgLAB8ZO/Ji7/XTTy0z8hg56+Jhs1v
Tcxgk72593Vqy9Q6Pqvbe7EiT8kAes141P4kj+D1TJ0teWoc6dbndK94cE1fE1oO
S7mlF9RiDK4Xq71EbKn5TINq0JsvBv2LHY7m3fPSMHAqrX077CEoy+Xi2PpNL+45

k2g7mTjU15dCOYWuXF/Ma9RiggjsR3fJ/KOulIqAHkG402O8WF87Ku6wNZUy4bE2
 QJY1jwwnBwej2sMjSjLpr16fzvPm7hTx20Og4gMZB2qTPtL+VcQ8oPSVUWuEDuAV
 Ds/pIMaQUr9EMPSqQumDXpzehQMe4FGaDUu0AF5ynuTacYKNd0am8QAA0mT+zB7S
 3Oml76opyuGSbkVqff5EpOqKZzk/QTlWFutby/3y3mn4qmEQ5abz74CYHVuFcQ7b
 vcYDhrhgNqGnMVqADM2LIEy14+SW1rjekytTOr+I9s76C1TG7wu3q4e1efZpGSjm
 z8DQG/TMK/pRF0yAiFk1PtqiD/VYcUxPQmaPMx6MulVARgJkvedqIVJcTF10sMIy
 UJPYGI49Udgb6m4KHK7Q3g8ZMf5eNGf17myC6mf1/PMSmb+19xI3cW1De4AJCrLn
 eiTrLL+kPYbsDjJLzwYAWa1N45ogcCFdKbRtVR6G4Se92b/CU/tdOEajhjl9lFCm
 pR/oet/vj8C+EH2wgjBKP59YwVTQyaqknZQxhfQIZINT2TCwLF2VT05qGU+TPhTm
 UDxOgTObCpELThELwI8D8DHHV9VTrE8SbyuBO07+/6B8m/Qz9NgHkPIpc8Zs05XQ
 15fzm+Ck0IEvY1pc76oazSqN2RtImopUnoB36IMZ1TghD5O+4yWZTAFpd/L/YNoU
 O2tqE+hiZ9/08f87g4jCGGNBBAEX+wiGUUkt38ridgrmXvI5Psa6LM4Fy4p0PVBN
 G4YoqMypp/pU+CeIlyx50N1v4HWhgdkyHN/twWTJyNGESpVjKdlsXmAMonKrJZGg
 SSKYMb0T4vxG6PjT6Xg6F7mCZmMAMztXzaEAUNqjr/1taVW+RplkzP8JvOTGn1
 zOvt3DkVWZqvjXjLHxEptCy2ja9K1PzvwXTZ1KotdAdc755M41I1P0oQSHLCX15w
 WAjyfgmMQOnpsK40K1wVLwvOW37vKxmh09R+2BMfNRdnXtIO7yKgeY1qsZrgmAzq
 nGTXthixWwsW2OHKLeZNBw31h16k1jDm+8tWEqe2kYVUVwX0VRVHJE+zspuhsK38
 HVt5vCJERCyXRSPYZmoUjgRKY8LpvzJ6U2rv8k+qo6FAIGY3o3sIF7baks05BM5r
 ME7dMGsPTppkCNLJZA4V5JM7lZAwPu0IsXvIeNQw9EK/Flo/7WftoAQADZ5fLx8p
 9XNA+/ycwSsCj6a776f0kfoL+Bx9bA7FRvZk3VY6nxT6USrcT4vrsYyANLc2xVDo
 nRWog6YpHLv2TtrLCqSqfltbeJxwHEez+0P2MDhVvJYpEeiyZdAAvov2YOF+PHyy
 FrAUaltnbuhem4aHs35aaMMmCGItXBV0/cVkW9dJn++8Q0ouM1TMBzFgEKdwVZRP
 LdP01nDyyh07WJFXK74f5y2ila2gJAVrg9VsuCuegKCMmB0SoxJ+10gF19H/F+qn
 3Hrx36LBy+tBj4EcRJS07q9m35hmZIRhE2zV7yfnpSYOWEHXsVxeL+aanx1dVIZ5
 D6oKjPH252uV9WKZdbvRgPgg/13gLAGTGXvPbPL+EwYeHZkDVCuU234159t+Db5w
 orVZheue5q9k1V0SaaNu+JawzU9UZg69m5QnJ9b5fyAMtAFVNVNlmtzZsonY0ovj
 KX6rj76Y4NcLjEKXwJzWDGJvZHV7D4KKgK+ptBpudlhAfmw1DWH3oFP2ue1z/262
 0sUDU3I3IZk2XDKbPkt1Z/3+WyEpbG+MSKeSvHKEENeqlHpRK56qBuid4QyfuH01
 cWgT2D+w/Nx4WQcz19h4LWYBecrUml8Wo53DQApeLJNMdUzNgeKKOFxs3an/y5/g
 NEJT4p+kCpgQfSHJ8sKujf0X8/HHoaxfH3Vd/V2wZrYCVf5IxECQ2xy01lorvU0w
 YbK6euf597puiFolZtRzOaSnuauUvVAQNthTwfOhUWsw0UC/i+jaS9m/4GkIoUH
 S6zPE7/w7KBrEne/4gtqgpSORO10YDnxOGNIFOMNUjZ1z1IKASa4AuU94hYtnix3
 dxg6Y2g/v8Gue8Z+RkoLGdjzavyu0AVgz502eH+u2BalxfpQpbQtVFxzEFceHDLZ
 w1IbrXQdbtoks9WVtqjVStiX1Yf07JQCK20WeGfaVfwvmd54VWBNypXScelRRhZ
 Ek0uX5FGd71159FYucHQ6TNPbS1fptvSfsiaCqPzU5Tqk1XLBMPdwHrJQU605usd
 T4no88uZnm0WE794m7CZ81ZpxhluRB3Dp67znf3gEYSFpTvtRvhRc/e71BBmPWZH
 NY+bvMfrfnWwgkR57Y3wrKLLMcUfH/R1PcXQ1KbLA4FGkUUvc61VW2u+wfHX1xX/
 s3ht5TA4CJ1tubjVmaSFViiFQDs5BHADZHVMSPdmpDVjogtBRYnDVNqIEZPWqdyA
 eAlLLPLNjthzVWmnWF8bBew5sWsj1V5aw+Ly5tCC472KLLM+t2NcVB440cBa/BSE
 p/vh3TEsoZ/m+UfK8EGLqNVs2vPZhuVW9i19cN/5ALp497jj7Pdq/LY19x8VIRjV
 EqbFPWIKeRDcBIvh4R1+0Z6n1HvILjv1N1NABnKqHwfjCQicvOaE701J3QWWbBjC
 dtOkxhC9+gBqDlq1YWgwbEzDPcFVzCmTPH9wHhshcmp2507lxqSxONeNcGMKy17
 yBSrKmaQrlescqLJLH/yofTni7sb+xeohrz+YYJraXlcdLgSK6Bzpf7wpWhMB7c
 Kyc5T3ReUPHrm8RIcaccjIwgxyJ8YW3iCpH2s+vdaJnEC1Aa6D+53+0aCFg0/2g
 asqTZ/iLws+bFux6MrNs8cohuvTF8Y6A/++cp40kp+PtSN8G7+g1CmkdZZdMg5u5
 9J8s8SIRsbVj3y8eH/DSWGQ0gMc+NYLaWBXNReVPndwWP7aqXjLysuRAVVgOfvJa
 zrWfU3JeUphCtGTh785hFePHTZ5IZBw+DaxvWHGX/5sIBokYH9E61224r3ikUXU
 DApjB42XlCyo386TU6OUzFE8xHaJ7o+nW09t6sWY99M+BYngsu5ghjqIz7EAZjU

```

BEB4pDKLcVf5tXVKSOSeIA/nauOxb8y+xve2ZkY8UARMwrтт7mqgqYgB6/gLD7Ah
Rw/Zs0+oQiNqv7XTY9c1U/FfAQ1RYiiz8o9fU783ccpsuw0PcgtnHWqyrw5I4v14
fRH0Iu+dI13B18fbPQnoVJkxbLTvG9p1aXf4fKPPYsR1zjIOSFSqimx/ogkNjlaq
4eG8h+lcyFIT2fmz4Pek1luASudAGGQn4AGPu/d9FsM6LJv01oYzcQVI13F1ASgz
Eo8/ks2dfhjeiMfHkG15aFybZAm1f/sEtbUX5rCGkf0REfa17TC2NpB+OVSIJKI
V8sLYNVsZc9eiBJTli81ZWUPzNaFtyk8zRcmd1OzUIvpESNve8x/USztcqIpMIwX
N2mlj8D1qwnFIOgqHEoMgWx3Dm9EMD5xjgCA9f1Q9dkD2WHVv62DnMUnSuYH3NKi
4fZ5EGXTNezry4SpXmgLiEOGpiXz/wSLP+/n4RvNfJ4DE0D27wiHchvTAYW8IJgo
9uJU/KuVEk+cmUVwAbqWimq2XpY4TyopHyVjSFy7a8iaYs/sd+u2E2EEfXiyVra5
UsJmo/RdgZSct0yLcYAKsO3gpXW1KSthrAUFYbSDlg7g5nQ9y2JyLsZGhjm+c1/I
6fEhOucX0MbaqMwP31pMw8LUKSKodiMXS+OlKzALyg3X1ObR1yK6PNK4XWs7L0+
a8nAdbRwoasr6SrenKYuTPkuRhLEkj0k+V4B7ilY8xGYuYjiZkxYxpZBwB8AM07m
ck4fGBGOOYdaGhraRy4DImP8SzVebtEj7i4wN7s+fHs3c8d7c6QuKOJhicyK6Hj+
spm0/oEd8vsVHieyu056IHduU4aeDkVoTYN2ks7itpuAv9wMOv6It2r4fob/aRSx
ExuZeT+RW/qnFpLDiUXa/z5VYZH32Ea6W/MUjoLc6VqzFGScE0FKJte+XiasJ8BG
yLuotJvLI5hCIz8gW8M4nSo8yly9VeyZ7Fn/DLsoJ32jQpYmhUjKjtNzqLcq6Wti

```

B.3.13. S/MIME Encrypted and Signed Over a Complex Message, Wrapped Message With hcp_minimal

This is a encrypted and signed S/MIME message using PKCS#7 envelopedData around signedData. The payload is a multipart/alternative message with an inline image/png attachment. It uses the Wrapped Message header protection scheme with the hcp_minimal Header Confidentiality Policy.

It has the following structure:

```

application/pkcs7-mime [smime.p7m] 9665 bytes
  (decrypts to)
  application/pkcs7-mime [smime.p7m] 6148 bytes
    (unwraps to)
    message/rfc822 inline 1923 bytes
      multipart/mixed 1818 bytes
        multipart/alternative 1132 bytes
          text/plain 375 bytes
          text/html 473 bytes
          image/png inline 232 bytes

```

Its contents are:

```

Content-Transfer-Encoding: base64
Content-Type: application/pkcs7-mime; name="smime.p7m";
  smime-type="enveloped-data"
Subject: [...]
Message-ID:
  <smime-enc-signed-complex-wrapped-minimal@lhp.example>
From: Alice <alice@smime.example>

```

To: Bob <bob@smime.example>
Date: Sat, 20 Feb 2021 12:08:02 -0500
User-Agent: Sample MUA Version 1.0

MIIB3AYJKoZIhvcNAQcDoIIbztCCG8kCAQAxggMQMIIBhAIBADBsMFUxDTALBgNV
BAoTBELFVEYxETAPBgNVBAsTCExBTVBTIFdHMTEwLWYDVQQDEyhTYW1wbGUgTEFN
UFMgU1NBIENlcnRpZmljYXRpb24gQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIeh7j00
Boq0MA0GCSqGSIB3DQEBAQUABIIBAHpthaX3pLTY1dQEps916ELlnCWjEQaqMJC
b7U14ds6WpbcE+m7YotdmTDc6sMudcq8QWt13YfuveYJMPp88TnfLOJlmxvh16zM
pBvxeDudVMaV5AhRsIDeZy7XeJbTUQbLvKNsfYaWpzcFQgw4pTbSj8adkH9ktJn
BpOb9B1gknnHni97s1F+6wc8y6UC1QmwsV6M3rFRhdx/QIT1Y+JsO8Za7ByfwWzZ
8mgmKCW1WhQKtZUZes335ES6TFg/rXQwZFC/g3K2gDVWQJ2KOG0Jfd+3gV8UhG7
XGwzJHn2H16D0+ryfmLq1EpdpH/n71xL0etM9wJmyXGCbxNFODQwggGEAgEAMGww
VTENMA8GA1UEChMESUVURjERMA8GA1UEC3MI TEFNUFMgV0cxMTAvBgNVBAMTKFNh
bXBsZSBMQUU1QUYBSU0EgQ2VydG1maWNhdG1vbiBBdXR0b3JpdHkCEzB8R0APhiY6
HGLS64Mv1sDXhpQwDQYJKoZIhvcNAQEBBQAEggEAldsl+A7Bcif3coy6alu5rzdj
R5dLE46I2ScGw8LvTkWxyGnpR4KoNvWvkQLJ7kbXAYLg+Td3abYLDxibz4s9DqLs
6fMo45Sdrifv4TLZ3JyP15Yc/ZjppjWcF6h35foI9SPuGOSeMD6nYX/d+Baa0Lxlm
ncAHYq+KUWhmWmkw2xDmiY6QWQgo1+Og+XLtVhbgjiHGJ+bYeVQLuQgq9Tn1vIHi
8EcvqA61XaP80AOPS1TL3Dph1MQaU7yEySyasiRSVLYA45iEA96JiPdLvdneG/2D
cLzhkZigGZHVvH+ZpPnr33S8BcTQG4W/ZHLwOmNB/To+JnAcSYoziSp78qs/7TCC
GK4GCSqGSIB3DQEHATAAdBg1ghkgBZQMEAEIEEPEfWWrJtkXQczxhrK23VBSAghIA
aS2akkBo7J9AikHtSGeZno2vqidZXAFA44XYi6sQysoQgX9G4Ovywjq23qbXFxShT
d9JP1tZcoL7KXlyCfHN+ua74NUetNmykoELZY5A8dzmPkdITjZcUEeEYiWk2t+b
VVP3qeCIFmL/NVakIF9v8/VLns17uzop/bx/lbV0GRWkc1ipS/75ZiY7jzHpognM
/6lpOKEJ6DYjOUJJS+cY7SfDy4dVudowyiFBrEPeUXJKUe95R4CmAiByVnO9oF9
7g3HiiIEJI3IMjNGTgloeXTVINkoniJu9FGRZn7W84mZ9J6mPTjtY1vSCZ1kr/U5
eGA406ZJWU/y3ziJdFRdhQIScUjg9GhMoyYHTYfAR4GF+HTbNHt+eRj7pqWRMetq
febcQhuqnMMiossq6zTSnD5ayVbKeDJTdownQjdR+Cyg1L+AVM202L1ZwziW/Yetb
57/s/DR0KNjhwRhUYYNhQ65g128BoI7MuaySnkMAqtcmnmAfMhQb4R6revFA9fLU
sCuN1MKsFmFKgSjWNWbBehlofCp3gvdApXBXMwCNLZLlIprA/n/6uzTe3+EsJeX0B
vL1YVknAS2/MwbbObqi jmgjR2Y01+h1w0UmXDTG2tFQkVEHcaEQogZ/Wf5Kamvny
kzrxGZBdjUWQ78DOWhRWcmczTuhYR05IIB31y8r8wUoybRydT13EdRCXy8OC3PS+
EeZ7H/2Pv0TnQysjAT10P8LGLiirG0LWmR0maQ0Chr6HU8o6mQHyaC5J8Km3mSEW
7krJmQLle6ecYCPXoi6Asur22Rh41pyHwo7dgtvuKf3Ipp8KQOSSD1gUkC9WUTPp
qP22f5R0JmE0tliw4h38DSLQC2M7uGmByCzXw4YSggF2paW54gQuDKEay2291pMX
mXIbUTdjVDymXuOBv4R+t8rcV09x8YY/DbqoGBKkY8F/SVrx1v1ENIba04MEUelz
xUkP+1cA6KdrPcTHjKAhR4Vq35XANYJG2T4NhF/kF+09OQu4UupdB/wgjJZ22HJc
XjQ1Qxs+f/Y5KsPNdkax8lDMwf9aSoSQIpyPm5PqjjHxnA62n9ho/hwXnk++61c/
3ZJWp0ck0pBqhx4rbQwMV1Pc08z17qYK2LVDPVyhkvUB61EK0gToOE615M1mmUUE
PC8NtmwJfC/rXFRIPi4gwYNCqn0BmGB91hXWTrmbNVT3H+wfshod48QuVr7bsvOf
DAmtDZM6eYWoAIUrOgoAg8OcZ+sPg6OzeiMoe7VENTa3m0whXHOTLxGoe4VBtQuz
52PTK+4a26qICcnAZvx4C2AgEba/sdUOKDjg53kR+JsQPPqS6/JVuL31eki9tGeo
UXnbhk5ZKUBoXj+nQCWwrKkreNEgV5fsHgeZwY3/+FXm6pk9qQ+/f++S5cnBm/Se
iSvKwKAXzc9/1GSgM/EKg6AcX4/4Lyr7sN0tIyZhLaVfsceLAU1oprJ53hED4HC
6E2CuFl1F+EwIYAaTQGPavujpFH/IiGy8qbaY9foYb01WKjnumHdyoGwXHw5CdFf
KiF0zW1a6fjvM+Z41d7SeKv+TzRp/cjVr7GavRHEp4EG53EQc9CIXLqmz32Ep3ph
kMsqm/REp/VswvSaog/YV82zgdK3POhTESIAB85awrPmfj6mCOu+ypnLYuuu8mo

ZNqIleDvcofTguOy68I5cI0QGLog8915R8umqPZcRPPa8I9gotIqkvUyPQHczLfL
oIHnLKwe7WxHcQQdbQaKuz5YAIewUz3RF32g4qmr6d7C/MdkDGQi7+d9+wp9wbDX
L5k1lauXJDzsNiz4E2wrpOrzuRF7vMrc6VoIxjSco4gkBI4ANZCwtqB+H+Ci8ubwT
VUQ+jDIpt5q50EXMEfqnzdpQ0tBWgWNS0b7vguUec//5MzGNx69rnaw+06zkjzri
JEmsTZiXPwEhse3yu+N6xAHtCF2/CiYgeHio0/toJyXpmbRsC/MaeWtHgd07KXj
LMC1fnTPUn30/aRl7rISnLWhcezEHEv0h6lK5mABH9VI/wUywoiQuigl34WCDHa/
Q0hpUc/wC/rFsHK34ZWSj5MAKtdXacbZT2ck8yK2HJRPdallzRurZd/A+rCebXRn
q/yHz8t/NYxz8n0mGRKpWu57Sdt+eAsT7YJ9LaLMYfSd7cJz0+8rhJKXxU0eSFdr
NryDjHMvoN5nPle7UFcu1Rt1kfjRnTyjw92wiTxskGeG4/HLC+Zlg85YMXq9thhh
+gtRMVrVsahRty8rRLglJmmZOXYT4/i2e+mqPTOgnqCABmnU1CkAvfwo6QVAYpTN
tTjEbS0dQ9FBPppQ9Npyv2bpfPJPiF0tCTC106JPCC+73agjOyQXT9zHvcQibNY
WWsvh7ri3wm2RYEbsheP7cPePWS/raFKNdRIRBcyqTcPv3YIBgiY229EmJMXagKr
3Z4KqNT6RdSwrLMOxbdHU88yK6OMRMfHOGHrPinfT81j0oiw4uxnmFnXUqcWXaC0
gv9V1/z8PvQE/YgY9uQNwbC/UEcZ7GttnxEZdV8nuJeFYr4o4/wJAbVcmKWh8JM
V21ERzWoTjv82VuBGSRFQw6K1LMUQAfZF0q+hnLzdTBjT945GXiRkeHcENxscd1H
br+HW2bg56ZTVerczUKODuHQvtviQR19oV+7uWv2BCEu5SwM05rEOjwKMIE04zgKc
CWflGpudQKJXXS24iQiMzPU3ZCxokBqgz/eQxcQAPk1rFcmGJvvGj6oN+FsJcXtR
jAZUnr+WumYu9f3GcTmlemtRqnnMALVzp0sJ6XDMih0xhey7UCWDwodz0w7P+IhJ
J8M4vvPwj5f70sVx29lhsZV/hBXG9Ww8anSrLLHYH6+o1tiSHEQKqHFQ2MdPwzk
xNtcmpGHEH9TJtiseGbU5hsYZeSj8kbphKcHO6gL76h6XXOcZpXNywo0TYTDcH1C
BNBIdskqGrL8gd6IoeP7fjG6f4syoeYjWhCM3RXgR2tNamFxD1P1lQeX/A1/WQ4p
p5bRSc9itl/uiMjVM/fic6eslIJ6XOGpYACPjqrT6JoCOEP+e4fHW5tjajaVbk+F
j14aZ3e5/WvYwJzkUulTuywdp79Sejk9kil/RAzvSOS2v+40sWXoFFdr1TRtaz1P
gciRlOsTAT5y4uiL8Yi+IPO8SjS6C/mbpyAfIFgY4cWu309zo656GN1OyEqBQSVg
bVYJEXEj4dcix2Ll0MbND0JfiFQz+pmyB2mPGYrktDG0hwV2w8CPHCOHEg4yCV8R
JoZRLQix/6WL4mTly9dLsMrurQ6CaQCauiu5XUB8P+aPVuilxOWxwVFWdEcX4if
Ns57xmGj5mNaCjSrtaw++043bfXR9NQQR788cReltBMTdEZCVZdQJ6/K0idnWfaJ
CYiz6KE9KUE2phZxbq2J7Rhk06dq4qicFYZAqvJXsxbOdJapQNwTm1HOE01MA6aQ
uOgZq8CWvVwZMRE/KRF8RPYgrhVPnfB6TVUoTdGRLgnz4S7dqwz6q9H86Jd2Fz86
W212B+LIrBkZGWgmz6QNMT2g85LaC5GnqMLKwsSf/cBUWc2rBjwYk0xb8jEmZwv1
8mz5o8YJN80qMAKyeL4aDF4naa6RPOBUavZP0bLAX8YHasz+85D91RPSrNwerI64
SqRRS20QfQCAAnpGv60IAGbLmNn3URRIcRfVP32kgLqk78cuSxlg/qhwvStNbdxL
AkISRZdOd+aJyU+y1JFRFmcMRD1Def5gTtJ4vWNKwiThJy5qPyW29NkKxBkU/6F0
0Z90MjUznhx6v5DjHlUxKL/v2OZ8VB0oDfcmIdu6XC5x93NQoJmyxP6R8MHHiK+v
WYeVVPPhfWtDbfd6dfKhyVyXO7w3pk/8tpruc2vwoe0uG/f0fCTUG1xCpCahTGJYg
Dm1/+XlgFWJUthK9NMZ6GouF+DJuNKaBBxQu9nJfHmUzz1+4eQf1bcKgQk9BjuV1
1zWXZeKNGXRIEaiVO2E87/m6fqR9Yv9me9QIhlmVvt2687/eoV5CteRKzL7+RIW
uMb91NatVjX9pIv5Zz9W91e1wJasoc4sqKYhG+GNuV17cir1xwmjJcZD6rHgjF2D
xCDwrKPudpbxkZL11eF7QYzvqylmgQx2v0cbshfIEovZAbD1zWzdqvpJSrWEDs5q
sXPdN36TAWVF5Rod1fueIo7bv0tCGQ0zrYu4FHPDLe9a7uGWZs4kzAnQBSKGRJR
METU1btRmvybibgU+8/Z5JA+4hbQsxjGAvpwLitXcPm05By0dULQBdR1HXXMd2mF
Q8XuScWoGQDxeCq0j4VGGGAUZEj8iW2UyU/q6vuSfzA3TAM10cu0dz7/WQxdqw+g
hYQx4N26R3DG4c8B5p1DcEENHzhBkkeEcAyCq17jDpPqwdxxaSHM4HjCibrgD/mN
HDyVgwbyy+aiucg3aq6EfpZHM16DVA+uwHs0WN5cqByrJqAiI2AWa9/rCoixTtkZ
b/hJbDIX1NZ2b8s8wzZt/MOeqyMRaDuR3LiGuX18y79ImYk8qr4bAcSDs15z1GHs
+5Zuzs3K3MEAMW2ff9c6QUmfkMGmdKtMtG/hdiqFcpzbX1SmxgZVw4EM1/OLftTI
Y/6k4QuE+TXh200eN15VGEMYam6+AMjWPC9u1I/AtMy5y3yOcuouRXayBqpfy5Hg
xbxzoQGhUqg7P+PnOMPjUn5bQdbHfnbyK4kv5sGQrocQ40qtK7VODaEx0mcm0wN3

Zs7jzcVxRC6bzH5yxR5zDdqzsh7qqXhRe8OZ4yhvc4mokrQrswiIL5kFt43gL3y
h/cI1Bp4KBX6pqf1IzoFuiO9scgFVRvtHygsgQ+UqWwuq8xMWgXFahuy9jrrPULs
iV2hQ96pKCKERQkt519xMKmT8/w7neq5rUuyOtCgLCt/E6NMwmpyZv4F8BQoHeAO
69PHQ7dg2uDeKAyy7szDr7EPA/1Jc/AiRRX30ohPEc6xqiYFO6U4Mc+Wrf15oW/2
SFuh5+2j09W1y0XVMSM9vXGfb1wiIp3QZqWUfavm4C0NyXLjfCkNg/M/rIjRFJXr
sToHAyus3wrRT+UVN4ARzT4thfejIx65026NFyAE8qeZNd/cgqcCLOzX0Wuh2uI7
opk10J2QCYuxsHHQf93VcnwuLhh8669HdPTuInw0poWzmy6nUTWifZ/MXlqq3WcG
E8mkjQps12vGJfHPAsW43/cwJ83dI1LKzJA0XHaURU9C0yvb1aU8QO47t2q6Ne1
FLdOIHWGPSvBbhW292F14iT2oSe3CQ8QjfKRW3686zJm1sJJjRLL1JtnAUOeIyNX
OwXB4pb6m2emOZWfp052z13bmAVc9/Ja5Ikqf8pCgLO04WZpKF4kJ+7wuoIbwfsx
mu5aD3C3+wzRZ5d8KEDuLGY6EgtSmhGw3jBUOS8hML621YiuqAwiTZL1mmXjcmY/
nB/YncK44CBsJelOJyInx6trRM9Buwo3K9+U17e/QlZgri1Zph5InrB0d+v0+bSy
iqYu31F/1AQp1xiJK3siURdEUWXYw10T2qbHRhJO3MwvSi/1Hz2jFdl61lJTC40e
vBPFwOt3wv47assyifSq1VL4wKGkpn69kjmVwQzFBO2oSxoYebosX0v1OCjHTpvz
Eg6986NX5P7sXt9LlQ48xBmrSjaB2Nmh2Vwhxt0/nNd4yyMUHpaGC/Hht/pnU1U6
2fTGcQa1LOxmZT721b8OSPkt6quz+8xDbpX/183FsM9Bt3/m8x7Nxx6HRHj+GVsv
Zo5epA+EX5gQNZ/EFg5FoNUuXu/j15AwMF5t16XoLxuyjdIdT0TkJ2/fYXqAjmWq
IV8IaPjpiJQ8jjuePNean4Uu3UI5d14katc8yW9HvTd3ANXpAO6Jz1+ujhPkRsd
r9xSuV63fNXg60C2wrFU/B2E/rAf7fP1lZlatvIb6AksnwXHaR2+apyI4tgoBEqqN
eHS5rqqR/vtEAYybOrz5bzQo9ZLBvqQ6Sy6ijaNbJJU72OMwLFNHdTuHYpeMcqy6
RhLORFX/OwyRecOYtyJm+8N4/nmea2gg4bdN2ajET9GXbEuIwBLUxYEPg22XIru
iC+Xqm7E+vcG0DynGLW5AR2HVRKnNFEUerCE0Mi3lns0tbp1s8FH8cLIEZpU/6Jr
4+A711E2aY30HIbXcMhGVkFRFKawZ1lGSY/3A0/zuWcPLRfvfI9iIco+73fDrXwg
CUg2KoHBh81rwmDzx9HBETHByO++sY+8FdYPtC5EmMHS2gICDSfcmiI8dC5J2bla
Zfv2s5rw81FMWx3IjmAt84jPNjFvXoCmlbWJnhX2YZP312MzdVrqlRQWLSZ9eQFp
WyYA4Dohp27izdz8Hk1162EMEsyjumHHdF18ZuY1GETLFyzJcJjJb4THJbi2S+yp
Z+83HZoTX9OWYh2M3/Si5jUuVxs0KSM8odJDNE+zbRmzgKLih19EWkFRaEPGld1H
q8uMXq0CHByd303MVR3z+WPQE+tZoxHjhtMVION/5cfKTqO4UWVTYup8pUYa8Ea0
4RvH1Dc6V7HARTwo71ai6vm81p3U1oOvVqIX9j7mx5+WlmpznM7KcIFCIQihXANM
Eu1/tbpFG6sOGvVachsz84P3laZzuFe6i+gjlz+Xr2PjNgshZJOHzLtuTuWsBMdw
12AoUC0A+icf5564zgsyYJ6I5iqKvFdL00zoVMElSpFqCdEka5IHYfPn1lwsAMnp
oqjcoxfwoXnDKxKFjS2Qhae3Iqnn64YDxCD+gtxHPe9QMRFQvbM52yPxLGPwrayr
1YhDipe4Nh67gYRaNdsmG7hnVA6z1GhaEypaP5AJ+YsuH85cMV7Ck11H19JFcx31
7ZCjw5FQGX5ThOoZBJeEp24yO9YPRnlo8Sy9gAhIc34ZeBoFfx08F1hu/Ii55n+c
yme3YGUazZHERIP8TwoSes0daEXzSn8oGwWspXRP282frfyUAhe8W9OU1KgLL2FI
bRZiV9S/F/QgoDkpxo1T2z1rMoAsOQ95Oy/9XtNw7ywsbLJVIVXNv0KCK+S79eIY
XxCvDW1ZSOLAxZKdstP9ZziAqkC5bANpMFZ1EUPxBSJCBEblcav4k7NV4fTYNQ4V
Niy8WS/OUMFWZHw5BITRjRx2bwmvaSEKuPPiGtZ41QV8j/jguZyZp5oH6pkG1C2B
AoixTqj4y9w6DbC5ruYke0o0px/nkH5V6NHGODzuyEPtkmYVqMezNDnx1qynqV
QrPZivsHT97MYbzj58Y8DbTx/hBr/uJ+ya39MsR+N+vpBV9t3ubM9i91906akTWg
rBbNwdU0ayLOR4q/TlmdYVmSoc1xDwVe8kLD9vMiNcwobOkzxZK7J2Qc4cromDf
8vNMzjYu4DmR7WaE9wFzUk3FpixeWJrhJpdNC9cUaZ3I6y7RSN143mKdZF54x6JX
AnSbwNE6dtuLKa07MutWuq2MbBDQDIGxattEmnRniwoJc1KdQYKtJM4MOQVfUTy2
xubs/4wjiS+YodmH1XGlnXP9N2SmFqE1puBJ/5hdp1BOIFGHUj7Kura71N/TqtRU
e3NjQlC5VbbYNIxvcSsKqaVBLESsxgEuC8pmJ4N8FBVc1WekXo5kn1NevMFJphMO
Fv9gyRo7NDCNRY7oY2zyzhnoZN1FHQR7GxaeKseOtNc7kM1QMeCdhZ53wPyLJ81
1G41CHDePQ1RtI/Kg3foHyNG4bvQ3vgPpT9s4T5fX7GStMQxh770i2Njo9OdNidJ
2+eyVuGFwwP4PNeEqEYe8iCGygDbGxh/I02zBPV3UgFpx/eWWx7Fwm9VBu8I3NYT

```

OOAtTUBOKWDXFZ+02uwKwZ1+Z8FBStB+HLuP5c03Iwo4gDkWWbSP2SoIeDnC0fbF
nAOHb8tHJ8AbeJGcnaE23nsgI+da162PL623w72uvK6SFvPNS+q93uPxNNKmh7Lu
rq0hQiaDtBSgSYRaIoLuA8Cuh2+K+AUKIc5mnC3VRpje/QqISgU6q/3tQ7LE/bip
qJONE7TiWw8hKRhOPgqRuLVDpYk8qaqujTt76rVZwY3Dd0rc5bX1jTp7YZpMKeg+
3YStlo2zgFrMc3niyYZsDPoNsZhXUJFLMIynBQO3+HpX1ve8WbyKJ5WqkS5E0H18
rHmLEJrQ4PYsu8yFosaRtFDDMfWA+pYSgnHSw9VAx1XS4Fs4uSPbprbuSNo+ARpY
PlM97viQDUdxB4co3vcChQYRv+j5fzxEOnd2ceKTj9XJ3Rrufra5KhBB47OOXxVj
HNp5W2ERPEBIRszF3p2J/V1HqRDd26MrORwfpZ4r5Jmv91NxKZyw+mnZqm+Sf0PF
/X9g5MCZtCrPWFH1AiRB8S2XUvbQMjh2c4BWPExc1Dw=

```

B.3.14. S/MIME Encrypted and Signed Over a Complex Message, Injected Headers With hcp_minimal

This is a encrypted and signed S/MIME message using PKCS#7 envelopedData around signedData. The payload is a multipart/alternative message with an inline image/png attachment. It uses the Injected Headers header protection scheme with the hcp_minimal Header Confidentiality Policy.

It has the following structure:

```

application/pkcs7-mime [smime.p7m] 9620 bytes
  (decrypts to)
  application/pkcs7-mime [smime.p7m] 6114 bytes
    (unwraps to)
    multipart/mixed 1848 bytes
      multipart/alternative 1136 bytes
        text/plain 387 bytes
        text/html 482 bytes
        image/png inline 236 bytes

```

Its contents are:

```

Content-Transfer-Encoding: base64
Content-Type: application/pkcs7-mime; name="smime.p7m";
  smime-type="enveloped-data"
Subject: [...]
Message-ID:
  <smime-enc-signed-complex-injected-minimal@lhp.example>
From: Alice <alice@smime.example>
To: Bob <bob@smime.example>
Date: Sat, 20 Feb 2021 12:09:02 -0500
User-Agent: Sample MUA Version 1.0

```

```

MIIBvAYJKoZIhvcNAQcDoIIbrTCCG6kCAQAxggMQMIIBhAIBADBsMFUxDTALBgNV
BAoTBELFVEYxETAPBgNVBAsTCExBTvBTIFdHMTEwLwYDVoQDEyYhTYW1wbGUgTEFN
UFMgU1NBIEN1cnRpZmljYXRpb24gQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIeh7j00
Boq0MAOGCSqGSIB3DQEBAQUABIIBADkVMpcJRfEa4tT386C5ia35Oz07sK9g9yn1

```

vdGRpk9pUDuldIeio6wLIzCtwl4TtjfxJ3m9sEL0KDMSszkV0AANUZwx1576jpm7
qEL/7d2D+WXVGAI560e6ihINfrnPUJmk6BCj2Vkj9918mX2FaDTtCQsVnrK/gDNu6
c8b8uJJBjeqbuDN8cyhATJA2+qS1/Fhoxieu2uiYU2CRjTfGELU0B5ReaksOxw9g
ICfc55w7fuiIpTo7egwLaPaA3m4yUGoQSfoe+FZm4tCpsyIufBR3YXRVMpFMS2Qf
k5G6ZQnLkxynZ3SEy+Xjq04q3HZS+3y1b3ikQ1o+7umpZI/eQ3kwggGEAgEAMGww
VTENMASGA1UEChMESUVURjERMA8GA1UECMI TEFNuFMgV0cxMTAvBgNVBAMTKFNh
bXBsZSBMQU1QUyBSU0EgQ2VydG1maWNhdG1vbiBBdXRob3JpdHkCEzB8R0APhiY6
HGLS64Mv1sDXhpQwDQYJKoZIhvcNAQEBBQAEggEAafaf61hcW19exMGYiSEijqEQ
wFqMk jInWObOGS4Tng36oAAiGiLJ3JBV4QEgcjr/FhJ6A1HeN/LFxBMhYBXiMrqm
d3HVndt1WSNOcEoyECUeaAPQxVKbvXCPGgst59nCtEzWE2Ct58RLkd431BAEt03
TPqKfzo7u0WADP1KHfxSpzJwmpj6HP2pKNaVZKN9w4ZTMHCwDRWR+3WXb+kwlP0
7ChjrmpLPuWRhRElljniRdx1tM8R60lmbB/6rjtpRXbKZH6jTYBRmOnzHJg9wsMo
WfGn/uYtvIegq4e2v/H5peA14Fp79u8ndV7c7xyPsGDbVjNARvy5hfYQF/m72jCC
GI4GCSqGS1b3DQEHAATAdbG1ghkgBZQMEAAQIEEBxcSQvgHyVtT5BnVpnby6uAghhg
I79rjVcQLwsmWeOaVCPz8zezMot4x7NhxWfX0RKueazhiW880A8ASrDW+77sbq83
HMur9uth951A3ICKuqZfb/Pj4GTxR2TNYDqiv5R79Wc0yf4gG2Gb/Kq7CE97/6pE
/9U65LLuMDXIdL1z73r1xjJtN7LVJ8HN8PuUgtT8gIEnw09IIP7aSh0T/xaV4F0L
Bahsnd4oRINgXxyE0gJB/v00rDpL5UxLwCoS4ods1PUY4M/03+IqoIuaJv8nakO
qrzULjcy7XFNxQCvVP8UDntvjoTZ7+RB4sLmRNd5qIp9R56dXjUMY8zize9qR0LI
B4f4fp5IcCxU78UO+JRu7IPJYbD+OXctx2pEOYwdL79cK3AERm2wSOF9xuQt//s2
CquqezZyeFs/i+WVqCjuSBJU0pnRS1gzvz3B6ulBPK/qehAGFpR5LHbulfjJWelny
0svqhMcozSvTbM7xf9sILSF0Xd13G7cdHXHsyYc8v4huc1ggFvcd/5vXO+QCetkh
H9vThqYQFd8tno4miPI1P7KvtypnLUeREpFt2pkuvy4pZ1+Z2J6cZI3DaoKvjI/M
4Nhh5SigtwrM4ZTweTwCojjvdr4iEWRLQ3KscA3X41AKm2XqoNNASzhLw45bj3t5
nJiAVobe4EiCefuqq0gq61Pz5WyePO2u/uG3mZKCsouebdQE14xhRub/aHaq70nK
cXLnAV8knPXC38r7h3LAGgCc2ZB1CQixS6ewaNz8oJPr+P5dd6TALhtev9Z8pz5
YeNwY95AmMNAvsFwAq4SGALAp2hH3w2yJTu6F04Caacxouy9bB9MAanJfxS+i1/
gkU4dn/3Em+wM0tEPznYckcrsFi+bQMyE6/DWiyalykCTr7I1TQGep71wsDaX5Qb
mfnhBDo7vOzGkqvchMMAxgD6HKBPojdvnMCMFMVAv33jErGGGkLxV61QntInFt6P
K9aGiS9EbU0v2spQQJZ8gXJRBwAP7E0c40EoDkSVnx2XBZD9CatZPnbgf7lgWdFS
tCka7NXluuRUV/R1GWA2AARMHwPAECzTdVfkQ9nSNqkeyZbcPazgr7WUKXM6SNEb
KgchxJTCfQ6dJC/+dD9MCJH8FNN7j41CgJ/Uaf8FeSHnvGnAhPogjqrENYjJM/gh
czK0XK2x3hzgq1f7If5CUqD41C0yzHALfHolKZQprZHJXw9+QhjhZcJ+uKovp+3x
mu5iWxhHpwF54Eo0OD97Z81UdDseypk8OwazoIKgFXm99jeBgv9TDhXQYwUIfAF5
Qnpp/CR1d6mfyv6wAAw//K+/fwz0PwK4RuXSg0upbodg9dM9O+dFOAidpd5Ruu/t
pGnP94ytVLIouSKq8rM/ZP0G151fLB56Ps7JjadBOFPz6nepHkMDwEzu5U8tqOq/
akx6ZakjqkTIVkhHC/HSypAC9d13AYm8XV/uAjoCpGiAZpLh9/1NqpVSadeQ/Zj1
8ZDJg6usgfxm9DPTvpxQ8+KuQMNY8vWJRrn6HCnoTh6eE440m0ot94pr1FOLLUuT
BANmXOYxSHPJ7IAduMUUVh6h2JMIhiVkfP+mZj/4Sy9iBc/8DS0SgpV1c1etv7F3
fGkzsDrMrdMT3Ywxf5dX9k8rIabWY0103YVHdfPUNK6r9sdlasvGqXVenMBANuv
ZhHPs8NtYgdbRfOafRtaEse6SNubEUI1ndJiDZE6hOdfIbOQ92++95XmEneODZ01
6kXy1HeheBzeOCe1w6TXxhkTaSBPcr9GRBeXoTThiLev4jZG4LDjRU39hziGKJQB
5hJBnL5DBfEY8uR9xNcHHZpcBSnWg7FWfSN0lywaCFmOKXrJp4ozjvVn3hdzWRP+
H50Wi7BNh83CSxqGEGuw3gnSrZtzcPnt3/pbNJovBf1L4RE39eVZuDT3d3n/1Qho
ae2X6PhEGOMMgSobXZzL7cYsQ0itfsLJUDLaoJXT7tTHhhyaxNUY6Aqk8R0d6FC+
07T6EL5cZQ5pg9ERt24WLufIQRUGah/ne6+ebdj1Gmc2DF+NM/+VGcLJ+3CSzu0e
fCxi31BhvGr6/62CFjPk5XMR3xRrnV1xgj/7A42/tFpOFiOQ7OI2Kp7x14y1cdoP
LmLP/6PgY498fadbyWqDEWnICOM60W5B+T12/p9d0U0MZoaFsmSKLO+5TSzjT7Jc

xptzejYn0T0t1/dwsYCSmvu8NRCsEAU7B02ZrTbzau98CrSOEQP51LJ0ploRv14w
 qXA5Qwm9prF9NS0u5pVN03iEqFGBYv2t/z1hWC8H2gJaV/0hqY6RcIsGWS9C6LHE
 qhX2OmpHao31E1Bit6XdWq7iDtpjwTQPJv6USeFbCxuqk9xSge6cBHeS1MQh3xBH
 0z75ey41DSTO+B4IwWjzHQM+JS9/edI2wq/yezQGpF0U+mULZk90OWTUXpacnx66
 DbOyeg1PiA8tYZPR47nHoNFEW4nGeF6gjHpWjse/a0c6Jx1ALd60QN6cpKrJfb+8
 y/Lkn1V4xgFHcsG1C3GNyMDTvA7A3CDCdCALCdXD5v1OFHwLJIemygKspIBZIP7
 v2mXqheE4arG06MTu5sCAPYB8L220WjdpGy9Q5c9lay52DvH65JnnfsrtopkKSfU
 RuVo2eNrGKKsSeL3wFUS/xjmSvYJDWDVScT/KNtRWi8FDuLw+lCq/eOC/CaQ95a/
 X6rKyGuElmUYLe1fiFJ86zZKhQ19+LOMjituYkizW68szy+5axC62aBP/Q6Dv+Vm
 2NlOVssZubRNnFvzq+Sx2Mr2GdnLC3wb/zFnYe3Ctm4WSJ72khpBfH66s6gzPZC9
 RXJdghEHdV8HiQ1YR1YrLlQfwON28p7PpMcOIJ7cemv1V93L1/ysxppMA3CZIm4Q
 ROUIAG7s5pl/jlG3D6wHmkibNs3uUS4S5TuZT52assAMpQPJm12tF+ubwEtRqhiA
 6s2u4jNOqEWyZCDNitKGzjtw8ifBvM0PDHRGtI9WFemCwtlppNto4R1kF8NjOfPf
 lupXyISaAFEgGgwbxx/o2WraNc9pOq7COjZZFAgW2DPA6eyC3yKcLT3GuInGuDlp
 DrzlrEfo2thkAyqsyG5lMNkzooihi7InouvIgUCmshAEr1qrjwGeBrCsdVnKur0E
 kEQFLtI3PychbXc4q6V3VjyNHL5oscmBqxoFVaMIbK3ApXNzuWw16hsMI1tHeTu
 zWAMuMnGlKbgL09iHUaTvUhzMaaK1R/dZWG1J420tB3L9aEud6lU4IhdEz1EjC1d
 7VEti422OQzeYU16Tg5WiHU/MxmsqOQsB09A0kHbz99nGeGsUNU5k9xfF2oVlfsD
 kKnNrdNq6xD+Bp3iFxjLxMsr6HzXNAQTRjTB8EaqCAp4BgkT9j9xMsUIY185eFu
 SI7Jgze8WAGAHQ9WSY2QxVbj05I0S1z8ZNY2Fv7JeDkCsePALuKcDXN1PHRoxsa
 bcpXn1oiJAb+PM0V4AGYoR3jy9+NznK1KeHYGi8lFA4I8uI7ukS9VBm89BHNGYI0
 ajV82mXIessCtaSClGjy5vWNIFrYyHKWNdx/vQgwV09EAfbhD5Q3X1SiwvCvdvl
 hQRWAF9E3GMXAg7q93r23Q/cIQpkaDHzOK+p637fnaEawuSDU5pTANgz5NdNSAPE
 Df8unnMf1L7cwl00ED9W05JHalTZBMZeJB0G2+074YE2HzZm5omS6fomxsQ5Ldoe
 jaCAMDTFXy9SaM1H/0R45750cyD+2xrJAWtgzam7JUiSeeWdpXdnTgkt7nrrpE9T
 eHHbf3v79yWbnq6ov2N2bUi8uoyZmGrnT8pRC6/0814qwZWm3GxsB4WBv/0EmTr
 20ARsnc9A/ve9E04TcsMLzBBPV8P8PouFogK6001+QATE8cBY3GekqAXAair7r1Nv
 Vlnz8UKFQt+KoDyZakAyxk6/haJajs9vKnRfJFNs12w9Yp1lbQsQXwaLwu6y2EQP
 V1ph2IN6BH1+v52YtLQ6ntEicX7wBEw1cJcCQAMILW9OSuWIrIYXSkDaQ3Sw+YDz
 onRiHnehKaW7HevSOZY+Kk/A6XozbAXxeuJv/LeCWALBXbz7r1kd9p/0t7M69bJR
 ysTKvNcnBEbHgMK7eggzd8saboT++vWn00Ye5VV2Jcg2Fom4x1rgscniJznxUUR
 926191qtFVNV/rjcDymU5mKGT1+1AU+LqS5/oT8adjEbAYyN1v92qSJPLQXeOBmJ
 McheNy1R6WsAXo8oF8VJ8l8fwM3Npt0439pKY9dXfVo0jH2FQXfCTyvlYZL40hEF
 Y8Do9OPbiBaKtU1lqH2hEUrogERXS7DLU1oS0yA6jD78eDD8fGs0KYomHiaLws7L
 m61aUjiU3RglTQ44hZfhqlfM1zUgCXc94u0wXuRdpik2abxTmCqcWnzPZJFGK6V
 kof1vZLfve2b9fdG4EB7uuQ+Q9IVJrTui1bh5d9k1S0A4fQ94Qo5Rcy2k9+xKU+Tn
 s7KUduEGalV10BtzfCMpd1XbHLat2lnAlsspZwYY0UCfc5f6HNclYA0C+8fCCbnD
 f+tRvZ0KxpgGr2t6z6b+3dZNNUNNBQIEW9UIP+TOQEgdzR1YL9gg3BowpQ1V+Koy
 dGFRKXcKDlyBPevC6jkf+GjE+ocDBtql2gCN1Qlfe5mXQMtFi4uce0KThx98kx/L
 ZJEWozvOoWSk7J+BhiWtbGt9yzeZJ6s29i+f8mtzyycmc85wJuzoPiv9dXmIyyXO
 NnnCnc2J3G6PydP/xNP4z5gcdVYwi96JC42Cc0uwRdZl8D5ONOLpZdLuEV4Y9vZu
 86jLXnWdF5pIf2JqB8rDjrUtU61jptnqFwmcXOQonYmcjzyb+UUfo/cgAalZvK7W
 4KzJ+NodwZVlnYq1WA6XkQfmxKjVIm5TTYE905y1znpKfz6oeXRltKsxrZCJns3r
 WysdeDewoUczT3UbZ5X0S7AKtUI3By8+CHHzKW1U0ZWGk9+wZeJT4cJIDaRM6eUO
 v2YHnDxXyR8o5VhGLE/UxR9oc4iPrZYleAG7amMapIIKmb26ZOJYcyKuwjNg1W1t
 mTzz0VI3tjshXgPWHEMiZyI59esnDD1XucN86YfpT6W4PMHz3+LzTutcxMpx2Yhd
 OfMmDFITE7bkJ+6oQrLOa+BjScN6jRUQsxUegyjrj00YW2ze1o+gXAcenzJzUX2hv
 V8C228zzHZUSNv6h+dRXdaztAu2QTtqPHFQawCqB3UX1u67Ulnlvxb7/JVshl2aS

hkioncKVxXhHKSps9i9uZOGgzRwmCo3ih8WdkSDUeD8e4m8Sj9aCYvPEyNld442n
HldVFGAnskP/hBeRYG56JJTN/W4Bzsy8b2K92y1QdZm1NVzwCBSp2r2k5eYGdPmO
c0lwT5xUKkubKqQmpdAzBCeAJBhOUY9QuCtyP1Cjz6WVaFG/QVvaXbByiI/20vIP
Z5T0+1t8QB2kE05KXSynWkxcyaelYHTkSdsTICUnmDgT6IyJGFuDFsGudtN0p6H9
1yCPKLElSNcL3z63fDngAivYZEOLyicVlnAGuKMzV5THg72IXU7V4N0Woff+dNDY
3jHsYCNyYXW700r8golnfgZgBzZoEeUWGMhFHyubXoaJOBcZhRG3CPggPnUY8ij
20UXJYo2X2r7+prRx6H7V1taYZA9os6VKoyM0i2V1cIYsOulneXd3H+eJp1dzJYr
1b1x2Cd2Fw4NmCUimekWxSFyhu5GPHcvqU00kA25Djktmsq9MKxZdtZ8WvNYnC4U
sh5m8JjYPQqvELzvt+E1szengbK5sQUam7Iln1zT7/3cYTB8sAJkuLcAy9u/Y9+M
y3xqq0VhH+4/joJ2w4Vm1YB8FT8Hm9Mq62hYz4XHhQOS/D5r6dvnDUqSZOVxMNV+
pHPQHUrUFQ4fAFWzN9IO6Pen2I fWDJKI9+ftVP/CwQxXFvG31zJdua1Kbo2IvuJN
Nn05Gc01PHgQFIMBy5pVTUwq1y1r+RTBRnv22/paJ3ih1r7iBpSKAqtlBEssB9HL
E3Nwkd2P/zM8vccDdoxjsL6Ss/sjwe5yU21CncXDcvRd/hpN60TXSWsw6Vn1N5fh
wE7NVmwQ+FQ2Hw0ro33zRiYsY/ZgIas1OedR/ybDho0BOcx5170IyEdowQpFajKs
W3NYVvaMtJZI7AANOHg7gxKx/TstLcKyZfSa4l0qnjjzLTVu5wyWQyWErtjv5U/m
1CCXzV/q3pBARgEnMhmwdRb4Xfp6Ik/LFzRddG/t5z8iMKgrVKA8EJeiOqo6iGiY
b6NJAvzaOb7SprYv0m0fow3nsWSCA3m0Vr4mEyCkQVeKZq/CEmWKD+XKV702YxiC
W1vyaQITxt+s8Pi3GqoPTfTg3TE4KoGUQymElcgBZqEJs1MFXWzldvspyS4hp00r
L0wq/o4RkYhXHMfibsAC39Dxxct0KHEJ6cFxaWf7ABIVwMk1EuKtm/QI1Gh351q
N064Qn4kwMhr5/g1YjIFKIJLU1MMKWg/bkqLx0L2eIUppD+UFzSC2EjvpimPTAhNx
RsZk4aWNscJI1lBgaeJpZ15ZoJjBQ146+QGcri2isW6BkiJ/d0L4MbQT3q5Ejedx
I8+xt3C6U40Icf6gQD0Zr3AgOQGTIa42iuYhAK6I3ieJan051yv3Pjfx9nxxdsos
EUvn8b8jG51iQpwbJEbh1UhbXFppv8BXDC3Dphm9NIR/v4456Q7KwZ/IDD/zUI74
K6JUXo1N4YuzDrXmZnMR6oHywLqvHmvXQd3F1KRpr8A9ofuQd05J1+YlHntrzquj
1wuU3soH+zNeMldLjOpGust8sdezM+6maqI/ILZ+5GA43RGU61td7yyGpfbG49Ml
SGBPSyMn6MhKynqbnMJp759xxTl9HeJ/pFg1BAvvQoCDJMEbl7V10LZlGpD0Db/7I
qUF/hkPg2siW/VctB0mgFZWLL0eh0s2zmzuZAFeTUmvtula0/R8YcujuEyw7nR/
8SmT4nxvd1j2n4dLW48ukpkahCkULWVR248qmZr+1DWYPuz4P70JsOSk2dois0sr
ZH/EgSGHRtyHbv7NxchaEWITkKuH+koQMYCE8g7WoW/kcsrqRuuV50PYqK1lmtZ8
5n7duXNnn08hLhahIcA9rXYchQ1PldIZCz3oI3VvRh94CQeyTjFzz1BCZOyESzWt
/aJcNHM7gRo2oYUyGymikspuvvKozoAiRPS4rTK88un3ojv1I8+JLZyiNHaNuOGz
uP5h/BuuwOckY3eLCgtTsapMqAMvybQB4hZqxywoEwKvZUwCA/HJkoxuwSeuM2uH
PmmxufmqWHndNg3BSCpN0xjclf5/ZGQZGREjYTKwY5QsyehITmHr3rCGM+QbDM3H
4YoGwPh6sa/TVIkX1a4z1ELVzDV1qN3+ecy34zJeZLfgn4f6cYJ1Qz8ga+WfTt67
QIq84sNMaKCaCnUldP2xVFDLwxzqMhHXrYEOrLgt3tGFRbxGJH7ecz02vHp8CWdq
VhPyB05RPFgch57GAsu1IVNwhKUYlGvFb/9aECYgONcxqNcvOCKGSVgyRDWGV0Sh
wPyluTaz+0QxSQGaYvU3THYzzQ852q09DbDhH8xR7QsDTpTbRr2Rk5CSNHw/gNsh
OqgdYL44V+ryJA52q/zBESoP1oyZX3Yy9c8PbI0n49sm8Y0KWbHoBhsywREdtTsH
0hKK5j1XjgaZY/pTen2D34xSh8guGQIseDi4DMAkRMAhMCQCD8sbZKk3ZBuJcB8J
JQioHhcIk7wHbcBrtL/P+MZkp3StzSncn/zr+2gd9H+Gs1dS/gun5ZpspGcCk3xT
tG7VqZxKyehEXeElCXgbNtwGKnsKOAgZ84MMNukFt3EIs1x9JR83581B6tpYeY/j
7zYSdwnUlXvtt/ETW682XYqVRBHS86vKunHANlEZv1eRLd8Nd9WM+5LmRM1o77N9
x8n/1qvmJpzVu8g9sQzy/31rWtN+f35p6ISDRs+KHOX9EYvpqrh/dwVacsD/XBIJ
T/La84y5fr9p6pNODlgBr0s9c3Vkw6isbZXNdYrSwYOAcrMzXJ/51Mxt4P8r4RQC
HVaPR/tewyb8GF46BQ/gllVnc8eQK6GH2yw3FZba4hKJ6HdGEytfvMUSdoSF2Do9
XUYR9Fq5BETHAGYx1RFfVR9K+BdqLJpD3Fx1UzZ3fFrmYjE5+vxe86HOo4x6j3WI
A41jep6yAgRzIFJ7f//L2+5/7drzD8jhjnhW2CKQZiSoSqTMAVqNA81BSdR1o8X8
Vf0P11sV1zr7VwyLFJ4K/QB1nLAOnj2wCGGASli00ns7w5IJJV4HbZx/cyDwyekA

B.3.15. S/MIME Encrypted and Signed Over a Complex Message, Injected Headers With hcp_minimal (+ Legacy Display)

This is a encrypted and signed S/MIME message using PKCS#7 envelopedData around signedData. The payload is a multipart/alternative message with an inline image/png attachment. It uses the Injected Headers header protection scheme with the hcp_minimal Header Confidentiality Policy with a "Legacy Display" part.

It has the following structure:

```
application/pkcs7-mime [smime.p7m] 10205 bytes
  (decrypts to)
  application/pkcs7-mime [smime.p7m] 6548 bytes
    (unwraps to)
    multipart/mixed 2157 bytes
      multipart/alternative 1431 bytes
        text/plain 485 bytes
        text/html 637 bytes
        image/png inline 236 bytes
```

Its contents are:

```
Content-Transfer-Encoding: base64
Content-Type: application/pkcs7-mime; name="smime.p7m";
  smime-type="enveloped-data"
Subject: [...]
Message-ID:
  <smime-enc-signed-complex-injected-minimal-legacy@lhp.example>
From: Alice <alice@smime.example>
To: Bob <bob@smime.example>
Date: Sat, 20 Feb 2021 12:10:02 -0500
User-Agent: Sample MUA Version 1.0
```

```
MIIdbAYJKoZIhvcNAQcDoIIIdXTCCHVkcAQAxggMQMIIBhAIBADBsMFUxDALBgNV
BAoTBELFVEYxETAPBgNVBAsTCExBTvBTIFdHMTEwLwYDVQQDEyhTYW1wbGUgTEFN
UFMgU1NBIENlcnRpZmljYXRpb24gQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIeh7j00
Boq0MA0GCSqGSIB3DQEBAQUABIIBAF3P8K//f2QuFu3CB1QYWA1UVOKdVUefYICd
TG2PVFlsq76rPSChX/WA765rYh7r1p7cpKSvcuGYkLHxA28CXiR8i77ZCcoxFVVR
vOqPGTZZ9eoNvpYa0qOai6KVhkRbGTwyXC6mi18N+Sy6tLcTR96jSLi8k4EDtKJs
v4cCrA4QRDEpNFyzftj48yfhKCBZSjn1PSeq6p5RW132SFKGe81k72ez4VV/pzK
idOG9ltviQ1ffeRFLI71VpEQov3fKckkxCo/h1DilcFAo88o7Tmc6U8DwiaMr8x4
rQXB5S8uBJLNUhrdFiNIftRM2OJp3ij5DM3YRBoUvnDaKfiEMQwggGEAgEAMGww
VTENMA8GA1UEChMESUVURjERMA8GA1UECXMITEFNUFMgV0cxMTAvBgNVBAMTKFNh
bXBsZSBMQUU1QUYBSU0EgQ2VydG1maWNhdGlvb2IiBBdXRob3JpdHkCEzB8R0APhiY6
HGLS64MvlsDXhpQwDQYJKoZIhvcNAQEBBQAEggEAGuRE7UAzm9ELVleX0vu8IXiY
vh/9cLBb2MvdmWGKIwHthSLxiZA5X64VxdGjFM1ZzPanUhhexMLTzaP3ADx57dat
SnmSfpT9XXbpkokCPBL+NBpA8e9vtWAOS7yIgfpwdJyBbfcYi0CHGqslq/ctRsVF
```

UyksjPX0dvJjqSM7Tnqd7F3FIToSdoelZtprDhh/opM/acJl++qovSgJyL8AZak7
mSU28HbTnBZD5iXxCppi0LH2wK6KfWPqSV3AG8wTpd1qF8v1Iv jF2Sur9Jx+hwKZ
1kNPKOH8G+PgnIA800gH2VDW4Husj64hxShEWzAXUFqNqHPwxFbf0h5Lu0S3DCC
Gj4GCSqGS Ib3DQEhATAdbglghkgBZQMEAJIEEIOER8IO7SembW0J+kNg4yqAghoQ
QCNckOUP TLID4uHVL44bv4N9/bwWoKN68FQvcoXFHbica+KkrxCMHO+nIrFVSNnC
FtVXB5N90rVy82ACdT1MWQzC/npdlfKQB41F8f5owkRSG101CZvxE/LqDhFNfLrV
xHdPWi6djHNRKK96S8HDwhR0FtCrCt6kSP62AO/U4x/FUAcQxxc/ad0OwHACucFe
IDeoHb8ne3FF3cyuh4Q1K5MdW9g9xp4Qw4nA6WUYFY9V78X8jYvwxCl5XRXKiWaH
rdeQCMdy78V56IvSXto85uCJDMgsvTs+xRyyQZpzm9dt6LWRMm4XNmkt8deoXn8g
K8G5QenEWFqj3uPVN7MSVYwA8WCx/qgCDtjeNZkM70EGhX6SXm8JRhmj3QHS0wth
rc6Tpc6mGZ8ZWBGXOv1GpL4JPB7jgEWm1qEnZOjofwyOLAQxhnqpPOEmTvfNSrm
/yeDFBz9qPX4Q/Z9OUUnPYybiVYoly8Flam5bJqneJR9XFUjv95E0rFkzwMv+ceLy
WaicDNCPbXI71Kqj2KdTLNefcSSRLmtEYqn14aKeIOMWA0HHfCkmf8SMkLGY2Cq1
DdH4sf02yoiXpCa3iE1BaoPavMrkVzudyrrzRXqIRIDci8ND4knhVdayLUFvyZ2yB
aNomiQ9AMtya2CCGh3GJfTwz2U1IzEaZ0n7ZczW+2pWBCMatvgQfhtCDEhmX1QGN
V2UGz26tMwf775yNhAoIdYesgzZp+tnGlm1MnwGgbWixyqM+FPO+Bmj7/g8/vKC+
zvuyYW9rwbU+VIMDQ+X6w1o6bzOYv/znSdKK15UI8nSmfkbechyN1BN9o+kX3uJR
Mw6gCShn+ouia7PK7iy7PCaEAApS8cRsT8XbYZoo83KcHZM2zaYZ5gGOPOnulcOX
GSmg27A1zRDjJcP0aEJ/StIwomT8641Ge39dprTU1Ij6L0pWWEa3x8M75HWMm92
phMd2f7X+eht71Ix+ne/tc+0BGdKcWRRWJDMIrfpX9WeJZaZmJhNzT+geR176CQ
OPmmtsFaNt0toVbWDTquzcHJqRNFwRmWl9jOxz1USPPtKAXIvtqgYXdfshLDpx50
V3ETsmomoElr0McvwHHB1rc3sVvNoP5cqjNdmT Yu/2iX8lc7BjFPfUQmwfMdhKb1
mYMEyzrWT+ABCCSzf9iNjKx59oKSuVdi3oFHOFgu5F723QCw94nHfWj1fhsx05tR
zaZpFuBwc+a2z1Pd9FCsu6KTLwdiUVR0AefsgM1l4+1AVVIEAsZt8p2el/f++w+7
T+OxYIpeN3/2sUr6tzJANWw+ldAmMmiPqlE+2XiSj1HFqIyeHUSKJVRhJxkbZIXo
e9tW6wU0pb5abx1A1E7rYiL4H1N9DoJStLEgrRADxYBCf76QQA10jR9JLmOF14w1t
a1UkC3MRpJeFfH12jdFeeHoABM6NaLhOzS7+MtokP+zQsFcLea5FmCmsSndJVu8v
esS4A1p2szK6fuxOhsPRdOAuia1U0mc5zRo1xW+VD7vB0JN+VxR7puh2b8/5qEd
XJgEedz08cDKRer7hSoLxDUdJq7Ruidwvtsz8lpXeVF8ferw+weDNrM1diDSv3J
kk7XOqvLwz5Ud5W5D1ffo2cl68LbejB6ZgUzV7QqCKIzEHfgOz7AmZ4rkw3L2qaN
7EmE6JC+JGsqQsAB+QQgmwmM/atuaDcUXnzKrrWHmNL1XJe8Cdpd3tmquKqp066C
qEriBqD1qKbtSZmYA80YRrjffFRKk5hXupimek9XJaXn3tOa6WwDniXvS+nE4+qyf
by1qy3ALwm2NVMFkVAU7qFTLgK53sppEEmDMYR6bMoDX7zk9vR1Gipb4JrOtkuAT
yZdVIgkW67kLHQtdyLSaGuJnJA07tAmW8UTMzNW1x1T7KYHDrqoMm6hvXKPhh1g
PaHGTRFxDebmW7hQ7nmcLHs9ca4cjbGafeNCZrNhm2BZ1D46gO61Nf56npjATYEL
saJmeJBAXtrvqgC78CfngLG5SEAcZnKuUYHnpOB0mCUdqL4KHB15LmUg6jBRIUuQ
4aZQHx5gJDCwhvcQCI9uAxtnhwxcwJ/KUwGntfBeyh71UAbLpNqjF9oJ2UQFAEol
j/qr9QQ56NJT4Re9obu9XWzR/120chZp1Yy8W0cP1MZRQU1zq/Fp7eDuYv6qy6jo
1yZfWLL/8u+zaL61XbgksEvDrR21Be1q1vhJApw/LC7Ju5Qucc5HTEtND+k5TW
XU1Q4QI9Vf3/jRsoCuW2jppgA1krLDAtxzHV4MkyDm9hqWheFSSqLGguud0MxFe1
j6q/ubZsIxt8Ce3NuAQcQMZdkUM+0e/4KEHFJPPUnfh6JbdX5wWJieOPRwt+1ceR
CaIpvQKaCPKPiGMWEyI5xHcHJDJDY3WVmSCTtm+gka/CpwZcI8+szy9JRuUnjyg
LviXjNEQm/413QFgW5eV3oa7aUjjSEuh5+DvYWFb86ECneJhQCXG7c4ke+aIE4ub
dx9dyOez2MjaYoeJmJy+xfNHYSfQfmd1MdarcPjv5oBdM2NFidPAVBgrQte8tSmz
rmjWb06jRzhn7LEMgJRZ7UGjgsIL+/+MO8KckFs32yJzEfz0QUXyhaxn0BVT/4QU
lfQs3C3Perbudo5GXbhXIDIwkIoWLwbUyZee30/Q0oNBpYsax7AAk/IuKNbdt5kz
LssTIXrLDnpFirt5pPDBFbaQTJrs1rPLTiIZIMwwJIOrYGbP+P0N9g9XoQal0qPh
Ub/O2CsSfragMboYltbhGMmSvPgnlC71dVztlpMJ9LZdoHHgdtH64WqBO30dSljg

rb0kUNNAz0Sj72N2w5PM7RQ2wzbwNirC0eBrul2CmT4cPTGzQdeA3ygoAWvHYJ0U
MYERTPUBHccQjOqicPZlPz2FEtw5+40jxzuzJgyJOqRnt/teJH/MFCkDLIDC2iOGa
JT1jsSqTQMOjJBNb+3vAF607LVoRAFapgMjjbJNHRvfNzMK2+PAbQQemEe8zOVQM
Ab3iyFIdJxQ11UiDrfh5/4myWu01BaFPZLCyJET64QX01XfSUaeYisf7ebvvcCbP
4ChqhxZcomqfs6gKhZNeV1v//8YDEwVwHwRaV7vxuGFhZycUsnXUS3JazFw1hUgb
3H31KL3QGyWPkK3ogmMD2HfKLVfblPdNBmu++jeAef5n0Gvau0oWOHn9vhhZ++mq
ZGbkhd5HyxOzglF8/MrEQmFrs/ISemFKtSN07qeifzpxqAu5blRv3rdx+4aDK2J
JaKBX/GSu2y6XkrZ8vHZ2LeXDABzZQ1K2cjZuzqWwNjHAg9n+xpOIR1GkLpGm+XJ
hbHWef7y+g93cpVAEmMY9dmfFRWFMDzdfNUgCuaV20JhWnqdRB4fPlbPobneyqLA
zFt8R9DjsY0Xy1KXnY07X6yDnJurVLTd7h8dYmV7XM4JGHMRqOWMjVz9ou0KfE+m
VbDkzN49wyy6lbuhVFhBsibXtKwladl9hapfGbdK5/XG5FctRbfzTPIZ7vfbrxP
JOKjfeTuvCX9igkNjdp0UbJWxdTCUwlor53jLGHZN6rQbjF7G1fKXiXGVgI3T9VC
P48zTCqoHKmWkStKjtfqO5vVFjOxmXLaLoDlwFtme3apTbbs0jedNav1tXjQNgw
Xms+N9DnPCMsXaYLVBlJ/8aVIFmAemuXcShVeu8cBynkRj9oM4q8Cf3nK498K7B/
WKv8qfCmzUUN0LVQWE3n9XV521MhYDRpUox0D3RCC9WedWXT5IQgJ1iBR9B17taK
pSRyEq+XzVqgIn8KkTSXinxMbXWYRCncYB8mUdHaEiULkw3QaxyQvODJyF+V4CWE
v+T0EeqT4QkVzH3AKdURw97F6FodhmJht6qT/F/WnoIvPSTq7OJQ/uzEs0aL0UP
L4oy5jHYpYgKnQzP3fI7DQsbCf+Nw1Z2+Cn8mXf2iA5Ps3lCVPObFPLQ1LG1Zc7o
6BkGub3bqmNp18/sgGHB/pEQT2gjtT1T1lJGGH5CoGE6+x6xqHssugolpH4+NreWd
08EBjGAOEDy4vjGAcZAIYgIJBzIeffDw61+R4K14Ljfehkmx6ANtXabGYI6NBs9
zOCIKNe611oHKZT5FuQiBCivdDyD6bLeoKtZHcfkBuTI2ZL9Ftzo1ODBzv6FjMP3
V1NJRT4UnsT/nvJaeqZVofqAvVBL2CRIWo3IjfkSkRothbvUN1ZmLQ+RtWeA35G
xjX54V1BAZxZcudbJ2kDUsAieSIRPWAPeywbvBWDvAme00PJXFUsTZ/S/aQXmg2
EBpACCUrGwYiybW3Q75cuTTwU0HTG9mQJsx+zDmNaaFP1201zB+kvv+G9ieDWrie
PGux3Fg6G5X4VXtUrEn6Hee4cDLBVbuVNX8vWO3cJvauzQZHq57wD7ixxcFyXk4f
pPevmSEX+3aQDhEabRe51NBzhH9DdzzXG+Cfcyjl/02xDgVZlIqventjBkkA6Qfp1
Rxx4FHHzqNmlbWM/P+CKHf5e/tojrhoIPsne4rVGFWPYMXigF9M29P1lut0KK4qDV
RuJJB9ruG3Rs6sqN4x/m8WJxvGjsObwvrbQh9yusV00pV6d7BswCBv50wnwrHWB
Ka8s+Bo9Ax8uTsPKBM1Cxu5BMKjWtC+3yRxU0zSjFu0vpae4FvqHqHqAwKJTqkmY
KBXnDbB72DTTLivTYYqgTrsx38Aopi2MwZJGdn4AEiauf0577rehC10lcCWUEmHN
X/12qsTLo2Ym9oQySoSW313ZKFzdfRlBmPd4QcL2ecedk+ZjsEGyJ0yNjv5NDPI7
yASEOLCqzTmiei33MpN2B2N2V1bhx7+B0Dfi2gdguoGACqwnIFRBrUK1cKPPAE5
zfIDDXp66XmUmvCwKEbCJPzND+6x8ypvKqyqbu5scS9xP6daSNY1QoDKSgLKlGm+
1424s193XfofotYJtZbpZANRfu/aUjV04Ptej3NosmScgp+mEozbMC8H1UKUJE1Z
g3LNZPYisTWNhHPTqjldPPr+4pleX0+YBaAjfizeh3aLcOr81gzKsfrxGVYs/oj8
JrY2oN3C6sHrdKJnL57AFzE0vF56/A45znvbfqSUQPI9y1ahE706ABHPhqk5/zxF
2brwm2BwDD06T205PghrDKwGwVqmfI3ckcd4UNMT8Gqwd/sw3Uf4W3nPFLLK7yD/Y
j8uT3TrjI5yY2KvIj6m23hTca35r7PEB7WcTOgsmFjTvWPOysOK0d5az3wbsV8DU
xbKzsGPSOCWy+ykdW8eN5LtE6GBFitu1rbw2DIYQk5dKtdUoohaM/x6BmXIGvmp+
pTTLVJHEYwuzTEEgzDBYPB4WVx2ziXGrfQiuBq71tBp587VNDpMkqpyoBUSCugj
Cfe58nW5DBGA8Q5sjAKhtcGIO5AkHC8LDQDdvWDTMqw5+d6WbAsTRESsL8XRHXIO
pDDcs0006LncRIJo9zdEsADDZomRxsB4xRcSETKevgAhtPPD0s8qEl2I+V9o9dcu
oFDBeALHR4KwaZ9xQDbhTw3w8QswZbzbYOrPB22eudzmLxrOCCim9mYM4vp9Gan0
/bvTWCjHt8AkyqR5y08VjOjHH9UGJIaCG++2/H8ij+ya5UVY8+Gfewt6TLlk+3Hy
y8HSNIBn+4G9DydfmUSD/j8x+L81YkRQ1Z5S3/peWTOhJOXV8StXSXcQb7umRy87
45hrrDdfcSZ6QeMhNVRv6ifh8ImIC5hCXMg9dfz4sMZR5tJRv+LDcL45OLZ5H+p4
TNxGHpDpkdDzrTMhb2r9oYMPjHvZygH1fWcpAtkDDy0fUCxvJZAKoVhKyW4IM3fp
FrlxJ/614a4M46CIgDMH12FoZj/wUw2VKDf3okpusY7y/R93akMEM1BIDCXGmUg

dy2OQI2FGjeongJUo8Cn8XGfMD4eWShqBUDc0zEiZT40Nx8Ao+qbwfGgwegBpxlu
xSWIM4eQ+YimqLpmMqN1qwK9cME3pKAHZnVBuWJ+8YxJZVz/R1CUmcjBj6WKDk3e
vbl2FQvB3Kas5vierHSTaNdFaRxZCwfCkFfhjShAHdbHYd3ftwdw4TG0Vo1j4bCJ
DyVn4v+/aZ006cgRwsmIvbjHQzYKItzegcn/6mNGuz5i8doi//cwhm6y1r8oxebT
d4CPHFnWl+rbtjV7nh3Px+8PZEcYOXOs+uvpdtGMSiao0651TFb5F5QBbtH6xODg
HvjZ60bVzK3C9ZTIkuE/JNQRQjHhhMikeXuv2k/QPysAo8TQvox5Pcg1DXSMn2Lh
MVj973B3mm/TXbBbagKFeQjCq/4nKiy3lDzGwR3rkVMEJzXcS7rgYkopzccH8XuW
17dSymQ24h2J/7mFotR3S1hGn5jrDWLT9oCyh9caExf58KBKm4LmsmSyTKj70UOd
5gQRsWxDezz7AvWNJo9OZwjaEpBQdcjte3KZX1Zxxv9scEsI4jDCQY3D++77vGon
8BcwQbQlLyzJnA7kSBW+QSo5DwceOU1DQqSa9/Kp0HANjy3mZxMp1Bg/+0uA+8nS
UCxC7DqQVva6xFECxaQwVA/fD/Y4NjhmFvxh1iBYC7iA34K4W0E8P++6fglm7gS6
XyYLVl+ExjJgJLn4xRC3556CGSr46XWYyLTESqZVWan6ThcxTdYeybeUXW4JOUJx
AlDIL3mM5447P5A6gmz9/VUuRkqPRQsdeOad7YQfWAe89carf7gQTqdsG7CjD+x8
0ivGprQjFxi5cwfC+NOCowZsFC/qdlr4NciDjsgwZnpP7QW9trho18evo6jsUiv+
+4kC2qdQ/Fm37xMcwTqTE5PEnsNX1302Qbhp6Pkbx7mrXsib4gTqz6Wyid5h07LW
Afwkvju/plsUV8gIWMRS1UnrmA9PepLt75pO6+u+7LDcYuHAoun/TC3N+AvC0ORE
CtRIiyMFPDw5v5sSeRidVpoRX2AV5/2ZncYnXizGk8FIv8C8dj/Mtd/GnFFIoT7x
9zvd3fX7PGdeIzptPDS181a1QbuvxUNiY/d+oa080/HkbzkoA8VaTL1HRxLJveMH
Snfa9GQFzHP1eOBuwPGnrTNHMLiREC4EQuHunyHyaZ7ut1eRwCXqDMYd5i9/Vclu
K8yuMt1kCyFG110zuCfSFQ2CO11eN8K8DKLiVAzIVvQuG3yaVTSwtNX90mP2qRkn
b608M+Xz3b0srajjxa5ZN4eKROuu+1KA2JeC00Bu4r9wHIS60toBgyWzkhkHqjkC
2n6c+4YPcMMi2XgFKF6T99hEzRr3rWKTksAJh/5dSVSQ19dH3Hwcy7C3WgyiuupI
qWkHmnpDMBUuL+YkF+Fxm2wU7mKDB5ee3GTO0MD19qZSpbHvrSk/ATudlAbgYXd
NGmHBF72S8VdS6PVPnsTpuNbkYAHMat+AmfdezW/FEWV2Q3riL6KA3thnmayFxA
GLCMQ0sm/4u9K1L2RCMZf2V9/v5InTRTAYEzo8sSp+5Zu9I6Rb7mwHZTgLMWOBQd
kjcbxygVSiBLWvyofQ9WkP3iyUVjsB2mf5ABk4SWMefiIld/aA11QvbcnrcnjbKw
b5jnYm6b6bKUJUzZoMGR2dzWi082TnFu03j1Su1+1DxhOB2LgKypeJGpTMD0smZD
jg2ZhpB8HAJCFqhose1n31YN2roINWEC0kyTDIyHYZmmubd64Upe/wYbJWAAI2gm
kj0B6+HBZatjHCdhFv7oR3+smnFUtff59LQ4x9eI6DkJ/3r/Iwyd+5XyZKoDJYJp
5jiwD6pQKW+VuYzG4TxoTc3GXIB5s/22yQI30v3sYG3uSQHviYmStGQxp3pVBA0q
+9xkOMpZp7nFrBA6C2obNabDpTofJef2aItfPPmuiIrrjQYpAc5o3542S12fQFmbQ
G2LumyaiTdGuH8uqNBtYnNDQFUSWfnyqcDFIoyLairThbgkMcB8PLip2O6TEKwFV
s304MG4vLdGYjBsus30axpSYXtS91JfYpGpCEZifkUR7yZw+sfb3JPAjeNelqs2H
1lcNEiMQzL50A8cOtzXftKbLU83H1DMhiCYnS49VqXgChYK8EPCnA0UoJ18CAahf
oRmOoK8N+LMEohQV6VcVL58ggwnR5oFGY6ZuBIv8jJcCS9uXiFZnnoCY8bgkxxvK
7d0kASdiN/eFnzJkPfoVHnkVLUI8kSIY0799iw3kl9dYxShfrma18Xcq0r7BKM9n
LChsKG41P0RLWLKRTNyI7J6cX484j5FswT8MWOAayc5s51MPUkTn5OX+bWyGV2eV
Th8QwyRTgo3DVcoqNWQ4+W12TEgXbiM8w7ZPXiWifGTTrL4vR+4y/H+BqKvJUjT7W
za33W6iRkgh1bd0jhbehmno6yRcpw6Zcu7ndW+Fdt1GBoOtiXjmqo1Bo00po2cdP
3ToOU8fH1/NExBG20S3Rqhl+IEtVq1Xrw5hVIF7FTF78CXeGpvjue4BAKoiR87Yo
mHnesyBocxOaTxGgiEucDWJtMnJ1L9oh/Ob/UAPQVQngkWSK9HgP+cGiJDkt7e2I
Ktd/Se70jZa5Tj0Ry5+9akSpa0HwNn24GtauqUmgnotP3QFxrO2FR1KiG6LbsfGH
8NrUGUVymMDePLAGDb4duc1asNjGJ2uSzs3GA5EKHqMdIV+VBj18klueffwn55Hz
h71qzW039NOQ/WyEJbmZWg7811CnW0dz8dD2ac/fWqpEmT3+pBsiJok+WxPKqv39
s7La32r0XAANEUcA3m79ExjUtD6Yfn3k1s83z1Zt7rgoI5jTVMSEdtaUctJ5/GkT
+ruh1fX05FpB8/8oq8hPLAvf5nLZcVtEBHcgKuIeFwPmqChyqPFxnRC6PjBzPVBH
ugfppbVP45xx284eJ8IpXSSXnFtmPhAzPkzNSTfYK3NG5I34qTSaksvCQWkPJThUd

B.3.16. S/MIME Encrypted and Signed Over a Complex Message, Wrapped Message With hcp_strong

This is a encrypted and signed S/MIME message using PKCS#7 envelopedData around signedData. The payload is a multipart/alternative message with an inline image/png attachment. It uses the Wrapped Message header protection scheme with the hcp_strong Header Confidentiality Policy.

It has the following structure:

```
application/pkcs7-mime [smime.p7m] 9840 bytes
  (decrypts to)
  application/pkcs7-mime [smime.p7m] 6276 bytes
    (unwraps to)
    message/rfc822 inline 2016 bytes
      multipart/mixed 1911 bytes
        multipart/alternative 1128 bytes
          text/plain 373 bytes
          text/html 471 bytes
          image/png inline 232 bytes
```

Its contents are:

```
Content-Transfer-Encoding: base64
Content-Type: application/pkcs7-mime; name="smime.p7m";
  smime-type="enveloped-data"
Subject: [...]
Message-ID: <95b9bb39-c028-5ff4-99b1-f179cb5d7585@lhp.example>
From: Alice <alice@smime.example>
To: Bob <bob@smime.example>
Date: Sat, 20 Feb 2021 12:11:02 -0500
```

```
MIICXAYJKoZIhvcNAQcDoIIcTTCcHEkCAQAxggMQMIIBhAIBADBbsMFUxDTALBgNV
BAoTBELFVEYxETAPBgNVBAsTCExBTvBTIFdHMTEwLwYDVQQDEyhTYW1wbGUgTEFN
UFMgU1NBIENlcnRpZmljYXRpb24gQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIeh7j00
Boq0MA0GCSqGSIsb3DQEBAQUABIIBAFa5urZzuiuJCF681wqMjpt5q6ecCrubcxrFW
ufCpLVF9IwsK5B8mBc/Y1Ao1Izm1ZLHe71vRftcPk07APU/bkaJ0YtXyELF67P9c
AvW8XQRf2oDHEYgVerva1KvWDxoCDgyBXIGfaD1wjaZKs2nAM4fnWfju+d6zcw5q
uArKn+BbUI43ryuHTDiaurzBwBEUps64ZyXNjP73X3xS1YV58OfftHQSHOKoPHg3
zebVKPSqARhugLWk06GxDMXAEjYZZBqrrYEgKNANwQ1lu72bFkD4gCXm4kIc9ezU
ZDNTctiFclShGZB4Kdmrrm66ogsxJ+Ecvw4YVakWbJE4+eV2g5gwgGGEAgEAMGww
VTENMASGA1UEChMESUVURjERMA8GA1UECxMITEFNuFMgV0cxMTAvBgNVBAMTKFNh
bXBsZSBMQU1QUyBSU0EgQ2VyYdG1maWNhdG1vbiBBdXR0b3JpdHkCEzB8R0APhiY6
HGLS64MvlsDXhpQwDQYJKoZIhvcNAQEBBQAEggEAKQ13b9qWHas/pyz/sKKJKkkz
DMpPlzOvhEtFBfsUoKvSrgDwWgmKhO/V+B7abpEzibR7I4rAadHzgU1wfbNf30cb
WqcCmyj+YA6w02rB0+y9X/SazD5+fmBwbDjnmWDXnggImy9xXrLjTl+7gII2J5Y0
JQXI96iSLWdFP6/Tq+Xj3HD/ZKL0+HgV6ncTNcpjkRPPuzm9vTMeU4qFVoNvTErI
```

V5vvmzvJccr8E+oyloP/xbd4qv9OrxbfFn5SAZ+HkypGkE5NAy3peSRDwQ6qLEM/
tKuYIewKJdv3xjJO0JyQxPRcA0FCEQp0Ovt/zPum3aJ5Rb+YpIJEVHhwd7gzgDCC
GS4GCSqGSIB3DQEHATAdbglghkgBZQMEAAQIEEInoQ5WN0SAuuCSGvprpRuCAghkA
Retlz414Eo8kzcdBnIBWQ/HdAhaJ8anHqEqq+Cko8a3zBHxAD3QsJ/Agje+62Cj/
1Mn64cw0oDarrIzkfzH7bqOjWOys7VmqEOX+v4WePKd0CoAzYO4J+ugOf7QcGPzj
unj5pXTjPmj7RvklVxhPG4DfYaFwpjQBAPLE6stWAV1Rdhv34LuIeKVJuG1114ZI
Xi/0ilWgeRglHdxXT3OrmrZpP8VAu5oH2tg1EkPHoKqeahyNLkA+fLqjGH3ODVOb
mphub7wyBNRDH8yyXZBJKoeT/jq2FQkNjworQgbL4YNYH6yysK8/rRwAlDZGpKFU
eeHZf4N4wwquwDAV3OgWJ2ugJIIVqIEB36JwQ5ocvWTZSudGe/HwgoG+YUzL/53s
Kyok994Lrrq9JQKYIkPIibF6ku46LAYmz8Jg1RMazE9zSWeqfyicqZk1bR9+r8dw
E7PK5p8EAEeiL7MLCcbUbxkqzVHnNFPjGsQbMCwKRx8ErPM9hgvmprSRTh0X26ZDO
rdTYZfkF6jfnMHXPsbSjx5nlpGVlc/VnxRJOKmEFFFD3rOigz1kV8x1Ib0R1xDJ
Spkyr3KVxvFvHNOM3/GO1Pnaq9fngKPMaQANwp7TqHrGp3pU7aCg1014LS2NPFfAW
o+jRrCPgs2jBcIC3ySvW1Ug32S4UH8eSFYvO9XbC5Yp0EZxhpzLST4Xk+VfDT5oS
LFoSO+PAis9cEgeolVrSwNudCkMyku8djsOR1OHUCd5XVnN4aXGDS1KF8YvwbDiK
vvjwb6NO99AAMx9YMHETIKmUs2GHuROkegdjm5rLqgdZ0mXIAAtUM1vau+MBrcf6
JdyQYp6b1i07005xb5gI0nS7GsSf/5iSaBRJwPz6s2wrlsG3hIOkqBaq2GBVNI4i
3wZcI7YvqFs0DO8hwJB40+I1lKHsu5+SlivBaOnMcu2Pzd6xXBZ7AdTKhSXrBdhH
Ge/Ly/00AYv1cawVvhh/weWV47y9bSef4B+8PVMh3WT22z7FWUlDpkEb4+Ovv2LZ
MfgrhWZHeCqElmrKbn5p3CmhP3B3NqYkFeB9PT3w4niTe+OHYZbrmunGUMXQJSah
3A+sRI1KDThxImwKy8D5EUEFICoNeUIAC2qv7KyLrI1RHBkZV1Waf3V4Day6ZtJX
Q2/oUM/Z8rrUmveCkr5oXm0z2CqDsaUJEjR6VDc8wP35WqjR9LFL1DsRhCLwukQI
RhdTvxu9gdhaDuov9QXKZCgkKhodE2IGMQ1W+fQf+39ZdsZvgS1HV1GSCFmb5Pet
n3c7ECQVvQ7SwA6/IMEj1D71Q7LPNGVdR8mcX3+RW8duiFecerWDYakOWS+On1Sx
sUh9FAEnNdK+YI5RvUfeS0Gii1D3SpaJ8OQw1vgTdCetw2ID5rvzFTA5OV5tpjI6
CTRK1q0JzV2gdSmRuKCTr5zoAti6NyI9v1qhvZr+zwpyWD4RrieaATjL4MaSNJU
mYE961MOVvIF8/Q/wXq5XPDrkibob/ak3iWSU9DUQuzPCUaOtw+Xo1GoAn0RrMW
KucqrbZmfeCO7v1bdWxju2LSfh8LA56h+OCAZqQFLiXeRcFVsrRMf4OGvku7sWOH
x5XsHZv8mqEsC3kP43Yceby64n2mxxX24b27xbk0J8RyqAOVGSPPivLIW9R1zScd
/iSe8/DyE625H7qmTezaRKaxbh5y1Y3+cMJz1GvJLYewQdjHCLCTVbRmG8yRRPAD
7siN3po+WEMlBpX7LnZP02v3xicnwD01NX2VQIw43WYf/9dbGBnxe7uz/Gmt3yMb
E92nayUAsBNfPJUIz0hwUS1C1eISG9UhBvH58caoQkMU3CTiMtvhr53GDdgK/cxX
1z6yN5peJPrMML1RSNBVmFR1ZRvf/iwchvVdmJmQq1hRcIbkzWMmNtalkVCfBRcD
s1k61glZzc3kdpf0oOWNPcqp5BpVHP2znONgalrjyxUaHEh7dKrZluNDXmioUzTe
pCEB3+IMVjpv5Hvs59XkeQR11Lol3Viu2bwKhh6A jV2/6c0jcn+K9LDDbSorkG30
3q9paSowTBY1YiI0vFOECCXRScnUcEEwR6GAnr5kYjJQZXLzKOBx6BiurpG+JF
EOchCrk/Ob/SHgGqHpBS015TspZryVfrLDbcR8JxTIn9LTmf67Gyb0R9jWMKX1ku
5dbscuLlIFOVBOFR5iNOTE36AJdzPh3v+/Ws9EGxf/ptwIakmB1Kab04yUPvuxWeo
NOvgDRVPA1A9jmlk1bHGJKNYOwuv06rzedIW4EhsxBr8kt9R2ELJW1A/TzzuEIBI
ox2BaqT2u/txvIdpicpnuAOE8Ae1o+9Zm66oM3ODAcBxkHqk9GLh8jotY8Wt7o/M
faZI/znUQ8bOyNXhNriA1N8+sXZaNXs6enoRNNovK4mvNVevT3VmSSNTB++tMb0
whqgHyba3c+Bds5cymzWzDD9Lk81KR+40AkaE7j9CEGqAGpvYqcDcODtwuLQuScd
OSyp00p49D/XTu+UCqw3gqCDDC/mM0xFaOviJv+8P6KerOCy4LOGpJxnPjg/o6FB
LIFv3ihEJ4Pk0DMEpNWHHgY6NqcdX1gLPrsbBjvLuKAB4BvOrcH1Uexufy9Aiq3P
B+QhhbU2nalxV7ITmWUENWm0hZkwwQ1YDFvii9G+EhJ+3j077ACzF24IBkILTr
VOyQOItYliM24CcfuHtVmJ8St3RVFpv/xJ6hwmdGKxOzrPSNuqHhkLXGWXDt/xsP
B2xbmu0HBKUXhpR6cgtNnZiVa+6sSXZa3GmB/vXh0FwGhU7F95z4+5tKTF5ZxjaR
ItfRMxBsvxWjfrYvvECR9em0dxN0Anom//+PZHgt+2G0/tUqgow7nUfXy4DHRNUp

Y98wavg3qQRZGSrnK0BTsjtEHN9au3arzZ5Xp69g7URznIP5OimdiYj+Yeo87tU5
 EryHhybdIF4WVE+JoYxf6rDIc3trm/1r6J7obw5aQQRr0Vj8Gbh2XaaSFcyuWax5
 FTwV87GDJ05XiLP5hk25q87j8zbm+UOUJV6LCFXBmL8yXucztcg9/GlznC83IadG
 VbzQNXF9TVEpql2SF3aCJNrrEHUxM56h4aio2jQIxo/v+nHVy5pYYWieY8mgF3lx
 g8ZtrORYYV7szHzETbz1i4MF6SOQH1B7q4ShOxrZfLb1lG6gUPOIgox0nK5dxnu
 DFcfYFiBerJjsvUIwpKAq5u3AJdunp7XQYgMKjV0xcMCuHR/1jpb5eSLNm9AauCK
 obq/JL7lDcL0Nr6XxhvDiqtNjFvD3OzdU8XpS15S9LdU+c/PrWmM5lJlqg2lLNKq
 FAK1nXcNLFqVobOkJ6Wf/ZyXg9cXQzFVM6SbSI3lyXfmi4ExNz0iBo1p4v7v4lyh
 VEfUCgVUAoYswcpSnw5g1hnwz+V4hQJ7vLq4j3i9bZi7pIWCw1qI7wWgyXxsBS7b
 NZ37cthex2uleGyMZ9YCASqKRggUtFgYDQBoIK/aspPg56sgCMsubuvfRjHm1pE7
 JBmHw6oHXOEwGQBUPW8VPE1qeNxsTTiAToP1L/ohUkZ6lg5LSWbiDPScCAHv4EZ
 kJGibe1JTJK35hvpqFCZOLJ54psjr+XGgJ1juE0nyG0+bltVZk/mlGaHVzBur1tq
 Yvd708BkUIQ3Q2URK60iUi29j5dnS2t60Sf9+v6i3Fn9wVYyeMoQ3Vx+2ZcaNBSF
 zef+LuDFHPRMakoe4pio0Z07wUqa8+oc9YSoxGBiJXVLeW+NUnf/iCAQCUCfIhEUE
 1DhhyeVmduzHRQjroBxypREZD1i0xANfdWjzgw5E1J6AB3iZhBZBTHFAJO4P1QtO
 yG346IVA3nbtOeeGw64/79zQR3/LH7IdJ5bVa2UbkRDeX5ApmS9uOQKGCY0AW
 Acg176FtnOZ5mIDCxYmP18wy9KQIi2iAz/b65sauY49ZtYcoKE6z4gsfnrgIKRaf
 f7taSiGf10nDIjnkBeZ2+ZjdUKNc4r06SQ1SFyMKmqsgmGDovckQKpzmiZcwAXF
 MQEOX39G2FNtuVXp6yQ1Xux+qGjlyk7U7QW+Tj3Fwra+7weQXK4s1U13EUnTfElY
 8jmEa1kz/76brf3qCE79EC+HjKzxmRwkLcAKA4f0ihLUjHGZArEbYM6gAMqSkC8T
 9Clond51z9Tvg1xCqQsISZbP4o87T4TPzwXXc6Ut6cJkuILsgZwVDPgorvY8uS6u
 vACffeqKhs08h/VVEHQ98CHvt77Z2dMKCCdKJsHsFml05FL9oQaX6LauE6sJEcq3
 VJSNs0wSMkLZPDNg85VrO/8kHaAMfmLU41cjunocgqkLkIGvTo0ej4IiF0UrGpyw
 o1UNBcNjcy0IhIgj0CiYj3tX5VaJFNWUY8AWE4sdYCO0WNmuqS88iTitRtuRnXWp
 SAzbLvFh0wGU58oc+S82bLD3vNMIq74n2QuyJlB2mq7nWuYz1lLE+UhlnasRw3Rj
 3BMQK6aZNOT9uYyfwF3iWKLZRKOhDgozqN3mltVEvHOSjy1R1LavGw9V1ZudRsw9u
 vHGkeePZAwmC90aS3DEwzEvHYebTQGQ7en92357TOQDibUT21r6ZAJXqHyqD8uYx
 qAPtGRwNnspAFV5ad43e6FoL+muM3gyY12hhfbkF8r/6rJwRWt6/hL801jP7DmfW
 vc2WPBTA/OZ84Ixu9I68w4ICrBSN+VqH2NkHQcUQALoTzyYBLdT5oEN+S8W6oNyJ
 tQ1+UcdjaBXMblf51/tFazIBwvZw9VYyas/N8zPRK2p6pPF8opsCRE5Kq/cuIrHZ
 fXgzoXH4VfIb/zGuzqEIZfCHgBW/ELX3u8140rrp9m/EFKjdg+/tA9zVEYVQW9+
 M6E3gpLhddhZcuVfLxQYogXU+jIm9K0VgGCsFFNpMP7DBDfTQ+M2QGJkj0b2a6Bc
 jgmIy9Zrn69p9sC+OmPOLv8c/lyV9HGSdqLAWQTeWYAkaeGk4/rhOh2i6/cUVWCu
 NSeHLnWPewb6OnSAIvQez/VAGlgYiSNjdmktfKs1v9Qi+FKEIy//14TU3Ce6VamE
 JcRE2QTHTr2hFBPSVM2nGgQfJJEK5093YZ2kLqblGZf6JawN6Z4Mma2ukTCpNgeZ
 XXSft3CnJtUJ9DJ7SRlMH51wDwgS27YNF5SL4vn8HF/2c88Igl0+1yJvXBI48ZR2
 ra/aQ01dJRj4IB3Qzi1ByAC38xSmHMk/zxcH7j3Xxd9wvm/PNNxhcn4bfe4bseHO
 GiLE9e7eU/H7TeEzfzN4CC1J1YWDOF6t7Jw2AXSfdq9r0pIq2/mVZeQ+PP1PwRzLx
 uMVJ8tgy1HYd3gfMo8Sok3dA4/0pNTfJ0ggaM8+0K0l4+fScbm09JskmDhXW7pUN
 IhygGYLOPXcn1u6Yua3TpX9zTww8dKD8iVmwAVISrdD7EF1AD6MkQsA6Z/tFuRrR
 egyD1twvVSOGsykAnyuQfQ2YT7nht/4wAyBGsD//iVZf6VQG869Ng4Dje4X6Bh1
 sl17L4Rcl88LmgVeyhR1b/1Ru2rJtn+eFWJRRn/uOJjF5479W/1Kd0EMme1Sjiyt
 EgQdT+S7Uve5onHYlbjHETKQ56nVhqu6BigLBW0zwb49JA2GukLGJQnvyKED7u4T
 d23K5bx4Aq1P/w0UwFYV7qMS8vnhbhv+YOVaGTTQXnDLqvnMujb1+nuUL2jDD+c
 syFkpm6uPbi45bzzuLuNEcuh2Q55mLrEMy0hVOYbRaZszGgv+AUrLIfoXzTZNwrX
 krP18o3/IYDtZc5LdKSM4wZdk2jM1E+2SxvsdP5gRXc8CVwZ/b3nOkXyGzvgFUb4
 Z3rCZX4J3ZjXRkhjCx+ACp+ASuz5C7RSr5Uox4dEiWnUOYjS6P07x9OwYKjbX/U1
 QfhTQBIEsRC6xrmG15zLT+6CnBF0GalLwcPbLxRTX4auRJMfy5Mn1HX7sQL6jEo3

c6hUtmfI2fcFotqVgwc5yciX4Yp38rqmRhUwFDRVrENyyApvk/uRSolCxnjiQca6
9GPC5brfg9PRglj1CSCZmhA6UrKy4xuKB/rGmK12rnHeuL+98ldK4R+dvC067eyn
pZjuwZ9PpGrCKsmib/reUwoU9yB4g/ycnE4SG/C6NRjy6gILdckQN0LJtvHw+axy
3TlT2ua04cX9dvxLtxPed08s/j+1TJjcBjG2HskT2WuHHz5h0oPTSxTvqxfYwZT3
nb4QiIMxMTBzh6LXYA+gM9as0QNvJjKG+v5/s6AVzPL3/J6Hn2biG9hXRha/TntH
JwIW8Pg0Dp1vhhLq1lXG8UFCsv1SY82sQpnZORkhBfLuznHYp4ZgMhRBR8BIOkto
TwqaaosuaXihSFTXt387mLmJMs55N79cFU4T6bJLhwLmW1TNeusli0vRjN45Cx
6owQ5CDcxU0nNeyoz2HjTSD3EDIdRbMzQs8iE0vNVMLKIg1YTsTr20dFMTaE9TFN
OeKML7L8cI3PTZt+fUg0Ezfy1YdAKHR0p/hVW7kz1Qyti5P727yrxeqOQNGhiFig
SYqi/OO/r8xtXjNG4nDJoUOpRPEasOYB9EZM/Gq+VewG7G+JG8pYU7azJpUjXcKQ
jaq6IRUXnSuQ1zmyEicnCAZ77bKoLqe0cmY5NJ78T+R2cZFFLrxEjhYyGAd70+LT
sNzLqrrH41P6rta90BM4EslmLv7oJHchdKiFZYCXqZXYW4IwIubHzb4yNF7ntoki
4Yk6qadQrQVZjF6tlZz8xevPwyodUC6tNcqMT7PunPwUA1flXHfWksPqm/J4RqEp
CgQZdkX//dWt5PW6/vKUK87BBcC1ISVM8NFpME+EufXLNP/7GmSOeSu7qnS0+Qz
yoLuC+4FFXxB1+ocpvHf4i0WWfme7qP737bCmWnDpBS3XwUMwG3U1krRnKUTL+rQ
vSmW9vSX0Q/xDcJIX6d2Lb4i5qHV0/o/BtQiQrP3F7f+r1sI4EQiuUMCBxsi8Zab
pC7wd/XWms1TED0yOsFRX/Nd8hXakrgC9X1RyoJ+mdMsI5fqsgIKIRhyhRmUeJXN
9D3FAu2c8PyP+bWiy1w+0KrlTSFOT3FMLF6DKUDQYLp1Vm/stmREJFXJsw5+qxbm
rtFI9hQHJiJNdxFvmxVcvurddJIt/D7PUEALklIdIQ50/mIhTUBgwwj12705bJ8ju
xFi/YkULINhdbIEt2/we04QAuew6Y3mAp4CR46OeWNIEtQeGL1tSj3nS171OX2L9
gsxwKtFhv/33n78w6XiK821wJTrRGfR94ZDLJAA0yoiOZdAgORS8+HodrOgMuOwL
t2Zct6RzT8Ni+L8gjI86UIepUe6QVZJMYgDr/nisD+gegJhxxuHTkJXWYPuDKNh
ACLgHS5iMh+0hnI2MLcoYO5shOLUVXahs1nJbeiJ2onEo/IG9EsUzzH+oIX+hGSo
nzdTu7MyoBte+VEYtV/7QkTKuhUa51kTyUwM1vfgTU15w1OxAhfp+sPcHdAtdf+O
xmaUqDurFcltQbvjHoU/bB8y/Bw5Ie8Q1ugu5EVSaIoavmrSgTCioF64z4wwqwDk
o1sBx4NBtjkU16m/CGW58geCIIioUCAXD6EpA11knha7gBd018Je9fMM+Dr/URfJ
AUv4cVByu0d0cHaPAwplNg6+CK2duL0uHg3LG8HIGuL6NHhM7G2D5/Ltw/Wi2t10
NqXI/OmdjwHXJ2Bnt1S0/cj07shAgnWigp8PiTN7nQh8U6ZA6TWqPm2uDFcY+Voz
40PLNFLJ5akdKBZ1w9mjtX/U+Uhkba2GoKehjaVcc3Bluyk+ww5i8RrrId00S2Q2
SZBCSCOLjU1X/9t/MjhFNHduhUzTGKS2PUolez3Zpuxxh39tt7UuHp9YGH0/KwBL
lgKGVgHggeGt6fgk5yJAfpz9rRGfl2vA39y+Bi9sn7KP0CdJ158rt3qW2Ka8Z1Kc
IVOjzofEcveQJ7NxVo18YTD1kyYmxDGXBuHWX7CdNWM+jzdHgoL4Q87WqKHticB0
Y+3d/RVb4oHMVXxNpxFXzX3Ogqp3Nr1G1z/nbqzFmyokBms0BeyPqzGkScdiazy+
w80USxqJR4KX12xkkadNHaXkCvJgkbVQIi0nRuoZn1PrPczmszFrsBlUKalxPG4g
ziJ5kC1WI5PvJYxEVKHNn4dCOYtli6rrPC21B+RNiH2KXdjc8+8xKJ4QKkrJ5sou
COsGRntFRVyzVT30Xe5NKqjjsdjFWThXkSbIhDDMORY044NnKKvK0AzS0WHwrYe1
4ZkrPY1Ta71YMg+kOCEW8wKiFW34JWRq1lhJlqJxolDwNy9oWkKtqUXuZ2rnXRWY
knSv1FJu01S2dQHAXo9bOJ+CCdWry/9UCnIB/4xwwezHU7NT9stBLcJgIf1RtQ0
mRnjevYNQpB7W9HqVRoExm47+jTJInDir6/fXm2kk+sonwyulCHhPJFRhkBdchc/
Ad+iZ5IK554dEI+e3JQesa5vKtTRtsmBdzIyEpkRXNA/Xm0AYWjEB0KDUVmr7TTE
3EUKKKEGHIMQy1GrVMcAiQ==

B.3.17. S/MIME Encrypted and Signed Over a Complex Message, Injected Headers With hcp_strong

This is a encrypted and signed S/MIME message using PKCS#7 envelopedData around signedData. The payload is a multipart/alternative message with an inline image/png attachment. It uses the Injected Headers header protection scheme with the hcp_strong Header Confidentiality Policy.

It has the following structure:

```
application/pkcs7-mime [smime.p7m] 9795 bytes
  (decrypts to)
  application/pkcs7-mime [smime.p7m] 6246 bytes
    (unwraps to)
    multipart/mixed 1941 bytes
      multipart/alternative 1132 bytes
        text/plain 385 bytes
        text/html 480 bytes
        image/png inline 236 bytes
```

Its contents are:

```
Content-Transfer-Encoding: base64
Content-Type: application/pkcs7-mime; name="smime.p7m";
  smime-type="enveloped-data"
Subject: [...]
Message-ID: <23abef5f-8781-5c95-a46c-61e3a4464d58@lhp.example>
From: Alice <alice@smime.example>
To: Bob <bob@smime.example>
Date: Sat, 20 Feb 2021 12:12:02 -0500
```

```
MIICPAYJKoZIhvcNAQcDoIIcLTCCCHckCAQAxggMQMIIBhAIBADBbMFUxDTALBgNV
BAoTBELFVEYxETAPBgNVBAsTCExBTvBTIFdHMTEwLwYDVoQDEyhTYW1wbGUgTEFN
UFMgU1NBIENlcnRpZmljYXRpb24gQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIEh7j00
Boq0MAOGCSqGSIB3DQEBAQUABIIBADPyejv9Q41LGeGoBdDpNDv6uYtRx1aRJOfn
f4sbWXR40/34962uv803XknQUPyKkYZ41bEXBUU3enu5MvV/CQIbHYBIxhYmAMD
vrw41JyVFN+yH53wtubTwSC8poa2TtjNv2S4nEgbsDQBbN7IR/DHKqCbUK7Am5t1
uuSHgMWpZrcRkUmBlkkqYym/kYfK51FnZbMSODJESjwQordhXJqv1RJFG6T0kw2a
GOTxsg7spf/dDxEyNMnqm5tLOArFLKOBxcpbJBPTWumUyKh2P+d8D/8pSGW351u
SVEfw5Zw4zX5klwBKLvowk07vI3oS1u5DKfQJ/5WOBucU0EqDGIwggGEAgEAMGww
VTENMASGA1UEChMESUVURjERMA8GA1UECxMITEFNUFMgV0cxMTAvBgNVBAMTKFNh
bXBsZSBMQU1QUyBSU0EgQ2VydG1maWNhdG1vbiBBdXR0b3JpdHkCEzB8R0APhiY6
HGLS64Mv1sDXhpQwDQYJKoZIhvcNAQEBBQAEggEAKK12BJjTcV/7qS94c1NAH+Nc
slgE+RXD3UJ4VQM1fu/X4uszwRQtE8eWO/ToCVp/g+WSFZIZDRBfhhv/7rFDF1s2
oRVHpoZr00sUrB6IQB7R+5WCueJomWRjYybAbAcFSuff3WzC9sh6o+hu8p69lnJm
7/ht/8X40bRHcno/68mPOu4UEl9jOphAxwAzVPc6DqAPztyBvTOIERp7JhfYUy9W
r0lWxuYsVFF0Z5NI0ZRybPAJPuBQUM38S880am6CxcgKgOR+QLy/s0HdiZQ63tbXG
```

NcRsbWIHMriC6xLWHl4cmq6VQdnSNGqoHVaqKAZlQjDgfwP4cQ9pFK3HaJJ3bTCC
GQ4GCSqGS1b3DQEhATAdbglghkgBZQMEAsIENbAf5M2+FbI0Ai6GKC0Vf+Aghjg
88tKi0DrMGsmUZonZvPq/tu/822mi0P1iqCEdG3Jby4dU+exxrgn3InoEZQI0QcL
go7Qm0xpqdSPHFp0ZPf3qDAIJub62gC6/kvshuxVyWeHySYp9qn9nwwesP8JLGBJ
iBqtQEjeRZPx17A0oLLalGfe5C88Z/zM4QqL3H0HuJzBM55W5pCm5Vv2fvAnnPw
q4S+YYV9z05elpolx8dQQm8+D3RGp+Dlp5nd/yiSgMSolIBzhnxK+jkPZ6dicKXV
CQwyIFfHB5k2J74wsGDYBqeZKIhGZuXEL2YQ9LwchmMv++AjoGOhXnoYdStCh7Lu
zI7eZqn1MriXFXJ4rMd58BXYByFrjDMoLiMXCD2dZF6wrCPDFEcktaEA4XFP32n
hkLdX6o2r+9uDS7vJX3RvcYvocXdk6VnwxB9664vLWuczw6BueYitlxU50dlS29
v7IpFTrcSYJOtqftglY38+L1fHGrfd0EKIeYlKeXv2TbT3ZDpiZOVe9Kuc993XWl
+5T+JGV02jidWgDgkP55TRnz+F+i0cowve6gcRrDVM87ECP/4qC3mh/stlg/AkvY
y6DFD45GTLkrMqeKcSHBo06jS3D+/BarpG4X01tNuHuoGd5DVhURSMNQXrtVxA6y
ro5iupYlJh/00sN8gHoCcwSq4v1Y20CwqmSrfY+8FhfZXBQA4sHP/apBVJDMTrgq
MRBxrZUHNmSwEaL/kFVMzNDPrVjU8RP2qqqufkeLU6si9+sZptEEERaqNWUyWzi
X6e62jWaxv8qOnuD/6zukqOx7tsQVpiJYPHDw+tVd76Yvefe5UCtp4/mBRFqZoz4
tZFM/nC52VuRNYDQ79h4YPQTryxvrgKaGEm5xDZLcM5MtJUylo/sNiK0yoZHVEix
d4DkM5/IbEOoJM1zhVp3fDh1qkkCEf1yKLSYV1HFamAN22U8ImlGsRSnZVmpLMM6
GuV78wRP/zwJJ0pYrwJf2SzyXI+K9vc6fZQUT9oLCV7mwRRuMN17HMJN/Qi64lq+
KaL9sTZKs84Nu8jAmjGLD1lKbvpAXJrInlnKoeoT3560Ih0lRHfXmh12ZtTl6qc
zUPROCNgcDePcmKwpUrS+DPsN9VznIFqWVsi2bsGFbA5pRxTiulA/rTgIT3/ToA
M4gp2mANIG3dtxKYDnJskUn6LoX7Hsbf9ALS12CFrA9Ma1o47ILNMhDzn8foho9e
do6cIw1LY/lbaxjh5sRFe6IEYI8Bsc5whhRRX5s2cxYtQprYfr+HDYl1LwJTOEF1
JD1wQ0yEBSmGFNE0wiZPM/iquwnfVsackMwFxf2eC1e0wcryRSH7qfTB69gZYJTr
lGiHxr9TKeKduXvk7Cpeim/SQkgi3cA69dwezdb1HAKCcb8zjpGp4hjHWXAnTrwV
kuf9s26nTClJU/z5XTJc3yP746MKHe+G46Qetn+h/DjvX612v5VKA+XEavB8eyex
5NYLFh0/JQ4zgvNB7DwW3T1+OxyQ+rcp1aj318sMmH7Zkcvk8Z1bn12TAHV1w0c5
GQnUCeoJV0guIC4KKjgneu0rNhxWUD7WczWwm5HIAvgq0bjZu9dzCIozXK9UJ5UN
hp5W/G5MLCqEzmweTXGidg9UBuOBRWjaAEoLsDZyr4E1E1QiIzY0VUoIRh9qt9tO
DvB3YksC/788W5jJX3Q5G+MjX7kxm4Y5fcXWHU7BwqMLKkpvy2qCNYC4z3rBPAGL
ftZ/sJdkR9uvClh9X5zU+JJNAE3R9LlDFW68cUIFxpw+bx43BCesis7r/plhW/Vp
4JS1x8am2uChAKNMQTjHxeGuaxEvoBjOwTT2D21i1f2KJp+SxKZtb9bTbJZvt/fX
/8nUUR6VdwSfgxmOEF+JCRI5U/z7V1Yv908BZ+wF0vvNbFGsFdR1UHEGusARNV1w
gRO27cfkJO1WDkqYWzWXXvwoTbTgVJ4i01GhA6nChdatU6m5nR8cXrUchXkZQ837
OsNAUN9sjsR11PA+bwM49kF1kysRRv7T1+uZ59hw3P1j/en95+GNORsJMwUJfAZp
bqx+8YdB2szVpBoFYy9eYmeAajdO6NYkzeXvYAOp739iFs+DQSYvoASUr6CxbBzi
6d3LO7c+OHsgGTiocgGtX8qcP7T3rHdd2njfPzhr980zHQbESJ7TazOsS1YtSOB4
5rL4nWDXBFqtd5ugCfYmtbMdyWH8xuOTPB7yCUjUI9AdnTEFGPPZlco+YHHCQMd
7K7A6C58piH2qzn2NuFcoo1+4uwVittRGS5ETrpEA3wPjiNtO18wt72MbtSygMec
36DWhQpUnnKOJ619jbQPooa14Gc/TlrsLtePsoJEi5UHkXiuKEVTH2yjp9RdlRYI
7YwecLbK5Hv1+Tw45k81X9IikMftdPbQ4sUanD3ErzKGOvccIcEQH947ZI3B1VJT
Uvah9ODsIdJ1a68GGJwFdydJdHI0WUqBiQ7190/33x3CzxtOTlte9dRkJkP751o
V+vLRDQ5HHChOSzWQ0VeGAsNa4AFgrO9HMcMTH5VYeeHVMZBCTKGpuC08PKehul
rxY+tQ3j1bYgPdL40IdyNFCVD4edYo1lsDkYofsGEjcv5J3umLHWcfLOSdcXylcj
OccGWIz1mAwSeOT8Qqk+8vM6fRKy2neC1QB5L4SFGrmnnVPg91KBEgaZt6E00MmlP
Cwrby5VET5Nf/w2jJWlhxnViPqlg6E2Zx9dRRTriGITd81FA87+dL7xZc+kt30tk
2RhG9yLW6OlIpBQC9akLEAlqq/ikJWziVrerWKZ0XQ7IAh7c3Q9Au83eRYqL3jEb
1nXN94Di6dfwGXi29FwYqxzkgz2P21t6KcpvrRIBk378yKn7jLVgkil/TEzQ5vce
quAinWS2WmF+iOaW7nhcIpYBO0HZK1DYSSlAraK0xvydsZTO5HdGkynJ6ddty0z9

j4Khe5VT7X6hrqIkOcfogl2GNXjelhNDUJS0YKRYvda56b2hbn7ppThsaydOmdG1
HxTq+/9ENaBmASEqcgF0/RoJ5ZcLv8+fww6qmxkQI+GG7PLyseI0GA/Zy/THHoD
uhikRUmY8eFAZNT10kL/4w1GFIG0Ik/ZGVHs7paRJhWeOcFhnGHqQ+4q6ocGcWmi
AZGIgzD7A7sb0zKxtbeSwwRqvS9fhussCMA7avcDni6WCVTxHSMnV1wCzM5CHEmt
rYQ6/kRKQ7mkJ7xWyHuKDb4e93+ZsBOomaM3AETVwagmeYiMKG8Ir7EswzuQLkau
Pe5qh3i33Z3UcNE+4jaD+Pg01LUOHPmsGkTi9hJSADwC7bZpRse52WtoJ7PoL0FJ
I/SNdk9yzLnDLPiOziNQiY2i+rLI5H2B1RwGRspyZiRw4MthuP4A261dhCscP3fI
TY+DQ9tV5NItvRVNa640EoX/CV/bwNIV8ciFrsGgpVrkAD7gmSdrK5IUxUEUaYh
v6LECYSmICQb1n0A+GxwCFrPWL3Ls59Q+8UxDjyqcPUA3A9jyz6GUGGAWN0YQxp
DXHHafrIKs8p5ixcjVili9Lz6Hni9XJGZC1Q+nxZQm1C5h55jft+UD0b423beluT
2O+M+Wenck90pxfbk7IPB9XOvBTj/WNQDWFbt2t2wzgyxZmGZ4x3ULMYHlyqGlu9
KpGu40w+3pAqtuF1fiXW2yBiv5exC+/vz/mfozBnW3PF7BpCmwqHXPP0IHwqCL5W
qtmnF3rz3SxUihGvIwDU/POC6PEXGixbP4xhmAyVH9kxYLOEK3Jil2QpL8UHh6w3
eXJwuztXaK4HUQHlL7a33LDRtI+fQ9JPfh1bXLJJsLw8Lor1oBgjV9CR3D19ESff
NFUj96B5QPwu10KAA3G5jtrBoNa0U+PWxyw3CUhi4d7gsy7eXpGJcc0JNgY6P65t
kXVIzY7RI6zGg+4RFES3uiaxG4oUyfIat4YYGq50ox5iwmOQgav6Y8CkGOQmZUmF
49CiEvsxVUxzUsmESGvvTXTeOsG550DX/XqyG44ieigPjCcMjRTQw2wO2CaNy1HC
8jMIMiteoLovVqThlAmHBnK03EqnOcrJ0isR5JHkv4WTpML0gU+oEkiDhJEKymqy
UAgNkwdZN+2dc7wYFSj8U3oMnVKjtQzgpRVZsanuMmTgaTlhY7+HmS15M4TjViqb
IOJ+mJLVyNr18zvp1hl/pAl1wepwoihSO4m3S0IjU+JWproQm6EtEPuW2VNfmIZ
cggeDENMq60qS8ZoX2wPulhXge40lFNSKHividiFYCqqW5SZ/obLqU6aetzZnSVT
KLfpQDqib1Izpz2wKJXvBiZgCfIp2gRLoushp7v57DoTlG48KBI8/a8b9x1CvxFVk
1Tx0irCIHSjcnI3OYSPURZQfZE/RZiiyxOrnMiloa2wP1lq+z8mDFikKcyqHNL7W
do3FS2GDA/hj5GJFV9SEtV3vBUmdqjSxyA5skxAXMleHwH19RlpoDmpAU5q/4/hyJ
8NLVJ6GGOZfjbbfJzLdh75qTgjbCj/tW1W0ChzhnjXRN9U2d4YCR3Uke51Soo/Fw
Jg5AZNo51cuygrvWAljeRgCmDfaHp67CYonsr4VuWy5JpuI9/lSzIk/19C0U9qY9
wh23xyRz5rG/9NfWmbh6auVHRGypfQAGNwwjs1F4hIFAAJ5WkmbPSRn+7SVMLDdW
FYOpNclimbnfapvsU9cQiTxkRB7NJfgazVxd6A6h/1rOZnmSuUPou/8NB71F9Jm
1rYt1Op9TF95Z9D3oFwsmCjhRAza/tlk7SicT8K+LJSGks+0yS0KvH9Ebs0V9jMC
vBMzfxEEVink5qvHNe907T5iivAf52jnTYMwVP5UwvNnseR0/q5/Z0dseLwqYbqS
BS3NRjHaV3c0Y8E+Koc4+1RrcE3w6mv0Fsu3IApwQj4AyKd7JDwsfzsz2iv2Upe4v
RMCzS7Tww4gy1SIEjqlr27iXgi0kR4ehLChh+k9WbyewNYWQWfJqvqzFT39ormMg
dTJDCQh08cUvmBf1MKimg/Tf2ng+3SvbnD7fkb9mqfCHzfQlmsRrwp7amGRj3f5l
CfMywN5Bo2si9UrKVgzMaMn10pIXwziUbsqiGyE9/8SqdlTbtVR9/x/XFUL4eEEQ
dUUCk/9qBkB3M15vquva6BUVj1hhiKFgnnpZ8eI9o4RL02UfBJRtgBzicI9I1GOB
+Dfveo85TdQLZB3duuEo1RMrnSKre0Ki50xp7I80guRkie+++71s3wixp42GENXb
pesxCaAZWreIJoVqFsqJLkpDhrh/C1VVC/DlMfYROf6rTKLdFsuJy1bxEEOXw1Q3
DkNIgPoy7x38a0TUj59t2H5xbfbQj3rRmbSuhVWlYgeGL9w/N4NXymW0ixs9QxHz
Y15/X+cYWrOV9zLhHvjhYAA3z8pevd3v7HgYvyayHH9FAOQOzwtiNP1DiJz9zVQy
XxDt1m9Y+rTdVxj36dzUd/EVAmuIgh7HA5TdC+2fwfcoMN+4cyFBNVw/FhnhvqY0
S788MBOudK6UPbTyPte9szSqkdVRLzTtjiURPGf3DACDPOVu7bzewbXN8f+KHjDK
aSdLktQifgbzdXFscZPOYHQXbs9zvztTU/xCligjvsDK/A+exn8QuBuLnumKZZz6
vW88zNPulJdZiqdszjEQt5TrMnSVBgcB3TeerA8GQCmgZlgnN+Jy5PIQHTz4oLu
mp8ZPBWd8DRsT59LltnWYKTDLCYTIN2Xx1YomfpUQDKnnvmct7W5usjD8VntHwOY
gJ3J+Rd8xPdQsnW4/HCX3uTjgpp/mUTqCYP+J+226n0ac+jdfDmi/otRn2jE9zvKG
7gKpFu/gGfXZvY8OUSdNP/h8+VctaUzbDqkbNkIIsyharupkDBvSJCW5qxybXB/a
k471+F9nug6jdyIi3Hqp0FvubcsSchYA1UP9EtUg0ae9hDB1tRY9GT1AaObd2xbI
zvvEBEecV1TlzaY9B6XaTG3VIt40i8S1BrDlJh50jc/qG3B7X3Tk9Vvyn2N6otF5

```

nidTIwwJ+HLGt4h6c+YsV1WZ1PZDta3n6/HNh/+pAdwSP/2t43PJMgJ10lS1xR1I
C/OUgu7gNndyg9sm0j8rpPUz7p5s7cTP1zGkyZ1VzEAc19dv2RFB4TV6z9h/BLWI
TUfx0RcH6Ny4mvPiQKUAduMHGNZoOHXEpsIQPvpqL/XDXeEZCgKIH7nZiaoirNWO
OG7cJU3F7Ko0EejbSsrG2HJVrDd09Tlfr7HP6/4Tu3h6qox1TuINjNCWs9wUqdx
3HNzXc+0JAKE1xiuoat5Y/aGnfabVUVB29ad8yFPtG4cv3ftWHM/N87Uezeni6f4
vsZhKLo06FcJ6xpmWD0Y0Hys1YtukQs8IhuKNYBBRTNFGrBlCqKJVn7MIszivld4
NGgmDpVQ6sgIr8EbIVVsQC/0WgzON1hsfLvweYfd0I8AaVfPwD39Q/y8DS1Lq/yq
of7KgAyObSxxqumY+hJwW71VuFGFiRiZDYi1bd0RaVb0qVnRF5pU7YkXYwby6wzF
77o1QUVcEoXMJvtWLnU7h3mI7fQ5F2F4a9bc1LGXDcNMHsfh3JaIlhXkmUbEyrqF
EBOuotyT8Jtz4a6rSG8vLCDEjfw/DKfM/2vtAg9CWb8u1Tj8Ir0j/0YP01VjNtKe
dQmi+Grcts/5cYbhewOIaoaD00N2Hy+7MQLMDrHo/NFlrCHtLUT+B0I7acnjAdit
v202eROGGQa9YDjMZ8tMhHVGyko46yep08AWm5RR4vVd8b3CbvFbzJy8wIGIB1sE
5D5rvWqgzKcVV1xRneE5k9uJwY7CeL1DnVX5Sks4mZoxgabfQEcR116SB5RFmSW
y1CDnTwMg64WCGG8XCWMnjEydtEGK2JoI1b5Zikor9F5WiqhQ29Ropv+CjekM7MP
F71W0+C0iB9PaQsn47J5Wuzhdt85RfLpCm56r57z9eMctbGfmhU13YMth9J71xOB
NZyBXUnAzQ7qIaOuFJ8ZxZT3V55hYAokF/Ph+6W/rHcSshEb1nzUQ8Yf4jqjLmcl
S9I1cVf2xkwWTS+6+xOMoEuqeGK6TF3brI+s8qmnimIIXYsspnzNun6fXcoXmh
6TOKCAoCHh3wWPk1lucj+JzK5LHDUhoBzccx1co1Vf4To9Lc3X07Svh5L9ZouJ2IM
NHqP5tv7V3dCyPfiLo4R0LGFQ9o3x4vQq1Q9Tt8VPi++Z93H7SqIy9/XNYAMtp2b
erhoi5Qc7p1zFgMN+oL7cO/r+jM3/Xt4uBdenLk1Ws9M9CC21Pg4vLvs7f5XNj9F
nKSSAqo/zxxnqrwsfLCEir4nIZaOSmQvFATKAumiIq/Bmljy3yJaNFhNuo8k44mi
6C5rChBO59FkqFJI6s3s0BW/ARDMPRzWzZLqEiaYQxXrvh/YWatmzdMcOGjObivG
R6cgEjJ3ycfymZ4c1/dQVqqeNGSfcuumI3eimiIq4txhUFaSQwkp8WI19n0yBnFm
ygdePhIuatf5n9yuKNLbTxamlog4Kd9m2iHGp9oYETf4xt9icTvNa1q6kEjkeJ29
jA17hx7ws5uAr1NIu5Yo9dmgzQ9c5DToQr3TPsNM0SnNR3S4nujNc5zyAybkGD9N
oirZ0yz3BMyWadhVACK26hYMEjdM/eE6Va2M8yg2aLXU+d1H+hR/C4RN0v50u7L
xnBmTU8y+AY/vb14042v1TcvL5IC0vOG5moFRgUziCcsncVcE1h5EBbwcK52dvWt
OCE0JR7HV323h/mBe2uMdCrsvRSdIO9/VqTU9PbVb13xGwz/mXpQrRjf/HLk1Bxx
8PNZU6gLQP7Ktgo9RTKV4ZgEcbsFrg/np4m0wb+wQrI4d6XX1vHMPit0ofu6M/e4
FoyKwg0Jf2Bcfq33eCeTa9tioa4G7d0ML4NqZi6sxaGG94XMMzu9nD6ewUN8hlxa
mhn+uLGFie3y1EvhI3ICCeJnZNfbPU5bXq8zuwqp/YJUUhoshBna+VO891W217v
koo01YxZB5GE/BvngnYDUPY7cGyutF03uRoFOHmc2Q76mW19hgdc1tFfCO950nre
d0cNqrMsmtryp7tJ7FpsD8QE2t/jWG5P1Ck+m/8GbeRk2qimvkch0M2jSIEUHLTr
ZNxIQ0dVtrMTtsLaATMTG1sH/AiY+Ajuzhbp10G8YVilyIYpxx6RSpRb6hvpLqC/
xZy4kBsOJfcpPiODphgcRLyNg+8ogdHwg7LXqT8vHQ6t3wfASSV1wetnWCQvfB8J
XjnBSSUXoTHhghvvpJ9SXxHRiA+XHgFYc6BOAepLYWMCuIzvxTweEsy6feQynVKWG
p9DiKuvC/v2gqse50u2E+E5rPQuTj8/SLrGUBw12i1TkQhUIYZMI0HYBDFxu9pyD
u1zx3DsnS1LWTzJr//wkr31Jd5L3WUerfEp4gAaq5hGCqkSZs4yC7YfnjiNyGWS2
FPFhOo2EhGBGLHCO+mSSYxMnkRi+sDUMzx8dljVByeM=

```

B.3.18. S/MIME Encrypted and Signed Over a Complex Message, Injected Headers With hcp_strong (+ Legacy Display)

This is a encrypted and signed S/MIME message using PKCS#7 envelopedData around signedData. The payload is a multipart/alternative message with an inline image/png attachment. It uses the Injected Headers header protection scheme with the hcp_strong Header Confidentiality Policy with a "Legacy Display" part.

It has the following structure:

```

application/pkcs7-mime [smime.p7m] 10380 bytes
  (decrypts to)
  application/pkcs7-mime [smime.p7m] 6676 bytes
    (unwraps to)
    multipart/mixed 2248 bytes
      multipart/alternative 1425 bytes
        text/plain 482 bytes
        text/html 634 bytes
        image/png inline 236 bytes

```

Its contents are:

```

Content-Transfer-Encoding: base64
Content-Type: application/pkcs7-mime; name="smime.p7m";
  smime-type="enveloped-data"
Subject: [...]
Message-ID: <9cfcaae2-9fec-5aca-9a29-c98da35b262d@lhp.example>
From: Alice <alice@smime.example>
To: Bob <bob@smime.example>
Date: Sat, 20 Feb 2021 12:13:02 -0500

```

```

MIId7AYJKoZIhvcNAQcDoIId3TCCHdkCAQAxggMQMIIBhAIBADBsMFUxDALBgNV
BAoTBElfVEYxETAPBgNVBAsTCExBTvBTIFdHMTEwLwYDVQQDEyhTYW1wbGUgTEFN
UFMgU1NBIEEN1cnRpZmljYXRpb24qQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIeh7j00
Boq0MA0GCSqGSIB3DQEBAQUABIIBAFyXL5Bdsrj47hCSCMZg5HssJuT0Wkfqzrt+
Uauk+xOG9fu/C2qZF1c6itV0sTYKogOf2UOEetIXbj4ad9TeExHOn3YdEbbKBp0a
KnYn5zyuaRc2VmBGwCrAcPaGLHL59ul93+Quyvp6t6T7L+y+rvgtOh6tMsCH2yVp
TGUj2FVg6FxB4kg63f1FB1ofpU10wSB8nn+dUzUqxP/Pwvt0yxhB89ea2+3C4nch
36wQPHM71la9981grPRH7RHBCwDyvny0LPipQ8v9p8bweJyVQ4oDqLdByO4XuNzL
XqZnTKmhXugkRs2pShYJa9P/YnVf6fPhc9mlz12R0UXZ00ezMZYwggGEAgEAMGww
VTENMA8GA1UEChMESUVURjERMA8GA1UECXMITEFNUFMgV0cxMTAvBgNVBAMTKFh
bXBsZSBMQUU1QyBSU0EgQ2Vydg1maWNhdG1vbiBBdXRob3JpdHkCEzB8R0APhiY6
HGLS64Mv1sDXhpQwDQYJKoZIhvcNAQEBBQAEggEAIaf5NTfAC/vD/MMeOH0+9ciT
ntt01b98ds2zwaGnUR9B567tVQjWS+hXSWYZ7BSdp4Mnt1QyeIsFadrHZp9RGnXS
gxfzpcBQm6400PesyumvXNwJnJIsGFScVJ2cfyFhdH8DM8yKcDBZclueiaTDTHXb
efDBndblmGaJESe99TiZSWu7dq1tVm81u7NnPdY7yM1IHPp8Ij0mxxrm/5pXN9Nv
ZK0Q1voE2pBgqZS2gzIoevepePkveqNYsMk666ThBmSR3RAelucLaRhCdGJ1lutn
my00M75Rn6A9U1NAEUa6HXXqqIx4G3XeRFvwjEX3gW+sd2+qlzNaIOK5VKVPDzCC
Gr4GCSqGSIB3DQEHATAAdBg1ghkgBZQMEAQIEEOSKKKhhbXhpND0X/10pLf+GAgHQ
fImfWw1xwLL9IO5jrbrEB+Nwv/IEPx/eZR77kGkohfz/1D2J14obHrkGO8DF+610
dlcXUtVeJ4EJeQdugoF3Zf41ulAF+skxo/0kbTZuReffOaGENU3beequQ0fi8yzd
UDGRc+HvYpmKFgy8YXdNexYYXaDGxBcVHx4WrPmczGeLE3KdnR8BR6630xU8zcV7
zmG9LH/7N8JimcVvphNpKpbgC0W4vck1wsJ4HsQ5/5XQ9bIrXvWxDLqCL7wNjHfY
MDHc582aczLwOcb/RVr83VN1JDLGe+FR/snhhxpM+yFNblpXcZiDnzVwpr/kVE55
B8Z5P/9VkhU+dG3opNmronOWOgoUdul0H4BaebYmIIRzvFFWetRSYmh1IZeJ2s4u
dCc1GclveZBB0fmXWYRjFlmbEKdo5vVN/wbilQaIfjbm4iQ4YkZZBmRFhsSqvlpm

```

GqTE5pm+A+4oscp+dnqMGDl0jzAWnyN7tlbkIW5vYlcnOddRpT2r93ZRZ/sFggog
 pkoYY5q9d2Vw+ghVPS19wToG1RoL8GuQ6SRTq8FN+vuJjT1dfyNhsYp7ia9+ttq
 Y5KdR+3e4u6SmVLWudC9k4jsglndrKNqXvVyd6NBPQpmeGaXGXhtQkzy3FBBfcsq
 mjwgKUmkpqsY2a8YZqRce2PgCuDSsXeYZvVfs0EDj17XnPadzjIBdLl9oUGaaD+i
 3q6j5y7xbyvjGc8T4TJCss7z50Louuxw/g5VBHHDz4huywugOR3SICAOFn665uTS
 zSXXuNi+jIiIaLOtPttqmOmPa4KXHZhQFiT/F8D578Wnt5hGV6fwHXOjvvi1JrsJ
 k5X0Eid+vY3THdmE0e+IWog5ViTK1j4Yc6I36CX/Ek8k6sjjLl1qKgKo0XXon2a3
 2MdZxnNuSPvx0EZ7b3GE1okJyChNPguG3J3yxOLeb24pQ+jDLmka1X/pLALIEZtR
 HUks6pNQ35eYoULzB2Sc24t3Xk1C2As9dS6xVXCxpoC/2f+SDOMJzCDi+3Cig+dU
 SZcqHGNmKdp27ScsNmtgeGp4qKPB9EVBCLSYHdWwuAlhj9bOuBC6zAEMfr4FnL1r
 bH/K7K1HyHjBwrIZmfvbEOMF3CYdX3kFwUnv71sqfwW863DrJpW6o0Fyzi9zecJS
 MHdj1mL3t5Yp3u0+z9+MVJpfgJfv3GDwoM+Cb4s2+kH/P101tUdZcAyohu8gcylq
 eJ1mfgrBBSILwrzLr0egML0guHdXWp1LncSswqYm52zcUWuo2M+gz2/vD+6t16OV
 Ax1GZQE4Vwwd+z765wfiQAv9OodQYhrdX0zblgdSSUCrLI/rc8CE40qZQM2q+Bx
 ZVzvFLQI15SgQMZ59IjZRCncOsunqtnN6VqUrbOvqrdYBFHjd9VI5qTL8CtvEcJW
 EBw5nsz2dPYXYjBzKqsxYGVxeEKiNyOt6XhFKAv2pFiiECi26XbnI1Pcq0BU+8iM
 KTV41Ku2lGp+DVftSxaBY2ge/hyYpFp5zTPelPSvDw8VEnAZn4BzFd4L5Qb5nNGh
 MOsOg2nbU2CFZJg7515qTODfgdeCDJkAbdJG+3g9Tp9rMb4tNsZ1j82OqoCHY4eE
 iHgw96FAF1vR3BSk7w0rNgAbCEt08fBKAdqp4XOivsNk6ows/3E3DyCuZdpW0hWN
 RZ+fcmWVgFaZa7hJAiiQxeX+b3ClbHBuEeRIPHns93uAA7Y9+Bi cm+9lp51Mwefe
 yEW2wH9V/d1vOPUnUIV6VSe64vBlkdbfexY8/C6z3owe6KyLJHiDnLK8sa/wHORP
 3pXMZ9ldHweG4pPeUmWFfQWgYDufiH2uRWSe9qLImGdL01yuKvt5bV8OznIGHhdn
 wW8GGIwZtzpL9IfShYVk3RAAEfUVO1e1B0C22fNaQZJZf0FAxByW3g/nkVxMw1nF
 9IRkiMWWYXK8f73YVrxfcN/NpJKxfkm60r8UrJKapDbbqbQ/phLVPyEufa/12/Q1
 qKklcxvTUIyJwnmMCUAv5P57QpWCmpJVhrZKJGgmquf2bjq2UKdtnuMJNcD5kV1z
 Xa+4oeSEFAhkhkDzoeJPCGrC8s/+OfObp69YMMlRl onrbaAOuiVyRL21tUpR4Nexu
 YVEwHRAkwM0L7qL9dMngEv/p65OqsiXXMuhn0oW2QaWP7YOJYCvrIZCDEsMiwfzW
 TgDARodbZ6z+X4PLf9xLALXZEGanQwc3Z6nz40EnJAYN5FKodLjMGUyXAtYfkUTV
 zF0e2RqVSRQ03/0Sz0nQEjgo07UhHIytprIX3JKqNENqzQFp7TON9RpTTgykxTT
 6Drz1yn/daFZubp3am80Hg704V9CWHGKi03E/Pm09UcQsb6cPbTe08QVzd3074ne
 unrho139p00UdBSiWAl1RcAcBiH2Am4g4ILgXMX+E5JTTUWCxUbtMtK7QXX+YzYu
 BdwnmvzNr4nLgM24Tcq5c+1DqT+fxMc8jy006IegdtABgGUqGdW/0jfdWID+v9Un
 FTf82vMpYCwzmeCX7/N4BAdLHBA6tjWQdn0kAhZ8QWNXO6X5TGQStEtpW4zrSe3s
 QWjJNN051ajQKX58QY95z/PntOWUrTmWC+pJJZhxFVWBAFOf1NKNse0WB6OFzbqZ
 ClrN14gCb6tWVR/F8nIJqICeOQHMBs9tFyl+FksXg5WwmrA4kflorihz+I9AbhBL
 PD5jdJJaGZeLYP8X1W0AaaSHa2p5V/cdDumDz/rnkzpbia9VN8/pLy2aWsvQE1qE
 R3Fxb7N8bU+1c/FG+ekaiC+mzBfaOq7WimFqk7rKV0gfSeHXTQVolKNceeIowKDY
 9YeodW61yVClzWypFh4x/icE2xzD+0hjm/beUpfUOCW1ehut9dwRmjUjhwK7ZivL
 rC4ex1D5KrT9npqcB+c00wy1ghr4xjn9xpiBIFmo4NJ+76777Puu4khUBuV/zYav
 fkupDpG7M19n0eX9x9oXQSLeEDagQXnqOVgxbOgCsJbssADsy9Q85mDqc4jJxc/Z
 MunEcErg01NIEOeu9wx/yiNu2ioPoVvIUf9qRzh1i6cZzpZOVkpsfC7KaunfyU9a
 BlIuZaI5ZclbeuLxjC005tCCLzpltdLNBbXAQzQEDz0CNDExsXhvsMq5oPwWbHbF
 IrTtyW10UYhiqfzKqcqjL56sd4cPz0AebxHRbi4TGWjG41lFkgtTjle4wRK+EGWm
 JtzZ8DwgU6szLrEc/R36Jc+vFNdi4+UE4tQxOior7/yRlJeaCjiWcliK3G20aM8h
 s0Yt90pHZc7C0c3v3ls5g4i8115DL/qk+4Q8PJNotFG9ScytPimD2SRNcHK0RAH
 mgEe5+MJUKxRcTLM9kXYC51cPn19yjRoJYVB68kyaC5sxs1DqS2cTTN5h8LymUGx
 pM1PUGdmKF+AV6ovcV51YTqm4FivtYFYIDfC7wSfgC9trWeFZuhNIjzmWXzYuTs
 o80LveeBRAfMgIbFS3fBQ9EiUs4IIuoVGoG64vg5HG4FxpialPHDdHJB8eT0CZ7i

XAQK/ml/DHino+SE3bNUIArL27v/e59Fc/USW5BeII6hrsmRhJgmzDf90Aw641nS
DKUdWYJVyMGAoS0hv8AGvxDDh93kSjAw1NUHieLCA2Ac6H8iv64napmdaeI4AOJx
DyRjzUT/MWJijxmfU1kszqQOIwq2ClFHKA031P5T3e6CyGIp8H1wM6IvYIiGu59
w1CXpHbhCxMS/BezX8SFq9mIMdyCu7HUQFakbRpRh0uMkMJ8p7ej72XGbnY0v/Ur
1WrQyRdOUFPympv4tOXFygDc0rjOR/Kwnlh0Kxk3ocm51mDUvWXpTrraSfQONIGOW
R1YUO+VCoD5D/F0MZ5cJpYBHF6EkKysfZ3sc1LkEarkW+iONWsOaJ2Ax77fz23ob
NaG9SYBkHV9e+xsmVTMt41RgtTsF8ptFxmJPJQ5ERDp0Lh//nPtmXYqtIrzIs2qK
2AuPwR8QjqHz+wjeo/xkjBsyHnQiB+nxFH2oQhwp8umEs9KJan3qa68fITChKZ6f
z6IzV9w4qn9EdLam713n04ZizXpN6SKOYQfOsfDyv5uvSPKH/jeskupt3JBLpqLv
aEXzY2DNZApFdvRmbjdl7t2DuyX1zh9bs8tP2IpMaV+6T2cH8AiNVUmoVzCFJSG
NFb0eWzhP+EFiLojHP8QfG7y8QX3YjbpGSfnapiXV3/nPg5xCarZC6ryz1G/c1j0
7HDfOmMxdllf/hSAi+CbRmGAsp8WI7cYH2Q+lwGiSwOsOYU22t1ivqdRm6cNux/Y
BeUDqWofYpDgn/UM1FGxKWvdqCeyrA3j5k1PTFO1AeKY/+QGRnASsnDC8UUP003M
VbiMD7Z0uB52J6tt/mpHcUXnZ2Lk0LrAacFdi5wxbz/LnN5A++QP+rkh6TMg7puM
FgfXQCg43+hYhbrkvwmiBFAJz1B91j1LSL2G0HzszyGcKNL1s9YoBKjb0xx8wIfw
eEfuYuoQstu4Ea788+n7ozmNS7kFQ6hYtPhCmUPhjuUTrWtWV1F89Zf4JiFihrzF
WUFj51aAjou8wzB0k6peInRy5xJ4rpwZiizM9eJruIvDD+HmMwU2UGcyjppXN9E
yi50cJEQQZoP7JB7fw9Emjq/WG1ODxRlezbmOHUfbqbbFVM/KP11iJ75OEQdKw1J
M4iTZWZ24e/aEqoGZ/R87dfG9ZKuu8o7i3QxOvn2cm57ywdG4NQV9Xj74FdVrLoM
U5nTKeimdkYc6BUhNDRWeoTzjFAWbGxBomgWoy2+mne6F4hVX08Kxv2YTG+yDeAn
iGxK0LiUW+F9GkqU1HPqAejMoIH6Z2zTyuTVJvc8ig3gUQLfCO2AJz9c0/pQILgH
npBgpq+4WdW0Yip+9lr3BP5KGU5mGHde1wxmL3A7/p6tMacOwOExhfIKIwUtE8c
1CXT+HUS9zjONA61tTVTPZkEY8KIMr6voINHUCvbd62P4W9ZEbxWuSoucc+XHo+
Bqk5r4vFgR5G3emt7qGsFennb3siQu/aB+jENycjzN7Rn1RCYiZvJAlqy3dLEe9
S/M1IfCWSLijcJMHgMvm4akifig1+wCrNq+S15End4xTAet/Ur7rzh1VSfQHxRM9
OVP5rL3vLgbYnHNOnBWgM8FV4hDBzsfLy4CRvNUvYiJ0eyqv5Wsi ft+4sSj3nwLO
COoNx7+oqX1ICOo7yiClW/DhakIVI5Ydm2TsBchKh9dSg+W/Ez6C2ph2v33x4ZBP
ucurUokYNqz7U0VSSYEtb/1EzBCWAM21PXdmPhtWAObQFtO5/816nDY3+QO+y0Au
81A5nhgzXIEoDwPafSjWJ5YUJf8tnftD/CiSH9KbmwQ1sTbvXAe49Jtdx28Jb+Rp
9E3QBexiFqppkkwAPi75CIb8yPVjauqBO8kJQcA0yookzBya/ouZC5uClVmACNrYl
8BA35zxa+/f4kmfufvE0abmUCTGxwVwJOan6uvaVqVMIN5Hj1j2TzcNmdqZyCwMW
JIAA1dAI5bTEYkUdctqD3CrV0eqQL+b/Lv10JZD58R+1iaNsQaUGpdsycW2aBFUC
XHiesdGLYCS/J2biGSDIrYYKho9ANkNRtMOXRAUR/dUaVikL8jMN2ka89RDyX41N
gdQH9OmUQP1oQcKImACQcB00QL162WLLnEKoP2P3VemkyMGRSditD6QPkfutn01f
6D8LCYRTb4/p91wzIxdov3XvpbaX//koMOWHwAdgDsBPK/MmRwPp8ym+yE+tuz+S
JI3Nv8L5KkshFraFseUpPcx41njBvQV0h7vP/hqwwnbFSJYPm380LK30s4rd1g5G
LNyaBIanTPrc4j78Sknd01I0Kha1JXSKX2U15TMmgOOyuP5wGBUJjAHpYqvTnZ7C
fUihEbg8mBx243NZP/XrH1OXtNzGv64BJdGNx8bmwW3guuo4fXG5aZ0AFzY1HMCi
UfFtEWA1B//GVpj4uxZ5B5nd8zNiQrMGL7B/xYGi1hAhDYN/JLwgnNkFWP2Uo8dU
2MPzCBug1ZLvzqXQWBR12M8JX17iyXKfKie+5921WocB32ZSc1BCrpe9cr1vzWfm
YJyC1GvHkAAy/b4XvRGrS4NmvdLgjjzWNzkDCru5dEc9+oPvf+/rsyP7709Hsde0Q
qAP2IwEF/YHJDIGVwqEiWdWHRbkfasLiqsEyXHZ6BGNFBaywfQCaZ4Y4dVUzryDC
mtz4YgXwsvOHcaY8UvHLU4c3/+FwYM+OXs1C5oYbk7D68KNeXxw1lui7WSBySalf
IGcm3OM2tZfwauLzXHwSRLy5gtIZj/RH3gfVQZ06ys4S1kzIbJgo81K6ysgyDIQ1
iHWzSxPnGUVz0GOJ2rHb1eYyPzPJlqqJkIgaJvDh3Zdnb1HK+GkIJGgXhgQCaYdQ
1hwIJzHOX1R/USDfxyGA396uz7cStejY7D9SN/taXdHUJp+TJi1vm20xMwWvhpkh
uyjbjVJTmyM5890j7dyTSBGBRFdR19y6ekmCdDi7Z6jYyEi9pMvMGUnWO42mHTL
ehLitRFGLX6vVF1HJocqMLvcs/yInAPWnfTtgBbe2028/rfWpkFnVTEkmEob1lpP

```

mhWSue/ldrOM9TL8TYtLF8+zF4+v/E1lvEfBlBiRLZSA8+D+uG3gGMDq20Lg4XOV
9cv4I4x2KSYKiVv4MnwJd9ih9IodTr4sdgeLLEd3CT15/fziP5jb9vFD+2c8NhZy
Qb7/0YPqtPZwgNrp5dB9n2qNm9y/cVhYf1C9pauNnLsdNIXBt5yXRu4kzNv/B56K
FtbDalYVdfLbhSEcW50DqpBFDKPzbtGdpCsOP/+ViQE1mtNNuTJYwQW4eBtIGfiT
37N/PvZyKn+9uoVDJaNG6iTeKj1WB/kNz+zdmuag3yx1kttc1jDpchMFqR1CUKDj
+SPrkp+DqlGC0TpvO+3JiN567WDV9CvjdfttHJ5zpgPe3lC4Muu0VYASuN3UrCXB
eQLee1ty7rk61M/RlgIizC8JAntPx4hfBb6ujZnyrujGRowG/TLsdQNODvj3Fw8r
i/huor6VwkJwC/FQxvjTNWcEL+MUu7cBv+O2Pd/gL70tyQP7eg0QENUcyUsZ5NXl
f/BJLERQWESr108fRTbkWLHN6/nouUZ/0c3AqC/SNHTuMky0Lcy5+33Xh1ktb1rz
6TRBojU19yjd+DnbmpGY5fDKhQeOUV+ydFSRUCu/1X5P8mkU5+kja8KIWP9HTRDu
3QtuUN/MGQ0iok8Hwr/3U9spCp1E5KsxWfxU+M/10KIqKWPcyW1bX8JUZRMMl5s
qSiZlBke7yuXFrZw+ubzDnoCZwNM37F685nJ08Wuk6giK6w1/q7ctKAv+mMmrq8+
2iKIrT/oWIA5iHkEGi56VrvqetNLoWo0HK1f8ZjsBd3Xc8SIYn2eWticKy8ch5n1
LyI11qnjphhUCz0b7wSLOA7d51cZ7yCPgW9uB7bMLHzefIjTGVNVT8ktRm9/4VK
OqQugt+L50OKRvZ6UpHXAZ/Mkd0Y8lcm39nD/h1DfxA/oIoEM9Ze7NQS0sxD+PCG
Pylc9Z61hys8KH1onuv7tyIZ1a2CITXJzP11cIi/cqbrUdBK6XVn1a9exfSxVH2l
XJPUCb3UIvS1750KAXJXVT+Hh+63LCzhUZaVvPR21tiYZI9exGKh3n2H+Mm+H8g
ODkrO6y+WnmhCwGFZLGUKJA8f2qq2HfJcL2RGV6C71ACc5PGQG5zbqUxmEXidQmj
cpykjsFcy7CsBWI/wmH5vX4A1TN17FFE2Gutasn/JICUXE2yoeabr35F3SbFmNLa
A+x4+MPbsq8eR1RK3/X9eGooP0fkQbuQDk1J8B9md0TlyXVn4DTSsSxNBK+HRBM
Q8GBkIVisBV51aFEEqIDYN6rklhEwAEi4U1c4Uv0IN24vMdaeX55wE5o7JjNFNcT
c7qoChUxRP65LsjoTOxM1lE6Ra7302PwaJZK3dsmLIE+7jaqdm3w689tw6sr9Mzc
hTK8nUwfkXWK00iLp1ESVIUG4E14xARjYgQMltYlra/wgFLoJkVBAEVMvVL6hRoL
JOKUTBDqwU9jvu7ZhgaseyOQ48+yY8yPET3CM2XCDIyoGAbc58qIC7vn1meuL4+F
otjxJW1xn2T6WoUtTUi6yCCRHHe+xcxlSvt1wr36M7i7IapqGlUdrRoKZsiPWHDP
liEPq1Y7105hK+pMZg58OmFB1eRkSZ1rZDzRZwCPErT7vGnZX3InSRtNuhjx7uTB
qn7yqv47r/xMPEPVshGj/KQpEu6+PtMzn8OmF1CqN69yPhc40VtNwyQwWHBBZ43j
Gx8v2IHL60HGy0yhdcSz5NdNdsBwhs0Yqn72xxMKYY/Ax/kVO4GP8kwl1f2mmvPq
a93lxxKUnuKRY1Jw1lgPnJomtLm4WjPqSXxgY0D9/vnDgfv/9PXjK4hNnDNvi+Ji
qwwAW7nLMF4uVkirCndrtlhdIDEaq/Wju+gvo+pC1lggRZJyuQhCwm2swB5jTuGh
c7V8X0KEgunWe+QXzMMBddU0MAIoHddnA1d0KqNjIRfnIw0Eb93j9zYK5U3cDjF6
LkmD9of2rbA8mWc7DDSIN1Zg1QQf+wwLzJ7yctHadK3dzNzdMiToQb41KtuKXdxM
sTHmhXcbeC5cPIWzbr5tQA6AtbusfwgUFek+jh1b69cw3Ibm8nCu2okSbJ6DeaX0
7/Q6D/wQCWV1HSQRpzCV1BESRzg823D/VPK1Cnx5qj1FupXyPH1h1j1BEongTww1
7LrfK3UGH4zgvraqlaMgDpOofQ36DvMge8RmholdlMRHqSuIzRhJVYL2z1AWaz6
unVy00hr5F1R+5FCynUNxu9XjofqNp4032Ihd+0IiOqORfObfPhFMLDFQgWCXnO3
W3LZR8epSit76AEYaw+6+FmrDPVmqGab0JgEOLctPNyYpM5XoVLM3/675GyKz/3E
dx0HTSm6BLyrY4h4FMVaI/nCu+MkiZmdZx8jDd4nSHya3NdNOjphJv5nW//WLEPO
6BOTjzVrI6YvHJuqkC3FssUY+VWZRC/+0iY1DYnaBWU=

```

B.3.19. S/MIME Encrypted and Signed Reply Over a Complex Message, Wrapped Message With hcp_minimal

This is a encrypted and signed S/MIME message using PKCS#7 envelopedData around signedData. The payload is a multipart/alternative message with an inline image/png attachment. It uses the Wrapped Message header protection scheme with the hcp_minimal Header Confidentiality Policy.

It has the following structure:

```

application/pkcs7-mime [smime.p7m] 9970 bytes
  (decrypts to)
application/pkcs7-mime [smime.p7m] 6366 bytes
  (unwraps to)
message/rfc822 inline 2082 bytes
  multipart/mixed 1977 bytes
    multipart/alternative 1144 bytes
      text/plain 381 bytes
      text/html 479 bytes
      image/png inline 232 bytes

```

Its contents are:

```

Content-Transfer-Encoding: base64
Content-Type: application/pkcs7-mime; name="smime.p7m";
  smime-type="enveloped-data"
Subject: [...]
Message-ID:
  <smime-enc-signed-complex-wrapped-minimal-reply@lhp.example>
From: Alice <alice@smime.example>
To: Bob <bob@smime.example>
Date: Sat, 20 Feb 2021 12:14:02 -0500
User-Agent: Sample MUA Version 1.0
In-Reply-To:
  <smime-enc-signed-complex-wrapped-minimal@lhp.example>
References:
  <smime-enc-signed-complex-wrapped-minimal@lhp.example>

```

```

MIICvAYJKoZIhvcNAQcDoIIcrTCCHKkCAQAxggMQMIIBhAIBADBsMFUxDTALBgNV
BAoTBElFVEYxETAPBgNVBAsTCExBTvBTIFdHMTEwLwYDQDEyYhTYW1wbGUgTEFN
UFMgU1NBIENlcnRpZmljYXRpb24gQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIEh7j00
Boq0MA0GCSqGSIb3DQEBAQUABIIBADCSchA3WYayfDB0SnAjlLRn3cTsjsbLknQh
iSnryqFniP70VlHS0exfVjnkzy5YxHRCrqluXfV7EB4GRaieVzIkQTUEnhfBB+oM
jXJzEZWi3Q/O3b/5AMsV8vks+gCf3eND5y/dxgFuzgTrYbE+M3XsfkiI4f9MaK9G
96uzaT0E1PLOCwQYUuWtPCffle484roJwg4++H+jWYpGvWhM1fGUu7dbNX779ErA
pAMmOS4cywx9W20uczJ2Vzaa7OAEbgXrSinji47uAMFNvb/g2toeXlM4bITvdjd1
JhBqQoxgIGdVLFmxG9aZzKIWWF9D62cEdnyCu/t7A2knMCPkAqUwggGEAgEAMGww
VTENMASGA1UEChMESUVURjERMA8GA1UECXMITEFNUFMgV0cxMTAvBgNVBAMTKFh
bXBsZSBMQUU1QUYBSU0EgQ2VydG1maWNhdGlvbiBBdXR0b3JpdHkCEzB8R0APhiY6
HGLS64Mv1sDXhpQwDQYJKoZIhvcNAQEBBQAEggEAedVmzC4JhhBlJEdXJ0NgpV+6
StlCprlktO3ukPWbOBwWuUoMKcyt8aCN+XHtdVUFNqSAqJPhtcTGjq9JC4qUn8TY
tiH5BLyu5JDU1d1f9FvnFgbHpdbiifKF3d2F8YRYEa6IrU//1qJdWisn1ZBoYBKe
t07G6mMg/L/4cGfyMGGbWxIBLZDImhxr/JvPS93jGRJChTPDnNbYKtL4t0rMquM8e
pE4ya8MpWXZwXZh3qGz4pcBrGPY7oEkho9pzOMmoHU/sD3CpmXyGewWHTOqZBqHQ
wwZkg8DeJseAai2r5yUx1f1M4cZ9LTcgUQNFOu+vOLjEL5eiG1zgmNhx8axcvDCC
GY4GCSqGSIb3DQEHATAcBg1ghkgBZQMEAIIEEDdlzb1BMvVv1ZjclFL6tSSAgHlg

```

XtLH7SSnyPyftXAJx/P8qkLeTa7PvaM2TOhUL9fQsc6oNeqG4tLomIbn32XajG2N
kxPAX9J8ywwHaFjs+OIXgBGKdyYDmdjRAnfi91lo52ywxkldXkiYJV5mfpKkIAp5
NBWYkfgYLSf3QeD0+6FGdLBXwBrNdv9yn4zn90qwkGRRzG8MQ0lMXVGJnqzqClZ9
CkHSPpeKS5vUH0WGlJOTRoEjadmLXG9dJyYHdFm2v/Y65mvOhmCnIfzieGydzTxS
kBXRFttXESPoXoJR6jX271KbVu1O5hAr3xxrO8qtrFUZ6ug9VvqMfwpTupsYiVV/
NSqB1KBHxpocubCF+84BUB1N6nElqZFdc23gVaDcqdD59QETfiCj6cvkuI2vrZYm
6RSK3s4mt50glry85o7TqAKqZxuGqRgEHMx1mBlXQcVzoiXai4NR0AwKTVrm0J5h
tSqs6zbd8asTjJo9+CcykzxVcx2900z0mRDb1z1dAFLTFAQsvmY/TfKj0kiA19MC
Tc6To0SUagvUV4G/2Sghh0zdDihTK55dSG9jygVOjVsnRtg36A4bYef71A4jNUbk
dz8AsP6JVgHwysQs0n8JVkNMEs1i1hr0Z/05BE3PqMhHKSz1Kh55zb6MoBJquyqr
7RF3w1mXb+tsM6yUR9WimlOgIon/GTxR0LHKkYeLX2wDhQCr2PbcIbAGIP+W20Bb
qjvjPxcjfoAQZfxsTYGUr/mTcaLZYvVN9L3wg2u60pqz+67KszKn3U3Efib/+6aT
8QV1weqEfjZRYc8o+S3g6JES/zPhCxykoo5Dmx1trvTOhK4+0V7U5SrUbC+S/oiE
hQZd+pk3AVS6x0oJtAQWoBWKDZvPSQ3VXPVN1wCnt3muq/xV540MqKtZ2XRkuCwy
Cez3s4AyhdX1Ko/8lsJ8isSWsqG51iTODRGjooFMFMz4NXmNZXE5ink2Ba6w80XS
CPmVQ3SKnuKaUGmknH5mAQfUrZCzrSqwppTuWNZ9xTTrz//kSiBSv7aAz5GVcm+5
fzLWdMoMoWwn8lnhEU0jX8vmZ17I5onAO1UjLNBtyomiZqz7PD/iIvEPm1XqgZpe
mJBYmIZI9cSB1lQF1I4R5GBhBTBT3fyPAT6iNkVmgWgL4KGOK32tz9XAJ+UJP4z/
pAqPyMKXcKsIq19YLJzFgi8ACRtuAC5tQI61vkd3N96oHZxJ2T7QQ+n3skFXXcud
v3buX1+N8M40VBFybsx3dAxUPuAtkb/tiYcXHL+9ivc+ncXZWm1j53tPMiodDEvR
zBb2s02cCX0+k3mdrCgLMvwlxN/874+CJ8C98XDijhc1LL+/vfe6VfH9iJhoW/Hu
5SLt+bTcr1bcmbgBkFwU075vp/9YIwFW1MQDa99PiwVYz7MDkWFZRGs8RUXaHh
BoCi908wzibGTjSkFvu42xccfqpOjYrFLktMp31G9JMyZvaIHxcYldJLIRB7m4fp
Mx5/mxfxmbjwJzIG3Y0uTOupzJtzb94GogLA4VyG3D4EzQyUWLMsvwSrJXSWuMHY
ZTgb7qqjb7DVk9YVpmmqAb2JhVJbPRs1Uc9yOSSr23KqrJ9Z+z8/8BJWEc//kMZ2
91NnMsZ9X9rPAuuQOjy1lvqBR3Gy6Wm+dRncqyPp1/NWnYJTKrbdM61VzuM1NFtxKn
d6eZqj94N0BQxvTn++F229TILgz404kJsVKGZc4af6aE18v4VMe95pYdiD8v1e5E
mLbAo/5jMu6En1Vz//U+0AiCa1oZ1BUNXweK6PNHSAUKF7RY1d3eMBqVdaIiYJf
dKkMtK2bGUsrTMbWhA210p8q0Iz7LNpCjXBYzDevLdWzn4vZq4obaJa3x/YOe8jT
iqxC4CZ6drM4eHKRjgHjTqnXKXZF3/+w3JpdNnUYe19nQTCid3DLQKYtrmuoSJ
SsJKFqkEC/lNt4Hopo4MQG1KDgaHBps06IVvAm9TFzd4IErtQSjgRYZkp4i+SfU
TLYzVQP/Up96jphXuQGmv9veks70NiFATyGzsUB/iCW7ys0NpcHN7vrXv4+3KyDZ
1d82+dqbYTk7HiFgAt9UtKoNRazBLF2E5xcax8TjEHsLNAN9nX4hwIjgyJMggXcR
9H+v3WHYsQyCHHGxE15b9/PLWuBDiYAKOQHCjHqujotRSpWqmiFP7D/QUUCGvsEg
RhtyC8KjJNtFmd/4PVGbct1mTEyMuk1Phxg/+uj+iEr8cHmdKvL6aT0VQ4dp11E2
jCzdqos0uW0ssSBjHfWDG9Y3NtZz+AeToty53LZGT1gYWHIWSjIMHoQSFcp/9UEp
rzoT8YxaNUjXs3IKzvfWdDtdVm2hmukVZaRcp0qMNA1ZNbnznq0u7Pw8jeBSc36g
HhsPJUgWlz1W5xr/BpduJwrmxFz100MXZKV9FFJXVXM0UMJ2P2FrZRqnaPAifmS6
12GBGfBcgTv0b3cAussB+lEjaeZuqQMwThMuKotawY5UKqhvSKJIp07NOBS5kQQ6
8SoXh5ekYcesMwVTzx4btEEJw0VgKfli4S+eFAtOy9tcBv4A9aIzdYZ8blIMAg4p
5+uVv++0G7OuS2FB1x7ANX1gsXG2HJ+IbwfUTmpn7XsI8e/dNe9P+F2h28390So2
Yp8YdobE4Kw/G2E9VQ9mvRK3BLs4nTcyYouHzcz41GoVQkB7MQrW4iiVY14BBCSW
TLVklqRXNqsw5yN67YRgmmFPgnIvsOzswlG0wzBul5sHXTFcn7p1jEkI2udfdoWe
j1Z/RgMOvLOV5H1Han0Kxsqx8IPcw7szf4Ccd1JGKHEVPkD0Vviga9oD6dC/vftu
nxM718ResIQuLiXtpgL6p1ZUxbGhsShurTdHb9fuPDaSV4SBeYXV+mWyNSA1ydv4
6sjVDSUCPiTm/fmq6XFHQwi8DC839RkrEls/YpHKJ2xuhSr+FvqMkh8UVuFyxMiv
OXnbBkHLAd9jnRjy6TgSU8EafYg7pAmqBcYvmrz1WYATaIoda84xeCctSpT34Du
+z6lsLddbwkgK/SyFKLjrSPNmoeJjHtRe+LL0yO9ZbbR6YunaGnKIEWS1CSBVVNR

fV8Fb6XAMHPbH/xVSBqy+slqT4IbcLlBAytHkgNzCYgUKx8RQLFUbIEcrN0TMbBt
OKQUvQHoJXY8pbwHRgPCgUyNQwLEyDVuJBnWt4bUOg6hcsLfm0tFhvXSMgWF4phh
ZOWp9rq/8LRik9zw59PF6rIxFuZ3WtXfRcLMGDq/hLJ/VNN7eCqQyTO/DIzPM2cO
EBOP04JVcYqcTy9FuaiJNE45+1BotjA1HdDwTPWyWFv8foLSwwgG8c1N7Z6LiOT3
PoBIBvNhT2McZ/GhOMXMr12T7I3zHtpvh3XecfmOlGsNiJZU3yxcygrkIshDbNTn
fAYo1U1TdgtTvqy/XuWFOdK+/t7sT4owjrLHYXeeZgTszYqL0gZdTJh7vhZ7nnN
p70NLf0bi+eIUbCHhy6rTawucjnxOoc0SI7q64GU2dHS1NYb/7wNSAm2DjcobNrq
iIVyWNd9g6a8JQOMBFBQtrbGGwFDfcCiFW63V+2KGWS+Wghq5mB+aXWocUoRgv02
yrQNuRUwSzbjTDgNsSE+5aCLCVIppoKwGU+LY5oNXfE7NTERx7zKfgv1G2y9df5p
2rSimOUEgLFJO7r74BzcTysTopqF1PsRYZgxiwGHFgyKqP8Y1vZLeMkdDYwUtu9v
RlfrU/e137/r1v1EKPfGoKPVtwsM0S6Ur95AZSr90/chy1xSp2wLaoaTY20crx3H
G7DDYH+Ldx6fWw1T6oJzNYq3Y5Oxfir22F3QVwsXs1WhrQhJhs5qqG1ucSj8kh4V
nCw8kkdrvAs7H5wuTmxR9hatO73vnpBNWooAmWO2kLLuVs8y/2KLPPjppRP/u8o14
o7rHFsestbznvbcPaoAtKFsEyUbmJewR3ApTuR2pubpZ95cdJgtc1ZHYyP183W+5
zbyNqwmmdQXvTocOqYh1TpxS0UjXwHgSCxjKbq1kiR9YqCiMR8Z1TmOoCm0LV09
xm/sArkY7g5NDZdLlpjTx1BtswE9KDuTHWv+58CDPeXyDz/429g9A6TzJ9mqS+S
WtmU9uh1pxs7yqlYAWSDUPY+uCAO2DFett6tO6RqUbmXEYail3Z4wc8qdzgkf5+R
TGlnB91DIN81rUmhGGGpWuhHKQ7YC+n3hiJdWNyUbbhSYQGxzgzOz/p5eO8GAvZP
wbf76FgkjFBKykXx151PQn6WJHR1S1KjJgJepf2jf5zBt35mSW82ngOrv0R1Ey5o
WafAkxvdx30uoax5IAdb7/g15P06FvsFQ71GFTY2/skYJMIltZa32M/4cqPkDIa0
urUs7R42M/46LASI4+e08inNtun0we42AW1qYDwuFPfxE3ZIZVzkc7+26Lj0oGGK
QrViXeJf2czzJowh9FQDHMZ9DQZOrksGev147v+vfnRULMwKCGEUEbuunDFJiO2J
cL2wtqr4H67YJ51wkn950LoCqHu34eXz24jmjVYVDBMJS9wH/jIfk+7Yn8L1bab
Fv7AMxx+iOePwF2ZNTIXLvwRqgtN7FpbOtCkoQZkdILEjLS22bCoZGt6eCwOv+I
KoP9n16KjEutLUmU133RWyW8BcyImgDUzaVo8CsjarznJLFizNYS/1wWSIzj0Eyb
zn1Z+f7AAqwm+hE9610hALUHReVYQfYxwAMxN1Ik1cmuFsuG2gFgRr0CaYcQYff
RyphlYlK1xMyuEM9b4UCZ1CIffXwDnjx1lukJjVXuujVWE01DVsA4pSzIupCiq9Y6
pA2ywyad2nS8vLvrydpG3elvkXaSp8wTJzJgOxJ7McPtZWNnaRt7Vb203hn318R7
Qe5NiZwmBFtSdSCIEvmZ412ewKSc6wDaJJyvK6C6/0ism2cyU6n2bmESxt0oD4L
gYSfQa6yhoks900Q+vmALnw3occupHg4CkI6+9Y3eLsx8U2NqAYVnWSuFxQ/E/tuc
7fAcsJouG/u38MSE4eB+4Yrfn4Nh7trE87HrwtOZPn8fWosPY1g0Qn1k6vg4IxY7
d0iFtCJmjnsxa339p128C1Em0i0Z0wcwHJKrMh9Axxk/pQtYP2V1/ggMh+1BE+It
PV/Et9pjlzspBNvKou2C0t1jQMo5TsEGX/fg1IVYNcobDx1s1tWWu7xWkREUOQmu
Oz4jWzU61G40o8VoeYquV68onBYIWCxsiZMTdwpPxxK3rtc+LIIdFByDac7QPWJ1F
NXNsr/9pF3viYyD3wcmKng4X5gtC/adM3OJUkaacrH4nPEjtCSPKLceeJFV1Zchq
yeZsZJBE6X2CvICUIIRGrSiaFHOPvzw1k8jgN/2FDKNvFhVxtB0KNasckImhZGUY
TF2oWkq1IVQQeZzCZezX5yqaq9G7Rmiegl8k6/CJaQs1VJ2t+gc17Wb3JR3uatx
ukfZE//8iOWuFvJXDikgkLryJev4v6e39nmF/myEKjmm1YLG3WnE21rCKkwkZVq6
HJqJ7sflZ+zzeBPQTFsaccuOPxF8wpXFzNNTNA/a2W+gjbSXZQTJkIVUjtaE9wNz
/TnybvsgGsTi0tGMAJXfCJL+yTp/EnRDM0F9Gdt98p1c5Jay9N6tNyMitxCo2QQZ
vjp0SBc8y0QUef09TUDnwQCb9M/aw3J2f4HtzLjdyPFRKAVLBvwEp2J+IZ9hpBPS
03yftnWko5nBKAZK2NV1sWjij/A6Pgev4XOih8HRSJvZzVDuDLioeStB52XguORr7
qVbMYuG+BcSwPLrVH3wLoRq2UV7cXvB6WomYNh3/Iv1DLdrVhhVd++P8jznG22W
F817vfuSogzzM9PYWylDFYEh9XiKiJAWpDr4QKx/K41rRiC/+UNNhnLr4QK/Zb
Y6JIYh6H8Zwq8es8aaWKYE9PhL06gMGZyf7gw0jHZLo/5GyI+tAhPMIDdeT/aHRs
klisHJ67A8WsNrHWXft7jNTKnVm310XIR9tAv5TTWYXntA0Z087f0YEWrywYKNHf
w5icQ1Up0wWRjNATgW4887N2yKqPdLNgxHaMdYxn0eKKvbhkJNadA4A+vGKcnVYm
I5ZdUp+AHaVARdf4oH8xmrP8o9Ty9e7boxLZGzBHTif3UzuAvD5B5rZNNJVSZ56W

VXoz6LxuD/NMGE1VUptqxCr6miaxeHC4Lc2CV+5FxFGkTy0Jpi6098crFfngKr7Tv
WEgWHbsjb3JdKT3rarCoxxxC0ccqI5h1JJ55UvCn1rZAVxBla4z4eG/UlIfmy3iR
76kNbWNqrG1T5rr7OwtxqhxBSrTRXC0AW6j1HM9YkvVcqoKgs/Mj7+hDVUdq1BtW
/DEyeBgAeQTekmMj1N0eclRqMmP2VhPjgKvXdsHvi5HRVI/sLELkBXfnZIIyiVR8
4EM9sJyJwRb7zrK/ZSjR8eY/xYi36iS24GKufZKkIq+q7+P41SBN9xGp998DzT6g
/31E9y/7n1Zb8sQZtN6wa2KV0Aov1t9YWPkQ1xdouaTp4MCJwPpPbD/vXrgXwcdM
fX05EGSuyqyyU7CDEG88P/xyBikToGRygfNKjagD4Yw9PW/1KswtjaFFIqtIPh3B
IeILYznxYvIp+FKAyFESPJioM9cI2/ge5u/SyVnk3Pz5rfkZmX5EjNdbDUGmQBCA
XjYgyEEjAfVptQ/rqdnkedOXnkdmSk7I6xX3QkM9jnhcNgzGWXsFwa9smwXubWfL
eW89gdes3PFxps5gn+VZNR3POJnvcd74q8cVuGDvCat8B1AdEeTDsX7BtEvMd/9+
Efqj+pQIhbvU6NIy6+9suDFhzJIpncPMZ1oLAC+bdqjmRM5eg/7okLM1FXDZtqzs
zAX94ijUP+6IwzISHWlmoMclvZ4zA6Z3HfFtdY+uAA5rRutmqtejv5FsKWENPq4D
fW2IF1joOBkq7AAKds13kFR3UmG5Cw016+EQ1mUPYIpaZyD9SPx1XZ7djtRE8OV
aOdiLENe1pwX3WGUy6rQo57iKwa+7hMw/rkdFxC5Op6wyGyazUKSk0QD1S/7MR9q
R3kKRrdSaVq7X6pRXsJJQsGQd2zjFXBM09i+C1RgjBWWP+8eSdhBk0xEoVyiHuM3
3ieTp/UpStDzz1UJZVYrd2rtF/VHVA+M17mrIkgA2eofShiezCw3JhIV2GN0cnI4
kfNo4wKuH0lsJVu+CYWw1gEaoH+nB2//H9R+fxZ/Luh/fakxB1KfIFe96YIVfc9m
AW6XsEVfnQhTWuCU4evFM7m0pjZS3MH4eL+usAB47MpfZqCPFJTMA12KxxOaAuk
AaHg7cmiCtpQGmdly+YUNBac4d47szdRhvDnGRyJpGwCYiJRQp3DZKvtwoodJKW3
YW3K/MTdnDRPwuGhVKb2AwcPwSC7kw0azRovVYOnTH0tRkOkNBWhXhHYORaxSZR
0hUAe6X28GGPaObt3cvduQJDW/eEbG6z5x1bvCvim9qhj8ahoWm2eLoSgmeJLa9X
p6L5JFGTLVaC9L0nIJraVcr70RGEN7DhubGufRQe2AViaJ7DNRUuNmbIOVumP1Q3
gnta0wo8bhUTEpZDLRJQWfbZxtwi2hXgzEqMNZS52yFAexyRvqZN+0rVVAAKbV6/
aJ/nwEK3jIapSCvgUOu+BzHGp6Xq3xcCca9gRWWYbxuXWmgPgcRNiFU1Mg/HrEiy
y8YwqWf4FwzmZ9sJGQVhkJSzIbxa89JGX7QpjhPov70b4wD/JU8vBgXDHDHkR1vK
aEqRdFbcv2G5i4hT17y5pxXYvJaLM7BnGDBvo3bbQhHYtBq1C2VkJHexvUpmLRVR2
nTcexY00MssxeYPvSaLQCWO2NZ+0LwFKx/0wXA3zcUsQnRQmghHsJessCCsBXEng
wpJcU2qCG1G1Nkz3dnAeTyzNI/h8hpauW07yZA4tas517z6j5vSSwMD4m1XHKBPc
MHQxE+GaHiMZe2FxtA5GQgkggstNxn3W8UcCNqSDkPKUNmHzPK1KL7MvM6g5Hidz
HGKj7NY/LzSQftcu5h3li2Yfa1ImpctUVZVhOf8T/halWo9Gp3F9+6TUvhvP9dVP
T33eCEPxBkz3RwUZSEmZXRUJbh3SSiFtFwn5RA9p8XZai/wurfOZsp55ma7r0M3C
2fomu+tcQ4BZJzMRWvzHd084jIry6gHcWK7Ppe1EgWDFSIcU/istOXimAxAUBY
k8RXxpbTVu7csDQBFsKEbdqsCy9QKwjOGobJYThkAvTfVFDutEiT6V1TN9kVIpQE
L8qjyRLqEAnBssW7z4JE+qINP/BDblTM71K1lSH86e4U8I/DzEA5OAx8uJAteVmN
Fqz/blzI5ggbe6R0pFtRD4sGpn0azCNyM1ks47czVaSjI3cEN+yU7GBXfAWriRcb
2sQs7tzqmILnTXfytItquZTBPvsOIM4TGIzzQ3yLdIp0lnzzBZ7zWeYNZFbesTw
/r/tl+q/aU4an5q3sgw1mN7ZEjP2+bc62mRJ/cC90mVJvXpPFG/wuzWdOBi680Na
DUGS2zNJPDlnLwQZKaN8HcB6FiXhMnrVIF4bgPoPoiRoAiU/psIaa26CKdso51Jp
y9DdzQLMM/7PZT1w9uRk61WBHGnUJXqGJMoZpJVblhFAKUzWUa6MREZMBqNBYszH
e/YMYKXibjeYXgEA8ln+Pw==

B.3.20. S/MIME Encrypted and Signed Reply Over a Complex Message, Injected Headers With hcp_minimal

This is a encrypted and signed S/MIME message using PKCS#7 envelopedData around signedData. The payload is a multipart/alternative message with an inline image/png attachment. It uses the Injected Headers header protection scheme with the hcp_minimal Header Confidentiality Policy.

It has the following structure:

```
application/pkcs7-mime [smime.p7m] 9925 bytes
  (decrypts to)
  application/pkcs7-mime [smime.p7m] 6342 bytes
    (unwraps to)
    multipart/mixed 2009 bytes
      multipart/alternative 1148 bytes
        text/plain 393 bytes
        text/html 488 bytes
        image/png inline 236 bytes
```

Its contents are:

```
Content-Transfer-Encoding: base64
Content-Type: application/pkcs7-mime; name="smime.p7m";
  smime-type="enveloped-data"
Subject: [...]
Message-ID:
  <smime-enc-signed-complex-injected-minimal-reply@lhp.example>
From: Alice <alice@smime.example>
To: Bob <bob@smime.example>
Date: Sat, 20 Feb 2021 12:15:02 -0500
User-Agent: Sample MUA Version 1.0
In-Reply-To:
  <smime-enc-signed-complex-injected-minimal@lhp.example>
References:
  <smime-enc-signed-complex-injected-minimal@lhp.example>
```

```
MIICnAYJKoZIhvcNAQcDoIIcjTCCHIkCAQAxggMQMIIBhAIBADBsmFUxDTALBgNV
BAoTBELFVEYxETAPBgNVBAsTCExBTVBTIFdHMTEwLWYDVQQDEyhTYW1wbGUgTEFN
UFMgU1NBIENlcnRpZmljYXRpb24gQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIeh7j00
Boq0MAOGCSqGSIB3DQEBAQUABIIBADE6mU323yt3WjthHoiqYZZ7xRs0RSluUkr5
I1v10lSNq5YQ95dD5vNuhMvjt/EtfgCJ7AO3aJNaldxCo/jIwbq5I6odTQZ7aEWN
BkZ1KMhtu+gDoczq+jPyGvpYX14x4yUtSwbpOI8nm2VMoYvNY9nBaqaXuraOLnGE
VeqcJ0lh+hkyb0rcx9cxLk92xMk71/HQK4lYD2uMSnec26UemFmvSbijnBoJqqhi
wDG/iUN6/7y05UYnku7+66Ub9Jj4pdtjMXAyF7LvVBncQ4L+aXMFJQQJTJK6Rfh7
bgogVv/ijZtSRmB6jKJZ0wHruSgKIGFi3GdUhFxf7URV+Xc6/QUwggGEAgEAMGww
VTENMASGA1UEChMESUVURjERMA8GA1UECXMITEFNUFMgV0cxMTAvBgNVBAMTKFNh
```

bXBsZSBMQU1QUyBSU0EgQ2VydG1maWNhdG1vbiBBdXRob3JpdHkCEzB8R0APhiY6
 HGLS64Mv1sDXhpQwDQYJKoZlIhvcNAQEBBQAEggEAc8xsqG6RvJzmyeWC4l7tECW7
 cLRsPjr0ZP+Nz9j5BjhH00A8KUh8deF6zC99DixeMVHxTWgYETK/yAKR6VASWglZ
 jT/PXeV3uwjFKFj+VfMzJ7OZmToXAJN5d1PzYeWeLjN7qRxo0/DvyjmvNcfwXI1/
 uwiTkdmokX8dyMk93E5Y1wwQ3fKQMiRIt4gngU8r4+qMZzpy7oPWQ72EukdIySkv
 wga+KasO7PeTLj8KS/dQ7DxQ7BLMjVF+1zbQ1vTujOPQmQ13u7+sNe7YbsIpMEMN
 R9CHHVfm12QdRm7KQhKJ0TMC2YeW/alCrLGNJ9eK9Qz1BfcUtJn9hWVsvij9fTCC
 GW4GCSqGS1b3DQEHATAAdBg1ghkgBZQMEAAIEEBQNNirHKPkL4TpSNOFBt7iAgh1A
 yX9nf2uy06ybPSZFJaD/uxbWjJNQfItZY7VyFAQBImHBg6MOT21WdzkvQA2njMSF
 xQx2zKxBaPtnRUKQCYyHaEygqrCT/cUcJx6nVSoEntZQcTTrgSiDL6TxHgSyp8O9
 d+VfULsu82GGbdylE6wesW6wZxJUCuD1DjfonJZUf/Zl2LviF0o12csXjaYpbTc
 13GtrcWDVdg+uqb57moD6ylinulseA1viro9dBUT9mki6073hZAO99/kbDfgSdTU
 KJ9qIZ19sjiFNaODN5noumSWzUiUSjIT03RE/iATAyjnrhW9Mwzbe4PtXzfHJuJE
 m5hgiQHyk1h5wv1Qi2vJ16joL8nSmU1871i92+X8S6dFBhpo918+mFsvWPCO+ZaS
 4TPoqFfY27sAjL+s0h0mHE+AzkQ2aSK23uowh1vTyFxy4VANikyVIJWNW3ULA5Uv
 iNirhafgxPwS5p4xiymX2ymqM/t7V9//sePvuVDzQo1EzO260i0lsFqTd0tun4Aj
 P3j0FrvkXv9BDEbq/alL1qkH7+9CyQtoRb0/hjCe+ZC1WU9T6b4Z7bvsOsibnwPB
 koEXLPuPMzxQMe1Q4P5jOdcTukoDJMX5nVjhHbQwZ6P+SSaKRntO+uJcGnUCeyG/
 MM3PHMzQOP3QcZUgppZxG1wxNQHmD1G8OnLS+VNWU5HQLaKg9xkgZi/ru7a0uPRLq
 GWIEYurZRSBOFjdqi/dAwsYSamekybCdBYPMdHNK3MnI2a1Bh6YQ2ML2KHDfLXm
 9sHgMA/0CTP1AbVgp49G4QhjfPK5XMEKqTqoBXILeGxwMABWV/QntYrdc4j0Shx9
 wHz/47YxRSALjvS2ZBATEavEzKIVSm0Mhvjv8mSPjkDoth/UvMIeiIKavyqpZPJc
 b6NVrnkYhiINruDUheOU/N4pPr6yF7Q+DdoJfmgRmIry4G8vj5j/36GDqs21h1lK
 rtNsC6A8hqBK1XvLobN+WsmVj01H9xjHbJ/TtAlb6DGP4n51sPx3vHU8pSc6vR6
 Bz9OT7wm8UYvRdyRUWb16jQW0FhO2jgsnKEXMuu+5JUR1J2e1s32hfPjBrr7B4px
 MYnh3O7OXTjyx/ES8tsOdukPqbFfBLLYbdvTNVjyvkJA1aek4+3o/XeC6Iuzuoook
 EECWH+JSArJpgH7n75dnpmQTGRu/ZnhqhChrXUnIsKIIXpThI0WouZXCruFzx+2Z
 NtAjJhfrJCxK0+miSLeZv2bsxN8Fb5YKNNYpzTqH+6nFHqzbZg1spkQMvDFuo3jP7
 LYrcEOI/WwbcQE/xIC3QgtimfkPodf741+2ZsNarHX2SB9ys3DoQZ1e0ryX7HeL4
 WImseW9kY/89f4HbYA3Q1MoASes8pZcoxcGaQM0lDXVYwRszcpgD8OxMA5BY7z7T
 r38uATXNDwecqCb649/MYQMCvRX1OyUQvApPVY1hf9N7LkLawCJPLPWEuHPES3z
 cp9K+zVWmne3o7uZ/Rxo/YwQoLt51BT991Ytjz8b2AWRAuzfLu8C+supge3WDBlk
 SdYxzACoUonQRa0QAPx409P7s/HKprk1JpzmciaNVV7qL0YVmlS3Rpp9wI0HidgC
 CSKcHq33Qq23do2+mKU1eZ6QQIm8ZLwgGuAnqSz1wo/SGSGF7FuCURzVjSabITsa
 vY1b2Sey0OodZyFyjoc8suyDbv3qUDRusFck1yAbAJithEuzwh9s1gVhP+QCLOGp
 ga4rgZgb/mVIUqfBuqzv20+IKfeE7Aj0ETpokFjW43Vaf688NhdPqtYVY1e7aHpm
 VZWx5dRr1Cp/sV/82MuTgpI6fdxi6o00oITeOB/xOYVaYROSPxG2g8d+gxI5fMUP
 isKIGVPdGFH/oyJ330J+r08eH5bdwQ4ZLJx8VNNZ5DQeJ1deeG3g/KLDKDXaU13c
 wmIRLwZ9ORAsirq3GQuVqgV0h5WRpxE9trhtLBUuuNyxCl1McvwZPQUfWqNLImQ3
 z/5kNguw/qmuzVcd0Zu66X/PiOyhIjINv1brGtGQm5PV1Zc80XAtxz/UjwZaf6yv
 +tukIzP8XHo1NSYi0I8qyro/DY3CUSdZzm0e0AbTSbX6GwDLvo4jqg2ZjJMI/aqa
 w+1bBrVSVvS8LSUGviRYCIjQTq4q7rGBS5DDcd09YGjdLn8swV+kZQ+Q6HORy3FI
 CNq/9f9GLn8On1bKFLDmRR3eA0dCP/FcMa+20/tfhweawpFw4RQEVt5qWxSTwRu
 11BghRZ1VMYvz/c8Jtk1bTavZcF96j1liuqRnU3svEV60fiITkvMemb5kReBnH1m0
 F2rgLSsgdPzLZX7jNnvZloj1ciEOVfqZU6ieS+yEfEDG5DKEZZ9bMUyVUUYM/PbI
 uVT14NuNHc0VknZ5D68iICSXZFEugGH12xb812GRpU73qve+Vwe1CapVxrXCnOP0
 MEYCu/ENIBTy/LTrfoE+kJPhWj56LZq2eJ7wTHwd/fx1Rjsvth/1qMLpWBbWwPd
 IKcskR1SLU3VgYOe4PhlgaQpd4IjGFFfBbgypjmBUA7DlaQ1BzwbHCpetKTWDcc

3CeCEn7AuFzFILL5PdFRbWzZr+Yoqlz+Z15cznBEWYU37fwnNWIpUrFPbRp7j3fh5e
j44Zz4yHkCB4iIvszmOO+PGIVvQW2PIZB9JPsyQ7mzONb9S3qx9VHs3+UCmTD91
IpJWdQBCcosDWgIGSs1+Xi8ty4rp/Q9ec5v4u3fOxqUX2s65N5eRwup2pfNwexyc
H6qsqFVKP9Y/bth6PdrO8qYDxYAP5iwKvQqh7/5vaHdJG0dnmlzJuiajmyTXKjMu
hOEssR1ZdT2d/ivnZnSQyCKKxSIUIIyWb+UrDoIe+GSUWtPlaoVG8mTc3NfxBa2
wPyJ6lpIEkNQABO8OJPfj1QXvqBnr6fln365yIKoG9G8va0MDgjGFYHk8ZFfx08t
MgpPS+W5j1L56+i1qb8V3dixzZxTD8prgd/xBU+fn5559La9faudpl+U9TwJZJBm
Q/WH4V8Q12sd0qATT6XUccYu0CX524eGUh7bHELejrjO5EW2W9N9hBiNy1InsPbP
UsCBCUuJCF+VEe4oN1UuMqqbDokVgd4d2AcPuhjfYpg7BJSwdxATBtkJHXYTpvMg
7Xh1Pj9YZCio+mU9wmHwD8Fv3S/V01tBYrboQtFu9Z+q4hJ0sY+ZE5qtmOpb071M
TFq26vAwGAOFFtx3xvf9feM8yLL09PigGmKg15Rw1VovasdEPaJMy74UwhnMMAQ1
P8XQldV0YUIaGT1uvoMGs98gXJogJ+1WObrglKhFVsP10wGyPEHfhnZ4HX+4IMvd
wio1U1tWUtDMY2rsgsz6Hp6Gc7+Dke3OzvVaL25RCwyHX1D9NmlohY/8dStcr3/9
DtvBTywp9GZofsmErjJuig4UCUZe311sLXzqxuaWLYd9qOUJaXugCNtmktc8Xsa
dXMfxAZc2igIMDDT0pVCufCm7DoA5zsnocDXWXTTf4Dza9Dk/EqyK4brFecXq/sE
Fq5csMwmyHysJAjEswhBxPKz2oIvVhRSOLpPIdlvrg17i4UasneOxmptLRWMLC0K
D6x6o9R74e5QydItuawdeQ7VaHcPdOcmJfKqW3RgOo4qyPUxUnnYYMmMTcH9p8/d
FKJhhr11ECw6hp7g8IwFsYV04pqv0lgAN5wfwu1C/VRf2n9zA1m+1CfRCldfvbtK
W7N0qD61af13Mt5HdcuTCUNNg7chnDPAMQ5PIL/x1s1ZtigWaAigxIGmVn9eAW2e
YLv/ckPed6ovZ1EnqW5qb3b+JBF80hVLjekgzYI5OE1kAiEs8FDhH9UIOGN2rv3o
V7gn1Aux9h8mBJKvr4KvWu2fouP1cQXJ1X77Thdj3asxyd4q7UrnAhzLNWGXYS+h
0jwLb496fameKx7qovgnCEPp2TUbjunP9kk/azloVgunNe+W6c741w3X7a0a73oq
LTdPZ/fNkWdpj6tEw2ufJ5Ez5TZ1RtNCdh9H+uk+tbiki61qmCSjwZ6wgZF0P9QA
Vko6aMCl+8oXiSthP8R5YRq8YTr+Tkft3WmUGRY9ssBweUJWJZCt4nMwMzZ3M1E
YOOhhZnreEVxo3Hn1RAF3VUGHrkrR2k47jF61FI6GMvMZBqYW7vGeSOjZW/gfOoZ
QGn2AFBLAMH8oCJoVBT6N2MMYIQxKBQrk2nQ09a4DbZeLn3IBJgiTYsv6w/Wrr0H
qTGF1N1080HaCcBGqRE71x8OwKk1tP0kYcQBITV6Ha+c0wT4xV4FY6SB+Q7wRh0z
5t2FuqHaoIjvLnGPip/93GENpFiS3qDoROFiN3xDkOM60CENAd1Bh+h4aJdM7eTa
b8wqsBqU9X2j1LOJYepG81MadrDvMvYnEPqJ7zPY0MZyfl5pRKA78+DHdeYuCikJ
ELq81gJjBoHOI4ZLTH2smh3cBDcI5dqV0ZTo387037NnOKT3KEfimaP6cQbEWDmN
L48gAVsGndEOQiea2j5nas7VszyAH4X1CZ/AYgQP80IzKZp888D4tMTw/lx8be5
EMU96NzWvJciyw5aQ33c1qJrF5UB7JJINYhQ3b3iVrPWScv3GqHYrgZrNO4Mvbq4
jS9wFUMGc1oAbd5p5RnJ5ewZx0JDncuhAG9GejZbJ87Dgd2IP0dqn9DtHVjuVb0E
XzuNNxhuBpKk+dwTDRQ6vNdc10fQg1lyZiizwzhsR9bqHtpbWQD7+8MGS6Kh1Yg7
r4uc+MNjuJvc9pMLAilzq1eJkb9JZcWa3v1Yx1f+8AmF80ZaDgiLKKxEb1oQ1hIt
WYd5b8S75fGrQugw0up4268p/X97GKLmKJQz5YeSSEKRA9ychpxB84nmFd9hMFNX
U4m91cwpXSkrf9pDGaZJ9R6kYigj1tv1DuNtGHxLDJXELHr9IVP8shWsWQQuipT
wZ0sBwWnpp14/O1IvbfErVBe/pCUPMiQhJLsgFEKcCLt2hs0iWW8yftcTEKS8m3
7aNOZJjkjKvm4/KYO2kvqx4sXt85fXxfChrGWUFGGXgugk1cKo9jMC2WzY/iEcsB
0pkzkOLLALYxfPc2HwMIK3jz69hoQwYz0DAbwtQQoChb/bbueyM/gwJxUuor4BYP
bWKXSfcdWDLBUFNK316JHb1nZ1VDxMz3Miqtc6vZrW9zfa0Gj6KRooDTd+TzprGc
uzdj2WJKJusQcU4PK0SiPCF+hMpFzvcnH/8d5JwD9BhJTn8ITFL7zHc+ju5k0Vtu
2c/ascRhbbnm6roX/SeZzoDs4kcYzQioE4GaDxyuzfbEbNt0We5I0pzdiV/lpd9Y
NqdrIRm1D1NjuBpDQSZGkEwCtD3y0RuYpR1LcQg1HI6hvhu50v6r3cBMXsQPycOn
mvjzTOZb4uv3Hd6uck1fGIqarFfhfoLPuqIvwVXJZJXFxkPEi77GbaVGcRHCGZC5
aMn+VjvRJSiAs0IESspjH/bQTIjP2hnrqQoYsd33v9dre4enTrOgZRQyo1GXEOFO
MsNT1r7QThBw4LdjPV1h1IchoebmOAixwh+HY9ahXkUoP184z2d6P31ruUpbd40p
18i3THpExutzeAPfQfsOhU7K6USyHT8M1a7NacGVqRISBGBMVg3QZEj/b49c5h+M

ymml2xXYe jmQVFLiM+3FnwAX9o+k93MZdICmi3UQHCVFdCb7fRqxrzrRLagLuXI
oW/M8CD1CLem2/wMINJwzpITtFRRZzB+op4ghtnLuIeOCIOtdRIrBTpOK5XQY+UO
fSmY1FfQ+FEBlyh4UNwarnSBdaTtAs6jyXzkDqtU6FYLPxqilbTruI9Mk+7zOXe
p9N3hHMZwNvN40PnzQgN2Bw4c1cbbqPHhozVfmbWsAFINw15FrrFzyAgeBff1hQU
k3D/Rdq6H/07XDqshc1fjgZzmev78S90j4cNC01xnxihU3/KA09fnBMHSYp4J1RN
+Chdh6sIm6tObJgKEzm+e988A3AgFzcYKVWhTX2nJ7q1vx/zb5RqCD2vVaBhS3Vf
0S5Hrkc5r19a1wLbsL1LbGNw8dkcL5lnhufvb3zbqS9k0Je jppJfs5JEM5bM9jcdR
bQxz6W9YWC1AHnHDNB6K5aZx4r0y17c04QVbUSAzULYQnCFJ5qyUvJ8/j3f7eNRZ
dmdj4Hkqda+Ct6tTJ/KPvefpL7C18QdiusJN+P6pb04s/9Z6PQjNnobj4StX+hA5
hxXc5dIQZ4Xdin8A8u jAbj0VjhbsBbu8bA1rflDPOfHbAG8onYF34gtzLLyC1o3g
PWOpqGcmGzKxzxwN3N9YfPEZ+VZI24EEE191fKQKyz2UE2/FiCa4cGdtrDrrfw4Y
RK9Eer1KY6nvqF01VzyeI1qxUv1ciTi7jd7Rpn+q92CGVku01PHOGMkBTWBiBhw
ep3X/eZGdV5WWZm+qnalooD/TxqiG9vymJkPzycrrxds7LgY1K5pLi jT9fJUAYfL
JcNvsFVx10YiUDnWmwSmRp59M4cTI+0hz5T7m8VlxB57bWmhkXEg79rQm/EczvoV
zv06tj4B5kFtxKKuAcYrgpZqdN1CQln0ae73eCdIZl6goNWty7N5wLaMhf6RsB+t
m6Zga36Ka98a+y6J46tpt1tvpW7wWpUMsDN0LRRgdCf1YQAWM78YTuK6Aob4DM1V
kgeDqA0ESmLbgB0c/mah50uNEPQD+/X41i8jv51wj3LV0nxyyzf3ehne16jvMu1m
Im+2vGokh9POvMPHIRJmPgt8QaoW7QyUDVo5G+n8t8WYHQBT8ZpCS1wg0MIuSMIZ
eHP2dr1uSkinIQ9fwnQRO0qQgLOK2iALtGCLE3BBYy1tKxDyv2K9jgxGvEkpOfsB
CBajFmYED6+/Ox0wTnT2bHzzy7p49vqE+EkQRVH08z0jzLa7KNEAMoku+27oyWwO
fPqiMZv6yoOkpG3LRgg9tHmPbCvqWixZufAzZJuv4/W04+Kq2Zq4uicGtIQyx7Hy
KyksxWIAVi18/bwt3MzjZTU3cav/kP2FLDos55ioXC0ZAC1dqqrMDZ/OqP5Gs jZB
WKJQpgi2L+zs0SiXbHdLmJTEDUQp4F5QSFE8HF1AAnHd/xx79VEOJPwubSWVXDda
dfGweNmFhaqacc7LMFracTy6uA jFRGeBuRc4n1ISbhfPbAr0AgOmUduGXh/QtmMs
hpcs5QNGNWUeuFmKdimpGe530DpPXWZt f6ERioKuacZVCEzmBkmHLTz8K+zm15yf
lHwx6n8s/hp95EsHzpQL166mrWpIowCODCyHAgrrtCqLMRtx02f105KqCGPRXvzm
He6TiQ/04Eiz2NrE1GsykFIkXaoB/uKNEXYU4MYG3hg1CoPN4BdQrPhkwf03ApF4
aRZ7qbZzkiuKGAVMC8oFGWS26yIwoyxDp90aLuzake3NLqVV/RwhDLAQtJDD9Qbc
i0q+ACKRS1XxEKRLj8u/8zw+MAPE/zcVg+tiPH7dS9sferMa0PK1fvWhfVVEiCAP
2j64xuWMAHgPMTleDsvLk/fvpVLfPo2qp/tC2ybmH+obUAgA3ad/repVvtH1BDLV
x+r5pDZCpftCgZKTYzSoWYCOfHw1L1DLbBe41MUCSWP IQtOxLTTctv1qISuxMq0W
5JyRfNaZ5OXYgqIhUwpZckycThFt4q2IfJ3cS06rcqGu47kCvMfytVWLNsuczkhE
PDBGhv6uMVk8r0vk+0jf8wJh/wL5evIY77qXPUIyufVPfoWJhy85oVVnJFqbDwX0
eoDk1VYGvi+0yhe+gQKMmXWE6GsHHPHrFWDkNnAPPRJ8xQqqtVC4cIHZ3KOHofFr
vYG8JnwCpdy2vkv4PtCLds+/jDIRLRvuCWD/HVk9Ove4eQH7Bjcs559eInQ+JSgd
Tq60srKAY1feM1cm3XeV01FJst1VGq+5DzD/XUIVjVzbEPMHKhgwZj/Dznt6AeK9
KNj8apWhYaYA4jt4wYA2tHyU3UuKvPEIr8+BOF7YLwDAWamXm1S/94454XUJHuHh
DQ61oKR0cuX2BY6Ze7J/WVyyUQM/qt1Q3R1TQwd5Hb+3MG8kFvn9EW2vnkr41jLY
AOzr+fmQyX5H4g/Vf6g/Ek6KmNAiNVgW7exsz7ZQXlraK0CEXJkPDzo9Q1e++0qh
O2XX2kr2FICjB5S8QoS80Z1Mwpc+J8dAztfk+hLj+vN1t3gz2F20/rB1XGXkV1K
XAtfo7GngbrG5PnKE2Yh7x8nTYdOdmWXDRnrVfwgo+q4mxeCiJbiZW+gohm2iV9T
FkwZ/AS7MDpR8pCDpvQfRyoTu68BmuVCuc/9VaiRz/icIg9jnLAYMyfCc5LhYUxy
spUrMiLp33LvsTd2GhmNnMXh4mWnIZ0Hj3HnizJrRzBhOrA0V87w0wUcDUzWfdf/
UNFtOX4IzMcAstdxAjDbDckem+z6QuGMYQ55x2FEmMLGjP0QsBZp9ESbpfJmqWJS
Ak7nYxqVtdJzFWS1G2btA13H5i6yynX335T7t1Em1cAtVcraXRijWOWz7ZoLtgZ0
MzgK0bU8ViUqT1G3bmwP1qFyjm75X8AS2rx7olard3CV918zGppn91jQHcW5LByi
zYHKnN97GVhKnREXnsrTQIe6OrvtrkKtOoz0rPG0gSY=

B.3.21. S/MIME Encrypted and Signed Reply Over a Complex Message, Injected Headers With hcp_minimal (+ Legacy Display)

This is a encrypted and signed S/MIME message using PKCS#7 envelopedData around signedData. The payload is a multipart/alternative message with an inline image/png attachment. It uses the Injected Headers header protection scheme with the hcp_minimal Header Confidentiality Policy with a "Legacy Display" part.

It has the following structure:

```
application/pkcs7-mime [smime.p7m] 10510 bytes
  (decrypts to)
  application/pkcs7-mime [smime.p7m] 6766 bytes
    (unwraps to)
    multipart/mixed 2314 bytes
      multipart/alternative 1435 bytes
        text/plain 487 bytes
        text/html 639 bytes
        image/png inline 236 bytes
```

Its contents are:

```
Content-Transfer-Encoding: base64
Content-Type: application/pkcs7-mime; name="smime.p7m";
  smime-type="enveloped-data"
Subject: [...]
Message-ID:
  <smime-enc-signed-complex-injected-minimal-lgc-rpl@lhp.example>
From: Alice <alice@smime.example>
To: Bob <bob@smime.example>
Date: Sat, 20 Feb 2021 12:16:02 -0500
User-Agent: Sample MUA Version 1.0
In-Reply-To:
  <smime-enc-signed-complex-injected-minimal-legacy@lhp.example>
References:
  <smime-enc-signed-complex-injected-minimal-legacy@lhp.example>
```

```
MIIEtAYJKoZIhvcNAQcDoIIePTCCHjkCAQAxggMQMIIBhAIBADBbsMFUxDTALBgNV
BAoTBELFVEYxETAPBgNVBAsTCExBTVBTIFdHMTEwLWYDVQQDEyhTYW1wbGUgTEFN
UFMgU1NBIENlcnRpZmljYXRpb24qQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIEh7j00
Boq0MA0GCSqGSIB3DQEBAQUABIIBAGR5655q1ldQrEn1+qj1llo1Gr+bLsb6vwGIH
YA/sZqzFUPrdFQZRoIqGr+mw9OFYhsaTjR+ZiK/19IZJUDSXOIqAN02kCRaLbe1R
822KrjNTYSKYNUI8mEMu1s8Mm/J3Rf6LDss3ZgcKKxDg5XqDtBG39VFTXgHVq5p5
xYKt88FM1Che6oMOBVnCEKLu9aNm6iaQx/1IPGUYpQfEY1VEFHEyJeD9UenyYR+f
07UYz1XOk0l79OlIxspqqrBehwscVirzy9XfDzWfc1A14GTtMp8n+7wm7BchMX/
7S86+FiypQQFv/nHoeEgE5Z4Cfm/m464/q86fJ80tv4iTNQ7mGIwggGEAgEAMGww
VTENMASGa1UEChMESUVURjERMA8Ga1UECxMIIEFNUFMgV0cxMTAvBgNVBAMTKFNh
```

bXBsZSBMQU1QUyBSU0EgQ2VydG1maWNhdG1vbiBBdXRob3JpdHkCEzB8R0APhiY6
 HGLS64Mv1sDXhpQwDQYJKoZlIhvcNAQEBBQAEggEAIK+kWh01GQu8sKhJuZf11zGB
 7uDFNxt/SEB+I31lUgQJuK6BjfXoFhDy0j6Wwi5KFfCOGip5PdSd/UqLidl0TJD9
 R7/j4ZIVZL2WBKNY5aFEoiy4v6/RAXRY7VNony/vSeH0ZTHyC2zC2mn5R4BU5Ry
 pcNTni458AedkjlZGhyh9qbf4XOBMWT7Se3P//h8a00rJsPpguLEr9eYk+SEmdor
 s/dvtN2Fa/c5sgf8Ha2j8zFEET0fe5727t3b4TPhLamne94RF2Ban2hYKyGthaOd
 E3s1E24n/cJP9iUtZ5FbFeL72Z87rQS6QKkRJUjyuutwsA2HzggcRaizMRVhyzCC
 Gx4GCSqGS1b3DQEHATAdbG1ghkgBZQMEAEIEEIZ0JLuCYpHS6PTGpDi013qAghrw
 yOPDrc10UUqt5eVulaxY+qP6Irw2lLxwF7HtbaDzc0iOv7rG7l22g1mfKvUf3vYS
 k6c2jzYBxR/f4ngS0oTGiZaRnEbD75gEuKOpwlmQDOc8Yv/NqU1t8Iqx8fq75VaW
 51SK+rw+BZ8AW/D+AIOKJxjqX89NFZaAkJEcohjAGTRz4wrUoLEpwFE5V6qzSqqF
 jJXm4SoDXH8ZAGMAlVyxxobZ5P04Agqn5CXxYkdLV5BoVhKzFizP6HtnKpdlmaMI
 Ct0AaJbvWjdC+vZ11igU/txiyp9io4VLFUNQjROGzk5p9gFWwQ6tWIF3tpsqGU+n
 cqhisLig6DvL8I0v4T15fYW8j09r0SiE1aBGRL6Psc4U6D5xeE7FosG01K8HVxfq
 IIqoOfr4f7eq4+cRxuegtLn/podCkfmfKfMFb5Naqsarc5r/63GMSufptc5RKROZ
 8ReYo1OJzNzgyUUYMzerv2J6Qya6ybcVHBFfLsK9j2XjGtFMG5MiXOHob3gJE15H8
 yWsNtiew4FXhStzWLwFHWJjPwZEQExqJxMRRmlCfJroW3NqCE1S8AQCseTab1jRm
 M9mVD78fG51p98iPe0JHYDrv3bsB0FaPhCaVhnxt5cdsqU42kblavmH6VPUqoygx
 QgKojyvQU1lzyZ7udh/M70eWVag731BLa62h5zCQ8V3F9Jly8s6r67da9h09dq3K
 5h3oxWUBcQh0rqqMwD23X1X2T5D46LoJAQIqOmb//askhoNr9BKL9y5K2gFQjI+T
 quMLP6ysZd+oszVbRtyNfKyFyJNmks0s0NZ5FgeLLc5h3y/fOM0U559PzVET/fEP
 R89dIDTt4lCRrT7N90YESQ8N+e/haJahnQDS78VX1q6nnrIerI/WLXR8eKQIL0Fs
 Quw/A7YQ4DOytsxOLUENGRunIPBePzu/gF37Dd81ZdcviTIBToLy1hIOPIMw2C0f
 vaqy+xwooSnwZnQmH+FiFuBOOScbhHmHKUjajmT/Ybx+A/8saXRN+SfizVi8tLXb
 XI4faBkFoVbYvuQh4PYHogTz8W3UjWhyVtmCicL55kMk9TSksxu5SGn+PpIFovJF
 zuxSk0Xm+7it3gtIisd++sZSRA2a/dYyFPOUnfOIBL5Nsq//H5sL7LYo9ynuJUD2R
 AI1wtAymmPt/+stRjbsq31b39I1b9A8rFyv2Dhi/p66Z6XLTSyM9gvCdBgxjvcp4
 opNEUsriap5zFtvDC3YvPmlYaWo2jK94mDa/F7VmJ52R32NGrTPf3h1prS+ma+2c
 wJRT/P2gVip86VOxTb+KgJSEGbIHhdJ9+gFjTNq9y0dgwhLqvN3rWFwU8H/nTa3v
 ymRTNEi/CCxcKctLgSckRZ5mMCjEJMqtqF13tT5BdMtUSWd75Iqu/uambE1iJ1/i
 9O3ZRB36f2uHGILpWfJTOyL4wsW3GgqteXmjBx6qyXhJ4pNc01y4HF0XIiWKKZEE
 0dIY8Rpx9c0Tw056YP4qHXAc2t/jJqTM9M6hB8y7Pdnh5XlW41tCc2qLXuZwKdqi
 uWHnMzCVmIykubDh87yZgzZb8BaWZbjYfnwXfsv+XgV7hiBGMB0QYRWFQQ05synd
 LzvG8WKcBdTDawuwvz2riK4n0p0YVBNTnJZBXsvS3GG0Jhjt05c+pZ1rLmRAUnu
 eosO3ZOb0z6bZGt9goGHAYdoIiOinUYMeTAEs910k334nCnTh79FZAd/aEInLupq
 dx2aNBzv0IdCPiERuWvMu6QLHQ7vyqou7ibu0eLWcs/IhJsnMXLj//qx1Cedax/5
 rerO59X4kK33h4IY0Ql09CF9Xs1CVhpVnVuw9Hp3C5i/fTdq+yR3xdQ8CAJWJid
 UfZb/nIbU2460JAnRvNi7LFW4Wwywv6uDbC3zTeVR/KUJ4Zg2uIMcpLUCUSVXK6U
 vTviCdljyuUxDoPjMPpf/s+4kCkR3ALqyZJMf6owMTBw8sQ6U3H75UNGertSkPEY
 A0sStLy/K5wtqoHbsANK8iUNFmlUdVh3yEafzz1gNxuW8y7xCN4ihlaBF6UBJf3x
 PggNcA7802kUcJeaFqC1k8WGodnaCy01XQA52xF45pdJ2HzGHRpKm2hqRYBjNOA
 2kS/8eTiufFmqHvoaXVvNspEwNaH+aJrsk9Tm1Pk5m/lvjB3kYsGofuUp94MPYAv
 PG96GHJoPNs4/KR88ECBQgjVruE9ozj062S4EeonEIQOipulAj6CXxYiQv5L1YO+
 HOOGvLAKDa506Yf0rcQF1ys7IVDGokVScJqCEYzIOfubhrw413Y4S16ka+ZgLKq9
 DLzuMXkNMqL7WqmK4pLx2kHvLqqLP5XjynagJHWNm0zYO8JDgWHxjbQxmaoNf49
 oIFXdzESzLnz3T+lK+OSyJjq32IJZbCWCzWcc8PX829b/KNo/a9VD/UCpMMz6E1E
 aSxE+ywyGo/gpW45d4ZRJxzWBT00BVvhrUC6NYjseSoNLUorVmWbzkqpnfO92bfi
 L5Fu4YnKbh6VCxnEUOmAMaCzXIWO1LMg5Myep9JrNnVPb+PYKhQm9QwVpwhxKwsG

+ /MKZ0eOjMHd6jk9GZxNDV0VuCcmtiLPuBW9+AxcAxjU5h4GH9fmH1ZMJDSIBDaR
qt /D1jTognJK26Lx8KmQ7yI /zUAKz0LwALxBbnV3f2600LQfqrA2MpTvdt0YKPCR
brmXI8ROZjGX0H3HZ607P2NRum /9hJAQx1 /ZR74gu7FpPUUIJjr3JEHQ3icNcS4h
9icI4wSSl0FngY3ONUdVH0tvCEYsMexrZo2rk7qasTFHoHTrbk jncbhw /dc0LXCX
79wCmue63UbkamFUz5827rDiRpEd7QwWg+RexkYeZ+b52Iloyti82ivolKeBKp+f
vsF2Ni+mag0zUPrra0lg0QYmOk1ZiCm+GtYNp1t0ROT1Yhlcq0743B6qvrBkqSM5
wYqMR+zIVBbqK0Kme1+C9AfPF4K6vsGmJKnRPWHXdsEtONzwGM06HhUhKXSKPYTN
EKdsM2Et4dWFjHDYBii jVna47yQbHVCm0a /1l8tA7xQTETyZoDdxg8e0WS4NnYSM
0nUOny1bKIN+N7Qj4brWegRmOFDvxas9He /msMOoYNMVWXM0Kc64UfLL4mRcq7fK
YVY90E5YKanWsNDku0NSbK9zI1QZt9ASOUvohQniIPGBNGO /X9JAgIsg7hy5 /z33
58b9JyBpufXxeCUP430eEm3HHQkNwK72BxsuBZK1Np28fdFgstOVs jDXFdmUpFt f
0jbiQ+GM+GWCYkfx7mSV3kCAWdLHJLQCEVj1XZbhtK6y5o0G9YP81m5nQnqyvyM
nG87JkhY1MpzPGKIKTxRHCPTRkGqxxkVEIOVEmvZAqZ3fHXzM3cRvRYER6RB70KYU
C0gvzTAgBr6W10ErYNKysjD+QG8FyfzbfYH /zXumG8jiiEqNKFU0Y0AxoAKHIQYH
a2Cz4Dzcbt9YdTf7V1FSFVWZspRivCGCmqsfD+pbz4Xc0REJf+fG6K4ytaIJFJqY
fVVe+Ecdt3oo7N+LL54jA2MBrbktXhpnHGMD3WaksG /JMOrMTKpcKEM6J0ouOAX
q /TeKF1fUKP /6ig5fN4HwCHRwXxGNThBvFz+gXUvZ8IddtYEqOpSqJ7z1PvD610
vqBFovrswN2E6hiLMGwS6n /P /o4HHbLXVBCCA9w5sApMsnfAQK5DzLxKiUU3xUjP
FIsFIVxWMJ0aCi9UulfTA5J7IOdCeoldJ2j3BmAKfHsNcvN8MfuG3gHLX3w6n+Bi
oXQVQqqD7pliHfXccgxYk070CtCuxi8OMB0mWFvDm6BHWEJx6BNNCoCdpVFtIF36
g0Hb9aVIc06pietUr45Mgw1AGCB05Tj9VGOROnErdQZChejOw7LsozFNT1x5wAnz
okTLIbhvHcKKNUTz5Lb9WwE15o1DRxmHfa8e0jYk8PrjDfJ5hSL2n /ug+SCb+w7dr
hzFsIhhhAFPt5Ezv0vdad3LAG8a08pgr+K+AbAtwth9Oa6ufLMMeUkR3AqrbTL4 /
svQX+yVqQsbEp65SgN4h4g46ZJLlyY5i38yXi5a8nFusWbLH /gW5qHLCN6103FuZ
NQP5L84K8HiBs7yKqVE0qdl+Gs jtNKUND0LxV6IsAobLtcX4WoYUE8d2FnfY /I2a
xII82SmhXg12Chyyzmz6odQNF29zFBVYONZS82N1JroHHMrwvI /ys5odt jNve9kT /
xKCjWAqj5X8rcnmch+kL24HNpFntNbddiPdfVcV3q5+Ma2V1A /ZH0BokPsj15yrt
CDFK3+4x5bRnFbNuMWUACVeORO0J1DHMwydG8jhqFv961NsYnKrVQShJwjmWSifP
I6VaR8kHo8ZJP93NNpXy7GnXeByF4hDTy+PDS97u1Zu2eXo9 /5txg4Ted6ts2tVa
L6nBR225Nne8t fasxOLnp4TyCOFbvAskPLQzFIAUv72Rh2iGxPq6S6300grFXD0J
kiHNjwh+Ixz+lp1GsK5oafRg+dAX09APDibR8X09iBhWtIJD9Rs7EsW1EX61 /T8
y6kV5CGNSxbFjiYgkNWF28EdSy18ipmd6a1wczNJ /uqvFxef /Vn94KqwrHkOwuIq
UwtXr2j /X18+0 /RBVeLARpvILQM37pWKB9T7+ /09QyAAedyET375Zs0Hr9sYcSgf
w /3vb9HX74 /cAGQVtQz2qeqCr1cSgKBd8riVirakIvdgGI83DoIim /EcHD7rKh4B
Uyb2Z0V5Mi70uncXn4MHsJwrpfiFZmgcXUfOKE35gNAqbTNi+m01z8bmQ+VO8qF8
Fj4hW8JjmfLxwjFE /gh7RjYOYrQM+JET8PFY1x6A2iJ51TKsCOXFGO5o0Obngv
01rRy8LFOLncR4f3syZhymcccrR6obIdqwdcz+1+zWdoLxoQsLQKrTqKnJez5GXC
kRXQ4YNJ98Ly8M+wcAz5bZCeqoq3e5BCCB8Z4g+I0ryLKirnFvSbXx1QWCiQv7sG
QDRFPve+moQkBJw9UfVdD2C+ofjPUZd8m87t1bKdxoz3lYSGVny12b1McsNUTQRI
Skyhri40iIvcheXuaAEXZ4YCW787ABIyc54DLv1XSnb22Pr /OJGLSjGDLu /U7Fe8
3iE90fCDPsfUU6yAsfNLRr2LcZhnRw0F+siRcEHe /naDountYq3W3UB5Vji8k /bw
5kvLoTUEIEb /UJn05uHX2tco5qIqdTyR2WL8BCLekJdpvzg52M+e88BX4S4coJ+w
MlgyXmG8TkSXT9GLGua+JEyE2qk5pG9dmhTO+K1CqKdrJX45N1CEh7C617sWC3rg
rdc5CQFh2gYv10ZOWJ76wn+LA5gUTU7pvhgdeDCEs3dTwyNHjCFYJedBH0jzFG4m
oJrThIYxfMkPTBLa6htHIgutpdOG6GD9nP7mKimUXq1jP1iaJMHTaQkoRGYsvP7y
2019eMvOQm6Ppm351ZOMpJnA00UwHLMJjWvH1WvTvh1vKjVKMwKscd79fybBk6XI
tHfBwKRHQaVQ7YvPUDjYfuyAhnJt1016fRiRN1MapwTTHg3tVZE2QoTfkKfM4km9h
+VQdyiUwkbpg5rFCVhQWI0+imqKFwoATjQm0+352eJB63jgvH7o9myg5RU+AK+6D

ssSVGjhp9vGOC3KbGY8ldHVhFjCWEApJ39REOxe4YkcCeaYTDMqhlldlhZIUWNbPZ
EdCnr1GaZ1EGeMQeu+Q0mIBM60ur/Mwr712cwMtzmbasFiC8zARsbkZQZh8ujXep
yMiWkXXGPKdYClal4pjoBmLrPaOXlrybD9K5mKZEopbpDPGYzge/C+tvPYCP8KpH
MGmaHYUwWdLlIPi2YDOFL3WAW3fa8ugJUNdnYV7I2sRAOql9JOQZaYxeGby1VJZh
EWRsYbauamQJ7TUORdBoivZOuJNoYKP0wJUipEiK7ZgJR8pvP6HLEoSyxu8dgVTS
gi39NrfE34xH+TMpuDp7K2florTNMVe6WMryOonuTCln3KxD2nCXr5pT+AtUzmZm
40lYzwDgIBlyNgSpxX9FML+mFqfT3mtfLm9Kt5YiF4/SXiEfi6Go9VV4xM/znwIg
RlaORawjDtZ+CzPsRU9v8Dr78xHFhiAp0ohwrzmOVHGbzK4d9jtI8yHqLmPEpKH
mV8vvDNgBbzKpst2Z6ahBma2hvOiI3JzE59PUxdg4GBQz20pieW6ghRaIyIVJVg7
Ot0cZ+wp+04X9pyUtKaEZMDfQMJO3F2Z/dvSP1538NsZieYj4PNuF1ToGG3AbB6a
Ccs3wK7TzG4bQtRnEUk5121U2zm5uxoUJTOrfS1iOKs5jGXN+mxow5H3D6QEGYgI
nzBhB5BUuRoiAJe8uAbUnT4r7aSB/LFxv6NP4HaF0qJv2YCE5KdV2//2dHNgL47k
pql9Cw53XRjr3xUnLO8+GjH5MWNfVwVLatSLBNgQrLswk2IrbHjEHcECrN9CtX76
P4/CkOcLqx7wS1LFvu82Pm6UHqhb9Ke4K075tNRDAjIDJ5v88/zbsu641AyfVXxma
ifHuNKgYhd9mk1IEjXfTvJPouyI5a9FabSs7kK9S+awuENvyhSJ6PQ0+MC+J5eW0
yW5SjQcCIXSkIKNhUTdVLUmEgjl1a7KRrbDjYF2u5GSa/seY7192laHnoXWnC6W3/
rGt+BsbuJhf+MqZf24zVWUcFhMJW6t6a/jguD2QH7opt9d7NLvzLNNStARxR0NAQ
0hXxldj3fk/6hrVO2IsuxPSAysG1TQhrwEuNsp8ff/cJhCj1XQ9JGoiWYP3+niaT
ZrYoaTbPRA/NOELG3Kmdsinzn8+EC1AKsh8cy8EwtNdl4MGiZnr0tZVJ3Y1YPzFj
wRr27iH7c1IzBfCkOV2oxvO/mEYhYxLffIuid5ph23QtSEa/4r2/m1H1LMD3Z1Cx
/6XOyeDx1bQBjnh0SEVoE1S6ATwS14sGE/DrNdVhotrdDHEBv6u9vcOzob5o4us7
mWBGfO28ypruRWxRaQ++H3ysrW1GPZY710jLjM0BwRiMg4aY7LxbbzJU+tf3mRbM
F5Brb0zRMKiniZtP5zKqIPTBIfvuymfQbrf8pEELVnSHgd8ZFWRUeBFgIFGH1i3c
VdL+n+utTjXUXRSkGKgXc21AaS7sU3ziloPgi2mU0TsJY20F4kWznPtUFGn36zBm
QM7sh18AFw+rskI6R9kO9v1Bd/SqBMxP16Egy0u+O9203iNKbldpyiFsynhd4Yj
oR0Tzr4KZF4KQ1zyc1bVgsrGNJKx0L6SmqYIchkwaP71VoZPdn+XYr37WSPM6U71
SkRkJMkxr++p8qgnY60BHXQW7u3ZBJgkSXuJk1zo1q/THVeNe/gDA99Qt2bc4YYZ
JD/9naGv4a6hzT/oWXvCOLmcdp4iN9Q8Z7Oc7GrQDLq5GdBnIogVIhCCUY3WBn0
XT1Lv5tZMztOsIxYEA/UsxgtMU0C8kRX2PhYSWFFyRKiF+I1EwZ+7NjCDtRI+1+2
hIG6DvYiOxi3FBZtyZxkBaoggv5Ah3wOPf4URjds7s6HjgvLdHMDJkuFL6qdUsG
fSn7+jRCAiJGkf/MCMBEH1bzQpnY1xT+LB93rguGV/PkoFFM5nZ0c9ZjPCVZ/ewv
ItqkF2oXuidYmLd3STxoH1MF1P5/qNruCWYrAo/M7dJ1W12zMwE9Dr4+VJ1OBZkw
AU1Sd14XGTIOLfby+cCS6RhSMf8XqJ2d2hxUX1hNgOaizsVp14HCTddKCuVfyp1z
t/H1EZJnar4UsLiCwsgB7vYRMMMA0XAhIn4Rmi3Y8HZga3/jLwHtGdPFYe1fVwOc
6VVeFVA+21vmXS4nKcOFgGWhLTQ/u+xhJMfY9mAzZSH5f74KK5FcNspC9/mOUQmv
tDvcoIWIJdxoHVNwCsuSVW8+IS1+25wST0wShD3sKaTVhgFPuQGbej2wCgirZkPQ
82FCxLDkzhL+goh85EGV8FuxMoo6gb1krFTxDF7MGdEv6RwOyj0PxLEgG/ctyu0e
Y46Peb435ScUFXTa5jU6yGOjHrzzjNN74wArI5FtFI5qgTDcd9DSwZFh15Adbj81
TamImut13IE6n7v5kuTnqEAM2y4He5d0Vnv/Ms5+1a12LaPgwpkykbz3WdScD1Kxc
+oFUTNXGfsi9C6/DiWdAB7btCmMxVA0KaFPq11HtUAoP+qxrrqwwL3aa3+rtC/wbX
EqG9w+6U6eMBbPw/li++M1aiAWSq7e2Ny1T7i3wy1V1cpSSFhrn2EX10IS1VmPwn
f9yzUwQ6yk3r5CaOXg+LmqWrebMnqXmYtHICGrzkk6c25sKY424S/d2ggJeCkUp7
MHh12qWj0rUtei+DKx3SjkHXhct20+t5E0zmaGQgGKL5C1HR8ODX/pmRH5qWILUs
F1K8Uf+NP6Vwmf3sYpyWchMKWRm1AdidbSGfh1fMarEh9kpxEXuGdcvqxIXfWfHm
ksitbzmMzHhfYx6UtN6VTp5BfYma3rD9dgAQxmkgmGKhEkKnEu6RLq7MVXwh6Kq
H63fldMdx81Dphv6tcpD57BS2748MbIkGpVGekpwg/HQJb4YY9bPOPpMKzrZ09w
aWdf5qJ8NK638ZEpOYFxoq71EAOjL5JrmRmhX9OuxyyIhbR89v1IfnCPnozN0s9D
DRqTLEi63UbiVMfSYTJz01Di0sFoQfMM14/8vqwh4NQU3b1C9GcMf/hOQyezuKvx

```

/UHnm64IeGuF2Q875R340q4T5xF/iQzMb6uBWAHCfVB3kDrETQ/nSGPu9qLWMkeG
RkCBrotadhbkddyTbqM9LaqIWPA2ROdr5W3PU0h6ZLUzh2hGRiF9pQ+wLj71YmIX
5FXnT3n2KzCEVc6XHpU9c+6PAa2nYfIgcslI8I1yyxJERzDeIBNh7m2ihYHyFQ+1
GGkjF2pWvVIN2hB+KS961UAwm+1vvRN9wx18YSpJ5T2BKNkg0pucDUYP7KYsiRd4
4TCHEqK0JeF3CzYYt9NvKHCulQMa49LARmcEndoKMS2975EqTpq0aP3TpnS/81Uc
E94iZftUsFKhs0yttvYS/fw2OSp62hmT2JIab230p4jd2wpwP8GA1KHZwWjjbRjB
F9vrhTYbWntat4k8AeEKj2ZjHJMOGmG3sSx33JcaBwWug69Pg7nEcxdP+GxbGyTZ
fPCC/s5G0gxtUc+Xk/sv6wI7gbd1BYAQnBVs4wUVNMw=

```

B.3.22. S/MIME Encrypted and Signed Reply Over a Complex Message, Wrapped Message With hcp_strong

This is a encrypted and signed S/MIME message using PKCS#7 envelopedData around signedData. The payload is a multipart/alternative message with an inline image/png attachment. It uses the Wrapped Message header protection scheme with the hcp_strong Header Confidentiality Policy.

It has the following structure:

```

application/pkcs7-mime [smime.p7m] 10185 bytes
  (decrypts to)
  application/pkcs7-mime [smime.p7m] 6526 bytes
    (unwraps to)
    message/rfc822 inline 2198 bytes
      multipart/mixed 2093 bytes
        multipart/alternative 1140 bytes
          text/plain 379 bytes
          text/html 477 bytes
          image/png inline 232 bytes

```

Its contents are:

```

Content-Transfer-Encoding: base64
Content-Type: application/pkcs7-mime; name="smime.p7m";
  smime-type="enveloped-data"
Subject: [...]
Message-ID: <38a0b7ba-76e0-5351-93e9-f44877e20e6e@lhp.example>
From: Alice <alice@smime.example>
To: Bob <bob@smime.example>
Date: Sat, 20 Feb 2021 12:17:02 -0500

```

```

MIIdXAYJKoZiHvcNAQcDoIIdTTCCHUkCAQAxggMQMIIBhAIBADBsmFUxDTALBgNV
BAoTBELFVEYxETAPBgNVBAsTCExBTVBTIFdHMTEwLwYDVQQDEyhTYW1wbGUgTEFN
UFMgU1NBIElcnRpb2Ml jYXRpb24gQXV0aG9yaXR5AhMPLSW9ETmXSs55CVIeh7j00
Boq0MA0GCSqGSIB3DQEBAQUABIIBABOfkRzWpe8+giahAB4aK4FyKBN6535VHc1c
4f+nf8otkhBtrdWQfFeOuErPGeHzRvmDmaMtJFf1J24hsbhV4RbQ+mbxJPxoqKT5
qOYSj218aZlRvM4E3Y5Cy8i6iFGDOKBVSc+RHv+UukIOs9MhLC3K/Tmf64MQKYL5

```

sGAepPwv36xSQR3VSrmioM5SuozXl892mtuk207bpPiTnXXs4LHCgzptWc85vq4S
jtS2AKUMUQOcUvyOoKlqQsERyy5BfkXE9jkjB90/ba/No5LUBnhfhYJpnmfEeU2F
JBldGcO7drxF3FQNHgvj49IJHYEXndC7L8LkDvL+vh3XSTvedLAWggGEAgEAMGww
VTENMAsgA1UEChMESUVURjERMA8GA1UECXMITEFNuFMgV0cxMTAvBgNVBAMTKFNh
bXBsZSBMQU1QUyBSU0EgQ2VydG1maWNhdGlvbiBBdXRob3JpdHkCEzB8R0APhiY6
HGLS64Mv1sDXhpQwDQYJKoZIHvcNAQEBBQAEggEAqPKO+X6DScUv9t34OnIzHRne
LvUyO32lwpOwyc6rFSLrBto/WGpAGS9NQBGFlk7roGzXH3BTMnzpF/sFj8rntQT
jHHqm6Cqvam9gBlnyu5/tihN9eHBCjF8M6OYksj18TomW99tm3KADcoe3AvPEOEg
39AJIsis8c+sayVKEG8gyeaDn/m6AK1AqeoWXJ4yZtV13B5J/mc8Td9m1dPQe74r
JuInHR5tM1DKLe8Lq79zs3SwcJgNzhKt9IH75ZxDTYEI7Bpqa2ZF4R31E1XrKR9A
fRa+/fqrMjTKFm4/1jeqVD5owrjk9iv/T3caGpI8WwGUopeOqBaeyYeLkjo0GTCC
Gi4GCSqGSib3DQEHATAdbglghkgBZQMEAIIEEP9IzMSducnaqC0xK5rxgoCAghoA
ZXRWVTctm+0Vn4Pow0nRtA7FkfnVfUcKZNFwHuZPpjNFDLiFPRr12UjvMLA20Wr5
8cISjkBlm/wWzoW0XAZStAlX2kvEluvdKtjS7ly/kexutYDclf456v9+H+IQZkbI
xtjniKhnd7NK1kkjvzRRcZ3D9MFfJMbS5ISvHL1055I+9vRzHJIbwExwL3ReRhHo
lcjdrMRzs0sTsRYEFyf+xxQT7yCTfz6xglkzAfrq+kilCCMbcJCXZOH91kv3HL6d
oTjW6RItWpALJQk34M1HJkHKU8yYnaIyKcDwu/b2wMgVhy3hcVzUIz1KoquOfm5y
gotJrpJOQqGqjfuBkzk7S/R30zLLLlY0wAXbEhiJsCepYrIneYrizMB5rwGGS7wP0
JPGBRwWUkXjPTlzhAkGRwk8pmFaiBC5KzF3JJ/mXwNwCT4lfgu9MX4uqhpUUs/FK
16V+LjSonYGATec9K9405eSszRKTi3z8BYtLHI0ur/e/P/easCJcb09zd3okONCC
83WKUaqw7VBLbxckfVDYg3S1VmdpHXPNrFUqkn/NGAKTYhJS4Wdlq2rKF0FPiC1S
6qux09Kh+aYJYX8SjbcFDBF1+1UjmIBAhNMqzbUaYo1jtnIjxL5fqCP+Z6Wy4izt
lg/zO3zVPP+ZGi7i0D5eBNYMBfHMZoJUSK+HqVvd80569khEGoQGwdm7SrjRLbf
SKcPu93aAucAlk8S5ur0xmR1nVbdpiu/VDYIgz4Vi4RFV4rjvU7aa4UV5rj4XJSS
IpBMlGYJhZ029ZxPGC7e8Ji1sClnYl7gnT8aMWeNkk4GN7ATWFZ0qoaENepGziUI
yRr/ORn1lumEmrTGjv3HZ7cbtovm0r58JtiUX/2CLaFAwFPjcp4GVfDqtsosrKdUIy
9xk/rM0vJshXr0UoEhfuJa6yqiH8F1/49UTCdDQgcKUEfkwza9OromsVngEF1jzG
AMTOPDVQRXcYc7DYaqzDiaIBXKYrt/zcdHpKeOYttHY707OKrzNEZTZ9y1q+k0jc
F6XuXMHWWcFN+Ca64d45ABOyooyxgYWRwQGq00KoWY6eVaub9b607eeTe7j8+HKb
LPQCop80JFHFOFohpGIAcWY00iv08tzfNbp5DNsktTd5ADog9ZK7WqIhnIPzOG/A
N32EEjnoMQIHVj9z0Vwthm02Ltnqw7buAj0B9gtua4ccM+taallzKHKBKIXdoAttr
L+35BH21J+OF0hKnKG2nLooYdYPmsaApQboaGy0bEnu3FFICix9yn9zbG1BD1IR4
GQthFIMiyxnkxSndAQCu6K4I8RGY3Tm45t1ugAgvopmd2ze5SGnBWovIG8/+LImJ
mcP1JFMn0EUdkWG+ckNJE1Q86C7dVpOLAl7Kkp6QWBcjmN1+WQ76JdmTMEPox5dV
cBY3xtbYlKOHHebwqmnbbq6R5GGD3dB2mU+41JJHzOyx/gqco2Hb8MDhbOtx9sb8j
zQlRc0NHrDzbrxRWZkuF1LUPJq0OtxWqVOzqdII3eUHhV3gzIuHW7k0XORJDqx2OR
jm7+dve04VHmoKmCeeDU/iLcG1B4ukErq3b4sluein7z59KFv5oQPY68z1Pj40Jb
J8bv/fL8vo7kEmsODALXYSetTtQY9h1oZ1jymzcz7gJAu9JQmkq7f6G3rzhabNan
el2tnTWLFJXOecpKtnOJzH8EzRWVZcelJKhgrUgzAwQeSazXXMCeRUdOe66EhMCT
0pBIzJKvuY0zs46nwrDc/HxqfdX4aj80wMoKjx1B5rAdB9b9beZXiZXgdY7kupg
53UyKJot7efeWiVrsrjk6quek6AuuZtgLbBHURM8kIPtEi11cnGvZU19Z2igoNs+
F7U/Arn/Cokl+OqmCCqLC2+nqf5UvpwSxSx+d3bz7b2osYrGEU2iA3sW3ucJn11v
Kf9thGwNgigqGDkDhejtYUD2PVvmeBpsrkUK5BmHW1OulhHvliL0KmZog5xUCdLK
wkUD2hntSMBovw9A8KS22ZzIKm+3U3tUtrEcp6WG4kcK0EQH5rrEU5/m70+sbSOS
TpXfe/0pOU4XounNi+K/LchIME6VizL58vm450DW99JRKRvssJpjY111XHVHr5uH
joKs+9xTapfStY8WKwX/cd7J8B2yQfxU7iy0de6kGLfUZLgw0uur1xrbOzVko4FR
q5BWOWgFKX8GUC14SFKGbyxhQG1V2Up1o4U1oLt7SNwA/Rd3Tcmxy904YDe0yHfL
VM8JBR6dOBGhHvcYYjEd5+1ITNFFkxLglq+MfFbi8eb0qfHzNexCShN2C0IRk/16

KX3OCOmRyZAusQXk/S1O/tdvUFaDwvmtiPBbuVaeTmiBIwMfLlqbkuphykeTSgWm
dCU9uBNjhI0/95BexW7+ifLjVpksBbKiyAdHe6+lcnszoNrZWW5P9vzqoPLLUjR8
DDhmEeW3ud6QUGZ7V8qL6Q6cxBpS11DvqIp7Srf9ue0RTcmv39gIrhoEszybmhZS
pMIInSaJhuYzYfXJS1rNMVIhms08DVO/6k7pm6QA7pMwAI8rntSEf9Z70Uzr5Wg1
eCSuOxHqL71pU49wNap8r3YYK2PuMUtGEzw+u9Hbf+F86NycDfhmVIET14Q35jQW
/6gpJy+hibsPLpPi6ne8bnvkdNV3dxFoV0o/rD5aiQeJLs0HvthxsMY7qpnV8LoU
FW+fN154VyGg7znerTYHyO1G7tmsxs/ejYaT0gG1rk2WE+9XAGpkZhapB1LUrCu
Y3fF0CPaNFc5kYoy3oNAambJViYrZw5J2zjaL4wb0FRjE3dmpHU+MlffRK86ke/S
MGai7HW/uAVZA0QpUxUzxnN/zH1xHNISrgStcFeYcNbcx1eMxoARVges7PPEYVKy
QapYffxV2BKZag4215PkxQmppYwpO+gmCYg2ff/Ilu2PN42BmfTe40pPrWSejuDC
CD3coLRnsgTydRsmIAB2XaApMepwEp6Bp6PyHU1BYRZBdqj+MuxVG2+MBfLYOS1R
B4V6ZQ+AA1UFu++eOVC5umDr8oCisITepnc62S5eQKp/40ia+JUJD/Sc1WNrGGO4
Dy6/2MC05EgKhQxNB5TmUlq0Bn7/JZOWI+zTBypsMzcHuiBHKy1CAvB3FU+W6L0m
XEuLZeokayYotg1J1fL0qZSziX2RFub2x3LEPer/NsXISfmfvaKa/2ZHPKQjmrW
FC7447gHXyrU1SzBH4SHPAoplCZMAhA3N439zGM90brt1Aq6XVeAxxiv1r0Ahd40
BrAD7ScUBGhPPEKp5zY7p4HSe0hEYdIumVmKOKY6Jl6X7Lialpjlus+Va7AEAhTv
HWKDWrlryafb5ixrAxR0fKltFtqp84yw09gUdf6Mam2nY/BFhktiyfmp4iBXDUD
gAAciNotDXSVMwdA4rbCGDf3Tdx+rJg4ny6mGNY57F1jXK8SdnLpyhb0EEtyr/Ot
75LCcXgBPmPN6y69pRF85ezLeUMINmzmLUQqTVupRFU4rFA2NnEUnFtbpKp1AY/H
pdKfuP6khZU/fCXRoogGeC2LkLIsOfCiIJA0hf3FShVH8z2hXhjsNrtx1aLUSFxee
rYEG374iuRjwcPkZC6dxzrcSpWhfRwS9nsVLVvXFskazr2690WfWiMDVUtM+XS1p
YefZr/+SRGnRa4Xwj9F7b+CC7bHT+otFr3IAay+RXVAifjHypRUpBVTpk64mp5H
ux8FF/qhgbLjP1PN16ZB6LXV1/HD7dK7Gkqlsfu2GTmjpQwSFx/SMXbdv1bfiQT5
8tR/nY6ZvL46jp0BTxEgezWwX6+SvFaQc/AgMo1L5JdJIUCX3+QUOKE0hVP2PMaT
GjVuRivkEyWzh6eKk9YJqdmB/oCptKFpdEOzh5yqgtPcmT6JQuJ6pJH7fA65E+i2
k9beHY1hd9pzcQiy5Tw7AcXsRX7SOQrdddg/ZK60kL9b6458jJLLTH1R57t89069
qGiN18bdrFenh9TiqqpbwqTAcmlHIKU8Nc+zs9Wbk0eqeMLMpEU5R4TO8EI6ojrZt
gn0hQw6jWcbA9a+plxiF2ShRYSaAcDvUybc4hSfMH2fcG0s05cjcHkVjbu8W8k09
tKedeNatpRXT3DJWAgKIhh/oWt0Lu3ulGCJxp1f7ip8E8195wrnDFFfIx/0Plrjj
vJpL7nmF1HoXqVLbTyreDAMLGBMYpXv1HH4ef5vrz10A6r8jqoDwo0pcLQzzZC20
4rLCKScgIC9+6Cy8cfd1tGkoMLb1BRM+80FO7pUwpt3/B5fnMQ+WshXocVYbcs26
17zUgFWV4Aga3TpicWqc+EUAhYt7DEbQ3c773y08sRjFihhacpBrI+7aDFJpbFkj
SpCKzY5ReQxbdZigcbxic1GGaUNQ/qfX28n/RvgIWgAOz7ytSme2pcEmp+jJLT9D
JJ88hzFliK1qLGCRwj3iVjROpgnAjd/yPpwB8TNyoEc5UrDNObOR1RX1dJWQSkRF
onCJ81dfXJOBnvtb0AABLwvqiA+jewXyRnD22gxx/m+uD/6jHJ+U587W/Yhr2Tg
OKR2zhLxgz01Nc52ik8geeCH1KtVWaKWysUg4CdINQTVtBhM0LtT76F0qW+AP0eO
yrN1F3ZfynT67Leat25Zy8biHCLaO5ccNMG5SEfzugj1zGosW9w/g22cqZ4k3FUG
uWagQYAZeaP7GZNGR+Mf2/x8YTqk2nUoeHt0Ehk3YQ8NMtgcE53T5Pa5op4sEQVZ
Mr5+LscPIMKOP1Q35uNIkhYMXtZp8/VNuERa3UvMv53Njc3THU/TGjfo+Ye85wss
sGmI68E1tTkYDhB2GI dmd/CD68E6Y/u3xhShP9zDqBUh3hHHjJbFF8DYpA9ACBt1
Ad40Vb11fXoAfY4ZtQ1UaObgkAkXyQ4yROFNmpWhS3RbRUsez1ie1Sg8PKSLy1YE
bbvvuQGkaYBivNoJhcdFc5ELqdz4F6vXHS1jzMg007leyL6TSFs5nIjvXSA3MsF
AFPGWuoKZAdjmwV3CbUgR7pzUJNBtJ1KlasPUd4sIAPKxp473AwUjYyFMX/2tWHT
DQa1wVph+pqad4n/GchKN3K4Pte6RTT7j2LG34+Woud0T+LILS7iVnw55PsP4P2a
qh5Yt2Ed2/wSN+WnPBooJPig35fhI4AiAocKjA9B01Rv85BaVi5UppyviB7YiZNXb
sU7BCMynM1SLqWqgL9HjaBlXzUNzPaU8zkzJva+/qkah/61CkCp4FL7QNNnbcQgi
sQW+C5Xi2QB5tDWNmRkRF9cwCwicpERhri/rQZqq/WV1BiDmbCEgujxfgo4mCse
80XUNsOqfRz40UAMIPUyZwaDiLgl8jXjQ7JTyJ8SmOggvnc0CAPua92rToZlM70m

CPzt68j9JSfxpyrGqQpa2c5CP6qJi+eJGfUoLtmc25vt9sYilZTciekJmNDRMMYR
7zDppxLNGYuT5Ly4afWq30QOUsk/CsOxd8JNsZ5FFNbG7uh0996CxQjJfSwah9KL
6Xp60mBNsYuD9ocaffelf2ShqF2KSJ/bkSeYcAIJ72m0l8EXPn+zKu5BDoanCRct
Y0A7rxp3N0Ga4T6JQNortN1w8mFfeUWSwi4PRYJFqDkb1VKvapN3oCovj3wqelwL
K0p00yFDML9/SxrmBFjioKf1lKhIRV0IA6t2+n4wuJciyUY/lGQqnn6qQje1GtJm
NpTAHHMgM7eJBJL6Zpmq6Nj4xnqiaoAuvd09Gjq1KpFR38j5DW8BN1VfJ+0fPoOD
nhLpYtWLA7cudQFWKBUNazW6YcfZeEzKExDdEab6CJ5bhZgbXEiw4Qde2snuVkJa
MpqvXgCtKkT6Vvm8embk JrNWw3ge10MRZQHUoBnv7D+ai+CveXKEm2sBMLw+qN5p
93ZHIW9LDyeJn9Xc+nuzBzqKxoA5UXA7hkPfoT9BVgIOcan1UeMtguyf1VjZdKCI
LzXvK5Uz5ZKIUK0WuXmoZHXPcCFfH/3VSpME1LgRXxfWRi4pYyuxFFW0gRPNCizK
MSHIUDYbyzdTPi7Ivp4I2vUTjLVuiQSjYKs4SFc0EKsP3jFxpQX1vDfu0sC2h2pm
kV3w15903AEwsj7VXg5zUzLMJ+8Kkv6/dVvevpu8+mIpuBQ6nv6roYU12QWeqPjh
18as6/TS919xm3ujanRQN7bxBJ8LBHUJPIuUe9iIj+2Yqv1YQFj0GdKj1Ntn4kS1
KFTg1Q5tewpiCiHndok48asn11TDZQrcncQfi/bQmG0BUwZni4v88DYhfQuxek7
hRWqcFqRziFxxInHI1+ABF0Vc1nwZeAiwwanRSgP1UzxMDRIkFwKmpnQC4NoDNAY
ECsnUX34Ffh/0hx40cjbvVxpUcQuJpTiN9EIXtJs41DKbwk3wWe9VfQCjji1khsh
X3K1X+1PY1/UvqHHfxHPaTPKNtrjYtWnASxLoVdF72o1BWGSatd/QDCRy38oVNF1
9oV+WwH6ISalLQJugqrcO2uVyIzsiKwFnFR5zqb5N0MdYSu9hXZ+j7IvL+ixFSMh
AKuGK1nNhKE91UfJq/rJoV2brpAa2PVuq5Kd1pY4MN6qEUy/UrocyPxV9cwp5d
IF/XPAfHFyBpXFV3LBiEOCvEpUc8TGNu119700biffTjPF8K01Gp8X0Th6uGoj/b
WWZyVRo0a8nx5W2qlabeKQ1waZoJee+HkLeuWqRk1Vb7kNsJVH6bJiX4zQErSyts
Gyz1psT/kms1dHic1EFAUKvqYPm05t90d+sL9QoB7XxpM0mtsMtgc8n4XXdoCf7w
iSsmnrQqIVPmGBKGUBimxvWyCN6mvWgi8ElgmBwtvdGlsPgAqr0nZrGs4gvd1wu1
Aw8mhxEE/brrjPs7o4BV13Q24eAfr7ANJRQabPapOie4EWExyUdaljkKsoLauboR
s+CjiB3TdNdRv9zfSBJEocFnQ4MaMvdYXKDVZ6ayeYVkrPCBP1RCMpwHtr3KrbCh
1uHPrtsLV2SQc16cn/EzQY5Lus6aGyB/KDSF+ONAUhv+BziNvh8ThGFB4L41xIYH
0nNdek9qtNoby1pJ+DAV/CSQRfdRrTMQuTKI+T5WqB8BVvvHCqQBP78Yyz2Do3K3
2JjAve03MdmMvDDmfdICdYmKt2Lc0p6oE60at1051zb/WUvnGcPKTVuJH1CnTQ1s
wI5QG7ALhT0MpoVmmJUstqgQELIeT1sQPkfl13g6HpG1V+42V3Gp2Ne4oMGni7pr
cSSoAAMOeDMUJHObx2B87iWUUpKC6UnlaYflgIixrbiqba4q4ZYrHdXv6YWNNQR
Dr1kw2XnPHdqRw7F0rvMCYITEP4Rc4DrzMHzTHI5Esp50K2657QkYinOwb07Ki65
fE1I7MGkjkfc+ToLDUIz2J9irtdTczS1QD1cghISCHq7jFVYjdt73ffVhUS2Nsw7
dlL7RX19TzmeyYTCpkqTsVsZ1ncZruj0fU1j6m0RmFQynMmD91zn8o5+HRBiFODy
plaipknwoHZjhRYiHqooZo/ODOYHQXA+0vjKQqquJKz9rkDeannMedtBH2Uq0aFW
jPT2P1EVsP591VXjwWwo2jTjk6F9AOaSB0LW0cwYxJJ08Ev+/NWID0WMEBwmoJ4m
cLxub2XHm2XUdgiXz6EUYReMoMzBSKfehJAZ6rkUxV0i7ZYRLBi+n1RN0XIkTu+o
4UKMLReeTMcKw5yQ1x6e0aQcRxw39FLgCRjF8e+feny2rK9OGXUoJgVU3+1LAj15
dQSi+dw+RqmvtncMqmeBhuEWF/KYbcvTiIRqMrPNnYE0CFRL+y0xS7QVv0Gvr3YL
WMOTTWJZ1wK+JDkrToS5UvoGo1PNDzi+md3sYV93BYqbMvzXvzIGF1wq4+h80wH6
0p7TMxaQK0nHVh36+FW0AZpWApF9NTDBMFxsUiWFENHs8wU13XBgwRBpDuoBqX0m
AgLfBgtXspJq3Qv2qfX7/ltEhG3FP8pJT5iu95AKQD4zm5UaIxqpJLCIOeagV4/
f26RrrdnNKJDPWuT6tke7tD2bKg6d8HJXh7FthEODVu/47P1kS59flwTswKpUP5L
Yelgxeg+T+gzcvaoJK5Ymqo1bH5dCEfF4GhZddT8bGDJ3twRgUHir9mpqVtn3C/7
/ak9jF6gwK1MnJo2QD+OM57TmqhDIzFEvYRn1fiIaMte4As4msnmsSULKG9i+uZ
i5c0Q/1xIoUZ2AZGMGvYlGsAZomj7hxiEkfauxUESHU+Bjrc6JiTzqt40o1tn2YP
q5FdnVsdCilp3vMwiH8K+vS570QD1U3Cd4qD9+Kv8UnFyJ5yc5wF8ryIcT+Dz+3G
bRihn7DAjck1JohqpiF/PnDzBQhUwKnc6Du/GE91lNGv6iEOJbRqeyli8WGMsJBj
p2zTWxHy90xvXqpg9Jci9JdG/ZQOe58RS8hT1u129qRKPkupf+L1c6GZqomxZ4us

```

h63bK4GMIjTOKyZwU5RrDm2Lo5EXizbVfUtKLgaZolxtVdPpbVNXcQNjXEPPjvrZ
HxJUuU7gfacXyeJwqj4+9Mkh1FXZ4QEaueqe+ZwrwAXLS+cN5PNNAKcEmYXnjAD7
dDs75K+hx3/LtHe1lbmYPjG0WwyaWfV5Tpz84PSz7FR+tmFbjnalqwLxNBmCGDDp
vClISYOwoWcJRMvXqZqTqWUqOAOggiz0VW1l+RO3z0TYbJLJsAci+AczKYRyzLGC
W4LqUchjKmgzXr0U17ERgR9v6doa0p+aJGrPflYs+VJZE5Lb1hMO/E/nrFtjCIGS
AAiD7/MLA5FRO0L72brj37aIXMrrZ9fWZMo5EwzRT+P7hzGMcICyH+1/52it05q5
K0r4TYyD3L9oTEpytBI7r3hmf6hr59aez9xbWhHaQYU=

```

B.3.23. S/MIME Encrypted and Signed Reply Over a Complex Message, Injected Headers With hcp_strong

This is a encrypted and signed S/MIME message using PKCS#7 envelopedData around signedData. The payload is a multipart/alternative message with an inline image/png attachment. It uses the Injected Headers header protection scheme with the hcp_strong Header Confidentiality Policy.

It has the following structure:

```

application/pkcs7-mime [smime.p7m] 10140 bytes
  (decrypts to)
  application/pkcs7-mime [smime.p7m] 6502 bytes
    (unwraps to)
    multipart/mixed 2125 bytes
      multipart/alternative 1144 bytes
        text/plain 391 bytes
        text/html 486 bytes
        image/png inline 236 bytes

```

Its contents are:

```

Content-Transfer-Encoding: base64
Content-Type: application/pkcs7-mime; name="smime.p7m";
  smime-type="enveloped-data"
Subject: [...]
Message-ID: <c6774fdb-3ef5-5293-ab2d-eca8b66b4bbf@lhp.example>
From: Alice <alice@smime.example>
To: Bob <bob@smime.example>
Date: Sat, 20 Feb 2021 12:18:02 -0500

```

```

MIIdPAYJKoZiHvcNAQcDoIIIdLTCCHSkCAQAxggMQMIIBhAIBADBBSMFUxDTALBgNV
BAoTBELFVEYxETAPBgNVBAsTCExBTvBTIFdHMTEwLwYDVQQDEyhTYW1wbGUgTEFN
UFMgU1NBIEIENlcnRpZmljYXRpb24gQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIeh7j00
Boq0MA0GCSqGSIb3DQEBAQUABIIBAHw/91uDg1fJb003YLEnXot6ooUedmQUuwrV
0+AAMXpx+Ag22aGkQndo1Enr51SPudU674Rqcmd8GhOYv/SN7k2hJHcVJ1NB1Bqk
KBlndk8OZ3CmHiV04gDZUaH0CvHsXFS/SV2fixL4CuPj1/Ko1O1AFuOU336iRXTe
cxiI6UL/n/feSVf0HNqSFgdnQs1/3pQIOA/33mSJBN9gLSZIohefKGYgzhjIO9EU
T3PKk7A59hZhZisoldMUSnuHOMRRHGbfPK1e9mMe3s/H8LXkqRXFeb9Dvme3R4pC

```

GHEEsT4zJJqOTwYc2o1qn83v22k1Tych2daG/sMgDp+1nYV4KIQwggGEAgEAMGww
VTENMA8GA1UEChMESUVURjERMA8GA1UECxMITEFNuFmGv0cxMTAvBqNVBAMTKFNh
bXBsZSBMQU1QUyBSU0EgQ2VydG1maWNhdG1vbiBBdXRob3JpdHkCEzB8R0APhiY6
HGLS64Mv1sDXhpQwDQYJKoZIhvcNAQEBBQAEggEAPe52qn0+vt6h8MkYH5DP9GdZ
UkyDSFBx4fkz1m1OivGHVrmeMAacHrU0EItthagq/gIoX3VL6+t0czMIm+19svu3a
tXUyCjDjOFS3gXm1wxg91rYWunz1Mj7sMBRt3RjvZXUKhluL1kz3f10J77Y9GoG8
rDj+BnVM4GHuKknTTSaQDYsXnarJOFtLMHFtMefuAf4bSxn/WyNU720tNYG1M0/O
pE+SZPEA+we615WjdmVjwsBZTlhQKxV8mFsAmsiukjWYAWHn5ZaPS0xA8W80NyEh
GF68xjy1tYBwLExtii2NqD+4at16axj/odar1/FTLCG4fUJeBWH3/ea6keEr9DCC
Gg4GCSqGS1b3DQEHATAdbG1ghkgBZQMEAIIEEGkoJQ9zwwq8mv0aBdHyfuSKAghng
Z6pgVbu/KHUwPthP3sxFazxNC2ZfrvCGWwUFaxAZQQR5D3WhHqUYWhWoMRP343rZ
NjZzBNA3KqDRoZ3Oj50M2ekjBb8d477Q2ytFz3wuC6+0jxFO17y9OUQBZn1BI2z
HdqO2YJhdmlaLkoRThsXHCdSsr1Jx1sp7fhkA83CcKai7z3T890f4z8q7pu+AUvG
v1MFYxQ+d63ezTucWXdjbbxgzN9iQGLP4kq21IeccX5Fr6gzwwotRCQsXj/wyTRX
pWjoVWfWedOoiMbAXsol20+idiam88MhdH0kSpxve/DAF51x14X7mMJJfogrSnao
ebrrzg+hojw09CMJvLFBVnlmy3EcdRfpeFsxUWK1Xnc1UycAv5jNHkERmz5gK056
a1BQFGkD38VsiH028KT9uNbpInx1FNsvfJ0u1YMrA04kuYcOvbUBDnF/ha8Tdj+v
d6No2b050+jf8OTBlIe1khM8jV/Cy3pYqixAm70gH+USuvVjvjLhBp/EJ2xWA/mv
MbvbesuyVERZpnvoQ1A3bayJAV3HyDZ1oJdmEM7/ynY6J1GpQaloTQcfvFbhUnYa
ooV199V2kXAWIj5cKEWFoLFHv3wgYQPK81BpqxKlp4/ZqGpnjG6I7liFNxDC7mzU
dNCK2fUu2XbSuXz1gz2XNML77LkD/0Bbv7c1ktiCQ6nNcd+Zhb2oeGO/WO1c1m/a
5ZFI3pW07vLNIAaOTQx1mBUOc7tvYi0Pvmnj1k+6UshdT2MJyUagcz6yPRWJftfg
LaPNphyRVTYPCAoy7TmfBNoy9VssOAbbxq8JjJOL4aV7mS0J56BHZLUNH4GQMYso
nEr6b75sRMov1sYAinDf0fg2gAzWrdAO06XjNQ6rdgrYbHPN7WqYhcstN+vTuGAP
ze42pN5L7ayKXKwrTivHB2jliP5pKNat2jz/MiLczfzEvSgburwpYVqk9t8zVEE
ICmsUK+vaF+GePy0LO3/G1bVBFPHGdFbTB3DAbo6R1hf+sys2/xR3Lc/8+mPJThO
3gAoMXTRRgBx4pTgiltGF7JjYbSQybNZ8f4Y13IOZ1uStTTXa0f85G0gYpTR3dI
cCk+ftDU3UALldQEr8sBm/hdWxYJ6yL5kw34R84/vL7yZhs02z3rfVv1/WNFNF/i
TX8G14PYT3IZo6AeSZ5Y01Z1/xx81D8t/azHhX+1n7LzVaZj2M/2/tqI22wWNjZb
yiORjDSjeJ5TvyElqVIFXYw7dz8vK0GGzjDTx/OS50hlmVhJ1rfY/IWMrHNhSVAP
H5vcjQ8duMhbPIWj1/w3bhOL6UwiI+X041cElTeABE/ZLfgA09Eon4+kbXWGBJMM
BYqWtsp8/tKqZQC1jWS6drh8v08jP7aMRNblNcYS9ZG4fpMdtJ81onJgDeLdUks1
uSH1CpGop3XGbfvOHn1YS+m/ftSMRvfJUXKIixKHRLIhhclwaxKXWzvf4Q4Tsl
iJkK8UeKOTXI2bdaNGkDGvW2Abo0YdiDqCe5v51XHiPecPxoGvzU4TT3625sK1mfi
4f4X4f9X+E7e+6iSiF8bs5rJZDENe7AwDLqGpupCYO618Oyuq/VDcnHFMCsgvvCJ
yaBk9nRIYJfL7H0uJyn6tj1Cqbu42m5zCM4ONiQ1GN140SgJykTKe5opSy5nkDDy
BMyBdnspolQ19HodvLtL923VfPDlcoS/MjSg7kRVPqQJdo7odN5sJUD9ldnFI6he
97w39ivE9zeGQkWM9gQtS0fy4QN6bLxrcqSbtSKpLvd0afpbaE7/zyswtPu1yhsj
AidFKrgOqyuiRdychkA06J1qSsbiBpvkOsFmeADqdKnG7lg4e3tmGME0rooIBfHq
txCMG9QzMeBaQVI6TqzA2xs/ta2OrokiN15YzjjHhLgwXN2Sr3eOXxUR3LNF5SZI
HrzY+oBoamyDFSFEJLAhfOJABA/bruPwCzIzraXq7YtkOJNZGSK1CvMpk1orMVrx
vdMcoGCT/UcGOLakk+3r6OeuHO0T4UWwO9/vEyxWWqUZusYiIR2hlzTgBae8F4nt
QLhb+sZquSC0a7tf90228eK7nfmUjXyhk07wTzkfL6vdxPvdzfrAVDMTMsE019aw
XcXgn7cMshA1qoY3GJwnFKvvHwZD+x81crpVEMXUbLnN10nseH35EWm3DHvHJR4H
ET+jbiQfXiRs/qEZAvpIzE2c4UuKEYyKPF3fFNKJ1/qWgAh6o3yURYD49ayP+7gW
wJYQ4Y04aaGPxURZxBAXeVS3t7oK7ptTa93ism8fxGVJZofraeCX/I8VIgdTXzzC
QI0smZydy+GKcQi60U2/S0eoQ0zmCd54Fh+Mg7YzJsyfxGhCoIVEkDknyP4rMBr3
71BZD05pxqWsFRoHun8Aw2nhb+TIUNAHK/6iBHq1RN1jhesfc5d7yEZDgva1RVDX

oZAhXBxcKz1GJGd1At/hzZDmj8MsxoIhRN6pCvBrN1x7OxJybtnp/6dKtE9A6VRM
ek/zdWKIdHiZOnNnp5SBnamRCx+pHECFtTuQyVmcvzbH2X/itmrxrLPiAfdLk11tW
Qv19Vo57I3MKfEWPVWVsMQs6gDk6n+hfSplhIKHS1jv4911B0RELdp8Av3i jCVae
jjAqi23xwAFUE6EtniNwwGyFGKMdbHRRNGsNiaUS49VP44x/60ae4cfUQ0t1qLXW
Z/fmGSB1LeQUqlnv1igfRW6u8bL0bRwrN+jOPWmxxAdS2ipjB3e8PIbNHDi+sYtW
B8SRWcQ1pDUETyY/hG17pqRtxFBgRZWXaQWMXwVh3lcexasEd6j2cIRklCk/70rf
H5zXVSw3LNDps90Xa0k9TnP5x1Yt1L89SDILy1DU1fpzhwhsyS3V5fhoGCdxbilS
qjA/pYvqjC4m1IS1ytjj3bMUvbP3x2etnqoVSGAtaH9ewHfCEndFlkMGIq1Wee8i
SC8hvNR8TcWIANzWxqlUF018EMQEN1OTAjE59K8sXa4gluyXjbN2K/DibdbzG7hL
XY+oQxLsW8uJdlZvfuiqLnmu1sNogAgrJCvq0XTG6dx3MuaTC4Uci jGpWvS0r1Xh
FO+4qmScEs9tg5xXRqRRhbu5BXAJ/TR1Z6vaSKUoeLQ49MC7CeBO6XTKHSPp06x5
Fjdyq189O62hnqKFa8MgMfwx+vpyyA4KSVPN36W18EPmYNABkTm1TbL1+SHwAMVX
qhDuDNRZv7o176CYrQrBqunwzGhV51vhkdT8uyqV9VtdfdpL3gpQHbqqIjSQT6/k
iDFMI81QLdHXv028jFSNL/huldQ6GLuO17tPsBWVoIcaKCFoz63dHfOQzPupT4wO
ZmDv/Yae7wLuhhDCFoe26A4mAWufXCKfdKouP7GygaLVzi4V2bYVmVWO36XDNDyI
6PETY9bQU+foHEhMLKdMpkb1LziWtclv9PIoR4dwKnufsnnCbZsgAPankJmbjP8p
tHvDrctJvqYCZHSyTqT5IWgOAp3c8K/RxD91wiFvCkEca0uZBUqTLwZJ1bbKLxEM
hLmtBn412q7ic+ud3zT5O2fAeuAw84tKKKbpt79jxiaz5EOATiBeEYmR6MNxux3u
TDvBabBA6h6Sc6NbQB5QpU8knGmoGyJm7nwNPsJtud7oQ0pjT//XIKAGE4xBLAT
qB44uBhwJETObjkeWKqVV/Umnv/TYf7CZaKIA5udixJwg1OLldPAXgNXRXVX2+2K
ArZABmju+eEKLZGqF1LIXO/20BaIJUbpK+DSappBovKoTGdSTfr83OECfvuP0BNU
+A2IkB74WzoVJm0oRGrhZJZ1J1C6X50Mqc0+RXtm2LBaa7k18RfnUQpRr14PPJ6Z
JL93AmFFZgGLt9N8ITg657MHvt2rtZpTb8c4vBDSbg8kuDH/CMYZfT4CpG7TmHTC
nevVRYNHwj/d7Kd+9T6UMly9LGMnJtP7yXPWuldLGLv0qk1wRQCfVN6ePHHLAW10
b40r6tL2kURqCL0QkIVxmJx3Iypyq4mRSnWcZTJ16hvWVW9P6e1XERXUSWF0GHRg
9JNFaENt+p+x8rocnrV4+AOg952uhH96F++0szz6T0am37SKfUFAvJV8XdtZwyVj
a3LAh8vJzhfV0WfRv110UxIZUVP4qm1K+cTpj304bE0hi1gQL6+26s34Vrv836SG
Gae+hYTGX1NFjReMi9r/X4YY9EDpKC5eETSnnZYSkP50163vDsVtTmZfkSXyT3vY
7p1UaF6AvZTdhapMKCelEq0yMiOMNSIqXC3VX12bd4miHuP8Z6FgKIn8vtc2dNpc
d+d3EA0+Gpt4L331okogHAnEHokiiZkvWJHyw6UDunRmJ3p0AxR1zmgGbfGLeuYV
BTP1XlyYHRHuWI+TVL+Qvc6c77Q5QRvX6RVLxeqSW+drnkHctGX4eWz082xy61s+
SBoOxt2JVPYvyiCA5cTkALyVh1bak9dHMPVe04U1f45c8mApm6xPT20187vnVBxd
gWwPxVaC90X1qXvaTvow08yvgLQPE0+eISkRCm3X26Wfyck8W6HsMrUE18Boa25H
/Txq2TdRTjkIkaE8ek2YOMdv+JFnkxbgUEijjJrt5rYDzD8M7yTePkrq80chx2WX
0qujD5dUkYXsGAB0Cyoe7RRwsHuzc39c3NMuMzKm6zBY2Q8jcC9N4ANzS22iq95
1nhN5/7dUkByuRmpXNqhKmkP6AA7h9H7YNeG8hdlmRB+3BeFidezv9t1PGs/mtdz
lmsI7yfIPDTXF/7gF5KpcwAhWQ9uMySeTHBZwrLP8mNoTcoH/Or7PRGUOR5Uvf9A
5GnEH4BhgnMKf4MB/TbhkNMocB1Jh2NFiq+HlnJRxxRoXXjZdIQj7wF7evcwHIzxE
I/BSUSCrLeYos08QnOLOHbfiJZM1thyqfJC2Hc22zmeIu7wNRMAlyQZMv/0z8qAk
Wd1MTpT2jFbn/uvFwuEBv6vbKC9Dm9NADBS9xg0P39FmhYtzCmrWuG/gQ+JP9RIe
vuw9wwjqxH+VEUwSxNtSAOFpyH1m2ggWSQuTBRFf1Sfj95PUMn6kgNFwaIxxLpow
quFfqhz5HIzdjLLAYFOz1+MephXGGNm/H8UMAV8tO1MjBIUqbVjbGSKf1p2oSvqT
+9q928fB8cDHy8rSFVUjEMiJT9uEQHBr7Xk3d2gOHBJA2iivjxcYe2yWa5qJZ1WB
ObKTxaLVbLvHac5XdX1vNtzzF+qo5C5UGRng93IIfYxw6V1kF6kQYJMusgceMLN
9aWDHsuVtdQR+mNP9FOkktTQ3GzYM/szBDi+ZaPmkswmnvA80Q4Qbrxp//TZFLkd
H1TiQpTk4XgQws7k4K4kv16K7Fn9snqqUBq9ODaxrEfvH8JS6pvuIvf+wwU0ID9H
23jaZ4wj1CkmzWj11G/jWBHiMhaXc81vS6C61OKyvVfoiJWovSdhqM2jgm2TYBSS
NI6hVgLPaQvFNgZuKopRgHjt/OQXFQBCUA0ijEBxBJ1ZDzk4xSxo5bsw+85W7Zz7

vzePF0LmT7Cy/qkGQW+RO4ID96w8Lq3+qX0aAi5oPwvA7G7Jtp+BhPucvehn3z5r
bl/aMEcoIqTd49gpcYZLqDPaD0SsOYBicShs/CtwqdoYDgwkzi1WfQK3KlrsJxPd
Us2VG1us7Els0zQKz0pJuFUz1xxyz0339tuh04Kc39DNPzv1acwkPHMVsYHjOqmD
zeWxpxHpiVJYX1V/CEHaOCTQH79WJZDHDWaiaXopVp9V96toArzz9nzffm+psJL
Gqv6P0DZbGxecnSXqQNw8nucoEK6pXSooFCpCCqWfo+xi29Mv3gA982UDEDubW7D
zpc6b3luSYEw13p7VMqWsbWsit zjt9MBq9g354SWnTMOF5yabvRoZa4gj2j3Of8Z
9pEkpEgHO2cQHEgrHvpFuAiNHk2qBmFiIp0/MUIeUOXVsrD9mUzoTe2W9YYeIAu9
4yE1cT1apMhOoFGurW351kxbR1GQ4zy+osg1kbuK3kAsk0HHkibRR/sXLMrHgy9Z
gdi3Kw2aU4nyzzMqueoK3rtC5u1IEfHMsRU1E76Q6TfS1gcITGDxwzJ1T9z3pfa5
lBet5lV9McbpOpQkvxGt00KvmVcqdxVSz1ZF3j15qkyz20pn7uyUWr16r4ppqIPk
KMkiOzLCKIiWfnnA3dDiF8a6otgX+bYgGbwXooZ8GIzIhqLkrJNvF5ufeZGaGSCo
iNT24WGBcnKJot6Zrr2K4mo/eNuvTrYv4dZt/rmWBUdEyug9VK0fiSGFYED9hUDA
uxGpRXxIU1Fq5w0H1H1tNH4mzQRIIMds9nw3xCbvPDiwoLodalK6KDXF2fy6EMgt
xSCLb8A1W8f/S0VtaDorNyN1ApTvXWX/tDSUa10swZpJBNB35vrYh8NOcK49j7Kb
ldEnsuzSROZX7hPzVwC9z9jS8IqNuX0nPr0mNli1gpxPOuW3UMDNr6gKBznKqcGo
HnWD1l2Air849gN1EAXcGcORuWb405d0hu61csSvYKvaEj4Mct76vDaeFEcb5Pzj
yUQ4Z2UFpp/KsnP3B2CE1zdxu1AstDRdO/x2dcDWLJjUy3c2wM+U9nvHvbxTnM12
gx5UV1M21UHEm4kiwAhYKjOMsnpx/HnNk8kqP50OB1WwusS3JTr76tzBtzQfocqW
HEOMvMy35x2Bh1q11PRTSh9c3mgSpXIPut014xvNBtVKh5GG3rTZf44qJkMbwy3d
C36hOWWkV/z7y5e0xERArT1CsFP+uDdGny3XGUPi0yjjz/XFy3UnxzSKGVQPaO1
E90Ezi8eMNRtx/gBy0s9KwgUvam+3dG525y1Gvbio2mrgLuTI2CKZiQBOTICXkP7
/A1RGp9W4wI23/Xt3hDW0XuBgvoJb6Ux1NabXMBov8MQF/KfWVJ7nnhqQDrRujuo
ya9Id5L57bLdP4SEHCWLVPERMDzRk9wpeVgivKN29Q2hhAU5RCgO9KjXWd1moJku
4FALtZErCqfkIHdLTN5GKeL+kYFI fUV8CV1r6D6MVwpN5QGzX2Y/+iat7iS4C4dY
MZ1HqMwkBRdxYjBBDYBiXGILjhgMGQ8HyZV/sJSYv3pDS4WfghTW3mSNqQ50cVz/
3uGzNe6ZkbE9EyGe/rRVCiBT5HkCpabG118Bj8MO+R19CM7ddVvO23WbaKt+Vw1
f+yzK+LAELR3XfAfqJPo7nK1UE2/QOLFdw0W4/uPbb61lRkp3lMW9NRznAQsUAuT
HgLQT7Q9hn23wBTiQwiBS3kej4Gi6wVW2Cj4o/8EPR0qn6ne6nhGhgcYHpkw1Uz6
Q19vjLyUFKjEo0NWOu6pgyDcfW4uGNzvsdxcnvRQ4+qVyHeXLEM5d2EhAw+TzW9
vWDpgYTTa/ZIILVjv3f4iKNZYs5PeUJWLX1IPQbrPPKFevufJk3ld8K8QRuxtNvx
aKp+scqFC36GXvCrGsR1HVaaWBCGKCL6DYZVTDTaWIwztIvCXu0zOR9D6hnsbmFn
t15MSUwr2B8GWm1I0yVgxp9U0tF4uTDFo9BlnPpJ+2QYjUEPXv1BqjEaw3iQsBK
h6XPNfRjqrRXJCbpcwZSisqSMKpgh88PB3F5Hjr6//UgVY4Z1wfyLSUgyZFIKBmKZ
8LAdemKui2WtsI1HMLTv+yWcbf/6m1F6qx9Rb111Q7OxGAP18JkfvBdNuFqu1iLm
ir9x10Y+8j/GcaYOEWc/CHxduAqpr03sEz45oM0kSD8ZfhhbfuYH/QrbEdZQd87
FkCzNVdV3ZjGiaoI4o/0CpmBfhU5xN5G4tXY9cCfIXEpkqvO3/guo0lkbNWBHJJU
WGLKvluSpoa6C9bfnaS9xr4YZjo1D1W9odFC9uE6aHyMNFkTt71YT2sTmbVG9Y1o
BWKv+DQAcAi6BECVv1bvY9UyhicbzGLFXRmFS+/pGSi6h40eF7uEkUivm1ZYnN/B
yKL3yEqV7CqpUYrBmAC5RLj0pgWsBER6B9wf5gfRL8LMZp3l06g/w3yJgH434L9H
Su/VZmVjrCzZIOxE/ZG1GGMUc61+Z3D/9lQMeVdWs94YhoFT4nn5SREDVa4+4YWw
sUokqK5i6los9mYlu/SJPxnwdCZxk/GyRRqH6Kk7IW2iWVXO8DEn2+n5szNLhv2E
7OazywsBB9jEH+CfJk1mgC2g7L7RbN4TDguMZvNGmtK3y50or3wRDMsCBX2iWG4r2
9HYAchFcmbEW1CL3A3y5MGIFTrrfIYmKAWB8foM6hhWWFVVTITxPqlvSZ6QXzOMA
VA7VL5TVx1tJotzLAbCKoYSRVmtJSEhsxTXHcWPX8YUpZvop0/dWsY6uJBkaadjv
Xdp6MyF0WPqs3TYKFjZChueaP8vq46vr6jP15h3tpxi5Jj+TWgqbOGmmn7reJKvx
xNFpPHjydvLC3FbHoda/sE+cbjDup/bbjsUdZIVGu1g67sMZc0Xk+eIiw3RIzcsO
f+c0AJz+6bGZ/k8xryPcGO1pud37J6F0nJZH9TrEAsjFJQtVmZoYbHDSzq0MVHw0
J0YksygeZn0aYHVA3gxfVcG2PbQpeXfnZyUsQt f jZOOEH9Wh1vh6bSFs+5TFbiUC

```

Twxyn5ssf2yjsxTrI+kCx1RfIe7r5/etsBUjQzPKju5VlXcg5msTqO2xj0QFKyjjZ
wci7X/lzVJvf6T/v//ItTWzmUFEJ+Bux0vo1jqdxlsgglwPyAEgKBoXVM4E4OJCL
vjc3vLlb8Yl134JcymIrLk1D8etIJdhNMsoil6oy7yFtyxmqHjJ+9EqbJRheflau
JWP7++n1NNtheB5YoLLGoRfgxA8pIpDrFlUxdYKN3mBX+IdaTk4f+gXoNpTXbtRD

```

B.3.24. S/MIME Encrypted and Signed Reply Over a Complex Message, Injected Headers With hcp_strong (+ Legacy Display)

This is a encrypted and signed S/MIME message using PKCS#7 envelopedData around signedData. The payload is a multipart/alternative message with an inline image/png attachment. It uses the Injected Headers header protection scheme with the hcp_strong Header Confidentiality Policy with a "Legacy Display" part.

It has the following structure:

```

application/pkcs7-mime [smime.p7m] 10790 bytes
  (decrypts to)
  application/pkcs7-mime [smime.p7m] 6968 bytes
    (unwraps to)
    multipart/mixed 2460 bytes
      multipart/alternative 1449 bytes
        text/plain 494 bytes
        text/html 646 bytes
        image/png inline 236 bytes

```

Its contents are:

```

Content-Transfer-Encoding: base64
Content-Type: application/pkcs7-mime; name="smime.p7m";
  smime-type="enveloped-data"
Subject: [...]
Message-ID: <aced3c9-111b-5a4f-bd80-34558da32b4d@lhp.example>
From: Alice <alice@smime.example>
To: Bob <bob@smime.example>
Date: Sat, 20 Feb 2021 12:19:02 -0500

```

```

MIIfHAYJKoZlIhvcNAQcDoII fDTCCHwkCAQAxggMQMIIBhAIBADBsMFUxDALBgNV
BAoTBELFVEYxETAPBgNVBAStCExBTVBTIFdHMTEwLWYDVQQDEyhTYW1wbGUgTEFN
UFMgU1NBIEEN1cnRpZmljYXRpb24qQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIEh7j00
Boq0MA0GCSqGSIB3DQEBAQUABIIBAAqBquNyGXBSi563D5scoeCEhSWiHeZcEBof
53CMvSnOvtdWust0R7xoMAJyq8ZDsQ/rIWOAvgm3xYi/8hVHowZtCe+dZozlkiG8
yLla7UpcJVVoqRZfMKoHwgySP0vNK+1BhgSQSPO6z1ilT2HBMeMBwjJ+6y9/CwOnr
hRXiQOWlBTBcLF/P+rpuAsFtv6jdxm/jzXEMgQe5j/aConPchgGzKH9XicC2YOz
RZDJs5Zc7cmnefTA3f0IH0Qa041g6ST8EnqimWsec/eNaAEakZOZZJRYAhgLXciD
1qjuByWAAAn4h9KnKXWg3VtZpX3I40YMLw319TGAJGnP5kh+DScwggGEAgEAMGww
VTENMASGA1UEChMESUVURjERMA8GA1UECxMITEFNUFUMgV0cxMTAvBgNVBAMTKFNh
bXBsZSBMQUU1QyBSU0EgQ2VydGlmaWNhdGlvbiBBdXRob3JpdHkCEzB8R0APhiY6

```

HGLS64MvlsDXhpQwDQYJKoZIHvcNAQEBBQAEggEAdhmdRrcVpFpMT38ZFuEl25Pt
kTT7HYAcroSov7Fuohjk7kukQyTQCG4y73sHeu/FZ1IPKzxxOU3kfBEbJunPykkc
VuFJPQJmrDpk4j5dvSqikvqU9cP/GliakTrCBiLdb7DO5jsA/8o+30mN4S8F4Mjw
gA6BY0DOT97FeTKpMohtlGhGpTtrVe8cVe1C2QPD0rKBYEgwJ7t83mzyaa j8Yws1
sUak jFY9hoTuwLspdiTqKbuUvEzaEaKrhO10WYqoTpLP jbl33KCo7fhtwj8zeVbR
Gb/1JbKsc6y/raPG0sTZXRcmQRmAjzEaNiYAmYaP6qdL0VuBQNDHEEf2bPopuDCC
G+4GCSqGS1b3DQEhATAdbG1ghkgBZQMEAEIEECnSfmBIh5urf+GVWT5DQ9mAghvA
jKFFJHAo+gcmKmrsfGJloYSxEavtM1O1VK8qtt1ITxGFRxoi2frbYzKjM0ELjKkE
0Qsux6e/ugDvnBtx85/00x+zECTF4jTU4u75oU+pXgAKDHkHQvn/SAeTaDMR2iKU
W1KJXpL98HqBBmaKzXGpXXt0WNKG2fnNs9+xOqzC1TkyUTTNOG81N0fkosHCBmdx
VY8Us1p+BjRKQ3DYIEHi3e0ktMCKSRh59s0J3rOpyAPeL8xtQF1SszjCSBociz/8H
OOECaDJ9Ryrhkd9E8tloeTWF8PD1VMsGq11F/eWPSGnDvKL0fvHPmq5nA5KMb9i0
4wgrRigvIn4yadhughQigM+wveRj7EpcXzaGeMusjc5Gzfau78VguIoIVPnBINwk
cYAm4hLR4SjksWjKctCREwCB5HhYmrc11adob7AKLSfVbGEGW+wjcoByHSQTLeyX
pjsImxrygb5WpczagBwIEt6AYk6kgWmsPtHF1FYtCHjdfv1Lr10zgvPuEHR0M9gA
0kWUnfSEdckaLw+c+Yade2q2Nct52wq4c4hcAvhJnJP0x25HWG0D0soCp74zx5jz
DuUvv6q77RFZtd/+ykYLYXhhMysKNq7d+3jUuQ1I3LStZ0K1xxeHsKN515AGNK7V
3HT6LAo1W4oOUObh/+bZRM6fPNMLsoTC+WHAiB4rOTUel jz7PEqTvpesDsbHbpORn
Oh5UKUuwTEH6FmfFUCMSlbeqyJoSghsa1F5cceJKRzKVR/ujYRcLJPoxRTVEWUhG
agsyQ5893TjS1xMFyvb2ZFq+11JdL/NU3980iWGWyg6FCck/UndwbV+DvRQ2pfgK
s1e97pSnL3w0JjMXpxs5WLWsf9wy2eTajsVMA9RWaHKXKpcIgtMIC6M8q4jWxie7
i4ZfrIVAHTbKaDaL1bGn7Y6nL2aWj1pLke9kE/gngZpKWEiAuG+MjY271AbNZB6x
zJL18Btd4VuzhmYnJCPBZ9q+YGV1TVtgbKeq5c9/06T6QvkeZU1QHUwYk jXvZAE
LZbGGzDcXVUWoBbP1fbzpzpWKuhzqrN8Jvof5e1SBuKe8nnQFUAKiHxz2shWkQvG7
gPkhDJdcWXErpohhnmEzE/deIGWRp2Kmh27/FW1FfKbF4s/UiYI2za6jNRmCSF8
FoTtHwOU23YdKfSgq+qx6Cr464wV1V7jUgBIfdMdhk9qx+1Nb3vnBpYnhg2tVHkb
aCUfOQxHN7FHySdTMunZSJ4DLHpS+e4ufY6jEmUwdgz/j+qTTDon9mwH3liyisj
o1nd0vA1ftBh4qfnjv5PQJ+C9vYhHU20V/uJt78jGdFP27qN31ZPj1Vq1/gPT/r6
BJzPXJx5FUqwhEkMUE6B5hd519hNNrrA1tvS1jugJGsoGUwbw1qffe29nGxTJKch
+pMJUOXRUDe105a75M9ui4r2nFb5yUhJh/KwBxTgfsuzQ2kZVksV1GRWzFOKeV/U
SoAJXR7mmxpKqcf002XdQYQq1k071iIGqxTQefTGNIsV+VSCK9VTjbd1RHBOoft2
lxW0GyLejwtfvRuFBozL268ZfyUI0xfqVRm/mjT27zBNoBDVsF3K8AGvuJRCIoVe
Pw2aki fn2+n8w9n3EDNkck4JDxkL9RQBULMYkxAcUwfxdXzPT/ixNHIEqj7VCu6k
qTLP r7Yt7qLe/BbndIs8u/rDc5SVWmdjzX3s0po7uw7XiII3ZvxWVmBhi65rJUzD
bx1pzA1+lRKGcnCetEPpnZuirmb33CKBrzLNxH5XTE1UfLr4g+kEwNFJL/ZvIwct
VzxwIGkeWfrkpdR27chlbGwXyZGeqhR7SgYoev9wvj55VKfGajWsb09Sw613FgS
lQ9fmgKv536pYlSYClfFSshuQiB0FVDlagtnb45FNGA2HaNtZuT+IWfwBpj308zI
fEGrm/NzhFFGnB/R8xqX3pB4nEQgbZ09Kbw7Bvd7XQ+2v6zQjHy82TP2Q2+vnDjN
fwwwJJ2mzT9QPcTjUu84RAT9ritBjh9QqU/pskeJJ+LW9s37uCEXICMnbaMT0btG
h8JBuYpxJ92M9511NSgv9pnex7PfuTdaq6CEyqnN2K5XFZQ9kVWYABucxCd++s05
uLOtbepB3MRJopgKpMYThCHAqd/Mcc+J3oO+Jw/g/zTw1q2tXiNw/smN1tsRO47x
ec/I9fK9VkxzXa34HQ6uhjhbzw+pzNUimlCCr/ZrAGGyUx+GqiAZjUWXuRtETFp
iYUpzM4+0Dnv5ziQNTWizAFWUTW++FmHpU8Nza4zqiVUKuxsFQC0I4zR9f1C6Ch
2oqRkuFIa203tff76D7h+BwoBK1Nm0yWjZFDpb71fbckQJZUQ9CbdpLdzY1SW1jyD
r09sRMtZeW0rE54k0XMyZ08MIUbvor1Xii f94QdhtWMFz2ne6rjBfgh4YF89QDtA
zjRBS0UeHXzv2N5LnYLaArADFGbhm4bhZVmgdQeHiPW5EaUF9PbaiWxs1E2dz710
DIgZaaz5ij3mWgRdu2uqBio7Abibsfhd0D3ImyEoB1AwisV3x8ucrTLjlm0lt4f
UXltfF7hdqRnKrtgQFe94pruaA8aWD8hMhLyycnOWhpmBHbWEAe2KTh3xC3XpVbr

V8IQdjSxY0AY1n5ktoAZG26Uoi1V34I6oImCyTTLKqbJv0KaO69Qszj3shoIjbtF
k7WeKn8xgSuqjl5t+n/6F+p10e4Tszh08+d2F11aBY27gGzxf7HGBbXY60hBZxg
vvC4DtZj3iYmzfZXRgbhgJLAjvtXRftxs90kHlHAFxlAgnrJWmUeVfzVif2/d10
VmJw/yg/u/d+HhFD18XXR8YRUHjCAsnews9F2I6B/abUpWELATTnL+SPYxdF0kC
Ip/+ziCiOZ6uiwNwiecu+VjzrZ0iGVARGHHHZOjTx1PlOcIsryOPOrJ6vGMiUsyr
cS9GYERRszavcaAQqYv/SF8Zi9VcuJA3ymyIHT1MaAghJYYzVcrr7NHWrU6+qf/S
zL3zJj3OGlUftX70tN41cJG9THfciWKILFgn5AdKiqOhqR2r0WffWY4E3/A2tKBe
AESRwu3p0K2UuCniE7UAg2P8C9jS/OdKQ7FepdUEwSCRJxb+jmm9o33NLvnkTItw
4jsHHjDfF4HxVx/vouoJ37SQqArYThgLcaEWHRRrNtmx/vPtYf+MrYviKgdCDgncs
ocBKICb0Uzi0NYNjnMp3j3rr69jzfxOH14WsmJlM3ANsyopuI9c5NeXEZiB1Sne
GAXPbcP1XERxd3HJ5gOB8+D7amyejIvJgqUpQIPBBYCYLFSIHukonEUt+Bj4HcfN
lBctOKTFVaEZsJhPpywdqKmwUuPn6Y4IVoEeQnxP3cSkk5vhgwZq+pfVvk6CMPnYx
ihmcuEiuZddzFL9IqaqJ57qni6yduEbo7AqGbaSDE6ISXtMvwjQHxnbWEAMbnQSU
BhmIdJA0BYy+GzjeKDX2SF/wejnmucBvVGBVPDyZ8bhj0ZH1jSBRvoeqxCnP7JkT
K3SFIWvTx4iulzpuqxyfQNIWFazRQRyrQqmVklz/u00t1mlrozRKPVDhFA6CwN15
djca+pBv7qMXDPSjNwgZLm4mhlwpuQM1m0frNdWjLjvo5X4k4B2SCLp2eRYLw/24
hi4Q0gs3yNSbV3VODnCj+VIpLFnwoOD1QyOH2GrEnREjJKSjqzCGbgBkXcvP03oE
dSioL/OvppL4c5FbQY135rQ6YtN8Ibww4QgCt3BEgPjUL820Pod0u/Fs5nOmOd0Y
/TAPLSUASRN0x3huZXPvPws4wHXtymYobUeiTz709iJGN1htySDhq6hHNBbzoIdh
OBSI7/j1UwOFLE6gAGIkxqxBRCKur/xUEia5MLFwsIDkd+MiAqRdtYHLZuVx4J5K
SgF08VucGPJNSkxMWpx3OM65CBMc9t7HR2EaMD230L5iF/maNyMH5X53OHib1Zg4
y8PaUdC1k6eoJc5qVzDf7a6xtuSr2d1R5gymyzG/22dLIpIL7o0jwcfraZrMouI
LoDtYkWxf8gHHMD4AmsrXY61PBECvrvI/s4CQ1Mvr8pChdtQJcuSH+lvuGUqqtFO
KnpdtecpSIALh0Eemdhet53LcpT2EUVY7Ns6N7PMHCgtQHOTPLJMkKRw4c3FWxPH
230C19w3+Wwvnnv+EDp6Eqza5QahCU7YeylteE2EY+1jaOfqe+jleTysem1lwz46U
wOSOM6X1zJhwNR1vqag7Ld4ZgtAUFjQjazR+Ko2IK9Lx6x+gxXkRDBtsdtUrdnLA
e0SVE8JdYQdJ55i7xhh46npC5ld5xX7igmlWFwPwj6V/5RoTKNYCdYo8UXX4NJ7B
yLYfK5yHF9KnLd9dTbxUuvOKYvdvKzgasfDhCd+SFwxLLR01JM8yDxmyy4rZEUwt
f+Q9DTtlbINMcIowXtJCi7afhZQRsEnDy1bzuaci72Dor6d266tnmDNTIQdLZ0j1
AivVD66/kTLb6Pp09BzZRY9x9P6SBHZ5RI50uyVJjSrm1SFjAKxxH/Kqkps06b6f
RBaSy1Jj/oBOFqgEehDZtyhFSKAftkd3qrfn9YhObhPltDwgOrUtSXRSpazqSzcg
ks/zcFjd9e51wPH4mPEOrRZuRjzWwrC2G8iZtAsVR8z3Ns2AWxoSDRSbE8IWxJYo
u9DbnvJV4Ri39N0u1cfadWiNePn22TMT5bszIrcqA1XiAMobfKok1xmAgPWlnAK
AaGhXgvumPCYp6+hNItX/PGId011iXyURVW9Jq/q9CotmaRM1j4q3JoHuleARjjW
Uf/jgzmcEFBYwftJJ6BJQtqhJ+HiBCVmJ1aFKNAXYcSfwBLaamN6SCQ8hXBuITe
TDqnbMo98r7amvNaIliwXtgYtz+FkfrZOWjgBDVJfrELmeoXbM8IoJ/zvnqUW8Yl
cMQjkHetmeIqGU2Ay9GduVQW7xV9Gc7kke7Sipnm/dQTL62rkPpA0qG17t5cPsBW
FUSCjBJR2RS1L0UcgZ1z1X6peuCN7XZwA2AvPPaZ8u2IWEqhyneOyms/4Zp5cr1L
ZfyceWokZ33zSGU7D8OPIXDkEcMas/a0hP7zYh+zQr7yazyxMOpnc16MNPJ4Ekeh
Dp1f6Rr/at8JRadz08iJujlWmcbdyCagg6v19gS1OmD5v7gcScZH0AOzYcYpntz
f36dd3VZfDT2heEkp+dmlNo5jiP/ZxANGy1qU+Dcq5vp/6KyHn1QZBMHw9KEfIAw
H04zUBXDBtiWiSx6UqW5bHR+nhKaB4oHpvnpGFekQZ01+5v/UbkAwJpEd3nPa96M
Xgt1oX0WR105AYfge1OzJo64KdryoImNNAqW2gOzn9b1HOeltkiNIwFdIU9gGHH
HdT7F3M60oInXO7X7b2Vw7y/7Ze9pWtnACP5k75EXXMGd9401c1pR99OX805kwdg
yFc6ZKvqEK/5rHRHwL12RfugI6Z43aY5nVtTQpJCUgw6HS4PzAEbNrHAQ1Ed+BZn
tGXvbtfo9ps115AO2HRS2YzdlrcQJqP5wD9gyT1hIzoTn6Z7eyIzYXGgte2GChFa
iC6V3SgPAPi6XheH50GBjllKFjPofRYiNJsQdJF8Oy/Ywo6ile8sByRx9jiASUzi
QSDxdMqt3m9ATbZQ3JoEGGuUohA5Wwn7ZhUDK1sfxp61h/1D2npjss98hYubDgck

a3jYmlyR9oh8KVlpSQ9ebaz2XXqmU2Egn9IOHQdQJ0wwqD7K5yneQ04/a1v3/0zG
jaliEfBgS81Dj4+iuucJUqTtS50K3H88zr11s1vr+KtFA0k8TESWk9ncDc2Uo+0w
jLIumCcDXzk+ZiUbd7bAdTYoCBKaPPj2RamY5K3/CYxRGdhuEra38Uyfk6S7Tjyr
UXvfEFZZVdP3UFv000Pw/p+iXnJusPZ7vZw7Zg5SCnO+RXtVnq18OS/HP9LbvX8g
3jgjABxluBtH2HmWyLiNhxZdG/OtgRzVYnBExVafqaBRtP7qNxI18u36U2p9IFn+
99UNmluZoup+yqVGzMDH7KUSTf36Oz9QpEghKwyohmK6u6s9FO3zHNVCKg2rvIOG
6iY8ro2q/KC4ioShoU+KM8DyBzAe8t8Yz/c06ipWlae+cMsBgulhqF7oAyyRJUX4
LMX1DAILi2FzmA2Cu347axP3woiquwG9GYiC+a3tfgzsnvVBay76JBPPUH2myy2L
1mxv1xewOjE+VRfBMGo6bPouWNqf1QGnDhWLwKyNzIAI7AiL/BHK7xhT4Be7+xWH
7P/Pd+9OZbYC4heifbXg/y+wYHBLVENsM9sM7qChuJSACuWQkNBBHJUQC1IzeGQb
Z1OdcjBQE+JNyJO8mo4cNhfIWlmJNH510jHRAzVO2qerF80uchQF7xWGV3qKg8P8
x5MAQDTiTiqKFGOHj5onM3Z6rbmRSRdbn6CJul65GGJjx6EnfXlpMG7IlgCFHv1U
CVlTnop5onytADFQih9LmjNvpHxonEOQ8wuEN9CiKEvFo/klEdiI/qRQhEV+KrX7
j/zsGEYfjMMbY6Uk40cPpZ70CwS4P7coHdTJQIX26inNN26UvQR9u48mhA0/ezuD
ttm0IHs7uK9IHOM1MBjSmEJxbDEvwND4srbj1Q0cv84bSPX3HHR0HGkwtPE4zqNq
Iw6eOpYUsJdDnyToq3A8Q+omzoz30YUzeBBRVvbf/Mwrd0Ci8+QcT9DbF4qUkVYT
xwGPQTnoLt+5DDPsfLESLblgXyxkYFavbnSlvNuAF1/AzD7C2T9GRvK7x7pleNrA
mwstYUVDPAL83egLxxqKDYeS7IPFZal3MJXO+/L8fr5zm+ZLh/fDFcHSTdkW/Mnh
pZfTjjc9NL701W2bpKUA VatptOqqsDNgX8lmXd2qetYTvVdc0rHrxz6moG8qtb2+
tzbi888edf615de8UTF9u4rTgN82IACEZC/78eeaiVOjOgUaQi/qY2yxtjFPOCZB
15Vwe/KkUMonf4btX1MAU0hSr83gQbhZR0ikKc9R42Mwuc0Ori3mWafVmJN/rB+E
hoF4756QzdkT7N93iGtOmeiccCu+nHZ6Mf/4wcoE2GzQ0w8LGMi2AxMxW5bBJTEA
/g5Eaug8JQ4dQ1srdw5Sn9CvaiyGOLvqiYMDj26YfPne75m29HmfFTgPi6xphEc0
Z/MCRP5kMXJuAm89d0KUZmXmRveNoudqmZ0VEXYZ086wn6u64Pj7RoN9N4gQYdZe
CZi33gShQfhpGVKMHK31Kc8tqB0I4PoPZF9QZu7pYa1Ki9VreFv4SA9X4182NEHM
sLOH1j+7Mr7k0zLXaFOL02X/uLUz+58aKeHo9TnH72j0Za71C7BoIcsVhdv1vHDz
+nw8bmeCHZA7mrThb5DUSG6J8TTDcAqAHxwD3R+vocAJGNDtE/6FvPHIUmLXOkY
Y+HPzvJhx4hN3plTXfLeB7ERgBsAQnnJYcZ/91sNNsC91ubbyC6X7Eu//V102nvv
Qo4M77evEo+ZW9vxyVxF+GjEuCeisCGztXKFFBhb3Z4XNNnClGP03GbAWAdnyI4T
T09QA7A0qwK5t4BtS57fuE8VgTEE2d29JmXM2J0vYqr1Bu7VWVvK8RjiejWi6g64
pA1NjrfACyitfbibkU51shu7pqrNKOrjiwewADLyUH/8s+HoPJCFellNqialOvMN
5Zy2nYs71GFw+Be6iNvLBeF2vvVhbnhRMbPCwMuQteJp3Vk1u98n78rVY0Q+G2wy
xGoJ5j020LckboH8IBIsp0t19Cb28x8AFTQnwWnXpjtmNAWwb9bakf+XvpLPkTlQ
/31+cHHBVIWzPBpbq8am8Ct2Ha1SRcOV3gF1U9jg3Us1pYdX7p0gqaQRgJoumcCu
/3tE8jye4VDUyWmCiIsO8mnyFGNq7qBb/Iq4AXegXMHTN/loDVWq1KaPoq2t23X
1UWly0KzV68q7jYQSyJCSAbhXl/K/lyY6YiRPukCu3c0jE66SFuVFeVbEPqsNuvU
cgTWLyDibMP3dzP1YTjVtjdsxs9kMoJcKyRG6uPVuD502Q/zrF+tb14Fu8tBscjM
q4xDg50fcXVH1HAZDDqaPYJEANRVVAEfi0apnrHC71W/Wit1gCGKyHtwpXNyGZqi
gTdtDQMIOTKXYcbA4qzaFRCXHAisVVALhzznSlcGPwKZuIKOR3FprlCqjBENzOwJ
959ySW84J3qoincGAl+gEJhXzCoRmb74+J0XwQxGJNz2EdPaQ9zn7fzS6EaBvioN
imKS94YwzD0bw4viUNxv+v9++hs/3Q5UL/TBrCTtaoUpzdkGGR/zoemj0S8LYLO2
6J17+U2N3i/Wcnpm8Y47LupdvbL+zddh8WQkmdJ7X8shVfHsUzSLxvYWnIQzdETY
+7xxzAY+W2309MSTJhGHR+xOcle/FB013ifpZo5qFRNasTWVLuPBZkwF3eFrSjCH
bnGre4WFFWLrOYR3Vfs1ZxczYJinI93N59nQUdN0FSTuoCT5ioIS2GQk1WoAbzRL
/7erGVX40mppmzB/tQ9wxXQoKZdWUyAJMRk1wV4XhnpUJScxJE+2HtBkaUi6I4/G
5wUs4i/cHAfrWksJOSII9zKx1EimwOGc1WcntB2+UCCb7cTJ2I5V6qmhAFK2ReX+
0Bcm8j8gmRjTeeKFon5Pp07CR/8FMr0X39D7VQmpc6t8hyA8xPhWwiRDdLwibMtj
7ZSNtVfiNMBofj+7k/INPNSe75DIuGaO+yAhizYYIJAF+HqObyMv+eBImiM3A6IT

C.1.1. Unprotected message

The resulting message would look something like this if it was sent without cryptographic protections:

```
Date: Wed, 11 Jan 2023 16:08:43 -0500
From: Bob <bob@example.net>
To: Alice <alice@example.net>
Subject: Handling the Jones contract
Message-ID: <20230111T210843Z.1234@lhp.example>
Content-Type: text/plain; charset="us-ascii"
MIME-Version: 1.0
```

Please review and approve or decline by Thursday, it's critical!

Thanks,
Bob

--

Bob Gonzalez
ACME, Inc.

C.1.2. Encrypted with hcp_minimal and Legacy Display

Now consider the message to be generated if it is to be cryptographically signed and encrypted, using HCP `hcp_minimal`, and the `legacy` variable is set.

For each Header Field, Bob's MUA passes its name and value through `hcp_minimal`. This returns the same value for every Header Field, except that:

`hcp_minimal("Subject", "Handling the Jones contract")` yields "[...]".

C.1.2.1. Cryptographic Payload

The Cryptographic Payload that will be signed and then encrypted is very similar to the unprotected message in Appendix C.1.1. Note the addition of:

- * the `protected-headers="v1"` parameter for the Content-Type
- * the appropriate HP-Obscured header for Subject,
- * the `hp-legacy-display="1"` parameter for the Content-Type
- * the Legacy Display Element (the simple pseudo-header and its trailing newline) in the Main Body Part.

```
Date: Wed, 11 Jan 2023 16:08:43 -0500
From: Bob <bob@example.net>
To: Alice <alice@example.net>
Subject: Handling the Jones contract
Message-ID: <20230111T210843Z.1234@lhp.example>
Content-Type: text/plain; charset="us-ascii"; hp-legacy-display="1";
  protected-headers="v1"
MIME-Version: 1.0
HP-Obscured: Subject: [...]
```

Subject: Handling the Jones contract

Please review and approve or decline by Thursday, it's critical!

Thanks,
Bob

--

Bob Gonzalez
ACME, Inc.

C.1.2.2. External Header Section

The Cryptographic Payload from Appendix C.1.2.1 is then wrapped in the appropriate Cryptographic Layers. For this example, using S/MIME, it is wrapped in an `application/pkcs7-mime; smime-type="signed-data"` layer, which is in turn wrapped in a `application/pkcs7-mime; smime-type="enveloped-data"` layer.

Then an external Header Section is applied to the outer MIME object, which looks like this:

```
Date: Wed, 11 Jan 2023 16:08:43 -0500
From: Bob <bob@example.net>
To: Alice <alice@example.net>
Subject: [...]
```

Message-ID: <20230111T210843Z.1234@lhp.example>
Content-Transfer-Encoding: base64
Content-Type: application/pkcs7-mime; name="smime.p7m";
 smime-type="enveloped-data"
MIME-Version: 1.0

Note that the Subject Header Field has been obscured appropriately by `hcp_minimal`. The output of the CMS enveloping operation is base64-encoded and forms the body of the message.

C.2. Composing a Reply

Next we consider a typical MUA reply interface, where we see Alice replying to Bob's message from Appendix C.1.

When Alice clicks "Reply" to Bob's signed-and-encrypted message with Header Protection, she might see something like this:

```

-----
Replying to Bob ("Handling the Jones Contract") .----.
      +-----+ | Send |
To:   | Bob <bob@example.net> | '-----'
      +-----+-----+
Subject: | Re: Handling the Jones contract |
      +-----+-----+
-----
On Wed, 11 Jan 2023 16:08:43 -0500, Bob wrote:

> Please review and approve or decline by Thursday,
> it's critical!
>
> Thanks,
> Bob
>
> --
> Bob Gonzalez
> ACME, Inc.

--
Alice Jenkins
ACME, Inc.
-----

```

Figure 2: Example Message Reply Interface (unedited)

Note that because Alice's MUA is aware of Header Protection, it knows what the correct Subject header is, even though it was obscured. It also knows to avoid including the Legacy Display Element in the quoted/attributed text that it includes in the draft reply.

Once Alice has edited the reply message, it might look something like this:

```

.------.
|   Replying to Bob ("Handling the Jones Contract")   | .-----.
|   +-----+ | Send |
|   To: | Bob <bob@example.net> | '-----'
|   +-----+ |
|   Subject: | Re: Handling the Jones contract |
|   +-----+ |
+-----+
|
| On Wed, 11 Jan 2023 16:08:43 -0500, Bob wrote:
|
| > Please review and approve or decline by Thursday,
| > it's critical!
|
| I'll get right on it, Bob!
|
| Regards,
| Alice
|
| --
| Alice Jenkins
| ACME, Inc.
|
+-----+

```

Figure 3: Example Message Reply Interface (edited)

When Alice clicks "Send", the MUA generates values for Message-ID, From, and Date Header Fields, populates the In-Reply-To, and References Header Fields, and also converts the reply body into the appropriate format.

C.2.1. Unprotected message

The resulting message would look something like this if it were to be sent without any cryptographic protections:

Date: Wed, 11 Jan 2023 16:48:22 -0500
From: Alice <alice@example.net>
To: Bob <bob@example.net>
Subject: Re: Handling the Jones contract
Message-ID: <20230111T214822Z.5678@lhp.example>
In-Reply-To: <20230111T210843Z.1234@lhp.example>
References: <20230111T210843Z.1234@lhp.example>
Content-Type: text/plain; charset="us-ascii"
MIME-Version: 1.0

On Wed, 11 Jan 2023 16:08:43 -0500, Bob wrote:

> Please review and approve or decline by Thursday,
> it's critical!

I'll get right on it, Bob!

Regards,
Alice

--
Alice Jenkins
ACME, Inc.

Of course, this would leak not only the contents of Alice's message, but also the contents of Bob's initial message, as well as the Subject Header Field! So Alice's MUA won't do that; it is going to create a signed-and-encrypted message to submit to the network.

C.2.2. Encrypted with hcp_null and Legacy Display

This example assumes that Alice's MUA uses hcp_null, not hcp_minimal. That is, by default, it does not obscure or remove any Header Fields, even when encrypting.

However, it follows the guidance in Section 2.5.8.1, and will make use of the HP-Obscured field in the Cryptographic Payload of Bob's original message (Appendix C.1.2.1) to determine what to obscure.

When crafting the Cryptographic Payload, its baseline HCP (hcp_null) leaves each field untouched. But it also knows that In-Reply-To, References, To, and Subject are all derived from Header Fields in Bob's original message.

For each of these Header Fields, it observes whether the origin Header Field was signed-and-encrypted or merely signed in Bob's original message.

In-Reply-To and References derive from Bob's original message's Message-ID field, which was merely signed. The To Header Field is derived from Bob's original message's From field, which was also merely signed. So these three Header Fields are passed through untouched.

But the Subject Header Field is derived from Bob's original message's Subject field (by prefixing Re: to it), and that Header Field is signed-and-encrypted, which the MUA can tell because the HP-Obscured: Subject entry in the Cryptographic Payload of Bob's message.

So Alice's MUA generates a new external Subject header by applying its derivation rules to the HP-Obscured: Subject value from Bob's message, yielding the value Re: [...].

C.2.2.1. Cryptographic Payload

Consequently, the Cryptographic Payload for Alice's reply looks like this:

```
Date: Wed, 11 Jan 2023 16:48:22 -0500
From: Alice <alice@example.net>
To: Bob <bob@example.net>
Subject: Re: Handling the Jones contract
Message-ID: <20230111T214822Z.5678@lhp.example>
In-Reply-To: <20230111T210843Z.1234@lhp.example>
References: <20230111T210843Z.1234@lhp.example>
Content-Type: text/plain; charset="us-ascii"; hp-legacy-display="1";
  protected-headers="v1"
MIME-Version: 1.0
HP-Obscured: Subject: Re: [...]
```

Subject: Re: Handling the Jones contract

On Wed, 11 Jan 2023 16:08:43 -0500, Bob wrote:

```
> Please review and approve or decline by Thursday,
> it's critical!
```

I'll get right on it, Bob!

Regards,
Alice

--
Alice Jenkins
ACME, Inc.

Note the following features:

- * the `protected-header="v1"` parameter to `Content-Type`
- * the appropriate `HP-Obscured` header for `Subject`,
- * the `hp-legacy-display="1"` parameter for the `Content-Type`
- * the Legacy Display Element (the simple pseudo-header and its trailing newline) in the Main Body Part.

C.2.2.2. External Header Section

The Cryptographic Payload from Appendix C.2.2.1 is then wrapped in the appropriate Cryptographic Layers. For this example, using S/MIME, it is wrapped in an `application/pkcs7-mime; smime-type="signed-data"` layer, which is in turn wrapped in a `application/pkcs7-mime; smime-type="enveloped-data"` layer.

Then an external Header Section is applied to the outer MIME object, which looks like this:

```
Date: Wed, 11 Jan 2023 16:48:22 -0500
From: Alice <alice@example.net>
To: Bob <bob@example.net>
Subject: Re: [...]
Message-ID: <20230111T214822Z.5678@lhp.example>
In-Reply-To: <20230111T210843Z.1234@lhp.example>
References: <20230111T210843Z.1234@lhp.example>
Content-Transfer-Encoding: base64
Content-Type: application/pkcs7-mime; name="smime.p7m";
  smime-type="enveloped-data"
MIME-Version: 1.0
```

Note that the Subject Header Field has been obscured appropriately even though `hcp_null` would not have touched it by default. The output of the CMS enveloping operation is base64-encoded and forms the body of the message.

Appendix D. Rendering Examples

This section offers example Cryptographic Payloads (the content within the Cryptographic Envelope) that contain Legacy Display Elements.

D.1. Example text/plain Cryptographic Payload with Legacy Display Elements

Here is a simple one-part Cryptographic Payload (Header Section and body) of a message that includes Legacy Display Elements:

```
Date: Fri, 21 Jan 2022 20:40:48 -0500
From: Alice <alice@example.net>
To: Bob <bob@example.net>
Subject: Dinner plans
Message-ID: <text-plain-legacy-display@lhp.example>
MIME-Version: 1.0
Content-Type: text/plain; charset="us-ascii"; hp-legacy-display="1";
  protected-headers="v1"
```

Subject: Dinner plans

Let's meet at Rama's Roti Shop at 8pm and go to the park from there.

A compatible MUA will recognize the hp-legacy-display="1" parameter and render the body of the message as:

Let's meet at Rama's Roti Shop at 8pm and go to the park from there.

A legacy decryption-capable MUA that is unaware of this mechanism will ignore the hp-legacy-display="1" parameter and instead render the body including the Legacy Display Elements:

Subject: Dinner plans

Let's meet at Rama's Roti Shop at 8pm and go to the park from there.

D.2. Example text/html Cryptographic Payload with Legacy Display Elements

Here is a modern one-part Cryptographic Payload (Header Section and body) of a message that includes Legacy Display Elements:

Date: Fri, 21 Jan 2022 20:40:48 -0500
From: Alice <alice@example.net>
To: Bob <bob@example.net>
Subject: Dinner plans
Message-ID: <text-html-legacy-display@lhp.example>
MIME-Version: 1.0
Content-Type: text/html; charset="us-ascii"; hp-legacy-display="1";
protected-headers="v1"

```
<html><head><title></title></head><body>  
<div class="header-protection-legacy-display">  
<pre>Subject: Dinner plans</pre>  
</div>  
<p>  
Let's meet at Rama's Roti Shop at 8pm and go to the park  
from there.  
</p>  
</body>  
</html>
```

A compatible MUA will recognize the hp-legacy-display="1" parameter and mask out the Legacy Display div, rendering the body of the message as a simple paragraph:

Let's meet at Rama's Roti Shop at 8pm and go to the park from there.

A legacy decryption-capable MUA that is unaware of this mechanism will ignore the hp-legacy-display="1" parameter and instead render the body including the Legacy Display Elements:

Subject: Dinner plans

Let's meet at Rama's Roti Shop at 8pm and go to the park from there.

Appendix E. Other Header Protection Schemes

Other Header Protection schemes have been proposed in the past. However, those typically have drawbacks such as sparse implementation, known problems with legacy interoperability (in particular with rendering), lack of clear signalling of sender intent, and/or incomplete cryptographic protections. This section lists such schemes known at the time of the publication of this document out of historical interest.

E.1. Original RFC 8551 Header Protection

S/MIME [RFC8551] (as well as its predecessors [RFC5751] and [RFC3851]) defined a form of cryptographic Header Protection that is similar to the "Wrapped Message" scheme specified in this document. In fact, the scheme originally defined in S/MIME is a subset of the "Wrapped Message" scheme specified in this document. The differences between the original and the updated scheme are outlined in Section 2.2.

E.2. Pretty Easy Privacy (pEp)

The pEp (pretty Easy privacy) [I-D.pep-general] project specifies two different MIME schemes that include Header Protection for Signed-and-Encrypted e-mail messages in [I-D.pep-email]: One scheme -- referred as pEp Email Format 1 (PEF-1) -- is generated towards MUAs not known to be pEp-capable, while the other scheme -- referred as PEF-2 -- is used between MUAs discovered to be compatible with pEp. Signed-only messages are not recommended in pEp.

E.3. "draft-autocrypt" Protected Headers

[I-D.autocrypt-lamps-protected-headers] describes a scheme similar to the "Injected Headers" scheme specified in this document. However, instead of adding Legacy Display Elements to existing MIME parts (cf. Section 2.3.4.1), "draft-autocrypt" injects a new MIME element "Legacy Display Part", thus modifying the MIME structure of the Cryptographic Payload.

Appendix F. Document Changelog

[[RFC Editor: This section is to be removed before publication]]

- * draft-ietf-lamps-header-protection-20
 - clarify IANA guidance about registration policy and designated expert review
 - emphasize that Content-Type parameter hp-legacy-display=1 belongs on all main body parts with a legacy display element
 - clean up/normalize pseudocode variable names and text (no algorithm changes)
- * draft-ietf-lamps-header-protection-19
 - improve text, capitalize defined terms, fix typos

- Clean up from AD review:
 - updates RFC 8551 explicitly
 - add "Legacy Signed Message" and "Ordinary User" explicitly to terms
 - tighten up SHOULDs/MUSTs for conformant MUAs
 - expand references to other relevant Security Considerations
 - drop nudge about non-existent Content-Type Parameters registry
 - clarify IANA notes to align with table columns
 - explicitly request HCP registry
 - add references to other header protections schemes, but move all of them to appendix
- * draft-ietf-lamps-header-protection-18
- only allow US-ASCII as modified output of HCP, adjusted ABNF to match
- * draft-ietf-lamps-header-protection-17
- More edits from WGLC:
 - clean up definition of "Header Field"
 - note leakage of encrypted recipient hints
 - clarify explanation of LDE generation
 - clarify how some obscured headers might not actually be private
- * draft-ietf-lamps-header-protection-16
- correct variable names in message composition algorithms
 - make text more readable
- * draft-ietf-lamps-header-protection-15
- include clarifications, typos, etc from comments received during WGLC

- * draft-ietf-lamps-header-protection-14
 - provide section references for draft-ietf-lamps-e2e-mail-guidance
 - encourage a future IANA named HCP registry if HCP development takes off
- * draft-ietf-lamps-header-protection-13
 - Retitle from "Header Protection for S/MIME" to "Header Protection for Cryptographically Protected E-mail"
- * draft-ietf-lamps-header-protection-12
 - MUST produce HP-Obscured and HP-Removed when generating encrypted messages with non-null HCP
 - Wrapped Message: move from forwarded=no to protected-headers=wrapped
 - Wrapped Message: recommend Content-Disposition: inline
- * draft-ietf-lamps-header-protection-11
 - Remove most of the Bcc text (transferred general discussion to e2e-mail-guidance)
 - Fix bug in algorithm for generating HP-Obscured and HP-Removed
 - More detail about handling Reply messages
 - Considerations around handling risky Legacy Display Elements
 - Narrative descriptions of some worked examples
 - Describe potential leaks to recipients
 - Clarify debugging/troubleshooting UX affordances
- * draft-ietf-lamps-header-protection-10
 - Clarify that HCP doesn't apply to Structural Header Fields
 - Drop out-of-date "Open Issues" section
 - Brief commentary on UI of messages with intermediate/mixed protections

- Deprecation prospects for messages without protected headers
- Describe generating replies to encrypted messages with stronger HCP
- * draft-ietf-lamps-header-protection-09
 - clarify terminology
 - add privacy and security considerations
 - clarify HCP examples and baselines
 - recommend hcp_minimal as default HCP
 - add HP-Obscured and HP-Removed (avoids reasoning about differences between outside and inside the Cryptographic Envelope)
 - regenerated test vectors
- * draft-ietf-lamps-header-protection-08
 - MUST compose injected headers, MAY compose wrapped messages
 - MUST parse both schemes
 - cleanup and restructure document
- * draft-ietf-lamps-header-protection-07
 - move from legacy display MIME part to legacy display elements within main body part
- * draft-ietf-lamps-header-protection-06
 - document observed problems with legacy MUAs
 - avoid duplicated outer Message-IDs in hcp_strong test vectors
- * draft-ietf-lamps-header-protection-05
 - fix multipart/signed wrapped test vectors
- * draft-ietf-lamps-header-protection-04
 - add test vectors

- add "problems with Injected Messages" subsection
- * draft-ietf-lamps-header-protection-03
 - dkg takes over from Bernie as primary author
 - Add Usability section
 - describe two distinct formats "Wrapped Message" and "Injected Headers"
 - Introduce Header Confidentiality Policy model
 - Overhaul message composition guidance
 - Simplify document creation workflow, move public face to gitlab
- * draft-ietf-lamps-header-protection-02
 - editorial changes / improve language
- * draft-ietf-lamps-header-protection-01
 - Add DKG as co-author
 - Partial Rewrite of Abstract and Introduction [HB/AM/DKG]
 - Adding definitions for Cryptographic Layer, Cryptographic Payload, and Cryptographic Envelope (reference to [I-D.ietf-lamps-e2e-mail-guidance]) [DKG]
 - Enhanced MITM Definition to include Machine- / Meddler-in-the-middle [HB]
 - Relaxed definition of Original message, which may not be of type "message/rfc822" [HB]
 - Move "memory hole" option to the Appendix (on request by Chair to only maintain one option in the specification) [HB]
 - Updated Scope of Protection Levels according to WG discussion during IETF-108 [HB]
 - Obfuscation recommendation only for Subject and Message-Id and distinguish between Encrypted and Unencrypted Messages [HB]
 - Removed (commented out) Header Field Flow Figure (it appeared to be confusing as is was) [HB]

- * draft-ietf-lamps-header-protection-00
 - Initial version (text partially taken over from [I-D.ietf-lamps-header-protection-requirements])

Authors' Addresses

Daniel Kahn Gillmor
American Civil Liberties Union
125 Broad St.
New York, NY, 10004
United States of America
Email: dkg@fifthhorseman.net

Bernie Hoeneisen
pEp Project
Oberer Graben 4
CH- 8400 Winterthur
Switzerland
Email: bernie.hoeneisen@pep-project.org
URI: <https://pep-project.org/>

Alexey Melnikov
Isode Ltd
14 Castle Mews
Hampton, Middlesex
TW12 2NP
United Kingdom
Email: alexey.melnikov@isode.com

LAMPS Working Group
Internet-Draft
Intended status: Standards Track
Expires: 21 August 2023

H. Brockhaus
D. von Oheimb
S. Fries
Siemens
17 February 2023

Lightweight Certificate Management Protocol (CMP) Profile
draft-ietf-lamps-lightweight-cmp-profile-21

Abstract

This document aims at simple, interoperable, and automated PKI management operations covering typical use cases of industrial and IoT scenarios. This is achieved by profiling the Certificate Management Protocol (CMP), the related Certificate Request Message Format (CRMF), and HTTP-based or CoAP-based transfer in a succinct but sufficiently detailed and self-contained way. To make secure certificate management for simple scenarios and constrained devices as lightweight as possible, only the most crucial types of operations and options are specified as mandatory. More specialized or complex use cases are supported with optional features.

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 21 August 2023.

Copyright Notice

Copyright (c) 2023 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document.

Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1.	Introduction	3
1.1.	How to Read This Document	5
1.2.	Conventions and Terminology	5
1.3.	Motivation for a Lightweight Profile of CMP	6
1.4.	Special Requirements of Industrial and IoT Scenarios	8
1.5.	Existing CMP Profiles	9
1.6.	Compatibility with Existing CMP Profiles	9
1.7.	Use of CMP in SZTP and BRSKI Environments	11
1.8.	Scope of this Document	11
1.9.	Structure of this Document	12
2.	Solution Architecture	13
3.	Generic Aspects of PKI Messages and PKI Management Operations	15
3.1.	General Description of the CMP Message Header	16
3.2.	General Description of the CMP Message Protection	18
3.3.	General Description of CMP Message ExtraCerts	19
3.4.	Generic PKI Management Operation Prerequisites	19
3.5.	Generic Validation of a PKI Message	21
3.6.	Error Handling	23
3.6.1.	Reporting Error Conditions Upstream	23
3.6.2.	Reporting Error Conditions Downstream	24
3.6.3.	Handling Error Conditions on Nested Messages Used for Batching	24
3.6.4.	PKIStatusInfo and Error Messages	25
4.	PKI Management Operations	27
4.1.	Enrolling End Entities	29
4.1.1.	Enrolling an End Entity to a New PKI	30
4.1.2.	Enrolling an End Entity to a Known PKI	37
4.1.3.	Updating a Valid Certificate	37
4.1.4.	Enrolling an End Entity Using a PKCS#10 Request	39
4.1.5.	Using MAC-Based Protection for Enrollment	41
4.1.6.	Adding Central Key Pair Generation to Enrollment	42
4.1.6.1.	Using Key Transport Key Management Technique	48
4.1.6.2.	Using Key Agreement Key Management Technique	48
4.1.6.3.	Using Password-Based Key Management Technique	50
4.2.	Revoking a Certificate	50
4.3.	Support Messages	53
4.3.1.	Get CA Certificates	54
4.3.2.	Get Root CA Certificate Update	55
4.3.3.	Get Certificate Request Template	57

- 4.3.4. CRL Update Retrieval 59
- 4.4. Handling Delayed Delivery 61
- 5. PKI Management Entity Operations 66
 - 5.1. Responding to Requests 66
 - 5.1.1. Responding to a Certificate Request 67
 - 5.1.2. Responding to a Confirmation Message 67
 - 5.1.3. Responding to a Revocation Request 68
 - 5.1.4. Responding to a Support Message 68
 - 5.1.5. Initiating Delayed Delivery 68
 - 5.2. Forwarding Messages 69
 - 5.2.1. Not Changing Protection 71
 - 5.2.2. Adding Protection and Batching of Messages 71
 - 5.2.2.1. Adding Protection to a Request Message 72
 - 5.2.2.2. Batching Messages 73
 - 5.2.3. Replacing Protection 75
 - 5.2.3.1. Not Changing Proof-of-Possession 76
 - 5.2.3.2. Using raVerified 76
 - 5.3. Acting on Behalf of other PKI Entities 77
 - 5.3.1. Requesting a Certificate 77
 - 5.3.2. Revoking a Certificate 78
- 6. CMP Message Transfer Mechanisms 78
 - 6.1. HTTP Transfer 79
 - 6.2. CoAP Transfer 82
 - 6.3. Piggybacking on Other Reliable Transfer 84
 - 6.4. Offline Transfer 84
 - 6.4.1. File-Based Transfer 85
 - 6.4.2. Other Asynchronous Transfer Protocols 85
- 7. Conformance Requirements 85
 - 7.1. PKI Management Operations 85
 - 7.2. Message Transfer 88
- 8. IANA Considerations 89
- 9. Security Considerations 91
- 10. Acknowledgements 91
- 11. References 91
 - 11.1. Normative References 91
 - 11.2. Informative References 93
- Appendix A. Example CertReqTemplate 96
- Appendix B. History of Changes 98
- Authors' Addresses 106

1. Introduction

[RFC Editor:

Please perform the following substitution.

* RFCXXXX --> the assigned numerical RFC value for this draft

- * RFCAAAA --> the assigned numerical RFC value for [I-D.ietf-lamps-cmp-updates]
- * RFCBBBB --> the assigned numerical RFC value for [I-D.ietf-lamps-cmp-algorithms]

Please also update the following references to associated drafts in progress to reflect their final RFC assignments, if available:

- * [I-D.ietf-lamps-cmp-updates]
- * [I-D.ietf-lamps-cmp-algorithms]
- * [I-D.ietf-ace-cmpv2-coap-transport]
- * [I-D.ietf-netconf-sztp-csr]
- * [I-D.ietf-anima-brski-ae]
- * [I-D.ietf-anima-brski-prm]

]

This document specifies PKI management operations supporting machine-to-machine and IoT use cases. Its focus is to maximize automation and interoperability between all involved PKI entities, ranging from end entities (EE) over any number of intermediate PKI management entities such as Registration Authorities (RA) to the CMP endpoints of Certification Authority (CA) systems. This profile makes use of the concepts and syntax specified in CMP [RFC4210], [I-D.ietf-lamps-cmp-updates], and [I-D.ietf-lamps-cmp-algorithms], CRMF [RFC4211] and [RFC9045], CMS [RFC5652] and [RFC8933], HTTP transfer for CMP [RFC6712], and CoAP transfer for CMP [I-D.ietf-ace-cmpv2-coap-transport]. CMP, CRMF and CMS are feature-rich specifications, but most application scenarios use only a limited subset of the same specified functionality. Additionally, the standards are not always precise enough on how to interpret and implement the described concepts. Therefore, this document aims to tailor the available options and specify how to use them in adequate detail to make the implementation of interoperable automated certificate management as straightforward and lightweight as possible.

Note: In the meantime RFC4210bis [I-D.ietf-lamps-rfc4210bis] and RFC6712bis [I-D.ietf-lamps-rfc6712bis] drafts were submitted incorporating the changes listed in CMP Updates [I-D.ietf-lamps-cmp-updates] into the original RFC text.

1.1. How to Read This Document

This document has become longer than the authors would have liked it to be. Yet apart from studying Section 3, which contains general requirements, the reader does not have to work through the whole document. The guidance in Sections 1.9 and 7 should be used to figure out which parts of Section 4 to Section 6 are relevant for the target certificate management solution depending on the PKI management operations, their variants, and types of message transfer needed.

Since conformity to this document can be achieved by implementing only the functionality declared mandatory in Section 7, the profile can still be called lightweight because in particular for end entities the mandatory-to-implement set of features is rather limited.

1.2. Conventions and Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

The key word "PROHIBITED" is to be interpreted to mean that the respective ASN.1 field SHALL NOT be present or used.

Technical terminology is used in conformance with RFC 4210 [RFC4210], RFC 4211 [RFC4211], RFC 5280 [RFC5280], and IEEE 802.1AR [IEEE.802.1AR_2018]. The following key words are used:

CA: Certification authority, which issues certificates.

RA: Registration authority, an optional PKI component to which a CA delegates certificate management functions such as end entity authentication and authorization checks for incoming requests. An RA can also provide conversion between various certificate management protocols and other protocols providing some operations related to certificate management.

LRA: Local registration authority, a specific form of RA with proximity to the end entities.

Note: For ease of reading, this document uses the term "RA" also for LRAs in all cases where the difference is not relevant.

- KGA:** Key generation authority, an optional system component, typically co-located with an RA or CA, that offers key generation services to end entities.
- EE:** End entity, typically a device or service that holds a public-private key pair for which it manages a public-key certificate. An identifier for the EE is given as the subject of its certificate.

The following terminology is reused from RFC 4210 [RFC4210], as follows:

- PKI management operation:** All CMP messages belonging to a single transaction. The transaction is identified by the transactionID field of the message headers.
- PKI management entity:** A non-EE PKI entity, i.e., RA or CA.
- PKI entity:** An EE or PKI management entity.

CMP messages are referred to by the names of PKIBody choices defined in RFC 4210 Section 5.1.2 [RFC4210] and are further described in Section 4 of this document.

The following terms are introduced in this document:

- CMP protection key:** The private key used to sign a CMP message.
- CMP protection certificate:** The certificate related to the CMP protection key. If the keyUsage extension is present, it MUST include digitalSignature.

1.3. Motivation for a Lightweight Profile of CMP

CMP was standardized in 1999 and is implemented in several PKI products. In 2005, a completely reworked and enhanced version 2 of CMP [RFC4210] and CRMF [RFC4211] has been published, followed by a document specifying a transfer mechanism for CMP messages using HTTP [RFC6712] in 2012.

CMP is a capable protocol and could be used more widely. RFC 4210 [RFC4210] and CMP Updates [I-D.ietf-lamps-cmp-updates] offer a very large set of features and options. On the one hand, this makes CMP applicable to a very wide range of scenarios, but on the other hand, a full implementation supporting all options is not realistic because this would take undue effort.

In order to reduce complexity, the set of mandatory PKI management operations and variants required by this specification has been kept lean. This limits development effort and minimizes resource needs, which is particularly important for memory-constrained devices. To this end, when there was design flexibility to either have necessary complexity on the EE or in the PKI management entity, this profile chose to include it in the PKI management entities where typically more computational resources are available. Additional recommended PKI management operations and variants support some more complex scenarios that are considered beneficial for environments with more specific demands or boundary conditions. The optional PKI management operations support less common scenarios and requirements.

Moreover, many details of the CMP protocol have been left open or have not been specified in full preciseness. The profiles specified in Appendix D and E of RFC 4210 [RFC4210] define some more detailed PKI management operations. Yet, the specific needs of highly automated scenarios for a machine-to-machine communication are not covered sufficiently.

Profiling is a way to reduce feature richness and complexity of standards to what is needed for specific use cases. 3GPP and UNISIG already use profiling of CMP as a way to cope with these challenges. To profile means to take advantage of the strengths of the given protocol, while explicitly narrowing down the options it provides to those needed for the purpose(s) at hand and eliminating all identified ambiguities. In this way the general aspects of the protocol are utilized and only the special requirements of the target scenarios need to be dealt with using distinct features the protocol offers.

Defining a profile for a new target environment takes high effort because the range of available options needs to be well understood and the selected options need to be consistent with each other and suitably cover the intended application scenario. Since most industrial PKI management use cases typically have much in common it is worth sharing this effort, which is the aim of this document. Other standardization bodies can reference this document and further tailor the PKI management operations to their needs to avoid coming up with individual profiles from scratch.

1.4. Special Requirements of Industrial and IoT Scenarios

The profiles specified in Appendix D and E of RFC 4210 [RFC4210] have been developed particularly for managing certificates of human end entities. With the evolution of distributed systems and client-server architectures, certificates for machines and applications on them have become widely used. This trend has strengthened even more in emerging industrial and IoT scenarios. CMP is sufficiently flexible to support them well.

Today's IT security architectures for industrial solutions typically use certificates for endpoint authentication within protocols like IPsec, TLS, or SSH. Therefore, the security of these architectures highly relies upon the security and availability of the implemented certificate management operations.

Due to increasing security and availability needs in operational technology, especially when used for critical infrastructures and systems with a high number of certificates, a state-of-the-art certificate management system must be constantly available and cost-efficient, which calls for high automation and reliability. Consequently, the NIST Framework for Improving Critical Infrastructure Cybersecurity [NIST.CSWP.04162018] refers to proper processes for issuance, management, verification, revocation, and audit for authorized devices, users, and processes involving identity and credential management. Such PKI management operations according to commonly accepted best practices are also required in IEC 62443-3-3 [IEC.62443-3-3] for security level 2 and higher.

Further challenges in many industrial systems are network segmentation and asynchronous communication. Also, PKI management entities like Certification Authorities (CA) typically are not deployed on-site but in a highly protected data center environment, e.g., operated according to ETSI Policy and security requirements for Trust Service Providers issuing certificates [ETSI-EN.319411-1]. Certificate management must be able to cope with such network architectures. CMP offers the required flexibility and functionality, namely authenticated self-contained messages, efficient polling, and support for asynchronous message transfer while retaining end-to-end authentication.

1.5. Existing CMP Profiles

As already stated, RFC 4210 [RFC4210] contains profiles with mandatory and optional PKI management operations in Appendix D and E. Those profiles focus on management of human user certificates and only partly address the specific needs of certificate management automation for unattended devices or machine-to-machine application scenarios.

Both Appendixes D and E focus on EE-to-RA/CA PKI management operations and do not address further profiling of RA-to-CA communication as typically needed for full backend automation. All requirements regarding algorithm support for RFC 4210 Appendix D and E [RFC4210] have been updated by CMP Algorithms Section 7.1 [I-D.ietf-lamps-cmp-algorithms].

3GPP makes use of CMP [RFC4210] in its Technical Specification 33.310 [ETSI-3GPP.33.310] for automatic management of IPsec certificates in 3G, LTE, and 5G backbone networks. Since 2010, a dedicated CMP profile for initial certificate enrollment and certificate update operations between EE and RA/CA is specified in that document.

UNISIG has included a CMP profile for enrollment of TLS certificates in the Subset-137 specifying the ETRAM/ETCS on-line key management for train control systems [UNISIG.Subset-137] in 2015.

Both standardization bodies tailor CMP [RFC4210], CRMF [RFC4211], and HTTP transfer for CMP [RFC6712] for highly automated and reliable PKI management operations for unattended devices and services.

1.6. Compatibility with Existing CMP Profiles

The profile specified in this document is compatible with RFC 4210 Appendixes D and E (PKI Management Message Profiles) [RFC4210], with the following exceptions:

- * signature-based protection is the default protection; an initial PKI management operation may also use MAC-based protection,
- * certification of a second key pair within the same PKI management operation is not supported,
- * proof-of-possession (POPO) with self-signature of the certTemplate according to RFC 4211 Section 4.1 [RFC4211] clause 3 is the recommended default POPO method (deviations are possible for EEs when requesting central key generation, for RAs when using raVerified, and if the newly generated keypair is technically not capable to generate digital signatures),

- * confirmation of newly enrolled certificates may be omitted, and
- * all PKI management operations consist of request-response message pairs originating at the EE, i.e., announcement messages (requiring a push model, a CMP server on the EE) are excluded in favor of a lightweight implementation on the EE.

The profile specified in this document is compatible with the CMP profile for 3G, LTE, and 5G network domain security and authentication framework [ETSI-3GPP.33.310], except that:

- * protection of initial PKI management operations may be MAC-based,
- * the subject field is mandatory in certificate templates, and
- * confirmation of newly enrolled certificates may be omitted.

The profile specified in this document is compatible with the CMP profile for on-line key management in rail networks as specified in UNISIG Subset-137 [UNISIG.Subset-137], except that:

- * A certificate enrollment request message consists of only one certificate request (CertReqMsg).
- * RFC 4210 [RFC4210] requires that the messageTime is Greenwich Mean Time coded as generalizedTime.

Note: As UNISIG Subset-137 Table 5 [UNISIG.Subset-137] explicitly states that the messageTime is required to be "UTC time", it is not clear if this means a coding as UTCTime or generalizedTime and if other time zones than Greenwich Mean Time shall be allowed. Both time formats are described in RFC 5280 Section 4.1.2.5 [RFC5280].

- * The same type of protection is required to be used for all messages of one PKI management operation. This means, in case the request message protection is MAC-based, also the response, certConf, and pkiConf messages must have a MAC-based protection.
- * Use of caPubs is not required but typically allowed in combination with MAC-based protected PKI management operations. On the other hand UNISIG Subset-137 Table 12 [UNISIG.Subset-137] requires using caPubs.

Note: It remains unclear from UNISIG Subset-137 for which certificate(s) the caPubs field should be used. For security reasons, it cannot be used for delivering the root CA certificate needed for validating the signature-based protection of the given response message (as stated indirectly also in its UNISIG Subset-137 Section 6.3.1.5.2 b [UNISIG.Subset-137]).

- * This profile requires that the certConf message has one CertStatus element where the statusInfo field is recommended.

Note: In contrast, UNISIG Subset-137 Table 18 [UNISIG.Subset-137] requires that the certConf message has one CertStatus element where the statusInfo field must be absent. This precludes sending a negative certConf message in case the EE rejects the newly enrolled certificate. This results in violating the general rule that a certificate request transaction must include a certConf message (since moreover, using implicitConfirm is not allowed there, either).

1.7. Use of CMP in SZTP and BRSKI Environments

In Secure Zero Touch Provisioning (SZTP) [RFC8572] and other environments using NETCONF/YANG modules, SZTP-CSR [I-D.ietf-netconf-sztp-csr] offers a YANG module that includes several types of certificate requests to obtain a public-key certificate for a locally generated key pair. Such messages are of the form ietf-ztp-types:cmp-csr from module ietf-ztp-csr and offer both proof-of-possession and proof-of-identity. To allow PKI management entities that use the module ietf-ztp-csr and also wish to comply with this profile, the ir, cr, kur, or pl0cr message MUST be formatted by the EE as described in Section 4.1, and it MAY be forwarded as specified in Section 5.2.

In Bootstrapping Remote Secure Key Infrastructure (BRSKI) [RFC8995] environments, BRSKI-AE: Alternative Enrollment Protocols in BRSKI [I-D.ietf-anima-brski-ae] describes a generalization regarding the employed enrollment protocols to allow alternatives to EST [RFC7030]. For the use of CMP, it requires adherence to this profile.

1.8. Scope of this Document

This profile on the one hand intends to reduce the flexibility of CMP to the generic needs of automated certificate management of machine end entities. On the other hand, it offers a variety of PKI management operations and options relevant for industrial use cases. Therefore, it is still a framework that supports further profiling by those addressing a specific use case or scenario, e.g., 3GPP/ETSI or UNISIG. There is room for further tailoring this profile. This

enables stricter profiling to the needs of concrete application areas.

To minimize ambiguity and complexity through needless variety, this document specifies exhaustive requirements for generating PKI management messages on the sender side. On the other hand, it gives only minimal requirements on checks by the receiving side and how to handle error cases.

Especially on the EE side this profile aims at a lightweight implementation. This means that the number of PKI management operations implementations are reduced to a reasonable minimum to support typical certificate management use cases in industrial machine-to-machine environments. On the EE side only limited resources are expected, while on the side of the PKI management entities the profile accepts higher requirements.

For the sake of interoperability and robustness, implementations should, as far as security is not affected, adhere to Postel's law: "Be conservative in what you do, be liberal in what you accept from others" (often reworded as: "Be conservative in what you send, be liberal in what you receive").

Fields used in ASN.1 syntax in Section 3, Section 4, or Section 5 are specified in CMP [RFC4210] [I-D.ietf-lamps-cmp-updates], CRMF [RFC4211], and CMS [RFC5652] [RFC8933]. When these sections do not explicitly discuss a field, then the field SHOULD NOT be used by the sending entity. The receiving entity MUST NOT require the absence of such a field, and if the field is present, MUST handle it gracefully.

1.9. Structure of this Document

Section 2 introduces the general PKI architecture and approach to certificate management that is assumed in this document.

Section 3 profiles the generic aspects of the PKI management operations specified in detail in Sections 4 and 5 to minimize redundancy in the description and to ease implementation. This covers the general structure and protection of messages, as well as generic prerequisites, validation, and error handling.

Section 4 profiles the exchange of CMP messages between an EE and the PKI management entity. There are various flavors of certificate enrollment requests, optionally with polling, central key generation, revocation, and general support PKI management operations.

Section 5 profiles responding to requests, exchanges between PKI management entities, and operations on behalf of other PKI entities. This may include delayed delivery of messages, which involves polling for responses, and nesting of messages.

Section 6 outlines several mechanisms for CMP message transfer, including HTTP-based transfer [RFC6712] optionally using TLS, and CoAP-based transfer [I-D.ietf-ace-cmpv2-coap-transport] optionally using DTLS, and offline file-based transport.

Section 7 defines which parts of the profile are mandatory, recommended, optional, or not relevant to implement for which type of entity.

2. Solution Architecture

To facilitate secure automatic certificate enrollment, the device hosting an EE is typically equipped with a manufacturer-issued device certificate. Such a certificate is typically installed during production and is meant to identify the device throughout its lifetime. This certificate can be used to protect the initial enrollment of operational certificates after installation of the EE in its operational environment. In contrast to the manufacturer-issued device certificate, operational certificates are issued by the owner or operator of the device to identify the device or one of its components for operational use, e.g., in a security protocol like IPsec, TLS, or SSH. In IEEE 802.1AR [IEEE.802.1AR_2018] a manufacturer-issued device certificate is called IDevID certificate and an operational certificate is called LDevID certificate.

Note: The owner or operator using the manufacturer-issued device certificate for authenticating the device during initial enrollment of operational certificates MUST trust the respective trust anchor provided by the manufacturer.

Note: According to IEEE 802.1AR [IEEE.802.1AR_2018] a DevID comprises the triple of the certificate, the corresponding private key, and the certificate chain.

All certificate management operations specified in this document follow the pull model, i.e., are initiated by an EE (or by an RA acting as an EE). The EE creates a CMP request message, protects it using some asymmetric credential or shared secret information and sends it to a PKI management entity. This PKI management entity may be a CA or more typically an RA, which checks the request, responds to it itself, or forwards the request upstream to the next PKI management entity. In case an RA changes the CMP request message header or body or wants to demonstrate successful verification or

authorization, it can apply a protection of its own. The communication between an LRA and RA can be performed synchronously or asynchronously. Asynchronous communication typically leads to delayed message delivery as described in Section 4.4.

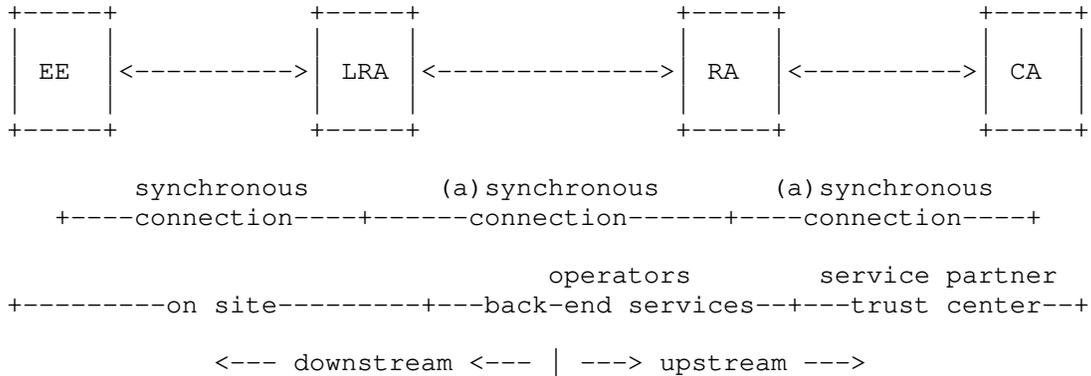


Figure 1: Certificate Management Architecture Example

In operational environments the certificate management architecture can have multiple LRAs bundling requests from multiple EEs at dedicated locations and one (or more than one) central RA aggregating the requests from the LRAs. Every LRA in this scenario has shared secret information (one per EE) for MAC-based protection or a CMP protection key and certificate allowing it to protect CMP messages it processes using its own credentials. The figure above shows an architectural example with one LRA, RA, and CA. It is also possible not to have an RA or LRA or that there is no CA with a CMP interface. Depending on the network infrastructure, the message transfer between PKI management entities may be based on synchronous online connections, asynchronous connections, or even offline (e.g., file-based) transfer.

Note: In contrast to the pull model used in this document, other specifications could use the messages specified in this document implementing the push model. In this case the EE is pushed (triggered) by the PKI management entity to provide the CMP request, and therefore, EE acts as the receiver, not initiating the interaction with the PKI. For example, when the device itself does only act as a server as described in BRSKI with Pledge in Responder Mode (BRSKI-PRM) [I-D.ietf-anima-brski-prm], support of certificate enrollment in a push model is needed. While BRSKI-PRM currently utilizes its own format for the exchanges, CMP in general and the messages specified in this profile offer all required capabilities. Nevertheless, the message flow and state machine as described in Section 4 must be adapted to implement a push model.

Note: Third-party CAs, not conforming to this document, may implement other variants of CMP, different standardized protocols, or even proprietary interfaces for certificate management. In such cases, an RA needs to adapt the exchanged CMP messages to the flavor of certificate management interaction required by such a non-conformant CA.

3. Generic Aspects of PKI Messages and PKI Management Operations

This section covers the generic aspects of the PKI management operations specified in Sections 4 and 5 as upfront general requirements to minimize redundancy in the description and to ease implementation.

As described in Section 5.1 of RFC 4210 [RFC4210], all CMP messages have the following general structure:

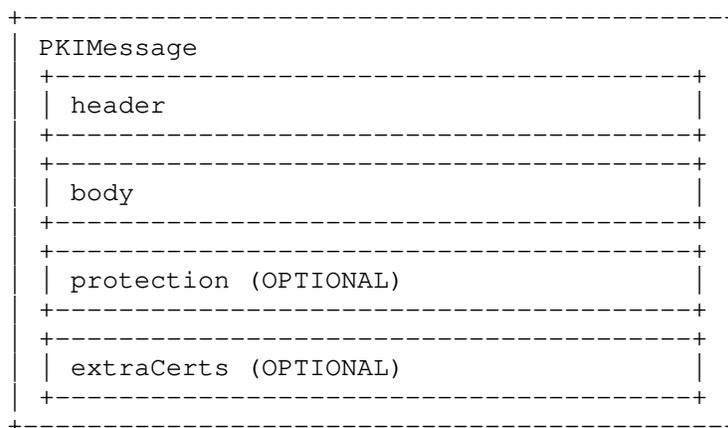


Figure 2: CMP Message Structure

The general contents of the message header, protection, and extraCerts fields are specified in the following three subsections.

In case a specific PKI management operation needs different contents in the header, protection, or extraCerts fields, the differences are described in the respective subsections of Sections 4 and 5.

The CMP message body contains the PKI management operation-specific information. It is described in Sections 4 and 5.

Note: In the description of CMP messages, the presence of some fields is stated as OPTIONAL or RECOMMENDED. The following text that states requirements on such a field applies only if the field is present.

The generic prerequisites needed by the PKI entities in order to be able to perform PKI management operations are described in Section 3.4.

The generic validation steps to be performed by PKI entities on receiving a CMP message are described in Section 3.5.

The generic aspects of handling and reporting errors are described in Section 3.6.

3.1. General Description of the CMP Message Header

This section describes the generic header fields of all CMP messages.

Any PKI management operation-specific fields or variations are described in Sections 4 and 5.

```

header
  pvno                                REQUIRED
    -- MUST be 3 to indicate CMP v3 in all cases where EnvelopedData
    -- is supported and expected to be used in the current
    -- PKI management operation
    -- MUST be 3 to indicate CMP v3 in certConf messages when using
    -- the hashAlg field
    -- MUST be 2 to indicate CMP v2 in all other cases
    -- For details on version negotiation see RFC4555
  sender                              REQUIRED
    -- Contains a name representing the originator which also
    -- protects the message
    -- For signature-based protection MUST be the subject of the CMP
    -- protection certificate
    -- For MAC-based protection MUST be the subject name of the
    -- certificate request, if available; otherwise, the NULL-DN
    -- (a zero-length SEQUENCE OF RelativeDistinguishedNames) MUST
    -- be used
    -- In a multi-hop scenario, the receiving entity cannot rely
    -- on the correctness of the sender field.
  recipient                            REQUIRED
    -- SHOULD be the name of the intended recipient; otherwise, the
    -- NULL-DN MUST be used
    -- In the first message of a PKI management operation: SHOULD be
    -- the subject DN of the CA the PKI management operation is
    -- requested from
    -- In all other messages: SHOULD contain the value of the sender
    -- field of the previous message in the same PKI management
    -- operation
    -- The recipient field shall be handled gracefully by the
    -- receiving entity, because in a multi-hop scenario its

```

```
-- correctness cannot be guaranteed.
messageTime                OPTIONAL
-- MUST be present if the confirmWaitTime field is present
-- MUST be the time at which the message was produced, if present
-- MAY be set by a PKI management entity to provide the current
-- time
-- MAY be used by the end entity for time synchronization if the
-- response was received within a short time frame
protectionAlg              REQUIRED
-- MUST be an algorithm identifier indicating the algorithm
-- used for calculating the protection bits
-- If it is a signature algorithm its type MUST be a
-- MSG_SIG_ALG as specified in [RFCBBBB] Section 3 and
-- MUST be consistent with the subjectPublicKeyInfo field of
-- the CMP protection certificate
-- If it is a MAC algorithm its type MUST be a MSG_MAC_ALG as
-- specified in [RFCBBBB] Section 6.1
senderKID                  RECOMMENDED
-- For signature-based protection MUST be used and contain the
-- value of the SubjectKeyIdentifier if present in the CMP
-- protection certificate
-- For MAC-based protection MUST be used and contain a name the
-- PKI management entity can use to identify the shared secret
-- information
transactionID              REQUIRED
-- In the first message of a PKI management operation: MUST be
-- 128 bits of random data, to minimize the probability of
-- having the transactionID already in use at the server
-- In all other messages: MUST be the value from the previous
-- message in the same PKI management operation
senderNonce                 REQUIRED
-- MUST be cryptographically secure and fresh 128 random bits
recipNonce                  RECOMMENDED
-- If this is the first message of a transaction: MUST be absent
-- If this is a delayed response message: MUST be present and
-- contain the value of the senderNonce of the respective
-- request message in the same transaction
-- In all other messages: MUST be present and contain the value
-- of the senderNonce of the previous message in the same
-- transaction
generalInfo                 OPTIONAL
implicitConfirm             OPTIONAL
-- RECOMMENDED in ir/cr/kur/pl0cr messages,
-- OPTIONAL in ip/cp/kup response messages, and
-- PROHIBITED in other types of messages
-- Added to request messages to request omission of the certConf
-- message
-- Added to response messages to grant omission of the certConf
```

```
-- message
-- See [RFC4210] Section 5.1.1.1.
  ImplicitConfirmValue   REQUIRED
-- ImplicitConfirmValue MUST be NULL
confirmWaitTime         OPTIONAL
-- RECOMMENDED in ip/cp/kup messages if implicitConfirm is
-- not included
-- PROHIBITED if implicitConfirm is included
-- See [RFC4210] Section 5.1.1.2.
  ConfirmWaitTimeValue   REQUIRED
-- ConfirmWaitTimeValue MUST be a GeneralizedTime value
-- specifying the point in time up to which the PKI management
-- entity will wait for the certConf message. The accepted
-- length of the waiting period will vary by use case.
certProfile              OPTIONAL
-- MAY be present in ir/cr/kur/pl0cr and in genm messages of type
-- id-it-certReqTemplate
-- MUST be omitted in all other messages
-- See [RFC4210]
  CertProfileValue       REQUIRED
-- MUST contain a sequence of one UTF8String element
-- MUST contain the name of a certificate profile
```

3.2. General Description of the CMP Message Protection

This section describes the generic protection field contents of all CMP messages. For signature-based protection, which is the default protection mechanism for all CMP messages described in this profile, the CMP protection key and CMP protection certificate are used. For MAC-based protection shared secret information is used as described in Section 4.1.5.

protection

```
-- If present, the same kind of protection MUST be used for all
-- messages of that PKI management operation.
-- MUST be present, except if protection is not possible for
-- error messages as described in Section 3.6.4.
-- For signature-based protection MUST contain the signature
-- calculated using the CMP protection key of the entity
-- protecting the message.
-- For MAC-based protection MUST contain a MAC calculated using
-- the shared secret information.
-- The protection algorithm used MUST be given in the
-- protectionAlg field.
```

The CMP message protection provides, if available, message origin authentication and integrity protection for the header and body. The CMP message extraCerts field is not covered by this protection.

Note: The extended key usages described in CMP Updates Section 2.2 [I-D.ietf-lamps-cmp-updates] can be used for authorization of a sending PKI management entity.

3.3. General Description of CMP Message ExtraCerts

This section describes the generic extraCerts field of all CMP messages. Any specific requirements on the extraCerts are specified in the respective PKI management operation.

extraCerts

- MUST be present for signature-based protection and contain the
- CMP protection certificate together with its chain for the
- first request and response message of a PKI management
- operation. MAY be omitted in certConf, PKIConf, pollReq, and
- pollRep messages. The first certificate in this field MUST
- be the CMP protection certificate followed by its chain
- where each element should directly certify the one
- immediately preceding it.
- MUST be present in ip, cp, and kup messages and contain the
- chain of a newly issued certificate.
- Self-signed certificates should be omitted from extraCerts and
- MUST NOT be trusted based on their inclusion in any case

Note: One reason for adding a self-signed certificate to extraCerts is if it is the CMP protection certificate or a successor root CA self-signed certificate as indicated in the HashOfRootKey extension of the current root CA certificate, see [RFC8649]. Another reason for including self-signed certificates in the extraCerts is, for instance due to storage limitations, a receiving PKI entity may not have the complete trust anchor as self-signed certificate available but just unique identification of it, and thus needs the full self-signed certificate for further processing (see also Section 9).

For maximum interoperability, all implementations SHOULD be prepared to handle potentially additional certificates and arbitrary orderings of the certificates.

3.4. Generic PKI Management Operation Prerequisites

This subsection describes what is generally needed by the PKI entities to be able to perform PKI management operations.

Identification of PKI entities:

- * For signature-based protection each EE knows its own identity from the CMP protection certificate and for MAC-based protection it MAY know its identity to fill the sender field.

- * Each EE MAY know the intended recipient of its requests to fill the recipient field, e.g., the name of the addressed CA.

Note: This name may be established using an enrollment voucher, e.g., [RFC8366], the issuer field from a CertReqTemplate response message content, or by other configuration means.

Routing of CMP messages:

- * Each PKI entity sending messages upstream MUST know the address needed for transferring messages to the next PKI management entity in case online-transfer is used.

Note: This address may depend on the recipient, the certificate profile, and on the used transfer mechanism.

Authentication of PKI entities:

- * Each PKI entity MUST have credentials to authenticate itself. For signature-based protection it MUST have a private key and the corresponding certificate along with its chain.
- * Each PKI entity MUST be able to establish trust in PKI it receives responses from. When signature-based protection is used, it MUST have the trust anchor(s) and any certificate status information needed to perform path validation of CMP protection certificates used for signature-based protection.

Note: A trust anchor usually is a root certificate of the PKI addressed by the requesting EE. It may be established by configuration or in an out-of-band manner. For an EE it may be established using an enrollment voucher [RFC8366] or in-band of CMP by the caPubs field in a certificate response message.

Authorization of PKI management operations:

- * Each EE or RA MUST have sufficient information to be able to authorize the PKI management entity for performing the upstream PKI management operation.

Note: This may be achieved for example by using the cmcRA extended key usage in server certificates, by local configuration such as specific name patterns for subject DN or SAN portions that may identify an RA, and/or by having a dedicated root CA usable only for authenticating PKI management entities.

- * Each PKI management entity MUST have sufficient information to be able to authorize the downstream PKI entity requesting the PKI management operation.

Note: For authorizing an RA the same examples apply as above. The authorization of EEs can be very specific to the application domain based on local PKI policy.

3.5. Generic Validation of a PKI Message

This section describes generic validation steps of each PKI entity receiving a PKI request or response message before any further processing or forwarding. If a PKI management entity decides to terminate a PKI management operation because a check failed, it MUST send a negative response or an error message as described in Section 3.6. The PKIFailureInfo bits given below in parentheses MAY be used in the failInfo field of the PKIStatusInfo as described in Section 3.6.4, see also RFC 4210 Appendix F [RFC4210].

All PKI message header fields not mentioned in this section like the recipient and generalInfo fields SHOULD be handled gracefully on reception.

The following list describes the basic set of message input validation steps. Without these checks the protocol becomes dysfunctional.

- * The formal ASN.1 syntax of the whole message MUST be compliant with the definitions given in CMP [RFC4210] and [I-D.ietf-lamps-cmp-updates], CRMF [RFC4211], and CMS [RFC5652] and [RFC8933]. (failInfo: badDataFormat)
- * The pvno MUST be cmp2000(2) or cmp2021(3). (failInfo bit: unsupportedVersion)
- * The transactionID MUST be present. (failInfo bit: badDataFormat)
- * The PKI message body type MUST be one of the message types supported by the receiving PKI entity and MUST be allowed in the current state of the PKI management operation identified by the given transactionID. (failInfo bit: badRequest)

The following list describes the set of message input validation steps required to ensure secure protocol operation:

- * The senderNonce MUST be present and MUST contain at least 128 bits of data. (failInfo bit: badSenderNonce)

- * Unless the PKI message is the first message of a PKI management operation,
 - the recipNonce MUST be present and MUST equal the senderNonce of the previous message or equal the senderNonce of the most recent request message for which the response was delayed, in case of delayed delivery as specified in Section 4.4. (failInfo bit: badRecipientNonce)
- * Messages without protection MUST be rejected except for error messages as described in Section 3.6.4.
- * The message protection MUST be validated when present and messages with an invalid protection MUST be rejected.
 - The protection MUST be signature-based except if MAC-based protection is used as described in Section 4.1.5 and Section 4.1.6.3. (failInfo bit: wrongIntegrity)
 - If present, the senderKID MUST identify the key material needed for verifying the message protection. (failInfo bit: badMessageCheck)
 - If signature-based protection is used, the CMP protection certificate MUST be successfully validated including path validation using a trust anchor and MUST be authorized according to local policies. If the keyUsage extension is present in the CMP protection certificate the digitalSignature bit MUST be set. (failInfo bit: badAlg, badMessageCheck, or signerNotTrusted)
 - The sender of a request message MUST be authorized for requesting the operation according to PKI policies. (failInfo bit: notAuthorized)

Note: The requirements for checking certificates given in RFC 5280 [RFC5280] MUST be followed for signature-based CMP message protection. Unless the message is a positive ip/cp/kup where the issuing CA certificate of the newly enrolled certificate is the same as the CMP protection certificate of that message, certificate status checking SHOULD be performed on the CMP protection certificates. If the response message contains the caPubs field to transfer new trust anchor information, the CMP protection is crucial and certificate status checking is REQUIRED. For other cases it MAY be acceptable to omit certificate status checking when respective information is not available.

Depending on local policies, one or more of the input validation checks described below need to be implemented:

- * If signature-based protection is used, the sender field MUST match the subject of the CMP protection certificate. (failInfo bit: badMessageCheck)
- * If the messageTime is present and
 - the receiving system has a reliable system time, the messageTime MUST be close to the current time of the receiving system, where the threshold will vary by use case. (failInfo bit: badTime)
 - the receiving system does not have a reliable system time, the messageTime MAY be used for time synchronization.

3.6. Error Handling

This section describes how a PKI entity handles error conditions on messages it receives. Each error condition should be logged appropriately to allow root-cause analysis of failure cases.

3.6.1. Reporting Error Conditions Upstream

An EE SHALL NOT send error messages. PKI management entities SHALL NOT send error messages in the upstream direction, either.

In case an EE rejects a newly issued certificate contained in an ip, cp, or kup message and implicit confirmation has not been granted, the EE MUST report this using a certConf message with "rejection" status and await the pkiConf response as described in Section 4.1.1.

On all other error conditions regarding response messages, the EE or PKI management entity MUST regard the current PKI management operation as terminated with failure. The error conditions include

- * invalid response message header, body type, protection, or extraCerts according to the checks described in Section 3.5,
- * any issue detected with response message contents,
- * receipt of an error message from upstream,
- * timeout occurred while waiting for a response,
- * rejection of a newly issued certificate while implicit confirmation has been granted.

Upstream PKI management entities will not receive any CMP message to learn that the PKI management operation has been terminated. In case they expect a further message from the EE, a connection interruption or timeout will occur. The value set for such timeouts will vary by use case. Then they also MUST regard the current PKI management operation as terminated with failure and MUST NOT attempt to send an error message downstream.

3.6.2. Reporting Error Conditions Downstream

In case the PKI management entity detects an error condition, e.g., rejecting the request due to policy decision, in the body of an ir, cr, pl0cr, kur, or rr message received from downstream, it MUST report the error in the specific response message, i.e., an ip, cp, kup, or rp with "rejection" status, as described in Section 4.1.1 and Section 4.2. This can also happen in case of polling.

In case the PKI management entity detects any other error condition on requests, including pollReq, certConf, genm, and nested messages, received from downstream and on responses received from upstream, such as invalid message header, body type, protection, or extraCerts according to the checks described in Section 3.5 it MUST report them downstream in the form of an error message as described in Section 3.6.4.

3.6.3. Handling Error Conditions on Nested Messages Used for Batching

Batching of messages using nested messages as described in Section 5.2.2.2 requires special error handling.

If the error condition is on an upstream nested message containing batched requests, it MUST NOT attempt to respond to the individual requests included in it, but to the nested message itself.

In case a PKI management entity receives an error message in response to a nested message, it must propagate the error by responding with an error message to each of the request messages contained in the nested message.

In case a PKI management entity detects an error condition on the downstream nested message received in response to a nested message sent before and the body of the received nested message still parses, it MAY ignore this error condition and handle the included responses as described in Section 5.2.2.2. Otherwise, it MUST propagate the error by responding with an error message to each of the requests contained in the nested message it sent originally.

3.6.4. PKIStatusInfo and Error Messages

When sending any kind of negative response, including error messages, a PKI entity MUST indicate the error condition in the PKIStatusInfo structure of the respective message as described below. It then MUST regard the current PKI management operation as terminated with failure.

The PKIStatusInfo structure is used to report errors. It may be part of various message types, in particular: ip, cp, kup, certConf, and error. The PKIStatusInfo structure consists of the following fields:

- * status: Here the PKIStatus value "rejection" MUST be used in case an error was detected. When a PKI management entity indicates delayed delivery of a CMP response message to the EE with an error message as described in Section 4.4, the status "waiting" MUST be used there.
- * statusString: Here any human-readable valid value for logging or to display via a user interface should be added.
- * failInfo: Here the PKIFailureInfo bits MAY be used in the way explained in Appendix F of RFC 4210 [RFC4210]. PKIFailureInfo bits regarding the validation described in Section 3.5 are referenced there. The PKIFailureInfo bits referenced in Sections 5.1 and 6 are described here:
 - badCertId: A kur, certConf, or rr message references an unknown certificate
 - badPOP: An ir/cr/kur/pl0cr contains an invalid proof-of-possession
 - certRevoked: Revocation requested for a certificate already revoked
 - badCertTemplate: The contents of a certificate request are not accepted, e.g., a field is missing or has a non-acceptable value or the given public key is already in use in some other certificate (depending on policy).
 - transactionIdInUse: This is sent by a PKI management entity in case the received request contains a transactionID that is currently in use for another transaction. An EE receiving such error message should resend the request in a new transaction using a different transactionID.

- notAuthorized: The sender of a request message is not authorized for requesting the operation.
- systemUnavail: This is sent by a PKI management entity in case a back-end system is not available.
- systemFailure: This is sent by a PKI management entity in case a back-end system is currently not functioning correctly.

An EE receiving a systemUnavail or systemFailure failInfo should resend the request in a new transaction after some time.

Detailed Message Description:

Error Message -- error

Field	Value
header	
	-- As described in Section 3.1
body	
	-- The message indicating the error that occurred
error	REQUIRED
pkIStatusInfo	REQUIRED
status	REQUIRED
	-- MUST have the value "rejection"
statusString	OPTIONAL
	-- This field should contain any human-readable text for
	-- debugging, logging or to display in a GUI
failInfo	OPTIONAL
	-- MAY be present and contain the relevant PKIFailureInfo bits
protection	RECOMMENDED
	-- As described in Section 3.2
extraCerts	RECOMMENDED
	-- As described in Section 3.3

Protecting the error message may not be technically feasible if it is not clear which credential the recipient will be able to use when validating this protection, e.g., in case the request message was fundamentally broken. In these exceptional cases the protection of the error message MAY be omitted.

4. PKI Management Operations

This chapter focuses on the communication of an EE with the PKI management entity it directly talks to. Depending on the network and PKI solution, this can be an RA or directly a CA. Handling of a message by a PKI management entity is described in Section 5.

The PKI management operations specified in this section cover the following:

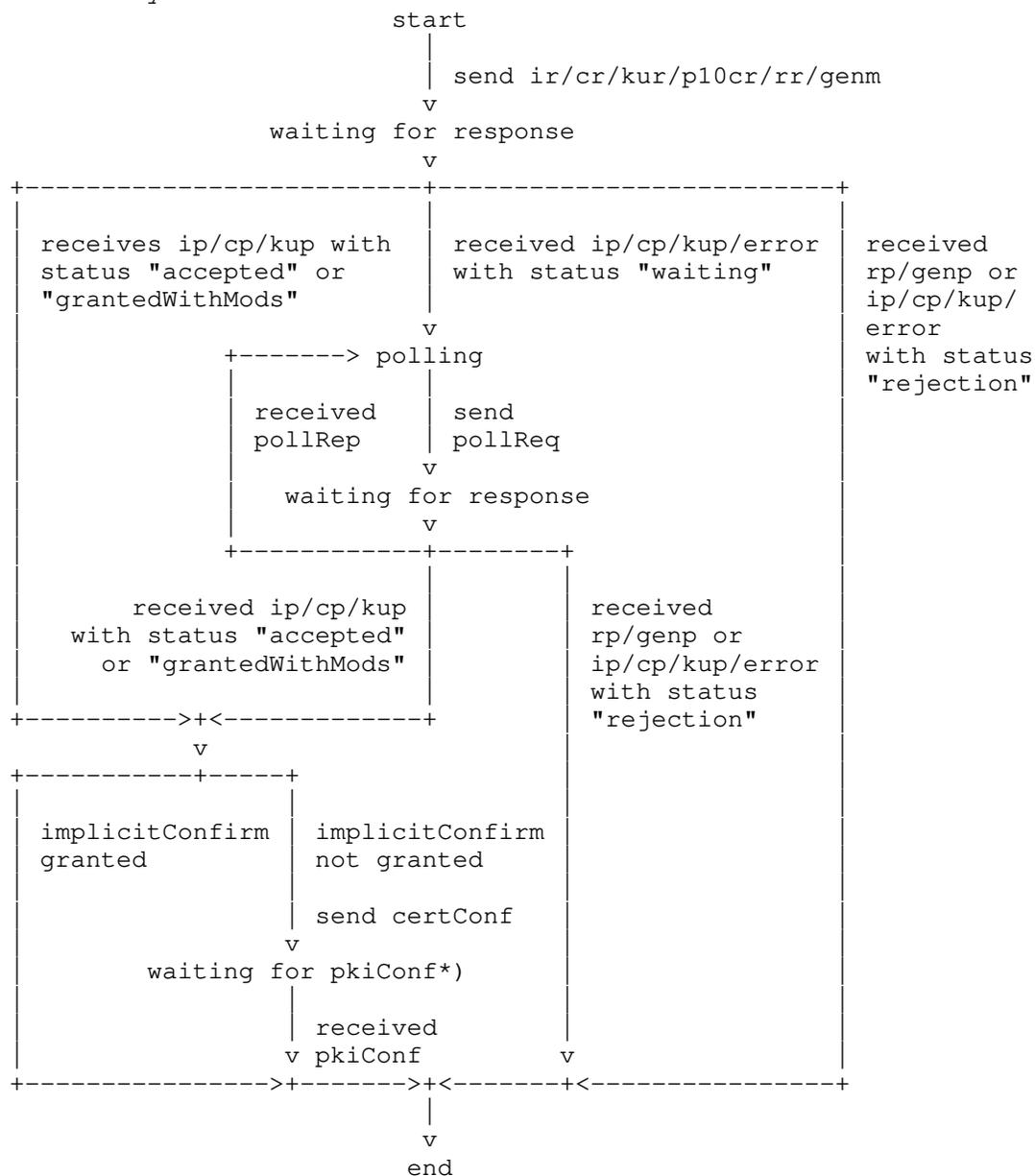
- * Requesting a certificate with variations like initial enrollment, certificate updates, central key generation, and MAC-based protection
- * Revoking a certificate
- * Support messages
- * Polling for delayed response messages

These operations mainly specify the message body of the CMP messages and utilize the specification of the message header, protection and extraCerts as specified in Section 3. The messages are named by the respective field names in PKIBody like ir, ip, cr, cp, etc., see RFC 4210 Section 5.1.2 [RFC4210].

The following diagram shows the EE state machine covering all PKI management operations described in this section, including negative responses, error messages described in Section 3.6.4, as well as ip/cp/kup/error messages with status "waiting", pollReq, and pollRep messages as described in Section 4.4.

On receiving messages from upstream, the EE MUST perform the general validation checks described in Section 3.5. The behavior in case an error occurs is described in Section 3.6.

End Entity State Machine:



*) In case of a delayed delivery of pkiConf responses the same polling mechanism is initiated as for rp or genp messages, by sending an error message with status "waiting".

Note: All CMP messages belonging to the same PKI management operation MUST have the same transactionID because the message receiver identifies the elements of the operation in this way.

This section is aligned with CMP [RFC4210], CMP Updates [I-D.ietf-lamps-cmp-updates], and CMP Algorithms [I-D.ietf-lamps-cmp-algorithms].

Guidelines as well as an algorithm use profile for this document are available in CMP Algorithms [I-D.ietf-lamps-cmp-algorithms].

4.1. Enrolling End Entities

There are various approaches for requesting a certificate from a PKI.

These approaches differ in the way the EE authenticates itself to the PKI, in the form of the request being used, and how the key pair to be certified is generated. The authentication mechanisms may be as follows:

- * Using a certificate from an external PKI, e.g., a manufacturer-issued device certificate, and the corresponding private key
- * Using a private key and certificate issued from the same PKI that is addressed for requesting a certificate
- * Using the certificate to be updated and the corresponding private key
- * Using shared secret information known to the EE and the PKI management entity

An EE requests a certificate indirectly or directly from a CA. When the PKI management entity handles the request as described in Section 5.1.1 and responds with a message containing the requested certificate, the EE MUST reply with a confirmation message unless implicitConfirm was granted. The PKI management entity then MUST handle it as described in Section 5.1.2 and respond with a confirmation, closing the PKI management operation.

The message sequences described in this section allow the EE to request certification of a locally or centrally generated public-private key pair. Typically, the EE provides a signature-based proof-of-possession of the private key associated with the public key contained in the certificate request as defined by RFC 4211 Section 4.1 [RFC4211] case 3. To this end it is assumed that the private key can technically be used for signing. This is the case for the most common algorithms RSA, ECDSA, and EdDSA regardless of potentially intended restrictions of the key usage.

Note: RFC 4211 Section 4 [RFC4211] allows for providing proof-of-possession using any method that a key can be used for. In conformance with NIST SP 800-57 Part 1 Section 8.1.5.1.1.2 [NIST.SP.800-57p1r5] the newly generated private key may be used for self-signature, if technically possible, even if the keyUsage extension requested in the certificate request prohibits generation of digital signatures.

The requesting EE provides the binding of the proof-of-possession to its identity by signature-based or MAC-based protection of the CMP request message containing that POP. An upstream PKI management entity should verify whether this EE is authorized to obtain a certificate with the requested subject and other fields and extensions.

The EE MAY indicate the certificate profile to use in the certProfile extension of the generalInfo field in the PKIHeader of the certificate request message as described in Section 3.1.

In case the EE receives a CA certificate in the caPubs field for installation as a new trust anchor, it MUST properly authenticate the message and authorize the sender as trusted source of the new trust anchor. This authorization is typically indicated using shared secret information for protecting an initialization response (ir) message. Authorization can also be signature-based using a certificate issued by another PKI that is explicitly authorized for this purpose. A certificate received in caPubs MUST NOT be accepted as a trust anchor if it is the root CA certificate of the certificate used for protecting the message.

4.1.1. Enrolling an End Entity to a New PKI

This PKI management operation should be used by an EE to request a certificate from a new PKI using an existing certificate from an external PKI, e.g., a manufacturer-issued IDevID certificate [IEEE.802.1AR_2018], to authenticate itself to the new PKI.

Note: In Bootstrapping Remote Secure Key Infrastructure (BRSKI) [RFC8995] environments, BRSKI-AE: Alternative Enrollment Protocols in BRSKI [I-D.ietf-anima-brski-ae] describes a generalization regarding enrollment protocols alternative to EST [RFC7030]. As replacement of EST simpleenroll, BRSKI-AE uses this PKI management operation for bootstrapping LDevID certificates.

Specific prerequisites augmenting the prerequisites in Section 3.4:

- * The certificate of the EE MUST have been enrolled by an external PKI, e.g., a manufacturer-issued device certificate.
- * The PKI management entity MUST have the trust anchor of the external PKI.
- * When using the generalInfo field certProfile, the EE MUST know the identifier needed to indicate the requested certificate profile.

Message Flow:

```

Step# EE                               PKI management entity
  1  format ir
  2                                     ->  ir      ->
  3                                     handle or
                                     forward ir
  4                                     format or receive ip
  5                                     possibly grant
                                     implicitConfirm
  6                                     <-  ip      <-
  7  handle ip

----- if implicitConfirm not granted -----
  8  format certConf
  9                                     ->  certConf ->
 10                                     handle or
                                     forward certConf
 11                                     format or receive pkiConf
 12                                     <-  pkiConf <-
 13  handle pkiConf

```

For this PKI management operation, the EE MUST include a sequence of one CertReqMsg in the ir. If more certificates are required, further requests MUST be sent using separate PKI management operations.

The EE MUST include the generalInfo field implicitConfirm in the header of the ir message as described in Section 3.1, unless it requires certificate confirmation. This leaves the choice to the PKI

management entities whether the EE must send a certConf message on receiving a new certificate. Depending on the PKI policy and requirements for managing EE certificates, it can be important for PKI management entities to learn if the EE accepted the new certificate. In such cases, when responding with an ip message, the PKI management entity MUST NOT include the implicitConfirm extension. In case the EE included the generalInfo field implicitConfirm in the request message and the PKI management entity does not need any explicit confirmation from the EE, the PKI management entity MUST include the generalInfo field implicitConfirm in the response message. This prevents explicit certificate confirmation and saves the overhead of a further message round-trip. Otherwise, the PKI management entity SHOULD include confirmWaitTime as described in Section 3.1.

If the EE did not request implicit confirmation or implicit confirmation was not granted by the PKI management entity, certificate confirmation MUST be performed as follows. If the EE successfully received the certificate, it MUST send a certConf message in due time. On receiving a valid certConf message, the PKI management entity MUST respond with a pkiConf message. If the PKI management entity does not receive the expected certConf message in time it MUST handle this like a rejection by the EE. In case of rejection, depending on its policy the PKI management entity MAY revoke the newly issued certificate, notify a monitoring system, or log the event internally.

Note: Depending on PKI policy, a new certificate may be published by a PKI management entity, and explicit confirmation may be required. In this case it is advisable not to do the publication until a positive certificate confirmation has been received. This way the need to revoke the certificate on negative confirmation can be avoided.

If the certificate request was rejected by the CA, the PKI management entity MUST return an ip message containing the status code "rejection" as described in Section 3.6 and the certifiedKeyPair field SHALL be omitted. The EE MUST NOT react to such an ip message with a certConf message and the PKI management operation MUST be terminated.

Detailed Message Description:

Initialization Request -- ir

Field	Value
header	-- As described in Section 3.1
body	-- The request of the EE for a new certificate
ir	REQUIRED
	-- MUST contain a sequence of one CertReqMsg
	-- If more certificates are required, further PKI management
	-- operations needs to be initiated
certReq	REQUIRED
certReqId	REQUIRED
	-- MUST be 0
certTemplate	REQUIRED
version	OPTIONAL
	-- MUST be 2 if supplied
subject	REQUIRED
	-- The EE subject name MUST be carried in the subject field
	-- and/or the subjectAltName extension.
	-- If subject name is present only in the subjectAltName
	-- extension, then the subject field MUST be a NULL-DN
publicKey	OPTIONAL
	-- MUST be present if local key generation is used
	-- MAY be absent if central key generation is requested
algorithm	OPTIONAL
	-- MUST be present if local key generation is used and MUST
	-- include the subject public key algorithm identifier
	-- MAY be present if central key generation is requested and
	-- if present, informs the KGA of algorithm and parameter
	-- preferences regarding the to-be-generated key pair
subjectPublicKey	REQUIRED
	-- MUST contain the public key to be certified in case of local
	-- key generation
	-- MUST be a zero-length BIT STRING if central key generation
	-- is requested
extensions	OPTIONAL
	-- MAY include end-entity-specific X.509 extensions of the
	-- requested certificate like subject alternative name, key
	-- usage, and extended key usage
	-- The subjectAltName extension MUST be present if the EE subject
	-- name includes a subject alternative name.
popo	OPTIONAL
	-- MUST be present if local key generation is used
	-- MUST be absent if central key generation is requested
signature	OPTIONAL

```

-- MUST be used by an EE if the key can be used for signing and
--   if used it MUST have the type POPOSigningKey
  poposkInput          PROHIBITED
-- MUST NOT be used; it is not needed because subject and
--   publicKey are both present in the certTemplate
  algorithmIdentifier  REQUIRED
-- The signature algorithm MUST be consistent with the publicKey
--   algorithm field of the certTemplate
  signature            REQUIRED
-- MUST contain the signature value computed over the DER-encoded
--   certTemplate
  raVerified          OPTIONAL
-- MAY be used by an RA after verifying the proof-of-possession
--   provided by the EE

protection            REQUIRED
  -- As described in Section 3.2

extraCerts           REQUIRED
  -- As described in Section 3.3

Initialization Response -- ip

Field                Value

header
  -- As described in Section 3.1

body
  -- The response of the CA to the request as appropriate
  ip                REQUIRED
  caPubs            OPTIONAL
  -- MAY be used if the certifiedKeyPair field is present
  -- If used it MUST contain only a trust anchor, e.g., root
  --   certificate, of the certificate contained in certOrEncCert
  response          REQUIRED
  -- MUST contain a sequence of one CertResponse
  certReqId        REQUIRED
  -- MUST be 0
  status            REQUIRED
  -- PKIStatusInfo structure MUST be present
  status            REQUIRED
  -- positive values allowed: "accepted", "grantedWithMods"
  -- negative values allowed: "rejection"
  -- "waiting" only allowed with polling use case as described in
  --   Section 4.4
  statusString     OPTIONAL

```

```

-- MAY be any human-readable text for debugging, logging or to
-- display in a GUI
failInfo OPTIONAL
-- MAY be present if status is "rejection"
-- MUST be absent if status is "accepted" or "grantedWithMods"
certifiedKeyPair OPTIONAL
-- MUST be present if status is "accepted" or "grantedWithMods"
-- MUST be absent if status is "rejection"
certOrEncCert REQUIRED
-- MUST be present if status is "accepted" or "grantedWithMods"
certificate REQUIRED
-- MUST be present when certifiedKeyPair is present
-- MUST contain the newly enrolled X.509 certificate
privateKey OPTIONAL
-- MUST be absent in case of local key generation or "rejection"
-- MUST contain the encrypted private key in an EnvelopedData
-- structure as specified in Section 4.1.6 in case the private
-- key was generated centrally

protection REQUIRED
-- As described in Section 3.2

extraCerts REQUIRED
-- As described in Section 3.3
-- MUST contain the chain of the certificate present in
-- certOrEncCert
-- Duplicate certificates MAY be omitted

Certificate Confirmation -- certConf

Field Value

header
-- As described in Section 3.1

body
-- The message of the EE sends as confirmation to the PKI
-- management entity to accept or reject the issued
-- certificates
certConf REQUIRED
-- MUST contain a sequence of one CertStatus
CertStatus REQUIRED
certHash REQUIRED
-- MUST be the hash value of the certificate.
-- The hash algorithm to use MUST be the hash algorithm indicated
-- in the below hashAlg field. If the hashAlg field is not
-- set, it MUST be the hash algorithm defined by the algorithm

```

```

-- identifier of the certificate signature or the dedicated
-- hash algorithm defined in RFCBBBB for the used certificate
-- signature algorithm.
  certReqId          REQUIRED
-- MUST be 0
  statusInfo         OPTIONAL
-- PKIStatusInfo structure should be present
-- Omission indicates acceptance of the indicated certificate
  status             REQUIRED
-- positive values allowed: "accepted"
-- negative values allowed: "rejection"
  statusString       OPTIONAL
-- MAY be any human-readable text for debugging, logging, or to
-- display in a GUI
  failInfo           OPTIONAL
-- MAY be present if status is "rejection"
-- MUST be absent if status is "accepted"
  hashAlg            OPTIONAL
-- The hash algorithm to use for calculating the above certHash
-- If used, the pvno field in the header MUST be cmp2021 (3). For
-- backward compatibility it is NOT RECOMMENDED to use this
-- field, if the hash algorithm to use can be identified by
-- other means, see above.

protection          REQUIRED
-- As described in Section 3.2
-- MUST use the same credentials as in the first request message
-- of this PKI management operation

extraCerts          RECOMMENDED
-- As described in Section 3.3
-- MAY be omitted if the message size is critical and the PKI
-- management entity caches the CMP protection certificate from
-- the first request message of this PKI management operation

PKI Confirmation -- pkiConf

Field              Value

header
-- As described in Section 3.1

body
  pkiConf          REQUIRED
-- The content of this field MUST be NULL

protection          REQUIRED

```

- As described in Section 3.2
- MUST use the same credentials as in the first response
- message of this PKI management operation

extraCerts RECOMMENDED

- As described in Section 3.3
- MAY be omitted if the message size is critical and the EE has
- cached the CMP protection certificate from the first
- response message of this PKI management operation

4.1.2. Enrolling an End Entity to a Known PKI

This PKI management operation should be used by an EE to request an additional certificate of the same PKI it already has certificates from. The EE uses one of these existing certificates to authenticate itself by signing its request messages using the respective private key.

Specific prerequisites augmenting the prerequisites in Section 3.4:

- * The certificate used by the EE MUST have been enrolled by the PKI it requests another certificate from.
- * When using the generalInfo field certProfile, the EE MUST know the identifier needed to indicate the requested certificate profile.

The message sequence for this PKI management operation is identical to that given in Section 4.1.1, with the following changes:

- 1 The body of the first request and response SHOULD be cr and cp. Otherwise ir and ip MUST be used.

Note: Since the difference between ir/ip and cr/cp is syntactically not essential, an ir/ip may be used in this PKI management operation.

- 2 The caPubs field in the certificate response message MUST be absent.

4.1.3. Updating a Valid Certificate

This PKI management operation should be used by an EE to request an update for one of its certificates that is still valid. The EE uses the certificate it wishes to update as the CMP protection certificate. Both for authenticating itself and for proving ownership of the certificate to be updated, it signs the request messages with the corresponding private key.

Specific prerequisites augmenting the prerequisites in Section 3.4:

- * The certificate the EE wishes to update MUST NOT be expired or revoked and MUST have been issued by the addressed CA.
- * A new public-private key pair should be used.
- * When using the generalInfo field certProfile, the EE MUST know the identifier needed to indicate the requested certificate profile.

The message sequence for this PKI management operation is identical to that given in Section 4.1.1, with the following changes:

- 1 The body of the first request and response MUST be kur and kup, respectively.
- 2 Protection of the kur MUST be performed using the certificate to be updated.
- 3 The subject field and/or the subjectAltName extension of the certTemplate MUST contain the EE subject name of the existing certificate to be updated, without modifications.
- 4 The certTemplate SHOULD contain the subject and/or subjectAltName extension and publicKey of the EE only.
- 5 The oldCertId control MAY be used to make clear which certificate is to be updated.
- 6 The caPubs field in the kup message MUST be absent.

As part of the certReq structure of the kur the oldCertId control is added after the certTemplate field.

```
controls
  type                RECOMMENDED
  -- MUST be the value id-regCtrl-oldCertID, if present
  value
    issuer             REQUIRED
    serialNumber       REQUIRED
  -- MUST contain the issuer and serialNumber of the certificate
  --   to be updated
```

4.1.4. Enrolling an End Entity Using a PKCS#10 Request

This PKI management operation can be used by an EE to request a certificate using PKCS#10 [RFC2986] format to interoperate with CAs not supporting CRMF [RFC4211]. This offers a variation of the PKI management operations specified in Sections 4.1.1 to 4.1.3.

In this PKI management operation, the public key and all further certificate template data MUST be contained in the subjectPKInfo and other certificationRequestInfo fields of the PKCS#10 structure.

The prerequisites are the same as given in Section 4.1.2.

The message sequence for this PKI management operation is identical to that given in Sections 4.1.1 to 4.1.3, with the following changes:

- 1 The body of the first request and response MUST be p10cr and cp, respectively.
- 2 The certReqId in the cp message MUST be -1.

Detailed Message Description:

Certification Request -- p10cr

Field	Value
header	
	-- As described in Section 3.1
body	
	-- The request of the EE for a new certificate using a PKCS#10 certificate request
p10cr	REQUIRED
certificationRequestInfo	REQUIRED
version	REQUIRED
	-- MUST be 0 to indicate PKCS#10 V1.7
subject	REQUIRED
	-- The EE subject name MUST be carried in the subject field and/or the subjectAltName extension.
	-- If subject name is present only in the subjectAltName extension, then the subject field MUST be a NULL-DN
subjectPKInfo	REQUIRED
algorithm	REQUIRED
	-- MUST include the subject public key algorithm identifier
subjectPublicKey	REQUIRED
	-- MUST include the public key to be certified
attributes	OPTIONAL
	-- MAY include end-entity-specific X.509 extensions of the requested certificate like subject alternative name, key usage, and extended key usage
	-- The subjectAltName extension MUST be present if the EE subject name includes a subject alternative name.
signatureAlgorithm	REQUIRED
	-- The signature algorithm MUST be consistent with the subjectPKInfo field.
signature	REQUIRED
	-- MUST contain the self-signature for proof-of-possession
protection	REQUIRED
	-- As described in Section 3.2
extraCerts	REQUIRED
	-- As described for the underlying PKI management operation

4.1.5. Using MAC-Based Protection for Enrollment

This is a variant of the PKI management operations described in Sections 4.1.1, 4.1.2 and 4.1.4. It should be used by an EE to request a certificate of a new PKI in case it does not have a certificate to prove its identity to the target PKI, but has some secret information shared with the PKI management entity. Therefore, the request and response messages are MAC-protected using this shared secret information. The distribution of this shared secret is out of scope for this document. The PKI management entity checking the MAC-based protection MUST replace this protection according to Section 5.2.3 as the next hop may not know the shared secret information.

Note: The entropy of the shared secret information is crucial for the level of protection when using MAC-based protection. Further guidance is available in the security considerations of CMP updated by [I-D.ietf-lamps-cmp-updates].

Specific prerequisites augmenting the prerequisites in Section 3.4:

- * Rather than using private keys, certificates, and trust anchors, the EE and the PKI management entity MUST share secret information.

Note: The shared secret information MUST be established out-of-band, e.g., by a service technician during initial local configuration.

- * When using the generalInfo field certProfile, the EE MUST know the identifier needed to indicate the requested certificate profile.

The message sequence for this PKI management operation is identical to that given in Sections 4.1.1, 4.1.2 and 4.1.4, with the following changes:

- 1 The protection of all messages MUST be MAC-based. Therefore, extraCerts fields of all messages do not contain CMP protection certificates and associated chains.
- 2 In case the sending entity does not know its own name by now, it MUST put the NULL-DN into the sender field. The senderKID MUST contain a reference the recipient can use to identify the shared secret information used for the protection, e.g., the username of the EE.

See Section 6 of CMP Algorithms [I-D.ietf-lamps-cmp-algorithms] for details on message authentication code algorithms (MSG_MAC_ALG) to use. Typically, parameters are part of the protectionAlg field, e.g., used for key derivation, like a salt and an iteration count. Such parameters should remain constant for message protection throughout this PKI management operation to reduce the computational overhead.

4.1.6. Adding Central Key Pair Generation to Enrollment

This is a variant of the PKI management operations described in Section 4.1.1 to Section 4.1.4 and the variant described in Section 4.1.5. It needs to be used in case an EE is not able to generate its new public-private key pair itself or central generation of the EE key material is preferred. It is a matter of the local implementation which PKI management entity will act as Key Generation Authority (KGA) and performs the key generation. This PKI management entity MUST use a certificate containing the additional extended key usage extension id-kp-cmKGA in order to be accepted by the EE as a legitimate key generation authority.

Note: As described in Section 5.3.1, the KGA can use the PKI management operation described in Section 4.1.2 to request the certificate for this key pair on behalf of the EE.

When an EE requests central key generation for a certificate update using a kur message, the KGA cannot use a kur message to request the certificate on behalf of the EE as the old EE credential is not available to the KGA for protecting this message. Therefore, if the EE uses the PKI management operation described in Section 4.1.3, the KGA MUST act as described in Section 4.1.2 to request the certificate for the newly generated key pair on behalf of the EE from the CA.

Generally speaking, it is strongly preferable to generate public-private key pairs locally at the EE. This is advisable to make sure that the entity identified in the newly issued certificate is the only entity that knows the private key.

Reasons for central key generation may include the following:

- * Lack of sufficient initial entropy.

Note: Good random numbers are needed not only for key generation but also for session keys and nonces in any security protocol. Therefore, a decent security architecture should anyways support good random number generation on the EE side or provide enough initial entropy for the RNG seed to guarantee good pseudo-random number generation. Yet maybe this is not the case at the time of requesting an initial certificate during manufacturing.

- * Lack of computational resources, in particular for RSA key generation.

Note: Since key generation could be performed in advance to the certificate enrollment communication, it is often not time critical.

Note: As mentioned in Section 2, central key generation may be required in a push model, where the certificate response message is transferred by the PKI management entity to the EE without a previous request message.

The EE requesting central key generation MUST omit the `publicKey` field from the `certTemplate` or, in case it has a preference on the key type to be generated, provide this preference in the `algorithm` sub-field and fill the `subjectPublicKey` sub-field with a zero-length BIT STRING. Both variants indicate to the PKI management entity that a new key pair shall be generated centrally on behalf of the EE.

Note: As the protection of centrally generated keys in the response message has been extended to `EncryptedKey` by CMP Updates Section 2.7 [I-D.ietf-lamps-cmp-updates], `EnvelopedData` is the preferred alternative to `EncryptedValue`. In CRMF Section 2.1.9 [RFC4211] the use of `EncryptedValue` has been deprecated in favor of the `EnvelopedData` structure. Therefore, this profile requires using `EnvelopedData` as specified in CMS Section 6 [RFC5652]. When `EnvelopedData` is to be used in a PKI management operation, CMP v3 MUST be indicated in the message header already for the initial request message, see CMP Updates Section 2.20 [I-D.ietf-lamps-cmp-updates].

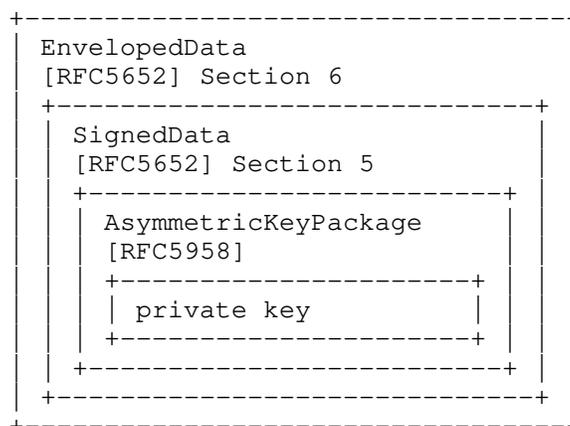


Figure 3: Encrypted Private Key Container

The PKI management entity delivers the private key in the `privateKey` field in the `certifiedKeyPair` structure of the response message also containing the newly issued certificate.

The private key MUST be provided as an `AsymmetricKeyPackage` structure as defined in RFC 5958 [RFC5958].

This `AsymmetricKeyPackage` structure MUST be wrapped in a `SignedData` structure, as specified in CMS Section 5 [RFC5652] and [RFC8933], signed by the KGA generating the key pair. The signature MUST be performed using a private key related to a certificate asserting the extended key usage `id-kp-cmKGA` as described in CMP Updates Section 2.2 [I-D.ietf-lamps-cmp-updates] to demonstrate authorization to generate key pairs on behalf of an EE. For response messages using signature-based protection, the EE MUST validate the signer certificate contained in the `SignedData` structure and SHOULD authorize the KGA considering any given `id-kp-cmKGA` extended key usage in the signer certificate. For response messages using MAC-based protection the EE MAY omit the validation as it may not be possible or meaningful to the EE. In this case the EE authorizes the KGA using the shard secret information.

The `SignedData` structure MUST be wrapped in an `EnvelopedData` structure, as specified in CMS Section 6 [RFC5652], encrypting it using a newly generated symmetric content-encryption key.

This content-encryption key MUST be securely provided as part of the `EnvelopedData` structure to the EE using one of three key management techniques. The choice of the key management technique to be used by the PKI management entity depends on the authentication mechanism the

EE chose to protect the request message. See CMP Updates Section 2.7 [I-D.ietf-lamps-cmp-updates] for details on which key management technique to use.

- * Signature-based protection of the request message:

In this case the choice depends on the type of the public key in the CMP protection certificate used by the EE in its request.

- The content-encryption key SHALL be protected using the key transport key management technique, see Section 4.1.6.1, if the key type supports this.
- The content-encryption key SHALL be protected using the key agreement key management technique, see Section 4.1.6.2, if the key type supports this.

- * MAC-based protected of the request message:

- The content-encryption key SHALL be protected using the password-based key management technique, see Section 4.1.6.3, if and only if the EE used MAC-based protection for the request message.

Specific prerequisites augmenting those of the respective certificate enrollment PKI management operations:

- * If signature-based protection is used, the EE MUST be able to authenticate and authorize the KGA, using suitable information, which includes a trust anchor.
- * If MAC-based protection is used, the KGA MUST also know the shared secret information to protect the encrypted transport of the newly generated key pair. Consequently, the EE can also authorize the KGA.
- * The PKI management entity MUST have a certificate containing the additional extended key usage extension `id-kp-cmKGA` for signing the SignedData structure containing the private key package.
- * For encrypting the SignedData structure a fresh content-encryption key to be used by the symmetric encryption algorithm MUST be generated with sufficient entropy.

Note: The security strength of the protection of the generated private key should be similar or higher than the security strength of the generated private key.

Detailed Description of privateKey Field:

```

    privateKey          REQUIRED
-- MUST be an EnvelopedData structure as specified in CMS
-- Section 6 [RFC5652]
    version            REQUIRED
-- MUST be 2 for recipientInfo type KeyAgreeRecipientInfo and
-- KeyTransRecipientInfo
-- MUST be 0 for recipientInfo type PasswordRecipientInfo
    recipientInfos     REQUIRED
-- MUST contain a sequence of one RecipientInfo, which MUST be
-- kari of type KeyAgreeRecipientInfo (see section 4.1.6.1),
-- ktri of type KeyTransRecipientInfo (see section 4.1.6.2), or
-- pwri of type PasswordRecipientInfo (see section 4.1.6.3)
    encryptedContentInfo
                                REQUIRED
    contentType        REQUIRED
-- MUST be id-signedData
    contentEncryptionAlgorithm
                                REQUIRED
-- MUST be the algorithm identifier of the algorithm used for
-- content encryption
-- The algorithm type MUST be a PROT_SYM_ALG as specified in
-- RFCBBBB Section 5
    encryptedContent    REQUIRED
-- MUST be the SignedData structure as specified in CMS
-- Section 5 [RFC5652] and [RFC8933] in encrypted form
    version            REQUIRED
-- MUST be 3
    digestAlgorithms
                                REQUIRED
-- MUST contain a sequence of one AlgorithmIdentifier element
-- MUST be the algorithm identifier of the digest algorithm
-- used for generating the signature and match the signature
-- algorithm specified in signatureAlgorithm, see [RFC8933]
    encapContentInfo
                                REQUIRED
-- MUST contain the content that is to be signed
    eContentType        REQUIRED
-- MUST be id-ct-KP-aKeyPackage as specified in [RFC5958]
    eContent            REQUIRED
-- MUST be of type AsymmetricKeyPackage and
-- MUST contain a sequence of one OneAsymmetricKey element
    version            REQUIRED
-- MUST be 1 (indicating v2)
    privateKeyAlgorithm
                                REQUIRED
-- The privateKeyAlgorithm field MUST contain the algorithm

```

```
-- identifier of the asymmetric key pair algorithm
    privateKey  REQUIRED
    publicKey   REQUIRED
-- MUST contain the public key corresponding to the private key
-- for simplicity and consistency with v2 of OneAsymmetricKey
    certificates  REQUIRED
-- MUST contain the certificate for the private key used to sign
-- the signedData content, together with its chain
-- The first certificate in this field MUST be the KGA
-- certificate used for protecting this content
-- Self-signed certificates should not be included and MUST NOT
-- be trusted based on their inclusion in any case
    signerInfos  REQUIRED
-- MUST contain a sequence of one SignerInfo element
    version      REQUIRED
-- MUST be 3
    sid          REQUIRED
    subjectKeyIdentifier
                REQUIRED
-- MUST be the subjectKeyIdentifier of the KGA certificate
    digestAlgorithm
                REQUIRED
-- MUST be the same as in the digestAlgorithms field of
-- encryptedContent
    signedAttrs  REQUIRED
-- MUST contain an id-contentType attribute containing the value
-- id-ct-KP-aKeyPackage
-- MUST contain an id-messageDigest attribute containing the
-- message digest of eContent
-- MAY contain an id-signingTime attribute containing the time
-- of signature. It SHOULD be omitted if the transactionTime
-- field is not present in the PKIHeader.
-- For details on the signed attributes see CMS Section 5.3 and
-- Section 11 [RFC5652] and [RFC8933]
    signatureAlgorithm
                REQUIRED
-- MUST be the algorithm identifier of the signature algorithm
-- used for calculation of the signature bits
-- The signature algorithm type MUST be a MSG_SIG_ALG as
-- specified in RFCBBBB Section 3 and MUST be consistent
-- with the subjectPublicKeyInfo field of the KGA certificate
    signature    REQUIRED
-- MUST be the digital signature of the encapContentInfo
```

As stated in Section 1.5, all fields of the ASN.1 syntax that are defined in RFC 5652 [RFC5652] but are not explicitly specified here SHOULD NOT be used.

4.1.6.1. Using Key Transport Key Management Technique

This variant can be applied in combination with the PKI management operations specified in Section 4.1.1 to Section 4.1.3 using signature-based protection of CMP messages. The EE certificate used for the signature-based protection of the request message MUST contain a public key supporting key transport and allow for the key usage "keyEncipherment". The related key pair MUST be used for encipherment of the content-encryption key. For this key management technique, the KeyTransRecipientInfo structure MUST be used in the contentInfo field.

The KeyTransRecipientInfo structure included into the EnvelopedData structure is specified in CMS Section 6.2.1 [RFC5652].

Detailed Description of KeyTransRecipientInfo Structure:

```

      ktri                REQUIRED
-- MUST be a KeyTransRecipientInfo as specified in CMS
-- Section 6.2.1 [RFC5652]
      version            REQUIRED
-- MUST be 2
      rid                REQUIRED
-- MUST contain the subjectKeyIdentifier of the CMP protection
-- certificate, if available, in the rKeyId choice and the
-- subjectKeyIdentifier MUST equal the senderKID in the
-- PKIHeader.
-- If the CMP protection certificate does not contain a
-- subjectKeyIdentifier, the issuerAndSerialNumber choice MUST
-- be used.
      keyEncryptionAlgorithm
                        REQUIRED
-- MUST be the algorithm identifier of the key transport
-- algorithm. The algorithm type MUST be a KM_KT_ALG as
-- specified in RFCBBBB Section 4.2
      encryptedKey       REQUIRED
-- MUST be the encrypted content-encryption key

```

4.1.6.2. Using Key Agreement Key Management Technique

This variant can be applied in combination with the PKI management operations specified in Section 4.1.1 to Section 4.1.3 using signature-based protection of CMP messages. The EE certificate used for the signature-based protection of the request message MUST contain a public key supporting key agreement and allow for the key usage "keyAgreement". The related key pair MUST be used for establishment of the content-encryption key. For this key management technique the KeyAgreeRecipientInfo structure MUST be used in the

contentInfo field.

The KeyAgreeRecipientInfo structure included into the EnvelopedData structure is specified in CMS Section 6.2.2 [RFC5652].

Detailed Description of KeyAgreeRecipientInfo Structure:

```

    kari                REQUIRED
-- MUST be a KeyAgreeRecipientInfo as specified in CMS Section
-- 6.2.2 [RFC5652]
    version            REQUIRED
-- MUST be 3
    originator        REQUIRED
-- MUST contain the subjectKeyIdentifier of the CMP protection
-- certificate, if available, in the subjectKeyIdentifier
-- choice and the subjectKeyIdentifier MUST equal the senderKID
-- in the PKIHeader.
-- If the CMP protection certificate does not contain a
-- subjectKeyIdentifier, the issuerAndSerialNumber choice MUST
-- be used.
    ukm                RECOMMENDED
-- MUST be used when 1-pass ECMQV is used, see [RFC5753]
-- SHOULD be present to ensure uniqueness of the key
-- encryption key
    keyEncryptionAlgorithm
                        REQUIRED
-- MUST be the algorithm identifier of the key agreement
-- algorithm
-- The algorithm type MUST be a KM_KA_ALG as specified in
-- RFCBBBB Section 4.1
-- The parameters field of the key agreement algorithm MUST
-- contain the key wrap algorithm. The algorithm type
-- MUST be a KM_KW_ALG as specified in RFCBBBB Section 4.3
    recipientEncryptedKeys
                        REQUIRED
-- MUST contain a sequence of one RecipientEncryptedKey
    rid                REQUIRED
-- MUST contain the subjectKeyIdentifier of the CMP protection
-- certificate, if available, in the rKeyId choice and the
-- subjectKeyIdentifier MUST equal the senderKID in the
-- PKIHeader.
-- If the CMP protection certificate does not contain a
-- subjectKeyIdentifier, the issuerAndSerialNumber choice MUST
-- be used
    encryptedKey
                        REQUIRED
-- MUST be the encrypted content-encryption key

```

4.1.6.3. Using Password-Based Key Management Technique

This variant can be applied in combination with the PKI management operation specified in Section 4.1.5 using MAC-based protection of CMP messages. The shared secret information used for the MAC-based protection MUST also be used for the encryption of the content-encryption key but with a different salt value applied in the key derivation algorithm. For this key management technique, the PasswordRecipientInfo structure MUST be used in the contentInfo field.

Note: The entropy of the shared secret information is crucial for the level of protection when using a password-based key management technique. For centrally generated key pairs, the entropy of the shared secret information SHALL NOT be less than the security strength of the centrally generated key pair. Further guidance is available in Section 9.

The PasswordRecipientInfo structure included into the EnvelopedData structure is specified in CMS Section 6.2.4 [RFC5652].

Detailed Description of PasswordRecipientInfo Structure:

```

        pwri                REQUIRED
-- MUST be a PasswordRecipientInfo as specified in CMS
--   Section 6.2.4 [RFC5652]
        version            REQUIRED
-- MUST be 0
        keyDerivationAlgorithm
                            REQUIRED
-- MUST be the algorithm identifier of the key derivation
--   algorithm
-- The algorithm type MUST be a KM_KD_ALG as specified in
--   RFCBBBB Section 4.4
        keyEncryptionAlgorithm
                            REQUIRED
-- MUST be the algorithm identifier of the key wrap algorithm
-- The algorithm type MUST be a KM_KW_ALG as specified in
--   RFCBBBB Section 4.3
        encryptedKey       REQUIRED
-- MUST be the encrypted content-encryption key

```

4.2. Revoking a Certificate

This PKI management operation should be used by an entity to request revocation of a certificate. Here the revocation request is used by an EE to revoke one of its own certificates.

The revocation request message MUST be signed using the certificate that is to be revoked to prove the authorization to revoke. The revocation request message is signature-protected using this certificate. This requires, that the EE still possesses the private key. If this is not the case the revocation has to be initiated by other means, e.g., revocation by the RA as specified in Section 5.3.2.

An EE requests revoking a certificate of its own at the CA that issued this certificate. The PKI management entity handles the request as described in Section 5.1.3 and responds with a message that contains the status of the revocation from the CA.

Specific prerequisites augmenting the prerequisites in Section 3.4:

- * The certificate the EE wishes to revoke is not yet expired or revoked.

Message Flow:

Step#	EE		PKI management entity
1	format rr		
2		-> rr	->
3			handle or forward rr
4			format or receive rp
5		<- rp	<-
6	handle rp		

For this PKI management operation, the EE MUST include a sequence of one RevDetails structure in the rr message body. In the case no generic error occurred, the response to the rr MUST be an rp message containing a single status field.

Detailed Message Description:

Revocation Request -- rr

Field	Value
-------	-------

header

-- As described in Section 3.1

body

-- The request of the EE to revoke its certificate

rr REQUIRED

-- MUST contain a sequence of one element of type RevDetails

-- If more revocations are desired, further PKI management

-- operations need to be initiated

```

certDetails          REQUIRED
-- MUST be present and is of type CertTemplate
  serialNumber       REQUIRED
-- MUST contain the certificate serialNumber attribute of the
--   certificate to be revoked
  issuer             REQUIRED
-- MUST contain the issuer attribute of the certificate to be
--   revoked
crlEntryDetails      REQUIRED
-- MUST contain a sequence of one reasonCode of type CRLReason
--   (see [RFC5280] section 5.3.1)
-- If the reason for this revocation is not known or shall not
--   be published the reasonCode MUST be 0 (unspecified)
protection           REQUIRED
-- As described in Section 3.2 and using the private key related
--   to the certificate to be revoked

extraCerts           REQUIRED
-- As described in Section 3.3

Revocation Response -- rp

Field                Value

header
-- As described in Section 3.1

body
-- The responds of the PKI management entity to the request as
--   appropriate
rp                  REQUIRED
  status            REQUIRED
-- MUST contain a sequence of one element of type PKIStatusInfo
  status            REQUIRED
-- positive value allowed: "accepted"
-- negative value allowed: "rejection"
  statusString      OPTIONAL
-- MAY be any human-readable text for debugging, logging or to
--   display in a GUI
  failInfo          OPTIONAL
-- MAY be present if status is "rejection"
-- MUST be absent if the status is "accepted"

protection          REQUIRED
-- As described in section 3.2

extraCerts          REQUIRED

```

-- As described in section 3.3

4.3. Support Messages

The following support messages offer on demand in-band delivery of content relevant to the EE provided by a PKI management entity. CMP general messages and general response are used for this purpose. Depending on the environment, these requests may be answered by an RA or CA (see also Section 5.1.4).

The general messages and general response messages contain InfoTypeAndValue structures. In addition to those infoType values defined in RFC 4210 [RFC4210] and CMP Updates [I-D.ietf-lamps-cmp-updates] further OIDs MAY be used to define new PKI management operations or new general-purpose support messages as needed in specific environments.

The following contents are specified in this document:

- * Get CA certificates
- * Get root CA certificate update
- * Get certificate request template
- * Get new CRLs

The following message flow and contents are common to all general message (genm) and general response (genp) messages.

Message Flow:

Step#	EE		PKI management entity
1	format genm		
2		-> genm	->
3			handle or forward genm
4			format or receive genp
5		<- genp	<-
6	handle genp		

Detailed Message Description:

General Message -- genm

Field	Value
-------	-------

header

-- As described in Section 3.1

```
body
  -- A request by the EE for information
  genm                                REQUIRED
  -- MUST contain a sequence of one element of type
  --   InfoTypeAndValue
  infoType                            REQUIRED
  -- MUST be the OID identifying one of the specific PKI
  --   management operations described below
  infoValue                            OPTIONAL
  -- MUST be as specified for the specific PKI management operation

protection                            REQUIRED
  -- As described in Section 3.2

extraCerts                            REQUIRED
  -- As described in Section 3.3

General Response -- genp

Field                                 Value

header
  -- As described in Section 3.1

body
  -- The response of the PKI management entity providing
  --   information
  genp                                REQUIRED
  -- MUST contain a sequence of one element of type
  --   InfoTypeAndValue
  infoType                            REQUIRED
  -- MUST be the OID identifying the specific PKI management
  --   operation described below
  infoValue                            OPTIONAL
  -- MUST be as specified for the specific PKI management operation

protection                            REQUIRED
  -- As described in Section 3.2

extraCerts                            REQUIRED
  -- As described in Section 3.3
```

4.3.1. Get CA Certificates

This PKI management operation can be used by an EE to request CA certificates from the PKI management entity.

An EE requests CA certificates, e.g., for chain construction, from an PKI management entity by sending a general message with OID `id-it-caCerts` as specified in CMP Updates Section 2.14 [I-D.ietf-lamps-cmp-updates]. The PKI management entity responds with a general response with the same OID that either contains a SEQUENCE of certificates populated with the available intermediate and issuing CA certificates or with no content in case no CA certificate is available.

No specific prerequisites apply in addition to those specified in Section 3.4.

The message sequence for this PKI management operation is as given above, with the following specific content:

- 1 the infoType OID to use is `id-it-caCerts`
- 2 the infoValue of the request MUST be absent
- 3 if present, the infoValue of the response MUST contain a sequence of certificates

Detailed Description of infoValue Field of `genp`:

```
infoValue          OPTIONAL
-- MUST be absent if no CA certificate is available
-- MUST be present if CA certificates are available
-- if present, MUST be a sequence of CMPCertificate
```

4.3.2. Get Root CA Certificate Update

This PKI management operation can be used by an EE to request an updated root CA Certificate as described in Section 4.4 of RFC 4210 [RFC4210].

An EE requests an update of a root CA certificate from the PKI management entity by sending a general message with OID `id-it-rootCaCert`. If needed for unique identification, the EE MUST include the old root CA certificate in the message body, as specified in CMP Updates Section 2.15 [I-D.ietf-lamps-cmp-updates]. The PKI management entity responds with a general response with OID `id-it-rootCaKeyUpdate` that either contains the update of the root CA certificate consisting of up to three certificates, or with no content in case no update is available.

Note: This mechanism may also be used to update trusted non-root certificates, i.e., directly trusted intermediate CA or issuing CA certificates.

The newWithNew certificate is the new root CA certificate and is REQUIRED to be present if available. The newWithOld certificate is REQUIRED to be present in the response message because it is needed for the receiving entity trusting the old root CA certificate to gain trust in the new root CA certificate. The oldWithNew certificate is OPTIONAL because it is only needed in rare scenarios where other entities may not already trust the old root CA.

No specific prerequisites apply in addition to those specified in Section 3.4.

The message sequence for this PKI management operation is as given above, with the following specific content:

- 1 the infoType OID to use is id-it-rootCaCert in the request and id-it-rootCaKeyUpdate in the response
- 2 the infoValue of the request SHOULD contain the root CA certificate the update is requested for
- 3 if present, the infoValue of the response MUST be a RootCaKeyUpdateContent structure

Detailed Description of infoValue Field of genm:

```

infoValue                RECOMMENDED
-- MUST contain the root CA certificate to be updated if needed
--   for unique identification

```

Detailed Description of infoValue Field of genp:

```

infoValue                OPTIONAL
-- MUST be absent if no update of the root CA certificate is
--   available
-- MUST be present if an update of the root CA certificate
--   is available and MUST be of type RootCaKeyUpdateContent
newWithNew                REQUIRED
-- MUST be present if infoValue is present
-- MUST contain the new root CA certificate
newWithOld                REQUIRED
-- MUST be present if infoValue is present
-- MUST contain a certificate containing the new public
--   root CA key signed with the old private root CA key
oldWithNew                OPTIONAL
-- MAY be present if infoValue is present
-- MUST contain a certificate containing the old public
--   root CA key signed with the new private root CA key

```

4.3.3. Get Certificate Request Template

This PKI management operation can be used by an EE to request a template with parameters for future certificate requests.

An EE requests certificate request parameters from the PKI management entity by sending a general message with OID `id-it-certReqTemplate` as specified in CMP Updates Section 2.16 [I-D.ietf-lamps-cmp-updates]. The EE MAY indicate the certificate profile to use in the `id-it-certProfile` extension of the `generalInfo` field in the `PKIHeader` of the general message as described in Section 3.1. The PKI management entity responds with a general response with the same OID that either contains requirements on the certificate request template, or with no content in case no specific requirements are imposed by the PKI. The `CertReqTemplateValue` contains requirements on certificate fields and extensions in a `certTemplate`. Optionally it contains a `keySpec` field containing requirements on algorithms acceptable for key pair generation.

The EE SHOULD follow the requirements from the received `CertTemplate`, by including in the certificate requests all the fields requested, taking over all the field values provided and filling in any remaining fields values. The EE SHOULD NOT add further fields, name components, and extensions or their (sub-)components. If deviating from the recommendations of the template, the certificate request might be rejected.

Note: We deliberately do not use "MUST" or "MUST NOT" here in order to allow more flexibility in case the rules given here are not sufficient for specific scenarios. The EE can populate the certificate request as wanted and ignore any of the requirements contained in the `CertReqTemplateValue`. On the other hand, a PKI management entity is free to ignore or replace any parts of the content of the certificate request provided by the EE. The `CertReqTemplate` PKI management operation offers means to ease a joint understanding which fields and/or which field values should be used. An example is provided in Appendix A.

In case a field of type Name, e.g., subject, is present in the CertTemplate but has the value NULL-DN (i.e., has an empty list of RDN components), the field SHOULD be included in the certificate request and filled with content provided by the EE. Similarly, in case an X.509v3 extension is present but its extnValue is empty, this means that the extension SHOULD be included and filled with content provided by the EE. In case a Name component, for instance a common name or serial number, is given but has an empty string value, the EE SHOULD fill in a value. Similarly, in case an extension has sub-components (e.g., an IP address in a SubjectAltName field) with empty value, the EE SHOULD fill in a value.

The EE MUST ignore (i.e., not include and fill in) empty fields, extensions, and sub-components that it does not understand or does not know suitable values to be filled in.

The publicKey field of type SubjectPublicKeyInfo in the CertTemplate of the CertReqTemplateValue MUST be omitted. In case the PKI management entity wishes to make stipulation on algorithms the EE may use for key generation, this MUST be specified using the keySpec field as specified in CMP Updates Section 2.16 [I-D.ietf-lamps-cmp-updates].

The keySpec field, if present, specifies the public key types optionally with parameters, and/or RSA key lengths for which a certificate may be requested.

The value of a keySpec element with the OID id-regCtrl-algId, as specified in CMP Updates Section 2.16 [I-D.ietf-lamps-cmp-updates], MUST be of type AlgorithmIdentifier and give an algorithm other than RSA. For EC keys the curve information MUST be specified as described in the respective standard documents.

The value of a keySpec element with the OID id-regCtrl-rsaKeyLen, as specified in CMP Updates Section 2.16 [I-D.ietf-lamps-cmp-updates], MUST be a positive integer value and give an RSA key length.

In the CertTemplate of the CertReqTemplateValue the serialNumber, signingAlg, issuerUID, and subjectUID fields MUST be omitted.

Specific prerequisites augmenting the prerequisites in Section 3.4:

- * When using the generalInfo field certProfile, the EE MUST know the identifier needed to indicate the requested certificate profile.

The message sequence for this PKI management operation is as given above, with the following specific content:

- 1 the infoType OID to use is id-it-certReqTemplate
- 2 the id-it-certProfile generalInfo field in the header of the request MAY contain the name of the requested certificate request template
- 3 the infoValue of the request MUST be absent
- 4 if present, the infoValue of the response MUST be a CertReqTemplateValue containing a CertTemplate structure and an optional keySpec field

Detailed Description of infoValue Field of genp:

```

    InfoValue                OPTIONAL
-- MUST be absent if no requirements are available
-- MUST be present if the PKI management entity has any
--   requirements on the contents of the certificate template
    certTemplate            REQUIRED
-- MUST be present if infoValue is present
-- MUST contain the required CertTemplate structure elements
-- The SubjectPublicKeyInfo field MUST be absent
    keySpec                OPTIONAL
-- MUST be absent if no requirements on the public key are
--   available
-- MUST be present if the PKI management entity has any
--   requirements on the keys generated
-- MUST contain a sequence of one AttributeTypeAndValue per
--   supported algorithm with attribute id-regCtrl-algId or
--   id-regCtrl-rsaKeyLen

```

4.3.4. CRL Update Retrieval

This PKI management operation can be used by an EE to request a new CRL. If a CA offers methods to access a CRL, it may include CRL distribution points or authority information access extensions as specified in RFC 5280 [RFC5280] into the issued certificates. In addition, CMP offers CRL provisioning functionality as part of the PKI management operation.

An EE requests a CRL update from the PKI management entity by sending a general message with OID id-it-crlStatusList. The EE MUST include the CRL source identifying the requested CRL and, if available, the thisUpdate time of the most current CRL instance it already has, as specified in CMP Updates Section 2.17 [I-D.ietf-lamps-cmp-updates]. The PKI management entity MUST respond with a general response with OID id-it-crls.

The EE MUST identify the requested CRL either by a CRL distribution point name or issuer name.

Note: CRL distribution point names can be obtained from a `cRLDistributionPoints` extension of a certificate to be validated or from an `issuingDistributionPoint` extension of the CRL to be updated. CRL issuer names can be obtained from the `cRLDistributionPoints` extension of a certificate, from the issuer field of the authority key identifier extension of a certificate or CRL, and from the issuer field of a certificate or CRL.

If a `thisUpdate` value was given, the PKI management entity MUST return the latest CRL available from the referenced source if this CRL is more recent than the given `thisUpdate` time. If no `thisUpdate` value was given, it MUST return the latest CRL available from the referenced source. In all other cases the `infoValue` in the response message MUST be absent.

The PKI management entity should treat a CRL distribution point name as an internal pointer to identify a CRL that is directly available at the PKI management entity. It is not intended as a way to fetch an arbitrary CRL from an external location, as this location may be unavailable to that PKI management entity.

In addition to the prerequisites specified in Section 3.4, the EE MUST know which CRL to request.

Note: If the EE does not want to request a specific CRL it MAY use instead a general message with OID `id-it-currentCrl` as specified in RFC 4210 Section 5.3.19.6 [RFC4210].

The message sequence for this PKI management operation is as given above, with the following specific content:

- 1 the `infoType` OID to use is `id-it-crlStatusList` in the request and `id-it-crls` in the response
- 2 the `infoValue` of the request MUST be present and contain a sequence of one `CRLStatus` structure
- 3 if present, the `infoValue` of the response MUST contain a sequence of one CRL

Detailed Description of `infoValue` Field of `genm`:

```
    infoValue                REQUIRED
-- MUST contain a sequence of one CRLStatus element
    source                    REQUIRED
-- MUST contain the dpn choice of type DistributionPointName if
-- the CRL distribution point name is available
-- Otherwise, MUST contain the issuer choice identifying the CA
-- that issues the CRL. It MUST contain the issuer DN in the
-- directoryName field of a GeneralName element.
    thisUpdate                OPTIONAL
-- MUST contain the thisUpdate field of the latest CRL the EE
-- has got from the issuer specified in the given dpn or
-- issuer field
-- MUST be omitted if the EE does not have any instance of the
-- requested CRL
```

Detailed Description of infoValue Field of genp:

```
    infoValue                OPTIONAL
-- MUST be absent if no CRL to be returned is available
-- MUST contain a sequence of one CRL update from the referenced
-- source, if a thisUpdate value was not given or a more recent
-- CRL is available
```

4.4. Handling Delayed Delivery

This is a variant of all PKI management operations described in this document. It is initiated in case a PKI management entity cannot respond to a request message in a timely manner, typically due to offline or asynchronous upstream communication, or due to delays in handling the request. The polling mechanism has been specified in RFC 4210 Section 5.3.22 [RFC4210] and updated by [I-D.ietf-lamps-cmp-updates].

Depending on the PKI architecture, the entity initiating delayed delivery is not necessarily the PKI management entity directly addressed by the EE.

When initiating delayed delivery of a message received from an EE, the PKI management entity MUST respond with a message including the status "waiting". In response to an ir/cr/kur/pl0cr message it must place the status "waiting" in an ip/cp/kup message, otherwise in an error message. On receiving this response, the EE MUST store in its transaction context the senderNonce of the preceding request message because this value will be needed for checking the recipNonce of the final response to be received after polling. It sends a poll request with certReqId 0 if referring to the CertResponse element contained in the ip/cp/kup message, else -1 to refer to the whole message. In case the final response is not yet available, the PKI management

entity that initiated the delayed delivery MUST answer with a poll response, with the same certReqId. The included checkAfter time value indicates the minimum number of seconds that should elapse before the EE sends a new pollReq message to the PKI management entity. Polling earlier than indicated by the checkAfter value may increase the number of messages roundtrips. This is repeated until a final response is available or any party involved gives up on the current PKI management operation, i.e., a timeout occurs.

When the PKI management entity that initiated delayed delivery can provide the final response for the original request message of the EE, it MUST send this response to the EE. Using this response, the EE can continue the current PKI management operation as usual.

No specific prerequisites apply in addition to those of the respective PKI management operation.

Message Flow:

```

Step# EE                                     PKI management entity
1  format request
   message
2                                     ->   request   ->
3                                     handle or forward
4                                     request
   format ip/cp/kup/error
   with status "waiting"
   response in case no
   immediate final response
   is available,
5                                     <- ip/cp/kup/error <-
6  handle
   ip/cp/kup/error
   with status
   "waiting"

----- start polling -----
7  format pollReq
8                                     ->   pollReq   ->
9                                     handle or forward pollReq
10                                    in case the final response
                                       for the original request
                                       is available, continue
                                       with step 14
                                       otherwise, format or
                                       receive pollRep with
                                       checkAfter value
11                                    <-   pollRep   <-
12  handle pollRep
13  let checkAfter
   time elapse and
   continue with
   step 7

----- end polling, continue as usual -----
14                                    format or receive
                                       final response on
                                       original request
15                                    <-   response   <-
16  handle final
   response

```

Detailed Message Description:

Response with Status "waiting" -- ip/cp/kup/error

Field Value

header

-- As described in Section 3.1

body

-- As described for the respective PKI management operation, with
-- the following adaptations:

status REQUIRED -- in case of ip/cp/kup

pKIStatusInfo REQUIRED -- in case of error response

-- PKIStatusInfo structure MUST be present

status REQUIRED

-- MUST be status "waiting"

statusString OPTIONAL

-- MAY be any human-readable text for debugging, logging or to

-- display in a GUI

failInfo PROHIBITED

protection REQUIRED

-- As described in Section 3.2

extraCerts OPTIONAL

-- As described in Section 3.3

Polling Request -- pollReq

Field Value

header

-- As described in Section 3.1

body

-- The message of the EE asking for the final response or for a
-- time to check again

pollReq REQUIRED

certReqId REQUIRED

-- MUST be 0 if referring to a CertResponse element, else -1

protection REQUIRED

-- As described in Section 3.2

-- MUST use the same credentials as in the first request message

-- of the PKI management operation

extraCerts RECOMMENDED

-- As described in Section 3.3

```

-- MAY be omitted if the message size is critical and the PKI
-- management entity caches the CMP protection certificate from
-- the first request message of the PKI management operation

```

Polling Response -- pollRep

Field	Value
-------	-------

header

```

-- As described in Section 3.1

```

body

```

-- The message indicates the delay after which the EE SHOULD
-- send another pollReq message for this transaction

```

pollRep	REQUIRED
---------	----------

certReqId	REQUIRED
-----------	----------

```

-- MUST be 0 if referring to a CertResponse element, else -1

```

checkAfter	REQUIRED
------------	----------

```

-- MUST be the time in seconds to elapse before a new pollReq
-- should be sent

```

reason	OPTIONAL
--------	----------

```

-- MAY be any human-readable text for debugging, logging or to
-- display in a GUI

```

protection	REQUIRED
------------	----------

```

-- As described in Section 3.2

```

```

-- MUST use the same credentials as in the first response

```

```

-- message of the PKI management operation

```

extraCerts	RECOMMENDED
------------	-------------

```

-- As described in Section 3.3

```

```

-- MAY be omitted if the message size is critical and the EE has

```

```

-- cached the CMP protection certificate from the first

```

```

-- response message of the PKI management operation

```

Final Response - Any Type of Response Message

Field	Value
-------	-------

header

```

-- MUST be the header as described for the response message

```

```

-- of the respective PKI management operation

```

body

```

-- The response of the PKI management entity to the initial

```

```

-- request as described in the respective PKI management

```

-- operation

protection REQUIRED

-- MUST be as described for the response message of the
-- respective PKI management operation

extraCerts REQUIRED

-- MUST be as described for the response message of the
-- respective PKI management operation

5. PKI Management Entity Operations

This section focuses on request processing by a PKI management entity. Depending on the network and PKI solution design, this can be an RA or CA, any of which may include protocol conversion or central key generation (i.e., acting as a KGA).

A PKI management entity may directly respond to request messages from downstream and report errors. In case the PKI management entity is an RA it typically forwards the received request messages upstream after checking them and forwards respective response messages downstream. Besides responding to messages or forwarding them, a PKI management entity may request or revoke certificates on behalf of EEs. A PKI management entity may also need to manage its own certificates and thus act as an EE using the PKI management operations specified in Section 4.

5.1. Responding to Requests

The PKI management entity terminating the PKI management operation at CMP level MUST respond to all received requests by returning a related CMP response message or an error. Any intermediate PKI management entity MAY respond depending on the PKI configuration and policy.

In addition to the checks described in Section 3.5, the responding PKI management entity MUST check that a request that initiates a new PKI management operation does not use a transactionID that is currently in use. The failInfo bit value to use is transactionIdInUse as described in Section 3.6.4. If any of these verification steps or any of the essential checks described in Section 3.5 and in the following subsections fails, the PKI management entity MUST proceed as described in Section 3.6.

The responding PKI management entity MUST copy the sender field of the request to the recipient field of the response, MUST copy the senderNonce of the request to the recipNonce of the response, and MUST use the same transactionID for the response.

5.1.1. Responding to a Certificate Request

An ir/cr/kur/p10cr message is used to request a certificate as described in Section 4.1. The responding PKI management entity MUST proceed as follows unless it initiates delayed delivery as described in Section 5.1.5.

The PKI management entity MUST check the message body according to the applicable requirements from Section 4.1. Possible failInfo bit values used for error reporting in case a check failed include badCertId and badCertTemplate. It MUST verify the presence and value of the proof-of-possession (failInfo bit: badPOP), unless central key generation is requested. In case the special POP value "raVerified" is given, it should check that the request message was signed using a certificate containing the cmcRA extended key usage (failInfo bit: notAuthorized). The PKI management entity should also perform any further checks on the certTemplate contents (failInfo: badCertTemplate) according to any applicable PKI policy and certificate profile.

If the requested certificate is available, the PKI management entity MUST respond with a positive ip/cp/kup message as described in Section 4.1.

Note: If central key generation is performed by the responding PKI management entity, the responding PKI management entity MUST include the private key in encrypted form in the response as specified in Section 4.1.6.

The prerequisites of the respective PKI management operation as specified in Section 4.1 apply.

If the EE requested omission of the certConf message, the PKI management entity MUST handle it as described in Section 4.1.1. Therefore, it MAY grant this by including the implicitConfirm generalInfo field or include the confirmWaitTime field in the response header.

5.1.2. Responding to a Confirmation Message

A PKI management entity MUST handle a certConf message if it has responded before with a positive ip/cp/kup message not granting implicit confirmation. It should check the message body according to the requirements given in Section 4.1.1 (failInfo bit: badCertId) and MUST react as described there.

The prerequisites of the respective PKI management operation as specified in Section 4.1 apply.

5.1.3. Responding to a Revocation Request

An rr message is used to request revocation of a certificate. The responding PKI management entity should check the message body according to the requirements in Section 4.2. It MUST make sure that the referenced certificate exists (failInfo bit: badCertId), has been issued by the addressed CA, and is not already expired or revoked (failInfo bit: certRevoked). On success it MUST respond with a positive rp message as described in Section 4.2.

No specific prerequisites apply in addition to those specified in Section 3.4.

5.1.4. Responding to a Support Message

A genm message is used to retrieve extra content. The responding PKI management entity should check the message body according to the applicable requirements in Section 4.3 and perform any further checks depending on the PKI policy. On success it MUST respond with a genp message as described there.

Note: The responding PKI management entity may generate the response from scratch or reuse the contents of previous responses. Therefore, it may be worth caching the body of the response message as long as the contained information is valid and current, such that further requests for the same contents can be answered immediately.

No specific prerequisites apply in addition to those specified in Section 3.4.

5.1.5. Initiating Delayed Delivery

This functional extension can be used by a PKI management entity in case the response to a request takes longer than usual. In this case the PKI management entity should completely validate the request as usual and then start processing the request itself or forward it further upstream as soon as possible. In the meantime, it MUST respond with an ip/cp/kup/error message including the status "waiting" and handle subsequent polling as described in Section 4.4.

Typically, as stated in Section 5.2.3, an intermediate PKI management entity should not change the sender and recipient nonces even in case it modifies a request or a response message. In the special case of delayed delivery initiated by an intermediate PKI management entity, there is an exception. Between the EE and this PKI management entity, pollReq and pollRep messages are exchanged handling the nonces as usual. Yet when the final response from upstream has arrived at the PKI management entity, this response contains the

recipNonce copied (as usual) from the senderNonce in the original request message. The PKI management entity that initiated the delayed delivery MAY replace the recipNonce in the response message with the senderNonce of the last received pollReq because the downstream entities, including the EE, might expect it in this way. Yet the check specified in Section 3.5 allows to alternatively use the senderNonce of the original request.

No specific prerequisites apply in addition to those of the respective PKI management operation.

5.2. Forwarding Messages

In case the PKI solution consists of intermediate PKI management entities (i.e., LRA or RA), each CMP request message coming from an EE or any other downstream PKI management entity MUST either be forwarded to the next (upstream) PKI management entity as described in this section or answered as described in Section 5.1. Any received response message or a locally generated error message MUST be forwarded to the next (downstream) PKI entity.

In addition to the checks described in Section 3.5, the forwarding PKI management entity MAY verify the proof-of-possession for ir/cr/kur/pl0cr messages. If one of these verification procedures fails, the RA proceeds as described in Section 3.6.

A PKI management entity SHOULD NOT change the received message unless its role in the PKI system requires it. This is because changes to the message header or body imply re-protection. Changes to the protection breaks end-to-end authentication of the message source. Changes to the certificate template in a certificate request breaks proof-of-possession. More details are available in the following sub-sections. Concrete PKI system specifications may define in more detail when to do so.

This is particularly relevant in the upstream communication of a request message.

Each forwarding PKI management entity has one or more functionalities. It may

- * verify the identities of EEs and make authorization decisions for certification request processing based on local PKI policy,
- * add or modify fields of certificate request messages,
- * replace a MAC-based protection by a signature-based protection that can be verified also further upstream, and vice versa,

- * double-check if the messages transferred back and forth are properly protected and well-formed,
- * provide an authentic indication that it has performed all required checks,
- * initiate a delayed delivery due to delays transferring messages or handling requests, or
- * collect messages from multiple RAs and forward them jointly.

Note: PKI management entities forwarding messages may also store data from a message in a database for later usage or audit purposes. They may also support traversal of a network boundary.

The decision if a message should be forwarded

- * unchanged with the original protection,
- * unchanged with an additional protection, or
- * changed with an additional protection

depends on the PKI solution design and the associated security policy (CP/CPS [RFC3647]).

A PKI management entity SHOULD add or MAY replace a protection of a message if it

- * needs to securely indicate that it has done checks or validations on the message to one of the next (upstream) PKI management entity or
- * needs to protect the message using a key and certificate from a different PKI.

If remaining end-to-end message authentication is required, an additional protection SHALL be added instead of replacing the original protection.

A PKI management entity MUST replace a protection of a message if it

- * performs changes to the header or the body of the message or
- * needs to convert from or to a MAC-based protection.

This is particularly relevant in the upstream communication of certificate request messages.

Note that the message protection covers only the header and the body and not the extraCerts. The PKI management entity MAY change the extraCerts in any of the following message adaptations, e.g., to sort, add, or delete certificates to support subsequent PKI entities. This may be particularly helpful to augment upstream messages with additional certificates or to reduce the number of certificates in downstream messages when forwarding to constrained devices.

5.2.1. Not Changing Protection

This variant means that a PKI management entity forwards a CMP message without changing the header, body, or protection. In this case the PKI management entity acts more like a proxy, e.g., on a network boundary, implementing no specific RA-like security functionality that requires an authentic indication to the PKI. Still the PKI management entity might implement checks that result in refusing to forward the request message and instead responding as specified in Section 3.6.

This variant of forwarding a message or the one described in Section 5.2.2.1 MUST be used for kur messages and for central key generation.

No specific prerequisites apply in addition to those specified in Section 3.4.

5.2.2. Adding Protection and Batching of Messages

This variant of forwarding a message means that a PKI management entity adds another protection to PKI management messages before forwarding them.

The nested message is a PKI management message containing a PKIMessages sequence as its body containing one or more CMP messages.

As specified in the updated Section 5.1.3.4 of RFC 4210 [RFC4210] (see also CMP Updates Section 2.6 [I-D.ietf-lamps-cmp-updates]) there are various use cases for adding another protection by a PKI management entity. Specific procedures are described in more detail in the following sections.

Detailed Message Description:

Nested Message - nested

Field	Value
header	-- As described in Section 3.1
body	-- Container to provide additional protection to original -- messages and to bundle request messages or alternatively -- response messages
PKIMessages	REQUIRED -- MUST be a sequence of one or more CMP messages
protection	REQUIRED -- As described in Section 3.2 using the CMP protection key of -- the PKI management entity
extraCerts	REQUIRED -- As described in Section 3.3

5.2.2.1. Adding Protection to a Request Message

This variant means that a PKI management entity forwards a CMP message while authentically indicating successful validation and approval of a request message without changing the original message authentication.

By adding a protection using its own CMP protection key the PKI management entity provides a proof of verifying and approving the message as described above. Thus, the PKI management entity acts as an actual Registration Authority (RA), which implements important security functionality of the PKI. Applying an additional protection is specifically relevant when forwarding a message that requests a certificate update or central key generation. This is because the original protection of the EE needs to be preserved while adding an indication of approval by the PKI management entity.

The PKI management entity wrapping the original request message in a nested message structure MUST copy the values of the recipient, recipNonce, and transactionID header fields of the original message to the respective header fields of the nested message and apply signature-based protection. The additional signature serves as proof of verification and authorization by this PKI management entity.

The PKI management entity receiving such a nested message that contains a single request message MUST validate the additional protection signature on the nested message and check the authorization for the approval it implies.

The PKI management entity responding to the request contained in the nested message sends the response message as described in Section 5.1, without wrapping it in a nested message.

Note: This form of nesting messages is characterized by the fact that the transactionID in the header of the nested message is the same as the one used in the included message.

Specific prerequisites augmenting the prerequisites in Section 3.4:

- * The PKI management entity MUST be able to validate the respective request and have the authorization to perform approval of the request according to the PKI policies.

Message Flow:

Step#	PKI management entity	PKI management entity
1	format nested	
2		-> nested ->
3		handle or forward nested
4		format or receive response
5		<- response <-
6	forward response	

5.2.2.2. Batching Messages

A PKI management entity MAY bundle any number of PKI management messages for batch processing or to transfer a bulk of PKI management messages using the nested message structure. In this use case, nested messages are used both on the upstream interface for transferring request messages towards the next PKI management entity and on its downstream interface for response messages.

This PKI management operation is typically used on the interface between an LRA and an RA to bundle several messages for offline or asynchronous delivery. In this case the LRA needs to initiate delayed delivery as described in Section 5.1.5. If the RA needs different routing information per nested PKI management message provided upstream, a suitable mechanism may need to be implemented to ensure that the downstream delivery of the response is done to the right requester. Since this mechanism strongly depends on the requirements of the target architecture, it is out of scope of this document.

A nested message containing requests is generated locally at the PKI management entity. For the upstream nested message, the PKI management entity acts as a protocol end point and therefore a fresh transactionID and a fresh senderNonce MUST be used in the header of the nested message. An upstream nested message may contain request messages, e.g., ir, cr, pl0cr, kur, pollReq, certConf, rr, or genm. While building the upstream nested message the PKI management entity must store the sender, transactionID, and senderNonce fields of all bundled messages together with the transactionID of the upstream nested message.

Such an upstream nested message is sent to the next PKI management entity. The upstream PKI management entity that unbundles it MUST handle each of the included request messages as usual. It MUST answer with a downstream nested message. This downstream nested message MUST use the transactionID of the upstream nested message and return the senderNonce of the upstream nested message as the recipNonce of the downstream nested message. The downstream nested message MUST bundle all available individual response messages (e.g., ip, cp, kup, pollRep, pkiConf, rp, genp, error) for all original request messages of the upstream nested message. While unbundling the downstream nested message, the former PKI management entity must determine lost and unexpected responses based on the previously stored transactionIDs. When it forwards the unbundled responses, any extra messages MUST be dropped, and any missing response message MUST be answered with an error message (failInfo bit: systemUnavail) to inform the respective requester about the failed certificate management operation.

Note: This form of nesting messages is characterized by the fact that the transactionID in the header of the nested message is different to those used in the included messages.

The protection of the nested messages MUST NOT be regarded as an indication of verification or approval of the bundled PKI request messages.

No specific prerequisites apply in addition to those specified in Section 3.4.

Message Flow:

Step#	PKI management entity		PKI management entity
1	format nested		
2		-> nested	->
3			handle or forward nested
4			format or receive nested
5		<- nested	<-
6	handle nested		

5.2.3. Replacing Protection

The following two alternatives can be used by any PKI management entity forwarding a CMP message with or without changes while providing its own protection and in this way asserting approval of the message.

If retaining end-to-end message authentication is required, an additional protection SHALL be added instead of replacing the original protection.

By replacing the existing protection using its own CMP protection key the PKI management entity provides a proof of verifying and approving the message as described above. Thus, the PKI management entity acts as an actual Registration Authority (RA), which implements important security functionality of the PKI.

Before replacing the existing protection by a new protection, the PKI management entity

- * MUST validate the protection of the received message,
- * should check the content of the message,
- * may do any modifications that it wants to perform, and
- * MUST check that the sender of the original message, as authenticated by the message protection, is authorized for the given operation.

These message adaptations MUST NOT be applied to kur messages described in Section 4.1.3 since their original protection using the key and certificate to be updated needs to be preserved.

These message adaptations MUST NOT be applied to certificate request messages described in Section 4.1.6 for central key generation since their original protection needs to be preserved up to the Key Generation Authority, which needs to use it for encrypting the new private key for the EE.

In both the kur and central key generation cases, if a PKI management entity needs to state its approval of the original request message it MUST provide this using a nested message as specified in Section 5.2.2.1.

When an intermediate PKI management entity modifies a message, it MUST NOT change the transactionID, the senderNonce, or the recipNonce - apart from the exception for the recipNonce given in Section 5.1.5.

5.2.3.1. Not Changing Proof-of-Possession

This variant of forwarding a message means that a PKI management entity forwards a CMP message with or without modifying the message header or body while preserving any included proof-of-possession.

This variant is typically used when an RA replaces an existing MAC-based protection by its own signature-based protection, because the upstream PKI management entity does not know the respective shared secret information, replacing the protection is useful.

Note: A signature-based proof-of-possession of a certificate request will be broken if any field in the certTemplate structure is changed.

In case the PKI management entity breaks an existing proof-of-possession, the message adaptation described in Section 5.2.3.2 needs to be applied instead.

Specific prerequisites augmenting the prerequisites in Section 3.4:

- * The PKI management entity MUST be able to validate the respective request and have the authorization to perform approval of the request according to the PKI policies.

5.2.3.2. Using raVerified

This variant of forwarding a message needs to be used if a PKI management entity breaks any included proof-of-possession in a certificate request message, for instance because it forwards an ir or cr message with modifications of the certTemplate, i.e., modification, addition, or removal of fields.

The PKI management entity MUST verify the proof-of-possession contained in the original message using the included public key. If successful, the PKI management entity MUST change the popo field value to raVerified.

Specific prerequisites augmenting the prerequisites in Section 3.4:

- * The PKI management entity MUST be authorized to replace the proof-of-possession (after verifying it) with raVerified.
- * The PKI management entity MUST be able to validate the respective request and have the authorization to perform approval of the request according to the PKI policies.

Detailed Description of popo Field of certReq Structure:

```
popo
  raVerified          REQUIRED
  -- MUST have the value NULL and indicates that the PKI
  -- management entity verified the popo of the original message
```

5.3. Acting on Behalf of other PKI Entities

A PKI management entity may need to request a PKI management operation on behalf of another PKI entity. In this case the PKI management entity initiates the respective PKI management operation as described in Section 4 acting in the role of the EE.

Note: The request message protection will not authenticate the EE, but the RA acting on behalf of the EE.

5.3.1. Requesting a Certificate

A PKI management entity may use one of the PKI management operations described in Section 4.1 to request a certificate on behalf of another PKI entity. It either generates the key pair itself and inserts the new public key in the subjectPublicKey field of the request certTemplate, or it uses a certificate request received from downstream, e.g., by means of a different protocol. In the latter case it MUST verify the received proof-of-possession if this proof breaks, e.g., due to transformation from PKCS#10 [RFC2986] to CRMF [RFC4211] certificate request format.

No specific prerequisites apply in addition to those specified in Section 4.1.

Note: An upstream PKI management entity will not be able to differentiate this PKI management operation from the one described in Section 5.2.3 because in both cases the message is protected by the PKI management entity.

The message sequence for this PKI management operation is identical to the respective PKI management operation given in Section 4.1, with the following changes:

- 1 The request messages MUST be signed using the CMP protection key of the PKI management entity taking the role of the EE in this operation.
- 2 If inclusion of a proper proof-of-possession is not possible the PKI management entity MUST verify the POP provided from downstream and use "raVerified" in its upstream request.

5.3.2. Revoking a Certificate

A PKI management entity may use the PKI management operation described in Section 4.2 to revoke a certificate of another PKI entity. This revocation request message MUST be signed by the PKI management entity using its own CMP protection key to prove to the PKI authorization to revoke the certificate on behalf of that PKI entity.

No specific prerequisites apply in addition to those specified in Section 4.2.

Note: An upstream PKI management entity will not be able to differentiate this PKI management operation from the ones described in Section 5.2.3.

The message sequence for this PKI management operation is identical to that given in Section 4.2, with the following changes:

- 1 The rr message MUST be signed using the CMP protection key of the PKI management entity acting on behalf of the EE in this operation.

6. CMP Message Transfer Mechanisms

CMP messages are designed to be self-contained, such that in principle any reliable transfer mechanism can be used. EEs will typically support only one transfer mechanism. PKI management entities SHOULD offer HTTP and MAY offer CoAP where required. Piggybacking of CMP messages on any other reliable transfer protocol MAY be used, and file-based transfer MAY be used in case offline transfer is required.

Independently of the means of transfer, it can happen that messages are lost or that a communication partner does not respond. To prevent waiting indefinitely, each PKI entity that sends CMP requests should use a configurable per-request timeout, and each PKI management entity that handles CMP requests should use a configurable timeout in case a further request message is to be expected from the client side within the same transaction. In this way a hanging

transaction can be closed cleanly with an error as described in Section 3.6 (failInfo bit: systemUnavail) and related resources (for instance, any cached extraCerts) can be freed.

Moreover, there are various situations where the delivery of messages gets delayed. For instance, a serving PKI management entity might take longer than expected to form a response due to administrative processes, resource constraints, or upstream message delivery delays. The transport layer itself may cause delays, for instance due to offline transport, network segmentation, or intermittent network connectivity. Part of these issues can be detected and handled at CMP level using pollReq and pollRep messages as described in Section 4.4, while others are better handled at transfer level. Depending on the transfer protocol and system architecture, solutions for handling delays at transfer level may be present and can be used for CMP connections, for instance connection re-establishment and message retransmission.

Note: Long timeout periods are helpful to maximize chances to handle minor delays at lower layers without the need for polling.

Note: When using TCP and similar reliable connection-oriented transport protocols, which is typical in conjunction with HTTP, there is the option to keep the connection alive over multiple request-response message pairs. This may improve efficiency.

When conveying CMP messages in HTTP, CoAP, or MIME-based transfer protocols, the internet media type "application/pkixcmp" MUST be set for transfer encoding as specified in Section 3.4 of CMP over HTTP [RFC6712] and Section 2.4 of CMP over CoAP [I-D.ietf-ace-cmpv2-coap-transport].

6.1. HTTP Transfer

This transfer mechanism can be used by a PKI entity to transfer CMP messages over HTTP. If HTTP transfer is used the specifications as described in [RFC6712] and updated by CMP Updates [I-D.ietf-lamps-cmp-updates] MUST be followed.

PKI management operations MUST use an URI path consisting of '/.well-known/cmp/' or '/.well-known/cmp/p/<name>/' as specified in CMP Updates Section 3.3 [I-D.ietf-lamps-cmp-updates]. It SHOULD be followed by an operation label depending on the type of PKI management operation.

PKI Management Operation	URI Path Segment	Details
Enrolling an End Entity to a New PKI	initialization	Section 4.1.1
Enrolling an End Entity to a Known PKI	certification	Section 4.1.2
Updating a Valid Certificate	keyupdate	Section 4.1.3
Enrolling an End Entity Using a PKCS#10 Request	pkcs10	Section 4.1.4
Revoking a Certificate	revocation	Section 4.2
Get CA Certificates	getcacerts	Section 4.3.1
Get Root CA Certificate Update	getrootupdate	Section 4.3.2
Get Certificate Request Template	getcertreqtemplate	Section 4.3.3
CRL Update Retrieval	getcrls	Section 4.3.4
Batching Messages Note: This path element is applicable only between PKI management entities.	nested	Section 5.2.2.2

Table 1: HTTP URI Path Segment <operation>

If operation labels are used:

- * Independently of any variants used (see Sections 4.1.5, 4.1.6, and 4.4) the operation label corresponding to the PKI management operation SHALL be used.
- * Any certConf or pollReq messages SHALL be sent to the same endpoint as determined by the PKI management operation.

- * When a single request message is nested as described in Section 5.2.2.1, the label to use SHALL be the same as for the underlying PKI management operation.

By sending a request to its preferred endpoint, the PKI entity will recognize via the HTTP response status code whether a configured URI is supported by the PKI management entity.

In case a PKI management entity receives an unexpected HTTP status code from upstream, it MUST respond downstream with an error message as described in Section 3.6 using a failInfo bit corresponding to the status code, e.g., systemFailure.

For certificate management the major security goal is integrity and data origin authentication. For delivery of centrally generated keys, also confidentiality is a must. These goals are sufficiently achieved by CMP itself, also in an end-to-end fashion.

If a second line of defense is required or general privacy concerns exist, TLS can be used to provide confidentiality on a hop-by-hop basis. TLS should be used with certificate-based authentication to further protect the HTTP transfer as described in [RFC9110]. In addition, the recommendations provided in [RFC9325] should be followed.

Note: The requirements for checking certificates given in [RFC5280] and either [RFC5246] or [RFC8446] must be followed for the TLS layer. Certificate status checking should be used for the TLS certificates of all communication partners.

TLS with mutual authentication based on shared secret information may be used in case no suitable certificates for certificate-based authentication are available, e.g., a PKI management operation with MAC-based protection is used.

Note: The entropy of the shared secret information is crucial for the level of protection available using shared secret information-based TLS authentication. A pre-shared key (PSK) mechanism may be used with shared secret information with an entropy of at least 128 bits. Otherwise, a password-authenticated key exchange (PAKE) protocol is recommended.

Note: The provisioning of client certificates and PSKs is out of scope of this document.

6.2. CoAP Transfer

This transfer mechanism can be used by a PKI entity to transfer CMP messages over CoAP [RFC7252], e.g., in constrained environments. If CoAP transfer is used the specifications as described in CMP over CoAP [I-D.ietf-ace-cmpv2-coap-transport] MUST be followed.

PKI management operations MUST use an URI path consisting of `/.well-known/cmp/` or `/.well-known/cmp/p/<name>/` as specified in CMP over CoAP Section 2.1 [I-D.ietf-ace-cmpv2-coap-transport]. It SHOULD be followed by an operation label depending on the type of PKI management operation.

PKI Management Operation	URI Path Segment	Details
Enrolling an End Entity to a New PKI	ir	Section 4.1.1
Enrolling an End Entity to a Known PKI	cr	Section 4.1.2
Updating a Valid Certificate	kur	Section 4.1.3
Enrolling an End Entity Using a PKCS#10 Request	p10	Section 4.1.4
Revoking a Certificate	rr	Section 4.2
Get CA Certificates	crt	Section 4.3.1
Get Root CA Certificate Update	rcu	Section 4.3.2
Get Certificate Request Template	att	Section 4.3.3
CRL Update Retrieval	crls	Section 4.3.4
Batching Messages	nest	Section 5.2.2.2
Note: This path element is applicable only between PKI management entities.		

Table 2: CoAP URI Path Segment <operation>

If operation labels are used:

- * Independently of any variants used (see Sections 4.1.5, 4.1.6, and 4.4) the operation label corresponding to the PKI management operation SHALL be used.
- * Any certConf or pollReq messages SHALL be sent to the same endpoint as determined by the PKI management operation.

- * When a single request message is nested as described in Section 5.2.2.1, the label to use SHALL be the same as for the underlying PKI management operation.

By sending a request to its preferred endpoint, the PKI entity will recognize via the CoAP response status code whether a configured URI is supported by the PKI management entity. The CoAP-inherent discovery mechanisms MAY also be used.

In case a PKI management entity receives an unexpected CoAP status code from upstream, it MUST respond downstream with an error message as described in Section 3.6 using a failInfo bit corresponding to the status code, e.g., systemFailure.

Like for HTTP transfer, to offer a second line of defense or to provide hop-by-hop privacy protection, DTLS may be utilized as described in CMP over CoAP [I-D.ietf-ace-cmpv2-coap-transport]. If DTLS is utilized, the same boundary conditions (peer authentication, etc.) as stated for TLS to protect HTTP transfer in Section 6.1 apply to DTLS likewise.

Note: The provisioning of client certificates and PSKs is out of scope of this document.

6.3. Piggybacking on Other Reliable Transfer

CMP messages MAY also be transfer on some other reliable protocol, e.g., EAP or MQTT. Connection, delay, and error handling mechanisms similar to those specified for HTTP in RFC 6712 [RFC6712] need to be implemented.

A more detailed specification is out of scope of this document and would need to be given for instance in the scope of the transfer protocol used.

6.4. Offline Transfer

For transferring CMP messages between PKI entities, any mechanism can be used that is able to store and forward binary objects of sufficient length and with sufficient reliability while preserving the order of messages for each transaction.

The transfer mechanism should be able to indicate message loss, excessive delay, and possibly other transmission errors. In such cases the PKI entities MUST report an error as specified in Section 3.6 as far as possible.

6.4.1. File-Based Transfer

CMP messages MAY be transferred between PKI entities using file-based mechanisms, for instance when an EE is offline or a PKI management entity performs delayed delivery. Each file MUST contain the ASN.1 DER encoding of one CMP message only, where the message may be nested. There MUST be no extraneous header or trailer information in the file. The file name extension ".pki" MUST be used.

6.4.2. Other Asynchronous Transfer Protocols

Other asynchronous transfer protocols, e.g., email or website up-/download, MAY transfer CMP messages between PKI entities. A MIME wrapping is defined for those environments that are MIME-native. The MIME wrapping is specified in RFC 8551 Section 3.1 [RFC8551].

The ASN.1 DER encoding of the CMP messages MUST be transferred using the "application/pkixcmp" content type and base64-encoded content transfer encoding as specified in Section 3.4 of CMP over HTTP [RFC6712]. A filename MUST be included either in a "content-type" or a "content-disposition" statement. The file name extension ".pki" MUST be used.

7. Conformance Requirements

This section defines which level of support for the various features specified in this profile is required for each type of PKI entity.

7.1. PKI Management Operations

The following table provides an overview of the PKI management operations specified in Sections 4 and 5 and states whether support by conforming EE, RA, and CA implementations is mandatory, recommended, optional, or not applicable. Variants amend or change behavior of base PKI management operations and are therefore also included.

The PKI management operation specifications in Section 4 assume that either the RA or CA is the PKI management entity that terminates the CMP protocol. If the RA terminates the CMP protocol it either responds directly as described in Section 5.1 or forwards the certificate management operation towards the CA not using CMP. Section 5.2 describes different options how an RA can forward a CMP message using CMP. Section 5.3 offers the option that an RA operates on behalf on an EE and therefore takes the role of the EE in Section 4.

ID	PKI Management Operations and Variants	EE	RA	CA
Generic	Generic Aspects of PKI Messages and PKI Management Operations, Section 3	MUST	MUST	MUST
IR	Enrolling an End Entity to a New PKI, Section 4.1.1	MUST	MAY	MUST
CR	Enrolling an End Entity to a Known PKI, Section 4.1.2	MAY	MAY	MAY
KUR	Updating a Valid Certificate, Section 4.1.3	MUST	MAY	MUST
P10CR	Enrolling an End Entity Using a PKCS#10 Request, Section 4.1.4	MAY	MAY	MAY
MAC	Using MAC-Based Protection for Enrollment, with IR, CR, and P10CR if supported, Section 4.1.5	MAY	SHOULD 1)	MAY
CKeyGen	Adding Central Key Pair Generation to Enrollment, IR, CR, KUR, and P10CR if supported, Section 4.1.6	MAY	MAY	MAY
RR	Revoking a Certificate, Section 4.2	SHOULD	SHOULD 2)	SHOULD 3)
CACerts	Get CA Certificates, Section 4.3.1	MAY	MAY	MAY
RootUpd	Get Root CA Certificate Update, Section 4.3.2	MAY	MAY	MAY
ReqTempl	Get Certificate Request Template, Section 4.3.3	MAY	MAY	MAY
CRLUpd	CRL Update Retrieval, Section 4.3.4	MAY	MAY	MAY
Polling	Handling Delayed Delivery,	MAY	MAY	MAY

	Section 4.4			
CertResp	Responding to a Certificate Request (IR, CR, KUR, and P10CR if supported), Section 5.1.1	N/A	MAY	MUST
CertConf	Responding to a Confirmation Message, Section 5.1.2	N/A	MAY	MUST
RevResp	Responding to a Revocation Request, Section 5.1.3	N/A	MAY	SHOULD
GenResp	Responding to a Support Message (CACerts, RootUpd, ReqTempl, CRLUpd if supported), Section 5.1.4	N/A	MAY	MAY
InitPoll	Initiating Delayed Delivery, Section 5.1.5	N/A	MAY	MAY
FwdKeep	Forwarding Messages - Not Changing Protection, Section 5.2.1	N/A	MUST	N/A
FwdAddS	Forwarding Messages - Adding Protection to a Request Message, Section 5.2.2.1	N/A	MUST	MUST
FwdAddB	Forwarding Messages - Batching Messages, Section 5.2.2.2	N/A	MAY	MAY
FwdReqKP	Forwarding Messages - Not Changing Proof-of-Possession, Section 5.2.3.1	N/A	SHOULD 1)	N/A
FwdReqBP	Forwarding Messages - Using raVerified, Section 5.2.3.2	N/A	MAY	MAY
CertROnB	Acting on Behalf of other PKI Entities - Requesting a Certificate, Section 5.3.1	N/A	MAY	N/A

RevROnB	Acting on Behalf of other PKI Entities - Revoking a Certificate, Section 5.3.2	N/A	SHOULD 2)	SHOULD 3)
---------	--	-----	--------------	--------------

Table 3: Level of Support for PKI Management Operations and Variants

- 1) The RA should be able to change the CMP message protection from MAC-based to signature-based protection, see Section 5.2.3.1.
- 2) The RA should be able to request certificate revocation on behalf of an EE, see Section 5.3.2, e.g., in order to handle incidents.
- 3) An alternative would be to perform revocation at the CA without using CMP, for instance using a local administration interface.

7.2. Message Transfer

CMP does not have specific needs regarding message transfer, except that for each request message sent, eventually a sequence of one response message should be received. Therefore, virtually any reliable transfer mechanism can be used, such as HTTP, CoAP, and file-based offline transfer. Thus, this document does not require any specific transfer protocol to be supported by conforming implementations.

On different links between PKI entities, e.g., EE-RA and RA-CA, different transfer mechanisms as specified in Section 6 may be used.

HTTP SHOULD be supported and CoAP MAY be supported at all PKI entities for maximizing general interoperability at transfer level. Yet full flexibility is retained to choose whatever transfer mechanism is suitable, for instance for devices and system architectures with specific constraints.

The following table lists the name and level of support specified for each transfer mechanism.

ID	Message Transfer Type	EE	RA	CA
HTTP	HTTP Transfer, Section 6.1	SHOULD	SHOULD	SHOULD
CoAP	CoAP Transfer, Section 6.2	MAY	MAY	MAY
Piggyb	Piggybacking on Other Reliable Transfer, Section 6.3	MAY	MAY	MAY
Offline	Offline Transfer, Section 6.4	MAY	MAY	MAY

Table 4: Level of Support for Message Transfer Types

8. IANA Considerations

This document defines new entries with the following content in the "CMP Well-Known URI Path Segments" registry (see <https://www.iana.org/assignments/cmp/>) as defined in RFC 8615 [RFC8615].

Path Segment	Description	Reference
initialization	Enrolling an End Entity to a New PKI over HTTP	[thisRFC]
certification	Enrolling an End Entity to a Known PKI over HTTP	[thisRFC]
keyupdate	Updating a Valid Certificate over HTTP	[thisRFC]
pkcs10	Enrolling an End Entity Using a PKCS#10 Request over HTTP	[thisRFC]
revocation	Revoking a Certificate over HTTP	[thisRFC]
getcacerts	Get CA Certificates over HTTP	[thisRFC]
getrootupdate	Get Root CA Certificate Update over HTTP	[thisRFC]

getcertreqtemplate	Get Certificate Request Template over HTTP	[thisRFC]
getcrls	CRL Update Retrieval over HTTP	[thisRFC]
nested	Batching Messages over HTTP	[thisRFC]
ir	Enrolling an End Entity to a New PKI over CoAP	[thisRFC]
cr	Enrolling an End Entity to a Known PKI over CoAP	[thisRFC]
kur	Updating a Valid Certificate over CoAP	[thisRFC]
p10	Enrolling an End Entity Using a PKCS#10 Request over CoAP	[thisRFC]
rr	Revoking a Certificate over CoAP	[thisRFC]
crts	Get CA Certificates over CoAP	[thisRFC]
rcu	Get Root CA Certificate Update over CoAP	[thisRFC]
att	Get Certificate Request Template over CoAP	[thisRFC]
crls	CRL Update Retrieval over CoAP	[thisRFC]
nest	Batching Messages over CoAP	[thisRFC]

Table 5: New "CMP Well-Known URI Path Segments" Registry Entries

< TBD: New entries must be added to the registry "CMP Well-Known URI Path Segments". >

9. Security Considerations

The security considerations as laid out in CMP [RFC4210] updated by CMP Updates [I-D.ietf-lamps-cmp-updates] and CMP Algorithms [I-D.ietf-lamps-cmp-algorithms], CRMF [RFC4211] updated by Algorithm Requirements Update [RFC9045], CMP over HTTP [RFC6712], and CMP over CoAP [I-D.ietf-ace-cmpv2-coap-transport] apply.

Trust anchors for chain validations are often provided in the form of self-signed certificates. All trust anchors MUST be stored on the device with integrity protection. In some cases, a PKI entity may not have sufficient storage for the complete certificates. In such cases it may only store, e.g., a hash of each self-signed certificate and require receiving the certificate in the extraCerts field as described in Section 3.3. If such self-signed certificates are provided in-band in the messages, they MUST be verified using information from the trust store of the PKI entity.

For TLS using shared secret information-based authentication, both PSK and PAKE provide the same amount of protection against a real-time authentication attack which is directly the amount of entropy in the shared secret. The difference between a pre-shared key (PSK) and a password-authenticated key exchange (PAKE) protocol is in the level of long-term confidentiality of the TLS messages against brute-force decryption, where a PSK-based cipher suite only provides security according to the entropy of the shared secret, while a PAKE-based cipher suite provides full security independent of the entropy of the shared secret.

10. Acknowledgements

We thank the various reviewers of this document.

11. References

11.1. Normative References

[I-D.ietf-ace-cmpv2-coap-transport]
Sahni, M. and S. Tripathi, "CoAP Transfer for the Certificate Management Protocol", Work in Progress, Internet-Draft, draft-ietf-ace-cmpv2-coap-transport-07, 27 January 2023, <<https://datatracker.ietf.org/doc/html/draft-ietf-ace-cmpv2-coap-transport-07>>.

[I-D.ietf-lamps-cmp-algorithms]
Brockhaus, H., Aschauer, H., Ounsworth, M., and J. Gray, "Certificate Management Protocol (CMP) Algorithms", Work in Progress, Internet-Draft, draft-ietf-lamps-cmp-

algorithms-15, 2 June 2022,
<<https://datatracker.ietf.org/doc/html/draft-ietf-lamps-cmp-algorithms-15>>.

- [I-D.ietf-lamps-cmp-updates] Brockhaus, H., von Oheimb, D., and J. Gray, "Certificate Management Protocol (CMP) Updates", Work in Progress, Internet-Draft, draft-ietf-lamps-cmp-updates-23, 29 June 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-lamps-cmp-updates-23>>.
- [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>>.
- [RFC2986] Nystrom, M. and B. Kaliski, "PKCS #10: Certification Request Syntax Specification Version 1.7", RFC 2986, DOI 10.17487/RFC2986, November 2000, <<https://www.rfc-editor.org/info/rfc2986>>.
- [RFC4210] Adams, C., Farrell, S., Kause, T., and T. Mononen, "Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP)", RFC 4210, DOI 10.17487/RFC4210, September 2005, <<https://www.rfc-editor.org/info/rfc4210>>.
- [RFC4211] Schaad, J., "Internet X.509 Public Key Infrastructure Certificate Request Message Format (CRMF)", RFC 4211, DOI 10.17487/RFC4211, September 2005, <<https://www.rfc-editor.org/info/rfc4211>>.
- [RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <<https://www.rfc-editor.org/rfc/rfc5280>>.
- [RFC5652] Housley, R., "Cryptographic Message Syntax (CMS)", STD 70, RFC 5652, DOI 10.17487/RFC5652, September 2009, <<https://www.rfc-editor.org/info/rfc5652>>.
- [RFC5958] Turner, S., "Asymmetric Key Packages", RFC 5958, DOI 10.17487/RFC5958, August 2010, <<https://www.rfc-editor.org/info/rfc5958>>.

- [RFC6712] Kause, T. and M. Peylo, "Internet X.509 Public Key Infrastructure -- HTTP Transfer for the Certificate Management Protocol (CMP)", RFC 6712, DOI 10.17487/RFC6712, September 2012, <<https://www.rfc-editor.org/info/rfc6712>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8615] Nottingham, M., "Well-Known Uniform Resource Identifiers (URIs)", RFC 8615, DOI 10.17487/RFC8615, May 2019, <<https://www.rfc-editor.org/info/rfc8615>>.
- [RFC8933] Housley, R., "Update to the Cryptographic Message Syntax (CMS) for Algorithm Identifier Protection", RFC 8933, DOI 10.17487/RFC8933, October 2020, <<https://www.rfc-editor.org/info/rfc8933>>.
- [RFC9045] Housley, R., "Algorithm Requirements Update to the Internet X.509 Public Key Infrastructure Certificate Request Message Format (CRMF)", RFC 9045, DOI 10.17487/RFC9045, June 2021, <<https://www.rfc-editor.org/info/rfc9045>>.
- [RFC9110] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP Semantics", STD 97, RFC 9110, DOI 10.17487/RFC9110, June 2022, <<https://www.rfc-editor.org/info/rfc9110>>.
- [RFC9325] Sheffer, Y., Saint-Andre, P., and T. Fossati, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 9325, DOI 10.17487/RFC9325, November 2022, <<https://www.rfc-editor.org/info/rfc9325>>.

11.2. Informative References

- [ETSI-3GPP.33.310]
3GPP, "Network Domain Security (NDS); Authentication Framework (AF)", 3GPP TS 33.310 16.6.0, 16 December 2020, <<http://www.3gpp.org/ftp/Specs/html-info/33310.htm>>.
- [ETSI-EN.319411-1]
ETSI, "Electronic Signatures and Infrastructures (ESI); Policy and security requirements for Trust Service Providers issuing certificates; Part 1: General requirements", ETSI EN 319 411-1 V1.3.1, May 2021,

<https://www.etsi.org/deliver/etsi_en/319400_319499/31941101/01.03.01_60/en_31941101v010301p.pdf>.

[I-D.ietf-anima-brski-ae]

von Oheimb, D., Fries, S., and H. Brockhaus, "BRSKI-AE: Alternative Enrollment Protocols in BRSKI", Work in Progress, Internet-Draft, draft-ietf-anima-brski-ae-03, 24 October 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-anima-brski-ae-03>>.

[I-D.ietf-anima-brski-prm]

Fries, S., Werner, T., Lear, E., and M. Richardson, "BRSKI with Pledge in Responder Mode (BRSKI-PRM)", Work in Progress, Internet-Draft, draft-ietf-anima-brski-prm-06, 11 January 2023, <<https://datatracker.ietf.org/doc/html/draft-ietf-anima-brski-prm-06>>.

[I-D.ietf-lamps-rfc4210bis]

Brockhaus, H., von Oheimb, D., Ounsworth, M., and J. Gray, "Internet X.509 Public Key Infrastructure -- Certificate Management Protocol (CMP)", Work in Progress, Internet-Draft, draft-ietf-lamps-rfc4210bis-03, 24 October 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-lamps-rfc4210bis-03>>.

[I-D.ietf-lamps-rfc6712bis]

Brockhaus, H., von Oheimb, D., Ounsworth, M., and J. Gray, "Internet X.509 Public Key Infrastructure -- HTTP Transfer for the Certificate Management Protocol (CMP)", Work in Progress, Internet-Draft, draft-ietf-lamps-rfc6712bis-03, 10 February 2023, <<https://datatracker.ietf.org/doc/html/draft-ietf-lamps-rfc6712bis-03>>.

[I-D.ietf-netconf-sztp-csr]

Watsen, K., Housley, R., and S. Turner, "Conveying a Certificate Signing Request (CSR) in a Secure Zero Touch Provisioning (SZTP) Bootstrapping Request", Work in Progress, Internet-Draft, draft-ietf-netconf-sztp-csr-14, 2 March 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-netconf-sztp-csr-14>>.

[IEC.62443-3-3]

IEC, "Industrial communication networks - Network and system security - Part 3-3: System security requirements and security levels", IEC 62443-3-3, August 2013, <<https://webstore.iec.ch/publication/7033>>.

- [IEEE.802.1AR_2018]
IEEE, "IEEE Standard for Local and metropolitan area networks - Secure Device Identity", IEEE 802.1AR-2018, DOI 10.1109/IEEESTD.2018.8423794, 2 August 2018, <<https://ieeexplore.ieee.org/document/8423794>>.
- [NIST.CSWP.04162018]
National Institute of Standards and Technology (NIST), "Framework for Improving Critical Infrastructure Cybersecurity, Version 1.1", NIST NIST.CSWP.04162018, DOI 10.6028/NIST.CSWP.04162018, April 2018, <<http://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.04162018.pdf>>.
- [NIST.SP.800-57pt1r5]
Barker, E B., "Recommendation for key management, part 1 :general", NIST NIST.SP.800-57pt1r5, DOI 10.6028/NIST.SP.800-57pt1r5, 2020, <<https://doi.org/10.6028/NIST.SP.800-57pt1r5>>.
- [RFC3647] Chokhani, S., Ford, W., Sabett, R., Merrill, C., and S. Wu, "Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework", RFC 3647, DOI 10.17487/RFC3647, November 2003, <<https://www.rfc-editor.org/info/rfc3647>>.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <<https://www.rfc-editor.org/rfc/rfc5246>>.
- [RFC5753] Turner, S. and D. Brown, "Use of Elliptic Curve Cryptography (ECC) Algorithms in Cryptographic Message Syntax (CMS)", RFC 5753, DOI 10.17487/RFC5753, January 2010, <<https://www.rfc-editor.org/info/rfc5753>>.
- [RFC7030] Pritikin, M., Ed., Yee, P., Ed., and D. Harkins, Ed., "Enrollment over Secure Transport", RFC 7030, DOI 10.17487/RFC7030, October 2013, <<https://www.rfc-editor.org/info/rfc7030>>.
- [RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, DOI 10.17487/RFC7252, June 2014, <<https://www.rfc-editor.org/info/rfc7252>>.

- [RFC8366] Watsen, K., Richardson, M., Pritikin, M., and T. Eckert, "A Voucher Artifact for Bootstrapping Protocols", RFC 8366, DOI 10.17487/RFC8366, May 2018, <<https://www.rfc-editor.org/info/rfc8366>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/rfc/rfc8446>>.
- [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>>.
- [RFC8572] Watsen, K., Farrer, I., and M. Abrahamsson, "Secure Zero Touch Provisioning (SZTP)", RFC 8572, DOI 10.17487/RFC8572, April 2019, <<https://www.rfc-editor.org/info/rfc8572>>.
- [RFC8649] Housley, R., "Hash Of Root Key Certificate Extension", RFC 8649, DOI 10.17487/RFC8649, August 2019, <<https://www.rfc-editor.org/info/rfc8649>>.
- [RFC8995] Pritikin, M., Richardson, M., Eckert, T., Behringer, M., and K. Watsen, "Bootstrapping Remote Secure Key Infrastructure (BRSKI)", RFC 8995, DOI 10.17487/RFC8995, May 2021, <<https://www.rfc-editor.org/info/rfc8995>>.
- [UNISIG.Subset-137]
UNISIG, "Subset-137; ERTMS/ETCS On-line Key Management FFFIS; V1.0.0", December 2015, <https://www.era.europa.eu/sites/default/files/filesystem/ertms/ccs_tsi_annex_a_-_mandatory_specifications/set_of_specifications_3_etcs_b3_r2_gsm-r_b1/index083_-_subset-137_v100.pdf>.

Appendix A. Example CertReqTemplate

Suppose the server requires that the certTemplate contains

- * the issuer field with a value to be filled in by the EE,
- * the subject field with a common name to be filled in by the EE and two organizational unit fields with given values "myDept" and "myGroup",
- * the publicKey field contains an ECC key on curve secp256r1 or an RSA public key of length 2048,

- * the subjectAltName extension with DNS name "www.myServer.com" and an IP address to be filled in,
- * the keyUsage extension marked critical with the value digitalSignature and keyAgreement, and
- * the extKeyUsage extension with values to be filled in by the EE.

Then the infoValue with certTemplate and keySpec fields returned to the EE will be encoded as follows:

```
SEQUENCE {
  SEQUENCE {
    [3] {
      SEQUENCE {}
    }
    [5] {
      SEQUENCE {
        SET {
          SEQUENCE {
            OBJECT IDENTIFIER commonName (2 5 4 3)
            UTF8String ""
          }
        }
        SET {
          SEQUENCE {
            OBJECT IDENTIFIER organizationalUnitName (2 5 4 11)
            UTF8String "myDept"
          }
        }
        SET {
          SEQUENCE {
            OBJECT IDENTIFIER organizationalUnitName (2 5 4 11)
            UTF8String "myGroup"
          }
        }
      }
    }
  }
  [9] {
    SEQUENCE {
      OBJECT IDENTIFIER subjectAltName (2 5 29 17)
      OCTET STRING, encapsulates {
        SEQUENCE {
          [2] "www.myServer.com"
          [7] ""
        }
      }
    }
  }
}
```

```

    }
    SEQUENCE {
      OBJECT IDENTIFIER keyUsage (2 5 29 15)
      BOOLEAN TRUE
      OCTET STRING, encapsulates {
        BIT STRING 3 unused bits
        "10001"B
      }
    }
  }
  SEQUENCE {
    OBJECT IDENTIFIER extKeyUsage (2 5 29 37)
    OCTET STRING, encapsulates {
      SEQUENCE {}
    }
  }
}
}
SEQUENCE {
  SEQUENCE {
    OBJECT IDENTIFIER algId (1 3 6 1 5 5 7 5 1 11)
    SEQUENCE {
      OBJECT IDENTIFIER ecPublicKey (1 2 840 10045 2 1)
      OBJECT IDENTIFIER secp256r1 (1 2 840 10045 3 1 7)
    }
  }
  SEQUENCE {
    OBJECT IDENTIFIER rsaKeyLen (1 3 6 1 5 5 7 5 1 12)
    INTEGER 2048
  }
}
}
}

```

Appendix B. History of Changes

Note: This appendix will be deleted in the final version of the document.

From version 20 -> 21:

- * Addressed comment from Murray checking each usage of key word "SHOULD" and changing it to "MUST", "MAY", or "should" where needed or adding an explanation how interoperability may be affected (see thread "Murray Kucherawy's No Objection on draft-ietf-lamps-lightweight-cmp-profile-18: (with COMMENT)")
- * Some minor editorial changes

From version 19 -> 20:

- * Addressed comment from John (see thread "[IANA #1261900] expert review for draft-ietf-lamps-lightweight-cmp-profile (cmp)")

From version 18 -> 19:

- * Addressed comment from Murray, moving section 'Convention and Terminology' after Section 1.1 and adding a paragraph on the use of key word "SHOULD", moving section 'Compatibility with Existing CMP Profiles' right before section 'Use of CMP in SZTP and BRSKI Environments', and adding a paragraph to section 'Scope of this Document' also clarifying the use of key word "SHOULD" (see thread "Murray Kucherawy's No Objection on draft-ietf-lamps-lightweight-cmp-profile-18: (with COMMENT)")
- * Updated Section 4.1.6 to reflect the changes to CMP Updates on guidance which CMS key management technique to use with central key management (see thread "CMS: selection of key management technique to use for EnvelopedData") and removed normative language regarding which key management technique to support

From version 17 -> 18:

- * Addressed comment from Paul (see thread "Paul Wouters' Yes on draft-ietf-lamps-lightweight-cmp-profile-16: (with COMMENT)")
- * Updated Section 4.3.4 with one minor correction and one clarification (see thread "Minor change to draft-ietf-lamps-lightweight-cmp-profile-17 on Section 4.3.4 CRL Update Retrieval")

From version 16 -> 17:

- * Addressed comment from Paul (see thread "Paul Wouters' Yes on draft-ietf-lamps-lightweight-cmp-profile-16: (with COMMENT)")
- * Addressed comment from Robert (see thread "Robert Wilton's No Objection on draft-ietf-lamps-lightweight-cmp-profile-16: (with COMMENT)")

From version 15 -> 16:

- * Addressed comment from Warren (see thread "Warren Kumari's No Record on draft-ietf-lamps-lightweight-cmp-profile-15: (with COMMENT)")
- * Addressed comment from Sheng (see thread "Intdir telechat review of draft-ietf-lamps-lightweight-cmp-profile-15")
- * Addressed comment from Niklas (see thread "Iotdir telechat review of draft-ietf-lamps-lightweight-cmp-profile-15")
- * Addressed comment from Erik (see thread "Erik Kline's No Objection on draft-ietf-lamps-lightweight-cmp-profile-15: (with COMMENT)")
- * Streamlined wording in two ASN.1 comments

From version 14 -> 15:

- * Added a reference to HashOfRootKey extension to note in Section 3.3
- * Addressed comment from Joel (see thread "Genart last call review of draft-ietf-lamps-lightweight-cmp-profile-14")
- * Addressed comment from Robert (see thread "Artart last call review of draft-ietf-lamps-lightweight-cmp-profile-14")

From version 13 -> 14:

- * Addressed comments from AD Evaluation (see thread "AD Review of draft-ietf-lamps-lightweight-cmp-profile-13")
- * Added a note to Section 1 informing about rfc4210bis and rfc6712bis activity
- * Added support for constrained PKI entities that can, e.g., only store a hash of a self-signed certificate as trust anchor and require the self-signed certificate to be provided in-line in extraCerts, see Section 3.3 and Section 9
- * Addressed idnits feedback, specifically changing the following RFC reference: RFC3278 -> RFC5753

From version 12 -> 13:

- * Some minor clarifications regarding 'exactly one element' -> 'sequence of one element'
- * Adding authors contact details

From version 11 -> 12:

- * Added a note to Section 4.1.6 to clarify the combination of central key generation with certificate update
- * Updated Section 4.3.4 for clarification that only one CRL per round-trip is requested
- * Updated Section 7.1 to fix a wrong change from the last update in the first two rows of Table 3

From version 10 -> 11:

- * Updated Section 3.2, 3.5, and 3.6.4 to define more clearly signature-based protection as the default and the exception for not protecting error messages as mentioned at IETF 113
- * Streamlined headlines in Section 4.1
- * Updates Section 6.1 and Section 6.2 regarding new well-known path segment for profile labels as discussed during IETF 113
- * Updated Section 7.1. on the support of PKI management operations required for EEs, RAs, and CAs as mentioned at IETF 113

- * Updates Section 8 adding well-known path segments on PKI management operations to be used with HTTP and CoAP
- * Capitalized all headlines

From version 09 -> 10:

- * Resolved some nits reported by I-D nit checker tool
- * Resolve some wording issues

From version 08 -> 09:

- * Updated Section 1.1 and 1.2 and converted Section 2.2 and 2.3 into more detailed tables in Section 7 (see thread "WG Last Call for draft-ietf-lamps-cmp-updates-14 and draft-ietf-lamps-lightweight-cmp-profile-08")
- * Updated Section 3.1 and 4.1.1 making implicitConfirm recommended for ir/cr/pl0cr/kur and providing further recommendations on its use (see thread "certConf - WG Last Call for draft-ietf-lamps-cmp-updates-14 and draft-ietf-lamps-lightweight-cmp-profile-08")
- * Updated Section 4.1.6 adding some clarifications regarding validating the authorization of centrally generated keys
- * Updated Section 4.3.4 adding some clarifications on CRL update retrieval (see thread "CRL update retrieval - WG Last Call for draft-ietf-lamps-cmp-updates-14 and draft-ietf-lamps-lightweight-cmp-profile-08")
- * Updated references to CMP Updates pointing to concrete sections (see thread "CRL update retrieval - WG Last Call for draft-ietf-lamps-cmp-updates-14 and draft-ietf-lamps-lightweight-cmp-profile-08"))
- * Corrected a couple of nits elsewhere

From version 07 -> 08:

- * Updates Section 4.1.6.1. regarding content of the originator and keyEncryptionAlgorithm fields (see thread "AD review of draft-ietf-lamps-cmp-algorithms-07")
- * Rolled back part of the changes on root CA certificate updates in Section 4.3.2 (see thread "Allocation of OIDs for CRL update retrieval (draft-ietf-lamps-cmp-updates-13)")

From version 06 -> 07:

- * Added references to [draft-ietf-netconf-sztp-csr] in new Section 1.5 and Section 4.1.4
- * Added reference to [I-D.ietf-anima-brski-ae] in new Section 1.5 and Section 4.1.1

- * Changed reference in Section 2 to [I-D.ietf-anima-brski-prm] as the PUSH use case is continued to be discussed in this draft after the split of BRSKI-AE
- * Improved note regarding UNISIG Subset-137 in Section 1.6
- * Removed "rootCaCert" in Section 3.1 and updated the structure of the genm request for root CA certificate updates in Section 4.3.2.
- * Simplified handling of sender and recipient nonces in case of delayed delivery in Sections 3.1, 3.5, 4.4, and 5.1.2
- * Changed the order of Sections 4.1.4 and 4.1.5
- * Added reference on RFC 8933 regarding CMS signedAttrs to Section 4.1.6
- * Added Section 4.3.4 on CRL update retrieval
- * Generalized delayed enrollment to delayed delivery in Section 4.4 and 5.1.2, updated the state machine in the introduction of Section 4
- * Updated Section 6 regarding delayed message transfer
- * Changed file name extension from ".PKI" to ".pki", deleted operational path for central key generation, and added an operational path for CRL update retrieval in Sections 6.1 and 6.2
- * Shifted many security considerations to CMP Updates
- * Replaced the term "transport" by "transfer" where appropriate to prevent confusion regarding TCP vs. HTTP and CoAP
- * Various editorial changes and language corrections

From version 05 -> 06:

- * Changed in Section 2.3 the normative requirement in of adding protection to a single message to mandatory and replacing protection to optional
- * Added Section 3.4 specifying generic prerequisites to PKI management operations
- * Added Section 3.5 specifying generic message validation
- * Added Section 3.6 on generic error reporting. This section replaces the former error handling section from Section 4 and 5.
- * Added reference to using hashAlg
- * Updates Section 4.3.2 and Section 4.3.3 to align with CMP Updates
- * Added Section 5.1 specifying the behavior of PKI management entities when responding to requests
- * Reworked Section 5.2.3. on usage of nested messages
- * Updates Section 5.3 on performing PKI management operation on behalf of another entity
- * Updates Section 6.2 on HTTPS transport of CMP messages as discusses at IETF 110 and email thread "I-D Action: draft-ietf-lamps-lightweight-cmp-profile-05.txt"
- * Added CoAP endpoints to Section 6.4
- * Added security considerations on usage of shared secret information
- * Updated the example in Appendix A

- * Added newly registered OIDs to the example in Appendix A
- * Updated new RFC numbers for I-D.ietf-lamps-crmf-update-algs
- * Multiple language corrections, clarifications, and changes in wording

From version 04 -> 05:

- * Changed to XML V3
- * Added algorithm names introduced in CMP Algorithms Section 7.3 to Section 4 of this document
- * Updates Syntax in Section 4.4.3 due to changes made in CMP Updates
- * Deleted the text on HTTP-based discovery as discussed in Section 6.1
- * Updates Appendix A due to change syntax in Section 4.4.3
- * Many clarifications and changes in wording thanks to David's extensive review

From version 03 -> 04:

- * Deleted normative text sections on algorithms and refer to CMP Algorithms and CRMF Algorithm Requirements Update instead
- * Some clarifications and changes in wording

From version 02 -> 03:

- * Updated the interoperability with [UNISIG.Subset-137] in Section 1.4.
- * Changed Section 2.3 to a tabular layout to enhanced readability
- * Added a ToDo to section 3.1 on aligning with the CMP Algorithms draft that will be set up as decided in IETF 108
- * Updated section 4.1.6 to add the AsymmetricKey Package structure to transport a newly generated private key as decided in IETF 108
- * Added a ToDo to section 4.1.7 on required review of the nonce handling in case an offline LRA responds and not forwards the pollReq messages
- * Updated Section 4 due to the definition of the new ITAV OIDs in CMP Updates
- * Updated Section 4.4.4 to utilize controls instead of rsaKeyLen (see thread "dtaft-ietf-lamps-cmp-updates and rsaKeyLen")
- * Deleted the section on definition and discovery of HTTP URIs and copied the text to the HTTP transport section and to CMP Updates section 3.2
- * Added some explanation to Section 5.1.2 and Section 5.1.3 on using nested messages when a protection by the RA is required.
- * Deleted the section on HTTP URI definition and discovery as some content was moved to CMP Updates. The rest of the content was moved back to the HTTP transport section

- * Deleted the ASN.1 module after moving the new OIDs `id-it-caCerts`, `id-it-rootCaKeyUpdate`, and `id-it-certReqTemplate` to CMP Updates
- * Minor changes in wording and addition of some open Todos

From version 01 -> 02:

- * Extend Section 1.6 with regard to conflicts with UNISIG Subset-137.
- * Minor clarifications on `extraCerts` in Section 3.3 and Section 4.1.1.
- * Complete specification of requesting a certificate from a trusted PKI with signature protection in Section 4.1.2.
- * Changed from symmetric key-encryption to password-based key management technique in Section 4.1.6.3 as discussed on the mailing list (see thread "`draft-ietf-lamps-lightweight-cmp-profile-01`, section 5.1.6.1")
- * Changed delayed enrollment described in Section 4.4 from recommended to optional as decided at IETF 107
- * Introduced the new `RootCAKeyUpdate` structure for root CA certificate update in Section 4.3.2 as decided at IETF 107 (also see email thread "`draft-ietf-lamps-lightweight-cmp-profile-01`, section 5.4.3")
- * Extend the description of the `CertReqTemplate` PKI management operation, including an example added in the Appendix. Keep `rsaKeyLen` as a single integer value in Section 4.3.3 as discussed on the mailing list (see thread "`draft-ietf-lamps-lightweight-cmp-profile-01`, section 5.4.4")
- * Deleted Sections "Get certificate management configuration" and "Get enrollment voucher" as decided at IETF 107
- * Complete specification of adding an additional protection by an PKI management entity in Section 5.2.2.
- * Added a section on HTTP URI definition and discovery and extended Section 6.1 on definition and discovery of supported HTTP URIs and content types, add a path for nested messages as specified in Section 5.2.2 and delete the paths for `/getCertMgtConfig` and `/getVoucher`
- * Changed Section 6.4 to address offline transport and added more detailed specification file-based transport of CMP
- * Added a reference to the new I-D of Mohit Sahni on "CoAP Transport for CMPV2" in Section 6.2; thanks to Mohit supporting the effort to ease utilization of CMP
- * Moved the change history to the Appendix
- * Minor changes in wording

From version 00 -> 01:

- * Harmonize terminology with CMP [RFC4210], e.g.,

- transaction, message sequence, exchange, use case -> PKI management operation
- PKI component, (L)RA/CA -> PKI management entity
- * Minor changes in wording

From draft-brockhaus-lamps-lightweight-cmp-profile-03 -> draft-ietf-lamps-lightweight-cmp-profile-00:

- * Changes required to reflect WG adoption
- * Minor changes in wording

From version 02 -> 03:

- * Added a short summary of [RFC4210] Appendix D and E in Section 1.5.
- * Clarified some references to different sections and added some clarification in response to feedback from Michael Richardson and Tomas Gustavsson.
- * Added an additional label to the operational path to address multiple CAs or certificate profiles in Section 6.1.

From version 01 -> 02:

- * Added some clarification on the key management techniques for protection of centrally generated keys in Section 4.1.6.
- * Added some clarifications on the certificates for root CA certificate update in Section 4.3.2.
- * Added a section to specify the usage of nested messages for RAs to add an additional protection for further discussion, see Section 5.2.2.
- * Added a table containing endpoints for HTTP transport in Section 6.1 to simplify addressing PKI management entities.
- * Added some Todos resulting from discussion with Tomas Gustavsson.
- * Minor clarifications and changes in wording.

From version 00 -> 01:

- * Added a section to specify the enrollment with an already trusted PKI for further discussion, see Section 4.1.2.
- * Complete specification of requesting a certificate from a legacy PKI using a PKCS#10 [RFC2986] request in Section 4.1.4.
- * Complete specification of adding central generation of a key pair on behalf of an end entity in Section 4.1.6.
- * Complete specification of handling delayed enrollment due to asynchronous message delivery in Section 4.4.
- * Complete specification of additional support messages, e.g., to update a Root CA certificate or to request an RFC 8366 [RFC8366] voucher, in Section 4.3.

- * Minor changes in wording.

From draft-brockhaus-lamps-industrial-cmp-profile-00 -> draft-brockhaus-lamps-lightweight-cmp-profile-00:

- * Change focus from industrial to more multi-purpose use cases and lightweight CMP profile.
- * Incorporate the omitted confirmation into the header specified in Section 3.1 and described in the standard enrollment use case in Section 4.1.1 due to discussion with Tomas Gustavsson.
- * Change from OPTIONAL to RECOMMENDED for use case 'Revoke another's entities certificate' in Section 5.3.2, because it is regarded as important functionality in many environments to enable the management station to revoke EE certificates.
- * Complete the specification of the revocation message flow in Section 4.2 and Section 5.3.2.
- * The CoAP based transport mechanism and piggybacking of CMP messages on top of other reliable transport protocols is out of scope of this document and would need to be specified in another document.
- * Further minor changes in wording.

Authors' Addresses

Hendrik Brockhaus
Siemens
Werner-von-Siemens-Strasse 1
80333 Munich
Germany
Email: hendrik.brockhaus@siemens.com
URI: <https://www.siemens.com>

David von Oheimb
Siemens
Werner-von-Siemens-Strasse 1
80333 Munich
Germany
Email: david.von.oheimb@siemens.com
URI: <https://www.siemens.com>

Steffen Fries
Siemens AG
Werner-von-Siemens-Strasse 1
80333 Munich
Germany
Email: steffen.fries@siemens.com

URI: <https://www.siemens.com>

LAMPS
Internet-Draft
Updates: 6960 (if approved)
Intended status: Standards Track
Expires: March 14, 2021

M. Sahni, Ed.
Palo Alto Networks
September 10, 2020

OCSP Nonce Extension
draft-ietf-lamps-ocsp-nonce-05

Abstract

This document specifies the updated format of the Nonce extension in the Online Certificate Status Protocol (OCSP) request and response messages. OCSP is used to check the status of a certificate and the Nonce extension is used to cryptographically bind an OCSP response message to a particular OCSP request message. This document updates RFC 6960.

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 March 14, 2021.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Terminology	2
2. OCSP Extensions	2
2.1. Nonce Extension	3
3. Security Considerations	3
3.1. Replay Attack	4
3.2. Nonce Collision	4
4. IANA Considerations	4
5. Changes to Appendix B. of RFC 6960	4
5.1. Changes to Appendix B.1. OCSP in ASN.1 - 1998 Syntax . .	4
5.2. Changes to Appendix B.2 OCSP in ASN.1 - 2008 Syntax . . .	5
6. References	5
6.1. Normative References	5
6.2. Informative References	5
Author's Address	6

1. Introduction

This document updates the usage and format of the Nonce extension used in OCSP request and response messages. This extension was previously defined in section 4.4.1 of [RFC6960]. [RFC6960] does not mention any minimum and maximum length of nonce in the Nonce extension. Lacking limits on the length of nonce in the Nonce extension, an OCSP responders that follow [RFC6960] may be vulnerable to various attacks like Denial of Service attacks [RFC4732], chosen prefix attacks to get a desired signature, and possible evasions using the Nonce extension data. This document specifies a lower limit of 1 and an upper limit of 32 to the length of nonce in the Nonce extension. This document updates [RFC6960].

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. OCSP Extensions

The message format for OCSP request and response is defined in [RFC6960]. [RFC6960] also defines the standard extensions for OCSP messages based on the extension model employed in X.509 version 3

certificates (see [RFC5280]). This document only specifies the new format for Nonce extension and does not change specification of any of the other standard extensions defined in [RFC6960].

2.1. Nonce Extension

This section replaces the entirety of the Section 4.4.1 of [RFC6960] which describes the OCSP Nonce extension.

The nonce cryptographically binds a request and a response to prevent replay attacks. The nonce is included as one of the requestExtensions in requests, while in responses it would be included as one of the responseExtensions. In both the request and the response, the nonce will be identified by the object identifier `id-pkix-ocsp-nonce`, while the `extnValue` is the value of the nonce. If Nonce extension is present then the length of nonce MUST be at least 1 octet and can be up to 32 octets.

A server MUST reject any OCSP request having a nonce in the Nonce extension with length of 0 octets or more than 32 octets with the `malformedRequest` OCSPResponseStatus as described in section 4.2.1 of [RFC6960].

The value of the nonce MUST be generated using a cryptographically strong pseudorandom number generator (see [RFC4086]). The minimum nonce length of 1 octet is defined to provide backward compatibility with older clients that follow [RFC6960]. Newer OCSP clients that support this document MUST use a length of 32 octets for the nonce in Nonce extension. OCSP responders MUST accept lengths of at least 16 octets, and MAY choose to ignore the Nonce extension for requests where the length of the nonce is less than 16 octets

```
id-pkix-ocsp          OBJECT IDENTIFIER ::= { id-ad-ocsp }
id-pkix-ocsp-nonce   OBJECT IDENTIFIER ::= { id-pkix-ocsp 2 }

Nonce ::= OCTET STRING(SIZE(1..32))
```

3. Security Considerations

The security considerations of OCSP, in general, are described in [RFC6960]. During the interval in which the previous OCSP response for a certificate is not expired but the responder has a changed status for that certificate, a copy of that OCSP response can be used to indicate that the status of the certificate is still valid. Including client's Nonce value in the OCSP response makes sure that the response is the latest response from the server and not an old copy.

3.1. Replay Attack

The Nonce extension is used to avoid replay attacks. Since the OCSP responder may choose to not send the Nonce extension in the OCSP response even if the client has sent the Nonce extension in the request [RFC5019], an on-path attacker can intercept the OCSP request and respond with an earlier response from the server without the Nonce extension. This can be mitigated by configuring the server to use a short time interval between the `thisUpdate` and `nextUpdate` fields in the OCSP response.

3.2. Nonce Collision

If the value of nonce used by a client in OCSP request is predictable, then an attacker may prefetch responses with the predicted nonce and can replay them, thus defeating the purpose of using nonce. Therefore the value of Nonce extension in the OCSP request MUST contain cryptographically strong randomness and MUST be freshly generated at the time of creating the OCSP request. Also if the length of nonce is too small e.g. 1 octet then an on-path attacker can prefetch responses with all the possible values of nonce and replay a matching nonce.

4. IANA Considerations

This document does not call for any IANA actions.

5. Changes to Appendix B. of RFC 6960

This section updates the ASN.1 definitions of the OCSP Nonce extension in Appendix B.1 and Appendix B.2 of [RFC6960] The Appendix B.1 defines OCSP using ASN.1 - 1998 Syntax and Appendix B.2 defines OCSP using ASN.1 - 2008 Syntax

5.1. Changes to Appendix B.1. OCSP in ASN.1 - 1998 Syntax

OLD Syntax:

The definition of OCSP Nonce Extension is not provided in Appendix B.1 of [RFC6960] for the ASN.1 - 1998 Syntax.

NEW Syntax:

```
Nonce ::= OCTET STRING(SIZE(1..32))
```

5.2. Changes to Appendix B.2 OCSP in ASN.1 - 2008 Syntax

OLD Syntax:

```
re-ocsp-nonce EXTENSION ::= { SYNTAX OCTET STRING IDENTIFIED
    BY id-pkix-ocsp-nonce }
```

NEW Syntax:

```
re-ocsp-nonce EXTENSION ::= { SYNTAX OCTET STRING(SIZE(1..32))
    IDENTIFIED BY id-pkix-ocsp-nonce }
```

6. References

6.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", DOI 10.17487/RFC8174, RFC 8174, BCP 14, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <<https://www.rfc-editor.org/info/rfc5280>>.
- [RFC6960] Santesson, S., Myers, M., Ankney, R., Malpani, A., Galperin, S., and C. Adams, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP", RFC 6960, DOI 10.17487/RFC6960, June 2013, <<https://www.rfc-editor.org/info/rfc6960>>.

6.2. Informative References

- [RFC4086] Eastlake 3rd, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, DOI 10.17487/RFC4086, June 2005, <<https://www.rfc-editor.org/info/rfc4086>>.

[RFC4732] Handley, M., Ed., Rescorla, E., Ed., and IAB, "Internet Denial-of-Service Considerations", RFC 4732, DOI 10.17487/RFC4732, December 2006, <<https://www.rfc-editor.org/info/rfc4732>>.

[RFC5019] Deacon, A. and R. Hurst, "The Lightweight Online Certificate Status Protocol (OCSP) Profile for High-Volume Environments", RFC 5019, DOI 10.17487/RFC5019, September 2007, <<https://www.rfc-editor.org/info/rfc5019>>.

Author's Address

Mohit Sahni (editor)
Palo Alto Networks
3000 Tannery Way
Santa Clara, CA 95054
US

Email: msahni@paloaltonetworks.com

LAMPS Working Group
Internet-Draft
Updates: 7030 (if approved)
Intended status: Standards Track
Expires: February 12, 2021

M. Richardson
Sandelman Software Works
T. Werner
Siemens
W. Pan
Huawei Technologies
August 11, 2020

Clarification of Enrollment over Secure Transport (EST): transfer
encodings and ASN.1
draft-ietf-lamps-rfc7030est-clarify-10

Abstract

This document updates RFC7030: Enrollment over Secure Transport (EST) to resolve some errata that were reported, and which have proven to cause interoperability issues when RFC7030 was extended.

This document deprecates the specification of "Content-Transfer-Encoding" headers for EST endpoints. This document fixes some syntactical errors in ASN.1 that were present.

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 February 12, 2021.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Terminology	3
3. Changes to EST endpoint processing	3
3.1. Whitespace processing	4
3.2. Changes sections 4 of RFC7030	4
3.2.1. Section 4.1.3	4
3.2.2. Section 4.3.1	4
3.2.3. Section 4.3.2	5
3.2.4. Section 4.4.2	5
3.2.5. Section 4.5.2	5
4. Clarification of ASN.1 for Certificate Attribute set.	6
5. Clarification of error messages for certificate enrollment operations	8
5.1. Updating section 4.2.3: Simple Enroll and Re-enroll Response	8
5.2. Updating section 4.4.2: Server-Side Key Generation Response	8
6. Privacy Considerations	8
7. Security Considerations	9
8. IANA Considerations	9
9. Acknowledgements	9
10. References	9
10.1. Normative References	9
10.2. Informative References	11
Appendix A. ASN.1 Module	12
Authors' Addresses	14

1. Introduction

Enrollment over Secure Transport (EST) is defined in [RFC7030]. The EST specification defines a number of HTTP end points for certificate enrollment and management. The details of the transaction were defined in terms of MIME headers as defined in [RFC2045], rather than in terms of the HTTP protocol as defined in [RFC7230] and [RFC7231].

[RFC2616] and later [RFC7231] Appendix A.5 has text specifically deprecating Content-Transfer-Encoding. However, [RFC7030] incorrectly uses this header.

Any updates to [RFC7030] to bring it inline with HTTP processing risk changing the on-wire protocol in a way that is not backwards compatible. However, reports from implementers suggest that many implementations do not send the Content-Transfer-Encoding, and many of them ignore it. The consequence is that simply deprecating the header would remain compatible with current implementations.

[I-D.ietf-anima-bootstrapping-keyinfra] extends [RFC7030], adding new functionality, and interop testing of the protocol has revealed that unusual processing called out in [RFC7030] causes confusion.

EST is currently specified as part of [IEC62351], and is widely used in Government, Utilities and Financial markets today.

This document therefore revises [RFC7030] to reflect the field reality, deprecating the extraneous field.

This document deals with errata numbers [errata4384], [errata5107], [errata5108], and [errata5904].

This document deals with [errata5107] and [errata5904] in Section 3. [errata5108] is dealt with in Section 5. [errata4384] is closed by correcting the ASN.1 Module in Section 4.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Changes to EST endpoint processing

The [RFC7030] sections 4.1.3 (CA Certificates Response, /cacerts), 4.3.1/4.3.2 (Full CMC, /fullcmc), 4.4.2 (Server-Side Key Generation, /serverkeygen), and 4.5.2 (CSR Attributes, /csrattrs) specify the use of base64 encoding with a Content-Transfer-Encoding for requests and response.

This document updates [RFC7030] to require the POST request and payload response of all endpoints use Base64 encoding as specified in Section 4 of [RFC4648]. In both cases, the Distinguished Encoding Rules (DER) [X.690] are used to produce the input for the Base64 encoding routine. This format is to be used regardless of any Content-Transfer-Encoding header, and any value in such a header MUST be ignored.

3.1. Whitespace processing

Note that "base64" as used in the HTTP [RFC2616] does not permit CR/LF, while the "base64" used in MIME [RFC2045] does. This specification clarifies that despite [RFC2616], that white space including CR, LF, spaces (ASCII 32) and, tabs (ASCII 9) SHOULD be tolerated by receivers. Senders are not required to insert any kind of white space.

3.2. Changes sections 4 of RFC7030

3.2.1. Section 4.1.3

Replace:

A successful response MUST be a certs-only CMC Simple PKI Response, as defined in [RFC5272], containing the certificates described in the following paragraph. The HTTP content-type of "application/pkcs7-mime" is used. The Simple PKI Response is sent with a Content-Transfer-Encoding of "base64" [RFC2045].

with: (RFCEDITOR: maybe artwork is the wrong choice here)

A successful response MUST be a certs-only CMC Simple PKI Response, as defined in [RFC5272], containing the certificates described in the following paragraph. The HTTP content-type of "application/pkcs7-mime" is used. The CMC Simple PKI Response is encoded in base64 [RFC4648].

3.2.2. Section 4.3.1

Replace:

If the HTTP POST to /fullcmc is not a valid Full PKI Request, the server MUST reject the message. The HTTP content-type used is "application/pkcs7-mime" with an smime-type parameter "CMC-request", as specified in [RFC5273]. The body of the message is the binary value of the encoding of the PKI Request with a Content-Transfer-Encoding of "base64" [RFC2045].

with:

If the HTTP POST to /fullcmc is not a valid Full PKI Request, the server MUST reject the message. The HTTP content-type used is "application/pkcs7-mime" with an smime-type parameter "CMC-request", as specified in [RFC5273]. The body of the message is encoded in base64 [RFC4648].

3.2.3. Section 4.3.2

Replace:

The body of the message is the binary value of the encoding of the PKI Response with a Content-Transfer-Encoding of "base64" [RFC2045].

with:

The body of the message is the base64 [RFC4648] encoding of the PKI Response.

3.2.4. Section 4.4.2

Replace:

An "application/pkcs8" part consists of the base64-encoded DER-encoded [X.690] PrivateKeyInfo with a Content-Transfer-Encoding of "base64" [RFC4648].

with:

An "application/pkcs8" part consists of the base64-encoded DER-encoded [X.690] PrivateKeyInfo.

Replace:

In all three additional encryption cases, the EnvelopedData is returned in the response as an "application/pkcs7-mime" part with an smime-type parameter of "server-generated-key" and a Content-Transfer-Encoding of "base64".

with:

In all three additional encryption cases, the EnvelopedData is returned in the response as an "application/pkcs7-mime" part with an smime-type parameter of "server-generated-key". It is base64 encoded [RFC4648].

3.2.5. Section 4.5.2

This section is updated in its entirety in Section 4.

4. Clarification of ASN.1 for Certificate Attribute set.

Section 4.5.2 of [RFC7030] is to be replaced with the following text:

4.5.2 CSR Attributes Response

If locally configured policy for an authenticated EST client indicates a CSR Attributes Response is to be provided, the server response MUST include an HTTP 200 response code. An HTTP response code of 204 or 404 indicates that a CSR Attributes Response is not available. Regardless of the response code, the EST server and CA MAY reject any subsequent enrollment requests for any reason, e.g., incomplete CSR attributes in the request.

Responses to attribute request messages MUST be encoded as the content-type of "application/csrattrs", and are to be "base64" [RFC4648] encoded. The syntax for application/csrattrs body is as follows:

```
CsrAttrs ::= SEQUENCE SIZE (0..MAX) OF AttrOrOID
```

```
AttrOrOID ::= CHOICE {  
    oid      OBJECT IDENTIFIER,  
    attribute Attribute {{AttrSet}} }
```

```
AttrSet ATTRIBUTE ::= { ... }
```

An EST server includes zero or more OIDs or attributes [RFC2986] that it requests the client to use in the certification request. The client MUST ignore any OID or attribute it does not recognize. When the server encodes CSR Attributes as an empty SEQUENCE, it means that the server has no specific additional information it desires in a client certification request (this is functionally equivalent to an HTTP response code of 204 or 404).

If the CA requires a particular cryptographic algorithm or use of a particular signature scheme (e.g., certification of a public key based on a certain elliptic curve, or signing using a certain hash algorithm) it MUST provide that information in the CSR Attribute Response. If an EST server requires the linking of identity and POP information (see Section 3.5), it MUST include the challengePassword OID in the CSR Attributes Response.

The structure of the CSR Attributes Response SHOULD, to the greatest extent possible, reflect the structure of the CSR it is requesting. Requests to use a particular signature scheme (e.g. using a particular hash function) are represented as an OID to be reflected in the SignatureAlgorithm of the CSR. Requests to use a particular

cryptographic algorithm (e.g., certification of a public key based on a certain elliptic curve) are represented as an attribute, to be reflected as the AlgorithmIdentifier of the SubjectPublicKeyInfo, with a type indicating the algorithm and the values indicating the particular parameters specific to the algorithm. Requests for descriptive information from the client are made by an attribute, to be represented as Attributes of the CSR, with a type indicating the [RFC2985] extensionRequest and the values indicating the particular attributes desired to be included in the resulting certificate's extensions.

The sequence is Distinguished Encoding Rules (DER) encoded [X.690] and then base64 encoded (Section 4 of [RFC4648]). The resulting text forms the application/csrattr body, without headers.

For example, if a CA requests a client to submit a certification request containing the challengePassword (indicating that linking of identity and POP information is requested; see Section 3.5), an extensionRequest with the Media Access Control (MAC) address ([RFC2307]) of the client, and to use the secp384r1 elliptic curve and to sign with the SHA384 hash function. Then, it takes the following:

```

OID:          challengePassword (1.2.840.113549.1.9.7)

Attribute:    type = extensionRequest (1.2.840.113549.1.9.14)
              value = macAddress (1.3.6.1.1.1.1.22)

Attribute:    type = id-ecPublicKey (1.2.840.10045.2.1)
              value = secp384r1 (1.3.132.0.34)

OID:          ecdsaWithSHA384 (1.2.840.10045.4.3.3)

```

and encodes them into an ASN.1 SEQUENCE to produce:

```

30 41 06 09 2a 86 48 86 f7 0d 01 09 07 30 12 06 07 2a 86 48 ce 3d
02 01 31 07 06 05 2b 81 04 00 22 30 16 06 09 2a 86 48 86 f7 0d 01
09 0e 31 09 06 07 2b 06 01 01 01 01 16 06 08 2a 86 48 ce 3d 04 03
03

```

and then base64 encodes the resulting ASN.1 SEQUENCE to produce:

```

MEEGCSqGS Ib3DQEJBzASBgcqhkjOPQIBMQcGBSuBBAAiMBYGCSqGS Ib3DQEJDjEJ
BgcrBgEBAQEWBggqhkjOPQDAw==

```

5. Clarification of error messages for certificate enrollment operations

[errata5108] clarifies what format the error messages are to be in. Previously a client might be confused into believing that an error returned with type text/plain was not intended to be an error.

5.1. Updating section 4.2.3: Simple Enroll and Re-enroll Response

Replace:

If the content-type is not set, the response data MUST be a plaintext human-readable error message containing explanatory information describing why the request was rejected (for example, indicating that CSR attributes are incomplete).

with:

If the content-type is not set, the response data MUST be a plaintext human-readable error message containing explanatory information describing why the request was rejected (for example, indicating that CSR attributes are incomplete). Servers MAY use the "text/plain" content-type [RFC2046] for human-readable errors.

5.2. Updating section 4.4.2: Server-Side Key Generation Response

Replace:

If the content-type is not set, the response data MUST be a plaintext human-readable error message.

with:

If the content-type is not set, the response data MUST be a plaintext human-readable error message. Servers MAY use the "text/plain" content-type [RFC2046] for human-readable errors.

6. Privacy Considerations

This document does not disclose any additional identities to either active or passive observer would see with [RFC7030].

7. Security Considerations

This document clarifies an existing security mechanism. It does not create any new protocol mechanism.

All security considerations from [RFC7030] also apply for the clarifications described in this document.

8. IANA Considerations

The ASN.1 module in Appendix A of this document makes use of object identifiers (OIDs). This document requests that IANA register an OID in the SMI Security for PKIX Arc in the Module identifiers subarc (1.3.6.1.5.5.7.0) for the ASN.1 module. The OID for the Asymmetric Decryption Key Identifier (1.2.840.113549.1.9.16.2.54) was previously defined in [RFC7030].

IANA is requested to update the "Reference" column for the Asymmetric Decryption Key Identifier attribute to also include a reference to this document.

9. Acknowledgements

This work was supported by Huawei Technologies.

The ASN.1 Module was assembled by Russ Housley and formatted by Sean Turner. Russ Housley provided editorial review.

10. References

10.1. Normative References

[errata4384]

"EST errata 4384: ASN.1 encoding error", n.d.,
<<https://www.rfc-editor.org/errata/eid4384>>.

[errata5107]

"EST errata 5107: use Content-Transfer-Encoding", n.d.,
<<https://www.rfc-editor.org/errata/eid5107>>.

[errata5108]

"EST errata 5108: use of Content-Type for error message", n.d.,
<<https://www.rfc-editor.org/errata/eid5108>>.

[errata5904]

"EST errata 5904: use Content-Transfer-Encoding", n.d.,
<<https://www.rfc-editor.org/errata/eid5904>>.

- [IEC62351] International Electrotechnical Commission, "Power systems management and associated information exchange - Data and communications security - Part 9: Cyber security key management for power system equipment", ISO/IEC 62351-9:2017, 2017.
- [RFC2045] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, DOI 10.17487/RFC2045, November 1996, <<https://www.rfc-editor.org/info/rfc2045>>.
- [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>>.
- [RFC2986] Nystrom, M. and B. Kaliski, "PKCS #10: Certification Request Syntax Specification Version 1.7", RFC 2986, DOI 10.17487/RFC2986, November 2000, <<https://www.rfc-editor.org/info/rfc2986>>.
- [RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006, <<https://www.rfc-editor.org/info/rfc4648>>.
- [RFC5272] Schaad, J. and M. Myers, "Certificate Management over CMS (CMC)", RFC 5272, DOI 10.17487/RFC5272, June 2008, <<https://www.rfc-editor.org/info/rfc5272>>.
- [RFC5273] Schaad, J. and M. Myers, "Certificate Management over CMS (CMC): Transport Protocols", RFC 5273, DOI 10.17487/RFC5273, June 2008, <<https://www.rfc-editor.org/info/rfc5273>>.
- [RFC5912] Hoffman, P. and J. Schaad, "New ASN.1 Modules for the Public Key Infrastructure Using X.509 (PKIX)", RFC 5912, DOI 10.17487/RFC5912, June 2010, <<https://www.rfc-editor.org/info/rfc5912>>.
- [RFC6268] Schaad, J. and S. Turner, "Additional New ASN.1 Modules for the Cryptographic Message Syntax (CMS) and the Public Key Infrastructure Using X.509 (PKIX)", RFC 6268, DOI 10.17487/RFC6268, July 2011, <<https://www.rfc-editor.org/info/rfc6268>>.

- [RFC7030] Pritikin, M., Ed., Yee, P., Ed., and D. Harkins, Ed., "Enrollment over Secure Transport", RFC 7030, DOI 10.17487/RFC7030, October 2013, <<https://www.rfc-editor.org/info/rfc7030>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [X.680] ITU-T, "Information technology - Abstract Syntax Notation One.", ISO/IEC 8824-1:2002, 2002.
- [X.681] ITU-T, "Information technology - Abstract Syntax Notation One: Information Object Specification.", ISO/IEC 8824-2:2002, 2002.
- [X.682] ITU-T, "Information technology - Abstract Syntax Notation One: Constraint Specification.", ISO/IEC 8824-2:2002, 2002.
- [X.683] ITU-T, "Information technology - Abstract Syntax Notation One: Parameterization of ASN.1 Specifications.", ISO/IEC 8824-2:2002, 2002.
- [X.690] ITU-T, "Information technology - ASN.1 encoding Rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER).", ISO/IEC 8825-1:2002, 2002.

10.2. Informative References

- [I-D.ietf-anima-bootstrapping-keyinfra] Pritikin, M., Richardson, M., Eckert, T., Behringer, M., and K. Watsen, "Bootstrapping Remote Secure Key Infrastructures (BRSKI)", draft-ietf-anima-bootstrapping-keyinfra-43 (work in progress), August 2020.
- [RFC2307] Howard, L., "An Approach for Using LDAP as a Network Information Service", RFC 2307, DOI 10.17487/RFC2307, March 1998, <<https://www.rfc-editor.org/info/rfc2307>>.
- [RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, DOI 10.17487/RFC2616, June 1999, <<https://www.rfc-editor.org/info/rfc2616>>.

- [RFC2985] Nystrom, M. and B. Kaliski, "PKCS #9: Selected Object Classes and Attribute Types Version 2.0", RFC 2985, DOI 10.17487/RFC2985, November 2000, <<https://www.rfc-editor.org/info/rfc2985>>.
- [RFC7230] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, DOI 10.17487/RFC7230, June 2014, <<https://www.rfc-editor.org/info/rfc7230>>.
- [RFC7231] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", RFC 7231, DOI 10.17487/RFC7231, June 2014, <<https://www.rfc-editor.org/info/rfc7231>>.

Appendix A. ASN.1 Module

This annex provides the normative ASN.1 definitions for the structures described in this specification using ASN.1 as defined in [X.680], [X.681], [X.682] and [X.683].

The ASN.1 modules makes imports from the ASN.1 modules in [RFC5912] and [RFC6268].

There is no ASN.1 Module in RFC 7030. This module has been created by combining the lines that are contained in the document body.

```
PKIXEST-2019
  { iso(1) identified-organization(3) dod(6)
    internet(1) security(5) mechanisms(5) pkix(7) id-mod(0)
    id-mod-est-2019(TBD) }

DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-- EXPORTS ALL --

IMPORTS

Attribute
FROM CryptographicMessageSyntax-2010 -- [RFC6268]
  { iso(1) member-body(2) us(840) rsadsi(113549)
    pkcs(1) pkcs-9(9) smime(16) modules(0)
    id-mod-cms-2009(58) }

ATTRIBUTE
FROM PKIX-CommonTypes-2009 -- [RFC5912]
  { iso(1) identified-organization(3) dod(6) internet(1) security(5)
    mechanisms(5) pkix(7) id-mod(0) id-mod-pkixCommon-02(57) } ;

-- CSR Attributes

CsrAttrs ::= SEQUENCE SIZE (0..MAX) OF AttrOrOID

AttrOrOID ::= CHOICE {
  oid          OBJECT IDENTIFIER,
  attribute    Attribute {{AttrSet}} }

AttrSet ATTRIBUTE ::= { ... }

-- Asymmetric Decrypt Key Identifier Attribute

aa-asymmDecryptKeyID ATTRIBUTE ::=
  { TYPE AsymmetricDecryptKeyIdentifier
    IDENTIFIED BY id-aa-asymmDecryptKeyID }

id-aa-asymmDecryptKeyID OBJECT IDENTIFIER ::= { iso(1) member-body(2)
  us(840) rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) aa(2) 54 }

AsymmetricDecryptKeyIdentifier ::= OCTET STRING

END
```

Authors' Addresses

Michael Richardson
Sandelman Software Works

Email: mcr+ietf@sandelman.ca

Thomas Werner
Siemens

Email: thomas-werner@siemens.com

Wei Pan
Huawei Technologies

Email: william.panwei@huawei.com