

lamps
Internet-Draft
Intended status: Informational
Expires: 26 August 2021

D.K. Gillmor
ACLU
22 February 2021

Guidance on End-to-End E-mail Security
draft-dkg-lamps-e2e-mail-guidance-01

Abstract

End-to-end cryptographic protections for e-mail messages can provide useful security. However, the standards for providing cryptographic protection are extremely flexible. That flexibility can trap users and cause surprising failures. This document offers guidance for mail user agent implementers that need to compose or interpret e-mail messages with end-to-end cryptographic protection. It provides a useful set of vocabulary as well as suggestions to avoid common failures.

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 26 August 2021.

Copyright Notice

Copyright (c) 2021 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 | 3 |
| 1.1. Requirements Language | 3 |
| 1.2. Terminology | 4 |
| 1.2.1. Structural Headers | 4 |
| 2. Usability | 4 |
| 2.1. Simplicity | 5 |
| 2.2. E-mail Users Want a Familiar Experience | 5 |
| 2.3. Warning About Failure vs. Announcing Success | 6 |
| 3. Types of Protection | 7 |
| 4. Cryptographic MIME Message Structure | 7 |
| 4.1. Cryptographic Layers | 7 |
| 4.1.1. S/MIME Cryptographic Layers | 7 |
| 4.1.2. PGP/MIME Cryptographic Layers | 8 |
| 4.2. Cryptographic Envelope | 9 |
| 4.3. Cryptographic Payload | 10 |
| 4.4. Types of Cryptographic Envelope | 10 |
| 4.4.1. Simple Cryptographic Envelopes | 10 |
| 4.4.2. Multilayer Cryptographic Envelopes | 10 |
| 4.5. Errant Cryptographic Layers | 10 |
| 4.5.1. Mailing List Wrapping | 11 |
| 4.5.2. A Baroque Example | 11 |
| 5. Message Composition | 12 |
| 5.1. Message Composition Algorithm | 12 |
| 5.2. Encryption Outside, Signature Inside | 13 |
| 5.3. Avoid Offering Encrypted-only Messages | 13 |
| 5.4. Composing a Reply Message | 14 |
| 6. Message Interpretation | 14 |
| 6.1. Rendering Well-formed Messages | 15 |
| 6.2. Errant Cryptographic Layers | 15 |
| 6.2.1. Errant Signing Layer | 15 |
| 6.2.2. Errant Encryption Layer | 17 |
| 6.3. Forwarded Messages with Cryptographic Protection | 17 |
| 6.4. Signature failures | 18 |
| 7. Certificate Management | 19 |
| 7.1. Peer Certificates | 19 |
| 7.1.1. Cert Discovery from Incoming Messages | 19 |
| 7.1.2. Certificate Directories | 19 |

| | |
|---|----|
| 7.1.3. Peer Certificate Selection | 19 |
| 7.1.4. Checking for Revocation | 20 |
| 7.2. Local Certificates | 20 |
| 7.2.1. Getting a Certificate for the User | 20 |
| 7.2.2. Local Certificate Maintenance | 21 |
| 7.2.3. Shipping Certificates in Outbound Messages | 21 |
| 7.3. Certificate Authorities | 22 |
| 8. Common Pitfalls and Guidelines | 22 |
| 9. IANA Considerations | 22 |
| 10. Security Considerations | 23 |
| 11. Document Considerations | 23 |
| 11.1. Document History | 23 |
| 11.1.1. Substantive changes from -00 to -01 | 23 |
| 12. Acknowledgements | 23 |
| 13. References | 23 |
| 13.1. Normative References | 23 |
| 13.2. Informative References | 24 |
| Appendix A. Test Vectors | 25 |
| Author's Address | 25 |

1. Introduction

E-mail end-to-end security using S/MIME ([RFC8551]) and PGP/MIME ([RFC3156]) cryptographic standards can provide integrity, authentication and confidentiality to MIME ([RFC4289]) e-mail messages.

However, there are many ways that a receiving mail user agent can misinterpret or accidentally break these security guarantees (e.g., [EFAIL]).

A mail user agent that interprets a message with end-to-end cryptographic protections needs to do so defensively, staying alert to different ways that these protections can be bypassed by mangling (either malicious or accidental) or a failed user experience.

A mail user agent that generates a message with end-to-end cryptographic protections should be aware of these defensive interpretation strategies, and should compose any new outbound message conservatively if they want the protections to remain intact.

1.1. 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] and [RFC8174]) when, and only when, they appear in all capitals, as shown here.

1.2. Terminology

For the purposes of this document, we define the following concepts:

- * `_MUA_` is short for Mail User Agent; an e-mail client.
- * `_Protection_` of message data refers to cryptographic encryption and/or signatures, providing confidentiality, authenticity, and/or integrity.
- * `_Cryptographic Layer_`, `_Cryptographic Envelope_`, `_Cryptographic Payload_`, and `_Errant Cryptographic Layer_` are defined in Section 4
- * A `_well-formed_` e-mail message with cryptographic protection has both a `_Cryptographic Envelope_` and a `_Cryptographic Payload_`.
- * `_Structural Headers_` are documented in Section 1.2.1.

1.2.1. Structural Headers

A message header whose name begins with "Content-" is referred to in this document as a "structural" header.

These headers indicate something about the specific MIME part they are attached to, and cannot be transferred or copied to other parts without endangering the readability of the message.

This includes (but is not limited to):

- * "Content-Type"
- * "Content-Transfer-Encoding"
- * "Content-Disposition"

FIXME: are there any non-"Content-*" headers we should consider as structural?

2. Usability

Any MUA that enables its user to transition from unprotected messages to messages with end-to-end cryptographic protection needs to consider how the user understands this transition. That said, the primary goal of the user of an MUA is communication -- so interface elements that get in the way of communication should be avoided where possible.

Furthermore, it is likely is that the user will continue to encounter unprotected messages, and may need to send unprotected messages (for example, if a given recipient cannot handle cryptographic protections). This means that the MUA needs to provide the user with some guidance, so that they understand what protections any given message or conversation has. But the user should not be overwhelmed with choices or presented with unactionable information.

2.1. Simplicity

The end user (the operator of the MUA) is unlikely to understand complex end-to-end cryptographic protections on any e-mail message, so keep it simple.

For clarity to the user, any cryptographic protections should apply to the message as a whole, not just to some subparts.

This is true for message composition: the standard message composition user interface of an MUA should offer minimal controls which indicate which types of protection to apply to the new message as a whole.

This is also true for message interpretation: the standard message rendering user interface of an MUA should offer a minimal, clear indicator about the end-to-end cryptographic status of the message as a whole.

2.2. E-mail Users Want a Familiar Experience

A person communicating over the Internet today often has many options for reaching their desired correspondent, including web-based bulletin boards, contact forms, and instant messaging services.

E-mail offers a few distinctions from these other systems, most notably features like:

- * Ubiquity: Most correspondents will have an e-mail address, while not everyone is present on every alternate messaging service,
- * Federation: interaction between users on distinct domains who have not agreed on a common communications provider is still possible, and
- * User Control: the user can interact with the e-mail system using a MUA of their choosing, including automation and other control over their preferred and/or customized workflow.

Other systems (like some popular instant messaging applications, such as WhatsApp and Signal Private Messenger) offer built-in end-to-end cryptographic protections by default, which are simpler for the user to understand. ("All the messages I see on Signal are confidential and integrity-protected" is a clean user story)

A user of e-mail is likely using e-mail instead of other systems because of the distinctions outlined above. When adding end-to-end cryptographic protection to an e-mail endpoint, care should be taken not to negate any of the distinct features of e-mail as a whole. If these features are violated to provide end-to-end crypto, the user may just as well choose one of the other systems that don't have the drawbacks that e-mail has. Implementers should try to provide end-to-end protections that retain the familiar experience of e-mail itself.

Furthermore, an e-mail user is likely to regularly interact with other e-mail correspondents who cannot handle or produce end-to-end cryptographic protections. Care should be taken that enabling cryptography in a MUA does not inadvertently limit the ability of the user to interact with legacy correspondents.

2.3. Warning About Failure vs. Announcing Success

Moving the web from http to https offers useful historical similarities to adding end-to-end encryption to e-mail.

In particular, the indicators of what is "secure" vs. "insecure" for web browsers have changed over time. For example, years ago the default experience was http, and https sites were flagged with "secure" indicators like a lock icon. In 2018, some browsers reversed that process by downplaying https, and instead visibly marking http as "not secure" (see [chrome-indicators]).

By analogy, when the user of a MUA first enables end-to-end cryptographic protection, it's likely that they will want to see which messages have protection. But a user whose e-mail communications are entirely end-to-end protected might instead want to know which messages do not have the expected protections.

Note also that some messages are expected to be confidential, but other messages are expected to be public -- the types of protection (see Section 3) that apply to each particular message will be different. And the types of protection that are expected to be present in any context might differ (for example, by sender, by thread, or by date).

It is out of scope for this document to define expectations about protections for any given message, but an implementer who cares about usable experience should be deliberate and judicious about the expectations their interface assumes that the user has in a given context.

3. Types of Protection

A given message might be:

- * signed,
- * encrypted,
- * both signed and encrypted, or
- * none of the above.

Given that many e-mail messages offer no cryptographic protections, the user needs to be able to detect which protections are present for any given message.

4. Cryptographic MIME Message Structure

Implementations use the structure of an e-mail message to protect the headers. This section establishes some conventions about how to think about message structure.

4.1. Cryptographic Layers

"Cryptographic Layer" refers to a MIME substructure that supplies some cryptographic protections to an internal MIME subtree. The internal subtree is known as the "protected part" though of course it may itself be a multipart object.

In the diagrams below, " (DOWNWARDS ARROW FROM BAR, U+21A7) indicates "decrypts to", and " (DOWNWARDS WHITE ARROW, U+21E9) indicates "unwraps to".

4.1.1. S/MIME Cryptographic Layers

For S/MIME [RFC8551], there are four forms of Cryptographic Layers: multipart/signed, PKCS#7 signed-data, PKCS7 enveloped-data, PKCS7 authEnveloped-data.

4.1.1.1. S/MIME Multipart Signed Cryptographic Layer

```
multipart/signed; protocol="application/pkcs7-signature"  
  [protected part]  
  application/pkcs7-signature
```

This MIME layer offers authentication and integrity.

4.1.1.2. S/MIME PKCS7 signed-data Cryptographic Layer

```
application/pkcs7-mime; smime-type="signed-data"  
  (unwraps to)  
  [protected part]
```

This MIME layer offers authentication and integrity.

4.1.1.3. S/MIME PKCS7 enveloped-data Cryptographic Layer

```
application/pkcs7-mime; smime-type="enveloped-data"  
  (decrypts to)  
  [protected part]
```

This MIME layer offers confidentiality.

4.1.1.4. S/MIME PKCS7 authEnveloped-data Cryptographic Layer

```
application/pkcs7-mime; smime-type="authEnveloped-data"  
  (decrypts to)  
  [protected part]
```

This MIME layer offers confidentiality and integrity.

Note that "enveloped-data" (Section 4.1.1.3) and "authEnveloped-data" (Section 4.1.1.4) have identical message structure and semantics. The only difference between the two is ciphertext malleability.

The examples in this document only include "enveloped-data", but the implications for that layer apply to "authEnveloped-data" as well.

4.1.1.5. PKCS7 Compression is NOT a Cryptographic Layer

The Cryptographic Message Syntax (CMS) provides a MIME compression layer ("smime-type="compressed-data"), as defined in [RFC3274]. While the compression layer is technically a part of CMS, it is not considered a Cryptographic Layer for the purposes of this document.

4.1.2. PGP/MIME Cryptographic Layers

For PGP/MIME [RFC3156] there are two forms of Cryptographic Layers, signing and encryption.

4.1.2.1. PGP/MIME Signing Cryptographic Layer (multipart/signed)

```
multipart/signed; protocol="application/pgp-signature"  
  [protected part]  
  application/pgp-signature
```

This MIME layer offers authenticity and integrity.

4.1.2.2. PGP/MIME Encryption Cryptographic Layer (multipart/encrypted)

```
multipart/encrypted  
  application/pgp-encrypted  
  application/octet-stream  
    (decrypts to)  
  [protected part]
```

This MIME layer can offer any of:

- * confidentiality (via a Symmetrically Encrypted Data Packet, see Section 5.7 of [RFC4880]; a MUA MUST NOT generate this form due to ciphertext malleability)
- * confidentiality and integrity (via a Symmetrically Encrypted Integrity Protected Data Packet (SEIPD), see section 5.13 of [RFC4880]), or
- * confidentiality, integrity, and authenticity all together (by including an OpenPGP Signature Packet within the SEIPD).

4.2. Cryptographic Envelope

The Cryptographic Envelope is the largest contiguous set of Cryptographic Layers of an e-mail message starting with the outermost MIME type (that is, with the Content-Type of the message itself).

If the Content-Type of the message itself is not a Cryptographic Layer, then the message has no cryptographic envelope.

"Contiguous" in the definition above indicates that if a Cryptographic Layer is the protected part of another Cryptographic Layer, the layers together comprise a single Cryptographic Envelope.

Note that if a non-Cryptographic Layer intervenes, all Cryptographic Layers within the non-Cryptographic Layer are not part of the Cryptographic Envelope. They are Errant Cryptographic Layers (see Section 4.5).

Note also that the ordering of the Cryptographic Layers implies different cryptographic properties. A signed-then-encrypted message is different than an encrypted-then-signed message. See Section 5.2.

4.3. Cryptographic Payload

The Cryptographic Payload of a message is the first non-Cryptographic Layer -- the "protected part" -- within the Cryptographic Envelope.

4.4. Types of Cryptographic Envelope

4.4.1. Simple Cryptographic Envelopes

As described above, if the "protected part" identified in the section above is not itself a Cryptographic Layer, that part is the Cryptographic Payload.

If the application wants to generate a message that is both encrypted and signed, it MAY use the simple MIME structure from Section 4.1.2.2 by ensuring that the [RFC4880] Encrypted Message within the "application/octet-stream" part contains an [RFC4880] Signed Message (the final option described in Section 4.1.2.2).

4.4.2. Multilayer Cryptographic Envelopes

It is possible to construct a Cryptographic Envelope consisting of multiple layers with either S/MIME or PGP/MIME , for example using the following structure:

```
A application/pkcs7-mime; smime-type="enveloped-data"
B   (decrypts to)
C application/pkcs7-mime; smime-type="signed-data"
D   (unwraps to)
E   [protected part]
```

When handling such a message, the properties of the Cryptographic Envelope are derived from the series "A", "C".

As noted in Section 4.4.1, PGP/MIME applications also have a simpler MIME construction available with the same cryptographic properties.

4.5. Errant Cryptographic Layers

Due to confusion, malice, or well-intentioned tampering, a message may contain a Cryptographic Layer that is not part of the Cryptographic Envelope. Such a layer is an Errant Cryptographic Layer.

An Errant Cryptographic Layer SHOULD NOT contribute to the message's overall cryptographic state.

Guidance for dealing with Errant Cryptographic Layers can be found in Section 6.2.

4.5.1. Mailing List Wrapping

Some mailing list software will re-wrap a well-formed signed message before re-sending to add a footer, resulting in the following structure seen by recipients of the e-mail:

```
H multipart/mixed
I  multipart/signed
J  text/plain
K  application/pgp-signature
L  text/plain
```

In this message, "L" is the footer added by the mailing list. "I" is now an Errant Cryptographic Layer.

Note that this message has no Cryptographic Envelope at all.

It is NOT RECOMMENDED to produce e-mail messages with this structure, because the data in part "L" may appear to the user as though it were part of "J", though they have different cryptographic properties. In particular, if the user believes that the message is signed, but cannot distinguish "L" from "J" then the author of "L" can effectively tamper with content of the signed message, breaking the user's expectation of integrity and authenticity.

4.5.2. A Baroque Example

Consider a message with the following overcomplicated structure:

```
M multipart/encrypted
N  application/pgp-encrypted
O  application/octet-stream
P    (decrypts to)
Q  multipart/signed
R  multipart/mixed
S  multipart/signed
T  text/plain
U  application/pgp-signature
V  text/plain
W  application/pgp-signature
```

The 3 Cryptographic Layers in such a message are rooted in parts "M", "Q", and "S". But the Cryptographic Envelope of the message consists only of the properties derived from the series "M", "Q". The Cryptographic Payload of the message is part "R". Part "S" is an Errant Cryptographic Layer.

Note that this message has both a Cryptographic Envelope and an Errant Cryptographic Layer.

It is NOT RECOMMENDED to generate messages with such complicated structures. Even if a receiving MUA can parse this structure properly, it is nearly impossible to render in a way that the user can reason about the cryptographic properties of part "T" compared to part "V".

5. Message Composition

This section describes the ideal composition of an e-mail message with end-to-end cryptographic protection. A message composed with this form is most likely to achieve its end-to-end security goals.

5.1. Message Composition Algorithm

This section roughly describes the steps that a MUA should use to compose a cryptographically-protected message that has a proper cryptographic envelope and payload.

The message composition algorithm takes three parameters:

- * "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 headers present (see Section 1.2.1).
- * "origheaders": the intended non-structural headers 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.
- * "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 "origbody", yielding MIME tree "output"
- * For each header name and value "(h,v)" in "origheaders":
 - Add header "h" of "output" with value "v"
- * Return "output"

5.2. Encryption Outside, Signature Inside

Users expect any message that is both signed and encrypted to be signed inside the encryption, and not the other way around.

Putting the signature inside the encryption has two advantages:

- * The details of the signature remain confidential, visible only to the parties capable of decryption.
- * Any mail transport agent that modifies the message is unlikely to be able to accidentally break the signature.

A MUA SHOULD NOT generate an encrypted and signed message where the only signature is outside the encryption.

5.3. Avoid Offering Encrypted-only Messages

When generating an e-mail, the user has options about what forms of end-to-end cryptographic protections to apply to it.

In some cases, offering any end-to-end cryptographic protection is harmful: it may confuse the recipient and offer no benefit.

In other cases, signing a message is useful (authenticity and integrity are desirable) but encryption is either impossible (for example, if the sender does not know how to encrypt to all recipients) or meaningless (for example, an e-mail message to a mailing list that is intended to be published to a public archive).

In other cases, full end-to-end confidentiality, authenticity, and integrity are desirable.

It is unclear what the use case is for an e-mail message with end-to-end confidentiality but without authenticity or integrity.

A reasonable MUA will keep its message composition interface simple, so when presenting the user with a choice of cryptographic protection, it SHOULD offer no more than three choices:

- * no end-to-end cryptographic protection
- * signing-only
- * signed and encrypted

5.4. Composing a Reply Message

When replying to a message, most MUAs compose an initial draft of the reply that contains quoted text from the original message. A responsible MUA will take precautions to avoid leaking the cleartext of an encrypted message in such a reply.

If the original message was end-to-end encrypted, the replying MUA MUST either:

- * compose the reply with end-to-end encryption, or
- * avoid including quoted text from the original message.

In general, MUAs SHOULD prefer the first option: to compose an encrypted reply. This is what users expect.

However, in some circumstances, the replying MUA cannot compose an encrypted reply. For example, the MUA might not have a valid, unexpired, encryption-capable certificate for all recipients. This can also happen during composition when a user adds a new recipient into the reply, or manually toggles the cryptographic protections to remove encryption.

In this circumstance, the composing MUA SHOULD strip the quoted text from the original message.

Note additional nuance about replies to malformed messages that contain encryption in Section 6.2.2.1.

6. Message Interpretation

Despite the best efforts of well-intentioned senders to create e-mail messages with well-formed end-to-end cryptographic protection, receiving MUAs will inevitably encounter some messages with malformed end-to-end cryptographic protection.

This section offers guidance on dealing with both well-formed and malformed messages containing Cryptographic Layers.

6.1. Rendering Well-formed Messages

A message is well-formed when it has a Cryptographic Envelope, a Cryptographic Payload, and no Errant Cryptographic Layers. Rendering a well-formed message is straightforward.

The receiving MUA should evaluate and summarize the cryptographic properties of the Cryptographic Envelope, and display that status to the user in a secure, strictly-controlled part of the UI. In particular, the part of the UI used to render the cryptographic summary of the message MUST NOT be spoofable, modifiable, or otherwise controllable by the received message itself.

Aside from this cryptographic summary, the message itself should be rendered as though the Cryptographic Payload is the body of the message. The Cryptographic Layers themselves SHOULD not be rendered otherwise.

6.2. Errant Cryptographic Layers

If an incoming message has any Errant Cryptographic Layers, the interpreting MUA SHOULD ignore those layers when rendering the cryptographic summary of the message to the user.

6.2.1. Errant Signing Layer

When rendering a message with an Errant Cryptographic Layer that provides authenticity and integrity (via signatures), the message should be rendered by replacing the Cryptographic layer with the part it encloses.

For example, a message with this structure:

```
A multipart/mixed
B  text/plain
C  multipart/signed
D  image/jpeg
E  application/pgp-signature
F  text/plain
```

Should be rendered identically to this:

```
A multipart/mixed
B  text/plain
D  image/jpeg
F  text/plain
```

In such a situation, an MUA SHOULD NOT indicate in the cryptographic summary that the message is signed.

6.2.1.1. Exception: Mailing List Footers

The use case described in Section 4.5.1 is common enough in some contexts, that a MUA MAY decide to handle it as a special exception.

If the MUA determines that the message comes from a mailing list (it has a "List-ID" header), and it has a structure that appends a footer to a signing-only Cryptographic Layer with a valid signature, such as:

```
H multipart/mixed
I  multipart/signed
J  [protected part, may be arbitrary MIME subtree]
K  application/{pgp,pkcs7}-signature
L  [footer, typically text/plain]
```

or:

```
H multipart/mixed
I  application/pkcs7-mime; smime-type="signed-data"
   (unwraps to)
J  [protected part, may be an arbitrary MIME subtree]
L  [footer, typically text/plain]
```

Then, the MUA MAY indicate to the user that this is a signed message that has been wrapped by the mailing list.

In this case, the MUA MUST distinguish the footer (part "L") from the protected part (part "J") when rendering any information about the signature.

One way to do this is to offer the user two different views of the message: the "mailing list" view, which hides any cryptographic summary but shows the footer:

```
Cryptographic Protections: none
H multipart/mixed
J  [protected part, may be arbitrary MIME subtree]
L  [footer, typically text/plain]
```

or the "sender's view", which shows the cryptographic summary but hides the footer:

```
Cryptographic Protections: signed [details from part I]
J [protected part, may be arbitrary MIME subtree]
```


6.2.2. Errant Encryption Layer

An MUA may encounter a message with an Errant Cryptographic Layer that offers confidentiality (encryption), and the MUA is capable of decrypting it.

The user wants to be able to see the contents of any message that they receive, so an MUA in this situation SHOULD decrypt the part.

In this case, though, the MUA MUST NOT indicate in the message's cryptographic summary that the message itself was encrypted. Such an indication could be taken to mean that other (non-encrypted) parts of the message arrived with cryptographic confidentiality.

6.2.2.1. Replying to a Message with an Errant Encryption Layer

Note that there is an asymmetry here between rendering and replying to a message with an Errant Encryption Layer.

When rendering, the MUA does not indicate that the message was encrypted, even if some subpart of it was decrypted for rendering.

But when composing a reply that contains quoted text from the decrypted subpart, the reply message SHOULD be marked for encryption, as noted in {#composing-reply}.

Alternately, if the reply message cannot be encrypted (or if the user elects to not encrypt the reply), the composed reply MUST NOT include any material from the decrypted subpart.

6.3. Forwarded Messages with Cryptographic Protection

An incoming e-mail message may include an attached forwarded message, typically as a MIME subpart with "Content-Type: message/rfc822" ([RFC5322]) or "Content-Type: message/global" ([RFC5355]).

Regardless of the cryptographic protections and structure of the incoming message, the internal forwarded message may have its own Cryptographic Envelope.

The Cryptographic Layers that are part of the Cryptographic Envelope of the forwarded message are not Errant Cryptographic Layers of the surrounding message -- they are simply layers that apply to the forwarded message itself.

The rendering MUA MUST NOT conflate the cryptographic protections of the forwarded message with the cryptographic protections of the incoming message.

The rendering MUA MAY render a cryptographic summary of the protections afforded to the forwarded message by its own Cryptographic Envelope, as long as that rendering is unambiguously tied to the forwarded message itself.

6.4. Signature failures

A cryptographic signature may fail in multiple ways. A receiving MUA that discovers a failed signature should treat the message as though the signature did not exist. This is similar to the standard guidance for about failed DKIM signatures (see section 6.1 of [RFC6376]).

A MUA SHOULD NOT render a message with a failed signature as more dangerous or more dubious than a comparable message without any signature at all.

A MUA that encounters an encrypted-and-signed message where the signature is invalid SHOULD treat the message the same way that it would treat a message that is encryption-only.

Some different ways that a signature may be invalid on a given message:

- * the signature is not cryptographically valid (the math fails).
- * the signature relies on suspect cryptographic primitives (e.g. over a legacy digest algorithm, or was made by a weak key, e.g., 1024-bit R.SA)
- * the signature is made by a certificate which the receiving MUA does not have access to.
- * the certificate that made the signature was revoked.
- * the certificate that made the signature was expired at the time that the signature was made.
- * the certificate that made the signature does not correspond to the author of the message. (for X.509, there is no subjectAltName of type RFC822Name whose value matches an e-mail address found in "From:" or "Sender:")
- * the certificate that made the signature was not issued by an authority that the MUA user is willing to rely on for certifying the sender's e-mail address.

- * the signature indicates that it was made at a time much before or much after from the date of the message itself.

A valid signature must pass all these tests, but of course invalid signatures may be invalid in more than one of the ways listed above.

7. Certificate Management

A cryptographically-capable MUA typically maintains knowledge about certificates for the user's own account(s), as well as certificates for the peers that it communicates with.

7.1. Peer Certificates

Most certificates that a cryptographically-capable MUA will use will be certificates belonging to the parties that the user communicates with through the MUA. This section discusses how to manage the certificates that belong to such a peer.

The MUA will need to be able to discover X.509 certificates for each peer, cache them, and select among them when composing an encrypted message.

7.1.1. Cert Discovery from Incoming Messages

TODO: incoming PKCS#7 messages tend to have a bundle of certificates in them. How should these certs be handled?

TODO: point to Autocrypt certificate discovery mechanism

TODO: point to OpenPGP embedded certificate subpacket proposal

TODO: compare mechanisms, explain where each case is useful.

7.1.2. Certificate Directories

Some MUAs may have the capability to look up peer certificates in a directory.

TODO: more information here about X.509 directories -- LDAP?

TODO: mention WKD for OpenPGP certificates?

7.1.3. Peer Certificate Selection

When composing an encrypted message, the MUA needs to select a certificate for each recipient that is capable of encryption.

To select such a certificate for a given destination e-mail address, the MUA should look through all of its known certificates and verify that all of the conditions below are met:

- * The certificate must be valid, not expired or revoked.
- * It must have a subjectAltName of type rFC822Name whose contents exactly match the destination address.
- * The algorithm OID in the certificate's SPKI is known to the MUA and capable of encryption. Examples include (TODO: need OIDs)
 - RSA, with keyUsage present and the "key encipherment" bit set
 - EC Public Key, with keyUsage present and the "key agreement" bit set
 - EC DH, with keyUsage present and the "key agreement" bit set
- * If extendedKeyUsage is present, it contains at least one of the following OIDs: e-mail protection, anyExtendedKeyUsage.

TODO: If OID is EC Public Key and keyUsage is absent, what should happen?

TODO: what if multiple certificates meet all of these criteria for a given recipient?

7.1.4. Checking for Revocation

TODO: discuss how/when to check for peer certificate revocation

TODO: privacy concerns: what information leaks to whom when checking peer cert revocations?

7.2. Local Certificates

The MUA also needs to know about one or more certificates associated with the user's e-mail account. It is typically expected to have access to the secret key material associated with the public keys in those certificates.

7.2.1. Getting a Certificate for the User

TODO: mention ACME SMIME?

TODO: mention automatic self-signed certs e.g. OpenPGP?

TODO: SHOULD generate secret key material locally, and MUST NOT accept secret key material from an untrusted third party as the basis for the user's certificate.

7.2.2. Local Certificate Maintenance

The MUA should warn the user when/if:

- * The user's own certificate set does not include a valid, unexpired encryption-capable X.509 certificate, and a valid, unexpired signature-capable X.509 certificate.
- * Any of the user's own certificates is due to expire soon (TODO: what is "soon"?)
- * Any of the user's own certificates does not match the e-mail address associated with the user's account.
- * Any of the user's own certificates does not have a keyUsage section
- * Any of the user's own certificates does not contain an extendedKeyUsage extension

TODO: how does the MUA do better than warning in the cases above? What can the MUA actually do here to fix problems before they happen?

TODO: discuss how/when to check for own certificate revocation, and what to do if it (or any intermediate certificate authority) is found to be revoked.

7.2.3. Shipping Certificates in Outbound Messages

TODO: What certificates should the MUA include in an outbound message so that peers can discover them?

- * local signing certificate so that signature can be validated
- * local encryption-capable certificate(s) so that incoming messages can be encrypted.
- * On an encrypted message to multiple recipients, the encryption-capable peer certs of the other recipients (to enable "reply all")?
- * intermediate certificates to chain all of the above to some set of root authorities?

7.3. Certificate Authorities

TODO: how should the MUA select root certificate authorities?

TODO: should the MUA cache intermediate CAs?

TODO: should the MUA share such a cache with other PKI clients (e.g., web browsers)? Are there distinctions between a CA for S/MIME and for the web?

8. Common Pitfalls and Guidelines

This section highlights a few "pitfalls" and guidelines based on these discussions and lessons learned.

FIXME: some possible additional commentary on:

- * indexing and search of encrypted messages
- * managing access to cryptographic secret keys that require user interaction
- * secure deletion
- * inline PGP, ugh
- * storage of composed/sent messages
- * encrypt-to-self during composition
- * cached signature validation
- * interaction between encryption and Bcc
- * aggregated cryptographic status of threads/conversations ?
- * Draft messages
- * copies to the Sent folder

9. IANA Considerations

MAYBE: provide an indicator in the IANA header registry for which headers are "structural" ? This is probably unnecessary.

10. Security Considerations

This entire document addresses security considerations about end-to-end cryptographic protections for e-mail messages.

11. Document Considerations

[RFC Editor: please remove this section before publication]

This document is currently edited as markdown. Minor editorial changes can be suggested via merge requests at <https://gitlab.com/dkg/e2e-mail-guidance> or by e-mail to the author. Please direct all significant commentary to the public IETF LAMPS mailing list: spasm@ietf.org

11.1. Document History

11.1.1. Substantive changes from -00 to -01

- * consideration of success/failure indicators for usability
- * clarify extendedKeyUsage and keyUsage algorithm-specific details
- * initial section on certificate management
- * added more TODO items

12. Acknowledgements

The set of constructs and recommendations in this document are derived from discussions with many different implementers, including Alexey Melnikov, Bernie Hoeneisen, Bjarni Runar Einarsson, David Bremner, Deb Cooley, Holger Krekel, Jameson Rollins, Jonathan Hammell, juga, Patrick Brunschwig, Santosh Chokhani, and Vincent Breitmoser.

13. References

13.1. 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>>.

- [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/info/rfc3156>>.
- [RFC4289] Freed, N. and J. Klensin, "Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures", BCP 13, RFC 4289, DOI 10.17487/RFC4289, December 2005, <<https://www.rfc-editor.org/info/rfc4289>>.
- [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>>.
- [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>>.

13.2. Informative References

- [chrome-indicators] Schechter, E., "Evolving Chrome's security indicators", May 2018, <<https://blog.chromium.org/2018/05/evolving-chromes-security-indicators.html>>.
- [EFAIL] "EFAIL", n.d., <<https://efail.de>>.
- [I-D.draft-bre-openpgp-samples-01] Einarsson, B. R., "juga", and D. K. Gillmor, "OpenPGP Example Keys and Certificates", Work in Progress, Internet-Draft, draft-bre-openpgp-samples-01, 20 December 2019, <<https://www.ietf.org/archive/id/draft-bre-openpgp-samples-01.txt>>.
- [I-D.draft-dkg-lamps-samples-05] Gillmor, D. K., "S/MIME Example Keys and Certificates", Work in Progress, Internet-Draft, draft-dkg-lamps-samples-05, 18 February 2021, <<https://www.ietf.org/archive/id/draft-dkg-lamps-samples-05.txt>>.
- [RFC3274] Gutmann, P., "Compressed Data Content Type for Cryptographic Message Syntax (CMS)", RFC 3274, DOI 10.17487/RFC3274, June 2002, <<https://www.rfc-editor.org/info/rfc3274>>.

- [RFC4880] Callas, J., Donnerhackle, L., Finney, H., Shaw, D., and R. Thayer, "OpenPGP Message Format", RFC 4880, DOI 10.17487/RFC4880, November 2007, <<https://www.rfc-editor.org/info/rfc4880>>.
- [RFC5322] Resnick, P., Ed., "Internet Message Format", RFC 5322, DOI 10.17487/RFC5322, October 2008, <<https://www.rfc-editor.org/info/rfc5322>>.
- [RFC5355] Stillman, M., Ed., Gopal, R., Guttman, E., Sengodan, S., and M. Holdrege, "Threats Introduced by Reliable Server Pooling (RSerPool) and Requirements for Security in Response to Threats", RFC 5355, DOI 10.17487/RFC5355, September 2008, <<https://www.rfc-editor.org/info/rfc5355>>.
- [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/info/rfc6376>>.

Appendix A. Test Vectors

FIXME: This document should contain examples of well-formed and malformed messages using cryptographic key material and certificates from [I-D.draft-bre-openpgp-samples-01] and [I-D.draft-dkg-lamps-samples-05].

It may also include example renderings of these messages.

Author's Address

Daniel Kahn Gillmor
American Civil Liberties Union
125 Broad St.
New York, NY, 10004
United States of America

Email: dkg@fifthhorseman.net

LAMPS Working Group
Internet-Draft
Updates: 4210 (if approved)
Intended status: Standards Track
Expires: 14 November 2022

H. Brockhaus, Ed.
H. Aschauer
Siemens
M. Ounsworth
J. Gray
Entrust
13 May 2022

Certificate Management Protocol (CMP) Algorithms
draft-ietf-lamps-cmp-algorithms-13

Abstract

This document describes the conventions for using several cryptographic algorithms with the Certificate Management Protocol (CMP). CMP is used to enroll and further manage the lifecycle of X.509 certificates. This document also updates the algorithm use profile from RFC 4210 Appendix D.2.

Status of This Memo

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

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

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

This Internet-Draft will expire on 14 November 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. Terminology | 3 |
| 2. Message Digest Algorithms | 3 |
| 2.1. SHA2 | 3 |
| 2.2. SHAKE | 4 |
| 3. Signature Algorithms | 5 |
| 3.1. RSA | 5 |
| 3.2. ECDSA | 6 |
| 3.3. EdDSA | 7 |
| 4. Key Management Algorithms | 7 |
| 4.1. Key Agreement Algorithms | 8 |
| 4.1.1. Diffie-Hellman | 8 |
| 4.1.2. ECDH | 8 |
| 4.2. Key Transport Algorithms | 10 |
| 4.2.1. RSA | 10 |
| 4.3. Symmetric Key-Encryption Algorithms | 11 |
| 4.3.1. AES Key Wrap | 11 |
| 4.4. Key Derivation Algorithms | 12 |
| 4.4.1. PBKDF2 | 12 |
| 5. Content Encryption Algorithms | 12 |
| 5.1. AES-CBC | 13 |
| 6. Message Authentication Code Algorithms | 13 |
| 6.1. Password-Based MAC | 13 |
| 6.1.1. PasswordBasedMac | 14 |
| 6.1.2. PBMAC1 | 14 |
| 6.2. Symmetric Key-Based MAC | 14 |
| 6.2.1. SHA2-Based HMAC | 15 |
| 6.2.2. AES-GMAC | 15 |
| 6.2.3. SHAKE-Based KMAC | 16 |
| 7. Algorithm Use Profiles | 16 |
| 7.1. Algorithm Profile for RFC 4210 PKI Management Message Profiles | 19 |
| 7.2. Algorithm Profile for Lightweight CMP Profile | 21 |
| 8. IANA Considerations | 22 |
| 9. Security Considerations | 23 |
| 10. Acknowledgements | 24 |
| 11. Normative References | 24 |

| | |
|--|----|
| 12. Informative References | 28 |
| Appendix A. History of Changes | 29 |
| Authors' Addresses | 32 |

1. Introduction

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.

In the following sections ASN.1 values and types are used to indicate where algorithm identifier and output values are provided. These ASN.1 values and types are defined in CMP [RFC4210], CRMF [RFC4211], CMP Updates [I-D.ietf-lamps-cmp-updates], or CMS [RFC5652].

2. Message Digest Algorithms

This section provides references to object identifiers and conventions to be employed by CMP implementations that support SHA2 or SHAKE message digest algorithms.

Digest algorithm identifiers are located in:

- * hashAlg field of OOBCertHash and CertStatus
- * owf field of Challenge, PBMPParameter, and DHBMPParameter
- * digestAlgorithms field of SignedData
- * digestAlgorithm field of SignerInfo

Digest values are located in:

- * hashVal field of OOBCertHash
- * certHash field of CertStatus
- * witness field of Challenge

In addition, digest values are input to signature algorithms.

2.1. SHA2

The SHA2 algorithm family is defined in FIPS Pub 180-4 [NIST.FIPS.180-4].

The message digest algorithms SHA-224, SHA-256, SHA-384, and SHA-512 are identified by the following OIDs:

```
id-sha224 OBJECT IDENTIFIER ::= { joint-iso-itu-t(2) country(16)
    us(840) organization(1) gov(101) csor(3) nistalgorithm(4)
    hashalgs(2) 4 }
id-sha256 OBJECT IDENTIFIER ::= { joint-iso-itu-t(2) country(16)
    us(840) organization(1) gov(101) csor(3) nistalgorithm(4)
    hashalgs(2) 1 }
id-sha384 OBJECT IDENTIFIER ::= { joint-iso-itu-t(2) country(16)
    us(840) organization(1) gov(101) csor(3) nistalgorithm(4)
    hashalgs(2) 2 }
id-sha512 OBJECT IDENTIFIER ::= { joint-iso-itu-t(2) country(16)
    us(840) organization(1) gov(101) csor(3) nistalgorithm(4)
    hashalgs(2) 3 }
```

Specific conventions to be considered are specified in RFC 5754 Section 2 [RFC5754].

2.2. SHAKE

The SHA-3 family of hash functions is defined in FIPS Pub 202 [NIST.FIPS.202] and includes fixed output length variants SHA3-224, SHA3-256, SHA3-384, and SHA3-512, as well as extendable-output functions (SHAKEs) SHAKE128 and SHAKE256. Currently SHAKE128 and SHAKE256 are the only members of the SHA3-family which are specified for use in X.509 certificates [RFC8692] and CMS [RFC8702] as one-way hash function for use with RSASSA-PSS and ECDSA.

SHAKE is an extendable-output function and FIPS Pub 202 [NIST.FIPS.202] prohibits using SHAKE as general-purpose hash function. When SHAKE is used in CMP as a message digest algorithm, the output length MUST be 256 bits for SHAKE128 and 512 bits for SHAKE256.

The message digest algorithms SHAKE128 and SHAKE256 are identified by the following OIDs:

```
id-shake128 OBJECT IDENTIFIER ::= { joint-iso-itu-t(2) country(16)
    us(840) organization(1) gov(101) csor(3) nistAlgorithm(4)
    hashalgs(2) 11 }
id-shake256 OBJECT IDENTIFIER ::= { joint-iso-itu-t(2) country(16)
    us(840) organization(1) gov(101) csor(3) nistAlgorithm(4)
    hashalgs(2) 12 }
```

Specific conventions to be considered are specified in RFC 8702 Section 3.1 [RFC8702].

3. Signature Algorithms

This section provides references to object identifiers and conventions to be employed by CMP implementations that support RSA, ECDSA, or EdDSA signature algorithms.

The signature algorithm is referred to as MSG_SIG_ALG in Section 7.2, RFC 4210 Appendix D and E [RFC4210], and in the Lightweight CMP Profile [I-D.ietf-lamps-lightweight-cmp-profile].

Signature algorithm identifiers are located in:

- * protectionAlg field of PKIHeader
- * algorithmIdentifier field of POPOSigningKey
- * signatureAlgorithm field of CertificationRequest, SignKeyPairTypes, and SignerInfo

Signature values are located in:

- * protection field of PKIMessage
- * signature field of POPOSigningKey
- * signature field of CertificationRequest and SignerInfo

3.1. RSA

The RSA (RSASSA-PSS and PKCS#1 version 1.5) signature algorithm is defined in RFC 8017 [RFC8017].

The algorithm identifier for RSASSA-PSS signatures used with SHA2 message digest algorithms is identified by the following OID:

```
id-RSASSA-PSS OBJECT IDENTIFIER ::= { iso(1) member-body(2)
  us(840) rsadsi(113549) pkcs(1) pkcs-1(1) 10 }
```

Specific conventions to be considered are specified in RFC 4056 [RFC4056].

The signature algorithm RSASSA-PSS used with SHAKE message digest algorithms are identified by the following OIDs:

```
id-RSASSA-PSS-SHAKE128 OBJECT IDENTIFIER ::= { iso(1)
  identified-organization(3) dod(6) internet(1) security(5)
  mechanisms(5) pkix(7) algorithms(6) 30 }
id-RSASSA-PSS-SHAKE256 OBJECT IDENTIFIER ::= { iso(1)
  identified-organization(3) dod(6) internet(1) security(5)
  mechanisms(5) pkix(7) algorithms(6) 31 }
```

Specific conventions to be considered are specified in RFC 8702 Section 3.2.1 [RFC8702].

The signature algorithm PKCS#1 version 1.5 used with SHA2 message digest algorithms is identified by the following OIDs:

```
sha224WithRSAEncryption OBJECT IDENTIFIER ::= { iso(1)
  member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-1(1) 14 }
sha256WithRSAEncryption OBJECT IDENTIFIER ::= { iso(1)
  member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-1(1) 11 }
sha384WithRSAEncryption OBJECT IDENTIFIER ::= { iso(1)
  member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-1(1) 12 }
sha512WithRSAEncryption OBJECT IDENTIFIER ::= { iso(1)
  member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-1(1) 13 }
```

Specific conventions to be considered are specified in RFC 5754 Section 3.2 [RFC5754].

3.2. ECDSA

The ECDSA signature algorithm is defined in FIPS Pub 186-4 [NIST.FIPS.186-4].

The signature algorithm ECDSA used with SHA2 message digest algorithms is identified by the following OIDs:

```
ecdsa-with-SHA224 OBJECT IDENTIFIER ::= { iso(1) member-body(2)
  us(840) ansi-X9-62(10045) signatures(4) ecdsa-with-SHA2(3) 1 }
ecdsa-with-SHA256 OBJECT IDENTIFIER ::= { iso(1) member-body(2)
  us(840) ansi-X9-62(10045) signatures(4) ecdsa-with-SHA2(3) 2 }
ecdsa-with-SHA384 OBJECT IDENTIFIER ::= { iso(1) member-body(2)
  us(840) ansi-X9-62(10045) signatures(4) ecdsa-with-SHA2(3) 3 }
ecdsa-with-SHA512 OBJECT IDENTIFIER ::= { iso(1) member-body(2)
  us(840) ansi-X9-62(10045) signatures(4) ecdsa-with-SHA2(3) 4 }
```

As specified in RFC 5480 [RFC5480] the NIST-recommended SECP curves are identified by the following OIDs:

```
secp192r1 OBJECT IDENTIFIER ::= { iso(1) member-body(2)
  us(840) ansi-X9-62(10045) curves(3) prime(1) 1 }
secp224r1 OBJECT IDENTIFIER ::= { iso(1)
  identified-organization(3) certicom(132) curve(0) 33 }
secp256r1 OBJECT IDENTIFIER ::= { iso(1) member-body(2)
  us(840) ansi-X9-62(10045) curves(3) prime(1) 7 }
secp384r1 OBJECT IDENTIFIER ::= { iso(1)
  identified-organization(3) certicom(132) curve(0) 34 }
secp521r1 OBJECT IDENTIFIER ::= { iso(1)
  identified-organization(3) certicom(132) curve(0) 35 }
```

Specific conventions to be considered are specified in RFC 5754 Section 3.3 [RFC5754].

The signature algorithm ECDSA used with SHAKE message digest algorithms are identified by the following OIDs:

```
id-ecdsa-with-shake128 OBJECT IDENTIFIER ::= { iso(1)
  identified-organization(3) dod(6) internet(1) security(5)
  mechanisms(5) pkix(7) algorithms(6) 32 }
id-ecdsa-with-shake256 OBJECT IDENTIFIER ::= { iso(1)
  identified-organization(3) dod(6) internet(1) security(5)
  mechanisms(5) pkix(7) algorithms(6) 33 }
```

Specific conventions to be considered are specified in RFC 8702 Section 3.2.2 [RFC8702].

3.3. EdDSA

The EdDSA signature algorithm is defined in RFC 8032 Section 3.3 [RFC8032] and FIPS Pub 186-5 (Draft) [NIST.FIPS.186-5].

The signature algorithm Ed25519 that MUST be used with SHA-512 message digest algorithms is identified by the following OIDs:

```
id-Ed25519 OBJECT IDENTIFIER ::= { iso(1)
  identified-organization(3) thawte(101) 112 }
```

The signature algorithm Ed448 that MUST be used with SHAKE256 message digest algorithms is identified by the following OIDs:

```
id-Ed448 OBJECT IDENTIFIER ::= { iso(1)
  identified-organization(3) thawte(101) 113 }
```

Specific conventions to be considered are specified in RFC 8419 [RFC8419].

Note: The hash algorithm used to calculate the certHash in certConf messages MUST be SHA512 if the certificate to be confirmed has been signed using Ed25519 and SHAKE256 with d=512 if signed using Ed448.

4. Key Management Algorithms

CMP utilizes the following general key management techniques: key agreement, key transport, and passwords.

CRMF [RFC4211] and CMP Updates [I-D.ietf-lamps-cmp-updates] promotes the use of CMS [RFC5652] EnvelopedData by deprecating the use of EncryptedValue.

4.1. Key Agreement Algorithms

The key agreement algorithm is referred to as `PROT_ENC_ALG` in RFC 4210 Appendix D and E [RFC4210] and as `KM_KA_ALG` in the Lightweight CMP Profile [I-D.ietf-lamps-lightweight-cmp-profile], as well as in Section 7.

Key agreement algorithms are only used in CMP when using CMS [RFC5652] EnvelopedData together with the key agreement key management technique. When a key agreement algorithm is used, a key-encryption algorithm (Section 4.3) is needed next to the content-encryption algorithm (Section 5).

Key agreement algorithm identifiers are located in:

- * `keyEncryptionAlgorithm` field of `KeyAgreeRecipientInfo`

Key wrap algorithm identifiers are located in:

- * `KeyWrapAlgorithm` parameters within `keyEncryptionAlgorithm` field of `KeyAgreeRecipientInfo`

Wrapped content-encryption keys are located in:

- * `encryptedKey` field of `RecipientEncryptedKeys`

4.1.1. Diffie-Hellman

Diffie-Hellman key agreement is defined in RFC 2631 [RFC2631] and SHALL be used in the ephemeral-static as specified in RFC 3370 [RFC3370]. Static-static variants SHALL NOT be used.

The Diffie-Hellman key agreement algorithm is identified by the following OID:

```
id-alg-ESDH OBJECT IDENTIFIER ::= { iso(1) member-body(2)
    us(840) rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) alg(3) 5 }
```

Specific conventions to be considered are specified in RFC 3370 Section 4.1 [RFC3370].

4.1.2. ECDH

Elliptic Curve Diffie-Hellman (ECDH) key agreement is defined in RFC 5753 [RFC5753] and SHALL be used in the ephemeral-static variant as specified in RFC 5753 [RFC5753] or the 1-Pass ECMQV variant as specified in RFC 5753 [RFC5753]. Static-static variants SHALL NOT be used.

The ECDH key agreement algorithm used together with NIST-recommended SECP curves are identified by the following OIDs:

```
dhSinglePass-stdDH-sha224kdf-scheme OBJECT IDENTIFIER ::= { iso(1)
    identified-organization(3) certicom(132) schemes(1) 11(11) 0 }
dhSinglePass-stdDH-sha256kdf-scheme OBJECT IDENTIFIER ::= { iso(1)
    identified-organization(3) certicom(132) schemes(1) 11(11) 1 }
dhSinglePass-stdDH-sha384kdf-scheme OBJECT IDENTIFIER ::= { iso(1)
    identified-organization(3) certicom(132) schemes(1) 11(11) 2 }
dhSinglePass-stdDH-sha512kdf-scheme OBJECT IDENTIFIER ::= { iso(1)
    identified-organization(3) certicom(132) schemes(1) 11(11) 3 }
dhSinglePass-cofactorDH-sha224kdf-scheme OBJECT IDENTIFIER ::= {
    iso(1) identified-organization(3) certicom(132) schemes(1)
    14(14) 0 }
dhSinglePass-cofactorDH-sha256kdf-scheme OBJECT IDENTIFIER ::= {
    iso(1) identified-organization(3) certicom(132) schemes(1)
    14(14) 1 }
dhSinglePass-cofactorDH-sha384kdf-scheme OBJECT IDENTIFIER ::= {
    iso(1) identified-organization(3) certicom(132) schemes(1)
    14(14) 2 }
dhSinglePass-cofactorDH-sha512kdf-scheme OBJECT IDENTIFIER ::= {
    iso(1) identified-organization(3) certicom(132) schemes(1)
    14(14) 3 }
mqvSinglePass-sha224kdf-scheme OBJECT IDENTIFIER ::= { iso(1)
    identified-organization(3) certicom(132) schemes(1) 15(15) 0 }
mqvSinglePass-sha256kdf-scheme OBJECT IDENTIFIER ::= { iso(1)
    identified-organization(3) certicom(132) schemes(1) 15(15) 1 }
mqvSinglePass-sha384kdf-scheme OBJECT IDENTIFIER ::= { iso(1)
    identified-organization(3) certicom(132) schemes(1) 15(15) 2 }
mqvSinglePass-sha512kdf-scheme OBJECT IDENTIFIER ::= { iso(1)
    identified-organization(3) certicom(132) schemes(1) 15(15) 3 }
```

As specified in RFC 5480 [RFC5480] the NIST-recommended SECP curves are identified by the following OIDs:

```
secp192r1 OBJECT IDENTIFIER ::= { iso(1) member-body(2)
    us(840) ansi-X9-62(10045) curves(3) prime(1) 1 }
secp224r1 OBJECT IDENTIFIER ::= { iso(1)
    identified-organization(3) certicom(132) curve(0) 33 }
secp256r1 OBJECT IDENTIFIER ::= { iso(1) member-body(2)
    us(840) ansi-X9-62(10045) curves(3) prime(1) 7 }
secp384r1 OBJECT IDENTIFIER ::= { iso(1)
    identified-organization(3) certicom(132) curve(0) 34 }
secp521r1 OBJECT IDENTIFIER ::= { iso(1)
    identified-organization(3) certicom(132) curve(0) 35 }
```

Specific conventions to be considered are specified in RFC 5753 [RFC5753].

The ECDH key agreement algorithm used together with curve25519 or curve448 are identified by the following OIDs:

```
id-X25519 OBJECT IDENTIFIER ::= { iso(1)
  identified-organization(3) thawte(101) 110 }
id-X448 OBJECT IDENTIFIER ::= { iso(1)
  identified-organization(3) thawte(101) 111 }
```

Specific conventions to be considered are specified in RFC 8418 [RFC8418].

4.2. Key Transport Algorithms

The key transport algorithm is also referred to as PROT_ENC_ALG in RFC 4210 Appendix D and E [RFC4210] and as KM_KL_ALG in the Lightweight CMP Profile [I-D.ietf-lamps-lightweight-cmp-profile], as well as in Section 7.

Key transport algorithms are only used in CMP when using CMS [RFC5652] EnvelopedData together with the key transport key management technique.

Key transport algorithm identifiers are located in:

- * keyEncryptionAlgorithm field of KeyTransRecipientInfo

Key transport encrypted content-encryption keys are located in:

- * encryptedKey field of KeyTransRecipientInfo

4.2.1. RSA

The RSA key transport algorithm is the RSA encryption scheme defined in RFC 8017 [RFC8017].

The algorithm identifier for RSA (PKCS #1 v1.5) is:

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

The algorithm identifier for RSAES-OAEP is:

```
id-RSAES-OAEP OBJECT IDENTIFIER ::= { iso(1) member-body(2)
  us(840) rsadsi(113549) pkcs(1) pkcs-1(1) 7 }
```

Further conventions to be considered for PKCS #1 v1.5 are specified in RFC 3370 Section 4.2.1 [RFC3370] and for RSAES-OAEP in RFC 3560 [RFC3560].

4.3. Symmetric Key-Encryption Algorithms

The symmetric key-encryption algorithm is also referred to as `KM_KW_ALG` in Section 7.2 and in the Lightweight CMP Profile [I-D.ietf-lamps-lightweight-cmp-profile].

As symmetric key-encryption key management technique is not used by CMP, the symmetric key-encryption algorithm is only needed when using the key agreement or password-based key management technique with CMS [RFC5652] `EnvelopedData`.

Key wrap algorithm identifiers are located in:

- * `parameters` field of the `KeyEncryptionAlgorithmIdentifier` of `KeyAgreeRecipientInfo` and `PasswordRecipientInfo`

Wrapped content-encryption keys are located in:

- * `encryptedKey` field of `RecipientEncryptedKeys` (for key agreement) and `PasswordRecipientInfo` (for password-based key management)

4.3.1. AES Key Wrap

The AES encryption algorithm is defined in FIPS Pub 197 [NIST.FIPS.197] and the key wrapping is defined in RFC 3394 [RFC3394].

AES key encryption has the algorithm identifier:

```
id-aes128-wrap OBJECT IDENTIFIER ::= { joint-iso-itu-t(2)
    country(16) us(840) organization(1) gov(101) csor(3)
    nistAlgorithm(4) aes(1) 5 }
id-aes192-wrap OBJECT IDENTIFIER ::= { joint-iso-itu-t(2)
    country(16) us(840) organization(1) gov(101) csor(3)
    nistAlgorithm(4) aes(1) 25 }
id-aes256-wrap OBJECT IDENTIFIER ::= { joint-iso-itu-t(2)
    country(16) us(840) organization(1) gov(101) csor(3)
    nistAlgorithm(4) aes(1) 45 }
```

The underlying encryption functions for the key wrap and content-encryption algorithms (as specified in Section 5) and the key sizes for the two algorithms MUST be the same (e.g., AES-128 key wrap algorithm with AES-128 content-encryption algorithm), see also RFC 8551 [RFC8551].

Further conventions to be considered for AES key wrap are specified in RFC 3394 Section 2.2 [RFC3394] and RFC 3565 Section 2.3.2 [RFC3565].

4.4. Key Derivation Algorithms

The key derivation algorithm is also referred to as KM_KD_ALG in Section 7.2 and in the Lightweight CMP Profile [I-D.ietf-lamps-lightweight-cmp-profile].

Key derivation algorithms are only used in CMP when using CMS [RFC5652] EnvelopedData together with password-based key management technique.

Key derivation algorithm identifiers are located in:

- * keyDerivationAlgorithm field of PasswordRecipientInfo

When using the password-based key management technique with EnvelopedData as specified in CMP Updates together with message authentication code (MAC)-based PKIProtection, the salt for the password-based MAC and KDF must be chosen independently to ensure usage of independent symmetric keys.

4.4.1. PBKDF2

The password-based key derivation function 2 (PBKDF2) is defined in RFC 8018 [RFC8018].

Password-based key derivation function 2 has the algorithm identifier:

```
id-PBKDF2 OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
  rsadsi(113549) pkcs(1) pkcs-5(5) 12 }
```

Further conventions to be considered for PBKDF2 are specified in RFC 3370 Section 4.4.1 [RFC3370] and RFC 8018 Section 5.2 [RFC8018].

5. Content Encryption Algorithms

The content encryption algorithm is also referred to as PROT_SYM_ALG in Section 7, RFC 4210 Appendix D and E [RFC4210], and the Lightweight CMP Profile [I-D.ietf-lamps-lightweight-cmp-profile].

Content encryption algorithms are only used in CMP when using CMS [RFC5652] EnvelopedData to transport a signed private key package in case of central key generation or key archiving, a certificate to facilitate implicit proof-of-possession, or a revocation passphrase in encrypted form.

Content encryption algorithm identifiers are located in:

- * contentEncryptionAlgorithm field of EncryptedContentInfo

Encrypted content is located in:

- * encryptedContent field of EncryptedContentInfo

5.1. AES-CBC

The AES encryption algorithm is defined in FIPS Pub 197 [NIST.FIPS.197].

AES-CBC content encryption has the algorithm identifier:

```
id-aes128-CBC OBJECT IDENTIFIER ::= { joint-iso-itu-t(2)
  country(16) us(840) organization(1) gov(101) csor(3)
  nistAlgorithm(4) aes(1) 2 }
id-aes192-CBC OBJECT IDENTIFIER ::= { joint-iso-itu-t(2)
  country(16) us(840) organization(1) gov(101) csor(3)
  nistAlgorithm(4) aes(1) 22 }
id-aes256-CBC OBJECT IDENTIFIER ::= { joint-iso-itu-t(2)
  country(16) us(840) organization(1) gov(101) csor(3)
  nistAlgorithm(4) aes(1) 42 }
```

Specific conventions to be considered for AES-CBC content encryption are specified in RFC 3565 [RFC3565].

6. Message Authentication Code Algorithms

The message authentication code (MAC) is either used for shared secret-based CMP message protection or together with the password-based key derivation function (PBKDF2).

The message authentication code algorithm is also referred to as MSG_MAC_ALG in Section 7, RFC 4210 Appendix D and E [RFC4210], and the Lightweight CMP Profile [I-D.ietf-lamps-lightweight-cmp-profile].

6.1. Password-Based MAC

Password-based message authentication code (MAC) algorithms combine the derivation of a symmetric key from a password or other shared secret information and a symmetric key-based MAC function as specified in Section 6.2 using this derived key.

Message authentication code algorithm identifiers are located in:

- * protectionAlg field of PKIHeader

Message authentication code values are located in:

- * PKIProtection field of PKIMessage

6.1.1. PasswordBasedMac

The PasswordBasedMac algorithm is defined in RFC 4210 Section 5.1.3.1 [RFC4210], RFC 4211 Section 4.4 [RFC4211], and Algorithm Requirements Update to the Internet X.509 Public Key Infrastructure Certificate Request Message Format (CRMF) [RFC9045].

The PasswordBasedMac algorithm is identified by the following OID:

```
id-PasswordBasedMac OBJECT IDENTIFIER ::= { iso(1) member-body(2)
  us(840) nt(113533) nsn(7) algorithms(66) 13 }
```

Further conventions to be considered for password-based MAC are specified in RFC 4210 Section 5.1.3.1 [RFC4210], RFC 4211 Section 4.4 [RFC4211], and Algorithm Requirements Update to the Internet X.509 Public Key Infrastructure Certificate Request Message Format (CRMF) [RFC9045].

6.1.2. PBMAC1

The Password-Based Message Authentication Code 1 (PBMAC1) is defined in RFC 8018 [RFC8018]. PBMAC1 combines a password-based key derivation function like PBKDF2 (Section 4.4.1) with an underlying symmetric key-based message authentication scheme.

PBMAC1 has the following OID:

```
id-PBMAC1 OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
  rsadsi(113549) pkcs(1) pkcs-5(5) 14 }
```

Specific conventions to be considered for PBMAC1 are specified in RFC 8018 Section 7.1 and A.5 [RFC8018].

6.2. Symmetric Key-Based MAC

Symmetric key-based message authentication code (MAC) algorithms are used for deriving the symmetric encryption key when using PBKDF2 as described in Section 4.4.1 as well as with Password-based MAC as described in Section 6.1.

Message authentication code algorithm identifiers are located in:

- * protectionAlg field of PKIHeader
- * messageAuthScheme field of PBMAC1
- * mac field of PBMPParameter
- * prf field of PBKDF2-params

Message authentication code values are located in:

* PKIProtection field of PKIMessage

6.2.1. SHA2-Based HMAC

The HMAC algorithm is defined in RFC 2104 [RFC2104] and FIPS Pub 198-1 [NIST.FIPS.198-1].

The HMAC algorithm used with SHA2 message digest algorithms is identified by the following OIDs:

```
id-hmacWithSHA224 OBJECT IDENTIFIER ::= { iso(1) member-body(2)
  us(840) rsadsi(113549) digestAlgorithm(2) 8 }
id-hmacWithSHA256 OBJECT IDENTIFIER ::= { iso(1) member-body(2)
  us(840) rsadsi(113549) digestAlgorithm(2) 9 }
id-hmacWithSHA384 OBJECT IDENTIFIER ::= { iso(1) member-body(2)
  us(840) rsadsi(113549) digestAlgorithm(2) 10 }
id-hmacWithSHA512 OBJECT IDENTIFIER ::= { iso(1) member-body(2)
  us(840) rsadsi(113549) digestAlgorithm(2) 11 }
```

Specific conventions to be considered for SHA2-based HMAC are specified in RFC 4231 Section 3.1 [RFC4231].

6.2.2. AES-GMAC

The AES-GMAC algorithm is defined in FIPS Pub 197 [NIST.FIPS.197] and NIST SP 800-38d [NIST.SP.800-38d].

Note: AES-GMAC MUST NOT be used twice with the same parameter set, especially the same nonce. Therefore, it MUST NOT be used together with PBKDF2. When using it with PBMAC1 it MUST be ensured that AES-GMAC is only used as message authentication scheme and not for the key derivation function PBKDF2.

The AES-GMAC algorithm is identified by the following OIDs:

```
id-aes128-GMAC OBJECT IDENTIFIER ::= { joint-iso-itu-t(2)
  country(16) us(840) organization(1) gov(101) csor(3)
  nistAlgorithm(4) aes(1) 9 }
id-aes192-GMAC OBJECT IDENTIFIER ::= { joint-iso-itu-t(2)
  country(16) us(840) organization(1) gov(101) csor(3)
  nistAlgorithm(4) aes(1) 29 }
id-aes256-GMAC OBJECT IDENTIFIER ::= { joint-iso-itu-t(2)
  country(16) us(840) organization(1) gov(101) csor(3)
  nistAlgorithm(4) aes(1) 49 }
```


Specific conventions to be considered for AES-GMAC are specified in RFC 9044 [RFC9044].

6.2.3. SHAKE-Based KMAC

The KMAC algorithm is defined in RFC 8702 [RFC8702] and FIPS SP 800-185 [NIST.SP.800-185].

The SHAKE-based KMAC algorithm is identified by the following OIDs:

```
id-KmacWithSHAKE128 OBJECT IDENTIFIER ::= { joint-iso-itu-t(2)
  country(16) us(840) organization(1) gov(101) csor(3)
  nistAlgorithm(4) 2 19 }
id-KmacWithSHAKE256 OBJECT IDENTIFIER ::= { joint-iso-itu-t(2)
  country(16) us(840) organization(1) gov(101) csor(3)
  nistAlgorithm(4) 2 20 }
```

Specific conventions to be considered for KMAC with SHAKE are specified in RFC 8702 Section 3.4 [RFC8702].

7. Algorithm Use Profiles

This section provides profiles of algorithms and respective conventions for different application use cases.

Recommendations like NIST SP 800-57 Recommendation for Key Management Table2 [NIST.SP.800-57pt1r5] and ECRYPT Algorithms, Key Size and Protocols Report (2018) Section 4.6 [ECRYPT.CSA.D5.4] provide general information on current cryptographic algorithms.

The overall cryptographic strength of a CMP deployment will depend on several factors, including:

- * Capabilities of the end entity: What kind of algorithms does the end entity support. The cryptographic strength of the system SHOULD be at least as strong as the algorithms and keys used for the certificate being managed.
- * Algorithm profile: The overall strength of the profile will be the strength of the weakest algorithm it contains.
- * Message protection: The overall strength of the CMC message protection
 - MAC-based protection: The entropy of the shared secret information or password when MAC-based message protection is used (MSG_MAC_ALG).

- Signature-based protection: The strength of the key pair and signature algorithm when signature-based protection is used (MSG_SIG_ALG).
- Protection of centrally generated keys: The strength of the algorithms used for the key management technique (Section 7.2: PROT_ENC_ALG or Section 7.1: KM_KA_ALG, KM_KT_ALG, KM_KD_ALG) and the encryption of the content-encryption key and private key (Section 7.2: SYM_PENC_ALG, PROT_SYM_ALG or Section 7.1: KM_KW_ALG, PROT_SYM_ALG).

The following table shows the algorithms listed in this document sorted by their bits of security. If an implementation intends to enroll and manage certificate for keys of a specific security, it SHALL implement and use algorithms of at least that strength for the respective PKI management operation. If one row does not provide a suitable algorithm, the implementer MUST choose one offering more bits of security.

| Bits of Security | RSA or DH | Elliptic Curve | Hash Function or XOF with Specified Output Length (d) | Symmetric Encryption |
|------------------|-------------------|---|---|----------------------|
| 112 | RSA2048, DH(2048) | ECDSA/ECDH (secp224r1) | SHA224 | |
| 128 | RSA3072, DH(3072) | ECDSA/ECDH (secp256r1), Ed25519/X25519 (Curve25519) | SHA256, SHAKE128 (d=256) | AES-128 |
| 192 | | ECDSA/ECDH (secp384r1) | SHA384 | AES-192 |
| 224 | | Ed448/X448 (Curve448) | | |
| 256 | | ECDSA/ECDH (secp521r1) | SHA512, SHAKE256 (d=512) | AES-256 |

Table 1: Cryptographic Algorithms Sorted by their Bits of Security

The following table shows the cryptographic algorithms sorted by their usage in CMP and with more details.

| Bits of Security | Key Types to Be Certified | CMP Protection | Key Management Technique | Key-Wrap and Symmetric Encryption |
|------------------|--------------------------------------|---|--|--|
| | | MSG_SIG_ALG, MSG_MAC_ALG | PROT_ENC_ALG or KM_KA_ALG, KM_KT_ALG, KM_KD_ALG | PROT_SYM_ALG, SYM_PENC_ALG or KM_KW_ALG |
| 112 | RSA2048, secp224r1 | RSASSA-PSS (2048, SHA224 or SHAKE128 (d=256)), RSAEncryption (2048, SHA224), ECDSA (secp224r1, SHA224 or SHAKE128 (d=256)), PBMAC1 (HMAC- SHA224) | DH(2048), RSAES-OAEP (2048, SHA224), RSAEncryption (2048, SHA224), ECDH (secp224r1, SHA224), PBKDF2 (HMAC- SHA224) | |
| 128 | RSA3072, secp256r1, Curve25519 | RSASSA-PSS (3072, SHA256 or SHAKE128 (d=256)), RSAEncryption (3072, SHA256), ECDSA (secp256r1, SHA256 or SHAKE128 (d=256)), Ed25519 (SHA512), PBMAC1 (HMAC- SHA256) | DH(3072), RSAES-OAEP (3072, SHA256), RSAEncryption (3072, SHA256), ECDH (secp256r1, SHA256), X25519, PBKDF2 (HMAC- SHA256) | AES-128 |
| 192 | secp384r1 | ECDSA (secp384r1, SHA384), PBMAC1 (HMAC- SHA384) | ECDH (secp384r1, SHA384), PBKDF2 (HMAC- SHA384) | AES-192 |
| 224 | Curve448 | Ed448 (SHAKE256) | X448 | |

| | | | | |
|-----|-----------|--|---|---------|
| 256 | secp521r1 | ECDSA (secp521r1, SHA512 or SHAKE256 (d=512)), PBMAC1 (HMAC- SHA512) | ECDH (secp521r1, SHA512), PBKDF2 (HMAC- SHA512) | AES-256 |
|-----|-----------|--|---|---------|

Table 2: Cryptographic Algorithms Sorted by their Bits of Security and Usage by CMP

To avoid consuming too much computational resources it is recommended to choose a set of algorithms offering roughly the same level of security. Below are provided several algorithm profiles which are balanced, assuming the implementer chooses MAC secrets and/or certificate profiles of at least equivalent strength.

7.1. Algorithm Profile for RFC 4210 PKI Management Message Profiles

The following table updates the definitions of algorithms used within PKI Management Message Profiles as defined in CMP Appendix D.2 [RFC4210].

The columns in the table are:

Name: An identifier used for message profiles

Use: Description of where and for what the algorithm is used

Mandatory: Algorithms which MUST be supported by conforming implementations

Optional: Algorithms which are OPTIONAL to support

Deprecated: Algorithms from RFC 4210 [RFC4210] which SHOULD NOT be used anymore

| Name | Use | Mandatory | Optional | Deprecated |
|--------------|---|-----------|------------------------------|--|
| MSG_SIG_ALG | protection of PKI messages using signature | RSA | ECDSA, EdDSA | DSA, combinations with MD5 and SHA-1 |
| MSG_MAC_ALG | protection of PKI messages using MACing | PBMAC1 | PasswordBasedMac, HMAC, KMAC | X9.9 |
| SYM_PENC_ALG | symmetric encryption of an end entity's private key where symmetric key is distributed out-of-band | AES-wrap | | 3-DES (3-key-EDE, CBC Mode), RC5, CAST-128 |
| PROT_ENC_ALG | asymmetric algorithm used for encryption of (symmetric keys for encryption of) private keys transported in PKIMessages | DH | ECDH, RSA | |
| PROT_SYM_ALG | symmetric encryption algorithm used for encryption of private key bits (a key of this type is encrypted using PROT_ENC_ALG) | AES-CBC | | 3-DES (3-key-EDE, CBC Mode), RC5, CAST-128 |

Table 3: Algorithms Used Within RFC 4210 Appendix D.2

Mandatory Algorithm Identifiers and Specifications:

RSA: sha256WithRSAEncryption with 2048 bit, see Section 3.1

PasswordBasedMac: id-PasswordBasedMac, see Section 6.1 (with id-sha256 as the owf parameter, see Section 2.1 and id-hmacWithSHA256 as the mac parameter, see Section 6.2.1)

PBMAC1: id-PBMAC1, see Section 6.1.2 (with id-PBKDF2 as the key derivation function, see Section 4.4.1 and id-hmacWithSHA256 as message authentication scheme, see Section 6.2.1). It is RECOMMENDED to prefer the usage of PBMAC1 instead of PasswordBasedMac.

DH: id-alg-ESDH, see Section 4.1.1

AES-wrap: id-aes128-wrap, see Section 4.3.1

AES-CBC: id-aes128-CBC, see Section 5.1

7.2. Algorithm Profile for Lightweight CMP Profile

The following table contains definitions of algorithms which MAY be supported by implementations of the Lightweight CMP Profile [I-D.ietf-lamps-lightweight-cmp-profile].

As the set of algorithms to be used for implementations of the Lightweight CMP Profile heavily depends on the PKI management operations implemented, the certificates used for messages protection, and the certificates to be managed, this document will not specify a specific set that is mandatory to support for conforming implementations.

The columns in the table are:

Name: An identifier used for message profiles

Use: Description of where and for what the algorithm is used

Examples: Lists the algorithms as described in this document. The list of algorithms depends on the set of PKI management operations to be implemented.

Note: It is RECOMMENDED to prefer the usage of PBMAC1 instead of PasswordBasedMac.

| Name | Use | Examples |
|--------------|---|--|
| MSG_SIG_ALG | protection of PKI messages using signature and for SignedData, e.g., a private key transported in PKIMessages | RSA, ECDSA, EdDSA |
| MSG_MAC_ALG | protection of PKI messages using MACing | PasswordBasedMac (see Section 9), PBMAC1, HMAC, KMAC |
| KM_KA_ALG | asymmetric key agreement algorithm used for agreement of a symmetric key for use with KM_KW_ALG | DH, ECDH |
| KM_KT_ALG | asymmetric key encryption algorithm used for transport of a symmetric key for PROT_SYM_ALG | RSA |
| KM_KD_ALG | symmetric key derivation algorithm used for derivation of a symmetric key for use with KM_KW_ALG | PBKDF2 |
| KM_KW_ALG | algorithm to wrap a symmetric key for PROT_SYM_ALG | AES-wrap |
| PROT_SYM_ALG | symmetric content encryption algorithm used for encryption of EnvelopedData, e.g., a private key transported in PKIMessages | AES-CBC |

Table 4: Algorithms Used Within Lightweight CMP Profile

8. IANA Considerations

This document does not request changes to the IANA registry.

9. Security Considerations

RFC 4210 Appendix D.2 [RFC4210] contains a set of algorithms, mandatory to be supported by conforming implementations. These algorithms were appropriate at the time CMP was released, but as cryptographic algorithms weaken over time, some of them should not be used anymore. In general, new attacks are emerging due to research cryptanalysis or increase in computing power. New algorithms were introduced that are more resistant to today's attacks.

This document lists many cryptographic algorithms usable with CMP to offer implementer a more up to date choice. Finally, the algorithms to be supported also heavily depend on the certificates and PKI management operations utilized in the target environment. The algorithm with the lowest security strength and the entropy of shared secret information define the security of the overall solution, see Section 7.

When using MAC-based message protection the use of PBMAC1 is preferable to that of PasswordBasedMac. First, PBMAC1 is a well-known scrutinized algorithm, which is not true for PasswordBasedMac. Second, the PasswordBasedMac algorithm as specified in RFC 4211 Section 4.4 [RFC4211] is essentially PBKDF1 (as defined in RFC 8018 Section 5.1 [RFC8018]) with an HMAC step at the end. Here we update to use the PBKDF2-HMAC construct defined as PBMAC1 in [RFC8018]. PBKDF2 is superior to PBKDF1 in an improved internal construct for iterated hashing, and in removing PBKDF1's limitation of only being able to derive keys up to the size of the underlying hash function. Additionally, PBKDF1 is not recommended for new applications as stated in Section 5.1 of RFC 8018 [RFC8018] and no longer an approved algorithm by most standards bodies, and therefore presents difficulties to implementer who are submitting their CMP implementations for certification, hence moving to a PBKDF2-based mechanism. This change is in alignment with [RFC9045] which updates [RFC4211] to allow the use of PBMAC1 in CRMF.

AES-GMAC MUST NOT be used as the pseudo random function in PBKDF2; the use of AES-GMAC more than once with the same key and the same nonce will break the security.

In Section 7 of this document there is also an update to the Appendix D.2 of RFC 4210 [RFC4210] and a set of algorithms that MAY be supported when implementing the Lightweight CMP Profile [I-D.ietf-lamps-lightweight-cmp-profile].

It is recognized that there may be older CMP implementations in use that conform to the algorithm use profile from Appendix D.2 of RFC 4210 [RFC4210]. For example, the use of AES is now mandatory for

PROT_SYM_ALG but in RFC 4210 [RFC4210] 3-DES was mandatory. Therefore, it is expected that many CMP systems may already support the recommended algorithms in this specification. In such systems the weakened algorithms should be disabled from further use. If critical systems cannot be immediately updated to conform to the recommended algorithm use profile, it is recommended a plan to migrate the infrastructure to conforming profiles be adopted as soon as possible.

Symmetric key-based MAC algorithms as described in Section 6.2 MAY be used as MSG_MAC_ALG. The implementer MUST choose a suitable PRF and ensure that the key has sufficient entropy to match the overall security level of the algorithm profile. These considerations are outside the scope of the profile.

10. Acknowledgements

Thanks to Russ Housley for supporting this draft with submitting [RFC9044] and [RFC9045].

May thanks also to all reviewers like Serge Mister, Mark Ferreira, Yuefei Lu, Tomas Gustavsson, Lijun Liao, David von Oheimb and Steffen Fries for their input and feedback to this document. Apologies to all not mentioned reviewers and supporters.

11. Normative References

[I-D.ietf-lamps-cmp-updates]

Brockhaus, H., Oheimb, D. V., and J. Gray, "Certificate Management Protocol (CMP) Updates", Work in Progress, Internet-Draft, draft-ietf-lamps-cmp-updates-18, 6 April 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-lamps-cmp-updates-18>>.

[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-11, 15 April 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-lamps-lightweight-cmp-profile-11>>.

[NIST.FIPS.180-4]

Dang, Quynh H., "Secure Hash Standard", NIST NIST FIPS 180-4, DOI 10.6028/NIST.FIPS.180-4, July 2015, <<https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf>>.

- [NIST.FIPS.186-4]
National Institute of Standards and Technology (NIST),
"Digital Signature Standard (DSS)", NIST NIST FIPS 186-4,
DOI 10.6028/NIST.FIPS.186-4, July 2013,
<[https://nvlpubs.nist.gov/nistpubs/FIPS/
NIST.FIPS.186-4.pdf](https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf)>.
- [NIST.FIPS.186-5]
National Institute of Standards and Technology (NIST),
"FIPS Pub 186-5 (Draft): Digital Signature Standard
(DSS)", October 2019,
<[https://nvlpubs.nist.gov/nistpubs/FIPS/
NIST.FIPS.186-5-draft.pdf](https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-5-draft.pdf)>.
- [NIST.FIPS.197]
National Institute of Standards and Technology (NIST),
"Advanced encryption standard (AES)", NIST NIST FIPS 197,
DOI 10.6028/NIST.FIPS.197, November 2001,
<[https://nvlpubs.nist.gov/nistpubs/FIPS/
NIST.FIPS.197.pdf](https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf)>.
- [NIST.FIPS.198-1]
National Institute of Standards and Technology (NIST),
"The Keyed-Hash Message Authentication Code (HMAC)",
NIST NIST FIPS 198-1, DOI 10.6028/NIST.FIPS.198-1, July
2008, <[https://nvlpubs.nist.gov/nistpubs/FIPS/
NIST.FIPS.198-1.pdf](https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.198-1.pdf)>.
- [NIST.FIPS.202]
Dworkin, Morris J., "SHA-3 Standard: Permutation-Based
Hash and Extendable-Output Functions", NIST NIST FIPS 202,
DOI 10.6028/NIST.FIPS.202, July 2015,
<[https://nvlpubs.nist.gov/nistpubs/FIPS/
NIST.FIPS.202.pdf](https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf)>.
- [NIST.SP.800-185]
Kelsey, John., Change, Shu-jen., and Ray. Perlner, "SHA-3
derived functions: cSHAKE, KMAC, TupleHash and
ParallelHash", NIST NIST SP 800-185,
DOI 10.6028/NIST.SP.800-185, December 2016,
<[https://nvlpubs.nist.gov/nistpubs/SpecialPublications/
NIST.SP.800-185.pdf](https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-185.pdf)>.

- [NIST.SP.800-38d] Dworkin, M J., "Recommendation for block cipher modes of operation :GaloisCounter Mode (GCM) and GMAC", NIST SP 800-38d, DOI 10.6028/NIST.SP.800-38d, 2007, <<https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38d.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>>.
- [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>>.
- [RFC2631] Rescorla, E., "Diffie-Hellman Key Agreement Method", RFC 2631, DOI 10.17487/RFC2631, June 1999, <<https://www.rfc-editor.org/info/rfc2631>>.
- [RFC3370] Housley, R., "Cryptographic Message Syntax (CMS) Algorithms", RFC 3370, DOI 10.17487/RFC3370, August 2002, <<https://www.rfc-editor.org/info/rfc3370>>.
- [RFC3394] Schaad, J. and R. Housley, "Advanced Encryption Standard (AES) Key Wrap Algorithm", RFC 3394, DOI 10.17487/RFC3394, September 2002, <<https://www.rfc-editor.org/info/rfc3394>>.
- [RFC3560] Housley, R., "Use of the RSAES-OAEP Key Transport Algorithm in Cryptographic Message Syntax (CMS)", RFC 3560, DOI 10.17487/RFC3560, July 2003, <<https://www.rfc-editor.org/info/rfc3560>>.
- [RFC3565] Schaad, J., "Use of the Advanced Encryption Standard (AES) Encryption Algorithm in Cryptographic Message Syntax (CMS)", RFC 3565, DOI 10.17487/RFC3565, July 2003, <<https://www.rfc-editor.org/info/rfc3565>>.
- [RFC4056] Schaad, J., "Use of the RSASSA-PSS Signature Algorithm in Cryptographic Message Syntax (CMS)", RFC 4056, DOI 10.17487/RFC4056, June 2005, <<https://www.rfc-editor.org/info/rfc4056>>.

- [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>>.
- [RFC4231] Nystrom, M., "Identifiers and Test Vectors for HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512", RFC 4231, DOI 10.17487/RFC4231, December 2005, <<https://www.rfc-editor.org/info/rfc4231>>.
- [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>>.
- [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>>.
- [RFC5754] Turner, S., "Using SHA2 Algorithms with Cryptographic Message Syntax", RFC 5754, DOI 10.17487/RFC5754, January 2010, <<https://www.rfc-editor.org/info/rfc5754>>.
- [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>>.
- [RFC8018] Moriarty, K., Ed., Kaliski, B., and A. Rusch, "PKCS #5: Password-Based Cryptography Specification Version 2.1", RFC 8018, DOI 10.17487/RFC8018, January 2017, <<https://www.rfc-editor.org/info/rfc8018>>.
- [RFC8032] Josefsson, S. and I. Liusvaara, "Edwards-Curve Digital Signature Algorithm (EdDSA)", RFC 8032, DOI 10.17487/RFC8032, January 2017, <<https://www.rfc-editor.org/info/rfc8032>>.

- [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>>.
- [RFC8418] Housley, R., "Use of the Elliptic Curve Diffie-Hellman Key Agreement Algorithm with X25519 and X448 in the Cryptographic Message Syntax (CMS)", RFC 8418, DOI 10.17487/RFC8418, August 2018, <<https://www.rfc-editor.org/info/rfc8418>>.
- [RFC8419] Housley, R., "Use of Edwards-Curve Digital Signature Algorithm (EdDSA) Signatures in the Cryptographic Message Syntax (CMS)", RFC 8419, DOI 10.17487/RFC8419, August 2018, <<https://www.rfc-editor.org/info/rfc8419>>.
- [RFC8702] Kampanakis, P. and Q. Dang, "Use of the SHAKE One-Way Hash Functions in the Cryptographic Message Syntax (CMS)", RFC 8702, DOI 10.17487/RFC8702, January 2020, <<https://www.rfc-editor.org/info/rfc8702>>.
- [RFC9044] Housley, R., "Using the AES-GMAC Algorithm with the Cryptographic Message Syntax (CMS)", RFC 9044, DOI 10.17487/RFC9044, June 2021, <<https://www.rfc-editor.org/info/rfc9044>>.
- [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>>.

12. Informative References

- [ECRYPT.CSA.D5.4]
University of Bristol, "Algorithms, Key Size and Protocols Report (2018)", March 2015,
<<https://www.ecrypt.eu.org/csa/documents/D5.4-FinalAlgKeySizeProt.pdf>>.
- [NIST.SP.800-57pt1r5]
Barker, Elaine., "Recommendation for key management:part 1 - general", NIST NIST SP 800-57pt1r5,
DOI 10.6028/NIST.SP.800-57pt1r5, May 2020,
<<https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r5.pdf>>.

- [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>>.
- [RFC8692] Kampanakis, P. and Q. Dang, "Internet X.509 Public Key Infrastructure: Additional Algorithm Identifiers for RSASSA-PSS and ECDSA Using SHAKEs", RFC 8692, DOI 10.17487/RFC8692, December 2019, <<https://www.rfc-editor.org/info/rfc8692>>.

Appendix A. History of Changes

Note: This appendix will be deleted in the final version of the document.

From version 12 -> 13:

- * Providing changes addressing comments from OPSDIR and GENART last call reviews

From version 11 -> 12:

- * Capitalized all headlines

From version 10 -> 11:

- * Changes on the tables in Section 7 after direct exchange with Quynh

From version 09 -> 10:

- * Removed the pre-RFC5378 work disclaimer after the RFC 4210 authors granted BCP78 rights to the IETF Trust
- * Implemented the changes proposed by Quynh, (see thread "Quynh Action: draft-ietf-lamps-cmp-algorithms-08.txt") and removed markers for Todos regarding this review of SHAKE and KMAC usage as well as on the tables in Section 7

From version 08 -> 09:

- * Updated IPR disclaimer

From version 07 -> 08:

- * Fixing issues from WG and AD review
- * Adding Note to Section 2.2, 3.3, and 6.2.3 regarding usage of SHAKE and KMAC and added Todo regarding checking respective notes

- * Added two tables showing algorithms sorted by their strength to Section 7 and added ToDo regarding checking these tables
- * Updates the algorithm use profile in Section 7.1
- * Updated and added security consideration on SHAKE, PasswordBasedMac, KMAC, and symmetric key-based MAC functions and added ToDo regarding checking the security consideration on SHAKE

From version 06 -> 07:

- * Fixing minor formatting nits

From version 05 -> 06:

- * Added text to Section 2 and Section 3.3 to clearly specify the hash algorithm to use for certConf messages for certificates signed with EdDSA (see thread "[CMP Updates] Hash algorithm to us for calculating certHash")
- * Updated new RFC numbers for I-D.ietf-lamps-cms-aes-gmac-alg and I-D.ietf-lamps-crmf-update-algs

From version 04 -> 05:

- * Minor changes and corrections in wording

From version 03 -> 04:

- * Added John Gray to the list of authors due to his extensive support and valuable feedback
- * Added some clarification of the use AES-GMAC to Section 6.2.1
- * Extended the guidance on how to select a set of algorithms in Section 7 and deleted former Section 7.1
- * Deleted the algorithms mandatory to support in Section 7.2 as discussed at IETF 110
- * Extended the Security considerations in Section 9
- * Minor changes in wording

From version 02 -> 03:

- * Moved former Appendix A to new Section 7 as suggested by Rich and Russ (see thread "I-D Action: draft-ietf-lamps-cmp-algorithms-02.txt")
- * Added a column to Table 1 in Section 7.2 to reflect the changes to RFC 4210
- * Updated Table 2 in Section 7.3
- * Added a paragraph to Section 9 to discuss backward compatibility with RFC 4210
- * Minor changes in wording

From version 01 -> 02:

- * Added Hans Aschauer, Mike Ounsworth, and Serge Mister as co-author
- * Changed to XML V3
- * Added SHAKE digest algorithm to Section 2 as discussed at IETF 109
- * Deleted DSA from Section 3 as discussed at IETF 109
- * Added RSASSA-PSS with SHAKE to Section 3
- * Added SECP curves the section on ECDSA with SHA2, ECDSA with SHAKE, and EdDSA to Section 3 as discussed at IETF 109
- * Deleted static-static D-H and ECDH from Section 4.1 based on the discussion on the mailing list (see thread "[CMP Algorithms] Section 4.1.1 and 4.1.2 drop static-static (EC)DH key agreement algorithms for use in CMP")
- * Added ECDH OIDs and SECP curves, as well as ECDH with curve25519 and curve448 to Section 4.1 as discussed at IETF 109
- * Deleted RSA-OAEP from Section 4.2 first as discussed at IETF 109, but re-added it after discussion on the mailing list (see thread "Mail regarding draft-ietf-lamps-cmp-algorithms")
- * Added a paragraph to Section 4.3.1 to explain that the algorithms and key length for content encryption and key wrapping must be aligned as discussed on the mailing list (see thread "[CMP Algorithms] Use Key-Wrap with or without padding in Section 4.3 and Section 5")
- * Deleted AES-CCM and AES-GMC from and added AES-CBC to Section 5 as discussed at IETF 109
- * Added Section 6.1.2 to offer PBMAC1 as discusses on the mailing list (see thread "Mail regarding draft-ietf-lamps-crmf-update-algs-02") and restructured text in Section 6 to be easier to differentiate between password- and shared-key-based MAC
- * Deleted Diffie-Hellmann based MAC from Section 6 as is only relevant when using enrolling Diffie-Hellmann certificates
- * Added AES-GMAC and SHAKE-based KMAC to Section 6 as discussed at IETF 109
- * Extended Section 9 to mention Russ supporting with two additional I-Ds and name further supporters of the draft
- * Added a first draft of a generic algorithm selection guideline to Appendix A
- * Added a first proposal for mandatory algorithms for the Lightweight CMP Profile to Appendix A
- * Minor changes in wording

From version 00 -> 01:

- * Changed sections Symmetric Key-Encryption Algorithms and Content Encryption Algorithms based on the discussion on the mailing list (see thread "[CMP Algorithms] Use Key-Wrap with or without padding in Section 4.3 and Section 5")

- * Added Appendix A with updated algorithms profile for RDC4210
Appendix D.2 and first proposal for the Lightweight CMP Profile
- * Minor changes in wording

Authors' Addresses

Hendrik Brockhaus (editor)
Siemens AG
Email: hendrik.brockhaus@siemens.com

Hans Aschauer
Siemens AG
Email: hans.aschauer@siemens.com

Mike Ounsworth
Entrust
Email: mike.ounsworth@entrust.com

John Gray
Entrust
Email: john.gray@entrust.com

LAMPS Working Group
Internet-Draft
Updates: 4210, 5912, 6712 (if approved)
Intended status: Standards Track
Expires: 8 October 2022

H. Brockhaus, Ed.
D. von Oheimb
Siemens
J. Gray
Entrust
6 April 2022

Certificate Management Protocol (CMP) Updates
draft-ietf-lamps-cmp-updates-18

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.

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.

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 8 October 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) | 4 |
| 2.1. New Section 1.1. - Changes Since RFC 4210 | 4 |
| 2.2. New Section 4.5 - Extended Key Usage | 5 |
| 2.3. Update Section 5.1.1. - PKI Message Header | 7 |
| 2.4. New Section 5.1.1.3. - CertProfile | 7 |
| 2.5. Update Section 5.1.3.1. - Shared Secret Information | 8 |
| 2.6. Replace Section 5.1.3.4 - Multiple Protection | 8 |
| 2.7. Replace Section 5.2.2. - Encrypted Values | 9 |
| 2.8. New Section 5.2.9 - GeneralizedTime | 11 |
| 2.9. Update Section 5.3.4. - Certification Response | 11 |
| 2.10. Update Section 5.3.18. - Certificate Confirmation Content | 12 |
| 2.11. Update Section 5.3.19.2. - Signing Key Pair Types | 13 |
| 2.12. Update Section 5.3.19.3. - Encryption/Key Agreement Key Pair Types | 13 |
| 2.13. Replace Section 5.3.19.9. - Revocation Passphrase | 13 |
| 2.14. New Section 5.3.19.14 - CA Certificates | 14 |
| 2.15. New Section 5.3.19.15 - Root CA Certificate Update | 14 |
| 2.16. New Section 5.3.19.16 - Certificate Request Template | 15 |
| 2.17. New Section 5.3.19.17 - CRL Update Retrieval | 16 |
| 2.18. Update Section 5.3.21 - Error Message Content | 17 |
| 2.19. Replace Section 5.3.22 - Polling Request and Response | 18 |
| 2.20. Update Section 7 - Version Negotiation | 22 |
| 2.21. Update Section 7.1.1. - Clients Talking to RFC 2510 Servers | 24 |
| 2.22. Add Section 8.4 - Private Keys for Certificate Signing and CMP Message Protection | 24 |
| 2.23. Add Section 8.5 - Entropy of Random Numbers, Key Pairs, and Shared Secret Information | 24 |

| | |
|--|----|
| 2.24. Add Section 8.6 - Trust Anchor Provisioning Using CMP Messages | 25 |
| 2.25. Update Section 9 - IANA Considerations | 26 |
| 2.26. Update Appendix B - The Use of Revocation Passphrase . . | 28 |
| 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 |
| 3.4. Update Section 6. - IANA Considerations | 32 |
| 4. IANA Considerations | 33 |
| 5. Security Considerations | 33 |
| 6. Acknowledgements | 33 |
| 7. References | 33 |
| 7.1. Normative References | 33 |
| 7.2. Informative References | 35 |
| Appendix A. ASN.1 Modules | 37 |
| A.1. 1988 ASN.1 Module | 37 |
| A.2. 2002 ASN.1 Module | 51 |
| Appendix B. History of Changes | 64 |
| Authors' Addresses | 70 |

1. Introduction

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.

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 [thisRFC]:

- * 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 pl0cr, 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 OIDs MAY be re-used.

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

CMP KGA: CMP Key Generation Authorities 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 therefore needs specific authorization. This authorization is with the CA certificate itself. Alternatively, the CA MAY delegate the authorization by placing the id-kp-cmKGA extended key usage in the certificate used to authenticate the origin of the generated private key or the delegation MAY be determined through local configuration of the end entity.

Note: In device PKIs, especially those issuing IDevID certificates IEEE 802.1AR Section 8.5 [IEEE.802.1AR_2018], CA certificates may have very long validity (including the GeneralizedTime value

99991231235959Z to indicate a not well-defined expiration date as specified in IEEE 802.1AR Section 8.5 [IEEE.802.1AR_2018] and RFC 5280 Section 4.1.2.5 [RFC5280]). Such validity periods SHOULD NOT be used for protection of CMP messages and key generation. Certificates containing one of the above EKUs SHOULD NOT use indefinite expiration date.

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 MAY be a MAC or a signature, depending on the information and certificates shared between the RA and the CA). Moreover, 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 is introduced by this document. 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. Moreover, 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 pl0cr 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 is introduced by this document. 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 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 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 TBD1}, SEQUENCE SIZE (1..MAX) OF CRLStatus
GenRep:    {id-it TBD2}, SEQUENCE SIZE (1..MAX) OF
           CertificateList | < absent >
```

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

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

< TBD: Add requested OIDs for id-it-crlStatusList (TBD1) and id-it-crls (TBD2). >

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 ErrorMessage 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).

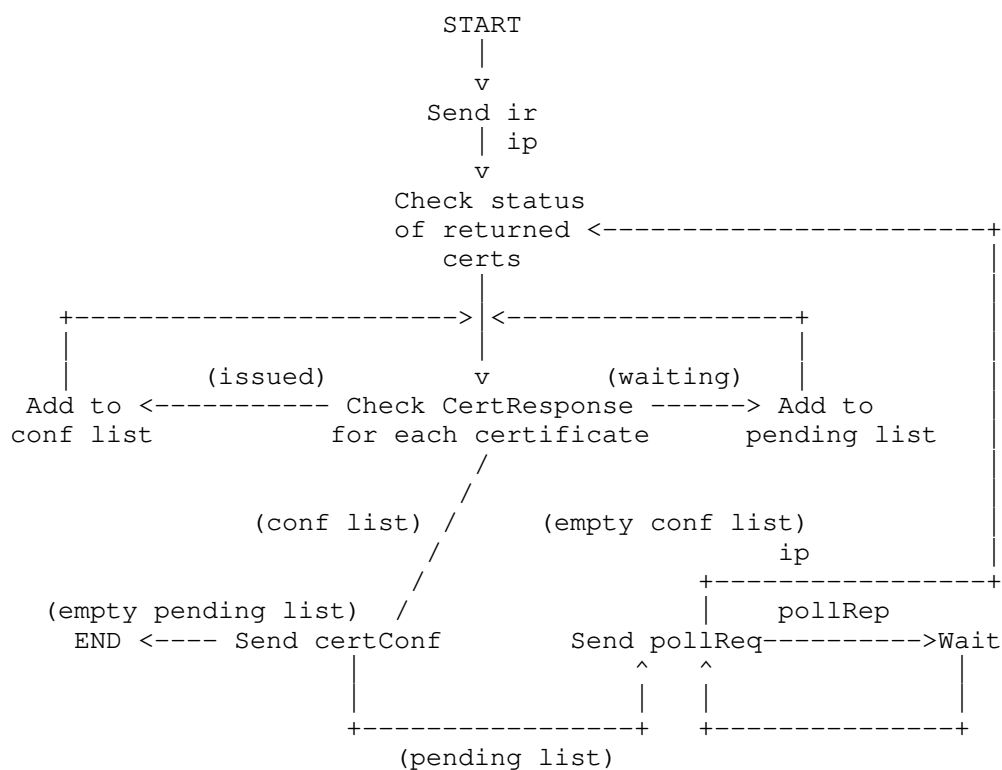
```
PollReqContent ::= SEQUENCE OF SEQUENCE {  
    certReqId    INTEGER }  
  
PollRepContent ::= SEQUENCE OF SEQUENCE {  
    certReqId    INTEGER,  
    checkAfter   INTEGER, -- time in seconds  
    reason       PKIFreeText OPTIONAL }
```

In response to an ir, cr, pl0cr, 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.

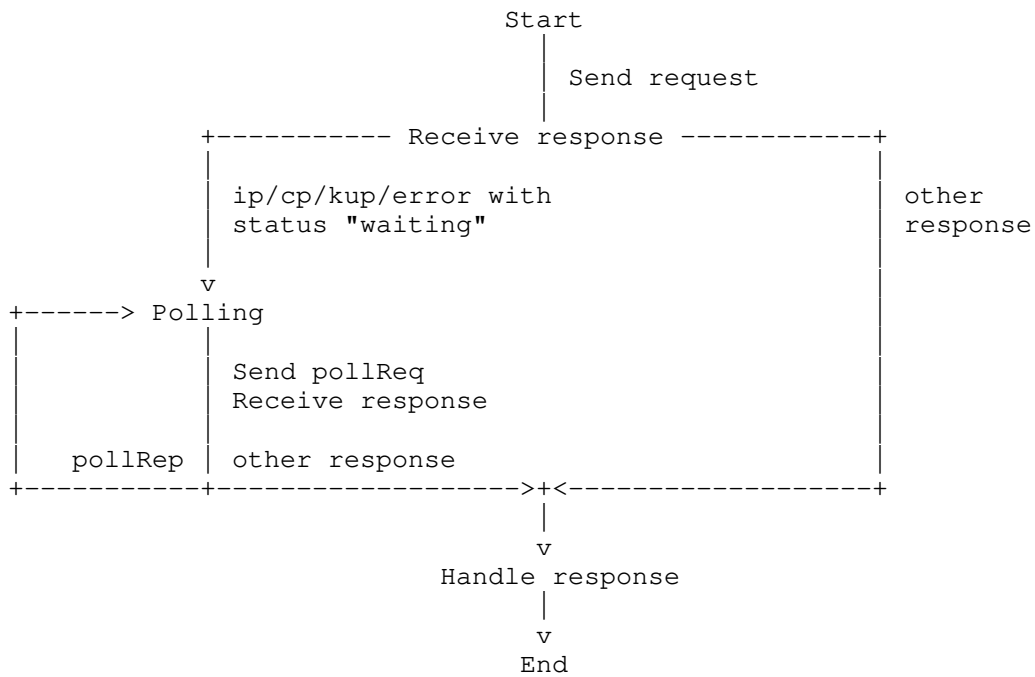
The following client-side state machine describes polling for individual CertResponse elements.



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 first three paragraphs with the following text:

This section defines the version negotiation between client and server used to choose among cmp1999 (specified in RFC 2510 [RFC2510]), cmp2000 (specified in RFC 4210 [RFC4210]), and cmp2021 (specified in this document). The only difference between protocol versions cmp2021 and cmp2000 is that EnvelopedData replaces EncryptedValue and the optional hashAlg field is added to CertStatus.

If a client does not support cmp2021 it chooses the versions for a request as follows:

- * If the 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 itself and the server.
- * If the client does not know what version(s) the server supports, then it MUST send a PKIMessage using the highest version it supports.

If a client supports cmp2021 and encrypted values are supposed to be transferred in the PKI management operation the client MUST choose the version for a request message containing the CertReqMessages data structure as follows:

- * If the client accepts EnvelopedData, but not EncryptedValue, then it MUST use cmp2021.
- * If the client does not accept EnvelopedData, but EncryptedValue, then it MUST use cmp2000.
- * If the client accepts both EnvelopedData and EncryptedValue:
 - If the client knows that the Server supports EnvelopedData (e.g., from a previous PKIMessage exchange or via some out-of-band means), then it MUST use cmp2021.
 - If the client knows that the server supports only EncryptedValue, then it MUST use cmp2000.
 - If the client does not know whether the server supports EnvelopedData or EncryptedValue, then it MUST send the request message using cmp2021.

If a client sends a certConf message and the signatureAlgorithm of the certificate to be confirmed does not specify a hash algorithm (neither in its OID nor in its parameters) there are two cases:

- * A client supporting cmp2021 MUST use cmp2021 in the certConf message.
- * A client not supporting cmp2021 will not be able to handle this situation and will fail or reject the certificate.

If a server receives a message with version cmp1999 and supports it, then the version of the response message MUST also be cmp1999. If a server receives a message with a version higher or lower than it supports, then it MUST send back an ErrorMsg with the unsupportedVersion bit set (in the failureInfo field of the

pKIStatusInfo). If the received version is higher than the highest supported version for this request message, then the version in the error message MUST be the highest version the server supports for this message type; if the received version is lower than the lowest supported version for this request message then the version in the error message MUST be the lowest version the server supports for this message type.

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. 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

When a CA acts as a CMP endpoint, it should not use the same private key for issuing certificates and for protecting CMP responses, to reduce the number of usages of the key to the minimum required.

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.

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 [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 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 without already trusting any of the CA certificates given in the message.

Moreover, 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. Update Section 9 - IANA Considerations

Section 9 of RFC 4210 [RFC4210] contains the IANA Considerations of that document. As this document defines a new Extended Key Usage, the IANA Considerations need to be updated accordingly.

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

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 | [thisRFC] |

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 | [thisRFC] |
| 18 | id-it-rootCaKeyUpdate | [thisRFC] |
| 19 | id-it-certReqTemplate | [thisRFC] |
| 20 | id-it-rootCaCert | [thisRFC] |
| 21 | id-it-certProfile | [thisRFC] |
| TBD1 | id-it-crlStatusList | [thisRFC] |
| TBD2 | id-it-crls | [thisRFC] |

Table 2: Addition to the PKIX CMP Information Types Registry

< TBD: Add requested OIDs for id-it-crlStatusList (TBD1) and id-it-crls (TBD2). >

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 | [thisRFC] |
| 12 | id-regCtrl-rsaKeyLen | [thisRFC] |

Table 3: Addition to the PKIX CRMF Registration Controls Registry

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 RFC 4211 [RFC4211].
-- *****
```

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 Appendix A.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 Appendix A.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. Moreover, delayed delivery of CMP response messages may be handled at transfer level regardless of the message contents. Since CMP Updates [thisRFC] 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 pollReq 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 [thisRFC]:

- * 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>
```

3.4. Update Section 6. - IANA Considerations

Section 6 of RFC 6712 [RFC6712] contains the IANA Considerations of that document. As this document defines a new well-known URI suffix, the IANA Considerations need to be updated accordingly.

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

6.1. Well-Known URI Registration

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: [thisRFC] [I-D.ietf-ace-cmpv2-coap-transport]
Related Information: CMP has a sub-registry at
[https://www.iana.org/assignments/cmp/]
```

6.2. CMP Well-Known URI Registry

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: [thisRFC] [I-D.ietf-ace-cmpv2-coap-transport]
```

4. IANA Considerations

This document contains an update to the IANA Consideration sections to be added to [RFC4210] and [RFC6712].

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.

< TBD: The temporary registration of cmp URI suffix expires 2022-05-20. The registration must be extended in time or update from provisional to permanent. >

< TBD: New protocol registry group "Certificate Management Protocol (CMP)" (at <https://www.iana.org/assignments/cmp>) and new registry "CMP Well-Known URI Path Segments" with the initial entry 'p' must be registered at IANA. >

5. Security Considerations

The security considerations of RFC 4210 [RFC4210] are extended in Section 2.22 to Section 2.24. No changes are made to the existing security considerations of RFC 6712 [RFC6712].

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 also 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-12, 6 April 2022,
<<https://datatracker.ietf.org/doc/html/draft-ietf-lamps-cmp-algorithms-12>>.
- [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., 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/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>>.
- [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>>.
- [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-10, 1 February 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-lamps-lightweight-cmp-profile-10>>.
- [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.800-90Arl] Barker, Elaine B. and John M. Kelsey, "Recommendation for Random Number Generation Using Deterministic Random Bit Generators", NIST NIST SP 800-90Arl, DOI 10.6028/NIST.SP.800-90Arl, June 2015, <<https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90Arl.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>>.

Appendix A. ASN.1 Modules

A.1. 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 [thisRFC]]. 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 [thisRFC]
-- 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
    pl0cr   [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 (SHA-1 recommended)
    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}
DHBMPParameter ::= SEQUENCE {
    owf                 AlgorithmIdentifier,
    -- AlgId for a One-Way Function (SHA-1 recommended)
    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; 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),
    -- invalid integrity, password based instead of signature or
    -- vice versa
    badRecipientNonce (13),
    -- invalid 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),
    -- invalid sender nonce, either missing or wrong size
badCertTemplate     (19),
    -- invalid 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 [thisRFC]

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 [thisRFC]
    -- 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 [thisRFC]
    -- 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 [thisRFC]

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 [thisRFC]

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 [thisRFC]
--   - 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 [thisRFC]
--      id-it-rootCaKeyUpdate OBJECT IDENTIFIER ::= {id-it 18}
--      RootCaKeyUpdateValue  ::= RootCaKeyUpdateContent
--      - id-it-rootCaKeyUpdate added in CMP Updates [thisRFC]
--      id-it-certReqTemplate OBJECT IDENTIFIER ::= {id-it 19}
--      CertReqTemplateValue  ::= CertReqTemplateContent
--      - id-it-certReqTemplate added in CMP Updates [thisRFC]
--      id-it-rootCaCert      OBJECT IDENTIFIER ::= {id-it 20}
--      RootCaCertValue       ::= CMPCertificate
--      - id-it-rootCaCert added in CMP Updates [thisRFC]
--      id-it-certProfile     OBJECT IDENTIFIER ::= {id-it 21}
--      CertProfileValue      ::= SEQUENCE SIZE (1..MAX) OF
--                               UTF8String
--      - id-it-certProfile added in CMP Updates [thisRFC]
--      id-it-crlStatusList   OBJECT IDENTIFIER ::= {id-it TBD1}
--      CRLStatusListValue    ::= SEQUENCE SIZE (1..MAX) OF
--                               CRLStatus
--      - id-it-crlStatusList added in CMP Updates [thisRFC]
--      id-it-crls            OBJECT IDENTIFIER ::= {id-it TBD2}
--      CRLsValue             ::= SEQUENCE SIZE (1..MAX) OF
--                               CertificateList
--      - id-it-crls added in CMP Updates [thisRFC]
--
-- 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          PKISStatusInfo,
    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 [thisRFC]
-- 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 [thisRFC]

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. 2002 ASN.1 Module

This section contains the updated 2002 ASN.1 module for [RFC5912]. This module replaces the module in Section 9 of that document. 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 [thisRFC]. 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 [thisRFC]

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 [thisRFC]
;

-- 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 (SHA-1 recommended)
 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 }
DHBMParameter ::= SEQUENCE {
    owf          AlgorithmIdentifier{DIGEST-ALGORITHM, {...}},
    -- AlgId for a One-Way Function (SHA-1 recommended)
    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; 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),
-- invalid integrity, password based instead of signature or
-- vice versa
badRecipientNonce  (13),
-- invalid 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),
-- invalid sender nonce, either missing or wrong size
badCertTemplate    (19),
-- invalid 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 [thisRFC]

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 pl0cr)
    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 [thisRFC]
    -- 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 [thisRFC]
    -- 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 [thisRFC]

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 [thisRFC]

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 [thisRFC]
--   - 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 [thisRFC]
-- id-it-rootCaKeyUpdate OBJECT IDENTIFIER ::= {id-it 18}
-- RootCaKeyUpdateValue ::= RootCaKeyUpdateContent
--   - id-it-rootCaKeyUpdate added in CMP Updates [thisRFC]
-- id-it-certReqTemplate OBJECT IDENTIFIER ::= {id-it 19}
-- CertReqTemplateValue ::= CertReqTemplateContent
--   - id-it-certReqTemplate added in CMP Updates [thisRFC]
-- id-it-rootCaCert OBJECT IDENTIFIER ::= {id-it 20}
-- RootCaCertValue ::= CMPCertificate
--   - id-it-rootCaCert added in CMP Updates [thisRFC]
-- id-it-certProfile OBJECT IDENTIFIER ::= {id-it 21}
-- CertProfileValue ::= SEQUENCE SIZE (1..MAX) OF
--                               UTF8String
--   - id-it-certProfile added in CMP Updates [thisRFC]
-- id-it-crlStatusList OBJECT IDENTIFIER ::= {id-it TBD1}
-- CRLStatusListValue ::= SEQUENCE SIZE (1..MAX) OF
--                               CRLStatus
--   - id-it-crlStatusList added in CMP Updates [thisRFC]

```

```
-- id-it-crls          OBJECT IDENTIFIER ::= {id-it TBD2}
-- CRLsValue           ::= SEQUENCE SIZE (1..MAX) OF
--                               CertificateList
--       - id-it-crls added in CMP Updates [thisRFC]
--
-- 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 [thisRFC]
-- 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

Note: This appendix will be deleted in the final version of the document.

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 "dtaft-ietf-lamps-cmp-updates and rsaKeyLen")
- * Updated the IANA considerations in Section 2.25 to introduce new OID for id-regCtrl-algId and id-regCtrl-rsaKeyLen (see thread "dtaft-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 ECU 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 AG
Email: hendrik.brockhaus@siemens.com

David von Oheimb
Siemens AG
Email: david.von.oheimb@siemens.com

John Gray
Entrust
Email: john.gray@entrust.com

LAMPS Working Group
Internet-Draft
Intended status: Standards Track
Expires: 8 September 2022

D.K. Gillmor
American Civil Liberties Union
B. Hoeneisen
pEp Foundation
A. Melnikov
Isode Ltd
7 March 2022

Header Protection for S/MIME
draft-ietf-lamps-header-protection-08

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 to offer a different mechanism that provides the same cryptographic protections but with fewer downsides when handled by legacy clients. Furthermore, it offers more explicit guidance for clients when generating or handling e-mail messages with cryptographic protection of message headers.

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 8 September 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 | 5 |
| 1.1. | Two Schemes of Header Protection | 6 |
| 1.2. | Problems with Wrapped Messages | 6 |
| 1.3. | Problems with Injected Headers | 7 |
| 1.4. | Motivation | 7 |
| 1.4.1. | Backward Compatibility | 7 |
| 1.4.2. | Deliverability | 8 |
| 1.5. | Other Protocols to Protect Email Header Fields | 8 |
| 1.6. | Applicability to PGP/MIME | 9 |
| 1.7. | Requirements Language | 9 |
| 1.8. | Terms | 9 |
| 1.9. | Document Scope | 10 |
| 1.9.1. | Out of Scope | 11 |
| 2. | Specification | 11 |
| 2.1. | Injected Headers Scheme | 12 |
| 2.2. | Wrapped Message Scheme | 12 |
| 2.3. | Sending Side | 12 |
| 2.3.1. | Composing a Cryptographically-Protected Message Without Header Protection | 12 |
| 2.3.2. | Header Confidentiality Policy | 13 |
| 2.3.3. | Composing with "Injected Headers" Header Protection | 14 |
| 2.3.4. | Composing with "Wrapped Message" Header Protection | 18 |
| 2.3.5. | Choosing Between Wrapped Message and Injected Headers | 19 |
| 2.4. | Default Header Confidentiality Policy | 19 |
| 2.4.1. | Minimalist Header Confidentiality Policy | 20 |
| 2.4.2. | Strong Header Confidentiality Policy | 20 |
| 2.4.3. | Offering Stronger Header Confidentiality | 20 |
| 2.5. | Receiving Side | 21 |
| 2.5.1. | Identifying that a Message has Header Protection | 21 |
| 2.5.2. | Updating the Cryptographic Summary | 22 |
| 2.5.3. | Rendering a Message with Injected Headers | 22 |
| 2.5.4. | Rendering a Wrapped Message | 25 |
| 2.5.5. | Guidance for Automated Message Handling | 27 |
| 2.5.6. | Affordances for Debugging and Troubleshooting | 28 |
| 2.5.7. | Rendering Other Schemes | 28 |

| | |
|---|----|
| 2.5.8. Composing a Reply to an Encrypted Message with Header Protection | 29 |
| 2.5.9. Implicitly-rendered Header Fields | 30 |
| 2.5.10. Unprotected Header Fields Added in Transit | 30 |
| 3. E-mail Ecosystem Evolution | 32 |
| 3.1. Dropping Legacy Display Elements | 32 |
| 4. Usability Considerations | 32 |
| 4.1. Mixed Protections Within a Message Are Hard To Understand | 33 |
| 4.2. Users Should Not Have To Choose a Header Confidentiality Policy | 33 |
| 4.3. Users Should Not Have To Choose a Header Protection Scheme | 33 |
| 5. Security Considerations | 33 |
| 6. Privacy Considerations | 33 |
| 7. IANA Considerations | 33 |
| 8. Acknowledgments | 33 |
| 9. References | 33 |
| 9.1. Normative References | 33 |
| 9.2. Informative References | 34 |
| Appendix A. Possible Problems with some Legacy Clients | 35 |
| A.1. Problems Reviewing signed+encrypted Messages in List View | 36 |
| A.2. Problems when Rendering a signed+encrypted Message | 36 |
| A.3. Problems when Replying to a signed+encrypted Message | 37 |
| A.4. Problems Reviewing signed-only Messages in List View | 37 |
| A.5. Problems when Rendering a signed-only Message | 38 |
| A.6. Problems when Replying to a signed-only Message | 38 |
| Appendix B. Test Vectors | 39 |
| B.1. Baseline Messages | 39 |
| B.1.1. No cryptographic protections over a simple message | 39 |
| B.1.2. S/MIME signed-only signedData over a simple message, No Header Protection | 40 |
| B.1.3. S/MIME signed-only multipart/signed over a simple message, No Header Protection | 42 |
| B.1.4. S/MIME encrypted and signed over a simple message, No Header Protection | 44 |
| B.1.5. No cryptographic protections over a complex message | 47 |
| B.1.6. S/MIME signed-only signedData over a complex message, No Header Protection | 48 |
| B.1.7. S/MIME signed-only multipart/signed over a complex message, No Header Protection | 50 |
| B.1.8. S/MIME encrypted and signed over a complex message, No Header Protection | 53 |
| B.2. Signed-only Messages | 57 |
| B.2.1. S/MIME signed-only signedData over a simple message, Wrapped Message | 57 |

| | | |
|---------|---|-----|
| B.2.2. | S/MIME signed-only multipart/signed over a simple message, Wrapped Message | 59 |
| B.2.3. | S/MIME signed-only signedData over a simple message, Injected Headers | 61 |
| B.2.4. | S/MIME signed-only multipart/signed over a simple message, Injected Headers | 63 |
| B.2.5. | S/MIME signed-only signedData over a complex message, Wrapped Message | 65 |
| B.2.6. | S/MIME signed-only multipart/signed over a complex message, Wrapped Message | 67 |
| B.2.7. | S/MIME signed-only signedData over a complex message, Injected Headers | 70 |
| B.2.8. | S/MIME signed-only multipart/signed over a complex message, Injected Headers | 73 |
| B.3. | Encrypted-and-signed Messages | 76 |
| B.3.1. | S/MIME encrypted and signed over a simple message, Wrapped Message with hcp_minimal | 76 |
| B.3.2. | S/MIME encrypted and signed over a simple message, Injected Headers with hcp_minimal | 79 |
| B.3.3. | S/MIME encrypted and signed over a simple message, Injected Headers with hcp_minimal (+ Legacy Display) | 82 |
| B.3.4. | S/MIME encrypted and signed over a simple message, Wrapped Message with hcp_strong | 85 |
| B.3.5. | S/MIME encrypted and signed over a simple message, Injected Headers with hcp_strong | 88 |
| B.3.6. | S/MIME encrypted and signed over a simple message, Injected Headers with hcp_strong (+ Legacy Display) | 91 |
| B.3.7. | S/MIME encrypted and signed reply over a simple message, Wrapped Message with hcp_minimal | 94 |
| B.3.8. | S/MIME encrypted and signed reply over a simple message, Injected Headers with hcp_minimal | 97 |
| B.3.9. | S/MIME encrypted and signed reply over a simple message, Injected Headers with hcp_minimal (+ Legacy Display) | 100 |
| B.3.10. | S/MIME encrypted and signed reply over a simple message, Wrapped Message with hcp_strong | 103 |
| B.3.11. | S/MIME encrypted and signed reply over a simple message, Injected Headers with hcp_strong | 106 |
| B.3.12. | S/MIME encrypted and signed reply over a simple message, Injected Headers with hcp_strong (+ Legacy Display) | 109 |
| B.3.13. | S/MIME encrypted and signed over a complex message, Wrapped Message with hcp_minimal | 113 |
| B.3.14. | S/MIME encrypted and signed over a complex message, Injected Headers with hcp_minimal | 116 |
| B.3.15. | S/MIME encrypted and signed over a complex message, Injected Headers with hcp_minimal (+ Legacy Display) | 120 |

| | | |
|--------------------|--|-----|
| B.3.16. | S/MIME encrypted and signed over a complex message, Wrapped Message with hcp_strong | 124 |
| B.3.17. | S/MIME encrypted and signed over a complex message, Injected Headers with hcp_strong | 128 |
| B.3.18. | S/MIME encrypted and signed over a complex message, Injected Headers with hcp_strong (+ Legacy Display) . | 132 |
| B.3.19. | S/MIME encrypted and signed reply over a complex message, Wrapped Message with hcp_minimal | 136 |
| B.3.20. | S/MIME encrypted and signed reply over a complex message, Injected Headers with hcp_minimal | 140 |
| B.3.21. | S/MIME encrypted and signed reply over a complex message, Injected Headers with hcp_minimal (+ Legacy Display) | 144 |
| B.3.22. | S/MIME encrypted and signed reply over a complex message, Wrapped Message with hcp_strong | 148 |
| B.3.23. | S/MIME encrypted and signed reply over a complex message, Injected Headers with hcp_strong | 152 |
| B.3.24. | S/MIME encrypted and signed reply over a complex message, Injected Headers with hcp_strong (+ Legacy Display) | 155 |
| Appendix C. | Additional information | 159 |
| C.1. | Stored Variants of Messages with Bcc | 159 |
| Appendix D. | Examples | 160 |
| D.1. | Example text/plain Cryptographic Payload with Legacy Display Elements | 160 |
| D.2. | Example text/html Cryptographic Payload with Legacy Display Elements | 161 |
| Appendix E. | Document Considerations | 162 |
| Appendix F. | Document Changelog | 163 |
| Appendix G. | Open Issues | 164 |
| Authors' Addresses | | 165 |

1. Introduction

Privacy and security issues regarding email Header Protection in S/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 structures for how message headers can be cryptographically protected, and provides guidance for implementers of MUAs that generate and interpret such messages. It 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 email header sections or fields and provides guidance to implementers.

One scheme is the form specified in S/MIME 3.1 and later, which 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 is documented in more detail in [RFC8551]. Experience has shown that this form does not interact well with some legacy MUAs (see Section 1.2).

Consequently, another form of header protection is introduced, where the protected header fields are placed directly on the Cryptographic Payload, without using an intervening message/* MIME object. This document calls this scheme "Injected Headers", and it is documented in more detail in this document, in Section 2.3.3 and Section 2.5.3.

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 the worst 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 a MUA.

In other cases, the user sees an attachment suggesting a forwarded email message, which -- in fact -- contains the protected email 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 headers 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 clients 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 client. But accommodations are described here that ensure a rough semantic equivalence for legacy clients even in these cases.

1.4.2. Deliverability

A message that cannot be delivered is less useful than a message with perfect cryptographic protections. 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 Email Header Fields

A separate pair of protocols also provides some cryptographic protection for the email 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 email (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 DKIM+DMARC not provide end-to-end protection. DKIM+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, DKIM+DMARC only provides cryptographic integrity and authentication, not encryption. So cryptographic confidentiality is not available from that suite.

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

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", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

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 Email 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 is a line beginning with a field name, followed by a colon (":"), followed by a field body (value), and terminated by CRLF; see [RFC5322].

- * **Header Section:** The Header Section is a sequence of lines of characters with special syntax as defined in [RFC5322]. It is the top section of a Message, and it contains the Header Fields associated with the Message itself.
- * **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:** cryptographic protection of email Header Sections (or parts of it) for signatures and/or encryption
- * **Cryptographic Layer, Cryptographic Payload, Cryptographic Envelope, Structural Headers, Main Body Part, User-Facing Headers, and MUA** are all used as defined in [I-D.ietf-lamps-e2e-mail-guidance]
- * **Legacy MUA:** a 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.
- * **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. (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:** a functional specification of which header fields should be obscured when composing an encrypted message with header protection. See Section 2.3.2.

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 headers as well as its body.

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 client 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.

1.9.1. Out of Scope

While the generation guidance aims to provide minimal disruption for any legacy client, such a client by definition does not implement this document.

Therefore, the document does not attempt to provide guidance for legacy clients.

Furthermore, this document does not explicitly contemplate unusual (and tricky) 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.

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 2.3.3), and MAY generate the Wrapped Message scheme (Section 2.3.4).

2.1. Injected Headers Scheme

The Injected Headers scheme places all header fields to be protected directly into 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".

Composing a message with the Injected Headers scheme is described in Section 2.3.3. Rendering such a message is described in Section 2.5.3.

2.2. Wrapped Message Scheme

The Wrapped Message scheme creates a message/rfc822 (or message/global) MIME object containing the message and all header fields to be protected, and then uses that encapsulated MIME part as the Cryptographic Payload.

Composing a message with the Wrapped Message scheme is described in Section 2.3.4. 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

[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 headers (`Content-*`) present.
- * `origheaders`: the intended non-structural headers 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 `origbody`, yielding MIME tree output
- * For each header name and value (h,v) in `origheaders`:
 - Add header h of 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 header field name `name` and 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.

For example, an MUA that only obscures the Subject header field by replacing it with the literal string [...] and does not offer confidentiality to any other header fields would be represented as (in pseudocode):

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

2.3.3. 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 needs one additional input in addition to the Header Confidentiality Policy `hcp` defined in Section 2.3.2.

- * `legacy`: a boolean value, indicating whether any recipient of the message is believed to have a legacy client. If all recipients are known to implement this draft, `legacy` should be set to false. (How a 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 [I-D.ietf-lamps-e2e-mail-guidance] for guidance on identifying the parts of a message that are a Main Body Part.

The revised algorithm for applying cryptographic protection to a message is as follows:

- * 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 [I-D.ietf-lamps-e2e-mail-guidance]):
 - + If hcp(h,v) is not v:
 - * Append (h,v) to ldlist
 - If ldlist is not empty:
 - o Identify each leaf MIME part of payload that represents the "main body" of the message.
 - o For each "Main Body Part" bodypart of type text/plain or text/html:
 - + Insert Legacy Display element header list ldlist into the content of bodypart (see Section 2.3.3.1 for text/plain and Section 2.3.3.2 for text/html)
 - + Add Content-Type parameter hp-legacy-display with value 1 to bodypart
- * For each header field name and value (h,v) in origheaders:
 - Add header field h of MIME part payload with value v

- * Set the protected-headers parameter on the Content-Type of payload to vl
- * Apply crypto to payload, producing MIME tree output
- * If crypto contains encryption:
 - Create new empty list of header field names and values newh
 - For header field name and value (h,v) in origheaders:
 - o Let newval be hcp(h,v)
 - o If newval is not null:
 - + Add newh[h] to newval
 - Set origheaders to newh
- * For each header field name and value (h,v) in origheaders:
 - Add header field h of 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.

2.3.3.1. 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.

For example, if the list of obscured header fields was [{"Cc", "alice@example.net"}, {"Subject", "Thursday's meeting"}], then a text/plain part that originally contained:

I think we should skip the meeting.

Would become:

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

I think we should skip the meeting.

2.3.3.2. 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 2.3.3.1). Instead of adding the obscured 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 SHOULD represent all the obscured header fields in a readable fashion. A simple approach is to assemble the text in the same way as Section 2.3.3.1, 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.

For example, if the list of obscured header fields was [("Cc", "alice@example.net"), ("Subject", "Thursday's meeting")], then a text/html part that originally contained:

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

Would become:

```
<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>
```

2.3.3.3. 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 [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.

2.3.3.4. 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 clients 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 client 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.

2.3.4. Composing with "Wrapped Message" Header Protection

The Wrapped Message header protection scheme is briefly documented in Section 3.1 [RFC8551]. This section provides a more detailed explanation of how to build such a message, and augments it with the forwarded parameter as described in [I-D.melnikov-iana-reg-forwarded].

To compose a message using "Wrapped Message" header protection, we use those inputs described in Section 2.3.1 plus the Header Confidentiality Policy hcp defined in Section 2.3.2. The new algorithm is:

- * For header field name and value (h,v) in origheaders:
 - Add header field h of origbody with value v
- * If any of the header fields in origbody, 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; forwarded=no, and whose body is origbody.
- * Else:

- Let payload be a new MIME part with one header field: Content-Type: message/rfc822; forwarded=no, and whose body is origbody.
- * Apply crypto to payload, yielding MIME tree output
- * If crypto contains encryption:
 - Create new empty list of header field names and values newh
 - For header field name and value (h,v) in origheaders:
 - o Let newval be hcp(h,v)
 - o If newval is not null:
 - + Append (h,newval) to newh
 - Set origheaders to newh
- * For header field name and value (h,v) in origheaders:
 - Add header field h of output with value v
- * Return output

Note that the Header Confidentiality Policy hcp is ignored if crypto does not contain encryption. This is by design.

2.3.5. 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 2.3.3 format.

2.4. Default Header Confidentiality Policy

An MUA SHOULD have a sensible default Header Confidentiality Policy, and SHOULD NOT require the user to select one.

The default Header Confidentiality Policy SHOULD provide confidentiality for the Subject header field by replacing it with the literal string [...]. 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.

```
[[ TODO: select one of the two policies below the recommended default
]]
```

2.4.1. Minimalist Header Confidentiality Policy

Accordingly, the most conservative recommended Header Confidentiality Policy only protects the Subject:

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

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.

2.4.3. Offering Stronger Header Confidentiality

A 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.

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 [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.

A 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).

A 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 forwarded has a value of no. 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.

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.

Rendering this information 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.

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. As in Section 2.5.4.2, each rendered header field found in G should be compared against the header field of the same name from E. If the value found in E matches the value found in G, the header field should be marked as signed-only. If no matching header field was found in E, or the value found did not match the value from G, the header field should be marked as signed-and-encrypted.

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 or generating a draft reply.

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.

A straightforward way for an HTML-capable MUA to do this is 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]. 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 `forwarded` is set to `no`, 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.

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. Each rendered header field found in Q should be compared against the header field of the same name from O. If the value found in O matches the value found in Q, the header field should be marked as signed-only. If no matching header field was found in O, or the value found did not match the value from Q, the header field should be marked as signed-and-encrypted.

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 SHOULD identify when the MIME part contains a legacy display element (see Section 2.5.3.3.1), and it SHOULD 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 legacy display lines (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 MUA itself, or problems with the SMTP transport path taken by the message.

A 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.

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.

While this document is not normative for those schemes, it offers guidance for how to identify and handle these other formats. In the following a list of systems that are known to generate email messages with end-to-end cryptographic protections that include header protection using a different MIME scheme.

2.5.7.1. Pretty Easy Privacy (pEp)

The pEp (pretty Easy privacy) [I-D.pep-general] project specifies MIME schemes for Signed-and-Encrypted email messages that also provide header protection [I-D.pep-email]. Similar to the "Wrapped Messages" scheme described in Section 2.3.4 and Section 2.5.4, pEp email messages are fully encapsulated in the Cryptographic Payload.

More information can be found in [I-D.pep-email].

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 Headers in Reply

As noted in [I-D.ietf-lamps-e2e-mail-guidance], an MUA in this position MUST NOT leak previously-encrypted content in the clear in a followup message. The same is true for protected header fields.

Values from any header field that was identified as either encrypted 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.

[[TODO: formally describe how a replying MUA should generate a message-specific Header Protection policy based on the cryptographic status of the headers of the incoming message]]

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 SHOULD use the value from the protected header field, and SHOULD NOT use any value found outside the cryptographic protection.

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 (e.g., to test whether they are covered by a valid DKIM-Signature, see [RFC6376]) before rendering or acting on them.

Specific examples 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.

FIXME: other examples of unprotected transit header fields?

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 client.

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 {#compose-injected-headers} 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.

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 section in [I-D.ietf-lamps-e2e-mail-guidance].

4.1. Mixed Protections Within a Message Are Hard To Understand

[[TODO]]

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

[[TODO]]

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

[[TODO]]

5. Security Considerations

[[TODO]]

6. Privacy Considerations

[[TODO]]

7. IANA Considerations

This document requests no action from IANA.

[[RFC Editor: This section may be removed before publication.]]

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, Claudio Luck, David Wilson, Hernani Marques, juga, Krista Bennett, Kelly Bristol, Lars Rohwedder, Robert Williams, 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., "Guidance on End-to-End E-mail Security",
Work in Progress, Internet-Draft, draft-ietf-lamps-e2e-
mail-guidance-02, 25 January 2022,
<<https://www.ietf.org/archive/id/draft-ietf-lamps-e2e-mail-guidance-02.txt>>.

- [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://www.ietf.org/archive/id/draft-ietf-lamps-header-protection-requirements-01.txt>>.
- [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>>.
- [RFC2046] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, DOI 10.17487/RFC2046, November 1996, <<https://www.rfc-editor.org/info/rfc2046>>.
- [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>>.
- [RFC5322] Resnick, P., Ed., "Internet Message Format", RFC 5322, DOI 10.17487/RFC5322, October 2008, <<https://www.rfc-editor.org/info/rfc5322>>.
- [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>>.

9.2. Informative References

- [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/>>.
- [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://www.ietf.org/archive/id/draft-ietf-lamps-samples-08.txt>>.
- [I-D.melnikov-iana-reg-forwarded]
Melnikov, A. and B. Hoeneisen, "IANA Registration of Content-Type Header Field Parameter 'forwarded'", Work in

Progress, Internet-Draft, draft-melnikov-iana-reg-forwarded-00, 4 November 2019, <<https://www.ietf.org/archive/id/draft-melnikov-iana-reg-forwarded-00.txt>>.

[I-D.pep-email]

Marques, H., "pretty Easy privacy (pEp): Email Formats and Protocols", Work in Progress, Internet-Draft, draft-pep-email-01, 2 November 2020, <<https://www.ietf.org/archive/id/draft-pep-email-01.txt>>.

[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-00, 3 March 2022, <<https://www.ietf.org/archive/id/draft-pep-general-00.txt>>.

[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/info/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/info/rfc3156>>.

[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/info/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/info/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/info/rfc7489>>.

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.

Hopefully, a non-legacy 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+encrypted Messages in List View

- * Unprotected Subject, Date, From, To are visible
- * Threading is not visible

A.2. Problems when Rendering a signed+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+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:

```

MIME-Version: 1.0
Content-Type: text/plain; charset="utf-8"
Content-Transfer-Encoding: 7bit
Subject: no-crypto
Message-ID: <no-crypto@lhp.example>
From: Alice <alice@smime.example>
To: Bob <bob@smime.example>
Date: Sat, 20 Feb 2021 10:00:02 -0500

```

This is the no-crypto message.

This message uses no cryptographic protection at all. Its body is a text/plain message.

```

--
Alice
alice@smime.example

```

B.1.2. S/MIME signed-only signedData over a simple message, No Header Protection

This is a signed-only S/MIME message via PKCS#7 signedData. The payload is a text/plain message. It uses no header protection.

It has the following structure:

```

application/pkcs7-mime [smime.p7m] 3852 bytes
  (unwraps to)
  text/plain 204 bytes

```

Its contents are:

```

Content-Transfer-Encoding: base64
Content-Type: application/pkcs7-mime; name="smime.p7m";
  smime-type="signed-data"
Subject: smime-one-part
Message-ID: <smime-one-part@lhp.example>
From: Alice <alice@smime.example>
To: Bob <bob@smime.example>
Date: Sat, 20 Feb 2021 10:01:02 -0500

```

```

MIILFwYJKoZIhvcNAQcCoIILCDCCCwQCAQEExDTALBglghkgBZQMEAgEwggFABgkq
hkiG9w0BBwGgggExBIIBLU1JTUUtVmVyc2lvcjogMS4wDQpDb250ZW50LVR5cGU6
IHRleHQvcGxhaW47IGNoYXJzZXQ9InV0Zi04IG0KQ29udGVudC1UcmFuc2Z1ci1F
bmNvZGluZz0KQ29udGVudC1UcmFuc2Z1ci1FbmNvZGluZz0KQ29udGVudC1UcmFuc2Z1ci1F
YWdlLg0KQ29udGVudC1UcmFuc2Z1ci1FbmNvZGluZz0KQ29udGVudC1UcmFuc2Z1ci1F
IFBLQ1MjNyBzaWduZWREYXRhLiAgVGhlDQpwYXlsb2FkIGlzIGEgdGV4dC9wbGFp
biBtZXNzYWdlLiBJdCB1c2VzIG5vIGhlYWV0ZWN0aW9uLg0KDQotLSAN

```

CkFsaWNlDQphbGljZUBzbWltZS5leGFtcGxldQqgggemMIIDzZCCAgAwIBAgIT
Dy0lvRE5l0rOQ1SHoe49NAaKtDANBgkqhkiG9w0BAQ0FAFBVMQ0wCwYDVQQKEwRJ
RVVRGMREwDwYDVQQLEwhMQU1QUyBXRzExMC8GA1UEAxMoU2FtcGx1IEExBTBVTIFJT
QSBdZXJ0aWZpY2F0aW9uIEF1dGhvcml0eTagFw0xOTExMjAwNjU0MThaGA8yMDUy
MDkyNzA2NTQxOFowOzENMA8GA1UEChMESUVURjERMA8GA1UECzMITEFNUFMgV0cx
FzAVBgNVBAMTDkFsaWNlIEExvdmVsYWNlMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8A
MIIBCgKCAQEAmPUp+ovBouOP6AFQJ+RpwP0DxxzY60n1lJ53pTeNSiJlWkwtw/cx
Qq0t4uD2vWYB8gOUH/CVt2Zp1c+auzPKJ2Zu5mY6kHm+hVB+IthjLeI7Htg6rNeu
Xq50/TuTSxX5R1I1EXGt8p6hAQVeA5oZ2afHg4b97enV8gozR0/Nkug4AkXmbk7T
Hnc8vvjMUJanZ/VmS4TgDqXjWShp1cI3lcvvBZMswt41/0HJvmsWqps6oQcAx3We
ag0yCNj1V9V9yu/3DjcYbwW2lJf5NbMHbM1LY4X5chWfNEbkN6hQury/zxnlsukg
n+fHbqvwDhJLAgFpW/jA/EB/WI+whUpqtQIDAQAB04GvMIGsMAwGA1UdEwEB/wQC
MAAwFwYDVR0gBBAwDjAMBgpghkgBZQMCAATABMBA4GA1UdEQQXMBWBE2FsaWNlQHNT
aW1lLmV4YVw1wbgUwEwYDVR0lBAwwCgYIKwYBBQUHAwQwDgYDVR0PAQH/BAQDAgUg
MB0GA1UdDgQWBBSiU0HVRDyAKRV8ASpW546vzfN3DzAfBgNVHSMEGDAwGBSRMI58
BxcMp/EJKGU2GmccaHb0WTANBgkqhkiG9w0BAQ0FAAOCAQEAgU14oJyxMpWpAy1
OvK6NEbM1lgD5H14EC4Muxqlu0q2XgXOSBHI6DfX/4LDsfX7fSIus8gWVY3WqMeu
OA7IizkBD+GDEu8uKveERRXZncxGwy2MfbH1Ib3U8QzTjqB8+dz2AwYeMxODWq9o
pwtA/1TOkRg8uuiVZfg/m5fFo/QshlHNaaTDVEXsU4Ps98Hm/3gznbvhdjFbZbi4
oZ3tAadrLE5K9JiQaJYOnUmGpFB8PPwDR6chMZeegSQAW++OIKqHrg/WEh4yiuPf
qmAvX2hZkPpivNJYdTPUXTSO7K459CyqbqG+sNo02kc1nTX185RHNrVKQK+LOYYWY
1Q+hWDCCA88wggK3oAMCAQICEzdBBXntdX9CqaJcOvT4as6aqdcwDQYJKoZIhvcN
AQENBQAwVTENMA8GA1UEChMESUVURjERMA8GA1UECzMITEFNUFMgV0cxMTAvBgNV
BAMTKFNhbXBsZSBMQU1QUyBSU0EgQ2VydGlmawNhdGlvbiBBdXRob3JpdHkwIBCN
MTkxMTIwMDY1NDE4WhgPMjA1MjA5MjcWnU0MThaMDsxDTALBgNVBAoTBE1FVEYx
ETAPBgNVBAStCEExBTBVTIFdHMRcwFQYDVQDEw5BbGljZSBMb3ZlbGZjZTCCASlw
DQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBALt0iehY0BY+TZp/T5K2KNI05Hwr
+E3wP6XTvYy6WWyTgBK9LCOWI2juwdRrjFBSXkk7pWpjXwsA3A5Gotz0FpfgyC70
xsVcF7q4WHWZwleYXFK1QHJD73nQwXP968+A/3rBX7Ph00DBbZnfitOLPgPEwjTt
dg0VQQ6Wz+CRQ/YbHPKaw7aRphZO63dKvIKp4cQVtkWQH6syTjGsgkLcLnau5LZ
DQUdsGV+SAo3nBdWCRYV+I65x8Kf4hCxqmqjV3d/2NKRu0BXnDe/N+iDz3X0zEoj
0fqXgq4SWcC0nsG1lyyXt1TL270I6ATKRJGJWiQVCCpDtC0NT6vdJ45bCSzsCAwEA
Aa0BrzCBrdAMBgNVHRMBAf8EAjAAMBcGA1UdIAQQMA4wDAYKYIZIAWUDAgEwATAe
BgNVHREEFzAVgRNhbGljZUBzbWltZS5leGFtcGx1MBMGA1UdJQQMMAoGCCsGAQUF
BwMEMA4GA1UdDwEB/wQEAwIGwDAdBgNVHQ4EFgQUu/bMsi0dBhIc164papAQ0yBm
ZnMwHwYDVR0jBBGwFoAUkTCOfAcXDKfxCSHlNhpNHGh29FkwDQYJKoZIhvcNAQEN
BQADggEBAH0JoJanzqmgaSN3/gqSQ4cbbmdj/R40BEPr+gXT+xiidfZ2iLnWYyTn
euK6AChwKfnNvOFb81V1iffRtF/KtmVEDMR/sYeqAH83KM5p3e121Vh40HhyI0qN
uz5oShNaACSioQ23WxHGvy9vsdVfnbhsplrWg9NQ2WbpCmK+2oMh2oY10Z/wvXMt
9cG6jbMvcdH4z0IOvg6mrYkKTM/RCGnumghxwYToj1OyD5Gs4D2IJCw+fx5ODxh5
2MbNRYXTus2ZPRPM8JXNQc4Gwv4km3M4rKnJDd6hnoQ9rNeozIcBVyybQYjfrgg4
DRvW9Ksk22OH4ConlB8f7R7s1LM2cSYxggIAMIIB/AIBATBsMFUxDALBgNVBAoT
BE1FVEYxETAPBgNVBAStCEExBTBVTIFdHMTewLwYDVQQDEyhTYW1wbGUgTEFNUFMg
U1NBIEEN1cnRpZmljYXRpb24gQXV0aG9yaXR5AhM3QQV57XV/QqmiXDr0+GrOmgnX
MA8GCWCGSAF1AwQCAaBpMBGCGSsqGSib3DQEJAZeLBgkqhkiG9w0BBwEwHAYJKoZI
hvcNAQkFMQ8XDTIxMDIyMDE1MDEwLwYJKoZIhvcNAQkEMSIEIESMi+9/LU1D
fGjj+6U50VNLfxbzvyVJ0wzwnTS114DyMA0GCSqGSib3DQEBAQUABIIBACJHeayB
UllC4GdcgdojTUjoeIy6UIbrSg/aKZgAkCB8Dwq0hdU10qiun6WKI/TxM5izpRvL

UsNBGmqknPBMFhvwX6KCrwFk0p0j5Y5DZqX30deiQiGTUv3NiwZGTrKJ3JkyymFO
 HGbe5Thrq3inRLVfileuIZewaJsnJhKfnEq9fS09icTJ5o1PDAH6mZbW6hpYmU3F
 KBk2qJNqJX6bo60rCogu3wXDj0wxnqEXmeNDH5/+L9UVZur+EWzviUc8Ldd/kP3L
 DOO7ivs10bAWe8Tbw7NjuP8ZlVvzcvj3nXWzZzxh2ymDIOvyJA+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 4156 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="76c";
  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
```

```
--76c
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
```

```
--76c
Content-Transfer-Encoding: base64
```

Content-Type: application/pkcs7-signature; name="smime.p7s"

MIIJ4AYJKoZIhvcNAQcCoIIJ0TCCcC0CAQExDTALBg1ghkgBZQMEAgEwCwYJKoZI
hvcNAQcBoIIHpjCCA88wggK3oAMCAQICEw8tJb0ROZdKzkJU6HuPTQGirQwDQYJ
KoZIhvcNAQENBQAwVTENMA5GA1UEChMESUVURjERMA8GA1UECzMITEFNUFMgV0cx
MTAvBgNVBAMTKFNhbXBsZSBMQU1QUyBSU0EgQ2VydG1maWNhdGlvbiBBdXRob3Jp
dHkwIBcNMTEwMDY1NDE4WhgPMjA1MjA5MjcwNjU0MThaMDsxDTALBgNVBAoT
BE1FVEYxETAPBgNVBAsTCExBTvBTIFdHMRcwFQYDVoQDEw5BbGljZSBMb3Z1bGFj
ZTCCAS1wDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAJqVKfqlLwLjj+gBUCfk
acKTg8cc2OtJ9ZSed6U3jUoiZVpMLcP3MUKtLeLg9rlmAfIDlB/wlbdmadXPmrsz
yidmbuZmOpB5voVQfiLYYy3iOx7YOqzXrl6udP07k0sV+UdSNRFXrfKeoQEFXgOa
Gdmnx4OG/e3plfIKM0dPzZLoOAJF5m500xzXPL74zFCWp2f1ZkuE4A6141koaZXC
N5XL7wWTLMLNf9Byb5ksKqUuqEHAMdlmNMgY9VfVfcrv9w43GG8FtpSX+TWz
B2zNS2OF+XIVnzRG5DeoULq8v88Z5bLpLJ/nx26r8A4SSwIBaVv4wPxAf1iPsIVK
arUCAwEAAaOBrzCBrdAMBgNVHRMBAf8EAJAAMBcGA1UdIAQQMA4wDAYKYIZIAWUD
AgEwATAeBgNVHREEFzAVgRNhbGljZUBzbWltZS5leGFtcGx1MBMGAlUdJQQMMAoG
CCsGAQUFBwMEMA4GA1UdDwEB/wQEAwIFIDAdBgNVHQ4EFgQUo1NB1UQ8gCkVfAEj
8OeOr83zdW8wHwYDVR0jBBGwFoAUkTCOfAcXDKfxCSHlNhpHGH29FkwDQYJKoZI
hvcNAQENBQADggEBAIFJeKCsTKcFqQMpTryuRGzJdYA+R9eBAuDLsatbtKt14F
zkRyOg31/+Cw7H8e30iLrPIfLWN1qjHrjgOyIs5AQ/hgxLvLir3hEUV2Z3MRsMt
jH2x9SG91PEM046gfPnc9gMGHjMTg1qvaKcLQP5UzPEYPLror2X4P5uXxaP0LIZR
zWmkw1RF7FOD7PFB5v94M5274XYxW2W4uKGD7QGnUZROSvSYkGiWdp1JhQXwfDz8
A0enITGXnoEkaFvviCqh64PlhIeMorj36pgL19oWZD6YrzSWHuz1F00juyuOfQs
qm6hvrDTqNpHNZ015fOURza1SkCvi9GFmNUPoVgwggPPMIICt6ADAgECAhM3QQV5
7XV/QqmiXDr0+GrOmgnXMA0GCSqGS1b3DQEBDQUAMFUxDTALBgNVBAoTBE1FVEYx
ETAPBgNVBAsTCExBTvBTIFdHMTExLWYDVoQDEYhTYW1wbGUgTEFNUFMgU1NB1EN1
cnRpZmljYXRpb24gQXV0aG9yaXR5MCAXDTE5MTEyMDA2NTQxOFoYDzIwNTIwOTI3
MDY1NDE4WjA7MQ0wCwYDVoQKEwRJRVRGMREwDwYDVoQLEwhMQU1QUyBXRzEXMBUG
A1UEAxMOQWxpY2UgTG92ZWxhY2UwggEiMA0GCSqGS1b3DQEBQUAA4IBDwAwggEK
AoIBAQC09InoWDgWPK2af0+StijSNOR8K/hN8D+1078oullsk4ASvSwjsCNo7sHU
a4xQU15JO6VqY18LANwORjrc9BaX4MguzsbFXBe6uFhlmVpXmFxSpUByQ+950MFz
/evPgP96wV+z4TtAwW2Z34rTiz4DxMI07XYNFUE0ls/gkUP2Gxzyms02kaYWTut3
SryCqeHEFbZfKb4urMk4xrIJC3CzWrus2Q0FHbBlfkgKN5wXVgkWFfiOucfCn+IQ
saqpol3f9jSkbtAV5w3vzfog8919MxKI9H614KuElnAtJ7BtZcs17dUy9u9COgE
yKriVokFQgqQ7XNDU+r3SeOWks7AgMBAAGjga8wgawwDAYDVR0TAQH/BAIwADAX
BgNVHSAEEDAOMA5GCMCGSAFlAwIBMAEwHgYDVR0RBBCwFYETYWxpY2VAC21pbWUu
ZXhhbXBsZTATBgNVHSUEDDAKBggrBgEFBQcDBDAOBgNVHQ8BAf8EBAMCBsAwHQYD
VR0OBBYEFv2zLlItHQYSHJeuKWgQENMGzmZzMB8GA1UdIwQYMBaAFJEWjnwHFWyn
8QkoZTYaZxxodvRZMA0GCSqGS1b3DQEBDQUAA4IBAQBziaI2p86poGkjd/4KkkOH
G25nY/0eNARD6/0F0/sYonX2doizcGMk53riugAocCn5zbzhW/JVdYn30UxfyrZ1
RAzEf7GHqgB/NyjOad3pdpVYeDh4ciNKjbs+aEoTWgAkoqENTlsRxlcvb7HVX524
bKZaloPTUNlm6QpivtqDIdqGJdGf8L1zLfxBuo2zL3HR+M9CDr4Opq2JCKzP0Qhp
7poIccGE6I9Tsg+RrOA9iCQsPn1+Tg8YedjGzUWF07rNmT0TzPCVzUAuBlr+JJtz
OKypyQ3eoZ6EPazXqMyHAVcsm0GI364IOA0b8PSrJNtjh+AgJ5QfH+0e7NSzNnEm
MYICADCCafwCAQEWbDBVMQ0wCwYDVoQKEwRJRVRGMREwDwYDVoQLEwhMQU1QUyBX
RzExMC8GA1UEAxMoU2FtcGx1IEExBTvBTIFJTQSBdZXJ0aWZpY2F0aW9uIEF1dGhv
cm10eQITN0EFee1lf0Kpolw69PhqzpqplzALBg1ghkgBZQMEAgGgATAYBgkqhkiG
9w0BCQMxCwYJKoZIhvcNAQcBMBwGCSqGS1b3DQEBJBTEPFw0yMTAyMjAxNTAyMDJa

MC8GCSqGSib3DQEJBDEiBCBBQlio2vX/u19qayJlCm1QL6VZY0fBeGz9o7nEzCRO
 +zANBgkqhkiG9w0BAQEFAASCARvWkQYbbPuADZ7KqyO9LuESdEfBxOF80sHKNz
 UXrHZo8JdKaKxr/cTAuzBvoTxsmqvzP3ItCBm+javqX22+tHTpqisz5jkoiWyNVS
 e+F++YX8mXokgQpY26mZ+15Mv8pYYhptn6zdkRU1+QOwwlDCc6ykkCZeXyc+Hf7c
 xqM6SqPMQ+G7wIF6P2jHCId8Xyl7sdbL0i6PjotesHU+7nQsCjgI/iVR/ubWUdFX
 CTg8HVy4p683V3Y9DoRNP4MlUdmon8JasHDvA0240JcXxhJnlzEYa4gOnwgu3kh9
 3Y+NeucYCT0bXCBq2RLVQSpdNZfScXKL9QvZ3FtB0r6Bmtky

--76c--

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
```

MIITXAYJKoZIhvcNAQcDoIITTCCE0kCAQAxxggMQMIIBhAIBADBsmFUXDTALBgNV
 BAoTBElFVEYxETAPBgNVBAsTCExBTvBTIFdHMTewLwYDVQQDEyhTYWlwGUgTEFN
 UFMgU1NBIEENlcnRpZmljYXRpb24gQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIeh7j00
 Boq0MA0GCSqGSib3DQEBAQUABIIBAE1K2Qo2Ln5O6L9qgFnOdvuAuXnh2dLiYWI
 tX7B9W2VMQctrxTipZfUe+Y4oV/Rxifp4gChJ2lCgt6A4hHyApDlyNqmRlpCT+ky6
 jOJlr907Jzy9nIADEjaeKTIHePPWEWPiF3Otlrvq25NobNAE/dzcSgaS+SHsfPgu
 vW6gA+lfzdoOKIWNv11AJfbDRw8DeDi5n8ZPLkb/gYteBpY5mC2Iu8TebZ5qstQH
 i8G01K4xb6E7eMdXKx+gyDxox1P79E4q3dCKwYPK/C6B3AaY52WW55js9mb79OH5
 6/XvIEez581v4a9d0iY7g+aoARyTPE9Z79miRYT0aagyYhblb14wggGEAgEAMGww
 VTENMASGa1UEChMESUVURjERMA8Ga1UECXMITEFNUFMgV0cxMTAvBgNVBAMTKFNh
 bXBsZSBMQU1QUyBSU0EgQ2VydgGlmawNhdGlvbiBBdXRob3JpdHkCEzB8R0APhiY6
 HGLS64MvlsDXhpQwDQYJKoZIhvcNAQEBBQAEggEAWANrcGMnwYd7bg/TA9Wagm3q

dbiZLg3NxHQZRLRySCFht5wGkq1XcD7bWYwF0hSKiI4AJxJapfGUDEpDk1FYBU4r
9zS/e1rwCnhwp09sLfbJPRVvMTgTZuCOaY25ovZWvWtkS9MRDH+WoM5SNTf4vHHu
kjcSx5hafbhyc5pPLLTryIjObYgKraIMBXix7XKtSR/G7uD+HSIzhYUXqY0q2uQ
w7XiijbRd4bq9zqBbXriYyhFdo/JsBnYckjmmKcTLp6DfYTEzILKBJOepEiY5X4J
0JPeFyGxs7WSKdp1JZLZtjbMwvtEuUAWZ+iXDr1x/rQhq7mZIWqIbG6QpxYX6zCC
EC4GCSqGSib3DQEHATAdbglghkgBZQMEAQIEEBDwXZa6LrdPCgLubNCkd3qAghAA
kaaty8gkFo4+y5iWeOqsbZ9paegmFbiGsTQxrta64sj8znKQfQKz6/g055IcDixI
STqxPMV+w01jv6+Azoy9qJP29UTL0mXAP0LDionSBTn/4VAwBMSUDRus6jkq045K
UXxmIpc03SeOnpCLksyij6QlnAO24SbKsBex7R5EXYXU7W1G/PCoz9SW1YrQuXJ9
cU5ONWldvYE4/WeDlm3pjbv3XKLNEWiaUIV0lKFRhR4v+FUedn6dlVYDgfJrH8xDC
kW9gQvI1ZBbnBOR/zkoDhMMKtTgTvmzLIauDEi2RWKzlvwCattvIkkrjt+SwWpvr
oc6i58XfCx/d0YHPp5AIU8pslawDtQXe5ecACY9J/K0OgX1G51HI+O2XMC9S9QYn
YPxA+CsRxmKhZQv9au48aQwmLBkhkXZq7FCve8GTnCLdU5AmtP6ff59lga7+hfb
VSz+jSodBL1Wn1IKw/lrBvXFem/A4mtY/W9y9EVhGyREuhoZDCiGRo/bPsyDNZBS
WAsjHLI3NJeUgHFFcEn5xOwDmhmJOehzs712pqrzMd0VrT4hALvvhSGB7nybL5dR
pabbxtpBgqzlwu6eoX1jSh5bF8/RsAJ8ldxvn8AWcFc8q81YfYOzjqf7ZnuumT10
18/rdepv/nfyiYCRhr2Eekj0F3bXj1TGloeCNTuUPcNHVX6+hQ7FY2CJm9JCqNhL
7whKhq+kKJuPugHble5d2rJfKnhRMIJAg8QqKy9eqKct4gW5FFT70wyB15YToJb
qVxb3BEZ6ulshpZ9IGVzS0Jmvke+Ptze86it00fQIJWfrFqoag83GcCuQEYyECIc
HXWFsZ1bQ1UD2+YSWBOzRBuUuJ3U66w3J5oDAYfYnieFNpUP0dhaAMsu7QQfLSza
T/GbSibQoFXcDx6MaZ5fbZ1iduvoZZfERNMe5vN+q/w9Lx5e8hf1EZmTNMuoRn90
wft/wuM06Cc8FR2Ft7QLu80jqePQ6tAYwvA5QOvpBN9A82DUWz0I9eRD19+S8Z+I
QgjbPcZ0ACFqLcFbT6uzrKp2vGSrA+IcS89+qBB+sKbtWPgTrK7Q1Jgc7NpHGyhZ
BlTAVXv4fPngqn+gSqGuerD/xmvszHMIHq6Q4ADxbxDE4R0yoV2afXUVyAMo85Q
eNG5WJ83Z12msJqx1+1EUzZoQXxvrZhm0bMziCjV/P1cu/ChtmuemopRxpplLbJv
/mChRaKv9TotDy2Dwzf5N5Xy58gb/0ktMXMdGpYts9awYc742TCscrTqutBAXtNM
dXA00yelkVHBBRCrcoUEWWhUGQKYmK0NQIpxduJYcLLhKMI+2QfyfdkODp1EtXbX9
LaZhPRI9osmmF0fnSkmt2mtD+W8uxBF7espDkUsidb8NiUtZBrSqTADQUIuAw5xG
322wFZ0DtPfm6nHpbYBfIGlIR4LyqTzyaSRJtMkMiDFgnMWrfNF6pMsToo+4GbARo
MWM9mq4XSMrKAinQu7T8UGWot9bMfMjrTrpfETgQCL4vur9nI1CbgcPWW14U2oBW
21T1duS0o2eRpeGA93U6zF7BbCm1EqPK45Qmm78NwMcI9i4GgHSG2ssEn8URmv0L
qp9+UmkhvLT26dZtkB0wPMEVOIWx3e+F34eVzno5jAbiJxuUIdDPDwQg7xtcLiF
lRsaiGx7MtWsP6paqGBrYdHcXnt8P8k2ywNqRicTSThG0P09CNDWFwNaKa+9Ia7a
EnWoFmNoNm/IUH+wbRQUnT7oh0qU2mxdgMnygDhEELe1+4tGCTAPThxSU3gxQyv0
w686bzZP9uGLoRfivmXKm73Wu0HtUefT1rNdPsJDfQEfo8mEY4EDMh+Fa50S9Yj6
SGe8X9jDateJLd+yL7xEvdEQ7FxHbqo7twj/g4Im0OeG2ngEchWlYcuOrlgog4bv
kWwcMhOCcQ/9242sgCTG/ATAVlix0Z16/WCzzY60Zxk1eAlP3Ar9NiQHGuVC1R0o
QxhlP/1KvyVMAQTtuEposNLUDXMydq81VrFuopYeJ3NJOPE7eA4BeIXNyhrhxqfX
j23tfb3/C4uHEmgjnfW1LZIjwWrOjoEZA2+lG+Si7YQWLLJWFNqEEH2rpxQMnvwv
282dIYpyY14PDLLN5nMltY8MeMaNp6Q8rOwTDozmmZ9RONzbKJL3FxSVENKgdJTf
v+gpLOvXou6qDdidAqxErGM0j68g8Rnsdw7Lj3FQH7JjLZiR3EQgGxRKDwTsV1rW
ODtsNyKBtHDBOn/zOFTmgTVpYol2x/kV22C1Wn9ZArHFgZDxDyDjjJqxJwHlgVdE
J+bUZ1C5DatXxvpFhrTpUz1dvsTsQ48cmepEiEnqYO/33uU7KIqjBxY527dagnR
q01ntVycY4wiLKjuJHHHy/b250RyxS/x6nVYJsORNXsvYcZ1zqHC7uh9eQStAyj6
zotbPet++u2REXKSzwhI+6mTCrFkfeHxt3BqTPAxHPxsZAmquayksNs8e94G5LnD
VLAbdtwuIdeuz3rDWObafnaOVXD8vzjoMpiZcYKubb9pdFQIdxpYXPYqWz2f+c8g
9VnLXaJpwqByOPtLT5knKWMbsXJ5Gc8sNIGl1blYnj5ao+z6JNV2qqWA8dukPM5Q
/KwmBvR9/RijeIEPGoqRcwUi92fuvVJV7oZf2ZCCGMLw8W4pSrzsfs/xdOJs1rTgN

trDrAOKlraCKJQ5zHwZyg+c65KUe+5voj4WTu27g/vWTmPjF70htA+UIYcsNVYU9
yGuznj6x/2EV7rLsUTpMqMFN0s4dQl4Hhfr4gaoDROb7bOdkVtWAavP4c18w1JA9
08X9kQNPqID0M0NOruz8JO8gyTlxyAmopnEDREvMT7JCGuwPM9YRE64pVPOZlAZm
STC7LY11zMhZL+RvhwBwQjKKeKN3hQM4/45BHGFVgg6k5iobcv78lZHW028SWila
dEgJLSobB9ieOTfrWqBrBBHjpaDwuyjS+QwjsF8SFLdRD5TY1IugUvW5Swnucikh
XlrK/FaRRQJGzUesrkN06LlpFiiRyW9nuDjdpaKV4P9pkeJHmtN3KF95LjJnXs+Z
07cF0sX2K7FY4GCfFxGPSsqbcR/6zAFHVPjgPGDH51yOTe05RWLhgGEWqt7mIeSD
ppJdnY1LDfK0AFbXAFnJxhNwlfJiLB4vdsFqXGSYXFAjns8vZR62PgSExxUMxrO6
P7oIAYisiU+9XuG40ok8RfCZgN2Qdy5oNDbyow8x3XR4BQu8+2sT9nLvJosjYnHt
8yHMhhAbJl5VVK1EaB2gMxmAISiCCKQQ4YlStMc/LUkl8XOdQmf9SF0LlpuuGEpM
V3BhXNxCreiXA8ulMtnytw++lhl3qapALVu5OsJBQ2sqrhC7VhZTfiRQHr5s/i97
OrBblZHv48NblW+tsS0Vl+jW/7AMUvQO+j7wYDI8Q2GplujJ08iHxZw/YDJR+up4
bmQjK3xySaCi9Ef58KY0j0Y8ITvS6lGMn0bCkL23UGNwISo2gPEcStd0ksZtlvGX
X37skWsFPD3M85DqQeckjv3PFzGQL7ZZLUQmmYqWg43DKrDJSZlD7VYHmTY0rrMj
gNo6iqzI+6Ygi81y14ZWTVeOFIH9tOKvjtuJz+90Qi9vEbDqF43+hiyWVg/aOke8
4TGy7BZp5j/+SCr78/LvTko/5gafEymhaQmmsR7hskt3AhjfTyUfq/cAtuIm39U2
MmXRwPdrzWASgy/lF0QnrgB0T85+ID58J9VaP78mI/BtKO20wWMTjbbabR7J3Rn+8
KW4H6eewVWBqghCnsJQuqibbZeFDjFgJ9kIaTvGD0TBehpp9TidmppXM4Dl4J+V/
u7dSL257Dz1Kkk42gK4Cs0PlDzwe888KIABF38AZ8dnWtD492eYxA9We6NB2rulo
K59o1oZdn+s1cF3DLfvPpyfkZ8o3EVgAPVXiDfHWuVp1gL8Cv5ahVlk9BJSd1CgC
Vwsm01V1E7QeNh3gNdQI88tu4wh5SVFk4U2cYI+dDMFUVDMzrUI3tKvWXNZOzn4V
Ce6Eu2JPiCcoYUwDHpsq5aJ9BPKBguhQqybDpAAkgSZLwhzAD7rEvo8TU8gzZ2KZ
zH506GoFtU4oNinnrvyHX96/bG/VlizeOE9YtQNYEfxxSOBsZD9jgd1pG4j/FDF1Z
Ib+KUuo8Y7GKlOu+1+/WIVcp0nIsyIC4zGdM6DThCT6nGrhKboduTgF5NRH/Hf03
Vrbj/ZarK0tlgzbzPgxotZiUfCvEuav9AVqxA2Zq5afs6bRfohqyFqwkHiYVl9C4
m00v4HisEFDDGG3f5+Zj/x6tnX9QxR81DOomUooh8aYs/iAz0nrKyux6GMHSLj8db
UbvQ+1VvNE3Fj0xu46HkKzGtFqpgXxzDLkE9e7NJ+Hw4tbOLfINQ0qS7iTcJmbwg
snexBuL6rf8NF28EdlqQzCPLZVhnOd1+KKJS7V/M8u/R/y22+IXzFSA2TlxhId09
IduZ3ByCz2HFJfVj7SameC3KANbRnBkdud1hclIBDS5Hhpqk4M8i3zmZRZWgLyjR
edtSaHuJAlHiKgAtQVeIz1L6Ilw3jVoHL0vOdISoQpoWWhejB9f47KRmUdbd5Pxb
Ot2y1XJKYFfoCQUs1xkNAaynSJAj97yEAZm7aDmE4bjs33pz4L3nYxO/KUY6EB/E
eGgPk3CdvT2JY5BuF0xXYRKQgZ06c9mXzavJJXXWQUUB5k2QG0uyKpmwNr2sdJQ
A8ehhmgGws+7qXwZQEcnC3W0vniGOBDYP3JVJPiNLFVQN9k8C1E7+0emFn2UcNyG
294h01G0uBPAbCdHAYDnNpVj5RS0EgY647agQHyp/gjSt4XeoacIKaalb4iGpT+C
4r2BqRcVUCdE3MRQFqiT6ccm+8h8eA7xtMB8c9OgUTEIKK/WSc0DUsCJB62Plgtj
KJ4xXQXTzzUCDMnACFp6mBTd3g2ZbnfHKSyJdAvPigVbA+Qhy2eWUTYpi6yjtIyT
eaQ2qafGppn85oLFkdgdme3Ty1UxOpAsqLyN1NAa6YT3D/0Jl3VnfhFKlmywWIG6
Z2SLd0r07xoBUuAKHkFUuRauGYbVbU/Frmdylv6I9DhCqV/XEDa/tHOa/LWugvb+
x5A+g+kZiTWRRLZYHungyjqUaf/zeJsPYRoQEi4KHAQ30xCDk/dhWdhDBnUXT8P
hzMj8VN3yJQA1vMNA5uefj2/+MIkLkz6+XP1/lJNLFHYi+EERgxJ2mFm/s02h9NF
NhyWBSBtsEwi+rVbfcRRBpVjR5MwUohNHMGxwgj7rzvUkDe47ueXDP74j+Jc1O68
r4jQ3sob123uSYryDHBZxZSbwjFU2ufe8W+XL/NGwTw04alHZfKsH4x4ZbGqwunf
U41kcOY/iJmuhL5mn2YYUE6w4oywZuLx5WCv2oAvQawMmNP9AeI1jCv9JiKa+8y0
sAa1LzD78Dg4FK08t3d13Q==

B.1.1.5. 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 1371 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="428"
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
```

--428

```
MIME-Version: 1.0
Content-Type: multipart/alternative; boundary="db9"
```

--db9

```
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
```

--db9

```
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>
--db9--

```

```

--428

```

```

Content-Type: image/png
Content-Transfer-Encoding: base64
Content-Disposition: inline

```

```

iVBORw0KGgoAAAANSUHEUgAAABQAAAAUCAyAAACNiR0NAAAAcELEQVR42uVTOxbA
MAgS739nO3TpRw20dqpbfARQeJ0ywiwYnCtkDKnbcLk66sqlT+zt9cidkE+6KwkZ
sgrzfcqVMpL2jo0447gYDpeArk+OnJHkIhAfTPRicihAf5YJrw7vjv0ZWRWM/uli
vdPf1QZ2kDD9xppd8wAAAAABJRu5ErkJggg==

```

```

--428--

```

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

```

```

MIIPHwYJKoZIhvcNAQcCoIIPEDCCDwwCAQExDTALBglghkgBZQMEAgEwggVIBgkq
hkiG9w0BBwGgggU5BIIFNU1JTUUtVmVyc2lvdjogMS4wDQpDb250ZW50LVR5cGU6
IGl1bHRpcGFydC9taXhlZDsgYm91bmRhcnc9IjExMCINCg0KLS0xMTANck1JTUUt
VmVyc2lvdjogMS4wDQpDb250ZW50LVR5cGU6IGl1bHRpcGFydC9hbHRlcm5hdG12

```

ZTsgYm91bmRhcnc9IjE5MyINCg0KLS0xOTMNckNvbnRlbnQtVHlwZTogdGV4dC9w
bGFpbjsyY2hhcnNldD0idXMtYXNjaWkiDQpNSU1FLVZlcnNpb246IDEuMA0KQ29u
dGVudC1UcmFuc2Zlci1FbmNvZGluZzZogN2JpdA0KDQpUaGlzIGlzIHROZSBzbWlt
ZS1vbmUtcGFydC1jb21wbGV4IG1lc3NhZ2UuDQoNC1RoaxMgaXMgYSBzaWduZWQt
b25seSBTL01JTUUGbWVzc2FnZSB2aWEgUETDUyM3IHNpZ251ZERhdGEuICBUaGUN
CnBheWxvYWQgaXMgYSBtdWx0aXBhcnQvYWx0ZXJuYXRpdmUgbWVzc2FnZSB3aXR0
IGFuIGlubGluZSBpbWFnZS9wbmcNCmF0dGFjaG1lbnQuIEl0IHVzZXMGb8gaGVh
ZGVyIHByb3RlY3Rpb24uDQoNCi0tIA0KQWxpY2UNCmFsaWNlQHNTaW1lLmV4YW1w
bGUNCi0tMTkzDQpDb250ZW50LVR5cGU6IHRleHQvaHRtbDsgY2hhcnNldD0idXMt
YXNjaWkiDQpNSU1FLVZlcnNpb246IDEuMA0KQ29udGVudC1UcmFuc2Zlci1FbmNv
ZGluZzZogN2JpdA0KDQo8aHRtbD48aGVhZD48dG10bGU+PC90aXR5ZT48L2hlYWQ+
PGJvZHK+DQo8cD5UaGlzIGlzIHROZSA8Yj5zbWltZS1vbmUtcGFydC1jb21wbGV4
PC9iPiBtZXNzYWdlLjwvcD4NCjxwPlRoaxMgaXMgYSBzaWduZWQtb25seSBTL01J
TUUGbWVzc2FnZSB2aWEgUETDUyM3IHNpZ251ZERhdGEuICBUaGUNCnBheWxvYWQga
XMgYSBtdWx0aXBhcnQvYWx0ZXJuYXRpdmUgbWVzc2FnZSB3aXR0IGFuIGlubGlu
ZSBpbWFnZS9wbmcNCmF0dGFjaG1lbnQuIEl0IHVzZXMGb8gaGVhZGVyIHByb3Rl
Y3Rpb24uPC9wPg0KPHA+PHR0Pi0tIDxici8+QWxpY2U8YnIvPmFsaWNlQHNTaW1l
LmV4YW1wbGU8L3R0PjwvcD48L2JvZHK+PC9odG1sPg0KLS0xOTMtLQ0KDQotLTEx
MA0KQ29udGVudC1UeXB1OiBpbWFnZS9wbmcNCkNvbnRlbnQtVHJhbnNmZXItRW5j
b2Rpbmc6IGJhc2U2NA0KQ29udGVudC1EaXNwb3NpdG1vbjoGaW5saW51DQoNCm1W
Qk9SdzBLR2dvQUFBQU5TVWhFVWdBQUFCUUFBUFBQUFVQ0FZQUFBQU5pUjBOQUFBQWNF
bEVrVlI0MnVWVE94YkENck1BZlM3MzluTzNUcFJ3MjBkcxBIzKFSUUVqT3l3aXdz
bkN0a0RLbmJjTGs2NnNxbFQrenQ5Y2lka0UrNkt3a1oNCnNncnY3FWTXBMMmpv
MDQ0N2dZRHBlQXJrK09uSkhrSWbZlRQum1jaWhBZjVZSnJ3N3ZqdjBaVlJXTS9l
bGkNCnZkUGYxUVoya0REOXhwcGQ4d0FBQUFCslJVNuvya0pnZ2c9PQ0KDQotLTEx
MC0tDQqgggemMIIDzCCAreGAWIBAgITDy0lvRE510roQ1SHoe49NAaKtDANBgkq
hkiG9w0BAQ0FADBMQ0wCwYDVQQKEwRJRVRGMREwDwYDVQQLEWhMQU1QUyBXRzEx
MC8GA1UEAxMoU2FtcGx1IEExBTBVTIFJTSBDBZlR0aWZpY2F0aW9uIEF1dGhvcml0
eTagFw0xOTExMjAwNjU0MThaGA8yMDUyMDkyNzA2NTQxOFowOZENMA8GA1UEChME
SUVURjERMA8GA1UECxMITEFNUFMgV0cxZzAVBgNVBAMTDkFsaWNlIEExvdmVsYWNl
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAmP+ovBouOP6AFQJ+Rpb
wpODxxzY60n1lJ53pTeNSiJlWkwtw/cxQq0t4uD2vWYB8gOUH/Cvt2Zp1c+auzPK
J2Zu5mY6kHm+hVB+IthjLeI7Htg6rNeuXq50/TuTSxX5R1I1EXGt8p6hAQVeA5oZ
2afHg4b97enV8gozR0/Nkug4AkXmbk7THNc8vvjMUJanZ/VmS4TgDqXjWShplcI3
lcvvBZMswt41/0HJvmswqpS6oQcAx3Weag0yCNj1V9V9yu/3DjcYbwW2lJf5NbmH
bm1LY4X5chWfNEbkN6hQury/zxnlsukgn+fHbqvDhJLAgFpW/jA/EB/WI+whUpq
tQIDAQABo4GvMIGsMAwGA1UdEwEB/wQMAAwFwYDVR0gBBawDjAMBgpghkgBZQMC
ATABMB4GA1UdEQQXMBWBE2FsaWNlQHNTaW1lLmV4YW1wbGUwEwYDVR01BAwwCgYI
KwYBBQUHAWQwDgYDVR0PAQH/BAQDAgUGMB0GA1UdDgQWBBSiU0HVRDyAKRV8ASpW
546vzfN3DzAfBgNVHSMEGDAWgBSRMI58BxcMp/EJKGU2GmccaHb0WTANBgkqhkiG
9w0BAQ0FAAOCAQEAgU14oJyxMpwWpAylOvK6NEbM1lgD5H14EC4Muxqlu0q2XgXO
SBHI6DfX/4LDsfX7fSIus8gWVY3WqMeuOA7IizkBD+GDEu8uKveERRXZncxGwy2M
fbH1lb3U8QzTjqB8+dz2AwYeMxODWq9opwtA/1TOkRg8uuivZfg/m5fFo/QshlHN
aaTDVEXsU4Ps98Hm/3gznbvhdjFbZbi4oZ3tAadR1E5K9JiQaJYOnUmGpFB8PPwD
R6chMZeegSQA++OIKqHrg/WEh4yiuPfqmAvX2hZkPpivNJYdTPUXTSO7K459Cyq
bqG+sNo02kc1nTXl85RHNRVKQK+L0YWY1Q+hWDCCA88wggK3oAMCAQICEzdBXnt
dX9CqaJcOvT4as6aqdcwDQYJKoZIhvcNAQENBQAwVTENMA8GA1UEChMESUVURjER
MA8GA1UECxMITEFNUFMgV0cxMTAvBgNVBAMTKFNhbXBsZSBMQU1QUyBSU0EgQ2Vy

dGlmaWNhdGlvbiiBbdXRob3JpdHkwIBcNMtkxMTIwMDYlNDE4WhgPMjA1MjA5Mjcw
 NjU0MThaMDsxDTALBgNVBAoTBElFVEYxETAPBgNVBAsTCExBTvBTIFdHMRcwFQYD
 VQQDEw5BbGljZSBMb3ZlbGFjZTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoC
 ggEBALT0iehyOBY+TZp/T5K2KNI05Hwr+E3wP6XTvvi6WWyTgBK9LCOWI2juwdRr
 jFBSXkk7pWpjXwsA3A5G0tz0FpfgyC7OxsVcF7q4WHWZWleYXFKlQHJD73nQwXP9
 68+A/3rBX7Ph00DBbZnfitOLPgPEwjTtdg0VQQ6Wz+CRQ/YbHPKaw7aRphZO63dK
 vIKp4cQVtkWQHl6syTjGsgkLcLNau5LZDQUdsGV+SAo3nBdWCRYV+I65x8Kf4hCx
 qqmjV3d/2NKRu0BXnDe/N+iDz3X0zEoj0fQXgq4SWcC0nsG1lyyXt1TL270I6ATK
 RGJWiQVCCpDtc0NT6vdJ45bCSzsCAwEAAaOBrzCBrdAMBgNVHRMBAf8EAjAAMBcG
 A1UdIAQQMA4wDAYKYIZIAWUDAgEwATAeBgNVHREEFzAVgRNhbGljZUBzbWltZS5l
 eGftcGx1MBMGA1UdJQQMMAoGCCsGAQUFBwMEMA4GA1UdDwEB/wQEAwIGwDAdBgNV
 HQ4EFgQUu/bMsi0dBhIcl64papAQ0yBmZnMwHwYDVR0jBBgwFoAUKTCOfAcXDKfx
 CShlNhpNHGh29FkwDQYJKoZIhvcNAQENBQADggEBAH0JoJanzqmgaSN3/gqSQ4cb
 bmdj/R40BEP+gXT+xiidfZ2iLNwYyTneuK6AChwKfnNvOFb8lVliffRTF/KtmVE
 DMR/sYeqAH83KM5p3el2lVh40HhyI0qNuz5oShNaACSioQ23WxHGvy9vsdVfnbhs
 plrWg9NQ2WbpCmK+2oMh2oYl0Z/wvXmt9cG6jbMvcdH4z0IOvg6mrYkKTM/RCGnu
 mgxwYToj1OyD5Gs4D2IJCW+fX5ODxh52MbNRYXTus2ZPRPM8JXNQC4Gwv4km3M4
 rKnJdD6hnoQ9rNeozIcBVyybQYjfrgg4DRvW9Ksk22OH4ConlB8f7R7s1LM2cSYx
 ggIAMIIB/AIBATBsMFUxDTALBgNVBAoTBElFVEYxETAPBgNVBAsTCExBTvBTIFdH
 MTEwLwYDQDEYhTYWlwbGUgTEFNUFMgU1NBIEIENlcnRpZmljYXRpb24gQXV0aG9y
 aXR5AhM3QQV57XV/QqmiXDr0+GrOmqnXMAsgCWCGSAFlAwQCAaBpMBGCSqGSIb3
 DQEJAZELBgkqhkiG9w0BBwEwHAYJKoZIhvcNAQkFMQ8XDITixMDIyMDE3MDEwMlow
 LwYJKoZIhvcNAQkEMSIEIAiYlRaTjUNCbHnrieg64m3mMEMTRF8kqt5E8+ogUh5/
 MA0GCSqGSIb3DQEBAQUABIIBAILQrmF19ls0ehRVddBjQESh5VnT+NxYWjofr2i0
 w5OoB4RU3+6bPs2i5Y+IZvdnQtkfux+L/Rmy+cK5tlK8J9taLXm3/mJO/57tW+C1
 E9WSBFb1Ik29FHbTuThrcSaE6Dr5zGwZBmlkcb3rx+AdYM8PMAhDd+ESwYwyjWk4
 A7zRNEAlpD4XZdiz0a/kULobW9W30KaQdJANQG0CX23puEW+wk9hzuuWX+IXeLwh
 4RlkXSigeWxlu44jrBGOzkr/UjonxvpjBzyvls6ltj0HekROzHy9tXEHyeP6BOzC
 kWKI9KZRyeZenYIOJRgqicDLdDgrZN5AoQqE+rBlK5i82l0=

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 5199 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="e18";
 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

--e18
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="831"

--831
MIME-Version: 1.0
Content-Type: multipart/alternative; boundary="ale"

--ale
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
--ale
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 smime-multipart-complex 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>--
Alice
alice@smime.example</tt></p></body></html>
--ale--

--831
Content-Type: image/png
Content-Transfer-Encoding: base64

Content-Disposition: inline

iVBORw0KGgoAAAANSUHEUgAAABQAAAAUCAyAAACNiR0NAAAAcELEQVR42uVTOxbA
MAgS739nO3TpRw20dqpbfARQEjOywiwYnCTkDKnbcLk66sqlT+zt9cidkE+6KwkZ
sgrzfcqVMpL2jo0447gYDpeArk+OnJHkIhAfTPRicihAf5YJrw7vjv0ZWRWM/uli
vdPf1QZ2kDD9xppd8wAAAAABJRu5ErkJggg==

--831--

--e18

Content-Transfer-Encoding: base64

Content-Type: application/pkcs7-signature; name="smime.p7s"

MIIJ4AYJKoZIhvcNAQcCoIIJ0TCCcC0CAQExDTALBg1ghkgBZQMEAgEwCwYJKoZI
hvcNAQcBoIIHpjCCA88wggK3oAMCAQICEw8tJb0ROZdKzkJU6HuPTQGirQwDQYJ
KoZIhvcNAQENBQAwVTENMASGA1UEChMESUVURjERMA8GA1UECzMITEFNUFMgV0cx
MTAvBgNVBAMTKFNhbXBsZSBMQU1QUyBSU0EgQ2VydGlmawNhdGlvbiBBdXRob3Jp
dHkwIBcNMtKxMTIwMDY1NDE4WhgPMjA1MjA5MjcwNjU0MThaMDsxDTALBgNVBAOT
BEI1FVEYxETAPBgNVBAsTCExBTvBTIFdHMRcwFQYDVQQDEw5BbG1jZSBMb3Z1bGFj
ZTCCAS1wDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAlJqVKfLwaLjj+gBUCfk
acKTg8cc20tJ9ZSed6U3jUoiZVpMLcP3MUKtLeLg9r1mAfID1B/wlbdmadXPmrsz
yidmbuZmOpB5voVQfiLYy3iOx7YOqzXr16udP07k0sV+UdSNRFxrfKeoQEFXgOa
Gdmnx4OG/e3plfIKM0dPzZLoOAJF5m500xzXPL74zFCWp2f1ZkuE4A6141koaZXC
N5XL7wWTLMLenF9Byb5ksKqUuqEHAMdlnmoNMgJY9VfVfcrv9w43GG8FtpSX+TWz
B2zNS2OF+XIVnzRG5DeoULq8v88Z5bLpIJ/nx26r8A4SSwIBaVv4wPxAfliPsIVK
arUCAwEAAaOBrzCBzDAMBGNVHRMBAf8EAJAAMBcGA1UdIAQQMA4wDAYKYIZIAWUD
AgEwATAeBgNVHREEFzAVGhRnbbG1jZUBzbWltZS5leGFtcGxlMBMGA1UdJQQMMAoG
CCsGAQUFBwMEMA4GA1UdDwEB/wQEAwIFIDAdBgNVHQ4EFgQUo1NB1UQ8gCkVfAEj
8OeOr83zdw8wHwYDVR0jBBGwFoAUKTCOfAcXDKfxCSH1NhpHGH29FkwDQYJKoZI
hvcNAQENBQADggEBAIFJeKCsTKcFqQMpTryuJRGzJdYA+R9eBAuDLsatbtKt14F
zkgRyOg31/+Cw7H8e30iLrPIfLWN1qjHrjgOyIs5AQ/hgxLvLir3hEUV2Z3MRsMt
jH2x9SG91PEM046gfPnc9gMGHjMTg1qvaKcLQP5UzpEYPLror2X4P5uXxaP0LIZR
zWmkw1RF7FOD7PFB5v94M5274XYxW2W4uKgd7QGnUZROSvSYkGiWDp1JhQxWfDz8
A0enITGXnoEkAFvvjiCqh64PlhIeMorj36pgL19oWZD6YrzSWHUz1F00juyuOfQs
qm6hvrDTqNpHNZ015fOURza1SkCvi9GFmNUPoVgwgqPPMIICt6ADAgECAhM3QQV5
7XV/QqmiXDr0+GrOmgnXMA0GCSqSISb3DQEBAQUAMFUDALBgNVBAOTBEI1FVEYx
ETAPBgNVBAsTCExBTvBTIFdHMTewLwYDVQQDEyhTYW1wbGUgTEFNUFMgU1NBIE
N1cnRpZmljYXRpb24gQXV0aG9yaXR5MCAXDTE5MTEyMDA2NTQxOFoYDzIwNTIwOTI3
MDY1NDE4WjA7MQ0wCwYDVQQKEwRJRVRGMREwDwYDVQQLEwhMQU1QUyBXRzEXMBUG
A1UEAxMOQWxpY2UgTG92ZWxhY2UwggEiMA0GCSqSISb3DQEBAQUAA4IBDwAwggEK
AoIBAQC09InoWDgWPK2af0+StijSNOR8K/hN8D+1078oullsk4ASvSwjsCNo7sHU
a4xQU15JO6VqY18LANwORjrc9BaX4MguzsbFXBe6uFhlmVpXmFxSpUByQ+950MFz
/evPgP96wV+z4TtAwW2Z34rTiz4DxMI07XYNFUEOls/gkUP2GxzymsO2kaYWTut3
SryCqeHEFbZfK4urMk4xrIJC3CzWruS2Q0FHbBlfkgKN5wXVgkWFfiOucfCn+IQ
saqpo1d3f9jSkbtAV5w3vzfog8919MxKI9H614KuElnAtJ7BtZcs17dUy9u9COgE
yKriVokFQgqQ7XNDU+r3SeOWwks7AgMBAAAgjga8wgawwDAYDVR0TAQH/BAIwADAX
BgNVHSAEEDAOMAAGCmCGSAFlAwIBMAEwHgYDVR0RBBCwFYETYWxpY2VAc21pbWUu
ZXhhbXBsZTATBgNVHSUEDDAKBggrBgEFBQcDBDAOBgNVHQ8BAf8EBAMCBsAwHQYD

VR0OBBYEFLv2zLiTHQYSHJeuKWqQENMgZmZzMB8GA1UdIwQYMBaAFJEwjnWHFwyn
 8QkoZTYaZxxodvRZMA0GCSqGSIb3DQEBDQUAA4IBAQBziaI2p86poGkjD/4KkkOH
 G25nY/0eNARD6/oF0/sYonX2doizcGMk53riugAocCn5zbzhW/JVdYn30UxfyrZl
 RAZef7GHqgB/NyjOad3pdpVYeDh4ciNKjbs+aEoTWgAkoqENTlsRxlcvb7HVX524
 bKZaloPTUNlm6QpivtqDIdqGJdGf8L1zLfXBuo2zL3HR+M9CDr4Opq2JCKzP0Qhp
 7poIccGE6I9Tsg+RrOA9iCQsPnl+Tg8YedjGzUWF07rNmT0TzPCVzUAuBlr+JJtz
 OKypyQ3eoZ6EPazXqMyHAVcsm0GI364IOA0b8PSrJNtjh+AqJ5QfH+0e7NSzNnEm
 MYICADCCAfwCAQEwbDBVMQ0wCwYDVQQKEwRJRVRGMREwDwYDVQQLEwhMQU1QUyBX
 RzExMC8GA1UEAxMoU2FtcGx1IEExBTBVTIFJTQSBdZXJ0aWZpY2F0aW9uIEFldGhv
 cml0eQITN0EFee1lf0Kpolw69PhqzpqplzALBglghkgBZQMEAgGgATAYBgkqhkiG
 9w0BCQMxCwYJKoZIhvcNAQcBMBwGCSqGSIb3DQEJBTEPFw0yMTAyMjAxNzAyMDJa
 MC8GCSqGSIb3DQEJBDEiBCDXOvk8vYdge4ktwwFa4GFP+Zxia/eTOacb5ZgEXQA7
 WjANBgkqhkiG9w0BAQEFAASCAQAIbfuFI8gxAWPFjnahNo6lRRGWj0U1S4GkRl6h
 LCNh5x49ns9BM51cZp+s5KhQSxhFdmuru+wCwgRk7KjzckAnizh70/dEYJmsjSZl
 zmLEGmtQ+q9MoyydZD9s2l9891WDjsCFjVIlhRkLTI7Zeh6+wQQpGKDbv0MoYQ95
 a9HPz6DuuCjCTCv+rUEOAys4X+dQsgDx3hsSITVoKDR1lkHVmZnjC4Byce6HY0Gn
 cEg/VqBGK4R70/46XTk/EgLPsnSPLPfc8Pclkw6yyF+QNYLV4tKvOKRvNJGf+Pjy
 GvJithBGOKFbOtWPPY+nFTMT+aNODuyAVQUmlbQIvz0/WXvU

--e18--

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

MIIZDAYJKoZiHvcNAQcDoIIY/TCCGPkCAQAxggMQMIIBhAIBADBBSMFUxD TALBgNV
BAoTBELFVEYxETAPBgNVBAsTCExBTvBTIFdHMTEwLWYDVQQDEyhTYWlwbGUgTEFN
UFMgU1NBIElcnRpb24gQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIEh7j00
Boq0MA0GCSqSIB3DQEBAQUABIIBAJGYWhyOEdeaxA1hlsqTJL/nwL8aIuFtQBnq
8aptWsaRxbkbfwd639JspX9Jzhc4gu50hiKu1HdJ2+IL7vvPRB49SfqiCst+ImD3
syFxFHjbmJSfPDNNukyut/SYV+DAHbvGiGxB0vCT8iW+qbKgwvQYcm2Kcs0UYV7ek
NXA7wkNjIygcYRSbg7Xdhv9HcGGtIshTBvws9DaYwmjo/8IlrXfeIusKU7dhZgMK
bVVbotXAYlbEFH6vpDFWK5pc+DPgVPFe8iA8z02k8HdtXEM44g++0/chZAiqe8uw
UARmERg+5Y+2dROAVHRWFvlow6qWw71jBmtf55abK6jJFhSIzmowggGEAgEAMGww
VTENMASGA1UEChMESUVURjERMA8GA1UECXMITEFNuFMgV0cxMTAvBgNVBAMTKFNh
bXBsZSBMQU1QUyBSU0EgQ2VydgLmaWNhdG1vbiBBdXR0b3JpdHkCEzB8R0APhiY6
HGLS64MvlsDXhpQwDQYJKoZiHvcNAQEBBQAEggEAB9sGmAYY1DHHoMQbd734joYE
SjbbvkHEPyOAlJI7FfGdAr4I+dmkYeBuvZVM1YWhtejPVALurNbbLkOEj+yPhGbTG
nxBGt08KsSGKCM1blIY9MpkbsdUs0rSkPs33cYeRLJwGTzAsTSy0txkCETlKQBgK
0JGNQHIu8gvpjyMr1RI5xHGVjvbdz0LiWeQPJmoqBFyO53sliYgWGiZmeqjVUSc9
LeQ1h0kHl+vF0QXaQil9+SpjRTlFe3MXdq3gmvwgkYPelF48YaBst45yyJh57+z
Z3pAX7dJgje75Msb1MKn7q/OsPf4Ux/yfWTVFxnJEGFG046FOWkVb2LSBRhqxtCC
Fd4GCSqGSIB3DQEHAAdBg1ghkgBZQMEAEIEEN/jbIuyBiQPvx9QS9tgtISAghWw
/W8bWpUqIZAatmwlv5kmA9az3Z9YUJnqm2X8mh1MO+UrRCcq/uk04cXYQaF0iqS+
M6torBqIrSRUMFkcC7k9TEaDFIuUYpRfp00AFGT/+imSNuouqRb69TcXkAHqfU7S
p9atNXNLr7tSxVec1j/uuW8cwTToPi7U/kHFCdGQt+YwMoUhD4gVp6lxWtgeNUE+
RNr/vN/hPSwXyWR/Wck4Vlc9AjGlwds4m4R9MzGHaaFWjOSgbkhm8dN/e0s409ze
8YzvbRc3GKz669zduW91LGzjbaGGd+X3Oug9zf6JPkdwvQAv6rPfQK6zbOBtNs7Q
KYm2APsaHFjItbN6/pM1E5ypYb+q+W+jQQqrbZOFziw1xFWWU0vUe0GwADCjEkKN
68ImJdvWjBlFvdFrGQLFRogHBwcyxCTtF9ZJcG88ldMGot5S7vKfSWY815ZEotr0
ZqgmIA5tiajWyasZPpqgz4Cz0pP6NJpeuTlpHrDKH/YjMvtdzzpnaBvFPMQJGu7Z
2gG5BX36PMHNFWDUi+L9fUnXl2pjuWqYPOS4WatITNaRP6NIyR3qsbSNZ0uqS7Ry
bZs9xvpYBsFIupxr6b3a2o1aSx4I0rjLiJdJYDesIjV2b+eis/vMi5HKbY2feFch
tTPdcv2KxP1yxflB5xF/jVxaFXlsRr7ZW3tPrWuR/oGhSn5DM6Rugg0zN7RoMAuu
9QxQRWS8eyw5VFxThQ/5pWVos2xwF3WtKVfuOXbhhKlWwwcZpiW32UvwnLG6QdLp
2FdmgD/MJMkGHOrB2LyUx6fABSOrOBz7iEe2uwPDTKIyLNj8uH4P9+09IaYnNHbT
mOjGGF4eTRVwRe8QTj8aQA+ObyxriGHEDNIXTF+QFES9+roo2zWbbOF2PT+C/LIA
Rmhtc0gFnpcCQ0iZNNssJDBlZhuliEGq5Vbm/UXqS1lb/vWtBmqrwUoBsrgXvkvx
HevFH4VrRQE8aIDCKMFDTme6Ti9zZyJh7sviuBQETt0rIQ4Hd8tVPR4B9VSIKbER
mgOosxcNkGEDPipr8Z+hioT07g1++ZhUbPQSY6biWrQmRemE4nIXisAEXfX5oPtrN
X9y92vgfUEF9q8c6uiVlh7MMt/U8WyjuoM/pEQRd24sAln+Hxytq99aStV0DQgg8
eC7RmmjtGToJkdeOPPjwZEn2QVloYuJs4jD4Aqrt+KlaooFh59tAacHt3KL7LO/c
U/sUFENJ9ouHlfmJd84xc5w0D4g01B53Ly6YRjLlzlrdlfhkU2OJiG9s7Ki6yC/a
4B7rA5cULoxyKiSi1QTDbTqo7CO0daORPKP7ZQWMTTrRhjeF4qfNJKWkTulKXJt

rIlw6XRj3xix+kYBrDHkzZI8Jp27Z41bkpcXu9U3iOHP+HPD8T8HepC2n63eQop2
+EJ2A06pintq029gtfssP7Tl4kybuimSgyaVLEIwcwzdI44fYg/Oiiezr66DSi/F
QggHZW3pgIdudD/CS4Uf7MdZid3S19NSBh3iAdiajotqXz7SEMCCt3YfdrIDFX7b
XQxhbVD/26zPKilLSYbAs634xeU91PUeDFvYdeA6uMSG05Fn+0D2ldT8vZiE5H2T
ud0buFrNqN8mnvAo6PxIDHqobXkTjcbdfDnPM43xGfvNPO8WUvGOHwSEhlzz+pvh
BeQ7XxOo/U0aNSXdT88TZ9v9z4VYCLaW2ko+WAd9PrmKLKcdqxmt0WT7z1ii2RG7
hLOpjKI4FHWFGwtXcx8YnXr4FDr6m87DhiYURQbLSV4iUfBgECFFhVuz4quYIyZn
yDrMlVJJ15vmZmwO1JKFSjMKyUZTJRPZaqRqjEu1hmLfUTKygTpFHWORx8HTkiDE
wWG4c3Jyh5AMSjYmTNnVgr/fqH1N56k9LD9ydWquMKe0HW3X2bhMQ6M+x031lb/k
XUbf71D2W+u2BJMGDnhvU2a1L42QPQebGjrbsb/Dmoq9BtJr1ldrB224aCbaYCSkN
dsQCCSPLCB/TXJAGoDSznw5f0OdG/gsaFEoQ2SvCrnACoQwkpz8HHYezx1QnV4Bn
kv7Cq70vb3wndsctTZrdR39fpB/rWILMer7kfsClrto7WK3p2QRgEAgDya82SWtJ
FJpOzO/6hW3ECIvq7TZHElWCvf/5gG6YsaDi36dBGfwUMI+NkAVOCCcKCLmro6ET
Rw0Yb3sawxuBrS5hOG4jCXcuN3lEC8AVVARho17xHU5nt+pfFTV4jt/ujh6iWxx8
zmwiPKO3tCaNAwSVHy3UHN9D8kz+ygMqMSQLFtzMnW8cty2Xf9YF5SiBefQflgM
HbI0dvzXxGstYSOjrQehUVLaW6gLnPuyssSDISubCQuF89AILtRpH+rETIq8Ai6L
tlvldsbl2ikHbVWe0z9f+EsXks1E2hO7GyPiK3TgwzVeT+t3z5wA0/3917qigGZ/
R6v3e2RhaBu6DSBhUX97hvJgn0rIjdkNv2A380mrW9Xz2ZXJhYkj5Isp5ch5wy8p
rW1leL6trfkuqozm174uYA44/DRqnEqqU6QhIeIJEAUeXilsfBittZ24twIulKx7
8S6g2BjuoBvv6RiWnW1gUtch45H844gqTrwjAr4j+CarCc8mYmI1LjaM9uVUOgt1
4q5+2m2f294KOKgiY45Q7Hit+TwqO+inWlskDqZAb04zn0/aZbdrqomWh+f7Nufd
Kv1FWAoljZg+ekAFFytBreBJsw+zah4yAz4W28gldylw44f68xNzCRg4SpoEm8Rp
gbQXVKzi7mFcfYn0R1GgFFldLDDLlV9F0b4hXYAgY3KV0qu6hfyrq6zAw8CRAPYkP
3rhV082VlFOaxIUia/U06vuXOWFzKMKciH8XEDvdPZycExa5HTzr9D7Je89csh5Z
AuQFRoHoshr3cDpiq+ML01HpL+b00l+tCkWLJSBE0y3JV4udFnWmESoqU4WAGKhP
+AWSZdwjySJEZnZtRgovk+fquvxnL6FjPJL/ohdEAQPeXfxbvgxQoeedDFCst9q
05G1Ekiq3VH4NDcGARDceGFag4oJU6Naw0rKAW3dzZQjZxU0c8a+CdVLV+ZaXYUC
rbopg4GKcAnCo2RP3tIXNvgHvnHWhWhtiys7hzVNpt06jXk0d7qIF7hClxq5aShe
kweXjMHYZJLjB/NT4JZoIgeyQKJAZkSSqbqBgbK3Mtuw5aZQaChuMr0MYyXbZ5Yv
4EABKcGUjlnIcsx4goKlsCnNVUIakz4oHCaxdKfGA/SyKbs8cgS+zusjpd9ankYh
tH8VGA06s0td3CvDhHV0X8S5kyU01LkyNhkXDCE5TnTEKRF4b7vLpNj71FzLYPC7
vc1FHNSFhyPjD+MGQsqohf1HozSjUMlt/Au72XxP8LXQgqJiRP0UkZ39IjRmt4BK
+rXt6baHjmcQfowjAhIPsqDNLgFRGGK4FSJ1hRb11kOFz4VHJ8604Akms2Mk5fF
kTXLOkxOEqyb+JBvd4J/NmW6wv1EZ7iHw+3nRS7E6o1+wef15b/axmVeJgU/h6KP
OfJZ8vDjzNtrkHFTbix4Vj7bzQFLlfiG17bP++hN+8ioJDSxob0/DijdcTvdJnzR
XJRgBH4iEEJrOcleQ5HIq2kLmUoYz+U4YpBVfBOKUyQfheYl689HphhUg2NES9w/
6am0jNfHpdUrRuBCHtBLIJySdyexq9Gzy/M5/+j51v29YXCLZo/lu6JpPXv21wGy
uG/+T5wFfKv1cIBVfwgYJJM4Whht7I9S6IAqpp35b0hLntYoyGAqttOSEENpM5wJKW
DGLeB4vye2vviK67ZACxcnqUrDePFYRFKUMSj+U/zeB62y/DVmZBkr7XAXiGBKbp
M5YMTuLmsz6uB2S9Pp1fuiw03qV4myPH1NQMtHZVnn/Fcgo+3rpW1zx3JSX+aMdT
eEran9uQRAyFMHod6k1tghZwvvZwGaU+9Oi7hyL2o4nJY1G/cqWvSK1E48u8aftK
oPv6RmpJDvJbh/uriqGZKNI27t50/IGBBcwRGeMBgqYYkmG4ss6cvbIcBcnyP/D
w4EoGDTLL+YU3vOZKUp5l8TEHYvtDGubf1nMt0uTT1Zk6savmLVEHOYObjpHGAVO
Mn5PvfV2L+QYi2mpCxAmarScHVJSysWXJ66Lzps4J0hI2mfxalyK/N+qW8dNrvkJ
tyokrjJfn03FVyD4j2Ph962pMLP9m0FsNBVa02ntBYojDYyd5MqXNcUMVkvaxORk
UTuUsCwU7CwIkTDpHtDt+9u8Lj139jkejwEAovh70EVDkGac1DCi0PVs/jq9ferb
V1T9QGbP8U2wp6pwVsJAdo4nuH+sn7HUsDxGP0/Zwz65dhSyd7eHLNSfEdxBMFSq
GyQ/RG03Rxx+sgtAKLjaB1S4Ra7xNLAKdx00dlyciNXPfHubDDhaib7BQE3qG7WY

9JYC9NeBS6qtfn5PBS9xaf5xtHLbIBegz0NRmct2KkamMIQsAJYRvcJ98mMXrFwO
qpqtQ0KHePJk7CLjUB8oQooWUuD7LGpmeSCnjTUSXqqJiW40ZWX0IWJYGkCEOLuZ
KrCIkTYimOq6fQBf6e6aAzrF1Wpdk7/7GXhiJf/agQnRkvrCP3xAeYnDBxDMnWmD
EKey12hNSGbEx/GEvM3c0odMtd6Hmko8X1G9OXevZWd10CiEFkqeL6faFO0v+rZc
gHF18L09KUOIxIjyPis3lKTrFLBqJnfzyHDeIiIlCCfqAgW/2ng3EK5sDs4fnvYN
DmNJIE0oDiDodIQrznGwn5Qsj2sG/aUgp8cNNdsLWn7diGmSrdJFZWji9/rluO60
lnwrMHbPBzEpEuFzjGs8TbN5Ww2CUfuSFBkB+dn7dkoORVppiakygh/OzSiNYpl
KCNU7RkGV45I+hadL7RU811L5F4Qimo7WQXW6F8fFEakURm4PU2cREpR86dhe/Xt
XNp6pvLjvgZb9G2CgtgDMgsZqSRlDa71B6ktIvg1js0blZ4Tcn4APcdi5F2Tm6Uj
h7V01OozajrZ4VGJVYI6DsBRPfa5DY+14f/ITDyONn9VBmnOlIQhwC1G411csAnW
L4T0bi2glM13BdafBAR0H7RePm08oohRiV9gB3lm9OXy7t9tyMdmfJSKEsXALnc5/
aE+7QfadJ1uaKI8MvFbKWB6x5KD+XHjNQ0NHOewM3aloJUp2Ok6CiNp9yekVAB
w8cIhvODtQysXPMj/q+wnuieOzkYHt9I2TA+wc4Bq+p6ZFGBIZUBzmb21h8SRqUw
HXC6D2VSMCBFjIVpePhYB8TbgEkY60obahPfkiq4BN1SnJc9rGK3ueMOcXLWyp8j
5enxquno55PmmeSvyU9VS5vwcUiLoEggLfmc3l0/XVlVpyFUs1ly1KjhBh0YfSDF
R0wTA3fMRH8v9UVQ1VcoNBS+FzXPk8wRm4Nbx0zQ/d6BqDeL25dvQw8qy0+CIntR
cMWV+BG5PIFFmL4N9fqwliHyK6ccIhp9KpUuVrpTTmmE2DuuJji0001ZU52DzaTg
GvRuEjZz/TryEYploSpya4iaNzqnaaWd/g4STf5EXzh192QBf7WJoct/EaioK+8T
hIpyR5qXBX0RK/+TlIT2+oOPFdEXXOI5II+0YTdYa+y1uV9qKnN3apBXS+7GLodr
fjOABQTPxkg1p6d7CTJU5gJLR+xQjkOKMvuQJn1WzeN3pkEFKaC/9SwoL/olvs5+
uCPe5QWUXNuCPyd9us8/mNsXse69SNK/of5/Zqn8NawfmQVMO8JaPWPWarqJXdoY
2Mt/UhmLgfrZ6QidZEQi6OPcLgNbbYY35VHGgYsHj8c07GYTo3p591KC6xEotY92
9MyKOGM8fw3dfAbBPXA4TqyUm6kD1J2FylsMMkyfR5WnQDsR+/Vxq5k5bTlJ1ZRF
8FZHewv5AitHWP8KknJv9yHpygUWgjlPtFTPI9JfC4OI4kTybfGkS67iIB72oojf
dLLyzdJ/WMy9HS1T6EncV0clQTVlsCpxvNmN7Wxt4BkYd0v8eLPm7d7saiwl38D5
TtHy3EgkOABSPPUoihuls1gJKoRq7hWT3CYf5UBCSa3Ocd7Qo2yKJNgDrRosp45j
X6u//xxA/LDXgrq+th28PN7i+E9ZkWhT16wdUbtFQBEOmpm5ZB3hq88mDk15v9vb
OnQnwGf6h3UWx/AzmPuRPu2C/7mEtB7/tUj9nqwCgjXIJ8oYhv2uD6IjoAZgRbwm
T7KoMb9T780h/0LealOBpZ2a9LZgNAIcDWWhb8fGcS537GIzIS6eZG31J2Pdb+ip
isCzrnRZmWJqR9MPHqUq0lhtLEuxd0RnuqQE+VnYydNvDu0p3L5nfINK9vtGWybkC
XRFbJS23dc0vS6ug29jGzLzjODz/S6TTvo0qgl2heFVfDYzD/z1pw2dPQA1k+RhO
dAG0tDQCiyVr719e64j4ZbFjMNF7QA+YJfMaQ1H1XEGQvF9oLA34dN9hiNAh2Ls
9ehAOIo7gs192SDDowDHSmJJr27A/BdGGc4vC+t8Bc7hjFza2ixJ9VkiH1pa8ZU9
aNNbLcnfb518/7DXgSpiVFncgsLaCZ3iORFXE/IsNX9+R0An0+y+r2mpdtDWglw
69g+EMg4dJw8u7pTTW4J47TCAECjF3WVybl8YpvVmgVsrTIL/jD1NWq66jTh2yC7
Kcc7IF1neMYTPw033hDTKDcY27lnz/BhdumwynboWzKTjyNuim6e/OdCKOJHT8YJ
8icUmzboi8iYjAwhSqu6t8OZBYIT7oItqzfkQMKKLWwuguJsRa3P6OY9Gg7FUZno
PXjOCpNyGzY0hg5VVk6FV+thB11MYmlnG16D50UbrH4tgnzkUwpUCMrXLdWr7dfp
19u77ICFSiWnIUTtah+s9TUULnBAL1TWyEN6dcqdtT2+HYzDN+FT9+HJsUabDIVP
9421qkTt5V1CWImXEPdeq4PqfE7LWtEA666xhpgzdnmmE35QHI/por/HS47TlxTV
38m+Laew31eEWGaiORbPI8X1NZqlfwjv39bpJH9nqMdaeY/kbgFCAsJyuW1nfJ4W
uiTUYsk0Cs9u70BdYYfo0+zdUgem+XM0epL9zh9gsKiJ4gfdbv8x0rmcXhIhaA/V
bRGj9MYxyBbCORCNCmt1oeX/GndLxj9azdHKugZdLzGTA0Dx84xRd9rDWOSxGv1/
bNVXqDqCaW7BcSi08pAnWlvwQ+m/p2Wxkzi7luxJhhHX7M8/k6mdJmmrB6SRf6S2
4oc7ojwI6vXTexWry42luQcrQTOMIFutqna5NYRy1ICuC0vm3WdNuRlFN7LkpaFq
evbT4zaksQOuDFoXIGIQ8kJ6HTEOA+v33uV7BZfqlo1yIetX1JnToGheZBMc3skU
pCQjWDeZA6u42Nz+ewytKgYRwr2trDE0bX3xMfH0+/o=

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] 4213 bytes
(unwraps to)
message/rfc822 566 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
```

MIIMIwYJKoZIHvcNAQcCoIIMFDCDBACAQExDTALBgIghkgBZQMEAgEwggJMBgkq
hkiG9w0BBwGgggI9BIICOU1JTUUTVmVyc2l1vbjogMS4wDQpDb250ZW50LVR5cGU6
IG1lc3NhZ2UvcmlzODIyOyBmb3J3YXJkZWQ9Im5vIg0KDQpNSU1FLVZlcnNpb246
IDEuMApDb250ZW50LVR5cGU6IHRleHQvcGxhaW47IGNoYXJzZXQ9InV0Zi04IgpD
b250ZW50LVRyYW5zZmVyLUVuY29kaW5nOiA3Yml0C1N1YmplY3Q6IHNTaW11LW9u
ZS1wYXJ0LXdyYXBwZWQKTWVzc2FnZS1JRDogPHNTaW11LW9uZS1wYXJ0LXdyYXBw
ZWRAbGhwLmV4YW1wbGU+CkZyb206IEFsaWNlIDxhbGljZUBzbWltZS5leGFtcGxl
PgpUbzogQm9iIDxib2JAc21pbWUuZXhhbXBsZT4KRGF0ZTogU2F0LCAyMCBGZWIG
MjAyMSAxMDowNDowMiAtMDUwMAoKVHpcyBpcyB0aGUgc21pbWUtb25lLXBhcnQt
d3JhCHBlZCBtZXNzYWdlLGoKVHpcyBpcyBhIHNTZS5lZC1vbmx5IFMvTU1NRSBt
ZXNzYWdlIHZpYSBQS0NTIzcgcz2l1bmVkrGF0YS4gIFRoZQpwYXlsb2FkIGlzIGEg
dGV4dC9wbGFpb2IbZXNzYWdlLiBjdB1c2VzIHROZSBXcmFwcGVkIE1lc3NhZ2Ug
aGVhZGVyYnByb3RlY3Rpb24gc2NoZW11LGoKLS0gCkFsaWNlCmFsaWNlQhNTaW11
LmV4YW1wbGUoIiIHpcjCA88wggK3oAMCAQICEw8tJb0ROZDkzJUh6HuPTQgIRQw
DQYJKoZIHvcNAQENBQAwVTENMAsgA1UECMESUVURjERMA8GA1UECXMITEFNUFMg
V0cxMTAvBqNVBAMTKFNhbXBsZSBMQU1QUyBSU0EqQ2VydGlmawNhdGlvbiBBdXR0

b3JpdHkwIBcNMTkxMTIwMDYlNDE4WhgPMjA1MjA5MjcwNjU0MThaMDsxDTALBgNV
BAoTBE1FVEYxETAPBgNVBAsTCExBTVBTIFdHMRcwFQYDVQQDEw5BbGljZSBMb3Zl
bGFjZTCCASiWdQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAJqVKfqLwaLjj+gB
UCfkacKTg8cc2OtJ9ZSed6U3jUoiZVpMLcP3MUKtLeLg9rlmAfIDlB/wlbdmadXP
mrszyidmbuZmOpB5voVQfiLYYy3iOx7YQqzXrl6udP07k0sV+UdSNRFxrfKeoQEF
XgOaGdmnx4OG/e3plfIKM0dPzZLoOAJF5m500xzXPL74zFCWp2f1ZkuE4A6l4lko
aZXCN5XL7wWTLMLenF9Byb5ksKqUuqEHAMdlnmoNMgjY9VfVfcrv9w43GG8FtpSX
+TWzB2zNS2OF+XIVnzRG5DeoULq8v88Z5bLpIJ/nx26r8A4SSwIBaVv4wPxAf1iP
sIVKarUCAwEAAaOBrzCBrdAMBgNVHRMBaf8EAJAAMBcGA1UdIAQQMA4wDAYKYIZI
AWUDAgEwATAeBgNVHREEFzAVgRNhbGljZUBzbWltZS5leGFtcGx1MBMGA1UdJQQM
MAoGCCsGAQUFBwMEMA4GA1UdDwEB/wQEAwIFIDAdBgNVHQ4EFgQUo1NB1UQ8gCkV
fAEj8OeOr83zdw8wHwYDVR0jBBgwFoAUKTCOfAcXDKfXCSHlNhpHGh29FkwDQYJ
KoZIhvcNAQEBBQADggEBAIFJeKCsTKcFqQMPTryujRGzJdYA+R9eBAuDLSatbtK
t14FzkgRyOg31/+Cw7H8e30iLrPIfLWN1qjHrjgOyIs5AQ/hgxLvLir3hEUUV2Z3M
RsMtjH2x9SG91PEM046gfPnc9gMGHjMTg1qvaKcLQP5UzpEYPLror2X4P5uXxaP0
LIZRzWmkw1RF7FOD7Pfb5v94M5274XYxW2W4uKGd7QGnUZROSvSYkGiWdp1JhqXw
fDz8A0enITGXnoEkaFvviCqh64PlhIeMorj36pgL19oWZD6YrzSWHuz1F00juyu
OfQsqm6hvrDTqNpHNZ015FOURza1SkCvi9GFmNUPoVgwggPPMIICt6ADAgECAhM3
QQV57XV/QqmiXDr0+GrOmgnXMA0GCSqGSib3DQEBDQUAMFUxDALBgNVBAoTBE1F
VEYxETAPBgNVBAsTCExBTVBTIFdHMTewLwYDVQQDEyhTYW1wbGUgTEFNUFMgU1NB
IENlcnRpb2ZlYXRpb24gQXV0aG9yaXR5MCAXDTE5MTEyMDA2NTQxOFoYDzIwNTIw
OTI3MDYlNDE4WjA7MQ0wCwYDVQQKEwRJRVRGMREwDwYDVQQLEWhMQU1UyYXRzEX
MBUGA1UEAxMOQWxpY2UgTG92ZWxhY2UwggEiMA0GCSqGSib3DQEBAQUAA4IBDwAw
ggEKAoIBAQC09InoWDgWpk2af0+StijSNOR8K/hN8D+1078oullsk4ASvSwjsCNO
7sHUa4xQU15JO6VqY18LANwORjrc9BaX4MguzsbFXBe6uFhlmVpXmFxSpUBYQ+95
0MFz/evPgP96wV+z4TtAwW2Z34rTiz4DxMI07XYNFUEOls/gkUP2Gxzyms02kaYW
Tut3SryCqeHEFbZfKb4urMk4xrIJC3CzWruS2Q0FHbBlfkgKN5wXVgkWFfiOucfC
n+IQsaqpol3df9jSkbtAV5w3vzfog8919MxKI9H614KuElnAtJ7BtZcs17dUy9u9
COgEyKriVokFQgqQ7XNDU+r3SeOWwks7AgMBAAAgjga8wgawwDAYDVR0TAQH/BAIw
ADAXBgNVHSAEEDAOMAAGCmCGSAFlAwIBMAEwHgYDVR0RBBCwFYETYWxpY2VAc21p
bWUuZUhxbXBsZTATBgNVHSUEDDAKBggrBgEFBQcDBDAOBgNVHQ8BAf8EBAMCBsAw
HQYDVR0OBByEFv2zLlTHQYSHJeuKWqQENMgZmZzMB8GA1UdIwQYMBaAFJEWjnwH
Fwyn8QkoZTYaZxxodvRZMA0GCSqGSib3DQEBDQUAA4IBAQBziaI2p86poGkjd/4K
kkOHG25nY/0eNARD6/oF0/sYonX2doizcGMk53riugAocCn5zbzhW/JVdYn30Uxf
yrZlRAZef7GHqgB/Nyjoad3pdpVYeDh4ciNKjbs+aEoTWgAkoqENTlsRxlcvb7HV
X524bkZa1oPTUN1m6QpivtqDIdqGJdGf8L1zLfxBuo2zL3HR+M9CDr4Opq2JCKzP
0Qhp7poIccGE6I9Tsg+RrOA9iCQsPn1+Tg8YedjGzUWF07rNmT0TzPCVzUAuBlr+
JJtzoKypyQ3eoZ6EPazXqMyHAVcsm0GI364IOA0b8PSrJNtjh+AqJ5QfH+0e7NSz
NnEmMYICADCCafwCAQEwbDBVMQ0wCwYDVQQKEwRJRVRGMREwDwYDVQQLEWhMQU1U
yYXRzEXMC8GA1UEAxMoU2FtcGx1IEExBTBTVBTIFJTQSBDZXJ0aWZpY2F0aW9uIEF1
dGhvcml0eQITN0EFee11f0Kpolw69PhqzpqplzALBg1ghkgBZQMEAgGgATAYBgkq
hkiG9w0BCQMxCwYJKoZIhvcNAQcBMBwGCSqGSib3DQEJBTEPFw0yMTAyMjAxNTA0
MDJAMC8GCSqGSib3DQEJBDEiBCct+Ik56mZTd2mpSgOXM38dS7jm5a1U2FDX9/58
cga1szANBgkqhkiG9w0BAQEFAASCAQcXKLkx51i140IOcH2tcWqcsQilPLgQ30ck
qhJL2X9/Cl22ibOGNwL8w3qSEBeGla+WtHw3bSqJx1ciRYcLs16ms23no5QoZ0pU
fRLmQuTEgObCf+syiTGNwLj8e+2aRVP1L9yEibin6+hFyp4s393zYhdMOPAP2ruI
lg+BxoWXUjXso+81PgqLawA+9KMI6tQZMnwI9LpGJmZfoSXdHWqWtjdotzZpqskm
Ihr8DBKtUetqgZ2zqD03zo3W2L6EmNM05BJUmqwAt/cN+X9kws5dAqtHDQhPNTa1

WUX0oTTkMzn1RAL0xfowEStSnfDOOzIqg+L7LgiMw9jhIgP4/uB2

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 4451 bytes
message/rfc822 596 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="20c";
  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
```

```
--20c
MIME-Version: 1.0
Content-Type: message/rfc822; forwarded="no"
```

```
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
```

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

--20c

Content-Transfer-Encoding: base64

Content-Type: application/pkcs7-signature; name="smime.p7s"

MIIJ4AYJKoZIhvcNAQcCoIIJ0TCCc0CAQExDTALBg1ghkgBZQMEAgEwCwYJKoZI
hvcNAQcBoIIHjpJCCA88wggK3oAMCAQICEw8tJb0ROZdKzkJU6HuPTQGirQwDQYJ
KoZIhvcNAQENBQAwVTENMA8GA1UEChMESUVURjERMA8GA1UECzMITEFNUFMgV0cx
MTAvBgNVBAMTKFNhbXBsZSBMQU1QUyBSU0EgQ2VydGlmaWNhdGlvb2IwBDBxRob3Jp
dHkwIBcNMTEwMDY1NDE4WhgPMjA1MjA5MjcwNjU0MThaMDsxDTALBgNVBAOT
BElFVEYxETAPBgNVBAsTCExBTvBTIFdHMRcwFQYDVQQDEw5BbG1jZSBMb3Z1bGFj
ZTCCASlWdQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAAJqVKfqlWaLjj+gBUCfk
acKTg8cc20tJ9ZSed6U3jUoiZVpMLcP3MUKtLeLg9r1mAfID1B/wlbdmadXPmrsz
yidmbuZmOpB5voVQfiLYYy3iOx7YOqzXrl6udP07k0sV+UdSNRFxrfKeoQEFXgOa
Gdmnx4OG/e3p1fIKM0dPzZLoOAJF5m500xzXPL74zFCWp2f1ZkuE4A6141koaZXC
N5XL7wWTLMLenF9Byb5ksKqUuqEHAMdlnmoNMgjY9VfVfcrv9w43GG8FtpSX+TWz
B2zNS2OF+XIVnzRG5DeoULq8v88Z5bLpIJ/nx26r8A4SSwIBaVv4wPxAfliPsIVK
arUCAwEAAaOBrzCBrdAMBgNVHRMBAF8EAjAAMBcGA1UdIAQQMA4wDAYKYIZIAWUD
AgEwATAeBgNVHREEFzAVgRNhbG1jZUBzbWltZS5leGFtcGxlMBMGA1UdJQQMMAoG
CCsGAQUFBwMEMA4GA1UdDwEB/wQEAwIFIDAdBgNVHQ4EFgQUo1NB1UQ8gCkVfAEj
8OeOr83zdW8wHwYDVR0jBBGwFoAUkTCOfAcXDKfxcSh1NhpHGh29FkwDQYJKoZI
hvcNAQENBQADggEBAIFJeKCsTKcFqQMPTryujRGzJdYA+R9eBAuDLsatbtKt14F
zkgrYog31/+Cw7H8e30iLrPIfLWN1qjHrjgOyIs5AQ/hgxLvLir3hEUV2Z3MRsMt
jH2x9SG91PEM046gfPnc9gMGHjMTg1qvaKcLQP5UzpEYPLror2X4P5uXxaP0LIZR
zWmkw1RF7FOD7Pfb5v94M5274XYxW2W4uKGd7QGnUZROSvSYkGiWdp1JhqXwfDz8
A0enITGXnoEkAFvvjiCqh64PlhIeMorj36pgL19oWZD6YrzSWHUz1F00juyuOfQs
qm6hvrDTqNpHNZ015fOURza1SkCvi9GFmNUPoVgwgGPPMIICt6ADAgECAhM3QQV5
7XV/QqmiXDr0+GrOmqnXMA0GCSqGSIb3DQEBAQUAMFUDALBgNVBAOTBElFVEYx
ETAPBgNVBAsTCExBTvBTIFdHMTewLwYDVQQDEyhTYW1wbGUgTEFNUFMgU1NBIE
N1cnRpZmljYXRpb24gQXV0aG9yaXR5MCAXDTE5MTEyMDA2NTQxOFoYDzIwNTIwOTI3
MDY1NDE4WjA7MQ0wCwYDVQQKEWRJRVRGMREwDwYDVQQLEWhMQU1QUyBXRzEXMBUG
A1UEAxMOQWxpY2UgTG92ZWxhY2UwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEK
AoIBAQC09InoWDgWPK2af0+StijSNOR8K/hN8D+1078oullsk4ASvSwjsCNo7sHU
a4xQU15JO6VqY18LANWORjrc9BaX4MguzsbFXBe6uFhlMvPxMfxSpUByQ+950MFz
/evPgP96wV+z4TtAwW2Z34rTiz4DxMI07XYNFUEOls/gkUP2GxzymsO2kaYWTut3
SryCqeHEfBzFk4urMk4xrIJC3CzWruS2Q0FHbBlfkgKN5wXVgkWFfiOucfCn+IQ
saqppo1d3f9jSkbtAV5w3vzfog8919MxKI9H614KuElnAtJ7BtZcs17dUy9u9COgE
ykRiVokFQgqQ7XNDU+r3SeOWwks7AgMBAAAgjga8wgawwDAYDVR0TAQH/BAIwADAX
BgNVHSAEEDAOMAAGCmCGSAFlAwIBMAEwHgYDVR0RBBCwFYETyWxpY2VAc21pbWUu
ZXhhbXBsZTATBgNVHSUEDDAKBggrBgEFBQcDBDAOBgNVHQ8BAf8EBAMCBsAwHQYD
VR0OBBYEFv2zLlthQYSHJeuKWqQENMgZmZzMB8GA1UdIwQYMBAAFJEwjnwHFWyn
8QkoZTYaZxxodvRZMA0GCSqGSIb3DQEBAQUAA4IBAQBziaI2p86poGkjD/4KkkOH
G25nY/0eNARD6/oF0/sYonX2doizcGMk53riugAocCn5zbzhW/JVdYn30UxfyrZl
RAzEf7GHqgB/NyjOad3pdpVYeDh4ciNKjbs+aEoTWgAkoqENT1sRxlcvb7HVX524
bKZa1oPTUNlm6QpivtqDIdqGJdGf8L1zLfXBuo2zL3HR+M9CDr4Opq2JCKzP0Qhp
7poIccGE6I9Tsg+RrOA9iCQsPnl+Tg8YedjGzUWF07rNmT0TzPCVzUAUblR+JJtz

OKypyQ3eoZ6EPazXqMyHAVcsm0GI364IOA0b8PSrJNtjh+AqJ5QfH+0e7NSzNnEm
 MYICADCCAfwCAQEwbDBVMQ0wCwYDVQQKEwRJRVRGMREwDwYDVQQLEwhMQU1QUyBX
 RzExMC8GA1UEAxMoU2FtcGx1IEExBTBTIFJTQSBDZXJ0aWZpY2F0aW9uIEF1dGhv
 cml0eQITN0EFee11f0Kpolw69PhqzpqplzALBglghkgBZQMEAgGgaTAYBgkqhkiG
 9w0BCQMxCwYJKoZIhvcNAQcBMBwGCSqGSIb3DQEJBTEPFw0yMTAyMjAxNTA1MDJa
 MC8GCSqGSIb3DQEJBDEiBCCcDIxr7wd3VCCz1VBG9nySvUJ/Fhzo26f78El/UUbj
 jTANBgkqhkiG9w0BAQEFAASCAQBUMGL40IZQmt3Nad/ymEUOLu3Dgfd/nYKuj6P
 fjKYJfb9UhtufZK9/WyVtytLsFJMYHZgUSWU3VbHk1L/c00469Rbqo6Cq1LRJPK
 uN2Eul2UCa+3ovMIQ8g0NBflXrdfR00VRqvfo91hLFkTxLfCDUG8ziRWOLWucgZg
 zkVXqEzvFyOtsSbr3GAY8l7wWg1l+PTFch04XF+rg7cNysKqGLtjxP9lN3PcURYv
 TmooTPY46kheab7ZAZKqQI6go7somKmMqD7UsctMLSVZo+EX5/N9vq5zmv7bfpoe
 Rgd+NZNQD+VYDIOU1FI5ZjyHjHrMcfpywHNBtBGLYhv3q4

--20c--

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] 4185 bytes
(unwraps to)
text/plain 239 bytes
```

Its contents are:

```
Content-Transfer-Encoding: base64
Content-Type: application/pkcs7-mime; name="smime.p7m";
smime-type="signed-data"
Subject: smime-one-part-injected
Message-ID: <smime-one-part-injected@lhp.example>
From: Alice <alice@smime.example>
To: Bob <bob@smime.example>
Date: Sat, 20 Feb 2021 10:06:02 -0500
```

MIIMDgYJKoZIhvcNAQcCoIIL/zCCC/sCAQExDTALBglghkgBZQMEAgEwggI3BgkqhkiG9w0BBwGgggIoBIICJE1JTUUtVmVyc2lvcjogMS4wDQpDb250ZW50LVRyYW5zZmVyLUVuY29kaW5nOiA3Ym10DQpTdWJqZWN0OiBzbWltZS1vbmUtcGFydC1pbmptY3RlZA0KTWVzc2FnZS1JRDogPHNtaW1lLW9uZS1wYXJ0LWluamVjdGVkQGxocC5leGFtcGxlPg0KRnJvbTogQWxpY2UgPGFsaWN1QHNTaW1lLmV4YW1wbGU+DQpUbz0gQm9iIDxib2JAc2lpbWUuZShhbXBsZT4NCkRhduGU6IFNhdcWgMjAgRmViIDlwMjEgMTA6MDY6MDIgLTA1MDANckNvb3RlbnQtVHlwZTogdGV4dC9wbGFpbjsgY2hhcnNldD0idXRmLTgiOyBwcm90ZWNOZWQtZGVhZGVyc20idjEiDQoNC1RoXMGaXMGdGhlIHNTaW1lLW9uZS1wYXJ0LWluamVjdGVkIGl1c3NhZ2UuDQoNC1RoXMGaXMGYSBz

aWduZWQtb25seSBTL01JTUUgbWVzc2FnZSB2aWEgUETDUyM3IHNPz251ZERhdGEu
ICBUaGUNCnBheWxvYWQgaXMgYSB0ZXh0L3BsYWluIG1lc3NhZ2UuIE10IHVzZXMG
dGhlIEluamVjdGvkIEhlYWRLcnMgaGVhZGVyDQpwcm90ZWN0aW9uIHNaGvTzS4N
Cg0KLS0gDQpBbGljZQ0KYWxpY2VAc2lpbWUuZXhhbXBsZQ0KoIIHpjCCA88wggK3
oAMCAQICEw8tJb0ROZdKzkJU6HuPTQGirQwDQYJKoZIhvcNAQENBQAwVTENMA5G
A1UEChMESUVURjERMA8GA1UECXMITEFNUFMgV0cxMTAvBgNVBAMTKFNhbXBsZSBM
QU1QUyBSU0EgQ2VydGhmaWNhdGlvbiBBdXR0b3JpdHkwIBcNMTkxMTIwMDYlNDE4
WhgPMjA1MjA5MjcWU0MTAhaMDsxDTALBgNVBAoTBELFVEYxETAPBgNVBA5TCExB
TVBTIFdHMRcwFQYDVoQDEw5BbGljZSBMbz3ZlbGFjZTCCASlWdQYJKoZIhvcNAQEB
BQADggEPADCCAQoCggEBAJqVKfqlWALjj+gBUCfkacKTg8cc2OtJ9ZSed6U3jUoi
ZVpMLcP3MUKtLeLg9rlmAfIDlB/wlbdmadXPmrszyidmbuZmOpB5voVQfiLYYy3i
Ox7Y0qzXrl6udP07k0sV+UdSNRFxrfKeoQEFXgOaGdmnx4OG/e3plfIKM0dPzzLo
OAJF5m500xzXPL74zFCWp2f1ZkuE4A6141koaZXCN5XL7wWTLMLenF9Byb5ksKqU
uqEHAMdlnmoNMgjY9VFVfcrv9w43G8FtpSX+TWzB2zNS2OF+XIVnzRG5DeoULq8
v88Z5bLpIJ/nx26r8A4SSwIBaVv4wPxAf1iPsIVKarUCAwEAAaOBrzCBrdAMBgNV
HRMBaf8EAJAAMBcGA1UdIAQQMA4wDAYKYZIAWUDAgEwATAeBgNVHREEFZAVgRNh
bGljZUBzbWltZS5leGftcGx1MBMGA1UdJQQMMAoGCCsGAQUFBwMEMA4GA1UdDwEB
/wQEAwIFIDAdBgNVHQ4EFgQUo1NB1UQ8gCkVfAEj8OeOr83zdw8wHwYDVR0jBBGw
FoAUKTCOfAcXDKfxCSH1NhpHGH29FkwDQYJKoZIhvcNAQENBQADggEBAIFJeKCC
sTKcFqQMpTryuJRgzJdYA+R9eBAuDLsatbtKt14FzkRyOg31/+Cw7H8e30iLrPI
FlWN1qjHrjgOyIs5AQ/hgxLvLir3hEUV2Z3MRsMtjH2x9SG91PEM046gfPnc9gMG
HjMTglqvaKcLQP5UzpEYPLror2X4P5uXxaP0LIZRzWmkw1RF7FOD7Pfb5v94M527
4XYxW2W4uKgd7QGnUZROSvSYkGiWdp1JhqXwfDz8A0enITGXnoEkaFvviCqh64P
1hIeMorj36pgL19oWZD6YrzSWHUz1F00juyuOfQsqm6hvrDTqNpHNZ015FOURza1
SkCvi9GFmNUPoVgwgGPPMIICt6ADAgECAhM3QQV57XV/QqmiXDr0+GrOmgnXMAOG
CSqGS1b3DQEBDQUAMFUxDALBgNVBAoTBELFVEYxETAPBgNVBA5TCExBTVBTIFdH
MTEwLWYDVQDEYhTYW1wbGUgTEFNUFMgU1NB1EN1cnRpZmljYXRpb24gQXV0aG9y
aXR5MCAXDTE5MTEyMDA2NTQxOFoYDzIwNTIwOTI3MDYlNDE4WjA7MQ0wCwYDVQK
EwRJRVRGMREwDwYDVQLEwhMQU1QUyBXRzEXMBUGA1UEAxMOQWxpY2UgTG92ZWxh
Y2UwggEiMA0GCSqGS1b3DQEBQAUA4IBDwAwggEKAoIBAQC09InoWDgWPK2af0+S
tijSNOR8K/hN8D+1078oullsk4ASvSwjsCNo7sHUA4xQU15JO6VqY18LANwORjrc
9BaX4MguzsbFXBe6uFh1mVpXmFxSpUBYQ+950MFz/evPgP96wV+z4TtAwW2Z34rT
iz4DxMI07XYNFUE0ls/gkUP2Gxzyms02kaYWTut3SryCqeHEFbZFkB4urMk4xrIJ
C3CzWrus2Q0FHbBlfkgKN5wXVgkWFfiOucfCn+IQsaqpol3d3f9jSkbtAV5w3vzfo
g8919MxKI9H614KuElnAtJ7BtZcs17dUy9u9COgEykRiVokFQgqQ7XNDU+r3SeOW
wks7AgMBAAGjga8wgawwDAYDVR0TAQH/BAIwADAXBgNVHSAEEDAOMAAGCmCGSAF1
AwIBMAEwHgYDVR0RBBCwFYETYWxpY2VAc2lpbWUuZXhhbXBsZTATBgNVHSAUEDDAK
BggrBgEFBQcDBDAOBgNVHQ8BAf8EBAMCBsAwHQYDVR0OBBYEFv2zLlTHQYSHJeu
KWqQENMgZmZzMB8GA1UdIwQYMBaAFJEWjnwHFWyn8QkoZTYaZxxodvRZMA0GCSqG
S1b3DQEBDQUAA4IBAQBziaI2p86poGkjd/4KkkOHG25nY/0eNARD6/oF0/sYonX2
doizcGmk53riugAocCn5zbzhW/JVdYn30UxfyrZlRAzEf7GHqgB/NyjOad3pdpVY
eDh4ciNKjbs+aEoTWgAkoqENTlsRxlcvb7HVX524bKZaloPTUNlm6QpivtqDIdqG
JdGf8L1zLFXBuo2zL3HR+M9CDr4Opq2JCKzP0Qhp7poIccGE6I9Tsg+RrOA9iCQs
Pn1+Tg8YedjGzUWF07rNmT0TzPCVzUAuBlr+JJtz0KypyQ3eoZ6EPazXqMyHAVcs
m0GI364IOA0b8PSrJNtjh+AqJ5QfH+0e7NSzNnEmMYICADCCAfwCAQEwbdBVMQ0w
CwYDVQKKEwRJRVRGMREwDwYDVQLEwhMQU1QUyBXRzEXMC8GA1UEAxMoU2FtcGx1
IEExBTBTIFJTSBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eQITN0EFee1lf0Kpolw6
9PhqzppplzALBg1ghkgBZQMEAgGaTAYBgkqhkiG9w0BCQMxCwYJKoZIhvcNAQcB

```
MBwGCSqGSib3DQEJBTEPFw0yMTAyMjAxNTA2MDJhMC8GCSqGSib3DQEJBDEiBCA7
4grfze+Y7DQEGFAYHyvRpNkuuZFR0V+RvSTvu4FGDANBgkqhkiG9w0BAQEFAASC
AQB1KYVvQNZpe3EKeM0XhJr1JNxneVmZWFCe15YFeRsO8FeIwJkV65YtFJKjOVVy
qYuZBGz4MsKaddXxAOXI/Q7cJ+70d9iOclmL3PD2/U6DowwhNfJoNSK7miYfMASV
42TMJWt0TlORJnvBitjkTuZDus1tp3xwxbrZTa4pyGaXEhBW/Fc4z6L+z8hpQv/
+6dw3+ORgfc67VTHVnsVVfb0UPrWvdxFdL5xYdqXxlhDsLMems2ttHHzvJc003Kq
As0xMHEmMpfdL5M69MAjvroOUv0SXETfQaxca7IKd+9xUNNRretZ9xz2kn2uD+k7
unTEyVGeHrWmQMw/8MdvEac/
```

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 4417 bytes
  text/plain 258 bytes
  application/pkcs7-signature [smime.p7s] 3429 bytes
```

Its contents are:

```
MIME-Version: 1.0
Content-Type: multipart/signed;
  protocol="application/pkcs7-signature"; boundary="12b";
  micalg="sha-256"
Subject: smime-multipart-injected
Message-ID: <smime-multipart-injected@lhp.example>
From: Alice <alice@smime.example>
To: Bob <bob@smime.example>
Date: Sat, 20 Feb 2021 10:07:02 -0500
```

--12b

```
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit
Subject: smime-multipart-injected
Message-ID: <smime-multipart-injected@lhp.example>
From: Alice <alice@smime.example>
To: Bob <bob@smime.example>
Date: Sat, 20 Feb 2021 10:07:02 -0500
Content-Type: text/plain; charset="utf-8"; protected-headers="v1"
```

This is the smime-multipart-injected 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

Injected Headers header protection scheme.

--

Alice
alice@smime.example

--12b

Content-Transfer-Encoding: base64

Content-Type: application/pkcs7-signature; name="smime.p7s"

MIIJ4AYJKoZIhvcNAQcCoIIJ0TCCCC0CAQExDTALBglghkgBZQMEAgEwCwYJKoZI
hvcNAQcBoIIHpjCCA88wggK3oAMCAQICEw8tJb0ROZdKzkJU6HuPTQGirQwDQYJ
KoZIhvcNAQENBQAwVTENMA8GA1UEChMESUVURjERMA8GA1UECzMITEFNUFNgV0cx
MTAvBgNVBAMTKFNhbXBsZSBMQU1QUyBSU0EgQ2VydG1maWNhdGlvbiBBdXRob3Jp
dHkwIBcNMtKxMTIwMDY1NDE4WhgPMjA1MjA5MjcwNjU0MThaMDsxDTALBglghNVAoT
BELFVEYxETAPBgNVBAsTCEExBTBVTIFdHMRcwFQYDVQQDEw5BbG1jZSBMb3ZlbGFj
ZTCCASIdQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAJqVKfqLwaLjj+gBUCfk
acKTg8cc20tJ9ZSed6U3jUoiZVpMLcP3MUKtLeLg9rlmAfIDlB/wlbdmadXPmrsz
yidmbuZmOpB5voVQfiLYy3iOx7YOqzXrl6udP07k0sV+UdSNRFxrfKeoQEFXgOa
Gdmnx40G/e3p1fIKM0dPzZLoOAJF5m500xzXPL74zFCWp2f1ZkuE4A6141koaZXC
N5XL7wWTLMLenF9Byb5ksKqUuqEHAMdlnmNMgjY9VfVfcrv9w43GG8FtpSX+TWz
B2zNS2OF+XIVnzRG5DeoULq8v88Z5bLpIJ/nx26r8A4SSwIBaVv4wPxAf1iPsIVK
arUCAwEAAaOBrzCBrdAMBgNVHRMBAf8EAjAAMBcGA1UdIAQQMA4wDAYKYIZIAWUD
AgEwATAeBgNVHREEFzAVgRNhbG1jZUBzbWltZS5leGftcGx1MBMGA1UdJQQMMAoG
CCsGAQUFBwMEMA4GA1UdDwEB/wQEAwIFIDAdBgNVHQ4EFgQUo1NB1UQ8gCkVfAEj
80eOr83zdW8wHwYDVROjBBGwFfoAUKTCOfAcXDKfxcSh1NhpHGh29FkwDQYJKoZI
hvcNAQENBQADggEBAIFJeKccsTKcFqQMPTryuJRgzJdYA+R9eBAuDLsatbtKt14F
zkgRyOg31/+Cw7H8e30iLrPIfLWN1qjHrjgOyIs5AQ/hgxLvLir3hEUV2Z3MRsMt
jH2x9SG91PEM046gfPnc9gMGHjMTg1qvaKcLQP5UzpEYPLror2X4P5uXxaP0LIZR
zWmkw1RF7FOD7Pfb5v94M5274XYxW2W4uKgd7QGnUZROSvSYkGiWDp1JhqXwfdZ8
A0enITGXnoEkaFvvjiCqh64PlhIeMorj36pgL19oWZD6YrzSWHuz1F00juyuOfQs
qm6hvrDTqNpHNZ015fOURza1SkCvi9GFmNUPoVgwgGPPMIICt6ADAgECAhM3QQV5
7XV/QqmiXDr0+GrOmgnXMA0GCSqGSIb3DQEBAQUAMFUDTALBglghNVAoTBELFVEYx
ETAPBgNVBAsTCEExBTBVTIFdHMTewLwYDVQQDEyhTYW1wbGUgTEFNUFNgU1NBIEEN1
cnRpZmljYXRpb24gQXV0aG9yaXR5MCAXDTE5MTEyMDA2NTQxOFoYDzIwNTIwOTI3
MDY1NDE4WjA7MQ0wCwYDVQQKEwRJRVRGRmREwDwYDVQQLEWhMQU1QUyBXRzEXMBUG
A1UEAxMQWxpY2UgTG92ZWxhY2UwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEK
AoIBAQC09InoWDgWPk2af0+StijSNOR8K/hN8D+1078oullsk4ASvSwjsCNO7sHU
a4xQU15JO6VqY18LANwORjrc9BaX4MguzsbFXBe6uFh1mVpXmFxSpUByQ+950MFz
/evPgP96wV+z4TtAwW2Z34rTiz4DxMI07XYNFUEOlS/gkUP2GxzymsO2kaYWTut3
SryCqeHEFbZfKb4urMk4xrIJC3CzWruS2Q0FHbBlfkgKN5wXVgkWFfiOucfCn+IQ
saqpold3f9jSkbtAV5w3vzfog8919MxKI9H614KuElnAtJ7BtZcs17dUy9u9COgE
yKriVokFQgqQ7XNDU+r3SeOWwks7AgMBAAGjga8wgawwDAYDVROTAQH/BAIwADAX
BgNVHSAEEDAOMA8GCMCGSAFLAwIBMAEwHgYDVROBBcwFYETYWxpY2VAc21pbWUu
ZXhhbXBsZTATBgNVHSUEDDAKBggrBgEFBQcDBDAOBgNVHQ8BAf8EBAMCBsAwHQYD
VR00BBYEF1v2zLIThQYSHJeuKWQqENMgZmZzMB8GA1UdIwQYMBaAFJEWjnwHFwyn
8QkoZTYaZxxodvRZMA0GCSqGSIb3DQEBAQUAA4IBAQBziaI2p86poGkjd/4KkkOH
G25nY/0eNARD6/oF0/sYonX2doizcGMk53riugAocCn5zbzhW/JVdYn30UxfyrZl

RAzEf7GHqgB/NyjOad3pdpVYeDh4ciNKjbs+aEoTWgAkoqENTlsRxlcvb7HVX524
 bKZa1oPTUNlm6QpivtqDIdqGJdGf8L1zLFXBuo2zL3HR+M9CDr4Opq2JCkzP0Qhp
 7poIccGE6I9Tsg+RrOA9iCQsPn1+Tg8YedjGzUWF07rNmT0TzPCVzUAuBlr+JJtz
 OKypyQ3eoZ6EPazXqMyHAVcsm0GI364IOA0b8PSrJNtjh+AqJ5QfH+0e7NSzNnEm
 MYICADCCAfwCAQEwbDBVMQ0wCwYDVQQKEwRJRVRGMREwDwYDVQQLEwhMQU1QUyBX
 RzExMC8GA1UEAxMoU2FtcGx1IEExBTBVTIFJTQSBDZXJ0aWZpY2F0aW9uIEFldGhv
 cm10eQITN0EFee11f0Kpolw69PhqzpqplzALBglghkgBZQMEAgGgaTAYBgkqhkiG
 9w0BCQMxCwYJKoZIhvcNAQcCBMBwGCSqGSIb3DQEJBTEPFw0yMTAyMjAxNTA3MDJa
 MC8GCSqGSIb3DQEJBDEiBCCXR0Udgr7J+TnI6kw8MpGtWVJPCnoAB+XfkDf78dWi
 cTANBgkqhkiG9w0BAQEFAASCAQCitU3JsEMd9FhqUu87UxYScDI1pDfZnXlvjges
 xBmmSy5lq5vvs+axKK/hTOR7YLSuLJLWxJgDCPEmHilhv5Tpj5mLH8qEXu4c+kK
 s9is53v0NvibhIvDEpnqNvL/kMVDak2gTqYHCE2Ij7qcWWNhnGdweMJZsBvLy/Xi
 BLad2t4qHY9lPaeMugDrxThNWEhjoDIOI5f7NpBPYvJgB7blcJhXqil5weYrJiGr
 hyTr56lff+Xjs8qjgrrzdJ8HHeUsxDJJulrX8auo+pIKudcu4lU8Ben2M9nCiVbEG
 aqbbPK7xip5c/YZEazWYAs8w+dif68J8Eo7QO/kkr45Tt5pf

--12b--

B.2.5. S/MIME signed-only signedData over a complex message, Wrapped Message

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 Wrapped Message header protection scheme.

It has the following structure:

```
application/pkcs7-mime [smime.p7m] 5631 bytes
  (unwraps to)
  message/rfc822 1613 bytes
    multipart/mixed 1549 bytes
      multipart/alternative 946 bytes
        text/plain 282 bytes
        text/html 380 bytes
        image/png inline 232 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-wrapped
Message-ID: <smime-one-part-complex-wrapped@lhp.example>
From: Alice <alice@smime.example>
To: Bob <bob@smime.example>
Date: Sat, 20 Feb 2021 12:04:02 -0500
```

MIIQOgYJKoZIhvcNAQcCoIIQKzCCECCCAQExDTALBglghkgBZQMEAgEwgGZjBgkq

hkiG9w0BBWggygZUBIIIGUE1JTUUTVmVyc2l1vbJjogMS4wQDdpDb250ZW50LVR5cGU6IG1lc3NhZ2UvcmlzODIyOyBmb3J3YXJkZWQ9Im5vIgOKDQPNSU1FLVZlcnNpb246IDEuMApDb250ZW50LVR5cGU6IG1lbHRpcGFydC9taXhlZDsGYm91bmRhcnc9IjhmZiIKU3ViamVjdDogc2l1pbWUt b251LXBhcnQtY29tcGxleC13cmFwcGVkCk1lc3NhZ2UtSUQ6IDxz bWltZS1vb mUt cGFydC1jb21wbGV4LXd yYXBwZWRAbGhwLmV4YW1w bGU+CkZyb206IEFs aWNlIDxhbGl jZUBzbWltZS5leGFt cGxlPgpUbzogQm9iIDxi b2JA c2l1pbWUuZXhhbXBs ZT4KRGF0ZTogU2F0LCAyM CBGZWIgmjAyMSAxMjowNDow MiAtMDUwMAoKLS04ZmYKTU1NRS1WZXJzaW9uOiAxLjAKQ29udGVudC1UeXB1oiBt dWx0aXBhcnQuYVWx0ZXJuYXRpd mU7IGJvdW5kYXJ5PSIxYUWUjCgotLT FhZQpDb250ZW50LVR5cGU6IHRleH QvcGxhaW47IGNoYXJzZXQ9InVzLWFf zY2lpIgpNSU1FLVZl cnNpb246IDEuMApDb250ZW50LVRYYW5zZmVyLUVuY29kaW5nOiA3Yml0CgpUaGlz IG1zIH RoZSBzbWltZS1vb mUt cGFydC1jb21wbGV4LXd yYXBwZWQgbWVzc2FnZS4K ClRoaxMGaXMgYSBzaWduZWQtb25seSBTL01JTUUGbWVzc2FnZSB2aWEgUEtDUyM3 IHNpZ251ZERhdGEUCBUaGUKcGF5bG9hZCBpcyBhIG1lbHRpcGFydC9hbHRlcm5h dGl2ZSBtZXNzYWdlIHd pdGggYW4gaW5saW51IG1tYWdlL3BuZwphdHRhY2htZW50 LiBJdCB1c2VzIHRoZSBXcmFwcGVkIE1lc3NhZ2UgaGVhZGVyIHByb3RlY3Rpb24gc2NoZW11LgokLS0gckFs aWNlCmFs aWNlQHNTaW11LmV4YW1w bGUKLS0xYWUKQ29u dGVudC1UeXB1oiB0ZXh0L2h0bWw7IGNoYXJzZXQ9InVzLWFf zY2lpIgpNSU1FLVZl cnNpb246IDEuMApDb250ZW50LVRYYW5zZmVyLUVuY29kaW5nOiA3Yml0Cgo8aHRt bD48aGVhZD48dGl0bGU+PC90aXR sZT48L2hlYWQ+PGJvZHk+CjxwPlRoaxMGaXMg dGhlIDxiPnNtaW11LW9uZS1wYXJ0LWNvbXBs ZXgt d3JhcHB1L2DwvYj4gbWVzc2Fn ZS48LE3A+CjxwPlRoaxMGaXMgYSBzaWduZWQtb25seSBTL01JTUUGbWVzc2FnZSB2aWEgUEtDUyM3IHNpZ251ZERhdGEUCBUaGUKcGF5bG9hZCBpcyBhIG1lbHRpcGFy dC9hbHRlcm5hdGl2ZSBtZXNzYWdlIHd pdGggYW4gaW5saW51IG1tYWdlL3BuZwph dHRhY2htZW50LiBJdCB1c2VzIHRoZSBXcmFwcGVkIE1lc3NhZ2UgaGVhZGVyIHBy b3RlY3Rpb24gc2NoZW11LjwvcD4KPHA+PHR0Pi0tIDxic i8+QWxpY2U8YnIvPmFs aWNlQHNTaW11LmV4YW1w bGU8L3R0PjwvcD48L2JvZHk+PC9odGl sPg otLT FhZS0t CgotLThmZg pDb250ZW50LVR5cGU6IG1tYWdlL3BuZw pDb250ZW50LVRYYW5zZmVy LUVuY29kaW5nOiBiYXNlNjJkQKQ29udGVudC1EaXNwb3NpdGl v bJjogaW5saW51Cgpp VkJPUncwS0dnb0FBQUFOU1VoRVVnQUFBQlFBQUFBVUNBWUFBUQUNOAiIwTkFBQUF jRWxfUVZSN DJ1VlRPeGJBCK1BZlM3MzluTzNUcFJ3MjBkcXB iZkFSUUvqT3l3aX Dz bkNOaORLbmJjTGs2NnNx bFQrenQ5Y2lkaOUrNkt3al oKc2dyemZjcVZNcEwyam8w NDQ3Zl lEcGVBCms rT25KSGtJaEFmVFBSAwNPaeFMNVlKenc3dmp2MFpxULdNL3VS aQp2ZFBMVFAmmtERDL4chBKohEBQFBSQpSVTVFcmtKZ2dnPtOKCi0tOGZML3S0K oIiHpjCCA88wg gK3oAMCAQCHeW8tJb0ROZdKzkJuh6HuPtQGirQDQYJKoZIhvcNAQENBgAwVTENMASGA1UECMESUVURjERMA8GA1UECxMIT EFNUFMgV0cxMTAvBgNV BAMTKFNhbXBsZSBMQU1QUyBSU0EgQ2VydGlmaWNhdGl vbiBBdXRob3Jp dHkwIBCN MTkxMTIwMDY1NDE4WhgPMjAlMjA5MjcW NjU0MThaMDsxDTALBgNVBAoTBELFVEYx ETAPBgNVBAS TCExBTvBTIFdHMRcwFYQYDVQQDEw5Bbg1jZSBMb3Zl bGFjZTCCASiW DQYJKoZIhvcNAQE BbQADggEPADCCAQoCggEBAJqVKfqLwaLjj+gBUCfkacKTg8cc2OtJ9ZSed6U3juoiZVPMLcP3MUKtLeLg9rlmAfIDLb/wlbdmadXPmrszyidmbuZmOpB5voVQfiLYy3iOx7YOqzXrl6udP07k0sv+UdSNRFxrfKeoQEFXgOaGdmnx4OG /e3plfIKM0dPzZLoOAJF5m500xxZPL74zFCWP2f1ZkuE4A6141koaZXCXN5XL7wWT LMLeNf9Byb5ksKqUuqEHAMdl nmoNMgjY9Vfvfcrv9w43GG8FtpSX+TWzB2zNS2OF +XIVnzRG5DeoULq8v88Z5bLpIJ/nx26r8A4SSwIBA vV4wPxAf1iPsIVKarUCAwEA AaoBrzCB RdAMBGNVHRMBaf8EAjAAMBcA1UdIAQQMALwDAYKIZIAAWUDAgEWATAE BgnVHREEFzAVBGnrhblG1jZUBzbWltZS5leGFt cGxlMBMGAIUdJYQMAGAOGCCsQAUF bwMEMA4GA1UdDwEB/wOEAFIDAdBgNVHO4EFqOUo1NB1UO8qCkvfAEj80eOr83

dw8wHwYDVR0jBBgwFoAUkTCOfAcXDKfxCSHlNhpNHGh29FkwDQYJKoZIhvcNAQEN
 BQADggEBAIFJeKCcsTKcFqQMPtryuJRGzJdYA+R9eBAuDLsatbtKt14FzkgRyOg3
 1/+Cw7H8e30iLrPIfLWN1qjHrjgOyIs5AQ/hgxLvLir3hEUUV2Z3MRsMtjH2x9SG9
 1PEM046gfPnc9gMGHjMTg1qvaKcLQP5UzPEYPLror2X4P5uXxaP0LIZRzWmkw1RF
 7FOD7PFB5v94M5274XYxW2W4uKGd7QGnUZROSvSYkGiWDp1JhqXwfDz8A0enITGX
 noEKAfVvjiCqh64PlhIeMorj36pgL19oWZD6YrzSWHUz1F00juyuOfQsqm6hvrDT
 qNpHNZ015fOURza1SkCvi9GFmNUPoVgwgGPPMIICt6ADAgECAhM3QQV57XV/Qqmi
 XDr0+GrOmgnXMA0GCSqGSIb3DQEEDQUAMFUXDTALBgNVBAoTBElFVEYxETAPBgNV
 BAStCEExBTBTIFdHMEwLwYDVQQDEyhtYW1wbGUgTEFNUFMgU1NBIEENlcnRpZmlj
 YXRpb24gQXV0aG9yaXR5MCAXDTE5MTEyMDA2NTQxOFoYDzIwNTIwOTI3MDY1NDE4
 WjA7MQ0wCwYDVQQKEWRJRVRGMREwDwYDVQQLEWhMQU1QUyBXRzEXMBUGA1UEAxMO
 QWxpY2UgTG92ZWxhY2UwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQC0
 9InoWdgWPK2af0+StijSNOR8K/hN8D+1078oullsk4ASvSwjsCNo7sHUA4xQU15J
 O6VqY18LANwORjrc9BaX4MguzsbFXBe6uFh1mVpXmFxpUBByQ+950MFz/evPgP96
 wV+z4TtAwW2Z34rTiz4DxMI07XYNFUE0ls/gkUP2Gxzyms02kaYWTut3SryCqeHE
 FbZfKb4urMk4xrIJC3CzWruS2Q0FHbBlfkgKN5wXVgkWFfiOucfCn+IQsaqpolD3
 f9jSkbtAV5w3vzfog8919MxKI9H614KuElnAtJ7BtZcs17dUy9u9COgEykRiVokF
 QgqQ7XNDU+r3SeOWwks7AgMBAAAgja8wgawwDAYDVR0TAQH/BAIwADAXBgNVHSAE
 EDAOMAwwGCMGSAFlAwIBMAEwHgYDVR0RBBCwFYETyWxpY2VAc21pbWUuZXhhbXBs
 ZTATBgNVHSEUDDAKBggrBgEFBQcDBDAOBgNVHQ8BAf8EBAMCBsAwHQYDVR0OBBYE
 FLv2zLlthQYSHJeuKWqQENMgZmZzMB8GA1UdIwQYMBAAAFJEWjnwHFWyn8QkoZTYa
 ZxxodvRZMA0GCSqGSIb3DQEEDQUAA4IBAQBziaI2p86poGkjd/4KkkOHG25nY/0e
 NARD6/oF0/sYonX2doizcGMk53riugAocCn5zbzhW/JVdYn30UxfyrZlRAZef7GH
 qgB/NyjOad3pdpVYeDh4ciNKjbs+aEoTWgAkoqENTlsRx1cvb7HVX524bKZaloPT
 UN1m6QpivtqDidqGJdGf8L1zLfxBuo2zL3HR+M9CDr4Opq2JCkzP0Qhp7poIccGE
 6I9Tsg+RrOA9iCQsPn1+Tg8YedjGzUWF07rNmT0TzPCVzUAuBlr+JJtz0KypyQ3e
 oZ6EPazXqMyHAVcsm0GI364IOA0b8PSrJNtjh+AqJ5QfH+0e7NSzNnEmMYICADCC
 AfwCAQEwbDBVMQ0wCwYDVQQKEWRJRVRGMREwDwYDVQQLEWhMQU1QUyBXRzExMC8G
 A1UEAxMoU2FtcGx1IEExBTBTIFJTQSBDZXRJ0aWZpY2F0aW9uIEF1dGhvcml0eQIT
 N0EFee11f0Kpolw69PhqzpqplzALBglghkgBZQMEAgGgaTAYBgkqhkiG9w0BCQMx
 CwYJKoZIhvcNAQcBMbWGCsqGSIb3DQEJBTEPFw0yMTAyMjAxNzA0MDJAMC8GCSqG
 SIb3DQEJBDEiBCDMOILEox46FkwxHI/3mD5yDe0N8CAfZ/xaQnI0a1yyOTANBgkq
 hkiG9w0BAQEFAASCAQBWzuGAP7C0InZ86JeaKimYKXpArooRzZnso+wJtXhZlmTX
 csHp783QCEKYE0F+rv1IrD+fcFULz8Lo7Mm+PWQbtkbx5uZR7IFLGLK+8i8wVCZj
 1Bs2lqpZ/qg1qP+ddCPwZuywITEGnjqqg76OHJOgxJniG3/teIy6dHMI20BogZjN
 kdVSbBhOa9GnTtnWJd2zH7t0tV16NyH3+pNn4DTUWR2IvRgxHky/KT7cIOTfQj9C
 HEizTlJQMDvHhoHs1WdwjAGjH3foH4CXP1/1bN+qBH2QAuRZ8+LueDcllQsPJXtc
 fUseHVMstoHac0rajLjDZ8FXSLCkmt06RRSQVsT0

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 5542 bytes
message/rfc822 1671 bytes
multipart/mixed 1607 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="ce9";
 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

--ce9

MIME-Version: 1.0
Content-Type: message/rfc822; forwarded="no"

MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="c33"
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

--c33

MIME-Version: 1.0
Content-Type: multipart/alternative; boundary="bb6"

--bb6

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
--bb6
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-wrapped</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 Wrapped Message
header protection scheme.</p>
<p><tt>-- <br/>Alice<br/>alice@smime.example</tt></p></body></html>
--bb6--

--c33
Content-Type: image/png
Content-Transfer-Encoding: base64
Content-Disposition: inline

iVBORw0KGgoAAAANSUhEUgAAABQAAAAUcAYAAACNiR0NAAAAcELEQVR42uVTOxbA
MAGS739nO3TpRw20dqpbfARQEjOywiwYnCtxDKnbcLk66sqlT+zt9cidkE+6KwkZ
sgrzfcqVmpL2jo0447gYDpeArk+OnJHkIhAfTPRicihAf5YJrw7vjv0ZWRWM/uli
vdPf1QZ2kDD9xppd8wAAAABJRU5ErkJggg==

--c33--

--ce9
Content-Transfer-Encoding: base64
Content-Type: application/pkcs7-signature; name="smime.p7s"

MIIJ4AYJKoZIhvcNAQcCoIIJ0TCCCC0CAQExDTALBglghkgBZQMEAgEwCwYJKoZI
hvcNAQcBoIIHjCCA88wggK3oAMCAQICEw8tJb0ROZdKzkJU6HuPTQGirQwDQYJ
KoZIhvcNAQENBQAwVTENMASGA1UEChMESUVURjERMA8GA1UECxMITEFNUFNgV0cx
MTAvBgNVBAMTKFhnbXBsZSBMQU1QUyBSU0EgQ2VydGlmaWNhdGlvb2ludXRob3Rl
dHkwIBcNMjEwMDY1NDE4WhgPMjA1MjA5MjcwNjU0MThaMDsxDTALBgNVBAoT
BELFVEYxETAPBgNVBAsTCExBTVBTIFdHMRcwFQYDVQQUDEw5BbGljZSBMb3ZlbGFj
ZTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBABJqVKfqlWALjj+gBUCfk
acKTg8cc2OtJ9ZSed6U3jUoiZVpMLcP3MUKtLeLg9rlmAfIDlB/wlbdmadXPmrsz
yidmbuZmOpB5voVQfiLYy3iOx7YOqzXrl6udP07k0sV+UdSNRFxrfKeoQEFXgOa
Gdmnx4OG/e3p1fIKM0dPzZLoOAJF5m500xzXPL74zFCWp2f1ZkuE4A6141koaZXC
N5XL7wWTLMLenF9Byb5ksKqUuqEHAMdlnmoNMgjY9VFVfcrrv9w43GG8FtpSX+TWz
B2zNS2OF+XIVnzRG5DeoULq8v88Z5bLpIJ/nx26r8A4SSwIBaVv4wPxAf1iPsIVK
arUCAwEAAaOBrzCBrdAMBgNVHRMBAf8EAAjAAMBcGA1UdIAQQMA4wDAYKYZIZIAWUD
AgEwATAeBgNVHREEFzAVgRNhbGljZUBzbWltZS5leGFTcGx1MBMGA1UdJQQMMAoG
CCsGAQUFBwMEMA4GA1UdDwEB/wQEAwIFIDAdBgNVHQ4EFgQUo1NB1UQ8gCkVfAEj
```

8OeOr83zdW8wHwYDVR0jBBgwFoAUkTCOfAcXDKfxCSHlNhpnHGh29FkwDQYJKoZI
hvcNAQENBQADggEBAlFJeKCcsTKcFqQMpTryujRGzJdYA+R9eBAuDLsatbtKt14F
zkgRyOg3l/+Cw7H8e30iLrPIfLWN1qjHrjgOyIs5AQ/hgxLvLir3hEUUV2Z3MRsMt
jH2x9SG91PEM046gfPnc9gMGHjMTglqvaKcLQP5UzpEYPLror2X4P5uXxaP0LIZR
zWmkw1RF7FOD7Pfb5v94M5274XYxW2W4uKGd7QGnUZROSvSYkGiWdp1JhqXwfDz8
A0enITGXnoEkaFvVjiCqh64PlhIeMorj36pgL19oWZD6YrzSWHUz1F00juyuOfQs
qm6hvrDTqNpHNZ015fOURza1SkCvi9GfmNUPoVgwggPPMIICt6ADAgECAhM3QQV5
7XV/QqmIXDr0+GrOmgnXMA0GCSqGSIb3DQEBDQUAMFUXDTALBgNVBAoTBE1FVEYx
ETAPBgNVBAStCEExBTBVTIFdHMEwLwYDVQQDEyhTYW1wbGUgTEFNUFMgU1NBIEEN1
cnRpZmljYXRpb24gQXV0aG9yaXR5MCAXDTE5MTEyMDA2NTQxOFoYDzIwNTIwOTI3
MDY1NDE4WjA7MQ0wCwYDVQQKEWRJRVRGMREwDwYDVQQLEWhMQU1QUyBXRzEXMBUG
A1UEAxMOQWxpY2UgTG92ZWxhY2UwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEK
AoIBAQC09InoWDgWPK2af0+StijSNOR8K/hN8D+1078oullsk4ASvSwjsCNo7sHU
a4xQU15JO6VqY18LANwORjrc9BaX4MguzsbFXBe6uFh1mVpXmFxpUByQ+950MFz
/evPgP96wV+z4TtAwW2Z34rTiz4DxMI07XYNFUEOlS/gkUP2Gxzyms02kaYWTut3
SryCqeHEfBzFk4urMk4xrIJC3CzWrus2Q0FHbBlfkgKN5wXVgkWFfiOucfCn+IQ
saqpol3f9jSkbtAV5w3vzfog8919MxKI9H614KuElnAtJ7BtZcs17dUy9u9COgE
ykRiVokFQgqQ7XNDU+r3SeOWwks7AgMBAAAgjga8wgawwDAYDVR0TAQH/BAIwADAX
BgNVHSAEEDAOMAAGCmCGSAFlAwIBMAEwHgYDVR0RBBCwFYETYWxpY2VAc21pbWUu
ZXhhbXBsZTATBgNVHSUEDDAKBggrBgEFBQcDBDAOBgNVHQ8BAF8EBAMCBsAwHQYD
VR00BBYEF1v2zLlthQYSHJeuKWqQENMgZmZzMB8GA1UdIwQYMBAAAFJEWjnwHFWyn
8QkoZTYaZxxodvRZMA0GCSqGSIb3DQEEDQUAA4IBAQBziaI2p86poGkjd/4KkkOH
G25nY/0eNARD6/0F0/sYonX2doizcGMk53riugAocCn5zbzhW/JVdYn30UxfyrZ1
RAzEf7GHqgB/NyjoAd3pdpVYeDh4ciNKjbs+aEoTWgAkoqENT1sRx1cvb7HVX524
bKZa1oPTUN1m6QpivtqDIdqGJdGf8L1zLfxBuo2zL3HR+M9CDr4Opq2JCkzP0Qhp
7poIccGE6I9Tsg+Rr0A9iCQsPn1+Tg8YedjGzUWF07rNmT0TzPCVzUAuBlr+JJtz
OKypQ3eoZ6EPazXqMyHAVcsm0GI364IOA0b8PSrJNtjh+AqJ5QfH+0e7NSzNnEm
MYICADCCafwCAQEwbDBVMQ0wCwYDVQQKEWRJRVRGMREwDwYDVQQLEWhMQU1QUyBX
RzExMC8GA1UEAxMoU2FtCGx1IEExBTBVTIFJTSBDZXXJ0aWZpY2F0aW9uIEF1dGhv
cm10eQITN0EFee11f0Kpolw69PhqzpqplzALBglghkgBZQMEAgGgATAYBgkqhkiG
9w0BCQMxCwYJKoZIhvcNAQcBMBwGCSqGSIb3DQEJBTEPFw0yMTAyMjAxNzA1MDJa
MC8GCSqGSIb3DQEJBDEiBCAv+o7fTfRF0qnpRsH2sYz0leh5w2W+5q6Nde9GJQWH
nTANBgkqhkiG9w0BAQEFAASCAQBrqtTw1eU834PA6rF6Vsac5dGAswyv4vh/EVxO
xBY7A+uEacaMOXRaSzktqehOkOGa3ld2bV6XmWbcR9kNvradw//dXOkctHW/cW6x
1BALj1aFABYmObCY/FTItu7nLGIAIQCM0W4OVHgH7I/QXOsz3o7hH68SWItJnLDy
cSEDzRKNh1v15cN0euY0mNA6HcvKchkiLWCj1pcJVMtq3FQE4GNee01x2Pz3ao7y
vDO/E/sliF2SiPS7GcgluywZ1ln5xAwR95/G/1UlqWFBXPAPgIMda1kDsqrI++tE
7aFVuQ9rEoAQJ8KeS8QWA/Lf/iefFfu0ESJxjRDdbJ3+gm5P

--ce9--

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.

Y2VAc21pbWUuZxHhbXBsZTwvdHQ+PC9wPjwvYm9keT48L2h0bWw+DQotLTl1Yy0t
DQoNCi0tNWRhDQpDb250ZW50LVR5cGU6IGltYWdlL3BuZw0KQ29udGVudC1UcmFu
c2Z1ci1FbmNvZGluZz0gYmFzZTY0DQpDb250ZW50LURpc3Bvc2l0aW9uOiBpbmVp
bmUNCg0KaVZCT1J3MEtHZ29BQUFBTlNVaEVVZ0FBQUJRQUFBQVVDQVlBQUFDtMlS
ME5BQUFBY0VsRVFWUjQydvZUT3hiQQ0KTUFnUzczOW5PM1RwUncyMGRxcGJmQVJR
RWpPeXdpdl1uQ3RrREtuYmNMazY2c3FsVCT6dD1jaWRrRSs2S3drWg0Kc2dyemZj
cVZNcEwyam8wNDQ3Z1lEcGVBCmsrT25KSgtJaEFmVFBSaWNpaEFmNVlKcnc3dmp2
MFpXUlDNL3VsaQ0KdmRQZjFRWjJrREQ5eHBwZDh3QUFBQUJKU1U1RXJrSmdnZz09
DQoNCi0tNWRhLS0NCqCCB6YwggPPMIICt6ADAgECAhMPLSW9ETmXSs5CVIEh7j00
Boq0MA0GCSqGSIsb3DQEBDQUAMFUDTALBgNVBAoTBELFVEYxETAPBgNVBAsTCEExB
TVBTIFdHMEtWlWYDVQQDEYhTYW1wbGUgTEFNUFMgU1NBIElcnRpb24g
QXV0aG9yaXR5MCAXDTE5MTEyMDA2NTQxOFoYDzIwNTIwOTI3MDY1NDE4WjA7MQ0w
CwYDVQQKEWRJRVRGMREwDwYDVQQLEWhMQU1QUyBXRzEXMBUGA1UEAxMQWxpY2Ug
TG92ZWxhY2UwggEiMA0GCSqGSIsb3DQEBBAQUAA4IBDwAwggEKAoIBAQCalsn6i8Gi
44/oAVAn5GnCh4PHHNjrSfWUnnelN41KImVaTC3D9zFCrS3i4Pa9ZgHyA5Qf8JW3
ZmnVz5q7M8onZm7mZjqQeb6FUH4i2Gmt4jse2Dqs165ernT905NLFf1HUjURca3y
nqEBBV4DmhnZp8eDhv3t6dXyCjNHT82S6DgCRZuTtMc1zy++MxQlqdn9WZLhOAO
peNZKGMVwjeVy+8FkyzC3jX/Qcm+ZLCq1LqhBwDhdZ5qDTI12PVX1X3K7/cONxhv
BbaU1/k1swdszUtjhflyFZ80RuQ3qFC6vL/PGewy6SCf58duq/AOEksCAW1b+MD8
QH9Yj7CFSmq1AgMBAAAGjga8wgawwDAYDVR0TAQH/BAIwADAXBgNVHSAEEDAOMAwwG
CmCGSAFlAwIBMAEwHgYDVR0RBBCwFYETYWxpY2VAc21pbWUuZxHhbXBsZTATBgNV
HSUEDDAKBggrBgEFBQcDBDAOBgNVHQ8BAf8EBAMCBSAwHQYDVR0OBBYEFKJTQdVE
PIApFXwBI/Dnjq/N83cPMB8GA1UdIwQYMBaAFJEwjnWHFwyn8QkoZTYaZxxodvRZ
MA0GCSqGSIsb3DQEBDQUAA4IBAQCBSXignLEynBakDKU68ro0RsyXWAPkfXgQLgy7
GrW7SrZeBc5IEcjoN9f/gsOx/Ht9Ii6zyBZVjdaox644DsiLQEP4YMS7y4q94RF
FdmDzEbDLYx9sfUhvdtXDN0oHz53PYDBh4zE4Nar2inCOD+VM6RGDy66K91+d+b
18Wj9CyGUc1ppMNURexTg+z3web/eDodu+F2MvtluLihne0Bp1GUTkr0mJBolG6d
SYal8Hw8/ANHpyEx156BJABb744gqoeuD9YSHjKK49+qYC9faFmQ+mK801h1M9Rd
NI7srjn0LKpuob6w06jaRzWdNeXz1Ec2tUpAr4vRhZjVD6FYMIIDzzCCAreAwIB
AgITN0EFee11f0Kpolw69Phqzpp1zANBgkqhkiG9w0BAQ0FADBMQ0wCwYDVQQK
EwRJRVRGMREwDwYDVQQLEWhMQU1QUyBXRzEXMC8GA1UEAxMoU2FtcGx1IEExBTBVT
IFJTQSBdZXJ0aWZpY2F0aW9uIEF1dGhvcml0eTAqFw0xOTExMjAwNjU0MThaGA8y
MDUyMDkyNzA2NTQxOFowOzENMAsGA1UEChMESUVURjERMA8GA1UECXMITEFNUFMg
V0cxFzAVBgNVBAMTDkFsaWNlIEExvdmVsYWNlMIIIBIjANBgkqhkiG9w0BAQEFAAO
AQ8AMIIBCGKAQEAtPSJ6Fg4Fj5Nmn9PkrYo0jTkfCv4TfA/pdO/KLpZbJOAER0s
I7Aja07B1GuMUFJeStulamNfCwDcDkY63PQW1+DILs7GxVwXurhYdZlaV5hcUqVA
ckPvedDBc/3rz4D/esFfs+E7QMftmd+K04s+A8TCNO12DRVBDpbP4JFD9hsc8prD
tpGmFk7rd0q8gqnhxBW2RZAeLqzJOMayCQtwslq7ktkNBR2wZX5ICjecF1YJfHx4
jrnHwp/iELGqqaNXd3/Y0pG7QFecN7836IPpdfTMSiPR+peCrhJZwLsewbWXLJe3
VMvbvQj0BMpEYlaJBUIKk01zQ1Pq90njlSjLowIDAQABO4GvMIGsMAWGA1UdEwEB
/wQCMAAwFwYDVR0gBBawDjAMBgpghkgBZQMCATAMB4GA1UdEQQXMBWBE2FsaWNl
QHNTaW1lLmV4YW1wbGUwEwYDVR0lBAwwCgYIKwYBBQUHAWQwDgYDVR0PAQH/BAQD
AgbAMB0GA1UdDgQWBBS79syyLR0GEhyXrilqkBDTIGZmczAfBgNVHSMEGDAWgBSR
MI58BxcMp/EJKGU2GmccaHb0WTANBgkqhkiG9w0BAQ0FAAOCAQEAc4miNqf0qaBp
I3f+CpJDhxtuZ2P9HjQEQ+v6BdP7GKJ19naIs3BjJ0d64roAKHAp+c284VvyVXWJ
99FMX8q2ZUQMxH+xh6oAfzcozmnd6XaVWHg4eHIjSo27PmhKE1oAJKKhDbdbEcZX
L2+x1V+duGymWtaD01DZukKYr7agyHahIXRn/C9cy31wbqNsy9x0fjPQg6+Dqat
iQpMz9EIAe6aCHHBhOiPU7IPkazgPYgkLD59fk4PGHnYxs1FhdO6zZk9E8zwlc1A

```

LgZa/iSbczsqckN3qGehD2sl6jMhwFXLJtBiN+uCDgNG/D0qyTbY4fgKieUHx/t
HuzUzzZxJjGCAGAwggH8AgEBMGwwVTENMAsgA1UEChMESUVURjERMA8GA1UECxMI
TEFNUFMgV0cxMTAvBgNVBAMTKFNhbXBsZSBMQU1QUyBSU0EgQ2VydGlmaWNhdGlv
biBBdXRob3JpdHkCEzdBBXntdX9CqaJcOvT4as6aqdcwCwYJYIZIAWUDBAIBoGkw
GAYJKoZIhvcNAQkDMQsGCSqGSIb3DQEHATAcBgkqhkiG9w0BCQUxDxcNMjEw
MTcwNjAyWjAvBgkqhkiG9w0BCQQxIgQgSnZFRpoKyudHBvkAo6hqyxtaGzBVpz8R
sk+FJtjH7PgWDQYJKoZIhvcNAQEBBQAEggEADAIUCPkW4o6qXePSs+Yh+ZPDq8Zy
v5hH1SNGGLmQP82ZDL/+zob54QvODTFnFb8SNL05nxIZlmZo/XtxRThlSiIy/Cnb
xL9dkylfOaOdtkc5MMv+W5AWQQ4CsJfkN+g9EPr+XcsFCn7Dsb/Vu836eZhSQ+tB
kttfKuhy/XKImI3fp5GLZhGu5NVWnwWC+1Um3AoKhmKhI3M8Kct84xpMGYXHJdlt
DfADNo6cWgQ0pQeF7mSh4gSneysep2koZNVx9LpCjoYzto6t5DorJBtBiZBr7qBg
jY68KcMpZ2N4IIPltcup96bHPeR+IkDqaF4EeeFIcYsEKBFRfkbF+qzgNw==

```

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 5510 bytes
  multipart/mixed 1637 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="34f";
  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

```

--34f

```

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
Content-Type: multipart/mixed; boundary="193"; protected-headers="v1"

--193

MIME-Version: 1.0

Content-Type: multipart/alternative; boundary="db5"

--db5

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

--db5

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>
--db5--
```

--193

Content-Type: image/png

Content-Transfer-Encoding: base64

Content-Disposition: inline

```
iVBORw0KGgoAAAANSUhEUgAAABQAAAAUCAyAAACNiR0NAAAAcElEQVR42uVT0xbA
MAgS739nO3TpRw20dqpbFARQEjOywiwYnCtkDKnbcLk66sqlT+zt9cidkE+6KwkZ
sgrzfcqVMpL2jo0447gYDpeArk+OnJHkIhAfTPRicihAf5YJrw7vjv0ZWRWM/uli
vdPf1QZ2kDD9xppd8wAAAAABJRU5ErkJggg==
```

--193--

--34f

Content-Transfer-Encoding: base64

Content-Type: application/pkcs7-signature; name="smime.p7s"

MIIJ4AYJKoZIhvcNAQcCoIIJ0TCCc0CAQExDTALBglghkgBZQMEAgEwCwYJKoZI
hvcNAQcBoIIHpjCCA88wggK3oAMCAQICEw8tJb0ROZdKzkJU6HuPTQGirQwDQYJ
KoZIhvcNAQENBQAwVTENMA8GA1UEChMESUVURjERMA8GA1UECmMITEFNUFNgV0cx
MTAvBgNVBAMTKFNhbXBsZSBMQU1QUyBSU0EgQ2VydG1maWNhdGlvbiBBdXR0b3Jp
dHkwIBcNMTEkxMTIwMDY1NDE4WhgPMjA1MjA5MjcwNjU0MThaMDsxDTALBglghkgBZQMEAgEwCwYJKoZIhvcNAQENBQADggEPADCCAQoCggEBABjVqfLwLjJj+gBUCfk
acKTg8cc2OtJ9ZSed6U3jUoizVpMLcP3MUKtLeLg9rImAfIDlB/wlbdmadXPmrsz
yidmbuZmOpB5voVQfILYYy3iOx7YQqzXrl6udP07k0sV+UdSNRFxrfKeoQEFXgOa
Gdmnx4OG/e3p1fIKM0dPzZLoOAJF5m500xzXPL74zFCWp2f1ZkuE4A6141koaZXC
N5XL7wWTLMLenF9Byb5ksKqUuqEHAMdlnmoNMgjY9VfVfcrv9w43GG8FtpSX+TWz
B2zNS2OF+XIVnzRG5DeoULq8v88Z5bLpIJ/nx26r8A4SSwIBaVv4wPxAf1iPsIVK
arUCAwEAAaOBrzCBrdAMBgNVHRMBAf8EAjAAMBcGA1UdIAQQMA4wDAYKYIZIAWUD
AgEwATAeBgNVHREEFzAVgRNhbGljZUBzbWltZS5leGFtcGx1MBMGA1UdJQQMMAoG
CCsGAQUFBwMEMA4GA1UdDwEB/wQEAwIFIDAdBgNVHQ4EFgQUo1NB1UQ8gCkVfAEj
8OeOr83zdW8wHwYDVR0jBBgwFoAUkTCOfAcXDKfXCSH1NhpHGh29FkwDQYJKoZI
hvcNAQENBQADggEBAIFJeKCsTKcFqQMPTryuJRgzJdYA+R9eBAuDLsatbtKt14F
zkgRyOg3l/+Cw7H8e30iLrPIfLWN1qjHrjgOyIs5AQ/hgxLvLir3hEUV2Z3MRsMt
jH2x9SG91PEM046gfpnc9gMGHjMTglqvaKcLQP5UzpeYPLror2X4P5uXxaP0LIZR
zWmkw1RF7FOD7Pfb5v94M5274XYxW2W4uKgd7QGnUZROSvSYkGiWDp1JhQXwfdZ8
A0enITGXnoEkaFvVjjiCqh64PlhIeMorj36pgL19oWZD6YrzSWHuz1F00juyuOfQs
qm6hvrDTqNpHNZ015fOURza1SkCvi9GFmNUPoVgwgGPPMIICt6ADAgECAhM3QQV5
7XV/QqmiXDr0+GrOmgnXMA0GCSqGSIB3DQEBDQUAMFUxDTALBglghkgBZQMEAgEwCwYJKoZIhvcNAQcCoIIJ0TCCc0CAQExDTALBglghkgBZQMEAgEwCwYJKoZI
hvcNAQcBoIIHpjCCA88wggK3oAMCAQICEw8tJb0ROZdKzkJU6HuPTQGirQwDQYJ
KoZIhvcNAQENBQAwVTENMA8GA1UEChMESUVURjERMA8GA1UECmMITEFNUFNgV0cx
MTAvBgNVBAMTKFNhbXBsZSBMQU1QUyBSU0EgQ2VydG1maWNhdGlvbiBBdXR0b3Jp
dHkwIBcNMTEkxMTIwMDY1NDE4WhgPMjA1MjA5MjcwNjU0MThaMDsxDTALBglghkgBZQMEAgEwCwYJKoZIhvcNAQENBQADggEPADCCAQoCggEBABjVqfLwLjJj+gBUCfk
acKTg8cc2OtJ9ZSed6U3jUoizVpMLcP3MUKtLeLg9rImAfIDlB/wlbdmadXPmrsz
yidmbuZmOpB5voVQfILYYy3iOx7YQqzXrl6udP07k0sV+UdSNRFxrfKeoQEFXgOa
Gdmnx4OG/e3p1fIKM0dPzZLoOAJF5m500xzXPL74zFCWp2f1ZkuE4A6141koaZXC
N5XL7wWTLMLenF9Byb5ksKqUuqEHAMdlnmoNMgjY9VfVfcrv9w43GG8FtpSX+TWz
B2zNS2OF+XIVnzRG5DeoULq8v88Z5bLpIJ/nx26r8A4SSwIBaVv4wPxAf1iPsIVK
arUCAwEAAaOBrzCBrdAMBgNVHRMBAf8EAjAAMBcGA1UdIAQQMA4wDAYKYIZIAWUD
AgEwATAeBgNVHREEFzAVgRNhbGljZUBzbWltZS5leGFtcGx1MBMGA1UdJQQMMAoG
CCsGAQUFBwMEMA4GA1UdDwEB/wQEAwIFIDAdBgNVHQ4EFgQUo1NB1UQ8gCkVfAEj
8OeOr83zdW8wHwYDVR0jBBgwFoAUkTCOfAcXDKfXCSH1NhpHGh29FkwDQYJKoZI
hvcNAQENBQADggEBAIFJeKCsTKcFqQMPTryuJRgzJdYA+R9eBAuDLsatbtKt14F
zkgRyOg3l/+Cw7H8e30iLrPIfLWN1qjHrjgOyIs5AQ/hgxLvLir3hEUV2Z3MRsMt
jH2x9SG91PEM046gfpnc9gMGHjMTglqvaKcLQP5UzpeYPLror2X4P5uXxaP0LIZR
zWmkw1RF7FOD7Pfb5v94M5274XYxW2W4uKgd7QGnUZROSvSYkGiWDp1JhQXwfdZ8
A0enITGXnoEkaFvVjjiCqh64PlhIeMorj36pgL19oWZD6YrzSWHuz1F00juyuOfQs
qm6hvrDTqNpHNZ015fOURza1SkCvi9GFmNUPoVgwgGPPMIICt6ADAgECAhM3QQV5
7XV/QqmiXDr0+GrOmgnXMA0GCSqGSIB3DQEBDQUAMFUxDTALBglghkgBZQMEAgEwCwYJKoZIhvcNAQcCoIIJ0TCCc0CAQExDTALBglghkgBZQMEAgEwCwYJKoZI
hvcNAQcBoIIHpjCCA88wggK3oAMCAQICEw8tJb0ROZdKzkJU6HuPTQGirQwDQYJ
KoZIhvcNAQENBQAwVTENMA8GA1UEChMESUVURjERMA8GA1UECmMITEFNUFNgV0cx
MTAvBgNVBAMTKFNhbXBsZSBMQU1QUyBSU0EgQ2VydG1maWNhdGlvbiBBdXR0b3Jp
dHkwIBcNMTEkxMTIwMDY1NDE4WhgPMjA1MjA5MjcwNjU0MThaMDsxDTALBglghkgBZQMEAgEwCwYJKoZIhvcNAQENBQADggEPADCCAQoCggEBABjVqfLwLjJj+gBUCfk
acKTg8cc2OtJ9ZSed6U3jUoizVpMLcP3MUKtLeLg9rImAfIDlB/wlbdmadXPmrsz
yidmbuZmOpB5voVQfILYYy3iOx7YQqzXrl6udP07k0sV+UdSNRFxrfKeoQEFXgOa
Gdmnx4OG/e3p1fIKM0dPzZLoOAJF5m500xzXPL74zFCWp2f1ZkuE4A6141koaZXC
N5XL7wWTLMLenF9Byb5ksKqUuqEHAMdlnmoNMgjY9VfVfcrv9w43GG8FtpSX+TWz
B2zNS2OF+XIVnzRG5DeoULq8v88Z5bLpIJ/nx26r8A4SSwIBaVv4wPxAf1iPsIVK
arUCAwEAAaOBrzCBrdAMBgNVHRMBAf8EAjAAMBcGA1UdIAQQMA4wDAYKYIZIAWUD
AgEwATAeBgNVHREEFzAVgRNhbGljZUBzbWltZS5leGFtcGx1MBMGA1UdJQQMMAoG
CCsGAQUFBwMEMA4GA1UdDwEB/wQEAwIFIDAdBgNVHQ4EFgQUo1NB1UQ8gCkVfAEj
8OeOr83zdW8wHwYDVR0jBBgwFoAUkTCOfAcXDKfXCSH1NhpHGh29FkwDQYJKoZI
hvcNAQENBQADggEBAIFJeKCsTKcFqQMPTryuJRgzJdYA+R9eBAuDLsatbtKt14F
zkgRyOg3l/+Cw7H8e30iLrPIfLWN1qjHrjgOyIs5AQ/hgxLvLir3hEUV2Z3MRsMt
jH2x9SG91PEM046gfpnc9gMGHjMTglqvaKcLQP5UzpeYPLror2X4P5uXxaP0LIZR
zWmkw1RF7FOD7Pfb5v94M5274XYxW2W4uKgd7QGnUZROSvSYkGiWDp1JhQXwfdZ8
A0enITGXnoEkaFvVjjiCqh64PlhIeMorj36pgL19oWZD6YrzSWHuz1F00juyuOfQs
qm6hvrDTqNpHNZ015fOURza1SkCvi9GFmNUPoVgwgGPPMIICt6ADAgECAhM3QQV5
7XV/QqmiXDr0+GrOmgnXMA0GCSqGSIB3DQEBDQUAMFUxDTALBglghkgBZQMEAgEwCwYJKoZIhvcNAQcCoIIJ0TCCc0CAQExDTALBglghkgBZQMEAgEwCwYJKoZI

```

cm10eQITN0EFee11f0Kpolw69PhqzpqplzALBglghkgBZQMEAgGgaTAYBgkqhkiG
9w0BCQMxCwYJKoZIhvcNAQcBMBwGCSqGSIb3DQEJBTEPFw0yMTAyMjAxNzA3MDJa
MC8GCSqGSIb3DQEJBDEiBCBpheScfJ+ESh8/z2r5jHx3Lw+5VkH8zTicO3HRGxfm
ozANBgkqhkiG9w0BAQEFAASCAQADy9VgxUcoI8DWKdyHqPM8nLuaHB1B/SONgbzi
4SlgIMs4wR6S02LpiG36z4/zFw0JUbvqwC2WJN7+W0Vra6ZX/x7Hfmv+uqdsMW6j
r8IXATRFWNm6GEbih2BsYABTNy8z0JGs+y6dcNNdDIwDJIkJETi+xvleFA0deoWI
PyHmUjppzzjOcTAKFnSsa4lwSB0ty8lZPW6u0klUx+VVGRkkgg/0uXTBBlyGD02gbw
q5893Rx03g5zzxaYJP03zyO/WW7FmCJNNQbyZbQD8R4rvR0hVna0r7XoW4Q+WZfU
Dz29oLszzmumpedAaP7q/M0jySdSjWfQn1W5hHHhAMIlwcqt

```

--34f--

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] 7345 bytes
  (decrypts to)
  application/pkcs7-mime [smime.p7m] 4436 bytes
    (unwraps to)
    message/rfc822 679 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

```

```

MIIVLAYJKoZIhvcNAQcDoIIVHTCCFRkCAQAxggMQMIIBhAIBADBBSMFUxDTALBgNV
BAoTBELFVEYxETAPBgNVBAsTCExBTBTIFdHMTEwLWYDVQDEyhtYWlwbGUgTEFN

```

UFMgU1NBIE1cnRpZmljYXRpb24gQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIeh7j00
Boq0MA0GCSqGSIB3DQEBAQUABIIBAFHb+aM8bhyJ1nFFuBDyyBVQf2IplykrvvYb
mKqBk08i2gecPSOMTkW5e2oQ4+WT4rtU4E0JXfMSA2KukKc+QUA3ycVCoL5zhetX
GsEx74S5P4JMY/uAoyBLEogGNI2lvagvgOGkqHJCZAjKjPNmqyTfafyv1Y4BQRQ+
WJi7mURDIbgrc0xfcC/yt7UWxFlfUhm6n7rTvRKhe4D0EOOB8yKupUgcDzBMTw5F
P9HEy0vFiJl2+LNKSsOPhVp0PbPkMCVi+ERTXEgV7C7BRVVYBiprpYJxJry09t3E
jmIupqH2MgXx1AKFpBsdlPWfI1mrMVZTBpRgy8Bds7CORgWbs0MwggGEAgEAMGww
VTENMA8GA1UEChMESUVURjERMA8GA1UECXMITEFNUFMgV0cxMTAvBgNVBAMTKFNh
bXBsZSBMQUU1QUYBSU0EgQ2VydGlmaWNhdGlubiBBdXR0b3JpdHkCEzB8R0APhiY6
HGLS64MvlsDXhpQwDQYJKoZIhvcNAQEBBQAEggEAX1PxPDD1V2Wo766+MhR821W8
pD0GWAM1ScYPggh4t5OFmSjFtyiqawhMcQhoRsAkGV387oXupYXH/1kaD7nIdZW+
pZK1/RZUU0txvlsRIPJduXcWm/Dsu0lQtQSfcg5Fas1SMjBpMI41BD2KC9M5meDP
NqHnzNMfV0ZiPO6x+bTCXhds8WTi/B2DDyXGjEan6RUFw6rKNXwbXoR0DJCMosF5
55gQuo1k040YMqYRwdsJGETr/r/JaEPwNekogAFuXBkNE3JQB7aVgePp8mI2NI1U
0nP6eXp95UwLsoA/zwbOv9XSYgQDCcQ0MWycXmmn4ysbeWi1p7P+6CLwgx/TNTCC
Ef4GCSqGSIB3DQEHAATAdbglghkgBZQMEAAIEEN9EoELwqIPQUHcQvENM3K+AghHQ
7MaGZ6VZ5f9fpYjTHCbQSjcBtsF3qd7/z94CkYE+Fdt4Xtm91GLDSRONaVuT9yV6
vd3hoFTCfrX1aQSzzHn3SPtIh7ySaTG70ctsXP33UjcmjzDbvvyvfi1lmxsct5rSx
e+cJ4z++pLB0vQeq1JlbuqY8SkSX9FyDZegnUD+zCB3qv7YSZEwD+EjifauMcr15
p29hRgVx522WoILf6Ty14stVYot76cyOYE5A1EUMxBg98tLLzNvgvpevmhZwNzby
B3v68cMTXh8Zm8UB6F17oxdLFIszhEMnM4v2RSWB507L5C4ab+zWpB58AcOeIesg
E9TvdhcJVsiQHLMtVqxXcyyzlh/TlglYznfi4+Q0gNTTS9kp5y2Jpl8AWiHV31JH
ltigpNDS1fbskC4ZUKNLmwMTed03kH2leAZGK9afAC+nNwKvSlhWovXXujmTwGao
8fQPC9cKfRS3tx5dOnEY5A6ZPbAx3SkcdHpUc/Z6Z9at0NnN80pp155sichJeP+Q
yoWX/IMhZwNksoiP1Wqa2KYGk8913EvBOOKMH3G/IOcilg75VxjfkQ/IrB6xrhb7
wY3YCV14MtJ4T9gi0rtkXxq6YfJ6LQVXP3BWpmlf3xwxQn3HUsQNFO/dESQMikOy
PgNT/wkwX0+v0XY59maI2tF9sMfiheLeRRjPDbwaXNCX4ghzpoA0KQ1+0/upcXPd
O2sskI3b3qh+gbRhTUOXAMA5i/POQ6QOj/0jxfbn081YdiHE49jlx5MA00u/yn2V
WKLdKXE570tX5Z3upvQvLVYuc7+hfsr0oIC/A+4UKzt3G3kjmHqKvkPeP4ytu5Cw
VxRQlhl+rWISO/EzflNHsgNwE/X3eOmub8vN1/fX9ng5hMVaz38pAQyQysr2Rg2s
ZDasrLS4kWuG0tv8gXD+Lm34r31bQf1+0NoVpJFV0iHYzBcmL+refdBec9Jfm0yI
KkX1YkAovvlnYL5ZYzP8E08hNtZW+rln041yyZa12hR1ORO61Bqxb9W23vTgU404
vIRppUbJrf6tmYQMiYXkC+Kugur1nBJtEbLQ2WurYfSkdrZYLg6+cs/K+sGgCMI
OGokK2ntwmLWHCVU9w15i+7G0HYxZkschUQeIokU2M6KePbp36Mb0vQ1VJhlqTmU
HdW6EDk+iXDNW72gZccDyPhzbhZT2g4iWH16xA5iydhE9le80boq4370lgMIHUKS
2+cEArcITxmKpDQWxREYF74jJyz2Yf8rZY4uI6j97+LHYlds7X5HIIq37xVUKUud
sDav+1XMqygilVzgdQ6MTKH29rK+/OKJhWZYN5HDGUIa4GzskjL9Sp93xG+sRvtP
tC2bhURNdHjg7HyYH+RldvxN74NiFrNCj39TXyw5Tzs44nxsVqghdu04BYMm5uGp
9rN4c7Asn7kfjg9rmntnmnmBotKncRM4W1ybT0zZ4QoBCv12306QKgl13Qiv4E2e
3l/POH7VEtTBeYph3JUHCjoF/DU7lQetAaH3sKDdRqvxb8pjvQKI+q3NLUHYMLdl
/HqrtNXq4ItRsfz+yYsEKlw68fPncK40EVjxD8e1kP9iccyhEWK9sS+zZmsJmRP1
+CzHNDV/3F4V2eaa+YRiBgerv8jjqKhozquzKBnFerDrGvBnctYkBCL04sGowv3c
uxADq5pwlsBo2XIwsA6/hKtCiJpkIOiPjawE+uKwDiQdGutdxOx5v/wk7McMU0qO
tjhrKGa3WqQ7w91LO/xqNVBsGxKSDsyCZuKnpYlg3MgRK5JEq7GngLiBKRN3EErD
f74gk2ZQ51+4leokY/3YTYhAFnDabzhxLK2vZxuc5JWOScoo/Ej7AATgKkhr1U/g
CHvGyXxqrozMu/Vks564d4QTx7SHCOzJs0pIeN79muMOWEFYBKqJWZPxyzZ+Bx9
p97BbhQwhJ3sCJPiWMrLUJCI3d/DDPkz8IPru7rBmuYfTJv2buakTrR4hwjg8oK1
2YnhHumejoHzR9EfDQe1F3hYZSzwCH64ODMsSXGCRZjps7GulKWvdRxAiZHHCCA8

98vBO6pjBFG+JlKVufCTecBAyFKQOToYBMiQ195wzucZjnEeFtBDlaSwTJAx8rM2
ROR5DasKHRqdV6i2LV4b/3Xq5CUqZw3Q/kZcdSQTrqtDafc5lTLs/dPdCVWr/XAh
wjBgP9a1Ki33QhB73CFNTM4T9HAgR4SkqqpfEQEWkcJOIE3K7pfcQbplvR2uIIdg
gExjg5vyMloBFEO2YBcBi8bzUKF+sVpIkaOyfeD/tUydl10e/eDkwMD6Mx01ssgT
POJKR7EggddGlm/BCB29IekA5Y4Ydc7Gs1OFhO8zC2LCm5OHfNgzCaOos6lZtpzA
II9ihCb2/P0VRO0XSJ4RoR9Srj4DJji/VlzhqqsWZJQyzqJMRJT15mQHf2tOmobj
PCHpkJVWJNjHphbKTCqfokzHh1YnOvTJ2f0svarDhV8H3q9cM+ODMDPFOARjZ/hi
ciDo6010MciMAYzh5CoAbLQgz1HNUZIM4CCqidPVzHyn11IifhH+yEWkXkkCO8QV
1kDFbwmBhLRPawpIxs7QuZ0aICJBdGZ2Xwx55VAbht7SObl1NYbM50QeMtpzJC7
0vKgPkoctvuqR8vO4lsIqxUc6vtHW8C8YWHhz8g9oLBPeR0o/0I4+AePScm/BICy
DrnYGfFM9C/rMU+PateE/dvsGiW6dTM+9SUFqEqwIOazGfAwE83G85ZVePQOQ7RB
jxvZkgNsg7DZkbuylEmSRUa5gR0wtTH+4jVtYo9Zqrjw7NOvn/OLIIYDcpxQBrUE
/ntfknMq8luYOMou8YJCIOtx/wL89sYZhJu49H657dGB/A2tpGRVsb820Iei7rhu
+9quDIPXoPgBCEPh8k5eLTF23XJTFti2sx0D7WULXwhiX0+0CfvQNFt8ptJURPB9/
GzNzN0brNex9YUbfEAeGh6BiopGLTAeauu/VSc6J0Dl2uxLtt/sqx5riBDvgiXpu
vp+N2213sejyMeQ1iO3EJkHAHNpAFbMi6uEeMVCNneg9IxJj8lodiCaWKxjQafhY
i97omBTNjLQWXj3gCyIr4gK8aD9jrcixrPrUuKlyO4jdSuprINoQcDLElT/yPd/O
OTwDZewzyg LHRI/2eg0JPhtjZer/m+stDLbRxnkhKGfwjTR7Redk0cX4oLPiyVI40
mRZ300kMZ53iYRvzrsChO+L7Z3D6q5nZ2vO5yKFvfHgcmY3RZW9WyaiCF+wnLGD+
gcOtrcMs+SYc1F01xCpCND2obYK0icvliqH4TpAuSrW0bYCTm6hzoDdbW10Btca1
08D6XVusUPgy4o683tf5TyqMZyqEssG6Uby+08HElcJ4p1jzb50VxwWFrMkfntREV
Birra5k4+/Td6nOWE/Ba6lCOWVC8cBylqpbkKsmlIWNrbbgZmfLx9hgflTtxtCZQ
+DaWbvzEEeH6qyGy8VR/rX6kU0+rHMIyohPbk35VysC/s87OfBsuUheFCigfC7xE
v69dle3NANXQpCE8OyI1L063AWlQBxEvEMfkutCX9LM/w2h7PI7DGu7lNa1CxTo
g/74mJrIT9lneVCK1EpkmEMCimLd5NzjUcGatCLu574LfGpsOEDRUDVi8HBJOAP
spptpgQ8LMAjnvWilPQZcbd/0WvRzzKEp8i5k3IvtVHi/aFu91ZvnopgDJe43L30
tT3Kt9d/ZjHRswW4MT8vnCiDkBNF7TtyTC/jUq6pOuHglfc5H6QRgEjow/maBCB/
ApoGhlvCv+7J8ExVzkesaqrcTWQpHmq2szcTpnnhjgzV5W9CHGv2R0GcQGhvkBB
Ds4wYl+OKDQhXczbqX7C9bJOjDb6hhlQhTt101/M5iBdW53k2OCcliV056KNLFhd
yLDvXZg7r7IuGo751b9urObCI/w2KGDfN3P4Y8yRseJeBY9m+txWMJNyhCyNJQnn
7jLZ3es8cx/zQC/6AUQtNrjHzM+sIoSxSHXnS61Akj21zY0qyn6pZa1PgVM0HIy6
I5r4BTGdIeI/kc6LoKhrrfgeQnH6PwZmmddNIFQo61a3lpXuWgOZfQWOILo7L+2dR
neQ5AYaQj0QdH8z8aYrIgwWfzxFzETtnGJkE/HoN/MNGSaMD2x5b4y8ObDpvAkG5
AD8/VxZosBJE1hTz/v7DBFY062MdYDbKHkBSOAxUPMI0ivu8yV5JzC6+x/98L+C7
NJTsg62OIWXqgAX+NHZbFDdeIYMcExoMH8R/mz1zLibFZG8f4Buv73rdhwuRQ1/F
aKAXL58eFl/ppkEvFEGrJhOKtXjQv2mElloseTc64JuG7wXql0/LW22Fiw+b9vP8z
aowf6DrVDB4CiZBvbJpyk/t8EtByn0JLq+Qp/f5FgIglB0DWteA1PVC22i0zlg/d
+aVKtOHRCSJXupP+jIjdJUekwJSZCid72SmwS61fCinPJlVedq700A/SrJ9eg5Om
Etg28g9N3x3BzC4Q+gI5CMSK1fC3d2xHohxxdkw02MJWdOXbjwPaPxgqYbngJC4E
WLCXLPtLw6XuTJ61QJRpf3kk6REmqnRlDz8Dmm3ocpCcNLa7Vo05LkCnZfUvmZc4
jw/2JwuLcZR9yooiuHRMZj/WOFzRhPmWQWwCESCqCkyfNnXLKVsOZfWaUbNapIbA
5EOZoVpFQYZRz00Q7vdSodDtJ0REPxyvbjGomJTYm8VgsICQZVTahU8cNkRgh3KF
tqULWhLK7TzOz12rrrr1+LuSqlpb+QM0Az4ALYByeWEKno920ZaCfa/DxxMitx/Zy
RDfAtYiUzOmtWKcJnGfPzuInCHQ7QRYh2+xDh/o9k5qSeSV+lrG4MlI0sptm4lfN
W6oEJR7Y99IoItlenqjicyLDYpJavZCgMjHznCSPffWziOB8Vy1vpbs80mTQlvN2
J2V6HqLTgDg27MO6vZoBjjsjBdW+AJcwOzzY0eMvT+hEkLqcSRXXEB40Wr/qtwFv
aLYhIToRENYvxBqQmXWL8iT2mCs57m1sr0tvP2t7J4DWbp4Coipy2IFLC4vZLK8
KgfPwDld7qdZewykn9tzisOdx83ta0qeXc02kXsvxglglx1hO+DL6oamH2G1BBz

```

yVvADnw3C72aV6BKL5XFjbW5WdqKr0/2Gh8EE6IPZIW9TlMbt2TxSTdGxXDgslBB
plIDq1Qo47imspSjw1lbZm/duczPWuDpNW1f9uHRyIPcA8QaqXA+hvgeLbVpJuJG
6Y11FEYeIl+0tX251S9qhkDCvZ8MIZZ2muqYoB/Bac/CsbkoGJHgF5kg1RNBMCZv
aUGnTA/PaUEDyHJY74VsJJFVv8Hbsvwi5M0AUuAIIy60lGL3VZqQRdQjInJKEIXp
szLOcHyaL8tHY0IRSP4XaSR6hiEbFJvbPUIKS4TqTr9N+mTlFeVkJXxjGJVqwcxn
GSohbJc93gt3r2sS7HAr5fhJI3xDyXIYhWmRIQatv1Kh5SXsg9wSVMNFn4D1Q149
Flb9J+ydb3ENJlVnOaKGC/hyGhULNAUTDyg+pqz3Nu5lweJgFNgz3/W/KPNnIFnM
6vJto9bEpNKATOOBLXW20ztJCjgH0DD7AvQAVTGu8208MBL8PueUDlUysqZduTay
f2aVXIcEfPFwXR8lzHtDe87Iu/RqKwPnkHy+nFRKUSVhyhQ3EgnWZpLRNzHgPxvf
C74UbBFRBARWFRty28HGPqM75jNsOIsquad+9gxleRsuPElklsljiXlvDTlreEYE/
EF56h9hdn88C7SEO4KFMbI/6ae62JQdpO7CPgq+5YGHMVUZeQHJZkfLAQUVTCRQt
cZH86BtnMyKPZeovEd0guyX0kv27gswviZXflh0ey5voAGw0EH9j6+z5SN0sPhry
AzwG8mH27qDlrrGCnlGx5FOS39+xtuuseqAW+iQgDk9IGrqAstMQYRWlkRYXKQlG
y/1c1Q5/M6kyq5M2iI9ggd7hrqTcEh9Xy1dRBPdCljXyWZo2eTnp0n9whXZbMtLu
lIZc102dTlWWXM7uLK3xDQS653AQKc8C46DW3GslHl5+jW00C5orPHh5xeLX9UO

```

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] 7305 bytes
  (decrypts to)
application/pkcs7-mime [smime.p7m] 4406 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

```

```

MIIIVDAYJKoZIhvcNAQcDoIIU/TCCFPkCAQAxggMQMIIBhAIBADBsmFUxDTALBgNV
BAoTBELFVEYxETAPBgNVBAsTCEExBTBVTIFdHMTEwLWYDVQQDEYhTYWlwbGUgTEFN
UFMgU1NBIElnRnRpb24gQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIEh7j00
Boq0MA0GCSqGSIb3DQEBAQUABIIBAEqWQtP9NMP0lborDI5F55uEoZxerbw2f8G8
04jR822TF4ehQnzqt1Smtb3q7XZZGz3OVYv0JOO2DWrWWbSzaaWHXwJ8HdM0vxio

```

87SvZMWXXzwrZSyrabmCte7HhJOo0FYqMphkC8UoGtIE+J5Z1XpZqjpiicTDHZPD
qKPIXCE026LS1uJO/1l/ON5cBrdMR1zEE/tn12vA3e95pUEM2ILObukZPPKLiTfr
ejLM2/oQUk1Ymh54leeC3dQA0xIf0Wktzrp4qt/qJPPKI/RCw/JL0Saf2x005pET
PBRhxQdPEyJkFbRIOM/FMa+LkAqzjHlJI6MbYs7a+zAZvqH/tXkwggGEAgEAMGww
VTENMASGA1UEChMESUVURjERMA8GA1UECXMITEFNUFMgV0cxMTAvBgNVBAMTKFNh
bXBsZSBMQU1QUyBSU0EgQ2VydG1maWNhdG1vbiBBdXRob3JpdHkCEzB8R0APhiY6
HGLS64MvlsDXhpQwDQYJKoZIhvcNAQEBBQAEggEABaLUv4/qgPpg9LQVoTcto3J
8+wK32x1FwCr3LzD4A+3AZGAzqgJ6roO/cyDbz6swNjZQb6IvsHrxn2hCLyGS7JZ
pxaqvNh0MTZ7ppvAAMY/cbtim6oo+aR+YBFMuUejNy2Lf4g9Quqs7C86BqWt/DDR
8012vrQcTRVqxxgtaJtTSHXPZVQeoTL9QvyvBR69XJ4fNvap1F5CVP1GONwVWgYd
7u1FQCViH1ASwcJ2VMYTAp2vWgrghn6taCB5NuzPH6TLqXM33bzaEZ9+7ya0kOyC
h6PtoIm+Sk504F3qTf3EZ9l+pZw9dYKmHXnJSXzhInzob22BUwmi8rmAhyz7YDCC
Ed4GCSqSISb3DQEHATAdBg1ghkgBZQMEAAIEECnEpHap3uuwIy1DMX4JXriAghGw
Y9Dgh6eaEPJSGb2YLpt5P4NZqy1iFQN5A5F/ejZ+0XBWbhPihaOCRKaixUL0XFX0
f1THjHHFDNcuiZ2dxbGtWtuCZkxt44ycJ2GOJpCNcWVn00aJckEyiPxbj4yu16d
pqbT2G4Pt6DEW8teMJFNpaM7AcGbp04KtF02zIy1PQRjQRafhF08+7Jkm8ndRPUP
bnfOdLq+oIerDaMDlr84VyUEasjJzIS5xh7+Igilkl09cGQViTaOEtDhhL19sWrn
Tdmrit+/jso6IPZKilkaA8U1sZ4B3gWEjyxOphDKmtzOY5P5hQNcbXquk6CQT+N0
2XB5h9OdYPQc5hSUY3PxG0WwUovzQGAQLH/LwCm57sjfSNdTYJO4NiJQB5kIzmSI
8KLqLquMser7JzSyhGaatw3zC9rZ152FUohJQk3OSIzeMhJoXrQ1lyWEQOSfdCFo
+iaV70jHoEYQtmcamzZwO118JN4FyufRh7DyCBi4RoDx7OwWgKr601VrhCPZNwV
r+8Ysuqprpb1YEP1E1cqL0ZxVX5z21UQ133U08p4CV9fW0TuuNnMFRARnfnwoXFS
ORqrSR45G/274tG2/j3R94EdomMSJ8/Zx/qf7fou+EkdhfVNB/6ANb2jAm37bUeg
I89QvN/BTVcXwhMDsYV6OqPMAHwD3B/O7yF8HjyRiVh78bUX9rU1pIgxSRmnnyB
1noOrWKpacjxQenLebNa8CZVG4ZpQRa3f/NXOcS17auNb/qoT/xtgcTaWb6jF5M/
D3uLDiILH/jCyDag1L7ItSzTKu2BCH9tNXy2DVV0FSMTyFOLrYaZpYGLULVoly+u
yBqTQram5ZxmWjGhM80snWlmaB4kQ1FBWoW++rnEbQ9JEL+n6UxTJHBbR6bNuY7u
5jjYih1tEKM7Y6cQbWn/PykRIjP76mukR/PI84WHQGP+n6K8QjCP32Ij5v0BdXCN
KftDYROYNGK168oej0ozUpPnz5LJw3vbDEFzMVVCjEY2qOD7EdTFAYoJnW4LuGW
W43/PKeEi7smTQWxGWrBIFxPwuNkyMOHLGiKXqSJSzj531jTiGasWVpHibEK1US
IWOXef/7Q/PZvCa8vxmVGowSQ7gWQTVEOhKi0MV71YuxDTWRacPetjFzkWZOzTHF
5gFV+/CY2W5VXVSKIR5mr/jjQtBu+7LOAep2MGq1u1LzJgXDaOkPR5Rz6orfCz70
M7oE85uq430h6goP4YKeCU1sxSE9YXRqICN83AhY7JCzrP4bKVnKdia56XEmxMKR
LQ29Z2zSakaIPKbSmxMuIqkn1OV29PGG1KztSDonWIFVVLJb6Qne8altI7zTxm11
IMi5zxcto96g35HGN1V0h9zJKA8xOf6q18yhfJnWQ0ONkMpFhHPTOXaU1r4hzm3
mPEnuG94PWMw6EKi485rsY0tZgE/PZr1slDsxmAO6r06mqwc5NfNZoHwN16FWFZ6
1uRmctWEMW7gHeqly4TfXH4QiRMXAuzDdrYnVjWGqN1k3zEY/v/ppxI/woU4wBmw
pxwr3LTvna/8jpk060hM8ZUkAs9zYbtQBGLqrSy1prf+nplrXDQhkiGbV3Lpx2H
hdM1jzMyvPJse5AyQ42L9w5SZA0vIA9t7Rn+i9LKxjpdMsY+zW7tggMhRTd6U9pY
kfRsOnDJJu1ypSBwbaEfZgiNtUkFwuzQRrfKLqjJeKCXw5cpad+f4xPPPC52UM5
RnJMTFe6UFlNmodzkyLr6pltMRmnLxs12uTXHR/9z8Ni/+mUWg8G/9aTwuJB1JO1
6Le8TE96yPlWqF//qSz8WJVWgTrfPGpQkwpzBWaV251LvqKzETel6/EY8zo/G3nN
ahl0W1aeBxbKm2VwtGwZM84bYwaH0cLPAQAvkFhv5zk+5pgC98rwiFhhXTefYA2P
OD9501UaTQTWkjrw6t2kzg6mQ7TF0Ee1i5EW+SxVKbd266MQgSZNhZXSFTgs8XA/
aNmXLx2DjpbQIjI5AzvE5YWeN+d51HDde4Z54sDp6GsqqYj136AHZIE6I1jxxi8U
p7J7Bkclzs/4FdY9cGfHTlhV7ugtAENq3w5whavoMgaQZIj0qi/PyLBSFrScCK15
3kfdaRRwdg4E43PqQDRW0e49oKWX6VxGzqV1sOhzo4Hq8GvMhvsjC9gJQK1hIeDY
otBZiHemOZQBq4rlJ6nVaWEPJkfebn8GB2xkogf3j+o16u4rv+djux87+QJ1h+cZ

vOIk/12eJaW3cxzBa/ckfph6TAPMlwEkcdxpLtF+dbNc7WHXK6NV8P5zPBTq58mC
iCpwhMnRUKY78wOdsAK5/oXl1bya5fFBSrVf71PPyADaw09puu5di9cJUy0GEcH9
dWWI29MnuhJ/+GPGlRt+X36CDc4UMuYHNqGI0Eqk6XuEUgZDwbsmpYUt0J2zBvu+
Rb4xAIb1a94wXzsAQ/4aVKaUsd6ofjychzccc6aU1vyQtqAOZPFP7S9z3dyN1LCA0
Uiat5crCQbVhJQNVmabkFBOWIF5kGIIErQmupnlukf8OFS+XGw8t24PPq4os2MnP
xtdZM0lmElwvFlcD2/thU8hfXUfYnT2qmObikJpXQE0e7BASAnYQj6u05eboEhfH
1bx1ZsZX+8bb504ah7QLfuqWAg9WTzdWooCpiCuYlAS/I7Ey2JW1tna3BZMCYMJi
SOD4yZG62wFp4QZFvv4WWKyG+NYdPj4XkHse7Yd7qTI5mxCr7bJtccBZi80JU19G
w+OvdypURyYXy1UYo1j55nFnEUX+IP3/pToBWpL7yRizP/Q98xEUjoOS1QV9rz8
ppg7XjBYZrns2JERC2L2xQUUfBgTtd28lNgCt02PwnF8F+KrS2w+kiJZI9CvN3ie
No/ufb4uOFLlJU+YWC2clKBb+5bxF1uVN2jhIfZRNxbzGVVifpTsIaz/qddsFtnI
8Y6yhImBpFCrdzt9GjsZjdNRFwTy60fJrXdkzwQgTwR8k4b8OF7AWYPxqgLRhRv
v2P26GOG2d7+BhGyZcaiz2y/eleVleG/rqfYqYHi+a3IDAa3Iq0hDg9IQ4x6/qh5L
viDAM70hN8kqGkg8//BaXvgETIIMyupmvi7nWpBVKozs/jGI90UCOSf8uJDDcbnP
XOnV47XI0XufAeIdxKa30hxw7b9UTqE6DAe0Vzc3qtWLscadPIxjHOoko+PGoUOe
A7w0vNwutU8beBDHkhz84Ni9hmSWOy9A+7J3XFMm7QxJJTmKoRe5bySvCy38god5
12WxVrlxuftoGPF8QYtLc5F7B+gx5i8Pv8eI/JJLMnGBdci9OUYkIe6IAw0zMXjz
0wPzIITHL815eJfE6cc+Gy+SwVosoa0RC43n0AzP4BWu4wRmJungQTSzMUM+6xb2k
ku3XkjdQLVY7qX7M7AbDr/7eK7oJWnixTyNY75zqObQaoyhgKJlD+6iwadBMVq8
SYpSY2EUfFSVM3+NeGVF/ANLoGcBHzYiokQy1HQZlTpB/2nYA3kBF19mZoUxN0fi
Ca8uDcGvB0MsHne8wvOMv9A4GCYYHSQxZ+SMtylTMtZ6qENDdRSz7JFC6jBaho3U
KM5+8iyAbX0h3PnMNUrtJ+9+nFHI+7Uiudkoel/ymgOZgJhrKkbSd6X9i0f2da/F
SeLx1jFtLx8GDkzWfI+8N/JOTsH0/0tI5gW4UUvWoRtF3XUMU6ZFpNkCK8GLUCqs
eCgzZdnCV0tYxvZnTQhZe9prONCe1bbRGcj/OeZRNKKH2CrjdLG811wFC47KfrMD
xRTM9wFxFsFDyr6VyhxoJpUEz2OjmxnStXyd3nofcVVR8kI9VxIqPbRTLvlzevRC
CMdeZPGMgvEPLXCWAKFTuqpTYwWBx+aHDGj8EPWoVKp/4DRwjwYMEyiErQjz+a6c
0Kg5lovNc0x3w5qx+7aU5hA8JF8YGj0+0j4HdNeFs0n5uAqSXI4IkaiMcik3F5I
pJRwI5VHLfm/UoeazisJ3IDq3TKAYpeh7lSJ6xotJkZnqlMBFzMA1vu/WMN8Ymye
1GUEFP1goRiukUOrfQDC1pfgYKXtvRsJRIFMPiaT/6kGDMA6OOVRjNOBO44OxuJJ
N2o71Q7+J6/Rig2Gck7bEVmmaZdj/lgrD7H2Hs/aUhFS5vQzdCnTiXBdcfUIyHM3
AsrOlzmwPgBup6FH4GW6oL64cFGmuSsCzkCwdXJKnt9AMq5h3efJVWhnRnldAYKo
bgkLdL4u21s9R802FQHqC9WahhGh7EF/fnVGE+yJkFI13jJUC7ZSU4W+QTLYR41e
ucYxmO+DmK9UDLOXyExJaSqohfaCba4nz+Dw2BFRSgV3JG3RcbsLsfcerXwQdyx1
R/u5ZRt3SThNNz/UIgkTZXTYMWZezQbHv6REvER0rwlDtmXpg0/rcPch6iGSKEi4
Wn365bCmBTYHd6mCOh8p2YycZoQBqGAXfSxz5q9OXJGIikrou7UfnSKTHqhubXz
PVMnWGbXuR5FrEYkR6sHQwpF4Hr9pbiqq4OZFXr0NvdC0fB7LL63x9XWV+TFXnPE
j9ycJeqxVQgB6fQ83nNfwb7WKCe4waoEARCZ2CNY14V3pePfZttMYwQDtHR7Ssko
VpjhqDqoQpMP3sdNFR7u7DqmwLkKhWArU1J0LynI72G2IutRxnOx4hWxiNizYntB
d9bjlUpCot7UYf6mDnadqFg6gQa69YiYuRR5JChc1P6LUSVTyNNMkCznkoPVOWGm
VQvaEPkWWZi2/YSmZqtBsuE2G2ggK6q0nRXC01GxjeNuoJkgaedceHrGFtnyfQBQ
gHG1j7L1HV840nwdJNS3nMhxceof7nQVsOyllcdHv7Flui5ZSxPzAJb6turW8ssy
xU8838uMVgqwnwVzj1Hz9mGguIeGX4rATS1tlvVR93GAebDWCEBiGg2hdJLfrvUF
Gru8B/HMtDc+HFwyDICgwVMrjixqb4QlOMZV8X8B2NdFG66U4KMG2KCMUeVU8ExX
sCMrf0/JEVC8uXZWUNXby7H1u4rMH257aYkhXwh/obKUx9DDqkwxW8QFjNeCQYq
+ACwiXXJlWOPg8CSXw5HqHdTLJHdtUXQ6qGuJMJB5VCDcn04SRv93e7wxnqYqPM
vQeKYt1gEx2SBn79jgkoZUCJ+GKqqdA2X01Ws+n/y139OSyckWHgEvHv+MzLjx5T
pAG71MwClyA5Tg1xiuYhliensL03XmszIm9qLTRD7tQ05RwC+fzpmBa6sU4eyQUe
ZnLupGijRq4IbhFWng18sDrS2dyVnib3tS3E8dnn9jTBDXxDnQrfqqlGnck+W7R0


```
n4c3EfHXenwQ1mkxudp5gefawftI8pa7VU9oVPdNHG2DbGtNfyrdcvKBjNV8k5Eq7
f2ScfXVavYXbDN0kFohBQZJCQNMEdrJRq6G1OoBmCu1joXpo48LWj/Wf4EM339nm
A0umfbUWwMMUHOtHDCdFwMUQ/pviN4J0u67f32f8WnK7FJGLqcKQSBmT710lp0wg
B1A2gBGUp3/OtsLsc5RZMSUyXYuqZ+qXjKkhEj8ApsB4sO8mEkho0KJRDqW0uu5o
yij7OfBY9kxe056y0xWee2Fw400SRscjAcuGkkiCzi8Beb9JriE5ddE9Hw9W5/Ai
Xyxn3C7Mv4ozpFzvKgW/bukNYIKdDZ2nWeqpnRoSyAbuHJ0FFdayEvx/XSSPdQ/t
g3V1bNrMbZMYr/QJkQqCvncusXK5OpFeOF/2jj+EnJrbubrOmTR+GzKAN88Qq67n
nMRrQVCOZ+3WiqlYkBY7nrVLfHW/AF8BDW+xqr6uNIO5u084yZRpStkE611JmZVY
MvTtm+Yb5trb/qUuzJbpgSRT40mlHynstP+vEEcM6ujVFSUEITFCQuaPKmZl/qHd
M+AqbdMRu6MLGBR1TX5rTVd6KIj2qDTmPbnV/6PK59T8Nv6Aekokdc5CtYgc4oKh
ftDRa60EjpLGiJgCQzT7khzTrHZMN9YxdtrTDBr4fHitqlr5RjU+Aymx+NL0CXmX
V+LiVvvQxHGpGiZEaV7onQ==
```

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] 7565 bytes
  (decrypts to)
  application/pkcs7-mime [smime.p7m] 4584 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
```

```
MIIVzAYJKoZIhvcNAQcDoIIIVvTCCFbkCAQAxggMQMIIBhAIBADBsmFUxDTALBgNV
BAoTBElfVEYxETAPBgNVBAsTCExBTvBTIFdHMTewLwYDVQQDEyhTYW1wbGUgTEFN
UFMgU1NBIElcnRpb2mljYXRpb24gQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIeh7j00
Boq0MA0GCSqGSIb3DQEBAAQIIBAC+Eq3peshJhf1JB/ataWrNRTuNhGtgwfe7q
0EmuJ93I3x04yobdlgfm+UQ8fBXZNobbjj57dkoxkbYEEtGKltv9PQrZ4Qw/e8UM
rgYA++xUC/h4dLTBBD+6U2KFinZFbVBJ7irGCZVB4ddzF2F9dMzZjMH9DOZIS4Yy
```

sB8Egd8ouTVQCLCfc7FB7i6f5qpffj3FibrPFQBrxFobqID08eoeQLv0oNkI4b78W
xdkG88IHfdWmjCr0+5Zj/1XdmMnuQfDaGV0r4FemW/gCj9UnQCF9Z6Yi3WQeCm9
xyEcMfUBWbBlpt5sBXqfV9JrdP6/5bQn53myy2B77XRrGmIzA04wggGEAgEAMGww
VTENMASGA1UEChMESUVURjERMA8GA1UECXMITEFNUFMgV0cxMTAvBgNVBAMTKFNh
bXBsZSBMQU1QUyBSU0EgQ2VydGlmaWNhdGlvb1BBdXRob3JpdHkCEzB8R0APhiY6
HGLS64MvlsDXhpQwDQYJKoZIhvcNAQEBBQAEggEAdfxEVSy21BQVbKsyGRIhEI8f
oJYQGAob33mMh9x08UAKGVuquskYMwZs2ZzPcFIPCBQquiecjXN5wxq1MWLaiRW
Uxg4tqnwezPRnGQD9GsJwm1V/n2JMhbMx/iXXYfvZ3f3mEwsUzfFKPkxmO/G3j6q9
zXW3J5c0ipriUdJHt26EF1lENbXUWSp32pwEjOXxp/nCHy4SphqyoHLgHTxQ9oTj
sJU9nMm2Td10Z+WtHuRMxLbFjFF41URAZ35aWJ5Iw+v0eBxQX1GxuNZ4CrmQsKrd
CE2hcL2vXsRECB4A+6596OGIon3R/BLQeLC8DPVdUFHvx1/N2REyW+hENgsY6jCC
Ep4GCSqGS1b3DQEHATAdbglghkgBZQMEAAIEEFVaSaXfBQR5J2dsolR7Q+6AghJw
pYrgLyq6XYnx9XGh8iNTbg26fpAqJ/xIjvD6nilunwHyMUMY08cEIIE7V6BClAKi
kBFwqycgmbgBhr0X92qKyLS/izZ5+QdJeqracwwwepbT33PZXDBY5q7hhIF90Bc/
GMId1bJGopFeyjjrczBpN30biBPas7kzVFfn/wGrRfTi2Mo7crR0v9znT0ixht33F
KB49E/QZFdtyp1Dz+2zIm+1WkYX9nsW+fyLcUo40HpywofaDHWpx1MrxUs2QF91
8CR68OCF/GSnUVJcySVKp1xXrEeoN7i3TX4+0BWSWlIVYp5g0vrJ1eD3vadQTJuB
gDk2Myz7mtAdswdqd/wpPn28tBGM5+GLwIm1TORyf2EqscrlptWJXsdDlsKf/u7u
Wgrz+Z6GFqiSVMB35k2o003E/hEu74u0H3zErccItnxq6ElhDmVBRgYO+3Nwmh7/
Cr7nfoSykkpZPona4ULy+309VrRI31cCheD+EIB+HRO58Ez22Axm1VINMq9ANC67
Gl/xTtJBiqb/PvT/mQiAgOvD95GtK152R361w630qTBcIV36ZN+zCC82AQIDKbOK
PmZm9nMyACvQ8oogodcctHvFcQj87+eqJRmmFU2/CMSreSXmuJzxXH6HGK+kanBN
1DAV8efJXHsD8+2V2on05j6WN7inIUdZeZGvEzrDeGu8mTdGQwCEoyH0PVRYYVKF
3luCk4eB50Rdze4z+FnoMBNjwykdbVySdLznKMX2jalrcbrP510I6dsUnG9T83xk
uvcNrf1RfNq63Iy/GsHMNnLntgSpLc4N+hz67uU6G2y2+dLiKA3ODBomVXDxqPvq
ozR2Lcaqb5Wi/T8YLB4AY6BUAAfTWrhT0FWeccWYe80+10PnPHbqKTejSDpkz26M
3wB2MtL0sCxSDJouFWohnAlwFbZ7heFdcuRXSk1k1rSbpEFH0sWiNCdfzmoSvVOa
yXOMR9fWCaYXXZJyZOZwTZ7KjFswa7LbZNJxPpV4RDNEOBGrGkxfGe8N2B0AuN46
7YkKLtQRy2f2BhZGKrKnGgNLPWvccBouqR0tvA09X7QUFAqfDJKJKAq2jPFfnjLI
S/DA1pUuMzGd1AAJPXkRXC5MvbepTLxPcwvo8Ucw+zzzmSTYQuyOyUwXqNyj+DpS
cdqSt5QYxQiMMBQ7QCA95OIGmwXXXX+PvpS6ShMGtidC3Q/h7M+oT+SCoCZqOVNW
ttRmECtUzx3t5IVRPE1shpsdNE7SyUS0KHUnguliMTWhaAWWfM03vppFHRMQ1PaJ
KPybWs+V+Pa0gOzstjTqKr0u5L0wX1CRtH+add6GnjZuzaJ+POZtC1CIGlHl8Rji
dCO472JhGSeEt/T5ugKEQ3gvVE38GdduYyDNL0u5Ef6vBRbY9mJCITfI134szZac
axoN5PKF2Gd5XM7kyU+DeHntXpvxfwDF/39ScoZ1FowlqHbxRcFEH+3YyhFfPvar
JELk7bmFE1CLce93CAmuVdxLjwMgXLD0FD2p7o03dgEcoMfuQBtk9LqainD0/b8U
N2FJbAyLGxKn+Dkorl+TF16ydeJIGQnIv6kvJYi2v6QdblmSCoY7rF3IuA72aO8A
dpenqKsPLgp5ltXAbND8d2gXVaLxyMOSJSSgKo0vZYKQYcPh5FeaIMtWhbpo6ci
ht3Wb3jFcXT7REyUTVibXcmwp5BfGF5HjtdsAUhuyZUWUfOdHWyirtORHoFlmokS
UMLRmaDa0CcoiJPXpQmvUivQxM+rPQUEHSTShwnx7hmjOUTUXxiaakiw0QNA9ZKL
+GMSQBAFiz+20NO4OGUtl86+ypHLQppCgOlYrbxcLKRvIIs4+VsaIvgCw5DGdStR
9jftX9HpMUXcIQvUIxZn+pWNMeTD9f1SchGQzYk1DcbLf+YCYM0GXnnVr1Xeu4Me
VlhyXuHUZdsghw4BpyRk12gvO4UQCvwrw/jLr3TO+msHMj8K7GzagwqzWBNJ3EV
UeuYufPYxk3nwsS9csq0WnH8i1YIBa63pdYH4VuRGWm8Y7vbI5/I0HTb603jYVB0
8Iwn+GdBK/UJe5scdKBgBPc/cg6M043WgzdQp0jYpRZbehyB/KVU/W9x8df3DkXa
DM53Ub6Is1CK8/eSrjkmjnytkf+JuVUxYB52yoDzg8JbxiBKwn7NcNN79k8hW3I
KZTSRImiDH4s59fzHmCMZYN4TrZ7aMC/jqMK1PfJjZRraM3aeRC4DvvHh0fD/bcB
rWzmZfFZeTjsKTYKhh4ehbgMKbBU0wMQYyG8HZ8XILgHNhGHZ3UqiEKGLY3tOE+P

9/2DFIerkICH5xybrAxcvDJeyMF8sWVxW6ZJ8Ka70OUMEmCfdcum625czluIs9u3
MyD6VCyef/j0TpqD+kn40IqQnfzL0QzrHA9Vp6k/pg3NpMhFc5ftr4QsBgyCDA5n
vKcsC5p2gi7/I9BgEw4aVu98QC05dtULTssn jxZZHXhggg40FEw0gv254T7r6yJz
gYa/tRiRzM4I1VILMvTdbC9leqBR2QSEBfjpBoPWJTXNcQfw+6lSdQCXC9LyIhwR
+8BNMu883XsyEW2nHu5pELYUuFFIG1LLAPL9h13BKbOg/Q0tvhHnjRZvujBGLLSK
rSbq9JZX9cT+r6R4kab92kbII2bEuBAOei7rNge0kLba5jTmsLiOSI38Vsr5AZok
pIbQl8SbqDennApKiIL9BuFCUfHG+uoM5hpg9B7ldmDyCAiFSazm/YsNwHqcHpEs
e183W3ds1EMf+VJ6St/mq5GJfAKH+vfp3qXaNqJ3WoaII+VAK0VJ42gxXgt dzo jS
pNX505etbGndrzjGEUSrcfXKhUduDklpB0wtAPewEXQFJj5pIZCO/KX2B6Xxe7xn
xGk3b3zY6FQfIMVX4VIYDA+eaTu2AvEJ+1HNAZNJmP0ly59VBUIf0vfARKnuh7fP
mQVBAguXLkzbZomaCs/WEYLFIN7dKw3gJw5nYyKNRjRUgW5PSRjsv4UVsCUIw2EJ
bWiJ6n2B0LM97iaDbMTlHUBb5O1HDNn0o5qgd0lqto+2BCsWJpqvCSNUXPW/kXGr
Suq0yAcjmaJG0vZSuN3/uUMdd7f8z+g/kzOw5tGz4m/Y3rx+WdM2IvyRuW5pVNWd
4NXI7onnvatoU9lPkXzaDpUTUj0bI3MOiGWESId8pyCDIAKjhud80in/kQsAoU9q
E8RFWlYopzYsXXG3bVWYVVG0qk8mew/5dYATHg5LnTNuQw1SGb61TSpwhNjh4uu0
0coedJlGD7+IDcwHZ/lOqIGXi1W0L02y+jT3GGUVQ6gM6b+JmTHgz9WREh/ewegV
Zz9jHHEoJ5XSGW4EcBE5UB5R4tg76KqZJTfrDOKifiLQ+bI/u4jPt6P2TpGd8rPU
2bHldJaImjko/zcMfq4hTxKiL8qxnPAjbmEwtCtlcO9ZOaloJM0r5CO1TnFo09uq
FaP0RHz3949Pue+6Khf4My25iOdUor5qA7kxPsV6H0zZegWtLhWQ5bQqedzp5/eM
LUZVgqQV4EqcWW6nVSHq14h1572C3wZey8LSkMhFPmNo+cdYCucDeA7Z+If2jvm
KotQrWLQ6GNUTV+uDM6y7YvVO+DK4C8mVvi4Kk85/7yQsLV1iDA9JWtH2D3+JTRv
MZ1E5RduDm/XBr18LGBp08kBGLl5sUg1Wf9bAb7VwoEgJf6YGPXxngTnQhpmSF3
05txmNA7C1079SfGJGaSlxLuJrTaXuZGHYEK3mWv7x1pRtUkzMam7nu9Fk0WEBsi
4TWBRNwFDLt+eRDhlcEZ3BXYXaYmd5cXZUYdGaQuwbBkd47MEwL/XEPRouqpwFG
IH3c2ZkmrugKkNgaKKJb8A196iXGBBz7JcofKzud8PK+3tWodXYM0y/KXNM3vime
QHrX3fid0vKpxrYJMgbcCkFaXWvGM3F6IksWK9R2LuPOS60MZ/IzPweiuQqMLgYK
iLqf4Xkpc+mI+9iFwbfVOg8b+0+bI7fBfrCFsGliDS5xeBsmB86h+fn+053BCZeZ
S6ltkJUml1KQxzSKvYfdvY3Atm/MYVQK6/bIVZg+BniwM8VEFY26BWz0lsxzK0UY
FbtfWN9vjObdqOtiSoTMFIjcGC1C7z1miluiExj0saHwbTKFuyHduJ+VRLm3+uto
ou9iSAahnyum4gnxQ7IIcceBe+/mp5SbG5G3EZwVQRkUHD/P/6fCJ2U2Qs12lmmI
HClgZBzFMe8HeDW4K1tTnk4YMOyTbn3qMPq0Qii8a6yjdXTYfoCXWzVWF80VmOkz
1wVLaNm3GRujlWDRURCzwbWDUV9/dm++kWwquY23VagcWgaTKLWTW4vuAq0rf1KJ
EYONqKKZHxBRyhg1+M6KQqGAGg3LZk6MqiMzABR8V6jmnLLbw1AIUCcaAGOynlZI
WpcaMisOUT3C3v2ChiEQtQrfWX9v1OY/ScwND6KDieQMqkzMrPUZU8we/Mms/ouG
tid4hMx3QZf2BTkCpEGDt9R5pkWYg1ZzL+7vsDouLCuDanUOsSfu5w+Qgp9aRB98
08g+RbMbBoi0lJAK7Bbj6pqXD/IXJ3PubuED+Q8TSG4YexDXGX1qvBPvxUsSlS1p
5XDmwx/ULLKV5UdRUlRtqmDjALIGAEww0awhTvvxaCHaRynxql/9fJYFQcZ8Jv1Q
j0Zjaqw8BS9rw1z+ZQDwYQbko0pBYWc/vKLib0YERvqph84iHWtvXfydd0poJvSa
KESUEko1Djp9ia+iEpUrwOQ4bU4cNXpqAlQtHy3ZntHWYdkWRRH8o5Fj9sYO+sDx
mQvwACvKaUb+o42n5AjjfgI4fYoFBtHJj8TKDVxfPGJfajp/Nb+/xmyor8jbnOQZc
ofBI9oZnZgGz8FdxG/eglZiUXHqGvs/fx7p2qjdcz5CMXbSzhvpiuMhDPGLDfDpN
6T/DEY50jN7dTthOhjYdPGYHZeH2o9dE6W60PlvREtOdHyJ0RZ0vwtWUzJEGadcb
HA1e6w72My04BwTL6SStDSfVh1UU7PSjqSA0mSc/8M/WjQJid2poodyKEOVijSID
P1a5dKKJPo9WZtRPQUSSUtOYodTTSYDYNhYQ3qVGgIIEZBgI9X1LVXsvnTSXCTv
1uUPGg/P7wNmfg0GEpJDPqudqE3j2s8JRNWsuqiE44QKle/3JlHewX0m7hNyoVVM
qZdJ3nkuA/7f68PW7+ctHoojLOxD41VLt+UjWgU2heqxsP4DlhTuSYprw14Mg/Fq
PyWkyh9qftKi8WKAw8VcfSj9jRQgk+YHtt38DV9mBrPd2h7QUNKPa16Gw489CTz
hKP6MrCVLwLveTAJvx0YCH8k+yq6bCB6zURI4L1qOiu7VskLyB54/TEcLDMTRmEQ

```

6Nsueo1eldOv6SyXILaQlJAbEZAXy0ZHGOy8YNbDm4y0caEhzr7Z2YmXrfEOo5Cb
Qmk/qtJb2cCNBIlyt/8DhseAE3ocYSDDGHFyyb1UneK+zmWdQIzKEch+ho0or2BG
X9B9kJ0dsk/1en/Ln4lPQoWCGshu23ftb7btgKzriChzNQYFdq2Lr/1VMwScD56K
8RuUopGOJ3mwBDqJJweqYZj6h4NtdY1LcOy1+f0lObhLzcGQZ80vec6Uz02RNKCR
j8l6g+bUQuQSrJAecEnRy92vzQfnKngKknC3HC66S3kVQpf7ssyo/cS+hnj/VtML
3tq6Sw4fdd+mWlEKk9L4CisIFbV/P7Q+6HyreiOnail84ltgWAlAEKU47SIchIoe
gT5Gak9VyqGOhuyJVSfEuyphI7EUIjXFK6MMz35oWkwT5tcroUT4zYfH/p3W08Og
rQjBqIJfvVNTjbsXUFebiMrRNAPXSuN9knQkqHgNdH/0T6HsPGEFxxvEFu38D+Qby
3WexSSUsUnH989T49sYCh6GSrk4h6hRl8Bhh7+UYg0alXi1SZzMMEMab7AGvuQI0
fKC/wXyhekq/1ZOtuEkDaTWvLedbHgaSKc/8WUItNLsefrR8iMgpTTQDsa0r+cEd
Zf1Nfv4eBMuSYAJ6fT5LmhDS5LlAbkz/1tBfYkkout37Uppu73u2tnx2lgtxHXaH
/4N94VfakQa5/J5s/yjx3YHb010Z1yEzgz3+GslWodD5HY4PpX8oJCKEKChqr8E5
D/d4XnQqvVepA/WLTXnk8j9ZRTkpSThqo1/v379XqFn6IbQlQyg57EuhSWJzFlbr
92AByhFG3CgJcjhWGBp49vlu0axM9Ahx94N/J0H4HRECCagNDsIOhfufB4/8OyOc
BrK7Ai7RR8LDtknZ8oxMdg==

```

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] 7345 bytes
  (decrypts to)
  application/pkcs7-mime [smime.p7m] 4432 bytes
    (unwraps to)
    message/rfc822 675 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

```

```

MIIVLAYJKoZIhvcNAQcDoIIVHTCCFRkCAQAxxgQMqMIIBhAIBADBBSMFUxDTALBgNV
BAoTBELFVEYxETAPBgNVBAsTCExBTBVTIFdHMTEwLWYDVQQDEYhTYWlwGUGuTEFN
UFMgU1NBIElcnRpb24gQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIEh7j00
Boq0MA0GCSqGSIb3DQEBAQUABIIBAHMG7sJRDCJDMqgvQrFh4sk9MkaJJY7q6B3r

```

hY87n3jM6UYk/ZaBi9uzcBlpDAF0hJkFLmo+PRUbFLUrmeYfQI6OuvVELpwIDWMP
cMtfz1XgKAO6fh/On6aoVhpfv9EmaG1rCU5ezDPPbaXW8caNi2/yvL0ustpqKOTj
cOLgMK45tPcHeIaSD+8A4P0uf/GLzEFhDPdJrt3mVq76UbAoIGasA/sDhhg0xygq
ZH3IPQoYShFEUmsK+RC9Sc9dmXtVYPByCEsPdhtieJyJw695dde8x17ZeWS+JZai
QK8pXZUdRL8El82+001HTXZYybfF05sFmJHQZ3L1ftF2Dqs800cwggGEAgEAMGww
VTENMASGA1UEChMESUVURjERMA8GA1UECXMITEFNUFMgV0cxMTAvBgNVBAMTKFNh
bXBsZSBMQU1QUyBSU0EgQ2VydG1maWNhdG1vbiBBdXRob3JpdHkCEzB8R0APhiY6
HGLS64Mv1sDXhpQwDQYJKoZIhvcNAQEBBQAEggEALvjSc3y8/+aA+Mk2+8tupO51
fsr8cR8BV0+aR/CYDXaeAFg6CPk12PnLcpFRZDdqitxfe7SpMgk0oT3IsBxvuOsr
0QckRlRLOwlv43Y9jJfMc7VINrB7bJ/cPHHgB07tPtB69/Qf252gsUs3UbWko8JU
JXBkymfUAe5+x8/gGQYNJdvNC+v9cmnwTORFF/IJ/WcGsyHPHxguR+JZqIJkSI8T
xjawV40qcahz5G/O3vLI8kxW96lSSmVE9WiuPafsmBp1KZN/6ilgaUOPFcsh1jln
fdnk3fToayCGwOaQvh/UYv1GTA06Rtnmz44YLZiGbvLFLGLvCXffwL1JLdl25DCC
Ef4GCSqGS1b3DQEHATAAdBg1ghkgBZQMEAAIEEJo6kOdMHnCo9aCxbG8k8qSAGhHQ
oifxeGRuuDaxdcCKEyNhsAqOP92jEteuI38u48FqaDfBniUs9wmW/EiEaTmXWvdB
f7df3XeOK8yGqYR4pcXSYSK8iGfLezceiIwABbXRS8eLcNT9Npc5MPopD/h4q1Vq
+Lliuv8P00Ih561cmKglrAmTebH1bnyjYlW6GH82/dscgRu4mihqvJTYQC3uaLY
H0dJnqyYV124/K0QyAPKCM3gR9gniHVlejlQKIVwOT649mTdZ6FVeMk9eaLQtKf3
mkx0trUzXduJnBj4cASKSovC8yySEuGwWu4kROf1g650ledfeU4SC91wPwzvHPD/
lk4R/gUA0UAo1Ij7GaNDZ4CqpZqPYOG2wJvCQjFK7MU9TgoPsSRXhlmZPCam4ecK
gdybUd0A4UTQ90lZiCrS0pDyKQyatn0u04SfKU/b97P9VwNageENERZTERoUx1T8
Vq9yBTKZIWoe/2wsVvVJaR2+SXunrla9HDwpHqDtZHHr6i9Ttnp08KMOCWLzbb+
lVrxswrexUtGPCJR162TBchhyOldIyz8eWmiUvHhLUKFnSUGh81MdQKItc0qJ9g3
iu3tSd05AEHxNf+2hKrrTZzWCClatSvyfRbW6/OmlIzh9+JUyJLcCywJbxQUuWRA
5pc3bHrd6/Ffldqgw1dbH9x0Q/r0lbKrWK98B+7/KIAfvy/XTW3NAJNdlzpzyl4
Ko4ujuBiRjZ0xRKIPSMOH4w76YeJowDi405Ea/F44hlTop5N/LYNVkpIVnGYrEHD
7s05/cjQTX+A98PpoFVKHxphV+jRiWdz7uUYlW6ClyrC7/H7VvkzdtPk07EyY+zXs
uThq5Js+uwgsbnqna613vTEF0p8f8k5fLi+HSgI/TYz/UtW7JknTl6k7TvLXuQWT
UWmrWrD/UKADkkehGkZHpMZe+RaImwRd/x10M9+ZBovlbf1DigfhrVimwTppKE2k
/S+GSXDS5r5ESN7OgIZv8swYtk6N18yoFiJBD+wwWU4u6JNL15RlJZZbki00Hhse
4Of2qogvmNfTpHbAU5DL4UWdehoK1fmPu4KaSpL2sRnTpqzyZEdAwG1JIOB0YAqE
ztszmcxi1s9KWQ/XdNJBG2QHvSMf4QTCuY2e+335Y9/ZC5WBphpAazRp9xfXc3de
Pl93N6ydfn09wT5k7TMeLOJrqPa84H06oRAYXqFYwiOVWRvyfrsInUv6AJfhrJBN
dA3ebIVCwrfG1w8OHerzDBo5yPclASLrmuPjaQ42CDrHqzfnMw9tHq5ZaJoCGF60
4mzqu9/99upVaaToFRsA40lUpRN2QoOYUBOl3Ck34mWGWg8vf6akYADylm0SrpRO
ym+/8WeERonQcc3YqrmVjzM/yh4RLp189oWWHIIHAaplYyuwCj+kjiOq2HNhvyuq
9acwfjQ7mKBfK1i7PAYdvWb9dt95VnY5LF+MvevJOdf1lEt6rISePs+AhoQCA1u1
B92MpDynfPUFOeRMx3do/zhVmY64qN7r1V0XxuuZXUW3WoopjdUzTmHycYBn7sM4
3U0d02yJgy+IqiTOusRaQGC3/IJiZmXoTL94wBsOB1++cP59GPYvm6qgm7i09fUW
VO4ik8lTEs1WegTez1Lr96dwkPv6mfFJQIDlxVoZ4LVRf3FbQa9cZS7wxSe6hgpI
0Y6YB/s21v13GpCX8RtHEKEk4Zc/9CrpUv+1/R3QXRvYOnQaWxc96w0/lVkoXcd
SRrlglhl6yY0QYvOmTbusUdC0QtrcQBRVcVeqqbfLhip9Nxe8vabPkoGQro+15sO
xk08Yw1Pt6oa0hh5NjqZaBpMhD0xAqHT0826xj7R5wp49KKtR90K4wuUy0OAWpFY
NdihipP1jGuCio13PPc+Vah0+ACMMdvEWjYk2qEy2TRbWooNB9szzUoQ7P0kKJx
LfMS07ecJ6sSsjcprskZgOsJQXtIcAgRMnxFFaCfeg2zjW1I5HC+jbiNtqda0aQQ
L0RZ1a3KWIIPNBql8u+cXXjfaBy4HQh1XmQEnStkLrx1JuAI1wxhXWYdsrjJ2xEW
hQBjBwcnTAc5i/vU8H+oI1Pnc32DF8qfa51wluLdoYl37PUMlerpXq+mPvL9cX/l
w2zd7Nc+UUezqOYPrBbwnrWOvG1msrjBPqKnJGHZJhlZOFLdmLa6inlsQBpX6kXb

K+8mpshqf472HOfje8/hrdLnOe9Qxdf8eNy10DHs2MzxkYRktNJFIEK6JHo62NSG
/aM1VJbKudK1V7FFd/hrOAVg+uLbrsaFBdI6EE868qQpDThpd3WnyX8HztTkm7Up
zpPuJeRarCgEk4RPLl3erYa7d+8lpD0hzZ0lQkEALbSlCV0uTW3RSd60fNp4gvXu
GCzrJ/gsevJrJNggz3QojIXW9RFaU1Wwy80yIWdTguCswGBjMdUBRghKQlM6LlHU
qqXGdRL742XbYU76RVNlTnjUvAFvumey5cylAck7Lm68hV8rhTBsWMAJCP6VYhY9
i2AiW440gsNOWu/uCLBNpxPlfA5UFYNx3fo5XriyTPumhkhkwsaF1N/jnWeXm8eUz
/ylnM5K6sD0gOX0ThLWVG90IC+qbMPNu5dOpCznI9DIup6dIhx8L2j+JoeqdsCBY
6Xt6KE8s1lLzAkYFFe5A57q1Tq/z1s/p/6TlhmRP/2IC+2sSX9EBqXGDD98gy66h
rBapI4n5N6Rnt1N5fnWJPVSnvFYIDQ145EmqPd/gUmMBF/AalgYLEdxc3xKOT+Gd
G0BcwQJdvmUp8rPGWgP5oy/qNIAdB3dnlfAdeOeeeeiGhaSpcwVhEaWofYS+IXUM
kGWNDccjDIZHvGyLNYSiHyAP6vOxZWzj2EWWUEAhtGodCQ74qm6JxRMGyVuBvyFD
MtZxMQE/AU/bPQmNBNCkN69NXyYW9Uk7p//Ef0EvZG4WYgQvaZlu4E/P8xOL6au0
pDcB5UWRoqkyU7jguMb7f167iCgkRTTfSLULD+lJv/4zf1Fv4F6cQhv+NaEAF48l
fCUFjEMtGLCP99xxnu3M6CdiabZNCyueGVkhzL/fq1JpVlgKRFEDFU/wfTe8D4QT
9tranwYyAVj3gd2f0ijr1Q5/9Ch0s83/X2CpSk8fHF0z3oBS7Gfyz45BugIhDqm1
NkX8J2vKlCBOx2Xo/3waf/Wf3aJOEFXKR9fc+TSO6DrSS6XGBSQXn95SsWrzuA9I
RuemiW8+wYbygIW4auucs+V60BRwG0wxAzn+0lX7zac+WHjerZui+E/7ehmFc8NP
ZW/FVftCYi6oc26dysKTzhpOUmh0WX4TvFHEX4KCL9QXTC/YaljrZTBFF+OtsJOi
oRDk2/yjrGU67Q1zK5escKJg0YdorZjMkfb0nNdjNOeJlflNL5eB8em/LEpaF+vK
aCWL8tVvuq8ggUZ6PHQNKqeIssJoSxrmCFSP0DEtjk2ZDGsHaHOJ8KUBLR+wiSs
g+NRIG3Uvch6kARJqN3AgWlBySV42A+C6x+BPUEbcwDv3qz0DLmfNob4WArd+jyk
42Gnk9VL/bbddnhCyzYyHCr1D0XMIzewqzFR9ppDbgCLMxb7Q7a+8Umlkddd/aC5
wFUAVB88JT1gJ+NqxHZs4BISTd91ElFslmx9yXD/dEUPGfqyl5tbTrbGQpfv393U
Q6L6cwZS1lRg+b7E777ZSuOWxJL92ATouJmzCYLjafIOjBN9BpGIymvi2QvUYgB9
9Bia2X/SRc1fc00VRK77c1GtW6Nj9L37eiXMKseQEWY3i94vY2Z61ytosB2BccSO
R0QRJSWzXCXTJ6btCnFhZUuGhrnG6ibGKYmrTJTzNcrN4yJ/eByDqOc0YBUR10S2
uGMqxwB0adJ9ci+r76ZLzdo7OvTIb+WGbOP3IIYeSjIsymkc+Shb04mAEcodrYX0
n3wYsjrhrYf4WIDxQhWJRUDbpty2LG14OGUOTPOQPDaKwnGIiBUiT554NJMvv6WW
KLEBxtJlJQ8LhN/jo9ZwxwI/FZ68pd0h4r5MhlatVxJHbLmnWmdd0L2b8w9UyBwM
ts/zY9bdjFndBgU3zmDsJkZgZdgGtzL9KbUwHDInvCKtODM+X7QQKHu482dRb/vo
uIkQDy6meuxdj8e/xzdSua2aSQhYaRXuZlE7uq4Eyn3OcJB/re3ORlsgKh5k+7hm
kSibtsFYymWvBzh/Mata98kYHs6BF+Rgx/FdA8989koFmkAb/B41NFKuTuS0DmK1
2SDKgHb6rmn+cftv1MOzfgJdnGobqa3NCEYnICWitPw6NAbql1vRWdKj2A91oMO9
YU1P/ZNox2vKWdH6rkgPfkJYVwdtVEwu1Nhaobu6p2c71RyCzJSYuAMshOyLXxgE
1mCup6EU6+IqLryA4WkD2IdpYbVP/tOdFLKY1fBcGJtSVdgJCXiC/krDLdKhrEkM
RCiIcf6ghG1En0Jpk0xU3OWMh+kD01MO2IJuwk4TlT0kBRqZAtYWQQYQv0xecZ/K
DvOXZNUQQSzxFSpnGo7wOLOh9gB5GOIbdqtAShYsCXbU3fuXl8/6Lojv+f0YBBN
capJh5oWBmJAmowJU3pL1JyABd5+R//cj1hQFApBKrs+cbP6ZO2cDabDWavBPPQ/
QQCPjbMENRsGrU5bdWRoG13qP8+FVka+aNHF0xtn+mc18scGhwfem6/hFgKyBCAZh
H7RmYuWoRZP73XPLYAM3sfwb1hLZSFNhbKHs00/Fg2b5MkFy3DwtMmqH+2vDLBv
6CJ8s0VTULjSk9b+ddvk6rgUy+Nce413s8Gq1ZfUUDV/AfYeovwoUhCIkKYj2DFS
jBB6Zvoo8Z7zQppqNOHIiz+02zoYKtLconQWBghVhn/A5ytYh05JZ72725Ajitae/
9iRvigf0u4hQrowNuR+5t6bjA+5nfpKimd/3G6JdvY+QcN3BizQ39ZyUrUr3pmY5
KkyHTZo1sazk9ZKQY8LU1/nM2IraTuFzLhP6Mttj8DR+zXDjoPX5xxsr9VWVlctG
Y1NPHo1SYvqScQ7K3LVVsiqAzbr7SHOABDF8ZtfwVqIDDmk7cubaTlUEdGA/tXTu
iQMYNv8iJ4MmE0tte0sRrPKKbnEP1f+UiSi2LDEYPuvXooGoroNFHzqPUX+6BswB
8GSEpsQDPzSJlYTugYrX+2PlM75c89dhfuidAHdubHMqurOUaWtTKTl57rd9en4e
HF0ZHQPxBgGQYQ7fT51WsXBZjxhWjHM4uDmc3WiST9DQX+blihoOGx3moRbbAR4

```

UsUJ8lopNmbY+Pf5XGvp92PtXzIBJyJlWfp0nCX3g4LhuwHpi5JOGu2nfKD2LZR9
l9OehrIncV0oF5rcJWwKnRZbTBjgozaxKwkUUfp/qEAteGYxEeAJC0wy4ZD3N2cS
r3I2871gQAni/LsF8CEAPaXE6swdSsfC0GWTi5W+jnDh2oeAWeUOqb10+vwLikC+
Xm4VabpnHPZPiozLRL6TaVEqvmBpvUXgZffUIXpXHsWbVpJuPsIzMlmgeKEdwUvD
Efcmdns0p3V5B4ZaXLfR6aHdtrDT+B8eNb1bB2wOP/IA7Up4NzVf9BtEzq2JKj18
mtSbNmSuhSGqYP3fKWV4inAgRQiDDw3bnazMh/mI17qMLa251zP9IJ5RNDRRWCjf
+mljnLpyYHb5RyZ4nqD4+w59YM9Q/v72C2cyL6WygYE4JVXIWdnrHPSTkjBBjoxD
P1WbthMP6DJcM5v9t8Rv8Mc8bPiUrKzMDCbXNcPJm1HDCnYrWXYqOvUpKvWn6zt
Q39rPppCdrHkNzFS20MsvWiw9KsWg2rb/ph+qh418ac8VdyXNcETVgkLeYHnue61
Rbb04HvCvu3bBNjy8D6yRlFVIVxH3Zy7+iz3fJ70VwlqqpmlnMsidx3vlykAeK1t
uo42n/3t82Dx/5s3p9rZnhWXUd00etjL88GpyzvdwtkYy3Nj/8afvB62iUwZ1fR5
rcnk1WkphSq9HL6brXQsS3lODDHsy8xIJlu5RrGD2MOIOy/rbMxNT5WnGoZ6j/RJ
Spn1f944h2LkyVFFNgIlq1W6MLfTNBrZZ6kMpJ8X39iL5KmkRQlme1rgJTtM4heK

```

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] 7305 bytes
  (decrypts to)
  application/pkcs7-mime [smime.p7m] 4402 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

```

```

MIIVDAYJKoZIhvcNAQcDoIIU/TCCFPkCAQAxggMQMIIBhAIBADBsmFUxDtALBgNV
BAoTBElfVEYxETAPBgNVBAsTCExBTvBTIFdHMTEwLWYDVQQDEyhTYW1wbGUgTEFN
UFMgU1NBIElcnRpb24gQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIeh7j00
Boq0MA0GCSqGSIb3DQEBAQUABIIBAFemxt6IIoOR5Kq2Jiucu85qezrNEQcYm6sV
Cuo2f+/3Qcmr85ho7PNGXSmj0LkmkvIAh4RYf2fH6jqYSYgsxQjT3jOcx70hhTms
zQV8e/UJvWRvXqHhPbtdDFketPi2CA++Y8zqvbl3L/dBeL+ltiQqcQprqy9RY5pH
FibcQ5OkxpIzBZQUL5NrjwRf16gujq+nGVRhphjwJWsCX+ypt6ZrrBPtje3Iudw6

```

/0MkMj2lJPEkgWvFEFNL/FkcNRzHlH3dQxqjaf28Jp7eY/3tF4NVHcirE9DSc6hV
7v5zVlVEtthdFE9shnbPxf+Sbww+M3ZTV0xJwGNwPwhM7ehf8wMwggGEAgEAMGww
VTENMASGA1UEChMESUVURjERMA8GA1UECXMITEFNUFMgV0cxMTAvBgNVBAMTKFNh
bXBsZSBMQU1UQUYBSU0EgQ2VydGlmawNhdGlvbiBBdXR0b3JpdHkCEzB8R0APhiY6
HGLS64MvlsDXhpQwDQYJKoZIhvcNAQEBBQAEggEAEZ8MBSyH2Tp59sokhPP1DnTLh
iblpxfhKGR1N86t0QjQcmsND8MhB4aM7BtgsymR3IcdKrchClmkt6ATp9anhFwz
7U93WrdRIUcSqLnwoCU5P6lGpM+w6XYJqWjpU2Yd76iYLPOYBeAftMbxdrOEwSch
KZH2jyGohfZXtA8jwGb3rV4sQ4EyZum5yfm0i8cOK7FPSPK/7pqtP797I9IBTOL
YdssDTrrNMDRBKZ8AXRO/UZFGyWAcXlSGSlwAQ4Ilg87lgUblYdKihC4VhH2Qn0m
YZG37Til6fmiZqAUfYJZp5nuJW8sUMzgrjzv8vu05u66W7LoEhCQQYTRSrxFYTCC
Ed4GCSqGSib3DQEHATAdbglghkgBZQMEAAIEEafIC8XIvnLoAcDMT8ITOq+AghGw
lTqzvMWiOchU/VM97L/YalUcMR5Gp9ca4N2T5OXhTDXkanfsUHQtikBHI9XXBP1h
Modt75Gunm5g+Jaj5K6hI2OtXZHGJFrh7MkZ6ttTNUeIHqtjacCA8j6Bunoa2qmT
MCCdHnTipVzH8tFfx8d5xcNETtOvuUjXwIpBMsehYbKBqpEG3qcS/Ke9chuwIEbwJ
vDwkagqw97Cyn+b+EWAj2hEKUGnS/YtztzsrhPwkhox3M+MG7eCJ577KUmIvrJcOZw
d7vku5E0Z075QiaFw40KaHVkqHsEEuAJ6FtQAOpwuHrTTZkMkTiZpETf40N4SPWu
uk0JIZpJvbxnZvktxbCDZV9FrGV/6TCpFgo0iAh28LWcjVkiFTS1kOtKqMFQxAu7
78W/dA6JSkli8OYPhevcyP8Ffyh+S1j+7cFirJPYKi/WS5oJn5vIZqzkJelySyf
vzGAiy84zd7AFevlZyHSJhYhvkpRa3Q9puIgF2DveqUvoFWuhhkg9SJ3QMgQVC6v
z8bPYqk+vG2btGT6FjlzZk/J0et2jpe+luFQ6qqVxQZQReUXaY3KZk3jSyub5M8U
RmIBw+l0eE7HXor+L/IMW2AV4TC45Crl6lYlboadPDyClJtsleWj7nlRkfrZTmAv
fgecHCgqAFIin76vB0uB7BcWXEJ1je8QBP9RHSadMFsxtO7QMwVXqMXLil4xaNPP
hUV3Z+YquW+rpMbb3WpF0lAzYtUbagK08eIzQmEa3nrpiX0so42imrrde3VgWiN
l/ZRyo9cPuCmmsdsJkxGfa2pdTecK52lE3Add8BI4qjF+W6ZhZnEmzkMiDuHGmoD
0OwvV+yV5S40HBhvGF1bBQR9xjKp2k5oIWLlSSbeUxpTw96sQ8Viu+MLgjubTjrl
bvWPHJzykokgM0VgZs0MwDQ6TNw3sSeI4wB/5btssUmjTwOinqjHbVjyityjM4WZ
5u7z29MaUNUY3I/rTBvN/RllEh/dBBBhlhCjbywizIQtoV146GRwPUGZeWymkNkt
xRqRxU+ecdzt3FZIDMjck4F1PqY0ylK06yevfI8mioUFU3HwNBpmkhfwgKx+K+WY
zoLatFBnvon9gemuVKvI/HblzOSqMXG30TQVzifza9Zhfeh9Hwz0cnknLCKYVYyq
NcQoTI6PyBZ44Rc5UmMr5o33OI0pffYHq0+QueAb15SskBOnCi6ELWBi6n38fVEB
Nh/7kpF0l9JqXnUwrs17jRMGp0gsM+sW9xaxbCkb8d6VOVS78gewysolaGe0AerO
qmQnNbfbzNH3IqxHGote/Y0husOkU5Kyglq6k3Aq7KCLtIlVLnyT+7rPmpf8jbrC
TlZmT3IaunHh3qS/c7xo0ybB1sFJzHdlrgwZ/FqMFGI65pynQ5zVGH37MspWs3L+
ZJ0wlnvA8Wle9cYGH41g/Ipz8Tl8hn4hhxP3XbQrPczDQ6i0cZn3Il84Iy0EyW/h
u0lLnQtzN9aes0ihuE8uL5H5DKF1G0L3zwE9eayxb9DXk+lwVLnCF06fGHgJFNt4
tbFIDW6y1ZLvsNT6FzWJUilD5i2lUIaMUDossMBzruTMGp8sTPqadxEtQRO8u/mU
ezKAKFr0DP86svfJfTmUK8mp9trqWpg5c6ftgN/7uG4fzq5DKAcPFbUspLH9J+Mw
WcbS3bojohXXNtpV4VgYbdjOqNFw5P2tKHRHSYFyHmu7eznQCrgklNNONJFQA9dr
3wHvLNshSt8ECsLarvnHUxyLCqn/i5Hy3ElzalmaliL7wYp3/7i+r1+qx39U6RCO
luHAZHWw2/IU5JkDkxjqRD0lkHgcfmwGdIBoKuHbcPxohwAlR7fD6ez0pnjW8RBo
AidbgUBlrWOOrLKFQMIabr7QDFrnmjLRQ6f19MJUtdsktb5E+r5odPTE/87yPS6w
wZxtM3xoFbIkMjzAjc3URxJRtDNVeeyKOCvnyxXO/QSS62Rs10/gOGmrpdiaA0yO
F3+n0jCBMkhtMmP7J2DiCDCwTCuuFglWJwxFe+TzeOzEOiiH5Pjce9PBTRgHJfnS
7apBM8IT+HatvHmC848/mt07Sg1ZpYQo+xBRjM4viMwSFYX+HeuiTQ4X/AxjGeT
sS0sOmozJwJiRkzwB95wY5yaTuSBLZgk1w0cakzfk6elcxVYiN7PUc1/GOR43sp3
soZF7Q+vI6pIbDzOXGH5gE8yrukDhHs6pnQJ5hVwi4KBo1R5dFNhYv2FsQHPVKC
ocw/Ng+jARRSHTEvRyvZTTe6levbTjG0ocCYx7j2rNsyoV8MX4b1XpECBODpOAUE
IcfQUUqYtqfs+m4h3QG1ch38u4UVUPAbhqCy14HHSsmA2y097eej/AlIKx1Q7AAH

oyjCVIIrKtZClfkfu6gPq9ft3L0aYqwQY4Ns9Br90qNyC57zvklvZDziNDy+/5NK
9raZxhPSJzek09erc68W5mRld/M3+hnHUJtldIfd1Ud5LdJSnqUd/7f98xxFrSR
zxyxdyPyCnRixl+mcCRoYsFwkYtnocmBcuKgoNtiGpmx9KZfbEk85xHW400BZPus
BQReMzmCHYMPWTsh4RrP0BjLkdjMrImwZ+P3fr1PE5CCTwie9z9h05gXnrc2isRF
PMhm28AVdi7HHNHw0eCBRLuz/TtgwZKK/ZsDJ9kx0NXCoWgvyLC2QhU9NT3q5jYr
LzSdyoaTypOzoYQT7rIgaQ6nyuo2gJlrtkKYGAAWKp3Z8QIWz1VfV7XDxekKnPK0
i2O63tw/PtB/PMRQMqRvO4lBP+M3GY67yROQ75RWfvaAmQhyYfa9p+1FkLnaXlq1
8sh6D/BSRAr0aaGBPdxY+M/WBNnIAR0e1VfcwUIav+X8j4/YJDGi7Rb8IJj3C7+P
9ev9NDQU3NICaUVlYOXo+PCa+WMVG6cHkK2u4GvYu2r5/v57RScgzDYpf0JwadAx
EINItmSH827SL6mLKPLPr6nvGhMZSONUSV9M0XqgGUWVlFPh/Vc7PV4qpi8F36Z
i898n6XP7u1L7TFUvWYHEbsK5x7luURECmlkCr+tueRKzfeFrtfnP12Y6mVt9JZ
fBKOGRI1ZaoghQ0IsC0JP3c1f4z6msuZwleDm2C98WpohbHX3D1AnCFSPz15RHS
/abEFKAJ2hfuaSQnc/nw9BWcceX1WNXxC1bA8GsXRguODW/BgfJ+1GsptFZORZqJ
u+XIpl7NaHPrQ18pnF+pRN5Zqzn+nD03H0Uu5tdKKpk0spe1QenzG4bDzy0UBJel
19cTFLJwz0sUXZStIGz5KhWmiIW909evFGE6q8lm4LxUcG5OaSqUNmZWmJ2dGwGN
72Q7Qyg3FSZRBbFDkkBYAWUFrnjrHEAQSFsD9NVjrCAVEEXEHfwnGncn2Ysh+gm8U
PojoVWH6R1BIAGDQbITeskofo32dyIn9RHWPqWf16914VXndx/5XO/bORTCqQSpFc
vaTwSt0NVkFVRvCsGG74SCEznwBulWd6ijslVKnOrZqlMXfzPiNUSTk3DEdwatsL
12yNVNiKoAdKK9oxbIyMHYHJXJWVluhWPy4gS43ND2P1lePBWC6DgnFQyIS2uPmD
sJ8V4fz6MYcLZQyfi0nOVWYRUE80vTKAczJ4u5hJ0HhhIXSoEqBJONSO9X1Ta7MW
uKmqm803X7JHEZcCa1kb1S01KeFXtVXRudVLhPP5Lc+o+DaxfvtOEpxjD3wjB208
Z3fYwkH0aW3sDo2aWSTuYC98UJ0/imqlxG8+4FrkWRkaoGetwt6oXaDY1RXE8GDY
FOBIxBrxAnc1lgv5dBxsjOmzQmNYCHtMG3T+AfdKmszSRyPNWhi8NeEK9G0PThul
LYezQjfkTm6zhq3JlM6Fn9DZ3CxxU7MZRqrVW0yXgsjlC0Mfb2WKiXZB7PZ21QKy
qi0hZoVubPHAoAK6rezhq0Amd01f3K/L6qVeilFMD7ilcP7r7dW/6hm2ZV4WS7Ck
W3R1ERI/HDGJ15NnWyYaXqcbwaRhpJma70FWE6c3lm5s1mcu64txxJDJSB4E4aI8
HVkz51slcwbue/YzdNUbNrr98iuAlh+3iJOZ1jKK3bHfb8zBZL9IDYFv+Hsb/fdb
tkTASb1fZUIp3u9OhvD91Vqb3IYriQix8RB6/6cmvk3L+1bDGNk8leupqSPrhIoT
YvDSVbQSYe93KGdNbyUelU/l3TervPau2dOLlqkPfoEs+TXThUUxzjyCvp3kapmh
MmbI3pVHQZKLfGym9BZcm80gOVMLsD/ICYwLfmMQbGXOVvQRBvn0rVLdbu3YK01l
MZci10F9Usak+agLidFmLlCBWnLk3uBNsjlzx/KkSFMPp9RBCpVDdtY2f4Fm1SSN
Mg+dmnVNqZHQuXA/Z2nuxwGKxrWF29crk8Nakha13UOX+qnBPUnRrs7X/IFhpsY5
OsGsD3US2ACHpoJAENsGoCpwJ0ydsQJ1926iSbQpcyL1avqxouPA70KoNWL8Jn6F
uuh/OM/NC2JhKNa3wbFMHg3btoAZiK1hhT8NKFbZ6P7QfDkrmp9j8kJK7nfWsiYp
psAur9z0EW//oWWAWR/xZ0E5rG0QUVfjTTWEMVQOwf6Q6cjj1EhxYrpIj0gA56li
Cw+ZUqUAYl1FHFEvVTPAeJD2XyZW0jwxaL67DyyxeGBLJj5dzTBbBiZ06vkMk7b+
u5Z/iGaMlmgN3jS0y8a13WAn/y35u6HZzteP8A42ZL4+fBsFL6cmIrWDYsLYEmB6
0owZ5Iz6xmqlXbfnkRZBDmixp2eeQPcMX8FnXK+61ZE1/AG1S1RSz5r8HoPOwI4
/3HE3uykVyRl3dWCnQG1A9V/2xw325/WgbvZ7z4gOxhwsYTNucIyCik3PR1j8OdD
GfEICpkLRCA/28hWE663wV93bRwVMqJi1MSTfxprAW10ChqZqe91RM5ijXbisdoG
yiwKF87xW5/1fEbBhVJAnXqjvMtDZbkBEteBDMOJ4yR2lWOj8/F+96IPUulX6N7
6BGczTT+dFe22fgjFqjOllOaA5H9d0A2meloaSpveDLWSd9k++tuhgbq5amEj0+V
o8qcJ8YdforXi39Tugmle1Pj1JFSfG7uH1LFNzBBKp+cfDWBtfnQnsFUKJoXT/d
21Xw19DKzGIzfzCjDyrXDQEdf9LzvH6VJ3CWJ9FwPbIw0rzo49ULXk140Uyy9nhA6
JJlX1sI4q6yWxUTSXQunbZH6LogTq9FshR5xAhkHmJhJAdDMkR/d3cBcDxKs0pdk
5PPw7R1w43Ledc+sV73bvEmD7r+mrQXfbYhvkP8nmLB8VkbPUqq2dqUwvnAq8WkZ
ggzcOKk8vETew+4B+ElzC3wUzpL+B908qhIJu2XHQQkKJraDaB4k7/jTt1gVfjQN
J3swWfsiDRKYUrPzZfac8+smCyy6FN1S37fGLOAIaDfCtiO1fZc1OhCXRHI3uRpl

```

dNXwFG6OepZTs+r3yLEpqH82vnbak35zhJTzgWWlUutcLLYLuulaTv85TntCV5du
tEPiR2f6oxgo+96zUxxpFAMU6+EzZ01IeGYy61+NTJ0aAOhWv1mpff2uDBEJtdnu
/i7WYT5qC6Pae0ZWihseLGI1U/CUMfdY295pCfCQSTS8016J93yHY5bWMwMyDw52
Vf584mGeE3a5/j9ju9qnjdl7Z5rjR7bc7oYKjCP+Pv+R3pOo7jhNhTKChipvH2Ik
xi+aa9nsTlYgNFMtmbFljhcsiTbPSOw6NpNfJmynWlduqM2Ra5ZSMOjdKtOEW5mL
HKN7LhzMs5nWvxM2m6J26kzfbM3+d5W361BvgU6v9oCE8uSobGI/sSNP0kgGU9Cx
A9kSrxMnhahtlC02aROS08PSeAcErUnyKJLOdrcACRM/T6iwROLI38Nn3E/PuqmF
XDcN6aosfk5Gz0WhEuIe7o4bEDcHTKkeZ90/qNyJuCTwh99VUEeN9T6PovTSTYr2
xpl2Dca+KXzEcDmT6bL3eyrBAMRW8HyfYTxAJntty0pL0gszHc9Im6q5Y+HvKOU2
Jck3hlnygfBehDUwsLTWPg==

```

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] 7540 bytes
  (decrypts to)
  application/pkcs7-mime [smime.p7m] 4576 bytes
    (unwraps to)
    text/plain 419 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

```

```

MIIVvAYJKoZIhvcNAQcDoIIIVrTCCFakCAQAxggMQMIIBhAIBADBBSMFUxDTALBgNV
BAoTBELFVEYxETAPBgNVBAsTCExBTVBTIFdHMTEwLWYDVQQDEyhTYWlwGUgTEFN
UFMgU1NBIENlcnRpZmljYXRpb24gQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIeh7j00
Boq0MA0GCSqGSIsb3DQEBAQUABIIBAEwle5fdKMS6hyob72qHYwMpicWoxWhovcMx
m9ncW3nWi8JUNK4Y306rc91m1a91Bnm6koyF5vbpMTU7MQgVK8Xsfmc8P15UeX7
9nO2hq9Nk5YDrbEy2VetDe+8FmyhJHM2AEKCRYJENj8JVN32v38+96h/H+JtAagN
hbEnXCjwumjHMPq3nqq+32oFDLLRppc1JZ1khgX2LCH7Mjfrp8ikVnSvAUa8tdtr
uWtEPqmUktYXUtad5ZqXQXual6KDi+0XCy44Ou+txnGyzY/iFBl/U9o1lQtMSBaq
hrCIF4WUgYlH1u3KN97+lm0qx1FcLQHGXz/eEhbejFEFwoFIoukwggGEAgEAMGww

```

VTENMASGA1UEChMESUVURjERMA8GA1UECxMITEFNUFMgV0cxMTAvBgNVBAMTKFNh
bXBsZSBMQU1QUyBSU0EgQ2VydGhmaWNhdGlvbiBBdXR0b3JpdHkCEzB8R0APhiY6
HGLS64MvlsDXhpQwDQYJKoZIhvcNAQEBBQAEggEAKi TmrMj8VnpLrV7+Exp1xnLj
Vd5b5eQeYlii jqqjmlUqj8JoYMSe5FokiSfC+lheSGabYyRZ7KxKY2NRScXNIX2Fz
r2Gv9imDrelGioRcACAbwJTj9aJYZqcY6NCKfvPvdhcs0sVvw1L3CX/iUbLk5xm
P73HitnQIGolSmgB3M1hEVNIrhSefymvaQcekGRrNAH4paHMsNJqJOY77FmSVzmr
YfKeta7EX4sYy4Gf+7akz3GTH+wBHbmEFJnKp+4EC4ABLo3N7AQokqU1bn5DdUXG
1JkgTT/wAqPW7wO/JDM0yv+yfjqJA/IsWKwwFG8UtW9maIP/NYDumgW4CzYqUVDC
Eo4GCSqGS1b3DQEHATAAdBg1ghkgBZQMEAAIEEKA15xixtiHfDNDPM9jjLlCAghJg
uO1Sou8ogsWNs1fQP2jjEv7BYc+c4S79FLRrC6DAccBIlKL5S789CjRrx2+aIXVi
A1/gBT9yGFF7ex0g+L5Q47TYi/kZYprf3V8l2nf7NCSCn4MOczDpL0h98q6F0aqq
9m6kIL6Z2LkTVCTLTuUfFv7WivKXqjw5G2rbgvKU1Biuw4hSn604yNsCrOvLvr9L
fb8UA1Msy6Og9VVZJEM57Ns5wDcTnCNfec13RLvQs0MtaX4qtK8DiY+A8maTM5PE
VmbwBnkYL1NEMv3KMhbYQdPN2YfXObYRVXg+HDuOd0wHx4TXKYK3frhgN+uII6hN
Py3gJmRR+HpK/kxCzXc2ZuyQLycQF2+Buv4bfW6PczVVGaw80iAWM5Ia9H7Tv/T
fycspPk62ce3cGdh/RUT78mc4pEKMaZvut8WTf0u5szet/NnSyH/VgnymZletHL/
9ijhv2lGfkhU1tEGlHE3OIkcQhZAFRhmMfMgDHcOAuATGcpybxmUA1VSF8F2pia66
frmrFyzmKEQ1ce9fuyd0DX5MbPtPtb3fDgOPwHoknczGnSF8GE0kqIRcs4wiz906
KrHSwKM78SxxcMnJS1Z2V7l1fIx5LmcSiidjYhsrlgyDDzUhqksK4/YyrdLS5CA
dVnW1Q/x7ALB+/gyW+2EYj4Fh1hREW03Haqc4lDECCIVNjvxjqmhE8MnkUihJnqJ
yQAH/U+wBcK2zce62XHZpMbJBK10SSGfjY+ofURfhjPPzfX1HSnDYMWAYkFWsUhk
4L95+YwIcaDoYlen1XyNdmqRu7HC1K5tVQwGW4ffIeaJ1xBe8NuOMaW3Tmn7KJrQ
Y/QWy2sR/dgT3aTOSU08sM+OHrmCW+44tdHfdsAGbQYrBX1l+2XtP/buOecSgkVb
1v7B+4d+1T/BfoJxhDVyZ5wIulKjBVYOLJe/dj8JuYwDk9RNYpSl2XEkgl/5vsOa
NpMNTmx/Sp6qw7OgqETPhZX6zFevW/Q81vnXiX/9bc3JEr7AmPmKoxij0JAI+a/g
HPHTUR/7AylcvuqAXs3N180TtOzzu5HU3YYqB3J9eeovmY8lTKyKS+bTgS9PQExr
3HtgzFoLQji+x6t5YcijdiEjsjD0R8ukMCRH0QMostGlodIrAla/3BU4+4epKtdl
WwzPGLaPONuW11351U0ArbfY0VKUa1IZj0lnKWs0Pr7CJsEObiRT+WMY1xDR6K3w
PUq0d8v+m8+gldNpqVJ/jm1U7BswjmKWNtChgJYebwpA4BuAVUAQJJtZyY+MioEJ
cNRngJF53JY3v9vBoQD/7g3CIzCI+UBiS/duaiVCTyzIwQ+T537LmrFWdDDHhZFC
S4k96TozHwcJQZT19GW4svAz2M8eZBTuBWoXtPn4sH/BHOC5yBjH5bHf5qg2vV1L
dCfWdg9T6AYewLUr9c2EPd2t8Z04SH+KwsruM4z1db1LibNf1PpxXIPB1tpnKOox
nQAGYRHDyBLyIJ7Mdwoz5QfS1Z0Q61ct77tM343Rf1C8voyh90yDQXhGxkvfGPFr
RP1EEZK7oANI0nhGkYwXkBsmDMR+KsC5VXA8tfXKkSACepXAbY+aqRCbUYRvnpV
AZ3iNObov/wWcnnvYFZC844eFjYYg0lkbXsFcig0iS37EcGN2jSRZaiV5kVq7hHF
+VUnwsSFthMwtK+Z0cuJjRLrslupM4fBbbVdRuSe2n2yvVZiZeXe59Jr6WwERlKi
n+sc1D/wkXIBrCRGclYoyW1JU4A531Pd46dgcgHNTuP8Yv/PW6zHc4HT6VYro9mI
wJosMTwIuL0W+Qr8/XLN+siI+XhdcaVGA480p3BxrjSeeqyWAC2QRVbWnF5YdXmp
NSkK0lZsceL9myNGEBk6UTZyDDzo7aJiOy1rqPCJD1hfXofYDyPlSByHE8zoMnwj
KVhOHUHE2Q4FDiCpSJO5qlvhSB2svgWlRtCBI02qevuCHugFvbUIAI4sN0XPc15y
1afNwNbXK3bQ+ZC8nXwKZRxQLRBbEk+YGP8XkDmXf59WjGoRJM01v/5gxZQAb1s/
g2VX/juutTVUt0GZP0umPmrnRQXjwTLtFjPIETj4AKUuGKEhr+i8uuN1vIDHJTZ8
qQiQddhek7kTGfPZ5GTHsx4U1NexaiolaHClN2oDY1kY0XBTlAmuU4kFo0sqfD6k
hVHv1F9/A3sQt3v5ygiV42HVAjbYZ11RHRKPLBFhuomDx19FBPbjGzF+cOKYRrnN
qdZpYGtCNKp1VDudQw0ffHFCTjebXmPPokgMrNtidWZGbf8wEPEf3VHE49gj7+1N
e5dwU1UXWQnfAs8VBIF4kSWhDG3gIFhD8IKoNTRPZeDL208bW6bfEBuKR1D9DE4z
rOot/hUAabFfA30AU1aMno7Rv5XNidY9sGTs39HSxL6CiGdHq1OoKErMW6vaVnZ2
z0FCLo9VBtXR5qAGQ5MgFLOPq+/rhK8+qNb/iPozMddYgxktJPiORCg4B0xhDySt

8IuzPhsNINyJ0+eclvG17TQPwX69jaUutQm3F82ldrLrFYXBDytfyz+APuWurZGH
NtBGj9JkKN50//7reaWeVkdSh8VwMwwfTCajrSQerUEu7rww0z+mGsSRzawuWahF
ZFpyNn2o/Pfn8eeBOW2E/2n/ndPvDf6jvAM8rL2rT3gGMktmYM4TvZxdFHG5gEFj
7M8itL9dqDTaeaHMKaFN6AYqPIhMTnYJa15iV3eKavPwE/t33q6oeNW8Rb+kv/lO
BMAVSzwxKti/MLt0xRe+x8+8HyvcxaINaojri2CYnbxCrH8HjTCsqVeiAIui4/q3
GjxEAIEfRoRmhRU+qcJl0lXCqDsAhn7NOCUtQx9zS8ueWVUJeTl9SsgQpjqSobg
Pc37RBBJP/QNHODoGUsYOLEmzMzAtC9YMOcXjmc7g/2S9dspURtpV5UaCwm6eAt
0quRoCr0aJbnG0zRWkoPRob6ZuIEOqMie1z2QsuhTacD7OygdRNU2wmBcBAe7RKY
J7dsJ/oYpJf6y+uBREB0AEUJhpErULETU1Z4uoLBWlqyP3drMYAAadoXXIBpLTN7E
9VkhIbQxhCuN7o2q6M2mykHAqEBaCa9KkL0UiouLiPC9Ygxr0FAUJpzFBb6dBXQn
Jo3JqNw7TyTzVf7PoZlV8hkQdrvrJ/67peI+rMZS2Cn9ut93AilRdO2v/fJmVucx
8cei7AJlboOdMzdKTKDpX4Opmo/EDw1/uR4M7bVwoGLiVx83lutJ3FFmsNcwH53e
FgyM2jdlWKNcm0EnNhi8Njr/8j+O8iBlaGD9rTlkDRb8RlcF9VtMrKKp3/AYf6wk
Ecenr9xcJxKzKxKigRNHmj1hEsJEyELICYoxlglyfyJRVEyVHoqO3OJ8cDeSwfFd
kDK2va4X7CPQvWFkkOTsv70vw+Q820SdksSiU4bq8rK37Hku3qFwErTgFT0Iph8OP
dz3TS00qpIWVYTRlCgWJnmwv2h20AAC25Cfwwa9ro+Xov5dr+/CZPEl/0wF4aZ3h
34uau8enUXV07sZ0ibmPUvw16lZd8vj3I+h4y6JbQTclHtaNxlFKvRubFomrc1MI
EqWbB24KS46W4U/l2qv2GD5SfiV9SjmwX8hYvhKlbY976BNL5VzbXN2lyU2LUKS
Uhlv9BxW9c3YCo80yDY3v7nFq49u3X9Xf3lQtDI839WfB5PnG/59UJhHWIxppe2
xn+mdUaGoyFLXSm2eZefp7C+4lwsceFWgxpVT6WekOoGyq/v0dRMHOruTOKyX2p
BIlzuqf0/2Vw9y3fNEJNsY0K+Afi6aevoHKQCWwr4tjc9YEawrzeVyNU2vZ3/YY3
2NsMxQP0a+JqMpIbv2I3GX0lIOhW0Ws5A1/Qzx9rA+bIAXbQKT0z0bHdGm/rj2C7
z6ngk24CxIJNc38+YPh06l7agY2bfXy5UJktxZPNxU23Os2GuqG4ymkfM2pf1nkN
TXFORhDNpaZKhSEEU3Vv/6f3wly9wffjptEsanbt7oHDTnrhlBZPzUvvGuR7rmjy7
AFqG+Ql6KduTqdWV3U5FqGk4RVuOKDj8rxkQDPZo23l76g0WRJOJY+aQi7uqQStL
v0BtVtALm1LNMHfAza9FKOcbNlc+fyWLkJ6cqAlDKUqIyUAFh3EqDQQk3wcwugQH
oBOZYRSsY8vSbFvA8mL2njUfxuhnGG+iUcNxJGaURHzhABBfrRHlBbmyDjW9gJPB
TQaDauQgHe0K5OZooPi9UIGPIGJCy0hgnF9MupBiMkBDJmOBLK1bx6Fwj7h/Qjrc
eVceaQtEbXvhiwH0BjiWckvSQ5tzoCeBE/9E2Bn/Nctf4ZUzWK1l1jAKU4Cb5Hm
pZTTueisZbC302FNqM7hOOFsEger24L2YH3TZOOaNaFdlpzPAzfbBqghQiFrREIo
NmvhXVVzse+pskp3bXscjdartlkb8tXIKNSilYP9TArBW1zBxy4hUqHv30hzS3uP
PmxfPlZPGgmmwu4sU9uAXG8WS+rWsDats8GmVnnx3kXJyessMHm7txv02T28Jp1k
cj7ciVmkfy3CBT5mMZ8Qn2pWAdtsXFn40QFE6CGTiSDkZ2LA1iisyWOqQSVg6o9A
TYNF7xBFqxi2AiebiERKRL03JOyZohVHNsbMWlCIMXUdH5TfUKAVLTkaAbB5b8Fc
lNJs9EUuUFpnuorJIO6sETOp/mejzkaHiq3yBsmHgSKoUHTN49lcjg5GM8WDXr0P
8XJqRBIme3ySH6QF0yY6vCDISvKlAPisi4LRd0zpnh6LJmMxoblTvMerzXCEL1lc
wzAzuhVBGNI3BBltQ5mYdTAj0+SSDLwILnJ9Sf44JuWc4HaUBzX3nf20G0T2bxh/
2N1BbY9U5Fjtp7R8dNU37NrhxTKjhXSabJ8w0x6dkuvq6uWLOhzAw1OQKkHmEquR
X0G5kRsWw0DYi0m/wfpvtBYZUKr124ejzLr19FLXYVx4cV08W0gnHxc4FW5mZl3e
LlSyxfe6EoYzj1Z96wM4buC9TDesk83m6TlKA1PlZvBbU+nnpgFL7dlRXrsTD0Rn
5nk06TDXqQba/sRzzvyd5zjF8LfIPedIB/X/zpqy5jK0Q0lFixhOpyWA9MYu2Se
6keqSwT2lnfR/ZhpqRhA39TnmoITHS1lAPPkf/Er+8ecJlSNVzfuVkrBVVvSq4Hn
tOA01lItIK24/z5wale8W4dGnURA2OGWEFm1YACq8Kl0+nSir0k4+VvJcm42+2vk
Zi1hE+FWxHQ9H+Pt58nJj5pNf1P6VH+up0515X0EHRikTm9ecyYPQETdG89ZbUw1
Hys/lNsNIKnuREwY5P/J+A5/s/+x194jNNBsv1Q8kLngufoxOQBbpwGjRxCTTfzbj
MHIONho7Xg0TbJQrq4l0U1goDW5tQsMH4VUg7ESiLzceMECYiQsnVLg+FOUqDvJ
4vaaGYvSTIaxlqppjL3qpHbmYa7+XXN0Vr8eHvr4XPB5PDualoEftZPA1z9dGCEAo
RPISOqbZBVx2SmC60mZ3ANIUnBDIA6/6VRaByAWoH43QkuC7c6Z85TIB7Wosx9c

```
KatEOhxIRGwcTvLf00vKY3bHb9aihWMDnpBEKUfIphU71iC9nCtiij16NokXMNAgo
SYvTbH9XJNfG7R3O9dINOfD+aKtVky3pP713HZyf/FiHyH5H+obcjX1HsKidTjg
BUptxdFQt6yVJaAy0xCZtUPV8Yrd3XZaAV1rX1tDsnfJe1Ab5u7CxpDYLiJdwLml
seOiMm6Uy0Nxr1UhfKmx9GPlrqMgm/U1Z2NBE5TKa51AI+3iIGWxPUOByT18/7S0
jYkvk2o09B9iPcYqxyUn7mS1vefRxxCmbIOP2lo39QYEX2zUsh3/kLoqxBwRk/Bj
/S2lLwfSx1lQrb3dJHyHyBIrMeGDCUXESmEv7n4JhF1SWjMYLTDY6TmlBefU1x+a
klwE9TszGKt+rCMCUpltt5axy0zPz3U7yJF63/j+kxT3YH7SJcmC47pOpALhG+dx
5zdQnZiTtioY07E8ZiaLPM2+42pYf9vhugpZhyA3R/EFJYBBqjo36Iw7Jh6gP9q
NzM4+CFNs3sdVuvkRNbDks6s9Z5FICjjK9BsYb2IjzyqDVMGdAxX0uuJTLLoXTJ+o
lis5qqmnmmylbPMoSsUiJZ+0ZQbw5m2NgNEZNrQEvfTj4L+R3tZfgCePtn1lVji6r
lUo4asH3v+jk5varkDOjOF9/mX4Ycu+TX3ItDx2c6kcbMsP7tknEMa7Xd006g2f0
```

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] 7605 bytes
  (decrypts to)
  application/pkcs7-mime [smime.p7m] 4626 bytes
    (unwraps to)
    message/rfc822 816 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
In-Reply-To: <smime-enc-signed-wrapped-minimal@lhp.example>
References: <smime-enc-signed-wrapped-minimal@lhp.example>
```

```
MIIV7AYJKoZIhvcNAQcDoIIIV3TCCFdkCAQAxggMQMIIBhAIBADBsmFUxDTALBgNV
BAoTBElFVEYxETAPBgNVBAstCEExBTBTIFdHMTewLwYDVQQDEyhTYW1wbGUgTEFN
UFMgU1NBIElcnRpb2ZmYXN0YXN0YXN0YXN0YXN0YXN0YXN0YXN0YXN0YXN0YXN0
Boq0MA0GCSqGSIb3DQEBAQUABIIBAIG/Nu5fmnMkn1fBsCANbQMYLALsx0mJWEly
TzK5u5MUntTeOq+fVAUULIJkXaF4inxIe6HSau/bWDWISRY5txztDBIrGLB2RZt7
Yq6OY4UVqXmd3EwkUab9wJVvj1ZTP408ijOAfpcJkzfcQD5J0ZLr3CRXz7JT1wR
```

CUHwhSBCMouy7/1M2fKeyI+ThUNFUQQRECIjAOPmMrQt1dYM+bXNPi4lY9BVM5qx
J8DQG9XNcQtPsIfz7ELwD20a7jGykPYUHzyFE681x+4KTBKjRZb9t2Ezeczydep9M
T92aV0ZU4A3Vd8buJG19sUvWcbFR6/vhT9TOHHpqRU00LJr20iswggGEAgEAMGww
VTENMA8GA1UEChMESUVURjERMA8GA1UECXMITEFNUFMgV0cxMTAvBgNVBAMTKFNh
bXBsZSBMQU1QUyBSU0EgQ2VydGlmaWNhdGlvb1BBdXRob3JpdHkCEzB8R0APhiY6
HGLS64MvlsDXhpQwDQYJKoZIhvcNAQEBBQAEggEAKUK3Tne27yc9+vIqGMeTO6/u
Ieg0Iav3LcaUwnCOGLjLZhlnpZEzC/SfNTobX7d/2yPH5oc4gDxGekJO2YyCkin5
RqpY1hIeCEWti45otBUinis/kAroFNbe7TOFJ9ck5tVXxLJ0WwG4mW+CoMLRF6o
E7tB3VSVplzvuapfi2/TrLtmCDb4rlAfyhTIEIqY8J2LuSEbmDm2RllrWNVhVPTo
9gQYfEz9VxyC6Ix13w18tJ7vAgvECibxVDj6AVkAB6ThJJGLE5YRQHsqbEdBQjBX
RBXfKBjTQ9eZqxRIKjfp1liYA+tNktr4WRyY6YUAlDWvb+GBV/qS2F78yJk4ETCC
Er4GCSqGS1b3DQEHATAdbGlghkgBZQMEAAIEEE9A9jv97rc3CRK2SsCiVP3yAghKQ
4aCK5bc+ic/OjQtKizK3Vidpqr/12OkW7gP/7UOS3BiPtRIUkwQuxlCsiCmFa8FS
B2Hv6npe8Pglkv6B4E/paVga591QdjPmnyoUmAWrH5ILbAdH1lgybzh545sSg5
xHftcE9xswAoBb0Es60qRMyNEbOilRKYIDVoXjFiyA5SxLCFxxZTveJGqQV6bErY
bEsTEhz578Cq+tmZVC6fRR/iSi1ZilyP7AYtCxUH1K5FSgt8qnxLSwk1kiRaBnMJ
Wtk3Ve+BETCUBTn6jYdL3rBLw8rx2bp+qUcVCu48KTW1Bk/eytSJ6Fn62hJnmNs3
m7U06C3nra2hvFWYhKva0JgOD+EyAiqGwWXOdD7jRS9js/dkFgguzVT9OewCvEb4
gGTXLtmTF5oiCipk5o8rRhQk8mkrXQSmfAKD0R7hav45BnaisfI+4rd3VRBqVONZ
wXVFioFEpq4hhA2FCV3owC+DiW+54F6gUEz0htkkfbJdD4r7+8u1Y8oLrEkPZGJU
7SojAM5yC7TErr4U9FCligOLjWKmeKud+rV+AGKUVEtgXlAe1C6EPQDY+uToSsP1
bwRmAroLwBBD1fttSRuS7089AsGqDNLbLfhoxrkwtyDG/lt0XbjWNNw+8a5y/nn
xnLklpqHvaHRSzrH6VAcSmrSuJUrJ+bxm7yPWqJbz17+8wrQa1FObsq7NBfUz1LD
93+hvK0mLVIWTPYq02QlkYgRNYEFSXgTbLslA519WChT75VhrwQrRT70JVP+RXwd
LT9su8myIiFWOZpEIjPsgSMAJs7EPDjTdcMkBEVyiQRICnra7lZsJJi5JQa6nVg
8pqD7tbH9ZH/AV/Z87q00VNUP3ppQW1kwaw3ZuLEH9DWfxVbrLxD+c9DjzTl+axI
voBsFnXWUQyW7CsirR0jhoM7sLcLXqv87Unwx1H7WgSiwKzNAoNj5gvZ8FB9xLw6
ZvndV4o8MOYKaXQuOkIo4fJ8xkja4g2suRFsOHUS8+EeuBKmMJhmOXx3P2TVFmY
jZcIPkXuHbJUMS3sCcDkwsN6Xbt7aa3jzqUpEJwwge3BG/1PC7Xeb4JgWH9uP5Hy
/JkC7Q4gfLcQXNvBE800MyGXpZCj9iXWNYsBAHLazBYpARpj+a2/nj+D0xjPYNo4
iwBzCBpOpva2C0f0MO7Axas7XDRRRuoP0bVeo/gDS7Nm7mq+HpH4RYdLP8Idr4ff
8wHmniHggUDFmVJnWAEePrMXZb2fCjr0zFAwHG7aL7GI4bH2tbN84uOYFGCUrAf9
qRe+7v7SGZiIqIXNvQCzsHkNbhSb1hOeAeKpMG+nkU4IHI2GGjs2291D7kEkKN0F
VA4f15pSlKLLEF8T4HhoWc8S8+sGxdXm4iujbis/yrkXH13bk46A55DNk+aCvDk1
nJatM4o58mMFun1LaCMUZ1/AQW3CFDRJxOU2Ae7VbgXRsb6gokkiL7hmxCOFNXwG
ff75Lo6/MywhXI8vANmoTBVNeOCO7atRVdzYZ3xvQ7tTgUgr2BCDQlw+1aDLso60
SxunTtZxDECm9V8mWeoQjzmWYLuYeCbaUfeoY0dhQfwlph8tOrunEfwrbfCMK1Gv
QX5b1eQURzZ/owrqE9/fUHHY+EjMrxk0T6+45cA+N3oOJS32KkIgv6+91GE43YKK
9eAiDYmrBaIoDMXAzpW0yyWmzPjSuKuolPsCKnVeMN1bM/1Iib1/lyjF0yegu4bS
0VIh+z/cNBg9Eetrbr2gR68d5mZzWXvB/Wfa6VM6Od16t7Kq30wiFUJ5OtVaRPkg
NSOeAXekL2rUQdmVJFwOtO6FmoYimgc+YD7b4HZICUSbpaernIhy9+ZS3iLrci3Y
9tiMlikwHpBX8ykQ59fI/i21SK+JVtqzjFOVq6hoRLegzQ/OSHuiEr+RWYmnGXH3
TLRaPx1xp4S5P5zEsrIGmkQVudXavewItYxq4vyEzC1BS7L4rK0XcK0n940IKJj5
YwOIj2uiGGew6AFVEF2GsO29XdpbM4XbuIrXMKVBV5VR8B06ppA8NcVOK0Pgvfho
66yomGxgvUn9V0v76+x/ZZpsyonbIsdfnoHmaK5gIfUcAKVIp8I2B7gN4tH8ut1+
YumRhc/R6Y37ZbeY9ZpMh1WFDJ04LOaiccFaU8yt0Grdhmg+VLQg+mzOUIZReTJb
VCP2201EGNisGeYp4sIqlVfziAtyPgpnvTN8qtUhoZOZ5ghK5x1B9nmmbhf2wjOGY
vB3dyw+dTkOBiH3tqqS90ATEddzJHHVV/oXzFAs6FtGbRFA0YpGvgYC+RUPYqvqj

lcm1OLq1EHl8tpQlrWzTEIGVUMePTRBW77CXSZGNh3yz+eC6l270KPKbhNbvZSQg
uI+NZXnGCdapQh8NIUmn4Suo/Kevo9/Z5WKg2k1gFI6rZVw2rdMuY0PVZfyuGTeE
KuLtXAMNZ8GVFBOq/uz6Goi06s5nFh7587LHc+X4bayK63tuKnkRdKJozqzChoU7y
P7zFJGWRORhe70vFwlihlYI2y9kHl1Y6GSzULYY2tYozH0cmAkYMnSTmeo5lq7Oh
NveHC6v1vVQZ6BUYN+6fm/jU8fuE8aTgrnREfdDNbPUF4G3hZz7Kyzu5KgWxWVjm
a7Jdl0MxVjUhqVtU52/H8eikdan1lQCSTtjnt8BP2apT8lXjzT2zdZsiIeEXhylX
03ao14tBqMDvpZ2Uriq0S3d406zZ8DdCA/4vqyVpdA5GYxj34Wg2tMN07XHZ+5iF
4D+Dra9pXS3mqmR+U/MUF495/9xM6+eKSN0e3gyHW3LLhMtnC/sNIodOmMvIkeXl
1VblCRNsO/vKpLm9TOgilK4uhk6//Nha+SoknZwZbKpV2HP/yjFm3/yopccmqRbJ
96z4Uwgqeq37EBPdrck7d395U29Wntzzh122iaUJyNYXmer9OqsH+tm71mJ6NWiR
KQ23Pj5h4nxvhDRAMD2tN65RfRPD+Qjz8QJ/6h9scXL2we2QuzNSZZ/IfITHt1Tj
c0Qp3HQGFH24JSf/QnhdPz06SUZp0rzRlYkgh97miSOzOZZt6K0oPYy/YeAC+kyL
K15Cu3F7fVrk/aYuU3TSSO10vfblioC3K74lWQZHMEd8nOF25++U7FspYVGa68Gq
lJiI/W8vhtTDUCdSwymn1NgsrVVG9ip7RCkSBjoibnup7nTOLbdi/yNTmgD+s/Fu
F9ieEEQN0/k8ARP71YAZR8YSaG2dLuYh/pRTpe3xoxLqwNyC6ck2eOWq0lK+LBOi
/T+b6HH2v64De8MGR33MNDf2DagAJ40/RlJJqXhLm6Jtn0ZB4C9gygJRUMv9KIV
li9yccYXs/dU+zYXiVOWedmN7vtm6lJTkWfet+gTRz4zS0z3UA2+dtiu8LLVm9oG
5Bgb8qiRF5WNXjaB+HC8lbpJfuIDzaJa/2QPAwFH3tG5ixKlN4/ryCwoGllkamDx
IiZPf+2itg/7CLDnomfCGn2XEelWxS8CGR+c+sH1k3umqpDJam0FZ8ylg7gaFUO3
QhpGY2kt7EvPhOXdbwMhNADHFCu9oEC/TLxknowMsdjme/vA1h00ttDWGdPnKQO
VYpCRFCQCVovNqbrC/kbRRiIZxnuPmcoRcI3lMqUDirZWfyxpMJsfgGCQXAMe72q
nCHGQGaRIC60JXosP0wFPSibg9HloaEAFaWeI6rMoKaLy2WL696rG/zxEQSOvB5
wTsHfS1UAAb70nCVoLu+0lS7mL2s5JPv6Hk0i0+wSi5uYMOpO6TUY2tZE3ay52zR
tJHKVK0rT7yTe6VQOr6PW//y7Ygqy+glBPVUJo8YV6oV4QF2vrj+StNKV457paQ4
+Ach6FXcShgXi6Em41W/wrBQEt2wzOUv2QKsx1T4rjtBk+hA1xfJoCYuJjiTqtT
HpdHHTPqX4WzGa+7Ke1r1YITR7TGAbo1PeJd0IMP8mu3zoRc1p15Te0mrXwM7CuA
7f+c5VIPIXaPxcQmGdPgrs9t9jzpV+JUpeokAtUpVJ+jcJtTaFf1SQqd/6w6rI3o
uvYT5IxS05EUu2nTYxjQRuTlonWNXkqVHEDGi99u/FrOgh9fZ10oX0FGtN4u5R6H
58uGsmJnWUE0VoJ+liSKb86wgwDjW8QOHnraoDBxAhtWTuydmuEhjGaFmQdNSKr
I3xC9o2Q4dqI/Kmht/fzrZiifbxvPleMkvaMUOKPEdWOQXaDeIAauR/Bgl4jyrvo
5GcTdHxa9DBvSpuhe/jpTk0029DBIKhTWPiUK2mCoRk+e1JILSi+k/q6P105sziD
TIBCjg7ba04EahFU7f8EReRzToWb2e+a2/F1DIwOr6o8SQJcrDi2MNORjEpOkAWP
HEAeTHh9WoJXnEsnHChwG+pshviy+tzInONjU3Q187xSubNse05u+ttKVTLtMER6H
AnU8UzFHkUDnpw6fjjJRfKYe7BQRm4uxeN+V3CjzNrK2VLQvMiUw5fcuEOboEhBT
dzhuObkrGKaUGGbuYIBhR5zVRQC3QsATra0ITzrPEBxGD2yY/PkpW+GhiV+6Qp57
fHtZSB0EQHOM3mihF0XJqLnx8dXAXJobdo5jNBXSo14os4fw88WdUCpmBDPpbWgD
Fy6hynY8tjtmeaGFQC6o8tzFNMnSH/Re7uO77x45Ly1WeHBhXHARumCEVRkI1Yg0
8WE9KLZ+TEkcTok4hMcYH27XnKSWElrUNV2ViuXKyH2jDZe21SvLO9kex+h8F13C
cfgeToh4pYrcxYB1Q+2Sehwy/nubL2pTbq09ZifaTyJaUyf6ilbAX82TUVSCRRn9
pqGlo6+sFZsKG/AitwV0xZ3DsuFbhVaePSArpAGJ6VLTMeHqHGy/20euCky6fsyE
DAU/W4DYjv9cN2BoATOxWkWKyI9IbGyN0Ob6E8LFPXoCswXAtuW/MdphWUHW1KED
v/WYC1ZYL8oRiZDAvNQGJxp7CI9iGaQCEsbwzoGw7AGsb7pt0lflJfVTNC28qSG
tCeI/HdZbUvdwUDRwePFXXSh8uhZEOWFNnqaIVTYDdbxnIHfnHNNjBczT+TjKK1z
s5A5dWgxCTLZcGKGmGcOmiw/KnNsAEJ5y7fFur4fvKrXQvQYctdYiJ5yX102gtci
IHiHmfohFr14TWB7iKEVj0+pfQqqnJIWYj5Sgd96UkR9F0l+Suc4lnTRqzSkOkYj
zkYZFaa7SPobvhK3N7as3niAgcb4VfTAoFXkOX7oVPPpDrHcd7UfZ/Vj2RnuO2/7
4o4aUm89U3k/9FgEapUL/rKCOoCGnazK+w4+Hcg2wzkgkSNFU/sqxEqY7cKAHjTt
TAXkYh/F1r7MizSf0uFRyksMEa3NSeDqNhDhHV0IbPandc9CWVT2eqU5uvOgsNPp

oLnDUUFc7rkQhQWlh39BaUzndXGU88LT5Lqb31Z8/8/AMMn4ZxowOTggd3Z0NSVe
ymrsuSGyuOEUEUagx9ipbomjzc5Cz1oOcF2D0/0ofzdTPkGFhb1NtOjutGbg5x50B
3bIphtV6lpFP+GapZKcX6e3081J/2AV2hJywbxN1AnLPnqmkGeHaU1nOp60JQ5TR
It8Oi/LjjNc5hrFa8zKU2aM+c0lXT0VQu9DvEkHqgkMBCH8B35NX1Xn7GYDzFwBs
NnGcrNvJl3y5LbJjdrORSyggVHjl5Rda8Nx3ihLdt2lkse6UBoUkZMJGwc3ZmpGW
2wX7+5Pv9ttUmQ4bx7xcKy0su4jQaOWpjoJl12G5Ju0BzRx0Vfvn2WGX4aY0AJR0
uIZgeibQy5/3hW5keuHgBlQ7134DgYMSSjj0C4PBvHnpSnuTjYPggE6+D7UrNnbX
x6PbWeP0soJxQfy3i26+f1Q2yPZcNIOszSulQdK36RteOR7C2XcQhsivgBbsM35Q
3E29rbMMFDfUzCZmdJNivvf+kvHID5I8RtX2p51YIQVcyItTunQkR9P/avTMBqyN
28vQ1zFk3RtJrpOuy8m0nOfNue4VpUV35u3FdYIa6RkqLB8ZBiLcSFoi559B9czW
C6zz4GlpohMNJbPN+dNbNFioTeSi0dE0vHlP++Xo3phOC3bBcRxNwEoIEYwxBS
uWQGBDNIdRHsYOVYSSiEx9QE0bOinnitTHLthPcpce0yMQkl+diABJe/J5IBPee8
O9sicjpgeFcIozBDz26njPOgLMl5o0xtKDsJltKlOM2g9NpA2kjXy/4uWliru69E
c592xssBoY3eEzoKdAOE2OHUBVnmA2v+kJc51y1BkY3YYi9LICEDPZvR0PTD172o
cJY2hGykCCDvfrTBjTuvIB5KeKgMfJRJDMtGAfzPESCXOZcDr4pXX4imljapeGUx

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] 7585 bytes
  (decrypts to)
  application/pkcs7-mime [smime.p7m] 4600 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
In-Reply-To: <smime-enc-signed-injected-minimal@lhp.example>
References: <smime-enc-signed-injected-minimal@lhp.example>
```

```
MIIV3AYJKoZIhvcNAQcDoIIvzTCCFckCAQAxggMQMIIBhAIBADBBSMFUxDTALBgNV
BAoTBELFVEYxETAPBgNVBAsTCExBTBVTIFdHMTEwLWYDQQDEyhtYWlwbGUgTEFN
```


UFMgUlNBIElnCnRpZmljYXRpb24gQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIeh7j00
Boq0MA0GCSqGSIb3DQEBAQUABIIBABgRQRzXTRs0Jqxr19ouqlyyOUVTZpzS4EN4E
rRGV0bK1OV1080iF4s730amfc1GowC6Y0ss5JBen3EQq5NmMsFXj1U5sSiFGGsX6
IjkVSHC9c9QtdJtXyEoqEhf2lGJ22FcLjU0M21XxtKMLArch5aouJO1+nTj8AIqk
25JNvqG2dpiLaN61T9hSnyZe7bqDUflBo5Xm5REOc6EBvO+lFgjtIJB73QWiGBu9
C9iPJPz7du0yIReoX0wtKClqUzrBEiqO64SNQ2MuLTLr12niNDfaQrvfDa62Y6Zz
RKPE+I461Bx2CEvp18cJVdmOLPE/41b6QPu3816L8/fSoKY0Ck8wggGEAgEAMGww
VTENMA8GA1UEChMESUVURjERMA8GA1UECXMITEFNUFMgV0cxMTAvBgNVBAMTKFVh
bXBsZSBMQUU1QUYBSU0EgQ2VydGlmawNhdGlvbiBBdXRob3JpdHkCEzB8R0APhiY6
HGLS64MvlsDXhpQwDQYJKoZIhvcNAQEBBQAEggEAbgf1LBuq/SuTA535o03f10T7
hFJz1cRgrOgdYfaJl+bAIAncrUXPCxEhAIJAV9DNOJnISnnTNW0E5ND32Dbcji83
GwhT2iC+Uzx+0auUYuuVZ/go7eHMUWrY1Vm5dqNq5JbTwVgWy81IC5CatZVYDVFW
o26J351tuF7mAAIaLYXOnUrLgqWpgq17zXjHrL0hADx1aJARcCY3Uv/PO1Y0sb83
1zQQs7Mu82fjhmJWqZ4yQX7rBKSk5V3aoPjFcj1w2vQWUXHqczJmr0ZHYiaZQuLT
gg1kNNSPNFV1fipXESE0ksP3ZoM+DzLahjfkSLiQTY1Gacasb9+oVwALBhUoCTCC
Eq4GCSqGSIb3DQEHAATAaBg1ghkgBZQMEAAIEEEKODP8WCdJVi34OU9/jVCwaAghKA
Ed5TzquhpH35bEbuVz9wfPotJOKJ6xieYlQEccchc8+87Log3fBKWsZolNwcRMZzW
Phe8p73CscBYylFWDtwWtTQfsu+pizFoH1B2u+byGhyr+cEVOcI2hSM7BTfzBEbR
RLAWNzse0Z1vW9MABUHhu/7QFVwV9LYaL+U1EEAvoPfnX1QP1WPbjyI14v+/4i4B
6jk2HBM1N2r7Kjk1+i0hdt8V7WXHRWifGO9rGmZzi4hVkfLiRkqOqXpghbsHodTL
mWf8LfmXatmz39ueE27ZJC/1KHgYfdFgQkTfSutBP05eP71JHPn3cb7ktJ3wmEj+
0iCyGySJ1wKB9EFbWPOo3ENWZ90csz4250Djzzx6HIUk5jA2ePiEw8VyoTCq77kc
n88G6ucn+7hApODGLazPByQeB40Tg4EwkVwa3fZ6CHENZfDNDjiqYtBtXLuh7KAT
elv3UmZ5PtoWGuUd/7MYNeGiZeVuALdFAzI9Z8uY1BEQE6kZQY4g1IAvvd09Xvu5
Z7LA4qbfbpw3708ps9KmKmlcrhmDs62DkZP261KUGC98FmpmKgpKmpb/V475+OlZ
FLJKE8LVPrhBQ1gJWSFmPCj5FTkWml+dAriVS+7RdkeohjOepRIw7ON+BODCpvsO
AKHry0k5ANJOZhIgyOPCByDs+AypJtqP18M0azkThmlFLBc1m6HDVroDk1pZkGib
hgANe0pnA87omyIXs3lWpkApS3Ri4Hr1JXj8sM1gqJABeQEOOcej3yI1IcKgVh1J
OYPfeRlibKzDhbIpVFs5QMzKNNwil/t2+VmuV9ReyelptdpXPFDp68ilPO/VCyMk
Uq6yKfU/3gtieCtCgYbh/5dAcYwAVwB4XvYqCO4Sxj369X90TBM5Ege/4e/jcNik
S4wJ1VNVIGs6W1QbAsQ0GwwyULguRbnmXuwXmLySLgKd3pqSeR6mM6HGGXe9rdSN
miIc53pdrWaaLRqP35oyOCjwd18xgaalAV2Un3AD+Lwwts2rSOpITfTLRHPYvN9
/44Hfmu1G/cxGTWfJrXq54hh+UteebSyKUx9Um4LGqs29HIx5skDVOxhzYPM3+J9
ZP/IVgnm/tqkzVvYd0s1SmHdhQyXuGt9BaWjii2JZdrQjbUv7KrtfLcGUNG13yzR
q4hyRecPQeCO89AryPZor5CQ2H1filibSDcILtCP2UDzScA9qd3lvMRZV83rFcYl
cRYGUyckJP6aJFYUPCXRIdei9/nSkLDCIjtVHESdyUtGFTv8DeTH208INYj5xjBv
cEtW1IM2DXft68jf9Z5XsnUM1QO2jhLdaUptBwmKDgzeQa3KESniqdceGLrTM1H0
lFgMPFEn9W/Ma3pdi2I21TnzIcS7ZaO+NG/2ZLKXMEVBrXVEU+R7heEo6mey9+qV
ftDsbNZJoB7mTlMf75Ut4jax9YReArT22jhHyxZ5NiUu1200emE6VM1H2t3UB5gS
9aoVqxh9xNiDMO+6Gh0xHbc3m712hWT6yIHYcPCHzC/wqBE7VE1jccq5PF3ZpfrBz
ZMVal8yGavhW+1F/Fl5GUpsyxJ7LR3RMUappLFdx+OBrAHWI3B59ZIDYTodigu6k
e4qJyNKMw1EGusefonkkAX/53Z63QXeORswKzW3cfydOvwfC0Hi0TQX4kqXj4MAG
N/gNFOVRpbUfLEmaWyohkVEKcgxqyYm2Qvw0oADhU/Loz9p6a1Fjz2E29DNsKtdT
uszU9+2D+9PptibTCm5BOEbgM27wSfTwjcyKpcZ1E+6SEiGVQthWIIj8cCSkp9uG
vTQrG0F1HCYZBIUixyzrCJoc1jBRv91cRrjG+xdVOrRX2gNKz/bgU+9e3MPW/MFe
uuhCqpee6qMBPJY7JQqa6qsJRDlhmjib2gCdSLsYr8+E/KGTWu1TDDb9bKq1I1lm
3LWl+d+VrGBz3H110N2PDgedjwHco3igrwt3dMicqF714R/aDCJXgQOb2PxoQoY
Eyg6vrAoykdSfrpFU6UDhXbnxBdlsRSQ5zfx49Rr+YHXOk/VWuQQkeWMA0m9nQ4C

BiU72A3+nP1lKh7mc0/3FXzSEuF7zzfhfU88tEVvzmTpVJkgNm70NEZ2tX6VBe9g
ycH24ytDbrYu5voZUP1CepPCdOTwq+uD1iU/UcIKxnsnxwPmnvqU/3Ch1/wOd8/V
4TwbNbRlSYit7Xt/3Kg63vkQa3wOBxZ5j/KOZLLPYkSy10JTzvE7Y1GlF8T8oeGP
li0RQbOaux8t+j9ZrHctxfDvbTOEOXYeVuQV2rnbvQcXg+KOA8Ef4TEfSnnnG/1
dW0Uvb+YxJjABh84LTf6X7ja8BTJIY+oyIMIptw3Iw3BKmpHe0DqZaJKatzZ2JP7
IaBmSS46Oxngqb3tIs/iX10OuvfoYFF8JP9VNw1Vacn40mU0YuGJi62oWugI5yPG
zjI1lcVAsiiTYMM8OUmw/UuTDwIgIO6AOSVNMMjWcihBOQSn5HgJNP3dc9JWCIZd
xM5npOLCukhsKgZqr3MHHroiP6Jn+UsYwoNvFeVkvZb7nZM9sqmrQ75JJPiqADfX
NpSGqNdGU6q4o7aCtjegr0coM4xyfyOEKyq04w5oXhYzAQ7qGvN4j0iw+WvtIX6x
kMv1cVXLzeJ/oNxLlaIgZjt+sN8MGtf1IBftWxfuGO+WkvWwu07D/BTsxexdfstQ
J40lhuuodlYSosSHMcTlYdDaRospOz9pvkjREwwb9RZtlnCjKALdVGeLDBLG3bc8
SX/LC//AosoGtlgzAftBa7/n3Xup3EqME+nXH1K0xjvED8jh6xchDA8U+tsghuCl
0OmY4Gf1qXtshxJOftbCGEOXJUGFLyYPUg8d8cn6aLwQiRi3D8OMzhDRSdz3KWw
M08i6lvavxGnwBPG+XIvDvxkzEaeEzrZ9Ea19/RnW+bZwxMwvC7Ecqk4q7o/djW+
FKjWedjngYAJIHSZCljRDosskfmgCEL4nfgMwVfqF+xS8bTxyQu5RxbwBPDk8EM9
ZN1EH4WY00hgN4N2oql1TUn8L2Ehx5JAhiTckZz+cp/nzKVpKArnjBQpCjTBUDiG
PT28zjiTkrZileKw1C2zwaQ8KOjMjRplAn1P6zSiuaYEtF/GW8nHzG9FcJoRlMKR
TUt05KBg7wgElRxPumyws1RL4cpIb2oWlyfSqiYNHdNCQykyuu/ubaQVg3VZyz03
CR15V3ErDa95ZM+cbaGx2JMXR29N6wTXEGi8FCMZpS5gTucp67yZtG3Ik+PPWkih
8bYskpn0AcPCl283neE57MhsEp+BOekq9tAx4IEWDVzL7w1EotLT5gp5iZlqMeQT
A4kCWEbcX0emotgo/KgYhSfgaSDa+LJqvFN10AqpWU0ApqrBkhDUUY97uznHWjXc
yS5rzHHDbro448nJpFo9ioCAWFYkWaEKRCELjU1qlfdaP+jHYIz48nuecCtuVOeU
gpdgE4EhL0mGG+y1j1wC6Isrqdj41aR5m3ZwMeucBE7RkyiCVMW8/GobcG40EqGn
grvjomjWljOIoJoEUzsv4ED7JjAbedGsA7WqGGzVTyyXbUVseSuYsb7eVy7I0VZF
KiPI06KglRA9AQYptnij3qku/RMQNWWrSjSSwUlm4FceY77GGo9BctQ7DdYSOMoa
ia2CYsL/nR12wRySdKzJOBmgBPDA+cFORwReVoBwGL4z1YB7jCBCpjKaB3zRrfwa
RGXiJQqS8frHtNaJ6+jQqa6myg6v1UPPRnEyPz69WyE5BVJOaSftCoixCtBI+Fnx
hJDiobd6WBzdueaB7Qc6W6tS79C+F50dUzbHeZLQNRXHztZX/H4TyJ2Jz7Bhy1hh
Haa5mIhgjdV985ZHUEBXIch5x85lmAjuQPADEi3chw00idxi+nbq/exCmsAxj6JC
cIuVA764o2gftaIAEj94JXMVy7Xi3en12L8wbUezyFZGKhUxwKilWFhvb3or70DM
yT4U/URV1HgDgeKAYOsAkTeSAZsK08cRvhxDrpLl7y5wOfxFkSbN/04KuJyB6YBe
Z/aUF4VZeNeg7fEmpW6XAVSorFQ6DgMLmY2TyIh5GswHwfcB7tqgYVYSieRM/ns
GZ9hks9nsg6NlaL5ueYYOyGs8MB50XHDS42uK18fvRI8qA5liX/CkCdUJC5Hlu4i
lt3BXM3Z25iaYaKEmosgNj4cMdreoKFckmq8nSBdeZdIJ0xWX4/ioBdOaQRTknIV
wSSQblutN5X/AZmKnF/65sv13IgnqkLQIbFCCaD2IAzS5itRuTcbK+KZbKSClNpg
U/qYmuh0TDeHHM0126VEPQXAQnxvtV/0MobXpswmuo91PVsbFgCU2IA0JDILki/a
xwacsoQszTnw9qn5BVmIodbTlBBfoDor1C/C2HrkeD/J3+jSX/35Zbb9GnuLn1wU
j/fQaGftHgt63pLqqMycYcVmiA0quvpMZyRmBGhHPyr+TcoLzkFNAsNswv6//U6
hxWkF6SAaIVWF7hTaePbDqIyeVlm4s2S5QhJw5IAsQokxff2C9GZTLDJpBlKv7oE
r3HBtOIs6Y2CzkCH9nXfQvbw2LWEgsAgq4dLk3Z2NRct/LZAWF3E5a1wW4YRRH7j
Oz18aACWB6WnKnz82+1v2FciFB9L8b0gNwU01u7sElayC2TQGzXAhu0riMtqBiJX
bLmCos3/VelP2TodcI9HmrjSPH5HOWNP0h3M7VgXHbohM9FgOZf+0GNaSI4Hr/3X
nvFuT6JgJUS4Nrq9uE2RpZ1XDvLUVrwe77tnLaqXmbLeHm/V/TXviqaxEEgtCSba
iWgsWkhjk8JL/Oa/HBSA5mhf8Sq1ru46/sJXjRdZ1wXGEVmCoSkJmgKTn2a/8K1g
XE1NMeTFZucz8WJDAC5DFvqthrHHcAcG8YMVTE4EzwfTrYe9dxfhjILMDjp8A3Dx
c7t1m/6g1c4nQTI471xs28iOsRw3upKY1T4S5MRqidQD2yKYbBVp0zMAwsybbq0ay
Bmugnz5xafztKADCg1mgQ4BzhXWz+0CMNj4txId3kMwGt7Qi20RDf7cDrv0S3krh
1DGWGS13fr9aaLiSh6m62v7hg5Jn4w12yXEGxAPj2TXzZwVGL9hmzbghxt8pJM/u

```

HR2vMKohagn56K3xIfwi8QrWDBr9r8OKj2Ia88v2i/QeQe8CqOVu6yR8xAxGQFiW
mJZ03enMP100rRF9wdj33CxaF2q2kVysid59tPfJanTHUYz+IFV6/NsRfMgye0gV
9k/ebq0x50IIjAjl1fIFj/jyupnblUteAILhvNBfkqiDkWg9Yhq2MZxgIGuJod
CLUq8fWt0iNV6WkStHZI402wMz3ek4YuIfyVrh+oxQcG6PihlwEu5wamZb2g0GDP
tqa8AD7v/mez1HR4a2xogj91DLz3RXH1RYOQHSbvRebgjZrntOG+gidcbQvsB2e
aS5X3SZXYQ0hbG4KACkWKWTj84Jxflp+KMfdybhVz9HneTtiLmsvlibPVj54ZuPc
YNmELThyCxjlsX6lmmtydIAoitzn+YrB+MWx06KnPbWW18AsH/gWNX0qtYIRxJjY
rZkvzOE0UgRBxdWuK9F10cbAfq6S3fIPMJycTlSa10A6ltq5XtjfozA2ckRutqV3
1n+JM3Lo55CMe9igKfi4sEuIPmFjQQccxh85PMZKXZv+k+EU/PgD21HxWLbply1n
lwSllaTC9kNAPlcvelROfuM5jqilqDF6Q6w8pwem2m+vUc0aV0CBGJvvz8+Y76Bo
fho7SD9SeBOnCsSxqlcOKaeWP110Y001wUfI061oTbSya/tbNGGaE+pXzIbhKCvv
wOTZ6t3+12dhZ0mx9OzolpxslASescGr4MDQePR6lecDPdgU6cJZMCzMiKrbZC1M
lFlApbM5HdkJOG0AVxHvbBP5u5SSfu5GGDcjiVp27A8kLGB1x1JkFr/ayVqyi0Zn
7QUQu85CxW0nxqFFkYxXfvWVpPvbzorPySEntj+ZmwdqB6asqBuHoW+WEVf/U4Sp
7YZ5c4Q6mP9/HZV3J+1b+BaFuuRQp8lwuvYuITRpobOncr3+U4Pr77vdBbzYFm65
kR5uZgS38rm3DX54qlUhb7AeWPNwqtEIAJA3soThkk+J4/GAIDM46cQaJdPfXikq
AuZkkSOqjH0qEQR2gprYNTTakISQXK3os+aSrdScZq87W55RQ4bW+1pwZjCnlEI5
zTgzG2iWGCAPHZvoCV0cv+Ln14a+rplNBORDHhDuN5Vxnd8R3QFz7iL6WOW8XPuW
VfhiLZMHR8/e0rgqlF7nEw8B8XYdKsPRpYDnrjWOUA=

```

B.3.9. S/MIME encrypted and signed reply 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] 7845 bytes
  (decrypts to)
application/pkcs7-mime [smime.p7m] 4806 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

```

In-Reply-To:

<smime-enc-signed-injected-minimal-legacy@lhp.example>

References:

<smime-enc-signed-injected-minimal-legacy@lhp.example>

MIIWnAYJKoZIhvcNAQcDoIIWjTCCFokCAQAxggMQMIIBhAIBADBsMFUxDTALBgNV
BAoTBElfVEYxETAPBgNVBAsTCExBTvBTIFdHMTewLwYDVQQDEyhTYW1wbGUgTEFN
UFMgU1NBIEEN1cnRpZmljYXRpb24gQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIeh7j00
Boq0MA0GCSqGSIB3DQEBAQUABIIBAGN10aq5o2OJUxeEgaKipbTTomG9IBdUTU2t
ZdTEG6d1H4121Dz0Q5zSqpMHqbqb/HQpqERcNiXtq0vu2aBMF48OoZo085R4kh1C
8uARKo/8CACUANfGIjje+oJPw1o8eaDT8CQL8/T2TZ012rfdQahxsIAR83/tFQMD
5EqnQVxHA9IM69Epdiwk4IrQjep6djishGG61WLrc8tbIXgBM7QHkdrEA9yJuWFp
zpnGgYtGHi3gPzE8H4MJK3hnZ3uNAWqHy/nLUw/BwzD6EOKM5CRoSkcwYI0yAYu2
zGrO7E5fvoqFFzBsYJp038zjw95tEOGUDeszdrGP2dPg16g5AjjwwggGEAgEAMGww
VTENMASGA1UEChMESUVURjERMA8GA1UECXMITEFNUFMgV0cxMTAvBgNVBAMTKFNh
bXBsZSBMQU1QUyBSU0EgQ2VydGhmaWNhdGlvbiBBdXRob3JpdHkCEzB8R0APhiY6
HGLS64MvlsDXhpQwDQYJKoZIhvcNAQEBBQAEggEAG67SjjL4JZnZLqZM62xH8Cdz
SGchx+DpraOfE5ehEpY40Jy9j8sF6Wu21MLUNRZHQ+pUlnky7tA0DCIwcIbJlWV1
PHfr/M0xf++3kfnJBFAjiGzp1ROhtpeP5p+qtky9VLxoArhI071rvEG0Z3u+6IO5
Z9OLz4jX51zZvi6XIQlp3wtBxap1hQ61BD3DWX3W21CdKw0mKPhHQLwig0kXFWUV
mpUs6oJZV3HlUp+ifN6znQJVWjDOAT08d2Rtq0y3RGvivEWB6ElLpy9vu6a6JWIL
1TTb/owfsyochfPx0ew4y/edwROayHmScjQ/ysa4ee5ehFnG691E1F0hKXJLozCC
E24GCSqGSIB3DQEHATAAdBg1ghkgBZQMEAEIEEOh03Ev+7rYyNOa01Xc4M6eAghNA
Mmz6xWWqsulbcv8u720fQPHh5AMbQdSGBizaAz6UlatqA09hSm7bPTSZH8HsuP3M
psIfwjyqa+yeFZqhx90wHClCMBJJUcsKTnzKjNL71IC4NntenOTMTkIuFcnWT4Nu
PBmKZjCN+8XqeJKQMyRT93MxKXNrMugo99jEnDip1vvEk94e9sp9F2tfBbAspxjw
vyDpJWsMyylq1GPFhcepiQz4ULyPCIiFgV/1xfvqu29NM2RC1WuBPR/2ro0k12Z6
f2idvHL65xzqBxJf1APVZLUVCkZzdPMGxgk7H4a/NULP7AL12zxMqIzMXilk+Hp6
H+9igKeybOrw2qYbZea2E7XxCJinzqdMpElS3ChlaRTyKCLyDF3mKnuX9IMcs1Y0
08dc23dKkLW85Vyc6MhZXPu5SADPFDNj43GGuyy7L6hqLNfr1vLfYRLGfiPwG1CU
KKVOvFNxRUi3bJq2vMzmErG/mUkMZxWiFLEv4pj3eT+5eCrlwNqYgsezECwtcYWw
KZrDYEp+2oG1Ijy00XLqWJf+aqAaHPaAUkyB3sbvTvCe5OwkV286LnXebzCJSCU
ZlpbGV81YlAXQQb5wKzKGpMn8jmUVOW89w+1v6zXmu8voytNqowE3QBsJHzVkBvO
6RJ0+CkcRRC6oy5aOPP0GmjPBxodwTmmkJYdqUqDd7QAqkrlbjITjFMgs9NAocrP
Xyzjkt96178e3BFIAoY34JCZa2rB3I/dEFpPILq9bgR42Jajc+hhikFB3aQm3Gx+
ljZHAyDjJbSNGsMS2k9Vv/yNAYyy2AFA957mhXHX5mIKimD89pJAhnfoL0D1202+
F8eDfn3nU81gOR4rTWadeCsTbJnWtABBcd3Yt6AZ5+aF+OnnWKdICvc4UTVD03Z
rP+bdla8xvTY2vkB+12P4h+NVRrXszuR/9z1gF3I+8MRbpANT7zEr8Szs3RhO69R
N/Kz1nro4K7zZWh5xfkX49khdcOORwk7ecBojed8JJXS+hi3D25BA517hll1eC5s
a90m4GVcZBvykdWd48VKmrdYttGAZTyW8afjfsfPqN6Q21vHuNMec6MM1wXZsqZ
nacCtsBJhfYrmOSdsFygTxg5Qsw6OC8EqFWZHtyUuXOUoFM5fJGClnXCri/er9xq
HybJe5s8nbyK62Kc7QQPPU3oPMN8ApyvvL5NrKKF7OxFPX3XGIvalVAKgoBUSWUa
44Usk/YQdw+3VngpUi7QJpn3fTbGMF31LHxoztmn7aNZ8cVDdidme096V+gbbUpq
ay6QKYXnOWB6PclfewkY/G5ETwdRrB6jtJ1bwJ+0b2LBD7wU5cWBd/MeWBzY14JO
ZfpwfHuw3V8VQasdCVmqrzb6EA+8NLu34NxSzVEejItaMz2aawCHnGbHnY1ThvsK
JQ3/JHBu44dkMSFPwiUOONdlwal7SBtjOOnjFnz2IMGpkyhMNOjjw9Jq7CUMPGda
mcfhosZo/jxC+6ZrSybIJsmOzyasMsXxgRUvjGVjf8rpmhou5ThWJu8rlfWg6pia
JbtyomU4c48lthqN/AaZNkkK1UsZg5uqHp08jnlbOFVgREb2bOWnUzG9Y+SDxWV+

0IhwAq3FamYXMGAWGkmgr6xi2EJAXLPe14qy+p+GzQ3wEHu21BLRTiAMpgJzqsXh
toundiL6k19C0g6oawjx42JcOHITrjYtO1ySkFaFiynKT+dvBV2rNigWpUDrTxJv
308zWn9sGTOo/iam0jSm0V9J0HptLw6BZhdqp/iZyre9wwouWP4uKzAY4Vi5c1lvu
e0KMJXaMglyKE8D0wg5MpxKPyloIoSXoMbFKh/hAjZoxQTgotxoYMeGLE8FOYw+l
9pZSm0EwtL2ImAA/qyDp6A6245mc0W46sDE2vUyKMfWPNVFlnwCFackni/Rzg36M
bVbxxpxGTy8GpSm4z4RI9EwwhbrdgzdyFD6qC8kXXGuXZpQ2n+elysdCmPSLcEy7
t0aXFBnyYMOI6eCBVNowQiZrQTp5aHxmxRgfeB/Ee45dfg2jvdryr7Cz6N034kad
Qv8gXyMx5Jfpjb6EIX/kGx1iFbMFkUNB1DAVO1gJkL9mvsNa3nk1ZA5u7StcCRuH
z4Qq0ST2uEkv00gT5UKh/SEW30Eg9AKf/G2kA1+4df192P3tP9JrJhFuxtCgrY/V
Q9mQV+R5Mapir1P/OAmdMogkgktmfT0/VBUEup4I4bL6RTGr2Hs0KYUzcUNEbKEo
F24QXk8dri3SZf5WtIYW5cGf1DptkKoUxGRsS5UHkfx8QXz1PG6PWPpHR9Gy7SDQe
FvDif7tegV8l2O7ak/v6TjoSyqXTq5IBjCpnmsHNoLd9pRVmfGwWzh6aL/CyeMqk
WOfOkIbKY4FIJUtu8dZmDRgEsq3O7cFnRdffFwAwodbrC9OAXdPHlpjAd7Ev/d6Q
F3YRA5ndYXDktkUWOpPwmooC07cKcYQsVFx9FeIt60Emvtd1+XY+zZF8i4kC38uP
sHaUBNYGAILyZEyouBqEQyB1lgcl/cQgx1c81lizK+J7IX1wcYgmwq/jrpJ/mBeUM
V3P4N1Hqjfh2yc7fGnVLE86barIMsqtdrZ58kMLdZNiQiwe9DZzOWmIx5BSrqWEb
tNtew/8ftKcMHRfAMyBkEAROWyyTty7QkvWlmAWDCGvt8rVuWIWlqk0gp2zAtMR
Fao6Io1thU2G8nPdEd0ntVssPQMmlhS4Bf16UAxpXUJ05KKgtyyzqxqWe+jGenxJ
/qulJNzhlgjWpUFJ/qnm7+Vk0W/HFvwMiY95Jd+dAxhkqhk69PKVpcr6uBwKJRjn
IHgr2jpEoySwZnK1R1ZLMtiKEpc+sM1vnCgf5qIAUVi7WmSS7WxI4h9OUdTVjz4/
bjmJSDI7ekPdvoD1P6DvS6atTCgu0NgxkG15zSnqOD2q5+187MGOiV6IL2vq/0Qk
oqwpKn2DCzLkO29XfVOPCZEyloJaufBlXWfqJIBA0EK9hQafalq2ObXwQ9VT+JQW
z+y25MbD5x7E7iqTTJPGNG+Lc1KVPVuryLz5aRsQdIa4AgvTZ0+Tgy2yt89tfZ2v
6dQ+VfyWtShtwaympKDDGRKmk5qlhr88UkI9Km8dlbmTicg+94+ot20tJE43pyDp
XpEohbQIfenTYtkkOWRr+7Q9XB1q49FMBTF0oMv+ygcy2622WF7cSIFIIDUzh3UQ
Ca17U/TfyKbpYxxeP5psXEI9q0fZg1N2Lc4CgyHt3CqTOZbie3+Vts13+YKZyJ8
Fwm11tLEmW67hezntqgf3ndcB0JRvoZSifIa3NLdYANJG+70yR3lG5L1ly3kJ3w3
1lKPCvy4theKWfYSAHxfx1+3nnPLV6PF7Z1TAZaHRukvKtnSOWb4Kvd06UWIN5Wi
GiWjiYQS01VDdq2CWMQ6v5QR+Kit2lmse6mxwHg87UW6TR8FPsA4F6GBZA0W4IdH
vLV2AjVR7G8UqWkcv7ET1/dE27daGrTf7Z82c09x/9sBuFXJk8gx17/rn9aOqRF0
2SY4CrMACJ8qnu9aakvtU+vN670pnFUAbOIEG66jJ7Wd2SbhgXDOUmThzoZWezM4
IIwVx1L1LxqF4FJvPEQjI32UcoViUU4GkG5SgXerArXeYKRwRGOMMoNccUcar0rm6
JuZMU58vcP9Uhz/HaRtaQUWjwG1N/I2Q1XJPX+Tzy4c3ae9pcoKoOFFL1VYSDLTi
4KFH5ElGswcW7kHfsibCxrZc9Q3dP6bT+YteuGvbbSHgP1YFp0Iw4ok4Dzi8EWGp
6KvdCH5mlqZYJgawSVISnxLPLUdbqY/49uExMm+HcvO1fXNcbV2SF/KnhdJ26w5y
VcuMB1/ze/mG9MAerxoFBRI029SRLhe39zsK2RNjDXDEi6R1q3F9oTQL/rCuFOG5
Cr1/ogQBFihF5Gyc8sqmVG6/f+p6dPcwHAX9US/WGI1zRR+qZ2TRW53zfe4CEgvi
YyRg6anigaS15moIjoR2k7ieadjMPHw/zDI1vTbIjR10i1w2e97yTT3o7dvjAJQF
yJ6tcnCP7pX+WC0pEYF6LVQiIs1xEZFnsnug22YBFpYfyxVO7m3H7LT1ZFjWdxpm
5JElz5wqdv7005yFo58JAs8fIpcD54VLQ9czDPpByq6M10JmasVc1EmdG98FgcuI
jGycJv01loomv91iojQHTc3mlfCDrPcMMDeELBfoeP5Xpd4ZhHOBwx/BjdUfQHI+
DALW+hazukHzcsCamfYh6XfffbqXKBg2r+4An7z2Bnb6xoQRB3TW4yibQ5XhDasi
kXsJ3m7Rx1Ja/scA8IqeEKD3xE2KWfARGBA4QSxv7/r3Q7/PHhCiBSQMZuLkPAXn
RDDmyHF14F4jU+L5zsrvy4qJ+nV6CwPIn5Py+6LuUnqe/ZHZv9MzsWbhbaChY+Gb
uUYSfUVGBy3pdVIBiHymgmpHj10xjDdD15WGRM8sI4yG0f6L0hCSm/fD0cIpihDZ
HMikn2GaNYTS5A50+GKRQPFYnm+lKHN/enyD6vOHITFgqJufjk9TtFD61t012kri
O8Yx+o8fFvFaeFUBaTWpPMi/ffgZio3ih+vRQx1MX2G3JdDolPPuRTR5ZbH+a3f6
aAueSmT53IFvv7280mVHUPN0VtjqHdkOT8p/+xVy1VwCt19h4xCLSQOKwWlZvEXw

W64AQfaJ/tELAdB2k0l7tR04tVlt0c94hgRld2r67TZzzPC5y2tBspXL29SabgtY
 CRCpmaF09VIH3o05brlBrj0glYdy7t6U+TfMDunWiLCDmYtweCs9kGeESiruTHdr
 GTiWBojp4HAsGP+3qYD0nfMXKELYgaPC/xtl7A3ON5tR0pwDkSckCzrwHYKWL66X
 KGOBDPW/o+Eq9BjwFN4n4lP4OXlcmGQqGBWHgnVSldditTAvFEEelpokqQI0G6Cf
 9/qeVR/kgY8/YmkwfSyL2b0xZMI1Yo7S54irqaP4j22vIKWA1RkrH9N0LV5sXAZy
 XJxZVx0PCOFQVnyJqCNX29qfQ2j/KLmHfaK5ZESCdUzyvPEQkxt4NtQT+tGuJGBY
 sWjK6jVA+CRw8xdLFZMwEZgoAhAVdW3bl75BmqSGGs72LvKs6535tfwsXMN4YJSe
 x7Ax4n9HoH9zNsrJ4sFCsaI2jdGY5cj3XjB4oNcjutsMLj0xLg54wo5AAEHik+4G
 qC9KPLpWiE7XXQFdUsMfByfqvFlj3iRERNdWCnhxk6xdXk29xaNgLh+uAEmG63Qb
 3DfVqsaCTed4N+gNf7sr/9xJ3PojwlcCXfCi04h7J+tRw5m3bdOyhibVerHftemb
 8skTeC6Sy27zEmeBj9suIyWeruTTAd77XzD7y+py6Mo7k0PV5nP7anGbVeKIZoSe
 /pLC7TzSoaEzR/1fYia93Rz7ZD2weqp+j+OUgCipefeOeCs7nwPThu/Qki2Z0cki
 F/pBP0xgIl2RRIPiInSWGq5WzfmdUo6BSkzz0PSJAa88yac7/Z/h8+rca7HGZzbB
 h1Y05I3Zx2oI2RxDWlZS/x3ZEw1Qx14PNzpfKn4tyLI fCk02fZoA2Yeb3s+NwASV
 SaSz95eSz3gaaa7QcdwvXjy9Q9obcuZuQt57NofpkeL9R6sv1SJG0+3W/He8D9q/
 yW46YufMjtUUXCmmQecEBvUDNkr5BdAfAcpqtvEHx8mp+CKPOU0EfRaXC6+mtzYD
 lQQBHBNXj0HwiKEMCmdJDMGv5hTwxLFJHPC0u4/cZLhebSqNxLM8siMH3zyua6z
 L1YWygKvdhf09syokQVndzz7M9rz8pKqv0sbVP3nn37Pu90jpEphZnY66cPbIQUR
 BmjA2DLAImK/u2KQEtwniIRYzWxmZxw+hiVMBaWHhmY0Dn5K+v3LQlnlUeIR5uwP
 /gdCM+F0Jy1FOPEfso9V/dVPa+sgXJc8Np42PGmgnpNUR7+MMh1EQ+1iNq41Yuq
 AsdKuq30cRy/5CC00IFz5tKDS0NpLKjEfa+LuZzPXd8i+MLthWEDPsi9/j+kwgjX
 2QanQPnMj2kJ9s15K22nMHTZwf0PI2B/3m3ic330yWaDJPm35z7U1YimwKLAPsg3
 91JJxNt6f79/cqZbGOau0lnffytR4/uSyra7AYmGUhSDFnd2FEpKTtztURPKviy
 kDHUtu8OnJE+0jJrg6HIxyf7NzVhgYUESyMFyL+MHEbf4h4R+DoV8pdqVhJLk5Zu
 Rtfejj0y6g53mq2e26I3y0iu9P9WMBowvmx3e5q0u+D8exIIM9V2aKfGFS0qynSB
 03BpRAofu6fjzSN6SxCaG/lC040NIegIf+FXcehxr2eVV9+q17dvc/bwOxer1bV4
 BBvuuRy9A039kW0B8wCQDq/tzAIjxItCTM2deFxlwB/fAbbIG+a/PVBxA7T+aYsF
 WGoNCxoFYe3TYXuVdp9FtSVlKIzW2E8LTT2pUfs1a7U22v4RnCFWtcjubRkaicoA
 eI5QRSnnESPlNF9Ci9TufpUPOxjOrImfoChuCftBoUUCLSKktXKzICP3wrRt9Vs
 8b8gb0Pg3hx5kSZjBJQ+yCeerDGGEU9eTa8lsJTEitk=

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] 7605 bytes
  (decrypts to)
application/pkcs7-mime [smime.p7m] 4616 bytes
  (unwraps to)
message/rfc822 810 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

MIIV7AYJKoZIhvcNAQcDoIIV3TCCFdkCAQAXggMQMIIBhAIBADBSMFUxDALBgNV
BAoTBEIfVEYxETAPBgNVBAsTCExBTVBTIFdHMTEwLWYDVQQDEYhTYWlwbGUUgTEFN
UFMgU1NBIENlcnRpZmljYXRpb24gQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIeh7j00
Boq0MA0GCSqGSIB3DQEBAQUABIIBAIEzSE7YJfWjy0TMQGEfYcrcBw2uruGZw+/k
QaHXEcEFdwDSaKvAzEFoNN0xMpZ090ybC5MHqteYMRpaax43TsCnes6XevL7o7FV
gSMI6CCnmV1Y2Dvj+oGPHk1/ZkFRPz+Hsrnvl65Fs19thjbtQ7LX9uKE8TBODLRF
nCnuyDdHx7iDJGI6xepIvD4M3zaUwpNa3fFi8XOC7UH7br6+UGCRQCZl9nrAU1W/
VvFRt+6XSWX171IU/0syMw4ghwS2tsLgZhIrDkFN1EokgVR8bDejaV9px7jH+d3m
FJ0t4hBjsZAFnggaecXwoKUaPqlj6X10e9cLtqwr+26h1TmA8X0wggGEAgEAMGww
VTENMA8GA1UEChMESUVURjERMA8GA1UECXMITEFNUFNgV0cxMTAvBgNVBAMTKFh
bXBsZSBMQU1QUyBSU0EgQ2VydGlmawNhdGlvbiBBdXRob3JpdHkCEzB8R0APhiY6
HGLS64MvlsDXhpQwDQYJKoZIhvcNAQEBBQAEggEACUHCgXEC4pKuedLh3CB0QLAW
HULF5htBXebTlJVES1voU9Smp5OkueBMptF18R2ojjM36C5d3xtdsBddVweJqNyA
Hgp9207qVoPyVXvp7BByoNRgZcrMx1pRoTREEjCX585MOXEbFUXRVRPohViZaOAM
dgdWFB02fcOWGh+RtwBF5Ege2zuJhTpF/ie7XIbN0LwsZrTDGdQ63VagvX3AS0m
TPJyeqUkstDWSzOIrOlplW/YjMcYNjDkygeNgppdV4SEUFYTNxz6rqql4E+a8LxX
IogOTMh2ruDPamtoAEMfsMvz9XUjSN4TRWXORLkzQeaIOjcPVjr6AHLJFG6etzCC
Er4GCSqGSIB3DQEHAAdBg1ghkgBZQMEAEgEEDyefElL8mhLfkZjaJQLY7KAghKQ
4f1OU+eyhjobu3iIzeCooq/a6JmdoGQbY6s656cODYMhlimkXQkRV1QEziLkAOi
aKPZy3zmuu00h5lnpduDqzFq16Clw8CY/99ep7I6vANjzmvh4pV0onCsR9GuYexq
65nR8oy9dXdCFP6vkGBFXcrTqnbPQrZF9DSxpXiicROjs5ybp8c1DbMJB0x9LQE
vHcxB5jaNGAsb+IVHZr3LjvO5V5T0/YsXn6aJXQAVU3b007iUjxgvGxQGSsShre7
F5qW99KiI2cc0c/wPtv4PyvgcVuLs/CFtvc9CfqbIAR/Vm4AupZUbaizLnpXSK3S
PKY01/8j0x8Eavv7LsO7R9WzZwS8zK5Zrx3aDRclXUMCXyQkel4nZvCOintGDoKo
QuSs4Fy3M826VYkKfc7uaVo7j5lzoSeNUeD0q5hpmrTnJ/ce8C9T0FES75jc6P3r
Q6yAakdLcsTL4XPc9Hi9stkX0pPrGYrK1HYaDBDBKZ92VdiEVGLX/41hltwX0f79
M/R1sbT4a2j9PsWKRI7Pva3L0nNGV0iajjBslyppdXLKNFBH02Vy4zoujccjj34Mr
SsrnW5EkoxUZGz1X9NAYV8N5/f8faUCYnSbfHg/QIK9WBKggCTm7e8Gq2iGgzVmx
Jpj85EkYXLDKs7tN4KhgJRp3ZYRFdRUtoq4SVNzNc3AhYDMVyBWcpDAIY/Y8ync
ZsHpEFB1Ypau4/vtj14MCjlIfOtRDf3oH7Z0Gp6ecWGFwkZ+P8muIY95FEfOofeH
gTzUi2M3NwbGVOSPpTMxZE5wesAvXaWVS2pN2KPMQLBXPVij7vqavbVdle3ld8JJ
cRJWxdVY03Tfe42TQRdKjYIXQmPrjRdx9d6TyyoZE00mGed1lv6Z7lxWcvGZDl8k
rMM30LF4IgQjCVr7EiAYIybviRYLNNKptCqLK/TvANteVYehb9yTynwevu1nFW5e
Uw3rihR3MJgCV7+zSvsjKHubdSpuu5adyMKfYpRyDQM94pKVEvEVxR8Ja51xyVB4
p8T3Y22rNWjlsBf0B7UAVqb/oDuN5oW2M8K53GVXEPUG+80dlR8r82Wq7ahSyae+
/jAZcaopN062hQvXXsIFj9vy/B2rdDu3hreUtFIjgLRcmKqmeXIvh7lcBLlhQ9Zm
EI+F7fIJJSynDna7PLsU0tANrE6lmn9XkdL9EVCVZK5LMFp8LtuGo8EMZ/MxZ2LQ

99duolum5gSBdZJYhrxb2rpmsVRrtLjzKCmywxOEBlyj3hYBNjFcdYhRd9RsMRgg
QjoZME5ovHdRyBABUiwOtyGIFD9rt8xNqjzHWEizeAzfj+WbDfWDz9qrys vx4Myg
scicK+yCWBwRvL2LbNb+uHhX879Ejj4zzkSlqDIuOTvGduojH+Ti6aZjEdnpfKGM
xHRFRHBI4hmwuiwzqO6h6CpuX/2aew8wByIAaomyGeTscBaJk0JumMxhSmeyImn
T9DTF4dUXR9cGES2qYquQcSSc2KNZpaRpVDNcTETNPLNh+vFUPJcv485g3e8EJIy
VS99+e2lECdjk+c+iHVMbTXdwSMEgrlydIlfrPCy2nwsajp9+4lhL2aPk3yEqSs6x
QHPO9cEKNuL7BG1Cp9wkr007CVayEWY9W0k912ARy637pYpgeQ/w3eNh1GjSuRK
pXZr7WWgT8MEuF0PJPOVWy2V49JmKjP4po+9/V+ewHievS/Z74/xozJnNhNqyYDp
56mGQ3FH5Q628WcPdk2V9h897AosHVYFrjFHlObWeUuQqQVctYqT6QtW/rITmQwE
85DzWoYELv6ng+IjSswQEeKFm7UIbz6UBPe5IVYJaa6nAXV9Ir0ErT0A8QLN/Inw
Buz4RnznGuXNgm7mONvWZrYnbwNKGsbo/LsMsKDmlCqDd/CRZLP2/r0mgNld6Iqy
wuFfFo9M18WXUY3veMD4J9+i1sm08jMQfIqKgBOOczsBt0sPn2yE9mgcsDgud095
jFz4g2E8RUSRJggj/av9nM1lSCYjnzKBezVvM/S/qJmGH0l8RbYSZlZBIJq+xxAv
xGKG0oNKVzHe8VtMUwBbi5k0Ox5oTrvJ/A3s36MrE0JlcBKV/jMMt2FyDE++PvFE
0X0zf1YsK5281jNBMBIA8GRbLb8+G/6q5RMf/epfy7c4oJRpDb1PVhSMWXmgUNxc
mLmCftewVJZvvtUu0WWcVWZ4s2GZOjtBF1qXcm8nBdY39drprA0pcrkL26XKWM7y
F+6CqWcGsMabViBtsY/BMVe026UCfXJfytMGyCeuno9d3p12VHCLM49TQcWIpZ
6yRLmKEYoXxvtThZE7WndatiUmS646xpsLmt0HpAhN9V/AJVUB5DPHDkFr75fWp+
GYsKyEDDIq/4U6gYlFkzWuNF3if8PWwT8Pbkia+2XWrUs9N0Tw+ugD8LkeobRw5M
gHcphVR6Zia3WvpXBe7u/rGgNqzRWHSDtT2UWKSJx32iPuQEVB7/KQNT6b1BhFrK
LUa6Xp1ZUtvdij09fNx9plaKquHQqjV00YTga++ZCrdLnEL0IxRMUbfz6tkf0fF+
gNnP7uaCt/1mXRyilDgb68oLxN8R/fCRTSVZibLhimWPRFXm0Qf8nznYR2+nOARW
K4SfFLhhB7QqsLHuQ6WB8k4vwewhAuNM6EDR9wSyp5wJ4/NRtwm8b+Vf9aYXweQ7
8n+mGBpKQBwStOllzU+pdDorM+jmLeky2hPVkr59IvEiZmnDQXdzEWZAVEC9jbsa
1lb8FnL6l0edbb1BkjfeaXn+hD3iRbz44vyHa/1/4fi717XNCyWMEL4Op/hezWdt
pGtexT+AoYw2uA9+qNkz70xtqcSczVkm3jWTJPLrYs1UUhI5HF8yH7NtbaySqPm
ybxysODBGfXz7qf/o/rg2SNHfSIcfr/itP0ZpnuHiCtFwIBYFLoY2ceMYeKfvrKX
9Ble9lgex4BtKL/uPFQopYWNPKAchseKIJzptZpPW2T37kt1UYzEhzieQpC6IDCn
qSZeQ/Nd56iF/kw78PQMDCGLdulJDh/nul8LD62GhCWpZMEGdxDJvP+VdycMEIkb
BHXLKlKm5NNAygyw2Wj6kiaPR3+/ZJBMuRzBFSxI87Zt/iXoHM9PYvyDcgjC8wwEK
z4jRNokSW2eSmgRp8ty0ZSWcgnnegymkRsYSYkIc7894qFP44PmYPNB981mLje3c
FsuvRcVny3r/KJ4XI140qbkYWwD8rkHbXohiYQx8N5VUqlfQCMYpPaqYf247fW1p
YJwOKXeOsJeiv5/uUiC6GzgUNABnBhZS5uFVKoCtVITzzOKpqAEFFMr6fGlnOMzv
Y9XwwT9fnM3XWB6RsXeHvSMKjQQXzOMxc23mtV0wse1Mg01UJVcLURyljWoY815F
DDNeBt5irzunTvX3eRCGz9oaJ6Dz16er72YqmHFyKEGFyFjCpOxMI3L1wZhUCRM0
MrsbtGKchcht9fmh2QouxTqH8T9r0vLlVrHyJhWwargNxQG+25ZPyb7pmBR9Fs+B
5PFhN2O3nOr9LbPdrDXxvsGexOwAwf5kp0LdM/8g+cn5qqSNGcj2jDagZ5j2IPbJ
9S7HmRxx/D0v5REfnwrc+WVPR+z83bYw1N6Ug9KB1S1lwE9E5DEUb4MWbnh3RCi8k
Uhh0ErIcBWByUooqZz1in408/ebhlPC2zYCOHqUP1AgVsycmvbZf68bHDZxJWPGz
w4EJYYCAF9DGbvaF+pA3TWnt7jmf8qLliwGCgC7U2XjsL6aTC1ql8QseE2OvvBLE
1lg4ZbXJXhs/rV9ZuKzzIE7MTQmZTY4923ROG/Bt9Bc/1AJ/a3e/mdYoZ+79TnQr
/sLP2FiqVHAOtLY8SQXnVP/Tes/Jc6EAXemoCR7fT+959WcC+vaow6MTngjk6JBb
YQUU5wNNF1/834tnvSLBI4IohjKbp/ZBqsctq6bg3pGb5MjfJgOxybX3G37CdccZ
yxd3N0+31XBWuEuUEzusUu1pqxK/TpVTcptV8IJJweiQjwYCESMsp0vHO44a5rui
WDiMaD0dgSiKgTl+4LiQsTTqVG1Hd3WB/16hUvIUeCmwbsDLZ7JZWy6b0PyQSQdi
AH2GwmcRRU0Kiebx942EDTkSTDudSCd8fcE9B3zg7VkgNkTRYHALUW/4kEm2LayA
Igg5Rkfe/t3w0wiDfiPkx6KZH//S5FpHgbFbPiXGLcKIozH0ocs5kT6L7vKc433K
es5nwUksT1iBdSP8fJjknUww179CqF5H3N00HUo3vN9Ghso3bvBvI0W0d84iuLk

70X098rJyQR8HBBiUFG6ze6ZY8hd4EY87dFY2/01p24iuQkLpXgxIRPmm2Z49Wvo
 2M1XLGIao+4D+sY3+E5RtOfjJ9oEUFZX1HJ5zjGB9poPJV2O/RSiRXpU4weIW2+t
 T4gvboMSMPZh4tccAsIMZxostc1LjBl3lrLzR62crJ0dOc3vKHhDrd9RdR2QM9yp
 ufaOAwJm+Ubb5+liqVPo5bwyXOxJZ5Q5cyBQRhwwFUL0y+tWwPmyGRlysoW+soFm
 w0NNGgn4qZFm300i7wkFJKlgZzo8t5d2XXx1yp063X6BYVLT+SGuTSNrpfk8MuWo
 0Q+6lyZ6UjZ5XLUgvyKFOyraKr3ETdfMCA/bDmx2FI/rFDhziwWgtYJpSaoEptP+
 I/+rZxfQEd1kzJ+SgvggUbpRXR6/UCHBcvjSnJNMyBRnjTU5j9FBfitay2L5ZOL8
 79hudV2c/NO+qTc1yMir5zQyYLFN5oIHUIOJRRTs1/kSu5Uk3i+ByDvAXG9nJ+I4
 t/zZ9FSvk4RatM+nHLbqQvA3lqfv8yoz9quVhEAMZRMticGWmwvPkchjZQdtzwTo
 vCKBC7M12xITparw+kZuD5tD2d62xn8vTAglhaFebf1I5N5dF58XgwOkqMeoYq+l
 mYNorq/q659Ac97jyJ35UEGsS8tbkWCahCj27WwkCcFnXMyfkRrDXasOyQWqZ8iQ
 mmZeVjJKrHNHAV5Xj8l+CI2BJlLwYyS/IwbK45UuIi1xcMAAx21J/HMk80Y8laDR
 qbbq5IPR2ndsYs2JYchBB06t4VXmcJSzK9Y9CFzK8OOOawFE3DpTjcl4ZCxdKSM
 MuTGLS2+ZYqM4buYp92HbeXBz+tjCaFp16wF1Pm3yRpm969smGt8Hhc0wkSvJI01
 LmFkXib4QXDx5ulHVDRH93B2tnq9kCG0Zs/AHaUkN5/TeFx2BIvMEJyQTNHf12Sn
 kF0+ao3jREVMhAadVzFq5Yvr907MFID/t29EEyWkk7NU1zmOjTzOt02ak040Pnog
 Qibu6gHHGFY6Aje3zHdIBEXnIETJdlvda//GG5ulfdb7bgJzoY/sdORb/U6Zy2za
 h1qJnifV7+0aTlaVDXD/F/Fsd+B8sK96e1MC0oB7YJ517ZxdZ09WJ/fNJaXBU1PS
 2065hVjG4S4XFYonkvE4Ig3OUntnwg6y4fx3ZUgUfo3XJtGhgyBIw6ZNRhrhyJHZ
 w89PxnGJpGTA6tDbJMUNSir6yvR9/uhgADhfVJsZdhSFKKre4BdDwn7gEtD3X2dx
 TbkFAs3TzfummzNH00C11v86RR8xx3jRGRqJLd5RtwaNUoTMIR6oFNx+1KOG/lp
 ADjBJU3otm8hC7Vp5HdTTk0mH36inha9dPTjFalx1OIUmj3V5icC2Z1LApdAuzD
 uAiYMQntZJGHawGLKOC9UspeMgmUiblo25gDMYsuG0stOfQZjQi9EQLQ2xyy4Ha
 RIRsLm+guqcYPQJgRhAOExlowEGqJqYoR4rmps7w/kAW7TrTrdXeXHLBbvavGtwo
 rt0mrTfHPhPmsYbQz/4T7Lsm2k60TjGbSm8tGgBRydJI5ly45U/FpNXVgykgXBMF
 P+hJLVMvKgHehLCoxn5sBE5Zzf8/PrqZ6c1iG/iBXgnbMW0+yKUQ8sVLvp92YpY7
 hKplcj7RKJL3HBxzUeuUhfGFaiq7MgpKml8vgnFXJoc/NL5N4eKLzn3TD0q/Xhid
 5lpZgm3+6c/mDgS4RUIqtHaALsVQhoMGdrK2Trlbi2VoKIhEong9UF2WxQJiDNhr
 VM99rYy6aX8H9bj70xYG+Kt101fEjP0+S1OEfxeLCEi/DSHQjPrEwumCW2dKz0Q1
 7G2u+qo6Zcml9eJp5ZX4GPHrlImX4+ngp27/cNDQML/pHZrTbT+h2HZiDObed3if
 Lj/pAB43Snah9bg7XoUWOE51NQoOq6uSG+bUFsuuprFeekcs850DtaryNWzpi+4/
 5bScqoMawu64YqNq/lpSCXImEEab9nXtn6q4aPjhKHEAhWD73YR0nP3kV6XUnlyF

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] 7565 bytes
  (decrypts to)
  application/pkcs7-mime [smime.p7m] 4592 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

MIIVzAYJKoZIhvcNAQcDoIIIVvTCCFbkCAQAxggMQMIIBhAIBADBsMFUxDTALBgNV
BAoTBElFVEYxETAPBgNVBAsTCExBTvBTIFdHMTewLwYDVQQDEyhTYW1wbGUgTEFN
UFMgU1NBIEIEN1cnRpZmljYXRpb24gQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIeh7j00
Boq0MA0GCSqGSIB3DQEBAQUABIIBAD2qfM1qd/wlIn5/weLGjTlvhLXq8DBtZlBx
74LEO41mLdlhgnRYsPIWC2PtjkC/seobOuZC+CV58bybhtZc98t+SPFhw/rCzvKD
r+TYWJWJ5klGojWrmZJXuXfUA6GW1KvNQYQV2xkntNjeOe0dUY/UwXDxNv2hwOSz
K0MpYY9/M847oDrGiWv4xDqLd7WrN+ztQiy+4b29oA4Hy40Ll/z9o3yNMYEeZ+ZU
oICNWAvSHhIHuHztoEhhGI0lwF7KFpygyjP34o5oC0MRfwyUPmqJEuj+/o265hfj
zKAZd20Dh01Y5f4cKRak/Nq7j0YAVUMftIn6Z1AI3NBdqAuncSAwggGEAgEAMGww
VTENMASGA1UEChMESUVURjERMA8GA1UECXMITEFNUFMgV0cxMTAvBgNVBAMTKFh
bXBsZSBMQUU1QyBSU0EgQ2VydGlmawNhdGlvbiBBdXRob3JpdHkCEzB8R0APhiY6
HGLS64MvlsDXhpQwDQYJKoZIhvcNAQEBBQAEggEAmcFRU9fU/PySxv4kLIQ1zBV4
nTTHsBv+6RGYcEomqToQCdNyyQie+HqTJh6M2/Cc1sbRuOVsrfhJc0RQqKG2VOa
huevYf4E/x7+3Ap17zzg6rOUfi0rScv8y5PYLaHe3AbZvJr/ilj5YKIj8+D6JnZe
WxSSPZTDhmnN+oTtePW9v+hfq6WomQ/VnUJTSQNUnkxTnhBK5MiOnwmIYBpOD5Z
29/dLzfgciF1gFtTdEjszQ05IkVB20IvP2hvyaciljfkMFXS3302jAuxLSPiAQIK
UYw8JQCLz+TEGT7jr2XKXTQqo2yv3dRTB9Y4P0/MglX8fbzqWLyOY94hK8fWMzCC
Ep4GCSqGSIB3DQEHAAdBg1ghkgBZQMEAEIEEBaBWCdD05Wk7rGu0j8AGnmAghJw
LWbI6Q5pWF3Q3tMokfjJ+6dzF8HNZm7De0S6Hu3eU/9w7ooJDnRsWbdr6B5QI3b5
fsXYC3Vfjp4iYgwikm2xX4AXzt07T4YU12V3yKNU5UKPhRLrbH6zb91+ghmZ3Nor
yEWWu2QuHVTg4xsCaEG/+LX71k2wJTI6Lk4QDH150yIN6KaivSZkqjN1160gQTp4
/0YdExevb/K2WX7w34kdq1KFg0Vju2hGrnPMhgpvfuzkQirtFtZ6FmeUXWm13lX9
Guf6GeL6F4r6aZqH5gz1JUvH++3OC6bzPG0MdkSVo5hELTYRvfclnSLbyYcoE38a
v9aMD1Rv8v45Nd3eCxC1G93Vh/EP8NOS02geATE0/mNk5f3jsZ9iFZOdRMZ+jVuB
100t/jCj9PiJaxLZ4+Vf9qB2CJ15PtBep8CfhN1lmGU3Z8LJbPAPUpRW/rzmTf0P
JbGJzL0mU39zRnEoIRDAFAaqTj5pVgqWiYVJhKkfs7fHXD6hHM7MXqpQXtc4KRPC
UJWjii4DhyEEeTscOx10QPrGqST5nNFbc6Hb8qFKc0/bIE//QGz9rGerH+cFxeKa
sOkevWj7Gb6EhMu2aGJMmnqoh0pNj2bp/5vZ6paFmhn37B89nJhLXqQeDcgg1mA
f7DzuAAN5CSw6KmiJocmaYe6RHZjCEZmILXHSRJoDoTEIIRQiV4NNGxah7Nw3gaw
wwASKf+dhnm6Kg+6y1mVIIPdgW/CjjLSUTvox7WeKdmlX4yjmJSASoCJM3NWGW3z
BVDdY3nxkSQ6QcpaK1N57MpOmK2Ejbn3ch8vQuj+croYomR72zD2mGNQ5iMzcl+
US5jIew4R49N1TavwubkQKXtxl6WnUgVGLFm2d+J7zGWT6tw88k7400ce8UwVpu
NBZduEjPtYnsyXIRxL5tYEPqUrSbrTbsK10WesjpTD9+i+fBqvF2Y832yXQeu97r
9JSQ11Q6XtyvsmY2lM5ahdzWS8cz2WSxMmJgVyGK1FX7REPjktHf6dkDM+GZs+6w
SBhDu4Lyf4yrtiwuNsoFlqn2rdhngQAkjshzsOOIcoctx8ionRi2p+nLn963tfZ
kYGcbbRaDs27nMBTFcncLpXFq8Phfmb6fI8Amv4JzptPtqnwU/ygonOdkKoMrqf
DUXXAJ7r/5otGqc/ABjuCOpe7TeAi4JZm0nnEnJM1SvvuJuPk2cJ18ippjYIF1lf

zkOU3aaxJtQKofPszkX6eBEuKWlTo9rlh6M7NqmZ3j9Q82SA8K2W43q0ImgYnded
h+5i3siTYThrXwSdN07hKtPI7c2ZE9J4ASDtTmWNmrb2i4u9bxF3+IG1ze81VZU2
WoJ4mqqsBYOE027tKn5IWVGKrCgJlmaKOCeumEi+iICajyyYOXz15mXu6Z6+84uDn
RxMCOxu/mualrIjt35zaUVuvkhMMJnkRijEcdbHk+ICM9x0DLnRQruuY9Kxwjgui
c8YACZcQf0SSMyQZTbMfJjVXvp1XUA0TqF5dCX4TorUEiWy7pclCmBvvAkOADjug
htFRym605C5HtjmVQonQWL5c5e5z4+cDOISgdkaEvVCqg0pu+MSvMLhjiqoQx7dZ
Mov5sdbk344oo/G0mokjLT3u52mhM00SighMtW+ABfzwBE16DP1I9sC9Ge999HsU
EU7hw6vEOIzMS08hsKTAcEBwpXX0chlum/emFkjglVnXgHGxYegMeziqQwkgaNV
UwuqPnnrFIce4xu7QZ7pcAcpcWVLUZhEtCK1vh8QPUBcdA7CSrcGWdXuzEZ5V0Xt
LpF2augMYQ+a9XFQjm2Lx0UZErfeN3plZ+1ci/ltQgVNuZCPABIFNedZpEKtOfR
czO5y++dqglPVOAdAP3bhY4cFSFfyoeOttJo4Ev1kph7Cgp9s1zR2QEUrwahlzMa
4zyeqnwomcZtbJfFysNTlIOT8FeRrynOImEzaJ5HoCRviceBUB2Y0X6uFcFllydv
lpEEIBfoI2opc5Zczm4x7sr+MUAAGbvVBRoXTn8L0r46JILp7hVY1Xt+DeoR3BEt
sKKSE+q3uuGbWcmhAxeoYZEZwt9VGFv5DPJyhugkn62dA6P6AXPHYf+NbIQIh0oM
HFRx+3xZwluTmCq4+Mf1LFekGuYenQnBEySm7ps3aLRBxjdKTuG59Z7nu1KIEljg
nyVhQfyDgyheDLdf4EWpb+moqjmfKnWlk83KSMRLR7v8EQyWYBO1jSCCoOTeEFez1
Z0E2ALHfEWKMFt8fGHd7VQoJlwoIoixNj5jYlM8xGBDvNbFDBCa/4e2CaAIj/AZp
lhRBXc6JJibLqOihgoxc5fMNTe2klv3qWa47QmbYnkQ1VV5C/u3mwBBlnFHSVHu5
s1MduNiVpN6Z6/Cex5nloPZK/7TqixnA6/058Ckrqf6nLZUGIT5gFo9RRYyGqbNU
ptIeBZqRpOxLoFanC2KSOFnJFhDAd4XVzaoXTEvyCjj9miTbccY9xh08ldAlWczh
ORItsVcqKhkVD25FH9kViSKjct1V2b1fqBAEcuqwytnB4gp2aUNCRmvu6RDPBpy/
yNAM6d9dgDCyW55KNpv2aUoJmSxEGLuZhSMJjbiZ/B43ipxJHwpMmP1Vj8y6UX6r
bzpaSRXhPv6RCdohH0Z6dY8rpO2PEufTa+4YNYcv5ehCY0AVcVSGGy4PgSiS+M9t
HezSWjMkqB/Oa3a7rEKO0Em/n9Y2L+h3npXY5BPACo590diiPdbOajojdp8s9DbH
kGepW9TxYpBKKSODBZJF7Gv/yUf1xJ23g+eZjnrGOBanTRImSe484pSgmScbOg8N
dW4Odnk4zyozg61obVAQShRtmBU2s1Ix6Y19zrVJUlxo77dldkybPob6mtgAauxZ
RDKT9uaaC03fm4GEJ9HEWfKwK2m4lt8EiHLrjz5Qar/XUW7JajxsJG9+d6pMztak
TKevdDYv+3Sr7+TSDUEYtYgPxbBdPtT8yXZa0vruA5BA9yazmxIfbK3HhKe9XFVW
CEPr1kHAd3g8t+xQFEvdKJEEfwrWd31KuqXCmPJqPEyT8uZ51NLG4xqb2oTM14v1
DcoREgm8ZFVpsvuwylItnwH6jluWV9yzetCoL4AbH/M8os92mzgL9OCygBl4PV1T
t1UGyDidOpv1Pa4tWvvzJQioGf49mPeatlPfv14W+IqqwlcKsDVBmq1MusOXgafm
qZ9nNYAnxLU07FfeN091jVyAEYMTW0Bg1xWU2Vo65GoZURH8mu50Hau5gD8FPOqJ
y13kUiZ8PKoQp+TCYfWs4IyEDXCo4+wKJ0TPVOhH8mBeAZBQsfmYEXtZhBGS1WxB
OMu9DJMuEXMS1UWFH0NEajhn1bdU1KD3KUvLXx61H35NoL6c8ER8AwHTB51wPWsp
hMiG6T1bhXc8mSrZ5Z9ftBXe+5NIN+eChmxUZpYtbv6wvUQJ5aq8iO2CTjBa5948
RhXCRenGzF2sa2tRVQjWoeMzU5G5NGo+v16bIZIZXv9GsWJdhQfiwJ8PEjdnGENf
gFb/zSPJbno41vgKhA5vp4r3T9IGR8wqID6Q4Tf6MnP6MkePwwzqh61plteHNE1V
2W71pbkL1n63ciSw+2frJ86QiDDeKMU5OFpWR+pt/6dGuHTSCOG61KILJRZDLRpg
Wg4hOEJOFID+9RU6DBZiNpW1FI5VZ2ZHYjrqSYEy8z+tenmX/yg42YFxI+1UL63
PAeyXDuNQ+D2OSrs5WqPz+ac9SGqA1NicNMDnLrm+82OG/4z/1xcTU1T1lewQRCD
VvXiTNx111PvW+/wdD5YGcRz/yjBSTqV+Xb1ALKPTk/qrlpHFerTxWw1BITpNEA2
kKM31YBpYZQK+ubTQexACbQeeE7129OG5r9rUetCTEehlvzglhiYrWoGzFOPXUET
G+ru146zMsDoJSALJuJjgZrEQX/BMumYdFHWPVxAXy7d01zchXUTU1bzTOMteAUs
Hn6hpaELCpuWYhKPQ30aN/Q2zWpat7jz1w6rm+NPTHbnw1loEOzJclaw9huFUCQZ
If/DRPBkZ9JT0dfZiz1ZqCxDXilpfYXHgFMWa6OMpcMYQ/yDOggqD7/z2fvwUdOU
N1Dv2HxpozKuBV6bF664gJ3qdHmHEteecKXjKbuzUbTrQLE/dsZIsgrvZYw/sMiZy
ErLCFA+pcGIeO6za9DFYVQheIpv6/y+gJgc/H8NPJXZVREbfbRqnhqkMGmnw65FB
1DRstzU1AYvq65aeLXkDaT/9wydtN57ebZWD7zbum6OrgEjdBtJWd3NuiUQf/pqY

dbKBfBifI8r8oUWomyJV3l7HOxXLZ07bwXt6sykngeZhnW6gULF0J2VqRShN62iL
 ycHtr7ug33fo+EGHE/FTia3Wg9SUJXgssrcxB++igW1Ou96AHA/Ub4IQZM9p1IpE
 BH4a07A0ia2DxYbpWCpeWZWuKmba5jEF8VIyVy3baic8L2cWmMPjPZ9+DyQpsemj
 RTutRPZUUI5pNUPiGvAby+c/s4zLFtKFFzk0/mE5MhFhwWs691lz1BOA/L3QRNX9
 py9AlucjDPOjFrJ4zmvDzdogkwkXGVSF4ELZgh6Jpe4ZKNqkIOXrv79GongnHm2Y
 alsrIFshEQj8TxXc3GT4W7HrzrbCjT8NLGE2YVq8xva6iOAX6DcpPLb0DH3fUcJh
 IYBE0Wxlr6ZSU4DaahCFEuNvKBtLv3oE8izP+SBDvo62etQXWS7ku4kQi3z9Xhlp
 1qjLh1ePnZXd060RlgrpvfwbmT6sFWrnRrOpeCkjU4YgMRJWwzyhWDJK9VvYpFv
 axcyjGzBgkmdh3+EV8ha+Owy6OCY95+9tZmv5c3jdBHrs8ErFh1AsYdFVWCeN9rW
 T3PcOGahl3AKqRWT1g4yPxIJSgcWxLR1238YLcd05LigKh6VDV10X1Agion5fyP4
 5o34WccEbM4qvror+sEBv1FJkA7k3965R1KlexSFkVqyaZbn5P5EgvY4MMgtCxez
 KvYoCaS26llcK8ofGVy/UTyV8B1N6ViBX5NPcKycjVNrnsroPIDztXjwRHjZiPud
 iboVmbLdGla3m5hoUUGeLi1jbTkH+OUVga+0rQy1QSNHX/MGTP4zV4Gcj5NU76CQ
 0XWweIntePs9LTNJCJfYKyLPcelDAJ31J0ia3Lqg4GtYEJbp4pq3rwdp8vF3etkb
 8QHUBcwfEPE3kyK1VYRPfwq4tpmLrfWtvoxf/mZ33TAoMa3e1p9SXHI+Ndb+Sob
 KL8Fyp43miL9wUFYKnv0Vo67do3cCXyOA6F/wbJw4V+oLdBS2amMqnMwpra94Scf
 L+BlnmzQsGvpl5nieCQE935uFDxfxGUatNbKbsqkX1ZOIORPplfX+TJrAfShBsSj
 E22uxGfQ0Bj2W/3tdFVKnkxzCuNtKECq1xQSuTaWkAHW5apFfpVBpWxzGO5eoiE8
 CadNkpr8YFGswCripoYqPgGHE68I96yIHal7H+ufo1XK7QH9ZtVSL7CEirYG0Xi
 ZhGhd1QwMBDAhI/57sF2xfGgv8UEm717/94isN0XPkSqEmmbjCbpGhRBvRmWggnX
 7DHoQj0viTY2Cj8B4f8ATvdCEuPY+JpCU3xWVdSTJSOXq9NH/isNzxWWxx2aCS2z
 T/K9o167FcXMJN8tH3TCs0VmXkYwID94DrPknaUXMPqr8fiTedByso764tCoK/bZ
 FcDRnUbdpn8UCN8koJF4UMp6mHwWxIg4ekX+V+REudBAWOXF9pRdury8xbVfB6A
 t+RvY9aZhTTr7sFFDHOSlhOnRndzfOVj5u0iiKmdmk4NDMF/gIMqlkQ6m2/vjAEu
 2H1p8DJ6XNsLCIZ4nwdqU5326tFOaeylTAcwSXox4M/23zzEHW20+DCSXn+Gad3v
 U0iN+AKsss6pGPFxzwBzaWBIPcdXmzV1w3J0oLiHQOx2IHkGXXEeaNPDBOa2PoY
 G/vQRsJCv3vgeYHuq+oKi0ORyElrLkFakmuSZjgG2Wo05B5tapxMHoW4plyNDDPJ
 0cezblxnqbDkceXcHa+nTeCouRCqd/P6YVz5ocD4BIdSvrda5GX+6U0bl/e+IDoP
 pHWKijdsU3DAM+uCJrE9EwZHDrkW2qL/Spp9AhtbdMsugaIqVuuTQyCWhoK+wpz7
 wjCdyk1XEMoCfQ8PAS1RyaSUz7fYAsIk9P+FZ6qwyvM9zhmvFQcNoj3E5ObIql8H
 GezlvPOeoDwieqKamAHWkEwefrUb6X4IK9w8dBjYQgCjnwPq9G0dWu+MbbP8xwE
 w7LgVMRJKMMD1lquSaKDrQ==

- 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] 7845 bytes
(decrypts to)
application/pkcs7-mime [smime.p7m] 4794 bytes
(unwraps to)
text/plain 431 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

MIIWnAYJKoZIhvcNAQcDoIIWjTCCFokCAQAxggMQMIIBhAIBADBBSMFUxDTALBgNV
BAoTBElFVEYxETAPBgNVBAsTCExBTVBTIFdHMTEwLWYDVQQDEyhTYW1wbGUgTEFN
UFMgU1NBIEEN1cnRpZmljYXRpb24gQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIeh7j00
Boq0MA0GCSqGSIb3DQEBAQUABIIBACGdltueYBykYh99Md439ZT6CO00DuOkUssi
mv3sON023lQTEH4IDhS8pYhggW0VuZxgSL6feXXdBPYdr8UHnTNzm2X8X2fSpZ+N
HcdEN2lH71tpKrFHxIznR1bEU7/Zb0maRg8+07g5f1cZb/e0dnjEOLQsEp1kUKik
wZQmfi0FJaFRTGEdQh29pQ7Ww5rVltn8jyZvr6IFqVPj0lhYJ3SciUdJxygMnF1N
FyIBlMNShELvkr8C4huv3q2L0r02QN/W8Tdf1PIDakY5ziJst5q6ILX6L2EypcuC
LBTFWAyYCsechbb0ZyZVFzg7+Yj/ELIeOg7ZC0iPjQhaB91luYwggGEAgEAMGww
VTENMASGA1UEChMESUVURjERMA8GA1UECXMITEFNUFNgV0cxMTAvBgNVBAMTKFNh
bXBsZSBMQUU1QUYBSU0EgQ2VydgLmaWNhdGlvbiBBdXR0b3JpdHkCEzB8R0APhiY6
HGLS64MvlsDXhpQwDQYJKoZIhvcNAQEBBQAEggEAAas3uwX/STpLX/1RqYFr8HSB
yVCdYgegx1TMbw98g/QmQgcNvrzFDvp6vF+VkgPOqTJAlFSQGWraNLTmbBQ94i6
NXuQ3fDrGzr0L15RbvpB0VcqreJOrOoJHrgkHKG13DRTIH6tC4mgmOMYZToCev+H
bWpijRzWYdFH8wGQxwgFWKHF2AnXprLBxe6Uub+drp2f1rASfBehX3Aid+6gYP1h
tOy57CV4WIRA9/Xr1fAyxkfmChdQHhBziiuvplUtSVVQf5UoB91KkjRbJhCe45IJ
mW2hG53SoHPyud6DIhDdUB0RzbTmnnsCnLNo03HohsszxDYJ3oa2Otu5UhpPxTCC
E24GCSqGSIb3DQEHATAAdBg1ghkgBZQMEAEgEAAas3uwX/STpLX/1RqYFr8HSB
3D4c/K+3F1f4LD3bnX4C/QHDrrX+DwkHhmMT7SdnP5EfQngHFRFaLT14d39XpHOM
sD7kubDwB5uW026zEoxDmgfcYPzeY1OKVzr/sakxiMRWybdMQyEkWQYWxVzaSTLm
+pvssI56CmM/ulmY6F/i8ncGRy2w+nuAjaHJfLXO9+NUzRPEweEoMccfZ81X8yoo
Zy9e9LVvXQJ2gFyA9/Ny3NFXV1K7LbHIV3oAwztLE36nRBlrQRyUswvrYdowbS2u
fVPzhi83lINf02rA+HJ9WWLRgQtc6oGDHJqEzXiMRQMuBFWj/6sdhISaoELUCSRO
4ET9+D/hdrPpVuyUc7ag1q9ihJnPV1Fod0ga0XHR9RziZg1qLknbroLOqj3mEFCd
y18HXQbUOCxNuIw7SoLffFoN/qv7hGOfkf3eFrChKmvD4A+fYezswa71fQ37Zkzs
hL7EeLHf8PhTPmQmQVd3EFVWhIUrNvR2Fy/1ZOJjvokFLpsAfMyh/gL+4SMxJkYr
MW5KtcDHH84o9J7ZYIZwhoc/Zr86uXtRVQN5cTJnPfMFsckBXD+KEWBTGuiXyzRJ
ZtzqfVvYwRm+4MTWmmRHq0CRuHsrbe1WCGmQ2zIjdUIOG2+cge8Uc2aAttFVXfnXs
SZ3K1JHmRkvDug4qdR62lDDg6zfJNnsStk8ej+y0fLKZJy1qs7/MxcIRjxFaoEvr
DdKbZk9Pk2pJutgsyU9p9bXN5qZdQWJSM6iZL0VVeolN0sZC1A6leeJUAzbytV5T
2ahvUGLR/zNMLSFyDUj4/et0/wuwqPValLT6VqrGlgylt9VAUm6nfdTjln5mIerB

tVobrisydTBQ3wwDKY8s9t4kebInwfJx5l/1FaDg+BfMmZfIxph+CEVdaWE+ORgD
97FgoyL7j60qzJvInsEUe8Bb5cm18fyMMYGM1ydDGHVUZUGI6OxFaJZMpAhuq37A
7z0/Y46ykepVvOzjZCBhNldwsW1AftSoWSEXHGbOmeI4rKELBiXqZ2Tids3Ny5Y5
WGRzzUYufn5rD4OULPIPbi25Fo7WydCFnOIHBSPaZNixaM4fcjSqCZcpXnuzKOGG
M6iGJ8F4rS3oFgXoeHDSM0CWnLS132zD/NKRklmLTiAgwEJ9BPG+NgNIouZkv45
EbFiYCGef4vBisukj0yDBvhlzTdrGAeHk2nqIF5B9DFc3GtuzKUjY/5xLQ9GIWuF
NFgu6DoHqVmoBaISDRKf1Yr4vxqWsoW6a9+yIOcTqLL6l19hu40c0SaYpPEZLV95
io9pBC4N9HPS8tVBzd/GAeK/BUiv1zordIx9GgwB/200pNkUyAuQ+DXL4yv/MROX
Dp2tM0TvUNIQNpbclSP3oGkEl1ld2IvTFsKJMXBkCe/oASFUQDD0C4Upv7B6usoJ
ZH5t1ne3dnxDQfBvhykXpWMXFEkktxpW5EwY5C17Br9f10LDX8wntj41F7ddxzDE
xwk0GOKYfyf7JTVnxefTyMCN8rYjEiQCa/KEgeZ2y9ORPG7tnDWpmSbRVOxPrmFDp
sIHsnefohCbNuoLfWbchSGX2nNqd7zSn4GRQWAUV2CP10/sVcsthEjTKsHrhMaVs
PoBrhEos6wS2PBa4zLsFKTe85ORkowEW+n7TGU64Nz+TNR8w2xJqZrhJEiMvS51r
uQ4fg1vi jfgwPlmufZfH9UcTzZ4EpeRvTqp/YrfclulIfilijS1qf2VkJ3VJ1trJxj
Jn0N5EDb/k3bNdxsD5GfuYga04bBtQ+8inyw1bC7BXtpRJaEdE4xbPvQ/xARTV1r
SwTwK9cmHzB86GM8KUqLtNhvMOJLitfVRLlRcMYXYcpKaaBvXkPUtKDFU7adHC57
OOz8WCgazSrM29c8IivKJKtxk0+zSZ5riscOhNXR7wuWPT1ZMMXWir0oKJIRO/Y1
XptrfKa8goaSOE6abQZHMjdUwehU2W5epgZAz5XIUS0yBXpqv6f+NRpB75zazfNU
39buyaJnytIABH4r6777ft3oLe/JIOEeput70P+imSENLrUlQnafte7ZaGMSAsQF
v3RnekZqnYqnUSPU7hK7vn+sXbkf5tI6ntF7/XXY/BMrk7bAk2dvjiekscZy0JsF
CFKjpI9Y+dJ91+CXBGduBmavKSZ7xGdYayVKLyQ1SnGNw+IGm0sJ1fR9AzzGI3pa
XPh55uuzGOFY5Y34kCO/+0KLbJ0ry7UQGGM8F3L1yLtKeFvYBj1pyAftb7VdMI3D
XlurTQ+03tPrWP2lwFPpB9nZp7i+8JaH5gJSec0w9uooEXEZHKhoDzE/wK51uJgC
wuPcTFMrXNI2nGainJW20FDTsOFZ0iit3cx54qT6w++P6iQRJOzAH2ncSkGz4DFC
mH1YqgrY69jGWDa8Trg0RDBQH1aUAmOAlhmyVLumqBdpfQN7mppB97DNNVRsDhSY
VnnhvJH1YVzGJ1vxE50CLTFz8vDHgQmJLfab9IdJ2hb9McpWGGqLLw/u+363yxsv
ijn5Raylovp5o7XF9t+NKpeGPNXamhbc22Yg08omXRSTv9RicnuPUK6WX9TGp6q6
916X/8rUNdDGKxwCfzVK2pknexitylhlrjMY7QQX5QD/MEZL2BHdVtjn2+DvoqqTZ
N8T9ow7vZVKgTM0TWy9of78D8KLMW8mHsq6nHD9X97RorkucD8avlQdjgTuHbQH2
wXgldxGGPQR+xDF4p40nfDvILW1EGndaYQH7qBJYvwE6uxO/6uk8otg8AzdfxRlK
60DBYDHk0N8JDQmek0bEHSy4CbuBZgDDZwQ1AG7ade0WSRUZ0ZwHGPfFEozYnFG8
fCluzUu00aPYUhDchIFyVOW30Twt0DwkEbcMzXXqBpXMzHD4Yk1TIKZY/ok9M3oa
Oei8xx3pPFJaxfSodmV/qXwv5b+f/UrmCwwC9gLI1jzg26o2KZK9SGQfAMf5HbqN
yzp/RyMKr88w6urhdFdXI7UvPAcsi4wOOA4Q3ANX0T5E/3M9oGRyKpUridBt0Pfe
Bmyr2Cq6yWDVs940vPm6blhOsOTx2KUTKMTxWbbKjLKob7C6srYllc4x9AzjbJX
XJu34KZxfbuRbL5mLzpu5BPXQE7VIZqwPXoYl+uvj4sAGq8RfHqpbeExVZAuGl+y
Tb0gGtwaIyb3xTMV86tkjzMFprxMgbj+iHAeU0k2wbF09Cq2wXGddBUEH2XZYCgv
aviaalJRhNKIhvr0zmvugsjnsF1X91MYJwGJbw2TbSxLLcKK6Buan3e83SNVZGPi
Tvvysyo4XebbkCxMy4Vnd+SYRfdPx2wfleJsq6LYqSrAA0DgvTjs/3hnVtGL1YQcd
jttli j0V8i0VicD5bNUbB132G5qy2BoflCkwdjINBZcx56fXKMOJU5cAf+XGD68p
shyNm+/cexdiiRjNGChN26m/yNiPAkCwrPacnj+Z/2DTvmFFutAtImSD5y30NOyH
YtxtuufCXpTgw1wzXcetvufyOHCquSLWIhB/usDLS8L/eqBJaezmF7dHa9oWLz22
SjiGi+R/WqiSSFgBHAznUd7Wm9cUitJxLpMzVJDeotOGcFyVI0nXUR43B54+phJu
B5UBU5DSt8VbjeHmLUa4VCw8q38vDbH7L4NkTd3pw381NrNuzmRyIxcq6Ta/zUmn
CbWBfA6WoBHdaq+Lp8q3VNBE4IkVJObiYWtAegODFUIlvaSinxnUIY13YePRXX8+5
QTGxKzosyzYBm2Xy9cA3DrEY7Vvi0jXzAtNozQRbiQY0dcmDpc1GocJPK7gNftPO
BeCwMhlJ3+UVg+vMeX51bAK3/gnMCSryxSgs9ku5v41tN95KZxfOTmEXg2r1SdDz
pvwkAXzp0wTyD1v12fAexu5KpFTSqauxy0tR682iWE1bXmPmqnrxrU3Gii0Tass43

KUtV7fRY6Lw9DO/hcY4HCbL0uCeCi0YTsm52GPBNPyJkVzQjBAlATxmgSrW05+ND
Ww3FoDL2ae81XWH4n3ZAZmRwTt3myeUm2UyBWDrXsQOb3MfENTrQDjoI4KjoHHy1
k0BOS7MfR2SmSJh24aBsZgGuTekTVhcqzJHn68b2H5VkIaiSTS8LNBa12L37LpOK
7jugglRMU3KHdgSS4ZrfreHn6R3Mjz380TRwms+6fs4d55mqLWtnE6KMzm79cSw8
flCcTKgYwpJdPX8qZR6BJKbR9kTeOdWcTgeJtoeWHMccVd7SLFa8Ya7MFAufnkX/
nKyGteImetM81f2OuOc9s8tdvH6MnRBCGs6TLBJ/6HR7gvkAO8mm7Q7hF8T1flhW
7SBcWyV0ombMqutB+VxvKpzWhg+dozChhIVijh4uHCEhgHrDKgCRvQ0xdvPTce/f
boPaaJtf28S1Jtoc+72AISoXv1QhQdInO5K36TOMhC47PTZMEVSyWkd+PluzO1ue
jVw9f4GfO9lmJ8Ly5VHT9auu/wLiJ7N1x1Fuyje1+hBU+eH6vtf/IPDZsYNTyo+7
r9hjMHdLYoDBqRplLxkEiOhD3j3VvJdTF0D84Ke97ICldKmdtpgTMeXgFI21OolZ
dZWUeBo2xeqqgJWYNK0XykgOi6uLjs3pW72taG3q7pIgn66rHdQD5rixjisP2uTM
yDznF+q5QbrtSASQ3YoghwqLnXQnWrOp0swcef95tLHCJu6k3NNXiaMVVAZ1WBih
UJ/Hw679GGOXVfVeIzLa1gcThjJ7Y7IU7ipbx8JpczGUXkLjtEuOYx1BBm5lqOd
F39q5YeNs0Z8DXg/Lo8xFgGKTzAuzDfmyM/vabHxFHTUJgyB/Dt/MrAGLztwvBjB
sffTcVoAnzv5Fv2er9Qxgl7psksLwfRkV59IclGPrxfgwdZM21b0A3FURCGWvTMe
QLUm9pmb7HsvBfzixhvWU4Wo/OatFWX591SAlSeaNaRqtPNAiyj5mdnvJ7Uj11FG
h+GAhGnn5yL27v9gvqkzBdU1q37eiNjjzu/m4YBZEKICz3buOVO2/io+vy1rxud5
aMed7LnIqkXn8qXz2KPouU9BTiHwXLPby4FzKF6vJVF6q870R6b0WEYu0uRwTjLg
y2dHTpVSjU9rhTu4fHmbvgDBgvRK1Y2GWf/d8DSb711SgWVZvq6SYtjxJigqNKYq
ekAKOGbchPbn0SRn1YkCCUzOzVI0nFs7SogYWbNv71I1IkE5xW93Anpytzo6H7iQ
wX+1hB1jm/Q5iiBYTJU364NCqJ+a2H93H41Bf7PSMhOW+RvSo07JUAsaOahQPjP8
c1NAGqPTShgHDWE/1PUHRZ2+AjuOBY9tIe+NH/EF0zPY7uMXhm4srokBSdn1rosB
6NAnIxY9DDK5LiLrkrpQXJJ3Dcii fm7ivE+/FRK/4gb4RRwmjxTUtNv2c9Q3apdwX
ZawER8MGwniMghNwU0plAdt5z+4aZ0nU6FW0S1eAsTZ1uR40BTf911sj211fDeoL
2ZeUBYWm+lmx3MGtJivYk93CmlJMBY8Mlclh/vT1FooJjt8EjjLBjzJhWacTbBO
9/F7XjLzyEaG3v5u7C5T/mdDhYYyoQQj//M34pIUuGb8EL4Heq2wKX/k14QG7RBy
PtKY8+Uso6DUFztfHwwyjafJKIcddFxiO/eQiIx813Uj/q5BGRRufrNcSVFAGDLE
zTvGsoZGWkr7zxUw/cfoRALzKa2h69SCFk4XcYkLLnQVEn27NXN3FhxQDH41f6qt
CpVIpqeJl300v5fDks3ne84iKGQkMnjdYRGJ2UzGvaxGA9NN28zdhPZKO3IqT3dC
2Nsq4TgBk/0wICjSg/vlMjaYVifBZo4H2Swb4CSbYh49S6upMHU+Kwx5R+x9TBNG
vKK14gPzebpQxtjeX/oIJE9WEUS9/StuHpVRnuhY15kbnD6XTos2crZHpQlCNm75
z4gqzHsG/ZXD//NkxsFPb6y7A0tmho17wiEbLZf7r2045YE/UGR5IcTcQ+q7dAu+
T6VXouyzcU927dN6PiKmVkd5E6+oR9zcMWopXvsR0cLR02+SzbtxIeQofq7TV4Gf
ZaU+1NtZOUSfGZR8erXiptDVThvRbk+SpjCydJUf6RKpmQ1TVod8tIEKH9JpBftn
lhmZ6VHKEM9391ifc2pDl9TkyX3IOQBoL01MuPRbpDJiDODIdZmbNltgmoE88maY
nZW3ZG6GhUjQsYSGEtuyZ6Ckbc+dlGIWaVYQJM/YycxZ5QxasmgHwQ9jEgoMfXiS
EfIBev7/ciyPU76nT/ZcExZ5OYaX9NHvNpL0KJzTNi7NXGK/JDI9gb6P1DTdwreH
6Fdw1kZe4ZX6TpCDrXl1FdL5bI6afUIZOp1iUZtICwVFTzY1hAlui0aD/79t0R0V
EjXZ0G3JdJmqdd50fqxVfcq/xwDOqqbJUvcVcWg2F6zAMfdwQFNGx1qpL2etFspL
vwelmtu1UUP2gUBXpQyPrmf4EM768VaLjRoAFu2v4/M8zalr3Wotokr9YfiFRPEH
EYAdFENn6A7DDE9uhFPJ+qasySYc1NwmdGtXVS5ynJw4GERicu7mJAa/L5fVzd6n
xDKBsoZSv0yR+1I5N1+79Q7L5xEl0bITWIL00J8pxTE=

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] 9470 bytes
  (decrypts to)
  application/pkcs7-mime [smime.p7m] 6002 bytes
    (unwraps to)
    message/rfc822 1819 bytes
      multipart/mixed 1755 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
```

```
MIIBTAYJKoZIhvcNAQcDoIIbPTCCGzkCAQAxggMQMIIBhAIBADBsMFUxDTALBgNV
BAoTBElFVEYxETAPBgNVBAsTCExBTvBTIFdHMTEwLWYDVQQDEyhTYW1wbGUgTEFN
UFMgU1NBIEU1cnRpb24gQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIeh7j00
Boq0MA0GCSqGSIb3DQEBAQUABIIBABfhanpcfRrENuk7s3Y/t208MLeCotKAgVuq
+YxkFGf1eaxIShygOHSwbXnGM+P3BCMmQ+iTm3smLm5KvZd01e9M1e4QERyC2//p
VNSbK6NWD+5sFc9YMZ9BrQDIkQ3gSDtVpZiCoNUh/IFYw0d0Bu55kTxrDliIbPdx
rPSwuyLw43V+ytTi+PpnlxvI7mGYNLZxHkFIaYlZqjpqdMphNko5TZBE2tXZP37+
MQ6slzZZ4nnUDIPO9u85PlEabQM4zbTd3gpdri8wZnNb16kqnoMR5/uv8JmAgvEw
hYY1akgApGMqM9G7wjVSd3vk2kXPR8iPUP7dszHXdlbog0G7hlEwggGEAgEAMGww
VTENMA8GA1UEChMESUVURjERMA8GA1UECXMITEFNUFMgV0cxMTAvBgNVBAMTKFVh
bXBsZSBMQUU1QyBSU0EgQ2VydGlmaWNhdGlvbiBBdXRob3JpdHkCEzB8R0APhiY6
HGLS64MvlsDXhpQwDQYJKoZIhvcNAQEBBQAEggEAEEn22GWE7bdTRn4fqNM0tQeXb
NqN2BYvLUaMBiM4mpghZq9GH3NcmFADp6SMPjrh87ldh7aKLQhOsBKLZ5eMlTUJr
3CCxczSGd+8urr8fnH2/aHmarkkz8YE8eUNIPlcCJbkAuw8cskDdHgE/xPYpcNsC
```


J5mwtcVnenPFt5M6Xg2TeaY7MYLV3nkToPhAr4wJsE+wFQv5sHSzP+W/HmoPzvxF
cpG3JKqIOoMmbvWjqFKBc31HsFrr6LOhilpt/WS5N9oiFvld9VdsxX4ihoXFHCh
KORL5MqJo+dW7iamwXl/EiqbT84z0r865OfvWgWFct2bjs60lvSR803LrHTP2DCC
GB4GCSqGSIB3DQEHATAdbglghkgBZQMEAAIEEAVgmUQkXr9d9o00LLRDaHKAGhfW
AN3p9ViMzS3HiNWsI5Fdj00ONoey+zwgLD/6NT+kvHSdr08mVxkiSlqiMlU+p73o
tuGu2G95XIXzhfdSa+FaFHo9R0+hPclhRgwIOaKnanBcib69zehPF/v4PzgniUi0
/102qOvemyfDyUNepw/LitYoTFFf0h8KxdqSrXAI0UYeajVqk5rdjh0WZ8k7rI4
BMNXgFeiQzKr49+jME51mkjDNuLUziNhuEHpHLwiNFDOUybSoLyN01ZTC7ckSpV4
FT1+m7FH+LRwAo5ZqugK4i8CXbkqRByxpt8b+oPZz+7mM+L1bPNENBSk84eVfNQu
6cppe6/gCjdel+Zwr3lmwdQ7TGt+nhgQU4+ZdtcEG9zZfcZ2EFxrYasInDZx6pY8
WlqmI1lVESiGBVq6mbDn6QzQRPiZwdxz5mIjmlghHcNuBLgTnzBXGOANLsYrgWh8
noVnm6548GvXbYygAghYOljGIrIA9k7wZzfyedIhYYEc07BKHE6x0rNtAZsEs1JO
Kln/cI3vAn425+Pfr8AJkzBnzzPwcBlDkUsxu2RNbuxAKD+OIa3gnXk0hdQc4pz
LJ80AT0AQ1lKReaqa3WfYTDzZ9vqF+MECs5t3CqYpYk0T2dk27pa7K+03Nzk1Re6
gxocneYhsbNvzBTRhZeZFlt6jhoyhWd0IZQC+G9WKXyYi1j56x2c04MCTMDse7y
qBiHsCK8qWkz6PqFz0VhiorX7j7Ke+qTylHEF3jNlBEM1xoW5pHp8Jvg5JeUYK2A
HovLtrQiF4suTS/f/FltIZzSY8eYslczdmCFJVAzdDdHAWTiqu17dlR/v5Ypr4N9
GyeXRUZWMHKZ7dlxriYwwCWSaT5MAtoO/mIFamf5CMhUcxCLkmdRx23T4UCA/yf
ffZbPBCO23GzAG4WJOKPyCVWBjIyMlTPinYq4c0lnEqTom3CHYLFNuWAd0iehuP
aAqYtMETlFXyu7+AtkWqBgbwmec3z8LLIJzWt1IOb9opoP9QIhTy7aUePT42eA9n
2r4rorVK4c7HxNCswBlSKbQrELyz1JiUcYeqPArb/jDE/LlgFH/D+wrL1zIAgr8E
kxzGBaPmxxq6dhDdHeEAU3oqoWbt4e0Fy+bVoiw540+e5NbvtM1+HAeKXzAy3fX+
Y8iavBhuLB0iDSDarP2Exc2dDO+rQOk6EYCVqaYh4WRA4iRe4hsW4WrwA9ccLGVi
+eTdm10/uJn59CcJEFs5bicctGtxTohpuzYE4V2BBBwXNu4KFvG1USuVdsH84Mhl
TtTo4ptQly0u90eyWWlSdaJORBMRMCj2AY+wwldRFpx10NtbGwQ7PtmemZktZgyf
UjL69zbu0qVOWW6h686uoOTkF1D6K2spPd7nLzjsu1KJjLCdQgbJNU20z3RswPq7
cSK659Uv7h/kagEhlY9AheJtXcbYxP/Tb7ieUQV+CmeGPM0xQceWd/LnSudqh3ZB
slRv7nDgIaKqoF5dZB8AASqs1W9f62CRy/Kgu+D0kbLvc4unid8yS/CiFXsPGkAw
LJwd5nihVJC2jw2GrfP17yhNW8TR87nbR/faqoyWmQkjyqy+ezNikgRM2Tr98fe+
CTofmHuFAOCAn4q9q40+p6YCDDJCYbyyP2nLipaOZBpVntoysfkvH7bBWC5qrFt+
xK7Yz1Pi4Dtw08K5F8nqaPdgJY5hSKoP2fPrJBwx40s92rOalZEdNA+Ig8zcMwqo
EYRE3BKxPBgChWxjuMcowBkNz6ZJzBSsfFYHz0/9NdDStBl32M29oNN5XBIYjbD
sS1NqmK7vJVkrSZIn8w5t1VQqo6B7SG34/sMPRZvfXLGvwDO0sn5g5NBj2to323R
rpNwXHRQao106IARwxTSCLk7+r7mjz3U3Cz0YTWpuZZK3yMKg9JbxAN6rG6fb02+
tideDrU5ibGI+VpBxPaoO/q7XBWks3Q3RX4502uoAPkYNBr4D6PoMXqlzrtMoSg+
PDKGTuZaw3RQ+5ED4tFWU1lVQACLDSzT4Q/7RWkff51b2aswy97gEoRCEUYc7GA/
KDSyviz8kGxEF/KxqGFZhYB1/Xs2VA/o1XUZsbr2YX/mhfn4iEvMUL+vI63YEkBR
KTQdM2UEw1MaqKSSyo4TGJ8WXG1WerWQ1Vpxn2HmeOb7mIYw0CC5vMrDsYJ4Dz4f
rAG3v2iqqG7aLpbnXe8BYLVMgcnciJWfav2lWNVUnHhG1IyeOvuvQRbt09RizxSK
fe/5rjxBBa7sPu8WDESre5Xg/C8GdbKk4vJxM6pUnYKLMGxpHO/XXWDlaIV4IuIG
HnfUZ9UzR3cilV53bmuWKL1AOMqvJ3Qcv0ltXdcQvk535uMu3VgRyrwd1wDbVRH2h
/ZTW5YEO95wjcCVjfd4YTXZOoinKBf7v2v2WDFCVOYQ4Frkertg/E/V+jcJ0usoS
qFny9JE2WQ3NSkYb1SEYQD0oiWH/6++kjknuMpWP2Ubc9UTERVD81RGPBNL+vAr5
ItFtD8iWBROcZg8iB3dWaM6Gs2zu1sYZCWvn18XVrHkQjvqv1iIeD9pyrmGKBqc6
jdlFfhY0Q4Ucy3GxE9yz/WT2SWWXOumq9PiAzOoh2jg45w7BWmsDnRx5W0woaJvI
1W+BXT1K/aJqnzDQELZCYLElG4jBkqmUvpkm6wtZ4vs3xwNMGo5vVLUkudC4ybag
nHrfb0t42o0IM4mtJOePslIEgLQ4dh3pdlhYF1ojcdwatHJ4yKjhl19UbjWcRFkV
BrzhlobPcvlpAx9ExiwJqp91ETrdGk0I/Kwr4sacP9+yb9tnuP9Y8M7KXn+K7Y5t

p6OXGLEAQsltWjK9b7XRI5y0FJwkMGFFjvKIVgLwDkeIYK5SNqsCgB+MoSwprtGJ
X7XWtd/6RICinOH+1AnAeB/WUVox4634qyh2GZC8vRvc2xNdKFDcLA3giC2/1tpb
CeQULpERCoy5Q/1jo+ShZSSmw3JbdcJFuDP4varTgf7Ft9mAWnd8xPtkTTYKgzMo
ZO6nxNnMdNBu/3+NYWVTSXuq40FUEmhkftP+GbVdU89jSr2oXsmTSd2PMWOUUnNgN
oJK7meDsHkOPjTlmg05wvvRy9FHN6TNWEfSAAVeJHJOyoSRQDdRtmek/9AXecNb9
wyKXyw3aGLlwb49hC4AE+w6zw8uAHNF6xYGBLaxW9jWyN+EEYG5mb5Co9MPsqTEa
+Nx4CMoj3VLFmk3Q8aYtIEmyQBkjY10pGAix8oINf9TTWvAgrHimCBQhsztQoHgZ
uByvSyChvendL4o2BsiozGAhUM21HC91L2FdtgVKEmYyXZEGWSdhMY7UD7uIPauo
7/+5o46AS1ZBAynSHi8oAETNni/oy4704a7yinNNcAsG+ZXH5mZU5akGiBJjPH7p
6REwmf11k+RGKS6sOIwdbXqgR3007qZPkesAKUVRB10xZkgEZ+DkZtOaULTxkxqJ
ED10TW/lZAm3wmTY86UhCsOiPRCMvsfughQisp4yZeEIwls3vblLf3r4FLvgBLRc
X9wdASPYHMPUWapeeYsaJJPZ23B478UIINoziz7dEl/OFGEHhKwiNTgRG2guXVks
QX+9LH4G+W9Kic5fwm/5M9gkQXOGu+OPIMgIy13RnyFr+5rFfnCcdq+FKC/w6N30
3/15JKrRup4exCfw5FXIeUpOtJP8W4HKv+cPtTJ21kXHHpXkMWswdcBWXGrb4Pp
rOII2htbmRcq/99mx9/7cWmp1ZY512GEhbd73CV4ZUaRO5JJV82Hbp3j467BorIT
D/hmJoUsusOypRvUJGGQ33m5uLOTqmQbuRk21SwNLYEoih0w6HK5Ayz1i4Jyrc0B
gxWkNkkWD8e1QcYsb5kD1ZeomK7HHAeXzZBmW+LeMrkFAOhXqDFC4HO+Reza8d8k
97RhAjNAHHdox0KoC6PY2dcu3VQEkyod8PizWgBtZYcjL6fsntjJNL/rDTl2Kfm2
XkKGG/2Q/2RHiOhGveEv6lMQN9CmvzIyB2Ijff5fpZLn/B0aedX8H1V33f/J/xsvA
nw2uAVziSucRJaEcSUoNV/cKgpV10uwBDcVeE7+p/k+RLY8aohN4J61WgATzV9+J
MFbRZXALyzLrVKk6y6Siog+7BisQajPtU/XncGfrRHxwHRJgoOoJM/jXq91KyW1V
YlUNu/ea/hz5xOUJ0D9AlChu3b2lZZ8lMAwnxxjyMvb7xRu+etoSpWYBB9/5B9gL
KXA21xpC8tdk0HVpPLH/kGwcZIsIr8GS8A2Unj/dreOIIW0+NxB/ERGPkbPEZ0qR
zBZZdkBbL8IckfMqP6w37k5ZXKHvJzQS6m2gFmNoXi0Eybx5cveSk/0ZxyohL4n
BA710uc+VoReh4st1zRWPbrOni7AuYeENDTH6kpQZ6Gdlkd0s05c1EPa+zDdPGJr
21nOL/vYAHVtW9eAFWU17W0zSbRH8Fu0UFBSiuZmRyPrdd+bUL/GTPATDDSEdidy
YBh/ihWM3PD10fgOrygqbpK/BmeOVEYesTHqjmdjLZU96NGMfmr0x+53a1YhFd4b
3sFFDdWdmDBh4eO+dELQkbT0ISLjmICTWw8TnKffjM3MDgy08VvjQP1ZiF7C6aaO
wCYNS1iX+B7vANKfj8Ax89jgqPqyJzmB8xbxPshvBvq7X718tWqXJnFuoFUrEhaz
l2h0WMxjY1P/r86Y1mzMlw17EwagREZq3sTIRclpu4qYN93RhsUOXFGRukYQLh7C
1VKgvOYGTynVDP9C2U07Rq/wPHc4u/6ZimtKJYddc3YqpVNXiQYv1unfGgz1UN9Y
tYQVDM9d5k+1tdqONOpG2SIDifSCRpc7fOg05h1o/3+3JSRYg54irwaln2AiaOK
xhrSXkWLtaV7yIAFr5J38rYA5lGoaYLUAP2NavHiYCHjIUjAkm5TxHENx0DVOuk7
IQvkXRRXckjBPWkvOQL6VwmauiPuQvYrWhUQHng4npHb/h+WY7RQcovz5tMtMOs+
RFIiORZmJS2Fze01fsR3Tztu6eUQBotYF85YKvGCo/bPsvN7hKdY1L1CjpsDtn7I
Q/dhxXWnvE2SXpzWBN9LrQrIKR4UYTCUiXXU+BnEodBJQD2z8/1Bf/r/JzUuHdo/
CzA1OGF1IcuUYkoBZild5ZrWoikcle6XCxXtUQ25yub8cq7V9Y8eOzpIee6VyYcy
NTEzqa1AQ/NPVOFICTt19b1P1Tmo2I42GmjWTE7mjdaNc19MXg8vmFqq6PaR1n2D
na66iGUEPfoNVGYFg0pR0DZAYWIE0ha9rY70cy7UbiQKRg73oBSMz1PGy+GNJwVg
75K9Gpkuu4iThey3BB6Kc6Qr8ab3CNoAf4z95VqfZ8eH6TwLjPwPrLbCa61iayBA
MKAqD8mtHmLclE+9F4L9hn3oVIEk3gVnKGWannZ64/RON75iwXJ2tijwJrfQsfP3
dteQX1sBrT/10Ui66PiOxMi83GwHNzkonFjia9Gn4FEOLTenDZowI+Fzp8uL9Ssz
slziDSogFCEjSJUmBVBKciUcwDlwwuJi8N4Hw7MUlMxx0gLWLUWe1t2eCdrDd2WH
vCCfZ+VG41q7d/7nrRKNnThBZohgg0H7DFIuIco5a/u41Er1vT3Cxb/LSBGWAHfh
BPC+vKdBAdle2gnyIxaJsv/8qPjbx1I09okQvIMygc6uA+ScX97RWbvWvFu5Pzig
NF101VSJqI9iO4r9jGm0P9nyDliAQFEcxqUNIQC0V98oZLSFA9q3jF+jqClAMdXL
Tj6WAZ6fEmamXo4VW1QkjwIIqWQA1DdyC2ffCZhHgtLL7MOqsOagvtMAPRzDGsl3
Dj4uMUPkhgOmj+LzZbylat51L2n9qtZRIQpIAzSpCiHIakxkCZaix+TLU6xIsIPi

TUw5t6QxmgDeqYbio5VYKClDb7LE+SjmESv0Ss4K2HoNXPiViw0GlvQYJoWpLiI5
 E94ftgR41MwWhwEpeb+fB6i1VS+KCyyFOjPBm1WOejlrPYoK1ZbRJqVGfiV15eN1
 bfvW01VoRqhGG/2YQqc4bnEjhKUYmPnqQ15HWeZGbZnlBQzyArU1s3WhLQxiP+00
 k9nh6ThhMD/NcynQpa5w45ozDhoLfDrE7W417oV5wcwRjkw89ylt8MRMr7XbJHjo
 OaWaIDc+BU9SNJWo+OCzxkHOB0/rYcUEHC57gh93KWThFdMSGppju8R10DshdQwtq
 ivJwyVI2s7csucaxcnao/dlSkEg00fUDTyMpXHsUE+TvAJvZbu9VA82oS59nyeJ9
 1Wnb4PJxHwP3v+3xp22MadG+wwQKQ4OsWS8QjMA+Ajp1ltz8bbVlKDaJQ+mX7fO6
 sXc4Q7h8J0AfaX7CHfe08enFdQhgTYdCIinGQVVF8E52tMp8bCosYKQ0/+Gs7Fs
 YSsqSMyrTkd/vTNzzBAT2MKM/qRtltAoR1rGH2GEYGYy97uCXmEK/CS60Lsu7CIm
 /JlSUTPMkfz/rGNQbNIhrmcoshyIXRMn5zJq/y3T3y63jbPRE2+w7tDDIMoFFjhU
 6ciiBm34QaHTg47LOhHjFRzqBPeAswaset9i5XjypsbPbaJcvFBA0IqxtXJp8J2o
 eUDpkKsW/Pji8EQqxP6/6nst2hdaWRtv1C4cW9mkCobZ1xvjqnugCN2ANye49yxm
 09jQYjUxanul/heIQcGPBnhOHMF02e/RwxscOqQdP+HVghQcuQOq/S4rtDAuvCF+
 PPfcB8MNbsWdD9IVeKkFXxqn3rtv1bs1WFCtVUjEI0cLorKixghPeYDmKNdDh5Ku
 1ctfIe3wwadx9TV3mvMyjEoz5z/rUstZgh2SmKT7NznKrGHASKKH/e+qnI02PvU6
 aWi2mVvHOVHG6Sg0RF4FZZeaZj87bXyQz97ainp9jiko2GCw1xuy5hjOCC0WsJb+
 UcfjBRqePQhSgo2LFT4+XtxbzosuCE74sefZLuNE4wX2cbQ1MPGh36drjY5vnygD
 bl7Zgj5j5kOfDn2rFWORdkgk2yJE7Gae0XnkwiFGEBSYNpNXZWgW00gTZxApQaAu
 N2SAKRgKvKzLJTtpgNSIrJ6H2MOU+ImQhobluQin43i265h9u7/GXSHarj9I5Rxm
 yOtUwzF7J6IKV02ZJyDuNUXzpLJJHh3tvQX88N1Y3oLBJ937j5xryIDHHNvX4bJP
 YpJka010Pv9JTQ1PRAVHe4gvpSxb1qnEb+xaqa2/Kz/1hDnVJuHpC3cQyLgTkk7k
 UtJn2j9z48MNK0Mbp7r9BeveVb39QGLfBV0oKnILQX2hv/8dkXvgN+I/tSjuW8k4
 sYXg/tUqwdulFEdncgA+RvAGIqvWrwwzZESO+BFPavv7anvn5y40s2l+r8NctgtK
 RLl34q/LH+w2J4OVlkMEjqf9xDctDAVWQ0Sdsqul94TCK8UpzNJAsc61QDdedlE
 nUsAKRQiYThJP6uwL7Xz3xAowcMyNyNLCxSLsgaYna7F3/rRoJr4oJErXX73zaVL
 EWjIuw21J5ba/5+XN4rHFKS1GtNNP8A5GCGNdbKxknowUZdMSWH2xDOXWExTnCJk
 HJPcmXu1PnWt5NOH920R3EpuFKrRcSKKniORKdNLo7jPLZ6r0KwuPN0QtWgmNzQC
 qSB0EWuRliZX+g1NR7cWkwfLixqVtER37OWpNEPr6YAXUFqgsFgKBNNM9etKVb1l
 82mWq9DRGbLCrhxbp8iAu4omBxQe1mGGRRT2WtBwkAvQr2O6sX/RU3nBvt4NvHwN
 yRyiTWYpfve8RzriZuZdCdYjagegbNVfPege0CYdhq3XYzf3AxxrUVEZaaC/GCZlq
 innWTPiXunVZyqF0v/UL6Xikh4f/1L8i6Zn3GKeeWHXHnyzsw2c44eTzBnkC5eqz
 F15GHyRMedBfg/3T8VnZSj/39dJ//+xSogpITDQc4yW7u5WKDvS47xQJ142yh4k7
 bIAuqxXgAt87MWUA2mLzuiFRpWFDZi99900EH+teaiezOXbqnv4EPNjWGxRDPbyr
 EIVNcKBxsk3zuFtGCsA2cEXLJIjcucV5Q5PscW5gBOqopPjNEC1B5Fa9LpftzIR/
 8QoTaaW3Hr5PrcMgEuRnfiBKriyKsXzbyRzsrozPlieA0ygm35QW0Tvr32QBWuS1
 wmSyyQOnKRpyLDGZUuUehGyY4C4AZ7utFzxG8SBodg=

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] 9515 bytes
  (decrypts to)
application/pkcs7-mime [smime.p7m] 6028 bytes
  (unwraps to)
multipart/mixed 1785 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

```

```

MIIBbAYJKoZIhvcNAQcDoIIbXTCCG1kCAQAxggMQMIIBhAIBADBsMFUxDTALBgNV
BAoTBELFVEYxETAPBgNVBAsTCExBTVBTIFdHMTEwLWYDVQQDEyhTYW1wbGUgTEFN
UFMgU1NBIElcnRpb24gQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIeh7j00
Boq0MA0GCSqGSIb3DQEBAQUABIIBA JMtpwbSzvTtudiTaWcx0TvyxUZpiHL+UmRp
WR9LJ8Ev18vh5FnKDB9TadYiAhseHiWnelYjygz/q5C8lV1HH+WwEihs6x7gIROb
IAudvBR12CMjm4HX7GKKCNDyFse+QRiRuuuQzLG3d0/2slCA33mCsOhkE7RRtjvz
yoxcoJ8U1z18BzFtjYnIcjqR/zkeMtaTdaw9S15wLSOCHhdnA10eYAnebMhpZM5t
NatVeDmlzoJAlqQKtaE/K+LWfhSm2Y2GKD2I7XaslJS0QBNDd00AF+537e4m/MY
RylhEzNmR0dz/Tyg6tyqakhXnPiQDQrv+RaXMH3RWDJWFZl1rYQwggGEAgEAMGww
VTENMA8GA1UEChMESUVURjERMA8GA1UECxMITEFNUFMgV0cxMTAvBgNVBAMTKFZh
bXBsZSBMQUU1QUYBSU0EgQ2VydgG1maWNhdGlvbiBBdXR0b3JpdHkCEzB8R0APhiY6
HGLS64MvlsDXhpQwDQYJKoZIhvcNAQEBBQAEggEAger+uW41F5G04kqx+ZTa0DlY
7GfQxltAwowLQzQPBlzUs0/Wk jzvXFDBcFkXL/8RPGyqT+5GvNxiloFEB/emqqTg
qee5jWKpur+BknpYLKQN5bpxrkeRSccljN2h0+msRhI6m6T7HIPs7GqdwT0C1rY
Zf0dl0+sKarYj3cR3YKV8BDD1kR+QhFLAmzRxryvhdXSyzah4KShupL2tcBpOYbQ
TFF5bj6DdNY8heOItu3/EzH7dzfJexThe3dFh7HtSEMkXiVcqNVIqEVtm90dzP2T
lOrxdnqUscbb+6lrIOxn+JjQmRLSt6JImEGQaKKxXuTzaR+PAERxHemp8HUm1jCC
GD4GCSqGSIb3DQEHAATA dBglghkgBZQMEAAQIEEKiRbYjQHnb/KW2jRBj95raAghgQ
74WlHyDzh0qIAn81LJx1D+JkALWg/z3P8Xhy++EjKaMAMy7CgoYG409ElUzKIDwi
taWh6ide0mxmDP8oKFVVtb2cSIHDWnJD8+Ox1F5eFiVo5zKIdaSRekpopoKFoP15
/Zck5ua5lprkBMIBuinwHLtHLAq7nVDP9sv3adB55mUKPCRstPSWIYAqqAEbWkoS
gQsbEH68Gxp1CaFswEn+GU6Nw4ffdbM3t4yNHNZ/4W6I6P03e5fRhclbU+6wg0s3
EhQNIgeP0aHYPjoShfAJ4IU+er9TV+UHN5Vz7FIEPE+Hpg9xNzL30fwhwN0Z1zRh
fX4HEd/nM2kZfezaF4DDoMj6g6N3n3mr195JZuuH9MoJhhTXEFZ6FIVAPG13RrWY
4KFEhIWCrnjz+KJ+PoG/5A8RJ9MSZLaWb2c9lmsj5+7WWwPqih+Ky60SXFnkSC9
uv9X76f4d2mIzBKzPt fpvka7b+LUua/lMCRxVN9q7eikC/po9yzc69kAezNAMwmF

```

xi9Ni7yoCL9aPibcqUv1ZlW/mPaJkOJ68o3lbb0KdR6dfW3ZtW+bI7IQwZnq2RDX
4/fxf5qZ3l1V0802oVvfmrr5xpRXVfyYx+wdln2DbG3k/NzUcn8aK1mZ6pNEDily
va4GJ3Vx1AHN1PxnL8sYFh2q0/VWLE2+wsL6RtQzS9vK5SLJDr19J+IY5Pn65Io1
O3gpsKC1bGODiaA0UGewRIUi9yfHzCMLwlmTWC9QBzGP/bqsp5bScgP6u02figP0
Kl53pevpRsrM4RLF+Jfcz/DoTso2qVSiBChh4qAiPmNXLmoR2YkY9LqWZHXc+Nfc
4b2cIUluCSTG+pXlp33B9OAytkfVNdj6SPhLv1+jASccJaPdY4Y33cboweJkdVq4
cBStfoG+nFQsDhXx7KKym19Tzce/tu3CngG19umIuL9rT2uksT14U4h5hsylbbCJ
IZZZhpF6JnvCN8xrKaX8LAcb1G2+DSbFvrkCZea+ej5v4sYiVBi40E88LQSB30bJ
FxREDenMDiKRTBEERjmqY6JtVsVrcm5H19/cDnrsVdAKbp+ToAncdA3Jy2bQpfhe
Ev7D/zDK2HPam2PODgCSX8ErGs0glz0O1rtXBd0+BEEb8o9CyBj6VY+Rk5B8H16Y
2lasrQGVpqnQTSeJJiFRGk2//3ZPmJvhXnINA9c1BlctC2g+UPwbxs87V2oRhQkE
oCdpPS/0GmtCFmG2pd07Ejbbwx2s5R//sfQEkYc/jkmu8u8xuaUbP+yPSIGaSUBN
aD+12wvCsOzAdZpv3oS/uQ41M6ICuCRdWhufcs5M3sNh6rjCk8TvfYIYOv9E36Qcu
5owhwVaHzjy99TCYWq6BbtjgPrwloYi15eqzUn/xPu+OnjqSkNlJ5V14QqutLAOK
VTfscKVW7rHDcmM2hbw1+rk+X/9F8tY/X8ekuy6Fha+NcYTjAsGwsMghGZ3I9YI4
Zw3lpucLV9M3jmLB1F2n9KHbZ920SvFMzuyTeSXM2nEnvRPOoCEzRHcSqew/JMty
2Qn/me+bp28rc4zDLOz3IAYot0SNC6sskGM6rGsXvUmKkqu3U6D+mI4yhdZL3wLl
xuwRHM5ERRguxQAZFrCuc5w22UGLlgIShUTowRLirZ/e9KjDg6GzsDRscQEgr4zj
kCRKsIVT9qotk6PjZXqcn5QJsylGhH6coGGQdBbBikwx0+XIOITCmtwIrMU4S79B
fp+Ll2KTyWT4HcILWA1voF8CUAFYqZMYEOvxCF1lyP7UbFe1NNRU96BxkCtOfcP4
2vwp2I+nViA4CwKyoizepwZqkLZERiSvvQAZah+UkvD7mni4MiWN2OVEPZMvTNmZ
p8VmOLRESZ0Ut67qh4leKq/c8pQtEhGARUP91n+H88bpFNVI3XhwJrLVEDXCKx7B
IKcEwE6Di32lZlinkiWMTJq2V5k5oUaAI4D87y04rabhc/4Pl7rMn6LX4vBKDMcRW
VgrlIy5+AgglFaGE4NqNLaxUHyn4tq8dZIVdg4lWm1NFONKKTtNfTtGTwdYNKFEF
14LzyWsvfkZvjUmlRWTITQz/rc0zkow98aYbQwWPWgAH7TK6tcsUSaUGi2nVm8FX
1JAMLH5KrQjBmyiKldpc33Pp2T4vb9C1OvVm7G8+E35XZ8bFdH3JmNcRB+bnh4dH
Bgn982jnBkTh8TjhksvNs+tlGzHDxh65caJO5t8HuDuX78oUVJfeVU/pmlj7Wa3p
P0OgW5tckWxyLYTmOhnVHUK0GS0lZZQPYGo6adCGiGz0ghAHiikI0UKy5zrosh/L
+nERmxlycUETbc2V5N518BHveOR83WlhRlqo0SzVlLPJwDqhwDyk3da23fUo0DDX
XTjgvokUk99fWlKzma6PQsFJaRbAcOJycBOP9tyPJyo+h3s8L9MoJ5S51xXPq4Bf
N/LIKISnci9+QtNCg/baPByMLUHULep41lC+aeFqPVT037EJ6ixe88PRpVGkvpdy
8b2SGlgeP/e7fIOM2lAcpET5HI3hbv6ILYAIM/U6iEZp0ClDMxt5nC3GiU0guSte
c2zBkcwz6idYRETZQnbXFiKDNvolGoR1vh5h9pOFFabcyjpY3dxDpjGMSlrrre/lX
RQF83BVCFKfGtZjuGSRc+Upe44sL2kxKjHrJTpeFp/gI88Jecm8UuwsFHIFAGdr
fczsiGKBjeBHUJlCM6ilQYNx9zQs/0DsF+WWBUzthv84Lw6sVDjZaGYkzjZSwzvg
iH6+ytZH26KVM3/QQ1qUB2EeLM8Jh3vNSK13BLsrHr8XqQm8wllKcySSS+mDCmLu
kmjwrXI8GbWyfkvKJmWi0WMEp4v+AQqltSSoNoQ+NYMzQe1vR+s9wzePrOmQxIpw
sdT3OxSr13r24K5R13YdhOD42YN+RSgU9m9MCLDg0Zst0n2FXfYh1+c02uwoSLbi
6GHviTKteFAkk10B2E1DDj0gbMMvnXIhvgFIM7GoPf9GU4bDuo5ohDdtCSJwU2qv
e4JtkYlVCY9zAcnmROqUSDpXvVlW1Q61FhzQ7GpkOuh0auGS7Sc3BTqX5s6Y1Smj
0dChIy2aDtXppCDxvpLjYBko17JKg74ZlwdzyJe4ohS+w4h3oQNRZMqIGR5M1WeE
6XCX5xELYh1lXT41SEL+ZXkIig0P+TywxnXmbQ74zY4o7+ttVTarYTjf4leBGjkRE
iqfTJGSCA+HDhMy+ULYRrdsbwWVpeN166anKpSK8hPZe50+ULzBjVz1rsL+KX7MF
h660epx9YwxzpEf9TK2SstH7dp3lbdMz96FL6ugWcTWSJa+ERyF4vt814y21A7W/
SZx4N5W+IzUG8kcws5UVcZajNEE80dm8blBxVoFJloKaRdlRS7aP+YPvaeOmI5l4
FeQ83kAri2oAcfmnk+yudvptSl3A5cmfKV1NCybx7vpK0ePwlg2UJtz1RiIC20Z
kCNMpLLN8hVkJHvJo2D9ic8IAmt4EGVQGCd6qp3Cv3RwYeVtJVgMnSw7j5HUpdFI
JiIZl7ZSNLW83CiiqJXFmkyJx7AxvEOXNC/00jZBtoOKU4RuGp/Uzpx93g+rao+7

97oyYsmNk8WVH5qk4LXhlNw0NuQnYEeFICIElNuZOJOJ/PjBI+hVvr6NtuEZ+0FV
J3zQYjMaFQ9qd9Ea jVHjHJVy jBCDoAoJNio8l8OFM8/X1NIMxri3nQYc4xfHP+yx
FgHbHiEcEhn1uHNARec+E6zXcVF/TmOhNovBPEROwhJhybxKAaaSKPzDZEovfJ8a
MpQxexpNSpkJ0u5gcEw6Z2xASX6Qn7RTPXwJo7hNYOvqDUVUdwQLPy8vJHgqn4iC
KAdclwMsJ7gTR2bgdZYfHGxU1XG4zKPvSZahp+uEcxiXhC5N67sC301A1oLmXKFX
YzGqt3ZhZu4XsPYWV2XEEa6S2Y+3ygke0HuloY/8aosF+3ow8UN0KJWinskyG3Rw7
t+ssGCQ2sGUTdpx7SOLwpwz1UgLuIJ3tvUK6l7fsCwU100uG6j6pqNIALRNiN6QM
ayUuu2lnKX2WTIiFf2UN5lppaGncolEwYoz1fF+0Xw1+xmoFJ+42QgjHrZSLf88P
w0jis03nKyTSNvqcJlv2yuloI01u90HO7qiCzpYbByPkFYNOyGhSFZ0aMl9vxoD6
O9tzFkNN9LZQBhaWxdubZJ0xdEsF7Fi4c0ZB2443iyMJvhkxeQ+GeS8sPrX10LQu
nUVLUXLG24DI2w8o7ihTWn8PtZNgcMbMf8c0g8+7yjmYRVtWcqJvfl/NtXv5f6x
FBQQIdQ05xhF2PyjUL+MO9xiUkC5YBBasBUm/cpPKflGnDiiqw5NRKGzdo6/5Pvv
pB5iUrukmlDJDROUHpniT0FIs2gVa6d2YIoZliXxY/eYu8i3laAS0/h8KXiU+fn
GdzpeVKZ3dr+UQwb+gMXafWV887yre6h70AA8gCWldAbkRaNj9CZeKlM6Z1lQilp
/NzqbHoCyvn2Ehrn8x8cFpEESBjau62otkaALHD032L2ijfiKqlq3AzTfgOhN2j1
IbvpXGhke9gEzJG15iWSqe7agSTb2AGGcgNaRlJP4/DW3nVf7SF01/J1dJP1C/w
RjmQVSxV+115g5bHxLr9BE4NOgAha0DDHZ4MVujaQaIj3XO3XcLhUROpbSC+cCzT
ZOmQ/QnCeimZlsfCmpn+hRxoV6BA8VBvI5pEprY7+YPiWGt3zqZF4Ot0UggbfZtM
WSDqYv5CoXdSaVvBOpofBidUdk/ASlgjdQBbXk3P/YBFoAbkbSPQopmlLmxcytfu
/W1GGf/VmK5/wm4QC8yu1nE+8b3iZuG2IxtHamQZR/qgwk9Qi1juhDiWnx3mITK
CJHeZhSR6zfF331p1G8mAYln6ZSfxrzQ4R5h8b/O/u4mf294VCNj5hoaTDhxEHmw
inflbhehkFbk4GQT2Rx7Ub9MU7mhkUpf01Ch7lIn8ci6jg0TS3Yr63gt3FpW8YRG
Cyauu/nUGZg4MXRfzEas/KNgcYayz7G/WK7puHvCfq/kiM2iaeRZ2BSBuWt7jLUQ
k5TgBmo5lSVsSsr2Cs6mTG30+5kS1AgLkFaxqynIN819dpBLdybUH2dxLcGN6Ue
wXhbqtnl1pnCJ8EtPKo2puWrXla5ke++q9/cZdAx9+hwB7+PLwVPSBX06IG6i3xu
LX2b3oxXcmTsFJ0V4AXFZGCwSXS15tPx4wZPRI6l5OJ/iVxJFxaSQXwoGs/KyjoK
B9dlqppJkzn6jxmCRt494/c7uVJePG/gm6PxxhWVWP+c/S2d28cypy85fIE1kATQP
YTBSHFzorJhH2dfD+vT5WWCwE5kTsORSiuLNlct3+m1N0gQU/OmAi76cwzpWd+w8
mtbwm9SY3el48FmHn1D4RFdZd3z/AWFVCMXJroEsUYuLL08Nrfx7Cap6lXCEEw0w
VxdjdFeaKOKFTIHBoTK4XUSmEYdMcjQv3lJ4zRGStjRuv4hFlawK/vhzC0ueOjqX
ZyBjUEE0GCFzu2UvZ9P1jbPbCOWOkM5TNg8Szm5J40FgtwFXr7yZLddFKsqw3F7x
N2Tc9q61PNXbyElOsPciD5vMpCBS7u3R1TP1UNJtoNf8qz/dvoDEh4FmKiVeznCV
7BFss4q1YEQH3JwVEGjAvcSUSggIpqNI8W8mbIT65vY6VKgP/WsyugD5AFruh5M+
qlt+Dni6ywMGC+CSQ1Y1lS8bVZviAEgCWZBs2PmP3HjuAFIcNo/hPd6fTK3HVp+V
6OYgIHSca3qXt4NBogbYyNFOYwwQq/dokmT41bNzaFbh29xGlKfmOq+1qzT6bQs
ZSvZ2DnyEtnBJ0t3OFR0hWBjUObR5DCfiqjw/ckkEe5rrmS1lDCPdmJA9fWTWR5D
EDICjnGMRrzIrINPKa3stnTRXNEujHw2FfpUIhXcd7IlrWJ+8EjaZKUDB9f4X28d
+DIR9tpvYhtB9/tWX/vK034ElxKGfLP3GpYmUnm+R8lv+v26JS7jndCylKmdbcjin
8l05tyykOqCt/hyJfTc+tt13w1TjZrdkJg7lJZ4p6gq4a35vn+gARo+X+RgOHffl
/aQYY0X8JbflBOI8BJ8NcvGJ0yaQXkTwGBDlGupCzIz0uUpsVTXUtkwBPgftsM7T
adqGCstJU44H56nQriTE+UJGSj0JZY5ch4nSTF49iwRvqtabrVUucM4TasERduFr
12QCvEVgPO30zvkuWJobau3tjH0e2INzAqG8txBYO5pi8StzGJ0sIgJCIXDHHKyI
pa/V28Es6RYKpneKJLZHIE8ISILgjj5bcowaSXL54hLYr1fCdJzKgxoQg4/tUMHvm
1B5Se5JfWER3K+4DLKkZ5EzWu39vwQovY1jrmDlramOCkxSmOvoVt3AGecaW8YOE
d0j/iQnMVUwqtik0zprqVr0CCnZah+HfB0CVBqmEi+ymR0Lmtl6GoLzX/d2Jfk76
eJi9iWDXqU3tQd7ya5fRmrEmQXxZ4F36sFHaBdp8ZVj9NMocDPavBRXCfsU4v1wq
7uFEXRN15y9mKlHQc4FGcrF81vYkbt6aSRZKdxwV3zaJN+vOBUSlRAa580lZrmr1
SuQ7XH7OIViuAcjpm1FcLzAFIX8UXAXflTvg8/T4fpzIbXL5KKebjYFBX3i2PUO/

ajofkSfOwiNjrpv/0VyeDeXreFoP8XQlzxQRrST9TRTPgK2A76u/4JSJzwjGc5Uq
sV4gTCwqFd9UEl+Ls4/P6RuDyGX6g14/XI/VLxLaDTW20EccANz0lxDQJALDR340
uSqRQf6/aIzUls+wGUV1WglFYXheY93z/Z9/M18EqF/DunA9WawYlbjl02GYIIyQ
ENowMwUKzCBoth8JPO/qm6xkNV7Nn/ZbEBAOwb9i2wIUGCJT2csM5GjiqpR7k0cH
ybYGZQlWpQKYTHHxUIkTYkzREtOa42m2O2U2A3NA45WultYY6r+9/eq7bltH/eFr
KkNw3S3R/LjDvYGviJThFUAAp4bsdHRO6Vq0B8X61ToqUCenMc4WFR6B/LCJ5oYy
Xpq3wY7d1CkwFEF8ZHmIIBDcV0rgtkQK45MhkWXkNoeNCQNb+VFHAgU901wC3qG/
CQBr1zF0mzMLE19OaSt8vR/uzdoCZksxDgElgNmM6tQeSuFdZyi7k9XgB/x2e1H5
2Ph+u3l3XDhfE3Ce5QULLs5TJFSXhc7x1trZOXLC4T4YJSpIg14LBzIXc9USQ3xM
UFgw1LUPilI6uu8IJ33B8OS9HZeLmUZAHKfgJ1409+UFwV7yWB1bDhDlIEN87LZz
DWGEUfSOXcUEjgoUMWfitfFtx/UXV8OzJB0TlvRTVY2c1ZzUY1fsGYzTz46DL+O1
BQ1o1LzehE9GxkGoGplyS340Ifx/nKWvVOrPCXmFyC+1sU4yYj4OXiFjdRuDY2dD
9vOEQ6A6TwGCBHaTheLYJz/B1N2iDF4xL8hBAIk+jPKugY59SqFGvWq+LBFg3u8
oy106YgAjcrKlREtAlXDbHvVBr5u0+XHUHfsEvP/tXhZG7GD45K55TSvIroq8Ext
zilBPlpEjJra+uFDDOAhzW43BuQw5Xa1PHg/lVh3bJ2YGuSJ7FHUjF5sgfnqOHT
gnV5e9J71CJju7AqDUcmAtK7Vf/lCF9kgyd+uwkfAJLDvic5wZxSpwEmaejteHpl
wPlKPKPE4MRDxXVJpqqxMjh/eXGn4n9lqGfkn+jlSTMgYaudnyXSoIe3GO0qfEZ5
LxeuW/QDMxOLgvVhaZLlQWKg6XVbCFwW5eTCSfZ2xfmSPFyF02coVICemUZbpnUL
kXVoToyXbk1Nuh9Qyisivoy5Mz+DZetDF03042Ric70OtWl0mIQDRQM7oPCECKOu
iEEDlk0ZkG5BCFy1uSiznlBEZJR6NcONZyTDrX9haA5SsUrtGYZFow1PQXgCI1Ey
jKVwenOKJbHo8ep728dd+aVBIw2sHnhzQcn5QNZ6URudhSavSM6CQJWqOsYOSrHc
SicxiL9CMzGXJzMG8ppnL2TgkiBjRmsst0sTT7Y19TFnScgXDjtwpimlSzkoQ4bY
a0Gw0jsN0F5k7k1SFjJQLe/fau99wQhJsTdnVeUA1SgzFLiEj0+Ba6z75muf4YaJ
3CwhLFtXiAia29lqNteJNQSJJKJa/NR9Qw9qEBwXuT/T7HxqZXfOAUqsiYeOJ9vOr
iskAuLrYCHbASEVkcHYOBw==

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] 10100 bytes
  (decrypts to)
  application/pkcs7-mime [smime.p7m] 6456 bytes
    (unwraps to)
    multipart/mixed 2094 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

MIIdHAYJKoZIhvcNAQcDoIIdDTCCHQkCAQAxggMQMIIBhAIBADBsMFUxDTALBgNV
BAoTBELFVEYxETAPBgNVBAsTCExBTVBTIFdHMTewLwYDVQQDEyhTYWlwGUGTEFN
UFMgU1NBIElcnRpb24gQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIeh7j00
Boq0MA0GCSqGSIb3DQEBAQUABIIBAEgnavqzNR+Do6JAxBP8F7JUcbB8kS7mxU+3b
foHqCQ/5k096KgY8libT3/JmQw+yAifncpIcl+22N0NqaqisYJj9dKA3Gjs/Uprb
bSN0zOavKBotza78JC1mzmIIKQ4Vy9QuStaxihfghKti9dZ5+elgenqQhZrq3wjX
MYBlnGKNgrXmNb/8HVb+ak+kxK9ZiRj7s2A3HBQz4kFOr2wcga3QHrnUFqlllFw+
Qod2RDSowp7uvZ/vdtVdVcywnCh7P45RUF01PL4WVr7AhzRDXsVmYWF1x+6uBz9M
NxOXJX3f7y5+eoTzMUWhJdUwcRM2z8EIT7EdG6I2n1XCgzT8jsIwggGEAgEAMGww
VTENMA8GA1UEChMESUVURjERMA8GA1UECXMITEFNUFMgV0cxMTAvBgNVBAMTKFZh
bXBsZSBMQUU1QUYBSU0EgQ2VydgG1maWNhdGlvbiBBdXRob3JpdHkCEzB8R0APhiY6
HGLS64MvlsDXhpQwDQYJKoZIhvcNAQEBBQAEggEAlNiYwrBY5HIxX9zMW8ERT4tV
H15QupEY6aXcOJhoRBLO3hrV9mEbo9vy3DlQkwWlOIsI4UqkQQ98dNeDQYEzy7TF
onupxyn8dy6gInGpUpqS6Vjnmvm+XbYthI6xuRu4wO1PEngPucsfjE79EARuh/e
2QZutFt0PgbtvejdiCDF7mJhFEAlaG0BHfYGxD142JRyQJ81LDB5MxsTD907MOGuF
mB6+zW7NWvTjYEsSZfge6Ycc1hcbFt/3yp8gthRh4eeJEtowBFMfxLQEIUII7ImC
CesYCwWlGmziG12d3hZkXR0nHd7xu/K1aw09mdvZepumsMwHXSod66y5U7Bw8jCC
Ge4GCSqGSIb3DQEHAATAgBg1ghkgBZQMEAAIEEBmL+hRticO6t0R79DViIhWAghnA
zZMLw8xZkhor3XzGSawL1D0tnmwJnq+wKQ1+2FrIFolxv4y/rsh4bZQW0ouD6wF/
MedLitVae7U9xInumho9P6VKuhRgWUCMMxG36qD/UwZ+s1V25Qfta09rt5cbCMN7/
e5da5VzohBicz6GzschM8HuV3uoNgxsLjZ88izGE7y/yptAcusUg2Fk+dWxnWwx1
KVsfEURo0qocwe4qcTEt000dPYZn8ebKFizhwAxD2g6jdFWefs+gm1pGQhKPGZJR
g7dC9sVURBP4FuvuUPvZE2OgSjM7pJCrALawOAUCaWMD+hAU25kX6Y0gydcpszdMW
Kus7jsPUKBfs9FwjX3gJnv9BFR7uJobL232X6ufvC11OgfzAzPE5GDo4nSdqpand
JEd0Db6ZWLux4Fduy5xtWqqmGLST7KMZBHMzpo94Z8Op2V9Wqa2hJ/DS4nB4voYD
fotZehldXoILB8HO7l/yq/6AyI3ousV0GdmtpneoeKyStj+WlPVaREIdmIzup2l2
+PySH1Kn5ckfcvz1RVQ12IL/Ba28Lx5KqBgbtMdGkfRmbGvH5DEYGiA7h98Q14Lv
6300MpgLbVjjs3h5QPUYp3tbdR5fUJZvAPJno8NtI5j7KgbPJmQaCNejpqJGWx2s
g2X0Vmcsj7X24PHknpccTXCq5cAf0rV/59KidMJgjMkVhYSnJVJ35wteFmKxYc/l
lrcU1E3TJMOoUCeIoTUR6BSwuR9v4pWxkY8y43HZEeqPK+lEx6m208woHTVx0YYoU
76/Y1JC7eDvPqFbOwDCUqSCOZmOzz0R67+pxHNSf6y78gLYEIySt3n8OViIRH69E
XMChkXCvj6dlynO+1zaFMr1XiFbxzlsIeqMAERp/QNAWAb3OIsqvAeZpG8Kb5byz
54+JpGZcYQ8VKkhrDt07oE5EM4ACNMQSzk4UfABX14npNvBUBKZ3FRQPUshETLF9
moSV41zGGV1pmgOY5Rzuwe5Td3Yj9inGZ2heTO3VywoBN2iBmk2chX2V9xoU05oA
If+wmAz2SM3vwT71krjWztCa25BH603RR0bFikxREWrfbS3stygzlq+fzmsmSnBe
n/2AXYZAUUV9J0M3jz2FPHr09/y2TBoso9ExI2kSy6pgwqngqj3q/tMyErXgl6FKOC
/OXMSn40cHCX7ZZ5ud+XJLG7bAb2izzG7jjguLihCL/WxUwQM46jzPaJXcg2ioR
PAQwITRSGWuWR3qyqaLeHBRzfP5KE6I7lIdy5cz2tD7LYrOnx3tJHjiXcyTbzRk2

69yMTNciAjBsIcg4VtJ6AKgF5Clxrdu/3iOMxtjOIkfvmlEl/pOsnhkprkQVUsgA
MjXBc1OyE675ukj0TYB5Br+AJfa+beSgZ/5Hd8H/vhfs/0v6Mh6eFyaoowh8w5HS
jQg4TKzfnRyLMqQC5Us7UNZS9KNtp/g2FlYHYJX3CK2+AZfPtqn18XnoCF49rK54
Sle9rwAAa9gSYeAsoiEm3tCF+UiFlauSwzDgK82I1TCM7Vm4gD0Lwgilz8OpDZXY
U7zavy4wLQhYVdKYQH+kItxxMvdyeu4v5+Fa2LH9+V/wg7lzMG9TmutDODGUXy/
LbXagLTbrPhLqPYVnk1kb+UMuSnrb+56tRJqdZnlC64kOEOG/nLa+K8p/ZE/2jZs
avakQn7ZXZs98aQ7NKFxNqJ9rgMNB1NMETrVA5Wtty+6WhlwfPw8Au/Md5gsdYfO
wckX/W+t+87UoW99z6b8zOkFpfFNEgacBD3EA8dR8TWIgMXUm/Sq6ih1OpInqI1
bbtuCqcgogz3uKgDDMZilb1taTautpKTFvPcJ9rMoxC1HYuXhyrn/VNCBGM0VE49
lAkoiyBesIPM5UQb+Ys6TQ7m/ALazY0PKLKPEWjCqnVtMkEjHIn5b6nDzVqWfoug
fw2Tnzi0k6OIgiVMBNmx8+zBj8wflkeqdbZ3hS6Akx+lHXVeNGFq47VKwojw0rIM
bUBUK8rMC+lxJWPebgu/l/+otzeBnSipu8sIA/5dEtVxkXExEKun/U/E7qQZD7jo
xscyuLlsrcfwUd4W9intgyf/86rfJUC8yeAl4QOciAhjZvRc4X0Cf/Y8peIHRHAS
YKjQQYhQuCT04IqVOnodAzd/oGtFe3nvPu2uNUCOD/Ct66dVHb+n+eB63qeB9T1C
cqj7AAMSA56ZM9jDICPs33k1Au6Z85gsPLxzySmfk1dtcYsdN1Inh6d+olcdJXtI
1TfoRY+1xhTfavxfq9asGoQjNEtDywdi8JV8vHQ5ja5fc7LE89qGSKc/lRbTg1ot
MSjnQSBVtjmPKNit2DlWwtdDdr/aAPyK57hsXpwYMotNSqCF+L8HirXdz6K+7zBG
lJd5uB8/EFP7oFi9+MpBSm56GYN8JByRJIFlrSCyK0GdrUb3/DJSd/sdheewQPDR
ra17SMB/aysgT2xu2cPqlbr+/D9bGA4kTJ7KXp6WZ67kuUC3JtKkaGiqfcESDKym
mIglSN1W3Bf1H7fLxgszAdRRyEw01MaipgGbFsoU7sIKgjeQ56sczb1/PBJ3xS07
GamCZ6m44m5DhVs3k1vawuZqrSTRsxFBVrSYajL6msNtLXu6l4IPD6x9RZ/OCDat
CgmX2rSBj3pg0Jx+Xlnr69sVhTnq9LbB+GSu85eC/siDkcUEXaV5TzSXakElafuv
ESgOpdPlYFDIyiJmW0SKS+5uZLGvyH+hi3UdIaDt+BjlmeBmI+Q8poXc+jAqk2Qf
/vG/2p9o7EWgawklcuch6zDa3r4iNXWlKQc+lojXQdP5YhprUCDKCGe2hdelk2ku
cdth0i5lM4YqZH71hP5Mldf4uR8iDRjyTJCKGcDPVAKTpUXQ+GlbGzZxerv6XBuc
Ouxmi36H1vzHOBrgGjw8FdxIosD/O25gSjUGr5Q209Yz0OpcvUziU/bSg1fsb8fH
9us8+rlf/qHf9Sa1HTd7g93kgTx996ne/D4xtnuc6R9bcUcYmoME14u5pRkHjAAq
pAv8c0dFypwWm77RLb3SdsIqhuIQ3TK34yh7wILMHOAvZD40/jYfDn8aMFz9zYBy
r8iB26Oyc7F7Gn52aZMoLoKuNYpJCE7UsM1N2pkyX5DhkDA/JHJW/5LotmFHFsfU
mtkh5PR3c/DcRjsSimjWAW3BDvyOULgDcGE3dVKzpfCEDwTt06+bIHHpLLv+otYt
uu3ZbQCNQmt9jCh7FbEYRLixr/as8MT2HiJNbBfrT5m5yyo9jAFgl5kLMRe3SDmc
5eevHjA7ymNRPVmDPAK2yoSG6agF39CmZfZzS07Cwdtha3+YfHIfEaB8tdSEC/YX
09g9AQSjTChfX1TK/bwitDFeTZhLEhQVUK5jCFJECQS2u0iqpgc7Hiv9MOBH9GxU
FM+E+h3Osw0gPmaEGXh9+2V+tR9EwzyE7VPjuUTv/aMl0qdOxIldZWM63BHBqrPn
ylp4MId6l4zULkZ9m5xnXpBEOHQ0vbwBn8+qtfRTI3axZSbWAJAxvisUtLZDEXIr
Q4ce+BNEMHlQnrKlSfZlIcwC9UwvzDfwkFm/zkiZ34NVPWHT9ep2zJIXkQrQ/ugY
HOQVEwgHODz88MESY01Vln2rC0nFTnSMbnwSpOH+cqn8gtlogwBNYBiYfrFbCSGi
7p4bUj04MTXG6cbhZr2ztouRuGN4PWS5aWshQgc204U7mkldftGRuGxOHD6uxr1B
Yl10JHEAQSG+Vm5mAPG7txzMHLdLlsScGdwviP4TsLmfObJsxyr8JQKJlB9a0W+2
r47lIxOZ6+sTkOFIbzoCEH8rlwlpUIJI9QTZtc32bDI3bfeO4DFqUMvrN3cpS2nK
Zr62fWlcm6s64r2cJmaMnolkwYB86gwZbZbxBlyxndMcIcsKblvpFpEcZg3b9aoH
M+54UC2/YKtGc/j9xDZgQrivnN9YdMlXq/SSa9rBNGYUiaLhkesXUFuc3Q6KzxtO
sw/OJVyZoDafAF/JnpcFFt0WaSbC4BCnLP5RSBjyHXTBYhN0JWDep/E4IJc8i8Ha
+LYIFuu7RySDJ4ciLleZ29rNlceQ4go2H4GX8F+RlniC3oXHYrth6Hp/STe5svk3
ZtblNLDP/ETyz2oE/007NbRmncVQ3/rijaRQX+Lwx59bclvxleLOOomatawh0+F06
UgC9UYXHplXBnJAFVQaScpez2hene/b3WMcl6lZaWFbslvGjCQqfuWXTkt8KSdE
8ts/s1PAmLln0a/35q4Hu9gMTGT6hmxHm9gyEPNyLsNW/LkDDypIeG4KQlhal2to
JILz8xufiltXOwmIImZWyGrMLWZriPhT2XL3uwutMht++0KCCpG2v2HdOLvaA/+8E

Y9/M5N4Vd6hSNGHKapfmyrZB9ECf7jnXEkjvD0u+Er2JJ5G73e2u/vY4H42Af48k
ZEKdBg6RRK7yZIsaD155TgOCCspcyoiHKmjWKzq3uhT76aKxmdi7gyf10GOSZjc2
zNUyWzjrCiehuZ/tDFUsG1hfjca158/RPKmXKdIUBHpm6FQTF9RKhc7hVqEjXju
cVvmaNc1g3hkvBPEu7ZGsWj4iXG8YxskrKGYB3L6RVbhJuSw7QobThAIH1nI17wC
5JnUgILU3HzPFmA8A5oC5CrMO3u7p+ambSZO26DRtYELKk3TuynuNwx/UejPWx7j
S5ejy62kE7vsOEN4mmazRRxDxQC1RjE+XrD9bQR7/G0z6b0dS3BdxDQgnXIAyhLA
Iaz52rMo0qtun6gNFR8ynICetkwAgmtg+fVKqCIIQuV5zE8nw0fPVQfG2hmFf175
6+btXw+wUdUJWML/NjquSF+HSP7QXVRzCOVyLsX968iIwym7G10e+thPXbGhXqGy
SKxx7ZSw0SVDn89z3N58/Lfdi1x84gcEa2wVksffysV10IzE7EKTtU7fbzYW6MI
ihGnXkuQvAYwgKPw86nirrdHXS8nDIwjiuo7//VFzAwnqqTQxkXzbyDQJWZBzZKg
PC5GqEe80mtvanHZFYFytM8PDOxgmTbcNj2QqvTY2XK2nhV27ce7LLK1KHTTDNm
P8APqv3zVYKugFx7dyCVwPEpgayshnf9wGVfyVd9qHRb5o3LNJjxq8Pg1BpSouzN
ocUY2xOES7bIGm+Apq7eJcl2vmC6eClapHg2U/S/p2T2w5FhWnonCAhO/U8DKBM
DsMb7+JcJMCIdpm/0KbA8X55f3kknShwaDmKJMoEzVXiFMBNBLW/js9DJrPmL+H
R58bo2I8yRhYnOmvNggyk/pp/JMzm8rJtcJTyI06M1sNuVUvNeisMP7yVgH/KLGE
734aRoPQWSJA3Iry9h01I+9zN+/0GB3db1zzImIP/17p88DPXeCW5My4M0hQYU2K
uuO2Jan1jJQs3h96Ps8NMbvRqZGqq2poWLe2PvDCzu23/XIDLjPQk7b1Ttoalrn
GfTjYW6W/5WqUrILkrdYWh5UBqtPdt+N0kBk3fzeOAheh6CGtt00T2+sRjXM0ABr
9g4BF8uBE7nMYF5KorUAdmmwgD3XzHkLTFB1VpD9TOLvQEGJ/15kdudRSRQmbWMe
iM61X6D9wFN3XoYBj6Zs0CbNWzLnicoRUIgSwvndNQHUjOx7snPw7EEpTahMbIf
MQILRvKV2PWXCjikZm6b3oiMv83UINinANxhP4qdQ/yHXJx8FtUGmLE8/Ar7wJqn
UTJ5oIC02rANqCJdnok6ISSofCYZ/6ok6u6W5sA/PuZKXLAvD4N+vM2ntvrySjcb
lPHKJFOPaComOLOZ60CCix1BXAtcejVknSE835sJiNCK/LDg5I4bkoZ6/SsbPS64
MkRCaOqeEK9aRD44B+UYzz1cxfAlbUFPihu34ohFgSL5T9n6NQQ5ARPvoZYSYCB5
Z79+bYs8W/c4+9F7GAsIy9WJWuJLK1s2gGLSsf7uMkQ2t4Zb1N+sNmIL/II2UvMp
macMvSTdxAT1VRmuvT0NX9Zh8M4PpNF4Fc6Uhh0hqnHza1jYBEkAHeZytB47Hmq/
OsRY5sHEoNJIsoU00U1QyKhf3CcyWovS1/CKWoasFNM07kb3Sc4sUmLBd964UkFL
THi+6MuOQvWusX01Ba5g8XGvMB9T2B23R3Tl61XIFOGRoQ6ZOgnvPmvaEv6LzW3v
lduQRgkUnYXOYDk0riNqIZ7o1u+60t1MvpU2MMnRoNrWgj3V2QpPyV9P97r51Yk2
wL27uGvJELbcYNI0ufY2js7L3cfQoY6+4SqCurlvF8z+RHKRHdz0D7V8pb60JOTA
+/ugp/qFXJYPqSi9ipmzoA8+qL378pusJ01XG+A0Bf+T00nzEzlePwflle5pkQxc
FpR9cFHYsr6aAmOqf9nCQhzcMPT7xQkfpn9hKMFwB51bMRD8NrYY0SH1pfaEMDuO
jMNdqO3IrOTRMuHA9WYsJK0wN/RM7LrLaSQPTqpWMMFZhF0FHgcrheuCh94Nvi/D
MEN/saGODFQJuqpyzRtwkMQGvNE7JW98MFk6gHxZIXisVn5BEksPfm3EqFci6UfC
bAn1/8XFw29Um2IynHBedf+fTmjxg0D+aazX1jxeGyZ6by8Dr1JMxq0yFO1HtLcv
XwaHFeKrF/tD88VHsiZuq+ek/AAZmrD6C4aSttJIPysF5htol101IJD9tPC+UfzS
f2oD2FGKE1Y1KPE3uPlkovnvNfdnV0CPq/17Zxfa30KRZTDstvtDc5+sZNxmbfVZ
ZnbfQv0g0vo/E8iG5V2Y+gVHRhHwIR8E71/n3JXV51xmchvvJQ9JNjH4siJEr+sH
7j5oXuEeFqWsISVHV+dlXTp3GZvTAiH2qgMDgbGSP696+VXsTp1h3L7/PEYKQkCG
d/nts jq4mGQhI49Je4oCq3+5qb9i9gU1H6g4YFLL5vhkumdkL4mw8KbQoF/0kmGS
EhzXvd0mPTrlSb14ObVcjh4pvhLJw5uc/AHgCukSkFde3n7Ml9mpqgcJzfHTPYiK
lBxfy000F0ZB5KgH/evozRKQZT5mLO0oFwBjtQJkGxXBhcyqTyCb3/zNGMonfk3P
Jc7+ooybNn80pZzYHVTaYT2MNVFqKfy0GHMBA5S6SaISzoKtxR2XMwKAMGqInt95
Ie7dK/Ief0WhN3iCexZeJ70dAFYmbAgghJYEFyOjPb1I6p7div5cn1R87Q45UQf
2VLRlOQvAR10yNk+DxXKFesn61mejZR+5HeeLcu5h1dOR/broo2IrZGvCK3oDWyV
meuvKtWP8oLn49fA20K56nG8OfkEKXNv/TvN2YqN51lNkU1E+d0v6vF3jFWbuqu4
al7lighPkUhWrVbXtSRydmNA/gjxkj/hPl1lMfyiVOIfQ1wrpUgVpH50t4+a/cYk
jqtEEqPQtL9Jf91Y1i37JJ0KI6mH7ZiYXhcuPOGEzdQxj2CZxCZgIwe7Lb6GAu75

dLAIFwtzLdkKfFXyVlZFKig8ADzESpevxu00TkNfX2hs8MB0nUFxziE2sY3XW5ih
vvaQc2o2KcpY+irZj+B1PoYPBaHqcxPAYgK4pdcUqkgjVmLSqxqyStrMYS4/g10r
cDWhFpYUAM6i55g5ojwK7WJ5HEws8+yUoniq1/d0PsiSfGOxm3P/cflbPHsXW0Fm
I6FO3TFT2eQjLU7ZkZTSq1TrRH27EHYJ2dr1QUM6aVKhSiHdqTS5hhpanPwfhd3+
1TZnWC9qLglpCwWjut+r9bqYS2hyFLbR7YCT3+jybEQBXDHhXy+Xy9jixADek9/
IGKnmujmTq8FlakLgilpuSBFV08tOTrIiKZ9jV70/un9T5IIq9eTPFu4dw47q67w
SUg+ped9JU1iMrer4gmdppjRIYheCUYSe9/9wmedaHLYkYnjNzHqZZ1SlxROM10d
zPe7heqZGurSFVamO12TKGYMYkPg9j/X0xejK0QQnkW8zP3Ptbb2z2u1/1IwPAQp
TraAOK74FHKLCkQV/B7Vc0TvoLbyNYWLwQkkLqvwLVb3FdgWSjO3ed0V4/lgJF+A
DRsBY5DLNf0hSXgfvGma3kPkN+oD8u93LuIFRJP8+fGjcb2bMC58LpyJhhVUHfQZ
JNVxhWn0bzuF8VSZbyek2NeIGLKDcziLrKB0ncnkeD9Yry/dgHN2ycWijJaI4TcY
ixuCz6wtR5zzpXt3tPuY8NSMMFLW3+SH+gwGRpLS0E4QXbFCdsWoiuLduN60A2gI
5pICZpsqe38z3M1yL2yYc0K14BvvN1AsDNXnAET9xadEyt+wHDY1x5VSWONM5+/2
vgBq6YJnDDgP3fLIUf23nYDH8RVkRvewaKFOB1q0TtWwb6mmTVXFDEEsjjshG9uT
uNuGWi3yej3Q00HaqWZ1hdj+gNYDBikIEyvTRwJYWVELYugW9KLJIBLA+Ha4tCbd
MrPj2jslCXcU3jznPA0f2elPPGC2UPhwFEfo4JsobAGnBbJmKlrFkGt0Cid4KjOq
hBJzY+nG18Lad+pAhPixagmYYr6L4g4aJADhORToqsuIleCw1MfGxpFYOhbdyJL5
NcQQwSZKRGvVBKuRafoCivkGrxdCaYbTWS27kVSvT5T7Y8REBMv6akipc5IrUi70
ouSl909sPj5dz9kJ0RPqTx1UCUN+5LTuzWRxT+EyOLFxx1CjibP8lSjovji+KG1F
yuHcQh7v9L/amc0MAfKv0VSMKJQuGGON/BaIK+yVidMO/P3VNiDHloPi8Aa1xLv
V7aAsUeu44NI+V3dnDW2KofLxCHsc44U+c/dpkyJWijRaoejiZ4U5G0Z4RxNRHI8
cov6b9CP2WhxfoCWqatcsg==

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] 9470 bytes
  (decrypts to)
  application/pkcs7-mime [smime.p7m] 5994 bytes
    (unwraps to)
    message/rfc822 1813 bytes
      multipart/mixed 1749 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

MIIBTAYJKoZIhvcNAQcDoIIbPTCCGzkCAQAxggMQMIIBhAIBADBBSMFUxDTALBgNV
BAoTBELFVEYxETAPBgNVBAsTCExBTBVTIFdHMTEwLWYDVQQDEYhTYW1wbGUgTEFN
UFMgU1NBIElcnRpb24gQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIeh7j00
BoQ0MA0GCSqGSIb3DQEBAQUABAE0WeE2CZplu4oxW9silJTfwzOsPhm847d7z
qIXcjfvT8bDwlFtlv/4KmZLDPdBnuisuVpyLo4nnCIwQJYpQgGBT6QS+49zKBE6
MCBAAtAEp0lEX96vni0EnBTirqrlyTpyCfovzY7Wit0AGZtagvTDbUFZ0x1zspCwd
jrQHxNGnPvIUgWOMZvE8xcUU7goh5lIMlCrTS0701VvwBcAl36MvP2cq5fMwshaq
5sG8Tisa8scczHgFPox8g4dRg3avvuiPIeIWlhFHs jHOyxK//eXvbIAPvqSX2kkN
XA2WosMZFaOFDbreUYfH3vXXKhM/bN/ppP0j79SP/Oo0zcZNRfswggGEAgEAMGww
VTENMASGA1UEChMESUVURjERMA8GA1UECxMITEFNuFMgV0cxMTAvBgNVBAMTKFNh
bXBsZSBMQU1QUyBSU0EgQ2VydGhmaWNhdGlvbiBBdXR0b3JpdHkCEzB8R0APhiY6
HGLS64MvlsDXhpQwDQYJKoZIhvcNAQEBBQAEggEAHCEttYGleFD18WMLL2c j4QA2
9ufo9YrcguLxRESaQfGsdjNWumX+O6TbxzRXIRWUDM7Fgya5itiSeRX9vVMPqmoE
IqvVaBvUJrC/vpqimtsZ1DzfMILZS++8zKvhe65KULce+nV5uQFdCqY0haaC+r6Q
vo/Ync/CML6Gjnp4wpc5DWfXawIFtETdqw3OlRjeC1LN9x2GmlrZRG4Ae220cevY
fSeUgEwOAhN0JK0dKJV2FTaSoCvlsjSpqeEvrA/7PPTXiNhX3MpW/5LdnLVrGLWi
nf/8vbIMVR11a6OuX5LlEbtuiMcrDBW37Fz87G2WVfaLEGK1kOpuAq4Hva6UbjCC
GB4GCSqGSIb3DQEHAQAdBg1ghkgBZQMEAEIEEHX14xKi0oQG2bn6PtoB3rCAghfw
CwTQY9uTkxfjYyQLL9GBme+B4ar0sIhiueLsLSpDqYscvN0BUJ8d0xE+TpJm0IbY
yB8K+Xu2ZuZEBKHDM6gkwMjUmvzrqaoFM9JdgEdV0xrEEAtZ5fo4CQSQdtOY1EcC
gXaeqcek2pnEtzdMvpecyxJI+Swcj87MWMQkZC76ukWAJAb5HrzxWR1KppuRWK1k
k4dSlEU+tkItRahClnfRNdHbi/N4IYHFR/FS8efbDILhfnCsNrzhirBKkn+xcm9S
ICK3vs9rLRSxMGD2N2gvZlni jo/rIS38E8qPvgabRYasxvJjpm9pYnw8bna19NA5
hH44E1Nmd5/hF4Mez1J2HU4Fm2illB31TE0MPz+k1U/luNpMfkgBUnLEzGchYr40
+BzewtTsctonsul06hFrrHim5LgtaRxiuAJXnqmArH1N62eoFxC3t5GW10209d6G
hEFalcWjh03xZfVomvog4BUa99tRlSgQf1jkLuSGbYr8mfzufkCnxOzEZEsummqO
pGaxc4oX5J4ZiiGCM1K9M2L1/tDjN48CcZ3i1VWB/Dqb6bKHF3eEoy6qQR4aPWeL
OQxiYK+mRcdtzOMcynvgGo74RmLMNk3rpjPDOM9ltd++8stxLRltZY4dlOfdlwu8
pO53BAi0nPEwze9ApPBqp5p/bPHUp1lJNAGXY8H6tnhgZ3x3RV/Ji9KGJ6GJmENx
SVI7r714zXjwM9FJHqCmzI2DKr7p5ysqZ+Qc8mw2CRsfz60LEKA6WGb0NlovfQXL
tTq0qIOhtYe9Ge0ztbKKnzbZQL9kQ/32dbfKasQxDczaHjNZ8dNGhNr+BQ5rVWm
+8FwxmvMZDIX6Py2wbJEREUGCGHh61UUGiX3G1VYFBnqI1GUxBUVzXxvGJ3cj5t7
4aX8GRvMBrZQxhwuSLxSFQ/rPyTAusVPphPbwAoav2ZaUIlblLr4yHbawssp81sD
svgW391I7SRDonvdo2+qs5nPW0l9leeD9I9wvZM8AQ5q7mxvQkY7WDqX2J6lxxzS
jP3+jvr5vGOYuPGYGOZeuFSZU7HZGnPGFRk3tWG5Q1dRGPi0TWXzV1eZzo08e0cw
K6EuDenwxOU7i1LpC2xRxuJgdN4adAi2+A0d4vyJWxIvkQtcbzj57ZKPt80raQXJ
l/bGRFGynFRuXE510jBbwdBzvseKMOvfNFqB4nv6FMT9zVpGsmpeSVDDUdKLDayO
sEqeOV2boFAP9EvIpmA6i+G27ECsh9cTU1YXueOdcBUHagCS9DT4oNt57euc29b/
yKd5Y5iE3R0v6Vquqewtpw1GS/F2De5x3ETXj86FmcML0aZ9Z2sZMJmVy/Dw+ixl

bjVKliDg/FQZzGwsyynEcBARKvdKwM07/oliYy5n8OouKlmIPUyUmDoix3fSlz8/
RXV30BYKER1NHxpPPxzhD95ECeWi68toMliKaMsTstv23mJNwAEh6TrdfXL4Ls6
HfE/32ohxglD4q+sKg8V5QG8wVBnGpwBXD0yuUxewye08Xwlm7Y/PbCJvuSEj4G9
zSOPXka1ViH3tcnFedmyBugNw+Gs1NHCo49wllf2+UCpaoJcC6zvD8gdQ737G1/p
tLvIrC6FZa4CP0PVE0omraIssica9iWZT1QaEWDZDSVlQQvBLfBpYA90XUHxEw2f
8vWTvVo+Wmx0nZMhlU8sen1kEcKVJNuRC6XDq3fHpVJXnPkdVKk9ssvJ8IfKPSL8
4cpG9bV7RrGymy0q3hDzbzCPVGe5EdT5EaQyQRiHOjDYx+SGyyHdNQD0nDOT6nh5
C+guv89wGLYFJnjpYOpKW9Ex8yo3Ib4ArrGLTzXqdZaMaA31oAqhlOPkfp15xPSY
cLEMnTcEGGt98VSHJO1Ku3WDSC57PYd8QJsoFD4ayoYwllm7Fc1X7CG3s4i6eJOy
evfhxLQLiW5NX2/xkCnEHhZ7wWyXc6EPA4CQw2Rz0wyYEjEj/JQbcWqdn9eQnqHF
600WW704x6zRtVMKYNkvOreAVL3Q7U5EyE4ralLZNc2E/4caDxANP7mXW8x+8QOx
uJ7KR4z036DYCtZvOF07d9k3w1wgMSxwJkBGiuIOP9QQ3xWXE49TncQlTiaFV2sN
Fc10JLlepjTDCSVilU+Jqwji2DZdAfeLtKkC8Ka4D6Bg+Aovdgg/0ev+dj8Pl+ek4
et1FTQ6Db/v2POfdiWLFdp1XzSHsEnQ1NMfzvintSUSfGB0qOWFwPUj5jfH8/4hX
D0pxPixHA8PI5/3gSPho+wxgnbsd/j72VH1A+S34IinR+OH4SW+A8qCzcF/JGP5P
2TSact6pbdx7dfdlcW0J+QC8ity5APj3cOss5XDe3gs95JBgZ1AXEHypZs6avgoB
empIh6BBYeeul+NuXmRxpzLQbsqNwivPmTK+Jab2Yw/ASZdqyBHJH8DLa6xi8yFI
134xG6zMMGqW3Vnxa1IS6ops1DenfDz2hCDG9m6J2CTqMiY7ec3uoT2QysRPjmL
cx/gtUxS1L3lu6dfC0buV7dcEzuBG0H7m/Lja6vk6Tr+P9D+jlcQyUExDvpGnEOj
fhVRK//WmqWlxJ+su/yMvnSj9e51K0GC3yYmMem8Zyx7xSWOXpnBrqRf/T3tCAHL
P4DgV/3jEjFtu0PKV7Hx05YEemLzppQ0GA1IVvnZa/myRLB//x1qVATvGVc7EFhr
vKtr6FYfLfa7FUDMiDH2cxWx6/Zit+17JT/PJaKTspmM7UuxWh6eBMEld7GZZMT4
zaYrPCTvK+ykLj0FMs1ddbQCuD8BROzV/KgmTpiLQSm1cLpkGSODxR0K+8YVXiGQ
tOyNFEDtniIJQ3VoejaeLPX8YnHJPft4R9qAysU9wFdGJ1VPNCuDH29pn/i6KAPU
Rl8ALoomj6W2htvLQtIrnXICrKNpvd3FyXS0/+kSqT1WMfK1XdaYxYK1f4AR+P5A
PGsmE9TA51fkeYild3osdmL7/3n+x8L0OIDVxps+XdAk4MsQlnqjoazCysc+v6yi
Y+eMl8nsaxiTt8d8JPS9BpBUi5N1TlCmGsdoYEBjMPEso4/irjuckLKxRDb8S3U0
o6eo6x5IrEQK3/pw6/Vngiay9f32Rc64roNaCKcfgSl4MFJA2g5I4zIjBCL4stzN
E3tHKN7dCggwABOSxThj1Bo8Q9/ZUPRNXYG1MduAWomNV5SR2tUChA+G8YH4ESNv
M74R4Ij2moY9P8P165M4iKWBGWZ9eHwgHKZTkDBNrOvfwJlCdrjinDhNUWRNtFB9
hkUY4ZAYqInsedNkZRI4PpSEl3jUtKHILRx4O55De37pwSFO04uZ0NNn7xhFyQYU
GXV0HxOHt+AkaF9PTLFb761N7WJvPHF43G16EYbOVYUDJ8XRktk1AMX4WH4bNz1n
ViY421calq1/NpziXwAUEBpKWm8BR6mcBvZzNWOW9C1tQjWW7JjK5FeRLlMYDMko
r07Ra6N4/3ZCk+e5bNbJUDAuzb8eqdmGP6X9aTEE9IM+sUNeSOCZsAZtmOknyU3A
0eLkJyhZafluOSIYkD9SrAcsO47mpycYfQhREhwCbzYdM4AX9y0TVsCmVRWBznMK
z8i9jdnnKQYsSdl31h4ZezvalEf4mWGDWY5bdXYwTwJfaFRPNzH7JqcMrQrgWJ9C
7Im2YgUbOfTCqfxbVGZLstzRcONhn1v9yJxm1LlaaC6fbApPfolBzXStoXHG2FB2
ABgF+3DvWtltSShKbmQUE00Ppn2uz5ghChxt/uUFupvAntbIoHQPzsvB3GHiyN2p
pGgScgaIelUp8AUA/htPDdY2Ia0hLmGaxF6lpO3yt+uzAaWE0CSSsUJBBAT+kf2Y
8WMH1+54KiyyuJkFU0Fq/4JQNQ0/JvZNNx3M44rpuTPwpecL9lygQmQ2OLphlKyJ
Ou4B8cJLexmiUz8BHotB+xKWfGdnT0OLzeNni+f8HzBPRivcWrpdyYyG0J/YZnF3
5+tbP1UsLo0G0jtXL1Egtg7lpcgFv2RSDzYIsYMI+C7evP9r7GPoZeqQoU5d2fh4
hi7XGx8Hz9F1G+qDWhCj3JQUjBNxIPiEbPlu3N5ec/lzv4sgUwNkCcGKooPpm2HT
ddHIYyRnAGm1/om3HwMiZ+pH61slauPah6padnXHkX4uxNwDURuSFbhczugAG4Qo
UDpgSuRw/51av1cLzEN42Y5FFkHWpVZSXF2+XTbODGYOWK4B2rD8nAP5XGbbKpOY
Zcu9I3Z+/jSkHo07NFK/SctQmcrkz7CBG8Zg4E6m1XTdI+G4pu2OV3AWSfnnUKj0
4WnRDhyqPb25EN1dTQAGm9R5ltwb/lVxWqFKjPrRWzkiF5ZFkjiBfPwV2uqYhAeJ
+KptyupEN67BuI887mN/v064HR/Vz93Uc4b2ypaOb9ZbMClgbmGuV7ckFU6yBuYd

RA+KadICGwJne8vTRf0KnUlccldqyz/Zz+uNZy9KMx1E7DtDOKU+0Zydl4Uoeqzv
4ExE9pD1QIc+XHvxeqQGk5wAYqM+65cw4JOPDJNTlKGaohzpyiJIBBMvh6Nlhg4/
Ac7lWYv8yIczLyNi4wR5Tvq4I142AH3h5y2pZrUR2yTaB6iCYA+jClpQsLpZoTn/
Ry4x/8wxc6+tXSXsJkTWaZCDyEIDX8TXJ6nvcDYQvLek5sLf9QWQeSU+VniT8jUF
vtC5q0Y7BXCA0ymKtHFSB+rr2jJRT+680orbac2nTacuMF/YcTKclX0TXbLRFrqd
hMsu9An0CLG5CTHIpb1VXhEzuophyalaWsXkfRkU7EteWNIv6Mfg8ASVyk7HTtE
Zgn/i4vhp5qzEB5ule1VIOevtWmYQxuIqxpqonucqf4AH321C5S3/G40aLpJBDS
DKsGVxF/u86KRZRn3euuy8aTz4pKxSaYp6IFpA5hNZYU8vk0YNdlwFd0K+d+JB4b
y4tm7ipaJ26YgWE3kX4v9PX3v40UHMqVg+0k66GF000/bveWv0wg0KtbXWab9c9
x03ZRWto0h/l+oylLPCSRonVbBoICJ5VHgME/bIvZUIGQMKeWv9f3VQsI1k4J+e7
JX7SG0bfnuMczVS7fz6FEAV/k+1Z9HvJGXLfjTLXAJQOU0gZYbsr6ZfaAWyUmgBP
M9BT4M6ucbdvNdKd5AFMyg/DFoH2yINOBjXgEOio+m+5x0YAKE2pUn0W/9xaw+zR
abZTJHJdEdBW5YXiscG0MJkt1WWVjy1fGq7y6mgi0XqTMf6cY57DzR9k7hmywrpT
6Bg9CStEDPEub8kNy+IafignKGkHdVwjXCC1Ly2U8P50sSiFmvG+9vukY/E/IBgB
J2x8j2OJQ6FaiQ8PBhxVo+gudwZTQ4NKpgCiI xv2CHERaI8ao+DM4uNmD5T/Kaci
QWWG0mA+SA3KVvqMreaYKnMmwvtTXbet8zMLHy6knEIBe0v4GplsLsr7IugcKANl
q/IahiURHLXnsmrLVPjojdzaK7uUJuuchZsuuYVJL4CnV/Uo69XvozltlZ0APY9i
apIFDpZuF8tTBEHTUluY8mCY918T8CqIcFEN1N5B6cieWhbNCzgR4C1Xl+YsCGgs
O9dFktOPKIMJv1k1WpDVIHb4Ae6Ogv6zIUmfneQlGZzYksOauSQialEhXYly/3Zo
vQOenTXQDo2WuPiJohwP3Dh6qQuDkqgPmnhZ0EggbxvT4xVAvRc2jwOag96XwqF
WcLgkKDeIcORd/JOBuCyMNPf1oQT4Tqse2TrGgRcbxwLrUAHRhmYhuzvnpjSt9x+
LCzkF21GNorizv5Nc8sPSDIzCNKjC725BS65BUaRBQm/XywyZ19TkQ9tZP4vkQ8Y
YIuejmuJFpu2WD+IhoLVKZgQoFckYjCAIdXK2XqYlpQFfUmcYmlcUbrLlyhwfVZd
PMFeFvUmIwmQxeZv6MYTyDWg0OwRLDAXsBlrDER0GPbxRsz8y5xrlNT5oayp3Ehs
JLdDuhChe3i/TGFHIuh2NUPBZsmGrNCMRCx8ersWKKKATqGm+344paa8AaaQTVxb
14Yx0JGR/21YqdS3NvnRwDDtoJwYieQb1rr3xXae9vFF5xXgtOCMMUiyu4GVuy/4
6FuDGu90AzayfOcjtpQLYTIP+P9CNEagX2y+/Phsh9lw3fbjkCWNG3/AOI/u+L3v
gyFaKP9wfi7uzcebxDlotFmdwSzLv04idtj1A5F3djh9ZXY/R4cHqVuPgTnTJ7YE
Q6NzLEH1WB/X0xX2w16GwA0k+hFVT/MX//+a4sf9dRETuzqbetGyvbqJ8whNQeh0
7ZyqtGRPxrBsipaqlA4NMTTjeT9usAJze02GuQK8FwBBhVXAKSjeyWX5eKiSiIlp9
X0ytTitsmax66xCgjmCU6a0zuGHMvb/fih2RnuQZoEVmU/YK8xPWsjhwR2vOo+HK
k0XPfZ01DZLV+ZNMn28Y1wtfBWt6EAqKsQNT/pdDWjcbnq51NOxGaK2yIuznyew8
KGk0I56x7sixMIfiyelv+vH5OzX68yxjxJ9Wf30DjclVWTS0rEi9DcPSXN2EB0UI
N2Ovqz17RjsA5+YDmkjk+DnPUrKJ1IW7B+7Tyx8Xec99AbsJ4kmnw12U56H1qCdR
HfOWgI7Ci0Sq0gFozVDV6sA+AYuDGURGaYdWkBM+4VvoZyb0ZSp1XW5TfrppRnmp
yJnmUrRWotuLYxHnV1Wsn4Tys2KAXYqbJSj0aGSuUXQxjzPrkqn5cLwxstaHUYr1
8TxNpQd3uzj2E2Y/Ud485aZR5d0VRA6GDqZc1V3IV3eYDxktBC00K8rT4jhBsUkq
oOEBj1HqIrRVXZ0XdFAjU05ihzgG1vTB//DOI7xzpmf080/ZREtNT7LubT5q2EEe
M2rJYeOK4anWYGL1IIsck4o5rAT3Wyrq3qReKPAk3Vo9u4PIjmZCX1RE6Yp17B6i
MoA/zdlp5fg3kNziivSSbTeMlvR+Vz3XD3/6IeRz6sTZJF2+J18N47+W7yxPFFKHM
mialKU73fNbJXXp/4/19bZAYFQoatqCsxqTJSAU17f6k1XVYsKnsnHMiZcvlJ5OP
/2Tg25JB4Cuif2UyYUDGTW7ZAWSnVQ56eYYPISqJE2+PBGC7a+7bKZLeZoRpzuh
iODsg8xhw+oLSRMO5i01myoPWxJV/hochADOHY+oyk+9Gy3YPHwNUYzAr5glMYME
m+BA5aY9992411kL6bs3JZsdROR4/m+eVBhFGQq47jeJWWcPT+iB9/jPWjfLEnzU
bK95G61z2uXASIDKVR0PZbs18/YjBHsgELlVgYXG4pnLO0L+jEEZK4PZHkOEFFZ0
0cGAVObokXoIYr47Kgy9RcxZ0APK3G1KmGzCzppqu1x981MyIx11V1ZDKFwRyYcZ
eZnQX1BdB4UkDTHBBqBDWXXpBHqe2lwrzrNDUTz68DegE7Fsy3RtNWBXddYnneyg
6w/rfYkj8i5prYqceBChIsHG0HHoXzpdKAqkBL6WH8klz2Iw3NuyDFwq0ubXHRMo

```

W8PFxlyh00cdfI3aecM0l7OH+eo/fFzMpQ3Fc9VwEYgFuMmT2BoPSeDLWpInOAKn
5p5sym5uRRfrosszXJi43DkQJuOmX8gAHM0IfdKkxC61x/GCQER6jLoNBnHq9egY
V3lzG1PdL2XjjgJ7Gm7S7CPTvO4uPi6/DW6xIHS1N8yAfvOQoORvUA+feom8lXkH
raLUgRGx/mMyAjvnDpE+QKvXNVRqEAPQl9p6txnh4uB5BvDn0Fvgqvi9TT0Zh0qM
m+rKKr4yJONSwAktkWlr+h8JdcOonx3AD8bMG2v6jNLQC0D8Tab2NGUiylruhF00
iGXn5rWe3q4mwmJhEOgTeVc42rURCOjIrh5njcvwm3kMIyoF2v8+1FloQcWwYulG
8wyAGJytXy8UNi/W4/MR4Td5tVNn3sXIjoRk9sZ907ILfIU+4c7067N5VtkAtDPT
BnyPvEaM/hyyXTxOZ2kVXx3pC2EB4HNQMI9AJfWFcpw/tPupk5JRf2bs4CD06tB3
GnPORggcMCjGh1IKY2we3OW+38sCY/lXgYd2FWOXupYeEytax0iQn5ZcJlMLIzQl
vAtwSP0ighGTimF563kRlmbveO5H/Tu4MWIj5kr/88nMMFWKdIY9FGONviwfeFxa
Ieem/FtXVZu6dn0kCG5Hzkwv5ITErz4gaAJpbCWgrb4=

```

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] 9490 bytes
  (decrypts to)
  application/pkcs7-mime [smime.p7m] 6020 bytes
    (unwraps to)
    multipart/mixed 1779 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

```

```

MIIBXAYJKoZIhvcNAQcDoIIbTTCCG0kCAQAxggMQMIIBhAIBADBbMFUxDTALBgNV
BAoTBELFVEYxETAPBgNVBAsTCExBTBVTIFdHMTEwLWYDVQQDEYhTYWlwGUGTEFN
UFMgU1NBIElcnRpb24gQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIeh7j00
Boq0MA0GCSqGSIb3DQEBAQUABIIBAHU8bGe/H5LsJ+SjrpHwt7+3o55WiMyCIM8u

```

JDC68NB26HoxcTlKAtf33RWDG0EF3HshliusIPEIu99f46HunvPjw3oIBJlXcMmQ
8CHOFlx+iX82VOPuiW0081W6+aVsK3zZF8gxiFoUh/Z+kgL06L58OPM8v+V2cwIa
ApYX+6UXWvVY4CBZgpFtv8/L5tvwIFX0Zv/Yl50d4U/jFzc7GVq8Baz9JC4UjPrw
5QYctj13CCCLNdssAzgxb0Gb/2qXUKPKNel4HxCBE9tWvtAT6N0pJ42iGEeC87yy
RRk8MhzpaVghBs84p17CCHt/5e2x0Db7RS4fFxr/KHjy0daW04wggGEAgEAMGww
VTENMASGA1UEChMESUVURjERMA8GA1UECxmITEFNUFMgV0cxMTAvBgNVBAMTKFNh
bXBsZSBMQU1UyBSU0EgQ2VydGhmaWNhdGlvbiBBdXRob3JpdHkCEzB8R0APhiY6
HGLS64Mv1sDXhpQwDQYJKoZIhvcNAQEBBQAEggEAjQA0LUQpww5dQ6rwcqcfmudD
4Vr95tB9KqWfa6dQkQ+ZGQPO/rJMcL7aH3xKJZai1UmzD+B7Qk12TVg/dCCkyxHC
9OIIRVw4Hd5H90/K1zxuX5D8bTFsZrbgQMhHTo6GnxZFbkHrW5Cj/XDYmpFSdORg
Sl/IpiWgxp7mkCM2eO5V8aQxf7gYn0AXW+IWIXnG5FsSO7ViTd3ar+/n0UhZDuYQ
iE5Sn0iw15b+snWR2u6ECu5COerDvmQA3y3p1DTBQzGpJnj2wWxkSqaunhJsF6/r
UCaRcXnjTtoFVWegVaY8P/5ZB3J2OpzJ2hBazyYi7t9623Qd03PHmT8/LeDN3TCC
GC4GCSqGSIb3DQEHAATAcBg1ghkgBZQMEEIrz8pPRFcIaD2K3N1GCAaGAgHgA
DGzTGc1dSEpAV7+00CnAVac9cDEwSOG4Loi6QQs2S3iKN1F14B2sdxpf0Qq5uvGu
vfr8Q4g6fYkQleyJdlvPLjaiA43chMeBl9+2qZVb59rkj19XX42EIhtSplGy5/IU
5S+BLxju5tV6lkj/akkXKotUDAAv7mtZqQE76C8W9NyLj3uKBAfKNngz7KcSQXz
1Cc4CTI3/S9C2BmVlGjKwLYxS4ZD5S0CEuc8NIUCpb7WzeesnVkn2ZFzoq5gLnrs
DBlgTxu8a2vUlcVklYNoHbqC+6IzkCaJUzo+372R05nRLFlwWAXe0Ur4yRa8P0rT
U0XJ/joOEIQHLkQ27DKmEd8DWEyJZsFz4uAqgcjh1kLrnCgSGU3Ao1KkzXvs9VVI
Db7+E73GQJ9gNU4dDjY+zrVC/ssM8Jmw1qKQ9SZ/3p8oLL8LVQ9hiC2j1XH03W52
yqph61WkdBL3snl8M4fCre7ukmBbY0Z8JCIFu2lqvMndcvuIy6ygUH/Mjhtz7soV
6E5/nuTKWZgQN38LFnm2YeILU0GsmBDjwfyV8S3aTRoQPk3ibMAICJi84SdzIo+
jNrYikZK3isLflU9PDfle13cmmLcZibK6cceDwFyjr5A0RNgz/D4LIEsQWaq3fW5
sw758e8mWCB0yXG317vh5TV2Y3wTy8gGief1Pfa8jCSVu3xnMxNARq9AcntgEUGR
k3C+UefHb1CWPC9+aW0/U83hFyelfCkuAIS39aFVbI3WHdruUd53cE+D1qaZ7A1X
Dga9uH5Yv4RFkrfbrTntd88k2yvlK6sXYCek+MS723E6NS+cRxpUk8d7qIbIXhDd
VAvnxtb2qrk7LB/lK0rvVyI2UaH1xh6Jry8TjJqxA7WpBnQ9EM8WBtruzQmqmg7F
S+l8EdGuKTqG3xhFjxK6Y/k6XndRiPwn+GpRv//11AFqbdm4eJ0kiG9ieSfzGoa9
cKQYw6C8u352uDaB6Ek7GYXMH7dywq9DJODtpoJQUWr4QX+m7Q1qmpljgLfzylib
qt1Zw8fYTq7FU3QIvmFTZBYkvoU3GLQEOWBS1rPGapUN12ntj/arj85BTODMZVVV
m1RN5qxtRJJA9IK5oImYheqq8T5wBQ9gftKDMVdb6pPNqwTu3nbjyItKA80L4k
c8Ibgh5bTunCVMJN1UIdxFoOEudnJzircKB7A1RfJDlgDq0WkaIBQsw3YV2npfaP
D4lkf8HvyvTE6QbEDurgon/rDy6TQ2+buPgrsoCRw9+yvm0CHjKDOvk07L5ZNo5M
LALNBBrTugyyM27hkmYKSjGx9740ijlZj3eK17DQ6XP1hxWPPfOFCYCPY5U9440g
1unbhT+q3F4x7Lk4U72603gj25h+SYJiAf+5jRCCUaOpjAaG4ex0s8kdEznvSLH3
0w9YmZr7w43Q+3C0IY/du3WCMkj0EgNwEALQIo1j3wEVOWIxNsynfEP7ilKGWX
/L4MkeACKDDYmbXkvM70khXuH0APAGmw5rwuEUH2Nvr4rTvRI7QKMnDJ9BiNKK6A
e2gySoYelX8c7NeqdoEVUUYigF3rB8LNOqOHqMM7AAsAyt/yjFYVXxze6PXS124Z
ohTlT3vJstmrAfSsyzc4q29tU8AiY0AT3xmUe7lN2/QNyzHIrp/KjC6OmNFvcDLE
dTXLSxCUSLJby/rJ+YH69BJxledxdfogY7JFIXM3+4Hii5/JAsuAGkGpjsmTvc2T
X9p11/08ChdT5m1wRo0PqgtXy3Sfyc4hlFDhDvCk0kP51Lpr9YHe51HSR5x52/+i
mcSbDu1LU+2wNdu6g8+OResU5LvI87Mt0sCFRvV7yawg3gIt3tZrsStS543vilWd
+rZ7NQFC+GK7wBeP8xcGmb6LgdxTpJQmW7bOfLkIzXQHd6cd/Ezm24X5WjMkFKeB
HRJPGK8i5FYjQW8I+26mctTjPmo3MN4m2aUzU934aKZWnnlHd21wahtXB2Z7CNJC
7gpsed8peXWUzQ+ZTf8nx+nMpQ80dB4CRJl8Ah+GWBultkL7P1VikJIOQWE4ef5P
+wSn1phsQDeZWxyIGjCrcDwah6KougXOu9liqv7Hcy5fbgSDH0dWTJ+mARcQYiP8
EgdkQ0rmiJJ3INAc1G5jle4545SJTrIJqC5j2q2oRgj7JHe515QlIfzpfCNOxi6v

Cv/51Srh9vovy9f6SE92adrBuYf6m10EpR0UT0iYHKPEwCFkA73K6X8crEUXvGA
PuvzXqqC1aK8kYcYYUKDy3wkY0L4XaO9iNHQ8YDC0bwUg7Gcexee5H5IOC4F1lRk
sAGVv6QwESYsAikD1qS3d+IJC0DLasJ3OtY6ibSjNBs64A/SWxSVgrmkvYUK8GYs
bRoLyedyYYWsaJLIE4w0SR4LEcNAUsS3IXFgmwzuZfwI6++kVnYnP/Mzfhai3pFOy
CWn3Q0n7egRd3athFzhalQMSO/F6Nqvp0cj/wQu0EbeVqnnv4hEi/QAVkzH6wWed
bo0JZaEOEFHHVtK5gHqTbcD7tIxiZGIri6mW4CbDxMYBsMdA7D+CfjmFedVCZTZN
Hhi90An3agODUXbE2W1tKMrUfxwOS2StF9MRWmjUtoqkqQmp9CSpucAxs57JThER
ex/IkrkJZUZ0dss7FoEB5kple+JLA0Ilq2EzakCkcoC60TkTY/X34c+azZPLeEDM
vfNA5xqoiMWOotE9WDh8w1XphW8IHD9ixwPCaZGUNx75sQjqQMxh9UcgRaaolVfo
XfjktjmfHbhTc/J3VyMxgvcS4WIU+w0Ru+DaDVzL/9Kl1Vdyrbel/SDzccYtDax3
RpgWZC8/8h996H/Xr3p6gmFS10cQApU/SlvU67Ka6A1aBEIJnrIbv0r7hefAJPe8
QIEyoz5WYJfaHphSg49BUuS/vQB5XbvDEbJbTutsF7NWd/6/8R6iNI4iRtFYxrSn
QCu/yy78iomVpwpFR5qdRpwIiyigs4Do8yIEeKB3WoylLHx0bsWrQqvQdVwEIszA
tMkq1W1BJMTqPE1aQY5dwtr/zde2gZIIv41NikHHaOE6D+q3cNwHgUcSeRU1BOWs
Y0KjEUhkb1tGLYVBsvtYio88JaQbsNom2MRBJE8eW3gNSIEyYn2BuUeu3MGcEuhb
x5kymYoD8rnk7UE6zrDc/pZuse8sPk/LMsPitFL1I1QXRjRyc4EhINUCjPI3fXyp
8rN74Eu+1R22AtXc95TzUr44sr5Xi2JC6ZD91jxexS1TRnoSkd/ODPD00hktkn49
9vLH1HGtGFRg32LW7SCS2gKQFRf+t8DHGQBKyNt/UoOWGdx9NyUeFS6bqQz1TR1z
sw6UpnfQt4UuJR02d8Hv4OC3IVq3n5NFEGi0301Fvi7v3TQ4Vd8j7nYH9BR7IeUb
eES3imAhN20cjEOy5cwn/pHh2TuZQpoEyLAKZJrZz157Uxu84xRPSY+OyDUU/4Rw
L3M1pFSTXjG7cJeWS6qYJx6W9M/K16XffQSVv+a9tgHkCk6fddrd4Zm2DzxJJZ37
jrdVAXzWoi2oFTLUccS4P/hFje9j9rk3iJRAEPVY7178UvyemgA9OwkYG342DQ4s
+IR1S0591YjYf0XyWfEWBbkdLk4Jtnt1ObNkIXVLeaXtZ9ErByUrG4Mw2Bxq/MLZ
/BEiYdcoHUFpZqMckAqyOrng/k+uTkDs5OBnBIg84i9EAzrFL3iCW///10MVAm18
edoavzvZ/fJ3JyClx/+n+Z0o/zbiB0CD61/nT9c+65UMbe8FLZ7Jfu7G883fKfK4
g9EOnjShWVRgW1xZoTm6n6q2U0cazxQeVMswCe0r5N8+hw5WgW/9KhEB56Yy756r
GdoIUv2dt0JBBE77EtCLU3QxqfaItSpSgErm3u7pwHFW8t0FbgaoB+Cfln+c7HWk
5G22Og916iK6k7Xba8HETpcviCtUbKS+SKXobr9ehgBNjQtmUG1MkUgxP9GRKBGk
M1WnUD9ZN3yyLyEsXyNYRr8psmcS/tHXcpU1TwyKfS2wrNfXUFxUggcyqfkUrYto
nTN/bThuRWHmluji69YLvuSGZTdjn6WvPhzG0D0WTaimHrH2LhIev0t7gd8p6461
Ke9ElGsTojuv+jE4W+a//BDVsMaXONzrmPJFPhHEq+ewSreJCn/dNIy7LwzHN0tp
RdNY3oNXm3qIQ4ocjo53nEPeChi5sMxmdHTzNvVS19s3baoLcrSfnSIszX6gevM
T3exb0F2ABkqEYLjK94VepPsTVJ8o5JIXaEMTFyXU42em+gGhFD/clr2moYlm71i
zbAFGP3KLDN+nMi2QXmoR14/4VhIs1Sdhs/OdlbsQKK4WBGyRhbcYepWTY0qPFh6
0vOxXtN/FYJc4b2h+hBTsdrGdiOBYDk3pfKbS4R5z9FnYbP2LYiWjZ7sbUW572J7
i4tdRsuAdJr2dA+TEk/d04x3xJkxmQ2xIaBmxmaRZbxGKUg2Jk/ndJGUMLIh7bNi
3Cni/051ZtrgXJZyWn4CbawvDIntdK06KetGrRs8CzeUTPz7XOpOucXC7CtDB5Am
W+s+imvEUX1fGqNoI+FJtevc/pcgrSFk1NFyRQ2F8R6hra70uy02W2Ta0FffFZtgx
OGboryID8EkpBvEr0rEjxSDzdWnTpbD1RlxKmh1Tocft0N4yRfa2MLAuMhIcKY3U
sKj+SeSfdq+v5UOuEvr4RDuEsWRgFlFeDjv1VDlGkDzR5weT1dlbYXv86oI4G/9V
pE/86WG2xzyEYrHuUW9/y37EglGUTRP357gGuZvqvLWLo8+TRRWBDHfxUcdlXpKW
R9ejNA6slpC9Pq7s4cB1zcYMH/tX4o85FCLkIa6PfNSE52Dui5AXo3HliBeUGE4p
FBBAbc2yK71L6vKp9ld+a7qhzMw+gEkt9bjLRJbSldiyTvCuisK2n+zw0NZ98ftn
duoTAWi2pKRw9Tj8csKNGb6XCZmVM0rA0sdQGjRK5L1WFJAhw/tuWA6ZPSXER59R
xFlfoqPCKogCImWSokmdUQ63dwSrr4rQsvKLRLQCfpv9c68CqFEV2fsIFtcfUAMz
eYibzi+Xl/t2XDPZ9DYpEopOGcfAXvUqSzqbbcAnvaOXHRECEJGmW22kvqgbDwiY
Hglt4LkyWAG2C+5MbFfB0u6U9NVgv3EnPZceDXYWhkUu9T7QvyQso+2vaOGt64
4Qs9he5jL9cLamEkdm1vKhSpJ+uig/lsrw8JS6ZNddyCACHKDuVw1W4y/A4Aj7Vk

IUBampf6j pzmlaYtkvFUG/X/PkKZYUZsX8XSRTJ7ngTSMfh6pj9ZjPbGOI8Qnob
sqdThBen8dLsMS3SS1jg9wqmhl1tKV+0Ni0x/xLy3weoC96ujika35zZh/048HKN5
6104KOA5PiQqmwGSVskQMy8kBZPF2IEormQuZUmrz5w1xVGyULNPNhUIscXDGv1+
0ws5mOu9BHnu7OSy9RjJIp71lfagI0//220jQ+kwxpaGsSRYN0k9ArR8Lii jUoUH
cxI/VRAa+ELehkMiAzHma0quZ1bztVKd1ISono5d++7W9c68myMreM5IHKI1DMXL
PfIEvCbhlTOcgetvn/y/6nQDMOTJuzehlp9un3rIvfVfJtbtId+md3gHa2JRceua
tKifW21hk1Ec5rU8x5n3Zcnf/fupeVkkT90fR3NNtZjLKPh+tgVOWiUUztU2Mjpl
eZ3p1IWgfdLKlMw9Ct2kMXMrEaJILDbC9pWd6LKUTpmXwJSDn2sifPQkFR/C1mAi
3IUQevSy+HdGEDJmD0lcEr4dIAT/rrAAsJB4fa09oNrU5uJ/gi++qKx0o1nMMMkS
36ZJhczlp7kiZ0mqF5aVGEAwRn7cOrrViHDEY8bVNTFTiJJKDjLro4w6dbaRJP8
xKJgXblHEOCDHf3u91gcKZ6bERuMPxTXcqVTGiRQjRmPgEPUE08ktgBA0Va6QoV0
1g+ntpIzRmek8t202ITq3Pfl4XW401s8MrjDu8U9KatnPlf0eaSjGnhRtJYZO+6z
vaRgNzqimwJUCyJiuDJjqn6TvwDvZ0P4qCbNLkpBQZjyevAcLg56nQImgBn+KZPZ
1kPOX93JWxW8jI2qt3xsTdbITluXVUPCm4AOMo9/LYE/g1/PLejwMmyCX3mw/dS1
avlPSQ78JwubirIjAcPz/iEsc+6TRobJWF17ixFC0fDWW4XwTzpzVqYkcn3qrdQ0
txX+bV2+6+F/ZMf40sXUN3RxsVveT99cGMyJyhpWytCGOE5tRd2xB14N2VsO6r1R
M/ZhnTrBjwmEZLzwKXMhnE3rRhubX3JMgQ42jLEZqt fyzGh3Qz5UOEN/eNwpTTLt
h0kqu9DX1/vN3MwTYaHH17MMniZsZwUALRLBwEUpMipuTOSiDarQqmzi0NFR1U6E
4TuxVFnnQzV12PdCccF6owNBxQX4jz6foY6VVuXTYaV11F1ykkwRrwPU7R2gY2V0J
c3a75T27GZq3EZLdPz7yQyMS9iAIvJzgjXvPPcXi7zbT+eUPPEc/D/jY5SUiirj1
OPy4xxb+yDrtilHLDzvZKLkOjT06S4RLA5CdZv0HLWKMvAUf9Qyb43PaqFNRjEta
TxixKyIrFKon2nzbOiNh8W8z404/1KBAodn1I1MhGZ4b5hOWsY0KY2sCr/rqRJs0
yxdFL7o4QwtOntfEep6gMBirUEpIHxkqfY1j3nBLuA6X4WkoARKLomRn1c004LO2
ox08bzSmTNNwt1B1K45DjxQft4huCMdIa5N5hRfPUlG4G+Z08tjZRYMKuHi78Ntn
SKtyo+9XOYCaioHnUOzhSd0wXpZAVtixrshKZJ8BeOSb2HhJW23hoPud5EI6h0tU
P8JT7Vfshp6nc0nm5uWc/hGb4+G2F6Qaea19ZodxPquv0OgzW51ts8V9rTlxKfKh
bXrrAYYVQqEXLw7qEeptTrEia2PEb/ALsXboBcvxJeHE2esGYFinD/w2k1bMwqaG
KebiMZTB98PvrrwTfi+mPl0wHA3FmRm4B1IPH18yqgPIqHZPWnKZHyN7D84vn0B3
c/jGgii3mYuiliNu78cI815dFgXektZv1A58e6zUO6kTd2ShOmT8NjkqOg1AACVT
5n9nfFBF+Wldf1NldFIdxc7Y1XCthli+RjuWC53vASEbdnzMFmCuT5bh8Hh82rFo
UbQ2Y5ssuqi6F/onzAh7XezjMGfZDEb1F5S4WrGnyJ1EcikxxJ/2zV41GacEXWda
kFvC8oHxlepSFtq9B2b9/ZJSVwy/p48UyJ0/buYFoYwME/FFvFA5BU4Wo4UvVPeH
iVDV8mC5cH5t2HubjV4332LFKpqSIqA6+BLhytDhOx9I4E6Ns078N5/US1vVZ86i
6w1yMcTT6SXn4N877apC2BgDR3T/byu34Y2zHUjTW/4YQJQQqFVQq9watpFShVbx
OmLPa8AkZOmScgvEQKUfP15p7zZXoNpWMSwTiALbDYiLTGVi0bh2EZ3voRqcalQ
oSS1HtLoxpSrWtydtXlRQZUT/c+crTac+rxw2XmgfT+kqovdHPqLXhfZQTXdtYRO
ruIAiWG0TbUUsBEVOqWY7RJjGf1WTnEyCNk7Sk6PdFqWz7T7hRNYCbEEdV14fbK+
rpxBbmdpNQxY4KQOumQIPxLj/iPtXkCSu5qVEgphYrBsahu9kaCuU2x61ggIqfir
xwqzwG/lJNu0NCPOjR2/R3nAieqNy3eus+yXDAa4L1YxdgQixBod7iDt/v1CZL6E
zGoDoJpm8hWnoBvuYYDbmA8fAkfIq4utPMHrpr+bOW/7a7PESN7dBV4onEWfQFaT
D/T33gyRT51y0Uwd7Sf/BothnNXSQYWX7+jwkUMR5yCszQCxGqjuBLGE9mFAjnxZ
1PG8K/hN2jFAyfl8vAs5ak/Ui2eDi3x8UQE3mFRTxvS/irNUS1c1Sf1AgPaEGZW1
fV35q+7N15gJrNsopoZ/X64U4CzNzk+6114IjbczrzKJqF4xWzRLmMxdZGsJrhjg
ox3JAAECGdYmfBdsu1TGiJ4J3/ooGsBU/xTgi532AyXGT8Vbd8jt2kug+K7KKBTp
xw+jRrSD9gW3kcUe3e4hqTxwVNUs1t5uqkjFKpMgdQ5Uz1t1kAVKEhGCMsOHGw+e
8lii+Oc+IggActRBZFM/DMucxfr4gTlVT8adbtODEr61/nWwQBumEdDR004PgXp8

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] 10075 bytes
  (decrypts to)
  application/pkcs7-mime [smime.p7m] 6444 bytes
    (unwraps to)
    multipart/mixed 2086 bytes
      multipart/alternative 1425 bytes
        text/plain 481 bytes
        text/html 633 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
```

```
MIIdDAYJKoZIhvcNAQcDoIIc/TCCHPkCAQAxggMQMIIBhAIBADBbMFUxDTALBgNV
BAoTBELFVEYxETAPBgNVBAsTCExBTBVTIFdHMTEwLWYDVQQDEyhTYWlwBgUgTEFN
UFMgU1NBIElcnRpb24gQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIeh7j00
Boq0MA0GCSqGSIb3DQEBAQUABIIBABYIMaFU0xsD/1Txj7lo116DStu37Nert2mk
49trfnEu2mQhv6MAkHx1/MoOvM9j5S/Q1YSfRhF5c7XVgUWL17xafpFcdxqwyK5J
BfPzYzqEjA+P/oGei2qVW/IvI5iJkbFD04TPw4Cvfab6wNOnAhLi1dJElxx1uUD
93ha4H0ng3pb7MBP4wyYCSecC16mqDo1TGCP6ejUEzn9GAAMayOVK6A5DxVe711M
UtAdjXwP3Gy4IRYTFfISTD3nKp51OaKSv8g9qQtGCuYdfJxW3eB0BpG6OmBLMiEU
/jv1oVMZp0NwmuT+BSbkdecwgwuWJgqOOFn/4aIDEmyHyC72fakwggGEAgEAMGww
VTENMASGA1UEChMESUVURjERMA8GA1UECxMITEFNUFMgV0cxMTAvBgNVBAMTKFNh
bXBsZSBMQU1QUyBSU0EgQ2VydGlmaWNhdGlvb1BbdXR0b3JpdHkCEzB8R0APhiY6
HGLS64Mv1sDXhpQwDQYJKoZIhvcNAQEBBQAEggEAEELJsGfVgEhqvwys2R6g8/Srk
JTe+rInzh0ZtMqt+7FoT1/5aaw3JDLnPsncJyVpqrXrWq41JadCL5ycSU0aspAEP
RLkQGexmMk27IYrhWxfTkALRWqrH3yvGihyuSwALfMWqX4uWgyJB4TGxN/xXfvW1
34jTPewe4JQOWGajIc/dgrKKDgQcbiT8v5UYw7d2ha8YpcUxII/t+RfXqknLDfRm
lGq3zXjwfmve1ABkYtvr7NZ5J1HsAQzMKn9m2Clw69ocgTgBqCHxVHJ8k+hHdXAZ
```

LlU5kclv1KxKklqtviEXZBtDXclcc+jxEqNT7ZI1t4FlnqqYgroVTvzsSpKMqDCC
Gd4GCSqGS1b3DQEhATAdbg1ghkgBZQMEAAQIEEB6qJs1pRAtedRCD+VoEQzeAghmw
cU0VsqppEoyC+vYCYkiY3Kbxt4jFIYYdaJPJRd8vfGkNtINaoFODWIXX7QSy7RU9E
Pd7Fc7zrQ+9FwrrSbxtMQZ4s4Z3cI4COUOMPR/7nlHspkKLyJQ16bEp6Z2GOjn8h
32kVC/Zb+ibVcKXyTABW0dRCt12f5Kai4jEtWxlrcx43SLS9NUEMDTqpsgphCS7L
BHfHSmaM5g/RCX2Wa41meHlkDnQsR7T86qK/Wbna6eOdYL9uyhILFC8UKZr2PSRC
1/hFb+4vgubeJtOIPOdtHCsTxZFmudj8t7Wusq9KdzBBLNu7afQLx2f/tMPI2Zx1
ZTjDuhsoPFZIH0Lp6MNNCCzWHVVeY6K1qMI6fqDHL10cFpWNManOERp/KfP/Gu9
8kRxF4uM7siFrHjdeOa8fjARE4luXNKUio7DRezKvXpN2dzg5CLTq0/U47x2DON
TomM9kAie/7SxOmbAOmMJlO3p91Zyez6+BmXZ1V7Udhvdcf50o+0y8X7sBLEwZB1
vzQvCRUvJeSm2k51hpNGv4GEA5fKKRQVdAITaCD1b9GJpmiqJmjt9YJl1n1M8SkuL
yxMBp9RDnraYcnrbgbyT60fnC62XYHmKMTObz9qMbcW4aweo7odM0DLhMpBEiu1
308VDGznIH+gPB8177rECe+mXVIRO+sU7RvOUOurgm0Bh4Gwxbluqb6UF+yUB9a4
4ItKE1EhYRuxIkfHr1rCvAj4mlAFSXjKakcI7wX3kFRTODz0vQe2uX90n+u2N1rY
ELTPpQhrnzqVSnQEPXEJPDjC87aIw2jkcsmdoKie+lt/PnoG99sF9Rf5q1NvG8rO
jK5FzIR16WRK/u8IqGuZKD1UjxuDuwXyyQJZOUq3xHF46YE/0iGTkpcMPBNDOMXj
CSrcJiq5FOi+Tw/TUBYhjYf+TrR31+cLFGUttZl0af6mfMX2y4nhRkd5I1Sy8TMk
+RrscOia0g+gWRWfpyDzpvme5QEJxsLuXv8UC92y0Epr7/OKUeCuJGRSU8iPePy
dqQSJV1kh8z5mG+3ioZdfekTvlfnicocY+yYecdOGCZEBRdzq3JxLEMinsIk8Th
W3cULAtziM7gie06byCMBkUuUDswPHLcQJdJJwpZ1lnKGv/vevR55tzUgdit8tvA
oLnJQO/90Yna1PQbL1eUHE1Zhzh8hqve/3iSGn2M61EGi0ASRh74WM5Qrwlfr/ax
l6L3GIzH1/Vr6dLQz15nPWijgVsl+lfGkagwpK1MX0veWj7WAGm1lFJHH6amN/oI
lpDtSGwlhzakM+QBTbPIQ3iWiPzA9xmiB9qXDfSInpogFMZVKHs8d9qpTAdSbXEH
Y1F5XoKatqjyA2A2kqQnX3DZNRdGeYsOPpV+qBBtBmIzmWv5qXM1unwQuB5nFEzf
ciq8LNboFTxM6Nb+2J8b84GNJH0RwQfjyDHU081z82HD1dFCCFbeFI8H4dg6vzQ2
dRVFqX5wG1jJI5ZsAafFLQaxiyViAfEcrrNwbTauinSqCwzW8VsKLe/+RsvsjKZp
QTgcJ/3DZVaqJGefNi2i5YerLizIRGA0UUFdPck5iDqW0o1G1R4kUxnQM1ttRxwj
m0K69dDcqrz0lqCd+X1LE1VSuQ+m6W/p6nylVy0hwcNZK2Rd6V/8CztIKs5hcmVs
m2YcrPtRB4ZntMqiRHKFHqX6K/bI+YJSArfVkhJ+top8M4qW3jFvGbk/d9GA+Xq1
Oe4+5cN07qdC0OEhtC061ZEodyDjfoBE6y2LDXXVDC7vAUKh52vG5FWLmpgUhy9l
brHdPnkrIo4hJEgbeTyhP0FSQKKKGV0h/PXBJWMkfYwztltBaUPhi22dV3/MSLBZ
z7dyc5Ly9wAP72qL3Cd6Kwsz6kvBAMdcqzR5PWvdjkVv8p1RuMWkv4UFazpM6sX6
ruNNgLCAYraByH/DbYU4kDMhCnpcVstZC6InBfMveoTsfcwSh0Qeb118SeqNBoI8
NjDIDwlwXR8fsWNj5Ek7Pormutwqhtj0aqsNRuXBo+iyUE32QAb+ErX+ukbW1FPf
ECA/Um/vZyP6TCZEMLCaxes9Yx4XCCGxrKboDwwwKieiosCzBRMZ+hg0zTqiWPe
uZIGWq3in1H3SPJhtNkbWvZwEpfkK/+soAQA6cNkKBemJxdjy0Cdvs4k+iWN8hVc
YN0eo9wG70iy/xLfmC7QGnlAsUAWBkzpuBmcPw0VCNIkTwul0OR/K9/mUUE1QYpq
g5BR++U0cilgbBuh4MqdYBSmXraC/Sc8V0XF8HMXFqLf63VvymmxKXu0YdcsQmzg
pLp/eA8DY3yEJqZYramSSUU5b2d9pBRhh/uiSX/KRNquVhIbyPmBr//C2E6CFSG6
xDFJcYaZJPUIkh7SDDI8gIOshGoJpvQFfBZJtfoVtjp8gGk/pdCyqqCN4/4J5Lq1
HIfNXAqfeKobox3KJLLK4aKUcsElZ3ws7zh+0IDdtq2KtiZxFaON7VfoYtPZCDZ
Nf1XvkGeI6/iZ2Tvpce7R/+ueMUAhbOk1IRm73tC3KNBjEcTmCd5ogHjnbU//FGL
APCFds6dq182nG97yAxLRRVK/Hf6K/wCPapULZ9T2fDc6uIy4ffe0DymnguoIhxH
0U27dBn7m5FpaY1GP3+yOm6syw99RaV8o2NOpNtu+RPRD/V/V43s7f5S7BcGtdVB
BZ3Q0ppHpU7UViCCSK6FnEEVYly37vF3uP8LRfJ3ZQ5N8957zXbF10wUvBKW9eLl
NJ81I+d6Z+g4VZn0vKQkgjIp8xhtkUCjNzwdCCISNABMD1ja/N1R+aL/zUEZpM5+
TC7KFqJdea7VB8LS5UJUINa7SuWuGCUNqAZ8h+2Q0LTC081/DMQCMiUyUYMZjj3T
qq0ZXr2KX1NfcjFx3J2Z46xLpIBx2Ui6psXapHrTZoORGGD6xg5PAYQoDfvo+u7X

RMxTvYGR0xM3XX2XaxXQYzuFvY0Ksb7aa0WR2DJW5OTq7r1i2CUUYv8s6UUBNrtK
wgTWi9HvExMKS1a6cZV07S3SDRXUf+ZGk3VROgtwX10Qfx4jPVs+Opp5YMQETKXQ
qPT9zaEC6bVKlm7ODT4Hq1AA+fPbWDCmdEn3r3LRQaKUfKTHs1pb+IT0xR8N4TcY
3BsKf4AQaNLcQd9Ewso+wztvmOLHPub5PXProp/1DHap4OR3WfgnVd/7kpboYYsQH
bx2fHcuX154kCCZ5oaJf5o99GDG1M83MJP9YOS1v9yff3ikVVNzvGgSCJhqNNx1I
fJ5UW9jrSOhlMdCA4nDAZx14VcT7HA/RtvQYk6REMjhpMM/f2mKRT+LA71x9Dd/R
wS74z4b893+hIoI+FdQhnzbO9c7LhsJDyQO+e9R1EgZj4Iudic7LPaB4ibtEZfMW
I2tiXcN7bjfPAimTxDcr7pHgXy70iAzrKMkeH0VZQUxytxvCdOKqiGpa7Q9rlcOV
YOv6Qc7L0XezowibtXMLHQrh/atZqHLGD3RkMk4wPws80QHfvvtJeU6r3ORr6sR9
+z5/FM9eOQpEV556J8VvLtIRI+NkqTAQ6vn3NVmVcn0W1//JEeixkeXSNg3201S+
VtgnhKMDIrRtaEX2riy9FfYZha/P4L/NtZV5YtlzbZIZ2wK8nUvC/pjWqR7bsGqx
yVpPXgydzIFVSRdSBjp2kCRvqMVahTPBXq2FJ7D05FZjtpJ02fiiD4h7r2KG5E/p
G1Lueal+1kTw8F8ewXqg/kuX0UyMT3XuWCS59CirpPZfqWi7m5CJv0EMCJmIqQ0
wEQ4SxYhxcz61SJMMccf2LK1Rn5yUWOFE1zAW+ORZeltXIBzQy6eGZjo1x0U02a4
SiQvMf2UtMW/TukODEMGyBmfGdj+hTXsbntSh+y4LrTOEbDPMtaIkHVOQ8bPG7Ch
XZkNkLS/zfMxep8UMs9kkfQNWsjAYWPOMtLEQkn5DEHL7BIARnWPzzjSRd8+mB7T
ss+B0Sza0FRmMwasR7an0j6H8LPGU/WRJieuPBuOocrLj3uY9nUms+VWnv50eKic
dc89aR+ev6JTzre5hDYZ+uQ8KLx4XsL+8VTSfTGsVGa45fIgUOfgkJsNqLdb84WG
85Y+7qkRt7/+NaXJ2e3JNddppqA3uLCM8TcQrj3fb25AEos4r1Fb5N/e083CLTlaf
H9Wc012oF08FXM1+uPFieLIjBkRshsWngD5G72GFgaLAAKe2xBRnh8bmQPiHeDe1
dZs2+kj4LmroR1Kg8yrMTbbQpItzGhIosOX0x0uCWm6XDMrIZV4+QFmdVlQKmtPH
JHF7KbltJ67EkfhKClacZNJtSdrcFIRSnlY7D6Mxain6sHM6EBUkmyL5zc6fmpXz
8dTwmkebR8/c2mdvuZzv9cP0AVzOH5LIG3OQCkeCyRfwpX4briGu+1Nf2G2YthmY
CN/UFvwl1DQygRunTPMibMlC89pgLHsth3xraH4bqwyXQ9Kka/Oz/XLn5WIEEbFT
n8pXpcU1zuH09WjBCEoz7kZAVYtov0fAbawJFhA8vyT/DnOdv4T5ZE3KSZAtgYZB
Ua4DrBi/1b7eJ7ed31kFhKCxQIzglroeb23hMEzRLcrw+3zE8HKm4E3TQj1N8est
nuivy2KsUNTzRhQvvh1t1LMx1Kp6C6XOZar6JHwS4F7xGrxS3iVGMrIQzqbPacgv
PD9w7N9jgnJ60R920jYH0CveVCGiLO3DYjQOIJYSAqxtP0HN4nKO8gnJb+FLooFa
4fLkjoe2K1gILv6weolQUvCtjycoYdiV5ivwppRpuGyujuOIwc/ATZsKrS/NySmE
/cVfFDNFdhjffynJuG+dS8Z502SGB8zmh3tbZDj/luwlyqznHzq7hHN+QdYmUIXr
/AXXEXdlmgJ9SartyGeBTrmtluft7wyetJ7Y4Uvu5TdLiRrHVuOwzQItsCB/xrny
e9xD3J+Zza+AffaEOnZtu4FMK0+gWO6oyZ6QuIXqZSaZtGmtTHCJ6ONu2nMWgIfq
Vm1NvNtebsAS7PZg7F1GGn2OFwzdZQN7TAZtxp0iYbGrOgO/lZc+yKbALzVTQuwo
4P+1WK4FoVzgwTCUwswgJeCb0bDwYwJ5dmzQo4kxZiYxGYawoXoxvigJrkZqPOIY
dlah8s3xzQMHNrt1AXLGOS8moIcBPGXQQ13i64M43bytLOOwn4rJfZb1gWDKvcrj
a5tVN0unSfHOCgrBSJuw8C4bN1zDwnQMeawjQctkEeDU2DexIq/GtYj9X8//TPTp
boLHSFY0dcseVbHWw8O98ZCBU4Qd13JC3WLMF75aFvOcnuZZzJxh21R+esprC8ME
7mNSr36wzWd7YlXxyjQJTHaS14A9GG3kHCvawTb06nSrwRgVOVSsfUw1Pg1t/NV3
WgeaQtUj9zn4nqPLHtEO7vCRR2d5P22ism08Nulu8mQN8JCNqH+qvK2RjOxESFEc
wzo/AliWVkcROjaYivbfN08fXsN8mal3iL7L1tBeZ3dyNxRGksC7Q3jo7Kfc9H25
XeDRabFI4RmbFXHSdEcb5IZvVRspZps32VSjaFORMztIppBy7ilNt03Xoa3ZAwqe
NKdZpuSm70uwlQBvZSDQYKIL/RNbZ1c2uVko04gRvh5akoZMZHbPh62RLzWvDU5Y
EEmeT8pS+B+Z+Ecy0tCuSUFfwe4IT4o039SCWWymA+F6JMI+nnRzzbFLgoSK+FVd
/nONHA59fN2Pfe3eP4GDWVgct78eHOGLU6QitnksyUXn5VdxdJjm4dPZeWEdVyhS
xUj/RKd20pSQj9L/+i7s9HSFCp0u9fe3mluqOdKLyM7tvpQZBFRpICDo9U+hKhZE
RR5Bzw1viLobNtWbatUxLC2xwCfILdsXPzww5mWL5JxsZQrANYtZb9/Otc8QSV5t
11/AnOLYu8dlY42NUbw+Vo3cEUlqkq4ULCMDqQVEwsYaTiOJIFXXfa35Jhzq32mZ
uBRQIUaacc2nNvp9sWGbaRVV/g84g67uqK3ZTrOGmcPrBoinoe9nMClgpgCq5ke0f

Dqi09ofQK7HsQtimRa3oPqa4+auijzi8aeE0fYjUUOenF/YQgDOx0L3ObDd5UiUW
5XqbObxCLr7ItG34aHjRsiGAml/jVSNcAGIjybVuB2r/XR95g24THvE+WIM02040
9v+GuSK8gkATcCnLeHEeolOvHBKYhJy0WC0TkJ16YTwXIC6NisObPeBoYa4sF02v
alvzVOx82uzKR+N9nIHTjZXNJ5QohQ1bduPYQcUU3tAOz33pk3tTCcs6hRYfUee1
x9IsI5AGh4jUoU8CXETUKKj1SDEP8yU9KX5M08+7Opom4VncYgGrGtRRsStdNb08
m+qa7Im2zggMucz1A/PSuCWlGrfuSUhGFDmylGXVHTrpvzx6DG7trSvmeO4WOLnK
rFezgGiJZTagiQLomXiQg4MtqRAfNcOdkW/+ojy1jdpckuyou+4SMjarHJkCOPWH
ToE428nTBq3ub4UaE3vMMoZ1JZAru8nC1EE5qq/bIHdSV0jTXLw5e1vSOUaBfm/8
nSeQyBYHJtQcqp0qIPbSMMa+IavQPa+DjzNX+VzRay0XaffjfspwwWwGg+cgNKL6D
HKtsqWJNuahAlmYLe4ktql9WHicJtQRPqrAKcwI9WGsaA5ckOvP91V0nIhIjLzup
3aHfD8Fa7oKLCpksD2jFNldJL8i4utOs7+GyLraPmQZMfAULwevozQadYi/kV7Q3
hI/WxFP+2bS+AJgerPrpixJOE5IQRdz3+d1RUP5pG51G6UL2VZQXcOhcta6yjuad
nr1C3mEY0LEreGf0QMGSnkDc+xFD9vn7pQ7mNazjY8UPyoC8LdAfQXpZz0LpCpWM
kBMj1VoMooH6FFu+1KQ6MGVB5yc1005mCvwtlqqVW2j337AsASvbulH2VK5PU7TR
oEX94PULdZNGmEyQGbJGep4br+z4GOKKw1PhcCTKzS4QXCKPSLNluolt90QDny81
We6WpVBIztUG9YU5JBsa0EYHenmV4VGTEx+GrXA624jI5ZPcYvHery3AAXb61SZ8
HbjZoDyMpWCLiKb1SMpjYUrRISH0Qc4TJzYCchYp9DXp0thekCvj+JsYJuDzRJ14
nRQKmFVLTKhk3tGDPsBEk15eE0gB0uni8oDkkgDAVd4YcnnoIPQErL9Urq6zUYOb
br5UNf20HmUUVfj6EN14dF1moBHwfKielyXaffJ91OkdLfJASznAT6iWV+EMrTAY
61tDu3ZmHdrokfuUCBUCb2m+Ruxiy8euVtvtYoy9Hz6QmkfDjzU/IUpszVbxkzI4
KMopbWacNq1+bwOq7Cm5KlsQ5hXWbKJcJAUFwp1f0T6KuzZQHxpuscVOHihk/MNP
lRVqu9hYnYH4Pguyq+IwxJx/lr4BWlu0U5ad4tNpjNvHYNaH88rYxSMXKZmYB1oV
WesNteubU9yZK6sVCv19xnUCmy/meLS3ZgPuI+AEvVgV39aWDrNTWG8ZE8pom5N3
eHxqtdJgocgeFzzhAXeyH0k/c5pul1f6iFveSulVPWRWPunAshICkpBlFIWvVhXS5
54IwqzIVGmGV//xcYZr17439S3H6+nCVGUdWJ39/j86LCzJlutdhVRcNNBKAMymR
hgUeBEFPb9cj41p6uSp9vQ3zKtwyRMAEPJjzTeEeOz4YroZi0nHnpQbU5aQ/6+Ex0
AWXMC17zMPJ1aiqP0gFFjXUDUaC/OE84vok2Fr/1+V1BozORMDUNiv4UCmyZE0p5
VeZ2SVI2dgS+2EeHM5L0lWTlXQOnj0CMU2w3W7mEGwQVb6su5R5Dze5o2+JhyWSJ
gcXdY+dgoi5nje2gL6rSx8Ng9uoDKxkWzbqn2cwjNd7fMbGfDapuhKAsK1c35h6p
n48Mlmlw2hIPSRp9/af/nJmLg6BowhIFJNh6DhdaArLJ4PziwBNDw+3yhyz14IXA
CfSEin4hIHtri0cONIU8wRT8Zyzm23UzcOJ4hpmV0JQnDYqA/S3s54zU46ctH4p+
I04XQoR9nfN248dxmCUxovOCx8oKodRMg7OR0EUkQ/NhY5bu3gaTbRD3R8JiiQg
7sRBFrQAYPojJQ7bg5NsgPj0jfhEdkW/ALVfSVb7yP2tSF9oVAxyUgM1fRSRg5B
AlpYCKze3jaSj05QZuxtohtwH9d4qppdyTMUPuGV7R9GoLydLHT194HeGJ4BwCktn
Z8RAeSwMpqi8wkeu+rw015OPYE6mndiIVQUKRuR5bWFSjm2CWxwQ4m7QvjIVjbd
8lGfKGPnoyWNC1DVCEec5jHk4V72X+U4mdG3Gm4vs3NzGi7aRpeGFXUWWuIBzu9B
sT+3qcG1z9s7WQ6eiPEaERS9UMVN+FXUrdrI0xyIw8GxFcCgmLIo3OLJWadiOq/s
+G+R6Q5AE11t84sz1mIrjyZsURpic43zojbzFbcP9mXdkZRwaOH11IGZm5JVouB
EkC67WMDgWg8fJ8+1C/X5cv2XnIHZQ0okcvFWmOWHhUkH997h13vLWMROW3lXldi
UuN/+maQS2grBs30QPJzB8c1cF7hBELFfdIK+GyJk4+Rf5Mlsqo0mMDJRbeA8F1+
v2VzU0k+X1aRky/89JLRHwKAfJTlmarsf4qIvGOQ0WKpJT//Olz95ONcjFHq2ule
OgxwxXeIvNmPASj18rx1jwj1FrbMCOAZfNi9j+3ygRK+Kk+g+5QYu8zkCbqoVD2
MycPrv/fsRjrzojVnBDFRWMX1YIsO/sxYxTAZS67kz9YQDj7J5ulsHNLuc8bn7Rm

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] 9775 bytes
  (decrypts to)
  application/pkcs7-mime [smime.p7m] 6222 bytes
    (unwraps to)
    message/rfc822 1978 bytes
      multipart/mixed 1914 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
In-Reply-To:
  <smime-enc-signed-complex-wrapped-minimal@lhp.example>
References:
  <smime-enc-signed-complex-wrapped-minimal@lhp.example>
```

```
MIICLAYJKoZIhvcNAQcDoIIcHTCCHBkCAQAxxggMQMIIBhAIBADBbsMFUxDALBgNV
BAoTBELFVEYxETAPBgNVBAsTCExBTBTIFdHMTEwLWYDVQQDEyhTYWlwGUGTEFN
UFMgU1NBIENlcnRpZmljYXRpb24gQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIeh7j00
Boq0MA0GCSqGSIB3DQEBAQUABIIBAIJ1TSnodbIX+xsUfgRTABHJ9Cp7TJAEjB4Q
8bJ2SJQsuXjbky2uXOISzL5ryCv37l6n7W+MLKlTPvXIprN5kkk9mlA1ZkCprRC
usJvS25o/h3x6yb+XnhWORI3hB+b87zo1ysoA7YcyF3Qq9YCe8bkrNrstnxe6uzW
T+1EhIhPRzZRpaJzXKer4JjxKKJYn3o+pLdsD9/TlsAJu8ueGodVcn3cnDH5oW8j
9BnAVIS7Bosh05moOD1jwg1taKZu02yCsVzIq7U1yQ/kXQbxMkdc3sCIJHSH7upn
3/filDlwvHZynaQc5oIrGaXfja7+B1mCJJ3pvCwRg1BTs+2OkhgwgGGEAgEAMGww
VTENMASGa1UEChMESUVURjERMA8Ga1UECxMITEFNuFMgV0cxMTAvBgNVBAMTKFNh
```

bXBsZSBMQUlQUyBSU0EgQ2VydGlmaWNhdGlvb1BbdXR0b3JpdHkCEzB8R0APhiY6
HGLS64MvlsDXhpQwDQYJKoZlIhvcNAQEBBQAEggEAAJJ0GA1RKCaiPrJ/hStHEF8Kh
MV66qx8SQ4zF77Q0N1rgxGmQagbwuJaWy50TKpbEet1lel1l4pERnA7ySapuOE+e
myif7rCxUr08+MyqSidsGm4wSUC9MwRfUC+t9CwwV55MG8ajowtd5WhCKPbStsup
9MXn5yMTNCMJPMWOTVx8b3dcQIREcjMK06ZF8s0Tv0ecI+FaCP/38Tt5IxZ7rx1r
3IlzBA5i1uHjKGP1sF6iYoLaFm36gWtCk36g++FRtROfmBa+PbRjX0HNU8efC8c
W+WbS5fHnf6jx6wgtRgfNnwz/IKBp3OYBhpbURNMRoixDwTk8jTg6nWnFJPxfzCC
GP4GCSqGS1b3DQEHATAAdBg1ghkgBZQMEAAIEEC03a1ouHG0V1jk5134aXauAghjQ
BZBWseHe0tDuv9ziM/TuHvOjqmGw7teVs9nTLzfGBG3tZSAK1VuSbG4eJazVrwQO
00G7DWRAtcKbTZ1CiOXGLEUm4wddP9TkZTLZR0jrg6y4zxr45Q23+ie6Wtw74p2Q
ZcRm90Zcv5Vq6rBThZdK946hmVdfNK6jB2ZQIfZ7ziemSrgxLO2cEx5dLNI0K3qc
61ZdmJc7phl0+5sH/vMpzzQu93ju7f28dGa0A/fgSkfAGE5PL6kYXLJJK1l1v16ph
mhi8tHK4xGR3ELSu2LHapl+BMrIOL2RF2LJF5qgejVyaRx4NoFvd5ghSATDuzi0I
h22eFXT0Fv7AwCgBjucQxgUtmCLyd4hJLzNwGQK0mJS/YyTccEtQ96HqP7+3aBnt
LFSP91kzFg7n7mnyffqspUK4jxCj1AXohRL0t6EzroIeuNXNF806Q1RZutbv8GIF
dkhSYvi/MC2AZmulbW3UgPz60Zm04QMVzKGZMNOzNn1ezPShXyTao8iSExrvU1F2
9VVvOVYXE4+e62V6MYdRgfu8bpKyLvOYl87718S6Hyua61S1/c06mUN2bQrtQK3m
ctf44uHVhQ1gPuyrGCUrGzcVcQrvRiTGGJsDSA+kweFN5hWDY76MsTytKnMfqtsp
59vV1OZJ5mQfCFyIhBttXHEaxUfaJH54aC3RT1Yt4yRS7qBikR9C97dwHbnsR1UX
cgAsxBsq/lgiajK8cBy8ZL6yXVra4A358r4R0TPeh0r2BDboYQKvr2ke3YZJfVoV
9DxNoqf4Lma2NyQCwXvkP8D+na5df4RfmOPSnAWL1lxPkQER84sHo6im28GNcWuJ
20ThMF6zKzRxM5bjfdgqnUMJdSXKsb7Akz0dRcF1G/uYCi7mVPn3SCAcMUsfu4NI
L1lLnwB6/EZf5mtVDe7O4iwmTss/75oYmy9jq62A1F07fsH/bN3CVERJufnKiTJI
x4DzG1Ndb0QU8vSCCR/RjLMuAPO8y4BEF3wxVkJyhgvHlg4HncmkdWFRT866XRw1
5BJNrpFOgUXGIwEPRCiA0v2dgCEfpcNkYH7kv1IVTse7OPqul0hjl0LMVyrCwmfG
9Ukg1fU8dsP7gegOPDcMt8UWYxynvqoPWYStSiuzTB9cOfmb4h5AUpBB1cnRge7L
fzaddRtVqyl415a+BFS4YPkC5/+h09TVpVVGmZ2y4jNXvYU5YK9Ju5e9jy3Vgks6n
kz0kykoimM9DBsrpMdTHnUQZZWLvUSJTJjuxQFhZiY8hsP6tMQcYr60RX7jK4nuy
7wwBZ/30HalxaTDtCXWZbme50tqCJkEmnZLOTUO7p0KVK8TWgV71rVZsht0xO+qJ
CtkOUNN2DybgIoBylKvfbt6s3ithd3sQdB5oSigA3MAxvCQtM/whtwyooXLWmT
5vuiPTiIRKzmFF4Hzt98/+tOeXtW64obDHACANfKLAG7dOy2sAOAOj4zG8ykW+Hp
y9QpwYz5nI6rlps5LiecUSvujsnZRP1LpakSkpjYVFxAckoiw0Y/aD+zrXXV2K
qQulnRrvyrCIYT78XX5Z9v00mmlLHAhmPAwvVjUKLC+A8dg76+JC3NFVAJQs4Fng
TePrqCLB/qmvgK8EN+15eDZ2IiApNsFmiDJLcQoNaDcMFOGYy4KTNU6C0BsZ508j
K4S6bz7JjsXDPtNQWrCQVbmZDpHeksenvytw7pF5ITbfWQ1Jaz/BHYGoOrlC1sYo
CTI3Hzf1geYf1Nu8EpKLks/Aa2P/mu3zvIJ9Mdl410X836tNkYyQjumpyIv9fFm6I
Y3M9x7SGsvNXjg4mihyXgA/cULLcT8crrL4qxxDz2VPEghQLv6FzycmOurgYzwrL
H91G4JpJV9vdevlRTrM7oUdIHP/RdKZ3IC7RAHySPAcJZnYPbB/p9WmZfp84Co8G
cSDH8TvaEiLmesuVnqGQIChqQhccjzHJL507GBXTIgyWUMciFDrtQ1fjS/gjUOWW
EuWN/vJnH6n5eKiAySg/J8DuAyFt1Ij2aVBJgwZp07mK9jVgEQU4kolfUyHm1WRb
ontUaYxmWKARMAmR9xUTUxOaXQ7G0XwPN/vu6n/r866Kk9bEqfArzIcxIq2IkRb
A9NbKkCG5gPLBpnzk0tAxDKABfxBTPyIQZkFtZdUmuRKZY0RRwjeP9IUI+gbbKtP
eVkv817JvVW6oSufkPdDr1l++vH9fW4epVh7ToI2SqcSNsj7vBRVbB4KiYB/8fx9
ZHcsCRZqxVcNpvia9WTfNZYdJCJCKwNKJg3EJ9hD+Gp0wqiRtZsikeMRMJ+M0B0g
zK27TfSCmCs+cEdNhbeFrTiObXEwabpGw2Jul1+djpA7xbtY34jFy/ZblISTYD5n
6ejtrVMWGMiRy7f84ib0U0RdPU06TkhLPjv0e7TWuP6jFEeZiZQHeNBvdtC9hE7
iwfrqXQ4+AjEy89FERTszHE7eA8qisp3wMGot6dfJ8ColUGMzgU2B7J7C5TFLcKZ
pJC+x/VjGpk2+kZ4lwP3GB3KN/ROGRsdoKqt5V2wmNXEOExs7WGCSLoC0WKZJk4g

1OP2udKMTcVvAUNqS6tbZK7amGEKvuEdPWqWEMQCa4DtURbrX/+r/003Kf/0MLrVv
LaoyH/qiTl015FLYR5hHIX71DY0umkqtOdUxf3CRBeHD7OmH/wyhCDts2Krp5h7t
HTEP6WTB10VglfAkmgVzTcgdp/Id10bJZocu020qkZQeJBfbDcXM/E2FR4wZ2jqy
mkbdRho5pGC2LFc0cH68jDQ++2QqtPoVYhV/k/eYU9DL+QXRx2VvCrliHQpoeExEw
o2ucVlbAo6TxPKvv01A12AJkzKMpoEfyKWY0jHulNB08dcdFcLNG5glGaIqufAck
nkoR7r32bweJ1r9hAgEX6cyoMxBW+318SeLr4Kkmk0wigq5bDTrXh7ahqZzmsxv3
091DgmSzILxN1x7FK6K/yISo5FY2x8WUZqZNz++1tk0aApJU6ZFtaw2/Aj+GwK8g
lS+OgI9obXun2mNyqEYZWaZ2Lo8zVhZ6rkZAsQwVL4Wz9OuL1Ko68fp5Hv8zFmul
kS5bh53wG6qEigAUg4P0E/vCy31KV2Nz96sN9/B2awyBQ0uVkhW0oAq+zOGU19Tq
uu5ilps9R5iwsbF7oFsZDCg6Mp+I+kTAVvEX9Kt0/d3HqtXmZHSBZlbr3it5evm2
dqxeY0djL/WEXfZrNG/CH1AV493f0NpdgkYluiU7gX5gf9jadWjOdmC98XDtPJkt
eFG6wXKv603FJuATpAKhLZDkFrtX34cdQmtxj76UwB+rCQqO+W8Ax0v0HmeEayeS
HDSKJy0SbgWm3np98sH5N3pdpYQ21WD7p+0M328r7LLdLj65vP/vup58rfG6dSM9
CCpNNHLf/qbmUhd+q00PoNmXPWvggluCltafTWrbL2Ibu57yEzfePsoeoJagVpCe
X0501Jphn154RWeYHy6Y02Lb/aRDtu9m4IxtmcOHopKCgXZxiIhkmTPQlCeAmrmE
EebmFvH88R3WcmU+QxTMmptpdUnXBJX3+8dSxojeTo646hFV5r0JkQSQLeSRb3R9
OMm95sO+v7c1aArJZzT4xDnBppqlZshaa7ZuMl+pgmx/UEVUyhyWvySJdQJMW85
crFlSVGO2u6WT8LAUTuinMj+WolVniO3mdx+MSlg9MgnuvTm9vRLPaJ3e19g8jWt
pAnJz1N6EYfDSglB9+Nq9aGfagMc/6vRtMWT9AyG/DWW6CCSddK65FedsWzvpPej
pVfjJP56fu9dH3jqWrlsVaQLyEpuc3ArCVzk3FaGStcjsQadQHkgYNAYaGmSK5PD
N/cbht8G/GfBGpGAg8K2wZlk/VBn/uri2mTgtBVu9JJ/joJHxSmRNjYrAFMSP3F
Q8Z/iLrLzXxYuDn9KYFlv+OUfDDmriaHO4CIdQlG3MUq9+OMdx+IRB1ZiXeI0m3U
StFOVYb1nN6b5zlgW/ZwU1Wy9Vl/1AyHcuT/+m0TazayURqhsSjuEkC/zpLwXCMv
72phlTGLP6PrwqpUYF+ZSADFcrno+Ct2ylyEKRoWcqcT4++J9fQLNZKKGKQTDsmI4
K1zKx9G2T8xPF1mU8ATlpkWKSP4TT7yROEWp1a+aS3VOZGDvIOHuPmEKH0ju2uWg
O9OEGFOe1JeElOnXp9nLFPDyJzVRzbPgZBANsmNTIGtWokTNZC6ACKv7wh9HHZn8
pF0iugZ+08N709qWj23ps0dQk7GSIGYLUII5WC7DLD6Sutruum9ddsZ5BVDNfg3v
0Hf2gz0M8cGLKKR6wUW4qvK66METjvJoKLoZwQyJIYZKLVR+B9ZQtUBhml1J/ju
8VTPig9loc+X5tt8T/FKa6kvd8/ELN7UQ2gLoSdC0pX4vTQrU3pq5gs+08NXieFy
Pmt51AYkPPdqytfTrrsqAQHbeemxIZ4R8ZPHoM6ObRC0ciG08QVpSuR6vdOLM48P
lI9AVIQk0U56KJ2NUIhZfBg42hk0pytEBwchIfbEU43fkVoEmfzucImO6DIU8WuA
nL1NxrT2dLrFCQ1kSlsqvHMc5NuhU9BhUSC803rjGPcNA2U6DMYr0omT/A6dgMMX
vK0l+f5ap76yzzQWNJmiVln9iOWj8W4ULXtOH9XmzagAD/2SNjbZarEWLDhN69RO
LNaIW7QIPYHWCccvQMtEauxdmfJFDx CZSu4EY2TC4x2YdOOWNb/gjH9UIxz8down
JDgdc08F/eDg2hzpL1SlkOuA5s00AWZR3UNJgGRikZvqvadhV1Qs0o/g5Q7eU9P
P9AO/U9HVEHJMSHO07NytGUVMilwn7V8yhPwoScLmusnPHVqAfUxhrXWY7jHYN8N
Puhk5IYR9hpxQnab87i8A2HV6d2ezgFAk6CDMFwVFqfHN4v8TzHRfBIXG1lXy8uy
NmzQIi/4AYD7ZkgXB1p9mThbZoKZSErCO0opPPwjVGwjGBew2yoIPWwxRlrlHhka
Y0QuWrLkiM+WYCKHwPuvW5mtYraBX4S++TmWSPCzpMjH7/TJUeYGGC+4hOaYPMCZ
W/bd4htjpnxA3gQk3cUL4ZJvKRGdV26vS9JE8v/A5xXv5rALMSO15+XQDPYxBMu8
ZQVAi qeu7kGyTd7ZqT2qwVmmyT+8R6fiRLouuwpl8q2EKpQL+Qec03o4Tl+aX3/w
5kyC1leXg3e9/TgcDxlwEvGKF3BJWbethR4HiX/J2/mTWk0qcq2GuqKEQ6USqA2u
65isAb+WbfwRcdPmkDRhthOg4H7IwZdLiviBrEzxImQ0Q+XZrV8CAVxKJvg+hiD+
wc9YfgK238F5vwIs1Hc0fdGFiuAJq51N+34k1lpx0uLS1x8dgHO5e9/dA+PXwvfp
epqFhgNRdu/3NdBseYlohfGYA1db5R9BGnLOA51T36zX9tuo+5jtrAxIIV2QFOCQ
mapV5wdB0kL7R9ha7sIv33e13nN03VAmKQoKITpynZ/giTCYdPpw3vvinwqkEbp9
9rr6gafwig5d8uQ3FBv3vffJYaOo7edP5HR0Qoae1VxbKV3uX4gLOMjxt57HDP0i
KV50cSAQARRvrdHAiPh64z4/hSofrn6rwhWIqu9iiUhdgTPYZkb3pkknG81jo00N

1bL/42EH+6CW+JTYjjWx+vHUi/uXMYBbSbR6pT5rxVnHU+SnhZrka8JZ22gKSnl
lnWrB0RfW1dXnEfQCKTPhuZ3jbaLiFoxhJzRL/BkDvJw56NrOHGqzchNF2MvOGYq
iXPx60a//5p5qe8+9ZJ0MwWPLbyXzQbwJp43r8027H1URNbBr+VY82F1pA/eIzwh
M/al7XH4rCdo5n/mdjo/owmTOHEBvls1r0g7Lk7sJHHm/XWk6rquNPF+fzKtPyTA
FMGMkMoHRAusqq4PFgzGkYNwly/105bLSnvkSE4R/fUW5tHtJsEsMNLjXQuHAHQ
QuRtL327OMulL/GCguKpCZ3OIZYfrPk6DSkS8c6SujU3HOGKeo5w5F6QTnYamgvt
T90AgoRGfDZ1e0bBv9LOeWFQsv0sOYSpuo94p7PRHeFDL/MiU4KpnJBabjldFrch
3ztE8PbhbcKAhwPQ4pfciOPLaqWAZzfQUIKDqMtTYoWErWDcgZpQn6VyXIK35MD/
j1qRb3FvM1U9yGqrHBuNMIPkSi17lvglGd01yS+rSjvDo7yxRkr+obhNXghrox1W
li3kZwRaj7n5TguEtnlFn24rdoHu025fVmrynWZGnb1QM1lmPqk4CPMeMC4GuvBF
3mnw0jYYo0S4x3RpjR5Ack44X1PrRzo9kd2d8UuPYNokIrhSyFUnzUj3T/U6f6Ud
VwEAS8QqdkStXyMnfGidkaF/O7PqdNxLYwqcOgVd4bln646z0+f6IhoqVNMJ3Nux
ftycLJHKLfS29P8JM6up10gAJMit2MJA8U1MCKIuPTsTzKNdoiQJnPsF3JhsMk
qnqSD0ZTGcgjJLhL9x/E0kkcvHXMwdmteY+jfmNXsvUex4AneP4I2Qo7FieEFHYs+NO
00VyiQQu83P5WoGgqP+UVbgdPMS6lkNTavqQ1+xoZupgtUERmZW0ntGs+dzxBlpy
jFPP5xP9PGcOkJ6vh8DDw8hqWE28hDPnf09Nz1YT2G8OkQOm6hbfzGVgig7aWWhP
0wKAXmLPrG8kKBKzL94kqEuMP/V9T721ASLv6gslpHJic7h7/vAqNyBVmZFCRLuh
C/KyDASVZoUovc3phQUOA9+5tptQ4rrPVtBJvq9vyIppu04ny/GL0q/QEIim+XSA
YvRd+owkDCE/Vz78bt+oNbHjdEJDvNSE5yJykCiw99pBlxTrlRgs3hMzU+LCjHYE
yZUGd7ufdF/EK27ofJWnJEOMQ21uNcIqTEeDEU6PIK01fSV6GIk8tGx1HjhoSE90
OGybZPh2W+Tm8xvOG/VRihnUxHgJop9naLiE6Rdx7Gaqi3hzX2PR/eMOHJ/ctpIh
3sxUpQWpQTPmxTGTjtguuJiRnwAMFOVHHx5xuNrJAehpW/5blrDEwiisB3LjKEqN
8zmT0JWJLcURaQ4dXT1z/JDfjNxrWv3T6cdLbntfTCgeb2CCXYM/BE4F7ZrKnHX
ERVLUEuHASnFQhdCt95vtGKAODdCLrCyB7wt4Q40Mq/2/R+MFF0JKYot7phsNJC5
RT2X+041jR2FiCnF34I5cmfkX1TuzcUCc1JmzKMwXbaTTLSBoo9vEUedA2+sBU0
/hMdr70zJmy1eZyi0j7V5cwutEjxUHBCXYBRrm1gzUD2/6uNF5SeMwqIB1W6epu
fOcK1fHSij3NzdLaCeKnWMMgzJTfqq/TeMvrsq1755bfj7XrPl70r/Fbl7I0//sP
TttmqPr6kGSfWk6RxWulwVpTJrfYKlmcuKfNite0PAsyYww5NZ2wfHm3ahnfPzvT
sUU4s6FYWCL/GxrBrjyJ4rseMZ4W4uhFhXOd5+HefFM3IROX9JteuO+FGsHN18ut
85HOSiEP3ZpOGvmsge6tDtbUH0/VtVS3rxadPPLQcF1M6Y/7Qg7lzH7wDPc5Ra+S
fHpw+vGoqRdS+ffYsn3zjjnlIrqZzKZU0HhDl7hUbgYcnX8KtpCqTkDDIeGzYrf
nFaie4ASWf1jorX8DqWnZ3SzwCp5yxpPWC8bn3kmf5F/yWP2Ioau6aNAYXI2HOG/
q5zz1A4V3NzPdvmGxgclq1JAEu3k+DXnVx9JXncAVn/QEfaXhcOnsPV4Jwp0KJ8t
rI3AbNhuYQ2wGgPiphnrrA2W9dU3hZ0Nmc7cSNoegFb3Fqd+917t9hcGBq2AJkxW
FeKuJ8XvMhCLs8sWx48lHp73ZnrSKGPD1NBQC96iUjLnWJ6ZfWUJiErThwnRqfa/
4+AhmYuP0ibIddFckfHI7p1lN1VUw5Gktb+86Si0QSQncuIdNP43VCvgGCFcwoF
NuPHftfkHvOe+GV0RbZrOgOmByYxVcGvPjFD/mGil5nhSdr1PW0FZ5UovRW6d99
P53zqoDgzKOAcIs3ykKkVtmWY1cnJtQanH9yE94cOHc4VJBO9kZK3SCRGw7OZPsp
HeAh+cHqRKckLZASb5MMVZAhSp7AI7bimxJxDLsHWKgUqY8468ytrzeeKUCAEd/I
ivpZOmNn6P2jxtk/EBKa/fRyft/virU8ZWUp50TgGYSrD7MBOW2kW0sQODnjpxON
FkelUOPPvaJ5cEeZuqRsg+vDOni2f0RBWdEgoCnn2MUN2bI3d7W15SqTYEZADOzz
/YED5L68ReWwAO/8jJOiJ2ZKOYSSE2EatJzCA2nwMG528CtBNXpQILZjohg01170
S80RHRpRB0VuPNQyXeSSl++1bPfbDe9GgYrExdCDaS1F44PaLyID8pchIdQVat64
ticmexkGwt2so1ihPDfr4FTH0ZC5NYKB+1WOk22WbZ9VroGp8KHhwOQXjLiOw4QV
QSa8PCulKbOEcx+uesAJjQ==

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] 9815 bytes
  (decrypts to)
  application/pkcs7-mime [smime.p7m] 6250 bytes
    (unwraps to)
    multipart/mixed 1946 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
In-Reply-To:
  <smime-enc-signed-complex-injected-minimal@lhp.example>
References:
  <smime-enc-signed-complex-injected-minimal@lhp.example>
```

```
MIICtAYJKoZIhvcNAQcDoIIcPTCCHdKCAQAxxggMQMIIBhAIBADBsmFUxDtALBgNV
BAoTBELFVEYxETAPBgNVBAsTCExBTVBTIFdHMTEwLWYDVQQDEyhTYW1wbGUgTEFN
UFMgU1NBIEENlcnRpZmljYXRpb24gQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIeh7j00
Boq0MA0GCSqGSIB3DQEBAQUABIIBAAABVM3TokL/MB1a9//O9TymFiSM57sFKquhq
3EvwWIqXByQQ8Gfvfi6RyXmhXU4FV6FngXgNxgrBofyJLSvSvfsyDiAlREdaGgok
6sDANNU783lpxijGUNWEw5v1E7ILrr/WH+bFuW/WM33gB7EVaOvdZ+O3mRLr1Mw+
P13Q6oXTucozDJGmjVo4f2gmxnLbx6xRXeAun131NU1V8Cx1o39a+nXLd3D9Yz58
i7WUxFIEcm/2VDlKr/MzzN9T6C7Dlpx30498umpq4hXvHx3l4vhJ+O13AuFqpnJe
a05OWrBWkPX3UrMiuIttiHkTyJkH0ry6gN4/HlOVhMPgcspvC2UwggGEAgEAMGww
VTENMASGa1UEChMESUVURjERMA8Ga1UECxMITEFNUFMgV0cxMTAvBgNVBAMTKFNh
bXBsZSBMQUl1QyBSU0EgQ2VydGlmaWNoZGlvdjBBdXRob3JpdHkCEzB8R0APhiY6
```

HGLS64MvlsDXhpQwDQYJKoZiHvcNAQEbbQAEggEAI00unyTORapnAw86gBmKRZ+M
//uPCpz3/XYH1yyokmrZ7VGct8RdGjsn9V7qQyTiSKboObYqYSTK+Sa5Dm5gg7GK
B0Z4ulUg6da2v6ouyOliW0FWUf/cap/Q5nRxNlcKizfqr542GyCMYxHZVjwY+i4V
yTfZoXezLkki6LHB1Zh5jIw4wOzn6jAeAlN/geZ+u9ZDWNb1RhFyjsj7298pi+ig
o+E4SmVY3W+yE01F7t9540Vfdq1JCjN3mljbcZ1XqEdsMCSWf108NWRT/H9NH1r6
DPFtHvXbUN2XTXma8Zq3EBsoJE2iPsVH90j/LCTk8RRwcr5xp7oeUs+8ms7npTCC
GR4GCSqGS1b3DQEhATAdbglghkgBZQMEAAQIEELhRkgO+qcMaD7mw8JnmTf2Aghjw
gIY1Z/Tmg0UvmhbF2WILG3DqXoUB8tUJF0hUF6f8jq9/D0o8dbAAcK4UTUcFyHC5
QMH/Em+i8wbbGtdnywcdvENrv8/0v13AuBh138bugj2sUhI/vkcZiKB0FkiBQN6E9
poX4a2SaH7CmfM1CuYdYYdi1wynVXHbPAX7OzCMVXEKvFaOxSO/Btc0qJhpadowP
kwLWygXmVmpg47xqCM1G67pRa2bTGs35vFQq4iItBv7E1NcSgxIrnKJjxUGxUxwz
sjd5ZpDmYfumgQKvud/xDrffJIYJ7wrOH5aAC94T7evWcqQ59Iz0e91lr1wtnAFT
625WYAWiPdcaIU4Uafi2MfyErXA4+QUtWOBxrs48S8Git3i0OXvhIdFTBu5IUC/
tsV8TtH25IFyqdJX0hSgU6ICQ5Atmbr/oEcU/7dT1KUvQCm7d8f/4m7diix2vGuo
q773cAvSpUq1fKcVrzzjVkhP3eY/AvU3/gz6238S7MSMaYEWpeTZSm8dof0kamDr
DQcS4S7LHYN1IXCusWoqESTecMTbLeyVQeq/uLqUADmBZaRWTEmnGi9ojRynvzEa
+9V/t0NSVGf8yav6jcb/1HO3p5T3K1otRWSQktfuEUuQbZw5hLiu9/1VQ3H1A0k+
4ZH11rd6shUbihUktcsNlnYjNieceSHzoJlfnTVtjSakVmvjgJNSCov19OW5+YzZXf
whYgJ4/QpsbSXB5STxFJo0F/JDKjr78FT1ZDw1piFUVkW6WAXJftC3NmqwoNSSKU
sVHk501VRxoAdB0y4Kg307XrQsbQIQzocJnL2aYNeiWd1ej7NcRaZmB67ibEibZS
7gWvsoJyNZe0mGltTj1hw1Z263GeaqJXkmU+odvMJwg/zPIFDxATORw2B14s3sC
YRs+c8rv2CdGx2sKUfEcOztT9ZO0o84GirylS2LxuUsAlctypzNPFdrLGP800xt3
HLZ57Jamzx7I0AZwxqyhleh24jyG1PsAqBPM6a+dK2HsZe2HqgAfQHS6lgPx3gAK
c4ZacXIffNntFRUWVLbSQni8xT6kherHh/x4Jip/bSDWPuANpRx6QkOJCKHPjuoW
ilKveilu4X0Hm/G80Nwh+eQV8RDpnhRbuqe7jLY1hVti/F0OZPLbyEdk8hIn1sv
FMeKqumCMk4dlr8jP8NeMohYT20Qbv03aU0w6x1MZgnDxnesvupxgPETygyNd5+j
xKOUMcRaYmMU/TqGhkxYTN3LwSLZpeqztz9wxI4l/+fobTOFNBgcAhguLD8+NO83
0ykPnfiK8yoRQKLeH4KdFpzSkzmZVdfZH1DJT/LITxUnEWABjtp3ozx74N6YJ394
vme45v3uUmCJjcSzmQXPtg2Xdkh+xf62Yb5n4NjUZ8ajxQqaYOE80RN+A7J4TWE
c+NSglnJYdUZnmnYlJxHNKI8iAw1LEir5KHi99TclDMCOKfoVzvb4drqC8QhEjPF
ikP6skclyTthVmWx4EpyEKAwS89MrooylF7DxpYigByznws2/6dithCOMsQ1Uh1U
ph4asCh5uuPIJ/9n/nCNCn7j1wQkb97WjfgFr15DWUcdTIw3LriyTBphW7A8JpV
RkWiwt1VMVJiukGu7HsDgLTyDtd5SPnaJxerrdgh6oUdaeLtgREpJbGmmepHZbh
STiqxKPYHzycYSz/imTHs6ziGlaVuVfK1U3+LGf39E3Xgjjzy2Jgpd9sPAIYXuB7i
+jW4lI93rkMg+RyUe4sNO9z0CiW8XCQ09m6a0KhAG7TY+1AHfSeTbibAWXAOjqBt
docVwG/za0mGOpf1MWPtTEQxk/xXP3opow1rEYdeVGNFwDK+byECjmS0chW21AZ
WPc0G1SbX9Gxd6MjmvDq3swf68SBWEkCUT6FzBHx1/tqx9jm/afuW792zsqXvzIs
XfRpkDnHVBdqdgpkmfbjmE84MdpvY4Ia7rSQq1bYqgzpuXEEjNFR+6q01sidKFEa
hA8LPbCB8YT86HGMWavJ2k0NmdDqKQJi/7QK5Dq6pN51JZZkJFCcysJCYDW7WXo
4HiX7QAXLRLfpaIxXsorcxidMODuXxDah7YOGfisV5WzksqgEsoZJrc21mlchFxx
1MI9Ashp7pZVyspWA5GhKXjEOD52kEGaZM2F41JFOzMWf+S8jgmAYoxKehWhIZ/n
/6nkBCxfe3N28PhGos5UWiuwe1D16KBndxnnbjsaTiT/tXYaD2U/7OPOF4Jdngfw
1zONBR2onlQneBnTp8uSsAx2K9IJ/kFchx3tJBFVzcWE9shyW8+KsArqe/HUEm0r
LdIqiFnXXZsETa/8MLaIfT0mX7k+m0Gpu5wH1jooJuopxARH61kNRKApdjQcodfV
yNDO+pIKp1yBy/Ryu2S3ur/Raf3TuRxU3aJzUmeLRhkezJBycRbbyibWYTgkwSXC
u8HnOaUv2D8Yr4OM4c1WQbOXozuw1IYq6jun+v6s+G0JSKHB2cyI+AG7gSDS/JEa
VDVrcklkt+mat/BDHSidprTXa4DrTLWUZGZFYjjI6WI5k2F7Iy8wk9LBdl34tkT6
okv503aW8U4e4XHDQ+Xqrx5NxTUV33nTChsLeS+cHYihaScLo4xQXTV8owWqPBZq

jVAVBkfkkq111DCp4Mpfe5nHWKqdXF3eUZyDpOsdPz0MPKHICjifpxFlthtmkV/io
L97zVJ1TN65cHSZ0QHg256U2qvB6nmJhxFP1PhnVg79PEV5G1AdmNVwfxP+oXkGe
JvP8I7FNBC4Si711tEyfRupEk4jaJuJZTH2KS4UXGnfrHCeeacCGn0nacFvWUXSR
mE4cvwYTI9A0bhdNPFn2QnTnazNWBkgXI1/A76x6H3Shzfs/bm5fPOnSyObWFamQ
/te5QwG8W3seg0lrKd5y2OJPdn8IXG945KrJ9htWRXlxihI+RRFSawJvD5WrsS2j
/LlobmLNEYcT3o1+mQKj9P5yJHtRfLzQiMzExSBw1SPG4uAnUhaG9bfxP8BbosZV
Q5wXBA6QmkzjTB662M+8fUC8t3ictgC1Sp4abIUNlsVnNrn/7BbjGK8g/G9s8HXH
mhPVUrMzOu6r74ZCj1ErciHW2ViJG9xaBp7cRerGeGORuQyIgbQSe1fb814D3Sz6
dxtB9z6Pin08MDfD2eWsAnfpVs/6HpV6z7BBjaOiisWMuX+eN0Pr766iWjpsgeLa
fg/Pkmqy89cJ6kn5I37/1JrkYU1537G4VCPq68bHUatUTBSSrczNRv2D4ircEupi
aHd2RiMiS1iUiYERGqzVXQG2cKzyolxVg1HzmDKJX6DtRV9WhvSfgrRDYR2/LGG2
KDQ+JO1L+wwYjb04XGjmiP5Cu7zJ+r41GwSMGZS7MNNXkXpvoCdF5tyzbVCp/gSi
fApUNNQILq2HyD3c4fkCH3hUpE99N40mfeNizOrns9oJZL1DScma3eiR1bj9HVaY
pkTdA41kNeZHLGzg1leyYrKfQ5OqP2JROOMtXQ6/Qz1Howm2b+QAvVSMEOa49mop
YIQ06GBJ1jso76394bD/h9Xm/Jsh08Gnlw+0w3QjINBnCtDzKNxYqbP3+APTWHjO
qRyIaGd3StrluNjbNsW2fRojm1Nvc7eJlndGpecnCcfcimLCqBK5dy6Yg3IwUDeQ1
iz4k2RzUM0R4NKzuVi/t9iZ/1j17NTRgQRKGQ5KDjs5iVKDgCHi97iWWqaDSYYh+
SS0jyzgxBDyLT54/cVYbPEBKaxFQrNV7caJm6ESjH/IZS4DKUaw6qlTgppL5KMhu
wrdiv5sgTlgWTKQ7uihPK5sHctk6pXkwLoKPt1K+h12DHFjUFM0+/ZJeqflmYrDd
VpkrSTATGYIRwm6xpmTZ9UJxyBL/LtUFWrzS1HhdRm0voZW5MmNtsvag/OYTjwrA
rjUGUotkVX+RP970jOOnsluJ3AjpKa8LwtfvqqfLYKXkmlhofkoh4CR2+bmh0iDv
EzXj/rNsg/8IGUGxrAwmTzFMSiw0Ek2hdmfsyqNXY2yOlixnd4uzQfnJMT3Sozeu
hVy7GjtPshut/MBPWvvpBSnLmBdqjF4S749iorKESScUSaECqXpip/ebdRLV7eD
3FIAPmFsgxYsLBEBRSN7Zs1gFEk9P26jE3b2GVvdlTT7QUz90Nsta6x4PopCOUtt
EepMDpP+/gcjgZuks3rAdq4908+qx4i/026C1FF/VyUIFVBpq9BlqxPhL5c2bq8C
wky8hOs1hyi6vkb06HF28EVAi99kIz1oZNWoe4J2EX+a/OuVJqI+EI+wAPD1CMY
ERB72QK0PDxqo8kpW7ZG2Y6bQJxnc1HyL2zu5vMcrWs8Wj2nRvuEpf72ScKVKgJE
K6hwd4ms6LEtTXg/Eo0qT1HdbUTtvqzLZNP5aMfeh43+j15f6MiapGpwcC7ysoeA
XnkS0w84nauOLsrCqXbmXHVBLPzqKbad5b2Spw68mqXu2n4ehKckkYfKJBhM+M3u
lbQz8Hxpdp6qqZ8k0lQW5s2ICrSr2Ecs+pf7019WPOkbMQ/zCtX7u45jLHW5eoT4
P4Q4QUH5x10wi+3j12dqs8TkVE07SsJ/WO8N1vi1WInS4bpjnHX1X7Lze5U+sZy
/hwzlg/AMBmYVFG1vpgDLGu+uK7qQe90dN1NqcF6JR8YZvcUc+KQApmXOJc7eyyg
Mg011DaFWaOUPNFA8Px4+zWq6j4R9WDKAdX6WutMRfQUZdDY01mB/S2A2ASTBiIR
Z7/6ss/Q8aAq7iFlOtWHD6KoupswIuAy6aPdUmSL1cwTiuVOppzazhv01RwC2JWN
9hKOO+TQe7bZLSXDiPHEhh6TdRhEoChJJBSSwiA3wkfb5d2y8P1Gd3WZby+gyScb
lgzOcu06tKesa2Bv+/YIK1h6X6FobFQDsBvfpleenUqNqH2SnrREV9Gqu+40bZ/K
rvrKuRaOq1ZHHz1BStWiVQvfgjAw8PwvpS6BVLly1CtVD0u55MiTCadYxAJmHfF8K
FE+AgT8f8JKpVFcu2DvsQ3tbU2Y85LFm+SNzEhZ2yaDmVvSU0zpMKDp8/uc/Y0H9
08TLriiMt8eEq08Zr7/02X6fCn3i7r2RtYzECBikY5LQJ9R9hIWRIYykEwKLDeEK
YfGCqtKWnKVRH9dD0JfFYaok5jnJo5zrtazBipDT1wq/f5QAvEKapOXR6U/bdV8q
ITyNJVxbXunZA2QHlopJu5PkH3Zf9R4awUpoozPwagjG7t3Odl+p7j1Wdn3ODTzD
5jAkWoeEw8OAu4sJp7tCY8Oh4qcGqB6pdhdMOxbSquiQ8DpM/frLnCn9AXgavnlf
7QSEZte1qEWoguQ0ZhMHeFLf41D/CcjUQ6ZiNYTLMFjVgU5oKLrxC6YyADkEt84z
X1vniPeE09HXD4AEkC4SZv0Td2pWLGvriWTXZQxZA66rbMVopPdbLh7R6S8DZNej
FsU1TZms4X7y4hfT9kG0daRgboLpS1PtNiXSsbrRBxd5qSRtic/3zo/7BVkl+tn3
lQO5dbcLpCWB1kugz/gawwa7saPJcy3H8DKKsguEetNDevwejSEazPt3drw8aYnk
2eVKNpGmAOL4MjaJHxea63UPKhA5mzEJGaqMvk6ZjTpkHMPVsoEhcDIBgEMMTsm
6MnXBW+uXWtiLi/ZaCRAyneogNlnhsPp/IiQ5iz6VHqeO8EwZe4OYoMTbFWSec33

my2r8kJ6FdYNMQsi4RbDf25edVOzZCwiAUOk4VgHWThZKqbQ5PMgx6NpCgOEmS57
5U1fXhdFXvhiC1f7YrPtaaDspWrD3RNYhmS6EUZ+rh2y7oU0Qrqt4TLowAeiODDm
7r0ib2e6B+0EzeUsQuc4j40K51VuNoSHMROQvE/Vbd29nz6jlgw6IITwOkC1/Uuq
M9CywwguyPoThftSFUaTGKAjhnYhuLEOXI0iSNfIbcceLuSARpcwUZuGd82Rwf23
17UvrNguK7U2HrHvMJ9DbV1ELQIEJBEPkdIBbaejKLxLSVrwSskjZis9fnXcqSy7
kAC8nfTr0aBVk5RaH28P4qRFEdROKY9SNKHrHnhi57scrzcToYVgB7pYyk0hgnh
zBdSxXWOTgBvOPuk3OY+ob+ZSdQ4pEhs10U8AT5V7XKy+SvphZ5CT/LkcIvuWvku
1RwgE1MFpAg+4BDqCAW41sGea5RPALHIwiEkDu1a217gMeOVFCN3QO34+/yYoTti
W59mBgVUABv2rbmL4+8KUAXBVpx5XmE9SEH9w7jOaCIX589K+gn++ea1PZrbuaQC
aSlUCTbV3t1Jt8SNYVm3pzqs34GSKEz54cWAIIVIKGODPLiza5hc++HnDOPBbk0ss
Fu4ertJzRwxsdIY6NkV+T3PKinYVOUhWaE9AfXVH1+U6Y0wV8TD5ZzMy6gYNGxAY
+elu881wbllovSNVfkka9rcVD0zyXaWVZ6bXNxxzJiT/ctcUdsVdiCYWY67RiwCjRv
Q4GE6JpIdviPsr+3WpVlggXDHUfhBdaRk6BqaYjEHJPwLC6Xcpp6tn6JUtaJKBvEv
kw+ry74RXm0iPY4zzN1uo8jRdhMcRo3QM43B2ny1UZrA0gvtS7jFzdbgcQSiJdxC
PMZqLRRzoJtOrfBEtrT+Gc1Zv0XU/FWtdLguk8FeAiNQKHESoD56t43Vu337yo3C
c6xXtOPMXb4Nov4nM4MBzYr1T0tk0JqJB+egAnskd4cNK5IJGaJWZLqAOKkEFPPb
wMr/DFN7Sk0iLKYocrDTNI/5SZOvGzdW+TK3NwwnNEfuDKKv1IGoM5aDZmC9wUpJ
INIVDIWT/jtvmik6uHShOuoM8JFGRcfA6wXmxmEzGiBPkucBE37RLXjU5EGtZ1P
OLwJyFevCtiWl+NhfzZAUxnF4Haowxao21hXZiwZ5Lj4N9VfiWaM6aW2SXhYomUF
T95mjF388hS5yMwC2Vd91uN39BN527R3VUKT/fKQOg0HwfERMFdGPsm0cvPfFh5J
A2KK3zhIANF/hQh8LbIQesy3gCe2RYLwQYuoU0gh9sTx4Pn2LGxuSd06Pm+Iuh3
Ve9/tOxxFOcANPrMpS8W2cTMTci2MxN2MpAu0ITu5VeUPkaptdHBjN8YewRGHJB
tuzGcPIkuPRDFtPu7vHTPJXZvpbH+MyvCyGKotSI3lboBo7PXgMMfS2mR0sn1vb3
gKPr2p8zMy5YcX6Tf9zecEr1GVNC9WGXGzOrGz0N9jQbDZKbXVyI1sZh+AjIw+6x2
ztqEbXNPb2uP/RSCqciSljodg+p+P/EAQAeT6C/AsN0sSkFtm5P+4//QQId+Mdi4
MufzV1VbtHwJOT+aNXwYKa/ahFwZeae2KXgSELroRoaoe/qoiOR+apq6uLg8FkM1
OtA27JRz+S2leXCeX23BrKXYmi96dN/E7Gd+qUSDa8OGDnq7+Dq+SLKy2WwtTkLT
0sSBPgZCoatFx2k4F8FnBr/USXQJJ+97iAWjqTVqbas/YvzdQKey4N/ZMrMq3oAy
q3Ei5pEo3Z54zQ0qGeNYH16OW68qU3fT1TP1qUOWRrJoalwxeLRRQiJHa0IwxInj
kwKKht5zbcACm4ExqUYKbsQzZBBp2HzIXiWYBRnWUFZtho8KME4rjkv2useTcTiB
dLwr1gJj0fhB5vBSe3Zxc6ct7OI//SlUKClB265o+ocbQCyilNLwMyLpfIkPPGUO
R06Fw0gNEelvEuPwI12InP4sRTUOP/bH/MJFSDOE/hCQNP7RptQ5Goa4kVAQvb9h
ewEu07Ky3k+LstjNwynhV1wIM2vC49kVqE1D4sFZrCzjov1ufxhvOjlyWpL7P1Gw
SZ4a4GGDiJr9weHvOD8Jgr98U1r3UDPxF900Ucvr2289xC5Yoaq0vkAs0HOAusd9
5Jws02rmlaE/xJgMoxDyWphpHvLXQMGvRdEdpwHDn6LgdXYUF8dMFOxFXRsJWRmR
IhtW5b/GPamGHJrE/2967Ow0anuM/jzV6DddDaU1Nj+saJa7SuZDQqs+n1XWjxH/
NBPab+4Wr1r3eJjTNxHkg+p92zJVXQqpHkZ45Zry4k3mRByws8u9NWQ9Y+3oQyKS
cDYd8gDc5yv4DHZXCvOqHrpPSdc1gtTDq8fdByeeM6B9Q1aUF7Pn3XeV9fNkApB6
6fqbecdqyzoFyD5+zjbD64+XOgsxDRGO0GPdnj9cAsXVoD1sG/iu9yyWcRdIgfL2
2U58BewyeOwnwdedNn/NiChYwJZpbL2Gly6IfZHRjXfbRsMNPuPi5ETWNzixs4TP
jCiZ4Mcs6Hg8mTLcwMs30teCY7bJuy6g4AxsRWDgOhS3VS0uKJDxtkYYIsxor8/7
o47xu+jpQaqXzpcjPs3NPKQCXhHcEQlqTKQ1XQ4iTZDDisBo0vbt/5M0no9ZqrGA
x068PKdPRefVDHkGOTDDTYMJzDi0ZsSZk5fCgXf6IurWeWVpB5Y3cMf+hIwmJnMZ
Hb5knGH1z5BRYXLFBvJTF5kGDxKa+naJrOZrJD0CexsFWKV01NhFHOOH6ba7srmw

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] 10445 bytes
  (decrypts to)
  application/pkcs7-mime [smime.p7m] 6720 bytes
    (unwraps to)
    multipart/mixed 2283 bytes
      multipart/alternative 1455 bytes
        text/plain 497 bytes
        text/html 649 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-reply@lhp.example>
e>
From: Alice <alice@smime.example>
To: Bob <bob@smime.example>
Date: Sat, 20 Feb 2021 12:16:02 -0500
In-Reply-To:
  <smime-enc-signed-complex-injected-minimal-legacy@lhp.example>
References:
  <smime-enc-signed-complex-injected-minimal-legacy@lhp.example>
```

```
MIIEHAYJKoZIhvcNAQcDoIIeDTCCHgkCAQAxggMQMIIBhAIBADBsmFUxDTALBgNV
BAoTBELFVEYxETAPBgNVBAsTCExBTVBTIFdHMTEwLWYDVQQDEyhTYWlwbGUgTEFN
UFMgU1NBIEENlcnRpZmljYXRpb24gQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIeh7j00
Boq0MA0GCSqGSIB3DQEBAQUABIIBAHrX9uy/oQe7y+9tegYXxu2T9sAJqzKM2LCf
sMR9WNLx5AZ1A8iQOHilviTas4EvYkPgJzfadPRq51F98h1MGWacPvYgKlbyVdu5
ubwR2pIkRpttWRULid17OwadsTnbL539iRWWzWMakKPEh00oSsrDAUbe2INawzzs
H/aJSTjtFZoeVtwRH+c7+WiTsK+L02MnbqBLhrIUjPXq753QToNcUYbj4iFWtnku
gUFfdkhrcwmzEzOmM66L9kwqvnfqjpCbX8A5Q0sVYGZc4nuXzgY4F8PYKtrGwq7c
tLX+CPJ4X2rqH3KqghhRu+TfeVtVR4RQ1TOP1YyFdjlGDqHbAC0wggGEAgEAMGww
VTENMASGA1UEChMESUVURjERMA8GA1UECXMITEFNUFMgV0cxMTAvBgNVBAMTKFhN
bXBsZSBMQUU1QyBSU0EgQ2VydGlmaWNhdGlvbiBBdXRob3JpdHkCEzB8R0APhiY6
HGLS64MvlsDXhpQwDQYJKoZIhvcNAQEBBQAEggEAmCHkN9nwaIBtxA0yYkitWt8D
```

1J/TjluocqIubp1kShAIee33NF+5T0u7bNLseaqu5urdWlCk5z4Qh94iUA+En95C
+kpTNvWtXyIoWgRX6C/TSup3ETM/DE6BpdKTM0VvzV3rNAkvIVzBocTYnULspM8L
be074Vkw1oJQzvVJ1U3kDA0a9s7R999SnfYw01MppzK7fjnKswLkaw7SpkfbR/4t
9ogphGdg+GgskkFarJF3tJtXRik6M6HZGvvsognVKJCdVF1EgLDsyerBr6WhvJk9
oHoXkwb8oXkJ/UNqq9Xu2Ymg3G2cL4bvHgQLoxTTtg0M4uYFrqkren7v07yzPTCC
Gu4GCSqGSib3DQEHATAdBg1ghkgBZQMEAQIEEJN4/uWhaPtLhndZalT20q2AghrA
DzzcVJ/3c+0CpoBDEH/meyCF1mPrf3+qcrvGwbBABAlrKY6u/r2YSJskaJa+GEIV
kMO+ImIqUfWlJtOHsT5xC0yU26T0kyFayWIPsdoewTsmUzbtv9jovXsuBAApDy9S
jjKhe4Z+A9mJoWZj4vc0Kma8qvWY7gWJoKvi/62hiycut12ppRGDpucRgTpSM4Sj
n7GVlA1gSO9od9GxQ2lbLYRx1m7GQ7ChVQPEYoLOENESrle14Hq8+0sCcNw6cP2J
6wiblt0DfMkIHHA+ZJmKJJ+JeNmC+p4saEHIqsPm6QVQEUhpy7T5TEw+dBrm4i96
PrsB3j06ZoTBzih0Y0UBccnul26JMom7QCmo4voFnhu0ZTh0J9E4H7Kf/g6oe0H0
NdtjqXvFwEmFyp+C+c1X9dJauWjNSvyq4mTbD25mtg4zKh0bHotFP4fEi0vPRaJF
x6N54uuJVAWSjmV68nb8POCEqPjQqKyMIXdWvFEotd1FLrExkxh6S5D6IB01Opon
7lrFDkHIAVvZycqZ34gItB0ELH+DEmIUr2d+/slzztYY5X1soQ2pkAPXugtYMq9j
VGUF0dv9piFo5g53FpDJAZKSYFhtSuZblvc+8ws3Fa57CUFoTVSszSbGL+1dPTXCB
50zvNXsmiMSpR0wLKCrsdi0fCwrKusHS7XOI/A5hoX8di28RGmR2gkp2zmPnbYyv
1xLSGqQ+PAS8nNMrdNAVS7oMEab/avu+0LuZqcaQhADX+jwApOTA2MC1B+BxvRow
iqdE/pYUh2BqZRePd81UffmR6PQQMCHe3HQDfalo5B5r6o5AipkjoVw3bpWgKwTz
SQMX509K/pjEkMcOnvdLx0J/YIn6kfJm1c+Wn+8QrmTfejOMi05Pp8TKAdGs1L58
2bizDK857Fkxjz0dUz0Xi/5hCZxTiKIDbdGrUj1RL6PsM0w6NFgGxct08G9r0u3Q
kFE48HI5EVs8Tvc/w9Gchy51CZ6zUWVXjuhkUHUT98teTvmLPR1DctrrfDcYD/kJ
woptpWkI48LzRZMRfKrIP7MorWT6vcrZl7SRleAPGmd4Yjsn88oiPgsgKcd0P90R
HndJFpkHaFF1lezy5+dk/sE4tY8xWRFxsPMVfp7tBYwQqN3x3BfuLFM6/7Af4FN
xGLIH9iGCBmfq1JPYk0XAL2oVy5Byz/xudIHqSgeqP3+2611MZ7kQ+3swrKXhx4+
MVBvg5NU1KbCBCmIp+fGv0FDiTo/ytC3kQIF00QgQ+ZiZb85Us/JzHw5jfwH1kYz
D/RKHPdhvbTzMX7+Ta+KthM2WbZhS953R7hWbTBL01KJYleS6KPjnv1cGSouFAd8
WP1WowObD4agmewE0AGJrXn99ZpzQaGyQnTBao78IkBrdqkMc//Q6SLGPbVRTWJm
5Pyz0NEn8sKXN4xT0mbHM5rphnHUz8CcfQtGG2YfiloqBWRG80xY0WDKeXoU8Lp8
nvbJL+1tLc3UdpqTuQPH9TtUTJHFPoiXD5ka4dGu9ciFv94h1UMGuqD3gueaHGLj
Dyy5ctxAY2RVmC0VSgXrj9RMKNGSTWPb9ysZegl5RJA9zioyPSz71IukJm0MOD1
j9UK6gTM1WxuFcaXZK13dfkB6DsMmctP7G+M1bTCQ96beE8J4b7W11w89Yd2pgt
c0Pa5xomT1c1sa5UTTXvetX+F+JDrkQT8A9jEZmFgW4R6xJ/17X1gY0g9YDqZUce
nNqLBH1nwXJ15xrGNo778M2SR5sNCb3rKHozT/xtCXqg6kbvuyynAPwJfvrulCnsX
mj7+2HHO83UJk0o14BfUfXMZZLY6CdwVEqRgtVXTR9ImEUKcNj1N2hzjFND8CEad
p4LfJ3N6YgEWqT5BsmJ1TqJ8ipZHSTpC6C10h9ejsAhs7MvMUDWVOTIzXuyqCNgD
1L5BdMUR4NkLeAo+EDS/T34gJCBtWg7MfD0ebVbMvMXdtq5JR29V0RauWSNbIfq
oc+ophaLmAovD/HByZzdeDFr2tSKxQ76xohE42HugO1JD14whXHbFR1vhM/j0bMA
JJ4sYx+U4qRIY0lopa4uQTpLqbjd9xgWSR3eQvq1l5cpL6rH4pHYHHihoYmcy5b3
v06Psq/uW4juVo5keQ+R1lyii/TMITHKdNZZvxfA6CMC1aPZxS8dkhgWIf+OfV7
VKX93zUvWUwOGNnir/JDPNqTtu4KqCve/pMM2WyATwvEwmwQnwfnOV2tgHTkGRV7
CKMnteCyuOsaLOyJ2H1IzhFg5i9573eMXQpajJAVFJ6NccSjJ9USCeP74EL885m8
cK/T3aTTyXs8w4jDIj4LkAyNnnc4RAY1lFfFLMaGVEmO4ELG9MepFIGeNV9ek15C
M9JpvEoHHWboyzqvddT6az++vW/D4F1ZCQFCwVGgt0bgOokcL/5FBbP8x7QjzdfQ
RBirgQr2hzSV944IU+x9nzhLPcs1y31BjjKkKCgK3bL5PsjRz1wd4CxBed/qkQfC
5lgaX+mo7p9fxnP1g2ZUIImv79ERuqHo2EL/RmlmpKFtCHEAxdbcPaZNRLzhvWjO
IN9XB1JqvEqP0PdJpCSPiZcp/RXVviRlCwa+GVFhJGymvKtn8a5pttqRBHQiYoni
R5E+5jd/aLdS8+/rwmVjxou3QtYNHi7z+kR/4IE41+Ih/Vemu/fyahAAqiYv837c

Wmewk6Brnc9h1bIDwsngSluGBSPHiIcK4wFRttqyl/0DkIukpoBjqyqTWsDulZ3X
k/1UrO+nP1cSzmD6FoYtag80yz7vqzSpk35b5oMf0oLxQB8nAEaQge4Jabxo1nuq
H0NIIm9l9yBC2RuIsEibGcNT4u3nLyVSetpioIEN5q+HYuXVNeprsrfeMNMnAFSfxg
fWETq/iRyF8lTChlI/lgUFSYjd8toXthEcsbC/rABnMrNAslGlsVy5d/wE3J0er7
mr5+vSh72veXZ7pvkk1lXu5Ued8jXHFJ5Z8tLKGs3oQrcx7TIa+23/R52uv1Nh84
VaREBqQe8cGcAIQ8HcwhG9/xcc5LL/yznWbZaJe1NKEhP0jaRsxxXaTlpIUO46B
L3LrvCuqfRVWk1ke3lXzmkaR8JN0QyDGLRWDyTtap786x+XefWKoejbzCffSpnwY
0cqGv1nfuf7BSE95cqhqlNS4be7EapwSe094iusmLrNsINSCDirMBmDc3yt9HwiL
NdJl1Pwuubxgn7Wv6y4jhw+gcfO+jnkKpnlaf1Dw+EZj0R00028dbItWmdsMS9LT
lv1/lhox6f0livULgwbD4UHawJpSZY1wg29Bh0+EyPUCGUZYCBGAfHAny12j9Wk2
B/gZ6d5HGE0XWGTsguOnNnFP6xyJq/kRonyBfvukT0AQ6dCTquBnCwLFmfobN3mZ
pwniDmCmlz9XFz3PQy7nHp0MvQhFEePu+0r/SY3PsoOZaMsBgVt3tMurznmYRXx4
sjrEyzQ7k8TAKhN7a2QmI66++OFp3HKIKowwAAIqE3zer/cJZEu25GsOESlfzZe6
9BYcQke1f8GGFjgl0DOu3B+ZNSeB5+Arjjvy7joocRiaN75yFVwFzPwCIEHfzMs
ts00IrmY0nhznA2bMt1lR4cjJZdfWQ3XTWf9/6+4ktE1erstISjvMBaCJdIrDTIC
rgNOXHmLpj9D/eqNgtQHdSneFwIiq90xRdI+vDU8kkf00tG8PvT1DacVXWPEMctk
PP2SrEstVocHcrh9joSrvo/UcIYQMR5DFJLkdSlfuTIG5F8KwEKWix9++m//rlx
5t18KTDQMLrM2VzSvyrjhuqRNAR7KjjPZjyFuMA6s9GMj3WR/HxKGgQ/R2+oyh9l
9Gykf7oCZJbVd9b7amMwUNahOTks0UFAmdebe06DwXIg29P5LM65IOSs75xRQDqz
Yx5wgxrLJeZmYsHZSzFfribNZ1liKGWiY3qjPy+hHbRVS6yeRvMj3oHSo9p9feN
rjFfR1K5icZf18Cyob414ampJqisPWpy8hy7muaSzNnFnIj1MSpqwsLpGBahia8A
5pwKqysj9Fh2PB1gSOUoqx40b+1eDp0ZpHUcbkWdXT7S8SX299ipeoOetAdSyoNf
UIbf8dKF+kUHVzKjPx2whDjfgCXRgn3wm57m3ytJIEWJNihrZ4SOfnYBi+PJSCHF
ZgLiRMJ1LiXoXhps9XeuKyxW7uBGiGqWbpDmC1k3R7aEN7es3mxJ86ZBs73y1DMB
RVp46c0ABaXyrMFLKZ1y9Cy4Y51j6c6H3Clac0yp90uERHUTrv00bHsMwNDG+Yd
s5UZXZPvj7/Gq2nStN32/1b0GZKxM9Lop/7pidZDDwALawHsasewceEDS3mNccVG
DxBBCtw2t1zLNkGXKYdnPryyeBi7IvtRy0GbNGzaRmrDIP4wcwHutGBs2VZVp70X
wAiJ44z1G8/tFCJ6FIYppK4Wks6Sa0LmkpAyXzZEa48CQ8VbgFWb5fuRw+mR3dVP
GtJLTrTtqXe3SqD8l9fwaf98gWT6iNw5mQYVHwHkMEj0/39DVwqJsuy1K6FKI8CE
WHZaOmFrn+8tekEdwBVSBFFc8dYrNSSYSsAKR6Pt5w6yYqlmj4qcUy7mw1hGCVkU
/MBZUtSxmt5xbMBOFhD5aOvFAr4dGUnE/vixJ0b8dJ9O9GGXCg7NQUMZhswh/CkI
3Xzx0M6LudHZsYPwmAeYVVML0Ud4v2zUMRaJQw0mxQdCIae+DcDr7z0hstApqFz
w3gkrgeCordiC/yCktzD3NtHXT6MEdsNoXkOTCFoyW24BiT2QvIZL7XK2K04tQmPg
KR4YgOCK26K06w3xs2b0lqHJv1jz0251w+04DtJCFktBhwiUnB/BjcwrgD0wpqeM
fUejPndVXg/SzM5XSAX00kFZa5/UtR6gGHSJ4KYuA/LL+82pglGygefXL7OGAfyl
VZz0Hj0RPMozUkkMOS8GKGKVNOPAMotr6MNU+2NryVpcu5mQ9ZCcIqtruhgS8NS
QOFp8my2x93TF/1OX1N6S65sHpjFTVDh6h4W+q21f7R6Ri+M0qiuxKMURLx1rbY+
A7VsQX10bqeS4uWt9WR1EZXF06IAkYFUPMDQf5XOkX4h29G2+QBvd2tx0IcGU9b6
kez4PeulMkX/6fBReyso+PPmc2rcQQyKByDZskE7lYNGg1cPulIaq3oX2tVmVtB5
kHjuLuFq4tw3UB7eYO2G0JnibQAC73+tKME+uA5+pyLi07C3RfVy+T5XJR+EQ1LY
BNS20NxiU79Y0F6NF/uL5pbtqTJQEItvxQo1jSwKtvDBwLE/2uWfs2L2PPglKkC4
Wi3myKYZR8j4U437sMF7XP7FyD475AYUu7xn6UpjuBXu2gTl8PRYHOMumaElpL25
MC1fUoiVyWjRZlIa35EsliR6GOrDhgt50rlCy5pQuOrzG0e5jj2sKkSkG33LQCKC
rzm/YhJdnnGHarmPuR73Kj3YgDhvz0hGEUmeTFeqo/URf01kKhcI8aaJfLsHrbva
yLduultV4Rz80ny9UHREdNk jV6kGeMDCqG0b0YKQmZ2U7lyo6qegNXTlvpURnnzq
nbaVtyQqJugGvZA/9lbAxEka+xWLX2VeC9Rva/RaTuluRS4TRFF2EzPzfz3s46uOr
7cimaZ7oITiv5SygK2K4oM2a4OPComnK0GQ07suAVmMtspZxZW4zELOMP+xJnWgB
JzXDIIWAOaIR3hEYhi98y0721G7o+WORYi53DjmfYaQYikx9IpIY2g1jIploNvVB

DWh1FuEk1pZ8jXyhNRAmbcMtSYmgptFcE+BDolDZh3Zc4ps0Wjcg/c7A9sqNf87
3lCKLoKP0UV6M6KRvat0AjLaVmwfNvQphUQzOGevQ+H0h+7WZJ21CVd8DxH6c7Id
m60ioaEHfr40EZysB+sPqPbSYPNZo+oH/nIGhub1AQ/zcXSupFd0S21DkStCKkK8
9W2xTfqiB3pB6zf5n4T+S8/RXXX91xfvMk4NWvd1KDX7N7PPpE00N2A105SXQt2L
pLauSBWVfb+YOIx072R7gHw000ZogqG+M4HfQAGvUk2P5AN2FcEn4nn/z158xwJd
YYWqWajg78qja6dLNhoY3mPupeQlLmHArU3NfdksCwpoyAnhEFAzzrzkulB1EZHT
746Zfiu6nBF8KPW1Zxk8SqfWYJsbXe4UJHL/2Pn0UCoqI55YZDBT9xsWhwrHMvXf
mMrvGQ7iKzMTqY54R8LqHF44NcyXIIdaX8rP7IguHq7EVNTRIjT3tW8k4kjaq4EsU
PkPIpQabENvUhdRolaLRAnq9PXJvjzW5bcM29y9tR5sMrJ5pXDOHp3pLZQYeDVWo
UzRwKRVZ+o3rraTnc4BVUZlrGwfsbI4AG8AesgiVgc9z03xVKjNTLYOWZOq/wC9p
enRMYkFTRteRx99FzrqVlyrkiYZ40l6DT1QorbUUzHArZG7mwedLIs4YyXaJTywD
Qo0fjyfEkhnM5TvUX+K5FMJjzG1Ls9N01sXDX5dkzBm8Gd/E0mSZ6wZ4cp8KvVmN
XXGjYzQjhr5fuEahuLzfumxjYrS4yB6nrmfsMTLxfRd9CI29YntG+dyOrR8m1Au6
QTgVfiZ4m4BZuznWoAvRfCmBYfOaoqOrZOWeVp0YPPFs4wDnuMXGqGtflZWEEKIdJ
HmJawnx5hzcRemznAAXQ5NTu0kUfXfxx74ar7bfpFm8CdnAfzEEVEjdfbJHm0uv9
sTV7Yq1EB5thCy21jjz/3d6rkUp/oFvL6L0XRBAPGAVFVQYj1M5XgsMRLZDz2VDy
UyBUn2avhiAxcFEiuhrrhrZOEwysP9VB6b890BeNuC001yKB6YqcDaiLRiYKdTfAu
87jH247gx/yX40/35ePBaSPjpSSdW0HB30cgFNfJguDcAAq2aPL8BnXCTQv+X6cd
pKrFSXu+a6IJc+77rTCEJXCu7KANqQiRJZDjEzVsIt58K3jiYxx5bGsE9w4BTI3p
7BwaeUn84WxI60Lnc4ggWQ6UEd0Wgsn7G0ISVkkVJFYLoY8fLoa8yucVPsyYn9Cz
p5CtmPbMeY+fJ9+g3f2szBylMly3whwC+Ac64U7PoQdIXtLJ2cRUefFLdwSmrdgH
mu6KSA31FhANmEl06pnQ9A8/mRwVBEU8k3eg+UNZv+19JRaOib3uPzgz+BeLP5nb
wKgl054R0kLFafvTooXvXI2HWGZe8Q1w+gJtg4U+hAspgubGYCqQAcuh1taS2yC6
7v+uhGluXu7z6CqjWSsRTytjLXngXQCvoJDxv4LX6Z44e+mw+S7BSi7Lr+eBxVYw
5Dz1JUokGejW1f5qNT/l5dfB6nDZoYXYyG/Gu5w7siTu8+CPpC5cifiFq2ttGH7
s8nRZqZEObK777EhFCOUrQQCKYtiigB1dczWf80e1E75LpbXNxfWXSxH/Fz8gJPW
17xridMMCR/wvMQX1SLtxwAP0BnkbtGLGDUZzH5CAv2+5PG7vbj7kWyNPnXWobps
jzEd957txPQ+hHY/e7jwW5IpQBrwEPMaC8pZ+INDJMMzFnCWv/1YL4D1piMMuDmn
r/bc6SdFGe0iwIu5/FQlRzc45abUlpwUOctnxGt39YmJhxc4PexJ661mY++9ZF55
8mrlf2yB00+0+RmS0HL/J9drHfowJak+pveva+wQpy6wLGaWZlByZoVwt7fod9Vu
Moq5/gBGk7smRG0aOuV55I1YEMxSJnQcYMTqsI37TV9GHsCmdQFULk+J5IdTc8Bm
MCSk5t03BR40FA1r40mQQvRvw7ed/TD7Xk6tjttJw+hrBzzy98F80J6sQW1aaOzx
D+s0nq+e+JMzb3pGh+FqVaiDG0oEp5zgcY9SU7cNAX78VUtEZdgqxisPW88SknL8
cyMPqmXrjlx4XY4lwVh57RP2xiI8TU4bjj5LBRIiUXh3R5OHIFFSxxVHhvkQEuS
vV9lGIqkneNiB+xy9U3Yd4pGt/VShSVH6rj+s9l2uKLx82mexYo/Vh9OnqKVwIo/
TGBGibjAUrjOoAGsLwV9AvOTAqsALA5FLJmGm5PCfSamUpDR6ksuoM/eWywoYa9U
lDqlIitNd7hPomHaE6P0sBIg4+zWEB/7yV//xCXqafBWJPmfZg+HZw/nUGZ2B2XW
0qFD15ZTvgJ/V1I6D69QqZrptp923A1HsAkQt9PIG3vOac7M8EsUMGzSVxjbcR1C
ZoKKzPI7f0WYpSz18mbQHbRuyLbyS4FCCiPUgIPcsJL3abSFDSgJTzPVPXD3ApLa
mePL2wM4ETERs3M8adrXJTtU9I7ApThgMK0ZtzOBWOfVHItDlJunvBVyr9snQ+c+
412hVUhi24LTl0UeqXH+KHclrZH43eBaS4UZk5pQMSHG1GmX/ctp8D++WiCt796D
4ey4odj5TI02nA3BJNd4089ZTgNkL8MAukgJG9kX+IDry+Nl8PkGzLF9W1/IMcqH
QutsvU+/3BIz7ZT5GzgLkYhKSnlapoHOX2eevqozAlpQCdbPmBwkaf0kV3jxkIIF
APK3119EiGPaqL5TCBXva7mG4dz0fAiF4eiG8D6VZbRZ+o6MscSIfDo+x4MpSD3Q
dWoz3qPPwmHBL15sUQjbb3fMlOuY+10tjTF1zLjTbdkK2V4nVuu0vu1EGBij0/xq
uOZZWD77iJ5EBZbjBow18cq/HMGRvWMA1BYmfUpdXj48WcQ8Iv1ba8os0HhRBKG
0iZilizxkzVZXfGsBFQB5RWX0mxXnOobwn1QVYTxVtISxekyBOidef823n0n+Tpp
H+Cd7IXiAvdttcUBto9zA0ILgt6VPiI+mXemGRsyUklMpU9vNx60UFS9KeJ2bPXA

```

+/OkxdJfwCVc0i85JvCfVHVdeYTRgou5A5eTffcQau7YwRYPKi/56AUSB3hUfMjr
ka2l1xki6bwcrP/Vn3FEEn3g22U77+RcPe3wm4NKSSxrpaA0PZqIvKfqeTxn5MNI6
5YT1l8+hLJva5dc/Yiss/fMRAPfd741cjkD0Vc6ezy9LYI1jhenC8ut3oJiBvgcC
7rm/ncD7aKXHMmlo48qTwus0ujNw8rHVDOMtGtXsRU/AoKGe+mv/XKV/owN8+sK7
9IYXne5CD/jMn04I2sWzEyP3kP4mZnNcpQFOM0vIZNs=

```

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] 9750 bytes
  (decrypts to)
  application/pkcs7-mime [smime.p7m] 6210 bytes
    (unwraps to)
    message/rfc822 1970 bytes
      multipart/mixed 1906 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

```

```

MIICHayJKoZIhvcNAQcDoIIcDTCCCHAKCAQAxggMQMIIBhAIBADBBSMFUxDTALBgNV
BAoTBELFVEYxETAPBgNVBAsTCExBTvBTIFdHMTEwLWYDVQQDEyhTYW1wbGUgTEFN
UFMgU1NBIEU1cnRpb24gQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIEh7j00
Boq0MA0GCSqGSIb3DQEBAQUABIIBABoiuTpkWhhfblL1RCZbuwLzmGm3w9XY0fj+
SDxw4qBddauIERLO6YUM7k29IGzo5RQXEr/+QU80QxKUEp2vjNSnGqGpLAj0VEy8
TI52mFbeqCPJ5LxD7SWOgmI3i4tuUWcwhlIkWj73sYwqd5pOl4letK8yVIBDqN1D
uOwTKe9j9zyxHO7gl7GtWB5HJ5jAYmsoGv2bbg1T6JxlbOUmFwgV1R4g+33YiMin
sjuHAZ0EmoFH7o58au/9BOfaVrWjOgjAdn1bJps58tByZjaBYekx2FHjhWl+Zi4Y
sF1FEtleDjxZ0Bm2rpaWw6ZeCUz50YnkymOS5mC7AT9biJtw8mgwggGEAgEAMGww

```

VTENMASGA1UEChMESUVURjERMA8GA1UECXMITEFNUFMgV0cxMTAvBgNVBAMTKFNh
bXBsZSBMQU1QUyBSU0EgQ2VydGhmaWNhdGlvbiBBdXRob3JpdHkCEZB8R0APhiY6
HGLS64MvlsDXhpQwDQYJKoZIhvcNAQEBBQAEggEAY2tgCqi1NB7E4JtYGs2o2huW
iyP3292KYFjvg29uqV7CkkBAasq/ejnnunH/LBQQOGZJ+lratsdSqr1TBUE7PVKO9
pr3izAQpj7NLTbbT5ntI+17+I9SCf3gOiLXOq+f30/IAqXcLG+JgoSIBIiMitZUf
6dsiRa7g5mDvnSdl8mZo85Mf9tPs2rGvYo0dUzhlVbCcmUmW0qoGVvIFsimm5URh
Z3o/hDSVhaD6n24mNEuQqLcppYhcGWK6PAF7lKsqspk4RUgtmK8GyqX2gM+qglqy
dk1wTPrw84ZEi5ERcZI3mn+8gG3C7fUqvWYeikNuQjHRqm4cK3A4TQbWmBGUNDC
GO4GCSqGSIB3DQEHATAAdBg1ghkgBZQMEAAIEECWAFD4S4DqPDZVZGaRlOuKAghjA
38c9JsS+smudhk2lgcqxGWgqxm6ZqJ6N5eg0EVWslLtBo01B0Zc74xnKfB+zTMfS
RQy/b0QNTfJW05pYEjct86o6AOoDUdNAT1mR0BlCKKyvwxYqHACje0+DejbItz5
LaJ0abNLDfgyrXR1qlwXaTUFM+r7sBYq9RDM65/kjFvv0dcKJlQHvcsGG0k6XzWT
1z18D5m43m52IQSMITBlGQmhFuI45+KTJ+U1E9Zs8EphSvLNX0vTs3QaWsF3/8
AWn7TxxH/zu9Qu0DykHkGOFNjRGWj7ufjC0ROQbjIaVioUKvSKcM1DkPbQ7tNfo9
T52RnWuJbExEwhwLXjoGjR5zwzMNWKIkYbGunRlPDrgq2fsd2ewlJu70URliKEQ2
ws7BzRjb3JYnWCy830jRuVGpvLzo2WpQyqWxhSYzxuAl2oseGqfZ44P0ezy8YX7+
09noeKem2F2+XFLDrV3mL3OSEGEYqiTg2aalRxFaiUpz2IM158Exm3xv87F3oGrV
/NsV2g1f8sjkwdQ4k2K6IC7Rplk5jUBixRGC1FsFoXKWofol9/r1RIS7fvKpWUBh
SdQQ6VxkOTeBdXG8oPm+KZ24XnEPniJ1AduGFU5EPFol7MCGCPG41SjtwZM9vxl
v3HV4zNDZAbUvST2YCDsgZSSH7hgt8r0MzuB72PUiCBrHXA35nuRACWKG7W+Hbnp
7DvW72vbdP4WJ9BPuYcENzmVLP/4le07FI1Llx90ieh+/abBC1FKsnITC2mo8OZs
j7uYnZOKpJLyHPUATsyZ6qh/Fd1ElhiYJxmsROVUV7zuQDdXbcP7qmDNvifNwaIB
AA88/wC58QAwLIvrQeOesAOqoWa6XA2FevLkKob4zcKBuV7zLEgHEEKmebx7T+EM
13MeN6Nm0bit7+eZN+zZ9MuiUM8qlqbFj7dEMAnMMY9rbaH2d2AmXHhXVm74aLjE
eX7vcYpSyFXTRheds9jpeYalct58v9UB2Db5f43Gx2jm6+hW1NEAuBhAlzdHJ/I
yHzgGTWkaY5mu+1e07CNTgVes0hCmNgyekYxApkPDzvWaJJSi6pqfNHGki18q4Dg
gr2jQqGWMzqQX0HBKDPioF6AK7lGIAoi+ME2/8x2/wFF9in1h+cKkayz1RTubYh+
9u1TM1ECKCCAg07MQSMqc0XB1YrUuOSZBOcu978xXqc4JnS3LvmNrvh6dhekr9H+
X0ao6tIUOPR75d4hwCVUy1U1EPXekxAjG5Gp0eY/rAOh2JUMd23ScmSOIbsVNq8a
8BKB35v2mGyOBnQ50JNM/4KTOB5EMJWAL0S36jpneZksgw/Xxi6rQC8wg3eFgBXw
WFrNqzG/4HoILHbq4cbQZqymebjygcN9GCjmqKGPBbDxoiNOpGQKfCM2cyCm7eWy
Zhy6/70p296ogFP6TfCGbr9ub4y1yxXJbs7hBqVEQAWqQRgT2d83Z+XcSYsPyz+D
CSunKBKbLZlaAKcbYvSogrz5d/ANlKGhtE5SHMhI+X32ygZG/A7V98rOfskk0qaT
NGkuZeQA/wOTjZLwVMlhfDRGuBEbmedpU9lC1MXHFw2KpDEH9oNDtc03cyQZ7Ov
H3RceOTy1TgEly26Mu295h9PBlxJjY667JKai9F58Bfn5K037hj694CjiimFjcV
JhvoP3fV01tnrGD4sq7soMJNyoeDSTBbuhjUZAweWMnekaE/cpuBvf4brdLYKRa
tWJuu3ZpYLJIU1kNh3udFdVWcRhVvRRiGWU47BzvM7G8ewbLWrZARMGm/elfSTUr
WSnRM8suJXFfCscjFUHvA7a2G26bH1pcFEFj/EjgAG82J3bQwtz6kS2+tdMas6b
ry+hI0UVjKJN8umUP0agp6uFmpaCnU1tRu/9Zu+P9SOM4kuZCK0TEoFgPBrjxPG4
kVtjAV1j8ELNPWdCPJzjaz5dwK0wPi1st3RC6kOsro4yzHSM3t6MY6Bun3KtXCiS
koqqWaUx3vUvM7piART7Xv+gAsBcYu/MyDkVJKBFO1NGRx9ycEuVcnVohru96B2f
R0SXF0B9WdG6sndF6SySzsU9f+Xq6LyvSSq0L5zVBQmswBx8DE/1S+oTWqFZz4ba
SKMZU3AdT8Nrg1EuJ/OAoX+eCbYAP9EXC6itTXDGqJjYlqVbryurznLO6Gz1ro8W
dbtNjD+YDjMPT4VFs1E+KGo+9J0YNLWVqU9HsRhj6/DsgSZYp4AKi8tu2P9YhRGL
4wGK42ziR83Q9oBXSkh24etn5MBMBqwZo4nUbYKQTYg1cfYF3P5WiJUhr7OttCmq
JtGiggPyGTx4AUfDb8bAEaisxiEZlMpj8/bsQWnQ8ghj0z1/Fr9isSMT4mpVH3PN
94ElJ1eR4GC9P0zEFrPNosC21534F/xNWT6AHFV5HXlMyVyHFPFTS0x2pP8O9S8x
mBpen3OjycqSsDogIDY3hsrTy5s33p16tNrvdsRxf+Wsb4BOBOXQ+VQrr+WhJWoD

76raYeLbdcZ/R+C842FPz+lMeoQpb8zo5LQTIgjn/Z68ulRe+S/w2OaZXzDVs5ub
HIFWswpy6z1gHwEGvRYC+th009p33IWMzEuSpGApMvJ7UsKJ6iNaJwpGFzrzjMRT
2LLAlvXST2XAmgWb8t+eWJ/ToWcizoqm6Cm3uZTnfIiqjInGbX0dNV3ZjHq5F9Ik
eVnsRXS3U9Jutuah+2pUK1iVSZ1hQUkKcY2MovGcCRoFdQ8Vbvlr3sZ+QCKOe7Cn
YF69COsOnpgtzPU0CfsyHRInzVkdqdbM1DqC17IBjHVoLyWN0TtcjxvxNDCD2ih19
DY6QtVCEOYX6/sE3HE9Y4MmVqHdzq4hX68xujAhZry+9dP4clz6imAIMakJAj9BE
ZSD10C1lnk+tdz5qlzRopWkDdjNzyVsLbVuo9jkjHcGzX5LpLCqkFUzCaMR7mCpU
Qg7AW1qF27B7HVuoCTy9U2/9+XrD1USS+qWG6Oic1+FSxFD9bEy67VhN3205twID
Xc/DtD0RP9JrNRqVHH13Ciy7t6HJDukaMW2xYSlrVTT+gZXUqh9y1WHwDUTw1Sch
/K5Des8IvZnWIMseAjTSEeygp/EsLOc7yWkuNOZXu9pxZE6dDITLvGuJDLcxax1/
dsbzXrtUjXffniZgkPAFnkWA165ligKu26yM5thczRFgQbH5vofHm730T5JKzGNk
jlq8qHRxkjiFy4S28Y5u7TJEaxwgeKN1/IPkU2ixuroKCgMWHJNUVCMYnRzYjKv1
GVnNyuTYdu02aeKTe4JvO7cC9uC+PkDlnbB9e12t5Qull1t6wF85A5/jMPzPomU2
Uxc4ZGaWM31zbPsUjWVOTf+DE23nVsrdklyBsNK52U63Zu6hEX1YD3NemyGLmBdT
b3nwAKZ/6s5tmTTy5QtqwoQ4snOnUWOuv8uXzy8HRIw10MHYkoAVMpui2qT6LxBu
nSvEukN66qhV6XTmzSX6rKzh49zPNFvWjFwH30Qb/E6ekHKzrF8z2IeQ37q0aZmE
a/RFmueAl1Ihwm0ym4MRYgjoMHcnXRfTGj2QqQwCVtQZ6lmJUAqLN7y7BbqTeR8P
K/oZxgd4pFD1AH2147ewBQxYy0DLw2d0OPXQIqmXwh6MNLKyBvD0oRRFm/3lkh2C
PYnWQNhKZ364zeVESivpW1R5pBB31NZPiKrRXEHRRAR6Sd2i2s+Md2mlhLvF0Pe8
m0pVUR1QzZ5+apPRThgbZKj9iPhha1UNiOZd9HCSS08cVrV6WZq+n+KmmudxkE30
G9rMGFn16DCEYg30cVOBG0FAphpir8RoLKpGU/Kw1nOzKiQNwXK7y5jfyYR6Q/cI
hbWJ5XBKXtFaAsd72S+tgo7dNcmnIpGjI5QEDdvcskX50TUf9CAnkDGgpUkxfQ17
zfvM39IDZir4dUTZnuI9fmmCrr4yzDp9pvmv8gXnquvUmv9i0HKOJXte6vwq9K68
HaOmtYBT7EKkZFz2FmPkdvf9n3q81LTm+L43xAKh8/vtREu8uqmrr07BoiqzBi/T
mG8WbZCgo5zR0wnErf2sSGWpTlYX5GAtiMvees5k4z1Ya2kNDEwxbWkC11ymVjr7
X4ktxrqaUSFb51w6NR4Y6Hoz/nR8CLvoes5/Q+cOeSd9atBNWXiyBcBj1Ct9pTW
ANQrde9jL0bz1zrJpz06poVw7SrcuqUU1GWPkScjaewysY381lp39GqrnWfuOfI4
y+BXxKikdQuanQOrxGE/P3Q3hM1pIc71LPWMw4kWeIYYO42zgUqY6y28QhZDaiKe
r7F6Ti1L/3or6LyH3TsT7W9m8P/9pab/odpm9/Hj5GD/vQTSrcEdw9rMTTbj9W1S
X5X2hKactPQYqQLM2pJfQDVLBjnp7SHrM3Pb3PxQnxS3MdtYB0P1LaiXm1gNx+IY
igQPeBKkzGpV3itLdGCSgqxEI33Tz3EF9sYdLhKY17gHUXbbRMq2wir5DRbzDiOB
sQp96CWHxqiZf4bmpxjvM7J8BU8vsWt6PUdnYwAxsYoizvocscFv0sc+rwj6nNJY
Himws1CETDr0HVkXQymqo+sNVGLkQh+Lp8tEIqKc4qZFeVtw/2YQec7tFZLp9KcJ
3yDy/g9WDJ+54ezBz/s7kX14BwXynjy1X6oBa7GKHMEcIhwJykVtym2iduR2yBYN
ifpOBdHMjBwvxNei6+tpZJ8wvUJyrV+xCzyZpNHfztuPbNDU+XH0SKXpk91tmyed
B0e2Wi1/+GtBOMubq/TEHNcEvMTwScaHidJ1bc8wkxpOdVoh6R1QGgD9KQ3VWecO
IkNaiMcduiCbRYtGj/u7ZDfF9GM2w6eJotswrRACGqlqHForJX07MkTrECMOKL+c
3t+YfiTEBOaud+vGoDkvnOU1yNL4p1YKASQIJauDffOY9u9a3zRUXotvnZPa341R
3nMPdZB03wOGY2aZ8DWrJZhUG5E+PqNeOV8yft21UTHxQxXy0uNbp4Z/LaVIREjz
9xppWh00CQ0Z+hQ6KkWpHKckRSLQ9uzDgrgKTy9ROt0Z+mB3Q0smMxFeV7kxHpomT
hAx3e8UBulOesksRLBbFmpGsduLJGQeu7itHjaY/FJCiqNtabh5+hjt4gby8rr7/
+wP4UuaEl5nqx2KwtAYNe/qFWT339gN3co+yvWrIPYtbkJYpxNQqkEdOHDZMQltz
QwsgeiV2XBYlqnqb3kAiwwp3ilJ6Vh1Nivt9ULe5IQOyz6er+dP15HoDOx4j1SMN
gl8Of16MDPdAynaBZplhHALSedNE9e4+P908AKfHgZBnKU+eK+2+I4u7NRYwXy60
AqiXD3RUC+SuPpUto5a/OTAMiOnEyxlSD631bB1mJieDNVgYoIyRlGuqMLDijdnA
BJlC13S0dj0e50MU/ik9uK/jv2ulYzkhbBZDwxDyp7GNKHRnwMeZrn+WYvFXiaYx
7B12tGr4qQWsUYhQX2jk1WNaD5/XXVp2xvGDDfrR64FLGWOTMkqNiuvoOjJNs6Z3
G5F/omCdHI9LFBnO833IoqRaT2Lkyqx7olsrMNO+NN7EP4220fZtffagcWdqXERB

1KyjxvM0ppRSmyKFHEiIGMhp008xRku9m3s+F7L+D5mU4cR7fvA/qQy8/WEB0GYW
HVD6fvxnEZ3GJhZ0cYbk8G4eW7V6//XrEND5yoEG2mHqc3atZQEUCePdEdtZeuk5
HWyKqwZ9B/b8r2LLXEBnZPrbH6047PQrz19A0Gr8F5W5act0DB35D7vSRHk16aJn
OTDkat3EW881tFSRj3rjpmf6Uah+igycI7Ca1D+25BfbX5mSwxmskDTxAHIiOVHA
MWmQEY2G/5OEsWGIA/45oBhZxfuBnyAXD2zRrNwNOQvwLdHZbC9fBSvQVPLttq4D
t0p6q/mCY45JZUKF7YwOIKkIMvUJYFWKDCXYFCfktrZruKFs8BS1ZNKWFq+itIha
0CtiEkHXWL7HFLJ2QvF1FJ6Jrv92w9jgUXy4tVAZkShOQMtTqm1Tj4XkgoSMYc6T
KjBr3gQj2z9nIeehx1sQNNuw2fECAAA3DZD5W1UacVUs9AJje+5tKKo22HHg8To90
4rWfDKZR/LU/6Hkkialkxm1XI5dgWeGI0ZFxlUG3pqDmRODK6Yzw01z2/3XcJ1dt
CmMckfCkQFKICMMg+0R13F56NaKjVSeGb0mP3eH7OkL4v13fKyeWxHL+OPRnz8nQ
CQud6Bb0JztTPHdRTK/jT6w8F2R2/o4/qB5oDfj+w1rzKSvcAKUPsNbOpTWzFhGa
kWdFZE51CWZVz08uYIBxd7gBcQnuoh5/aJwykUGNj1Nv4e+fy38nBb0WMnp+GFuS
zorlITl3Hx5PUgz4e7x4pHVNzww93e1wmp08cdwmE6tJ8CyzRDGBBzRHKYgGYVui
XgLu/HmH4QOak8n2CX521DSRO//8FFGGSBSRTP2yHX3yRou0y2D10Ups0ruo+4FK/
APG0pzyouSOP+I3nLNUMEcvK0cA+s5D3+wcOqQLL3XaLeNpx1LPkhPYi43Xchfms
E7Z10YalmEWXxuCurLmBjM6NQRU28t4XCfoIz1blJV1Vc7B4134r4erV4G02sbr/
xTbLC848s8OFTYBf5GxqAmo2riVfsH0LYld6AnIjaai63Tf9V1ktnPGwHgefyoDN
JdKZAtkJyTen/tEwD8LkVBzHEjMN2axaM1+sQBj3RLWyn9y/74GiPXfHiOalOrI8
HB+9F2in4+R7OWzdIj1MxE2tTOWabZ+NxNpFt+iIrzHlps9SYjFr++ThfEWKjC
AaoPlnoyP/sTcxCEJLGwjr8nDTMln3HWHVLle//yyFzSV6eeIaTDZhAdMFNd5Is
gokg8DckveJwsBlZZzqWG1luuSKnzbwxGPOYzSPUrVTEeJaa7X7fTGNQhV0NHNhTR
SWKzni0hohpk1TtPSWlMxybyfJcJkK6ZZou34GE+0419jfcYYRMisU3+pgm0VJhy
sAQvO5/VdAswT6rgjS05Gq5ipj4+binjK7qpT+yPRkfQcvbPUnq+jJ8UCo99fye
cfN8JK7z1qj/hF9IkeNwGZqSr6OFmMDj9yorE+jlls0siwKbrfPdVrE5GZ7391G0
efVYKq14IkBIStxUHIRjWe2MTn1FpIhFPibStlmKPJH2purDDIESB68P4rcvn21t
SesHG15q18PECPiB3AaVJEA4dat89Rt1PH8MO8WLjWgDZ6TOEsK5cJK+EYKL7Yjx
JJO7u5QWY82oFy4ofWsWTqLdB7M09vPvjM6aeNm2noStTyf80rikW/KZpvj7UbSO
tkmV0zccSLvG49PXt5TkJ4cv/moxzggTUTBUzHzEfrQTMF3cMrOAKBew7UAQLJ6n
icV4etOsQBNGiXg0jfvmkSZ+nJ1hnzaODhno/PeQ+YYUW73jKeu/ItMnr4OqZw4U
AFRjavvXktsxwy9v+0/wgaIC4dAEgh1/i7wQFRiaJkZrYFulr6f9vebJwzudFS0F
0qNn3WIJqsXJXE+skopVmmuyaOaMhy65BtehYJ9qOUcweAfZHJl01I6b0mtXbZN1
Wm0Xu8GIW6hPA16/X9nb1Me13Ii/UOtI9+a5UaIdSiv1T4CiWUMuQvHbkTAggyqG2
wki/+pBwjZ9Rzx5L3jB/gYx+5kGoTMdkP8ECH33Ghd0yDhhIotlfTqSdxFxSemdb
qKqen1/IOvk+0DjlyMsVw+/WnvNuYg11BcNp0jWjSE5NOavyrbs/5q+MG/QRfWZT
B245IVLCgzvoacuEIVUKt1fgxuNisAzts1/xdMwDLa7gab+B9rm4LI1o7f1Ttosp
3P4oNVFBuDgY51pQVzMobKU3OUvBs1keTKf2G9A2tLTaOdCRXD0LN3QOE/qsOcW
2VA+J+0xwmOOSGC4KCSbi7CrKcMg/FzhtdBfwyFUXX04wMzrETfDdd7vKDY9JJyc
dfjxfi3gOWpzcTimLXa6bqjttcre7zOdu+fw033+Vc3iF6dRksSMvVJNfPnTOLgM
a+NplsG/fVIK1q7Z/vOcH8roepjboBd+isHKw09v8IWRx1gd8cBjFkieixnBlleY
gwnlof+ZV4Way+5CG6hZw9mBvNXPPud8QZWu5K3cNSb9QklDF+ZvkJ+ACEAQDpEK
xqINQv1oKDt5dGPSCXwD1cCljVOadk2cQ6hE4OD1IGA1DdPXXK9Rnx/BQAI2K1P0d
BjFTc6OF11LhCATKqdQiWotLtPN8P7910L6dqSXTJVoJY5sPxY6aQts77PKggSQq
2AtliQ7HmdnqHeZqAXKBM4bbHRr7PDiwgWw7t/ypE6gQ+M8p3CxZymaWFjvix+5Z
ZqnTv7pRvpE3nmvBtOMUyPjGa2AJEE5nH/wN2vqBbRcZ6ZdFvC7zsFv5mpQym+1f
dYZcgQ1KSCt4RjO6p/8R3pZFMwirr6hihe8YlHwTiv9FskZd6a1yiROyGwGL1+x0

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] 9795 bytes
  (decrypts to)
application/pkcs7-mime [smime.p7m] 6238 bytes
  (unwraps to)
multipart/mixed 1938 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
```

MIiCpAYJKoZIhvcNAQcDoIIcLTCCCHcCAQAXggMQMIIBhAIBADBsMFUxD TALBgNVBAoTBELFVEYxETAPBgNVBAsTCExBTVBTIFdHMTewLwYDVQQDEYhTYWlwBgUGTEFN UFMgU1NBIEIn1cnRpZmljYXRpb24gQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIeh7j00 Boq0MA0GCSqGSIB3DQEBAQUABIIBACHSAmfQHyENXaa4Q2w2DOX1R3r9EC4cvYoT j19WfwiSgCZkQs08Aai9ARBM1XNRKV0NY9ocrCo+RAAv+2xgHyBbZAvZYGa2SmRT rSqdHUtXYXiWkBRjXHer7Yi+96T6zGZ5iUyz/aeJBC+DkFgkAkVlIayyi5QH/uLz5 tjd096w8lZj2s/2UslouHw/oCs7KpleZzI9j/6MP0f+vpTelu5G1WYmumgKrF7MS 68ABr7N1V+hkMkSXo2u1CzPamQjLqHRjxJco0LFubArK1Rkn1i1GcCb9dITJh/d CPLaXptgCNHE8ZL4b40reSbA9UEKTPxA03oNph5Qt2eZLdvwdr4wggGEAgEAMGww VTENMASGa1UEChMESUVURjERMA8Ga1UECXMITEFNUFMGv0cxMTAvBgNVBAMTKFNh bXBsZSBMQU1QUyBSU0EgQ2VydGlmaWNhdGlvbiBBdXRob3JpdHkCEzB8R0APhiY6 HGLS64MvlsDXhpQdQYJKoZIhvcNAQEBBQEGgEAAQKdTFs7ZNTe908FdkPkY3u/ BoxJYAVFN+1/NDdaSSZb3FL6Z4zdwVSIo379RUTPSLfa0jQ9vYpDJZs2D1FoFtP lQbYVf/+SQWECLRSuU2MJTr+W+xyvgVRG6pAERbPZemHtGCv02hY0JqVt6U9vwm8oZ J96jh38tirDZ0688VjIBOVowEJJXHPIf/xv5dek4EqDvqQ9SAg2f6YvL8Ipb5t1S xN7dEQyAk38FU19ubnjOpHuPsSYQ16TmwdJ6tkL2Jf5X+Jb0Zi7vTVqA8CvEWQW1

ekLv0xrFrOgAxpY+mKs3etVDZJGXFNK1/aiwXcBtrmhzzxivA+yLrVGyCJvVBzCC
GQ4GCSqGSib3DQEHATAdBg1ghkgBZQMEAAQIEENrXasWjkevTOXaTor1IK6+AgHjg
P741U1HNTIuPnFwDRMNU/sFhk400LG+v/6DUQ5W6212BS61HgIRwI9sLmaUrwYCO
Yk2Fz0uvtOYr7IltVsC0LSg89JCsMh55IaZ/7CV+VVffH0mSiA+ItCAwVdUcPWSI
L/ppMzr/meOgS2KBv3qWPmtUkVVoiZHZbKO/C+MRd1EqcL9VqMAej/I/QWp1vtQa
cSmZD7g3loZ4FL/8OAVtc/baOv4b8/MKJ7MzVmVmUHaM/uHeVHMnj8oeNPzdp8fE
AQLX91/Wdi1QUYmEcFmVqQ/dTcYWVDYy8TviIupGDHYW8YU/TgIUzn2d3DNx57eZ
8USIUicJgE0uY61ItMPFuRtQuXj0s2oBbEY2ncWtVQOA/CFyPwSWMFFR071wRzLY
nOoLilOhMmzZjGBoAKXEL+Py2aoK0a4nFXzKbYMtWB8ecVYfEXcl1oAwAa01Zw1S
VoFZzt2NQ4AD45j9oCEXjuhW7X7eEJLMagHXPZPOUJ/B/HxZe4K1qmV3TmLyHRON
fzZOvMfy9Qlq3SbdLWvJJ13I6/R+GKEdJnQ1USaq2Ba jucENyfuckvJIo7Gs/FSY
AhC6w1vAuN8c+tkbiVQ3xbgkWeZ1BHcViaJ5XFGL0qe jccj5X2n3kz79UZSgGEkN
C0do6ahpLHqu3fECZv/yy8a34Kml71+70PKOv7VUt073ajLq6e6/gx1bm5rfODV2
n6yB9vT3JGzb8qjrrEhNshQp5TUjywgZcqFvJmNZ9dusXAPkHE3L29TZx51wtXmJv
DvC4ZdzsZQq9T3H8Zs9uU3bJNM5y jTk2JXIX5J1uvbwy3F0c j1XqdU3iL6dxE8dQ
eyS++mxO+yj7qXwx9ZWY+TAzcCGBaWXL3vhgK4qkiKuHl7QIWrrEmDNq1EyQ7J1
D9E1PZ5dMFCTRoYf9OIwGYRDiwhys2H5DWLOmQZwSfSYK30K9stbA3LgQ2GbAWJ1
vjTYH5yeWcAH5MA7SI+Sya7U3B4A0rz0YtHcVq8//QLb/h6Gfy70OmP5w6acuQ1Y
aNlnwKK40FB0h1jmVlBg4HL8zQKLaud5e8ObT9/KIn9rZQRcSg4wWw1TYPOOQbFK
gKBmYBbIU0z7Q10Gbg1J3iS8Gu/dCt7MZkyDwgEGKJuPLuN1omOiToDLs18jwUr
Wt15bSsXfu3hCNpiFhm+ns5XrLPtBE/fLgzfZUFNSg4vKV/s5QzZnMroJnyan468
UI83csnYxe+Gyf5YHjxLCflwhZP7PJnztOho7072tIS21EZKPqIIEh8W/m/QuMzz
ajEmEPV+NvLVPM390uS41J7paCIjoBRj3saGMf1WTZx8821b+QCZgjkvTfcAnFFO
SP7iYopxJ3x2SK41sBHmic/PBCKbJcFUG6EJqntVdPz+/ZDMCAoMYwloM7417r+O
9fuoH9z1mTR6TbT3X+e3N+8dQXRhuDeD7eJCb8gAD+c6Z/MZuAoAz1ZK5t1+RybL
hyJ8JEKtT27f+DXv02k5bwfcsmkbKecqPCjQv7ObNFTmDMS50ZTQzKC4QcOR/zfw
UG8SCmbY7mlARW5xieWIUX3qDbG5MI3M3DyR928prPrT/eFjCY2iTsp/FGThwdCF
I9HQ+910h+h50zkmm4Z4yyphWU89HOMhKiGCOye9je9D2IhoGr6885e0A/xklss6
4EovfOu4KE4FB91UXvFXTMK4tw9NS8Swi52dzhNNDkzX9jeSjcohoaUSnspO2G9G
wgEOkKcGhtggc1/O9luIMG9MaFDUAiUR8eUdF1KJVoxK17B4ew5PwzyFqXX/CMZY
lLcrC+gbe2dnYJbSGDChv1mFM3lat9A7qYfebrVgtnvZAJLsF6rXUZA0XUOX7rN6
cN2omrfsUMqQqJfACQKcMVke9MRDvPPfLR4vTbsLwbOFj60xE6Axsru3Vz1oC03a
0RSozFfRqlujIHJSTPJwMFnqR0pnvxBrHfYn00qnXBBiT5rQgJvF56I2dYCPICQ0
nZ2m8Jr4Ne8uL/NFUAIEgJrHzWrC/xdBdmk9/mQYsFLFXVqsjVzbCAHCPjGemwqH
0ks8YgiiPaC8Ij0YOYu8XI7RCTs+pdWQZNntQPmicXP3zTbim9nzHRYXVEDVYS4h
oDW/4UCbjkoSxnfn8nP2dx9vhwrtfCDubBnBc7wHCJgloFNCEUnmxKd7Tlou6BF+
NxUc7PVDq26Rv3rMepQ67YbjU94zp6cvlGAEBgXa2c+1q6m4BK09zGpUCfmRpVzz
GY3bzBrjoswbOY1N19dLIWcqHD9A9RCMVikpezwgyCHXvklMwsWNfBGdDDM6ZJkw
v1f0MW0GMMsQeQ8aTtTy1UZrAibv5/uo8GJtKqDAn/D2c4IaLPjkkE01xPlvaj4y
dm+VEp5/PaidiKfSyLiBO3xw7OVH6/V80lRCD5cd2C5zEVwWNzYC0ipjUhP4NaWV
QKx37Zn32bR3NUqyV2tyAFGwksFMi7+xjXBd714NMGGExp744j7cWwlihetUaTOG
QwVq1AFK7q2QVefSZ1bpEPyZAz5fLuwuZ1QgONbJhQyBLCKgobgrTVUYPGpRiBhh
xMTfD6NtwWLA9ZGyzdNgbId2QJmaXwBZTwNNRClnBcCJkBj1MYgOGT7xtozAq1W/
3SqfTTjNwFwXdeEe+tgga2aL/BvJPnAzQzxptuSKhM0sLvi70ripJ5ODwSIiYRS6r
me4vE85xUBBOXZ6EPo4YB0GoJsGG9kvDp1tUYhcPmHyFkEDgGjxCDPzIWAyPtNli
3L3x6jDRKJ/AqPqC4laOaA6t/qbUmEXEGkcSAPS2BdJ1Fduk5ae1nkUZ4obvGZfB
zxraCTXp0SOZ2IdDVgzgLK0aEza1VEiHyw8ikekAxoNcRqZNTpugCushSbdHFW6g
OpaKf9fNWM2MdoYDDuUFqR+tFhvdgAqCpnHXG/AARLUw85OQoY920VnRFNqeBxBx

osxBQWkhj2Msw7DgU1XhIXD5djfY5UQhpov/uxLMf4ti13LGC+xdjiX6Mh1ZeMNa
gneEzmb3YCROwldNCr3spnULVqQIqubaSp4DO4WPN7GiWr04gwncQOPy/HEu3t1V
+EzO6TIG3BB3EMU6dIo1L7tq69pTQeOTQm3Gf7itpp7MSIjpAc9y4kmtXbD2r0qJ
A5As6tiIchm8qHnP0iyCTv2TA+zKXMu/YDPj8KJJ2PWRoz9KTn2Cn8OMDcINnZle
LMNOs3THxzRMvGsv2E5Z0+0SWZtz23SJQp+aaKdXbE84z6rLNnaRSwHdkwFQtLs8
7P4kLJzaF1YxaC66+/Z6LR0WUg9wrx2lptK+0o3gvSFSkK/mFWlrit2z67ebaHww
KOk9XfI2683nxXFyCEBMBzuNVDS3aAb7biO61wkMD/1RqQxApZ8x+WSfXulPZ0Z2
ehc560tuM6c+ZZRwNUBCy02cALsEOVcQGbkQgiOwY0ubHqGBLyYLWyQTuX/TLmbe
OXvWNbwuQbAXxRcDL920Aqgj4qBjRqT2J7DQlZnkw0jyxjde1hfd0er+X/X5s4M/
PGapcQWQp3xacBld4K5STV6XLynqFgxe+cI+Gfye683wNWZtPRkonoFv9VnOKa08
Q9K0E9Tv0WzDXH1B5nwFMW4ld0j6JiWi22M/dSUwpLcbHml6XhfUgLG9rPYPElRa
7tRxq/6MfSwOdy435zsxkUx6eB2Yt9rEcFWEmZjNTBIZ8Efa77cquLJzFv/oFfCB
jHpTrVr5a2uDiv0migp7upYC65rmMlAhcAJioFfb575g2P6t+q+fMeLOX3S8H6wVo
fsTXpKwhFiWYp+MGUH83pqvYqngfNBDd55ITQveLl54h6EVuFuGuiC0oGC019EHS
jHrzf76ruy2EnoJmXqljXsGMUXHXLsxvo8XZM7Qr0bXEj5gt1bWUaaV7hIkCMTis
pNVz5ZzsklqxFO0+cTcPlrh9X7RLjMHkDgWR6k+mbOLexwmXUlKrlr2oGVFrArEW
ACcAa3z40TIw3oAdRpmvY0TH1InKc70fVaJW7SQU9qXXGH6iPSfjTVj8xjuPkyY8
VobraL10ekLXrHOEx3o2ylYfLhS8sNuyE015lKuXbfucUFU7aELaa0FYQv1k75ma
0Cb7+pJvZDXoGaGdjovnjktD140GqutOqBlf1Q7VAabgbI71vzJbIzmlVKJfW7ii
L++lwuHQG2IYUsxTG9P2LM5LqIvD8uQyH4duCSKEY27JD3nR6fkayv7+EG68N2TF
OdCtm3GEa1+HYCQGww2K7TROeY9B1GepsI0MraAUEwcJcmOJoRv117j1FNolgyxo
Zr7EIAH84gKLPgK7j5WBGVvpAIGtztid4j/7MYnNa4aOAgmtYLnMtnp+iQflkJXP
71SBol2nqMq52JEIuMW2Xfw075FcrFhTh/82U7jlojFsCsaiRvB0CWlt6d906nFM
e8dTMk7rApCgdj2CTQjA8KlQw53qo82XyZeI8X+UZdklpQzWIOrz0IR9XC7oWtv4
/D0VtBpwp2m3TswH+iX3Z/wUBiq+Oca2zCf2AEUNOQlb4gUPl/+WmdYt0OyhrAlT
jvHV12pvycM59MEQfQzaFFqPKOeQON7NSrq3Q4T4p8Uk0ltAaA+K9GDOYthBQrKZ
IyqN+t2nuaCONQB0yZdPaZsKUQtDZlUVNe+1C7PM7hG63oKz/5QXoVAWB5jrXZ+f
bhO9XP/wf3KD4ANaACVcGteJsECi8a9zCQU4Hwm184bc61jLXAAUAI2/RqF2FYR0
ywq7PTI3LNH47WCimxjaCdULyBIBYhOgTQdeQ55W2lqTLUsNSwoOog6C0Ng/FfRY
DgJb62fflG8NrQCildGgJKi3SGafe+4+2dheCyIS1TO+3OBbkj2wQxgvzht9Fmae
MZdf2vJg9i35pieoEIQ5QHBNR4W4yoZuBv8GtnAuKYcPHAnSzJGla+omMCbaCcu
gupCvuY9P+mR7ML8/vH9VaTW7u6M95PEcj9QiiRVZdVDUmBGipWr4oxMkfK+sCdv
TqEmJ4HgZlOj5Z8HQRl8XZ/HwG356bs8e2tZF68IBFWDEFcZP4BZ3qV22kbo1fyO
8E6hQqsnfJMXCymYkQIwEWodj5mkAYErffjieuVJ2HgKWCuV/KKsbE7DkT9hHkjdY
Hii8rmuAkWik6QA+1QnpK8x+oLYiiICBYpEUyCB1ryWAbY02WBNj4Yw1+do+AGgU
whJj/yPGISuNTUHL0Gd3AbFplsgj1HKua8+7XLy5UDrRHXoQBzBuJN70nbmRYXP
sVOWNOSNnkLSxNDwsO1T0X6BmYv6qDg0u/hq0s1Bmn2aKW5JBr17MQqfPZ8pKDhN1
ZBrILg6Fu2ThJyQUjWLVdmsNEaLz1Gi8A4om0Vww8qhkPN4ar+B5tbJwakdYne03
10WDrZI+w7cNLMUB5u+BqtHm8UNsQF6mY1YLnGcmr8l4hv86yB91RwPcJUK5ua+w
+JjE2DWb/zG/feWM9rgIyGz5TSmfzfyeWUfw4FV70n8EsKzTPGZpBxVK8Qp/S3uK
NhSXczrlgmeHdF3lip1QaX61GV1s/IkoepnPLxzHA1oXQY8FUgT8Ib5+1GKFNBz8
bam2Fd1Lrm2Y7m9qd0oAjM5QII5vMpraulzDfxAZugVh2G0DC6cqBxdqtuzUswUj
gzb6y3WZRCr9MZpsRTpe43HHm5t0U6JEpqxxjFwK3hDRCSckRdqt6I3MnA0EnyX
11ByNr2o6cCaw2yK/sVz9GouxBMFdy4599ES1uKYFvSMA/8nJr0IXNa13mLz+am
cvYemKIitkg/7aa/cVGKHWravWxN/kTdhb7cJ9Fu05TZYXZvIKFNt6qaUzyE9XwK
WFwx1Alk4s0CjFdAu17vRR4wW8V1caD1GR8DZdAjFw7gu2+x+J2XW02Z6z7ulv0T
Nf3byz3gU7pdXCvpF1Dkck8LpxpGmUPycwkwedrGedJ7HwwITBitheixm32+txLo
07TBiVKt2+NjswiToqMiLaymqDjmj4EGYFWpRwXQmkZm8qVAW5Y7jGogTibD5a5u

uvdJQzGupuFcVphJUO5XUulnuWc/qN4lym+UJsd0qZuqU2QhfirT6lYSQg/ELX82
d9ekPyX5qS73C3qb1zgagY8FssaWdW60mmUsCmetOg8osqWFyVRb4KQxTbVT2U+9
8kX08I3w/0Pjclz75I8kpbS/JSGMsUKChvGDTOf1nbBKUSA2ZkxPx5gujoXGxRl8
UIefG+ACT2MdBBWjsFMZF/b/SPieVb7dnVOP6bdQYt3bn3OKxA4GGPvWmZhOUk3A
8UV240yhdvgUFSCvfWjD+N/4JmHjOvx5Jniw2qi2sxxIA8Q6s872ktESgGG7eWh7
+okS+UHTTreV7auJBHgMGSNue79Wa7fJiVZXeVdVQJjAJXyFsTlID6alM++9yOM4
kO8o5juEMt6Gy2100Je5oupYuFj7zCmN9lnWQgSIqNlr4igs1W99S22KXZN9OWpM
M1+J7aG1b0BKsXA1KIFMY+iCcOpUHBDeTIIUR7wQ8bDQdjwa97/iw0LEJT7yuuO
1G0tAbZOB1RrJmoae+2Uz2bcilZGHTqVp+WhjNxXtBoCIxGCsP5YA40IEfdgf9qq
RKfQBVt6gBEg2PsR1SLCiJrETK41FHWvLHa+sxIVRbbkjQGvBiFY5PTF7m54DtOC
3RIw0yso7Kx66fP9kBG1QUKM5MQmedw6/xju3f6IZdHFAmThBI/s7bZgUqIqYHXN
Q56Rmulqiff3H6IfGWVYQKvfEhKzW0W+mrr1li2DxYQL5PZcqaTfJMNrvS2OMwYh
SFUJ92V9bGw+NyJtAfoHpyO1DAXv9tGU6odlQsECCTY48Avs3F//cjuWnxgul+Zl
7PSn1Vtpa+EbMW05NHAnQkrPvaungyDsFja/bF+0iSSvGGSRarzXglH/TUbr708y
NrK6GGwJXnV801VelBXEbClpks0VbMCxtOY/VhCOq5iGtD2U1wmz30A/uXtCioBq
UmCEX21E+DeAV1cGLX4881Wx/W96qNEvYMBKAND+k7MYJQeKcVOBA7i0T9WYQ6Gg
MEiQiFp9Fqep405VwLnvU+jlJCX79gKOr0IqXMu1LoVn0LvZusZhluUZg+LdcZml
Vs89SAuTz4EdRu7K/hxugECIPzizw3DGn5xnuMSdkGNoLHLtTZlefXqiG+0Ru066
DA4cIoKYOCELWFnCIjXIuVc9PuiOljCmMP1NzTK34bzJFx10qa6fwqn8dpqYyDaQ
viHoR9fcmuVWtHzinc0W2DrCkzbENviMZaxCdQwGCfo4vVNPRLwrnk920tcWrh8
WBcWstpRe5y7V20GnCNfPARPAFfXhkoU6SgyDds16t0aBlPoNf6/KLJ5e7fovnWuL
vdqBzPlMECtWuEJaqr4B4zqrb1txNcNhR1f4laQxTlyPp2sP0CStOSIly+9zCSwF
4lyIz70JLeT+0x4DmcVMkkdu0iwuVBzhx0cjjfcanELiT+f+ET9Gfac8MaEjTi9f
IGmyra408a7ZnEcJgqY+H+uNW9AGneSqVyuFnV2C18at6JfCckHVbIsMOAkRTwo
a+llodcyhIjnFQRaWf4y5Z+T3mWwQ6j6Gkbr6Qkqxq7L8AGEcXhjaLLHpUlrDz1A
5vuMSkorZhgxV10vicWzcqNqffXE5ojf8GDhoYEAN4JHONI7uB2EMkON+XTp3OZ
uYCDqzWj/3dEuaYpq1m9HBLp9TawR2gMRADCPNZVp1cmjWbcQNR1Z7JqeGZKyyl6
cYvcyKsR+g00/sQ/z/t8rzgP50+n8GtqYKQoS6RM1NXTu5qrE4wmOr2nIWM9q2bh
2H5W1UUEd0fExbIVYKIIuhWb7N7VAMwm5K4+fIfIVVCQJegNmvs/FUe4MbVxV6yQ
XOEyajiHmv3amKe98fWQIqtIke71zvHCNbbSVomRnBZfMdP9jkg/vNPuGRXDOL47
liRD1XX0jP83F/UDsyiGHyy9HRortlhJfn0UOhdDaEszsTpxJjw80bRc5X8gmuiT
QW8DtD5P5IjBDuctN9wC+BL0EuoLT72eyUxrtoqLjm45QpBqDA8c33I+5A4hwZs4
BnEigRXIv/I2gfyagiRyAQlZTJrkB+T+DVsmxdfAqZxf5pGfYLE45Bid+vjb0DF4
BSPUw8ILhQjw+LmtgtMia4i0IZgYHGGRU1EoLXF2jLadBqU+FRA8f0f3CCnCHsAm
xQ64ultaz/jen8ESHvx19c0NDmGczJINqX4zWNX+loRENaU1fISuRGQ5jF1+SWmC
8lixgMcuCivGcuTnZHvWkSmcSpis9pa06pBZv7RieLBqlAch58Mqur9P7zXTdNnO
mNmK7k/ucs94XpGQiXImMric05OgVg/3kxthe/D3F+fHz4LZPLtqdIhkJdCnNp/e
UbFF9A+6bvCyvMzEXZi0aI8fY8BekHIrr9QJ4meKPh2IpYFTLo9/EBGHOqs/VvA3
3RQSFHsIYe7r2+h6JDgTwtB0zG8B63mNY6rxgWR3q5k=

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] 10425 bytes
  (decrypts to)
application/pkcs7-mime [smime.p7m] 6704 bytes
  (unwraps to)
multipart/mixed 2273 bytes
  multipart/alternative 1449 bytes
    text/plain 493 bytes
    text/html 645 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: <acced3c9-111b-5a4f-bd80-34558da32b4d@lhp.example>
From: Alice <alice@smime.example>
To: Bob <bob@smime.example>
Date: Sat, 20 Feb 2021 12:19:02 -0500

```

```

MIIE DAYJKoZIhvcNAQcDoIIId/TCCHfkCAQAxggMQMIIBhAIBADBsMFUxDTALBgNV
BAoTBElFVEYxETAPBgNVBAsTCExBTvBTIFdHMTewLwYDVQQDEyhTYW1wbGUgTEFN
UFMgU1NBIEEN1cnRpb24gQXV0aG9yaXR5AhMPLSW9ETmXSs5CVIeh7j00
Boq0MA0GCSqGSIB3DQEBAQUABIIBAIz93W2Y+UAs5hfJv0FVshsVqpt+3nDEwVwW
1CbA0ElWkeDUJEA3temKIObvlca8GuinuFRfBNobC6Qh74dtjjDjD2Vy3mi+VJ13
ERB/OM1wdrMRtdJrwTwV7zPC0rfHhpenbvNqpKsdszIVitiiHeG7dG2oRrJ6Jyfq
ceU313EXthbLhXNRBA17tWRd4DtpBH9Wk+3M9v7tGQLOFW5sLczK+Btqgmed+/ns
mQNf1/8T+aA5ttkzwHYJJ/Fj6GMxWaKWLpGkGtE1V00ED2NpDHLNwciWMG3MgUC
tT2aF5yASW93vBhV3Wg/gdw1p5zTF6RXI7/Z0tSE5PjLpyqYrWkwggGEAgEAMGww
VTENMASGA1UEChMESUVURjERMA8GA1UECXMITEFNUFUMgV0cxMTAvBgNVBAMTKFZh
bXBsZSBMQUU1QUYBSU0EgQ2VydgG1maWNhdGlvbiBBdXRob3JpdHkCEzB8R0APhiY6
HGLS64MvlsDXhpQwDQYJKoZIhvcNAQEBBQAEggEAbcc4GVOP/kh6RDuyhTj+6a j4
Vw/bzLNJYkgln3XsWd53MeP IzQQ2m4/w2qMxeOFjCV0j0XA1FpKJH9XFUjeKKC9p
WYYOLCu7zYgXD+9rsxA3EzG3EM0S6x1FI81273MZ1mUNDFKWhl3e+PYyharTwa0N
aRe+ZdxDT6nJfKE0Rj5AJIzk+mqZkyfJqmWINhNB1QZmUdJIBUJ2Fj2TjW089fGf
RCyTlW2TeC9L4D4g77ZZopfPLE5mUYzJds+pg4gvdujbdGWcj+L9r5MfoVVjy0hL
AvgZUbgPhyFy9wovvXxgsjLrVG91D2yy0djtLJ30rIvG4QUdoOmGI3FwTWafzzCC
Gt4GCSqGSIB3DQEHAAdBg1ghkgBZQMEAEIEEJiDLZUQgf3R+buYZMt1K/WAghqw
ghkHRdlK8epExfgdOk0WgxpB5DMVZGINLxdkqNItRYZOEfZCL1+hs9S7J1fRggC6
rGyhDGGxDGKzg4ACTv+WoGH/Ghz47DLQggop/TbkOwr4aAS9HTfrHWuOAXdID1XS
4C18yoQWJyWYmgEYqMT0q6A5ScnYnYitQoTzVgSm37/vkKZ35Q5PhkTwG5QtbbQln
56/oyWYYjBldnSGtzAR6DLH3COiPS/b6mDJSHWhvqWlu5IjyMPqegZvIuNLA7Q+d
StCCdpDF1pJx8B7knuGIuevf2vfXHYHOontYmXH9WxV+UFBY3k3GmiCnpCdWszGv
6Fn1FOZMbUy2rk7k2zHTwluUdy5HbiQ+VrbBPI03WgGWA915B5oDeB8NZ0GdhXqS
FxmjlPnkrDsxNTTv8+vHjLyIlGnstSDixjwvpJkcn8LUf6bfAl1Pv5ChAKhIECLF

```

yJJsav2uwXI7by0noyTd0x6/Bzut35DBhHxuiwfPp8QSE/bgHg+vT0nLZKTIuEkD
66kpORIEciQxSfh7rhSrFYksA4wF64TjQBONWHf18pFTRpqTLUaI0K9F+ib2abp4
o8gk59yymY57ABKz6ZqE/6bn8cEcsREbpKaHkx1r3568Fy7ErBelDgs3DAn5DHoF
FHCRGtUzpxzRXEd5efHf7NrFnN/qyNVxdzcRqqats71vjjOQRZtUJHKAYpDFF6mU
ePcxW+iGTACvgpkvm2ZnTjID4li5Q57gmns1rywGpIE8BJePAfcH8+ccjyEhPGUD
XS/DPT+w+bs9GV0nZFrKGMpLs7iheaAR9twp7EF47wPgSNVfLZWq2fdkUcBxrwj8
cpbEI0leNwurQZGKz99aoaoMdYs6TUAxtI3/P9+Gu4M6DHjsnPeRrv/A4K79VNVp
z4NdM/vy+fBpS0Ef3kIdFc7gv7CkqFr05FR2i7MP69MDsUvwulb10jmeY371EVhd
NWHL6gBbuAgP05qjk0fk7ZsS6pHK1wocgRpeJtC9PPc/GtrB/hXCfgBLSL/xVCwu
z9MSrgRsSHDI f8H51fJJKgKt/5DmghLE9U8lQPTTZe6pLF0i9k0mgxVS1aWXogvxi
fm7yz7bHYLgeQR774mTP85h/ei3brsA92JJRCe162EBXExD8nBpUPDVXlvid6i5G
dwKDJdwfTB1Gixfp4SRoU+QzclKyuKuJ40YoN0OPsOPmRHeL6r+A9QbCxx/+gv4c
cgmzX17pczTW4MMo/TkhKstD75VySPweHSDiJ2ETsCbF9/OCyeGzOEIN8csUSFYx
PanRCmvP2E2ER58hngQOJcTack5qTl8hs+Vw+C9LBQ0noQfm7e9i1HMKaFEQoPWX
JwHRkmPaz0FiEQTjhI7nZTWfpxa2sslnkERQ91SgsT1cYJ0xQ3GeXWlPp5t7yfbQ
UZyYkbHvqC4MczW5rmSdPqxJiFkZF2uX8+OwOdF8DIwT8AKAp4MS+/Lg2iuYeq/Q
YVOMdXSaQkBXsLiBIXYEypSNPIfc6+24NgxTMzEpyJw1CNb5iV4Va8erDaYhr80
frClSk+xKC3nGNw8cnz5D1LFLz6px28dB7C7dSvYNU2YAR3xtDphOP2zZVydudI+
Wa7FGFpWJHHfGPAtnBnebwcR5hffICVV2ATiNwHJG6I8I1W6b2Uzo8V0v7sDj4EW
9pcuh96H27VG54UoM9xZcdkMq8q1mH02nD602xc9MIacOGOLV/pQP56MvFYfr1Zf
ysOUBaJW9B0lWsVEOyP4IkEovt1KaTvA2v+HMqlk2ok1EA2cYk8tcpnJg/fHm3K
LmTxldf0/2bfyBWLNYkcIsCxOjAoB7uoisPMNPCLRwniXfZgrspDt3yE4vzbXYMQ
49I/BYdxddJH1lfrDGqMyZ9OYS6aWoFQuSY1p3mI9IcrJul54SxeaALNaDnEX1h7
wfBejR4aGfD9AjuoVXyVdLI39difUDggMITPbbqO4eFeZph6D9sdTyG5Li2k+WxL
FQ+iVwj6/teLAIlyPgks13kbEP6CLICmQ8wpleF888YG4BVC4H1BjgzVTWj58Whe
E9zJ5xPhewf2p8a4PPs/75+3GNriNDjwZM1jtERMn1t96UfO2DCKiC5RYs2J7s9t
Cdguovv4PsN1pi0bD96Toe6yLDDDeBa7Y50FRBpN/YW83HS7HInPCiFjFozxc//JR
rvtYVjXzHvWWl8mdua+0k41E8WVCk/pn6cAlg/+HnkNh7UTB+QYdFHVr4HSI9HB
DPZ0H1zvGKG7jpx8AqgxLDItBq7JI7Kyo86fPQ7G15vjSZD53prvZz0tToR3j0dq
uZ/oH4IWqW4GiwsK4fUweHfOB5qDqaQTdm4jz2Dv4JJNS+C0QJxh2Nb0sZXV42NR
ITZIST8tS3MdUnxgH6KXV+AnvFWh0Tn51Dq1JVbWGRTGjw6qjnFNomU557ygDmMr
CEYBxPcj37jYbd0D7mCHD6L6ztrDXxWsJms0X2oILzTJFI fVc8+5gPbzYgbn4CxD
ZffnrHA89rZiKcWn8fhqHPjeeADP65ywVfTqGltw5Vvt78+aBKCKXOLPZGzQ7R26
zxUVtxIG51AK70JNN6EqWTtk+Imd/EIUjIVZN9TbQBxATUaert5x5dziaA0lf9eVG
LmC3mkW9uQCzBxtwacxYUdw/VaV5VsbkTA61md+7B/FKvzmIq4F6sKSTiYuEFfOO
paFQf6FaftiIgqigpDu7ogNeR4YLL4ZQWZj+xTeY01Kx4OxPxCAfMZZ+sVoXGCnG
qCvuppwQ1X+rZItB3YErkf2nflk6J/XrSjVfr3mcmvw2QBsAmxK4Na4hLl03mRT0
JgbbmKXqN1TfZT2qWhjDQNmC7mo9hKUOmnm4Dy1RNE1Q1XMG//0G3oqKjFX21Gizy
wv07CYHuTuBBsyyXXj1ZARsGuzGbOkX5EaBri42M+VVGtqG8g5uPbfY/8i+/BuDs
08u1YlPBrvyzJOE7YbHOBdsJOGi546DSO57Bexrmfsgs+yoPEpfm1DqAgppm00++Z
agWFbj5JtLRMv00vFYMBgQU1FkZLmcvNA2tAVNUwC5xbFcEXxg7/4xQGXTA/B2/A
Oo/2kDsQ3o9Gwfx+OmYE8Sb6rEyIiVBymM5AzlEpF0lVFMGLzmRN85cwlDtBnKGF
b/vP7caw7LBoJCHUSasGmY2Mg2k+jmfybs573h3x9XAtGfbAN9YAR9qtRmkj8Wvr
4VpOSKGSV5zyfxWwdgNESHhH0HKFnj5hHKgcjgOmUWSEMx7+qDiF83uZ22xpyZnt
Wckbw7AcxznIPON7HJMKasZ2Oy+n1WgQw//n1prdn1Fv1YSu5j5HUp0wp9D544s2t
I/b9D/TAVEbBK8+m4mCf0PvqG1zuxrjt4A1pAwG9zVtNebdYw1YvxTwVvxdem7Bi
qzXc2YsmHLEkCkRsyqOgjr3k4IN3vMWICv7YQBX58NzvqeIA35hlCUc/wM41bH5O
EaCuWzAIiKNeGJ9tTvcPb0WAM0crq+G3CQwZyCxPQkCmKZWnyweO6yxpMzfH80aq

DBJIsYKOhYl+YZLu4i196BZY3wZ4Jh4rrHHA07NpsoS5ZLTh9+5OE6WkLR4sc0Kn
41fQZdFq7Lh1i2fD5A014zin+/1FY6FQ0iiBFBYOYhPB9WMbaO/T6HRsKfHS/2xU
G9cS2xdrLP1MNxv72PEY5EDMegsd0owKk6HpwmfNNU8iOg0AougZ1hmm+R5OBEu7
nJlccBRPusfZ4U2pG5MBwuo9ZQ/CetLMPtm/glixKoq+esl/ENTXoT7amGSA5nAN
ivxH/kKsGHNe2oh9QVaXeYtVwEknn/fPYcuOu1R1jqfnqqLS5pdTSSTOByJjImyA
/KbmzyEOgZDXLup3pAC6PXYaV0Y8FNdqs2eAg+jQZZ67foGYQeXbZ07t4W5LH/qB
Zt+78EN01NVBoHHdt1EdAcs57bzviVdbJw7GtjccwJhLEdJTxSDOOrTtI+wGJINp
KjNDBnBRE1KyE7Us/ev5yRQWWghoi+17Mias6eTXucKMNGz6mS9aNruTMDcQomjN
pdyfeN0mYcYSwU5RcrxSoZrWo88soKJ+vwsC+kQj2CWW15a1OmdZKPQMcfuS5XTi
SdXjZwckT9CcwoB4ElKxuni2mjLPODKwByYF1DV2fckV4P9oJRSD9400ZiwqI5KG
a3yCtmNGW2AJKVRWuW9uXgNR/ouMGwxKbHhJZeBJAebZspgxC2OAq4aZDQh3BQt
b9vfySSPFRn6nu3z6qWfvMAMjQfcyrydW+NphJEISLjm58kKM5NAW55bo4Zo7we4
eesbotmSaAVhpdDz+JbubsqGm0QhADLBP8A20Uj42jbgirqJ91AuuQik9uJDUjXy
gPPVSYH//iL6iyP6/hlk+Efcet77i34Zr36mn0rKKdtWzmi0JHn1z7zzLhG8Duez
pqxYAUqFtkktrLOFjt4863P0U9i/aWM+TcLZXdYhTM+dZLZViUUCACsHkhSs3i4h
9R6weSgV9WMOKn7ZhCa1WnnRuIFuN4+wzZtJlXklm2T9Zq+1lBB9vQmJXquctdKG
YlqqNQwGs3y5lcs13FylU6H9iSDz2eXtyr2srHniRNC3XdQ/CQQ7csM60nKvPRSl
agdyDj6ZWe6gdV1vrZuXyQHEoVYPCSXibqYTl6PsUHfwPfk16ZIJfkbT+gUWgfjP
MxsBsRJoW4nA8hJjsPaYXWj/+yyA+MELghFfCt1TVTOT7D4p//B8zUpVpFGirxJC
LtuB7P82/o9gn1EgLBeyPqC8uU/2L5gljkUr18zxTDnuu8uN5T+Dm7t9KhOzz4Nr
MlamGKrFCZpDlkFQHRuZuCd003fja+Z/TxbKkHT6tAS8KBpA4hkg6R/XZiSX2aX
dzf+8snt0yRyHvIKLsLuVl/oz7TJm0E6WhBxnAaXQRBYL1Qf0Qw586/TxJMbGmKG
bOodHUqyyodGrLhdzO+aZIbcceXR8tVF2pHvWUo1lKjld+RrzHP2wYqckinh2ie
sKzou6qmtfojHrZxv/hooe5UxuRQB5LoBY5tEPklx5CI/8MFtZg3Eb+uU3q+/TJG
2KAUnkqJsyNiLoggMcKasWkbbLm3g7nyq3eewRSdGinxwicCXqiC9zX9A1Fp4jHN
rLv0QtM0tbKjJbk/ttHqAaFC4/+CQ0YnWeNxxqzTrF/JCCnr1v1/grN14ei8wizb2
Uby05vA5hUgbgWDUPGvr+2tj118Q2Y+XGz13b76ype0TPFk9g3d2S1NUIcakiup2
e0PhHCXsVPIjxih+XiYUIeh0oxyWKAknsZPhausQZ7R1ArI2GBdRGFJBQO4rhIlC
2Bn1NUXF7IwH4Siza4mJvt+psd84SluVVBD1JYvFKJxrCQacY8OWNPZXqhy0aY0u
IWnCDYHWuLOK17RUfDaAHaiNwZ3LpppCxjtRl9s1P4u17b1LPk030nu9k6qj109
YYBSPq6wSqYvJ2vWYdebU0rLHm0R0MqZHKSScZfB+gypWgXi4dIKy5lS+DWKQQmr
lS+pxCx/Gab/yNjGIAMklWHr1EB+8xc3Tt1BeCBj4YdJgxdFzvOg9jnDr9J1CLGu
K9OkCAuzAqltaX6ot2KCWkzKNimmnd4i9p6pADukioRRBmftZf+cjk4LXoPGNPvp
cXl0aXeJz1t+9SII2eoQQhDZzyromghHXKdi9Oq1WV4kDAZG8cNVWTiLs462AHum
zSuI4VklWN1v1F7w1a1SX9/I5hL5pq2ldmyUCCANp7TWLr51ACPc11PEk5JdmAHA
nrMW7wGgf6Tr8i8LX0s8jGljaRdDVHmIfKpBhitfBuYcm+S3NN3uin1ZnyKhE8RU
KfiW4ZoQCEXOhc01Yjiiqq7VUB9g4kEKVt7y3LkuP3d3VkGndYDEwrCtyUOoXymx9
hpR/33z96eYICVKPxYEsCGAv802RcvviU48ZfDvKxDv0AdJLGu0BXANZKuJs0SAS
3cm/vIMqBgjGZ18Je+d5yEXUqrv1IiWFJtA6rdYdbvg6zTUIdpTriWV6e6KjgvMh
xCe2RPJn+vyEtmwwLMhs8pL1zqMAIS/cTxrQ73wyOgpI+ilkqUgfvfsJqrrHoQKt
agXmgsbOb6a2XTLqmymLmkcJyWpwwuRZGBMwFPYtIpwzcxISrn/AR7m4fXoHz13E
UFCwKWYXV7PvcNgGlzdlqBesPNM0tiRAhNclp+Vs18WLYaJfGCfDcc1WoH6ApGmM
HFaDEB51+UTNr1+lgemzs7E6Bq4vYFwIWQhCrSsv7UmQv554YOzqsqib+mGYa706
w6NZxLI3KniU4GZoTHT0Z+3HnhBJr9zMV0LQ88XAmqbJFiEGV20Bn62qu9SECOI9
UGYObQt1ZbsMAdi3GXbPIRl1j4nyQgxxaDrzw2RSkr0t7lPe7GrjuWJPp0Chz4jqX
FaAjxn3rWhqlDaSKe0kCsIh5bJ5dzfAfuTPYqNID3chp3SKn3PbfeIS8qf98NDT9
sWFrAeQtweFREuKymaYDZo319W4KRZpS8eYJfJ5Li2bPn6j67i5kTwCm6C4qzWNB
xncykXYsTZyc2+3Jy/0GKuG3twqA+1Nehq9cq7vAbKNKM+GvZ2LP2rcK49oVsMJc

Tp+iIZIoqC+2Ak8ZlEoV56oCzkVSFzJMmcN5PRUIeG4i69CdPTN1l4p/OxhuM1e6
EaIoR4Vr2CdnFQS2ftv8Mukp1+aT8YT+6RiVeJWr4/G30fby7uuQUqS/Hr7eSAC8
NxuVQooLc1y8dXtkpIrRMzojukX/1x2MALkq8w5V5v/Qw/Nz2PYa3UOAZ+p65ikY
O/T7PHpfTjNt1D7m/3WQvrmHa85P9Z9ehvmT/H27WrEwN/eomv+Ozk5+1XuPlZmX
PjnOBFIdbS0fFSu3zUJit0/uMx3vzDWVvQkh5L2xBcknrNt4yBz8Jt/FRgSmEebU
caMYI3iMLz4nxjNYtwf/BOJF+4smrptS5LGm10IQdnxEgda8gfmRyJGFUYGzFj/o
lctglZ7myHIAb5SaCEt2JlVn/D+tEyv4p5aRnEuFN4rHfVeoZrunN27voartSRCv
fxA4GSH+kzSRhpEH1UUY7kWuHJaLAKZQDdGtTEv3yqhKJZOU6FQAFJMjrwacUYgq
mErRvluDN1lB4CCPpzaC7FHM7jIY+5pqYXU5wW2Wic9bhCuzbWPmGu/JIFpS1PWj
xSV89+maQe4Q2bmTuSXMPof1DXRgB0dB+kzKLIyyv42NF9K2c5IS6Mqf3rRmpUL+
7mFV5iBibV+ZYLQHKGN1ev2OQuYlTc0Zgal5xP4fNn6lCOW+T1nl7K2jjRKDWDxX
LvDexplV8no/dsm1FXub+eGlC+MKxj+v12Rb1k9W2pj8ui5X54CVfxT20l8jtgcr
l9GTMH+9CyWpN/lqnrS2LV9MFnjv1mVJ9QBCHESFVIi3SK6M9KbW4iqoPzsed3Pv
HbJ76KwLj0bcSLnLdXoqp5XQUUh50ULEZN2IYhx4FNKZABcf9Uyr3o/h7EqAejUi
MR7qBOMjegXuj3X0lnIZSK6ds5LFXyb9hTa/O7CIO86BQpk+xZpWku2oxRbAR66
f79naAQwchYfDbIzc8XqFMT6TQHuk61DsWhQtyRpLkbuONYMqSuWp3i4DcoMhsNt
SEHxUWf3qxkDO/cjGJ1QRP721TP9UFGRjh9gRplL15yNSfPeNFUQfvAGS87K5xP+
WPTEJRFIdTfzf5SHM4DA8+2eJSSm7ii8iq/bEubxwMc6mO9YfauvuOwyXuEw5En9
kNsZDnBgefWRanDIkwGQHZOjs77wm1i15Bf8ik8wpluWI4qtkeNmnHLmbCAkvv7w
PXZc2hecs0rN0Ly0xsQtXBvD5psc/V3nm9N9DteCIOBJZNQeTGpY5cukWvN378Aq
tOx0KiTEZVVFklYVSdmiJssxB/VF+8NvhjC4qtUNqRm2UIvFbGQnb272FL5tN6ow
Bpg/wd9+26GPhZ9Xf1+pgGHMPQBOUqY+jVjJGScm8CkSJ9btOVdak7JYzGHACWvK
KtTve7W+HErYj2fiZbgXbpGitaa01Vr7tqpsO6bYqxqFayDNLTLUwrU+i0CFLPwgD
JJp7LrUkUbYNPWR4UDX5eoIMvLo3SMJk3FYi8cp3mx7NYzFXbs4aq2CCZnOWFTl
ZrY9cc5RTxaP8MGMSVS3EqxK2GmEs3oC/Ww9BBG9bQ2enxeBqseA2Tx6RkhgLDzS
WegxW26LVlmtkk0e0sEc42vMXt96kTMrOKpq/sThDpiflXaMTOGiOasI/ArISJwi
Z8B2W/io964PpAcDc6Qo5AKqAhjMxqFQy3bVsAdCotMDXSYYOkHDJ8yKdssNumKov
9i2iNagzcGUx085i0jerD3FiYmCw77X0gTeJ5S3EKcm8NN+X8WuwgsHTjIwWSkeG
GhoYUB6PZY/NS7hKl2pV+ob+S5cDKI8I60buhlquEc2K5NpMrfIux6h+Rmd2AYb9
WyBRA6uNeb8JZQkQdPzNnS9RGXPY19mPU617gmV0mNT5xuVbAYbwu64AOVL6au5U
V6VauvPlDPeSQSDHCbtFfSd82zp8IRTQgo34EjzYQQhrX31KW4fkSkOIY5xvM/9
xVAW+8Svq+eYWb5ue7VHS/+n/PTdEi4kB6UPJ5gStfe716YeC+caOejkoCzvneBV
MdAoR04DsRgdmIFNn/vS0k2RbEnVLusz3ZkaOT2Zf3SztXqSa764+OjXhFZQc8K7
l4hFxmXaBgNmWT5vDQNEb86hBx3zkkhTlPZJLdJJgEJ0FiUJYThos+xCMPHtsHMz
v7qaTE0YyukXFCKvxbByalL6CjduTJjZQXcxbIJBh67l0ZRjANlRkU/WJTY1REo9
2juia+Gg56gsjt/qM0VJjR+pYktjfvCFdUWYwZu+WTLdhEVm1sgjKdaHuastGnBE
wEzTYffcbXAG/4pc8A5msfCJexqNBr4QNWLMhN2kpZCIOAJRfzci9hKd/xsI8AXa
CY+q3wTzJDEzcGrG1Vh6PRKFtnuuk7MEjAHmz0Po/Suh7PPjCUABihvcac5rnDRA
kdqDZ+jCgU/KgsmEzQdxxR8M/iAmvDrYFMKlDlC6zPw5JxYbSMh/tdrPjBdbpjQY
pQf5xCsbK0kMJs6ZAvrjg==

Appendix C. Additional information

C.1. Stored Variants of Messages with Bcc

Messages containing at least one recipient address in the Bcc header field may appear in up to three different variants:

1. The Message for the recipient addresses listed in To or Cc header fields, which must not include the Bcc header field neither for signature calculation nor for encryption.
2. The Message(s) sent to the recipient addresses in the Bcc header field, which depends on the implementation:
 - a) One Message for each recipient in the Bcc header field separately, with a Bcc header field containing only the address of the recipient it is sent to.
 - b) The same Message for each recipient in the Bcc header field with a Bcc header field containing an indication such as "Undisclosed recipients", but no addresses.
 - c) The same Message for each recipient in the Bcc header field which does not include a Bcc header field (this Message is identical to 1. / see above).
3. The Message stored in the 'Sent'-Folder of the sender, which usually contains the Bcc unchanged from the original Message, i.e., with all recipient addresses.

The most privacy preserving method of the alternatives (2a, 2b, and 2c) is to standardize 2a, as in the other cases (2b and 2c), information about hidden recipients is revealed via keys. In any case, the Message has to be cloned and adjusted depending on the recipient.

Appendix D. 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 (headers 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 (headers 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. Document Considerations

[[RFC Editor: This section is to be removed before publication]]

This draft is built from markdown source, and its development is tracked in a git repository (<https://gitlab.com/dkg/lamps-header-protection>).

You may also be interested in the latest editor's copy (<https://dkg.gitlab.io/lamps-header-protection/>).

While minor editorial suggestions and nit-picks can be made as merge requests (<https://gitlab.com/dkg/lamps-header-protection>), please direct all substantive discussion to the LAMPS mailing list (<https://www.ietf.org/mailman/listinfo/spasm>) at spasm@ietf.org.

Appendix F. Document Changelog

[[RFC Editor: This section is to be removed before publication]]

- * 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])

Appendix G. Open Issues

[[RFC Editor: This section should be empty and is to be removed before publication.]]

- * Ensure "protected header" (Ex-Memory-Hole) option is (fully) compliant with the MIME standard, in particular also [RFC2046], Section 5.1. (Multipart Media Type).
- * Decide on whether or not merge requirements from [I-D.ietf-lamps-header-protection-requirements] into this document.
- * Decide on whether or not specification for more legacy HP requirements should be added to this document.
- * Verify ability to distinguish between Messages with Header Protection as specified in this document and messages without header protection, and update receiving guidance accordingly.
- * Privacy Considerations Section 6
- * Security Considerations Section 5

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 Foundation
Oberer Graben 4
CH- CH-8400 Winterthur
Switzerland
Email: bernie.hoeneisen@pep.foundation
URI: <https://pep.foundation/>

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: 14 November 2022

H. Brockhaus, Ed.
D. von Oheimb
S. Fries
Siemens
13 May 2022

Lightweight Certificate Management Protocol (CMP) Profile
draft-ietf-lamps-lightweight-cmp-profile-12

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 special and complex use cases are supported as well, by features specified as recommended or optional.

Status of This Memo

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

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

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

This Internet-Draft will expire on 14 November 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 | 4 |
| 1.1. How to Read This Document | 4 |
| 1.2. Motivation for a Lightweight Profile of CMP | 5 |
| 1.3. Special Requirements of Industrial and IoT Scenarios | 6 |
| 1.4. Existing CMP Profiles | 7 |
| 1.5. Use of CMP in SZTP and BRSKI Environments | 7 |
| 1.6. Compatibility with Existing CMP Profiles | 8 |
| 1.7. Scope of this Document | 10 |
| 1.8. Structure of this Document | 10 |
| 1.9. Convention and Terminology | 11 |
| 2. Solution Architecture | 12 |
| 3. Generic Aspects of PKI Messages and PKI Management Operations | 14 |
| 3.1. General Description of the CMP Message Header | 15 |
| 3.2. General Description of the CMP Message Protection | 17 |
| 3.3. General Description of CMP Message ExtraCerts | 17 |
| 3.4. Generic PKI Management Operation Prerequisites | 18 |
| 3.5. Generic Validation of a PKI Message | 19 |
| 3.6. Error Handling | 21 |
| 3.6.1. Reporting Error Conditions Upstream | 21 |
| 3.6.2. Reporting Error Conditions Downstream | 22 |
| 3.6.3. Handling Error Conditions on Nested Messages Used for Batching | 22 |
| 3.6.4. PKIStatusInfo and Error Messages | 23 |
| 4. PKI Management Operations | 25 |
| 4.1. Enrolling End Entities | 27 |
| 4.1.1. Enrolling an End Entity to a New PKI | 28 |
| 4.1.2. Enrolling an End Entity to a Known PKI | 34 |
| 4.1.3. Updating a Valid Certificate | 35 |
| 4.1.4. Enrolling an End Entity Using a PKCS#10 Request | 36 |
| 4.1.5. Using MAC-Based Protection for Enrollment | 38 |
| 4.1.6. Adding Central Key Pair Generation to Enrollment | 39 |
| 4.1.6.1. Using Key Agreement Key Management Technique | 45 |
| 4.1.6.2. Using Key Transport Key Management Technique | 46 |
| 4.1.6.3. Using Password-Based Key Management Technique | 47 |
| 4.2. Revoking a Certificate | 48 |
| 4.3. Support Messages | 50 |

| | | |
|--------------------|--|-----|
| 4.3.1. | Get CA Certificates | 53 |
| 4.3.2. | Get Root CA Certificate Update | 53 |
| 4.3.3. | Get Certificate Request Template | 55 |
| 4.3.4. | CRL Update Retrieval | 57 |
| 4.4. | Handling Delayed Delivery | 59 |
| 5. | PKI Management Entity Operations | 64 |
| 5.1. | Responding to Requests | 64 |
| 5.1.1. | Responding to a Certificate Request | 65 |
| 5.1.2. | Responding to a Confirmation Message | 65 |
| 5.1.3. | Responding to a Revocation Request | 66 |
| 5.1.4. | Responding to a Support Message | 66 |
| 5.1.5. | Initiating Delayed Delivery | 66 |
| 5.2. | Forwarding Messages | 67 |
| 5.2.1. | Not Changing Protection | 69 |
| 5.2.2. | Adding Protection and Batching of Messages | 69 |
| 5.2.2.1. | Adding Protection to a Request Message | 70 |
| 5.2.2.2. | Batching Messages | 71 |
| 5.2.3. | Replacing Protection | 73 |
| 5.2.3.1. | Not Changing Proof-of-Possession | 73 |
| 5.2.3.2. | Using raVerified | 74 |
| 5.3. | Acting on Behalf of other PKI Entities | 74 |
| 5.3.1. | Requesting a Certificate | 75 |
| 5.3.2. | Revoking a Certificate | 75 |
| 6. | CMP Message Transfer Mechanisms | 76 |
| 6.1. | HTTP Transfer | 77 |
| 6.2. | CoAP Transfer | 79 |
| 6.3. | Piggybacking on Other Reliable Transfer | 81 |
| 6.4. | Offline Transfer | 81 |
| 6.4.1. | File-Based Transfer | 82 |
| 6.4.2. | Other Asynchronous Transfer Protocols | 82 |
| 7. | Conformance Requirements | 82 |
| 7.1. | PKI Management Operations | 82 |
| 7.2. | Message Transfer | 85 |
| 8. | IANA Considerations | 86 |
| 9. | Security Considerations | 88 |
| 10. | Acknowledgements | 88 |
| 11. | References | 88 |
| 11.1. | Normative References | 88 |
| 11.2. | Informative References | 90 |
| Appendix A. | Example CertReqTemplate | 92 |
| Appendix B. | History of Changes | 94 |
| Authors' Addresses | | 100 |

1. Introduction

[RFC Editor: please delete]: The labels "RFC-CMP-Updates" and "RFC-CMP-Alg" in ASN.1 Syntax need to be replaced with the RFC numbers of CMP Updates [I-D.ietf-lamps-cmp-updates] and CMP Algorithms [I-D.ietf-lamps-cmp-algorithms], when available.

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]. Especially CMP, CRMF, and CMS are very feature-rich standards, while in most application scenarios only a limited subset of the specified functionality is needed. Additionally, the standards are not always precise enough on how to interpret and implement the described concepts. Therefore, this document aims at tailoring the available options and specifying at an adequate detail how to use them to make the implementation of interoperable automated certificate management as straightforward and lightweight as possible.

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 Section 1.8 and Section 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. 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.

Though CMP is a solid and very capable protocol it is so far not used very widely. The most important reason appears to be that the protocol offers a too 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 a choice to have necessary complexity more on the EE or PKI management entity side, it has been pushed towards PKI management entities, where typically more computational resources are available and the development can be consolidated better. 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 [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.

As also 3GPP and UNISIG already put across, profiling is a way of coping with the challenges mentioned above. 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 all the general and applicable aspects of the general protocol are taken over and only the peculiarities of the target scenarios need to be dealt with specifically.

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 do not need to come up with individual profiles from scratch.

1.3. 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 needs in operational networks as well as availability requirements, especially on 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, while PKI management entities like Certification Authorities (CA) typically are not deployed on-site but in a more protected environment of a data center or trust center. Certificate management must be able to cope with such network architectures. CMP offers the required flexibility and functionality, namely self-contained messages, efficient polling, and support for asynchronous message transfer while retaining end-to-end security.

1.4. 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.

1.5. 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 different types of certificate requests to obtain a public-key certificate for a locally generated key pair. One option is using a CMP p10cr message. Such a message is of the form `ietf-ztp-types:cmp-csr` from module `ietf-ztp-csr` and offers both proof-of-possession and proof-of-identity. To allow PKI management entities to also comply with this profile, the p10cr message must be formatted by the EE as described in Section 4.1.4 of this profile, and it may be forwarded as specified in Section 5.2.

In Bootstrapping Remote Secure Key Infrastructure (BRSKI) [RFC8995] environments, BRSKI Asynchronous Enrollment BRSKI Asynchronous Enrollment [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.

1.6. 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, neither).

1.7. Scope of this Document

To minimize ambiguity and complexity through needless variety, this document specifies exhaustive requirements on 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").

When in Section 3, Section 4, and Section 5 a field of the ASN.1 syntax as defined in CMP [RFC4210] and [I-D.ietf-lamps-cmp-updates], CRMF [RFC4211], and CMS [RFC5652] and [RFC8933] is not explicitly specified, it SHOULD NOT be used by the sending entity. The receiving entity MUST NOT require its absence and if present MUST gracefully handle its presence.

1.8. Structure of this Document

Section 2 introduces the general PKI architecture and approach to certificate management that is assumed in this document. Then it lists the PKI management operations specified in this document, partitioning them into mandatory, recommended, and optional ones.

Section 3 profiles the generic aspects of the PKI management operations specified in detail in Section 4 and Section 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, exchange 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 [RFC6712] transfer optionally using TLS, and [I-D.ietf-ace-cmpv2-coap-transport] transfer 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.

1.9. 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], 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.

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: 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 its locally reachable 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. Especially the communication between an LRA and RA can be performed synchronously or asynchronously. Synchronous communication describes

a timely uninterrupted communication between two communication partners, while asynchronous communication is not performed in a timely consistent manner, e.g., because of a delayed message delivery.

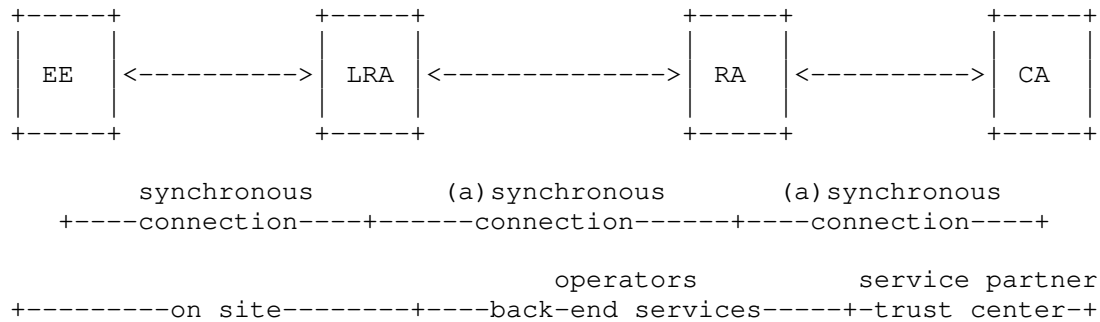


Figure 1: Certificate Management Architecture Example

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 (re-)protect CMP messages it processes. The figure above shows an architecture example with at least 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: CMP response messages could also be used proactively to implement the push model towards the EE. In this case the EE acts as receiver, not initiating the interaction with the PKI. Also, when using a commissioning tool or a registrar agent as described in BRSKI with Pledge in Responder Mode (BRSKI-PRM) [I-D.ietf-anima-brski-prm], certificate enrollment in a push model is needed. CMP in general and the messages specified in this profile offer all required capabilities, but the message flow and state machine as described in Section 4 must be adapted to implement a push model.

Third-party CAs may implement other variants of CMP, different standardized protocols, or even proprietary interfaces for certificate management. Therefore, the RA may need to adapt the exchanged CMP messages to the flavor of certificate management interaction required by the CA.

3. Generic Aspects of PKI Messages and PKI Management Operations

This section covers the generic aspects of the PKI management operations specified in Section 4 and Section 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:

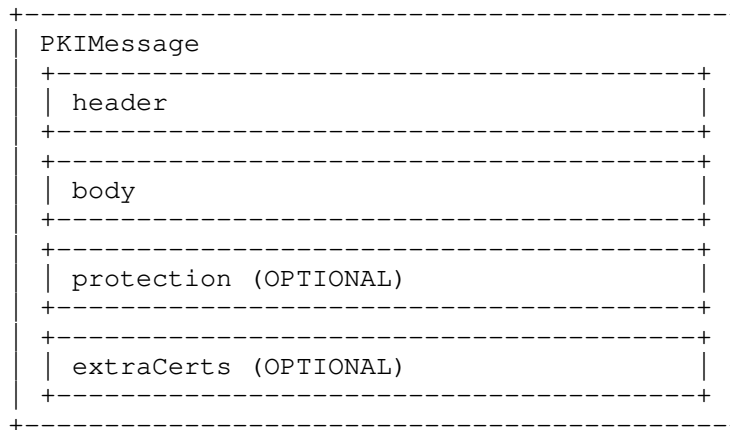


Figure 2: CMP Message Structure

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.

The CMP message body contains the PKI management operation-specific information. It is described in Section 4 and Section 5.

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 with signature-based protection.

In case a message has MAC-based protection the changes are described in Section 4.1.5. The variations will affect the fields sender, protectionAlg, and senderKID.

Any PKI management operation-specific fields or variations are described in Section 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 RFC-CMP-Updates
sender                              REQUIRED
-- SHOULD contain a name representing the originator of the
-- message; otherwise, the NULL-DN (a zero-length
-- SEQUENCE OF RelativeDistinguishedNames) MUST be used
-- SHOULD be the subject of the CMP protection certificate, i.e.,
-- the certificate corresponding to the private key used to sign
-- the message
-- In a multi-hop scenario, the receiving entity SHOULD NOT 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                        RECOMMENDED
-- MUST be the time at which the message was produced, if present
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 [RFC-CMP-Alg] Section 3 and
-- MUST be consistent with the subjectPublicKeyInfo field of
-- the protection certificate
-- If it is a MAC algorithm its type MUST be a MSG_MAC_ALG as
-- specified in [RFC-CMP-Alg] Section 6.1
senderKID                RECOMMENDED
-- MUST be the SubjectKeyIdentifier of the CMP protection
-- certificate in case of signature-based protection
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: SHOULD 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
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 [RFC-CMP-Updates]
CertProfileValue         REQUIRED
-- MUST contain exactly 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 with signature-based protection, which is the default protection mechanism for all CMP messages described in this profile. The private key used to sign a CMP message is called "protection key" and the related certificate is called "protection certificate". Any included keyUsage extension SHOULD allow digitalSignature.

protection

- RECOMMENDED for error messages
- REQUIRED for all other messages
- MUST contain the signature calculated using the private key
- of the entity protecting the message. The signature
- algorithm used MUST be given in the protectionAlg field.

Generally, CMP messages MUST be protected, but there are cases where protection of error messages specified in Section 3.6.4 is not possible and therefore MAY be omitted.

For MAC-based protection as specified in Section 4.1.5 and Section 4.1.6.3 major differences apply as described there.

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 with signature-based protection. Any specific requirements on the extraCerts are specified in the respective PKI management operation.

extraCerts

- SHOULD contain the CMP protection certificate together with
- its chain, if needed
- If present, 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.
- Self-signed certificates SHOULD be omitted from extraCerts,
- unless they are the same as the protection certificate and
- MUST NOT be trusted based on their inclusion in any case

Note: For maximum compatibility, 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:

- * Each EE SHOULD know its own identity to fill the sender field.
- * Each EE SHOULD 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.

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)
- * The message protection MUST be validated:
 - The protection MUST be signature-based except if
 - o MAC-based protection is used as described in Section 4.1.5 and Section 4.1.6.3 or
 - o protection is omitted in certain error messages as described in Section 3.6.4.(failInfo bit: wrongIntegrity)
 - The senderKID SHOULD identify the key material used for verifying the message protection. (failInfo bit: badMessageCheck)
 - The protection, if present, MUST be validated successfully. 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 SHOULD 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.

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 SHOULD match the subject of the CMP protection certificate. (failInfo bit: badMessageCheck)
- * If the messageTime is present, it SHOULD be close to the current time. (failInfo bit: badTime)

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.

3.6.1. Reporting Error Conditions Upstream

An EE SHALL NOT send error messages. PKI management entities SHALL NOT send error messages in 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. 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 SHOULD 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.

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, it MAY ignore this error condition and handle the response 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: certConf, ip, cp, kup, and error. The PKIStatusInfo structure consists of the following fields:

- * status: Here the PKIStatus value "rejection" MUST be used.

Note: 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" is 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 Section 5.1 and Section 6 are described here:
 - badCertId: A kur, certConf, or rr message references an unknown certificate
 - badPOP: An ir/cr/pl0cr/kur 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 has already been used 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 | RECOMMENDED |
| -- | SHOULD be 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 |
| -- | MAY be omitted if protection is technically not feasible |
| extraCerts | RECOMMENDED |
| -- | As described in Section 3.3 |

Note: 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.

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:

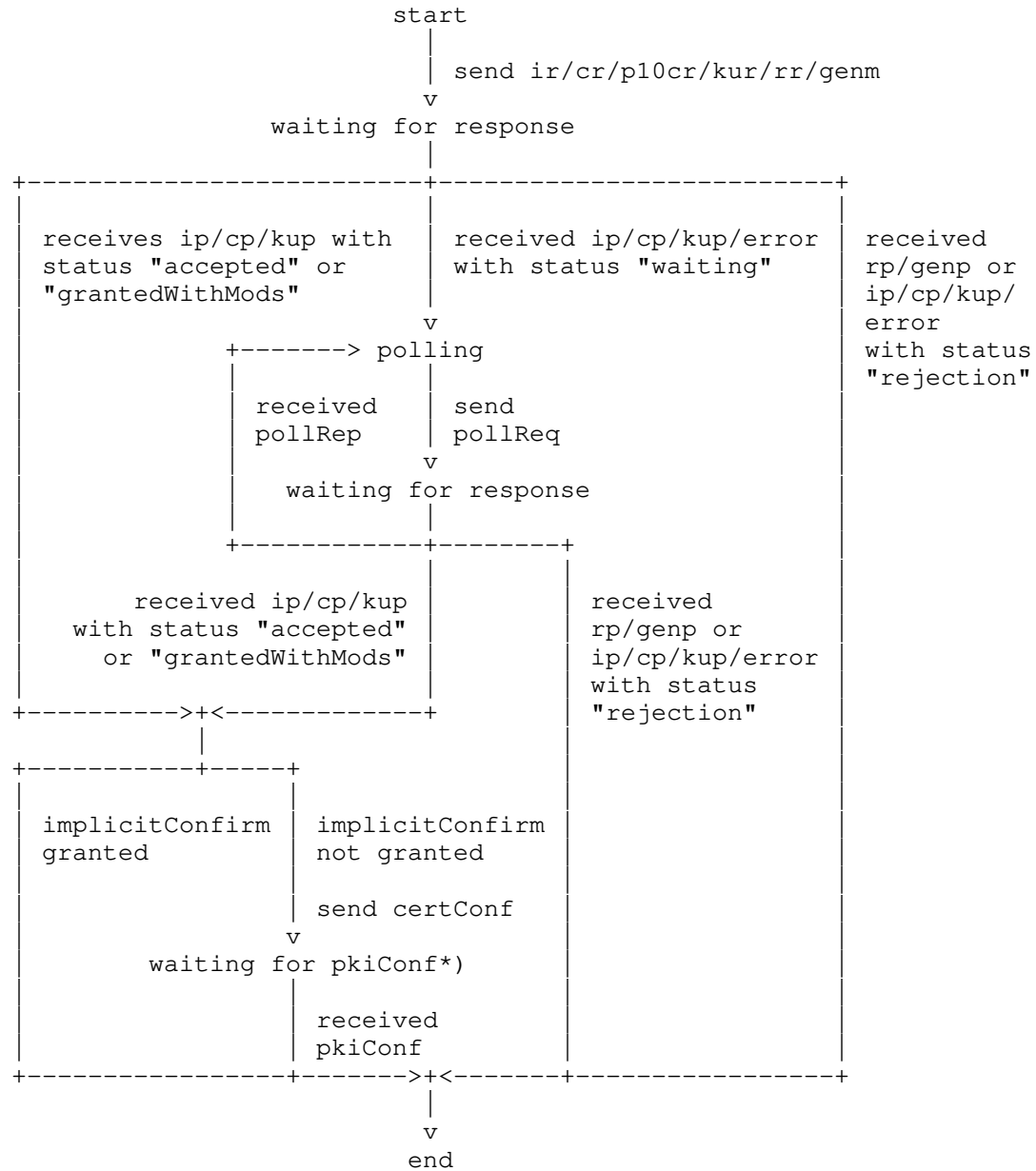
- * Requesting a certificate with variations like initial enrollment, certificate updates, central key generation, and MAC-based protection
- * Revoking a certificate
- * Support messages

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 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 and ECDSA, regardless of potentially intended restrictions of the key usage.

Note: In conformance with NIST SP 800-57 Part 1 Section 8.1.5.1.1.2 [NIST.SP.800-57plr5] 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 Asynchronous Enrollment (BRSKI-AE) [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 exactly one CertReqMsg in the ir. If more certificates are required, further requests MUST be sent using separate PKI management operation.

The EE SHOULD include the implicitConfirm extension in the header of the ir message as described in Section 3.1, unless it knows that certificate confirmation is needed. 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 PKI management entity does not need any explicit confirmation from the EE, it MUST include the extension as described in Section 3.1. This prevents explicit certificate confirmation and saves the overhead of a further message round-trip.

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 is 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 no certifiedKeyPair field. 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 exactly one CertReqMsg

-- If more certificates are required, further PKI management

-- operations MUST 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        REQUIRED
  algorithm        REQUIRED
-- MUST include the subject public key algorithm identifier
  subjectPublicKey  REQUIRED
-- MUST contain the public key to be certified in case of local
-- key generation
  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          RECOMMENDED
-- MUST be used by an EE if the key can be used for signing and
-- has 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 exactly 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
-- Self-signed certificates SHOULD be omitted
-- 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 exactly one CertStatus
CertStatus                              REQUIRED
certHash                                REQUIRED
-- MUST be the hash of the certificate, using the hash algorithm
-- indicated in hashAlg, see below, or the same one as used to
-- create the certificate signature
certReqId                                REQUIRED
-- MUST be 0
statusInfo                              RECOMMENDED
-- 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 certHash
-- SHOULD NOT be used in all cases where the AlgorithmIdentifier
-- of the certificate signature specifies a hash algorithm
-- If used, the pvno field in the header MUST be cmp2021 (3)

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 extraCerts 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 extraCerts 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, respectively.

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 SHOULD 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 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 a legacy PKCS#10 [RFC2986] request instead of CRMF [RFC4211]. This offers a variation of the PKI management operations specified in Section 4.1.2.

In Secure Zero Touch Provisioning (SZTP) [RFC8572] environments, SZTP-CSR [I-D.ietf-netconf-sztp-csr] describes the use of a CMP p10cr message as a form of certificate signing request (CSR) to optionally include in device bootstrapping to obtain an identity certificate as part of the onboarding information. Such a CSR is of form ietf-sztp-types:cmp-csr from module ietf-sztp-csr. The requirements given below on p10cr message MUST be adhered to.

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 Section 4.1.2, 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.
- 3 The caPubs field in the cp message SHOULD be absent.

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 Section 4.1.1 to Section 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 SHOULD replace this protection according to Section 5.2.3 in case the next hop does 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 Section 4.1.1, with the following changes:

- 1 The protection of all messages MUST be MAC-based.
- 2 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.
- 3 The extraCerts of all messages does not contain CMP protection certs and associated chains.

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 fields 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 perform 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.

As described in Section 5.3.1, the KGA can use one of the PKI management operations described in the sections above to request the certificate for this key pair on behalf of the EE.

Note: When performing central key generation for a certificate update, the KGA cannot use the old EE credentials for protection. Therefore, the PKI management operation described in Section 4.1.2 SHOULD be used instead of Section 4.1.3 to request a certificate for the newly generated key pair on behalf of the EE.

Generally speaking, in machine-to-machine scenarios it is strongly preferable to generate public-private key pairs locally at the EE. Together with proof-of-possession of the private key in the certificate request, 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 it 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 2.19 and Section 2.20 [I-D.ietf-lamps-cmp-updates].

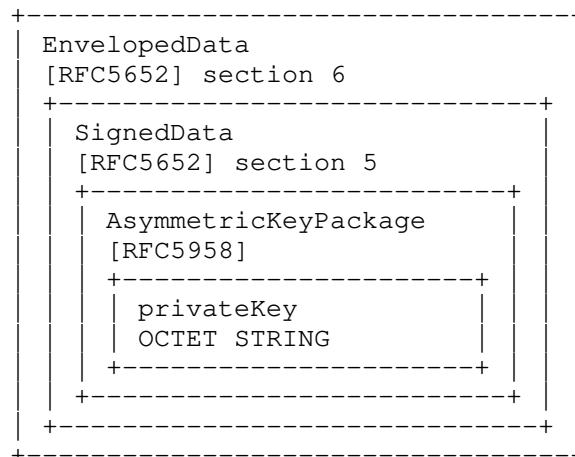


Figure 3: Encrypted Private Key Container

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. The EE SHOULD validate the signer certificate contained in the `SignedData` structure and verify the presence of this extended key usage in the signer certificate.

Note: When using password-based key management technique as described in Section 4.1.6.3 it may not be possible or meaningful to the EE to validate the KGA signature and the related certificate in the `SignedData` structure since shared secret information is used for initial authentication. In this case the EE MAY omit this validation.

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 more details on which key management technique to use.

* Signature-based protection of the request message:

- The content-encryption key SHALL be protected using the key agreement key management technique, see Section 4.1.6.1, if the certificate used by the EE for protecting the request message allows the key usage `keyAgreement`. If the certificate also allows the key usage `keyEncipherment`, the key transport key management technique SHALL NOT be used.

- The content-encryption key SHALL be protected using the key transport key management technique, see Section 4.1.6.2, if the certificate used by the EE for protecting the respective request message allows the key usage keyEncipherment but not keyAgreement.
- * 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.

If central key generation is supported, support of the key agreement key management technique is REQUIRED and support of key transport and password-based key management techniques are OPTION, for two reasons: The key agreement key management technique is supported by most asymmetric algorithms, while the key transport key management technique is supported only by a very few of them. The password-based key management technique shall only be used in combination with MAC-based protection, which is a sideline in this document.

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          OPTIONAL
-- 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 exactly 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
-- RFC-CMP-Alg 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 exactly 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 and [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 exactly 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 exactly 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 digestAlgorithms
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
-- 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 RFC-CMP-Alg Section 3 and MUST be consistent
-- with the subjectPublicKeyInfo field of the KGA certificate
signature REQUIRED
-- MUST be the digital signature of the encapContentInfo

NOTE: 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 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 allow for the key usage "keyAgreement" and therefore, 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 certificate,
-- and thereby identifies the sender's public key.
-- MUST contain the same value as the senderKID in the
-- message header
        ukm                                RECOMMENDED
-- MUST be used when 1-pass ECMQV is used
-- 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
-- RFC-CMP-Alg Section 4.1
-- The parameters field of the key agreement algorithm MUST
-- contains the key wrap algorithm
-- The algorithm type MUST be a KM_KW_ALG as specified in
-- RFC-CMP-Alg Section 4.3
        recipientEncryptedKeys
                                REQUIRED
-- MUST contain exactly one RecipientEncryptedKey element
        rid                                REQUIRED
-- MUST contain the rKeyId choice
        rKeyId                            REQUIRED
        subjectKeyIdentifier
                                REQUIRED
-- MUST contain the same value as the senderKID in the
-- respective request message header
        encryptedKey
                                REQUIRED
-- MUST be the encrypted content-encryption key

```

4.1.6.2. 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 allow for the key usage "keyEncipherment" and not for "keyAgreement". Therefore, 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 choice
    subjectKeyIdentifier
                                REQUIRED
-- MUST contain the same value as the senderKID in the
-- respective request message header
    keyEncryptionAlgorithm
                                REQUIRED
-- MUST be the algorithm identifier of the key transport
-- algorithm
-- The algorithm type MUST be a KM_KT_ALG as specified in
-- RFC-CMP-Alg Section 4.2
    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
-- RFC-CMP-Alg 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
-- RFC-CMP-Alg 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 the revocation of an own certificate 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 exactly one RevDetails structure in the rr message body. In 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 exactly one element of type RevDetails

-- If more revocations are desired, further PKI management operations MUST 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 exactly 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 exactly 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

In the following the aspects common to all general messages (genm) and general response (genp) messages are described.

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 exactly 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 exactly 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 2.13 [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`:

| | |
|--|----------|
| <code>infoValue</code> | OPTIONAL |
| -- MUST be absent if no CA certificate is available | |
| -- MUST be present if CA certificates are available | |
| -- MUST be a sequence of <code>CMPCertificate</code> | |

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`, which SHOULD include the old root CA certificate in the message body, as specified in CMP Updates Section 2.14 [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., 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 entities do 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
-- available
```

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 2.15 [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.

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 2.15 [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 2.15 [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 2.15 [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 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 2.16 [I-D.ietf-lamps-cmp-updates]. The PKI management entity MUST respond with a general response with OID `id-it-crls`. If no `thisUpdate` value was given by the EE, the PKI management entity MUST return the latest CRL available. If a `thisUpdate` value was given, the PKI management entity MUST return the latest available CRL in case this CRL is more recent, otherwise the `infoValue` in the response message MUST be absent.

The EE MUST identify the requested CRL either by its CRL distribution point name or issuer name. The CRL distribution point name can either be provided from the CRL distribution points extension of the certificate to be validated or from the issuing distribution point extension from the CRL to be updated. If no CRL distribution name is available to identify the CRL, the EE can use the issuer name either from the certificate to be validated or from the CRL to be updated.

The PKI management entity SHOULD treat a CRL distribution point name as an internal pointer to identify a CRL for which is available at the PKI management entity directly. It is not intended as a way to fetch an arbitrary CRL from an external location.

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 exactly one `CRLStatus` structure
- 3 if present, the `infoValue` of the response MUST contain exactly one CRL

Detailed Description of `infoValue` Field of `genm`:

CRLSource REQUIRED

- MUST contain exactly one CRLSource structure
- 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

- SHOULD contain the thisUpdate field of the latest CRL form the issuer specified in the given dpn or issuer field,
- in case such a CRL is already known by the EE

Detailed Description of infoValue Field of genp:

infoValue OPTIONAL

- MUST be absent if no CRL to be returned is available
- MUST contain exactly 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 an ip/cp/kup/error message including the status "waiting". 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. 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 |
| pKISStatusInfo | REQUIRED -- in case of error response |

-- PKISStatusInfo 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 extraCerts 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 |
|------------|----------|

```
-- 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 extraCerts 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 SHOULD 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 on reporting failure as described in Section 3.6.4 is transactionIdInUse. If any of these verification steps or any of the essential checks described in the following subsections fails, the PKI management entity MUST proceed as described in Section 3.6.

The responding PKI management entity SHOULD 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/pl0cr/kur 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 SHOULD 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 perform also 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 in the response the privateKey field as specified in Section 4.1.6. It may have issued the certificate for the newly generated key pair itself if it is a CA, or have requested the certificate on behalf of the EE as described in Section 5.3.1, or have received it by other means from a CA.

The prerequisites of the respective PKI management operation as specified in Section 4.1 apply.

Note: If the EE requested omission of the certConf message, the PKI management entity SHOULD handle it as described in Section 4.1.1 and therefore MAY grant this by including the implicitConfirm extension 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 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 still valid, 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 MUST completely validate the request as usual and then start preparing the response or forward the request 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.

Note: 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 SHOULD be forwarded to the next (upstream) PKI management entity as described in this section and otherwise MUST be answered as described in Section 5.1. Any received response message or 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/pl0cr/kur 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 necessary. The PKI management entity SHOULD only update the message protection and the certificate template in a certificate request message if this is technically necessary. 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,
- * store data from a message in a database for later usage or audit purposes,

- * provide traversal of a network boundary,
- * replace a MAC-based protection by a signature-based protection that can be verified also further upstream,
- * 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.

The decision if a message should be forwarded

- * unchanged with the original protection,
- * unchanged with a new protection, or
- * changed with a new protection

depends on the PKI solution design and the associated security policy (CP/CPS [RFC3647]).

A PKI management entity MUST replace or add 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.

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 SHOULD 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.

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 must 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 take over the recipient, recipNonce, and transactionID of the original message to 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 towards the next PKI management entity and on the downstream interface from the PKI management entity towards the EE.

This PKI management operation is typically used on the interface between an LRA and an RA to bundle several messages for offline 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 a suitable mechanism may need to be implemented. 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 SHOULD 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 SHOULD bundle the 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 can determine lost and unexpected responses based on the previously stored transactionIDs. When it forwards the unbundled responses, any extra messages SHOULD be dropped, and any missing response message (failInfo bit: systemUnavail) MUST be answered with an error message 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 SHOULD 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.

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 verify the protection provided and approve its content, including any modifications that it may perform. It MUST also check that the sender, 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, unless the regCtrl OldCertId is used to strongly identify the certificate to be updated.

These message adaptations MUST NOT be applied to certificate request messages described in for central key generation Section 4.1.6 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 SHOULD NOT change the transactionID nor the sender and recipient nonces.

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.

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 a signature-based 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.

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 SHOULD verify the received proof-of-possession.

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. HTTP SHOULD and CoAP MAY be used for online transfer. CMP messages MAY also be piggybacked on any other reliable transfer protocol. 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 CMP client component SHOULD use a configurable per-request timeout, and each CMP server component SHOULD use a configurable per-response 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 minimize the need for polling and maximize chances to handle transfer issues at lower levels.

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, though is not required from a security point of view.

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 5.3 of RFC 2510 [RFC2510], Section 2.4 of CMP over CoAP [I-D.ietf-ace-cmpv2-coap-transport], and Section 3.4 of CMP over HTTP [RFC6712].

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 SHOULD use URI paths 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] 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 CA Certificates | getcertreqtemplate | Section 4.3.1 |
| CRL Update Retrieval | getcrls | Section 4.3.4 |
| Batching Messages | nested | Section 5.2.2.2 |
| Note: This path element is applicable only between PKI management entities. | | |

Table 1: HTTP URI Path Segment <operation>

Independently of any variants used (see Section 4.1.5, Section 4.1.6, and Section 4.4) the operation label corresponding to the PKI management operation SHALL be used.

Any certConf or pollReq messages are 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 is 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 [RFC2818]. The CMP transfer via HTTPS MUST use TLS server authentication and SHOULD use TLS client authentication.

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 is acceptable using shared secret information with an entropy of at least 128 bits. Otherwise, a password-authenticated key exchange (PAKE) protocol is RECOMMENDED.

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 SHOULD use URI paths 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] 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>

Independently of any variants used (see Section 4.1.5, Section 4.1.6, and Section 4.4) the operation label corresponding to the PKI management operation SHALL be used.

Any certConf or pollReq messages are 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 is 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].

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 Section 6.1 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 SHOULD 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 offline EE 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 in this section 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 RFC 2510 Section 5.3 [RFC2510]. 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 which type of PKI entity.

7.1. PKI Management Operations

The following table provides an overview of the PKI management operations specified in Section 4 and Section 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, KUR, and P10CR if supported, Section 4.1.5 | SHOULD | SHOULD 1) | SHOULD |
| CKeyGen | Adding Central Key Pair Generation to Enrollment, IR, CR, KUR, and P10CR if supported, Section 4.1.6 If supported, key agreement key management technique is REQUIRED, and key transport and password-based key management techniques are OPTIONAL. | MAY | MAY | MAY |
| RR | Revoking a Certificate, Section 4.2 | SHOULD | SHOULD 2) | SHOULD |
| CACerts | Get CA Certificates, | MAY | MAY | MAY |

| | | | | |
|----------|--|-----|------|--------|
| | Section 4.3.1 | | | |
| 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, Section 4.4 | MAY | MAY | MAY |
| CertResp | Responding to a Certificate Request (IR, CR, KUR, and Pl0CR 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 |

| | | | | |
|----------|--|-----|-----------|--------|
| FwdRepKP | Forwarding Messages - Not Changing Proof-of-Possession, Section 5.2.3.1 | N/A | SHOULD 1) | N/A |
| FwdRepBP | Forwarding Messages - Using raVerified, Section 5.2.3.2 | N/A | MAY | MAY |
| CertOnB | 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 |

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.

7.2. Message Transfer

CMP does not have specific needs regarding message transfer, except that for each request message sent, eventually exactly 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 transfer is RECOMMENDED to use for 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 CA Certificates 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] |
| crt | 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.

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-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-13, 13 May 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-lamps-cmp-algorithms-13>>.
- [I-D.ietf-lamps-cmp-updates]
Brockhaus, H., Oheimb, D. V., and J. Gray, "Certificate Management Protocol (CMP) Updates", Work in Progress, Internet-Draft, draft-ietf-lamps-cmp-updates-18, 6 April 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-lamps-cmp-updates-18>>.

- [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/info/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>>.

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.
- [I-D.ietf-anima-brski-ae]
Oheimb, D. V., Fries, S., Brockhaus, H., and E. Lear, "BRSKI-AE: Alternative Enrollment Protocols in BRSKI", Work in Progress, Internet-Draft, draft-ietf-anima-brski-ae-01, 6 April 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-anima-brski-ae-01>>.
- [I-D.ietf-anima-brski-prm]
Fries, S., Werner, T., Lear, E., and M. C. Richardson, "BRSKI with Pledge in Responder Mode (BRSKI-PRM)", Work in Progress, Internet-Draft, draft-ietf-anima-brski-prm-03, 29 April 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-anima-brski-prm-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>>.
- [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>>.
- [RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, DOI 10.17487/RFC2818, May 2000, <<https://www.rfc-editor.org/info/rfc2818>>.
- [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/info/rfc5246>>.
- [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/info/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>>.
- [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/filebrowser/download/542_en>.

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 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 1.4.
- * 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 (editor)
Siemens AG
Email: hendrik.brockhaus@siemens.com

David von Oheimb
Siemens AG
Email: david.von.oheimb@siemens.com

Steffen Fries
Siemens AG
Email: steffen.fries@siemens.com

Network Working Group
Internet-Draft
Updates: 7299 (if approved)
Intended status: Informational
Expires: 10 April 2022

R. Housley
Vigil Security
7 October 2021

Update to the Object Identifier Registry for the PKIX Working Group
draft-ietf-lamps-rfc7299-update-02

Abstract

RFC 7299 describes the object identifiers that were assigned by Public-Key Infrastructure using X.509 (PKIX) Working Group in an arc that was allocated by IANA (1.3.6.1.5.5.7). A small number of object identifiers that were assigned in RFC 4212 are omitted from RFC 7299, and this document updates RFC 7299 to correct that oversight.

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 10 April 2022.

Copyright Notice

Copyright (c) 2021 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. IANA Considerations | 2 |
| 2.1. "SMI Security for PKIX CRMF Registration Controls for Alternate Certificate Formats" Registry | 2 |
| 3. Security Considerations | 3 |
| 4. References | 3 |
| 4.1. Normative References | 3 |
| 4.2. Informative References | 3 |
| Author's Address | 4 |

1. Introduction

When the Public-Key Infrastructure using X.509 (PKIX) Working Group was chartered, an object identifier arc was allocated by IANA for use by that working group. After the PKIX Working Group was closed, [RFC7299] was published to describe the object identifiers that were assigned in that arc. A small number of object identifiers that were assigned in RFC 4212 [RFC4212] are not included in RFC 7299, and this document corrects that oversight.

The PKIX Certificate Management Protocol (CMP) [RFC4210] allocated id-regCtrl-altCertTemplate (1.3.6.1.5.5.7.5.1.7), and then two object identifiers were assigned within that arc [RFC4212], which were intended to be used with either PKIX CMP [RFC4210] or PKIX Certificate Management over CMS (CMC) [RFC5272] [RFC5273] [RFC5274] [RFC6402].

This document describes the object identifiers that were assigned in that arc, established an IANA registry for that arc, and establishes IANA allocation policies for any future assignments within that arc.

2. IANA Considerations

IANA is asked to create one additional registry table.

2.1. "SMI Security for PKIX CRMF Registration Controls for Alternate Certificate Formats" Registry

Within the SMI-numbers registry, an "SMI Security for PKIX CRMF Registration Controls for Alternate Certificate Formats (1.3.6.1.5.5.7.5.1.7)" table with three columns has been added:

| Decimal | Description | References |
|---------|---------------------------|------------|
| ----- | ----- | ----- |
| 1 | id-acTemplate | [RFC4212] |
| 2 | id-openPGPCertTemplateExt | [RFC4212] |

Future updates to the registry table are to be made according to the Specification Required policy as defined in [RFC8126]. The expert is expected to ensure that any new values are strongly related to the work that was done by the PKIX Working Group. In particular, additional object identifiers should be needed for use with either the PKIX CMP or PKIX CMC to support alternative certificate formats. Object identifiers for other purposes should not be assigned in this arc.

3. Security Considerations

This document populates an IANA registry, and it raises no new security considerations. The protocols that specify these values include the security considerations associated with their usage.

4. References

4.1. Normative References

- [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>>.
- [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/info/rfc8126>>.

4.2. Informative References

- [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>>.
- [RFC4212] Blinov, M. and C. Adams, "Alternative Certificate Formats for the Public-Key Infrastructure Using X.509 (PKIX) Certificate Management Protocols", RFC 4212, DOI 10.17487/RFC4212, October 2005, <<https://www.rfc-editor.org/info/rfc4212>>.
- [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>>.
- [RFC5274] Schaad, J. and M. Myers, "Certificate Management Messages over CMS (CMC): Compliance Requirements", RFC 5274, DOI 10.17487/RFC5274, June 2008, <<https://www.rfc-editor.org/info/rfc5274>>.
- [RFC6402] Schaad, J., "Certificate Management over CMS (CMC) Updates", RFC 6402, DOI 10.17487/RFC6402, November 2011, <<https://www.rfc-editor.org/info/rfc6402>>.

Author's Address

Russ Housley
Vigil Security, LLC
516 Dranesville Road
Herndon, VA, 20170
United States of America

Email: housley@vigilsec.com

lamps
Internet-Draft
Intended status: Informational
Expires: 6 August 2022

D.K. Gillmor, Ed.
ACLU
2 February 2022

S/MIME Example Keys and Certificates
draft-ietf-lamps-samples-08

Abstract

The S/MIME development community benefits from sharing samples of signed or encrypted data. This document facilitates such collaboration by defining a small set of X.509v3 certificates and keys for use when generating such samples.

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 6 August 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. Requirements Language | 4 |
| 1.2. Terminology | 4 |
| 1.3. Prior Work | 4 |
| 2. Background | 5 |
| 2.1. Certificate Usage | 5 |
| 2.2. Certificate Expiration | 5 |
| 2.3. Certificate Revocation | 5 |
| 2.4. Using the CA in Test Suites | 6 |
| 2.5. Certificate Chains | 6 |
| 2.6. Passwords | 7 |
| 2.7. Secret key origins | 7 |
| 3. Example RSA Certification Authority | 7 |
| 3.1. RSA Certification Authority Root Certificate | 7 |
| 3.2. RSA Certification Authority Secret Key | 8 |
| 3.3. RSA Certification Authority Cross-signed Certificate | 9 |
| 4. Alice's Sample Certificates | 10 |
| 4.1. Alice's Signature Verification End-Entity Certificate | 10 |
| 4.2. Alice's Signing Private Key Material | 11 |
| 4.3. Alice's Encryption End-Entity Certificate | 12 |
| 4.4. Alice's Decryption Private Key Material | 13 |
| 4.5. PKCS12 Object for Alice | 14 |
| 5. Bob's Sample | 17 |
| 5.1. Bob's Signature Verification End-Entity Certificate | 17 |
| 5.2. Bob's Signing Private Key Material | 18 |
| 5.3. Bob's Encryption End-Entity Certificate | 19 |
| 5.4. Bob's Decryption Private Key Material | 20 |
| 5.5. PKCS12 Object for Bob | 21 |
| 6. Example Ed25519 Certification Authority | 24 |
| 6.1. Ed25519 Certification Authority Root Certificate | 24 |
| 6.2. Ed25519 Certification Authority Secret Key | 25 |
| 6.3. Ed25519 Certification Authority Cross-signed Certificate | 25 |
| 7. Carlos's Sample Certificates | 26 |
| 7.1. Carlos's Signature Verification End-Entity Certificate | 26 |
| 7.2. Carlos's Signing Private Key Material | 27 |
| 7.3. Carlos's Encryption End-Entity Certificate | 27 |
| 7.4. Carlos's Decryption Private Key Material | 27 |
| 7.5. PKCS12 Object for Carlos | 28 |
| 8. Dana's Sample Certificates | 29 |
| 8.1. Dana's Signature Verification End-Entity Certificate | 29 |
| 8.2. Dana's Signing Private Key Material | 30 |
| 8.3. Dana's Encryption End-Entity Certificate | 30 |
| 8.4. Dana's Decryption Private Key Material | 30 |
| 8.5. PKCS12 Object for Dana | 31 |
| 9. Security Considerations | 32 |

| | |
|--|----|
| 10. IANA Considerations | 32 |
| 11. Document Considerations | 32 |
| 11.1. Document History | 32 |
| 11.1.1. Substantive Changes from draft-ietf-*-07 to draft-ietf-*-08 | 32 |
| 11.1.2. Substantive Changes from draft-ietf-*-06 to draft-ietf-*-07 | 33 |
| 11.1.3. Substantive Changes from draft-ietf-*-05 to draft-ietf-*-06 | 33 |
| 11.1.4. Substantive Changes from draft-ietf-*-04 to draft-ietf-*-05 | 33 |
| 11.1.5. Substantive Changes from draft-ietf-*-03 to draft-ietf-*-04 | 33 |
| 11.1.6. Substantive Changes from draft-ietf-*-02 to draft-ietf-*-03 | 33 |
| 11.1.7. Substantive Changes from draft-ietf-*-01 to draft-ietf-*-02 | 33 |
| 11.1.8. Substantive Changes from draft-ietf-*-00 to draft-ietf-*-01 | 34 |
| 11.1.9. Substantive Changes from draft-dkg-*-05 to draft-ietf-*-00 | 34 |
| 11.1.10. Substantive Changes from draft-dkg-*-04 to draft-dkg-*-05 | 34 |
| 11.1.11. Substantive Changes from draft-dkg-*-03 to draft-dkg-*-04 | 34 |
| 11.1.12. Substantive Changes from draft-dkg-*-02 to draft-dkg-*-03 | 34 |
| 11.1.13. Substantive Changes from draft-dkg-*-01 to draft-dkg-*-02 | 34 |
| 11.1.14. Substantive Changes from draft-dkg-*-00 to draft-dkg-*-01 | 34 |
| 12. Acknowledgements | 34 |
| 13. References | 35 |
| 13.1. Normative References | 35 |
| 13.2. Informative References | 36 |
| Author's Address | 37 |

1. Introduction

The S/MIME ([RFC8551]) development community, in particular the e-mail development community, benefits from sharing samples of signed and/or encrypted data. Often the exact key material used does not matter because the properties being tested pertain to implementation correctness, completeness or interoperability of the overall system. However, without access to the relevant secret key material, a sample is useless.

This document defines a small set of X.509v3 certificates ([RFC5280]) and secret keys for use when generating or operating on such samples.

An example RSA certification authority is supplied, and sample RSA certificates are provided for two "personas", Alice and Bob.

Additionally, an Ed25519 ([RFC8032]) certification authority is supplied, along with sample Ed25519 certificates for two more "personas", Carlos and Dana.

This document focuses narrowly on functional, well-formed identity and key material. It is a starting point that other documents can use to develop sample signed or encrypted messages, test vectors, or other artifacts for improved interoperability.

1.1. 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.

1.2. Terminology

- * "Certification Authority" (or "CA") is a party capable of issuing X.509 certificates
- * "End-Entity" is a party that is capable of using X.509 certificates (and their corresponding secret key material)
- * "Mail User Agent" (or "MUA") is a program that generates or handles [RFC5322] e-mail messages.

1.3. Prior Work

[RFC4134] contains some sample certificates, as well as messages of various S/MIME formats. That older work has unacceptably old algorithm choices that may introduce failures when testing modern systems: in 2019, some tools explicitly mark 1024-bit RSA and 1024-bit DSS as weak.

This earlier document also does not use the now widely-accepted PEM encoding (see [RFC7468]) for the objects, and instead embeds runnable Perl code to extract them from the document.

It also includes examples of messages and other structures which are greater in ambition than this document intends to be.

[RFC8410] includes an example X25519 certificate that is certified with Ed25519, but it appears to be self-issued, and it is not directly useful in testing an S/MIME-capable MUA.

2. Background

2.1. Certificate Usage

These X.509 certificates ([RFC5280]) are designed for use with S/MIME protections ([RFC8551]) for e-mail ([RFC5322]).

In particular, they should be usable with signed and encrypted messages, as part of test suites and interoperability frameworks.

All end-entity and intermediate CA certificates are marked with Certificate Policies from [TEST-POLICY] indicating that they are intended only for use in testing environments. End-entity certificates are marked with policy 2.16.840.1.101.3.2.1.48.1 and intermediate CAs are marked with policy 2.16.840.1.101.3.2.1.48.2.

2.2. Certificate Expiration

The certificates included in this draft expire in 2052. This should be sufficiently far in the future that they will be useful for a few decades. However, when testing tools in the far future (or when playing with clock skew scenarios), care should be taken to consider the certificate validity window.

Due to this lengthy expiration window, these certificates will not be particularly useful to test or evaluate the interaction between certificate expiration and protected messages.

2.3. Certificate Revocation

Because these are expected to be used in test suites or examples, and we do not expect there to be online network services in these use cases, we do not expect these certificates to produce any revocation artifacts.

As a result, none of the certificates include either an OCSP indicator (see id-ad-ocsp as defined in the Authority Information Access X.509 extension in S.4.2.2.1 of [RFC5280]) or a CRL indicator (see the CRL Distribution Points X.509 extension as defined in S.4.2.1.13 of [RFC5280]).

2.4. Using the CA in Test Suites

To use these end-entity certificates in a piece of software (for example, in a test suite or an interoperability matrix), most tools will need to accept either the Example RSA CA (Section 3) or the Example Ed25519 CA (Section 6) as a legitimate root authority.

Note that some tooling behaves differently for certificates validated by "locally-installed root CAs" than for pre-installed "system-level" root CAs). For example, many common implementations of HPKP ([RFC7469]) only applied the designed protections when dealing with a certificate issued by a pre-installed "system-level" root CA, and were disabled when dealing with a certificate issued by a "locally-installed root CA".

To test some tooling specifically, it may be necessary to install the root CA as a "system-level" root CA.

2.5. Certificate Chains

In most real-world examples, X.509 certificates are deployed with a chain of more than one X.509 certificate. In particular, there is typically a long-lived root CA that users' software knows about upon installation, and the end-entity certificate is issued by an intermediate CA, which is in turn issued by the root CA.

The example end-entity certificates in this document can be used with either a simple two-link certificate chain (they are directly certified by their corresponding root CA), or in a three-link chain.

For example, Alice's encryption certificate (Section 4.3, `alice.encrypt.crt`) can be validated by a peer that directly trusts the Example RSA CA's root cert (Section 3.1, `ca.rsa.crt`):

```
ca.rsa.crt  alice.encrypt.crt
```

And it can also be validated by a peer that only directly trusts the Example Ed25519 CA's root cert (Section 6.1, `ca.25519.crt`), via an intermediate cross-signed CA cert (Section 3.3, `ca.rsa.cross.crt`):

```
ca.25519.crt  ca.rsa.cross.crt  alice.encrypt.crt
```

By omitting the cross-signed CA certs, it should be possible to test a "transvalid" certificate (an end-entity certificate that is supplied without its intermediate certificate) in some configurations.

2.6. Passwords

Each secret key presented in this draft is represented as a PEM-encoded PKCS#8 [RFC5958] object in cleartext form (it has no password).

As such, the secret key objects are not suitable for verifying interoperable password protection schemes.

However, the PKCS#12 [RFC7292] objects do have simple textual passwords, because tooling for dealing with passwordless PKCS#12 objects is underdeveloped at the time of this draft.

2.7. Secret key origins

The secret RSA keys in this document are all deterministically derived using provable prime generation as found in [FIPS186-4], based on known seeds derived via [SHA256] from simple strings. The validation parameters for these derivations are stored in the objects themselves as specified in [RFC8479].

The secret Ed25519 and X25519 keys in this document are all derived by hashing a simple string. The seeds and their derivation are included in the document for informational purposes, and to allow re-creation of the objects from appropriate tooling.

All RSA seeds used are 224 bits long (the first 224 bits of the SHA-256 digest of the origin string), and are represented in hexadecimal.

3. Example RSA Certification Authority

The example RSA Certification Authority has the following information:

- * Name: Sample LAMPS RSA Certification Authority

3.1. RSA Certification Authority Root Certificate

This certificate is used to verify certificates issued by the example RSA Certification Authority.

```
-----BEGIN CERTIFICATE-----
MIIDezCCAmOgAwIBAgITcBn0xb/zdaeCQlqp6yZUAGZUCDANBgkqhkiG9w0BAQ0F
ADBVMQ0wCwYDVQQKEwRJRVRGMREwDwYDVQQLEwhMQU1QUyBXRzExMC8GA1UEAxMo
U2FtcGx1IEExBTBTIFJTSBDZXJ0aWZpY2F0aW9uIEFldGhvcml0eTagFw0xOTEx
MjAwNjU0MThaGA8yMDUyMDkyNzA2NTQxOFowVTENMA8GA1UEChMESUVURjERMA8G
A1UECzMITEFNUFMgV0cxMTAvBgNVBAMTKFNhbXBsZSBMQU1QUyBSU0EgQ2VydGlm
aWNhdGlvbiBBdXRob3JpdHkwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIB
AQc2GGPTEFVNdi0LsiQ79A0Mz2G+LRJlX2vNo8STibAnyQ9VzFrGJHjUHRX/Omr
OP3rDCB2SYfBPVwd0CdC6z9qfJkcVxDclhK+VS9vKncL0IPUYlkJwWuMpXa1Ielz
+zCuV+gJv83Uvn6wTn39MCmyu7nFPzihcuOnbMYOCdMmUbi1Dm8TX9P6itFR3hi
IHpSKMbkoXlM1837WafFx57kBIoIuNjKeyPIuK9wGUAeppc5QAHJg95PPEHNLmM
yhBzC1mgkyozRSeSrKxq9XeJKU94lWGaz0zb4karCur/eiMoCk3YNV8L3styvcMG
1qUDCAaKx6FzEf7he9RN6L3bAgMBAAGjQjBAMA8GA1UdEwEB/wQFMAMBAf8wDgYD
VR0PAQH/BAQDAgEGMB0GA1UdDgQWBBSRMI58BxcMp/EJKGU2GmccaHb0WTANBgkq
hkiG9w0BAQ0FAAOCAQEACDXWlJGjzKadNMPcFlZInZC+Hl7RLrcBDR25jMCXg9yL
IwGVEcNp2fh4+YHTRTGLH8laPADMdUGHppfcfqwjесavt/m00T0S0LjJ0RVm93fE
heSNUHUigVR9njTVw2EBz7e2p+v3tOsMnunvm6PIDgHxx0W6mjzMX7lG74bJfo+v
dx+jI/aXt+iih5pi7/2Yu9eTDVu+S52wsnF89BEJeV0r+EmGDxUv47D+5KuQpKM9
U/isXpwC6K/36T8RhhdOQXDq0Mt9lTZ4dJTT0m3cmo80zzcxskMDStZHOOzCBtBq
uIbwWw5Oa72o/Iwg9v+W0WkSBCWEadf/uK+cRicxrQ==
-----END CERTIFICATE-----
```

3.2. RSA Certification Authority Secret Key

This secret key material is used by the example RSA Certification Authority to issue new certificates.

```
-----BEGIN PRIVATE KEY-----
MIIE+wIBADANBgkqhkiG9w0BAQEFAASCBAKgwggSkAgEAAoIBAQC2GGPTEFVNdi0L
siQ79A0Mz2G+LRJlbX2vNo8STibAnyQ9VzFrGJHjUhRX/OmrOP3rDCB2SYfBPVwd
0CdC6z9qfJkcVxDclhK+VS9vKncL0IPUYlkJwWuMpXaIelz+zCuV+gjjV83Uvn6w
Tn39MCmyu7nFPzihcuOnbMYOCdMmUbilDm8TX9P6itFR3hiIHpSKMbkoXlM1837
WaFfx57kBIoIuNjKEYPIuK9wGUAeppc5QAHJg95PPEHNHlmMyhBzClmgkyozRSeS
rkxq9XeJKU94lWGaz0zb4karCur/eiMoCk3YNV8L3styvcMG1qUDCAaKx6FZEf7h
E9RN6L3bAgMBAAECggEAE3tFhsm7DpgDlro+1Sk1kjbHssR4sOBHb4zrPp6c18PO
6T8gWuBcj1DzOzykNTzaMaDxAia4vuxVJB1mberkNHZTFqyb8bx3ceSEOct3aoyq
5fiFpR0L6Balvgg8RTvNCAIApHNa4pVx0XD8Wq+h7mlUAOYGBie5U08/P2qWjcOz
+zcheyYXJS/iiu0t2/F0ihEWGcXBmoc8D++n7mKst2jkAHD4w1PN2MgVqnmagpBz
gobFNmCZyZpDS+PPTtQZ1XvdGF5Sodc+Fz+jpWunlkqxDHE4UIZzDA/HAABgORbm
aEZAvsOs9ZExeqOtqu2fPB7zF/1JKdRk4UJOUxS0OQKBgQDJwonP5RwvO0sYoCiw
zuFcYTmN/hI3R3viKuxr19CH6+mvuIU85ooIHF6TlouZwhk+6+Vx7rcXdS554DT4
2RbVrX/5i/MOzx8c8IIwoZJIasLz+vx8F4n6hyhV65bXN7AIBojMh2dt8tP2MZ/R
VEfsk4mNmO6yKuzYAfjjZiCnCQKBgQDnDH9UYUIPkg0PSvViKQFJFCB9BJPFhld2
pIgoziw/JZzM3W3IWU0KWG7UxS0T3xmn3IX6xmWW4vX1/088ybObZWYP0edb61GM
I9DoI5igndLgDwyOL2PFuZh5pqqc09DE+cpJW4nNoudqTNmCrjhmxNCGKgGjld8z
/OkSccvywwKBgDd0ReajRUziEjDxjF2UbzKx81zJsX4KIs22GidHqSRCv1cy80Qa
5WN3ULNiyB350HCP69wDFMXym5rJoQjPvh6GIuhYKv4V8fffxkYv5kx5uWiXZVJ
7v2x+m8rMqlyv+pkyWLV8KKytHmdiBzD+oTWx7r4ueLjtaxngzx93pAoGBAKpR
rR9PnroKHubSE/drUNZFLvnZwPDv6108T978tONL372pUT9KjR8eN31DaMpoQOpc
BqvpSoQjBLtlnDysV2krI0RwMIOzAWC0E9C8RMvJ6+RdU50Q1BSyjjvLGaKi5AAHk
PTk8cGYV01BCHGLX8p3XYfw0xQaHxtuVCV8eYgCvAoGBAIZeiVhc0YTJOjUadz+0
vSozA1arg5k2YCPCGF7z+ijM5rbMk7jrYixD6WMjTokVLHDSVxMBpbA7GhL7TKy5
cepBH1PVwxEI18dqN+UoeJeBpnHo/cjJ0iCR9/aMJzI+qiUo3OMDR+UH99NiddKN
i75GRVLAeW0Izgt09EMEiD9joDswOQYKKwYBBAGSCBIATERMCKGCWCGSAFlAwQC
AgQcpcG3hHYU7WYaawUiNRQotLfwnYzMotMTAtli6Q==
-----END PRIVATE KEY-----
```

This secret key was generated using provable prime generation found in [FIPS186-4] using the seed a5c1b7847614ed661a6b0522351428b4b7f09d8ccca2d99302dd62e9. This seed is the first 224 bits of the [SHA256] digest of the string draft-lamps-sample-certs-keygen.ca.rsa.seed.

3.3. RSA Certification Authority Cross-signed Certificate

If an e-mail client only trusts the Ed25519 Certification Authority Root Certificate found in Section 6.1, they can use this intermediate CA certificate to verify any end entity certificate issued by the example RSA Certification Authority.

```
-----BEGIN CERTIFICATE-----
MIIC5zCCApmgAwIBAgITcTQnnf8DUsvAdvkX7mUemYos7DAFBgMrZXAwWTENMAsG
A1UEChMESUVURjERMA8GA1UECzMITEFNUFMgV0cxNTAzBgNVBAMTLFNhbXBsZSBM
QU1QUyBFZDI1NTE5IENlcnRpZmljYXRpb24gQXV0aG9yaXR5MCAXDTEwMTIxNTIx
MzU0NFoYDzIwNTIwOTI3MDY1NDE4WjBVMQ0wCwYDVQQKEwRJRVRGMREwDwYDVQQL
EwhMQU1QUyBXRzExMC8GA1UEAxMoU2FtcGx1IEExBTVBtIFJTSBDZXXJ0aWZpY2F0
aW9uIEF1dGhvcml0eTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBALYY
Y9MQVU12LQuyJDv0DQzPYb4tEmVtfa82jxJOJsCfJD1XMWsyKeNSFFf86as4/esM
IHZJh8E9XB3QJ0LrP2p8mRxXENzWEr5VL28qdwvQg9RiWQnBa4ylldrUh6XP7MK5X
6CNXzdS+frBOff0wKbKa7ucU/OKFy46dsxg4J0yZRuLUObxNf0/qK0VHeGIgelIo
xuSheUzXzftZoV/HnuQEigi42MoTi8i4r3AZQB6mlz1AAcmD3k88Qc0eWYzKEHMK
WaCTKjNFJ5KuTGrld4kpt3iVYZpnTNviRqsK6v96IygKTdglXwvey3K9wwbWpQMI
BorHoVkr/uET1E3ovdsCAwEAAa8MH0wDwYDVR0TAQH/BAUwAwEB/zAXBgNVHSAE
EDAOMAwwGCMCGSAFlAwIBMAIwDgYDVROPAQH/BAQDAgEGMB0GA1UdDgQWBBSRMI58
BxcMp/EJKGU2GmccaHb0WTAfBgNVHSMEGDAWgBRropV9uhSb5C0E0Qek0YLkLmuM
tTAFBgMrZXADQQBnQ+0eFP/BBKz8bVELVEPw9WFXwIGnyH7rrmLQJSE5GJmm7cYX
FFJBGyc3NWzlxxyfJLsh0yYh04dxdM8R5hcD
-----END CERTIFICATE-----
```

4. Alice's Sample Certificates

Alice has the following information:

- * Name: Alice Lovelace
- * E-mail Address: alice@smime.example

4.1. Alice's Signature Verification End-Entity Certificate

This certificate is used for verification of signatures made by Alice.

```
-----BEGIN CERTIFICATE-----
MIIDZzCCAreGawIBAgITN0EFee11f0Kpolw69PhqzpqplzANBgkqhkiG9w0BAQOF
ADBVMQ0wCwYDVQQKEwRJRVRGMREwDwYDVQQLEwhMQU1QUyBXRzExMC8GA1UEAxMo
U2FtcGx1IEExBTBVTIFJTQSBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eTagFw0xOTEx
MjAwNjU0MThaGA8yMDUyMDkyNzA2NTQxOFowOzENMA8GA1UEChMESUVURjERMA8G
A1UECxMITEFNUFMgV0cxZzAVBgNVBAMTDkFsaWN1IExvdmVsYWN1MIIBIjANBgkq
hkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAtPSJ6Fg4Fj5Nmn9PkrYo0jTkfCv4TfA/
pdO/KLpZbJOAER0sI7AjaO7B1GuMUFJeSTu1amNfCwDcDkY63PQW1+DILs7GxVwX
urhYdZlaV5hcUqVAcKpvedDBc/3rz4D/esFfs+E7QMfTmd+K04s+A8TCNO12DRVB
DpbP4JFD9hsc8prDtpGmFk7rd0q8gqnhxBW2RZAeLqzJOMayCQtws1q7ktkNBR2w
ZX5ICjecF1YJFhX4jrnHwp/iELGqqaNXd3/Y0pG7QFecN7836IPpdfTMSiPR+peC
rhJZwLSewbWXLJe3VMvbvQjoBMpEYlaJBUIKk01zQ1Pq90njlsJL0wIDAQABo4Gv
MIGsMAwGA1UdEwEB/wQCMAAwFwYDVR0gBBawDjAMBgpghkgBZQMCATABMB4GA1Ud
EQQXMBWBE2FsaWN1QHNTaW11LmV4YW1wbGUwEwYDVR01BAwwCgYIKwYBBQUHAWQw
DgYDVR0PAQH/BAQDAgBAMBOGA1UdDgQWBBS79syyLR0GEhyXrilqkBDTIGZmczAf
BgNVHSMEGDAWgBSRMI58BxcMp/EJKGU2GmccaHb0WTANBgkqhkiG9w0BAQ0FAAOC
AQEAc4miNqfOqaBpI3f+CpJDhxtuZ2P9HjJQE+Q+v6BdP7GKJ19naIs3BjJ0d64roA
KHAp+c284VvyVXWJ99FMX8q2ZUQMxH+xh6oAfzcozmnd6XaVWHg4eHIjSo27PmhK
EloAJKKhDbdbEcZXL2+x1V+duGymWtaD01DZZukKYr7agyHahixRn/C9cy31wbqN
sy9x0fjPQg6+DqatiQpMz9Eiae6aCHHBhOiPU7IPkazgPYgkLD59fk4PGHnYxs1F
hdO6zZk9E8zwlc1ALgZa/iSbczisqckN3qGehD2s16jMhwFXLJtBiN+uCDgNG/D0
qyTbY4fgKieUHx/tHuzUszZxJg==
-----END CERTIFICATE-----
```

4.2. Alice's Signing Private Key Material

This private key material is used by Alice to create signatures.

-----BEGIN PRIVATE KEY-----

```
MIIE+gIBADANBgkqhkiG9w0BAQEFAASCBCwggSjAgEAAoIBAQC09InoWDgWPK2a
f0+StijSNOR8K/hN8D+l078oullsk4ASvSwjsCNo7sHUa4xQU15JO6VqY18LANwO
Rjrc9BaX4MguzsbFXBe6uFhlmVpXmFxSpUByQ+950MFz/evPgP96wV+z4TtAwW2Z
34rTiz4DxMI07XYNFUEOlS/gkUP2GxzymsO2kaYWTut3SryCqeHEFbZfKb4urMk4
xrIJC3CzWruS2Q0FHbBlfkgKN5wXVgkWFfiOucfCn+IQsaqpold3f9jSkbtAV5w3
vzfog8919MxKI9H6l4KuElnAtJ7BtZcs17dUy9u9COgEykRiVokFQgqQ7XNDU+r3
SeOWwks7AgMBAAECggEAFKD2DG9A1u77q3u3p2WDH3zueTtiqgaT8u8XO+jhOI/+
HzoX9eo8DIJ/b/G3brwHyfh17JFvLH1zbghsn5bghJTz3r+JcZZ5l3srqMV8t8zjI
JEHOKC3szH8gYVKWrIgbaQot1H9Ti8J2oKk2aymqBFr3ZXpBUCTWpEz2s3FMBUUI
qCEsAJqsdeCh+kt43X5kvAom7LC1DHiE6RKfhMEub/LGNHSwY4dmzhaG6p95FJ1h
s8HoURI2ReVpsTadaKd3KoYNc1lcfmwdZs/hFs7xmmwXKMmlonhlmzHqD1/BqeJ
Hc8MP4ueDdyVgIe/uVt1Q9NcRQbuokkDyDYMYV6hzQKBgQD75ahYGFgzZnRktSE3
w/2rUqTYIwxx2PQz5G58PcsTZM89Hj4aZ0oLmudHbrTQHluRNcHoXEI62rs0cVPs
D7I1ZOLfs+SSTeNEXxD57mjyyufpV650cNc1mSJAmMX2jWQ8ndnOuWPcc5J6fNvT
au0a7ZBOaeKHnA8XXL3GYilM9QKBgQC35xKi7f2JmGtsYY21tfRuDUm6EjhMW6b7
GWnI9IXF8TGj15s7oDEYvqSPTJdB6PAb/tZwdbj9mB4qj176x1kB/N7GO97408UP
/PdHkU7duyf5nRq1mrI+yGFHVsgD3l3rc+akYdKcC207e6IRMST1ZFoznC6qNgpi
nNTuDz4ZbwKBgA5Dd9/dKkm77gvY69Objn6oBFuUsO5VaaaSlcsFOL2VZMLCNqQJ
+NLFZ7k8xJJQVcEIOT2uE7X/csBKdoUUCnL5nnsqVZQPQwI5G937KQguYlMZLte
WmFX1X/w5qzKXtWr3ox9JPFzveSfs1bqZBi1QQmfp0skhBo/jyNvpYUNaoGAMNkw
GhcdQW87GY7QFXQ/ePwOmV49lgrCT/BwKPDk18l5ZgvfL/ddEzWQgH/XraoyHT2T
uEuM18+QM73hfLt26RBCHGXK1CUMMzL+faQc7sjH1YX1kleFASg4rrprcrKqoR+KB
YSiayNhAK4yrf+WN66C8VPknbA7us0L1TEbAOAECgYEAtwRiiQwk3BlqENFypyc8
0Q1pxp3U7ciHi8mni0kNcTqe57Y/2o8nY9ISnt1GffMs79YQfRXTRdEm2St6oChI
9Cv5j74LHZXkgEVFfO2Nq/uwSzTZkePk+HoPJ04WtAdokZgRAyyH10gEae8R189e
yBX7dutONALjRZFTrgl8CuegOzA5BgorBgEEAZIIEggBMSswKQYJYIZIAWUDBAIC
BBySyJ1DMNPY4x1P3pudD+bp/BQhQd1lpF5bQ28F
```

-----END PRIVATE KEY-----

This secret key was generated using provable prime generation found in [FIPS186-4] using the seed 92c89d4330d3d8e31d4fde9b9d0fe6e9fc142141dd65a45e5b436f05. This seed is the first 224 bits of the [SHA256] digest of the string draft-lamps-sample-certs-keygen.alice.sign.seed.

4.3. Alice's Encryption End-Entity Certificate

This certificate is used to encrypt messages to Alice.


```
-----BEGIN CERTIFICATE-----
MIIDzzCCAreGAWIBAgITDy01vRE510rOQ1SHoe49NAaKtDANBgkqhkiG9w0BAQ0F
ADBVMQ0wCwYDVQQKEwRJRVRGMREwDwYDVQQLEwhMQU1QUyBXRzExMC8GA1UEAxMo
U2FtcGx1IEExBTBTIFJTSBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eTagFw0xOTEx
MjAwNjU0MThaGA8yMDUyMDkyNzA2NTQxOFowOzENMA8GA1UEChMESUVURjERMA8G
A1UECXMITEFNUFMgV0cxZzAVBgNVBAMTDkFsaWN1IExvdmVsYWN1MIIBIjANBgkq
hkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAmP+ovBouOP6AFQJ+RpwpoDxxzY60n1
lJ53pTeNSiJlWkwtw/cxQq0t4uD2vWYB8gOUH/Cvt2Zp1c+auzPKJ2Zu5mY6kHm+
hVB+IthjLeI7Htg6rNeuXq50/TuTSxX5R1I1EXGt8p6hAQVeA5oZ2afHg4b97enV
8gozR0/Nkug4AkXmbk7THNc8vvjMUJanZ/VmS4TgDqXjWShplcI3lcvvBZMswt4l
/0HJvmsWqps6oQcAx3Weag0yCNj1V9V9yu/3DjcYbwW2lJf5NbMHbM1LY4X5chWf
NEbkN6hQury/zxnlsukgn+fHbqvDhJLAgFpW/jA/EB/WI+whUpqtQIDAQABo4Gv
MIGsMAwGA1UdEwEB/wQCMAAwFwYDVR0gBBawDjAMBgpghkgBZQMCATABMB4GA1Ud
EQQXMBWBE2FsaWN1QHNTaW1lLmV4YW1wbGUwEwYDVR0lBAwwCgYIKwYBBQUHAWQw
DgYDVR0PAQH/BAQDAgUGMB0GA1UdDgQWBBSiU0HVRDyAKRV8ASPw546vzfN3DzAf
BgNVHSMEGDAWgBSRMI58BxcMp/EJKGU2GmccaHb0WTANBgkqhkiG9w0BAQ0FAAOC
AQEAgl4oJyxMpwWpAylOvK6NEbM1lgD5H14EC4Muxqlu0q2XgXOSBHI6DfX/4LD
sfx7fSIus8gWVY3WqMeuOA7IizkBD+GDEu8uKveERRXZncxGwy2MfbH1Ib3U8QzT
jqB8+dz2AwYeMxODWq9opwtA/1TOkRg8uuiVZfg/m5fFo/QshlHNaaTDVEXsU4Ps
98Hm/3gznbvhdjFbZbi4oZ3tAadR1E5K9JiQaJYOnUmGpfB8PPwDR6chMZeegSQA
W++OIKqHrg/WEh4yiuPfqmAvX2hZkPpivNJYdTPUXTSO7K459CyqbqG+sNo02kc1
nTXl85RHNrVKQK+LOYWYlQ+hWA==
-----END CERTIFICATE-----
```

4.4. Alice's Decryption Private Key Material

This private key material is used by Alice to decrypt messages.

-----BEGIN PRIVATE KEY-----

```
MIIE+gIBADANBgqhkiG9w0BAQEFAASCBCwggSjAgEAAoIBAQCalsn6i8Gi44/o
AVAn5Gnck4PHHNjrsfWUnnelN41KImVaTC3D9zFCrS3i4Pa9ZgHyA5Qf8JW3ZmnV
z5q7M8onZm7mZjqQeb6FUH4i2Gmt4jse2Dqs165ernT905NLFFlHUjURca3ynqEB
BV4DmhnZp8eDhv3t6dXyCjNHT82S6DgCREZuTtMc1zy++MxQlqdn9WZLhOAOpeNZ
KGmVwjeVy+8FkyzC3jX/Qcm+ZLCq1LqhBwDhdZ5qDTII2PVX1X3K7/cONxhvBbaU
l/k1swdszUtjhflyFZ80RuQ3qFC6vL/PGeWy6SCf58duq/AOEksCAW1b+MD8QH9Y
j7CFsmq1AgMBAAECggEADgxoWEDDRE5yEZ+s7TMw+WH2o+3X0OrryqnsLbOyv34I
wAAUWK7qZyjd9rSDOAtBOgFhQNXyHwZLT+0iHs1CIfqJMZ8wy1iFHBCIphoMSWs5
/D+idXrUef5Y23rClBxXH0glUnSGXnpUH4ehV6p1lvZMh40JKEoMC4cpydlSzxrw
+VGCC1+pXv/tTW3Rb2qoW09JoWY+Epcssrw5N8OFIFODh4QfbLN6pVtT28aQ4pf/
1KhLoapjFzXSyp/jrcNjYJ9qRdSAbZsKOJ2yZ0yqjLHDCDipFty+W0pkUZcJhsgu
CglStt7tKgSvAV/nejN8e/vA91/AACKBCNcLzEoLgQKBgQC4eTM6BDCz1usXJBK4
SRC/WwUthJZzzfOk2GmwrODCTRYhWQSDjBfiQNboazHobVPz45qP10fot2iPEHeX+
VWAXTNrN69M9LEzxygA3s761Ae jBR3fBLWkzLYqPB3oZwSIE7CrWHTXJipFWZv+X
FG1R418fnRCUMJ4j85qem5iyqQKBgQDWhQMJu7FC02fr83qsIdLwqhiDtTpwUN3j
qfp7JoEZOXbm3TgM1xPAkrQTUgfr2ZhXGtUwsuKH yi fXQEycrTkBOg0gqAfG0fnv
ybyXK6/guctHJQiy641L39kPuvQkKB+YO60B/oF6zbyFvqanoKXjpspObN3i3yBU
X5/EOu/LLQKBgQCUVvHWeWAgSg+pgBx9jGOnPK4hOCKznRJ7qyuo37Tv+E317lFf
vYFv1YSd4CJmmiUCkZTvK3FkL7HrFo/HwSeQFQE7aDkN8jX9bPPFv8K+UoNgkGp
LA8YVFrDQSPYadFNvYvsuXhzJLZSYGjPOGHgI5JufYLDZ4UDK/T97ekQYQKBgDDM
ORCxxvTYGiW2USVu3EkaqFDtnMmH27G6LNxuudc/dco2cFWbZ0bbGFN8yYiBCwJl
fDGDv7wb5FIgypqtn4lpvjHUHA6hX90gShT3TTTsZ0SjJJGgZEeV/2qyq+ZdF/
Ya+ecV26BzR1Vfuzs4jBnCuS4DaHgxcuWW2N6pZRAoGAWTovk3xdtE0TzvDerxUY
18hX+vWJGy7uZjeg14cFecSkOR4iekVxrEvEGhpNdeB2GqdLgp6Q6GPdalCG2wc4
7pojP/0inc4RtRRf3nZHaTy00bnSe/0y+t00UbkRmtXhnViVhCcOt6BUcsHupbu2
AduB72KLk+gvASDduatGjqqOzA5BgOrBgEEAZIIEggBMSswKQYJYIZIAWUDBAIC
BBwc90hJ90RfRmxCciUfX5a3f6Bpiz6Ys/Hugge/
```

-----END PRIVATE KEY-----

This secret key was generated using provable prime generation found in [FIPS186-4] using the seed 1cf74849f7445f466c4272251f5f96b77fa0698b3e98b3flee8207bf. This seed is the first 224 bits of the [SHA256] digest of the string draft-lamps-sample-certs-keygen.alice.encrypt.seed.

4.5. PKCS12 Object for Alice

This PKCS12 ([RFC7292]) object contains the same information as presented in Section 4.1, Section 4.2, Section 4.3, Section 4.4, and Section 3.3.

It is locked with the simple five-letter password `alice`.

-----BEGIN PKCS12-----

MIIX+AlBAzCCF8AGCSqGSIB3DQEHAAACCF7EEghetMIIXqTCCBI8GCSqGSIB3DQEH
BqCCBIAwggR8AgEAMIEEdQYJKoZIhvcNAQcBMBwGCiqGSIB3DQEMAQMwDgQIWQKs
PyUaB9YCAhTCgIIESCsrTOUTY394FyrjkeCBSVldw7I3o9oZN7N6Ux2KyIamsWiJ
77t7RL1/VsXsBLjVV8Sn5+/o3mFjr5NkyQbWuky33ySVy3HZUdZc2RTooyFedRi8
x82dzEaVmab7pW4zpoG/IVR6OTizcWJOooGoE0ORim6y2G+iRZ3ePBUq0+8eSNYW
+jIWov9abdFqj9j1bQKj/Hrdje2TCdl6a9sSlTFYvIxBWUdPlZDwvCQqwiCWmXeI
6T9EpZldksDjr5N+zFhSLORwABGRU8jXSU9AEsem9DFxoqZq8VsQcegQFY6aJcZO
Xel7IECIAgK8nZlKCTzyNVALxeFw0ijWnW4ltDaqcC6GepmuINiqqdD94YAOHxRl
1lKU4mLknSJ36W4T7vaI4fp98sK0nGpaDzQheu6BbQ+dVd44q52MDwvqvD0Y7UjF
IVEP3V9Ebf641CR0mIcVCUynxb3aaKjhqBKTGbySktPue974rDPIArMs2Heo8y3
cq+f7Jce0IVCglRatN6rSyJBF8JlBQW5pZGco8AwTMlpK3RrdIDziheA8DIBB+KT
4JZBO6UprlcZ5wBY6ncXW5E4feb57Cd3bB+zJuubBX9f4yG/J0cSF59w92c/6Qb
i4EFk6tAiz19PxuLLWjco71e69Jiav19Ph/WJpf/XCEurw7K+VAeZALFW41G/D30
WIBRC2shisHB3j8+3fNpCvi4Fy3EkZNW4lrZFAjBbtloCkx5rcfRS7vxucAvC5X9
4bm0xEcdOysnuplH77u+CWWxjCk414SlKZTUbwclA0B6yRDvojUMZkDzMQsxyYjn
JG5QhMFQrTyAlwCgJsP/rAf5xPhG2p+9Qul0yiBIIzWvKNKRQKL+YLcvYvTh1bhj
rUflYzzvviyXCY9LcX2GBop9yBFJzIcmKfL0MGua6WIkWX2BIjhGttu6VThmRHuf
OsqNg/ZrNCTYa7e1D6gWP5uFRecSZdAsf+OXTe6M7e/vaN4Go4A3H8+d53SYQP6n
pTt/a0DTHzY77aNMh+mzkIHC1W3zUdlS48tUyJMiAN3Tt+RfhHZfgloJ7IdcYdM2
01I+UD/5L9ghxN8dh13Fi3rDyn6Y5xB1xFuZ0mLjoEI+3Pr1+B9Kgf+o/hxFttfx
luPlXcHt0a4gBr6g7fWGNssfw5S6g6hS9UDTAYOpvLaatil2TZmeYZziJl9ssv36
kr1VaRV9xcQCbY05ucD+buymFXPn/rhVdxhgIydmvOtdzDozy0WFDTVgJUBNeRnC
eMVD6AlWdW0lMBqOcIlJS0aY2FWm8Kju62XZA8YIRow1Lysuq3zIqDmzmqJFKwuA
mRMZmUVhophMen86rwob3Z87gNbyy1U/dXi+s6Vybx/kiwDXjfyhWBnhnlgkhgiv
oOhtGtt+yAlICVuhQ1ElOQeQN04C5QTUOd1WOj489Ft6wpvm0tqc16NpnRYUhbCoF
XhFr4wswggR3BgkqhkiG9w0BBwagggRoMIEZAIBADCCBF0GCSqGSIB3DQEHATAc
BgoqhkiG9w0BDAEDMA4ECPoEFEHQGB9dAgIU5oCCBDAOrGHYn47xkt1J1VvWQZN
BYIMFzLN6p2/zKotGf7EMdgSdw1xkhKTWxunfoP/gfRD6boXTAA7ukJDsHXZrFXF
KjI4HI2oa/NihwqctphcLonBJXcofuHv+loP9MPLtwu3MolwsWTiHpf5XmxMoZQw
fbrp2ohLugJO1ZRB9RfAupaAhtFg9lpLotXEpz7GULEyOnYh9R8iu9bSel8bp14S
+AoxzXD4gYiEU6Yi0/47aRstd3H4u3ERDnUKSoqVstslRSKnK/WrGYUwoy7kNDwy
DBitfosMY0rpWEe5rXTBwJkBodcl3LBpDbNzdbrZw+e+yObJ9zfRlMpl0xVfoiJi
q9UbRdgN2yo0RKwF6c63V2RdF5tjQHnNIM3K3tC9zEis11jgn9LeOLB9Cd1qyE4P
WfmHN0gwqDFleX96TmUipmYM63H6jcbnSc6p7eIZtCrqGjhsTqFwcMg04WaxWeHD
ffLXSZdzIUB+zfc8tftUUEOUX3tX41loU7K8uAuQTSK/AXwUj+MbQVhlz8te4FVr
w4ulZ184IYqhD3VdIOxXiZkfSKChRz8/7QacrXFvfkKrcrxS2iHMoxhoJ7WETNtI
slW5R5runj61r50VT4HCFNFQfGBbTtV9AdP7yka9aQDWxPCoXfgeb1Q01F/BigzW
02JP5Lcrw7ia0y88QbTzWhi57d4he50Ip0wHUiGPh7s792ml1tvuSpRKJkOXWv6h
qAj5AsBB8JNvgXP71Ytx2vMdw6gqzQcxASJ4UHQg0CxmiodLUP+FHAY1CPNSjBR
pHrTilUFi/+9hYneQci++qPvkCqMuGHVxamd4OLanGJN1NxElDyMeduapX5rXuPn
g66LPey9GQuE3SBNC2dmjuOy7d8fWXEZqhqltPfsuwVzdnWbluAcjRfQPN0+uWe4
zihYisXK3lqA557dRqdSv+6GL6/OZQOCTaYMyZIWD9js2gU6T3q2j8uk1LncL9n8
aSpQ5xWspBXpZxo39fG6CMeqzZlFCqrVqWYhdXbtXn90x/pimmWolcqAxv+xythW
BMx+il1JEdbcj015wjmScWNPWlM4AVSholpZhs9Mq6rvqBXi1HJgjd0DpSLCE0xh
/GNoXoOX3LrxfCIDEhT8LyZ2NE59yh3t6pm88soFzaAghdjblFkc79nBbcl4NLKg
SmL/7GktkxEznOiSYfnfJ905kjZC08d8RnoGfrDDUWD2ZihbbxOCq4E3E0Zt13aH
JOXRBOZLC9L2JNeSNiBZZGykh+Pi4TsIzXL2UPQ+dy4DDaEf8yamyY04dlhFsnhd

qr94Y9E30/rpF0yUb2gCehEgT9nppVuMeridsCkHqemmgVr/52Xv/XK9dx4+YBjL
4/3Id0/yVJURqDIHH8o4ogF4rflkz0a1rZ9nJFugP0UM8oNysaL9yr7/Dli1juV0
MIIDZwYJKoZIhvcNAQcGoIIDWDCCA1QCAQAwwgNBNBgkqhkiG9w0BBwEwHAYKKoZI
hvcNAQwBAzAOBAidIqBxZFwvagiCFCKAggMgTzrUv4/12Jqnv3AL+P6990uX1ybZ
NcTwC+hMRV0Ho0FuAAybzdSRBAaZchl+8GheU8yz7IYWmLn1PNHx1Z8inIYfmTfk
Pa34Rk8s/RxJIE8LMYL1qjk/FMq/Fpgc0S65S6bXvJ69Hb8gtAoGW8P1b0dd9bvG
NbAk00h5r+IWiH4U8zGpcqWDWRgieGICsY00Hvx4KKMV6FIjFVCTZevORVoyzmSX
ZZgxqrbjw4CZqOWReHPi3aEt5xVX3BihRGi4Eiyia6yU10VOZTGBKqWUeKmOA5Gw
SX3mH/kLiya3gwwGvdq1ncXcl7V1STN1HFyp4ebGKg4CsZ6NkWjocwq2PwM/TqoZ
5i02tqvOeR8lX7LrSegxGH8lKw3nMV4dH5txoVt9hddZCKKGcJ5Z8FlzxFP4BFuF
7hOmRpUPdxiahJ/GkXDVIaw6BJKd4Q9e6sJjYxTeq4uOP6V4PMuDU7F98X/d9sEx
2X3blcJxua7xtOnKAPsWEyWBg98B+CKG6KwO5s8TlZVmlk15FCUjvFoKCiWIKF4N
vGLiWOIP/jj9N6Gqp4gNbm51zNFGZ7gZAtvsBSGQSOUPgfZcx2mRxpBmcX8tm5YJ
hmy9EDK13umUUGKrP0rG8c7/MVAQegSKqQuXSfMK6KknXGe7jwjs7xaQaRm9fFHS
0KbGU3MsLxRGjW/jzjUNAEWDiSYPCVo8E/kd8LEtvjAowF772y9o0X1ZzcP7HWcl
oYcO/WSSh4e+FAbgqLo/8KIkGzJ23BAcdx8XAtxzUZhRdHaIttnwaJsTr4TCwq8C
XxJG5u44/z6imqQrVOaXQfvk6sSNGdG62TkacYg2K63D9hcg+TbZPPVSSStWXyJ8S
N84anzTOxblyx6aw6IL+uBLC4jISgNFijaF5pwjLSbgTs5Z7skZdCam80xYmdJVO
ES/uqFCQFUSamXXNbotviQk8jWuJFz+BXzPYJN3t+3mp6SmgTZ2zP8FUQEE4GbSH
DqYV621DcWRo/mao8xxX/mvkKm4ddGBldiusoHZaL4gdo2A1qThSMnMBSciC+JEj
DqOr70XhHccTDW8wggWUBGkqhkiG9w0BBwGgggWFBIIFgTCCBX0wggV5BgsqhkIG
9w0BDAoBAqCCBSYwggUiMBWGciqGSib3DQEAMQmWdGQIehcRLmVUApMCAhQOBIIF
AHb5dXZKzCeRUo2ZSj0oyuFS3zQ5HhKyfapsyCqbYCKv/lSzNYWvuda7xafa+uOM7
/wCB9sWdz0MTpaBMHwx9hvibZiY65oM+ry4tTuKKQOJl37OsnjB0dSNTKszsI3fa
PUjslXqIH3aClshD7OqhIRGZzRjK44PjYwV626oQrgVtTYR9NYTdee+SbBzBkEt/
EpWipftWXGR6tSYJQn99eO9Vih8HyQvwIpidUh3pCF0low4VZyAqIWOHcw9TAjB
XNv+qfdH7fiX9wM5/GvnQReIsqjXCuoc6pSQIAqD/f+I/dlF2ZmqM7KwX0LGRER9
OWZGyF734pN9GLbNetWm6rKxmlSI/5m6+2Jxxfann16P+vBSEgWJ/I8GnJAdzIbB
Tyfjog4Gi2+lmrPzK7+C79ntM9nfsr4xVzy/BknwZiaJksd4VvOGKS9nfm6shtBJ
B9uR+GJfthtsvIVUHN0kz2r/lVzMSRbOg9yR53hv1H/nXCmUjWz/BvobmoaVBCm
mOnnYZTHMNarIVYdLQFif5ZLH7WV/XVEVioRntNRiKsK96VAHm5XboWQGCqL0heh
IX3NilylgenGmlaF1SQNMvLDko1ILDTKrINvPmjG/WFoLntPjFPtYZsooT1jjXLw
3VTSodtgKQNdPYOEidSJqWIS87fzrCB2Wmwys0iGfdsuNhSaqNqa0dMO6FiW2fku
x7H+w7SX1/n9YeZUNLOcewLcC7E8IA1IarjglZE1L6Yb2ldXxV9q3PPowKuGnah0
TKnD6mLn5BIGOGTzF1VspXRrJhFrcLe+xsJR1r6niI3bcMWXXy7gbm1X/CRE902I
ynxEloDR+xZ6rjPwDJP7kVf4GvA8trCGrot4pbJbmwlBeMIylScdQoHENyqrenOn
RMmXZaKz13njtq7Wk78qoJq0a6Vh/sde0KcOPFkyTZdMB1Tztm0K2VJU3jUVzPlM
0WY2fyGD0A89o1+/MiNsgiaEghGybXBYipOex+p7j1GIRN/CKmpWsqjZnB78kyXm
Z6AE1vC6neD/7zANInDkzXiun6ic72LoBX3JGiCSuM6hIPJ0AcDwlzTDu0H2rCQN
w+tiVJ2v4KbgeKoc6beQb5fZHS7VsWHikIcpwqB5ngwt34wHgFG0nTS41ZmvzSJ7
FMRVGmsDYkDTpZzgNOaxiUBQMCEvxNIE3nAmA+dvB7w6XRQVSUsL+vBFhHiWGZ7h
k5sCeHElewXK0SyJADgfflYq3EfEgZ13h4wtoSfbBVtzbbyg2LNegUCLfIJKc7fm
T7X7JSxbjOgndMHEeMdVb+NFxbgsXYrYD8rC2A815cQzZrsxblbvgybEJz+NU/52
UgGrPmdjJKuGBK/V2zor6qPvKyId1Gb4QQuIoyClwhZ+qk9nE4Eft84y7ISgMywH
+lw87HrSHKfppqzQhCx1rLu53IYK/4PhE7BYC9Q4tvIsZXSGZ+nju4tyzERSlaNe5
njUeIENr4B/+kXULwVdcvMFHqUFJmKfai8FUga7gyipZ+654clGgJjnNB01va8Jc
dtdPRRW4gwdrVn8u8J78KBzt6ChkrpKRV8VeWKBk91hcT0ZNpJnNqhDrkfzHBqP0
Uo133I7P7C+h9sNDI153W6IOIodyQE0Av1WxHo4y/ld1VeGDab7hOSDq9ZMpm9n1

```

En7F6/1/s4IUZHja/qRrK9hD4M0Xq0LhFXuUzuipo49OMUAWGQYJKoZIhvcNAQkU
MQWeCgBhAGwAaQBjAGUwIwYJKoZIhvcNAQkVMRYEFKJTQdVEPIApFXwBI/Dnjq/N
83cPMIIFlAYJKoZIhvcNAQcBoIIFhQSCBYEwggV9MIIFeQYLKoZIhvcNAQwKAQKg
ggUmMIIFIjAcBgoqhkiG9w0BDAEDMA4ECKq4DtyiayOyAgIUUpQSCBQAKQtKPOS4s
LE6Os7nP4RaJWBuYXl27V/o6TusBRBqQoPzP+aC+O99wgisEKedyB47bAzC04sba
4q8UkERAsYHcEhdD2hGRCL7ou9jTtrr4RgZpa5V9CJCBO0t4bqy2lUefOpm6no+R
X840uyM4q5Q+cfHlrTQla/a+gLglbptoEkH/4dFR3ELYiXcM5UrBYTJOHcyME8c+
TXbpf7kip1TtLsr1ZyU5zrWcxngrBxwFA+O85W/uVR3QZSW+EGx/VCYwGruZ1Nyt
BvBYjsYsnC+yKYXbqL81DgOePy+eh6VX64SwBLXcWcY+NK2EZrhzrUFjl+PXFYK3
IVVPJhTE9o7gJA0hzvAanOluWXozD3/WPQaXhyIJDwM2MjznjL2MBydpy9K8Cio7
XaV6PX8DsZIZkfI4DAz5f7G7WbwUq3IjPPPWiuV+JsR+dnqzWDJ22SXc+AdQP2sK
qmVp8gOpHOSvLXXE76c5rUcZCZD+gGv1avO7YttWqbDqLj6oQEIJ8LX0Qvwd0YEH
etE0bJ5uv2njhQDhLkH/JIbmFSgJZeM8dtKHb8f5wZc2B+nXGB+TFboGzSuP7gaW
u1vKsJNqT/J/FYEgcami2F+td7z1sGfbR9ckAcxXeb2uPVbCJla50gR1z9qVm5Hb
5f53X7aoQQp3F3LDGQmJ+GFQ/oXXwabqn4TvNO9KDhxpGcMMU9RnugUfNU9GBec0
vfrzmVKZdmJ36HOMMnLvqRakRhCV3kGABXY83hwUv17E1qASLKcAWIachkCCGpBG
yGtP2IOZTn7PsLJR1BzKnePa7MgFcgOCToIpdQnCTtAsalmBmls480LN3GB5ojeG
bQvNf9TAvia0tg5VuT4/O48V6uYSJsIZsawm3tGA/LjxyfV1aLddQT5Zf5ZX9BX+
K/PB4oYAFxtUpMK/aL5G1MvppUJ9CjqAtnoKE+EkdQmyZ1VoDO9ih44zuRx6XV4A
EYafNB8ygjRHGsVPW0/M0Es0w16wzJHTuf/15fD/nH7Xh5MzhCF0CtvLn8v+S1Po
i2/4006pS2byjUFRbeCpzEpRxdv90Lcb9ALdy0yG9u41W3yInKNFnaWBulfoPFCE
ZT92M1BgwJA8ZcydtiunRNAH5iWLSPLoUpODlv6En+rat+PoyRXIy2fLHBL25aw
LhABoZPgRsCiLsiNiohfnyngksrQKeRgOlaBMT92J8r1E4sUKirQlcOdiWBE6vmBS
XzyN/twvfgPNIXgR0rw6c7VhhS+hNTrsttg/xcfvJ/bftDbKm+RZL+yQoOkkAf9R
5tizyMdBlaMrpfrBxvNtMiykbZ88SYoA70Trwab2aHqLuVhs8OjXGBEOqmSudcS
dV1EhBpo9HBsDZZi0IwOp5/B9fCHdnThCTiUm80eQ6mX2/DB9L1Nh7gHOyLL3azT
m12D0ZpZNaXyxLzdiRiAdwpWZmmegOOG70yi0D5eIhx6cbnBU6Ygdp+pFFVYHfA
vc5Czrne2OPhXX2k00kbwawr9AfrFjIfAEbBFx5GBGr/lSiUQSkbUC/s209YgaOg
WTYt3KXPzrThJJGZnnXZRTGfIi6vp8RsnPX35+Dxe/Lp3gXDdIJeWG6XVA8t3fsp
coTqPkm/XGNMmOZ81KX/ReVdP+dC93sov2DuDZbYGPmH1D47bOOiA68GD64DeuNt
Q8MhWk8VRR1FqcuwB0T0bc+SIKEInkvYmDFAMBkGCSqGSib3DQEJFDEMhgoAYQBs
AGkAYwB1MCMGCSqGSib3DQEJFTEWBBs79syyLR0GEhyXrilqkBDTIGZmczAvMB8w
BwYFKw4DAhOEF0/nnMx9hiloZOS+JkJAu+H3/jPzBAj1OQCGvaJQwQICKAA=
-----END PKCS12-----

```

5. Bob's Sample

Bob has the following information:

- * Name: Bob Babbage
- * E-mail Address: bob@smime.example

5.1. Bob's Signature Verification End-Entity Certificate

This certificate is used for verification of signatures made by Bob.

```
-----BEGIN CERTIFICATE-----
MIIDYjCCArKgAwIBAgITaqOkD33fBy/kGaVsmPv8LghbwzANBgkqhkiG9w0BAQOF
ADBVMQ0wCwYDVQQKEwRJRVRGMREwDwYDVQQLEwhMQU1QUyBXRzExMC8GA1UEAxMo
U2FtcGx1IEExBTBVTIFJTQSBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eTagFw0xOTEx
MjAwNjU0MThaGA8yMDUyMDkyNzA2NTQxOFowODENMAsGA1UEChMESUVURjERMA8G
A1UECxMITEFNUFMgV0cxFDASBgNVBAMTC0JvYiBCYWJiYWdlMIIBIjANBgkqhkiG
9w0BAQEFAAOCAQ8AMIIBCgKCAQEASnAF0glRof9NjBKke6g+7RLrOgRfwQjch+2z
m0Af67FJRNrEwTuOutlWamUA3p9+wb7XqizVHOQhVesjwgp8Pjpo8Adm8ar84d2t
tey1OVdxaCJuNe7SJjfrwShB6NvAm7S8CDG3+EapK09fzn2pWwaREQ6twWtHi1QT
51PduRtiQ1oqsuJk8LBDgUMZlKUsaXfF8GKzJlGuaLRl5/3Kfr9+b6VkCDuxTZYL
Zxt6+a3/QkaC3I9m2ygPubtHFJB5P5+s8boROSKm1OB1gsLow8eF9S7OtcGGeooZ
JiJUQCR14NaU5bIyfkEZV2YStXwdztoEJJ2fRURIK+8Ynw1B3QIDAQABo4GtMIGq
MAwGA1UdEwEB/wQCMAAwFwYDVR0gBBAwDjAMBgpghkgBZQMCATAMBwGA1UdEQQV
MBOBEWJvYkYBzWltZS5leGFtcGx1MBMGA1UdJQQMMAoGCCsGAQUFBwMEMA4GA1Ud
DwEB/wQEAwIGwDAdBgNVHQ4EFgQUF8WEe9Cn73aQOLizbwi8krWeK5QwHwYDVR0j
BBGwFoAUkTCOfAcXDKfxCSHlNhpnHGh29FkwDQYJKoZIhvcNAQENBQADggEBAG7e
QY6Px7WZC5vCbF5hjOitxoz3oyM+LRcSTGWOYXdm1wsNUzy3lpE3dtADvevRtsP8
uN7xyfK6XZBzhShA/BtkkqYGiFvXDpluOxWmqC0WPmc1PNK2mHil+pGMfvnUwnxd
6gKcHED5p+bUhDyIH2fy9hGyeOUs8nvi+7/HwBipN+nA/PfsPn+aU411K6qDoG/i
kwyuiWcFFlc5yE5rkAe2J0/a4+HtzNmTK4jB/4GbyI6x1UszPlEqKE+Es10Xut/y
UWL5nKKaqpRRd0Pq371MpFQs2+zXt4fGheKzZU3XXrIPcAPyJjWiyU1DzpqgSJM
OIp/HtXdFscHb9+Qic8=
-----END CERTIFICATE-----
```

5.2. Bob's Signing Private Key Material

This private key material is used by Bob to create signatures.

-----BEGIN PRIVATE KEY-----

```
MIIE+wIBADANBgkqhkiG9w0BAQEFAASCBCBgwggSkAgEAAoIBAQDmcAXSCVGH/02M
EqR7qD7tEus6BF/BCNwf7bObQB/rsU1E2sTBO4662VZqZQDen37BvteqLNUc5CFV
6yPCCnw8mmjwB2bxqvzh3a217LU5V3FoIm417tImN+vBKEHo28CbtLwIMbf4RqmQ
71/OfalbBpERDq3Ba0eLVBPnU925G2JDWiQy4mTwsEOBQxmUpSxpd8XwYrMmUa5o
tGXn/cp+v35vpWQIO7FNlgtN3r5rf9CRoLcJ2bbKA+5u0cUkHk/n6zxuHE5IqbU
4HWCWujDx4X1Ls61wYZ6ihkmIlRAJHXglpTlsjJ8oRlXZhK1fB3O2gQknZ9FREgr
7xifCUHdAgMBAAECggEABcQglfTtieZ+O/aNdU149NK0qx97GLTBjIguQEDDBVFK
2lu4PhBg9AdgAUqLH1PE+eq65JaGZwvFH8X1Ms2AKiRzYsPOQIoJ4n1hc69uiEN9
Ykcv4QH0vvtCtWYjJyb5By9WPeLH6QynJ6FlBoSqxhURSWyYfTuwqt1OHEhsUuH
d3N5BmbFiRBNj4aIA9zz+i5xL0m33kMKai/Ajj3sIOAJsZ5ZVAhYbC8sCt1Xevb6
i4lp9S6GSwGC19by+ly9WC1QGtb5GDotvChMvmZS/O3NeDc6xC/LZoQcHNvGiZd7
f1g6iEkJlCYK+D7xsd7Y630w75Haj0vn1hiJObSA+wKBgQDxv8jp2D6IVRGgYfaC
nUU3Mg70wagX1fgPHO9Sk6e9c8CgORh2uwWjpTawu88xBGFyZ+xnWqr7GCNsltas
3m94ri4A4R94+5uL8+oOLC26gMDfzAtD1Q3k/h919YLk89tonQEUBCFZJdphThEb
vg2W+nNsEVcQGUC1zhX0AyGMswKBgQD0BYk3sdGQbBA/hYD1EYsZfYebUiYv21Tt
VGRgTohKfclRAWotGP9YRbKyEVkBLhjgkXzS9xGqKywP71z9Iny+zDGBzk8ElB/g
1S7FGGX50TG0ISfaFTYdxt4mN9pduZE2b1T/26uyU8DXCEBhF/OqhwQjJqKTYTT
Rl3Ara5fLwKBgQDQyVtjIyD2q8naY2D8c4mo3vHtzyc21tQzcUD8Z4vSYps1hbos
KN/48qJmRv3tjqP+o+SXasYKsFE/4pIroLxTVNNkbQm6ektfttwp0lyPG8340wLk
97HVW0ig/tX6mOWg1yBsm+q9TKTrrvmlpRGlme6BQgSYy4r504u3VlnYwKBgQC1
B4FvWyDhTVQHwaAfHUG3av/k+T++KSg6gVKJf1Nw1x8ZW5kvnbcJC3pAlgTnyZFyK
s5n5iwI1VZEtdbKTt1kqKCP8tqAV9p9AYWQKrgzxUJsOuUWcZc+X3aWEf87IIPNE
iQKfXiZaQuZ23T2tKvsoZz8nqg9x7U8hG3uYLV26HQBKGC0J/C21yW25NwZ5Fudh
PsQmVH7+YydJaLzHS/c7PrOgQFRMdejvAku/eYJbKbUv7qsJFIG4i/IG0CfVmu/B
ax5fbfYZtoB/0zxWaLkIEStVWaKrSKRdTrNzTAOreeJKsY4RNp6rvmpgojbmIGA1
Tg8Mup0xQ8F4d28rtUeynHxzoDswOQYKKwYBBAGSCBIIATErMCKGCWCGSAFlAwQC
AgQc9K+qy7VHPzYOBqwy4AGI/kFzrhXJm88EOouPbg==
```

-----END PRIVATE KEY-----

This secret key was generated using provable prime generation found in [FIPS186-4] using the seed f4afaacbb5473f360e06ac32e00188fe4173ae15c99bcf043a8b8f6e. This seed is the first 224 bits of the [SHA256] digest of the string draft-lamps-sample-certs-keygen.bob.sign.seed.

5.3. Bob's Encryption End-Entity Certificate

This certificate is used to encrypt messages to Bob.

```
-----BEGIN CERTIFICATE-----
MIIDYjCCArKgAwIBAgITMHxHQA+GJJocYtLrgy+WwNeGlDANBgkqhkiG9w0BAQ0F
ADBVMQ0wCwYDVQQKEwRJRVRGMREwDwYDVQQLEwhMQU1QUyBXRzExMC8GA1UEAxMo
U2FtcGx1IEExBTBVTIFJTQSBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eTagFw0xOTEx
MjAwNjU0MThaGA8yMDUyMDkyNzA2NTQxOFowODENMAsGA1UEChMESUVURjERMA8G
A1UECxMITEFNUFMgV0cxFDASBgNVBAMTC0JvYiBCYWJiYWdlMIIBIjANBgkqhkiG
9w0BAQEFAAOCAQ8AMIIBCgKCAQEAgHAlBNMiBik8iJqwHk/yDoFWwj8P9Z1uYdq
1aqIuofvj0AyjdA8TbsBRGdmvaIOSQOepsNjW1ko71E8H1Ds9JHn1E+tzH3mKfn+
G2erY+alkMJTXPvMAUdCA8+e10J7k91gYXDpzIWrp3Kc0xTlsJ8tGJ6mhydJX3wP
0/HuyHpfKQQfDusPH8S5yidPciWuB7Wj0X4xY1pUAz2rSSAlnGvhEzKFbW43BPjY
XPUnRWMtXFyaldjq6Eb9M/klbhdZheDLLsJLUSXYU70r9VXGM/qcjd/NhWYphCeB
cqswaM5mXLYdm0mFmqoecF62mUE0DiNdhwKTtnefd0c1l+D3FQIDAQABo4GtMIGq
MAwGA1UdEwEB/wQCMAAwFwYDVR0gBBAwDjAMBgpghkgBZQMCATABMBwGA1UdEQQV
MBOBEWJvYkYBzW1tZS51eGFtcGx1MBMGA1UdJQQMMAoGCCsGAQUFBwMEMA4GA1Ud
DwEB/wQEAwIFIDAdBgNVHQ4EFgQUSrOsMVMCSZxN42554CVhlT6IYiUwHwYDVR0j
BBGwFoAUKTCOfAcXDKfxCSHlNhpnHGh29FkwDQYJKoZIhvcNAQENBQADggEBAC2c
Y8FgaxgB+Dx9gAFj35aelvgzYiWI3Ax3FSxogo/GzpK//LB4215oeBuKXbm0ixBn
4nojxD7Pm1M0i+ilAvVNJNaHY9TtgIgq8V/C0C7vL8SdBN01e5ZRI764ohu9ivYv
Ixvvt7gzvSTpe+NUTli09xNgsC8v19WB/BwkqMagDqMxqCxT4fyrVwpxNBke75j
E6Q3xCjfdOWYcfMLK7EsTSgimYuonZjN7v/yqTdjn/iVH+agL/2M1SfiU36w/Yf1
7EM09uKGH/Javh+2Vjd0j8rE/q2Iaac5VI91M6xz5oDZUknycBKKinR+nJWMt5AK
UAaL2Mjl3YtrUGBpxxY=
-----END CERTIFICATE-----
```

5.4. Bob's Decryption Private Key Material

This private key material is used by Bob to decrypt messages.


```
-----BEGIN PRIVATE KEY-----
MIIE/AIBADANBgqhkiG9w0BAQEFAASCBBKkwggS1AgEAAoIBAQCq0cCUE0yIEiTy
ImrAeT/IOgVbCPw/lnW5h2rVqoi6h++OgDKN0DxNuwFEZ2a9og5JA56mw2NbWSju
UTweUOz0kefUT63MfeYp+f4bZ6tj5qWQw1Nc+8wBR0IDz57U4nuT3WBhcOnMhas/
cpzTFOWwny0YnqaHJ0lffa/T8e7Iel8pBB8O6w8fxLnKJ09yJa4HtaPRfjFjWlQD
PatJICWca+ETMoVtbjce+Nhc9SdFYylcXJrV2OroRv0z+SVuF1mF4MsuyMtRJdhT
vSv1VcYz+pyN382FZimEJ4FyqzBozmZcth2bSYWagh5wXraZQTQOI12HApO2d593
RyWX4PcVAgMBAAECggEAEvPt6aAQjEJzHfiKnqt1U7p4UKb5Ef4yFrE7PdTLkeK2
RjncIhb6MeevVs8gO6co7Zn8tuUT95U3cOXLhVOWTvaHYeurTXaknICz3IeOoS18
skiVZko70uJ8pR6asWUlr/zOjLEwZ7RnEUWet97oM0YeA07LDFDkF7eUq//6bfzT
ewr/QfDDsv+erwJBh+9CRHOJyTuDH1WeGxYV8VK3M6VhdTjFxxXfhrQ4pBe5J/UA
17Bd2GM8Urg6VYzVo6x4ajnc1H/ezYLdc459poTffv6Fg2trqFVAj2IrQ1Aeqjda
lemsa6Np801mUgknq3fjKS13RYGBv/48rCHOT8eRgQKBgQDM5TuS4ANQjOYoOgtF
xoVjbVlndOo+SmdFkZihzQHxchLY9HXe5H1bLflIMXz/nERx1+SmYuuJk0EdiM9r
HOCcHRLfBmC7t0GdVvLDHSAX8Ec47LbtKZqyM1U9dn7Z+5q4iywqpaP8pP3+oY57
cgtQax1jle3xhRAj65c1lRBmQQKBgQDVbLqK6wKdfSdZuMZGUtOY0rtamBDCgEU6
rEqBAYCPy5NpFlpomUFcYKWT/wbReFqtuyq2OyiATB0yHHMko46BUtN7qX/m/skt
DHWXVWs1+G4IgEMVokM9jjrkgdY5grrJ68sagKC+bgv35BizHP IggQuO6qnPSrM9
bevwBQEj1QKBgQCiPE/zeBSnzyjeaTdLxGkR1R+ZX2WqdNdYqnQkiWMkf1aSmt5J
4raEj+GhLC5BZsZ6+z480M6XXFWOwSkbMv5WH1824KHvgKcf0h00iR1EVyjn1gDx
wKOQvjjycMhs3FpXn0arjCcZS2wGSgPGEpUR4JJhpcfaF6kphZsWDWzV1AQKBgQC2
ivbKltNhj4w2q1m7EGC3F5bz15jOI1QTKQXYbspM8zwz6KuFR3+1+Wvlt30ncJ9u
dOXFU7gCdBeMotTBA7uBVUxZOtKQy19bTorNU1wNn1zNnJbETDLi1WH9zCdkrTIC
PtFK67WQ6yMFdWzClgEy5YjzRjbTe/rukbp5weHluQKBgQC+WfachEmQ3NcxSjBR
kUxCcida8REewWh4AldU8U0gFcFxF6YwQI8I7ujtnCK2RKTECG9HCyaDXgMwfArV
zf17a9xDJL2LQKrJ9ATeSo34o9zIkpbJL0NCHHocOqYdHU+VO2ZE4Gu8DKk3siVH
XAaJ/RJSEqAIMOgwfGuHohhto6A7MDkGCisGAQQBkggSCAExKzApBg1ghkgBZQME
AgIEHJjImYZS1Ykp6InjQZ87/Q7f4KyhXaMGDe34oeg=
-----END PRIVATE KEY-----
```

This secret key was generated using provable prime generation found in [FIPS186-4] using the seed 98c8998652958929e889e3419f3bfd0edfe0aca15da3060dedf8ale8. This seed is the first 224 bits of the [SHA256] digest of the string draft-lamps-sample-certs-keygen.bob.encrypt.seed.

5.5. PKCS12 Object for Bob

This PKCS12 ([RFC7292]) object contains the same information as presented in Section 5.1, Section 5.2, Section 5.3, Section 5.4, and Section 3.3.

It is locked with the simple three-letter password bob.

-----BEGIN PKCS12-----

MIIX6AIBAzCCF7AGCSqGSIB3DQEHAAACCF6EEghedMIIXmTCCBIcGCSqGSIB3DQEH
BqCCBHgwggR0AgEAMIEEbQYJKoZIhvcNAQcBMBwGCiqGSIB3DQEMAQMwDgQIe/d6
qDQ/28QCAhQGgIIEQJKA5kzRVm9d6rEwC/0RyBSgpPuSROUQTjspt6EhBZlgHc3u
FTCPaO5P/vpeWacnBRarGFN3DmqA3JT+59bmRpGdiP3Zrlk2EbHi0yrd2P3UFDnX
qRkKI+7pf6eOHWJRntJA+KJS8v3tZ/hpiEKAeav/Mq0IFNFyEiZpCkbKCX5auDb1
p5c3J2MNng/WNBfpGUHkVIZuIF3H+8LffgayRsDsppoUMffR+GmdL8nxLiqhraHD
+Iqr3LpEroNi/iZQWUTFTUlaePf/2KMqaHOuy41IVvcH1jIcLXHGNaa66S8AP/Hj2
TJPPg/lve76DvaGdEnx4QJd4pBFQac90zmhxU1HZrvzubK9t4e5lr80wpd2djvZK
wSLzUgtQZXq8pSslr85vrb3KItDYGF6SZpX029FS7rY3uYth5SYVUQWdUYYY3S0/
nsaLg4MCWUO4Sh7nYJZ15Ijkk9LS7JhmwKvzHRRTXbLyRDH06e+jCRgLCu2WSUq
1bEr9Jy0ucK8zNPTf8HWBTS0ubvy4JfO3mVp4REX/8ozX1LztWGb1FGbyaJ9Y4ga
LM3JpKxMtb1UTxoAyj3iFwGLGZFGKB1Wp1r+OdkKkC4dloFE22IINfLdRNLV9mPO
aGZhsDheB8iVotN0lu91BLU68Q7AL1ryXWUSjouKGRSU6uMDLZ7rw0w1ZC1m4oLG
BF8Cm04ELmbOci78fBs/qDX1f3BJazcNtciamEsQPYRGkHASBRYtoDfVy6mTT40o
obdrZigcvCwttdBu7RtynAQVZ8DvKzxFGhe2p2Yc9H5A5ML7IwqNtYzheduBAQTE
jAU2jMqwnZN5wULEnH2TF6KAQNrKdtBYMbqkToKgx5Zf+cJZbyQq7WM6nVfOM7g
kcFdeHDn/CWoSNHI1+JA3wSDM06zkU5HMD2MpT1RLTSaemImUKCAGYieJmwNQxR9
aYHBBw5BNBw1XRB7WRka2Uah0Xq/wAgAI/o9L+mShDRFJjFi+8AV3KR0WWHg02O
9qchX7P5H3Sy/tq8yUQIo1+hRiRjkfi9qy6AxIRttrK4WbW4scUtBZSk9uFkTVU
ybnV6WvBpn2SrnwF/ElueKARVmouWJ/7fiLJXk6wVvVtuBZw2gE5QGfuCwq0PQsC
xPx8MhN1kZYDVCgsyUr/LMHeKNc31S2HLGQK7kh/o+QQazafiJocQ+kRbS1VX1D
nQ1Ihz4zvKsBgZHpoe3wQcfAY5sp2ubepsZ5T/YHkmroBmvA4glvi7nlCetgxXrh
2V6OXvaZ+BnfsYxJeUZGnNMNEDF1zS7xB18ojtT5JN0o+9tLsdikdikl69IsVv+2
eCv9Go+whl9cSAL24rkzdKVuiIAXS7tzel3eWGjdKq3Ke+tfJtobSGrB39xgLVr
3ho63hd+qTUyjcAhVL3hAJinv+/KT0JlR8fq+CDsXMnCEWugHhwB+66N0r876MIE
bwYJKoZIhvcNAQcGoIIEYDCCBFwCAQAwwgRVBqkqhkiG9w0BBwEwHAYKKoZIhvcN
AQwBAZAQBAjiGuDSkfG4UwICFLWAaggQogyL08hPtU152dkO+BVimcGXW3FmDrT0D
gU3Drd0P76KzYzd2lLuGb9dx84wx0XnFIXeBM4F3QSDbCK4tOuJ6JRaEeUoCAYzd
XyHtLjVeuozt2xHBDUgQVE0ldZHtk1VUGzLSchalrXjcwpa4+8xqqoVM3C15uBh6
QLUNey8Z3YlK1k018Tdge60OUrg72BPKppNfJlN4TnOfwMVMA/qHAJl4pL1YDpmc
5BZm4tMg0HvPiz96uwjEhw1GZFGOGZIOgeVJuqCNIzPDjCFEDgnCw6sciS5Bi+dX
Km0VUdamSr93e2eEPLbzxZR0E0A3IcoJ66iHuZpU9YhKzsAIhLMxT8kf81I0ZZzj
8N+PlhNkjdvWuJLg77pkXxQJyvuT0e2oc9r/DCHjckneen3+E66IKsYbib7sX4g6
2oFBJS+7xQopy69pC8jCn3fx61t7AFx2RiVuvHY/eU4sXoWkJNqQ3Vxj2SPWKjzJ
4IiVWvXiFiQjJotDfDGYPGukJXn62Lbb8CFgam9s4jDKnr0LHIngVeUIgi4wkvvA
QzZTzXfUApezQgQy4x+ogdiYF1U0a0OaqvrGRiiJlMdRi0/MDy+jzkX5cULhxkF
vdBNCirv+3zBaiJ5Eu6q0zP5Cxi2qXhSbehZqvTPB4dD/vu9yxHpZmUCvzm7H213
Tdrb9WxH0c92ZpBzsfica1smVwTDFVga/kqN6noPw0qWZANik27/+apsTkBYaVpa
jpfN9eydi5eV2+pEQV08fh40JfIKbHS012E3Gp/rPm91VgmCmjBWh+Dilk4qgF/f
lsxWgzXNOxPntpohnM6AZDxW9Sk+BELDLYS4WFwUg679BsJG6hQqAZKvG/8agSH2
k+TKKYUbXbFVCB0+iuNZIwgf4qxGzvI5+Iok+OcxuGCqWou30QbfECEG01QbKETn
ic3kMiZ5Cxt7NQsuyEYAQ/AmvM4qo0x7Tw1r7tR8BcAEF6fGxd2VXIV8Tr/pXGO2
HL+0iIHs+Ob67z1Thr7wUB4tCp9LC3IIWdsr7KcSRNEMxpUIFI0etCjNgCU3iT+R
9152150fWNGxQfaXTEyMVNaT1HpwihIisSb9QHbagaRLbYmqJ+ILSECADYQPEWf+
LT01tcOhkIb6BiwVWUu0OqNj6ILJM2XvmknATyUj9MYcd77xOJzMrJE5VtaM5BVT
oRpcOLfhYOmihceGSEqXX5golkqfLUze7zls1NWMYTTLw6tC6I+c/IUIWJnZT4m2
RbTQ0krfPn94zbTjrg42HS5+Ke3ySV6Fv8MZ+s93yY1v9iB6cVPEUteLRc+C7e7t

lw0bQ2+MyAkjenS5Td+3tC7lR4202CSFY2SaOsRv+EaYjTGzf9F3TM706o5+VZrM
gtIKtw2okRcjRhaKDFhui6jo46YYzWbrgOS3vzc60VcwggnNbqkqhkiG9w0BBwag
ggNYMIIDVAIBADCCA00GCSqGSIB3DQEHATAcBgoqhkiG9w0BDAEDMA4ECEyHXPVs
ncxTAgIUQ4CCAYDSBLYeFnsa4vtKApbLnd9FENDYeYqkKmj0lkDagMqHC22/nQ9v
gz2lOo5FQJoaJx/WSorQt0JnylQP9vZd2t+bkfoaXOR0MtmFY5SotYEudJplrCz+
ZEw8JlePJRP0Q3lnwEiSk5NnXLRWNzurIeuyZEdlVbTvi/rF22sRWlmU335L67zj
PlsPeXkBPiYCPHw8E4rkaC8G1ko5wyrnhuqL4ItzhvOORvgRaDflpP9WTj9LVUv
FD5D59zgbOptaW0jIw4Jp1IGXIEZlYnW4KfKWy2YJvsXiuLHvN3Z8qL6VtxNGk1s
g340uKkUuLzmtDJqGT9RVkoYBXxN7KYesbSttONhPwcv/MxHrEo8TGHZAvbmwgft
hOUrc/WVtUopPEs4QgrsA8d0MrSd5lVtPW0XPsbPEnlLuh7dqAlmgztYlP4Yztk2/
JJ+E4MosmhrjBkZm2N5WuGLDC5m9KF/5JjNVwQ7e8gMeUv/3gizgCG/4Mgng0VGG
IxGzzBoQXPWCKdT3sLQVyt4/pqPBpZYNP09bmkkY/UIalunNB+WWpLokKSzD5wRv/
2xmNO2D37DnHwTFYC5lZblKz7FGjOgCwG95Vpc8NQ8aG5rqpQ+muq/Jil5mXgNw
IDeM4bawa01UKEZzTGQUb3gsJMGiVOhgtOrBi09Kx/2PJo1UuwZGcho4oGSVR7KH
lLgIuC8aIQDyFURVYRCNwOw5U7JN5arkvZ4ty0/qk5UbJxQuDkF8o6ZdVi03l0Do
C+6zvncDx4HvUd6uQ+u/kZfr8qfwM5o6D2qXhS/ZHskq2xwIzb47uUuaeg3yOZJ
++na7gC+ibthHXNnSHUvPbpCn9qViFhZilcQZYq0tZxDKa0E/pzEP/IA4IG24wEL
GnyuUIHXBS9T0MchTx17BglycOPRDNFKzMQfUXYlRAErK76cs3y4VQDbfYDiOzsa
lqqMapiX4i/qKFdRvDuLxtZQbVA/rNumm40LPUQ5OvEngIESA74G+//YQbVjbmJP
y+hm7/15q5LRO9YxCS49KGLz4NG1QMWjnfkPOCNVZVpaQ7TPGOIYZBL6kTCCBZgG
CSqGSIB3DQEHAAcCBYkEggWFMIIFgTCCBX0GCyqGSIB3DQEMCgECOIIFLjCCBSow
HAYKKoZIHvCNAQwBAZA0BAIo/0ICbTbZLQICFOWEggUIFWT/JI8UjJQPfYTFonJE
o8zEbpYWXkboqw6/zZsMGmAnUPgQNQDxyuLVprS5jUc437kVB2M3F0x8DjmEpeb
tHfIoyjoXF7jdnA4EF38tss0K1nMPmSgl02iYztOqsOvBpfe05Hj4Ovhi26J9Pz
TwPcgl3QQPqfWv7CwgGVn4/hntBARIpSE4gAlfAcqkxtJBm0lQwDoAdsOKOMsYnt
gWajprlJ3Hm+34NPL04Usf1OpcesPUJ4CBxNyLXxjjsOzD78WVvKY+N+j89xTsy
z5Y0fEkFqrc18pgBQxH72jBwSCm5YwHz3BhWQgr2bpWJ1f2LWcVsnrN9tx6RhQtA
AkcyNgX/ksp5EW4JTo+o6oXLRhXIYauRrUrisMY++b8ZJTp6C1t0RW2QdggMZghS
ZgaW6FSC6Dy2Dd/ezdkYUCgiEtq8eSx/8WDw6Va2iGVSNt4/p/OJ97yN5yOJ0K1
g0hAteBU+I3E74PQ9RK84FfJvyHDBC6fvYZW/ouMcgp3YmAF+dTm74Hq88X4daV+
/UPYf/cVpyiwcBTg6H3jrkrs0yKoWLiFfrIvMNBeeKZ+fl2Enw1MFzkLI4VGD/UeR
wrbbhN0SHkh5lIGtu0yRTfQ6msYQpkw+jr7QwJIdQyrAoaVaRotVyyvgtOLlHw8r6
o7v36yoNov3kDPW7DfbSVTWX5lIyQn8NqMwa4N1clWT8ukfZXSaYykFSqF3w5zal
a4iIhu03GjDcfiWLMU1YVAUcvSmcIULElOW7FKiJc8OadeIu0JBySRSEvf7B3w8l
eYUs+u/hlptrZZKhe1JdAtlszvHJ0DD0kMqA6Ig4yomscGSol/sRUqpecIQwVZTC
RRq9dJOFJkKhKD5Eo9E0Z2snp0lfpUF5qlMeBjpYgkX7jhyFyvq+qDqBAY8izvkc
ruE69WooBVyqrqKHURjWtY+rhzcB4+HL72wZKzLnY3iUjJ1UANxm8mC9fpD1Njt/
7epqzPyZ2Kd4GJVYi8sQpFKf4tRHDrotI5iUB78qjlEBp1w4qvRn/jc4ii7+Bas8
mz/AJ25Qevic44Vj+eT2YXXafDivrmoeBuVMIBbD066YnuBC2CeKydnWdiARzc3I
fhcuhVwq7riotYfyDqd4e0Jy7Y57pbwv4QwzlyCXRjSwiFQ7/fRa2Cx8xtxKcC/A
4LGnXAKISy+uNbDWA7AYaP6RmGgMcAniXy3F1zvxnE3bv68tXRF9vjuEChUq56N6
992qhoBuHP0J/mRitw+JoI4m/OfnEUGT3bNyxpEFyA7aXBE91aQdSXl4a97nC0/R
SFH/fRwPFYgxr3XdcIf3Cw5PDs25YNsXWcsDCVeJWMFrowmDwa8sBkY270+rGv7
6qXvb/uGD3M2C+DySVy55Zd42wjghSezgY6taT0tqKfLOS6V14ELU78Q6va2o8Ml
cUdi343tOi60MZgCDUwPP8TjKZINh8u1KNhzgppwNLzlgE0dd200l3bbzdZ6uio3R
52WQWRck17Z9lUesCJavytCAi0mMefMxBPMODnUi608TPDRA0mcohE5rybwDXAo
B/VUbwgM0/qCpZ7VcSKN1lUuoE9+Kho0NK/gymEvntMxGNNI8arV8UkeFollPhrt
umvdwqBVCeN8TBj5vXo6Hu+eKB7AVwjBk/rRHpZxnnVGXbm8HzM+kjib2cYldius

```

VRJ/1+Q9GXuo135tQbobgcMzAmqAqZp9kDE8MBUGCSqGSib3DQEJFDEIHgYAYgBv
AGIwIwYJKoZIhvcNAQkVMRYEFqzrDFTakmcTeNueeA1YZU+iGI1MIIFkAYJKoZI
hvcNAQcBoIIIFgQSCBX0wggV5MIIFdQYLKoZIhvcNAQwKAQKgggUmMIIFiJAcBgoq
hkiG9w0BDAEDMA4ECCNi2K1bMEiBAgIUdgSCBQDLIXo4ExcyE8+4aiZIJj/Wnh/SV
VVR0n7s4PGCbXt+VrOHd9YzTuUicAqIcHH62dv7NSy+fgqZG7SmVR1IodadFe+5u
sAzXoyyhhEe2c+ToeVbr5rs+vBvQUyh6X5XTV5QVOAkWsyKGjyfdy86x1Q8cL2D2
BM+RpkmlcFtjgWcB46U6S6w50sG7XOKSCMI4a6rnHPVgPPdXMrj3VSPJY8bhBqED
PVTnfSHf/wKZrTi5403F33B5jt6Cm9+9m9Fed8n+81w59rRom72CY9Xii/ULER9T
HwjxOZOQ+dIm123KauwexuOGjii0UR8MeM/AOn7UNys+bZTulgdPWW/mDhJ+eLAT
nhJw5ro/AWA6YVXG+t5k9LjdJ1ZmqS4bJxvBwilpEGoh0MM6Yp0dr1XM4mT/E0JM
WD458Ngs05CuCpWAUXGdQmgrVsFrrV0HTyHeVLDhe43J3GI6HCWJV0eDQzzma03A
M+IooRDkTHnJMaxUXphKTag5+f/smNYEhzVjZeIc8GFZ36eSI4BNHGSXFACWLu2T
hkzpxMMg50JAUhBYxqE/fVevLUH4JPLgz869wk8gRlUBo6ihQGrnsx7ZO5IsYahE
Yjz0N05PVPJYMLSyMovG9i+LpzQ49gIBzPu2fdLR41u5n505mG1Y4aJ7OCJxMORY
hWHuctHdGdpJsgiq8+1iiUwmfyCfb0ZL3ePMU+W0zkAsyn22aK8jDBLLVZ1vOZIV
qR3Gx4QFPsk6qCMQOE58VkMUMxYvClzTwSeEMu66eND/AKTE+XXV/d9bmSmWGk7Y
8XrDKLKfmrdr1IeondVJv5mk12YKxBPQGeUqK5XJUa2dzH9zvFEX8iYzdt4281QC
iXJ3qwmBt+8RoOLBt4KyOs2e2ZSZnjrL9004oUsHIOyEfjwnWoLhKbkmun8GJxoB
2yCzTawVQf9/qIUxASzcp23AV6Lflk9Of79HYPW3cQJAtjf6XBVE1xVZPkfTuC3y
VLuf1js2ed/ctpHg9nuId/xHFH7t4HbmU3/ZufE1GHnsRQ3kbnqA5WXerd9UzeoD
aVDjFXGrITp8env08GXYvwWGXL15010DuJSv1E+1yww86SNjBYUTx0r0CJjjTk2
7vIUhAYUEA+J71IeifqqPDKYXnrCdUEajbfEdek30WiLR+ChEvEp48Mla6UVTLM/
mjiwbsxm5QlGccmz13e32RiyrfsB+RyllmzeJtydP2IHkWK7pww9y0lPK0QtZs
66IGZKqEXrWBk9QFYDX42gAy/xTfglco4KO7akhp3UzTIQyTXnt+OsOScc+ArVm/
dwC1m+ZxybtOcVyadjpKWydyfAr3aTkGxX6RmHrEWrlR9BnMGPyEsDs+yeVNs1Qd
Dhff/bQLwCLXdxGLWwLe6kitUiYi8F3bdfPjR7R611EUvJrBm7YlmgdxRCJ02LFLG
n09iSMNe5vmiNaKiuzfb4Dp9dqEMhmJfdsTURagfJIYqULoe08EIIozahivbzoWV
A6oPAkk2D8DnTiMegX4IZ/Zb3LPxJKAeX03Ys1YQrNSNZ3B2ZISBapzGzhFZfRVz
POMXhN53pDhlxkw0btKkblYA9CvP+kzgwkeZCy/Mlq/HbO38CV1NKzay3yg4nteh
J+v9/k7gaqKmo3ZWMGk0WGBv/GFxYhmeNd14Y65D9TlYPm/zrXSsyGoOqZgSA6H1A
gogzwwSaGwx9n/o6czE8MBUGCSqGSib3DQEJFDEIHgYAYgBvAGIwIwYJKoZIhvcN
AQkVMRYEFBfFhHvQp+92kDi4s28IvJK1niuUMC8wHzAHBgUrDgMCGgQUgwafFeGU
n9Q1rAOUcGw+KWxk+8EECJ1vqXe6ro0FAgIoAA==
-----END PKCS12-----

```

6. Example Ed25519 Certification Authority

The example Ed25519 Certification Authority has the following information:

* Name: Sample LAMPS Ed25519 Certification Authority

6.1. Ed25519 Certification Authority Root Certificate

This certificate is used to verify certificates issued by the example Ed25519 Certification Authority.

```

-----BEGIN CERTIFICATE-----
MIIBtzCCAWmgAwIBAgITH59R65FuWGNFHoycON3iWesrXzAFBgMrZXAwWTENMAsG
A1UEChMESUVURjERMA8GA1UECzMITEFNUFMgV0cxNTAzBgNVBAMTLFNhbXBsZSBM
QU1QUyBFZDI1NTE5IENlcnRpZmljYXRpb24gQXV0aG9yaXR5MCAXDTEwMTIxNTIx
MzU0NFoYDzIwNTIxMjE1MjEzNTQ0WjBZMQ0wCwYDVQQKEwRJRVRGMREwDwYDVQQL
EwhMQU1QUyBXRzE1MDMGA1UEAxMsU2FtcGx1IEExBTVBTEIEVkmjU1MTkgQ2VydGlm
aWNhdGlvbiBBdXRob3JpdHkwKjAFBgMrZXADIQCEgUZ9yI/rkX/82DihqzVIZQZ+
RKE3URyp+eN2TxJDBKNCMEAwDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMC
AQYwHQYDVR0OBBYEFGuilX26FJvkLQTRB6TRguQua4y1MAUGAyt1cANBAFAJr1Wo
QjzwT0ph7rXe023x3GaLPMXMwQI2Of+apkdG2mH9ID6PE1bu3gRRqIH5w2tyS+xF
Jw0ouxkJyAyXEQ4=
-----END CERTIFICATE-----

```

6.2. Ed25519 Certification Authority Secret Key

This secret key material is used by the example Ed25519 Certification Authority to issue new certificates.

```

-----BEGIN PRIVATE KEY-----
MC4CAQAwBQYDK2VwBCIEIAAt889xRDvxNT8ak53T7tzKuSn6CQDe8fIdjrCiSFRcp
-----END PRIVATE KEY-----

```

This secret key is the [SHA256] digest of the ASCII string draft-lamps-sample-certs-keygen.ca.25519.seed.

6.3. Ed25519 Certification Authority Cross-signed Certificate

If an e-mail client only trusts the RSA Certification Authority Root Certificate found in Section 3.1, they can use this intermediate CA certificate to verify any end entity certificate issued by the example Ed25519 Certification Authority.

```
-----BEGIN CERTIFICATE-----
MIICvzCCAaegAwIBAgITR49T5oAgYhF5+eBYQ3ZBZIMuujANBgkqhkiG9w0BAQsF
ADBVMQ0wCwYDVQQKEwRJRVRGMREwDwYDVQQLEwhMQU1QUyBXRzExMC8GA1UEAxMo
U2FtcGx1IEExBTBTIFJTSBDZXXJ0aWZpY2F0aW9uIEF1dGhvcml0eTAqFw0yMDEy
MTUyMTM1NDRaGA8yMDUyMDkyNzA2NTQxOFowWTENMA8GA1UEChMESUVURjERMA8G
A1UECxMITEFNUFMgV0cxNTAzBgNVBAMTLFNhbXBsZSBMQU1QUyBFZDI1NTE5IEN1
cnRpZmljYXRpb24gQXV0aG9yaXR5MCAwBQYDK2VwAyEAhIFGfciP65F//Ng4oas1
SGUGfkShN1Eccfnjdk8SQwSjFDB6MA8GA1UdEwEB/wQFMAMBAf8wFwYDVR0gBBaw
DjAMBggpgghkgBZQMCATACMA4GA1UdDwEB/wQEAwIBBjAdBgNVHQ4EFgQUa6KVfboU
m+QtBNEHpNGC5C5rjLUwHwYDVR0jBBgwFoAUkTCOfAcXDKfxCSHlNhpNHGh29Fkw
DQYJKoZIhvcNAQELBQADggEBAGV0x0OeZgYlRKixMcztiiKxxJDbmRat1pcipD15
1n8kiBoGhsT4fNZJVoL0OQBw/WTMntL+qcAk2itqZCNIEZeGk1U1jXBaz5tkDRAf
f/v99LEcsZTcuIbnJqz35danQkp4/upG4hPkfx+nbc1bsVylrITwIGOpnGhz7z3m
VCK03DFE3Qt4w9mlv9yuMse33nmsBGXog/XZvM2JRY0iKt0xksQqQD9uYm7MoMeH
qQs3Ot7EaoPj54xyWvy42run6TLUye64D94SNjB/q/wjL96bsVIKGrRn10T1ybCh
4F5HD00hQZgP15Dlblrq+vskN8MSk5nuD+6z1VsugioW0+k=
-----END CERTIFICATE-----
```

7. Carlos's Sample Certificates

Carlos has the following information:

- * Name: Carlos Turing
- * E-mail Address: carlos@smime.example

7.1. Carlos's Signature Verification End-Entity Certificate

This certificate is used for verification of signatures made by Carlos.

```
-----BEGIN CERTIFICATE-----
MIICBzCCAAbmgAwIBAgITP14fVCTRtAFDeA9zwYoXhR52ljAFBgMrZXAwWTENMA8G
A1UEChMESUVURjERMA8GA1UECxMITEFNUFMgV0cxNTAzBgNVBAMTLFNhbXBsZSBM
QU1QUyBFZDI1NTE5IEN1cnRpZmljYXRpb24gQXV0aG9yaXR5MCAwBQYDK2VwAyE
EwhMQU1QUyBXRzEWMBQGA1UEAxMNQ2FyYyBzZS1FRlcm1uZzAqMAUGAyt1cAMhAMLO
gDIs3mHITYRNYO+RnOedrQ5/HuQHXSYPYAKaS98ito4GwMIGtMAwGA1UdEwEB/wQC
MAAwFwYDVR0gBBawDjAMBggpgghkgBZQMCATABMB8GA1UdEQQYMBaBFgnhcmxvc0Bz
bWltZS5leGFtcGx1MBMGA1UdJQQMMAoGCCsGAQUFBwMEMA4GA1UdDwEB/wQEAwIG
wDAdBgNVHQ4EFgQUZIXjO5wdWs3mC7oafwi+xJzMhD8wHwYDVR0jBBgwFoAUa6KV
fboUm+QtBNEHpNGC5C5rjLUwBQYDK2VwA0EAwVGQWbdy6FQIPtFsaWvG2/US2fnS
6B+BzgCrkGQKWX1WgkTj4MEOqL+0cFXLr7ZQ2DQUo2iXyTAu58BR6btCCQ==
-----END CERTIFICATE-----
```

7.2. Carlos's Signing Private Key Material

This private key material is used by Carlos to create signatures.

```
-----BEGIN PRIVATE KEY-----
MC4CAQAwBQYDK2VwBCIEILvvxL741LfX+Ep3Iyye3Cjr4JmONIVYhZPM4M9N1IH
-----END PRIVATE KEY-----
```

This secret key is the [SHA256] digest of the ASCII string draft-lamps-sample-certs-keygen.carlos.sign.25519.seed.

7.3. Carlos's Encryption End-Entity Certificate

This certificate is used to encrypt messages to Carlos. It contains an SMIMECapabilities extension to indicate that Carlos's MUA expects ECDH with HKDF using SHA-256; uses AES-128 key wrap, as indicated in [RFC8418].

```
-----BEGIN CERTIFICATE-----
MIICNDCCAeagAwIBAgITfz0Bv+b10MAT79aCh3arViNvhDAFBgMrZXAwWTENMAsG
A1UEChMESUVURjERMA8GA1UECxMITEFNuFMgV0cxNTAzBgNVBAMTLFNhbXBsZSBM
QU1QUyBFZDI1NTE5IENlcnRpZmljYXRpb24gQXV0aG9yaXR5MCAXDTIwMTIxNTIx
MzU0NFoYDzIwNTIxMjE1MjEzNTQwWjA6MQ0wCwYDVQQKEwRJRVRGMREwDwYDVQQL
EwhMQU1QUyBXRzEWMBQGA1UEAxMNQ2FybG9zIFRlcmluZzAqMAUGAytlbGZhAC5o
MczTIMiddTUYTc/WymEqXw8hZm1QbIz2xX2gFDx0o4HdMIHaMCsGCSqGSib3DQEJ
DwQeMBwwGgYLKoZIHvcNAQkQAAMwCwYJYIZIAWUDBAEFMAwGA1UdEwEB/wQCMAAw
FwYDVROgBBAwDjAMBgpghkgBZQMCATABMB8GA1UdEQQYMBaBFGNhcmxvc0BzbWlt
ZS5leGFtcGxlMBMGA1UdJQQMMAoGCCsGAQUFBwMEMA4GA1UdDwEB/wQEAWIDCDAd
BgNVHQ4EFgQUgSmg+iOgSyCMDXgA3u3aFss0JbkWwYDVROjBBGwFoAUa6KVfboU
m+QtBNEHpNGC5C5rjLUwBQYDK2VwA0EAzss75UzFuADPfd4hQdo5jyAQ3GvkyvI
BdBGNWtJ1eT1WuMaIMh1rH4vPGPd9scwW+sqd9fG+pv3MShl+zKAQ==
-----END CERTIFICATE-----
```

7.4. Carlos's Decryption Private Key Material

This private key material is used by Carlos to decrypt messages.

```
-----BEGIN PRIVATE KEY-----
MC4CAQAwBQYDK2VuBCIEIIH5782H/otrhlY9Dtvzt79ffsvpcVXgdUczTdUvSQsK
-----END PRIVATE KEY-----
```

This secret key is the [SHA256] digest of the ASCII string draft-lamps-sample-certs-keygen.carlos.decrypt.25519.seed.

7.5. PKCS12 Object for Carlos

This PKCS12 ([RFC7292]) object contains the same information as presented in Section 7.1, Section 7.2, Section 7.3, Section 7.4, and Section 6.3.

It is locked with the simple five-letter password carlos.

-----BEGIN PKCS12-----

```
MIIKZgIBAzCCCpYGCsSgSIb3DQEHAAcCCocEggqDMIKfzCCAvCGCSqGSIB3DQEH
BqCCAAugwggLkAgEAMIIC3QYJKoZIhvcNAQcBMBwGCiqGSIb3DQEMAQMwDgQIwS3R
pTlmkyMCAhS7giICsGKkBM0nci9VHfXqOTWy/lkKyQeF5bwsF/9gZrqUym1KtHZF
a4rSJIPUctmzqVnhGmFW9m+LEi7Em9rRmUIQbDZt4kQDG5eDk7AdhyDnB3uZDG1W
4cAeUVXJMzGfnwtzy5TzBZzEo5nnVX74A1+PDW9wdpbv2TiriL0m29fBT+7HVS9F
Z/95XokSwbb6mmCYeGiPpNEaoeUeuU4zrh/k+JJqDuqNsU66I30wH0CFmk3aarBV
3LkEecjKfknzgMOZqiKZu8D2hEUjsGQ9ALsRn7P+hIWNFIgjjvqgcCMTF8fLK1C/8
vYGD+HOpnn23nLele4b/qpfYx5kJO0K1ZolSpqUQ7Bu6gectUceyOgi7CjRScuV
ew7918ZY0ugyYoIWAT0kecPM0TFtxAn19JPXo4jBYAlwUtx7GYAlDkgZCb/0dbkv
4L+PAeJK4kVDREDQ6ch/6/hlqU8xHeNzdagEWYL6FxDiHebASxIvZzqkLd7RV9m
dL1FXst9R9G74jOs0WMMFmd9toyOhD0q6G19catOro1CVS/CKaC0CucsJfiKrlJ/
duQkt/JwcELveuOg60u2uaGKUqHmFhd3+6omk+wNB0Y+0D5MmBZ/xnrVELGmzp94
q0f/HfZPT6sxkYBGUp2eUA/qR/zimNG3TuGVch/MdnduuVhvAYLyhlgbA8yRm+I/
zGCVuAqhsHITTx7Fqc3tyVp/mLYU00QuwmgAw6NhzwKZf5N+tR0DZGcgw8rZpeJA
yTxVfcjzXvoShxog7RroR9Nc4FwJhWI4BO241OHFEiQZeRk8vzI8WIFXnn6t42/q
j1mV7Ba42zxPEGoY3mObKwjR6rDp6KwmmfkghpwMPU3qP2/ASV8WT1+9GIYHc5Am
9CmSOTiQM1uW70Ra2k5ZM1wnbKNyMRbjUB/yHwwwggKvBgkqhkiG9w0BBwaaggKg
MIICnAIBADCCApUGCSqGSIB3DQEHATAcBgqhkiG9w0BDAEDMA4ECOMzXMste/8a
AgIU1ICCAmgXa+q2JhTLvWs5jSKLdMninTk5uB6HhOsDKYR9GDg/cABqUFxycROG
JeJuewIRkJsfdXJi+TSRtnQOppyVM9oRUdxcBGuCI98fEbLmVyr7KF8GudTgC+b
eaLjn6HYkWp7lWdvsFG8BEy6Jqi3/tP9PgNvpCYgVVM7yx6SX8QArcLSQkxbTsv
Ae0iN18H89W9xOHEz4Z2qHYyb7f0pPHrmpTGC6qmtvolgNRsKTF0wYeQ5Sy/9U3f
oM6bIcrOvHDksaco4+5n0zeySDETY8W4m01K0uC/t0oTOScYGBerhVr0DQapZGT/
Ej5LpgjXOuosAoT3IKnMwK3C00Z8oBzcvgSpeAa/V/OTKDPZb22yq6sEaHAPoUqb
cKRJmB6HC5mdLs3n0uPlv1ZuYsHu7Evt0Uhn59pbklJDiCgM+4SFgKTRbd6Xt8bf
GHkWNmpv4pQL7jjzA3epP2DHyc8MJaDvleWY7Z3t/IETkzVxf1Lo8kT2ledz12cm
uFVK9i1MW3eJuYiRyFXFPgVsuNi/HFniJXFgxzAncP7fFP5MCsOo6daiEjJjemKf
J3D+HdD60gFih/eX9V+tG14y7/jtxCRA/54mit4sCy3LC0++1Ep9AtFwGYrDw825
uGj27a7mE26qgGdGXdzT9UJ8FfUsIoRPrG38Q4mhS10pTarNucWOGjkftZiKJLay
rfMRf3HYxOI/7iupfxYlK/4/FODijaHzAfSdQf2Bo7csPaz2HQkK/0nyO+tt68S9
pUCjEfV6Liy22tang/jXxPFbBDK/P68Mnmgr8C3PcYhPJCo/K0JR2/8F8pVVEqd5
MIIDPwYJKoZIhvcNAQcGoIIDMDCCAYwCAQAwggMlBgkqhkiG9w0BBwEwHAYKKoZI
hvcNAQwBAzAOBAho9g0tQyYTvwICFIAggL43SpNCoshZX3ikmK1mOIJpS2Ah8Xv
94S/5NA8kwHtaNXpLrjYr3CyRL93USm55uvGAtECR/Eb1ON9zeo2p0gK2JPSbDr6
/1oovo7UoZNRoRBZ8pUegVWJswNWjqvzVu5JIRmpD05XjVDKHbFqiXAqtj9/w3q0
Qq/p/M9UrLWD93hyLNdIppWr2KR2it9mASTKEHX9dqXcTOG0Kp2GmrfGNteGL02j
qVKZaZyYI8gkSxhVLS9zzgf1OynAkzYQsoo+GKhDAW1fJECemAyPc3L+eeARw/SY
qld5QVwxKfYpIJ2wiiavdeRVNBWiW7Ti+P9PtPx/hV22NNLwMhvnJcHaSS1PaOi
SjoxFJ1EJWGES0QwcwM8iN3oVuqT5HU/edMgx9TLNTiElg2GEq59I/RwBtCL8Dh
```



```

OzKnUb4PUlZ81+HimV3KPI8g3cduhYaBR4HfqAhMnc+w5HXI6J3C1NtAE/izZ1Y2
Od7l+GTJfjPgziy0hjqlbMt8uU9D9aPr2XjNOWoKRSojae16v8bLx+dFn6RMxFUS
g3nLEZ6EDpyrJfpGPm6mPgZKSXtnHuFcbS+utkRuVAtqu07r2XpkGBIJLNVIRHU
5gLACbTj9TPcAce6RLoaYSDgOuFK0YZMdwzhsAI0YMPyHsUEZpQ5tjWSBY6ENbvF
7+QhmDnf6N3Bj+vxUtGS40pVsYCGbmOD7UM5QpUxIgVkpPrfRokOZs/fi9sW+Xy6
eQ2Brbn3t9C2TASORYzFbuBwuTCqFW/rXHS6iffJpx2eAg3DCqaUAJjptSV/yzj4
vxiXlDB3fMRcpNd5Je7DoHS4axuj7SLHdpNoUHs+qQsG6yDM5BEuXWGxo/L9sGhe
XQrUnkZ4m4g01sfGTOfDNurXx/oP0ym+B50q6nLUWv0tYZpmCVil358dIEGPPSMY
AMXh05tIPFdYSJS3WLs0cxy5X4sXZl5w16Pzeb9SF5topqRUb5PDTfVr2bQUMwThp
99FcOQf6cg8HXyT+8b4qKp9WyjCBxAYJKoZIHvcNAQcBoIG2BIGzMIGwMIGtBgsq
hkiG9w0BDAoBAqBaMFgwHAYKKoZIHvcNAQwBAzAOBAgNhfODEdzSrQICFF0EOCEq
FielpeicS9OSXNQjLwbN3k08lYM2HqeSZoEKJ4JSF1V1kWW3xwfu5azKrGEYBfGM
d8renRijmUIwGwYJKoZIHvcNAQkUMQ4eDABjAGEAcgBsAG8AczAjBgkqhkiG9w0B
CRUxFgQUgSmg+ioGsyCMDXgA3u3aFss0JbkwcgQGCsGSIb3DQEHAACBtgSBszCB
sDCBrQYLKoZIHvcNAQwKAQKqWjBYMBwGCiqGSIb3DQEMAQMwDgQINFCqIEMfd9UC
AhS1BDGzruEsSaBY+Cm9WKR8HhH3JXh+AoMSrwdCKytWt+MNIXB0jY2QZHDn3u
Fn7qHw06MDthnKniazFCMBsGCSqGSIb3DQEJFDEOHgWAYwBhAHIAbABvAHMwIwYJ
KoZIHvcNAQkVMRYEFGSF4zuchVrN5gu6Gn8IvsSczIQ/MC8wHzAHBgUrDgMCGgQU
8nOYIWrnJVXEur957K5cCV3jx5cECJDjaZkfY4FnAgIoAA==
-----END PKCS12-----

```

8. Dana's Sample Certificates

Dana has the following information:

* Name: Dana Hopper

* E-mail Address: dna@smime.example

8.1. Dana's Signature Verification End-Entity Certificate

This certificate is used for verification of signatures made by Dana.

```

-----BEGIN CERTIFICATE-----
MIICAzCCAbWgAwIBAgITaWZI+hVtn8pQZviAmPmBXzWfnjAFBgMrZXAwWTENMAsg
A1UEChMESUVURjERMA8GA1UECzMITEFNUFMgV0cxNTAzBgNVBAMTLFhnbXBsZSBM
QU1QUyBFZDI1NTE5IENlcnRpZmljYXRpb24gQXV0aG9yaXR5MCAXDTEwMTIxNTIx
MzU0NFoYDzIwNTIxMjE1MjEzNTQwWJA4MQ0wCwYDVQQKEWRJRVRGMREwDwYDVQQL
EwhMQU1QUyBXRzEUMBIGA1UEAxMLRGFfuYSBib3BwZXIwKjAFBgMrZXADIQCy2h3h
hkaKDY67PuCuNLnnrQiHdSWYpPlgFsOif85vrqQBrjCBqzAMBGNVHRMBAf8EAjAA
MBcGA1UdIAQQMA4wDAYKYZIAWUDAgEwATAdBgNVHREEFjAUGRjKjYw5hQHNTaW1l
LmV4YW1wbGUwEwYDVROlBAwWCgYIKwYBBQUHAWQwDgYDVROPAQH/BAQDAGbAMB0G
A1UdDgQWBBRIA4bBabh4ba7e88wGsDOsVzLdljAFBgNVHSMEGDAWgBRropV9uhSb
5C0E0Qek0YLkLmuMtTAFBgMrZXADQDpORBZitZxGYUjxnoKVLicWL5xner97it5
VKxEf8E7AeAp96POPEu//2jXnh4qAT40ymW0wrqxU1NT8WW/dSgC
-----END CERTIFICATE-----

```

8.2. Dana's Signing Private Key Material

This private key material is used by Dana to create signatures.

```
-----BEGIN PRIVATE KEY-----
MC4CAQAwBQYDK2VwBCIEINZ8GPfmQh2Amp+uNIzZMbzvvyTOltwvEt13usjnUaW4N
-----END PRIVATE KEY-----
```

This secret key is the [SHA256] digest of the ASCII string draft-lamps-sample-certs-keygen.dana.sign.25519.seed.

8.3. Dana's Encryption End-Entity Certificate

This certificate is used to encrypt messages to Dana. It contains an SMIMECapabilities extension to indicate that Dana's MUA expects ECDH with HKDF using SHA-256; uses AES-128 key wrap, as indicated in [RFC8418].

```
-----BEGIN CERTIFICATE-----
MIICMDCCAeKgAwIBAgITDksKNqnvpuyaO2gkjlIdwN7zpzAFBgMrZXAwWTENMAsG
A1UEChMESUVURjERMA8GA1UECzMITEFNUFMgV0cxNTAzBgNVBAMTLFNhbXBsZSBM
QU1QUyBFZDI1NTE5IENlcnRpZmljYXRpb24gQXV0aG9yaXR5MCAXDTEwMTIxNTIx
MzU0NFoYDzIwNTIxMjE1MjEzNTQwOjA4MQ0wCwYDVQQKEwRJRVRGMREwDwYDVQQL
EwhMQU1QUyBXRzEUMBIGA1UEAxMLRGFuYSBib3BwZXIwKjAFBgMrZW4DIQDgMaI2
AWkU9LG8CvaRHgDSEY9d72Y8ENZeMwibPugkVKOB2zCB2DARBgkqhkiG9w0BCQ8E
HjAcmBoGCyqGSib3DQEJEAMTMAsGCWCGSAFlAwQBBTAMBgNVHRMBAf8EAjAAMBcG
A1UdIAQQMA4wDAYKYIZIAWUDAgEwATAdBgNVHREEFjAUGRjkYW5hQHNTaW1lLmV4
YW1wbGUwEwYDVROlBAwwCgYIKwYBBQUHAWQwDgYDVROPAQH/BAQDAgMIMB0GA1Ud
DgQWBBSd303UBe+a7GCGvCdtBOnOWtyPpDAfBgNVHSMEGDAWgBRropV9uhSb5C0E
0Qek0YLkLmuMtTAFBgMrZXADQQD6f7DCCxXzpnY3BwmrIuf/SNQSf//Otri7USkd
9GF+VthGS+9KJ4HTBCh0ZGuHIU9EgnfgdSL1UR3WUkL7tv8A
-----END CERTIFICATE-----
```

8.4. Dana's Decryption Private Key Material

This private key material is used by Dana to decrypt messages.

```
-----BEGIN PRIVATE KEY-----
MC4CAQAwBQYDK2VuBCIEIGxZt8L7lY480Eq4gs/smQ4weDhRNMLYHG21StivPfz3
-----END PRIVATE KEY-----
```

This seed is the [SHA256] digest of the ASCII string draft-lamps-sample-certs-keygen.dana.encrypt.25519.seed.

8.5. PKCS12 Object for Dana

This PKCS12 ([RFC7292]) object contains the same information as presented in Section 8.1, Section 8.2, Section 8.3, Section 8.4, and Section 6.3.

It is locked with the simple four-letter password dana.

-----BEGIN PKCS12-----

```
MIIKtgIBAzCCCN4GCSqGSIB3DQEHAAACCCm8EggprMIIKZzCCAu8GCSqGSIB3DQEH
BqCCAuAwggLcAgEAMIIC1QYJKoZIhvcNAQcBMBwGCiqGSIB3DQEMAQMwDgQIZNqH
TA2APx0CAhQXgIICqK+HFHF6dF5qw1WM6MRCXw11VKrcYBff65iLABPyGvWENnVM
TTPpDLqbGm6Yd2eLntPzvJoVe5Sf2+DW4q3BZ9aKuEdneBBk8mDJ6/Lq1+wFxY5k
WaBHTA6LNml/NkM3za/fr4abKFQnu6DZgZDGBzh2BsgCMmO9TeHgZyepsh3WP4ZO
aYDvSD0LiEzerDP1OBgjYahcNLjv/Dn/dFxt003or010TTUoQCqeHJ0oq3hJtSI+
8n0iXk6gtf1/ROj6Jrt/3Aqz/mLMIhuxIg/5K1wxY9AwFT4oyflapNJozGg9qwGi
PWVtEy3QDNvAs3bDfiNQQAfJOEHv2z3Ran7sYuz3vE0FnPfA81oWbazlydjB0P/B
OQ+s6VLbsAosnZq9jv2ZVrCDaDA1/g7oD7fY8qmaC6O2q5/Z3KusfMt+r9En2v81
H2vjgrpxnDIXjYuLzdrnNE/s1RtqadOGR/WQ358RG+yUmRUbHYHGnkjn9fOGLasI
ZUV0aowivcWyF/kR7QV3VvexgqJMX6k1vzSXR0J/tnA+1/WPWy1mCJe1jGogYqSV
txtVB61Qmc2XP48F7wyaQZvdAU9zfe11/tHAaKKJWBpE11IuAEkGtIP6ozYJBFjH
I11tBA8fiJtnug+S4OvSgjtSRV/+kSEiW4F+pwE8RuTYfUu7q+Ew0LYdLgkH5OyE
sn0b62UFpR/ElD9exWzohrFbIdUCbjtssXucruAqPNhW/abT0zicWu5nvf+Pniow
2VxvhwoGt5jZ+1kaR5Z+1/GpbMgq47EUyGCgKv+5GACJxUxINZqLbACJ/MhLfYPB
eJrXz8f5Cigm1wZLisYCnuc8cGCXjNqNkUlqtzodM8xv4gcgT/zILxmJTzP2q4n
YA4yBQx5/n2G2dZC+pf3kAfbXcp0MIICpwYJKoZIhvcNAQcGoIICmDCCApoCAQAw
ggKNBgkqhkiG9w0BBwEwHAYKKoZIhvcNAQwBAzA0BAjxuoiaSZDbnwICFH+AggJg
k2hcNYt00+15uLqXdiNhr5Q0JkYcrHdo0wR6G5AgLmWI+TYi+P8EZUjdIJ4TJ3b4
6xv7+3pT8cbEFf6PXcfS8/sCfM7FaV3SpLACLZbBJV52OKE0CAGALZOLuIz5mGVU
tWI2h1x587KeIv5GRPixumDebT3Gmkkp9Qoi55hjTgn68olSgDaJF8o5wnfODhKS
o110a3x9OwkJSN1AXfmBfj33KnT8Dc4bTfAZylS5o1zCtaEqnct2Urb4PeO3LfHB
ErBsvY8HE4D7qh6P5ftXHQHAX/b3hbU8jQP1tR0N90h0SiLi//ebCeGXWQRdVjL5
+VQrh1QF5d4Kz9Zx79oC36g7C2BxCQomur/F9TT12NPzPpaEGGo61jB6myAHnYw9
rCxbSxBvbtEt1gJnxxblY5Q4ukgyjzK6431Bwq2+iNL0vGc9o2c5ELUPU9zGeLBZ
tXWvdX27aOHjusPFDZ170C5zHiYs1FU6Tkn9Aotc424Q3d2IRTTcYnnjs1VSilSr
4bRyB8zBAQmdQrniBW++7eJm3m/EOU0Yy0noUT169m8KNJrmSspMvKS6pyiYHR4I
BvAikRIjvdtQvJdQJ+Uyr+HH5daE6go1W1917b2bXj/41mvXYkYJ6W8x0km1RYhH
QJZphW1vNcrHKO46Unk48Qc/5J5tI+6UDTXFr//V34vcpQ2ktp0MAK11rBH549ef
CsGQTGoq8XHPkhkseHEEMRmOJDeKTVkKx8xNhbw395yFCIxfF2NHeDLXP+JyW+nH
Iy2fnBDlyTiPF7YXyGiPjPAGK8LS8GUE+Zq2rWqrGNkwggM/BgkqhkiG9w0BBwag
ggMwMIIDLAIBADCCAYUGCSqGSIB3DQEHATAcBgoqhkiG9w0BDAEDMA4ECOJ/s3Y
f5bgAgIUnYCCAvi4NaYP4lpAtuXtE02Zqgl9aLFwsj9B/rikBo6O1ZR/lSryJ4PJ
VGyY6NyBPjG67glJVMYiI3Hge+j66FXKXD/AaiMVD21ZmfrH935S14ZUKS9tpTJL
QDw3eJpDEDqJUfJZJ/ybgpRAKoNjhCE3B7F7+WMI8Pr70M1Fbw7ytUCAjOf18sIW
prUA8f809dLiGgiWyjE5HMzSXEib5IMRpq5x4Q28pBrT8rVYgoQSSyVkfHtU7LDi
Bm68RfBgEl7jIqLdrt2kKxHC3/lC4xXQgFNXEQ056aRp8Yu4VpoRwraVLU03tJk+
pflzFfmUei/JtiFlC6uf0PvC2B5h6kAZocE11LxGIDFH7fTd6dzP7qTDbUQ+uEk3
qsgktT2pcoVnxTanvQmTCEZM9ZKXC5/z7Gkm+z83lGLDDU9oNyRSrxHrRBIVgH4w
```

```
3aGH1v6kfYOWwwwaghQOQIZFyzGVRKXsP7AslL+n4ti83lTxqSUZX2qy9LpI4Tjp
5A/NLMKo3uqmHF1TLnnYUqoppe88FNY8T/LXnHp0KTkuXFmdKJtp1/ydqh18jBk7
nfLcQFdf1R/5okysblRtaMu1lhelymT7MoM8u5C8ceIO7uWX8NI5B/IB+Yn2BvzZ
9LXoSia/wHjTu7UK610o7WQo9qTYeli1x+HsmJaOC6hpaQh6b33VWDrHJb17c/4Z
tvQ9qAzqkqIhFWMRXNK+32jFVAgXrD8U1QHW2ip5s7W/XtmlAegrhGlnSQgJezYl
OnE/t2PDWuPeW94kR0uv1fNsh6p1LyZYf/BaqhoGCHsa/ipD86viVSZDgJ8ASVLF
eLUK3HYFMhJ+MLEzZJffYZAOnbYoyNPNc0vc7dpbk+ZMnlb5bDFcMCpm7+fWOjsC
nsNNL9nqQ1NHHCJRKGuX05rujftbPM7R3GLT9d/u5e9YY5cXORiDLxomFfflj2Yh
uRoyX+8WzEst98I/KmAraWKXnxOP1FEWajtnCrnGCezDKO3xEHTQhECpg+z704mj
MjN6MIHABgkqhkiG9w0BBwGggbIEga8wgawwgakGCyqGSib3DQEMCgECofowWDac
BgoqhkiG9w0BDAEDMA4ECL2BzlVw+YZkAgIUugQ4YOyEjke53NDvCFR0ciUHZ7re
f9/wPx5TgV3qzGhfr4bP2rdpiOt9hAHVK5cmUAR7+wjAJiYdLUQxPjAXBgkqhkiG
9w0BCRQxCh4IAGQAYQBuAGEwIwYJKoZIhvcNAQkVMRYEFJ3fTdQF75rsYIa8J20E
6c5a3I+kMIHABgkqhkiG9w0BBwGggbIEga8wgawwgakGCyqGSib3DQEMCgECofow
WDacBgoqhkiG9w0BDAEDMA4ECFw78Uk8K64uAgIU+gQ4id0jRb3JyEM5fdpaeQR+
YEE Mn+Y5KavplVD5HtgQQY9hhppbQqG4af7KY+MT6xus6oNEQeJAE5wxPjAXBgkq
hkiG9w0BCRQxCh4IAGQAYQBuAGEwIwYJKoZIhvcNAQkVMRYEFJ3fTdQF75rsYIa8J20E
zAawM6xXmt2WMC8wHzAHBgUrDgMCGGUzSoHpcIerV2lCvCOjAe5ZVhs2M8ECC5D
kkzl2MltAgIoAA==
-----END PKCS12-----
```

9. Security Considerations

The keys presented in this document should be considered compromised and insecure, because the secret key material is published and therefore not secret.

Any application which maintains a denylist of invalid key material should include these keys in its list.

10. IANA Considerations

IANA has nothing to do for this document.

11. Document Considerations

[RFC Editor: please remove this section before publication]

This document is currently edited as markdown. Minor editorial changes can be suggested via merge requests at <https://gitlab.com/dkg/lamps-samples> or by e-mail to the author. Please direct all significant commentary to the public IETF LAMPS mailing list: spasm@ietf.org

11.1. Document History

11.1.1. Substantive Changes from draft-ietf-*-07 to draft-ietf-*-08

- * Apply editorial cleanup suggested during review
- 11.1.2. Substantive Changes from draft-ietf-*-06 to draft-ietf-*-07
- * Correct document history
 - * Restore PKCS12 for dana and bob from -05
- 11.1.3. Substantive Changes from draft-ietf-*-05 to draft-ietf-*-06
- * Added outbound references for acronyms PEM, CRL, and OCSP, thanks Stewart Brant.
 - * Accidentally modified PKCS12 for dana and bob
- 11.1.4. Substantive Changes from draft-ietf-*-04 to draft-ietf-*-05
- * Switch from SHA512 to SHA1 as MAC checksum in PKCS#12 objects, for interop with Keychain Access on macOS.
- 11.1.5. Substantive Changes from draft-ietf-*-03 to draft-ietf-*-04
- * Order subject/issuer DN components by scope.
 - * Put cross-signed intermediate CA certificates into PKCS#12 instead of self-signed root CA certificates.
- 11.1.6. Substantive Changes from draft-ietf-*-02 to draft-ietf-*-03
- * Correct encoding of S/MIME Capabilities extension.
 - * Change "Certificate Authority" to "Certification Authority".
 - * Add CertificatePolicies to all intermediate and end-entity certificates.
 - * Add organization and organizational unit to all certificates.
- 11.1.7. Substantive Changes from draft-ietf-*-01 to draft-ietf-*-02
- * Added cross-signed certificates for both CAs
 - * Added S/MIME Capabilities extension for Carlos and Dana's encryption keys, indicating preferred ECDH parameters.
 - * Ensure no serial numbers are negative.
 - * Encode keyUsage extensions in minimum-length BIT STRINGS.

11.1.1.8. Substantive Changes from draft-ietf-*-00 to draft-ietf-*-01

- * Added Curve25519 sample certificates (new CA, Carlos, and Dana)

11.1.1.9. Substantive Changes from draft-dkg-*-05 to draft-ietf-*-00

- * WG adoption (dkg moves from Author to Editor)

11.1.1.10. Substantive Changes from draft-dkg-*-04 to draft-dkg-*-05

- * PEM blobs are now sourcecode, not artwork

11.1.1.11. Substantive Changes from draft-dkg-*-03 to draft-dkg-*-04

- * Describe deterministic key generation
- * label PEM blobs with filenames in XML

11.1.1.12. Substantive Changes from draft-dkg-*-02 to draft-dkg-*-03

- * Alice and Bob now each have two distinct certificates: one for signing, one for encryption, and public keys to match.

11.1.1.13. Substantive Changes from draft-dkg-*-01 to draft-dkg-*-02

- * PKCS#12 objects are deliberately locked with simple passphrases

11.1.1.14. Substantive Changes from draft-dkg-*-00 to draft-dkg-*-01

- * changed all three keys to use RSA instead of RSA-PSS
- * set keyEncipherment keyUsage flag instead of dataEncipherment in EE certs

12. Acknowledgements

This draft was inspired by similar work in the OpenPGP space by Bjarni Runar and juga at [I-D.bre-openpgp-samples].

Eric Rescorla helped spot issues with certificate formats.

Sean Turner pointed to [RFC4134] as prior work.

Deb Cooley suggested that Alice and Bob should have separate certificates for signing and encryption.

Wolfgang Hommel helped to build reproducible encrypted PKCS#12 objects.

Carsten Bormann got the XML sourcecode markup working for this draft.

David A. Cooper identified problems with the certificates and suggested corrections.

Lijun Liao helped get the terminology right.

Stewart Brant and Roman Danyliw provided editorial suggestions.

13. References

13.1. 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>>.
- [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>>.
- [RFC5958] Turner, S., "Asymmetric Key Packages", RFC 5958, DOI 10.17487/RFC5958, August 2010, <<https://www.rfc-editor.org/info/rfc5958>>.
- [RFC7292] Moriarty, K., Ed., Nystrom, M., Parkinson, S., Rusch, A., and M. Scott, "PKCS #12: Personal Information Exchange Syntax v1.1", RFC 7292, DOI 10.17487/RFC7292, July 2014, <<https://www.rfc-editor.org/info/rfc7292>>.
- [RFC7468] Josefsson, S. and S. Leonard, "Textual Encodings of PKIX, PKCS, and CMS Structures", RFC 7468, DOI 10.17487/RFC7468, April 2015, <<https://www.rfc-editor.org/info/rfc7468>>.
- [RFC8032] Josefsson, S. and I. Liusvaara, "Edwards-Curve Digital Signature Algorithm (EdDSA)", RFC 8032, DOI 10.17487/RFC8032, January 2017, <<https://www.rfc-editor.org/info/rfc8032>>.
- [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>>.

- [RFC8479] Mavrogiannopoulos, N., "Storing Validation Parameters in PKCS#8", RFC 8479, DOI 10.17487/RFC8479, September 2018, <<https://www.rfc-editor.org/info/rfc8479>>.
- [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>>.

13.2. Informative References

- [FIPS186-4] "Digital Signature Standard (DSS)", National Institute of Standards and Technology report, DOI 10.6028/nist.fips.186-4, July 2013, <<https://doi.org/10.6028/nist.fips.186-4>>.
- [I-D.bre-openpgp-samples] Einarsson, B. R., juga, and D. K. Gillmor, "OpenPGP Example Keys and Certificates", Work in Progress, Internet-Draft, draft-bre-openpgp-samples-01, 20 December 2019, <<https://www.ietf.org/archive/id/draft-bre-openpgp-samples-01.txt>>.
- [RFC4134] Hoffman, P., Ed., "Examples of S/MIME Messages", RFC 4134, DOI 10.17487/RFC4134, July 2005, <<https://www.rfc-editor.org/info/rfc4134>>.
- [RFC5322] Resnick, P., Ed., "Internet Message Format", RFC 5322, DOI 10.17487/RFC5322, October 2008, <<https://www.rfc-editor.org/info/rfc5322>>.
- [RFC7469] Evans, C., Palmer, C., and R. Sleevi, "Public Key Pinning Extension for HTTP", RFC 7469, DOI 10.17487/RFC7469, April 2015, <<https://www.rfc-editor.org/info/rfc7469>>.
- [RFC8410] Josefsson, S. and J. Schaad, "Algorithm Identifiers for Ed25519, Ed448, X25519, and X448 for Use in the Internet X.509 Public Key Infrastructure", RFC 8410, DOI 10.17487/RFC8410, August 2018, <<https://www.rfc-editor.org/info/rfc8410>>.
- [RFC8418] Housley, R., "Use of the Elliptic Curve Diffie-Hellman Key Agreement Algorithm with X25519 and X448 in the Cryptographic Message Syntax (CMS)", RFC 8418, DOI 10.17487/RFC8418, August 2018, <<https://www.rfc-editor.org/info/rfc8418>>.

[SHA256] Dang, Q., "Secure Hash Standard", National Institute of Standards and Technology report, DOI 10.6028/nist.fips.180-4, July 2015, <<https://doi.org/10.6028/nist.fips.180-4>>.

[TEST-POLICY] NIST - Computer Security Division (CSD), "Test Certificate Policy to Support PKI Pilots and Testing", May 2012, <https://csrc.nist.gov/CSRC/media/Projects/Computer-Security-Objects-Register/documents/test_policy.pdf>.

Author's Address

Daniel Kahn Gillmor (editor)
American Civil Liberties Union
125 Broad St.
New York, NY, 10004
United States of America

Email: dkg@fifthhorseman.net

Individual
Internet-Draft
Intended status: Informational
Expires: 5 May 2022

T. Ito
SECOM CO., LTD.
T. Okubo
DigiCert, Inc.
S. Turner
sn3rd
7 November 2021

General Purpose Extended Key Usage (EKU) for Document Signing X.509
Certificates
draft-ito-documentsigning-eku-02

Abstract

[RFC5280] specifies several extended key usages for X.509 certificates. This document defines a general purpose document signing extended key usage for X.509 public key certificates which restricts the usage of the certificates for document signing.

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 5 May 2022.

Copyright Notice

Copyright (c) 2021 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 | 2 |
| 2. Conventions and Definitions | 3 |
| 3. Extended Key usage for DocumentSigning | 3 |
| 3.1. Extended Key Usage Values for Document Signing | 3 |
| 4. Using the Document Signing EKU in a Certificate | 4 |
| 5. Implications for a Certification Authority | 5 |
| 6. Security Considerations | 5 |
| 7. IANA Considerations | 6 |
| 8. Normative References | 6 |
| Acknowledgments | 7 |
| Appendix A. ASN.1 Module | 7 |
| Authors' Addresses | 7 |

1. Introduction

[RFC5280] specifies several extended key usages for X.509 certificates. In addition, several extended key usage had been added[RFC7299] as public OID under the IANA repository. While usage of any extended key usage is bad practice for publicly trusted certificates, there are no public and general extended key usage explicitly assigned for Document Signing certificates. The current practice is to use id-kp-emailProtection, id-kp-codeSigning or vendor defined Object ID for general document signing purposes.

In circumstances where code signing and S/MIME certificates are also widely used for document signing, the technical or policy changes that are made to code signing and S/MIME certificates may cause unexpected behaviors or have an adverse impact such as decreased cryptographic agility on the document signing ecosystem and vice versa.

There is no issue if the vendor defined OIDs are used in a PKI (or a trust program) governed by the vendor. However, if the OID is used outside of the vendor governance, the usage can easily become out of control (e.g. - When the end user encounters vendor defined OIDs, they might want to ask that vendor about use of the certificate, however, the vendor may not know about the particular use. - If the

issuance of the cert is not under the control of the OID owner, there is no way for the OID owner to know what the impact will be if any change is made to the OID in question, and it would restrict vendor's choice of OID management. etc.).

Therefore, it is not favorable to use a vendor defined EKU for signing a document that is not governed by the vendor.

This document defines a general Document Signing extended key usage.

2. Conventions and Definitions

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. Extended Key usage for DocumentSigning

This specification defines the KeyPurposeId id-kp-documentSigning. Inclusion of this KeyPurposeId in a certificate indicates that the use of any Subject names in the certificate is restricted to use by a document signing.

Term of "Document Sign" in this document is digitally sign contents that are consumed by humans. To be more precise, contents are intended to be shown to human with printable or displayable form by means of services or software, rather than processed by machines.

3.1. Extended Key Usage Values for Document Signing

[RFC5280] specifies the EKU X.509 certificate extension for use in the Internet. The extension indicates one or more purposes for which the certified public key is valid. The EKU extension can be used in conjunction with the key usage extension, which indicates how the public key in the certificate is used, in a more basic cryptographic way.

The EKU extension syntax is repeated here for convenience:

```
ExtKeyUsageSyntax ::= SEQUENCE SIZE (1..MAX) OF KeyPurposeId
KeyPurposeId ::= OBJECT IDENTIFIER
```

This specification defines the KeyPurposeId id-kp-documentSigning. Inclusion of this KeyPurposeId in a certificate indicates that the use of any Subject names in the certificate is restricted to use by a document signing service or a software (along with any usages allowed by other EKU values).

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

4. Using the Document Signing EKU in a Certificate

[RFC8358] specifies the conventions for digital signatures on Internet-Drafts. This is one of the intended use cases for the general document signing EKU described in this document. [RFC8358] uses CMS to digitally sign a wide array of files such as ASCII, PDF, EPUB, HTML etc. Currently, there are no specification regarding EKU for certificates signing those files except those which are defined by the software vendor.

The signed contents of Internet-Drafts are primarily intended to be consumed by human. To be more precise, contents are intended to be shown to human in a printable or displayable form by means of services or software, rather than processed by machines. To validate the digital signature which is signed to contents intended to be consumed by human, implementations MAY perform the steps below as a certificate validation:

The implementation MAY examine the Extended Key Usage value(s):

1. If there are no restrictions set for the relying party and the relying party software, the certificate is acceptable.
2. If there are restrictions set for the replying party and relying party software, proceed as following.

Each Restriction on the EKUs can be "Excluded EKU" or "Permitted EKU" and handled.

The procedure is intended to permit or prohibit presence of a certain EKU or complete absence of EKUs. It is outside the scope of this document, but the relying party can permit or prohibit combinations of EKU. A consideration on prohibiting combination of EKUs is described in the security consideration section of this document.

2.1. Excluded EKUs procedure "Excluded EKU" is an EKU which the relying party or the relying party software prohibits. Examples of "Excluded EKU" are, presence of anyEKU or complete absence of EKU extension on a certificate. If an EKU of the certificate meets the conditions set by the "Excluded EKU" restriction, the relying party or the relying party software rejects the certificate.

2.2. Permitted EKU procedure "Permitted EKU" is an EKU which the relying party or the relying party software accepts. Examples of "Permitted EKU" are, presence of this general document signing EKU and/or protocol specific document signing-type EKUs. If an EKU of the certificate meets the condition set by a "Permitted EKU" restriction, the certificate is acceptable. Otherwise, relying party or the relying party software rejects the certificate.

When a single software has capability to process various data formats, the software may choose to make the excluded and permitted decisions separately in accordance with the format it is handling (e.g. text, pdf, etc).

5. Implications for a Certification Authority

The procedures and practices employed by a certification authority MUST ensure that the correct values for the EKU extension are inserted in each certificate that is issued. Unless certificates are governed by a vendor specific PKI (or trust program), certificates that indicate usage for document signing MAY include the id-kp-documentSigning EKU extension. This does not encompass the mandatory usage of the id-kp-documentSigning EKU in conjunction with the vendor specific EKU. However, this does not restrict the CA from including multiple EKUs related to document signing.

6. Security Considerations

The usage of id-kp-documentSigning EKU intends to prevent id-kp-emailProtection from being used for none-email purposes and id-kp-codeSigning used to sign objects other than binary codes. This EKU does not introduce new security risks but instead reduces existing security risks by providing means to separate other EKUs used for communication protocols namely, TLS or S/MIME etc. in order to minimize the risk of cross protocol attacks.

To reduce the risk of specific cross protocol attacks, the relying party or relying party software may additionally prohibit use of specific combination of EKUs.

While a specific protocol or signing scheme may choose to come up with their own ECU, some may not have significant motive or resource to set up and manage thier own ECU. This general document signing ECU may be used as a stop gap for those that intend to set up their own ECU or those who do not intend to set up an ECU but still would like to distinguish from other usage.

Introduction of this id-kp-documentSigning ECU value does not introduce any new security or privacy concerns.

7. IANA Considerations

This document requests that IANA make two assignments. One for the id-kp-documentSigning object identifier (OID), as defined in Section 3.1, for the ECU from the "SMI Security for PKIX Extended Key Purpose" (1.3.6.1.5.5.7.3) registry. Another for the id-mod-docsign-eku, as defined in Appendix A, for the ASN.1 module [X.680] from the in the "SMI Security for PKIX Module Identifier" (1.3.6.1.5.5.7.0) registry. No further action is necessary by IANA.

8. 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>>.
- [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>>.
- [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>>.
- [RFC8358] Housley, R., "Update to Digital Signatures on Internet-Draft Documents", RFC 8358, DOI 10.17487/RFC8358, March 2018, <<https://www.rfc-editor.org/info/rfc8358>>.
- [X.680] ITU-T, "Information technology - Abstract Syntax Notation One (ASN.1): Specification of basic notation", ISO/IEC 8824-1:2015, November 2015.

Acknowledgments

We would like to thank Russ Housley for verifying the ASN.1 module.

Appendix A. ASN.1 Module

The following ASN.1 module provides the complete definition of the Document Signing EKU.

```
DocSignEKU { iso(1) identified-organization(3) dod(6) internet(1)
  security(5) mechanisms(5) pkix(7) id-mod(0) id-mod-docsign-eku(TBD1) }

DEFINITIONS EXPLICIT TAGS ::=

BEGIN

-- EXPORTS ALL --

-- IMPORTS NOTHING --

-- OID Arc --

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

-- Document Signing Extended Key Usage --

id-kp-documentSigning OBJECT IDENTIFIER ::= { id-kp TBD2 }

END
```

Authors' Addresses

Tadahiko Ito
SECOM CO., LTD.

Email: tadahiko.ito.public@gmail.com

Tomofumi Okubo
DigiCert, Inc.

Email: tomofumi.okubo+ietf@gmail.com

Sean Turner
sn3rd

Email: sean@sn3rd.com

LAMPS
Internet-Draft
Intended status: Standards Track
Expires: 16 August 2022

M. Ounsworth
J. Gray
S. Mister
Entrust
12 February 2022

Composite Encryption For Use In Internet PKI
draft-ounsworth-pq-composite-encryption-01

Abstract

With the widespread adoption of post-quantum cryptography will come the need for an entity to possess multiple public keys on different cryptographic algorithms. Since the trustworthiness of individual post-quantum algorithms is at question, a multi-key cryptographic operation will need to be performed in such a way that breaking it requires breaking each of the component algorithms individually. This requires defining new structures for holding composite encryption data.

This document defines a content encryption process following the hybrid model as described in the NIST Post-Quantum Crypto FAQ. This draft defines three composite encryption modes. First, Composite Key Transport using Encryption primitives which encrypts a message (typically a content encryption key) for a recipient with a composite public key composed entirely of encryption keys by encrypting it with multiple one-time-pad keys, each encrypted under a different recipient public key. Second, Composite Key Transport using Encryption and KEM primitives is the generalization of the previous mode to support a mixture of encryption and KEM algorithms. Third, Composite Key Exchange is the most general and supports establishing a shared secret using any combination of encryption, KEM, and key exchange primitives where a master shared secret is generated using NIST SP 800-56Cr2.

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 16 August 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. Terminology | 4 |
| 2. Composite Key Transport using Encryption primitives | 5 |
| 2.1. Algorithm Identifier | 6 |
| 2.2. Public key and key usage | 6 |
| 2.2.1. Composite-OR | 6 |
| 2.3. Algorithm parameters | 7 |
| 2.4. Encryption process | 7 |
| 2.5. Decryption process | 8 |
| 3. Composite Key Transport using Encryption and KEM primitives | 10 |
| 3.1. Algorithm Identifier | 10 |
| 3.2. Public key and key usage | 11 |
| 3.2.1. Composite-OR | 11 |
| 3.3. Algorithm parameters | 11 |
| 3.4. Encryption process | 11 |
| 3.5. Decryption process | 13 |
| 4. Composite Key Exchange | 13 |
| 4.1. Algorithm Identifier | 13 |
| 4.2. Public key and key usage | 14 |
| 4.2.1. Composite-OR | 14 |
| 4.3. Algorithm parameters | 14 |
| 4.4. Encapsulation Process | 15 |
| 4.5. Decapsulation Process | 17 |
| 5. In Practice | 19 |

| | |
|---|----|
| 6. IANA Considerations | 19 |
| 7. Security Considerations | 19 |
| 7.1. IID property of KEM primitives | 19 |
| 7.2. Composite-OR modes | 20 |
| 7.3. Policy for Deprecated or Unacceptable Algorithms | 20 |
| 8. Appendices | 20 |
| 8.1. ASN.1 Module | 20 |
| 8.2. Intellectual Property Considerations | 20 |
| 8.3. Making contributions | 21 |
| 9. Normative References | 21 |
| Authors' Addresses | 22 |

1. Introduction

During the transition to post-quantum cryptography, there will be uncertainty as to the strength of cryptographic algorithms; we will no longer fully trust traditional cryptography such as RSA, Diffie-Hellman, DSA and their elliptic curve variants, but we will also not fully trust their post-quantum replacements until they have had sufficient scrutiny. Unlike previous cryptographic algorithm migrations, the choice of when to migrate and which algorithms to migrate to, is not so clear. Even after the migration period, it may be advantageous for an entity's cryptographic identity to be composed of key pairs associated with different public-key algorithms.

The deployment of composite public keys and composite encryption using post-quantum algorithms will face two challenges:

- * Algorithm strength uncertainty: During the transition period, some post-quantum signature and encryption algorithms will not be fully trusted, while the trust in legacy public key algorithms will start to erode. A relying party may learn some time after deployment that a public key algorithm has become untrustworthy, but in the interim, they may not know which algorithm an adversary has compromised.
- * Backwards compatibility: During the transition period, post-quantum algorithms will not be supported by all clients.

This document provides mechanisms to address algorithm strength uncertainty by building on `reference draft-ounsworth-pq-composite-pubkeys` by providing formats for both wrapping a content encryption key using multiple public key encryption mechanisms, or performing key exchange using a combination of encryption, key encapsulation, and key exchange primitives. The issue of backwards compatibility is addressed with support for Composite Or recipient keys in each mode.

This document is intended for general applicability anywhere that content encryption or key exchange is used.

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.

The following terms are used in this document:

ALGORITHM: An information object class for identifying the type of cryptographic operation to be performed. This document is primarily concerned with algorithms for producing encryption keys.

BER: Basic Encoding Rules (BER) as defined in [X.690].

COMPONENT ALGORITHM: A single basic algorithm which is contained within a composite algorithm.

COMPOSITE ALGORITHM: An algorithm which is a sequence of one or more component algorithms..

DER: Distinguished Encoding Rules as defined in [X.690].

PUBLIC / PRIVATE KEY: The public and private portion of an asymmetric cryptographic key, making no assumptions about which algorithm.

PRIMITIVE PUBLIC KEY / SIGNATURE: A public key or signature object of a non-composite algorithm type.

SIGNATURE: A digital cryptographic signature, making no assumptions about which algorithm.

SECRET or SHARED SECRET: Cryptographic material established between two parties. May be generated by one party and send encrypted to the other, or may be the output of an exchange of public information between two or more parties that generates a unique shared value for all involved parties.

KEY DERIVATION FUNCTION: A function used to derive secure secret keys using shared secrets, hashing and other cryptographic primitives.

COMPOSITE ENCRYPTION KEY: A structure that contains a sequence of content encryption keys, or secrets used to derive a content encryption keys.

2. Composite Key Transport using Encryption primitives

In this composite encryption mode, a message to be encrypted is provided by the calling application. This message to be encrypted is assumed to have length less than the maximum message size of the chosen encryption algorithms, as is the case when a suitably-sized symmetric key is encrypted.

This mode is compatible with protocols requiring a key transport primitive, such as CMS' KeyTransRecipientInfo [RFC5652].

Composite Key Transport using Encryption primitives uses a trivial XOR one-time-pad scheme, as defined in Section 2.4. It transports n one-time-pad secret keys of the same length as the content to be encryption, where n is the number of recipient component public keys, and each one-time-pad secret key is encrypted under a different recipient component public key. The trivial XOR key-sharing scheme requires the recipient to use all component private keys in order to recover the content encryption key. Note that it would be possible to use an "n of m" or "threshold" secret sharing scheme if it was desired for the recipient to be able to complete the key transport using a subset of their private keys, but that mechanism is not defined in this document.

EDNOTE: we have not been able to find a reference and security analysis for the trivial XOR key-sharing scheme. This may need review by CFRG. We could re-frame this process as "a one-time pad with $n-1$ one-time pad keys, which we transport using the recipients public keys", then this could leverage one-time pad security analysis.

Composite encryption uses the following structure:

EDNOTE: Should a different composite OID be used to determine the type of composite encryption (Key Transport or Key Agreement?). Probably not because the desired key usage will be handled in the protocols that uses this primitive.

CompositeEncryptedKey ::= EncryptedKey{ SEQUENCE SIZE (2..Max) OF OCTET STRING}

EDNOTE: This ASN.1 probably does not compile. The intent is that this fits into any EncryptedKey field, but defines some structure within the existing EncryptedKey ::= OCTET STRING, but I'm not sure exactly how to specify that.

Where each OCTET STRING within the SEQUENCE contains an encrypted one-time-pad secret key encrypted under one of the recipient component public keys. The CompositeEncryptedKey MUST list encrypted values in the same order as the recipient public key's component keys.

2.1. Algorithm Identifier

The id-alg-composite-encryption object identifier MUST be used to identify the usage of this mode

```
id-alg-composite-encryption OBJECT IDENTIFIER ::= {  
  id-alg-composite-encryption OBJECT IDENTIFIER ::= {  
    joint-iso-itu-t(2) country(16) us(840) organization(1) entrust(114027)  
    Algorithm(80) Composite(4) id-alg-composite-encryption(4) }
```

EDNOTE: this is a temporary OID for the purposes of prototyping. Permanent OIDs should be requested from IANA, see Section 6.

2.2. Public key and key usage

The recipient MUST have a composite public key which supports key transport operations. Where the recipient public key has an associated keyUsage as specified in [RFC5280], it MUST have keyUsage: keyEncipherment. In other words, the mechanism specified in this section applies only if all of the recipient's public keys are associated with encryption algorithms.

2.2.1. Composite-OR

The design intent of this mode is to support migration scenarios where a recipient has been provisioned with a composite key containing algorithms that its peers may not yet support. This mode allows the sender to encrypt for a subset of the recipient's public keys. Support for Composite OR subset encryption is indicated by the recipient at key generation time by marking its composite key with the id-composite-or-key algorithm identifier as defined in ~~~cite properly draft-ounsworth-pq-composite-keys~~~. To maximize security strength of the ciphertext, clients SHOULD encrypt for as many keys as they support and as the migration and compatibility situation allow.

Policy mechanisms defining allowed subsets of algorithms could be applied here, but are out of scope of this document. As defined in this document, a recipient marking their public key as id-composite-or-key must accept the risk that a sender may encrypt sensitive data for it using any one of its component keys in isolation. Composite Or is a direct tradeoff of lower security for increased migration flexibility.

2.3. Algorithm parameters

The composite key transport using encryption mode does not require additional parameters, and therefore any associated Params are ABSENT.

2.4. Encryption process

The process for performing Composite Key Transport using Encryption primitives is as follows:

The first $n-1$ one-time-pad keys are random bit strings of the same length as the content encryption key. The final one-time-pad key is computed by XOR'ing the content encryption key with each of $n-1$ previous keys.

Input:

| | |
|------------------------|---|
| n | The number of recipient component public keys |
| P_1, P_2, \dots, P_n | Recipient component public keys |
| A_1, A_2, \dots, A_n | Cryptographic algorithms to be used with public keys P_1, P_2, \dots, P_n |
| CEK | The Content Encryption Key |
| SIZE | The size of the Content Encryption Key in bits |

Output:

| | |
|------------------------|--|
| E_1, E_2, \dots, E_n | EncryptedKey values corresponding to each recipient component public key |
|------------------------|--|

Intermediate values:

| | |
|----------------------------|---|
| S_1, S_2, \dots, S_{n-1} | One-time-pad secret keys to be encapsulated by each component algorithm |
| C | One-time-pad ciphertext of the CEK under S_1, S_2, \dots, S_{n-1} |

Generation Procedure:

1. If recipient public key is of type id-composite-or-key, determine the


```
index of the last recipient public key to be encrypted for
  i_last := index of last Pi to be encrypted for
Else,
  i_last = n
```

2. To generate secret keys S_n , compute the following

```
C = CEK
for i := 1 to n

  a. If id-composite-or-key and Pi is to be skipped
    Ei := emptyOctetString
    continue to next i

  b. If i == i_last
    Ei = encrypt(C, Pi, Ai)
    break
  Else,
    Si := random_bits(SIZE)
    C := C XOR Si
    Ei = encrypt(Si, Pi, Ai)
```

3. Output E_1, E_2, \dots, E_n

Where `random_bits(SIZE)` is a cryptographically-secure random bit generator outputting `SIZE` bits, and where `emptyOctetString` is the octet string of length 0.

EDNOTE: we currently do not define a composite `algorithmID` type to carry A_1, A_2, \dots, A_n . We may need to add one analogously to the `CompositeParams ::= SEQUENCE SIZE (2..MAX) OF AlgorithmIdentifier` that we have in the composite signatures draft.

If the sender does not support Composite Or encryption, this algorithm may be simplified by omitting step 1, 2a, and the `if i == i_last` statement in 2b.

The design intent is that Composite Or encryption with a single recipient key collapses to being equivalent to direct encryption of the CEK.

2.5. Decryption process

To obtain the content-encryption key from a `CompositeEncryptedKey`, each component algorithm **MUST** be used to decrypt the set of one-time-pad keys. The keys are then XOR'ed together to recover the content encryption key.

Input:

| | |
|--------------------|---|
| n | The number of recipient component public keys |
| SK1, SK2, ..., SKn | Recipient component secret keys |
| A1, A2, ..., An | Cryptographic algorithms to be used with public keys P1, P2, ..., Pn |
| E1, E2, ..., En | EncryptedKey values corresponding to each recipient component public key |

Intermediate values:

| | |
|-----------------|--|
| S1, S2, ..., Sn | One-time-pad keys and ciphertext to be decapsulated by each component algorithm |
|-----------------|--|

Output:

| | |
|-----|----------------------------|
| CEK | The Content Encryption Key |
|-----|----------------------------|

Generation Procedure:

1. Recover each one-time-pad key
for i := 1 to n
 if Ei == emptyOctetString
 Si := emptyOctetString
 Else,
 Si := decrypt(Ei, SKi)
2. Recover the CEK;
For each one-time-pad key
 CEK = S1
 for i := 2 to n
 if Si != emptyOctetString
 CEK = CEK XOR Si
3. Output CEK

The if statement in step 1 (and ensuring its proper bit length for the XOR in step 2) is the only modification required to support Composite Or encryption. The designers have intentionally omitted a check that the recipient key is of type id-composite-or-key because even if the sender erroneously used composite or subset encryption for a recipient key which is not of type id-composite-or-key, the damage has already been done by encrypting and transmitting the data, no further harm can be done by decrypting it. However, where appropriate, clients SHOULD indicate a warning to users that this data was transmitted with weaker encrypting than their public key allows.

EDNOTE: investigate whether this is actually a special case of the next mechanism, and therefore both sections can be folded together.

3. Composite Key Transport using Encryption and KEM primitives

This composite encryption mode is the generalization of the mode defined in Section 2 to support a composite recipient public key which may contain a mixture of one or more encryption component algorithms with zero or more key encapsulation mechanism (KEM) component algorithms.

This mode is compatible with protocols requiring a key transport primitive, such as CMS' KeyTransRecipientInfo [RFC5652].

Security consideration: for a recipient composite public key to be applicable to this mode, all component KEMs MUST produce a shared secret whose bits are independent and uniformly distributed (aka "uniformly IID" or "uniformly random" or "full entropy") and therefore the shared secret is safe to use directly as a symmetric key. If a recipient public key contains component KEMs which are not known to have this property, then implementors SHOULD use the more general mode described in Section 4 which incorporates the use of a key derivation function. See Section 7.1 for a further discussion of this security consideration.

EDNOTE: also put this in the Security considerations section.

3.1. Algorithm Identifier

The id-alg-composite-kem object identifier MUST be used to identify the usage of this mode

```
id-alg-composite-kem OBJECT IDENTIFIER ::= {
  id-alg-composite-encryption OBJECT IDENTIFIER ::= {
    joint-iso-itu-t(2) country(16) us(840) organization(1) entrust(114027)
    Algorithm(80) Composite(4) id-alg-composite-kem(5) }
```

EDNOTE: this is a temporary OID for the purposes of prototyping. Permanent OIDs should be requested from IANA, see Section 6.

3.2. Public key and key usage

The recipient MUST have a composite public key which supports key transport or key encapsulation operations. Where the recipient public key has an associated keyUsage as specified in [RFC5280], it MUST have keyUsage: keyEncipherment. In other words, the mechanism specified in this section applies only if all of the recipient's public keys are encryption or KEM algorithms.

In addition, for a recipient composite public key to be applicable to this mode, all component KEMs MUST be capable of producing a shared secret of SIZE bits, where SIZE is the length in bits of the content encryption key (CEK) to be transported. This is assumed for the remainder of this section.

3.2.1. Composite-OR

The design intent of this mode is to support migration scenarios where a recipient has been provisioned with a composite key containing algorithms that its peers may not yet support. This mode allows the sender to encrypt for a subset of the recipient's public keys. Support for Composite OR subset encryption is indicated by the recipient at key generation time by marking its composite key with the id-composite-or-key algorithm identifier as defined in ~~~cite properly draft-ounsworth-pq-composite-keys~~~.

Policy mechanisms defining allowed subsets of algorithms could be applied here, but are out of scope of this document. As defined in this document, a recipient marking their public key as id-composite-or-key must accept the risk that a sender may encrypt sensitive data for it using any one of its component keys in isolation. Composite Or is a direct tradeoff of lower security for increased migration flexibility.

3.3. Algorithm parameters

The composite key transport using encryption and KEM mode does not require additional parameters, and therefore any associated Params are ABSENT.

3.4. Encryption process

Given these conditions are met, the encryption process defined in Section 2.4 is modified as follows:

Input:

| | |
|-----------------|--|
| n | The number of recipient component public keys |
| P1, P2, ..., Pn | Recipient component public keys |
| CEK | The Content Encryption Key |
| SIZE | The size of the Content Encryption Key in bits |

Output:

| | |
|-----------------|--|
| E1, E2, ..., En | EncryptedKey values corresponding to each recipient component public key |
|-----------------|--|

Intermediate values:

| | |
|-----------------|---|
| S1, S2, ..., Sn | One-time-pad secret keys to be encapsulated by each component algorithm |
|-----------------|---|

Generation Procedure:

1. If recipient public key is of type id-composite-or-key, determine the index of the last recipient public key to be encrypted for
 $i_last := \text{index of last } P_i \text{ to be encrypted for}$
 Else,
 $i_last = n$
2.
 - for i := 1 to n
 - a. if id-composite-or-key and P_i is to be skipped
 $E_i := \text{emptyOctetString}$
 continue to next i
 - b. If $i == i_last$
 continue to next i
 Else, if P_i is of type KEM:
 $S_i, E_i := \text{encaps}(P_i)$
 $CEK := CEK \text{ XOR } S_i$
 Else:
 $S_i := \text{random_bits}(\text{SIZE})$
 $CEK := CEK \text{ XOR } S_i$
 $E_i := \text{encrypt}(S_i, P_i)$
3. Encrypt the final CEK value
 $E_{i_last} = \text{encrypt}(CEK, P_{i_last})$
4. Output E1, E2, ..., En

Where `random_bits(SIZE)` is a cryptographically-secure random bit generator outputting SIZE bits, and where `emptyOctetString` is the octet string of length 0.

If the sender does not support Composite Or encryption, this algorithm may be simplified by omitting step 1, 2a, and the if `i == i_last` statement in 2b.

The design intent is that Composite Or encryption with a single recipient key collapses to being equivalent to direct encryption of the CEK.

3.5. Decryption process

The decryption process defined in Section 2.5 applies directly where `decrypt()` is substituted for `decaps()` when the underlying primitive is a KEM.

4. Composite Key Exchange

This mode is the most general in that it supports a composite recipient public key which MAY contain an arbitrary mixture of encryption, key encapsulation mechanism (KEM), and key agreement component algorithms. Due to the nature of key agreement algorithms, this mode cannot take a content encryption key as input, but instead generates a master shared secret as an output. As such, the nomenclature in this mode differs from the modes above.

This mode is compatible with protocols requiring a key agreement primitive, such as CMS' `KeyAgreeRecipientInfo` [RFC5652].

Composite key exchange uses the underlying primitive to either encrypt for, encapsulate, or interactively do key agreement with each of the recipient's public keys, then all shared secrets are concatenated together and a KDF is applied as prescribed by NIST SP 800-56Cr2 [SP80056cr2].

4.1. Algorithm Identifier

The `id-alg-composite-keyex` object identifier MUST be used to identify the usage of this mode

```
id-alg-composite-keyex OBJECT IDENTIFIER ::= {
  id-alg-composite-encryption OBJECT IDENTIFIER ::= {
    joint-iso-itu-t(2) country(16) us(840) organization(1) entrust(114027)
    Algorithm(80) Composite(4) id-alg-composite-encryption(6) }
```

EDNOTE: this is a temporary OID for the purposes of prototyping. Permanent OIDs should be requested from IANA, see Section 6.

4.2. Public key and key usage

The recipient MUST have a composite public key which supports key transport, key encapsulation, or key exchange operations. Where the recipient public key has an associated keyUsage as specified in [RFC5280], it MUST have keyUsage: keyEncipherment, keyAgreement. This mode is the most general and places the fewest restrictions on the recipient public key.

EDNOTE: I think this violates our public key draft where we say that the public key's KU MUST apply to all components. ... we did not want mixing of signatures and encryption keys, but I think in this case we do want to allow mixing of keyEncipherment and keyExchange keys. Not sure how to fix that.

4.2.1. Composite-OR

The design intent of this mode is to support migration scenarios where a recipient has been provisioned with a composite key containing algorithms that its peers may not yet support. This mode allows the sender to encrypt for a subset of the recipient's public keys. Support for Composite OR subset encryption is indicated by the recipient at key generation time by marking its composite key with the id-composite-or-key algorithm identifier as defined in `~~~cite properly draft-ounsworth-pq-composite-keys~~~`.

Policy mechanisms defining allowed subsets of algorithms could be applied here, but are out of scope of this document. As defined in this document, a recipient marking their public key as id-composite-or-key must accept the risk that a sender may encrypt sensitive data for it using any one of its component keys in isolation. Composite Or is a direct tradeoff of lower security for increased migration flexibility.

4.3. Algorithm parameters

The composite key exchange mode requires additional parameters to specify the KDF used to combine shared secrets into a master shared secret.

Params ::= KeyDerivationAlgorithmIdentifier

The KeyDerivationAlgorithmIdentifier type is specified in [RFC5652]. The KeyDerivationAlgorithmIdentifier definition is repeated here for completeness.

KeyDerivationAlgorithmIdentifier ::= AlgorithmIdentifier

4.4. Encapsulation Process

Composite key exchange uses the underlying primitive to either encrypt for, encapsulate, or interactively do key agreement with each of the recipient's public keys, then all shared secrets are concatenated together and a KDF is applied as prescribed by NIST SP 800-56Cr2 [SP80056cr2].

Input:

| | |
|-----------------|---|
| P1, P2, ..., Pn | Public keys for the n component encryption algorithms, a CompositePublicKey |
| SIZE | The size, in bits, for shared secrets to be combined by both parties into a content encryption key. This value SHOULD correspond to the size of the content encryption key. |
| KDF | A key derivation function |

Output:

| | |
|-----------------|--|
| E1, E2, ..., En | EncryptedKey values corresponding to each recipient component public key |
| M | Master shared secret |

Ciphertext and master secret Generation Procedure:

1. Generate a set of one-time-pad secret keys of the same length as the content encryption key
 - for i := 1 to n
 - a. if id-composite-or-key and Pi is to be skipped


```
Si = emptyOctetString
Ei := emptyOctetString
continue to next i
```
 - b. if P1 is of type KEM or keyExchange:


```
Si,Ei := encaps(Pi)
else:
  Si := random_bits(SIZE)
  Ei := encrypt(Si, Pi)
```
2. Generate Z via concatenation


```
Z = S1 || S2 || .. || Sn
```
3. Generate the master shared secret via a KDF


```
M = KDF(Z)
```
4. Output M


```
Output E1, E2, ..., En
```

Where emptyOctetString is the octet string of length 0 that serves as a no-op or identity element for the concatenation in step 2.

In cases where KDF is extensible output function, the length of M must be carried in the KeyDerivationAlgorithmIdentifier defined in Section 4.3.

EDNOTE: It isn't clear to us how one uses the defined HKDF algorithmid (RFC 8619) here. Those OIDs specify a hash, but no output length or seed or info parameter either implicitly or explicitly. But we also don't see how it would be used with CMS either, for the same reason. ..?

If the sender does not support Composite Or encryption, this algorithm may be simplified by omitting step 2a.

EDNOTE: investigate whether step 3 really belongs here, or whether the surrounding protocol (ex. CMS EnvelopedData) will perform a final KDF anyways. We believe that outputting an IID master secret is consistent with modern KEM behaviour.

4.5. Decapsulation Process

Input:

n The number of recipient component public keys
 SK_1, SK_2, \dots, SK_n Recipient component secret keys
 E_1, E_2, \dots, E_n EncryptedKey values corresponding to each recipient component public key
 KDF A key derivation function

Intermediate values:

S_1, S_2, \dots, S_n Shared secrets to be encapsulated by each component algorithm

Output:

M Master shared secret

Master Secret Recovery Procedure:

1. Recover each shared secret
 - for $i := 1$ to n
 - if $E_i == \text{null}$
 - $S_i = \text{EMPTY_STRING}$
 - $S_i := \text{decrypt_or_decaps}(E_i, SK_i)$
2. Generate Z via concatenation
 - $Z = S_1 || S_2 || \dots || S_n$
3. Generate the master shared secret via a KDF
 - $M = KDF(Z)$
4. Output M

"EMPTY_STRING" indicates a string or byte array of length zero so that that value is essentially omitted from the concatenation in step 2.

The if statement in step 1 is the only modification required to support Composite Or encryption. The designers have intentionally omitted a check that the recipient key is of type id-composite-or-key because even if the sender erroneously used composite or subset for a recipient key which is not of type id-composite-or-key, the damage has already been done by generating a master secret and potentially transmitting data encrypted with it, no further harm can be done by decrypting it. However, where appropriate, clients SHOULD indicate a warning to users that this data was transmitted with weaker encrypting than their public key allows.

5. In Practice

This section addresses practical issues of how this draft affects other protocols and standards.

6. IANA Considerations

The following OIDs are to be assigned by IANA. The authors suggest that IANA assign OIDs for composite encryption on the id-pkix arc:

```
id-alg-composite OBJECT IDENTIFIER ::= {  
    iso(1) identified-organization(3) dod(6) internet(1) security(5)  
    mechanisms(5) pkix(7) algorithms(6) composite(??) id-alg-composite-encryption  
    (??)}
```

```
id-alg-composite-kem OBJECT IDENTIFIER ::= {  
    iso(1) identified-organization(3) dod(6) internet(1) security(5)  
    mechanisms(5) pkix(7) algorithms(6) composite(??) id-alg-composite-kem(??)}
```

```
id-alg-composite-keyex OBJECT IDENTIFIER ::= {  
    iOBJECT IDENTIFIER ::= {  
        iso(1) identified-organization(3) dod(6) internet(1) security(5)  
        mechanisms(5) pkix(7) algorithms(6) composite(??) id-alg-composite-encryption  
        (??)}
```

7. Security Considerations

7.1. IID property of KEM primitives

Composite Key Transport using Encryption and KEM primitives defined in Section 3 directly uses the shared secret output from the underlying KEM primitive as a one-time-pad key to encrypt the CEK. Therefore the output of the KEM primitive needs to meet the security properties of a one-time-pad key, namely that its bits are independent and identically distributed (IID). In particular, key agreement schemes such as ECDH or SIKE do not produce shared secrets that meet this requirement and therefore MUST use the fully general mechanism Composite Key Exchange defined in Section 4.

EDNOTE: Should this be brought to CFRG to decide which KEMs are appropriate to use with this mechanism? It may be possible that we need to run the KEM output through a KDF; but we're trying to avoid needing to carry a KDF AlgID here.

7.2. Composite-OR modes

Composite-OR eases migration at the expense of security. For composite encryption and key encapsulation, the weakening of security is entirely at the discretion of the sender, since once data has been encrypted and transmitted with weak ciphers, there is nothing the recipient can do to protect the data against record & decrypt attacks. Clients performing Composite Or encryption operations **MUST** ensure that the recipient's public key is of type id-composite-or-key before producing a ciphertext with a subset of the recipient's public keys.

For some cases of composite key exchange, notably when the underlying key exchange primitive is used in a fully interactive (aka "ephemeral-ephemeral") mode, the sender cannot begin encrypting data until the recipient has completed the key exchange. The recipient **SHOULD** reject the connection if one or more null ciphertexts are encountered when the recipient's public key is not of type id-composite-or-key.

7.3. Policy for Deprecated or Unacceptable Algorithms

Within the context of composite encryption, the sender holds the responsibility to ensure that chosen algorithms are of sufficient strength prior to encrypting and transmitting sensitive data under them. Composite is designed to provide security redundancy and to remain strong as long as at least one of the component algorithms remains strong.

When encrypting for a Composite-OR public key and using a subset of the recipient's public key, then these redundancy guarantees no longer apply. The sender **SHOULD** employ a policy mechanism to ensure that they are using a combination of algorithms of sufficient strength. Even though this document does not define such a policy mechanism, but implementors making use of Composite-OR encryption are strongly encouraged to implement a policy mechanism.

8. Appendices

8.1. ASN.1 Module

~~ TODO ~~

8.2. Intellectual Property Considerations

The following IPR Disclosure relates to this draft:

<https://datatracker.ietf.org/ipr/3588/>

We are grateful to all, including any contributors who may have been inadvertently omitted from this list.

This document borrows text from similar documents, including those referenced below. Thanks go to the authors of those documents.
"Copying always makes things easier and less error prone" -
[RFC8411].

8.3. Making contributions

Additional contributions to this draft are welcome. Please see the working copy of this draft at, as well as open issues at:

<https://github.com/EntrustCorporation/draft-ounsworth-pq-composite-encryption>

9. 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>>.
- [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>>.
- [RFC3211] Gutmann, P., "Password-based Encryption for CMS", RFC 3211, DOI 10.17487/RFC3211, December 2001, <<https://www.rfc-editor.org/info/rfc3211>>.
- [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>>.
- [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>>.
- [RFC5652] Housley, R., "Cryptographic Message Syntax (CMS)", STD 70, RFC 5652, DOI 10.17487/RFC5652, September 2009, <<https://www.rfc-editor.org/info/rfc5652>>.

- [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>>.
- [RFC8411] Schaad, J. and R. Andrews, "IANA Registration for the Cryptographic Algorithm Object Identifier Range", RFC 8411, DOI 10.17487/RFC8411, August 2018, <<https://www.rfc-editor.org/info/rfc8411>>.
- [SP80056cr2] NIST, "SP 800-56c Rev. 2: Recommendation for Key-Derivation Methods in Key-Establishment Schemes", August 2020.
- [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 8825-1:2015, November 2015.

Authors' Addresses

Mike Ounsworth
Entrust Limited
2500 Solandt Road -- Suite 100
Ottawa, Ontario K2K 3G5
Canada

Email: mike.ounsworth@entrust.com

John Gray
Entrust Limited

Email: john.gray@entrust.com

Serge Mister
Entrust Limited

Email: serge.mister@entrust.com

LAMPS
Internet-Draft
Intended status: Standards Track
Expires: 16 August 2022

M. Ounsworth (Editor)
Entrust
M. Pala
CableLabs
12 February 2022

Composite Public and Private Keys For Use In Internet PKI
draft-ounsworth-pq-composite-keys-01

Abstract

With the widespread adoption of post-quantum cryptography will come the need for an entity to possess multiple public keys on different cryptographic algorithms. Since the trustworthiness of individual post-quantum algorithms is at question, a multi-key cryptographic operation will need to be performed in such a way that breaking it requires breaking each of the component algorithms individually. This requires defining new structures for holding composite keys, for use with composite signature and encryption data.

This document defines the structures `CompositePublicKey`, `CompositePrivateKey`, which are sequences of the respective structure for each component algorithm. This document makes no assumptions about what the component algorithms are, provided that they have defined algorithm identifiers. The only requirement imposed by this document is that all algorithms be of the same key usage; i.e. all signature or all encryption. This document is intended to be coupled with corresponding documents that define the structure and semantics of composite signatures and encryption.

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 16 August 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. Terminology | 3 |
| 2. Composite Structures | 4 |
| 2.1. Algorithm Identifier | 5 |
| 2.1.1. Composite Public Key | 5 |
| 2.1.2. Composite-OR Public Key | 5 |
| 2.2. Composite Keys | 6 |
| 2.2.1. Key Usage | 6 |
| 2.3. Composite Public Key | 6 |
| 2.4. Composite Private Key | 7 |
| 2.5. Encoding Rules | 7 |
| 3. In Practice | 8 |
| 3.1. Textual encoding of Composite Private Keys | 8 |
| 3.2. Asymmetric Key Packages (CMS) | 8 |
| 4. IANA Considerations | 9 |
| 5. Security Considerations | 9 |
| 5.1. Reuse of keys in a Composite public key | 9 |
| 5.2. Policy for Deprecated and Acceptable Algorithms | 10 |
| 5.3. Protection of Private Keys | 10 |
| 5.4. Checking for Compromised Key Reuse | 11 |
| 6. Appendices | 11 |
| 6.1. ASN.1 Module | 11 |
| 6.2. Intellectual Property Considerations | 12 |
| 7. Contributors and Acknowledgements | 12 |
| 7.1. Making contributions | 13 |
| 8. Normative References | 13 |
| Authors' Addresses | 14 |

1. Introduction

During the transition to post-quantum cryptography, there will be uncertainty as to the strength of cryptographic algorithms; we will no longer fully trust traditional cryptography such as RSA, Diffie-Hellman, DSA and their elliptic curve variants, but we will also not fully trust their post-quantum replacements until they have had sufficient scrutiny. Unlike previous cryptographic algorithm migrations, the choice of when to migrate and which algorithms to migrate to, is not so clear. Even after the migration period, it may be advantageous for an entity's cryptographic identity to be composed of multiple public-key algorithms.

The deployment of composite public keys, and composite signatures and composite encryption using post-quantum algorithms will face two challenges

- * **Algorithm strength uncertainty:** During the transition period, some post-quantum signature and encryption algorithms will not be fully trusted, while also the trust in legacy public key algorithms will start to erode. A relying party may learn some time after deployment that a public key algorithm has become untrustworthy, but in the interim, they may not know which algorithm an adversary has compromised.
- * **Backwards compatibility:** During the transition period, post-quantum algorithms will not be supported by all clients.

This document provides a mechanism to address algorithm strength uncertainty by providing formats for encoding multiple public keys and private keys values into existing public key and private key fields.

This document is intended for general applicability anywhere that keys are used within PKIX or CMS structures.

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.

The following terms are used in this document:

ALGORITHM: An information object class for identifying the type of cryptographic key being encapsulated.

BER: Basic Encoding Rules (BER) as defined in [X.690].

COMPONENT ALGORITHM: A single basic algorithm which is contained within a composite algorithm.

COMPOSITE ALGORITHM: An algorithm which is a sequence of two or more component algorithms, as defined in Section 2.

DER: Distinguished Encoding Rules as defined in [X.690].

PUBLIC / PRIVATE KEY: The public and private portion of an asymmetric cryptographic key, making no assumptions about which algorithm.

2. Composite Structures

In order for public keys and private keys to be composed of multiple algorithms, we define encodings consisting of a sequence of public key or private key primitives (aka "component algorithms") such that these structures can be used as a drop-in replacement for existing public key fields such as those found in PKCS#10 [RFC2986], CMP [RFC4210], X.509 [RFC5280], CMS [RFC5652], and the Trust Anchor Format [RFC5914].

This section defines the following structures:

- * The id-alg-composite is an OID identifying a composite public key.
- * The CompositePublicKey carries all the public keys associated with an identity within a single public key structure.
- * The CompositePrivateKey carries all the private keys associated with an identity within a single private key structure.

EDNOTE 2: We have heard community feedback that the ASN.1 structures presented here are too flexible in that allow arbitrary combinations of an arbitrary number of signature algorithms. The feedback is that this is too much of a "footgun" for implementors and sysadmins. We are working on an alternative formulation using ASN.1 information object classes that allow for compiling explicit pairs of algorithmIDs. We would love community feedback on which approach is preferred. See slide 30 of this presentation: <https://datatracker.ietf.org/meeting/interim-2021-lamps-01/materials/slides-interim-2021-lamps-01-sessa-position-presentation-by-mike-ounsworth-00.pdf>

2.1. Algorithm Identifier

2.1.1. Composite Public Key

The Composite algorithm identifier is used for identifying a public key and a private key. Additional encoding information is provided below for each of these objects.

When using this algorithm identifier it is implied that all component keys **MUST** be used in an AND relation; any cryptographic operation using this composite public key **MUST** use the it as an atomic object and use all component keys. This mode has the strongest security properties and is **RECOMMENDED**.

There is an additional security consideration that some use cases such as signatures remain secure against downgrade attacks if and only if component keys are never used in isolation and therefore it is **RECOMMENDED** that component keys in a composite key are uniquely generated.

```
id-composite-key OBJECT IDENTIFIER ::= {  
    joint-iso-itu-t(2) country(16) us(840) organization(1) entrust(114027) Algori  
    thm(80) Composite(4) CompositeKey(1) }
```

EDNOTE 3: this is a temporary OID for the purposes of prototyping. We are requesting IANA to assign a permanent OID, see Section 4.

2.1.2. Composite-OR Public Key

EDNOTE: This section was written with the intention of keeping the primary Composite OID reserved for the simple and strict mode; if you want to do either a simple OR, or a custom policy then we have given a different OID. We are still debating whether this is useful to specify at issuing time, or whether this is adding needless complexity to the draft.

The Composite-OR algorithm identifier is used for identifying a public key and a private key. Additional encoding information is provided below for each of these objects.

When using this algorithm identifier, component keys **MAY** be used in an OR relation meaning that any one of the component keys may be used by itself. Implementors may also define more complex processes and policies using this algorithm identifier, for example allowing some algorithms by themselves and others only in combination. This mode is provided for applications that need to issue long-lived composite keys in a way that allows for backwards compatibility or staged adoption of new algorithms.

```
id-composite-or-key OBJECT IDENTIFIER ::= {  
    joint-iso-itu-t(2) country(16) us(840) organization(1) entrust(114027) Algori  
    thm(80) Composite(4) entrust-Algorithm-Composite-OR(3) }
```

2.2. Composite Keys

A composite key is a single key object that performs an atomic signature or verification operation, using its encapsulated sequence of component keys.

The ASN.1 algorithm object for composite public and private keys is:

```
pk-Composite PUBLIC-KEY ::= {  
    IDENTIFIER id-alg-composite  
    KEY CompositePublicKey  
    PARAMS ARE absent  
    PRIVATE-KEY CompositePrivateKey  
}
```

EDNOTE 4: the authors are currently unsure whether the params should be absent (ie this structure simply says "I am a composite algorithm"), or used to duplicate some amount of information about what the component algorithms are. See Section 2.3 for a longer ENDOTE on this.

2.2.1. Key Usage

For protocols such as X.509 [RFC5280] that specify key usage along with the public key, any key usage may be used with Composite keys, with the requirement that the specified key usage MUST apply to all component keys. For example if a Composite key is marked with a KeyUsage of digitalSignature, then all component keys MUST be capable of producing digital signatures. id-alg-composite MUST NOT be used to implement mixed-usage keys, for example, where a digitalSignature and a keyEncipherment key are combined together into a single Composite key object.

2.3. Composite Public Key

Composite public key data is represented by the following structure:

```
CompositePublicKey ::= SEQUENCE SIZE (2..MAX) OF SubjectPublicKeyInfo
```

The corresponding AlgorithmIdentifier for a composite public key MUST use the id-alg-composite object identifier, defined in Section 2.1, and the parameters field MUST be absent.

A composite public key MUST contain at least one component public key.

A CompositePublicKey MUST NOT contain a component public key which itself describes a composite key; i.e. recursive CompositePublicKeys are not allowed

EDNOTE: unclear that banning recursive composite keys actually accomplishes anything other than a general reduction in complexity. In particular, with the addition of Composite (AND mode) and Composite-OR (OR mode), recursion actually allows full boolean expression. Is this valuable?

Each element of a CompositePublicKey is a SubjectPublicKeyInfo object for a component public key. When the CompositePublicKey must be provided in octet string or bit string format, the data structure is encoded as specified in Section 2.5.

2.4. Composite Private Key

The composite private key data is represented by the following structure:

CompositePrivateKey ::= SEQUENCE SIZE (2..MAX) OF OneAsymmetricKey

Each element is a OneAsymmetricKey [RFC5958] object for a component private key.

The corresponding AlgorithmIdentifier for a composite private key MUST use the id-alg-composite object identifier, and the parameters field MUST be absent.

A CompositePrivateKey MUST contain at least one component private key, and they MUST be in the same order as in the corresponding CompositePublicKey.

2.5. Encoding Rules

Many protocol specifications will require that the composite public key and composite private key data structures be represented by an octet string or bit string.

When an octet string is required, the DER encoding of the composite data structure SHALL be used directly.

When a bit string is required, the octets of the DER encoded composite data structure SHALL be used as the bits of the bit string, with the most significant bit of the first octet becoming the first bit, and so on, ending with the least significant bit of the last octet becoming the last bit of the bit string.

In the interests of simplicity and avoiding compatibility issues, implementations that parse these structures MAY accept both BER and DER.

3. In Practice

This section addresses practical issues of how this draft affects other protocols and standards.

~~~ BEGIN EDNOTE 10~~~

EDNOTE 10: Possible topics to address:

- \* The size of these certs and cert chains.
- \* In particular, implications for (large) composite keys / signatures / certs on the handshake stages of TLS and IKEv2.
- \* If a cert in the chain is a composite cert then does the whole chain need to be of composite Certs?
- \* We could also explain that the root CA cert does not have to be of the same algorithms. The root cert SHOULD NOT be transferred in the authentication exchange to save transport overhead and thus it can be different than the intermediate and leaf certs.
- \* We could talk about overhead (size and processing).
- \* We could also discuss backwards compatibility.
- \* We could include a subsection about implementation considerations.

~~~ END EDNOTE 10~~~

3.1. Textual encoding of Composite Private Keys

CompositePrivateKeys can be encoded to the Privacy-Enhanced Mail (PEM) [RFC1421] format by placing a CompositePrivateKey into the privateKey field of a PrivateKeyInfo or OneAsymmetricKey object, and then applying the PEM encoding rules as defined in [RFC7468] section 10 and 11 for plaintext and encrypted private keys, respectively.

3.2. Asymmetric Key Packages (CMS)

The Cryptographic Message Syntax (CMS), as defined in [RFC5652], can be used to digitally sign, digest, authenticate, or encrypt the asymmetric key format content type.

When encoding composite private keys, the `privateKeyAlgorithm` in the `OneAsymmetricKey` SHALL be set to `id-alg-composite`.

The parameters of the `privateKeyAlgorithm` SHALL be a sequence of `AlgorithmIdentifier` objects, each of which are encoded according to the rules defined for each of the different keys in the composite private key.

The value of the `privateKey` field in the `OneAsymmetricKey` SHALL be set to the DER encoding of the SEQUENCE of private key values that make up the composite key. The number and order of elements in the sequence SHALL be the same as identified in the sequence of parameters in the `privateKeyAlgorithm`.

The value of the `publicKey` (if present) SHALL be set to the DER encoding of the corresponding `CompositePublicKey`. If this field is present, the number and order of component keys MUST be the same as identified in the sequence of parameters in the `privateKeyAlgorithm`.

The value of the attributes is encoded as usual.

4. IANA Considerations

The ASN.1 module OID is TBD. The `id-composite-key` and `id-composite-or-key` OIDs are to be assigned by IANA. The authors suggest that IANA assign an OID on the `id-pkix` arc:

```
id-composite-key OBJECT IDENTIFIER ::= {  
    iso(1) identified-organization(3) dod(6) internet(1) security(5)  
    mechanisms(5) pkix(7) algorithms(6) composite(??) }
```

5. Security Considerations

5.1. Reuse of keys in a Composite public key

There is an additional security consideration that some use cases such as signatures remain secure against downgrade attacks if and only if component keys are never used in isolation and therefore it is RECOMMENDED that component keys in a composite key are uniquely generated. Note that protocols allowing public keys using the `Composite-OR` algorithm identifier will have a more difficult time preventing downgrade and stripping attacks and therefore it is RECOMMENDED to use the default `AND` mode unless the application has a strong need for backwards compatibility and migration.

5.2. Policy for Deprecated and Acceptable Algorithms

Traditionally, a public key, certificate, or signature contains a single cryptographic algorithm. If and when an algorithm becomes deprecated (for example, RSA-512, or SHA1), it is obvious that structures using that algorithm are implicitly revoked.

In the composite model this is less obvious since a single public key, certificate, or signature may contain a mixture of deprecated and non-deprecated algorithms. Moreover, implementers may decide that certain cryptographic algorithms have complementary security properties and are acceptable in combination even though neither algorithm is acceptable by itself.

Specifying a modified verification process to handle these situations is beyond the scope of this draft, but could be desirable as the subject of an application profile document, or to be up to the discretion of implementers.

2. Check policy to see whether A1, A2, ..., An constitutes a valid combination of algorithms.

```
if not checkPolicy(A1, A2, ..., An), then
    output "Invalid signature"
```

While intentionally not specified in this document, implementors should put careful thought into implementing a meaningful policy mechanism within the context of their signature verification engines, for example only algorithms that provide similar security levels should be combined together.

EDNOTE 11: Max is working on a CRL mechanism to accomplish this.

5.3. Protection of Private Keys

Structures described in this document do not protect private keys in any way unless combined with a security protocol or encryption properties of the objects (if any) where the CompositePrivateKey is used (see next Section).

Protection of the private keys is vital to public key cryptography. The consequences of disclosure depend on the purpose of the private key. If a private key is used for signature, then the disclosure allows unauthorized signing. If a private key is used for key management, then disclosure allows unauthorized parties to access the managed keying material. The encryption algorithm used in the encryption process must be at least as 'strong' as the key it is protecting.

5.4. Checking for Compromised Key Reuse

Certificate Authority (CA) implementations need to be careful when checking for compromised key reuse, for example as required by WebTrust regulations; when checking for compromised keys, you MUST unpack the CompositePublicKey structure and compare individual component keys. In other words, for the purposes of key reuse checks, the composite public key structures need to be un-packed so that primitive keys are being compared. For example if the composite key {RSA1, PQ1} is revoked for key compromise, then the keys RSA1 and PQ1 need to be individually considered revoked. If the composite key {RSA1, PQ2} is submitted for certification, it SHOULD be rejected because the key RSA1 was previously declared compromised even though the key PQ2 is unique.

6. Appendices

6.1. ASN.1 Module

<CODE STARTS>

```
Composite-Signatures-2019
{ TBD }
```

```
DEFINITIONS IMPLICIT TAGS ::= BEGIN
```

```
EXPORTS ALL;
```

```
IMPORTS
```

```
  PUBLIC-KEY, SIGNATURE-ALGORITHM
  FROM AlgorithmInformation-2009 -- RFC 5912 [X509ASN1]
  { iso(1) identified-organization(3) dod(6) internet(1)
    security(5) mechanisms(5) pkix(7) id-mod(0)
    id-mod-algorithmInformation-02(58) }
```

```
  SubjectPublicKeyInfo
  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) }
```

```
  OneAsymmetricKey
  FROM AsymmetricKeyPackageModuleV1
  { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1)
    pkcs-9(9) smime(16) modules(0)
    id-mod-asymmetricKeyPkgV1(50) } ;
```

```
--
```

```
-- Object Identifiers
--

id-alg-composite OBJECT IDENTIFIER ::= { TBD }

--
-- Public Key
--

pk-Composite PUBLIC-KEY ::= {
    IDENTIFIER id-alg-composite
    KEY CompositePublicKey
    PARAMS ARE absent
    PRIVATE-KEY CompositePrivateKey
}

CompositePublicKey ::= SEQUENCE SIZE (2..MAX) OF SubjectPublicKeyInfo

CompositePrivateKey ::= SEQUENCE SIZE (2..MAX) OF OneAsymmetricKey

END

<CODE ENDS>
```

6.2. Intellectual Property Considerations

The following IPR Disclosure relates to this draft:

<https://datatracker.ietf.org/ipr/3588/>

7. Contributors and Acknowledgements

This document incorporates contributions and comments from a large group of experts. The Editors would especially like to acknowledge the expertise and tireless dedication of the following people, who attended many long meetings and generated millions of bytes of electronic mail and VOIP traffic over the past year in pursuit of this document:

John Gray (Entrust), Serge Mister (Entrust), Scott Fluhrer (Cisco Systems), Panos Kampanakis (Cisco Systems), Daniel Van Geest (ISARA), Tim Hollebeek (Digicert), and Francois Rousseau.

We are grateful to all, including any contributors who may have been inadvertently omitted from this list.

This document borrows text from similar documents, including those referenced below. Thanks go to the authors of those documents.
"Copying always makes things easier and less error prone" -
[RFC8411].

7.1. Making contributions

Additional contributions to this draft are welcome. Please see the working copy of this draft at, as well as open issues at:

<https://github.com/EntrustCorporation/draft-ounsworth-pq-composite-keys>

8. Normative References

- [RFC1421] Linn, J., "Privacy Enhancement for Internet Electronic Mail: Part I: Message Encryption and Authentication Procedures", RFC 1421, DOI 10.17487/RFC1421, February 1993, <<https://www.rfc-editor.org/info/rfc1421>>.
- [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>>.
- [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>>.
- [RFC5652] Housley, R., "Cryptographic Message Syntax (CMS)", STD 70, RFC 5652, DOI 10.17487/RFC5652, September 2009, <<https://www.rfc-editor.org/info/rfc5652>>.

- [RFC5914] Housley, R., Ashmore, S., and C. Wallace, "Trust Anchor Format", RFC 5914, DOI 10.17487/RFC5914, June 2010, <<https://www.rfc-editor.org/info/rfc5914>>.
- [RFC5958] Turner, S., "Asymmetric Key Packages", RFC 5958, DOI 10.17487/RFC5958, August 2010, <<https://www.rfc-editor.org/info/rfc5958>>.
- [RFC7468] Josefsson, S. and S. Leonard, "Textual Encodings of PKIX, PKCS, and CMS Structures", RFC 7468, DOI 10.17487/RFC7468, April 2015, <<https://www.rfc-editor.org/info/rfc7468>>.
- [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>>.
- [RFC8411] Schaad, J. and R. Andrews, "IANA Registration for the Cryptographic Algorithm Object Identifier Range", RFC 8411, DOI 10.17487/RFC8411, August 2018, <<https://www.rfc-editor.org/info/rfc8411>>.
- [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 8825-1:2015, November 2015.

Authors' Addresses

Mike Ounsworth
Entrust Limited
2500 Solandt Road -- Suite 100
Ottawa, Ontario K2K 3G5
Canada

Email: mike.ounsworth@entrust.com

Massimiliano Pala
CableLabs

Email: director@openca.org

LAMPS
Internet-Draft
Intended status: Standards Track
Expires: 12 August 2022

M. Ounsworth
Entrust
M. Pala
CableLabs
8 February 2022

Composite Signatures For Use In Internet PKI
draft-ounsworth-pq-composite-sigs-06

Abstract

With the widespread adoption of post-quantum cryptography will come the need for an entity to possess multiple public keys on different cryptographic algorithms. Since the trustworthiness of individual post-quantum algorithms is at question, a multi-key cryptographic operation will need to be performed in such a way that breaking it requires breaking each of the component algorithms individually. This requires defining new structures for holding composite signature data.

This document defines the structures `CompositeSignatureValue`, and `CompositeParams`, which are sequences of the respective structure for each component algorithm. This document also defines processes for generating and verifying composite signatures. This document makes no assumptions about what the component algorithms are, provided that their algorithm identifiers and signature generation and verification processes are defined.

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 12 August 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. Terminology | 3 |
| 2. Composite Identifiers and Structures | 4 |
| 2.1. Algorithm Identifier | 5 |
| 2.2. Composite Keys | 5 |
| 2.2.1. Key Usage Bits | 5 |
| 2.3. Composite Signature | 6 |
| 2.4. Encoding Rules | 6 |
| 3. Composite Signature Processes | 7 |
| 3.1. Composite Signature Generation Process | 7 |
| 3.2. Composite-OR Signature Generation Process | 8 |
| 3.3. Composite Signature Verification Process | 9 |
| 3.4. Composite-OR Signature Verification | 11 |
| 3.4.1. Composite-OR Legacy Mode | 11 |
| 4. In Practice | 12 |
| 4.1. Cryptographic protocols | 13 |
| 5. IANA Considerations | 14 |
| 6. Security Considerations | 14 |
| 6.1. Policy for Deprecated and Acceptable Algorithms | 14 |
| 7. Appendices | 14 |
| 7.1. ASN.1 Module | 14 |
| 7.2. Intellectual Property Considerations | 16 |
| 8. Contributors and Acknowledgements | 16 |
| 8.1. Making contributions | 17 |
| 9. Normative References | 17 |
| Authors' Addresses | 18 |

1. Introduction

During the transition to post-quantum cryptography, there will be uncertainty as to the strength of cryptographic algorithms; we will no longer fully trust traditional cryptography such as RSA, Diffie-Hellman, DSA and their elliptic curve variants, but we will also not fully trust their post-quantum replacements until they have had sufficient scrutiny. Unlike previous cryptographic algorithm migrations, the choice of when to migrate and which algorithms to migrate to, is not so clear. Even after the migration period, it may be advantageous for an entity's cryptographic identity to be composed of multiple public-key algorithms.

The deployment of composite signatures using post-quantum algorithms will face two challenges

- * Algorithm strength uncertainty: During the transition period, some post-quantum signature and encryption algorithms will not be fully trusted, while also the trust in legacy public key algorithms will start to erode. A relying party may learn some time after deployment that a public key algorithm has become untrustworthy, but in the interim, they may not know which algorithm an adversary has compromised.
- * Backwards compatibility: During the transition period, post-quantum algorithms will not be supported by all clients.

This document provides a mechanism to address algorithm strength uncertainty by building on `~~ reference draft-ounsworth-pq-composite-pubkeys ~~` by providing formats for encoding multiple signature values into existing public signature fields, as well as the process for validating a composite signature. Backwards compatibility is addressed via the Composite-OR mechanism described herein.

This document is intended for general applicability anywhere that digital signatures are used within PKIX and CMS structures.

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.

The following terms are used in this document:

ALGORITHM: An information object class for identifying the type of cryptographic operation to be performed. This document is primarily concerned with algorithms for producing digital signatures.

BER: Basic Encoding Rules (BER) as defined in [X.690].

COMPONENT ALGORITHM: A single basic algorithm which is contained within a composite algorithm.

COMPOSITE ALGORITHM: An algorithm which is a sequence of two or more component algorithms, as defined in Section 2.

DER: Distinguished Encoding Rules as defined in [X.690].

LEGACY: For the purposes of this document, a legacy key or signature is a non-composite key or signature.

PUBLIC / PRIVATE KEY: The public and private portion of an asymmetric cryptographic key, making no assumptions about which algorithm.

SIGNATURE: A digital cryptographic signature, making no assumptions about which algorithm.

2. Composite Identifiers and Structures

In order for signatures to be composed of multiple algorithms, we define encodings consisting of a sequence of signature primitives (aka "component algorithms") such that these structures can be used as a drop-in replacement for existing signature fields such as those found in PKCS#10 [RFC2986], CMP [RFC4210], X.509 [RFC5280], CMS [RFC5652].

This section defines the following structures:

- * The id-alg-composite is an AlgorithmIdentifier identifying a composite signature object.

The sa-CompositeSignature AlgorithmIdentifier and the corresponding CompositeParams identify the algorithm(s) used in a composite signature.

- * The CompositeSignatureValue, carries a sequence of signatures that are generated by a CompositePrivateKey, and can be verified with the corresponding CompositePublicKey.

EDNOTE 2: the choice to define composite algorithm parameters as a sequence inside the existing fields avoids the exponential proliferation of OIDs that are needed for each combination of

signature algorithms in other schemes for achieving multi-key certificates. This scheme also naturally extends from 2-keypair to n-keypair keys and certificates.

EDNOTE 2a: We have heard community feedback that the ASN.1 structures presented here are too flexible in that allow arbitrary combinations of an arbitrary number of signature algorithms. The feedback is that this is too much of a "footgun" for implementors and sysadmins. We are working on an alternative formulation using ASN.1 information object classes that allow for compiling explicit pairs of algorithmIDs. We would love community feedback on which approach is preferred. See slide 30 of this presentation: <https://datatracker.ietf.org/meeting/interim-2021-lamps-01/materials/slides-interim-2021-lamps-01-sessa-position-presentation-by-mike-ounsworth-00.pdf>

2.1. Algorithm Identifier

The following object identifier is used for identifying a composite signature. Additional encoding information is provided below for each of these objects.

```
id-alg-composite OBJECT IDENTIFIER ::= {  
    iso(1) identified-organization(3) dod(6) internet(1) private(4)  
    enterprise(1) OpenCA(18227) Algorithms(2) id-alg-composite(1) }
```

EDNOTE 3: this is a temporary OID for the purposes of prototyping. We are requesting IANA to assign a permanent OID, see Section 5.

2.2. Composite Keys

A Composite signature MUST be associated with a Composite public key as defined in `reference draft-ounsworth-pq-composite-pubkey`.

2.2.1. Key Usage Bits

For protocols such as X.509 [RFC5280] that specify key usage along with the public key, then the composite public key associated with a composite signature MUST have a signing-type key usage.

If the keyUsage extension is present in a Certification Authority (CA) certificate that indicates id-composite-key, then any combination of the following values MAY be present:

```
digitalSignature;  
nonRepudiation;  
keyCertSign; and  
cRLSign.
```

If the keyUsage extension is present in an End Entity (EE) certificate that indicates id-composite-key, then any combination of the following values MAY be present:

digitalSignature; and
nonRepudiation;

2.3. Composite Signature

The ASN.1 algorithm object for a composite signature is:

```
sa-CompositeSignature SIGNATURE-ALGORITHM ::= {  
  IDENTIFIER id-alg-composite  
  VALUE CompositeSignatureValue  
  PARAMS TYPE CompositeParams ARE required  
  PUBLIC-KEYS { pk-Composite }  
  SMIME-CAPS { IDENTIFIED BY id-alg-composite } }  
}
```

The following algorithm parameters MUST be included:

CompositeParams ::= SEQUENCE SIZE (2..MAX) OF AlgorithmIdentifier

The signature's CompositeParams sequence MUST contain the same component algorithms listed in the same order as in the associated CompositePrivateKey and CompositePublicKey.

The output of the composite signature algorithm is the DER encoding of the following structure:

CompositeSignatureValue ::= SEQUENCE SIZE (2..MAX) OF BIT STRING

Where each BIT STRING within the SEQUENCE is a signature value produced by one of the component keys. It MUST contain one signature value produced by each component algorithm, and in the same order as in the associated CompositeParams object.

The choice of SEQUENCE OF BIT STRING, rather than for example a single BIT STRING containing the concatenated signature values, is to gracefully handle variable-length signature values by taking advantage of ASN.1's built-in length fields.

2.4. Encoding Rules

Many protocol specifications will require that composite signature data structures be represented by an octet string or bit string.

When an octet string is required, the DER encoding of the composite data structure SHALL be used directly.

When a bit string is required, the octets of the DER encoded composite data structure SHALL be used as the bits of the bit string, with the most significant bit of the first octet becoming the first bit, and so on, ending with the least significant bit of the last octet becoming the last bit of the bit string.

In the interests of simplicity and avoiding compatibility issues, implementations that parse these structures MAY accept both BER and DER.

3. Composite Signature Processes

This section specifies the processes for generating and verifying composite signatures.

This process addresses algorithm strength uncertainty by providing the verifier with parallel signatures from all the component signature algorithms; thus breaking the composite signature would require breaking all of the component signatures.

3.1. Composite Signature Generation Process

Generation of a composite signature involves applying each component algorithm's signature process to the input message according to its specification, and then placing each component signature value into the CompositeSignatureValue structure defined in Section 2.3.

The following process is used to generate composite signature values.

Input:

K1, K2, ..., Kn Private keys for the n component signature algorithms, a CompositePrivateKey
M Message to be signed, an octet string

Output:

S The signatures, a CompositeSignatureValue

Signature Generation Process:

1. Generate the n component signatures independently, according to their algorithm specifications.

 for i := 1 to n
 Si := Sign(Ki, M)
2. Encode each component signature S1, S2, ..., Sn into a BIT STRING according to its algorithm specification.

 S ::= Sequence { S1, S2, ..., Sn }
3. Output S

Since recursive composite public keys are disallowed in ~~ Reference draft-ounsworth-pq-composite-pubkeys sec-composite-pub-keys ~~, no component signature may itself be a composite; ie the signature generation process MUST fail if one of the private keys K1, K2, ..., Kn is a composite with the OID id-alg-composite.

A composite signature MUST produce and include in the output a signature value for every component key in the corresponding CompositePrivateKey. For this mode, please see Composite-OR in section Section 3.2.

3.2. Composite-OR Signature Generation Process

EDNOTE: This section was written with the intention of keeping the primary Composite OID reserved for the simple and strict mode; if you want to do either a simple OR, or a custom policy then we have given a different OID. We are still debating whether this is useful to specify at issuing time, or whether this is adding needless complexity to the draft.

If the algorithm ID of the public key associated with this signature is id-composite-or-key then the signer MAY use only a subset of the component keys and therefore produce fewer signatures than the number of component keys.

Composite-OR signature generation uses the same structures and algorithms as Composite, with the difference that the signature generation process may emit a null instead of a signature value in step 1 for one or more component algorithms. A Composite-OR signature MUST NOT be entirely null; it must contain at least one valid signature.

The design intent of this mode is to support migration scenarios where an end entity has been issued keys on algorithms that either itself or the peer with which it is communicating do not (yet) support. This design allows for both the mode where the signer omits signatures that it knows its peer cannot process in order to save bandwidth and performance, and the mode where it includes all component signatures and allows the verifier to choose how many to verify. The latter is RECOMMENDED for signatures that need both short-term backwards compatibility as well as long-term security.

EDNOTE: Do we want to allow a Composite-OR with only a single signature to produce non-composite signatureAlgorithm and signatureValue as per [RFC5280]? Advantages: bandwidth savings of an extra OID and some sequences with one element. Disadvantages: ambiguous whether a signature is traditional or composite until you look at the corresponding public key.

3.3. Composite Signature Verification Process

Verification of a composite signature involves applying each component algorithm's verification process according to its specification.

In the absence of an application profile specifying otherwise, compliant applications MUST output "Valid signature" (true) if and only if all component signatures were successfully validated, and "Invalid signature" (false) otherwise.

The following process is used to perform this verification.

Input:

P Signer's composite public key
M Message whose signature is to be verified, an octet string
S Composite Signature to be verified
A Composite Algorithm identifier

Output:

Validity "Valid signature" (true) if the composite signature
 is valid, "Invalid signature" (false) otherwise.

Signature Verification Procedure::

1. Parse P, S, A into the component public keys, signatures,
and algorithm identifiers

P1, P2, ..., Pn := Desequence(P)
S1, S2, ..., Sn := Desequence(S)
A1, A2, ..., An := Desequence(A)

If Error during Desequencing, or the three sequences have
different numbers of elements, or any of the public keys P1, P2, ..., Pn or
algorithm identifiers A1, A2, ..., An are composite with the OID
id-alg-composite then output "Invalid signature" and stop.

2. Check each component signature individually, according to its
algorithm specification.
If any fail, then the entire signature validation fails.

for i := 1 to n
 if not verify(Pi, M, Si), then
 output "Invalid signature"

if all succeeded, then
 output "Valid signature"

Since recursive composite public keys are disallowed in ~~ Reference
draft-ounsworth-pq-composite-keys sec-composite-pub-keys ~~, no
component signature may be composite; ie the signature verification
procedure MUST fail if any of the public keys P1, P2, ..., Pn or
algorithm identifiers A1, A2, ..., An are composite with the OID id-
alg-composite.

3.4. Composite-OR Signature Verification

EDNOTE: This section was written with the intention of keeping the primary Composite OID reserved for the simple and strict mode; if you want to do either a simple OR, or a custom policy then we have given a different OID. We are still debating whether this is useful to specify at issuing time, or whether this is adding needless complexity to the draft.

When the public key associated with the signature being verified has algorithm id-composite-or-key, then an alternate verification processes MAY be used, at the discretion of the implementor. In this section we provide some examples of alternate verification processes.

If the signature is a traditional (non-composite) algorithm and value or a composite signature with a single component, then it MAY be considered valid if it verifies under one of the component keys.

If the signature is composite, then the implementor MAY implement policy for which combinations are acceptable.

EDNOTE: Does this mean Composite-OR end entity certificates need to be issued by a PKI that is marked as Composite-OR all the way to the top so that verifiers that do not support all the algorithms don't fail? Need to think more about the security implications of allowing a Composite-or in an end entity cert implicitly turning all Composite algIDs into Composite-or algIDs in its cert chain.

EDNOTE: Do we need to specify the semantics of verifying an "n of m" subset signature? I suspect that specifying this in general will be a rat's nest of edge cases, so I propose to "leave this to the implementor".

3.4.1. Composite-OR Legacy Mode

The Composite-OR Legacy Mode is provided to facilitate migration by allowing existing PKI entities (including root CAs, intermediate CAs, and end entities) to have their existing keys re-certified inside a Composite-OR structure along with Post-Quantum keys, and for signatures made by that key prior to the migration to remain valid. Note that Composite-OR Legacy Mode is only provided for signature verification, and not for signature generation; legacy signatures SHOULD NOT be produced from a Composite key.

EDNOTE: to further solidify this, we could add a clause that Legacy Mode signatures are to fail if the signature was produced after notBefore date of the Composite-OR certificate?

In Composite-OR Legacy Mode, a legacy signature algorithm and legacy signature value MAY be validated against a Composite-OR public key. The legacy signature algorithm is to be interpreted by the verifier as a sa-CompositeSignature with CompositeParams in the following way:

```
CompositeParams {legacyAlgorithmIdentifier, null, ..., null}
```

with the correct number of nulls to match the Composite-OR public key that the signature is being verified against. For the purposes of a signature validation under Composite-OR Legacy Mode, a null AlgorithmIdentifier is considered to be a match for the corresponding algorithm in the Composite-OR public key.

The legacy signature value is to be interpreted by the verifier as a sa-CompositeSignature with CompositeParams in the following way:

```
CompositeSignatureValue {legacySignatureValue, null, ..., null}
```

with the correct number of nulls to match the Composite-OR public key that the signature is being verified against. The verification algorithm in section Section 3.4 applies.

Security consideration: when implementing Composite-OR Legacy Mode, it is important to catch the edge case of {null, null, ..., null} for both AlgorithmIdentifier and SignatureValue and return Invalid Signature.

It is RECOMMENDED that Composite-OR Legacy Mode be implemented as an optional mode in the verifier that can be enabled or disabled by runtime configuration or policy.

EDNOTE: the signing public key is often identified in the signed document by issuer+serialNumber or by an SKI containing a hash of the public key value. Might need X.509 extensions identifying the SKI of the legacy cert it's replacing?

4. In Practice

This section addresses practical issues of how this draft affects other protocols and standards.

~~~ BEGIN EDNOTE 10~~~

EDNOTE 10: Possible topics to address:

- \* The size of these certs and cert chains.

- \* In particular, implications for (large) composite keys / signatures / certs on the handshake stages of TLS and IKEv2.
- \* If a cert in the chain is a composite cert then does the whole chain need to be of composite Certs?
- \* We could also explain that the root CA cert does not have to be of the same algorithms. The root cert SHOULD NOT be transferred in the authentication exchange to save transport overhead and thus it can be different than the intermediate and leaf certs.
- \* We could talk about overhead (size and processing).
- \* We could also discuss backwards compatibility.
- \* We could include a subsection about implementation considerations.

~~~ END EDNOTE 10~~~

4.1. Cryptographic protocols

This section talks about how protocols like (D)TLS and IKEv2 are affected by this specifications. It will not attempt to solve all these problems, but it will explain the rationale, how things will work and what open problems need to be solved. Obvious issues that need to be discussed.

- * How does the protocol declare support for composite signatures? TLS has hooks for declaring support for specific signature algorithms, however it would need to be extended, because the client would need to declare support for both the composite infrastructure, as well as for the various component signature algorithms.
- * How does the protocol use the multiple keys. The obvious way would be to have the server sign using its composite public key; is this sufficient.
- * Overhead; including certificate size, signature processing time, and size of the signature.
- * How to deal with crypto protocols that use public key encryption algorithms; this document only lists how to work with signature algorithms. Encoding composite public keys is straightforward; encoding composite ciphertexts is less so - we decided to put that off to another draft.

5. IANA Considerations

The ASN.1 module OID is TBD. The id-alg-composite OID is to be assigned by IANA. The authors suggest that IANA assign an OID on the id-pkix arc:

```
id-alg-composite OBJECT IDENTIFIER ::= {  
    iso(1) identified-organization(3) dod(6) internet(1) security(5)  
    mechanisms(5) pkix(7) algorithms(6) composite(??) }
```

6. Security Considerations

6.1. Policy for Deprecated and Acceptable Algorithms

Traditionally, a public key, certificate, or signature contains a single cryptographic algorithm. If and when an algorithm becomes deprecated (for example, RSA-512, or SHA1), it is obvious that structures using that algorithm are implicitly revoked.

In the composite model this is less obvious since a single public key, certificate, or signature may contain a mixture of deprecated and non-deprecated algorithms. Moreover, implementers may decide that certain cryptographic algorithms have complementary security properties and are acceptable in combination even though neither algorithm is acceptable by itself.

Specifying a modified verification algorithm to handle these situations is beyond the scope of this draft, but could be desirable as the subject of an application profile document, or to be up to the discretion of implementers.

2. Check policy to see whether A1, A2, ..., An constitutes a valid combination of algorithms.

```
if not checkPolicy(A1, A2, ..., An), then  
    output "Invalid signature"
```

While intentionally not specified in this document, implementors should put careful thought into implementing a meaningful policy mechanism within the context of their signature verification engines, for example only algorithms that provide similar security levels should be combined together.

7. Appendices

7.1. ASN.1 Module

<CODE STARTS>

Composite-Signatures-2019
{ TBD }

DEFINITIONS IMPLICIT TAGS ::= BEGIN

EXPORTS ALL;

IMPORTS

PUBLIC-KEY, SIGNATURE-ALGORITHM
FROM AlgorithmInformation-2009 -- RFC 5912 [X509ASN1]
{ iso(1) identified-organization(3) dod(6) internet(1)
security(5) mechanisms(5) pkix(7) id-mod(0)
id-mod-algorithmInformation-02(58) }

SubjectPublicKeyInfo
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) }

OneAsymmetricKey
FROM AsymmetricKeyPackageModuleV1
{ iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1)
pkcs-9(9) smime(16) modules(0)
id-mod-asymmetricKeyPkgV1(50) } ;

--
-- Object Identifiers
--

id-alg-composite OBJECT IDENTIFIER ::= { TBD }

--
-- Public Key
--

pk-Composite PUBLIC-KEY ::= {
IDENTIFIER id-alg-composite
KEY CompositePublicKey
PARAMS ARE absent
PRIVATE-KEY CompositePrivateKey
}

CompositePublicKey ::= SEQUENCE SIZE (2..MAX) OF SubjectPublicKeyInfo

CompositePrivateKey ::= SEQUENCE SIZE (2..MAX) OF OneAsymmetricKey

```
--  
-- Signature Algorithm  
--  
  
sa-CompositeSignature SIGNATURE-ALGORITHM ::= {  
    IDENTIFIER id-alg-composite  
    VALUE CompositeSignatureValue  
    PARAMS TYPE CompositeParams ARE required  
    PUBLIC-KEYS { pk-Composite }  
    SMIME-CAPS { IDENTIFIED BY id-alg-composite } }  
  
CompositeParams ::= SEQUENCE SIZE (2..MAX) OF AlgorithmIdentifier  
  
CompositeSignatureValue ::= SEQUENCE SIZE (2..MAX) OF BIT STRING  
  
END  
  
<CODE ENDS>
```

7.2. Intellectual Property Considerations

The following IPR Disclosure relates to this draft:

<https://datatracker.ietf.org/ipr/3588/>

8. Contributors and Acknowledgements

This document incorporates contributions and comments from a large group of experts. The Editors would especially like to acknowledge the expertise and tireless dedication of the following people, who attended many long meetings and generated millions of bytes of electronic mail and VOIP traffic over the past year in pursuit of this document:

John Gray (Entrust), Serge Mister (Entrust), Scott Fluhrer (Cisco Systems), Panos Kampanakis (Cisco Systems), Daniel Van Geest (ISARA), Tim Hollebeek (Digicert), and Francois Rousseau.

We are grateful to all, including any contributors who may have been inadvertently omitted from this list.

This document borrows text from similar documents, including those referenced below. Thanks go to the authors of those documents.

"Copying always makes things easier and less error prone" -
[RFC8411].

8.1. Making contributions

Additional contributions to this draft are welcome. Please see the working copy of this draft at, as well as open issues at:

<https://github.com/EntrustCorporation/draft-ounsworth-composite-sigs>

9. 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>>.
- [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>>.
- [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>>.
- [RFC5652] Housley, R., "Cryptographic Message Syntax (CMS)", STD 70, RFC 5652, DOI 10.17487/RFC5652, September 2009, <<https://www.rfc-editor.org/info/rfc5652>>.
- [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>>.
- [RFC8411] Schaad, J. and R. Andrews, "IANA Registration for the Cryptographic Algorithm Object Identifier Range", RFC 8411, DOI 10.17487/RFC8411, August 2018, <<https://www.rfc-editor.org/info/rfc8411>>.
- [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 8825-1:2015, November 2015.

Authors' Addresses

Mike Ounsworth
Entrust Limited
2500 Solandt Road -- Suite 100
Ottawa, Ontario K2K 3G5
Canada

Email: mike.ounsworth@entrust.com

Massimiliano Pala
CableLabs

Email: director@openca.org

LAMPS
Internet-Draft
Intended status: Standards Track
Expires: 16 August 2022

M. Ounsworth
S. Mister
J. Gray
Entrust
12 February 2022

Explicit Pairwise Composite Keys For Use In Internet PKI
draft-ounsworth-pq-explicit-composite-keys-01

Abstract

With the widespread adoption of post-quantum cryptography will come the need for an entity to possess multiple public keys on different cryptographic algorithms. Since the trustworthiness of individual post-quantum algorithms is at question, a multi-key cryptographic operation will need to be performed in such a way that breaking it requires breaking each of the component algorithms individually. This requires defining new structures for holding composite public keys and composite signature data. This draft defines a structure generic enough to be useful beyond the post-quantum transition for any situation where a widely-supported but untrusted algorithm is being migrated to newer cryptography.

This document defines structures for binding an explicit pair of cryptographic algorithms together into a single object identifier, and it provides ASN.1 structures for encoding these pairwise composite public keys, private keys in wire protocols, as well as using them in conjunction with composite signatures, encryption and key transport mechanisms.

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 16 August 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. Terminology | 3 |
| 2. Composite Structures | 4 |
| 2.1. Composite Keys | 5 |
| 2.2. Composite Private Key | 5 |
| 2.3. Composite Signature | 6 |
| 2.3.1. Explicit Signature Params | 6 |
| 2.3.2. Explicit Composite Signature Algorithm | 7 |
| 2.3.3. Explicit Encryption and Key Exchange Params | 7 |
| 2.4. Encoding Rules | 7 |
| 3. In Practice | 7 |
| 3.1. PEM Storage of Composite Private Keys | 8 |
| 3.2. Asymmetric Key Packages (CMS) | 8 |
| 3.3. Cryptographic protocols | 9 |
| 4. IANA Considerations | 9 |
| 5. Security Considerations | 9 |
| 5.1. Policy for Deprecated and Acceptable Algorithms | 10 |
| 5.2. Protection of Private Keys | 10 |
| 5.3. Checking for Compromised Key Reuse | 11 |
| 6. Appendices | 11 |
| 6.1. ASN.1 Module | 11 |
| 6.2. Examples of defining explicit pairs | 12 |
| 6.3. Intellectual Property Considerations | 13 |
| 7. Contributors and Acknowledgements | 13 |
| 8. References | 13 |
| 8.1. Normative References | 13 |
| 8.2. Informative References | 15 |
| Authors' Addresses | 15 |

1. Introduction

During the transition to post-quantum cryptography, there will be uncertainty as to the strength of cryptographic algorithms; we will no longer fully trust traditional cryptography such as RSA, Diffie-Hellman, DSA and their elliptic curve variants, but we will also not fully trust their post-quantum replacements until they have had sufficient scrutiny. Unlike previous cryptographic algorithm migrations, the choice of when to migrate and which algorithms to migrate to, is not so clear. Even after the migration period, it may be advantageous for an entity's cryptographic identity to be composed of multiple public-key algorithms.

The deployment of composite public keys and composite signatures using post-quantum algorithms will face two challenges

- * Algorithm strength uncertainty: During the transition period, some post-quantum signature and encryption algorithms will not be fully trusted, while also the trust in legacy public key algorithms will start to erode. A relying party may learn some time after deployment that a public key algorithm has become untrustworthy, but in the interim, they may not know which algorithm an adversary has compromised.
- * Backwards compatibility: During the transition period, post-quantum algorithms will not be supported by all clients.

This document provides a mechanism to address algorithm strength uncertainty by providing formats for encoding multiple public keys and private keys into existing fields.

This document provides structures to encode explicit composite algorithm identifiers and parameters for use with composite signature, encryption, and key transport mechanisms defined in ~
TODO cite corresponding drafts properly ~.

This document is intended for general applicability anywhere that public key or private key structures are used within PKIX protocols.

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.

The following terms are used in this document:

ALGORITHM: An information object class for identifying the type of cryptographic operation to be performed. This document is primarily concerned with algorithms for producing digital signatures, though the public key structure could just as easily hold encryption keys.

BER: Basic Encoding Rules (BER) as defined in [X.690].

COMPONENT ALGORITHM: A single basic algorithm which is contained within a composite algorithm.

COMPOSITE ALGORITHM: An algorithm which is a sequence of one or more component algorithms, as defined in Section 2.

DER: Distinguished Encoding Rules as defined in [X.690].

EXPLICIT COMPOSITE: Composite structures where the AlgorithmIdentifier OID explicitly defines the component algorithms. This case allows simplification and compression of the data structures.

GENERIC COMPOSITE: Composite structures that are agnostic to the choice of Algorithms that they carry.

PUBLIC / PRIVATE KEY: The public and private portion of an asymmetric cryptographic key, making no assumptions about which algorithm.

PRIMITIVE PUBLIC KEY / SIGNATURE: A public key or signature object of a non-composite algorithm type.

SIGNATURE: A digital cryptographic signature, making no assumptions about which algorithm.

2. Composite Structures

In order for public keys and signatures to be composed of pairs of algorithms, we define encodings consisting of a sequence of public key and signature primitives (aka "component algorithms") such that these structures can be used as a drop-in replacement for existing public key or signature fields such as those found in PKCS#10 [RFC2986], CMP [RFC4210], X.509 [RFC5280], CMS [RFC5652].

This section defines the following structures:

~~ TODO ~~

2.1. Composite Keys

A composite key is a single key object that performs an atomic signature or verification operation, using its encapsulated pair of component keys.

Explicit pairs can easily be defined by simply providing an OBJECT IDENTIFIER and two existing PUBLIC-KEY types to the pk-explicitComposite object class, and assigning an OID to the resulting structure. See examples of defining explicit pairs in Section 6.2.

```
-- TODO - CERT-KEY-USAGE should contain the intersection of the usages from first
PublicKey, secondPublicKey and the four listed below
-- pk-explicitComposite - Composite public key information object
```

```
pk-explicitComposite{OBJECT IDENTIFIER:id, PUBLIC-KEY:firstPublicKey, FirstPublic
KeyType, PUBLIC-KEY:secondPublicKey, SecondPublicKeyType} PUBLIC-KEY ::= {
    IDENTIFIER id
    KEY ExplicitCompositePublicKey{firstPublicKey, FirstPublicKeyType, secondPubl
icKey, SecondPublicKeyType}
    PARAMS ARE absent
    CERT-KEY-USAGE {digitalSignature, nonRepudiation, keyCertSign, cRLSign}
}
```

The following ASN.1 object class then automatically generates the public key structure from the types defined in pk-explicitComposite.

```
-- ExplicitCompositePublicKey - The data structure for a composite public key
-- sec-alg-identifier and SecondPublicKeyType are needed because PUBLIC-KEY conta
ins
-- a set of public key types, not a single type.
-- TODO The parameters should be optional only if they are marked optional in the
PUBLIC-KEY
```

```
ExplicitCompositePublicKey{PUBLIC-KEY:firstPublicKey, FirstPublicKeyType, PUBLIC-
KEY:secondPublicKey, SecondPublicKeyType} ::= SEQUENCE {
    firstPublicKey SEQUENCE {
        params firstPublicKey.&Params OPTIONAL,
        publicKey FirstPublicKeyType
    },
    secondPublicKey SEQUENCE {
        params secondPublicKey.&Params OPTIONAL,
        publicKey SecondPublicKeyType
    }
}
```

2.2. Composite Private Key

EDNOTE: THIS IS WRONG. (copied from generic draft) we need to do some work to come up with a private key structure.

The composite private key data is represented by the following structure:

CompositePrivateKey ::= SEQUENCE SIZE (2..MAX) OF OneAsymmetricKey

Each element is a OneAsymmetricKey [RFC5958] object for a component private key.

The corresponding AlgorithmIdentifier for a composite private key MUST use the id-alg-composite object identifier, and the parameters field MUST be absent.

A CompositePrivateKey MUST contain at least one component private key, and they MUST be in the same order as in the corresponding CompositePublicKey.

2.3. Composite Signature

The structure pk-explicitComposite contains all the necessary information in order for the ASN.1 compiler to generate composite signature structures that are explicitly bound to the specified pair of algorithms.

EDNOTE: Is this helping, or adding complexity for no reason? In theory, explicit composite public keys can be used with generic composite signature and encryption structures (ie the SEQUENCE OF model).

2.3.1. Explicit Signature Params

The following ASN.1 object class then automatically generates the signature params structure from the types defined in pk-explicitComposite.

```
-- ExplicitSignatureParams - The data structure for composite signature parameter
s
-- TODO firstParams and secondParams should be optional only if they are marked o
ptional
-- in SIGNATURE-ALGORITHM
```

```
ExplicitSignatureParams{SIGNATURE-ALGORITHM:firstAlg, SIGNATURE-ALGORITHM:secondA
lg} ::= SEQUENCE {
    firstParams firstAlg.&Params OPTIONAL,
    secondParams secondAlg.&Params OPTIONAL
}
```

EDNOTE: we need some help from the community on the ASN.1 here: "OPTIONAL" is not really the right semantics here; we really mean that they params here should be present or absent when the corresponding params are present or absent in ExplicitCompositePublicKey, which ought to be enforcable by the ASN.1 compiler, but we can't figure out the syntax for declaring that.

2.3.2. Explicit Composite Signature Algorithm

The following ASN.1 object class then automatically generates the signature algorithm structure from the types defined in pk-explicitComposite.

```
-- TODO - Would it be possible to make these definitions compatible with n signature algorithms instead of 2? Is it desired?
-- sa-explicitCompositeSignatureAlgorithm - Composite signature algorithm information object
```

```
sa-explicitCompositeSignatureAlgorithm(OBJECT IDENTIFIER:algId, SIGNATURE-ALGORITHM:firstAlg, PUBLIC-KEY:firstPublicKey, FirstPublicKeyType, SIGNATURE-ALGORITHM:secondAlg, PUBLIC-KEY:secondPublicKey, SecondPublicKeyType) SIGNATURE-ALGORITHM ::= {
    IDENTIFIER algId
    VALUE ExplicitCompositeSignatureValue{firstAlg.&Value, secondAlg.&Value}
    PARAMS TYPE ExplicitSignatureParams{firstAlg, secondAlg} ARE required
    PUBLIC-KEYS { pk-explicitComposite{algId, firstPublicKey, FirstPublicKeyType, secondPublicKey, SecondPublicKeyType} }
    SMIME-CAPS { IDENTIFIED BY algId }
}
```

2.3.3. Explicit Encryption and Key Exchange Params

```
-- TODO -- Need analogous structures to the signature ones above.
```

2.4. Encoding Rules

Many protocol specifications will require that the composite public key, composite private key, and composite signature data structures be represented by an octet string.

When an octet string is required, the DER encoding of the composite data structure SHALL be used directly.

When a bit string is required, the octets of the DER encoded composite data structure SHALL be used as the bits of the bit string, with the most significant bit of the first octet becoming the first bit, and so on, ending with the least significant bit of the last octet becoming the last bit of the bit string.

In the interests of simplicity and avoiding compatibility issues, implementations that parse these structures MAY accept both BER and DER.

3. In Practice

This section addresses practical issues of how this draft affects other protocols and standards.

```
~~~ BEGIN EDNOTE 10~~~
```

EDNOTE 10: Possible topics to address:

- * The size of these certs and cert chains.
- * In particular, implications for (large) composite keys / signatures / certs on the handshake stages of TLS and IKEv2.
- * If a cert in the chain is a composite cert then does the whole chain need to be of composite Certs?
- * We could also explain that the root CA cert does not have to be of the same algorithms. The root cert SHOULD NOT be transferred in the authentication exchange to save transport overhead and thus it can be different than the intermediate and leaf certs.
- * We could talk about overhead (size and processing).
- * We could also discuss backwards compatibility.
- * We could include a subsection about implementation considerations.

~~~ END EDNOTE 10~~~

### 3.1. PEM Storage of Composite Private Keys

CompositePrivateKeys can be encoded to the PEM format by placing a CompositePrivateKey into the privateKey field of a PrivateKeyInfo or OneAsymmetricKey object, and then applying the PEM encoding rules as defined in [RFC7468] section 10 and 11 for plaintext and encrypted private keys, respectively.

### 3.2. Asymmetric Key Packages (CMS)

The Cryptographic Message Syntax (CMS), as defined in [RFC5652], can be used to digitally sign, digest, authenticate, or encrypt the asymmetric key format content type.

When encoding composite private keys, the privateKeyAlgorithm in the OneAsymmetricKey SHALL be set to id-alg-composite.

The parameters of the privateKeyAlgorithm SHALL be a sequence of AlgorithmIdentifier objects, each of which are encoded according to the rules defined for each of the different keys in the composite private key.

The value of the privateKey field in the OneAsymmetricKey SHALL be set to the DER encoding of the SEQUENCE of private key values that make up the composite key. The number and order of elements in the sequence SHALL be the same as identified in the sequence of parameters in the privateKeyAlgorithm.

The value of the `publicKey` (if present) SHALL be set to the DER encoding of the corresponding `CompositePublicKey`. If this field is present, the number and order of component keys MUST be the same as identified in the sequence of parameters in the `privateKeyAlgorithm`.

The value of the attributes is encoded as usual.

### 3.3. Cryptographic protocols

This section talks about how protocols like (D)TLS and IKEv2 are affected by this specifications. It will not attempt to solve all these problems, but it will explain the rationale, how things will work and what open problems need to be solved. Obvious issues that need to be discussed.

- \* How does the protocol declare support for composite signatures? TLS has hooks for declaring support for specific signature algorithms, however it would need to be extended, because the client would need to declare support for both the composite infrastructure, as well as for the various component signature algorithms.
- \* How does the protocol use the multiple keys. The obvious way would be to have the server sign using its composite public key; is this sufficient.
- \* Overhead; including certificate size, signature processing time, and size of the signature.
- \* How to deal with crypto protocols that use public key encryption algorithms; this document only lists how to work with signature algorithms. Encoding composite public keys is straightforward; encoding composite ciphertexts is less so - we decided to put that off to another draft.

### 4. IANA Considerations

This draft does not define any OIDs, however derivative drafts that define concrete algorithm pairs will. The authors suggest that IANA assign OIDs for explicit composite pairs on the `id-pkix` arc under a `composite()` arc.

```
id-alg-composite OBJECT IDENTIFIER ::= {  
    iso(1) identified-organization(3) dod(6) internet(1) security(5)  
    mechanisms(5) pkix(7) algorithms(6) composite(??) }
```

### 5. Security Considerations

### 5.1. Policy for Deprecated and Acceptable Algorithms

Traditionally, a public key, certificate, or signature contains a single cryptographic algorithm. If and when an algorithm becomes deprecated (for example, RSA-512, or SHA1), it is obvious that structures using that algorithm are implicitly revoked.

In the composite model this is less obvious since a single public key, certificate, or signature may contain a mixture of deprecated and non-deprecated algorithms. Moreover, implementers may decide that certain cryptographic algorithms have complementary security properties and are acceptable in combination even though neither algorithm is acceptable by itself.

Specifying a modified verification algorithm to handle these situations is beyond the scope of this draft, but could be desirable as the subject of an application profile document, or to be up to the discretion of implementers.

2. Check policy to see whether A1, A2, ..., An constitutes a valid combination of algorithms.

```
if not checkPolicy(A1, A2, ..., An), then
    output "Invalid signature"
```

While intentionally not specified in this document, implementors should put careful thought into implementing a meaningful policy mechanism within the context of their signature verification engines, for example only algorithms that provide similar security levels should be combined together.

### 5.2. Protection of Private Keys

Structures described in this document do not protect private keys in any way unless combined with a security protocol or encryption properties of the objects (if any) where the CompositePrivateKey is used (see next Section).

Protection of the private keys is vital to public key cryptography. The consequences of disclosure depend on the purpose of the private key. If a private key is used for signature, then the disclosure allows unauthorized signing. If a private key is used for key management, then disclosure allows unauthorized parties to access the managed keying material. The encryption algorithm used in the encryption process must be at least as 'strong' as the key it is protecting.

### 5.3. Checking for Compromised Key Reuse

CA implementations need to be careful when checking for compromised key reuse, for example as required by WebTrust regulations; when checking for compromised keys, you MUST unpack the CompositePublicKey structure and compare individual component keys. In other words, when marking a key as revoked for key compromise, the individual component keys should be marked, not the composite key as a whole.

## 6. Appendices

### 6.1. ASN.1 Module

<CODE STARTS>

```
Composite-Signatures-2019
{ TBD }
```

```
DEFINITIONS IMPLICIT TAGS ::= BEGIN
```

```
EXPORTS ALL;
```

```
IMPORTS
```

```
  PUBLIC-KEY, SIGNATURE-ALGORITHM
```

```
  FROM AlgorithmInformation-2009 -- RFC 5912 [X509ASN1]
    { iso(1) identified-organization(3) dod(6) internet(1)
      security(5) mechanisms(5) pkix(7) id-mod(0)
      id-mod-algorithmInformation-02(58) };
```

```
  SubjectPublicKeyInfo
```

```
  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) };
```

```
  OneAsymmetricKey
```

```
  FROM AsymmetricKeyPackageModuleV1
    { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1)
      pkcs-9(9) smime(16) modules(0)
      id-mod-asymmetricKeyPkgV1(50) } ;
```

```
--
```

```
-- Object Identifiers
```

```
--
```

```
id-alg-composite OBJECT IDENTIFIER ::= { TBD }
```

```
--
```

```
-- Public Key
--

pk-Composite PUBLIC-KEY ::= {
    IDENTIFIER id-alg-composite
    KEY CompositePublicKey
    PARAMS ARE absent
    CERT-KEY-USAGE
        { digitalSignature, nonRepudiation, keyCertSign, cRLSign }
    PRIVATE-KEY CompositePrivateKey
}

CompositePublicKey ::= SEQUENCE SIZE (2..MAX) OF SubjectPublicKeyInfo

CompositePrivateKey ::= SEQUENCE SIZE (2..MAX) OF OneAsymmetricKey

--
-- Signature Algorithm
--

sa-CompositeSignature SIGNATURE-ALGORITHM ::= {
    IDENTIFIER id-alg-composite
    VALUE CompositeSignatureValue
    PARAMS TYPE CompositeParams ARE required
    PUBLIC-KEYS { pk-Composite }
    SMIME-CAPS { IDENTIFIED BY id-alg-composite } }

CompositeParams ::= SEQUENCE SIZE (2..MAX) OF AlgorithmIdentifier

CompositeSignatureValue ::= SEQUENCE SIZE (2..MAX) OF BIT STRING

END

<CODE ENDS>
```

## 6.2. Examples of defining explicit pairs

To add support for a new pair of algorithms, all that is required is the following two constructs:

id-sa-entrust-sha256RSAandECDSA OBJECT IDENTIFIER ::= { 1 2 3 4 }

```
sa-entrust-sha256RSAandECDSA SIGNATURE-ALGORITHM ::= sa-explicitCompositeSignatureAlgorithm{
    id-sa-entrust-sha256RSAandECDSA,
    sa-sha256WithRSAEncryption,
    pk-rsa,
    RSAPublicKey,
    sa-ecdsaWithSHA256,
    pk-ec,
    ECPoint
}
```

TODO: run this through an ASN.1 compiler and list here what the final generated structures look like.

### 6.3. Intellectual Property Considerations

The following IPR Disclosure relates to this draft:

<https://datatracker.ietf.org/ipr/3588/>

## 7. Contributors and Acknowledgements

This document incorporates contributions and comments from a large group of experts. The Editors would especially like to acknowledge the expertise and tireless dedication of the following people, who attended many long meetings and generated millions of bytes of electronic mail and VOIP traffic over the past year in pursuit of this document:

John Gray (Entrust Datacard), Serge Mister (Entrust Datacard), Scott Fluhrer (Cisco Systems), Panos Kampanakis (Cisco Systems), Daniel Van Geest (ISARA), and Tim Hollebeek (Digicert).

We are grateful to all, including any contributors who may have been inadvertently omitted from this list.

This document borrows text from similar documents, including those referenced below. Thanks go to the authors of those documents.  
"Copying always makes things easier and less error prone" - [RFC8411].

## 8. References

### 8.1. Normative References

- [RFC1421] Linn, J., "Privacy Enhancement for Internet Electronic Mail: Part I: Message Encryption and Authentication Procedures", RFC 1421, DOI 10.17487/RFC1421, February 1993, <<https://www.rfc-editor.org/info/rfc1421>>.
- [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>>.
- [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>>.
- [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>>.
- [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>>.
- [RFC7468] Josefsson, S. and S. Leonard, "Textual Encodings of PKIX, PKCS, and CMS Structures", RFC 7468, DOI 10.17487/RFC7468, April 2015, <<https://www.rfc-editor.org/info/rfc7468>>.
- [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>>.

- [RFC8411] Schaad, J. and R. Andrews, "IANA Registration for the Cryptographic Algorithm Object Identifier Range", RFC 8411, DOI 10.17487/RFC8411, August 2018, <<https://www.rfc-editor.org/info/rfc8411>>.
- [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 8825-1:2015, November 2015.

## 8.2. Informative References

- [I-D.ounsworth-pq-composite-sigs]  
Ounsworth, M. and M. Pala, "Composite Keys and Signatures For Use In Internet PKI", Work in Progress, Internet-Draft, draft-ounsworth-pq-composite-sigs-03, 28 July 2020, <<http://www.ietf.org/internet-drafts/draft-ounsworth-pq-composite-sigs-03.txt>>.

### Authors' Addresses

Mike Ounsworth  
Entrust Limited  
2500 Solandt Road -- Suite 100  
Ottawa, Ontario K2K 3G5  
Canada

Email: [mike.ounsworth@entrust.com](mailto:mike.ounsworth@entrust.com)

Serge Mister  
Entrust Limited  
1000 Innovation Drive  
Ottawa, Ontario K2K 1E3  
Canada

Email: [serge.mister@entrust.com](mailto:serge.mister@entrust.com)

John Gray  
Entrust Limited  
1000 Innovation Drive  
Ottawa, Ontario  
Canada

Email: [john.gray@entrust.com](mailto:john.gray@entrust.com)



LAMPS Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: 8 September 2022

M. Richardson, Ed.  
Sandelman Software Works  
O. Friel  
Cisco  
D. von Oheimb  
Siemens  
D. Harkins  
The Industrial Lounge  
7 March 2022

Clarification of RFC7030 CSR Attributes definition  
draft-richardson-lamps-rfc7030-csrattrrs-02

Abstract

Enrollment over Secure Transport (EST) is ambiguous in specification of the CSR Attributes Response. This has resulted in implementation challenges and implementor confusion. This document updates EST and clarifies how the CSR Attributes Response can be used by an EST server to specify both CSR attribute OIDs and also CSR attribute values that the server expects the client to include in its CSR request.

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 8 September 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  |
| 2. CSR Attributes Handling . . . . .                                              | 3  |
| 2.1. Current EST Specification . . . . .                                          | 3  |
| 3. Updated CSR Attributes Handling . . . . .                                      | 4  |
| 3.1. Option two: Extend CSR structure to allow values: . . . .                    | 4  |
| 3.2. Option three: explicit content for the key<br>specification . . . . .        | 5  |
| 3.3. Option four: explicit members for unique attributes . . .                    | 7  |
| 3.4. Option five: more specific structure, simpler<br>extensions . . . . .        | 7  |
| 4. Co-existence with existing implementations . . . . .                           | 7  |
| 4.1. Use a new MIME type . . . . .                                                | 7  |
| 4.2. Use a new end point of the new format . . . . .                              | 8  |
| 4.3. Insist new format is upwardly compatible with old<br>format . . . . .        | 8  |
| 4.4. Return new format to new clients only . . . . .                              | 8  |
| 5. Whether or not to Base64 encoding of results . . . . .                         | 8  |
| 6. Examples . . . . .                                                             | 8  |
| 6.1. RFC8994/ACP subjectAltName with specific otherName<br>included . . . . .     | 8  |
| 6.2. EST server requires public keys of a specific size . . .                     | 8  |
| 6.3. EST server requires a public key of a specific algorithm/<br>curve . . . . . | 8  |
| 6.4. EST server requires a specific extension to be present .                     | 9  |
| 7. Security Considerations . . . . .                                              | 9  |
| 7.1. Identity and Privacy Considerations . . . . .                                | 9  |
| 8. IANA Considerations . . . . .                                                  | 9  |
| 9. Acknowledgements . . . . .                                                     | 9  |
| 10. Changelog . . . . .                                                           | 9  |
| 11. References . . . . .                                                          | 9  |
| 11.1. Normative References . . . . .                                              | 9  |
| 11.2. Informative References . . . . .                                            | 10 |
| Authors' Addresses . . . . .                                                      | 10 |

## 1. Introduction

Enrollment over Secure Transport [RFC7030] (EST) has been used in a wide variety of applications. In particular, [RFC8994] and [RFC8995] describe a way to use it in order to build out an autonomic control plane (ACP) [RFC8368].

The ACP requires that each node be given a very specific SubjectAltName. In the ACP specification, the solution was for the EST server to use section 2.6 of [RFC7030] to convey to the EST client the actual SubjectAltName that will end up in its certificate.

As a result of some implementation challenges, it came to light that this particular way of using the CSR attributes was not universally agreed upon, and in fact runs contrary to section 2.6. Section 2.6 says that the CSR attributes "provide additional descriptive information that the EST server cannot access itself". This extends to specifying that a particular attribute should exist, but not to the point of having the EST server actually specify the value.

The way in which the CSR attributes were understood by [RFC8994] turns out to be invalid. This document, therefore, updates section 2.6 to define this behavior.

This document also updates section 4.5 to include revised ASN.1 that covers all uses and is backward compatible with the existing use.

Additional examples are provided in an appendix.

## 2. CSR Attributes Handling

### 2.1. Current EST Specification

The ASN.1 for CSR Attributes as defined in EST section 4.5.2 is:

```
CsrAttrs ::= SEQUENCE SIZE (0..MAX) OF AttrOrOID

AttrOrOID ::= CHOICE (oid OBJECT IDENTIFIER, attribute Attribute )

Attribute { ATTRIBUTE:IOSet } ::= SEQUENCE {
    type    ATTRIBUTE.&id({IOSet}),
    values  SET SIZE(1..MAX) OF ATTRIBUTE.&Type({IOSet}{@type}) }
```

That section also states the following:

the values indicating the particular  
attributes desired to be included in the resulting certificate's  
extensions

This has been interpreted by some implementations as meaning that the CSR Attributes response can only include values for the attribute OIDs that the client should include in its CSR, and cannot include the actual values of those attributes. This is further reinforced by the example:

```
Attribute:  type = extensionRequest (1.2.840.113549.1.9.14)
            value = macAddress (1.3.6.1.1.1.1.22)
```

This example illustrates that the 'value' specified is an attribute OID, for example the macAddress OID, and not the value (such as "10-00-00-12-23-45") of the attribute itself.

There is no clearly documented mechanism with supporting examples that specifies how a CSR Attributes response can include a value for a given attribute such as SubjectAltName.

EST section 4.5.2 also states the following:

The structure of the CSR Attributes Response SHOULD, to the greatest extent possible, reflect the structure of the CSR it is requesting.

This statement aligns closely with the goal of this document. Additionally, EST Extensions [RFC8295] Appendix A has an informative appendix that outlines how a full CSR can be included in the CSR Attributes response.

### 3. Updated CSR Attributes Handling

The WG will pick one option as part of the adoption call.

#### 3.1. Option two: Extend CSR structure to allow values:

This ASN.1 needs fixing.

```

CsrAttrs ::= SEQUENCE SIZE (0..MAX) OF AttrOrOID

AttrOrOID ::= CHOICE (oid OBJECT IDENTIFIER,
                      attribute Attribute,
                      value Value )

Attribute { ATTRIBUTE:IOSet } ::= SEQUENCE {
    extType  ATTRIBUTE.&id({IOSet}),
    extAttr  SET SIZE(1..MAX) OF ATTRIBUTE.&Type({IOSet}{@type})
}

Value { ATTRIBUTE:IOSet } ::= SEQUENCE {
    extType  ATTRIBUTE.&id({IOSet}),
    type     ATTRIBUTE.&Type({IOSet}{@type}),
    value    OCTET STRING
}

```

This would just add a value to the SEQUENCE:

```

OBJECT challengePassword
SEQUENCE
  OBJECT subjectAltName
  SET
    OBJECT someACPGoo
SEQUENCE
  OBJECT id-ecPublicKey
  SET
    OBJECT secp384r1
    OBJECT ecdsa-with-SHA384

```

For example:

```

0 30: SEQUENCE {
2 28: SEQUENCE {
4 3: OBJECT IDENTIFIER subjectAltName (2 5 29 17)
9 21: SET {
11 19: [1] {
13 17: UTF8String 'hello@example.com'
: }
: }
: }
: }

```

### 3.2. Option three: explicit content for the key specification

The following options support complete and unambiguous specification of

- \* CSR ingredients optionally including values to use,
- \* the type of the public key, which is given in the form of a public-key algorithm,
- \* and the hash algorithm to use for the self-signature.

CSR ingredients may be the subject DN, any X.509 extensions, and special attributes like a challenge password.

For specifying the type of keys allowed in CSRs, they use a to-the-point KeySpec type. It can be defined for instance as

```
KeySpec ::= CHOICE {  
    keyAlg AlgorithmIdentifier,  
    rsaKeyLen INTEGER  
}
```

The keyAlg type is used to specify public-key algorithms and can include parameters, such as the name of an elliptic curve. The rsaKeyLen choice allows specifying the size of RSA keys, which it is not possible using values of type AlgorithmIdentifier.

The keySpec could also be a sequence of such specs, such that the server can give several key types from which the client can choose, e.g., EC keys on certain curves and/or RSA keys of certain sizes.

Stick for syntactic backward compatibility with

```
CsrAttrs ::= SEQUENCE SIZE (0..MAX) OF AttrOrOID
```

Each OID given in AttrOrOID must occur only once.

Plain OIDs are used mostly for challengePassword.

Attributes are used mostly for any X.509 extensions, subject DN, key spec, and hash alg, while defining new generally usable OIDs for

- \* a subject DN of type Name
- \* a key spec of type KeySpec
- \* a hash alg spec of type AlgorithmIdentifier

to be given on demand as attribute IDs of type ATTRIBUTE.&id({IOSet}).

### 3.3. Option four: explicit members for unique attributes

Define a new and more to-the-point type, which does not require new OIDs:

```
CsrAttrs ::= SEQUENCE {  
    oids      SEQUENCE OF OBJECT IDENTIFIER,  
    attrs     SEQUENCE OF Attribute,  
    subject   [0] Name OPTIONAL,  
    keySpec   [1] KeySpec OPTIONAL,  
    hashAlg   [2] AlgorithmIdentifier OPTIONAL  
}
```

Each OID given in oids or attrs must occur only once.

The oids are used mostly for requiring a challenge password.

The atttrs are used mostly for requiring certain X.509 extensions.

This is, typically just challengePassword and extensionRequest are used.

### 3.4. Option five: more specific structure, simpler extensions

Define a new fully to-the-point type, which does not require any (direct) OIDs:

```
CsrAttrs ::= SEQUENCE {  
    subject      Name OPTIONAL,  
    extensions   SEQUENCE OF Extension,  
    challengePassword BOOLEAN,  
    keySpec      [0] KeySpec OPTIONAL,  
    hashAlg      [1] AlgorithmIdentifier OPTIONAL  
}
```

## 4. Co-existence with existing implementations

There are some ways in which the new CSRattributes could co-exist with RFC7030.

### 4.1. Use a new MIME type

The client can signal that it supports the new attribute format by using an Accept: header in the transaction. This acts as a signal to a server that it can/should return the attributes in the new format.

#### 4.2. Use a new end point of the new format

Clients that want to use the new format would use a new end point, such as "csrvalues" which would only support the new format. A client which supported both would have to try both "csrvalues" and then fall back "csrattrs" if the EST server did not support the new format. Some uses (such as [RFC8994]) require the new format, so if it was not supported, that would be a protocol error.

#### 4.3. Insist new format is upwardly compatible with old format

ASN.1 encoding is self-describing, and some formats proposed above could possibly be parsed by legacy clients without a problem.

#### 4.4. Return new format to new clients only

The Registrar may know which clients are which by the kind of authentication that they do. An [RFC8994] client which has just performed a [RFC8995] enrollment would be assumed to require the new format only. A client which authenticates with an LDevID for a renewal would be strongly identified, and the Registrar could be programmed whether to return new format, or legacy CSR attributes.

#### 5. Whether or not to Base64 encoding of results

[RFC8951] clarified that the csrattrs end point was to be Base64 encoded even though the HTTP transport was 8-bit clean.

If this document establishes a new end point, then the new end point will not be base64 encoded according to current HTTP usage.

#### 6. Examples

##### 6.1. RFC8994/ACP subjectAltName with specific otherName included

TBD

##### 6.2. EST server requires public keys of a specific size

TBD

##### 6.3. EST server requires a public key of a specific algorithm/curve

TBD



#### 6.4. EST server requires a specific extension to be present

TBD

### 7. Security Considerations

All security considerations from EST [RFC7030] section 6 are applicable.

#### 7.1. Identity and Privacy Considerations

An EST server may use this mechanism to instruct the EST client about the identities it should include in the CSR it sends as part of enrollment. The client may only be aware of its IDevID Subject, which includes a manufacturer serial number. The EST server can use this mechanism to tell the client to include a specific fully qualified domain name in the CSR in order to complete domain ownership proofs required by the CA. Additionally, the EST server may deem the manufacturer serial number in an IDevID as personally identifiable information, and may want to specify a new random opaque identifier that the pledge should use in its CSR. This may be desirable if the CA and EST server have different operators.

### 8. IANA Considerations

None.

### 9. Acknowledgements

TODO

### 10. Changelog

### 11. References

#### 11.1. Normative References

- [BCP14] 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>>.
- [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>>.

- [RFC8994] Eckert, T., Ed., Behringer, M., Ed., and S. Bjarnason, "An Autonomic Control Plane (ACP)", RFC 8994, DOI 10.17487/RFC8994, May 2021, <<https://www.rfc-editor.org/info/rfc8994>>.
- [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>>.

## 11.2. Informative References

- [RFC8295] Turner, S., "EST (Enrollment over Secure Transport) Extensions", RFC 8295, DOI 10.17487/RFC8295, January 2018, <<https://www.rfc-editor.org/info/rfc8295>>.
- [RFC8368] Eckert, T., Ed. and M. Behringer, "Using an Autonomic Control Plane for Stable Connectivity of Network Operations, Administration, and Maintenance (OAM)", RFC 8368, DOI 10.17487/RFC8368, May 2018, <<https://www.rfc-editor.org/info/rfc8368>>.
- [RFC8951] Richardson, M., Werner, T., and W. Pan, "Clarification of Enrollment over Secure Transport (EST): Transfer Encodings and ASN.1", RFC 8951, DOI 10.17487/RFC8951, November 2020, <<https://www.rfc-editor.org/info/rfc8951>>.

## Authors' Addresses

Michael Richardson (editor)  
Sandelman Software Works  
Email: [mcr+ietf@sandelman.ca](mailto:mcr+ietf@sandelman.ca)

Owen Friel  
Cisco  
Email: [ofriel@cisco.com](mailto:ofriel@cisco.com)

Dr. David von Oheimb  
Siemens  
Email: [dev@ddvo.net](mailto:dev@ddvo.net)

Dan Harkins  
The Industrial Lounge  
Email: [dharkins@lounge.org](mailto:dharkins@lounge.org)