

Workgroup: openpgp  
Internet-Draft: draft-autocrypt-lamps-protected-headers-02  
Published: 20 December 2019  
Intended Status: Informational  
Expires: 22 June 2020  
Authors: B.R. Einarsson . juga D.K. Gillmor  
*Mailpile ehf* *Independent* *ACLU*

# Protected Headers for Cryptographic E-mail

---

## Abstract

This document describes a common strategy to extend the end-to-end cryptographic protections provided by PGP/MIME, etc. to protect message headers in addition to message bodies. In addition to protecting the authenticity and integrity of headers via signatures, it also describes how to preserve the confidentiality of the Subject header.

## 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 22 June 2020.

## Copyright Notice

Copyright (c) 2019 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
  - 1.1. Requirements Language
  - 1.2. Terminology
    - 1.2.1. User-Facing Headers
    - 1.2.2. Structural Headers
- 2. Protected Headers Summary
- 3. Cryptographic MIME Message Structure
  - 3.1. Cryptographic Layers
    - 3.1.1. PGP/MIME Cryptographic Layers
    - 3.1.2. S/MIME Cryptographic Layers
  - 3.2. Cryptographic Envelope
  - 3.3. Cryptographic Payload
    - 3.3.1. Simple Cryptographic Payloads
    - 3.3.2. Multilayer Cryptographic Envelopes
    - 3.3.3. A Baroque Example
  - 3.4. Exposed Headers are Outside
- 4. Message Composition
  - 4.1. Copying All Headers
  - 4.2. Confidential Subject
  - 4.3. Obscured Headers
  - 4.4. Message Composition without Protected Headers
  - 4.5. Message Composition with Protected Headers
- 5. Legacy Display
  - 5.1. Message Generation: Including a Legacy Display Part
    - 5.1.1. Legacy Display Transformation
    - 5.1.2. When to Generate Legacy Display
  - 5.2. Message Rendering: Omitting a Legacy Display Part
    - 5.2.1. Legacy Display Detection Algorithm

- 5.3. Legacy Display is Decorative and Transitional
- 6. Message Interpretation
  - 6.1. Reverse-Copying
  - 6.2. Signature Invalidation
  - 6.3. The Legacy Display Part
  - 6.4. Replying to a Message with Obscured Headers
- 7. Common Pitfalls and Guidelines
  - 7.1. Misunderstood Obscured Subjects
  - 7.2. Reply/Forward Losing Subjects
  - 7.3. Usability Impact of Reduced Metadata
  - 7.4. Usability Impact of Obscured Message-ID
  - 7.5. Usability Impact of Obscured From/To/Cc
  - 7.6. Mailing List Header Modifications
- 8. Comparison with Other Header Protection Schemes
  - 8.1. S/MIME 3.1 Header Protection
  - 8.2. The Content-Type Property "forwarded=no" {forwarded=no}
  - 8.3. pEp Header Protection
  - 8.4. DKIM
  - 8.5. S/MIME "Secure Headers"
  - 8.6. Triple-Wrapping
- 9. Test Vectors
  - 9.1. Signed PGP/MIME Message with Protected Headers
  - 9.2. S/MIME multipart/signed Message with Protected Headers
  - 9.3. S/MIME application/pkcs7-mime SignedData Message with Protected Headers
  - 9.4. Signed and Encrypted PGP/MIME Message with Protected Headers
  - 9.5. Signed and Encrypted S/MIME Message with Protected Headers
  - 9.6. Signed and Encrypted PGP/MIME Message with Protected Headers and Legacy Display Part
  - 9.7. Multilayer PGP/MIME Message with Protected Headers
  - 9.8. Multilayer PGP/MIME Message with Protected Headers and Legacy Display Part

9.9. Signed and Encrypted S/MIME Message with Protected Headers and Legacy Display
9.10. Encrypted-only (unsigned) S/MIME Message with Protected Headers and Legacy Display
9.11. Encrypted-only (unsigned) PGP/MIME Message with Protected Headers and Legacy Display
9.12. An Unfortunately Complex Example
10. IANA Considerations
11. Security Considerations
11.1. Subject Leak
11.2. Signature Replay
11.3. Participant Modification
12. Privacy Considerations
13. Document Considerations
13.1. Document History
14. Acknowledgements
15. References
15.1. Normative References
15.2. Informative References
Authors' Addresses

## 1. Introduction

E-mail end-to-end security with OpenPGP and S/MIME standards can provide integrity, authentication, non-repudiation and confidentiality to the body of a MIME e-mail message. However, PGP/MIME ([RFC3156]) alone does not protect message headers. And the structure to protect headers defined in S/MIME 3.1 ([RFC3851]) has not seen widespread adoption.

This document defines a scheme, "Protected Headers for Cryptographic E-mail", which has been adopted by multiple existing e-mail clients in order to extend the cryptographic protections provided by PGP/MIME to also protect the message headers. This scheme is also applicable to S/MIME [RFC8551].

This document describes how these protections can be applied to cryptographically signed messages, and also discusses some of the challenges of encrypting many transit-oriented headers.

It offers guidance for protecting the confidentiality of non-transit-oriented headers like Subject, and also offers a means to preserve backwards compatibility so that an encrypted Subject remains available to recipients using software that does not implement support for the Protected Headers scheme.

The document also discusses some of the compatibility constraints and usability concerns which motivated

the design of the scheme, as well as limitations and a comparison with other proposals.

This technique has already proven itself as a useful building block for other improvements to cryptographic e-mail, such as the Autocrypt Level 1.1 ([\[Autocrypt\]](#)) "Gossip" mechanism.

## 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

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 or both.
- *Cryptographic Layer*, *Cryptographic Envelope* and *Cryptographic Payload* are defined in [Section 3](#)
- *Original Headers* are the [[RFC5322](#)] message headers as known to the sending MUA at the time of message composition.
- *Protected Headers* are any headers protected by the scheme described in this document.
- *Exposed Headers* are any headers outside the Cryptographic Payload (protected or not).
- *Obscured Headers* are any Protected Headers which have been modified or removed from the set of Exposed Headers.
- *Legacy Display Part* is a MIME construct which provides visibility for users of legacy clients of data from the Original Headers which may have been removed or obscured from the Exposed Headers. It is defined in [Section 5](#).
- *User-Facing Headers* are explained and enumerated in [Section 1.2.1](#).
- *Structural Headers* are documented in [Section 1.2.2](#).

### 1.2.1. User-Facing Headers

Of all the headers that an e-mail message may contain, only a handful are typically presented directly to the user. The user-facing headers are:

- Subject
- From
- To
- Cc
- Date
- Reply-To
- Followup-To

The above is a complete list. No other headers are considered "user-facing".

Other headers may affect the visible rendering of the message (e.g., References and In-Reply-To may

affect the placement of a message in a threaded discussion), but they are not directly displayed to the user and so are not considered "user-facing" for the purposes of this document.

### 1.2.2. 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

Note that no "user-facing" headers ([Section 1.2.1](#)) are also "structural" headers. Of course, many headers are neither "user-facing" nor "structural".

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

## 2. Protected Headers Summary

The Protected Headers scheme relies on three backward-compatible changes to a cryptographically-protected e-mail message:

- Headers known to the composing MUA at message composition time are (in addition to their typical placement as Exposed Headers on the outside of the message) also present in the MIME header of the root of the Cryptographic Payload. These Protected Headers share cryptographic properties with the rest of the Cryptographic Payload.
- When the Cryptographic Envelope includes encryption, any Exposed Header MAY be *obscured* by a transformation (including deletion).
- If the composing MUA intends to obscure any user-facing headers, it MAY add a decorative "Legacy Display" MIME part to the Cryptographic Payload which additionally duplicates the original values of the obscured user-facing headers.

When a composing MUA encrypts a message, it SHOULD obscure the Subject: header, by using the literal string ... (three U+002E FULL STOP characters) as the value of the exposed Subject: header.

When a receiving MUA encounters a message with a Cryptographic Envelope, it treats the headers of the Cryptographic Payload as belonging to the message itself, not just the subpart. In particular, when rendering a header for any such message, the renderer SHOULD prefer the header's Protected value over its Exposed value.

A receiving MUA that understands Protected Headers and discovers a Legacy Display part SHOULD hide the Legacy Display part when rendering the message.

The following sections contain more detailed discussion.

## 3. 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.

### 3.1. 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".

#### 3.1.1. PGP/MIME Cryptographic Layers

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

##### 3.1.1.1. PGP/MIME Signing Cryptographic Layer (multipart/signed)

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

##### 3.1.1.2. PGP/MIME Encryption Cryptographic Layer (multipart/encrypted)

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

#### 3.1.2. 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.

##### 3.1.2.1. S/MIME Multipart Signed Cryptographic Layer

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

##### 3.1.2.2. S/MIME PKCS7 signed-data Cryptographic Layer

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

##### 3.1.2.3. S/MIME PKCS7 enveloped-data Cryptographic Layer

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

#### 3.1.2.4. S/MIME PKCS7 authEnveloped-data Cryptographic Layer

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

Note that enveloped-data ([Section 3.1.2.3](#)) and authEnveloped-data ([Section 3.1.2.4](#)) 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.

#### 3.1.2.5. 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.

### 3.2. 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 (see the example in [Section 3.3.3](#)).

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.

### 3.3. Cryptographic Payload

The Cryptographic Payload of a message is the first non-Cryptographic Layer - the "protected part" - within the Cryptographic Envelope. Since the Cryptographic Payload itself is a MIME part, it has its own set of headers.

Protected headers are placed on (and read from) the Cryptographic Payload, and should be considered to have the same cryptographic properties as the message itself.

#### 3.3.1. Simple Cryptographic Payloads

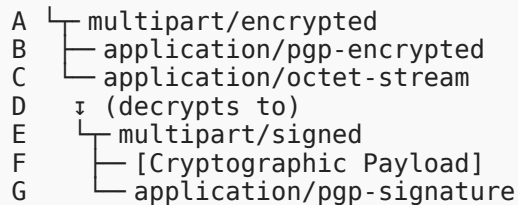
As described above, if the "protected part" identified in [Section 3.1.1.1](#) or [Section 3.1.1.2](#) is not itself a Cryptographic Layer, that part *is* the Cryptographic Payload.



If the application wants to generate a message that is both encrypted and signed, it MAY use the simple MIME structure from [Section 3.1.1.2](#) by ensuring that the [\[RFC4880\]](#) Encrypted Message within the application/octet-stream part contains an [\[RFC4880\]](#) Signed Message.

### 3.3.2. Multilayer Cryptographic Envelopes

It is possible to construct a Cryptographic Envelope consisting of multiple layers for PGP/MIME, typically of the following structure:



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

As noted in [Section 3.3.1](#), PGP/MIME applications also have a simpler MIME construction available with the same cryptographic properties.

### 3.3.3. A Baroque Example

Consider a message with the following overcomplicated structure:



The 3 Cryptographic Layers in such a message are rooted in parts H, L, and N. But the Cryptographic Envelope of the message consists only of the properties derived from the series H, L. The Cryptographic Payload of the message is part M.

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 O compared to part Q.

## 3.4. Exposed Headers are Outside

The Cryptographic Envelope fully encloses the Cryptographic Payload, whether the message is signed or encrypted or both. The Exposed Headers are considered to be outside of both.

## 4. Message Composition

This section describes the composition of a cryptographically-protected message with Protected Headers.

We document legacy composition of cryptographically-protected messages (without protected headers) in [Section 4.4](#), and then describe a revised version of that algorithm in [Section 4.5](#) that produces conformant Protected Headers.

#### 4.1. Copying All Headers

All non-structural headers known to the composing MUA are copied to the MIME header of the Cryptographic Payload. The composing MUA SHOULD protect all known non-structural headers in this way.

If the composing MUA omits protection for some of the headers, the receiving MUA will have difficulty reasoning about the integrity of the headers (see [Section 11.2](#)).

#### 4.2. Confidential Subject

When a message is encrypted, the Subject should be obscured by replacing the Exposed Subject with three periods: ...

This value (...) was chosen because it is believed to be language agnostic and avoids communicating any potentially misleading information to the recipient (see [Section 7.1](#) for a more detailed discussion).

#### 4.3. Obscured Headers

Due to compatibility and usability concerns, a Mail User Agent SHOULD NOT obscure any of: From, To, Cc, Message-ID, References, Reply-To, In-Reply-To, (FIXME: MORE?) unless the user has indicated they have security constraints which justify the potential downsides (see [Section 7](#) for a more detailed discussion).

Aside from that limitation, this specification does not at this time define or limit the methods a MUA may use to convert Exposed Headers into Obscured Headers.

#### 4.4. Message Composition without Protected Headers

This section roughly describes the steps that a legacy MUA might use to compose a cryptographically-protected message *without* Protected Headers.

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 1.2.2](#)).
- **origheaders**: the intended non-structural headers for the message, represented here as a table mapping from header names to header values.. For example, origheaders[ 'From' ] refers to the value of the From header that the composing MUA would typically place on the message before sending it.
- **crypto**: The series of cryptographic protections to apply (for example, "sign with the secret key corresponding to OpenPGP certificate X, then encrypt to OpenPGP 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 header name `h` in `origheaders`:
  - Set header `h` of output to `origheaders[h]`
- Return output

## 4.5. Message Composition with Protected Headers

A reasonable sequential algorithm for composing a message *with* protected headers takes two more parameters in addition to `origbody`, `origheaders`, and `crypto`:

- `obscures`: a table of headers to be obscured during encryption, mapping header names to their obscuring values. For example, this document recommends only obscuring the subject, so that would be represented by the single-entry table `obscures = { 'Subject': '...' }`. If header `Foo` is to be deleted entirely, `obscures[ 'Foo' ]` should be set to the special value `null`.
- `legacy`: a boolean value, indicating whether any recipient of the message is believed to have a legacy client (that is, a MUA that is capable of decryption, but does not understand protected headers).

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

- if `crypto` contains encryption, and `legacy` is `true`, and `obscures` contains any user-facing headers (see [Section 1.2.1](#)), wrap `orig` in a structure that carries a Legacy Display part:
  - Create a new MIME leaf part `legacydisplay` with header `Content-Type: text/plain; protected-headers="v1"`
  - For each obscured header name `obh` in `obscures`:
    - If `obh` is user-facing:
      - Add `obh: origheaders[ob]` to the body of `legacydisplay`. For example, if `origheaders[ 'Subject' ]` is `lunch plans?`, then add the line `Subject: lunch plans?` to the body of `legacydisplay`
  - Construct a new MIME part wrapper with `Content-Type: multipart/mixed`
  - Give wrapper exactly two subparts: `legacydisplay` and `origbody`, in that order.
  - Let payload be MIME part wrapper
- Otherwise:
  - Let payload be MIME part `origbody`
- For each header name `h` in `origheaders`:
  - Set header `h` of MIME part payload to `origheaders[h]`
- Set the `protected-headers` parameter on the `Content-Type` of payload to `v1`
- Apply `crypto` to payload, producing MIME tree output
- If `crypto` contains encryption:
  - For each obscured header name `obh` in `obscures`:

- If `obscures[obh]` is `null`:
  - Drop `obh` from `origheaders`
- Else:
  - Set `origheaders[obh]` to `obscures[obh]`
- For each header name `h` in `origheaders`:
  - Set header `h` of output to `origheaders[h]`
- return output

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

## 5. Legacy Display

MUAs typically display user-facing headers ([Section 1.2.1](#)) directly to the user. An encrypted message may be read by a decryption-capable legacy MUA that is unaware of this standard. The user of such a legacy client risks losing access to any obscured headers.

This section presents a workaround to mitigate this risk by restructuring the Cryptographic Payload before encrypting to include a "Legacy Display" part.

### 5.1. Message Generation: Including a Legacy Display Part

A generating MUA that wants to make an Obscured Subject (or any other user-facing header) visible to a recipient using a legacy MUA SHOULD modify the Cryptographic Payload by wrapping the intended body of the message in a `multipart/mixed` MIME part that prefixes the intended body with a Legacy Display part.

The Legacy Display part MUST be of Content-Type `text/plain` or `text/rfc822-headers` (`text/plain` is RECOMMENDED), and MUST contain a `protected-headers` parameter whose value is `v1`. It SHOULD be marked with `Content-Disposition: inline` to encourage recipients to render it.

The contents of the Legacy Display part MUST be only the user-facing headers that the sending MUA intends to obscure after encryption.

The original body (now a subpart) SHOULD also be marked with `Content-Disposition: inline` to discourage legacy clients from presenting it as an attachment.

#### 5.1.1. Legacy Display Transformation

Consider a message whose Cryptographic Payload, before encrypting, that would have a traditional `multipart/alternative` structure:

```
X  └─ multipart/alternative
Y  └─ text/plain
Z  └─ text/html
```

When adding a Legacy Display part, this structure becomes:

```
V └─ multipart/mixed
W └─ text/plain ("Legacy Display" part)
X └─ multipart/alternative ("original body")
Y └─ text/plain
Z └─ text/html
```

Note that with the inclusion of the Legacy Display part, the Cryptographic Payload is the `multipart/mixed` part (part V in the example above), so Protected Headers should be placed at that part.

### 5.1.2. When to Generate Legacy Display

A MUA SHOULD transform a Cryptographic Payload to include a Legacy Display part only when:

- The message is going to be encrypted, and
- At least one user-facing header (see [Section 1.2.1](#)) is going to be obscured

Additionally, if the sender knows that the recipient's MUA is capable of interpreting Protected Headers, it SHOULD NOT attempt to include a Legacy Display part. (Signalling such a capability is out of scope for this document)

## 5.2. Message Rendering: Omitting a Legacy Display Part

A MUA that understands Protected Headers may receive an encrypted message that contains a Legacy Display part. Such an MUA SHOULD avoid rendering the Legacy Display part to the user at all, since it is aware of and can render the actual Protected Headers.

If a Legacy Display part is detected, the Protected Headers should still be pulled from the Cryptographic Payload (part V in the example above), but the body of message SHOULD be rendered as though it were only the original body (part X in the example above).

### 5.2.1. Legacy Display Detection Algorithm

A receiving MUA acting on a message SHOULD detect the presence of a Legacy Display part and the corresponding "original body" with the following simple algorithm:

- Check that all of the following are true for the message:
- The Cryptographic Envelope must contain an encrypting Cryptographic Layer
- The Cryptographic Payload must have a Content-Type of `multipart/mixed`
- The Cryptographic Payload must have exactly two subparts
- The first subpart of the Cryptographic Payload must have a Content-Type of `text/plain` or `text/rfc822-headers`
- The first subpart of the Cryptographic Payload's Content-Type must contain a property of `protected-headers`, and its value must be `v1`.
- If all of the above are true, then the first subpart is the Legacy Display part, and the second subpart is the "original body". Otherwise, the message does not have a Legacy Display part.

## 5.3. Legacy Display is Decorative and Transitional

As the above makes clear, the Legacy Display part is strictly decorative, for the benefit of legacy decryption-capable MUAs that may handle the message. As such, the existence of the Legacy Display part and its `multipart/mixed` wrapper are part of a transition plan.

As the number of decryption-capable clients that understand Protected Headers grows in comparison to the number of legacy decryption-capable clients, it is expected that some senders will decide to stop generating Legacy Display parts entirely.

A MUA developer concerned about accessibility of the Subject header for their users of encrypted mail when Legacy Display parts are omitted SHOULD implement the Protected Headers scheme described in this document.

## 6. Message Interpretation

This document does not currently provide comprehensive recommendations on how to interpret Protected Headers. This is deliberate; research and development is still ongoing. We also recognize that the tolerance of different user groups for false positives (benign conditions misidentified as security risks), vs. their need for strong protections varies a great deal and different MUAs will take different approaches as a result.

Some common approaches are discussed below.

### 6.1. Reverse-Copying

One strategy for interpreting Protected Headers on an incoming message is to simply ignore any Exposed Header for which a Protected counterpart is available. This is often implemented as a copy operation (copying header back out of the Cryptographic Payload into the main message header) within the code which takes care of parsing the message.

A MUA implementing this strategy should pay special attention to any user facing headers ([Section 1.2.1](#)). If a message has Protected Headers, and a user-facing header is among the Exposed Headers but missing from the Protected Headers, then an MUA implementing this strategy SHOULD delete the identified Exposed Header before presenting the message to the user.

This strategy does not risk raising a false alarm about harmless deviations, but conversely it does nothing to inform the user if they are under attack. This strategy does successfully mitigate and thwart some attacks, including signature replay attacks ([Section 11.2](#)) and participant modification attacks ([Section 11.3](#)).

### 6.2. Signature Invalidation

An alternate strategy for interpreting Protected Headers is to consider the cryptographic signature on a message to be invalid if the Exposed Headers deviate from their Protected counterparts.

This state should be presented to the user using the same interface as other signature verification failures.

A MUA implementing this strategy MAY want to make a special exception for the Subject: header, to avoid invalidating the signature on any signed and encrypted message with a confidential subject.

Note that simple signature invalidation may be insufficient to defend against a participant modification attack ([Section 11.3](#)).

### 6.3. The Legacy Display Part

This part is purely decorative, for the benefit of any recipient using a legacy decryption-capable MUA. See [Section 5.2](#) for details and recommendations on how to handle the Legacy Display part.

## 6.4. Replying to a Message with Obscured Headers

When replying to a message, many MUAs copy headers from the original message into their reply.

When replying to an encrypted message, users expect the replying MUA to generate an encrypted message if possible. If encryption is not possible, and the reply will be cleartext, users typically want the MUA to avoid leaking previously-encrypted content into the cleartext of the reply.

For this reason, an MUA replying to an encrypted message with Obscured Headers SHOULD NOT leak the cleartext of any Obscured Headers into the cleartext of the reply, whether encrypted or not.

In particular, the contents of any Obscured Protected Header from the original message SHOULD NOT be placed in the Exposed Headers of the reply message.

## 7. Common Pitfalls and Guidelines

Among the MUA authors who already implemented most of this specification, several alternative or more encompassing specifications were discussed and sometimes tried out in practice. This section highlights a few "pitfalls" and guidelines based on these discussions and lessons learned.

### 7.1. Misunderstood Obscured Subjects

There were many discussions around what text phrase to use to obscure the Subject:. Text phrases such as Encrypted Message were tried but resulted in both localization problems and user confusion.

If the natural language phrase for the obscured Subject: is not localized (e.g. just English Encrypted Message), then it may be incomprehensible to a non-English-speaking recipient who uses a legacy MUA that renders the obscured Subject: directly.

On the other hand, if it is localized based on the sender's MUA language settings, there is no guarantee that the recipient prefers the same language as the sender (consider a German speaker sending English text to an Anglophone). There is no standard way for a sending MUA to infer the language preferred by the recipient (aside from statistical inference of language based on the composed message, which would in turn leak information about the supposedly-confidential message body).

Furthermore, implementors found that the phrase Encrypted Message in the subject line was sometimes understood by users to be an indication from the MUA that the message was actually encrypted. In practice, when some MUA failed to encrypt a message in a thread that started off with an obscured Subject:, the value Re: Encrypted Message was retained even on those cleartext replies, resulting in user confusion.

In contrast, using ... as the obscured Subject: was less likely to be seen as an indicator from the MUA of message encryption, and it also neatly sidesteps the localization problems.

### 7.2. Reply/Forward Losing Subjects

When the user of a legacy MUA replies to or forwards a message where the Subject has been obscured, it is likely that the new subject will be Fwd: ... or Re: ... (or the localized equivalent). This breaks an important feature: people are used to continuity of subject within a thread. It is especially unfortunate when a new participant is added to a conversation who never saw the original subject.

At this time, there is no known workaround for this problem. The only solution is to upgrade the MUA to support Protected Headers.

The authors consider this to be only a minor concern in cases where encryption is being used because confidentiality is important. However, in more opportunistic cases, where encryption is being used routinely regardless of the sensitivity of message contents, this cost becomes higher.

### 7.3. Usability Impact of Reduced Metadata

Many mail user agents maintain an index of message metadata (including header data), which is used to rapidly construct mailbox overviews and search result listings. If the process which generates this index does not have access to the encrypted payload of a message, or does not implement Protected Headers, then the index will only contain the obscured versions Exposed Headers, in particular an obscured Subject of ...

For sensitive message content, especially in a hosted MUA-as-a-service situation ("webmail") where the metadata index is maintained and stored by a third party, this may be considered a feature as the subject is protected from the third-party. However, for more routine communications, this harms usability and goes against user expectations.

Two simple workarounds exist for this use case:

1. If the metadata index is considered secure enough to handle confidential data, the protected content may be stored directly in the index once it has been decrypted.
2. If the metadata index is not trusted, the protected content could be re-encrypted and encrypted versions stored in the index instead, which are then decrypted by the client at display time.

In both cases, the process which decrypts the message and processes the Protected Headers must be able to update the metadata index.

FIXME: add notes about research topics and other non-simple workarounds, like oblivious server-side indexing, or searching on encrypted data.

### 7.4. Usability Impact of Obscured Message-ID

Current MUA implementations rely on the outermost Message-ID for message processing and indexing purposes. This processing often happens before any decryption is even attempted. Attempting to send a message with an obscured Message-ID header would result in several MUAs not correctly processing the message, and would likely be seen as a degradation by users.

Furthermore, a legacy MUA replying to a message with an obscured Message-ID: would be likely to produce threading information (References:, In-Reply-To:) that would be misunderstood by the original sender. Implementors generally disapprove of breaking threads.

### 7.5. Usability Impact of Obscured From/To/Cc

The impact of obscuring From:, To:, and Cc: headers has similar issues as discussed with obscuring the Message-ID: header in [Section 7.4](#).

In addition, obscuring these headers is likely to cause difficulties for a legacy client attempting formulate a correct reply (or "reply all") to a given message.



## 7.6. Mailing List Header Modifications

Some popular mailing-list implementations will modify the Exposed Headers of a message in specific, benign ways. In particular, it is common to add markers to the Subject line, and it is also common to modify either From or Reply-To in order to make sure replies go to the list instead of directly to the author of an individual post.

Depending on how the MUA resolves discrepancies between the Protected Headers and the Exposed Headers of a received message, these mailing list "features" may either break or the MUA may incorrectly interpret them as a security breach.

Implementors may for this reason choose to implement slightly different strategies for resolving discrepancies, if a message is known to come from such a mailing list. MUAs should at the very least avoid presenting false alarms in such cases.

## 8. Comparison with Other Header Protection Schemes

Other header protection schemes have been proposed (in the IETF and elsewhere) that are distinct from this mechanism. This section documents the differences between those earlier mechanisms and this one, and hypothesizes why it has seen greater interoperable adoption.

The distinctions include:

- backward compatibility with legacy clients
- compatibility across PGP/MIME and S/MIME
- protection for both confidentiality and signing

### 8.1. S/MIME 3.1 Header Protection

S/MIME 3.1 ([RFC3851]) introduces header protection via message/rfc822 header parts.

The problem with this mechanism is that many legacy clients encountering such a message were likely to interpret it as either a forwarded message, or as an unreadable substructure.

For signed messages, this is particularly problematic - a message that would otherwise have been easily readable by a client that knows nothing about signed messages suddenly shows up as a message-within-a-message, just by virtue of signing. This has an impact on *all* clients, whether they are cryptographically-capable or not.

For encrypted messages, whose interpretation only matters on the smaller set of cryptographically-capable legacy clients, the resulting message rendering is awkward at best.

Furthermore, formulating a reply to such a message on a legacy client can also leave the user with badly-structured quoted and attributed content.

Additionally, a message deliberately forwarded in its own right (without preamble or adjacent explanatory notes) could potentially be confused with a message using the declared structure.

The mechanism described here allows cryptographically-incapable legacy MUAs to read and handle cleartext signed messages without any modifications, and permits cryptographically-capable legacy MUAs

to handle encrypted messages without any modifications.

In particular, the Legacy Display part described in [Section 5](#) makes it feasible for a conformant MUA to generate messages with obscured Subject lines that nonetheless give access to the obscured Subject header for recipients with legacy MUAs.

## 8.2. The Content-Type Property "forwarded=no" {forwarded=no}

Section A.1.2 of [\[I-D.draft-ietf-lamps-header-protection-requirements-01\]](#) refers to a proposal that attempts to mitigate one of the drawbacks of the scheme described in S/MIME 3.1 ([Section 8.1](#)).

In particular, using the Content-Type property `forwarded="no"` allows *non-legacy* clients to distinguish between deliberately forwarded messages and those intended to use the defined structure for header protection.

However, this fix has no impact on the confusion experienced by legacy clients.

## 8.3. pEp Header Protection

[\[I-D.draft-luck-lamps-pep-header-protection-03\]](#) is applicable only to signed+encrypted mail, and does not contemplate protection of signed-only mail.

In addition, the pEp header protection involved for "pEp message format 2" has an additional `multipart/mixed` layer designed to facilitate transfer of OpenPGP Transferable Public Keys, which seems orthogonal to the effort to protect headers.

Finally, that draft suggests that the exposed Subject header be one of `"=?utf-8?Q?p=E2=89=A1p?="`, "pEp", or "Encrypted message". "pEp" is a mysterious choice for most users, and see [Section 7.1](#) for more commentary on why "Encrypted message" is likely to be problematic.

## 8.4. DKIM

[\[RFC6736\]](#) offers DKIM, which is often used to sign headers associated with a message.

DKIM is orthogonal to the work described in this document, since it is typically done by the domain operator and not the end user generating the original message. That is, DKIM is not "end-to-end" and does not represent the intent of the entity generating the message.

Furthermore, a DKIM signer does not have access to headers inside an encrypted Cryptographic Layer, and a DKIM verifier cannot effectively use DKIM to verify such confidential headers.

## 8.5. S/MIME "Secure Headers"

[\[RFC7508\]](#) describes a mechanism that embeds message header fields in the S/MIME signature using ASN.1.

The mechanism proposed in that draft is undefined for use with PGP/MIME. While all S/MIME clients must be able to handle CMS and ASN.1 as well as MIME, a standard that works at the MIME layer itself should be applicable to any MUA that can work with MIME, regardless of whether end-to-end security layers are provided by S/MIME or PGP/MIME.

That mechanism also does not propose a means to provide confidentiality protection for headers within an

encrypted-but-not-signed message.

Finally, that mechanism offers no equivalent to the Legacy Display described in [Section 5](#). Instead, sender and receiver are expected to negotiate in some unspecified way to ensure that it is safe to remove or modify Exposed Headers in an encrypted message.

## 8.6. Triple-Wrapping

[\[RFC2634\]](#) defines "Triple Wrapping" as a means of providing cleartext signatures over signed and encrypted material. This can be used in combination with the mechanism described in [\[RFC7508\]](#) to authenticate some headers for transport using S/MIME.

But it does not offer confidentiality protection for the protected headers, and the signer of the outer layer of a triple-wrapped message may not be the originator of the message either.

In practice on today's Internet, DKIM ([\[RFC6736\]](#)) provides a more widely-accepted cryptographic header-verification-for-transport mechanism than triple-wrapped messages.

## 9. Test Vectors

The subsections below provide example messages that implement the Protected Header scheme.

The secret keys and OpenPGP certificates from [\[I-D.draft-bre-openpgp-samples-00\]](#) can be used to decrypt and verify the PGP/MIME messages.

The secret keys and X.509 certificates from [\[I-D.draft-dkg-lamps-samples-01\]](#) can be used to decrypt and verify the S/MIME messages.

All test vectors are provided in textual source form as [\[RFC5322\]](#) messages.

For easy access to these test vectors, they are also available at `imap://bob@protected-headers.cmrg.net/inbox` using any password for authentication. This IMAP account is read-only, and any flags set or cleared on the messages will persist only for the duration of the specific IMAP session.

### 9.1. Signed PGP/MIME Message with Protected Headers

This shows a clearsIGNED PGP/MIME message. Its MIME message structure is:

```
└ multipart/signed
  └ text/plain ← Cryptographic Payload
  └ application/pgp-signature
```

Note that if this message had been generated without Protected Headers, then an attacker with access to it could modify the Subject without invalidating the signature. Such an attacker could cause Bob to think that Alice wanted to cancel the contract with BarCorp instead of FooCorp.

```

Received: from localhost (localhost [127.0.0.1]); Sun, 20 Oct 2019
09:00:17 -0400 (UTC-04:00)
MIME-Version: 1.0
Content-Type: multipart/signed; boundary="fee";
    protocol="application/pgp-signature"; micalg="pgp-sha512"
From: Alice Lovelace <alice@openpgp.example>
To: Bob Babbage <bob@openpgp.example>
Date: Sun, 20 Oct 2019 09:00:00 -0400
Subject: The FooCorp contract
Message-ID: <pgpmime-signed@protected-headers.example>

--fee
Content-Type: text/plain; charset="us-ascii"; protected-headers="v1"
From: Alice Lovelace <alice@openpgp.example>
To: Bob Babbage <bob@openpgp.example>
Date: Sun, 20 Oct 2019 09:00:00 -0400
Subject: The FooCorp contract
Message-ID: <pgpmime-signed@protected-headers.example>

Bob, we need to cancel this contract.

Please start the necessary processes to make that happen today.

(this is the 'pgpmime-signed' message)

Thanks, Alice
--
Alice Lovelace
President
Example Corp

--fee
content-type: application/pgp-signature

-----BEGIN PGP SIGNATURE-----

wnUEARYKAB0FAl2sWlAWIQT rhbtfozp14V6UTmPyMVUMT0fjjgAKCRDyMVUMT0fj
jtl0AQDtIsRWZVCjbB3TISlcyxLpBfwjaXXV0is5+c4Gd2NNgwEAipDF3m5zIt7t
29cFwQusmCqKqKfdJUf6H0UPF5L/zAI=
=+M9u
-----END PGP SIGNATURE-----

--fee--

```

## 9.2. S/MIME multipart/signed Message with Protected Headers

This shows a signed-only S/MIME message using the multipart/signed style (see Section 3.5.3 of [RFC8551](#)). Its MIME message structure is:

```

└─ multipart/signed
   └─ text/plain ← Cryptographic Payload
      └─ application/pkcs7-signature

```

Note that if this message had been generated without Protected Headers, then an attacker with access to it could modify the Subject without invalidating the signature. Such an attacker could cause Bob to think that Alice wanted to cancel the contract with BarCorp instead of FooCorp.

Received: from localhost (localhost [127.0.0.1]); Tue, 26 Nov 2019  
20:03:17 -0400 (UTC-04:00)  
MIME-Version: 1.0  
Content-Type: multipart/signed; boundary="179";  
protocol="application/pkcs7-signature"; micalg="sha-256"  
From: Alice Lovelace <alice@smime.example>  
To: Bob Babbage <bob@smime.example>  
Date: Tue, 26 Nov 2019 20:03:00 -0400  
Subject: The FooCorp contract  
Message-ID: <smime-multipart-signed@protected-headers.example>

--179  
Content-Type: text/plain; charset="us-ascii"; protected-headers="v1"  
From: Alice Lovelace <alice@smime.example>  
To: Bob Babbage <bob@smime.example>  
Date: Tue, 26 Nov 2019 20:03:00 -0400  
Subject: The FooCorp contract  
Message-ID: <smime-multipart-signed@protected-headers.example>

Bob, we need to cancel this contract.

Please start the necessary processes to make that happen today.

(this is the 'smime-multipart-signed' message)

Thanks, Alice  
--  
Alice Lovelace  
President  
Example Corp

--179  
Content-Transfer-Encoding: base64  
Content-Type: application/pkcs7-signature; name="smime.p7s"

MIIFhQYJKoZIhvcNAQcCoIIFdJCCBXICAQEExDTALBgIghkgBZQMEAgEwCwYJKoZI  
hvcNAQcBoIIDcJCCA24wggJWoAMCAQICFGeCtFlzUkvB9HFHGWrw/RGKqkwLMA0G  
CSqGSIb3DQEBDQUAMC0xKzApBgNVBAMTIlNhXBsZSBMQU1QUyBDZXJ0aWZpY2F0  
ZSBBDXRob3JpdHkwIBcNMtKxMTIwMDY1NDE4WhgPMjA1MjA5MjcwNjU0MThaMBkx  
FzAVBgNVBAMTDkFsaWNlIExvdmVsYWNLMTIIBIjANBgkqhkiG9w0BAQEFAAOCQA8A  
MIIBCgKCAQEAw+6t+WXRTiQM8yRjWQ2fbFewCodIZUX6BY02TeZuEXoEAGEsmo0N  
6LlotcUTdGr39FE2K8IytOKkXVexswgAqBCqv8YjVDrI3yV82wrm5Td32TDlw7IS  
igak4ZSu+UowPQs8Y03oxqImP4onZNHvdZ3it9EggmgUyZX0dmQ6z509yDzHpLMA  
E2rXxfYcPXQwPvx4tcqbTf2htEP7PYnBa8a+sts0F7I7kD5ozGYI9dGg/XGs1LYE  
WAoH5YZgNFdbkJdcKG2FPAwFcVZ/hoGm6soxkDKMrYScTbP+fqH8MV11DP821Po0  
vtSEnaF8UURbaths2yKpAB2WUJvgW5xa4QIDAQABo4GXMIGUMAwGA1UdEwEB/wQC  
MAAwHgYDVR0RBBCwFYETyWxpY2VAc21pbWUuZXhhbXBsZTATBgNVHSUEDDAKBggr  
BgEFBQcDBDAPBgNVHQ8BAf8EBQMDB6AAMB0GA1UdDgQWBBSsLlRapP1VGK8u6GZE  
ONEl0dcAeTAFBgNVHSMEGDAWgBS3UK1zwIg9ssN6WgzZlPf3gKJ32zANBgkqhkiG  
9w0BAQ0FAAOCQAQEAe+qOGM+8q1UhXKV6i63BrXS0Kvd2iglxAggszUC6eMnrIem6  
6mmRzSbcGHCeU6m1MpvYSe9IiR0IxjTfsgGUdZbbXtBxSmCASj0BCbphvvtoam1G  
i8+LZd0gR2kDwr//TYjW06vUfXPwerNWMx4cKpFobdmvgLYCeAZKRvoPjJmTEffw  
K00cCxSifTpTFiwZhFxXKSCtdB6T2rE9JxJfzJqLUrvvEZwpQIt8hX8kym/vKw+1  
cbsl3rag2enVP/f4qg/0mUuzkCI8sLXd+N5gAs9wdUZRCtB0gOnUAH9m7RrpqkdC  
ogKdypGEQHj6GiamJAe2WndOp4BZdBtBRzjfuzGCAdkwggHVAgEBMEUwLTERMcKqG  
A1UEAxMiU2FtcGxliExBTvBTIENlcnRpZmljYXRlIEF1dGhvcml0eQIUZ4K0WXNS  
S8H0cUcZavD9EYqqTAswCwYJYIZIAWUDBAIBoGkwGAYJKoZIhvcNAQkDMQsGCSqG  
SIb3DQEHAQACBgkqhkiG9w0BCQUxDxcNMtKxMTI3MDAwMzAwWjAvBgkqhkiG9w0B  
CQQxIgQgGeoQw8WdmjB606EKGR5n1oMuV7Te1Vjfa2oB2ebW390wDQYJKoZIhvcN  
AQEBBQAEggEABblYEWsnYyzL3jTS3AoPr93YKksIZr5q/b8Y5/1rMxdYxPm+iRe0  
RHRgpbfQeiqZXzRXtMohfoIkh7RmdQoSV40pwiUmNU+f0ZEAu8cMVJM6gdyUD+1D  
JwDnr+YNLV/1UUGhqx0FEEx0a/4092KYBD4eRQw4KDWrkfh9dLSj0Bs14thrZYGZ

```
e7ut3FN5TBruZfmqMy50xZ9yUW91YyQUBLiIcuF185y5ZW/aQCxBKBbrNNGXLJbo
8yKFJqSPiWZvwUmVQvfgL182hg8230JTtP4VImcUakTF0+k+BM//qqKXYr\X/tZn
QzG+4ZH/XM1vgHL7ShjHS6TS0Hz20DqD6Q==
```

--179--

### 9.3. S/MIME application/pkcs7-mime SignedData Message with Protected Headers

This shows a signed-only S/MIME message using the multipart/pkcs7-mime style (see Section 3.5.2 of [\[RFC8551\]](#)). Its MIME message structure is:

```
└─ application/pkcs7-mime smime-type="signed-data"
  └─ (unwraps to)
    └─ text/plain ← Cryptographic Payload
```

Note that if this message had been generated without Protected Headers, then an attacker with access to it could modify the Subject without invalidating the signature. Such an attacker could cause Bob to think that Alice wanted to cancel the contract with BarCorp instead of FooCorp.

```
Received: from localhost (localhost [127.0.0.1]); Tue, 26 Nov 2019
 20:06:17 -0400 (UTC-04:00)
Content-Transfer-Encoding: base64
Content-Type: application/pkcs7-mime; name="smime.p7m";
  smime-type="signed-data"
MIME-Version: 1.0
From: Alice Lovelace <alice@smime.example>
To: Bob Babbage <bob@smime.example>
Date: Tue, 26 Nov 2019 20:06:00 -0400
Subject: The FooCorp contract
Message-ID: <smime-onepart-signed@protected-headers.example>
```

MIIHhQYJJKoZIhvcNAQcCoIIHdjCCB3ICAQExDTALBglghkgBZQMEAgEwggIJBGkq  
 hkiG9w0BBWGgggH6BIIIB9KNvbNrlbnQtVHlwZTogdGV4dC9wbGFbjsgY2hhcnNl  
 dD0idXMtYXNjaWkiOyBwcm90ZWNOZWQtaGVhZGVYyc30idjEiDQpGcm9t0iBBbGlj  
 ZSBMb3ZlbgFjZSA8YWxpY2Y2Vac2lpbWUuZXhhbXBsZT4NC1RvOiBCb2IgQmFiYmFn  
 ZSA8Ym9iQHNtaW11LmV4YW1wbGU+DQpEYXRlOiBUDWUsIDI2IE5vdiAyMDE5IDIw  
 OjA2ojAwIC0wNDAdWQpTdWJjZWNOc0iBUAGUgRm9vQ29ycCBjZ250cmFjdA0KTWVz  
 c2FnZS1JRDOGPhNTaW11LmV4ZW9uZXh0c0tqc2lbnmVkQHByb3RlY3RlZC10ZWFKZXJz  
 LmV4YW1wbGU+DQoNCkIjvYiwgd2UgbmVlZCB0byBjYW5jZWwgdGhpcyBjb250cmF  
 dC4NCg0KUGxlYXNlIHNOYXJ0IHROZSBuZWnlc3NhcnckgcHJvY2Vzc2VzIHRvIG1h  
 a2UgdGhhdCBoYXBwZW4gdG9kYXkuDQoNCih0aGlzIGlzIHROZSANC2lpbWUtbn25l  
 cGFydC1zaWduZWQnIG1lc3NhZ2UpDQoNC1RoYW5rcywgQWxpY2UNCi0tIA0KQWxp  
 Y2UgTG92ZWxhY2UNC1ByZXNpZGVudA0KRXhhbXBsZSBDb3JwDQqgggNyMIIDbjCC  
 AlagAwIEAgIUZ4K0WXNSS8H0cUcZavD9EYqqTAswDQYJKoZIhvcNAQENBQAwLTER  
 MCKGA1UEAxI2FtcGxIEXBVTIENlcncRpZmJjYXRlIEF1dGhvcml0eTAqGAFW0x  
 OTExMjAwNjU0MThtaG8yMDYyMDkyNzA0NTQx0FowGTEXBBUGA1UEAxMQ0WxpY2Ug  
 TG92ZWxhY2UwggEiMA0GCSqSISb3DQEBAQUAA4IBDwAwggEKAoIBAQQDD7q35ZdG2  
 JAzzJGNZDZ9sV7AKh0hLrfoFjTZN5m4RegQAYSyag43ouWilxRN0avf0UTYrwjK0  
 4qRdV7GzCACoEKq/xiNUOsjfJXzbCubln3fZM0XDshKKBqThlK75SjA9CzXg7ejG  
 oiY/iidk0e91neK30SCCABTJlFR2ZDrPk73IPMeksx0Tatff9hw9dDA+/HilyptN  
 /aG0Q/s9icFrxr6y2zQXsjUQpMjMZgj10aD9cazWVGryCgflhmA0V1uQl1wobYU8  
 DAVxVn+GgabqyjGQMoythIK0Gn5+ofwxXXUM/zbu+g6+1ISdoXxRRftq2GzbIqka  
 HZZQm+BgabFrhAGMBAAGjCzwgZUdYDVR0TAQH/BAIwADAeBgNVHREFFzAvDwRnh  
 bZlJZUBzbWltZS5lGfGtcGx1bMBGA1UdYJQMMAA0GCCsGAQuFBwMEMA8GA1UdRwEB  
 /wQFAwMHoAAAwHQYDVR00BBYEFKwuVfQk/VUYry7oZkQ40SXR1wB5MB8GA1UdIwQY  
 MBaAFLdSTXPAiD2yw3paDP0U9/eAonfbMA0GCSqSISb3DQEBDQUAA4IBAQB76o4Y  
 z7yrVSFcpXqLrcGtdI4q93aKCXECCcZnQLp4yesh6brqaZHNJtwYcJ5TqbUym9hJ  
 70iJE4jGNN+yAZR1ltte0HFKYIBKM4EJumG++2hqbUaLz4t106BHaQPCv/9NiNY7  
 q9R9c/B6s1YzHhwqkWh2a+AtgJ4BkpG+g+MmZMQV/Ao7RwLFKJ90LMWLBMEXFcp  
 IJN0HpPasT0nEL/MmotSu+8RnClAi3yFfyTKb+8rD7VxuyXetqDZ6dU/9/iqD/SZ  
 S70QIjywtD343mACz3B1RlFxmHSA6dQA42btGumqR0KIap3KKYRAePoaJqYkBT7A  
 d06ngF2LGOGFHON+7MYIb2TCCADUCAQEWRTAtMsswkQYDVVQDEYJTY1wbWUgTEF  
 NUFmG02VydGlmawNhdGUGUgYXV0aG9yYXRS4ARngRZc1JLwRfRxxLq8P0RiqpMCzAL  
 BglghkgBZQMEAgGgaTAYBgkqhkiG9w0BCQMxCwYJKoZIhvcNAQcBMBwGCSqSISb3  
 DQEJBTEPFw0xOTExMjAwNjU0MThtaG8yMDYyMDkyNzA0NTQx0FowGTEXBBUGA1UE  
 AxMQ0WxpY2UwggEiMA0GCSqSISb3DQEBDQUAA4IBAQB76o4Yz7yrVSFcpXqLrcGtd  
 I4q93aKCXECCcZnQLp4yesh6brqaZHNJtwYcJ5TqbUym9hJ70iJE4jGNN+yAZR1lt  
 te0HFKYIBKM4EJumG++2hqbUaLz4t106BHaQPCv/9NiNY7q9R9c/B6s1YzHhwqkWh  
 2a+AtgJ4BkpG+g+MmZMQV/Ao7RwLFKJ90LMWLBMEXFcpIJN0HpPasT0nEL/MmotSu+  
 8RnClAi3yFfyTKb+8rD7VxuyXetqDZ6dU/9/iqD/SZS70QIjywtD343mACz3B1RlF  
 xmHSA6dQA42btGumqR0KIap3KKYRAePoaJqYkBT7Ad06ngF2LGOGFHON+7MYIb2T  
 CCADUCAQEWRTAtMsswkQYDVVQDEYJTY1wbWUgTEFNUFmG02VydGlmawNhdGUGUgY  
 XV0aG9yYXRS4ARngRZc1JLwRfRxxLq8P0RiqpMCzALBglghkgBZQMEAgGgaTAYBgkq  
 hkiG9w0BAQEFAASCAQAcryWkSiBgrrc/aDF1Z4KRnoRpr+fOutQSLV7k0Tgeztx/kJCIiuLvjUxLr  
 TuxlyUWCKUPb6TKLYASPjpwDXrNzqmGs1pJmWHTZwUhbFVxt16FaQZkDSATtvhQU39Rsot2j1pP/UV  
 J7+5FPQwNc4dt7MFw7ju4TBHo2VrzjZ2K8ioELPxsix0CAp3ytkhf1Umw6bC5M/u  
 owj3a6xzAl4fw5+pxZw0JdbrYn5kmPieKssYy2/+y0wzrtIyTHW5dY7DoWwXDXtD  
 cmCGHk08bry+MnMy3PwvXiX0warQo1fnhXB5tlk2K9YdiDc0tnAshEBXAudnxlPK  
 JGzeJVUfbbM0

Unwrapping the PKCS7 SignedData yields the following internal message:

```
Content-Type: text/plain; charset="us-ascii"; protected-headers="v1"
From: Alice Lovelace <alice@smime.example>
To: Bob Babbage <bob@smime.example>
Date: Tue, 26 Nov 2019 20:06:00 -0400
Subject: The FooCorp contract
Message-ID: <smime-onepart-signed@protected-headers.example>
```

Bob, we need to cancel this contract.

Please start the necessary processes to make that happen today.

(this is the 'smime-onepart-signed' message)

Thanks, Alice

--

Alice Lovelace  
President  
Example Corp

## 9.4. Signed and Encrypted PGP/MIME Message with Protected Headers

This shows a simple encrypted PGP/MIME message with protected headers. The encryption also contains a signature in the OpenPGP Message structure. Its MIME message structure is:

```
└─ multipart/encrypted
   └─ application/pgp-encrypted
      └─ application/octet-stream
         └─ (decrypts to)
            └─ text/plain ← Cryptographic Payload
```

The Subject: header is successfully obscured.

Note that if this message had been generated without Protected Headers, then an attacker with access to it could have read the Subject. Such an attacker would know details about Alice and Bob's business that they wanted to keep confidential.

The protected headers also protect the authenticity of subject line as well.

The session key for this message's Cryptographic Layer is an AES-256 key with value 8df4b2d27d5637138ac6de46415661be0bd01ed12ecf8c1db22a33cf3ede82f2 (in hex).

If Bob's MUA is capable of interpreting these protected headers, it should render the Subject: of this message as BarCorp contract signed, let's go!.



Received: from localhost (localhost [127.0.0.1]); Mon, 21 Oct 2019  
07:09:28 -0700 (UTC-07:00)  
MIME-Version: 1.0  
Content-Type: multipart/encrypted; boundary="ca4";  
protocol="application/pgp-encrypted"  
From: Alice Lovelace <alice@openpgp.example>  
To: Bob Babbage <bob@openpgp.example>  
Date: Mon, 21 Oct 2019 07:09:00 -0700  
Message-ID: <pgpmime-sign+enc@protected-headers.example>  
Subject: ...

--ca4  
content-type: application/pgp-encrypted

Version: 1

--ca4  
content-type: application/octet-stream

-----BEGIN PGP MESSAGE-----

wV4DR2b2udXyHrYSAQdAH1KRyK7qZzNpI7TVprCPo/a0TW9R5hBKcTkKES1Fo3Yw  
mtDplfGFN2JMzQ10Vbe2gbcyhrYfs+7Fd4eoZ0geE2cUYn5M951I0se1W+MdMZ/j  
wcDMA3wvqk35PDeyAQv/ePyXTBTU98wzM5LcwhWZcCmxCTtgqHmjJmymQKQqJuCA  
flrZPG6V6RyidGwmJYf2uDdmlhAHxFbYAalkI+/V3Sn050SejKvspUtuRnB0W8Ps  
luWQ6ANww/o4y/2/SkIodRmwaIBbs/4CaDQivSeBueHnPu0EqxTBNI47dQx9mkdB  
Z5PscuUVSsq2SmdIrCM9aLyOuf60NVhdp3mYQaVH12dX19wjZtcLTR74t66I/Wsc  
FH0NiGii/ioJS9LGLlnaRiS7carLbtw0s2yJJZPZeRozMPi0o8zgne77wdoF+NyU  
LkGtqXvLbPPA9SDGTHgkJ6H+wUhh00GWebYwpN3F6R7Su10LYRkQ8kok0mJmZokg  
qhDueENW2RsZIg06sydGFaRY5BoGe2EBkcXUVBWqYEMH3Zxz/kAEylVY5sZ0qcae  
PALvTF6Y4nNVGVylUvcuJ4DsQbi2AueD7TL28ha1xJTkzLHlt4UyU878eUfdVL0M  
FF+hwbxlo6RBT4uurMee0sHrAUDHma9Kx6XrALINbIL5lfMKKXnKhfQYpfbYbz8J  
jVFz0zCxMqmdHZLe/G9mxoksvXrbFf8b5DHfDYGCRvbj+CzERo6KCceaVSpKVGL8  
xiwHrjg+vwfn9EG9j+vp3jB39wES/IZZThSnf0JvJA4ePVnfbxcxMqgg/S2isyHf  
NAP89ZlX5mznom9efKUoojodNNFsmIt+YNaHEtnjZl+BXstGkXX0iurEt5HuEyRz  
+cyjwpmQChz6PuY0Ehsj42mMyGa3167H2kIqtKtxIfL5/qm1df1mLEc7SpmU+uHV  
58D22bl/Ukr8vmFu09z7V2U7zXz+FtohuVpeTr3l0UVEFEIGIT4JUqxiaVZqMsZE  
6DKj6X+fzXdxMyrDd/LD2ikZdlqTuvsuuiFW10tEbuIKRoYU16u8t44/KYoHCQK  
BWxhyh7LPpf0Gkema3KY0D7yG4caTWmN5GSskGyKqQjiCxa0jKqTlqfNBtxBh4/6  
8Ijf/cmlSNjC6ghzuwtNG7wr0mSC0pjQsl7b16Im7F0mP67pputqcFrZ0IzVbrS8  
vVe0+1X3/5VnmYHCilaI4l1n3wGRTLc/j4LIoGNGLJJ9Le0z0DlfIwfIy9aVUDXo  
48awW8hYu4Ck42GIJQP9HsQ9fbFzHmyUHHs4h+XGHTbPFqiPyzsoAT8KDTLMj4y  
CKWaqmqXMkuaD7hMc42xW8ziq2ZXZCv1ajDclbkg5rx9R6n4dZL6Cajt7wK2mMHt  
giNkCqLU2LuPhw/R9comDDJPFmb6WB/PBrnTrUwrFy4/6du5uK09kwLIUu82UVhm  
5xHVqybxIkHGeVNXqRSe3M3w8ERbkXqNp3s7BrGGb1bYdlrPf8h1PTeWi9vfXUdn  
wFHR0g3xjeQ9orvJZl5jPuk5NryF2J/iNEh7+sE=  
=NT2A

-----END PGP MESSAGE-----

--ca4--

Unwrapping the Cryptographic Layer yields the following content:

```
From: Alice Lovelace <alice@openpgp.example>
To: Bob Babbage <bob@openpgp.example>
Date: Mon, 21 Oct 2019 07:09:00 -0700
Subject: BarCorp contract signed, let's go!
Content-Type: text/plain; charset="us-ascii"; protected-headers="v1"
Message-ID: <pgpmime-sign+enc@protected-headers.example>
```

Hi Bob!

I just signed the contract with BarCorp and they've set us up with an account on their system for testing.

The account information is:

```
    Site: https://barcorp.example/
    Username: examplecorptest
    Password: correct-horse-battery-staple
```

Please get the account set up and apply the test harness.

Let me know when you've got some results.

(this is the 'pgpmime-sign+enc' message)

Thanks, Alice

```
--
Alice Lovelace
President
Example Corp
```

## 9.5. Signed and Encrypted S/MIME Message with Protected Headers

This shows a simple signed and encrypted S/MIME message with protected headers. Its MIME message structure is:

```
└─ application/pkcs7-mime smime-type="enveloped-data"
   └─ (decrypts to)
      └─ application/pkcs7-mime smime-type="signed-data"
         └─ (unwraps to)
            └─ text/plain ← Cryptographic Payload
```

The Subject: header is successfully obscured.

Note that if this message had been generated without Protected Headers, then an attacker with access to it could have read the Subject. Such an attacker would know details about Alice and Bob's business that they wanted to keep confidential.

The protected headers also protect the authenticity of subject line as well.

The session key for this message's Cryptographic Layer is an AES-256 key with value 12e2551896f77e24ce080153cda27ddd789d399bdd87757e65655d956f5f0b7 (in hex).

If Bob's MUA is capable of interpreting these protected headers, it should render the Subject: of this message as BarCorp contract signed, let's go!.

Received: from localhost (localhost [127.0.0.1]); Wed, 27 Nov 2019  
01:15:28 -0700 (UTC-07:00)  
MIME-Version: 1.0  
Content-Transfer-Encoding: base64  
Content-Type: application/pkcs7-mime; name="smime.p7m";  
smime-type="enveloped-data"  
From: Alice Lovelace <alice@smime.example>  
To: Bob Babbage <bob@smime.example>  
Date: Wed, 27 Nov 2019 01:15:00 -0700  
Message-ID: <smime-sign+enc@protected-headers.example>  
Subject: ...

MIIPVQYJKoZIhvcNAQcDoIIPRjCCD0ICAQAxggLCMIIBXQIBADBQFMC0xKzApBgNV  
BAMTILNhbXBsZSBMQU1QUyBDZXJ0awZpY2F0ZSBBdXRob3JpdHkCFCJT7jBtAgsf  
As3lycE+0t95phvCMA0GCSqGSIb3DQEBAQUABIIBAKswTLBs+STeesZIYAf7Gqsj  
Za0rdUeDTSxt8RCa010EHb2lqKzHRwwPJkCLm6Glb09nYnQiFrEl6jbWTG3hMRD  
0St9kyqgeg+MxXr2g4LoXAT+8hg/qBoF//tX+bzxhx0gx8wjxBc3bvp4esCJro7Aq  
tx56BtVsI06TA0NT0Ca0cnMhIo09raR6JQX+DoPynKeXihny6TFDP7eopCgorCfR  
o5903ZMvau6Q9KixZy3Yae8fa0ZdJu3FahIZTPdBHzbmirLxcYgp+cbTpW+Yno2  
X5GJ8eq8Y0qcc/8r6Xd3REarUx02Yb02D6cgDj+aNnnsoG1/9psaYl8W1MSc2/Qw  
ggFdAgEAMEUwLTERMCkGAlUEAxMiU2FtcGxliExBTvBTIENlcnRpZmljYXRlIEF1  
dGhvcml0eQIUZ4K0WXNSS8H0cUcZavD9EYqqTAswDQYJKoZIhvcNAQEBBQAEggEA  
RHhTarDqNLzXSaBokp2L3EwDv1lKiGtMSMUQuPelNoC2nNYU1yzAF4jd+1UUo4Uu  
quiHg5Hn44a9MejrVmQRLd5IEJiZGD8m5Jguu0jn0ooyA6EEWUpMn6h0AKlaCiXd  
kwTivKfhQFJe9Eb6TKqtvt2IEu3kXFfJKi+VyQw49+RXBmajDKJoHtumMJs8k4Ll  
kJah+wD+snwHg2LCiJeSVHmpf4RvSiIJSvk206IeTxN3JecNbBpKLtIoy/CjWEZv  
G3Pj/zkBbb+XhHbXo+Zk/e3aLT0VG/cldx6Ti8zAr0YNAzgt1G7dmJ3mnNPitEwN  
04qIozhT2Qn8P95AEV5PsDCCDHUGCSqGSIb3DQEHAUTAUBggqhkiG9w0DBwQIUzdf  
vwulBs+AggXQMK121v6l07W1r96RW0rs0HzsIvGyfyRTT1UuZRxVL09BQZstI5ss  
5Zv8BogoKA0mLaNBKM755joUzbF5f/jMYhkW3q0Het9/HRH0mOnCSnoT4i2yzNdi  
0tj8ixPT4sgPe9F0TKke9CzoJ967kj9D8u7Ik2goojttt3ViJkv3a1qrWDMiJRIJ  
g0TTA6ZaQep5L92vtCobhD+i7iaktEpmbYucXs8jjMmwyxCfXHXGD/fwDk3UDgeu  
8a5f66YepZdbLKB61A3rBwJMvQubuXEIEb04tG0Fgwx3Ao2NshN+XRk/y+uhQKdC  
5ZduTxk5sokA+H4nzVv0IUKAAI+8FwY5ZWFGlnCKUM/wvrGHQq3R/utChFau0HxD  
7vZQLM91TcQzVWdHfJGPTp+ekjRlu9UqatQgclog0bw3PGYLJc90GL7AZHAsYncU  
jsMbdswuFuYNHJ8Lr5VMO6L4bCNMy+tQB0fYTF1el+i9S3r3SWdBp+uLiKgDQ52  
/o4shxoi+Y0f9k8wRR0iDKQwzcJuABlpG9A9qjsQNqBKF5t5p3l3ihH1mfh8FaPl  
ab0aDC7uunY5g44qXcG9YS+j5wUFuxgYyGkVcJq3xiit9YbEy8uPxJFz4g0vNC+r  
uUSszbtLyHkhv7vnCTAlmjg9eDpW/tEC/85pL0V1HUooD05eRfkjU+1XsccX8DG  
iCax2C6W3cc1SC/d3a1+270cgvPdDcb7zuL3v6qqqbN+7GDrCQHQRfMD2vd6+XGk  
NWZQMBZVHmdCCKGL9YaH0RgkGH5beTRKEV1wBafuV0wTEwL/FuZzD4oHr0aP3GL0  
cLxi44her/hNxtxDc2Lw0VQcxD8A550kCt9+u9M5/YPj41FwyH6kdh86p958gzF5  
EpwCnQDe+s70rwFVW00DEJhqtEcRCSSW8dS4hVEhVxQJ56liJP+VZ+LTUJBel4  
mfSpSqxeJnmyY0nmhEbZKVbK95a1WYMJCEpk2nlg/bQGqJKRryGwbEF9WqqHuvPo  
Bv/BfinoUL3Kd3g+hgSCR4mCg5EhEsCx21jEqEggzb2XMcA+knGUYxSwj322pZfW  
LDh50gkL3GQSm9f0vjDk40GwZv8HUdLXuAQ/J19PaFmaDkd4jzRi37VBqdDgLY3  
u6K+oFKhG4oqQYa/er+ZGAqqlDtmu8HGCsjm6kGZvSAocJg0UnLPBNI0/iB0BYGf  
KJk302jy8kfAXGSiWryDNbTuDzFMD0zsbHbM07A00R0GwKv5TxAf1EHHTxGb3IKI  
jRkVBL7QdRtDH03zlxv0lnFwiuCrzLrQdUuEG/0wt8RaNr+p8hAo0YEGbB9jmbax  
CSLLWeNbM0o8eIi3Mft4qmDXp3TEuHHru8kbvA36vQ8+dunSf2BcecyM6UAYBqaw  
SCcxQmEcyMuyjSLVerVfML5lwlmM+qabxHq0hpJHnCR3Vl2qX3CiRwpVlNaBVyTf  
793bAm7DU7G+Tzt5gdgE4s41aZt8fFXyclhH1QLPNSnctxJjuW1gJJ0h51iCQJp2  
TgzDw35oqvBxbN3yqCFjScsQXPXYErGwKlRAkUurff4x/ZAizFkmjdpyaIK9JBw  
QRyrYYQ8pJhXJe9BrP30S6evFlsWZW1MaoQc0UMWsuVucE0e4AQRGLPixDjJWW7L  
I6AQ3KUW6ggzDJksaYHDiuEoBa7vcYoTar+/AhNjYMjkQX/3kptQryqy+xke0t80  
EPQER0Wur2IpvM6YsvI/SoeFwxMb4Zm5AFvvibiCCmmoJc4A9E1tZ/sMstHyZ5iu  
tJqu1M5B0DIoFdB5pzbZYCKgN2n7EY23JS7E/oz0rzYu0IVUJVtB5awqmuSLmI+N  
R91g4FMEfLYC1HYKYlaknX2zmrX8+Z8MEJNM2K0q8wPBnm860pGeJmlZhFwT2x0R  
eJpKcflGroXYh2Gb6BxwIfKj00TXCoIFP02JbTJ7clC/2ei0BN6JxywPkH4renaP  
SkuNBgbexfZGBhMTlR+CtKLEUmw5bxBDWjjcvzWDPhy/VurLQxh0qYnbhZW21SV  
4qMrJ4uGXehylNp0FD+HR4mB2epYcw3dFj4cGN3B2Y5Nn0Tw0Z7fi4S0BPdvYjP9  
LL5WZ6p90mII9wcunGCRnLUUYumRnIbhVHIbTTIRI5PUSVFfEuotrDZ9oZcwYk07

fQX21gJCzvJyp8ft01HX4Kc4mN/FMPgGcmq70N335yQ4mQ/eSvTNn7E+35ZGn9f8  
PI7QPJRhdUkBZCnwyv+0wK2VzySxnqNfPaZk168foGRd9eFCw80L4U+SuLDQH6ZT  
o++VKk4Ce2jx1khoig16wic0dVFwt4bmybNz4u/qdobYr5fs7dKPHH002SBvAl60  
16foheiBtV2VA8mEBA1BhcNmKYegu+RGhmGfNDuZB8XdbPQ6M+N+ilEj/6rr+wgD  
gcmEyAGNwJkmWpbyrm9M4lDtzemv5N5V32ppGizEt6c0xlkiULllwGdWey3+YRez  
7b+Kl/uIpDuRbp5Tf43dyPsy/cx4DNm5kAB4CcyyVLXPaqXm0lLEPYBmaMW30+D2  
5v4Wj1qwIR05qgI8FyVnX6sm/oucfcg5l172edaCG8f42gIMNfQBgWVMsSG7Nt00x  
dJo/OGtACwnY47ohMFG0BejWueAksdnqVWCIt0989iBHgegNx5jUCycB/Y0m0xh0  
pfeNjA9PwZMUpjlqrjDFIan/UFYAZH5ISSV7G30oRKJ3TTEshShXP2K3cn7Fa9W+  
H/jyTEQGfCiTq7Xx5Fr0IJBmKjylkF7oGlibXJgKKRm0iD/sGNTaSJ6Pl8/K6dEz  
zsMwEFTawnWVq32Xn3d6/+FADZ9lGhC5WwVgaQHRb/9Ejt1mBdptmXjEj5w0Y0ib  
xFer54LrQgvBWEYRqDneh3bI53BudbTl7YitqULVGETe+k1T0NbcyElrr2Y/NKHK  
rPMarAfByookkJrDtVh3VrAm2ows70wvKGyoNybjlyczjt7xosatZlXkbg9mtR5i  
E2l9ajSR4SzQjHoboRy0CwL5ZgLV/+yp3jTkNcUkFDRtkVbGfascBIME0ifUGfvP  
mJ9AQHZxdfm99KlQjCZzR8CBuVR+zsT43jr91CQKSSEvPML6vVRV2thiWw3VGgP+  
c8i5zj6+zCnLEdSwiIeFw0J9/ewKSdU9pGrA00QtXbYQlDCKuGKlVgy6jJCeglDH  
T6gVny5ip593wWwF0VxVEWUygi6JCdS27b5+P/wlNjTrzpZ4yWDCpyogyrTl9f1/  
GgvdGuWWinKSL0yh1fJlP9WoDWcqH980hJXLV+X30C+tmMofytmHgXN8jjVsWSRa  
VWrFUarMs2hZDWf6e6ncwvMC8QliisZrKXQNckxvBuh5hug9WKurVj4CIWnoqXFh  
Oql0+VbqZSj+TT5pCN//370vsIZIn5UbrpDmUP0rUvdTGz9iWQRUL6R2g2h286s6  
pAGHv9luXCoPJ5uPTwcbBSl/js6J+K5McyqRl4fucacfVFnMuDpET/tTleAR0P3F  
DOBKqV5Y000rWMexzMLJUEQ/eGSwfp7wv8on7jeGxAexMqyWCrhRk9G2ZwiT4L7Q  
rX4NIDj6oujCCkeFUATs0pGKwEFGmpbEUfD0sioWoVYJZPs09kAGq6bhbKAC0keZ  
v95ha/3CleYXGUUntzLsCx+c9Zp/wl+0PcT3ZSWhmRbXiIvz+ntHVe47PHxbvH6a  
ZG7YGc/9u3jTvJJyYtQ054uGET/eFWSxCUo5/Vfshe0uLdXN7JnVi6ooF+c7WUZd  
6lFwfDwNf8z0Gws3EotozrWyBgKS5VFP99vZM64nSqu9v5PSzmb0AY/Zc5KhVXVY  
zQqm03keXq92Fejtgdyd/09ITZf5GkMQVU7+IT52JxFRQplkbTHJj4HRGtGHtIyPW  
Rmf9qSZz8QgVyAUKK1k+kLBJTHN3CwIB6S9h042HWEFvLVl8wPWW5aLYTsVMGnMU  
aZ35M35odjrVY9B0INMPL53Hm7qH1w/h9QCv+xsFmanYsoylwbuKW2TcSnWB74C7  
Wy0NmCkaM+Jwe0gygffWicLGJ3jKWccykTUZtodzlectNHh24puZICnvfwjte+n  
eSQqJfHMsra6V8BcshpwmvPyLHnkU+2KyhQ84300R/qaXAYJ7EWRBEFe4EipxzfL  
zQF0LwbhpaStpcj0LjFEHmQiWx8ASzE1LMSfZo148sXYEwsJL7t5tWs=

Unwrapping the outer Cryptographic Layer of this message yields the following MIME part (with its own Cryptographic Layer):

Content-Transfer-Encoding: base64  
Content-Type: application/pkcs7-mime; name="smime.p7m";  
smime-type="signed-data"

MIIIkWYJKoZIhvcNAQcCoIIiHCCCCIACAQExDTALBgIghkgBZQMEAgEwggMXBgkqhkiG9w0BBWgggMIBIIBDZy206IEFsaWNlIExvdmVsYWNlIDxhbGljZUBzbWltZS5leGFtcGxlPg0KVG86IEJvYiBCYWJiYWdlIDxib2JAc2lpbWUuZXhxbXBsZT4NCkRhduGU6IFdlZCwgMjcgcTm92IDlwMTkgMDE6MTU6MDAgLTA3MDANCln1YmplY3Q6IEJhckNvcnAgY29udHJhY3Qgc2lnbmVklCBsZXQncyBnbyENCkNvbnRlbnQtVHlwZTogdGV4dC9wbGFpbjsGy2hhcnNldD0idXMtYXNjaWki0yBwcm90ZWNOZWQtaGVhZGVyc0idjEiDQpNZXNzYWdlLlU0e0iA8c2lpbWUtc2lnbitlbmNacHJvdGVjdGVkLWhlYWRLcnMuZXhxbXBsZT4NCg0KSGkgQm9iIQ0KDQpJIGplc3Qgc2lnbmVklHROZSbjb250cmFjdCB3aXR0IEJhckNvcnAgYW5kIHROZXkndmUgc2V0IHVzIHVwIHdpdGgNCmFuIGFjY291bnQgb24gdGhlaXIgc3lzdGVtIGZvciB0ZXN0aw5nLg0KDQpUaGUgYWNjb3VudCBpbmZvcmlhdGlvbiBpczoNCg0KICAgICAgICBTaXRl0iBodHRwczovL2JhcnNvcnAuZXhxbXBsZS8NCiAgICBVC2VybmFtZTogZXhxbXBsZWVucnB0ZXN0DQogICAgUGFzc3dvcuQ6IGNvcnJlY3QtaG9yc2UtYmF0dGVyeS1zdGFwbGUNCG0KUGxlYXNlIGdldCB0aGUgYWNjb3VudCBzZXQgdXAgYW5kIGFwcGx5IHROZSB0ZXN0IGhhcm5lc3MuDQoNCkxldCBtZSBrbm93IHdoZW4geW91J3ZlIGdvdCBzb21lIHJlc3VsdHMudQoNCih0aGlzIGlzIHROZSAnc2lpbWUtc2lnbitlbmMnIG1lc3NhZ2UpDQoNCiRoYW5rcywgQWxpY2UNCi0tIA0KQWxpY2UgTG92ZWxhY2UNCiBzZXNpZGVudA0KRXhxbXBsZSBDb3JwdQoggggNyMIIDbjCCAlagAwIBAgIUZ4K0WxNss8H0cUcZavD9EYqqTAswDQYJKoZIhvcNAQENBQAwLTERMCKGA1UEAxMiU2FtcGxleXBTVBTIENlcnRpZmljYXRlIEF1dGhvcml0eTagFw0x0TExMjAwNjU0MThaGA8yMDUyMDkyNzA2NTQx0FowGTEXMBUGA1UEAxMQ0WxpY2UgTG92ZWxhY2UwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQQDD7q35ZdG2JAzzJGNZDZ9sV7AKh0hlRfoFjTZn5m4RegQAYSyag43ouW1xRN0avf0UTYrwjK04qRdV7GzCACoEKq/xiNU0sjfJXzbCubln3fZM0XDshKKBqThLk75SjA9CzXg7ejGoiY/iidk0e91nek30SCCaBTJlfrZ2DrPk73IPMeksx0Tatff9hw9dA+/HilyptN/aG0Q/s9icFrxr6y2zQXsjuQpjmZgj10ad9cazWvGRYcflhmA0V1uQllwobYU8DAVxVn+GgabqyJGQMoythIK0Gn5+ofwxXXUM/zbu+g6+1ISdoXxRRFtq2GzbIqkAHZZQm+BbnFrhAgMBAAGjgZcwgZQwDAYDVR0TAQH/BAIwADAeBgNVHREEZAVGRNhbGljZUBzbWltZS5leGFtcGxleMBMGA1UdJQOMMAoGCCsGAQUFBwMEMA8GA1UdDwEB/wQFAwMHoAAwHQYDVR00BBYEFKwuVFqk/VUYry7oZkQ40SXR1wB5MB8GA1UdIwQYMBaAFldSTXPAiD2yw3paDPOU9/eAonfbMA0GCSqGSIB3DQEBDQUAA4IBAQB76o4Yz7yrVSFcpXqLrcGtdI4q93aKCECCcZnQLp4yesh6brqaZHNJtwYcJ5TqbUym9hJ70iJE4jGNN+yAZR1ltte0HFKYIBKM4EJumG++2hqbUalZ4tl06BHAQPCv/9NiNY7q9R9c/B6s1YzHhwqkWh2a+AtgJ4BkpG+g+MmZMQV/Ao7RwLfkJ90lMWLBmEXFcpIJN0HpPasT0nEl/MmotSu+8RnClAi3yFfyTKb+8rD7VxuyXetqDZ6dU/9/iqD/SZS70QIjywtD343mACz3B1RlFxmHSA6dQAF2btGumqR0KiAp3KkYRAePoaJqYk7Zad06ngFl0G0FHON+7MYIB2TCCAdUCAQEwRTAtMsSwKQYDVQQDEyJTYWlwbGUgTEFNUFMgQ2VydGlmawNhdGUgQXV0aG9yaXR5AHRnRZRc1JLwFRxRxlq8P0RiqpMCzALBgIghkgBZQMEAgGgaTAYBgkqhkiG9w0BQCMxwYJKoZIhvcNAQcBMBwGCSqGSIB3DQEBTEPFw0x0TExMjcw0DE1MDBaMC8GCSqGSIB3DQEBDEiBCC5A+mnkPofr5VZKP+y+n5m21txluYik0ynnkYbtCaH+jANBgkqhkiG9w0BAQEFAASCAQAgfVYYJu+aUcwjlfOT//l8p4L0BcB3WBEax7msyZcptuaJtWaLedzgi+nGHfhl/02wzTvCjx+LTHGouU83ILpEdDaxEDqzNgdgEJF7wswM7N31PhjpQyH+HbrJTH0tF+/xREgCG14yRs5yAX0kvkFDmd55svukInxeSb97LhQHqGpJLH5FBstWwBKQitNn8eB3g6h+c43zp4nBXoS2aFiUvYdWugw4QHW7T7dcSX5gAEHt/dm2q4oH0g9YtHmRp0mqdNQSUmK7vomEk0kv2XWmlf3znKWe8QPdlihgrh0ASyT1oBmnpEVwvsSkhqoxkGcrrSefUZy5h0wKfNSqRW

Unwrapping the inner Cryptographic Layer yields the Cryptographic Payload:

```
From: Alice Lovelace <alice@smime.example>
To: Bob Babbage <bob@smime.example>
Date: Wed, 27 Nov 2019 01:15:00 -0700
Subject: BarCorp contract signed, let's go!
Content-Type: text/plain; charset="us-ascii"; protected-headers="v1"
Message-ID: <smime-sign+enc@protected-headers.example>
```

Hi Bob!

I just signed the contract with BarCorp and they've set us up with an account on their system for testing.

The account information is:

```
    Site: https://barcorp.example/
    Username: examplecorptest
    Password: correct-horse-battery-staple
```

Please get the account set up and apply the test harness.

Let me know when you've got some results.

(this is the 'smime-sign+enc' message)

Thanks, Alice

```
--
Alice Lovelace
President
Example Corp
```

## 9.6. Signed and Encrypted PGP/MIME Message with Protected Headers and Legacy Display Part

If Alice's MUA wasn't sure whether Bob's MUA would know to render the obscured Subject: header correctly, it might include a legacy display part in the cryptographic payload.

This PGP/MIME message is structured in the following way:

```
├─ multipart/encrypted
│   └─ application/pgp-encrypted
│       └─ application/octet-stream
│           └─ (decrypts to)
│               └─ multipart/mixed ← Cryptographic Payload
│                   └─ text/plain ← Legacy Display Part
│                       └─ text/plain
```

The example below shows the same message as [Section 9.4](#).

If Bob's MUA is capable of handling protected headers, the two messages should render in the same way as the message in [Section 9.4](#), because it will know to omit the Legacy Display part as documented in [Section 5.2](#).

But if Bob's MUA is capable of decryption but is unaware of protected headers, it will likely render the Legacy Display part for him so that he can at least see the originally-intended Subject: line.

For this message, the session key is an AES-256 key with value 95a71b0e344cce43a4dd52c5fd01deec5118290bfd0792a8a733c653a12d223e (in hex).

Received: from localhost (localhost [127.0.0.1]); Mon, 21 Oct 2019  
07:18:28 -0700 (UTC-07:00)  
MIME-Version: 1.0  
Content-Type: multipart/encrypted; boundary="924";  
protocol="application/pgp-encrypted"  
From: Alice Lovelace <alice@openpgp.example>  
To: Bob Babbage <bob@openpgp.example>  
Date: Mon, 21 Oct 2019 07:18:00 -0700  
Message-ID: <pgpmime-sign+enc+legacy-disp@protected-headers.example>  
Subject: ...

--924  
content-type: application/pgp-encrypted

Version: 1

--924  
content-type: application/octet-stream

-----BEGIN PGP MESSAGE-----

wV4DR2b2udXyHrYSAQdAXX1u0LNgj2o6biKu64RULx3PY/gcetRoyN0WNoXG8zow  
LF4DhnBs27vQkh1BIU4KmJF0wwjLwuRvS/J4NvCqqcEYwiPdhp5q5ftn7wrq2W5s  
wcDMA3wvqk35PDeyAQv+M8gxGXm9ecpcotEX+90M9EY5N8V7FmZ6ydRpBXgWvCpB  
Nr6qk90s0vIlhiN1IJB173mEb5LdMj3wtRwGP3DB4AoPabIMXh/hCcNAhaWusVH0  
AK33oDjH3rhntORMve0qq4QhRzUGR1ctYWRNBXgKC/n3Bmp7mHAzfb4RyBGXDXsI  
TCXAb2qDnk06vTCVaHJ/ggBInSb12iYPkhDtoxNF0P7U97LSVgSoDe1s6TRDfpb  
9K667gVyhkTnBvys+EqWbe7Bz5MJqxn9NQxh7HTdY2kXSKGGe1DUrAzLKRpT78fQ  
002DLHR9EUh30hYQEPnuKAdYHJquXB5Ui0bJpQ5UDEt3Msv0bUD7k21MQk5K6iyh  
1wcxtXm/kPqQ3e0pVm8iaRve/VrpZEgA0/9PcvQJ0VCWQ/fZEBVmH3ojIoZF9WJE  
jB3FwPS21VLJhaZFTGU7xOKSz/x0K2M8meAsa7nx0TaetmieRA2L+wBaHhoUz77L  
9ihYLIBPNvkb49jnF3ft0sI2AYM9DWi3Ki7uWnw/Ue7jiu8dseBTvuxXU7XYPS+l  
k3nqqtkCjDziq+ojjw3+ahsfNNIrcFTizjZqGG5AK+dwjiTY3T4fJ4b07513+2uj  
/tJE7p6IuuxlE+qlpI1PrX7JFHpihbxsWnwT2RBgo+sdeVko3HbyWtfLnfwI+eNo  
njB1DvhWg4C61ilnbRU+osbnZSoSqJSdHCHqn06YfL75sdHrhDiXzV5+LPiaqHoD  
S1w0LknIFD91G03PXaae3ENJgE9CFz4v0jNw2+kASuH80DwnKiM0rmG78rY4u652  
Hc02p0ZQAX2QeK0UidsjQQaKRtZ5sys6QUBs46lgMSnHljQun4g8hlvoDH/7Zz4a  
kMgbZj7TRPU2EaApRX9JZub7nD90DJkqtLJef9ncmI3QwBjCLXylsL/olUhUjFAZ  
VNbbInqEba+LLio4HUozBAjrVW0rAt776lBSR4n72DdMjMKZ5osxPLtAVce9KeV  
s1cdKffbF4VDoe97eRq5ua4KJW/c+8WGW1u/vzPA7Zj6rR+gaWKqw4rnlys4+M2b  
LHugg+cF0k/sEfrmEuHyefYvms9Ht2icbiSTbqN+ApXuC9QtNRb/XnEw5lCH+dB0  
EYm/W0qSDXMcv0ZaZ379uFkXqiECLF11iA3K89BV1VFXgatnLHbNBdpm+mmJlz+  
MY0NTCASFv0Bri4Y7j6kS0ZMnfol+84j/nVCpBej8QrXqbpL+/6xrBURcA1Sb+Xu  
XRF1Veybr1bj1TcP7aDLzZtQ8pk+8zyxy9d0ePPcBDZlnDXCALf9eXJ/HX/6EYNT  
30h+kmF7UxghUGUnyTfBMhnBD5oNi+0GVyDwyRv5jfYc5FWwX0mcRjigPlofLmo9  
7eL0mYmmp0L2DdNiVer/Dl5g8HRSVaRceHJVUrnM+M2xzCkdrTHJSh7MBU0TwUd+  
RXYQgfPu8xbeouLnSTVC5Kuu13VA8Q1/Y6KcjQTgjNvr0zjHTxjKek5fokNxivFQj  
1fkAIM9w2k0=  
=+l7i

-----END PGP MESSAGE-----

--924--

Decrypting the Cryptographic Layer yields the following content:

```
From: Alice Lovelace <alice@openpgp.example>
To: Bob Babbage <bob@openpgp.example>
Date: Mon, 21 Oct 2019 07:18:00 -0700
Subject: BarCorp contract signed, let's go!
Content-Type: multipart/mixed; boundary="6ae"; protected-headers="v1"
Message-ID: <pgpmime-sign+enc+legacy-disp@protected-headers.example>
```

```
--6ae
content-type: text/plain; protected-headers="v1"
Content-Disposition: inline
```

Subject: BarCorp contract signed, let's go!

```
--6ae
Content-Type: text/plain; charset="us-ascii"
```

Hi Bob!

I just signed the contract with BarCorp and they've set us up with an account on their system for testing.

The account information is:

```
    Site: https://barcorp.example/
    Username: examplecorptest
    Password: correct-horse-battery-staple
```

Please get the account set up and apply the test harness.

Let me know when you've got some results.

(this is the 'pgpmime-sign+enc+legacy-disp' message)

Thanks, Alice

```
--
Alice Lovelace
President
Example Corp
```

```
--6ae--
```

## 9.7. Multilayer PGP/MIME Message with Protected Headers

Some mailers may generate signed and encrypted messages with a multilayer cryptographic envelope. We show here how such a mailer might generate the same message as [Section 9.4](#).

A typical PGP/MIME message like this has the following structure:

```
├─ multipart/encrypted
│   └─ application/pgp-encrypted
│       └─ application/octet-stream
│           └─ (decrypts to)
│               └─ multipart/signed
│                   └─ text/plain ← Cryptographic Payload
│                       └─ application/pgp-signature
```

For this message, the session key is an AES-256 key with value 5e67165ed1516333daeba32044f88fd75d4a9485a563d14705e41d31fb61a9e9 (in hex).



Received: from localhost (localhost [127.0.0.1]); Mon, 21 Oct 2019  
07:12:28 -0700 (UTC-07:00)  
MIME-Version: 1.0  
Content-Type: multipart/encrypted; boundary="024";  
protocol="application/pgp-encrypted"  
From: Alice Lovelace <alice@openpgp.example>  
To: Bob Babbage <bob@openpgp.example>  
Date: Mon, 21 Oct 2019 07:12:00 -0700  
Message-ID: <pgpmime-layered@protected-headers.example>  
Subject: ...

--024  
content-type: application/pgp-encrypted

Version: 1

--024  
content-type: application/octet-stream

-----BEGIN PGP MESSAGE-----

wV4DR2b2udXyHrYSAQdApTCCVZLqLBNWL55la9dZGb01aPtMkIFXYo8D0KgIpCcw  
gm5Vfq0ECRjoZqCwveFWGqRknz0lc+eau5fcbenmEW8J1E0FjpoBEnFo9vYb6PrU  
wcDMA3wvqk35PDeyAQwAwiumTVdntVxYn6dnGUoaga2txqCsxioqn4JgfmGrIfBf  
+BEHyt/a43rWwfi3QycCKg483Fqx0YG3HHJEiwdFmE3XdoHmTRKfHuSiyzCNxPz  
AK2cwloBtD3w6zs+m0Y7Ytq83ghyBeX0aGmgCZqGhL60In5Qu+w3Vmxcl9d2+BTs  
Z0JzxcHACRvq2tD0RRmyhjWKqVdd2akllMy1pcXLIediUiEI5MA3TaWUk/uVDsUq  
S6JtL0dEy0s49Z+f1cGfEyGCGU6TqV0Yun0bl3A7/0JjYC+75eCv89s/q4W1UM1M  
ps02X7xNlhgREncwvaoQbvfvfSlxHgWGCZDL8+0/7XC5EDyK4LAR912SG4Desr9e  
k9Fn3bH6Tt71vpH0nByKCh0m2/apFEMLXSq7DMiJEN4spbc4D3iBnxYqEH99e052  
KNjrHaoG59bZ6TNjj/JN+E5sQzDxic0004Qccg9M7iFh6eBL0uBhBpRxbeoXQkl3  
1mzI8XpyFoGu0HH0I0Cs0sJGAUnVvA0LGq7wjKpy0bWQLB2YVCKU6C8GnX6GUcLm  
SMovYhGKfpb+LUu+UM1BZ9vd9D/tsMd2WBw5tM1ncfRuST0hVeFgTEGiCrBn7sdb  
UFTV+jb5CktQMwj5vWLVPHMIUeISwoAQJ10Nu0qFnVTJ2bZ0dxZeV6NDYPYCERuR  
Sh980UxdjGLvw/LtMThKJRUR3S2TcmKSwGen5a96S+lAAmMJN5wLrH+X76UuRvV+  
07m6KDas0+fEIWXKYHGjJi10n8MnkVE4dSDKgUNukVRoBAB9Iqn11zWb6IX7f11M  
k8C+8F5Y1xxEG3CCeYdTKSiIkDvBV8oFGrFCYXW02bLWFpCZ0t2qDfWX5SvXj+EZ  
KxAiZobwQEW16WYp4Mk0Ppf0UrBXkfnLBieRg04o5j5Y//EXKpv8TSBxRbe0VfRk  
x11HNbaNeBtID4N2HfjsqUX3y2ZH3m7HWLwkQeX6Yw5qqSWQjC8fklx0ku+brAaM  
ayudhVFKiD5PVfe1NrVv5dDSbj5VyQkoESi2zLmd4SLoFIMp8/lfSnpl0ZF4krFb  
wIF8wd+zT2307fN4DRKjuqFVr0Yl8oh9iPJN0xXSyygeo+JWWfYPu41vf+viRZMh  
aj1nhJoa9UghiYfXuDu+VjzZuM22C/9gVbXMSuY1PaKffBleTNhCT7JWlmhNBW6t  
ouH6dZ2X60lXECmByzKy+d8Dun21G2nLuE82QP9y7/QZ2g+OSWZAA2IIDiH2tEib  
8CNSVwZXNpSeqH5u3+aRE1M5EzslbLU78Ryrxt6lNAzEHD42Fif+qaH0WW52wV2H  
vnaxJW0yQ1o4W6W+BPtKqtE7t8JgTetxldKHIIdWCMXg2isxWMMIE12QEc26+bQnz  
h+kDrTqxtp8rSfhLSQi4TRoudxx8mMjwFEwnRIFRQG7eGNPaqZYF3dz/neN/fy0p  
JbflgFJAtrSiL00aZ+iT8640tcaLOHk0LNGEuyJR1d0C9tuyldarvKR0v0i4jhY6  
UxDkknDkq0IzTmczFyAH3lBLRPMZNZ1z  
=YU4k

-----END PGP MESSAGE-----

--024--

Decrypting the encryption Cryptographic Layer yields the following content:

```
Content-Type: multipart/signed; boundary="80b";
  protocol="application/pgp-signature"; micalg="pgp-sha512"

--80b
From: Alice Lovelace <alice@openpgp.example>
To: Bob Babbage <bob@openpgp.example>
Date: Mon, 21 Oct 2019 07:12:00 -0700
Subject: BarCorp contract signed, let's go!
Content-Type: text/plain; charset="us-ascii"; protected-headers="v1"
Message-ID: <pgpmime-layered@protected-headers.example>
```

Hi Bob!

I just signed the contract with BarCorp and they've set us up with an account on their system for testing.

The account information is:

```
    Site: https://barcorp.example/
    Username: examplecorptest
    Password: correct-horse-battery-staple
```

Please get the account set up and apply the test harness.

Let me know when you've got some results.

(this is the 'pgpmime-layered' message)

Thanks, Alice

```
--
Alice Lovelace
President
Example Corp
```

```
--80b
content-type: application/pgp-signature
```

```
-----BEGIN PGP SIGNATURE-----
```

```
wnUEARYKAB0FAl2tvLAWIQT rhbtfozp14V6UTmPyMVUMT0fjjgAKCRDyMVUMT0fj
jjjqAPw0j0QI/Sr3vG0hiAKmfBgmB7VhKiUbfFWKRwKkzJ/kAD/e0jMNvaZ5MG1
fw6xQXpB1vRrY9Ttz3zr+TfLnHFwQM=
=4v4Q
```

```
-----END PGP SIGNATURE-----
```

```
--80b--
```

Note the placement of the Protected Headers on the Cryptographic Payload specifically, which is not the immediate child of the encryption Cryptographic Layer.

## 9.8. Multilayer PGP/MIME Message with Protected Headers and Legacy Display Part

And, a mailer that generates a multilayer cryptographic envelope might want to provide a Legacy Display part, if it is unsure of the capabilities of the recipient's MUA. We show here how such a mailer might generate the same message as [Section 9.4](#).

Such a PGP/MIME message might have the following structure:

```
└─ multipart/encrypted
   └─ application/pgp-encrypted
      └─ application/octet-stream
         └─ (decrypts to)
            └─ multipart/signed
               └─ multipart/mixed ← Cryptographic Payload
                  └─ text/plain ← Legacy Display Part
                     └─ text/plain
                        └─ application/pgp-signature
```

For this message, the session key is an AES-256 key with value  
b346a2a50fa0cf62895b74e8c0d2ad9e3ee1f02b5d564c77d879caae7a0aa70 (in hex).

Received: from localhost (localhost [127.0.0.1]); Mon, 21 Oct 2019  
07:21:28 -0700 (UTC-07:00)  
MIME-Version: 1.0  
Content-Type: multipart/encrypted; boundary="32c";  
protocol="application/pgp-encrypted"  
From: Alice Lovelace <alice@openpgp.example>  
To: Bob Babbage <bob@openpgp.example>  
Date: Mon, 21 Oct 2019 07:21:00 -0700  
Message-ID: <pgpmime-layered+legacy-disp@protected-headers.example>  
Subject: ...

--32c  
content-type: application/pgp-encrypted

Version: 1

--32c  
content-type: application/octet-stream

-----BEGIN PGP MESSAGE-----

wV4DR2b2udXyHrYSAQdAC1Ly20ZdEVNBoA4HUfvQJgdpSkeLPzYiPR/TW0apEx0w  
gPck901y4gnu01fnptzYiIaZKMWis7jPqmH2jQRhnG1Q0JKS1PeCfTS9207oQiD1  
wcDMA3wvqk35PDeyAQwAqIL7jcN2Rm5u4qhMfvT7by7nUKCOaP/H+kMPIsXP2Kxf  
MlRVnrrsCgJ6j5htt48HGddpEgLLZceK3vg8WlRWSpmstpdGxxE7HZqXHkMNk8V+  
8EVWlHGWBmxisA7/J00rt4HQJnHm0ldrIXgWjIA+Vpu/zFA542qQH78jr9Ghhp/C  
Q32V0rCY/PsFxabPIYS9wWh1Ym3+VQFndCVSpXCZHS1Qilt9XGj4X712QcvgL2Pp  
glaulvNob899d0Io4Noj7p+cx4yMkWpi9dqHu0me23aixieBbzQopzY3gleVgXhc  
HFhUzje7DybtVq0em4xnpNPWxq2b+WBeu+SvXFO2buHhWmMCLbKf6ggg0d3CRKcPt  
h5MLF3dFE1kj3B0LxJqF0Iny2EhWZvvmDQgG4uncEGo1siQhEiutQL2WClzuHGzs  
T8eEHKeATEPqRQHm395Ivr5bt08gg4tnIkfBBULPgnEfY07Llc+393a0MgW9bLbn  
UZTmNiS1FKXYzHxpUAD0sKBAe03UKSoYJ5b5yBghMZCCS9L9dm8llJVsmh022DC  
lMPpRsSm79hnFww0+Yud+i4z24C8WdivWBNoZz0M1hA5cwoQoXaxalL5GpZ/UWAd  
XNC6QwaCB2ioTFueq8SJAHzur2V89FMUuPmSaB3y072vko/468nLnjwCcZDpbWCS  
fVwcTz8bvyZfcYA2ugRPii4NM1+bYJHhtr6CIOjN0FkE5t0LAx04vPAX5CYABTm7  
HQn063YJJlTtJB1SjWmzmK5vqxtXFe0Byc/msdQX8goxS3G6RNPVHabESaqVrG4i  
F+TyzqiMFTZdLjiJXiKcFHwDoLUwA/Fxka5/BwRCM5LX3LITAvvqYy0TkaQH0SeN  
bfqCf4kWzuNhTfZM3wFgaA+FvYC8M7PKiE9y1+TiEUQMa+j0rcrf2+Nzt8mT6WU  
eQRwf9XzgmPVNarQpStomff6dJVaxloNCwKKk3LtGRWkV0EibKtFwPi+m7h3BgWn  
NQHVt1MXXV8LyKipH1ZpB3WUHjGqL13es0FwR4W+U9/qzgn6kN7kZP+yj0qXutCR  
GsjoVvwN6FU8cjv4nK1H65cobBAqP0iWEvLt1e351cwQWwUL1V/B3jWM3Wqui/hR  
lOQ9TW/WdP1/VT2Heb3503IJKJYnt0McT8aYooCLUCQmxlg4Ks1y4hP5mLlurjdV  
qBrvDNbRsW27GnyuUm8/oS1qpYS0gIrMe4BMXpwLca6xvXELNcm2Lo10qh3MhW5J  
IVjGkQDV2vM76qsFbDpHeb00XBKfccyx9wZD09M0A0XV08o/yh8H/Mcn/s0paVsv  
gdf6JElYfWC0d7J44ymzonw0kbC6F7UZgpWlY5gGlga2EPwwaFkTH22D8MH0rwKA  
JBJCvaGxEmzrV4WlaE77LUJoDs6chIF/GKcntsBvvyvjsrFLPK/2/RtrUEkP2G4e  
svWddqSECPYEFYMvzfJMwa2G0uXCLiATP8NTSle0cZ9sPKE9U162JVJ+y/t0z8z/  
oZ4SdrgAEdJSBwyev8bd1WCbRn0y0xuQHmVmhtCm4Ps506+sGWL+PDnywrwvyP7  
X1b8YpYCwaHS8md9AW2Jgcdj6p3Hc2Bs7zLMqzsc0pdivXR=

=Fb+8

-----END PGP MESSAGE-----

--32c--

Unwrapping the encryption Cryptographic Layer yields the following content:

```

Content-Type: multipart/signed; boundary="03a";
protocol="application/pgp-signature"; micalg="pgp-sha512"

--03a
From: Alice Lovelace <alice@openpgp.example>
To: Bob Babbage <bob@openpgp.example>
Date: Mon, 21 Oct 2019 07:21:00 -0700
Subject: BarCorp contract signed, let's go!
Content-Type: multipart/mixed; boundary="6ae"; protected-headers="v1"
Message-ID: <pgpmime-layered+legacy-disp@protected-headers.example>

--6ae
content-type: text/plain; protected-headers="v1"
Content-Disposition: inline

Subject: BarCorp contract signed, let's go!

--6ae
Content-Type: text/plain; charset="us-ascii"

Hi Bob!

I just signed the contract with BarCorp and they've set us up with
an account on their system for testing.

The account information is:

    Site: https://barcorp.example/
    Username: examplecorptest
    Password: correct-horse-battery-staple

Please get the account set up and apply the test harness.

Let me know when you've got some results.

(this is the 'pgpmime-layered+legacy-disp' message)

Thanks, Alice
--
Alice Lovelace
President
Example Corp

--6ae--

--03a
content-type: application/pgp-signature

-----BEGIN PGP SIGNATURE-----

wnUEARYKAB0FAl2tvswWIQTrhbtfozp14V6UTmPyMVUMT0fjjgAKCRDyMVUMT0fj
js14AQD2G0rZXkuKxZPY0l6AJFKiAFphRt+5V9gj3HEXKvQKPAD/bZy+vW9j1+e4
MLi0blojjFocLx/6MvQBoI3P9a591Qs=
=18GL
-----END PGP SIGNATURE-----

--03a--

```

## 9.9. Signed and Encrypted S/MIME Message with Protected Headers and Legacy Display

This shows the same signed and encrypted S/MIME message as [Section 9.5](#), but formulated with a Legacy Display part so that Its MIME message structure is:

```
└─ application/pkcs7-mime smime-type="enveloped-data"
  └─ (decrypts to)
    └─ application/pkcs7-mime smime-type="signed-data"
      └─ (unwraps to)
        └─ multipart/mixed ← Cryptographic Payload
          └─ text/plain ← Legacy Display Part
            └─ text/plain 445 bytes
```

The Subject : header is successfully obscured.

Note that if this message had been generated without Protected Headers, then an attacker with access to it could have read the Subject. Such an attacker would know details about Alice and Bob's business that they wanted to keep confidential.

The protected headers also protect the authenticity of subject line as well.

The session key for this message's Cryptographic Layer is an AES-256 key with value 09e8f2a19d9e97deea7d51ee7d401be8763ab0377b6f30a68206e0bed4a0baec (in hex).

If Bob's MUA is capable of interpreting these protected headers, it should render the Subject : of this message as BarCorp contract signed, let's go!.

Received: from localhost (localhost [127.0.0.1]); Wed, 27 Nov 2019  
01:24:28 -0700 (UTC-07:00)  
MIME-Version: 1.0  
Content-Transfer-Encoding: base64  
Content-Type: application/pkcs7-mime; name="smime.p7m";  
smime-type="enveloped-data"  
From: Alice Lovelace <alice@smime.example>  
To: Bob Babbage <bob@smime.example>  
Date: Wed, 27 Nov 2019 01:24:00 -0700  
Message-ID: <smime-sign+enc+legacy-disp@protected-headers.example>  
Subject: ...

MIIQjQYJKoZIhvcNAQcDoIIQfjCCEHoCAQAxggLCMIIBXQIBADBFMCOxKzApBgNV  
BAMTIhNhXBSZSBMQU1QYyBDZXJ0awZpY2F0ZSBBdXRob3JpdHkCFCJT7jBtAgsf  
As3lycE+0t95phvCMA0GCSqGSIb3DQEBAQUABIIBAFbDR6j4ZB/Mo9BQygYItwFc  
P+4r04d1ak51hc1DpSqyhiMcGahA3yxDRbZ4W1rbmC/s3d5+0WXYgs1nNMQJ48F  
f45BtNTNslPZ1+NZVbkoVJ08Bxv1rjB8/qWuSUSroqzn9enS8DUBxxPL5aSWKQQN  
G2IaH9BUKMXLPUYA46GATLy94IS4fZqwBtNNBP5eiIIPc90ggy+7At5GG7rVMN0M  
G5FL0oq52SYUe1167jp378JI+2dkAlq5+Cru/ZE2Rdw3DrMDAF05GwC7fWKg4zPm  
IHZj92caVj1IyfTmGogT2o5tLMqn61BkptqxZwHDr3FI/aYo4vcHgmlKR/TdbHww  
ggFdAgEAMEUwLTERMcKGA1UEAxMiU2FtcGxIEExBTvBTIENlcnRpZmljYXRlIEF1  
dGhvcml0eQIUZ4K0WXNSS8H0cUcZavD9EYqqTASwDQYJKoZIhvcNAQEBBQAEggEA  
hXeYVSUsT1EBZ/+AjwyEcnlM0kuFMaNVGLBMhAZzAsy012rrZTWbqWkCA3abgm/M  
CuZX7mQL0I79KZdmClGpLx6gQFjLemHaClQV0ZNdX4DxakWuME/kCMqbo4MZXSStT  
a0MHLKUdoMt72Rz4YBzNQCL7ePaii5w6Nd2KD7yJAirLYUMJEjVweVaMI9y9Lmb0  
vb0g0iuoUe0vp9B20LRcIX37nN5D1GG4tHLPjBD43gC8iqxZQf0uah2cWD1mAG5R  
oBgIDKXPy2eVbcMdSa0irDKYZ49WFe9Lad9q3mHHbFs6K6/yuBm/thMedCJkZTHo  
jiPvYdYf8IjFEd368I+DujCCDa0GCSqGSIb3DQEHATAUBggqhkiG9w0DBwQIsblA  
JX/RU9aAgg2I0VXWfs5fc/Yad2qvawUVNX+L0bjA6/+t9WxuV2em0eBYzQGjo7q+  
xaIXQwbbFlej27efGhxUYDwBNS56c0uI0Ta7jxv50FZhZQLRzoFp0bbZ+uVC4eP  
bFHarRQiPzlg900XAS00RW+U0tqN5raZ3Ry2lKwXxuStZ0pX666Rz4c8PrmMb4/B  
aQYn6iKcT6fDU2TpSbwY9iph6kZczSeewK+pIj9nXfjDKXScs8D2Raezev2ciq/V  
ZRpRH8JxieimI2yeBmEzTCq11TDYycDfMHB6reGaiCGX//8kAwtskzRyNlV61unY  
ZKSNhVKLwKmcQH1V1Nd3oLApT41EeM2oWedUqNBYqB+XGCD4DUYdm1e+4h73d4dn  
JTkCdadxEn+9RRvZ4YMLw3mvT997Dy3rTXT29dj14TstZZf2063pY0TpYy0HZy6Z  
Juglqoe/vdcJ9SP0SfJE6VWCEvjxB+eGgheFLKqzK8Hs/Bm0/wDKpSFgEp0PnkJ4  
HJ2Uzgn1Emo6gBDJt+qn3s2UnowcMsTgellhKvgzVq59LTyRyWL5U8XMBsXT4qjm  
0LkRvDk0IjMQH7kqvWbpPlnWpLko/VVoxifldeEgWAqFVRp7f5Y+nNQttAYV79uk  
MXvR+5YfKvmQaerflPqXBJdbB65ovikSVsy/kAboGpRG1oAZ40DdwdGyiGIzyyc  
lE0x/8+gY8BqWzRtWX4GySKyZ50/+xkJe5ss0IXPCgq/09bdihsRn57v4V4SpdD0  
k3g/Dce+LzCRL8uTbUhrhZnjKSjRc3fFaD/BpLYjEDbnGF0ICslN3vb2xWUK1u4M  
uUH9r7lH/DCb0+TxIBtx0nP7W02bz8gGJAXEVEqk6pjxx0YqfS9/uBrrAY8P21Y9  
PFLdeHzEdYemq3il+4S70U3uNUuAYijxmCRs7JQxZ9puA0iaTME9gKlyikzsLtVZ  
f+9osk2nYgFxlL0AiYabd5cU2GNW33TKdDMNBsB7lx77J9erVLZpPKNo4vgHA7b  
owrDaYe0AgcZm79fvmR0RdtIZI91MouEhkdhapiXmypszzJR/M00t3Y+oU/ks+yV  
Sle0S0h4V8wJRJYG/9VVurm8012ke2U3EGFLVnSv/IYtpssC+U4McRCmakKCrGU7  
0hL5JKBQN/DFTu4pV39IQLLLhg3wzA2FSkyIL5gEbS6sP9GTPo5LlNm2nYfJQX9A  
sHKSrfh68dvjSNExxi/8hdmFnnRwbAnUCI/W0bG0kKdhe0fdQ1AAHtL07G65X1Cx  
RctbAJWa93M+iRUN6qnB+viBPPnI1Mc7i6mPYzgtPrM9bYqEZz69pQtHcGTfx0Ru  
tm+/h36CRzJBfXodBZbwQ9mZAzfkKdlArLZYIEBUw30RQnQ7UljG8KsZpUhTxCc  
gvMoExtlvkXcYLRUBffZWy0i6FePzQjuCk1w580dweJgXprEAWsvyxhmVdg4jUpX  
MYKE0tZI9xwujyWjAC00myYqTdmqyds+BgfBn96XiA90FUH2C0/GAomhNs8uPS0  
T3Gt7Ld/FByxEVrtL9A37X6bAwZ001j5tHmdXFPmMVep0R8zsWtPn3RyGAjcgq6  
50wJRwhvofdI7wilZ0KUBsAaPj3MK52cRyD19VXKNNwt2bLDV6gcWQ8+QEMusxfp  
1Dc9N9DSs+w3lGsFfpoeQ53/fXcVNjM6Bv89bH9anLGYdCdRGvZsvw+xRuglykqb  
xLtL2lB6wzLRFREJoWTzCVsdpIZ8znPmk1cB0wDlbMeu6sddHmv+6fpyuvQfQmdj  
D8WLRtuyxax94TmBlhJCFYxm0/y4IvLx5C60GIRTKHpBYL/M0RjrbIszXEgcogzU  
bdwjLIhdEnpJ5vy0uXwhltce8BDpenmHE7y1kHvPBiUG3vB7AIXqhohFsJU3AYUj  
d1TvFKS2ASizUTLuq0Ydbnz3AxMfmnZe8qYkNu2zRyGL2xTa58f/MwsHKakk30mS  
9JFZLrkVWZKXoARctuahYtWBAsyKaWVnNB6zGcdX1MGVcc1930Z6QWHyydtZpQc  
ivNdEGdGv9B0K7/ngNdVgD5Wd29AMMFnS8+55mLfRZDCjUmshSySaf6Ein4HD9Hr  
vk6dJvBPjnI5UjeUPjmH+wcZKIjLHW/aV/6/zoxzBh61rWflr/daec+CFZE/+epr

LRRYSmv8oY47fF4duDDhoexcVP/CH+A2Hr400fciL4vKy3nuUDCNa59x09JWv4NL  
n3MQypC9bcaVPkXa7TK3ECq1Jgv8gwfdh5/ovG50dZA4uIc0+aqcskt/PD252c63  
0Znww3RXXf46KT4GdK05A377ixkUMkznnCMvottmkPxjnhQjAs0g3bJeQk8EoX8f  
Pq0If4i7SRBSDtb20H1pPmk0RVPtxlRDTVj3vS3Lci4xADFgC09n9nIvP0/55aaU  
06StbJtLmpubS5giuDH3uftwuyRiLqM3gtbSKPdoTk+dJhHXbbpBknL4XYTPxSsR  
IIaRds6w30vf7/IscyunMcquJls0929SSa93UevKEIZbqbV9oGIqwikiUMdVZK09g  
rW0F//Ts4a5nYdEQth/fq3JnwqeHvvUfKdasK4TtrTnUBX7qZk/K3Y1fZwjKdd/8  
t9t1z7Kb2d9hWwtY7xP8liDluVFTsq8NM54ZC2218X5ViWzlyFmF2LXvRixsmYJv  
Tz8lUUnC2B/Etm1kkU4zrYK0/L77EikkVl+B7BXfEqx6ow41j7e1YZYaqmZ9mph+  
UieSdzqVYxhPwT25DrkU3r74iS28gKsbFhUaNklaF005iDwsKgBXT+wdZqlYQ6Fo  
oPe66025iJMwK8t+d53jEduHezH02sTMAuf2hpdaZo7+rP/hRTReAR6CmI7nkWhP  
z5Kno9S+XhiSP+WTSps0A4ubx0T94mL8N0VvSZA76T230bVAP5VI/bwv6Grighor  
Kpsjt7dhSJRv+RHv95sAWBeW1Fgv8X0PSAZ0mpJV2qc3x3Qmj0MXIR+7+3GLUr8+  
Dit3CE1hwtXg0W0tc8kuBTfQD+wNSa9r0eUyFscEBBLjpEVbLjggVdNv4Hc+fsbT  
g1JzZuUIDQZoE02xLjxD+I7vLZKQa0J1JeZ70+NqmSxsvSnwCWtJEWNMMxYNfwsP  
rdj1zPLqn3rzSBqhroNbaDgn86BTwIqfhr+AKbvevxS6bI8IbyKm9u3BFr9cuawx  
Sp1QM3NtqNSTv67qR4A6U/ZyPUJd01bxo8F3oRmJq0t7Jc93rFgkhBJ2+eMtrA75  
0m5tB9LBVSL5U5yLP0C001QE5pqk5yuhJLT9Dyss8bWDRbSWKj83e4YXhPnq71Bm  
001czyLLVNUlDc69Tf7FXjtIxh2yJv0T3zeLBPX0jU0it+gAma4vgrh8/mMXnNiq  
OLsVow8aKqm+0fd6m13K5riDFgXgNI9lbvPKUSWlEqDMEqXk1oAqD4Nb5NTGSFpQ  
Q4G+cHAXJCu7vcXBaZnP8uMP5IAkdG5jIPvvMRwg/aqkl/KbL98oYZ5+1xr0MuKA  
LT1uCJ4MMB0lWsa1He4jPe8LneSupw7vAXlbo2Vzc0I6oCSY5hV+cGQRY+LjW81q  
Cu5nLq8bwgnZMSlPmwr0YrKmvh8YKyG0rmTadxykC5IC+XbrLDsw2Jd9mLIjUQ/V  
4ibjeb+e0QGob22W0plCLnHGW/SnYei8KG1dxs/ahS+8vQdrI880ZJx2QJnrz0Ej  
ux6tKv4mvUkqYA5hlTFeT3PTR54yA+YLCCLMfBDx4ykPQnYUBj70NHuNSUYt1CJy  
faZ7cWAbhgH+wLTfDVBVeW5D4FRbM8dMTPXyFC5ygwTJ0iDu3vQKyyDkmiX7sEaC  
P1JN2V55uacyR8ZAG5+Mlc4ZMx83kAIZZXTCdqa1EX8yda31FI2rDHmvW/82bmjL  
pvI4Nnn9+zzJtDVCJ0B2VAZ3Edov5GzPikm3un4+mvyhUZpH4sbT0+VhPCsr1+zn  
bDJyNw4AswxaaJKh2+7wBiU6h+9TP/lI8SAJHtZL7zHBH8tD10ptksLRWDs9vYqp  
/3T86S2vxJL5DvLFJSAZrY0E3InS+keGmTMCdAl9I8zIworC/8uQp0N8ESebEVjA  
aHotBk59lj/0W4JZ3tQkcdQWkpnUfW/x9xE2wthacHLRzYDDsFByjEqkQr0MU8VF  
EGij9RCC97zyFrhv0xJm1C6wX0pcuEcuPTNBf38WyBTIfmVHHZ/I5YKk5cdWG7Hq  
fmccV5GKrs2BseR683HM+/u50sq0km9UrqjgFR1Dj fDoRKp0guP9PqkJAnwG2nv1  
hmNtXumzkF0otP5LDKLJ84MP8Wnb006iEdD48Lra+cLRAIIuLX4A0wRQjViDp7n  
0ByI6ZcQd4DTMHnFPRvMkNMLYn13LghD6P9TTjQZ0KC0Cwmc2TMCihJlvz0YX6Cc  
wJZYL01ltgfnHEuh8ijv0u3d/BUpsknYKBSJGUyMEZ9iUtbFPVfXBGSTi3gcWHtL  
IrM7wjsWjWHWSvZKWUs+YWWJTwj0apG6ViGllw0AqR9C48uLKgFWPbMoTpolnp69  
eiiJ5ZHxB0i7SI80D+r65b+fqafzVIJXVEI0zu/miIlbYBnGkhLI/Naw1m2e1qVJ  
milJBjXLAT3pEJDh8b3Lpgw=

Unwrapping the outer Cryptographic Layer of this message yields the following MIME part (with its own Cryptographic Layer):



[illegible]

```
From: Alice Lovelace <alice@smime.example>
To: Bob Babbage <bob@smime.example>
Date: Wed, 27 Nov 2019 01:24:00 -0700
Subject: BarCorp contract signed, let's go!
Content-Type: multipart/mixed; boundary="6ae"; protected-headers="v1"
Message-ID: <smime-sign+enc+legacy-disp@protected-headers.example>

--6ae
content-type: text/plain; protected-headers="v1"
Content-Disposition: inline

Subject: BarCorp contract signed, let's go!

--6ae
Content-Type: text/plain; charset="us-ascii"

Hi Bob!

I just signed the contract with BarCorp and they've set us up with
an account on their system for testing.

The account information is:

    Site: https://barcorp.example/
    Username: examplecorptest
    Password: correct-horse-battery-staple

Please get the account set up and apply the test harness.

Let me know when you've got some results.

(this is the 'smime-sign+enc+legacy-disp' message)

Thanks, Alice
--
Alice Lovelace
President
Example Corp

--6ae--
```

## 9.10. Encrypted-only (unsigned) S/MIME Message with Protected Headers and Legacy Display

This shows the same encrypted message as [Section 9.9](#), but formulated without a signature layer, so it is "encrypted-only".

Note that the lack of any signature layer means that the only forms of cryptographic protection these header receive is confidentiality.

An arbitrary adversary could forge a message with arbitrary headers (and content), and package it in this same form. Consequently, the only thing "protected" about the headers in this example is confidentiality for any obscured headers (just the Subject in this case).

Presenting the cryptographic properties of the headers of such a message in a meaningful way to the end user is a subtle and challenging task, which this document cannot cover.

Its MIME message structure is:

```
└ application/pkcs7-mime smime-type="enveloped-data"
  └ (decrypts to)
    └ multipart/mixed ← Cryptographic Payload
      └ text/plain ← Legacy Display
        └ text/plain
```

For this message, the session key is an AES-256 key with value  
e94f6aaef7f14d6ceeac770c46d7f4885e81fbeatf1462d0fdadfce6c581525e2 (in hex).

```
Received: from localhost (localhost [127.0.0.1]); Wed, 27 Nov 2019
 01:27:28 -0700 (UTC-07:00)
MIME-Version: 1.0
Content-Transfer-Encoding: base64
Content-Type: application/pkcs7-mime; name="smime.p7m";
  smime-type="enveloped-data"
From: Alice Lovelace <alice@smime.example>
To: Bob Babbage <bob@smime.example>
Date: Wed, 27 Nov 2019 01:27:00 -0700
Message-ID: <smime-enc+legacy-disp@protected-headers.example>
Subject: ...
```

```
MIIG5QYJKoZIhvcNAQcDoIIIG1jCCBtICAQAxggLCMIIBXQIBADBFC0xKzApBgNV
BAMTIlNhXBsZSBMQU1QUyBDZXJ0awZpY2F0ZSBBdXR0b3JpdHkCFCJT7jBtAgsf
As3lycE+0t95phvCMA0GCSqGSIb3DQEBAQUABIIBADEhlzhFzYj6tUAdsRCrSiLL
d9cgKtLAesJ4cDY4szFWAbnwrCmEcFxfDU0jbfQCYCG80Sxd+xntni73I7PI2rR
QLjk3w9VhLwFRzy7qyJi2CavjKTxysX9f36+FXA+THfVQRM5ypiyYJg91X51PNX
hJj3DHrnXqKeSL/zlhdT9r+s6XAUCBSvL99BGnODWhNIZtPDzt8fMncgarfw+D5F
IZJb6+wX30tkztHkpHHKrrDPveyfnLS/p06Gi3ekrrhBtMQMRb9PA/E+ivDPktsm
aKg00auw4oZSKW3f4ukYhbnndbbagNsnTfs/QFy/p+hhKTrfCd0h1N8mTzedVX0w
ggFdAgEAMEUwLTERmCkGA1UEAxMiU2FtcGx1IEExBTBVTBIENlcnRpZmljYXRlIEF1
dGhvcml0eQIUIZ4K0WXSNS8H0cUcZavD9EYqqTAswDQYJKoZIhvcNAQEBBQAEggEA
FaK5QaPXJ133D2uybQt//oeDm6PkCAFW9YV0gjnLLz6FD54Dt2i1KCQu1Xlg9W3P
1zJdYX0ftDgilylNfmt/muEsvbRfFtMWUq0VGirHz//BwMY2cW/ocinF0514iviL
MLElumsXRnWVIVik/uh7AmqXjPkRZgRgIMUbSbtmW4DDja+ZM0vmqFQ1iUIlApth
FpjFfPDHHD8isLTBi2iK6dEN3DIJFGbg5o3nK6yAhVZ7x3LfFNSNVDDSY5mPFG9
Vm6uRgEE3Y5P6DbXXo6MHTgg0XY2f4y6MEWh0g37NT9aFAfzBBxJloSBWp00fZnV
K1DvAwPaemSRz9oWdCBM8DCCBAUGCSqGSIb3DQEHATAUBggqhkiG9w0DBWQIsFkN
8DEx8muAggPgWGF2WsPq3/a9jUa5GA0YFPiINuETCGTNaEXiVxnT0h0CF+EhZ0T2
HFCiZEM0dz005zt9wDvVAREaCSH7ZWG9D9wJF9x+ttqQbzMuJ2AdKuo0H73kClvkx
pHxANLhkY7hzIqRb/eLGS5D7Xh8iCDiFecXDh7EHqD/R+sflN9aHK0cKyY36kesBQ
R8aHZbbFnnD+oXSDNIPcntGG3BSGMxsWu0p+rpTKeIHWFIungDNKsLIy3kWleENw
FVICjUF6QhI1HYW6BeXuV40GV200kmB24rYEW1Jg0hAtY+5rn2mRoyxvUC87bjQ
hLu6xgPmhun9J324eM5aYVwkmVBnRW9hyxClZ7Sv0zLL7LGQ0VQG+zWHeJ+h/M2j
mQpLgAUEGxxNCm5ASHuXPIN6pSvr0VplrT8kKLpPMYEWmTX2/rB04P8I8uNrQYD
AyX8p0/l2ArczkWzGTz2luBahrD+cTZPApe5SeyX0xWBl1Lmb0G8o4twBeeBLiHP
XwYvttx0JYG/hc/lmMpEemJqwJ9uZ3wGD03dIhhDX20j4ek/7jT6yqJh8C1H+PqA
+HNfNXsFQDrR0RoqJS8YVEiYRDQNYePy2ugzLTh88nPtJp92hY7bk9zL3AYaiVFH
+szLLoyzfM9D+geZemR8XfI2ijGnrWmLnypah/zA6J6RwemhuiMklZGYG85hMU9H
K4CFVM+m7xYxKpwFVnmkVZjzWInirJhehElhtCXpx/IFGxH9CPbCyEZV1WVStrl/
0fWTGicMXez6hVQCadWCXy96/eLIX0rC54gSoIJX2TD6jdVEu1YptutyGI6KdQ2p
yXwhs98Uj7DM3nmFeAcjjN3e8pPoX7aG8eP+MfmHLWN6jA44jMaJmIdp9J20g74J
MdjvnHa/cGibW/RamPiF0bN0F94A83vcpUfU/zZ8cFHi/3/LN6Rm9+3/giGRZa9E
Y6e2/CEq1cUbPQ09fPwRjMjZCfDce71DKe+ZFGdYtFR7JwDEeZ6BB4Ff4rXctcWD
PgUJqUGv/SXBcFn4cNUK9MYyqVuIovd/T7FMf+i3c5MH6BRCvft/i5aeBR+A26Gk
2awtBPYdHW6+AslrFjncBbtPD1U6vX9AWuC0k0MQYnNkTWS8gTvsriXJZ6Zu5iFE
ExNuFz7YcnMKngu0n2ph5azzeMm83AYzWXzZPu3mdr5Siuu/Ke38oADKP+BZ08Za
XVvKvvnRXPX09kG9hgveMRU9K0cxn82XoGPNZib+9SPa2zYx5P6HX1Bqe/cmKAen
FKEiJLSTP2/pc6AWAICqJl978HaUHfMFIn7jEUppAifpAwqNcIGSw5w=
```

Unwrapping the single-layer Cryptographic Envelope of this message yields the following MIME structure:

```
From: Alice Lovelace <alice@smime.example>
To: Bob Babbage <bob@smime.example>
Date: Wed, 27 Nov 2019 01:27:00 -0700
Subject: BarCorp contract signed, let's go!
Content-Type: multipart/mixed; boundary="6ae"; protected-headers="v1"
Message-ID: <smime-enc+legacy-disp@protected-headers.example>
```

```
--6ae
content-type: text/plain; protected-headers="v1"
Content-Disposition: inline
```

Subject: BarCorp contract signed, let's go!

```
--6ae
Content-Type: text/plain; charset="us-ascii"
```

Hi Bob!

I just signed the contract with BarCorp and they've set us up with an account on their system for testing.

The account information is:

```
    Site: https://barcorp.example/
    Username: examplecorptest
    Password: correct-horse-battery-staple
```

Please get the account set up and apply the test harness.

Let me know when you've got some results.

(this is the 'smime-enc+legacy-disp' message)

Thanks, Alice

```
--
Alice Lovelace
President
Example Corp
```

```
--6ae--
```

## 9.11. Encrypted-only (unsigned) PGP/MIME Message with Protected Headers and Legacy Display

This shows a comparable encrypted-only (unsigned) message, like [Section 9.10](#) , but using PGP/MIME instead of S/MIME.

Note that the lack of any signature layer means that the only forms of cryptographic protection these header receive is confidentiality.

An arbitrary adversary could forge a message with arbitrary headers (and content), and package it in this same form. Consequently, the only thing "protected" about the headers in this example is confidentiality for any obscured headers (just the Subject in this case).

Presenting the cryptographic properties of the headers of such a message in a meaningful way to the end user is a subtle and challenging task, which this document cannot cover.

Its MIME message structure is:

```
└─ multipart/encrypted
  └─ application/pgp-encrypted
    └─ application/octet-stream
      └─ (decrypts to)
        └─ multipart/mixed ← Cryptographic Payload
          └─ text/plain ← Legacy Display
            └─ text/plain
```

For this message, the session key is an AES-256 key with value  
4f3e7e3cb4a49747f88d232601fa98a29d7427e8f80882464cfbca3dcb847356 (in hex).

```
Received: from localhost (localhost [127.0.0.1]); Mon, 21 Oct 2019
07:30:28 -0700 (UTC-07:00)
MIME-Version: 1.0
Content-Type: multipart/encrypted; boundary="c07";
protocol="application/pgp-encrypted"
From: Alice Lovelace <alice@openpgp.example>
To: Bob Babbage <bob@openpgp.example>
Date: Mon, 21 Oct 2019 07:30:00 -0700
Message-ID: <pgpmime-enc+legacy-disp@protected-headers.example>
Subject: ...
```

```
--c07
content-type: application/pgp-encrypted
```

Version: 1

```
--c07
content-type: application/octet-stream
```

-----BEGIN PGP MESSAGE-----

```
wV4DR2b2udXyHrYSAQdAX8p0+U8WbFNtCeGX5no1X1mSPqdmwrJIVWVZT8LS/yIw
lv+vor/Wsh7cKBofslYlPR4u/01EKjj+XkgD+h1BEtHDHp9ckuzBHm0I6YL0AZU
wcDMA3wvqk35PDeyAQwAiGcX6KN1jS+gHFAUcWwvc672CPP0hIhS91BGz4MMiV/G
Prm+dwIE5V7I6Sh7XMEons1Z7EdUbpXp/0ufCTQwrkXlzTTIt/0TMZkZxpDvLPpA
EzkdW2edtMhbTtqbGzjXg0sBVqnRZP6CaTfCba5tsVF0J8X0+WL1ARQSDVKWPuob
uXT+s4sZIam0JjnrXGYCD5NTjQt4UUmXlyXxQLEwN90wMLs8DrQ5kxcMHUU6kjDT
7icQRtsuIXXzrj0AVie0/Vd1ItKjrIo3eMvpi8G3GtB5VXYB2RPGKY6/cMISYGbx
s7aJVLW0Trri04p4vFi0I6iM1Y0dinbgCbzTXK+aYJpw5TmG/V5sHfRQXu77HBll
8BZdC+s6v5MWSdB9qVyvnd/e97mfi+ySa4Lw4yeLJFz70euL8C1SeQWhTmWIkwn6
FjiLFoxzkkLUE8vxcAYIUuzFMPCEUEXjH8EoLBwFz4jD0TQ4FJqn6lv9AEiJS4P4
mkgKdrvGqCSkZu6DpLgi0sGGAYu7ECCJLDcNTM6/S6o9AU9LcJJPgbd2wIylJyFY
D6ygG0D5skuKRsj7I/VJLx5SI6rkfTqd+vXcVcEX7vuhFAap988haqxS4fsFb/0L
CeLwZH94Y9hAP7Rz/hDiwHKcV1S0eAFFEfZ3u7kmMM2+o7zePIeimHbjSDjSAts5
GhZV7UDFyy6RnhSYgTNHw0hZToEPPLbH0mTzNZNp3tiS3apvYe6Yx9fCspd63Cet
tW5Y0vCpH00hJPIIv0ucVZsstn56SDBaYh70Fgq7M5UeK3AZ5KvH4cee4qd0KBgK
JZXBtIsoMICQj6Xw7ecmwP05huh1EQ0cfqdSuEu+k2ifgn0MApe85syK/d4yVxUB
wSj7Jk5r2Ytqe8ZXVoM4kYIKxVpuXmxb78KoUPvBUKLzq0MHwYpk2BjPQjZ8xqL7
oKQ8ywpm90SBB7DCGES7oIgrG5ZMovqVknppdJ3TrvkdgWtctbGe/Pb1WapMamQ/
a99+zfc9k63hDV6GW7mM7AiT05cqk0vYeynJShTpszf0eiIe+smM/3As4HJstCx7
Wiej+lM/Rqxp81nP8R78+a16iyIdbHZ6LSxD5vKgZbht30Qng0goZ3XQZXmIV/cZ
hVpPIEDgUzQi3qJq9P0PejosLQZhU41k0cyDdLZmPm70IRG7+b2X8JRbmhtg8FMA
szxT753uRpiGsKYb3dm0X9JYcDVbe9gFoIj2PktU2L96I9J79IVn9gtEeMYdR6Xn
w9rKgAyGiieepz5ygl9cRaGVfLnesAB
=zBUs
```

-----END PGP MESSAGE-----

--c07--

Unwrapping the single-layer Cryptographic Envelope of this message yields the following MIME structure:

```
From: Alice Lovelace <alice@openpgp.example>
To: Bob Babbage <bob@openpgp.example>
Date: Mon, 21 Oct 2019 07:30:00 -0700
Subject: BarCorp contract signed, let's go!
Content-Type: multipart/mixed; boundary="6ae"; protected-headers="v1"
Message-ID: <pgpmime-enc+legacy-disp@protected-headers.example>
```

```
--6ae
content-type: text/plain; protected-headers="v1"
Content-Disposition: inline
```

Subject: BarCorp contract signed, let's go!

```
--6ae
Content-Type: text/plain; charset="us-ascii"
```

Hi Bob!

I just signed the contract with BarCorp and they've set us up with an account on their system for testing.

The account information is:

```
    Site: https://barcorp.example/
    Username: examplecorptest
    Password: correct-horse-battery-staple
```

Please get the account set up and apply the test harness.

Let me know when you've got some results.

(this is the 'pgpmime-enc+legacy-disp' message)

Thanks, Alice

```
--
Alice Lovelace
President
Example Corp
```

```
--6ae--
```

## 9.12. An Unfortunately Complex Example

For all of the potential complexity of the Cryptographic Envelope, the Cryptographic Payload itself can be complex. The Cryptographic Envelope in this example is the same as ([Section 9.8](#)). The Cryptographic Payload has protected headers and a legacy display part (also the same as [Section 9.8](#)), but in addition Alice's MUA composes a message with both plaintext and HTML variants, and Alice includes a single attachment as well.

While this PGP/MIME message is complex, a modern MUA could also plausibly generate such a structure based on reasonable commands from the user composing the message (e.g., Alice composes the message with a rich text editor, and attaches a file to the message).

The key takeaway of this example is that the complexity of the Cryptographic Payload (which may contain a Legacy Display part) is independent of and distinct from the complexity of the Cryptographic Envelope.

This message has the following structure:

```
└─ multipart/encrypted
   └─ application/pgp-encrypted
      └─ application/octet-stream
         └─ (decrypts to)
            └─ multipart/signed
               └─ multipart/mixed ← Cryptographic Payload
                  └─ text/plain ← Legacy Display Part
                     └─ multipart/mixed
                        └─ multipart/alternative
                           └─ text/plain
                              └─ text/html
                                 └─ text/x-diff ← attachment
                                    └─ application/pgp-signature
```

For this message, the session key is an AES-256 key with value  
1c489cfad9f3c0bf3214bf34e6da42b7f64005e59726baa1b17ffdefe6ecbb52 (in hex).

Received: from localhost (localhost [127.0.0.1]); Mon, 21 Oct 2019  
07:33:28 -0700 (UTC-07:00)  
MIME-Version: 1.0  
Content-Type: multipart/encrypted; boundary="241";  
protocol="application/pgp-encrypted"  
From: Alice Lovelace <alice@openpgp.example>  
To: Bob Babbage <bob@openpgp.example>  
Date: Mon, 21 Oct 2019 07:33:00 -0700  
Message-ID: <unfortunately-complex@protected-headers.example>  
Subject: ...

--241  
content-type: application/pgp-encrypted

Version: 1

--241  
content-type: application/octet-stream

-----BEGIN PGP MESSAGE-----

wV4DR2b2udXyHrYSAQdArYyyCfDzUyr02W1QjJmXivzmT6XooGh6HMhPLmD/pkIw  
jPsIvobM6mmvctBWNgsg2IUvX3clXJum+/UmVuk5BQv0xk6x6kDt2WtwE3fWhop3  
wcDMA3wvqk35PDeyAQv+JZG91UzU5NJOY1Yxoadl8bNBkTdLBWN8DJEMhJd+Hm5  
KDjxBtAHWcsjzkiEdZcoR9EvrfFWBCTo+AmfnDi5YEJaX6GNr61VHKDcxowCrNsC  
lwfdXX+TIE0cwX7RWlyvWGxCS7alVHuxUa/hDe7DklAIx0icdTKz+lpDYFT8T9E  
Q/jtkk95paCzmtZ53RkaEMzizaJXD+B2s0/pBp6aJGxYMRf4yhez+b4HakUz2GK6  
tvFoN/qqXT97+cpREAhDFqtgHp6QmW4UUTgWaZ7G7TSDU7AuuiZxGCC5yGj0l19B  
iwm9xoG6YvjQxKbq6klaRZabUzFxyIKcuU8iDM9eZFLHu0QFhZKYSEmVaVNB9G1C  
i30ncaq7Ylkj73o90ogsilQwqdTRNZKz+65mPSzKj6HI7gu1w9Yf0MHcsHNPg9sI  
qTE/a88b17fc5qEEzkk8gmtnKyDI1bRvhxkrRNGWNeW6ZUEFdinYi5fAD5QYXMSW  
rIB+ELy/ZUYHHy31UAvS0sPRAXgbRmpFyrFzGgZMfkSbH2n+ngl+2lrDjnABUetE  
vSdvPCL57jS+w4MaUHz7wSjv1QnzBvRts/AJAvnFYhRYe5vP3wfDIKndpnhCz7EE  
QUE5d3upWL2fQ2UP/hLUWjBc6FhD+GFbyw38XomjBvvznT2NAFdZRLqqXfdw+dkG  
/daknCkHtyZ3Z1kQkTyyE0kuIopr2cJUWLgh0Euv00Ei842NsaDeKa05GepNXl0c  
9M9ScoUurCUGCa31tCe54GyceWs390ir6uiTeiJ5m1lN0KpuoDfiHKvVdM05Ge8+  
SLxZ03gyXEUPV//lhqqy3DwgYmL4M7SjxpJFLeu/YbguQuu4jpp/XBgZkc0eB//F  
FHShbmH6oEi59auutJ3I/NWl6n8EI0mRex64RYP8Bu3SLvVfsxlkjXHZk3XX52n  
vU4oUgHTpzUkJ8NxxmP0ZY8tu5MB7wBRp2Cqxx+r0KyHQPOrLU7iej0tXMHYHzwh  
QZ3/6BX9GR9ZBovqdZW0IzswjEradRfJXv0dL9QEL6V41m1tnFpeuaeNGCpMVqxN  
zvQf1T6z1JnX/hG0XwkKmFYz92MaeofNjx6ke++cAgfdRAqQxp77RkfBZdjtdFVV  
DggHI67I7DSs/sF+0ftJRet6E7rJlXYKJ24aB8ZkpLRU/eRvPXTaNNluoI7nMG2p  
Uf/LBTS+H+2jd5PB7vcIsvrTRuvCDqktniTk2eF3yYNHVEPlP7TmqpIVLXIFgc2Z  
NygS02HGQ56Cv8/HZKxaJ1tZDbUy9fVRtetj1lpsol5CfoGi8IVInI6gMWu3IBbb  
gqpV00YldQintY/BK49Q0y3lSh/5tgz+n6CZVxPxP1j+kVz0UGNy+SeThDC+H+lY  
d6Dd5+M+H5b/+XANBMKArzQVxDCSPtvi08qf1bwmZBB/ryylpLLDHpoYg0LC3Dk  
X/ICCAyk6n3Rz4IyupFuKNaEaiIwpjZZjqYtHbvMNjj+55crArYlfdadpTPeX5q8  
2QUg03J5ShkTlgp/a6qBuoUC3yHDcA0EiqGCMsF4Mmny6MtyzkKQXlgBHCDSG0y0  
NTnhfJxiKs1cahWf7ix9p05dn3lTqr1+t9usJtrZuhugVW0nbzQgfA4DNULbTsu5  
odSTwvrBczga7+JcvDJ+QELLiP8n1QcU2VkvCVwy5RHkwWzY0J84jYlh1VZEbbWa  
YDFXbQzCWGRcjubwb5Eet6pEPiNnTVvo6gGQx21Bue5kTslIZ01wRLiioU3vP4T0  
x4/6AaJt8MmSxXiGd9fjTT5ej7iawzH9qXQ40Umj3MvWni0rhRittRZyjXVaxdYG  
/F9sj5kkN0zFsSNaK3+Mi96Il6h6h4aYmVbrd1zapA8oqj6MpZRSeLL0HiHqmbcC  
IMXywNeKw2ZZSM6FNjU33fEDIQn0+jXLVazdkmqtBB0sUiuBuvMrKoJtr79rmiXC  
K77CmcJbikYpM0hnMyDfrtQqCEW4dKZ1c8uuFJQrEhRbQ24KP+Dq70ynNi0DalKN  
s4RgECgNgjES6ow4eIDS7vTo3xctCtXfzI5pkw8ub1rSM+Q=  
=wxHa

-----END PGP MESSAGE-----

--241--

Unwrapping the encryption Cryptographic Layer yields the following content:



Content-Type: multipart/signed; boundary="c72";  
protocol="application/pgp-signature"; micalg="pgp-sha512"

--c72

From: Alice Lovelace <alice@openpgp.example>  
To: Bob Babbage <bob@openpgp.example>  
Date: Mon, 21 Oct 2019 07:33:00 -0700  
Subject: BarCorp contract signed, let's go!  
Content-Type: multipart/mixed; boundary="6ae"; protected-headers="v1"  
Message-ID: <unfortunately-complex@protected-headers.example>

--6ae

Content-type: text/plain; protected-headers="v1"  
Content-Disposition: inline

Subject: BarCorp contract signed, let's go!

--6ae

Content-Type: multipart/mixed; boundary="8df"

--8df

Content-Type: multipart/alternative; boundary="32c"

--32c

Content-Type: text/plain; charset="us-ascii"

Hi Bob!

I just signed the contract with BarCorp and they've set us up with  
an account on their system for testing.

The account information is:

Site: <https://barcorp.example/>  
Username: examplecorptest  
Password: correct-horse-battery-staple

Please get the account set up and apply the test harness.

Let me know when you've got some results.

(this is the 'unfortunately-complex' message)

Thanks, Alice

--

Alice Lovelace  
President  
Example Corp

--32c

Content-Type: text/html; charset="us-ascii"

<html><head></head><body><p>Hi Bob!

</p><p>

I just signed the contract with BarCorp and they've set us up with  
an account on their system for testing.

</p><p>

The account information is:

</p><dl>

<dt>Site</dt><dd>

<a href="https://barcorp.example/">https://barcorp.example/</a>

</dd>

```

<dt>Username</dt><dd><tt>examplecorptest</tt></dd>
<dt>Password</dt><dd>correct-horse-battery-staple</dd>
</dl><p>
Please get the account set up and apply the test harness.
</p><p>
Let me know when you've got some results.
</p><p>
(this is the 'unfortunately-complex' message)
</p><p>
Thanks, Alice<br/>
-- <br/>
Alice Lovelace<br/>
President<br/>
Example Corp<br/>
</p></body></html>

--32c--

--8df
Content-Type: text/x-diff; charset="us-ascii"
Content-Disposition: inline; filename="testharness-config.diff"

diff -ruN a/testharness.cfg b/testharness.cfg
--- a/testharness.cfg
+++ b/testharness.cfg
@@ -13,3 +13,8 @@
 endpoint = https://openpgp.example/test/
 username = testuser
 password = MJVMZlHR75mILg
+
+[barcorp]
+endpoint = https://barcorp.example/
+username = examplecorptest
+password = correct-horse-battery-staple

--8df--

--6ae--

--c72
content-type: application/pgp-signature

-----BEGIN PGP SIGNATURE-----

wnUEARYKAB0FAl2twZwWIQT rhbtfozp14V6UTmPyMVUMT0fjjgAKCRDyMVUMT0fj
jnUTAP9YDBbjItEr14L3f/hpRDdkiexX96wHRZ0ZlP4VlsPbmgeA/zNQ5GZx0W70
EyF6maqK0Dedw/FXsbl32iFiXMGaTgY=
=EuL1
-----END PGP SIGNATURE-----

--c72--

```

## 10. IANA Considerations

FIXME: register content-type parameter for legacy-display part

MAYBE: provide a list of user-facing headers, or a new "user-facing" column in some table of known RFC5322 headers?

MAYBE: provide a comparable indicator for which headers are "structural" ?

## 11. Security Considerations

This document describes a technique that can be used to defend against two security vulnerabilities in traditional end-to-end encrypted e-mail.

### 11.1. Subject Leak

While e-mail structure considers the Subject header to be part of the message metadata, nearly all users consider the Subject header to be part of the message content.

As such, a user sending end-to-end encrypted e-mail may inadvertently leak sensitive material in the Subject line.

If the user's MUA uses Protected Headers and obscures the Subject header as described in [Section 4.2](#) then they can avoid this breach of confidentiality.

### 11.2. Signature Replay

A message without Protected Headers may be subject to a signature replay attack, which attempts to violate the recipient's expectations about message authenticity and integrity. Such an attack works by taking a message delivered in one context (e.g., to someone else, at a different time, with a different subject, in reply to a different message), and replaying it with different message headers.

A MUA that generates all its signed messages with Protected Headers gives recipients the opportunity to avoid falling victim to this attack.

Guidance for how a message recipient can use Protected Headers to defend against a signature replay attack are out of scope for this document.

### 11.3. Participant Modification

A trivial (if detectable) attack by an active network adversary is to insert an additional e-mail address in a To or Cc or Reply-To or From header. This is a staging attack against message confidentiality - it relies on followup action by the recipient.

For an encrypted message that is part of an ongoing discussion where users are accustomed to doing "reply all", such an insertion would cause the replying MUA to encrypt the replying message to the additional party, giving them access to the conversation. If the replying MUA quotes and attributes cleartext from the original message within the reply, then the attacker learns the contents of the encrypted message.

As certificate discovery becomes more automated and less noticeable to the end user, this is an increasing risk.

An MUA that rejects Exposed Headers in favor of Protected Headers should be able to avoid this attack when replying to a signed message.

## 12. Privacy Considerations

This document only explicitly contemplates confidentiality protection for the Subject header, but not for

other headers which may leak associational metadata. For example, From and To and Cc and Reply-To and Date and Message-Id and References and In-Reply-To are not explicitly necessary for messages in transit, since the SMTP envelope carries all necessary routing information, but an encrypted [RFC5322] message as described in this document will contain all this associational metadata in the clear.

Although this document does not provide guidance for protecting the privacy of this metadata directly, it offers a platform upon which thoughtful implementations may experiment with obscuring additional e-mail headers.

## 13. 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://github.com/autocrypt/protected-headers> or by e-mail to the authors. Please direct all significant commentary to the public IETF LAMPS mailing list: [spasm@ietf.org](mailto:spasm@ietf.org)

### 13.1. Document History

Significant changes between version -01 and -02:

- Added S/MIME test vectors in addition to PGP/MIME
- Legacy Display parts should now be text/plain and not text/rfc822-headers
- Cryptographic Payload must have protected-headers parameter set to v1
- Test vector sample Message-Ids have been normalized
- Added encrypted-only (unsigned) test vectors, at the suggestion of Russ Housley

Changes between version -00 and -01:

- Credit Randall for "correct horse battery staple".
- Adjust test vectors to ensure no line in the generated .txt format exceeds 72 chars.
- Minor formatting cleanup to appease idnits.
- Update references to more recent documents (RFC 2822 -> 5322, -00 to -01 of draft-ietf-lamps-header-protection-requirements).

## 14. Acknowledgements

The set of constructs and algorithms in this document has a previous working title of "Memory Hole", but that title is no longer used as different implementations gained experience in working with it.

These ideas were tested and fine-tuned in part by the loose collaboration of MUA developers known as [Autocrypt].

Additional feedback and useful guidance was contributed by attendees of the OpenPGP e-mail summit ([OpenPGP-Email-Summit-2019]).

The following people have contributed implementation experience, documentation, critique, and other feedback:

- Holger Krekel
- Patrick Brunschwig
- Vincent Breitmoser
- Edwin Taylor
- Alexey Melnikov
- Russ Housley

The password example used in [Section 9](#) comes from [\[xkcd936\]](#).

## 15. References

### 15.1. 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>>.
- [RFC4880]** Callas, J., Donnerhacke, 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>>.
- [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>>.

### 15.2. Informative References

- [Autocrypt]** "Autocrypt Specification 1.1", 13 October 2019, <<https://autocrypt.org/level1.html>>.
- [I-D.draft-bre-openpgp-samples-00]** Einarsson, B., juga, j., and D. Gillmor, "OpenPGP Example Keys and Certificates", Work in Progress, Internet-Draft, draft-bre-openpgp-samples-00, 15 October 2019, <<http://www.ietf.org/internet-drafts/draft-bre-openpgp-samples-00.txt>>.
- [I-D.draft-dkg-lamps-samples-01]** Gillmor, D., "S/MIME Example Keys and Certificates", Work in Progress, Internet-Draft, draft-dkg-lamps-samples-01, 20 November 2019, <<http://www.ietf.org/internet-drafts/draft-dkg-lamps-samples-01.txt>>.
- [I-D.draft-ietf-lamps-header-protection-requirements-01]** 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, <<http://www.ietf.org/internet-drafts/draft-ietf-lamps-header-protection-requirements-01.txt>>.
- [I-D.draft-luck-lamps-pep-header-protection-03]** Luck, C., "pretty Easy privacy (pEp): Progressive Header Disclosure", Work in Progress, Internet-Draft, draft-luck-lamps-pep-header-protection-03, 5 July 2019, <<http://www.ietf.org/internet-drafts/draft-luck-lamps-pep-header-protection-03.txt>>.

**[OpenPGP-Email-Summit-2019]** "OpenPGP Email Summit 2019", 13 October 2019, <<https://wiki.gnupg.org/OpenPGPEmailSummit201910>>.

**[RFC2634]** Hoffman, P., Ed., "Enhanced Security Services for S/MIME", RFC 2634, DOI 10.17487/RFC2634, June 1999, <<https://www.rfc-editor.org/info/rfc2634>>.

**[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>>.

**[RFC3851]** Ramsdell, B., Ed., "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification", RFC 3851, DOI 10.17487/RFC3851, July 2004, <<https://www.rfc-editor.org/info/rfc3851>>.

**[RFC6736]** Brockners, F., Bhandari, S., Singh, V., and V. Fajardo, "Diameter Network Address and Port Translation Control Application", RFC 6736, DOI 10.17487/RFC6736, October 2012, <<https://www.rfc-editor.org/info/rfc6736>>.

**[RFC7508]** Cailleux, L. and C. Bonatti, "Securing Header Fields with S/MIME", RFC 7508, DOI 10.17487/RFC7508, April 2015, <<https://www.rfc-editor.org/info/rfc7508>>.

**[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>>.

**[xkcd936]** Munroe, R., "xkcd: Password Strength", 10 August 2011, <<https://www.xkcd.com/936/>>.

## Authors' Addresses

### **Bjarni Rúnar Einarsson**

Mailpile ehf  
Baronsstigur  
Iceland  
Email: [bre@mailpile.is](mailto:bre@mailpile.is)

### **juga**

Independent  
Email: [juga@riseup.net](mailto:juga@riseup.net)

### **Daniel Kahn Gillmor**

American Civil Liberties Union  
125 Broad St.  
New York, NY, 10004  
United States of America  
Email: [dkg@fifthhorseman.net](mailto:dkg@fifthhorseman.net)