

Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 14, 2021

B. Hoeneisen
pEp Foundation
A. Melnikov
Isode Ltd
July 13, 2020

Header Protection for S/MIME
draft-ietf-lamps-header-protection-00

Abstract

Privacy and security issues with email header protection in S/MIME have been identified for some time. However, the desire to fix these issues has only recently been expressed in the IETF LAMPS Working Group. The existing S/MIME specification is to be updated regarding header protection.

This document describes the problem statement, generic use cases, and the S/MIME specification for header protection.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

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

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

This Internet-Draft will expire on January 14, 2021.

Copyright Notice

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

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	3
1.1.	Requirements Language	4
1.2.	Terms	4
2.	Problem Statement	6
2.1.	Privacy	6
2.2.	Security	6
2.3.	Usability	6
2.4.	Interoperability	7
3.	Use Cases	7
3.1.	Interactions	7
3.1.1.	Main Use Case	7
3.1.2.	Backward Compatibility Use Cases	7
3.2.	Protection Levels	9
4.	Specification	9
4.1.	Main Use Case	10
4.1.1.	MIME Format	10
4.1.2.	Inner Message Header Fields	15
4.1.3.	Wrapper	16
4.1.4.	Outer Message Header Fields	16
4.1.5.	Receiving User Facing Message Header Fields	18
4.1.6.	Header Field Flow	18
4.1.7.	Sending Side Message Processing	20
4.1.8.	Receiving Side Message Processing	21
4.2.	Backward Compatibility Use Cases	21
4.2.1.	Receiving Side MIME-Conformant	21
4.2.2.	Receiving Side Not MIME-Conformant	22
5.	Security Considerations	22
6.	Privacy Considerations	23
7.	IANA Considerations	23
8.	Acknowledgments	23
9.	References	23
9.1.	Normative References	23
9.2.	Informative References	24
Appendix A.	Additional information	25
A.1.	Stored Variants of Messages with Bcc	25
Appendix B.	Document Changelog	26
Appendix C.	Open Issues	26
	Authors' Addresses	27

1. Introduction

A range of protocols for the protection of electronic mail (email) exists, which allows to assess the authenticity and integrity of the email headers section or selected header fields (HF) from the domain-level perspective, specifically DomainKeys Identified Mail (DKIM) [[RFC6376](#)] and Sender Policy Framework (SPF) [[RFC7208](#)], and Domain-based Message Authentication, Reporting, and Conformance (DMARC) [[RFC7489](#)]. These protocols, while essential to responding to a range of attacks on email, do not offer (full) end-to-end protection to the header section and are not capable of providing privacy for the information contained therein.

The need for means of Data Minimization, which includes data sparseness and hiding all technically concealable information whenever possible, has grown in importance over the past several years.

A standard for end-to-end protection of the email header section exists for S/MIME version 3.1 and later. (cf. [[RFC8551](#)]):

In order to protect outer, non-content-related message header fields (for instance, the "Subject", "To", "From", and "Cc" fields), the sending client MAY wrap a full MIME message in a message/RFC822 wrapper in order to apply S/MIME security services to these header fields.

No mechanism for header protection (HP) has been standardized for PGP/MIME (Pretty Good Privacy) [[RFC3156](#)] yet.

Several varying implementations of end-to-end protections for email header sections exist, though the total number of such implementations appears to be rather low.

Some LAMPS WG participants expressed the opinion that regardless of the mechanism chosen, it should not be limited to S/MIME, but also applicable to PGP/MIME.

This document describes the problem statement ([Section 2](#)), generic use cases ([Section 3](#)) and the specification for Header Protection ([Section 4](#)).

[I-D.ietf-lamps-header-protection-requirements] defines the requirements that this specification is based on.

This document is in early draft state and contains a proposal on which to base future discussions of this topic. In any case, the final mechanism is to be determined by the IETF LAMPS WG.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [[RFC2119](#)].

1.2. Terms

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

- o Man-in-the-middle (MITM) attack: cf. [[RFC4949](#)], which states: "A form of active wiretapping attack in which the attacker intercepts and selectively modifies communicated data to masquerade as one or more of the entities involved in a communication association."
- o S/MIME: Secure/Multipurpose Internet Mail Extensions (cf. [[RFC8551](#)])
- o PGP/MIME: MIME Security with OpenPGP (cf. [[RFC3156](#)])
- o Message: An Email Message consisting of Header Fields (collectively called "the Header Section of the message") followed, optionally, by a Body; cf. [[RFC5322](#)].

Note: To avoid ambiguity, this document does not use the terms "Header" or "Headers" in isolation, but instead always uses "Header Field" to refer to the individual field and "Header Section" to refer to the entire collection; cf. [[RFC5322](#)].

- o Header Field (HF): cf. [[RFC5322](#)] Header Fields are lines beginning with a field name, followed by a colon (":"), followed by a field body (value), and terminated by CRLF; cf. [[RFC5322](#)].
- o Header Section (HS): The Header Section is a sequence of lines of characters with special syntax as defined in [[RFC5322](#)]. It is the (top) section of a Message containing the Header Fields.
- o Body: The Body is simply a sequence of characters that follows the Header Section and is separated from the Header Section by an empty line (i.e., a line with nothing preceding the CRLF); cf [[RFC5322](#)]. It is the (bottom) section of Message containing the payload of a Message. Typically, the Body consists of a (possibly multipart) MIME [[RFC2045](#)] construct.
- o MIME Header Fields: Header Fields describing content of a MIME entity [[RFC2045](#)], in particular the MIME structure. Each MIME Header Field name starts with "Content-" prefix.

- o MIME Header Section (part): The collection of MIME Header Fields. "MIME Header Section" refers to a Header Sections that contains only MIME Header Fields, whereas "MIME Header Section part" refers to the MIME Header Fields of a Header Section that - in addition to MIME Header Fields - also contains non-MIME Header Fields.
- o Essential Header Fields (EHF): The minimum set of Header Fields an Outer Message Header Section SHOULD contain; cf. [Section 4.1.4](#).
- o Header Protection (HP): cryptographic protection of email Header Sections (or parts of it) for signatures and/or encryption
- o Protection Levels (PL): One of 'signature and encryption', 'signature only' or 'encryption only' (cf. [Section 3.2](#))
- o Protected: Protected refers to the parts of a Message where protection measures of any Protection Level have been applied to.
- o Protected Message: A Message that protection measures of any Protection Levels have been applied to.
- o Unprotected: Unprotected refers to the parts of a Message where no protection measures of any Protection Levels have been applied to.
- o Unprotected Message: A Message that no protection measures of any Protection Levels have been applied to.
- o Submission Entity: The entity taking care of further processing of the Message (incl. transport towards the receiver), after protection measures have been applied to.

Note: The Submission Entity varies among implementations, mainly depending on the stage, where protection measures are applied to: It could be e.g. a Message Submission Agent (MSA) [[RFC6409](#)] or another (proprietary) solution. The latter is particularly relevant, if protection is implemented as a plugin solution. Some implementations may determine the destination recipients by reading the To, Cc and Bcc Header Fields of the Outer Message.

- o Original Message (OrigM): The message to be protected before any protection related processing has been applied on the sending side.
- o Inner Message (InnerM): The message to be protected, i.e. which wrapping and protection measures are applied to on the sending side or the result of decryption and unwrapping on the receiving side respectively. Typically, the Inner Message is in clear text. The Inner Message is a subset of (or the same as) the Original

Message (cf. [Section 4.1.2](#)). The Inner Message must be the same on the sending and the receiving side.

- o Outer Message (OuterM): The Message as handed over to the Submission Entity or received from the last hop respectively. The Outer Message normally differs on the sending and the receiving side (e.g. new Header Fields are added by intermediary nodes).
- o Receiving User Facing Message (RUFM): The message used for rendering at the receiving side. Typically this is the same as the Inner Message.
- o Data Minimization: Data sparseness and hiding of all technically concealable information whenever possible.

[2.](#) Problem Statement

The LAMPS charter contains the following Work Item:

Update the specification for the cryptographic protection of email headers - both for signatures and encryption - to improve the implementation situation with respect to privacy, security, usability and interoperability in cryptographically-protected electronic mail. Most current implementations of cryptographically-protected electronic mail protect only the body of the message, which leaves significant room for attacks against otherwise-protected messages.

In the following a set of challenges to be addressed:

[[TODO: Enhance this section, add more items to the following.]]

[2.1.](#) Privacy

- o Data Minimization, which includes data sparseness and hiding all technically concealable information whenever possible

[2.2.](#) Security

- o MITM attacks (cf. [\[RFC4949\]](#))

[2.3.](#) Usability

- o User interaction / User experience

2.4. Interoperability

- o Interoperability with [[RFC8551](#)] implementations

3. Use Cases

In the following, the reader can find a list of the generic use cases that need to be addressed for Messages with Header Protection (HP). These use cases apply regardless of technology (S/MIME, PGP/MIME, etc.) used to achieve HP.

3.1. Interactions

The following use cases assume that at least the sending side supports Header Protection as specified in this document. Receiving sides that support this specification are expected to be able to distinguish between Messages that Header Protection - as specified in this document - has been applied to and (legacy) Mail User Agents (MUAs) not implementing this specification.

[[TODO: Verify once solution is stable and update last sentence.]]

3.1.1. Main Use Case

Both the sending and receiving side (fully) support Header Protection as specified in this document.

The main use case is specified in [Section 4.1](#).

3.1.2. Backward Compatibility Use Cases

Regarding backward compatibility, the main distinction is based on whether or not the receiving side conforms to MIME according to [[RFC2046](#)], ff., which in particular also includes [Section 2 of \[\[RFC2049\]\(#\)\]](#) on "MIME Conformance". In the following an excerpt of paragraphs relevant in this context:

A mail user agent that is MIME-conformant MUST:

[...]

- Recognize and display at least the [RFC822](#) message encapsulation (message/rfc822) in such a way as to preserve any recursive structure, that is, displaying or offering to display the encapsulated data in accordance with its media type.

- Treat any unrecognized subtypes as if they were "application/octet-stream".

[...]

A user agent that meets the above conditions is said to be MIME-conformant. The meaning of this phrase is that it is assumed to be "safe" to send virtually any kind of properly-marked data to users of such mail systems, because such systems will at least be able to treat the data as undifferentiated binary, and will not simply splash it onto the screen of unsuspecting users.

[[TODO: The compatibility of legacy HP systems with this new solution, and how to handle issues surrounding future maintenance for these legacy systems, will be decided by the LAMPS WG.]]

[3.1.2.1](#). Receiving Side MIME-Conformant

The sending side (fully) supports Header Protection as specified in this document, while the receiving side does not support this specification. However, the receiving side is MIME-conformant according to [[RFC2045](#)], ff. (cf. [Section 3.1.2](#)),

This use case is specified in [Section 4.2.1](#).

Note: This case should perform as expected if the sending side applies this specification as outlined in [Section 4.1](#).

[[TODO: Verify once solution is stable and update last sentence.]]

[3.1.2.2](#). Receiving Side Not MIME-Conformant

The sending side (fully) supports Header Protection as specified in this document, while the receiving side does not support this specification. Furthermore, the receiving side is **not** MIME-conformant according to [[RFC2045](#)], ff. (cf. [Section 3.1.2](#)).

This use case is specified in [Section 4.2.2](#).

3.2. Protection Levels

The following Protection Levels need to be considered:

a) Signature and encryption

Messages containing a cryptographic signature, which are also encrypted.

b) Signature only

Messages containing a cryptographic signature, but which are not encrypted.

c) Encryption only

Messages that are encrypted, but do not contain a cryptographic signature.

[[TODO: There are further "Protection Levels" to describe for the receiving side, e.g. encrypted and signed (only after encryption), etc.]]

4. Specification

This section contains the specification for Header Protection in S/MIME to update and clarify [Section 3.1 of \[RFC8551\]](#) (S/MIME 4.0).

Note: It is likely that PGP/MIME [\[RFC3156\]](#) will also incorporate this specification or parts of it.

This specification applies to the Protection Levels "signature & encryption" and "signature only" (cf. [Section 3.2](#)):

Sending and receiving sides MUST implement the "signature and encryption" Protection Level", which SHOULD be used as default on the sending side.

Certain implementations may decide to send "signature only" messages, depending on the circumstances and customer requirements. Sending sides MAY and receiving sides MUST implement "signature only" Protection Level.

It generally is NOT RECOMMENDED to send a message with Protection Level "encryption only". On the other hand, messages with Protection Level "encryption only" might arrive at the receiving side. While

not targeted to Protection Level "encryption only", this specification is assumed to also function for "encryption only". Receiving sides SHOULD implement "encryption only".

[[TODO: Further study is necessary to determine whether - and if yes to what extent - additional guidance for handling messages with "encryption only" protection (as well as other variations) at the receiving side should be included in this document.]]

4.1. Main Use Case

This section applies to the main use case, where the sending and receiving side (fully) support Header Protection as specified herein (cf. [Section 3.1.1](#)).

Note: The sending side specification of the main use case is also applicable to the cases where the sending side (fully) supports Header Protection as specified herein, while the receiving side does not, but is MIME-conformant according to [\[RFC2045\]](#), ff. (cf. [Section 3.1.2](#)) and [Section 3.1.2.1](#))

Further backward compatibility cases are defined in [Section 4.2](#).

4.1.1. MIME Format

Currently there are two options in discussion:

1. The option according to the current S/MIME specification (cf. [\[RFC8551\]](#))
2. An alternative option that is based on the former "memory hole" approach (cf. [\[I-D.autocrypt-lamps-protected-headers\]](#))

4.1.1.1. S/MIME Specification

As per S/MIME version 3.1 and later (cf. [\[RFC8551\]](#)), the sending client MAY wrap a full MIME message in a message/RFC822 wrapper in order to apply S/MIME security services to these header fields.

To help the receiving side to distinguish between a forwarded and a wrapped message, the Content-Type header field parameter "forwarded" is added as defined in [\[I-D.melnikov-iana-reg-forwarded\]](#). Certain mailing applications might display the Inner Message as an attachment otherwise.

The MIME structure of an Email message looks as follows:

<Outer Message Header Section (unprotected)>

<Outer Message Body (protected)>

<MIME Header Section (wrapper)>

<Inner Message Header Section>

<Inner Message Body>

The following example demonstrates how an Original Message might be protected, i.e., the Original Message is contained as Inner Message in the Protected Body of an Outer Message. It illustrates the first Body part (of the Outer Message) as a "multipart/signed" (application/pkcs7-signature) media type:

Lines are prepended as follows:

- o "O: " Outer Message Header Section
- o "I: " Message Header Section
- o "W: " Wrapper (MIME Header Section)


```
O: Date: Mon, 25 Sep 2017 17:31:42 +0100 (GMT Daylight Time)
O: Message-ID: <e4a483cb-1dfb-481d-903b-298c92c21f5e@m.example.net>
O: Subject: Meeting at my place
O: From: "Alexey Melnikov" <alexey.melnikov@example.net>
O: To: somebody@example.net
O: MIME-Version: 1.0
O: Content-Type: multipart/signed; charset=us-ascii; micalg=sha1;
O:   protocol="application/pkcs7-signature";
O:   boundary=boundary-AM
```

This is a multipart message in MIME format.

--boundary-AM

W: Content-Type: message/RFC822; forwarded=no

W:

```
I: Date: Mon, 25 Sep 2017 17:31:42 +0100 (GMT Daylight Time)
I: From: "Alexey Melnikov" <alexey.melnikov@example.net>
I: Message-ID: <e4a483cb-1dfb-481d-903b-298c92c21f5e@m.example.net>
I: MIME-Version: 1.0
I: MMHS-Primary-Precedence: 3
I: Subject: Meeting at my place
I: To: somebody@example.net
I: X-Mailer: Isode Harrier Web Server
I: Content-Type: text/plain; charset=us-ascii
```

This is an important message that I don't want to be modified.

--boundary-AM

Content-Transfer-Encoding: base64

Content-Type: application/pkcs7-signature

[[base-64 encoded signature]]

--boundary-AM--

The Outer Message Header Section is unprotected, while the remainder (Outer Message Body) is protected. The Outer Message Body consists of the wrapper (MIME Header Section) and the Inner Message (Header Section and Body).

The wrapper is a simple MIME Header Section with media type "message/[RFC822](#)" containing a Content-Type header field parameter "forwarded=no" followed by an empty line.

The Inner Message Header Section is the same as (or a subset of) the Original Message Header Section (cf. [Section 4.1.2](#)).

The Inner Message Body is the same as the Original Message Body.

The Original Message itself may contain any MIME structure.

4.1.1.2. Alternative Option Autocrypt "Protected Headers" (Ex-"Memory Hole")

An alternative option (based on the former autocrypt "Memory Hole" approach) to be considered, is described in [\[I-D.autocrypt-lamps-protected-headers\]](#).

Unlike the option described in [Section 4.1.1.1](#), this option does not use a "message/RFC822" wrapper to unambiguously delimit the Inner Message.

Before choosing this option, the following two issues must be assessed to ensure no interoperability issues result from it:

1. How current MIME parser implementations treat non-MIME Header Fields, which are not part of the outermost MIME entity and not part of a message wrapped into a MIME entity of media type "message/rfc822", and how such messages are rendered to the user.

[\[I-D.autocrypt-lamps-protected-headers\]](#) provides some examples for testing this.

2. MIME-conformance, i.e. whether or not this option is (fully) MIME-conformant [[RFC2045](#)] ff., in particular also [Section 5.1. of \[RFC2046\]](#) on "Multipart Media Type). In the following an excerpt of paragraphs that may be relevant in this context:

The only header fields that have defined meaning for body parts are those the names of which begin with "Content-". All other header fields may be ignored in body parts. Although they should generally be retained if at all possible, they may be discarded by gateways if necessary. Such other fields are permitted to appear in body parts but must not be depended on. "X-" fields may be created for experimental or private purposes, with the recognition that the information they contain may be lost at some gateways.

NOTE: The distinction between an [RFC 822](#) message and a body part is subtle, but important. A gateway between Internet and X.400 mail, for example, must be able to tell the difference between a body part that contains an image and a body part that contains an encapsulated message, the body of which is a JPEG image. In order to represent the latter, the body part must have "Content-Type: message/rfc822", and its body (after the blank line) must be the encapsulated message, with its own "Content-Type: image/jpeg" header field. The use of similar syntax facilitates the conversion of messages to body parts, and vice versa, but the distinction between the two must be understood by implementors. (For the special case in which parts actually are messages, a "digest" subtype is also defined.)

The MIME structure of an Email message looks as follows:

<Outer Message Header Section (unprotected)>

<Outer Message Body (protected)>

<Inner Message Header Section>

<Inner Message Body>

The following example demonstrates how an Original Message might be protected, i.e., the Original Message is contained as Inner Message in the Protected Body of an Outer Message. It illustrates the first Body part (of the Outer Message) as a "multipart/signed" (application/pkcs7-signature) media type:

Lines are prepended as follows:

- o "O: " Outer Message Header Section
- o "I: " Message Header Section


```
0: Date: Mon, 25 Sep 2017 17:31:42 +0100 (GMT Daylight Time)
0: Message-ID: <e4a483cb-1dfb-481d-903b-298c92c21f5e@m.example.net>
0: Subject: Meeting at my place
0: From: "Alexey Melnikov" <alexey.melnikov@example.net>
0: MIME-Version: 1.0
0: Content-Type: multipart/signed; charset=us-ascii; micalg=sha1;
0:   protocol="application/pkcs7-signature";
0:   boundary=boundary-AM
```

This is a multipart message in MIME format.

--boundary-AM

```
I: Date: Mon, 25 Sep 2017 17:31:42 +0100 (GMT Daylight Time)
I: From: "Alexey Melnikov" <alexey.melnikov@example.net>
I: Message-ID: <e4a483cb-1dfb-481d-903b-298c92c21f5e@m.example.net>
I: MIME-Version: 1.0
I: MMHS-Primary-Precedence: 3
I: Subject: Meeting at my place
I: To: somebody@example.net
I: X-Mailer: Isode Harrier Web Server
I: Content-Type: text/plain; charset=us-ascii
```

This is an important message that I don't want to be modified.

--boundary-AM

Content-Transfer-Encoding: base64

Content-Type: application/pkcs7-signature

[[base-64 encoded signature]]

--boundary-AM--

The Outer Message Header Section is unprotected, while the remainder (Outer Message Body) is protected. The Outer Message Body consists of the Inner Message (Header Section and Body).

The Inner Message Header Section is the same as (or a subset of) the Original Message Header Section (cf. [Section 4.1.2](#)).

The Inner Message Body is the same as the Original Message Body.

The Original Message itself may contain any MIME structure.

[4.1.2](#). Inner Message Header Fields

It is RECOMMENDED that the Inner Message contains all Header Fields of the Original Message with the exception of the following Header

Field, which MUST NOT be included within the Inner Message nor within any other protected part of the message:

- o Bcc

[[TODO: Bcc handling needs to be further specified (see also [Appendix A.1](#)). Certain MUAs cannot properly decrypt messages with Bcc recipients.]]

[4.1.3.](#) Wrapper

The wrapper is a simple MIME Header Section followed by an empty line preceding the Inner Message (inside the Outer Message Body). The media type of the wrapper MUST be "message/rfc822" and MUST contain the Content-Type header field parameter "forwarded=no" as defined in [[I-D.melnikov-iana-reg-forwarded](#)]. The wrapper unambiguously delimits the Inner Message from the rest of the message.

[4.1.4.](#) Outer Message Header Fields

To maximize Privacy, it is strongly RECOMMENDED to follow the principle of Data Minimization (cf. [Section 2.1](#)).

However, the Outer Message Header Section SHOULD contain the Essential Header Fields and, in addition, MUST contain the Header Fields of the MIME Header Section part to describe the encryption or signature as per [[RFC8551](#)].

The following Header Fields are defined as the Essential Header Fields:

- o From
- o To (if present in the Original Message)
- o Cc (if present in the Original Message)
- o Bcc (if present in the Original Message, see also [Section 4.1.2](#) and [Appendix A.1](#))
- o Date
- o Message-ID
- o Subject

Further processing by the Submission Entity normally depends on part of these Header Fields, e.g. From and Date HFs are required by

[[RFC5322](#)]. Furthermore, not including certain Header Fields may trigger spam detection to flag the message and/or lead to user experience (UX) issues.

For further Data Minimization, the value of the Subject Header Field SHOULD be obfuscated. In addition, the value of other Essential Header Fields MAY be obfuscated. Further Header Fields MAY be obfuscated, though simply not adding those to the Outer Message Header Section SHOULD be preferred over obfuscation. Header Field obfuscation is further specified in [Section 4.1.4.1](#). Header Fields not obfuscated should contain the same values as in the Original Message.

The MIME Header Section part is the collection of MIME Header Fields describing the following MIME structure as defined in [[RFC2045](#)]. A MIME Header Section part typically includes the following Header Fields:

- o Content-Type
- o Content-Transfer-Encoding
- o Content-Disposition

The following example shows the MIME Header Section part of an S/MIME signed message (using application/pkcs7-mime with SignedData):

```
MIME-Version: 1.0
Content-Type: application/pkcs7-mime; smime-type=signed-data;
             name=smime.p7m
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename=smime.p7m
```

Depending on the scenario, further Header Fields MAY be exposed in the Outer Message Header Section, which is NOT RECOMMENDED unless justified. Such Header Fields may include e.g.:

- o References
- o Reply-To
- o In-Reply-To

4.1.4.1. Obfuscation of Outer Message Header Fields

If the values of the following Outer Message Header Fields are obfuscated, those SHOULD assume the following values:

- * Subject: ...
- * Message-ID: <new randomly generated Message-ID>
- * Date: Thu, 01 Jan 1970 00:00:00 +0000 (UTC)

[[TODO: Consider alternatives for Date e.g. set to Monday 9am of the same week. The Impact of obfuscated Date HF content to certificate validation is for further study, in particular regarding legacy clients.]]

In certain implementations also the From, To, and/or Cc Header Field MAY be obfuscated. Those may be replaced by e.g.

- o To: Obfuscated <anonymous@anonymous.invalid>

Such implementations may need to ensure that the Submission Entity has access to the content of these Header Fields in clear text and is capable of processing those. This is particularly relevant, if proprietary Submission Entities are used.

A use case for obfuscation of all Outer Message Header Fields is routing email using onion routing or mix networks (e.g. [pEp.mixnet]).

Note: It is for further study to what extent Header Field obfuscation adversely impacts spam filtering.

4.1.5. Receiving User Facing Message Header Fields

The Receiving User Facing Message SHOULD be a verbatim copy of the Inner Message.

4.1.6. Header Field Flow

The Following figure depicts the different message representations (OrigM, InnerM, OuterM, RUFM) and which parts those are constructed from:

OrigM	InnerM	Outer(S)	OuterM(R)	RUFM
			<Trace-HF>	
		From (OrigM)	= From	
		To (OrigM)	= To	
		Cc (OrigM)	= Cc	
		Bcc (OrigM)	= Bcc*	
		Date (OrigM)	= Date	
		Message-ID (OrigM)	= Message-ID	
		Subject (new)	= Subject	
		<MIME-HSp> (new)	= <MIME-HSp>	
		PROTECTED:	PROTECTED:	
		<Wrapper> (new)	= <Wrapper>	
From	> From	> From	= From	> From
To	> To	> To	= To	> To
Cc*	> Cc	> Cc	= Cc	> Cc
Bcc*				
Date	> Date	> Date	= Date	> Date
Message-ID	> Message-ID	> Message-ID	= Message-ID	> Message-ID
Subject	> Subject	> Subject	= Subject	> Subject
<More HF>	> <More HF>	> <More HF>	= <More HF>	> <More-HF>
<MIME-HSp>	> <MIME-HSp>	> <MIME-HSp>	= <MIME-HSp>	> <MIME-HSp>
<Body>	> <Body>	> <Body>	= <Body>	> <Body>
		<Signature>* (new)	= <Signature>	

Legend:

- o OuterM(S): Outer Message (OuterM) at sending side (before handing it over to the Submission Entity)
- o OuterM(R): Outer Message at receiving side (as received by the last hop, before decryption and/or signature verification is applied to)
- o InnerM: Inner Message (that protection is applied to)
- o RUFM: Receiving User Facing Message
- o More-HF: Additional Header Fields (HF) in the Original Message (OrigM)
- o Wrapper: MIME Header Section; with media type (message/rfc822) to unambiguously delimit the inner message from the rest of the message.

- o MIME-HSp: MIME Header Section part to describe the encryption or signature as per [[RFC8551](#)]
- o Trace-HF: Header Fields added in Transit (between sending and receiving side) as per [[RFC5322](#)]
- o >: taken over / copied from last column
- o =: propagates unchanged, unless something unusual (e.g. attack) happens
- o *: HF that is often not present (also further HFs, e.g. To, may not be present). If a HF is not present, naturally it can neither be taken over nor propagated.
- o (new) / (OrigM): HF or MIME-HSp is generated depending on the decision in [Section 4.1.7.1](#), while '(new)' / '(OrigM)' designate the default.

[4.1.7.](#) Sending Side Message Processing

For a protected message the following steps are applied before a message is handed over to the Submission Entity:

[4.1.7.1.](#) Step 1: Decide on Protection Level and Information Disclosure

The entity applying protection to a message must decide:

- o Which Protection Level (signature and/or encryption) is applied to the message? This depends on user request and/or local policy as well as availability of cryptographic keys.
- o Which Header Fields of the Original Message shall be part of the Outer Message Header Section? This typically depends on local policy. By default the Essential Header Fields are part of the Outer Message Header Section; cf. [Section 4.1.4](#).
- o Which of these Header Fields are to be obfuscated? This depends on local policy and/or specific Privacy requirements of the user. By default only the Subject Header Field is obfuscated; cf. [Section 4.1.4.1](#).

[4.1.7.2.](#) Step 2: Compose the Outer Message Header Section

Depending on the decision in [Section 4.1.7.1](#), compose the Outer Message Header Section. (Note that this also includes the necessary MIME Header Section part for the following protection layer.)

Outer Header Fields that are not obfuscated should contain the same values as in the Original Message (except for MIME Header Section part, which depends on the Protection Level selected in [Section 4.1.7.1](#)).

[4.1.7.3](#). Step 3: Apply Protection to the Original Message

Depending on the Protection Level selected in [Section 4.1.7.1](#), apply signature and/or encryption to the Original Message, including the wrapper (as per [\[RFC8551\]](#)), and set the result to the message as Outer Message Body.

The resulting (Outer) Message is then typically handed over to the Submission Entity.

[[TODO: Example]]

[4.1.8](#). Receiving Side Message Processing

When a protected message is received, the following steps are applied:

[4.1.8.1](#). Step 1: Decrypt message and/or check signature

Depending on the Protection Level, the received message is decrypted and/or its signature is checked as per [\[RFC8551\]](#).

[4.1.8.2](#). Step 2: Construct the Receiving User Facing Message

The Receiving User Facing Message is constructed according to [Section 4.1.5](#).

The resulting message is handed over for further processing, which typically involves rendering it for the user.

Note: Further study is needed to determine whether or not the Outer Message Header Section, as received from the last hop, is preserved for the user, and if so, how this is to be achieved.

[4.2](#). Backward Compatibility Use Cases

[4.2.1](#). Receiving Side MIME-Conformant

This section applies to the case where the sending side (fully) supports Header Protection as specified in this document, while the receiving side does not support this specification, but is MIME-conformant according to [\[RFC2045\]](#), ff. (cf. [Section 3.1.2](#)) and [Section 3.1.2.1](#))

The sending side specification of the main use case (cf. [Section 4.1](#)) MUST ensure that receiving sides can still recognize and display or offer to display the encapsulated data in accordance with its media type (cf. [\[RFC2049\]](#), [Section 2](#)). In particular, receiving sides that do not support this specification, but are MIME-conformant according to [\[RFC2045\]](#), ff. can still recognize and display the Message intended for the user.

[[TODO: Verify once solution is stable and update last sentence.]]

[4.2.2](#). Receiving Side Not MIME-Conformant

This section applies to the case where the sending side (fully) supports Header Protection as specified in this document, while the receiving side neither supports this specification **nor** is MIME-conformant according to [\[RFC2045\]](#), ff. (cf. [Section 3.1.2](#) and [Section 3.1.2.2](#)).

[I-D.autocrypt-lamps-protected-headers] describes a possible way to achieve backward compatibility with existing S/MIME (and PGP/MIME) implementations that predate this specification and are not MIME-conformant (Legacy Display) either. It mainly focuses on email clients that do not render emails using header protection (in a user friendly manner) and may confuse the user. While this has been observed occasionally in PGP/MIME (cf. [\[RFC3156\]](#)), the extent of this problem with S/MIME implementations is still unclear. (Note: At this time, none of the samples in [\[I-D.autocrypt-lamps-protected-headers\]](#) apply header protection as specified in [Section 3.1 of \[RFC8551\]](#), which is wrapping as Media Type "message/RFC822".)

Should serious backward compatibility issues with rendering at the receiver be discovered, the Legacy Display format described in [\[I-D.autocrypt-lamps-protected-headers\]](#) may serve as a basis to mitigate those issues (cf. [Section 4.2](#)).

Another variant of backward compatibility has been implemented by pEp [\[I-D.pep-email\]](#), i.e. pEp Email Format 1.0. At this time pEp has implemented this for PGP/MIME, but not yet S/MIME.

[5](#). Security Considerations

[[TODO]]

6. Privacy Considerations

[[TODO]]

7. IANA Considerations

This document requests no action from IANA.

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

8. Acknowledgments

The authors would like to thank the following people who have provided helpful comments and suggestions for this document: Berna Alp, Claudio Luck, Daniel Kahn Gillmor, David Wilson, Hernani Marques, juga, Krista Bennett, Kelly Bristol, Lars Rohwedder, Robert Williams, Sofia Balicka, Steve Kille, Volker Birk, and Wei Chuang.

9. References

9.1. Normative References

- [I-D.ietf-lamps-header-protection-requirements]
Melnikov, A. and B. Hoeneisen, "Problem Statement and Requirements for Header Protection", [draft-ietf-lamps-header-protection-requirements-01](#) (work in progress), October 2019.
- [RFC2045] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", [RFC 2045](#), DOI 10.17487/RFC2045, November 1996, <<https://www.rfc-editor.org/info/rfc2045>>.
- [RFC2046] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", [RFC 2046](#), DOI 10.17487/RFC2046, November 1996, <<https://www.rfc-editor.org/info/rfc2046>>.
- [RFC2049] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Five: Conformance Criteria and Examples", [RFC 2049](#), DOI 10.17487/RFC2049, November 1996, <<https://www.rfc-editor.org/info/rfc2049>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

- [RFC5322] Resnick, P., Ed., "Internet Message Format", [RFC 5322](#), DOI 10.17487/RFC5322, October 2008, <<https://www.rfc-editor.org/info/rfc5322>>.
- [RFC8551] Schaad, J., Ramsdell, B., and S. Turner, "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 4.0 Message Specification", [RFC 8551](#), DOI 10.17487/RFC8551, April 2019, <<https://www.rfc-editor.org/info/rfc8551>>.

9.2. Informative References

- [I-D.autocrypt-lamps-protected-headers]
Einarsson, B., juga, j., and D. Gillmor, "Protected Headers for Cryptographic E-mail", [draft-autocrypt-lamps-protected-headers-02](#) (work in progress), December 2019.
- [I-D.melnikov-iana-reg-forwarded]
Melnikov, A. and B. Hoeneisen, "IANA Registration of Content-Type Header Field Parameter 'forwarded'", [draft-melnikov-iana-reg-forwarded-00](#) (work in progress), November 2019.
- [I-D.pep-email]
Marques, H., "pretty Easy privacy (pEp): Email Formats and Protocols", [draft-pep-email-00](#) (work in progress), July 2020.
- [pEp.mixnet]
pEp Foundation, "Mixnet", June 2020, <<https://dev.pep.foundation/Mixnet>>.
- [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>>.
- [RFC4949] Shirey, R., "Internet Security Glossary, Version 2", FYI 36, [RFC 4949](#), DOI 10.17487/RFC4949, August 2007, <<https://www.rfc-editor.org/info/rfc4949>>.
- [RFC6376] Crocker, D., Ed., Hansen, T., Ed., and M. Kucherawy, Ed., "DomainKeys Identified Mail (DKIM) Signatures", STD 76, [RFC 6376](#), DOI 10.17487/RFC6376, September 2011, <<https://www.rfc-editor.org/info/rfc6376>>.
- [RFC6409] Gellens, R. and J. Klensin, "Message Submission for Mail", STD 72, [RFC 6409](#), DOI 10.17487/RFC6409, November 2011, <<https://www.rfc-editor.org/info/rfc6409>>.

[RFC7208] Kitterman, S., "Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1", [RFC 7208](#), DOI 10.17487/RFC7208, April 2014, <<https://www.rfc-editor.org/info/rfc7208>>.

[RFC7489] Kucherawy, M., Ed. and E. Zwicky, Ed., "Domain-based Message Authentication, Reporting, and Conformance (DMARC)", [RFC 7489](#), DOI 10.17487/RFC7489, March 2015, <<https://www.rfc-editor.org/info/rfc7489>>.

Appendix A. Additional information

A.1. Stored Variants of Messages with Bcc

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

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

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

Appendix B. Document Changelog

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

- o [draft-ietf-lamps-header-protection-00](#)
 - * Initial version (text partially taken over from [\[I-D.ietf-lamps-header-protection-requirements\]](#))

Appendix C. Open Issues

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

- o Ensure "protected header" (Ex-Memory-Hole) option is (fully) compliant with the MIME standard, in particular also [\[RFC2046\]](#), [Section 5.1](#). (Multipart Media Type) [Section 4.1.1.2](#).
- o Decide on format of obfuscated HFs, in particular Date HF ([Section 4.1.4.1](#))
- o Impact on spam filtering, if HFs are obfuscated ([Section 4.1.4.1](#))
- o More examples (e.g. in [Section 4.1.7](#))
- o Should Outer Message Header Section (as received) be preserved for the user? ([Section 4.1.8.2](#))
- o Decide on whether or not merge requirements from [\[I-D.ietf-lamps-header-protection-requirements\]](#) into this document.
- o Decide what parts of [\[I-D.autocrypt-lamps-protected-headers\]](#) to merge into this document.
- o Enhance Introduction [Section 1](#) and Problem Statement ([Section 2](#)).
- o Decide on whether or not specification for more legacy HP requirements should be added to this document ([Section 3.1.2](#)).
- o Verify simple backward compatibility case (Receiving Side MIME-Conformant) is working; once solution is stable and update paragraphs in [Section 4.1](#), [Section 3.1.2.1](#) and [Section 4.2.1](#) accordingly.
- o Verify ability to distinguish between Messages with Header Protection as specified in this document and legacy clients and update [Section 3.1](#) accordingly.

- o Improve definitions of Protection Levels and enhance list of Protection Levels ([Section 3.2](#), [Section 4](#)).
- o Privacy Considerations [Section 6](#)
- o Security Considerations [Section 5](#)

Authors' Addresses

Bernie Hoeneisen
pEp Foundation
Oberer Graben 4
CH-8400 Winterthur
Switzerland

Email: bernie.hoeneisen@pep.foundation
URI: <https://pep.foundation/>

Alexey Melnikov
Isode Ltd
14 Castle Mews
Hampton, Middlesex TW12 2NP
UK

Email: alexey.melnikov@isode.com

