Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 9, 2020

A. Melnikov
Isode Ltd
B. Hoeneisen
Ucom.ch

July 08, 2019

Problem Statement and Requirements for Header Protection
draft-ietf-lamps-header-protection-requirements-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 issue has been expressed in the IETF LAMPS Working Group only recently. The existing S/MIME specification is likely to be updated regarding header protection.

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

This document describes the problem statement, generic use cases, and requirements. Additionally it drafts possible solutions to address the challenge. Finally some best practices are collected.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

[[Note: Please be advised that this document is early work-in-progress, and will substantially change in future revisions.]]

A range of protocols for the protection of electronic mail (email) exist, which allow to assess the authenticity and integrity of the email headers section or selected header fields 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 headers section and are not capable of providing privacy for the information contained therein.

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

A standard for end-to-end protection of the email headers section exists for S/MIME since version 3.1. (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 has been standardized for PGP (Pretty Good Privacy) yet.

End-to-end protection for the email headers section is currently not widely implemented - neither for messages protected by means of S/MIME nor PGP. At least two variants of header protection are known to be implemented.

This document describes the problem statement, generic use cases (Section 3) and requirements for header protection (Section 4). Additionally it drafts possible solutions to address the challenge. However, the final solution will be determined by the IETF LAMPS WG. Finally, some best practices are collected.

[[TODO: enhance this section]]

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:

- Header Field:: cf. [RFC5322]

- Header Section: cf. [RFC5322]

- Signed-only message: a multipart/signed or application/pkcs7-mime containing SignedData message which doesn't contain any encrypted layer. I.e. this is a message which is not encrypted and not encrypted + signed.

- 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."

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.

[[TODO: enhance this section]]

3. Use Cases

In the following, we show the generic use cases that need to be addressed independently of whether S/MIME, PGP/MIME or any other technology is used for which Header Protection (HP) is to be applied to.

3.1. Interactions

The main interaction case for Header Protection (HP) is:

1) Both peers (sending and receiving side) fully support HP

For backward compatibility of legacy clients - unaware of any HP - the following intermediate interactions need to be considered as well:

2) The sending side fully supports HP, while the receiving side does not support any HP

3) The sending side does not support any HP, while the receiving side fully supports HP (trivial case)

4) Neither the sending side nor the receiving side supports any HP (trivial case)

The following intermediate use cases may need to be considered as well for backward compatibility with legacy HP systems, such as S/MIME since version 3.1 (cf. [RFC8551]), in the following designated as legacy HP:

5) The sending side fully supports HP, while the receiving side supports legacy HP only

6) The sending side supports legacy HP only, while the receiving side fully supports HP

7) Both peers (sending and receiving side) support legacy HP only

8) The sending side supports legacy HP only, while the receiving side does not support any HP

9) The sending side does not support any HP, while the receiving side supports legacy HP only (trivial case)

Note: It is to be decided whether to ensure legacy HP systems do not conflict with any new solution for HP at all or whether (and to which degree) backward compatibility to legacy HP systems shall be maintained.

[[TODO: Decide in which form legacy HP requirements should remain in this document.]]

3.2. Protection Levels

The following protection levels need to be considered:

a) signature and encryption

b) signature only

c) encryption only [[TODO: verify whether relevant]]

4. Requirements

In the following a list of requirements that need to be addressed independently of whether S/MIME, PGP/MIME or any other technology is used to apply HP to.

4.1. General Requirements

This subsection is listing the requirements to address use case 1) (cf. Section 3.1).

G1: Define the format for HP for all protection levels MIME structure, Content-Type (including all parameters, such as "charset" and "name"), Content-Disposition (including all parameters, such as "filename"), and Content-Transfer-Encoding.

G2: To foster wide implementation of the new solution, it shall be easily implementable. Unless needed for maximizing protection and privacy, existing implementations shall not require substantial changes in the existing code base. In particular also MIME libraries widely used shall not need to be changed to comply with the new mechanism for HP.

G3: There SHOULD be only one format that covers all Protection Levels (cf. {{protection-levels}}))

[[TODO: Should this one remain in the document?
   If yes, consider improve / rewrite sentence
]]

G4: Ensure that man-in-the-middle attack (MITM) cf. {{RFC4949}}, in particular downgrade attacks, are mitigated as good as possible.

4.1.1. Sending Side

GS1: Determine which Header Fields (HFs) should or must be protected at least for signed only email.

GS2: Determine which HFs should or must be sent in clear of an encrypted email.

GS3: Determine which HF should not or must not be included in the visible header (for transport) of an encrypted email, with the default being that whatever is not needed from GS2 is not put into the unencrypted transport headers, thus fulfilling data minimization requirements (including data spareness and hiding of all information that technically can be hidden).

GS4: Determine which HF to not to include to any HP part (e.g. Bcc).

4.1.2. Receiving Side

GR1: Determine how HF should be displayed to the user in case of conflicting information between the protected and unprotected headers.

GR2: Ensure that man-in-the-middle attack (MITM) cf. {{RFC4949}}, in particular downgrade attacks, can be detected.

4.2. Additional Requirements for Backward-Compatibility With Legacy Clients Unaware of Header Protection

This sub-section addresses the use cases 2) - 4) (cf. Section 3.1)

B1: Depending on the solution, define a means to distinguish between forwarded messages and encapsulated messages using new HP mechanism.

4.2.1. Sending side

BS1: Define how full HP support can be indicated to outgoing messages.

BS2: Define how full HP support of the receiver can be detected or guessed.

BS3: Ensure a HP unaware receiving side easily can display the "Subject" HF to the user.

4.2.2. Receiving side

BR1: Define how full HP support can be detected in incoming messages.

4.3. Additional Requirements for Backward-Compatibility with Legacy Header Protection Systems (if supported)

This sub-section addresses the use cases 5) - 9) (cf. Section 3.1).

LS1: Depending on the solution, define a means to distinguish between forwarded messages, legacy encapsulated messages, and encapsulated messages using new HP mechanism.

LS2: The solution should be backward compatible to existing solutions and aim to minimize the implementation effort to include support for existing solutions.

4.3.1. Sending Side

LSS1: Determine how legacy HP support can be indicated to outgoing messages.

LSS2: Determine how legacy HP support of the receiver can be detected or guessed.

4.3.2. Receiving Side

LSR1: Determine how legacy HP support can be detected in incoming messages.

5. Options to Achieve Header Protection

In the following a set of Options to achieve Email Header Protection. It is expected that the IETF LAMPS WG chooses an option to update [RFC8551] wrt. Header Protection.

5.1. Option 1: Memory Hole

Memory Hole approach works by copying the normal message header fields into the MIME header section of the top level protected body part. Since the MIME body part header section is itself covered by the protection mechanisms (signing and/or encryption) it shares the protections of the message body.

[[TODO: add more information on memory hole]]

5.2. Option 2: Wrapping with message/rfc822 or message/global

Wrapping with message/rfc822 (or message/global) works by copying the normal message header fields into the MIME header section of the top level protect body part

[[TODO: consider rephrasing, as not only the header fields is copied, but also the content.]]

and then prepending them with "Content-Type: message/rfc822; forwarded=no\r\n" or "Content-Type: message/global; forwarded=no\r\n", where \r\n is US-ASCII CR followed by US-ASCII LF. Since the MIME body part header section is itself covered by the protection mechanisms (signing and/or encryption) it shares the protections of the message body.

5.2.1. Content-Type Parameter "forwarded"

This section outlines how the new "forwarded" Content-Type header field parameter could be defined (probably in a separate document) and how header section wrapping works:

This document defines a new Content-Type header field parameter [RFC2045] with name "forwarded". The parameter value is case-insensitive and can be either "yes" or "no". (The default value being "yes"). The parameter is only meaningful with media type "message/rfc822" and "message/global" [RFC6532] when used within S/MIME signed or encrypted body parts. The value "yes" means that the message nested inside "message/rfc822" ("message/global") is a forwarded message and not a construct created solely to protect the inner header section.

Instructions in [RFC8551] describing how to protect the Email message header section [RFC5322], by wrapping the message inside a message/rfc822 container [RFC2045] are thus updated to read:

> 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. It is up to the receiving client to decide how to present this "inner" header section along with the unprotected "outer" header section.
>
> When an S/MIME message is received, if the top-level protected MIME entity has a Content-Type of message/rfc822 or message/global without the "forwarded" parameter or with the "forwarded" parameter set to "no", it can be assumed that the intent was to provide header protection. This entity SHOULD be presented as the top-level message, taking into account header section merging issues as previously discussed.

5.3. Option 2.1: Progressive Header Disclosure

This option is similar to Option 2 (cf. Section 5.2). It also makes use the Content-Type parameter "forwarded" (cf. Section 5.2.1).

pEp for email [I-D.marques-pep-email] defines a fixed MIME structure for its innermost message structure. Security comes just next after privacy in pEp, for which reason the application of signatures without encryption to messages in transit is not considered purposeful. pEp for email, either expects to transfer messages in cleartext without signature or encryption, or transfer them encrypted

and with enclosed signature and necessary public keys so that replies can be immediately upgraded to encrypted messages.

The pEp message format is equivalent to the S/MIME standard in ensuring header protection, in that the whole message is protected instead, by wrapping it and providing cryptographic services to the whole original message. However, for the purpose of allowing the insertion of public keys, the root entity of the protected message is thus nested once more into an additional multipart/mixed MIME entity. The current pEp proposal is for PGP/MIME, while an extension to S/MIME is also on the roadmap.

pEp has also implemented the above (in Section 5.2.1) described Content-Type parameter "forwarded" to distinguish between encapsulated and forwarded emails.

More information on progressive header disclosure can be found in [I-D.luck-lamps-pep-header-protection].

5.4. Examples

Examples in subsequent sections assume that an email client is trying to protect (sign) the following initial message:

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

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

Without message header protection the corresponding signed message might look like this. (Lines prepended by "O: " are the outer header.)

```
  O: Date: Mon, 25 Sep 2017 17:31:42 +0100 (GMT Daylight Time)
  O: Message-ID: <e4a483cb-1dfb-481d-903b-298c92c21f5e@matt.example.net>
  O: Subject: Meeting at my place
  O: From: "Alexey Melnikov" <alexey.melnikov@example.net>
  O: MIME-Version: 1.0
  O: content-type: multipart/signed; charset=us-ascii; micalg=sha1;
  O:  protocol="application/pkcs7-signature";
  O:  boundary=.cbe16d2a-e1a3-4220-b821-38348fc97237

     This is a multipart message in MIME format.
     --.cbe16d2a-e1a3-4220-b821-38348fc97237
     Content-Type: text/plain; charset=us-ascii

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

     --.cbe16d2a-e1a3-4220-b821-38348fc97237
     Content-Transfer-Encoding: base64
     content-type: application/pkcs7-signature

     [[base-64 encoded signature]]

     --.cbe16d2a-e1a3-4220-b821-38348fc97237--
```

5.4.1. Option 1: Memory Hole

The following example demonstrates how header section and payload of a protect body part might look like. For example, this will be the first body part of a multipart/signed message or the signed and/or encrypted payload of the application/pkcs7-mime body part. Lines

prepended by "O: " are the outer header section. Lines prepended by "I: " are the inner header section.

```
O: Date: Mon, 25 Sep 2017 17:31:42 +0100 (GMT Daylight Time)
O: Message-ID: <e4a483cb-1dfb-481d-903b-298c92c21f5e@matt.example.net>
O: Subject: Meeting at my place
O: From: "Alexey Melnikov" <alexey.melnikov@example.net>
O: MIME-Version: 1.0
O: content-type: multipart/signed; charset=us-ascii; micalg=sha1;
O:  protocol="application/pkcs7-signature";
O:  boundary=.cbe16d2a-e1a3-4220-b821-38348fc97237

   This is a multipart message in MIME format.
   --.cbe16d2a-e1a3-4220-b821-38348fc97237
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@matt.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.

   --.cbe16d2a-e1a3-4220-b821-38348fc97237
   Content-Transfer-Encoding: base64
   content-type: application/pkcs7-signature

   [[base-64 encoded signature]]

   --.cbe16d2a-e1a3-4220-b821-38348fc97237--
```

5.4.2. Option 2: Wrapping with message/rfc822 or message/global

The following example demonstrates how header section and payload of a protect body part might look like. For example, this will be the first body part of a multipart/signed message or the signed and/or encrypted payload of the application/pkcs7-mime body part. Lines prepended by "O: " are the outer header section. Lines prepended by "I: " are the inner header section. Lines prepended by "W: " are the wrapper.

```
  O: Date: Mon, 25 Sep 2017 17:31:42 +0100 (GMT Daylight Time)
  O: Message-ID: <e4a483cb-1dfb-481d-903b-298c92c21f5e@matt.example.net>
  O: Subject: Meeting at my place
  O: From: "Alexey Melnikov" <alexey.melnikov@example.net>
  O: MIME-Version: 1.0
  O: content-type: multipart/signed; charset=us-ascii; micalg=sha1;
  O:  protocol="application/pkcs7-signature";
  O:  boundary=.cbe16d2a-e1a3-4220-b821-38348fc97237

     This is a multipart message in MIME format.
     --.cbe16d2a-e1a3-4220-b821-38348fc97237
  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@matt.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.

     --.cbe16d2a-e1a3-4220-b821-38348fc97237
     Content-Transfer-Encoding: base64
     content-type: application/pkcs7-signature

     [[base-64 encoded signature]]

     --.cbe16d2a-e1a3-4220-b821-38348fc97237--
```

5.4.3. Option 2.1 Progressive Header Disclosure

This looks similar as in option 2. Specific examples can be found in [I-D.luck-lamps-pep-header-protection].

[[TODO: include an example of the same style.]]

6. Sending Side Considerations

6.1. Candidate Header Fields for Header Protection

[[TODO: This section is very early stage and needs more work.]]

For a signed-only message, it is RECOMMENDED that all "outer" header fields are identical to the "inner" protected header fields. This would mean that all header fields are signed. In this case, the "outer" header fields simply match the protected header fields. And in the case that the "outer" header fields differ, they can simply be replaced with their protected versions when displayed to the user.

[[TODO: Decide whether "Bcc" header field should be excluded. Also verify whether this requirement applies generally or just for specific implementations.]]

When generating encrypted or encrypted+signed S/MIME messages which protect header fields:

1. If a header field is being encrypted because it is sensitive, its true value MUST NOT be included in the outer header. If the header field is mandatory according to [RFC5322], a stub value (or a value indicating that the outer value is not to be used) is to be included in the outer header section.

2. The outer header section SHOULD be minimal in order to avoid disclosure of confidential information. It is recommended that the outer header section only contains "Date" (set to the same value as in the inner header field, or, if the Date value is also sensitive, to Monday 9am of the same week), possibly "Subject" and "To"/"Bcc" header fields. ("From", "Date", and at least one destination header field is mandatory as per [RFC5322].) In particular, Keywords, In-Reply-To and References header fields SHOULD NOT be included in the outer header; "To" and "Cc" header fields should be omitted and replaced with "Bcc: undisclosed-recipients;".

   But note that having key header fields duplicated in the outer header is convenient for many message stores (e.g. IMAP) and clients that can't decode S/MIME encrypted messages. In particular, Subject/To/Cc/Bcc/Date header field values are returned in IMAP ENVELOPE FETCH data item [RFC3501], which is frequently used by IMAP clients in order to avoid parsing message header.

3. The "Subject" header field value of the outer header section SHOULD either be identical to the inner "Subject" header field value, or contain a clear indication that the outer value is not to be used for display (the inner header field value would contain the true value).

Note that recommendations listed above typically only apply to non MIME header fields (header fields with names not starting with "Content-" prefix), but there are exception, e.g. Content-Language.

Note that the above recommendations can also negatively affect anti-spam processing.

7. Receiving Side Considerations

7.1. Which Header Fields to Display to User

When displaying S/MIME messages which protect header fields (whether they are signed-only, encrypted or encrypted+signed):

1. The outer header fields might be tampered with, so a receiving client SHOULD ignore them, unless they are protected in some other way(_). If a header field is present in the inner header, only the inner header field value MUST be displayed (and the corresponding outer value must be ignored). If a particular header field is only present in the outer header, it MAY be ignored (not displayed) or it MAY be displayed with a clear indicator that it is not trustworthy(_).

   (*) - this only applies if the header field is not protected is some other way, for example with a DKIM signature that validates and is trusted.

7.2. Mail User Agent Algorithm for deciding which version of a header field to display

[[TODO: describe how to recurse to find the innermost protected root body part, extract header fields from it and propagate them to the top level. This should also work for triple-wrapped messages.]]

8. Security Considerations

This document talks about UI considerations, including security considerations, when processing messages protecting header fields. One of the goals of this document is to specify UI for displaying such messages which is less confusing/misleading and thus more secure.

The document is not defining new protocol, so it doesn't create any new security concerns not already covered by S/MIME [RFC8551], MIME [RFC2045] and Email [RFC5322] in general.

9. Privacy Considerations

[[TODO]]

10. IANA Considerations

This document requests no action from IANA.

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

11. Acknowledgments

The authors would like to thank the following people who have provided helpful comments and suggestions for this document: David Wilson, Steve Kille, Wei Chuang, and Robert Williams

Essential parts of [I-D.luck-lamps-pep-header-protection] have been merged into this document. Special thanks to its author Claudio Luck. For further Acknowledgments, please refer to Acknowledgments section of [I-D.luck-lamps-pep-header-protection].

David Wilson came up with the idea of defining a new Content-Type header field parameter to distinguish forwarded messages from inner header field protection constructs.

12. References

12.1. Normative References

[RFC2045] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, DOI 10.17487/RFC2045, November 1996, <https://www.rfc-editor.org/info/rfc2045>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[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>.

12.2. Informative References

[I-D.luck-lamps-pep-header-protection]
Luck, C., "pretty Easy privacy (pEp): Progressive Header Disclosure", draft-luck-lamps-pep-header-protection-03 (work in progress), July 2019.

[I-D.marques-pep-email]
Marques, H., "pretty Easy privacy (pEp): Email Formats and Protocols", draft-marques-pep-email-02 (work in progress), October 2018.

[RFC3501] Crispin, M., "INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1", RFC 3501, DOI 10.17487/RFC3501, March 2003, <https://www.rfc-editor.org/info/rfc3501>.

[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>.

[RFC6532] Yang, A., Steele, S., and N. Freed, "Internationalized Email Headers", RFC 6532, DOI 10.17487/RFC6532, February 2012, <https://www.rfc-editor.org/info/rfc6532>.

[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. Document Changelog

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

- draft-ietf-lamps-header-protection-requirements-00
  - Initial version

Appendix B. Open Issues

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

- Enhance Introduction and Problem Statement sections

- Decide in which form legacy HP requirements should remain in this document

- Signed-only protection needs further study

  * pEp only does header protection by applying both signing and encryption. Technically it is also possible to sign, but not encrypt the protected messages. This needs further study. Feedback from IETF-104: Probably no need to specify it, but need to document the case.

- Should requirement G3 remain? If you consider improve / rewrite it.

- Add more text on Memory Hole

- Rephrase Section 5.2

- Add example to Section 5.4.3

- Resolve question regarding Bcc in Section 6.1

- Rewrite Section 6.1

- Write Section 7.2

- Correct terminology for Header(s) and Header Fields throughout the document (editorial).

  * Header: Whole Header Section of the message

  * Header Field: Part / single Line inside a Header (Section)

Authors' Addresses

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

Email: alexey.melnikov@isode.com

Bernie Hoeneisen
Ucom Standards Track Solutions GmbH
CH-8046 Zuerich
Switzerland

Phone: +41 44 500 52 40
Email: bernie@ietf.hoeneisen.ch (bernhard.hoeneisen AT ucom.ch)
URI: https://ucom.ch/