Header Protection (HP) in S/MIME

LAMPS @ IETF-108 / Tuesday, 28 July 2020

draft-ietf-lamps-header-protection-00

Bernie Hoeneisen / Alexey Melnikov

Main Use Case

- Both – the sending and receiving - sides (fully) support Header Protection
  - i.e. as specified in this new specification
- This should also work for receiving sides that are MIME-conformant (see next slide)

MIME Conformance on rendering of RFC822 message encapsulation

Excerpt of Section 2 of RFC 2049 on "MIME Conformance":

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.
- [...]

Backward Compatibility Use Cases

Sending side (fully) supports Header Protection as specified in this new specification, and

1) Receiving side MIME-conformant
   - According to RFC 2046, ff.
     - In particular also Section 2 of RFC 2049 (cf. previous slide)
   - Main Use Case should work for those

2) Receiving side **not** MIME-conformant
   - Clients that cause serious rendering issues for wrapped (incl. forwarded) messages

Issue - Backward Compatibility

- To what extent should the new standard accommodate implementations that are **not conformant to MIME**?
  - Or rather remind to fix their broken implementation?
  - Something in between
    - e.g. “Legacy Display” (cf. Sect. 5 of draft-autocrypt-lamps-protected-headers)

Not discussed right now!
(discussion at the end of this presentation or on the list)

Protection Levels

- Signature and encryption
  - MUST implement for both sides
  - SHOULD be *default* on sending side
- Signature only
  - SHOULD implement on sending side
  - MUST implement on receiving side
- Encryption only
  - NOT RECOMMENDED on sending side
  - MAY implement on receiving side

Issue – Protection Levels

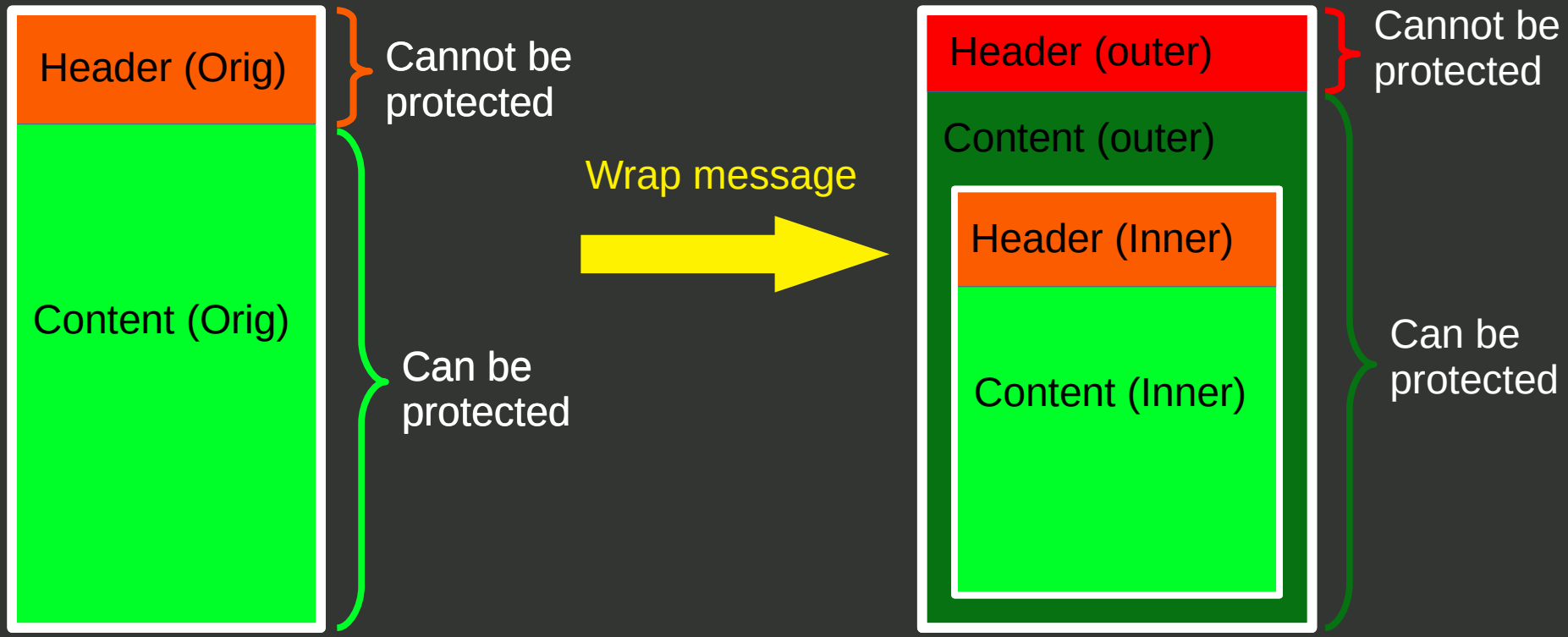
- Specification is targeted for
  - Signature and encryption
  - Signature only
- Variations / corner-cases may pop up at receiving side, e.g.
  - Encryption only
  - Encryption before signature
  - Signature and encryption, but
    - Signature fails to validate
    - Signature validates but the signing certificate revoked
  - Signature only, but
    - with multiple valid signatures, layered atop each other?
- Which of those and to what extent do we need to document those?

MIME Format (Main Use Case)

- Two proposals
  1) RFC 8551 (S/MIME 4.0)
  2) Autocrypt “Protected Headers” / “memory-hole” successor

  (separate slide with issue MIME Format below)
- Outer and Inner Message
  - Outer Message Header Section (HS)
    - Protection not possible
  - Outer Message Body
    - Protection possible
    - Contains Inner Message (HS and Body)
      - Inner Message HS same as (or a subset of) the Original Message HS
      - Inner Message Body same as the Original Message Body
- Original Message itself may contain any MIME structure.

HP in S/MIME since version 3.1



Privacy by Default.

Example

Outer Message Header Section (unprotected)

```
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
```

Outer Message Body (protected)

```
    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--
```

Wrapper (only proposal 1)

Inner Message Header Section

Inner Message Body

Signature

Privacy by Default.

Issue – MIME Format

S/MIME 4.0 / RFC 8551 (wrapping as “message/rfc822”)

+ MIME conformant
+ MIME parser proof

+ S/MIME conformant
? Possibly rendering issues with non-MIME conformant legacy MUAs
- Extra line(s) for wrapper

Autocrypt “Protected Headers” (based on “memory-hole” concept)

? MIME conformance unclear
? Existing MIME parser treatment unclear

- S/MIME to be changed
+ Reduces rendering issues with non-MIME conformant legacy MUAs
+ Slightly shorter (no wrapper)

MIME Conformance on Non-MIME Header Fields in body parts

Excerpt of Section 5.1 of RFC 2046 on "Multi Part Media Type":

> 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**. [...]

Header Fields Flow

Original Message [OrigM]		Inner Message		Outer Message (Sending Side)		Outer Message (Receiving Side)		Receiving User Facing Message
						<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>		
				<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

Trace-HF:
Header Fields added in transit

MIME-Hsp:
MIME Header Section part

>
taken over / copied from last column

=
Propagates unchanged (normally)

*
HF often not present

Privacy by Default.

Composition of Inner Message

- The Inner Message Header Section is the same as (or a subset of) the Original Message Header Section

Issue – Original Message → Inner Message

- The Inner Message is either:
  - Same as Original Message
  - Original Message without Bcc
- Depending on the (Bcc) recipient
  - cf. issue Bcc Handling

- Any other variants to consider?

Composition of Outer Message (1/2)

- Header Section (HS) SHOULD contain the “Essential” Header Fields (EHF), which are:
  - From
  - To / Cc (if present in Original Message)
  - Bcc (if present in Original Message and needed)
    - Separate slide with issue Bcc Handling below
  - Date
  - Message-ID
  - Subject
- HS also contains MIME Header Section part:
  - Content-Type, Contain-Disposition, etc.
- HS MAY contain further HFs
  - e.g., References, Reply-To, In-Reply-To

Issue – Original Message → Outer Message

- The Outer Message HS normally contains:
  - Essential Header Fields (EHF)
    - From, To, Cc, (Bcc), Date, Message-Id, Subject
  - MIME HS part
    - e.g. “multipart/signed”
  - Other HF are optional
- Depending on the (Bcc) recipient
  - cf. issue Bcc Handling below
- Any other variants to consider?

Not discussed right now!
(discussion at the end of this presentation or on the list)

Obfuscation of (Outer Message) HF

- Subject HF SHOULD be obfuscated
- Other EHFs MAY be obfuscated
- Obfuscation likely has impact to spam filtering

Issue – Obfuscation of Header Fields

- Should we recommend any specific format for obfuscation? e.g.
  - Subject: …
  - Subject: [...]
  - Date: Thu, 01 Jan 1970 00:00:00 +0000 (UTC)
    - Impact to certificate checking?
  - Date: *<set to Monday 9am of the same week>*
  - Message-ID: *<a new randomly generated Message-ID>*
  - From: Obfuscated <anonymous@anonymous.invalid>
  - To: Obfuscated <anonymous@anonymous.invalid>
- Impact to Spam filtering?

Receiving User Facing Message

- The Receiving User Facing Message (RUFM) is typically the same as the Inner Message
  - No merging of Outer Message with Inner Message HS

Issue – Outer Message → RUFM

1. The Receiving User Facing Message (RUFM) is the same as the Inner Message
   - Any other variants to consider?

2. As a consequence, the RUFM contains no information on the Outer Message HS
   - Preserving Outer Message HS might be useful, e.g. for
     - Debugging (Trace HFs)
     - Detecting attacks (HFs different)
   - Do we need to standardize a means to provide the Outer Message HS to the user?

Issue – Bcc Handling

- Encrypted Messages with Bcc need to be split:
  1) The same Message to all To and Cc recipients, without Bcc HF
  2) Message(s) to Bcc recipient(s) vary among implementations
     a) One Message per Bcc recipient
        Bcc HF contains recipient address the message is sent to
     b) The same Message for all Bcc recipients
        Bcc HF with an indication, e.g. "Undisclosed recipients"
     c) The same Message for all Bcc recipients
        without Bcc HF (same as same as 1)
- No specification on this found in S/MIME
  - 2a is most privacy-preserving, but may result in many Messages
  - 2b and 2c are easier/more efficient to handle, but leak privacy information via encryption keys and certs

Composition of Wrapper

- Simple MIME Header Section
  - Media type "message/RFC822"
    - “Here comes a nested email”
  - Precedes Inner Message (inside the Outer Message Body)
- MUST contain Content-Type header field parameter "forwarded=no"
  - To distinguish between forwarded and wrapped message
  - As proposed in draft-melnikov-iana-reg-forwarded
- Only in Proposal 1

Issue – Content-Type header field parameter forwarded

- To distinguish between forwarded and wrapped message draft-melnikov-iana-reg-forwarded proposes
  - Content-Type header field parameter "forwarded" (for message/rfc822)
- Should this be defined more broadly to cover other “message types”, e.g.
  - Forwarded
  - Wrapped (e.g. for Header Protection)
  - Rejected (non deliveries / bounces)
  - ML-hold (message to be assessed by a mailing list admin)
  - ML-discard-action (mailing list admin reply to this will discard)
  - ML-digest-item
- What format?
  - e.g., “message-type=forwarded”, “message-type=wrapped-inner”?

Questions / Discussion

Backup Slides

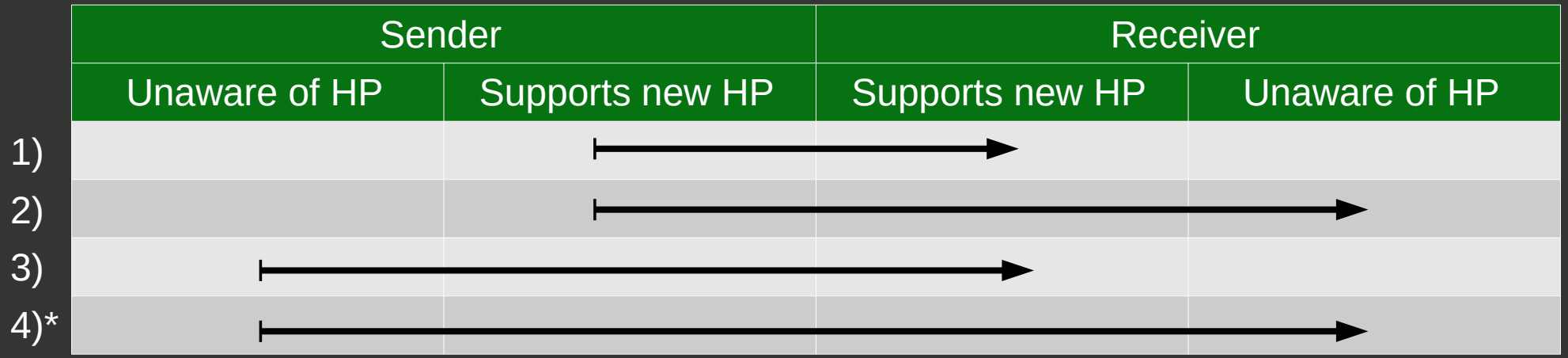
Background

- New Work Item on Header Protection (HP) to be added to the LAMPS Charter requested from IESG:

  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.

Interaction Cases (1/3)

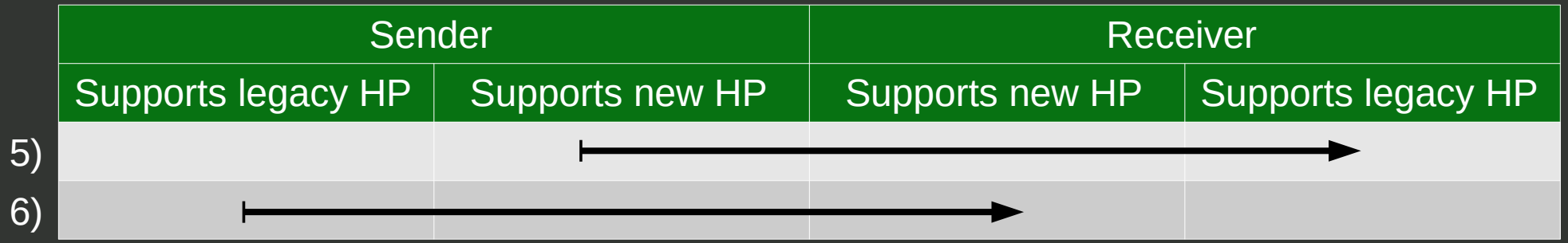
- Which interaction cases are in scope?



* trivial case

Interaction Cases (2/3)

- Which interaction cases for interoperability with legacy HP are in scope?
  - S/MIME HP since version 3.1
  - Other implementations (incl. PGP)?



Interaction Cases (3/3)

- Interactions between clients not supporting new HP
  - Probably out-of-scope
  - Though, may need to be documented

	Sender		Receiver	
	Unaware of HP	Supports legacy HP	Supports legacy HP	Unaware of HP
7)		←	→	
8)		←	→	→
9)	←		→	

Sending side processing

1. Decide
   - Protection Level (e.g. Signature & Encryption)
   - Header Fields (HFs) of Original Message to include
   - HFs to obfuscate
2. Compose Outer Message
   - HS depends on choices in 1.
3. Apply Protection
   - Depending on Protection Level choice in 1.

Resulting (Outer) Message handed over to Submission Entity

Receiving Side processing

1. Decryption and/or signature checking
2. Extract Receiving User Facing Message (RUFM)

Resulting (Inner) Message rendered to user