lamps
Internet-Draft
Intended status: Informational
Expires: 4 May 2021

D.K. Gillmor
ACLU

31 October 2020

Guidance on End-to-End E-mail Security
draft-dkg-lamps-e2e-mail-guidance-00

Abstract

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

Status of This Memo

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

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

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

This Internet-Draft will expire on 4 May 2021.

Copyright Notice

Table of Contents

1. Introduction

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

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

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

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

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 ([RFC2119] and [RFC8174]) when, and only when, they appear in all capitals, as shown here.

1.2. Terminology

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

* _MUA_ is short for Mail User Agent; an e-mail client.

* _Protection_ of message data refers to cryptographic encryption and/or signatures, providing confidentiality, authenticity, and/or integrity.

* _Cryptographic Layer_, _Cryptographic Envelope_, _Cryptographic Payload_, and _Errant Cryptographic Layer_ are defined in Section 4

* A _well-formed_ e-mail message with cryptographic protection has both a _Cryptographic Envelope_ and a _Cryptographic Payload_,

* _Structural Headers_ are documented in Section 1.2.1.

1.2.1. Structural Headers

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

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

This includes (but is not limited to):

* "Content-Type"

* "Content-Transfer-Encoding"

* "Content-Disposition"

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

2. Usability

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

so keep it simple.

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

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

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

3. Types of Protection

A given message might be:

* signed,
* encrypted, or
* both signed and encrypted.

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

4. Cryptographic MIME Message Structure

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

4.1. Cryptographic Layers

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

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

4.1.1. S/MIME Cryptographic Layers

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

4.1.1.1. S/MIME Multipart Signed Cryptographic Layer

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

This MIME layer offers authentication and integrity.

4.1.1.2. S/MIME PKCS7 signed-data Cryptographic Layer

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

This MIME layer offers authentication and integrity.

4.1.1.3. S/MIME PKCS7 enveloped-data Cryptographic Layer

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

This MIME layer offers confidentiality.

4.1.1.4. S/MIME PKCS7 authEnveloped-data Cryptographic Layer

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

This MIME layer offers confidentiality and integrity.

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

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

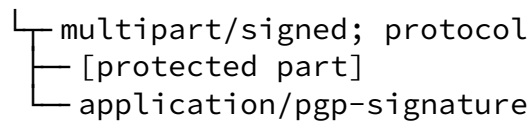
4.1.1.5. PKCS7 Compression is NOT a Cryptographic Layer

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

4.1.2. PGP/MIME Cryptographic Layers

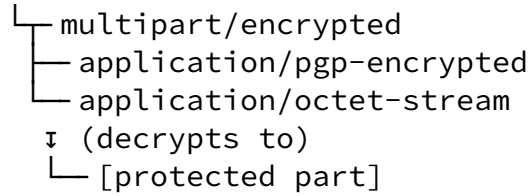
For PGP/MIME [RFC3156] there are two forms of Cryptographic Layers, signing and encryption.

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



This MIME layer offers authenticity and integrity.

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



Note that for PGP/MIME, this MIME layer can offer any of:

* confidentiality (via a Symmetrically Encrypted Data Packet, see Section 5.7 of [RFC4880]; a MUA MUST NOT generate this form due to ciphertext malleability)

* confidentiality and integrity (via a Symmetrically Encrypted Integrity Protected Data Packet (SEIPD), see section 5.13 of [RFC4880]), or

* confidentiality, integrity, and authenticity all together (by including an OpenPGP Signature Packet within the SEIPD).

4.2. Cryptographic Envelope

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

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

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

Note that if a non-Cryptographic Layer intervenes, all Cryptographic Layers within the non-Cryptographic Layer _are not_ part of the Cryptographic Envelope. They are Errant Cryptographic Layers (see Section 4.5).

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

4.3. Cryptographic Payload

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

4.4. Types of Cryptographic Envelope

4.4.1. Simple Cryptographic Envelopes

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

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

4.4.2. Multilayer Cryptographic Envelopes

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

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

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

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

4.5. Errant Crytptographic Layers

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

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

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

4.5.1. Mailing List Wrapping

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

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

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

Note that this message has no Cryptographic Envelope at all.

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

4.5.2. A Baroque Example

Consider a message with the following overcomplicated structure:

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

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

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

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

5. Message Composition

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

5.1. Message Composition Algorithm

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

The message composition algorithm takes three parameters:

* "origbody": the traditional unprotected message body as a well-formed MIME tree (possibly just a single MIME leaf part). As a well-formed MIME tree, "origbody" already has structural headers present (see Section 1.2.1).

* "origheaders": the intended non-structural headers for the message, represented here as a 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 X.509 certificate X, then encrypt to X.509 certificates X and Y"). This is a routine that accepts a MIME tree as input (the Cryptographic Payload), wraps the input in the appropriate Cryptographic Envelope, and returns the resultant MIME tree as output.

The algorithm returns a MIME object that is ready to be injected into

the mail system:

* Apply "crypto" to "origbody", yielding MIME tree "output"

* For header name "h" in "origheaders":

  - Set header "h" of "output" to "origheaders[h]"

* Return "output"

5.2. Encryption Outside, Signature Inside

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

Putting the signature inside the encryption has two advantages:

* The details of the signature remain confidential, visible only to the parties capable of decryption.

* Any mail transport agent that modifies the message is unlikely to be able to accidentally break the signature.

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

5.3. Avoid Offering Encrypted-only Messages

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

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

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

In other cases, full end-to-end confidentiality, authenticity, and

integrity are desirable.

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

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

* no end-to-end cryptographic protection

* signing-only

* signed and encrypted

5.4. Composing a Reply Message

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

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

* compose the reply with end-to-end encryption, or

* avoid including quoted text from the original message.

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

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

In this circumstance, the composing MUA SHOULD strip the quoted text

from the original message.

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

6. Message Interpretation

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

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

6.1. Rendering Well-formed Messages

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

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

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

6.2. Errant Cryptographic Layers

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

6.2.1. Errant Signing Layer

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

For example, a message with this structure:

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

Should be rendered identically to this:

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

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

6.2.1.1. Exception: Mailing List Footers

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

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

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

or:

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

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

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

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

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

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

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

6.2.2. Errant Encryption Layer

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

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

In this case, though, the MUA MUST NOT indicate in the message's

cryptographic summary that the message itself was encrypted. Such an indication could be taken to mean that other (non-encrypted) parts of the message arrived with cryptographic confidentiality.

6.2.2.1. Replying to a Message with an Errant Encryption Layer

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

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

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

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

6.3. Forwarded Messages with Cryptographic Protection

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

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

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

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

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

6.4. Signature failures

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

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

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

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

* the signature is not cryptographically valid (the math fails).

* the signature relies on suspect cryptographic primitives (e.g. over a legacy digest algorithm, or was made by a weak key, e.g., 1024-bit R.SA)

* the signature is made by a certificate which the receiving MUA does not have access to.

* the certificate that made the signature was revoked.

* the certificate that made the signature was expired at the time that the signature was made.

* the certificate that made the signature does not correspond to the author of the message.

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

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

7. Common Pitfalls and Guidelines

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

FIXME: some possible additional commentary on:

* indexing and search of encrypted messages
* managing access to cryptographic secret keys that require user interaction
* secure deletion
* inline PGP, ugh

8. IANA Considerations

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

9. Security Considerations

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

10. Document Considerations

[RFC Editor: please remove this section before publication]

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

10.1. Document History

11. Acknowledgements

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

12. References

12.1. Normative References

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

[RFC3156] Elkins, M., Del Torto, D., Levien, R., and T. Roessler, "MIME Security with OpenPGP", RFC 3156, DOI 10.17487/RFC3156, August 2001, <https://www.rfc-editor.org/info/rfc3156>.

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

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

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

12.2. Informative References

[EFAIL] "EFAIL", n.d., <https://efail.de>.

[I-D.draft-bre-openpgp-samples-01] Einarsson, B., juga, j., and D. Gillmor, "OpenPGP Example Keys and Certificates", Work in Progress, Internet-Draft, draft-bre-openpgp-samples-01, 20 December 2019, <http://www.ietf.org/internet-drafts/draft-bre-openpgp-samples-01.txt>.

[I-D.draft-dkg-lamps-samples-02] Gillmor, D., "S/MIME Example Keys and Certificates", Work in Progress, Internet-Draft, draft-dkg-lamps-samples-02,

24 December 2019, <http://www.ietf.org/internet-drafts/draft-dkg-lamps-samples-02.txt>.

[RFC3274] Gutmann, P., "Compressed Data Content Type for Cryptographic Message Syntax (CMS)", RFC 3274, DOI 10.17487/RFC3274, June 2002, <https://www.rfc-editor.org/info/rfc3274>.

[RFC4880] Callas, J., 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>.

[RFC5355] Stillman, M., Ed., Gopal, R., Guttman, E., Sengodan, S., and M. Holdrege, "Threats Introduced by Reliable Server Pooling (RSerPool) and Requirements for Security in Response to Threats", RFC 5355, DOI 10.17487/RFC5355, September 2008, <https://www.rfc-editor.org/info/rfc5355>.

[RFC6376] Crocker, D., Ed., Hansen, T., Ed., and M. Kucherawy, Ed., "DomainKeys Identified Mail (DKIM) Signatures", STD 76, RFC 6376, DOI 10.17487/RFC6376, September 2011, <https://www.rfc-editor.org/info/rfc6376>.

Appendix A. Test Vectors

FIXME: This document should contain examples of well-formed and malformed messages using cryptographic key material and certificates from [I-D.draft-bre-openpgp-samples-01] and [I-D.draft-dkg-lamps-samples-02].

It may also include example renderings of these messages.

Author's Address

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

Email: dkg@fifthhorseman.net