Internet-Draft
K. Yamamoto
IIJ Research Laboratory
Expires in six months
January, 1998

MIME Security with OpenPGP (OpenPGP/MIME)

<draft-yamamoto-openpgp-mime-00.txt>

Status of this Memo

Abstract

This memo specifies how to protect a MIME object within an OpenPGP object and how to embed the OpenPGP object into a MIME object based on security multipart.

1. Introduction

Pretty Good Privacy (PGP)[PGP] version 2 was a de facto standard of cipher programs in the Internet. So, PGP version 2 was a good candidate for security services of e-mail messages.

PGP version 2 itself has the cleartext signature format and ASCII Armor[OldPGP] to exchange protected objects by e-mail. However, these formats themselves are not suitable to be used in the context of MIME[MIME].

Several efforts were made to integrate MIME and PGP version 2. As a result, PGP/MIME[OldPGPMIME] based on security multipart[SECMULTI] was standardized.

PGP version 2 depends on patented and/or licensed cryptographic technologies. To be free from these problems and to introduce richer functionality, OpenPGP[OpenPGP] (aka PGP version 5 or 6) has been standardized. It is, thus, necessary to synchronize PGP/MIME with OpenPGP.

This memo aims to specify how to protect a MIME object within an OpenPGP object and how to embed the OpenPGP object into a MIME object based on security multipart. With this format, signature, encryption, decryption, verification, and/or combined services are available to protect security of MIME objects.

Throughout this memo, PGP object and OpenPGP object refer to objects presented in the format define in [OldPGP] and [OpenPGP], respectively. Note that the OpenPGP format is a super set of the PGP format, so OpenPGP objects includes PGP objects. Also, PGP/MIME and OpenPGP/MIME indicate the format defined in [OldPGPMIME] and in this memo, respectively.

Readers are assumed to be familiar to [OpenPGP] and [OldPGPMIME].

2. Standard Keywords

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 [KEYWORDS].

3. Design Goals

The design goals of OpenPGP/MIME are as follows:

(1) OpenPGP/MIME should bring a security mechanism to MIME objects, whose security level is the same as OpenPGP itself.

(2) OpenPGP/MIME should be more friendly to MIME-aware viewers/composers than to PGP-aware viewers/composers. This is because the number of MIME-aware viewers/composers is much larger than that of PGP-aware viewers/composers.

(3) For the signature service, it is more important to deliver a signed object itself than to protect the object. In other words, it is more important for MIME viewers to extract the object than to verify the signature.

(4) It is important to maintain backward compatibility with PGP/MIME and to encourage the migration from PGP to OpenPGP.

4. Conceptual Model

This section describes conceptual model of the OpenPGP/MIME composer and viewer. This conceptual model is descriptive and does NOT impose any restrictions or requirements on implementations.

4.1 OpenPGP/MIME composer

Composers that conform to the specification defined in this memo are called OpenPGP/MIME composers. An OpenPGP/MIME composer consists of a MIME composer and an OpenPGP engine.

The MIME composer creates MIME objects, which consist of a content header and a content body. The MIME composer cannot create PGP or OpenPGP objects. So, PGP or OpenPGP objects are opaque to the MIME composer. The MIME composer also does not care about version control for PGP and OpenPGP. However, the MIME composer may give control information to the OpenPGP engine.

The OpenPGP engine signs and/or encrypts input data (which is a MIME object), then produces a PGP or OpenPGP object. Version control for PGP and OpenPGP, including packet formats and versions for each packet type, is carried out by the OpenPGP engine. The OpenPGP engine cannot create MIME objects.

Figure 1 illustrates this conceptual model of the OpenPGP/MIME composer for the signature service.

```
        An object to be signed
                 |
                 v
          +---------------+
          | MIME composer |---------+
          +---------------+         | A MIME object to be signed*
                 |                  v
   A MIME object |            +--------------------------+
   to be signed* |            | OpenPGP signature engine |
                 |            +--------------------------+
                 v                  |
          +---------------+         |  An OpenPGP object of
          | MIME composer |<--------+  a detached signature
          +---------------+
                 |
                 v
        An OpenPGP/MIME object
```

Figure 1, the conceptual model of OpenPGP/MIME composer
Note: Data flows illustrated by arrows are different
      for the signature and encryption services.
      The two objects marked with "*" are copies in the
      signature service.

4.2 OpenPGP/MIME viewer

Viewers that conform to the spececification defined in this memo are called OpenPGP/MIME viewers. An OpenPGP/MIME viewer consists of a MIME viewer and the OpenPGP engine.

The MIME viewer analyzes MIME objects. The MIME viewer cannot analyze PGP or OpenPGP objects. So, PGP or OpenPGP objects are opaque to the MIME viewer. The MIME viewer also does not care about version control for PGP and OpenPGP. However, the MIME viewer may give control information to the OpenPGP engine.

The OpenPGP engine decrypts and/or verifies a PGP or OpenPGP object (then extracts plain output data, which is a MIME object, for the decryption service). The OpenPGP engine cannot analyze MIME objects.

Figure 2 illustrates this conceptual model of OpenPGP/MIME viewer for the verification service.

```
            An OpenPGP/MIME object
                     |
                     v
              +---------------+
              |  MIME viewer  |---------+ The MIME object in the first part*
              +---------------+         | The PGP object in the second part
                     |                  v
  The MIME object    |            +-----------------------------+
  in the first part* |            | OpenPGP verification engine |
                     |            +-----------------------------+
                     v                  |
              +---------------+         | A report on verification
              |  MIME viewer  |<--------+
              +---------------+
                     |
                     v
      The MIME object in the first part*
```

Figure 2, the conceptual model of OpenPGP/MIME viewer
Note: Data flows illustrated by arrows are different
      for the decryption and verification services.
      The two objects marked with "*" are copies in the
      verification service.

5. PGP or OpenPGP object

PGP or OpenPGP objects can be represented by native raw binary octets or by ASCII Armor. ASCII Armor was required for PGP/MIME[OldPGPMIME] wherever PGP objects are used. However, MIME viewers must accept MIME-encoded objects anyway. So, the following are required wherever PGP or OpenPGP objects are used throughout this memo:

(1) OpenPGP/MIME composers SHOULD generate PGP or OpenPGP objects encoded with ASCII Armor. They SHOULD NOT generate OpenPGP objects encoded with a MIME encoding.

(2) OpenPGP/MIME viewers MUST accept PGP or OpenPGP objects encoded both with ASCII Armor and with a MIME encoding.

This requirement affects all PGP or OpenPGP objects including the detached signature, the encryption, and the public key distribution.

NOTE:

There are two kinds of ASCII Armor boundary for a detached signature. PGP uses the "-----BEGIN PGP MESSAGE-----" string while OpenPGP uses the "-----BEGIN PGP SIGNATURE-----" string. OpenPGP/MIME composer SHOULD generate the latter and MAY generate the former for backward compatibility. OpenPGP/MIME viewer MUST accept both.

6. The Signature Service

For the signature service of OpenPGP/MIME, multipart/signed is used. The value of the protocol parameter is "application/pgp-signature". OpenPGP/MIME composers MUST generate the OpenPGP/MIME object defined as follows:

```
Top level:
    Content-Type: multipart/signed
    Required parameters: boundary
    Required parameters: protocol=application/pgp-signature
                         (case-insensitive)
    Optional parameters: micalg (see below)
    Optional parameters: domain (see below)
    Content body is defined as follows:

The first part:
    An MIME object to be signed.
    Content-Transfer-Encoding: MUST be other than "binary".

The second part:
    Content-Type: application/pgp-signature
    Required parameters: none
    Optional parameters: none
    Content body: an OpenPGP object which contains a detached
                  signature
```

A multipart/signed object of OpenPGP/MIME is created as follows:

(1) The MIME composer prepares an object to be signed according to a local convention.

(2) The MIME composer converts the object to a MIME object according to the MIME canonical form. That is, a content header is created and the object becomes a content body. The content body MUST be in 7bit or 8bit. So, if the object is "binary", an appropriate MIME encoding MUST be applied(line delimiters are converted by the MIME encoding). Note that this specification allows an "8bit" content body.

(3) The MIME object is copied. One MIME object will be passed to the OpenPGP engine while the other MIME object will be the first part of multipart/signed.

(4) The MIME composer or the OpenPGP engine converts line delimiters of the latter MIME object (including those of the content header) to <CR><LF>.

(5) The OpenPGP engine calculates a signature over the latter MIME object and creates an OpenPGP object which contains the detached signature.

(6) The MIME composer creates a multipart/signed object. The first part is the former MIME object created in step (3). The MIME composer treates this object as opaque. The second part is the OpenPGP object created in step (5). If this OpenPGP object is native raw binary octets, an appropriate MIME encoding is applied to it.

The OpenPGP object for the detached signature MUST NOT include literal packets.

For the protocol parameter, "application/pgp-signature" SHOULD be lower-case. Since both [SECMULTI] and [OldPGPMIME] doesn't clearly say this parameter is case-insensitive, some PGP/MIME viewers may accept lower-case only.

The micalg parameter MAY be omitted. Just in case, the micalg parameter MAY be specified for multipart/signed. If specified, the value of the micalg parameter SHOULD be "pgp-md5" or "pgp-sha1", which will be ignored by OpenPGP/MIME viewers.

The domain parameter specifies whether or not 8bit characters are contained in the first part. If contained, "8bit" MUST be specified. Otherwise, "7bit" MAY be specified. (If this parameter is not present, OpenPGP/MIME viewers treats that the first part consists of 7bit characters only.)

NOTE:

PGP/MIME requires conversion of a MIME object to a 7bit content transfer encoding before calculating a signature. This is because several message transfer agents(MTA) convert 8bit messages into 7bit in some cases during delivery.

Since there is strong demand for 8bit messages without any encoding, typically to transfer European charcter sets, OpenPGP/MIME allows to sign 8bit MIME objects. The domain parameter associates the original content transfer encoding, so OpenPGP/MIME viewers can tell the possibility of modification by MTAs when verification fails.

If the domain parameter is not present, it is treated as "7bit", which is exactly the same as PGP/MIME.

NOTE:

The micalg parameter is mandatory for multipart/signed. And PGP/MIME defined the values of the micalg parameter. However, this memo doesn't require the micalg parameter for two reasons.

The first reason is that this parameter is not necessary for OpenPGP/MIME. The micalg parameter was designed hoping that one-pass operations could be implemented to calculate a hash value in the MIME level. However, hash calculation is closed in the OpenPGP engine in the conceptual model.

The second reason is that there is no perfect way for the MIME composer to know which hash was actually used in an OpenPGP object produced by the OpenPGP engine. Remember that OpenPGP objects are opaque for the MIME composer.

The micalg parameter should be optional and the specification of multipart/signed should be revised in the future.

IMPLEMENTATION NOTE:

A user may have two or more secret keys. For instance, one is for an RSA secret key for PGP and the other is an ElGamal/DSS secret key for OpenPGP. OpenPGP/MIME composers SHOULD create an OpenPGP detached signature (excluding PGP) by default. OpenPGP/MIME composers MAY create a PGP detached signature for backward compatibility. OpenPGP/MIME composers SHOULD provide a user with a mechanism to select which secret key is used for signature calculation.

There are two ways to support multiple signatures. One is storing them in an OpenPGP object. The other is storing a multipart/* object in the second part. Such functionality is outside the scope of this memo.

7. The Encryption Service

For the encryption service of OpenPGP/MIME, the content type multipart/encrypted is used. The value of the protocol parameter is "application/pgp-encrypted". OpenPGP/MIME composers MUST generate the OpenPGP/MIME object defined as follows:

```
Top level:
    Content-Type: multipart/encrypted
    Required parameters: boundary
    Required parameters: protocol=application/pgp-encrypted
                         (case-insensitive)
    Optional parameters: none
    Content body is defined as follows:

The first part:
    Content-Type: application/pgp-encrypted
    Required parameters: none
    Optional parameters: none
    Content body: "Version: 1"
    Note: existence of this part is for historical reason.
          The content body is meaningless while it seems to
          control versions.
```

The second part:

Content-Type: application/octet-stream
Required parameters: none
Optional parameters: none

Content body: an OpenPGP object which encrypts an MIME object.

A multipart/encrypted object of OpenPGP/MIME is created as follows:

(1) The MIME composer prepares an object to be signed according to a local convention.

(2) The MIME composer converts the object to a MIME object according to the MIME canonical form. That is, a content header is created and the object becomes a content body. The content body need not to be encoded with a MIME encoding. The content body MAY be encoded with an appropriate MIME encoding(line delimiters are converted to the MIME encoding).

(3) The MIME composer or the OpenPGP engine converts line delimiters of the MIME object (including those of the content header) to <CR><LF>.

(4) The OpenPGP engine encrypts the MIME object and creates an OpenPGP object according to user IDs specified by the MIME composer.

(5) The MIME composer creates a multipart/encrypted object. The first part is the application/pgp-encrypted object defined above. The second part is the OpenPGP object created in step (4) which is labeled as application/octet-stream.

For the protocol parameter, "application/pgp-encrypted" SHOULD be lower-case. Since both [SECMULTI] and [OldPGPMIME] doesn't clearly say this parameter is case-insensitive, some PGP/MIME viewers accept lower-case only.

IMPLEMENATION NOTE:

The OpenPGP engine selects receiver's public keys according to the specified user IDs. For example, if there are two public keys for a user ID, the OpenPGP engine may encrypt a session key with both keys.

8. Combined Services

This section describes two typical combined services for OpenPGP/MIME.

8.1 Signed-then-Encrypted Service

There are two methods for the signed-then-encrypted service. Note that the two methods are identical in the service point of view.

8.1.1 Signed-then-Encrypted Service with Security Multipart

Signed-then-Encrypted service can be implemented combining with multipart/signed and multipart/encrypted. OpenPGP/MIME composers MAY produce this format. OpenPGP/MIME viewers MUST accept this format.

OpenPGP/MIME composers first create a multipart/signed object according to the procedures defined in Section 6. Then they creates a multipart/encrypted object according to the procedures defined in Section 7 from step (3) as if the multipart/signed object was created in step (1) and (2).

8.1.2 Signed-then-Encrypted Service with Atomic OpenPGP

The OpenPGP engine itself can create an OpenPGP object which signs-then-encrypts input data. OpenPGP/MIME composers MAY produce this format. OpenPGP/MIME viewers MUST accept this format.

OpenPGP/MIME composers create this format according to the procedures defined in Section 7.

8.2 Encrypted-then-Signed Service

Encrypted-then-signed service can be implemented combining with multipart/encrypted and multipart/signed. OpenPGP/MIME composers MAY produce this format. OpenPGP/MIME viewers MUST accept this format.

OpenPGP/MIME composers first create a multipart/encrypted object according to the procedures defined in Section 7. Then they creates a multipart/signed object according to the procedures defined in Section 6 from step (3) considering as if the multipart/encrypted object is created in step (1) and (2).

9. The Decryption Service

OpenPGP/MIME viewers MUST be able to decrypt the OpenPGP/MIME object defined in Section 7 and 8. The decryption procedures are as follows:

(1) The MIME viewer extracts the first part and the second part from the multipart/encrypted object. The first part, the application/pgp-encrypted object, is ignored. The OpenPGP object stored in the second part is extracted by removing its content header and decoding according to the value of content transfer encoding.

(2) The OpenPGP engine extracts the MIME object by decrypting (and verifying, if necessary) the OpenPGP object. Both the MIME object and a resulting report (on decryption and verification) are passed to the MIME viewer.

(3) The MIME viewer or the PGP engine converts line delimiters of the MIME object into the local form.

(4) The MIME viewer analyzes the MIME object in the context of MIME.

OpenPGP/MIME viewers MUST ignore the first part even if it doesn't conform to the format defined in Section 7.

Mismatch between the value of the protocol parameter and the value of content type in the first part would happen. This error handling is implementation dependent.

IMPLEMENTATION NOTE:

OpenPGP/MIME viewers SHOULD inform the resulting report of step (2) to users.

10. The Verification Service

OpenPGP/MIME viewers MUST be able to verify the OpenPGP/MIME object defined in Section 6 and 8. The verification procedures are as follows:

(1) The MIME viewer extracts the first part and the second part of the multipart/encrypted object. The first part is the signed MIME object. The OpenPGP object stored in the second part is extracted by removing its content header and decoding according to the value of content transfer encoding.

(2) The signed MIME object is copied. One MIME object will be passed to the OpenPGP engine while the other MIME object will be analyzed in the context of MIME later.

(3) The MIME viewer converts line delimiters of the former MIME object into <CR><LF>.

(4) The OpenPGP engine verifies the signature with the former MIME object and the OpenPGP object. A resulting report (on verification) is passed to the MIME viewer.

(5) If the verification fails, the MIME viewer checks whether or not the domain parameter has a correct value. If the value is "8bit" and the first part consists of 7bit characters only, the first part may be alternated by MTAs.

(6) The MIME viewer analyzes the latter MIME object in the context of MIME.

OpenPGP/MIME viewers MUST ignore objects in literal packets if they exist and MUST accept the first part of the multipart/signed object as the signed MIME object.

OpenPGP/MIME viewers MUST ignore the micalg parameter if it exists.

Mismatch between the value of the protocol parameter and the value of content type in the second part would happen. This error handling is implementation dependent.

IMPLEMENTATION NOTE:

The PGP/OpenPGP signature packet specified has "canonical text document" type and "binary document" type. PGP/MIME and OpenPGP/MIME allows these two types for the signature service.

On signature calculation, for canonical text document, line delimiters are converted into <CR><LF> by the OpenPGP engine, then a signature is calculated. For binary document, a signature is calculated over the original input.

On verification, for canonical text document, the line delimiters are first converted into <CR><LF> by the OpenPGP engine, then verified. For binary documnt, verification is carried out over the original input.

Suppose that an OpenPGP/MIME composer itself converts line delimiters of a MIME object into <CR><LF> then calls the OpenPGP engine telling it that this object is a binary document, the signature type will be binary document. Please note that this signature is valid in the context of OpenPGP/MIME and signature verification should succeed. If another OpenPGP/MIME viewer on a system whose line delimiter is <LF> calls the OpenPGP engine without line delimiter conversion, the verification fails because the OpenPGP engine never converts line delimiters of binary document.

So, to verify signatures, the MIME viewer MUST convert line delimiters of the first part to <CR><LF> by itself.

IMPLEMENTATION NOTE:

OpenPGP/MIME viewers SHOULD inform the resulting report of step (5) and (6) to users.

11. Distribution of PGP Transferable Public Keys

To distribute PGP public keys in the context of MIME, the following content type is defined:

```
Content-Type: application/pgp-keys
Required parameters: none
Optional parameters: none
```

OpenPGP/MIME composers MUST embed PGP public keys in this MIME object. For a filename contained in Content-Disposition:, the ".asc" suffix SHOULD be used when encoded with ASCII Armor.

OpenPGP/MIME viewers MUST accept this format. OpenPGP/MIME viewers MUST NOT automatically add the PGP public keys in this MIME object to a public keyring for security reasons. OpenPGP/MIME viewers SHOULD interact with a user to decide how to treat the PGP public keys.

12. Historical Note

Many ideas were proposed to integrate MIME and PGP in the past.

The first trial was the applicaton/pgp content type. An applicaton/pgp MIME object embeds a PGP object, which it signs or encrypts or signs-then-encrypts into a MIME object. This approach is against the design goal (2). That is, it is difficult to extract the inside MIME object from the PGP object even if it is a clear text signature.

Multipart/signed achieves the design goal (2). If a MIME-aware viewer doesn't know multipart/signed, it is treated as multipart/mixed. So, the MIME object in the first part is easily extracted.

[OldPGPMIME] adopts multipart/encrypted for encrypted PGP objects to align to the standard way, security multipart. Since PGP or OpenPGP doesn't have functionality to separate encrypted data and its control keys, the first part of multipart/encrypted defined in this memo is mostly empty.

If all security mechanisms adopt multipart/encrypted for encryption services, MIME viewers can abstract reporting routines (that tell users MIME objects were encrypted). However, S/MIME[SMIME] doesn't adopt multipart/encrypted but does adopt the applicaton/* approach for the encryption service. Thus, this merit is being lost. The reason why this memo uses multipart/encrypted is just for backward compatibility.

Another approach proposed time to time is the text/pgp, which embeds a cleartext signature PGP object, for MIME-unaware but PGP-aware viewers. This approach is against the design goal (2) and (3). Note that since lines started with "-" are escaped in a cleartext signature, it is hard for MIME-aware viwers to extract the original object.

13. Examples

This section contains examples of OpenPGP/MIME objects.

(1) The following is a simple OpenPGP/MIME object for the signature service.

```
Content-Type: Multipart/Signed; boundary="simple"
        protocol="application/pgp-signature";
        domain="7bit"

--simple
Content-Type: Text/Plain; charset=us-ascii
Content-Transfer-Encoding: 7bit

This is a text object to be signed with OpenPGP/MIME.

--simple
Content-Type: Application/Pgp-Signature
Content-Transfer-Encoding: 7bit

-----BEGIN PGP SIGNATURE-----
Version: PGPfreeware 5.0i for non-commercial use
MessageID: M13gFu827jrvjz0U5rM0iWLMfKpt2PzB

iQA/AwUANmJ8V6eXeOJHtjA3EQKOugCfUKYA9iw3KssVncR3hW2oSKKGFDAAn27d
m8dNeZ0d4UO9MLuCITa0sAS1
=czIa
-----END PGP SIGNATURE-----

--simple--
```

(2) The following is a simple OpenPGP/MIME object for the encryption service.

```
Content-Type: Multipart/Encrypted; boundary="simple"
        protocol="application/pgp-encrypted";

--simple
Content-Type: Application/Pgp-Encrypted
Content-Transfer-Encoding: 7bit

Version: 1

--simple
Content-Type: Application/Octet-Stream
Content-Transfer-Encoding: 7bit

-----BEGIN PGP MESSAGE-----
Version: PGPfreeware 5.0i for non-commercial use
MessageID: vdnh0wRmUsog31/QvX620a0hlwLtD+m5

hQCMAw9kihyeT3RNAQP/fLxae4lOuvZxWoZN62cm/3/9K2BSg0OphyqSQu6WVV7u
IPgbvaRqPz7ejJITXEtr+GVR01OEAOENxYTKU/s+vv//XQ1AOljWCuXBJRU0K/L0
Qm9kCw6vprAF8Mwm0QjquOx7PRd/GhqjK5PVBBuXPfFFqlE5ARtQ+27qPYLck7WF
AQ4DYDI64N0KZe0QBACjjBbDgTtYKVPW2J53DxnmwnNTH4t7GuDQ14jfzCYNOeFa
2rpxdd54NgtB932Kh0lyAY7JCAJG2oaKedT7w6owAkdX6bUw4T9RkkWCytsKLPcM
S35OoiLajnS9LXvY7EEEIrmXB2D5AFdPsBLOg9nOJy9giB98hk/OiSYvVqPzigP/
RLmgCjQwlqCGuxAmp9mpqI76mMXfpR5ZN7zXd0hjI2nh9Vnu9/DSUTrzn6mNV/Gt
xN7Afl6YIW/16FbwHozwve2vyJyCRT5XHvFe294eBkJvcdCIIOsW/IMT99NvK3rQ
CK7cwxe1nml33e/QMTh9F3mRibk9fQBQDgS4XOXob5Ckg8JfJbFEA3w5HmwUWfPl
sG0iXQJB6/pSCtV8m1ZBEgJSPSvcwWfR0MCgZr7uyvaxTDgIKE7fcX3a/DzGuYwc
XEWAudEYYjGSWDL21iLMj1g7FehsfDqZZBAMpOr+l/iSxjOypeDhZKYzf/qKK73r
HmgY23tNmMJndoiePbHf7a4KNcUD
=SXzQ
-----END PGP MESSAGE-----

--simple--
```

14. Security Consideration

This memo brings a security mechanism to protect MIME objects. The security level of OpenPGP/MIME is believed the same as that of OpenPGP.

References

[KEYWORDS] S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997.

[MIME] The primary definition of MIME. "MIME Part 1: Format of

Internet Message Bodies", RFC 2045; "MIME Part 2: Media Types", RFC 2046; "MIME Part 3: Message Header Extensions for Non-ASCII Text", RFC 2047; "MIME Part 4: Registration Procedures", RFC 2048; "MIME Part 5: Conformance Criteria and Examples", RFC 2049;

November 1996.

[OldPGP] D. Atkins, W. Stallings, and P. Zimmerman, "PGP Message Exchange Formats", RFC 1991, August 1996.

[OldPGPMIME] M. Elkins, "MIME Security with Pretty Good Privacy (PGP)", RFC2015, October 1996.

[OpenPGP] J. Callas, L. Donnerhacke, H. Finney, and R. Thayer, "OpenPGP Message Format", RFC2440, November 1998.

[PGP] P. Zimmermann, "The Official PGP User's Guide", MIT Press, 1995.

[SECMULTI] J. Galvin, S. Murphy, S. Crocker, and N. Freed, "Security Multiparts for MIME: Multipart/Signed and Multipart/Encrypted", RFC1847, October 1995.

[SMIME] S. Dusse, P. Hoffman, B. Ramsdell, L. Lundblade, and L. Repka, "S/MIME Version 2 Message Specification", RFC 2311, March 1998.

Acknowledgement

The author would acknowledge Thomas Roessler for discussing on technical issues and Tony Mione for reviewing the earlier version of this draft.

Authors' Addresses

Kazuhiko YAMAMOTO
Research Laboratory, Internet Initiative Japan Inc.
Takebashi Yasuda Bldg., 3-13 Kanda Nishiki-cho Chiyoda-ku, Tokyo
101-0054 JAPAN

Phone: +81-3-5259-6350
FAX: +81-3-5259-6351
EMail: kazu@iijlab.net

Open Problems

RFC 1847:

- Should make the micalg parameter optional?
- Should allow 8bit objects for the signature service?

Changes from RFC 2015

To be written.