UNIT II

E-MAIL SECURITY & FIREWALLS

Electronic Mail Security

- **S/MIME**
  - S/MIME emerge as the industry standard for **commercial and organisational use**

S/MIME

- **Secure/Multipurpose Internet Mail Extension (S/MIME)**
  - Provides a consistent means to send and receive secure MIME data
  - Security enhancement
  - Not only restricted to e-mail, but can be used with any transport mechanism that carries MIME data, such as HTTP
  - Allowing secure messages exchange in mixed-transport systems
  - It will emerge as the industry standard for commercial and organisational use

SMTP

- **Simple Mail Transfer Protocol (SMTP)**
  - Internet standard
  - RFC821 in 1982
  - Send and receive mail messages
  - But used only for sending messages to a mail server
  - For retrieving messages use POP
  - Messages are sent only in NVT (Network Virtual Terminal) 7-bit ASCII format
  - This is an 8-bit character set in which the seven lowest-order bits are the same as ASCII and the highest-order bit is zero

SMTP

1. SMTP **cannot transmit executable** files or other binary files.
2. SMTP **cannot transmit text data that includes national language characters** because these are represented by 8-bit codes with values of 128 decimal or higher, and SMTP is limited to 7-bit ASCII.
3. SMTP servers may **reject mail message over a certain size.**
4. SMTP gateways that **translate between ASCII to EBCDIC** suffer translation problems.
5. Some SMTP implementations do not adhere completely to the SMTP standard defined in RFC .

MIME

- *MIME Description*
- *MIME Header*
- *MIME Security Multiparts*
- *MIME Security with OpenPGP*

MIME

- MIME
  - Allow transmission of non-ASCII data through e-mail
  - Arbitrary data is encoded in ASCII and then transmitted in a standard e-mail message
  - It is a supplementary protocol of SMTP - extension to SMTP
  - It allows non-ASCII data to be sent through SMTP
  - Not a mail protocol and cannot replace SMTP
  - Provides a general structure for the content type of Internet messages and allows extensions for new content-type applications
  - To accommodate arbitrary data types and representations, it includes information that tells the recipient the type of the data and the encoding used
  - Content-type declaration must contain two identifiers
    - A content type and a subtype, separated by a slash
      - image/gif
      - text/html
      - video/mpeg

MIME

- *MIME Description*
  - It transforms non-ASCII data at the sender’s site to NVT ASCII data and delivers it to the sender side SMTP (client SMTP)
  - Then it is sent through the Internet
  - The receiver side SMTP (server SMTP)receives the NVT ASCII data and delivers it to MIME
  - MIME transforms back to the original non-ASCII data

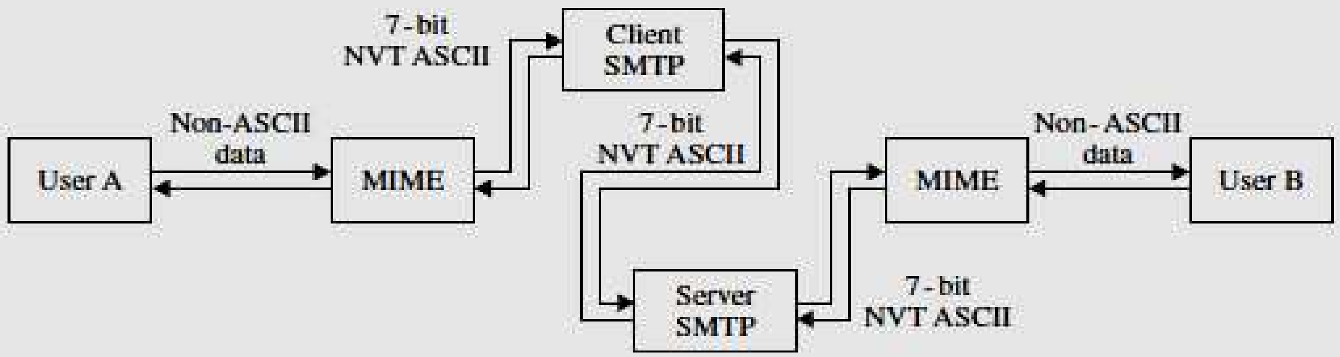
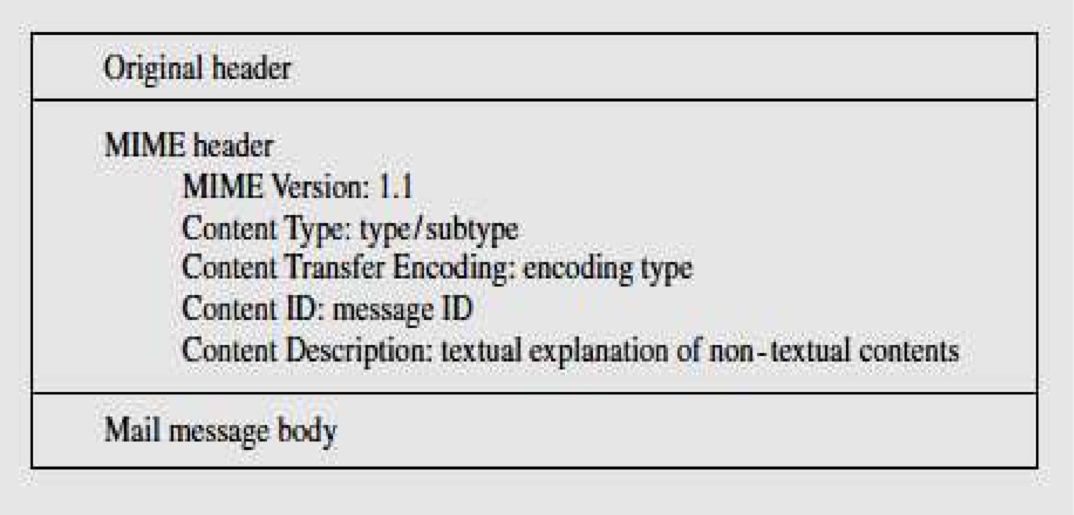


Figure 9.6 MIME showing a set of transforming functions.

MIME

- *MIME Header*
  - Five headers added to the original SMTP header

1. MIME Version
2. Content Type
3. Content Transfer Encoding
4. Content Id
5. Content Description



Original header

MIME header
MIME Version: 1.1
Content Type: type/subtype
Content Transfer Encoding: encoding type
Content ID: message ID
Content Description: textual explanation of non-textual contents

Mail message body

Figure 9.7 MIME header.

MIME

1. MIME Version

- The current version is 1.0

2. Content Type

- Defines the type of data used in the message body
- The content type and the content subtype are separated by a slash “video/mpeg”
- Seven different types of data
  - *Text:*
  - *Multipart:*
  - *Message:*
  - *Image*
  - *Video*
  - *Audio*
  - *Application*

MIME

2. Content Type

- *Text:*
  - *The original message is in 7-bit ASCII format*
- *Multipart:*
  - *The body contains multiple, independent parts*
  - *The multipart header needs* to define the boundary between each part
  - Each part has a separate content type and Encoding

MIME

2. Content Type

- *Multipart:*
  - Multipart/signed content type
    - specifies how to support authentication and integrity services via digital signature
    - Definition of multipart/signed:
      - MIME type name: multipart
      - MIME subtype name: signed.
      - Required parameters: boundary, protocol and micalgo
      - Optional parameters: none
      - Security considerations: must be treated as opaque while in transit.
    - The multipart/signed content type contains exactly two body parts.
      - The first body part is the one over which the digital signature was created, including its MIME headers
      - The second body part contains the control information necessary to verify the digital signature.

MIME

2. Content Type

- *Multipart:*
  - Definition of multipart/encrypted:
    - » MIME type name: multipart
    - » MIME subtype name: encrypted
    - » Required parameters: boundary and protocol
    - » Optional parameters: none
    - » Security considerations: none.
  - The multipart/encrypted content type contains exactly two body parts
    - » The first body part contains the control information necessary to decrypt the data in the second body part
    - » The second body part contains the data which was encrypted and is always labelled application/octet-stream

MIME

- ***2.Content Type***
  - Message:
    - The message body is
      - Whole mail message
      - Part of a mail message
      - Pointer to the message
    - Three subtypes are currently used:
      - RFC 2822
      - Partial
      - External body

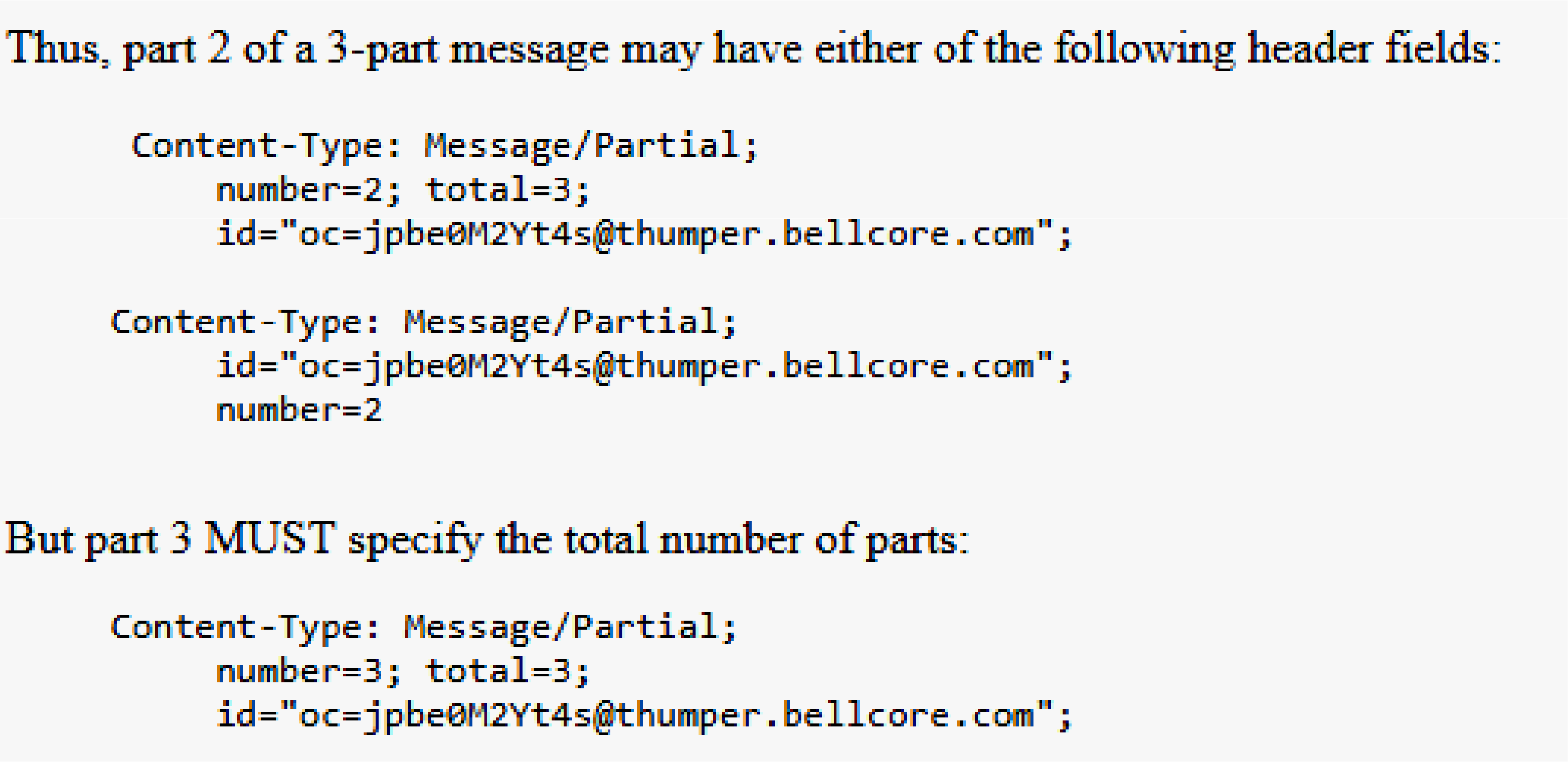
MIME

- ***2.Content Type***
  - Message:
    - **“Message/rfc822**" indicates that the body contains an encapsulated message, with the syntax of an RFC 822 message
    - **“Message/Partial”** A subtype of message, "partial", is defined in order to allow large objects to be delivered as several separate pieces of mail and automatically reassembled by the receiving user agent.
      - Three parameters must be added: ID, number and total. The id identifies the message and is present in all the fragments. The number defines the sequence order of the fragment. The total defines the number of fragments that comprise the original message.
    - **“Message/external-body“** The external-body subtype indicates that the actual body data are not included, but merely referenced. In this case, the parameters describe a mechanism for accessing the external data.

MIME

2. *Content* *Type*

Multipart: Example



Thus, part 2 of a 3-part message may have either of the following header fields:

```
 Content-Type: Message/Partial;
     number=2; total=3;
     id="oc=jpbe0M2Yt4s@thumper.bellcore.com";

Content-Type: Message/Partial;
     id="oc=jpbe0M2Yt4s@thumper.bellcore.com";
     number=2
```

But part 3 MUST specify the total number of parts:

```
Content-Type: Message/Partial;
     number=3; total=3;
     id="oc=jpbe0M2Yt4s@thumper.bellcore.com";
```

MIME

2. Content Type

- Image:
  - The original message is a stationary image, indicating that there is no animation
  - Subtype - Joint Photographic Experts Group (JPEG), Graphics Interchange Format (GIF)
- Video:
  - The original message is a time-varying image (animation)
  - Subtype - Motion Picture Experts Group (MPEG)
  - If the animated image contains sound, it must be sent separately using the audio content type

MIME

2. Content Type

- Audio:
  - The original message contains sound
- Application:
  - The original message is a type of data not previously defined
  - Subtypes - octet-stream and PostScript
    - Octet-stream is used when the data represents a sequence of binary data consisting of 8-bit bytes
    - PostScript is used when the data is in Adobe PostScript format for printers that support PostScript

MIME

3. Content Transfer Encoding

- Defines the method to encode the messages into ones and zeros for transport
- There are the five types of encoding:
  - 7 bit
  - 8 bit
  - Binary
  - Base64
  - Quoted-printable

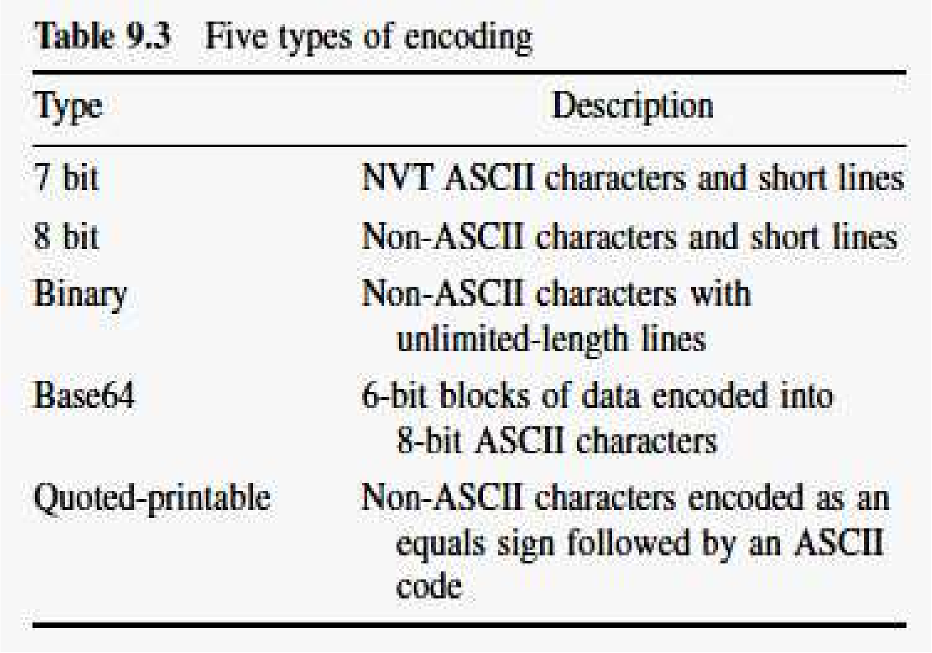


Table 9.3 Five types of encoding

Type	Description
7 bit	NVT ASCII characters and short lines
8 bit	Non-ASCII characters and short lines
Binary	Non-ASCII characters with unlimited-length lines
Base64	6-bit blocks of data encoded into 8-bit ASCII characters
Quoted-printable	Non-ASCII characters encoded as an equals sign followed by an ASCII code

MIME

3. Content Transfer Encoding

- *7 bit:*
  - *NVT ASCII encoding.*
  - *No special transformation is* needed
  - The length of the line should not exceed 1000 characters
- *8 bit:*
  - *This is 8-bit encoding*
  - *Non-ASCII characters can be sent*
  - *Length of the* line still should not exceed 1000 characters
  - SMTP is able to transfer 8-bit non-ASCII characters - MIME does not do any encoding here
  - Base64 (or radix-64) and quoted-printable types are preferable
- *Binary:*
  - *This is 8-bit encoding*
  - *Non-ASCII characters can be sent*
  - *Length of the* line can exceed 1000 characters
  - MIME does not do any encoding here
  - .
- *Base64 :*
  - *Sending data made of bytes when the highest bit is not* necessarily zero.
  - Base64 transforms this type of data of printable characters which can be sent as ASCII characters.
- *Quoted-printable:*
  - *Base64 is a redundant encoding scheme*
  - *The 24-bit non-ASCII data* becomes four characters consisting of 32 bits. We have an overhead of 25%. If the data consists of mostly ASCII characters with a small non-ASCII portion, we can use quoted-printable encoding. If a character is ASCII, it is sent as it is; if a character is not ASCII it is sent as three characters.

MIME

4. Content Id

- This header uniquely identifies the whole message in a multiple message environment:
- Content Id: id = *<content id>*

5. Content Description

- This header defines whether the body is image, audio or video:
- Content Description: *<description>*

MIME

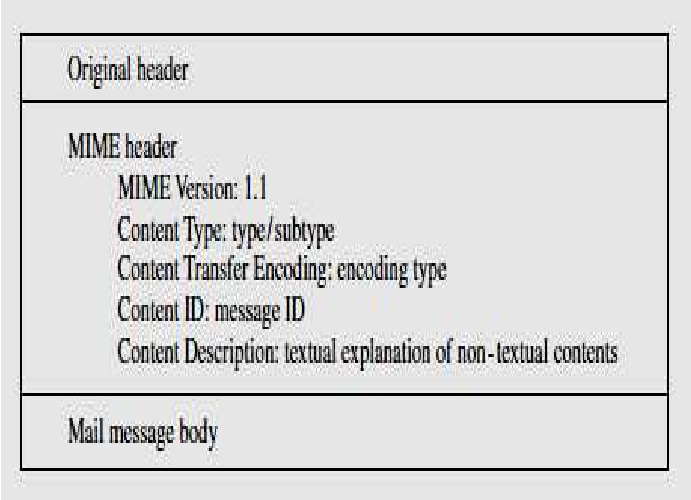
- *Example 9.5*
  - *Consider an MIME message that contains a photograph in standard GIF* representation. This GIF image is to be converted to 7-bit ASCII using Base64 encoding as follows:

```
From: myrhee@tsp.snu.ac.kr
To: kiisc2@kornet.net
MIME Version: 1.1
Content Type: image/gif
Content Transfer Encoding: Base64
. . . data for the gif image . . .
```

MIME

- *MIME Header :*
  - Five headers added to the original SMTP header

1. MIME Version 1.1
2. Content Type
   - *Text:*
   - *Multipart: - multiple independent parts (signed,encrypted)*
   - *Message:*
     - Whole mail message : Message/RFC 2822
     - Part of a mail message : Message/Partial
     - Pointer to the message : Message/External
   - *Image*
   - *Video*
   - *Audio*
   - *Application* - octet-stream and PostScript
3. Content Transfer Encoding
   - 7 bit
   - 8 bit
   - Binary
   - Base64
   - Quoted-printable
4. Content Id
5. Content Description



Original header

MIME header
MIME Version: 1.1
Content Type: type/subtype
Content Transfer Encoding: encoding type
Content ID: message ID
Content Description: textual explanation of non-textual contents

Mail message body

Figure 9.7 MIME header.

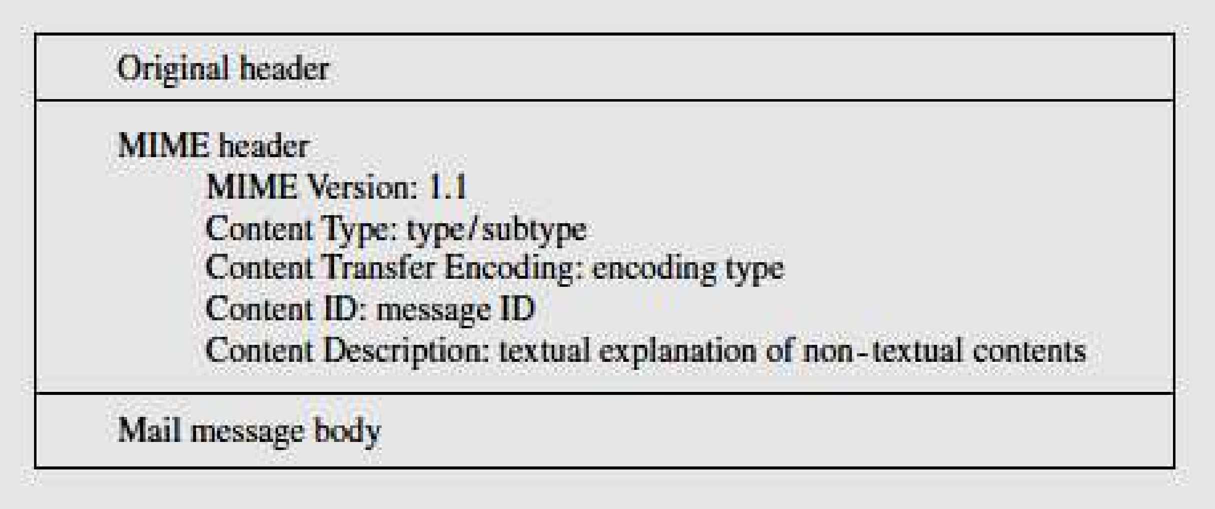
MIME

- **MIME Security Multiparts**

Refer: Content Type- *Multipart:*

Email message
Two part

Header	**Body**
(Field /value : Multipart/Signed or Multipart/Encrypted)	**(MIME format) (Protected data and control information)**



Original header

MIME header
    MIME Version: 1.1
    Content Type: type/subtype
    Content Transfer Encoding: encoding type
    Content ID: message ID
    Content Description: textual explanation of non-textual contents

Mail message body

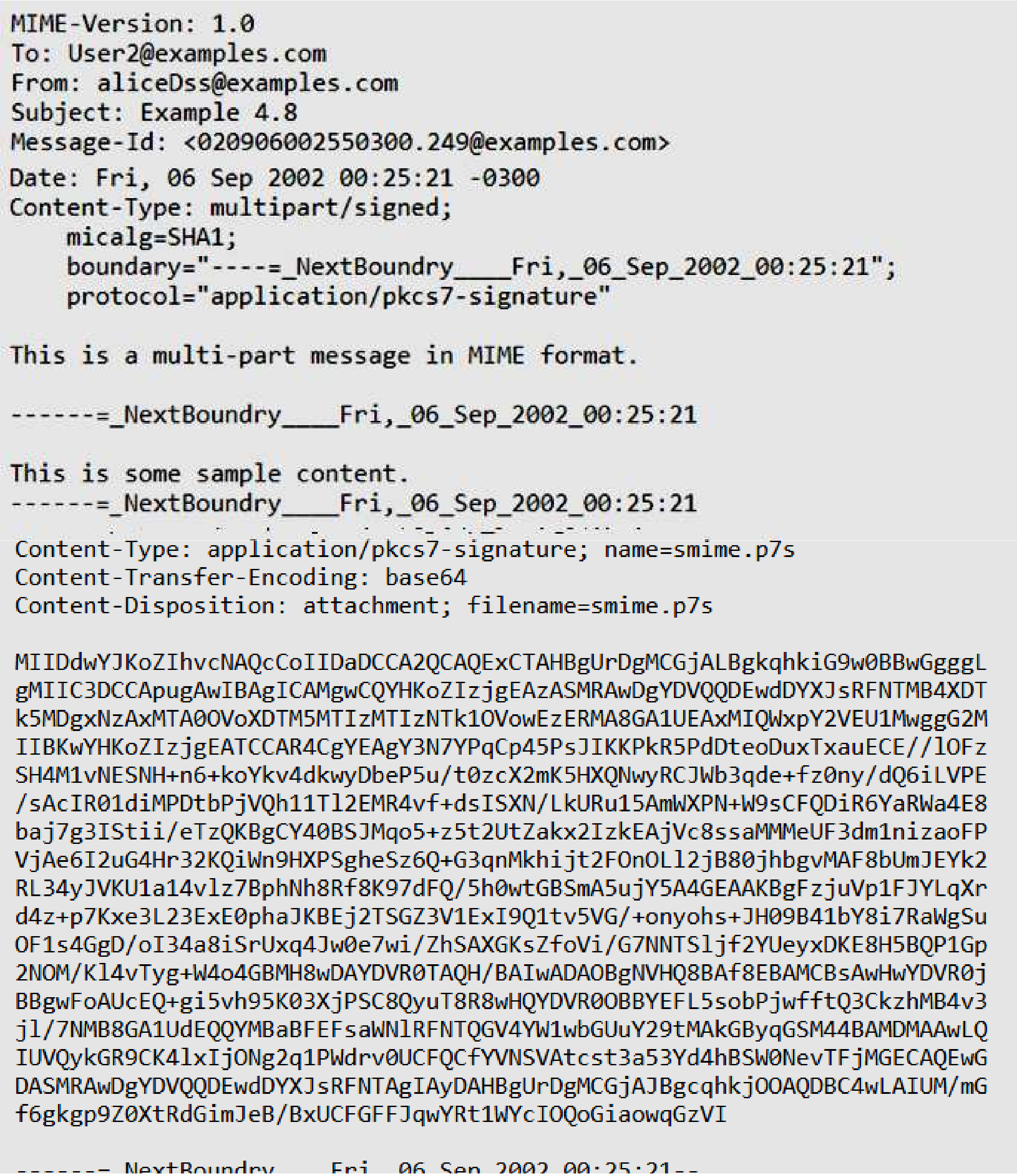
Figure 9.7 MIME header.

MIME

- **MIME Security Multiparts**
  - Basic MIME by itself does not specify security protection
  - MIME agent
    - Provide security services
    - Employ a security protocol mechanism
    - Two security subtypes
      - Multipart/Signed
      - Multipart/Encrypted
    - Each of the security subtypes
      - There are exactly two related body parts:
        - » one for the protected data
        - » one for the control information
    - Should recognise a security multipart body part
    - Identify its protected data and control information

MIME

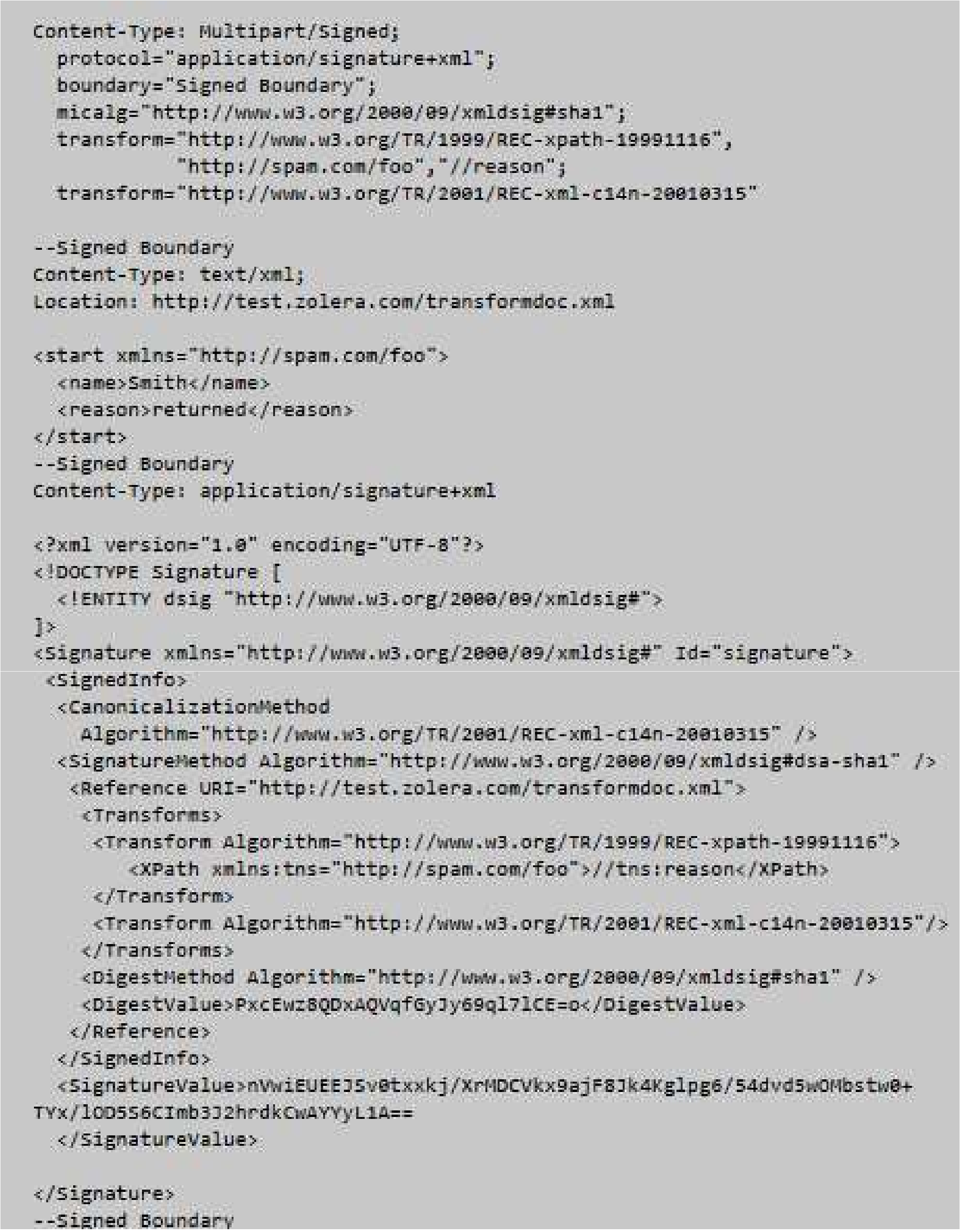
- *MIME Security Multiparts*
  - Content type - Multipart/signed
    - Via digital signature Supports
      - Authentication
      - Integrity services
    - Contains exactly two body parts
      - The first body part is the one over which the digital signature was created, including its MIME headers
      - The second body part contains the control information necessary to verify the digital signature
    - The Message Integrity Check (MIC) is the quantity computed over the body part with hash function to support of the digital signature service



```
MIME-Version: 1.0
To: User2@examples.com
From: aliceDss@examples.com
Subject: Example 4.8
Message-Id: <020906002550300.249@examples.com>
Date: Fri, 06 Sep 2002 00:25:21 -0300
Content-Type: multipart/signed;
    micalg=SHA1;
    boundary="----=_NextBoundry____Fri,_06_Sep_2002_00:25:21";
    protocol="application/pkcs7-signature"

This is a multi-part message in MIME format.

------=_NextBoundry____Fri,_06_Sep_2002_00:25:21

This is some sample content.
------=_NextBoundry____Fri,_06_Sep_2002_00:25:21
Content-Type: application/pkcs7-signature; name=smime.p7s
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename=smime.p7s
```

```
MIIDdwYJKoZIhvcNAQcCoIIDaDCCA2QCAQExCTAHBgUrDgMCGjALBgkqhkiG9w0BBwGgggL
gMIIC3DCCApugAwIBAgICAMgwCQYHKoZIzjgEAzASMRAwDgYDVQQDEwdDYXJsRFNTMB4XDT
k5MDgxNzAxMTA0OVoXDTM5MTIzMTIzNTk1OVowEzERMA8GA1UEAxMIQWxpY2VEU1MwggG2M
IIBKwYHKoZIzjgEATCCAR4CgYEAgY3N7YPqCp45PsJIKKPkR5PdDteoDuxTxauECE//lOFz
SH4M1vNESNH+n6+koYkv4dkwyDbeP5u/t0zcX2mK5HXQNwyRCJWb3qde+fz0ny/dQ6iLVPE
/sAcIR01diMPDtbPjVQh11Tl2EMR4vf+dsISXN/LkURu15AmWXPN+W9sCFQDiR6YaRWa4E8
baj7g3IStii/eTzQKBgCY40BSJMqo5+z5t2UtZakx2IzkEAjVc8ssaMMMeUF3dm1nizaoFP
VjAe6I2uG4Hr32KQiWn9HXPSgheSz6Q+G3qnMkhijt2FOnOLl2jB80jhbgvMAF8bUmJEYk2
RL34yJVKU1a14vlz7BphNh8Rf8K97dFQ/5h0wtGBSmA5ujY5A4GEAAKBgFzjuVp1FJYLqXr
d4z+p7Kxe3L23ExE0phaJKBEj2TSGZ3V1ExI9Q1tv5VG/+onyohs+JH09B41bY8i7RaWgSu
OF1s4GgD/oI34a8iSrUxq4Jw0e7wi/ZhSAXGKsZfoVi/G7NNTSljf2YUeyxDKE8H5BQP1Gp
2NOM/Kl4vTyg+W4o4GBMH8wDAYDVR0TAQH/BAIwADAOBgNVHQ8BAf8EBAMCBsAwHwYDVR0j
BBgwFoAUcEQ+gi5vh95K03XjPSC8QyuT8R8wHQYDVR0OBBYEFL5sobPjwfftQ3CkzhMB4v3
jl/7NMB8GA1UdEQQYMBaBFEFsaWNlRFNTQGV4YW1wbGUuY29tMAkGByqGSM44BAMDMAAwLQ
IUVQykGR9CK4lxIjONg2q1PWdrv0UCFQCfYVNSVAtcst3a53Yd4hBSW0NevTFjMGECAQEwG
DASMRAwDgYDVQQDEwdDYXJsRFNTAgIAyDAHBgUrDgMCGjAJBgcqhkjOOAQDBC4wLAIUM/mG
f6gkgp9Z0XtRdGimJeB/BxUCFGFFJqwYRt1WYcIOQoGiaowqGzVI
```

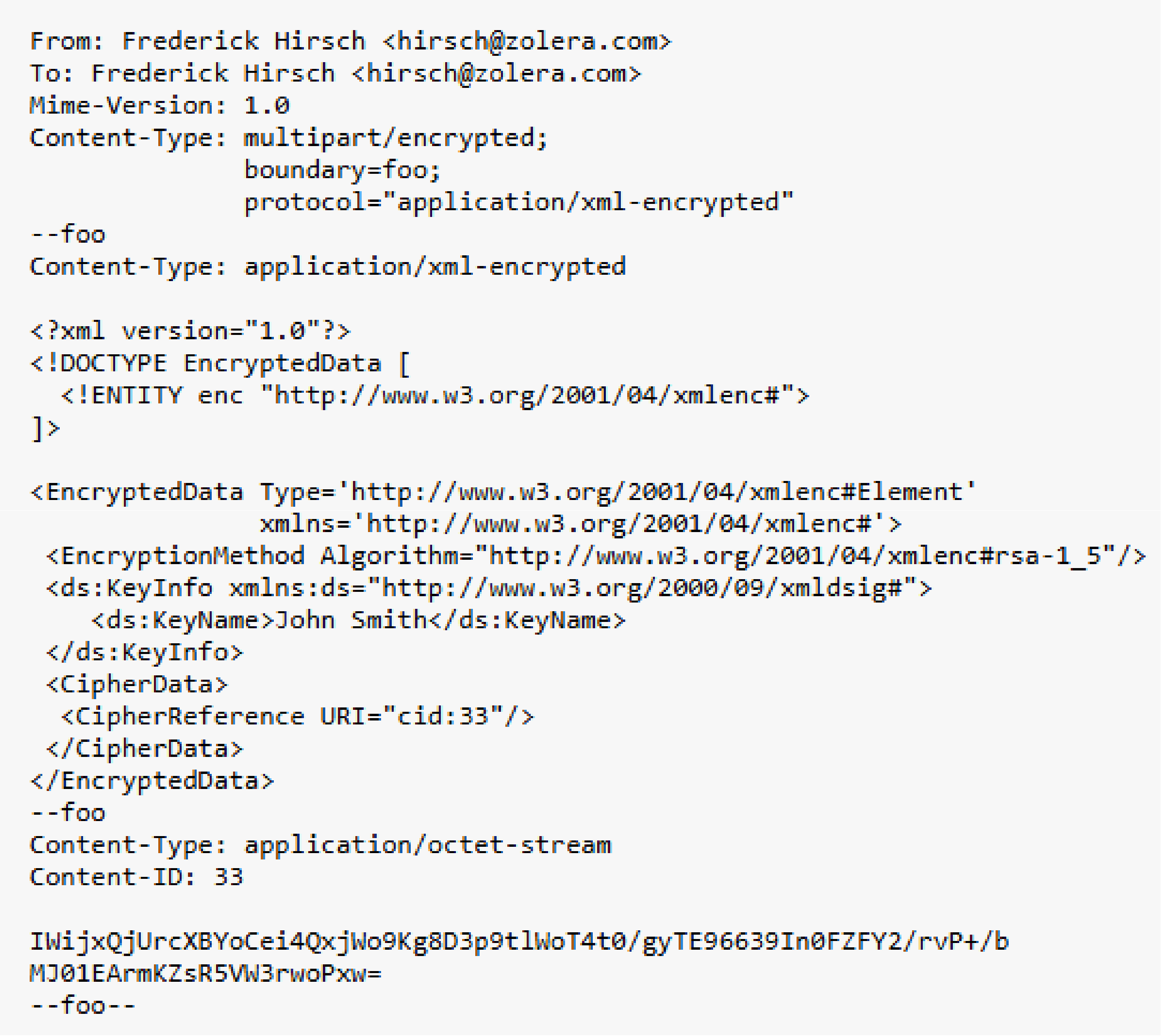
```
------=_NextBoundry____Fri,_06_Sep_2002_00:25:21--
```



```
Content-Type: Multipart/Signed;
  protocol="application/signature+xml";
  boundary="Signed Boundary";
  micalg="http://www.w3.org/2000/09/xmldsig#sha1";
  transform="http://www.w3.org/TR/1999/REC-xpath-19991116",
             "http://spam.com/foo","//reason";
  transform="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"

--Signed Boundary
Content-Type: text/xml;
Location: http://test.zolera.com/transformdoc.xml

<start xmlns="http://spam.com/foo">
  <name>Smith</name>
  <reason>returned</reason>
</start>
--Signed Boundary
Content-Type: application/signature+xml

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Signature [
  <!ENTITY dsig "http://www.w3.org/2000/09/xmldsig#">
]>
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#" Id="signature">
 <SignedInfo>
  <CanonicalizationMethod
    Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
  <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1" />
   <Reference URI="http://test.zolera.com/transformdoc.xml">
    <Transforms>
     <Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
        <XPath xmlns:tns="http://spam.com/foo">//tns:reason</XPath>
     </Transform>
     <Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
    </Transforms>
    <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
    <DigestValue>PxcEwz8QDxAQVqfGyJy69ql7lCE=o</DigestValue>
   </Reference>
  </SignedInfo>
  <SignatureValue>nVwiEUEEJSv0txxkj/XrMDCVkx9ajF8Jk4Kglpg6/54dvd5wOMbstw0+
TYx/lOD556CImb3J2hrdkCwAYYyL1A==
  </SignatureValue>

</Signature>
--Signed Boundary
```

MIME

- *MIME Security Multiparts*
  - Content type - Multipart/encrypted
    - Via encryption Support
      - Confidentiality
    - Contains exactly two body parts
      - The first body part contains the control information necessary to decrypt the data in the second body part
      - The second body part contains the data which was encrypted and is always labelled application/octet-stream



```
From: Frederick Hirsch <hirsch@zolera.com>
To: Frederick Hirsch <hirsch@zolera.com>
Mime-Version: 1.0
Content-Type: multipart/encrypted;
              boundary=foo;
              protocol="application/xml-encrypted"
--foo
Content-Type: application/xml-encrypted

<?xml version="1.0"?>
<!DOCTYPE EncryptedData [
  <!ENTITY enc "http://www.w3.org/2001/04/xmlenc#">
]>

<EncryptedData Type='http://www.w3.org/2001/04/xmlenc#Element'
               xmlns='http://www.w3.org/2001/04/xmlenc#'>
 <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5"/>
 <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:KeyName>John Smith</ds:KeyName>
 </ds:KeyInfo>
 <CipherData>
  <CipherReference URI="cid:33"/>
 </CipherData>
</EncryptedData>
--foo
Content-Type: application/octet-stream
Content-ID: 33

IWijxQjUrcXBYoCei4QxjWo9Kg8D3p9tlWoT4t0/gyTE96639In0FZFY2/rvP+/b
MJ01EArmKZsR5VW3rwoPxw=
--foo--
```

MIME

- **MIME Security with OpenPGP**
  - OpenPGP message format can be used to provide privacy and authentication using the MIME security content type
    - "application/pgp-encrypted”
    - "application/pgp-signature“
    - "application/pgp-keys"
  - The integrating work on PGP with MIME suffered from a number of problems
    - Most significant of which was the inability to recover signed message bodies without parsing data structures specific to PGP

MIME

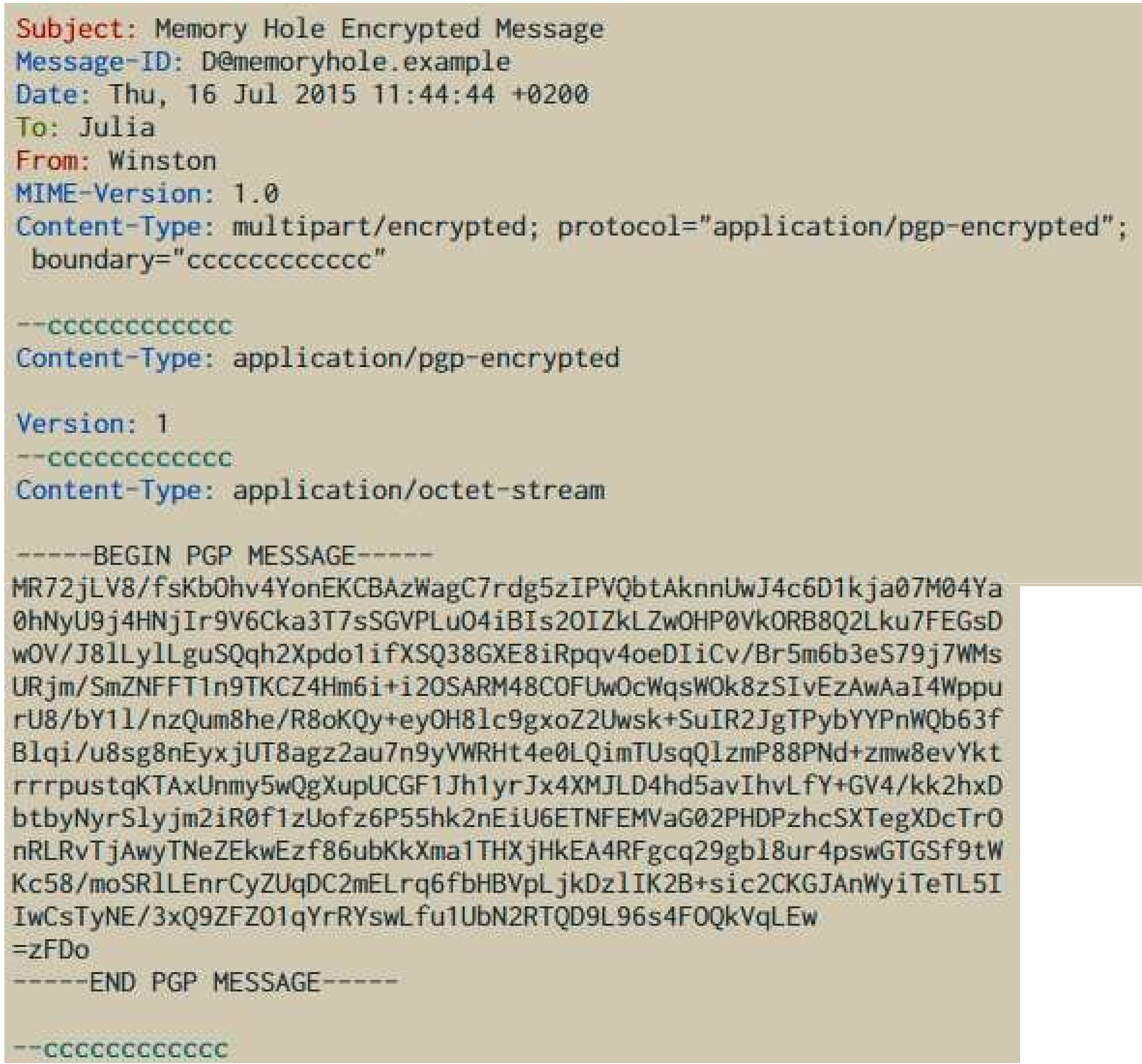
- **MIME Security with OpenPGP**
  - RFC 1847 defines security multipart formats
  - PGP
    - Generate
      - ASCII Armor (or)
      - A stream of arbitrary 8-bit octets
    - When
      - Encrypting data (or)
      - Generating a digital signature (or)
      - Extracting public-key data (or)
  - The ASCII Armor output is the required method for data transfer
  - When the data is to be transmitted in many parts, the MIME **message/partial** mechanism should be used

MIME

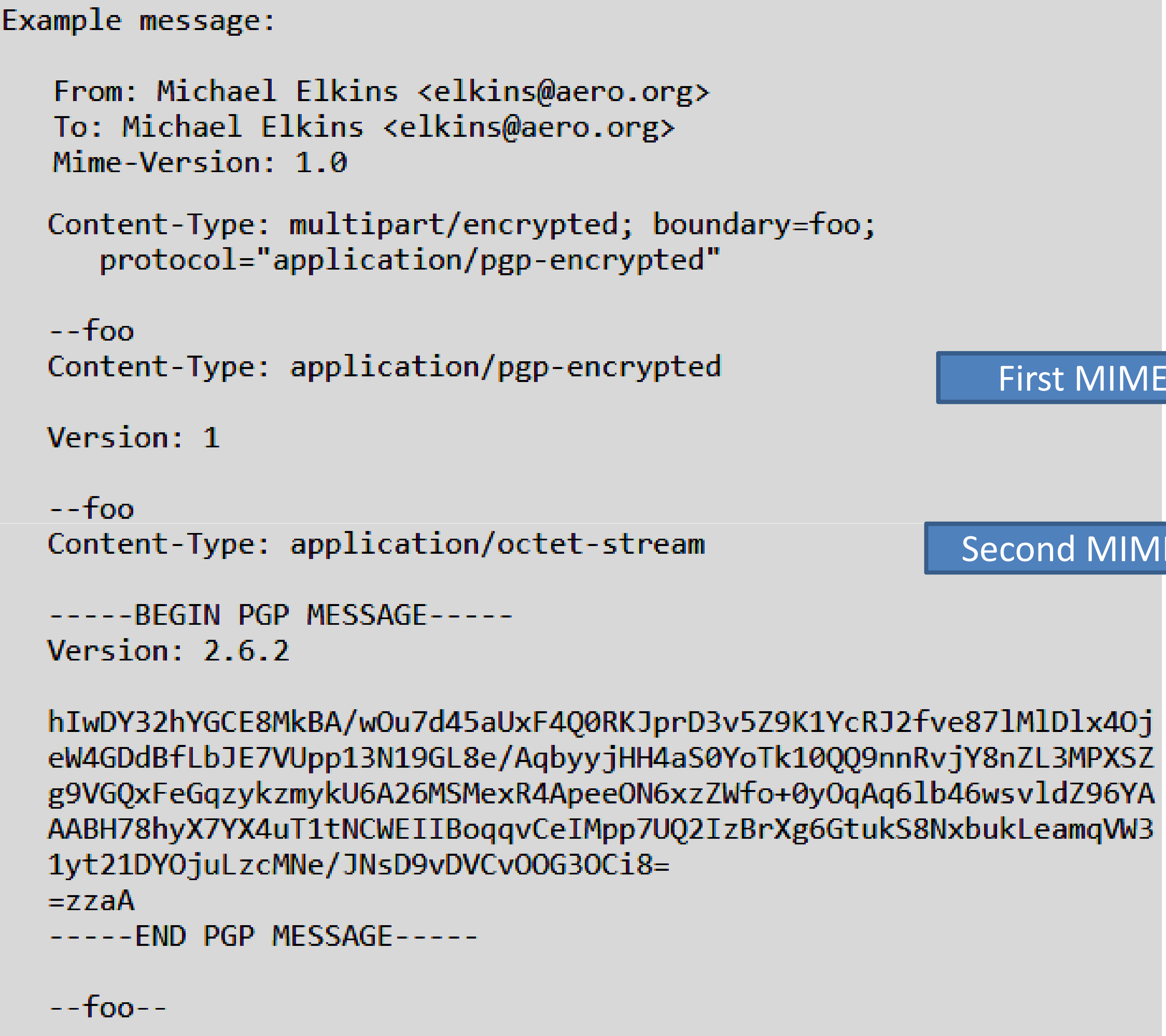
- **MIME Security with OpenPGP**
  - MIMG Agents treat and interpret multipart/signed and multipart/encrypted as opaque
    - Means that the data is not to be altered in any way
  - If the next hop do not support MIME then
    - Data in 8 bit is converted to some other format
    - This is not allowed in multipart/encrypted, multipart/signed form
    - Such conversion presents serious problems of invalidating the signature
    - For this reason all data signed according to this protocol must be constrained to 7 bits

MIME

- **MIME Security with OpenPGP**
  - **OpenPGP encryption**
    - Before OpenPGP encryption, the data is written in MIME canonical format (body and headers)
    - OpenPGP encrypted data is denoted
      - *Content type: multipart/encrypted*
      - Protocol parameter: application/pgp-encrypted
    - The multipart/encrypted MIME body must consist of exactly two body parts
      - First body contains the control information
        - » With content type ‘application/pgp-encrypted’
      - Second MIME body part must contain the actual encrypted data
        - » With a content type of ‘application/octet-stream’.



```
Subject: Memory Hole Encrypted Message
Message-ID: D@memoryhole.example
Date: Thu, 16 Jul 2015 11:44:44 +0200
To: Julia
From: Winston
MIME-Version: 1.0
Content-Type: multipart/encrypted; protocol="application/pgp-encrypted";
 boundary="cccccccccccc"

--cccccccccccc
Content-Type: application/pgp-encrypted

Version: 1
--cccccccccccc
Content-Type: application/octet-stream

-----BEGIN PGP MESSAGE-----
MR72jLV8/fsKbOhv4YonEKCBAzWagC7rdg5zIPVQbtAknnUwJ4c6D1kja07M04Ya
0hNyU9j4HNjIr9V6Cka3T7sSGVPLuO4iBIs2OIZkLZwOHP0VkORB8Q2Lku7FEGsD
wOV/J8lLylLguSQqh2Xpdo1ifXSQ38GXE8iRpqv4oeDIiCv/Br5m6b3eS79j7WMs
URjm/SmZNFFT1n9TKCZ4Hm6i+i2OSARM48COFUwOcWqsWOk8zSIvEzAwAaI4Wppu
rU8/bY1l/nzQum8he/R8oKQy+eyOH8lc9gxoZ2Uwsk+SuIR2JgTPybYYPnWQb63f
Blqi/u8sg8nEyxjUT8agz2au7n9yVWRHt4e0LQimTUsqQlzmP88PNd+zmw8evYkt
rrrpustqKTAxUnmy5wQgXupUCGF1Jh1yrJx4XMJLD4hd5avIhvLfY+GV4/kk2hxD
btbyNyrSlyjm2iR0f1zUofz6P55hk2nEiU6ETNFEMVaG02PHDPzhcSXTegXDcTrO
nRLRvTjAwyTNeZEkwEzf86ubKkXma1THXjHkEA4RFgcq29gbl8ur4pswGTGSf9tW
Kc58/moSRlLEnrCyZUqDC2mELrq6fbHBVpLjkDzlIK2B+sic2CKGJAnWyiTeTL5I
IwCsTyNE/3xQ9ZFZO1qYrRYswLfu1UbN2RTQD9L96s4FOQkVqLEw
=zFDo
-----END PGP MESSAGE-----

--cccccccccccc
```



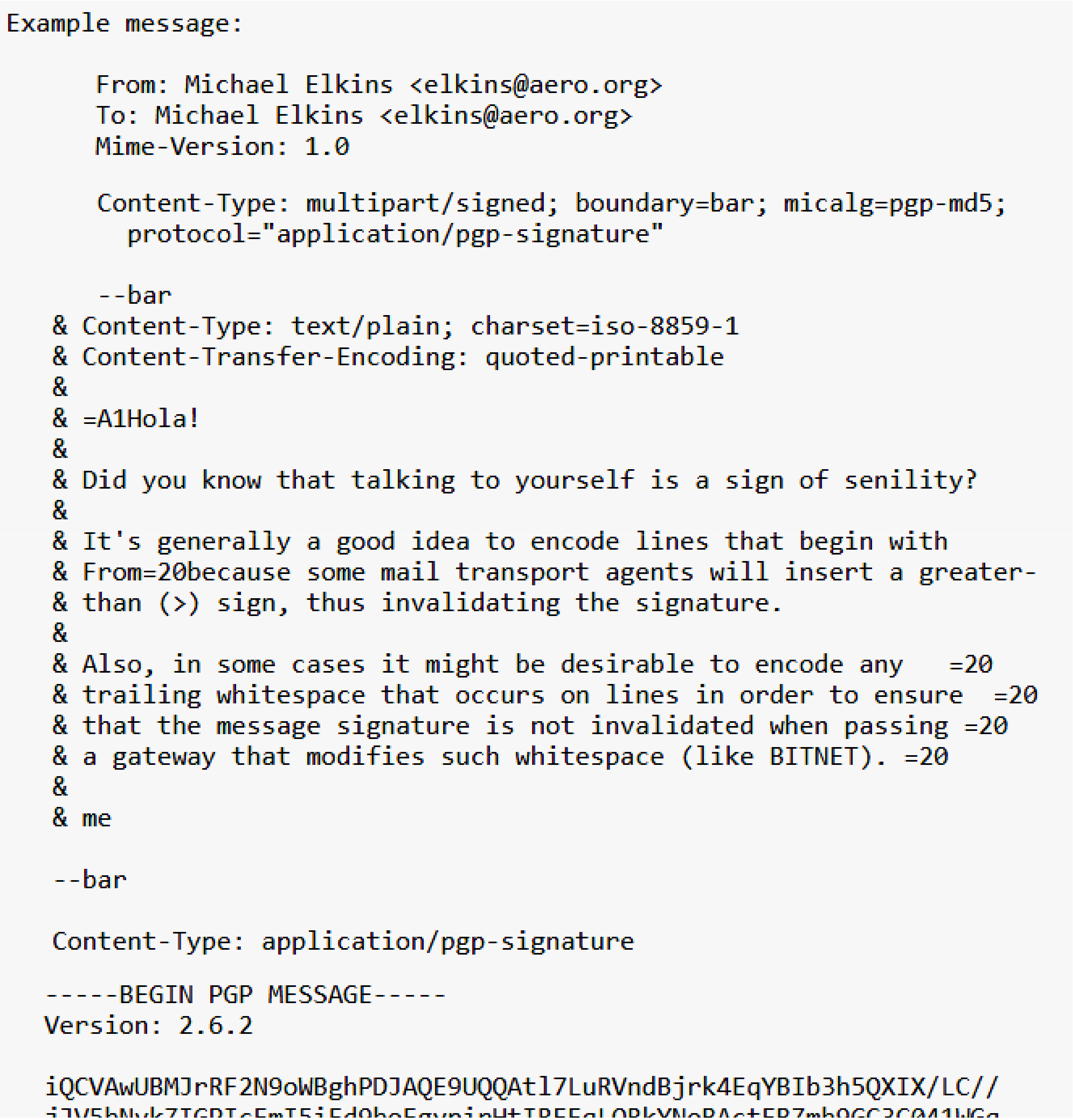
```
Example message:

   From: Michael Elkins <elkins@aero.org>
   To: Michael Elkins <elkins@aero.org>
   Mime-Version: 1.0

   Content-Type: multipart/encrypted; boundary=foo;
      protocol="application/pgp-encrypted"

   --foo
   Content-Type: application/pgp-encrypted

   Version: 1

   --foo
   Content-Type: application/octet-stream

   -----BEGIN PGP MESSAGE-----
   Version: 2.6.2

   hIwDY32hYGCE8MkBA/wOu7d45aUxF4Q0RKJprD3v5Z9K1YcRJ2fve87lMlDlx4Oj
   eW4GDdBfLbJE7VUpp13N19GL8e/AqbyyjHH4aS0YoTk10QQ9nnRvjY8nZL3MPXSZ
   g9VGQxFeGqzykzmykU6A26MSMexR4ApeeON6xzZWfo+0yOqAq6lb46wsvldZ96YA
   AABH78hyX7YX4uT1tNCWEIIBoqqvCeIMpp7UQ2IzBrXg6GtukS8NxbukLeamqVW3
   1yt21DYOjuLzcMNe/JNsD9vDVCvOOG3OCi8=
   =zzaA
   -----END PGP MESSAGE-----

   --foo--
```

First MIME body

Second MIME body

MIME

- **MIME Security with OpenPGP**
  - **OpenPGP signed messages**
      - *Content type: multipart/signed*
      - Protocol parameter: application/pgp-signature
      - *micalg parameter :* pgp-<hash- identifier>“
        - » *Eg:* pgp-md5, pgp-sha1, pgp-ripemd160, pgp-tiger192, pgp-haval-5-160
    - The multipart/signed body must consist of exactly two parts
      - First part contains the signed data in MIME canonical format, including a set of appropriate content headers describing the data
        - » Wth a content type of ‘text/plain’.
      - Second part must contain the OpenPGP digital signature
        - » Wth a content type of ‘application/pgpsignature’.
  - Note "&"s in the following example indicate the portion of the data over which the signature was calculated.



```
Example message:

      From: Michael Elkins <elkins@aero.org>
      To: Michael Elkins <elkins@aero.org>
      Mime-Version: 1.0

      Content-Type: multipart/signed; boundary=bar; micalg=pgp-md5;
        protocol="application/pgp-signature"

      --bar
   & Content-Type: text/plain; charset=iso-8859-1
   & Content-Transfer-Encoding: quoted-printable
   &
   & =A1Hola!
   &
   & Did you know that talking to yourself is a sign of senility?
   &
   & It's generally a good idea to encode lines that begin with
   & From=20because some mail transport agents will insert a greater-
   & than (>) sign, thus invalidating the signature.
   &
   & Also, in some cases it might be desirable to encode any   =20
   & trailing whitespace that occurs on lines in order to ensure  =20
   & that the message signature is not invalidated when passing =20
   & a gateway that modifies such whitespace (like BITNET). =20
   &
   & me

   --bar

   Content-Type: application/pgp-signature

   -----BEGIN PGP MESSAGE-----
   Version: 2.6.2

   iQCVAwUBMJrRF2N9oWBghPDJAQE9UQQAtl7LuRVndBjrk4EqYBIb3h5QXIX/LC//
   jJV5bNvkZIGPIcEmI5iFd9boEgvpirHtIREEqLQRkYNoBActFBZmh9GC3C041WGq
```

MIME

- **MIME Security with OpenPGP**
  - OpenPGP signed messages
    - When the OpenPGP digital signature is generated:
      - The data to be signed must first be converted to its content-type specific canonical form
      - An appropriate Content Transfer Encoding is applied
        - *<CR><LF>*
      - MIME content headers are then added to the body, each ending with the canonical *<CR><LF> sequence*
      - Any trailing white space must be removed from the signed material
      - The digital signature must be calculated over both the data to be signed and its set of content headers
      - The signature must be generated as detached from the signed data so that the process does not alter the signed data in any way.

MIME

- **MIME Security with OpenPGP**
  - OpenPGP signed messages
    - Upon receipt of a signed message
      - Convert line endings to the canonical *<CR><LF> sequence before the signature can* be verified
      - Pass both the signed data and its associated content headers along with the OpenPGP signature to the signature verification service

MIME

- **MIME Security with OpenPGP**
  - OpenPGP signed messages
    - Sometimes it is desirable both to digitally sign and then to encrypt a message to be sent
    - This encrypted and signed data protocol allows for two ways of accomplishing this task

MIME

- **MIME Security with OpenPGP**
  - Open PGP signed messages
    - Two ways
      - » The data is first signed as a multipart/signature body, and then encrypted to form the final multipart/encrypted body
      - » The OpenPGP packet format describes a method for signing and encrypting data in a single OpenPGP message
        - Reduce processing overheads