S/MIME

Heinrich Moser*

December 2001–January 2002

E-mail communication is insecure. E-mails can be read and modified as they are passed through the Internet as clear-text. S/MIME is an attempt to standardize a protocol used to encrypt and digitally sign e-mail correspondence.

1 Concepts

E-Mail is usually sent over the Internet as plain text. It can be read and altered by anyone whose server it passes through. Therefore, two basic needs have emerged:

Confidentiality The e-mail can only be read by the intended recipient. This is ensured using *encryption*.

Authentication The e-mail has been written by particular person and has not been altered on its way over the Internet. This can be accomplished using *digital signatures*.

S/MIME specifies a protocol to encrypt and digitally sign e-mail messages. The current version v3 is specified in RFC 2633[1].

Due to the large amount of people participating in e-mail correspondence, symmetric key systems (i.e. encryption key = decryption key) would not be practicable, because for everyone to be able to privately send e-mails to everyone else, $\frac{n(n-1)}{2}$ keys would need to be exchanged. Therefore, symmetric keys are only used as temporary session keys in order to take advantage of their faster processing abilities, whereas asymmetric keys are used as permanent keys.

S/MIME uses public key cryptography (an asymmetric system) to sign and encrypt e-mail. Basically, every participant has two keys: A *private key*, which is kept secret and a *public key*, which is available to everyone. Files or mails encrypted using someone's private key can only be decrypted using his public key and vice versa.

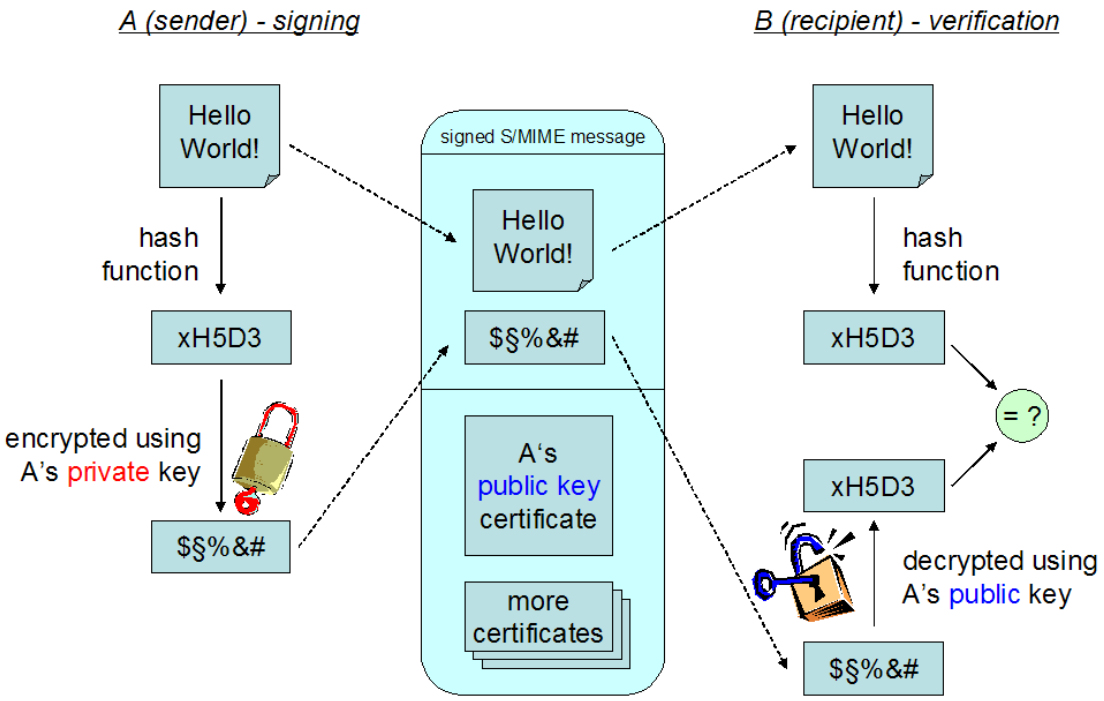

Figure 1: signed mail

1.1 Signed Mail

In general, a message could be signed by person A by just encrypting the message using his private key (= *signing*). Recipient B can try to decrypt the message using A's public key (= *verifying*). If he succeeds, he can be sure that the message is authentic and has not been altered with, because a message, that can be decrypted using A's public key must have been encrypted using A's private key (to which only A has access).

However, for the sake of performance and ease-of-use, S/MIME does signing a bit differently:

- Only a message digest is encrypted, which is faster than encrypting the entire message.
- Therefore, a copy of the original, unsigned message must be included with the mail.

The following steps are taken in order to create a signed message:

1. The user writes the message as clear-text.

*E-mail: h.moser.jun@moserware.at

2. The message digest is being calculated (using SHA-1[2] or MD5[3]).

3. The message digest is being encrypted using the signer's private key (DSS[4] or RSA[5]).

Signed E-mail	
original message	clear-text
signer's public key	to enable the recipient to verify the signature
algorithm identifiers	to tell the receiver's software which hash function and encryption algorithm to use
encrypted message digest	which will be decrypted using the signer's public key in order to verify the signature
public key certificates	(optionally) to prove the authencity of the signer's public key (i.e. to prove, that the signer is really who he appears to be)

Table 1: Contents of a clear-signed e-mail

There are two ways of encoding a signed mail:

clear-signed A clear-signed message contains the original message as clear text. This enables non-S/MIME-compatible mail reader software to read the contents of the message.

opaque-signed Using opaque-signing, the original message is not included as clear text but base64[1]-encoded. This eliminates the risk of the original text being changed while being transmitted (e.g. through automatic conversion performed by some mail transfer agent on the way). If the text was changed even slightly, the message digest would be different, thereby invalidating the digital signature. However, opaque-signed messages have the drawback that they can not be read by non-S/MIME-compatible mail readers.

1.2 Encrypted Mail

An encrypted message, sent by A to B, can only be read by B. This is ensured by encrypting the message using B's public key, which is available to everyone. However, only B can decrypt the message, because only he owns his private key.

Again, to enhance performance, S/MIME implementations do something slightly different:

- The message is not encrypted using B's public key but instead using a randomly created symmetric session key. Symmetric encryption/decryption is faster than asymmetric algorithms.

[1]See Section 2.2 for details.

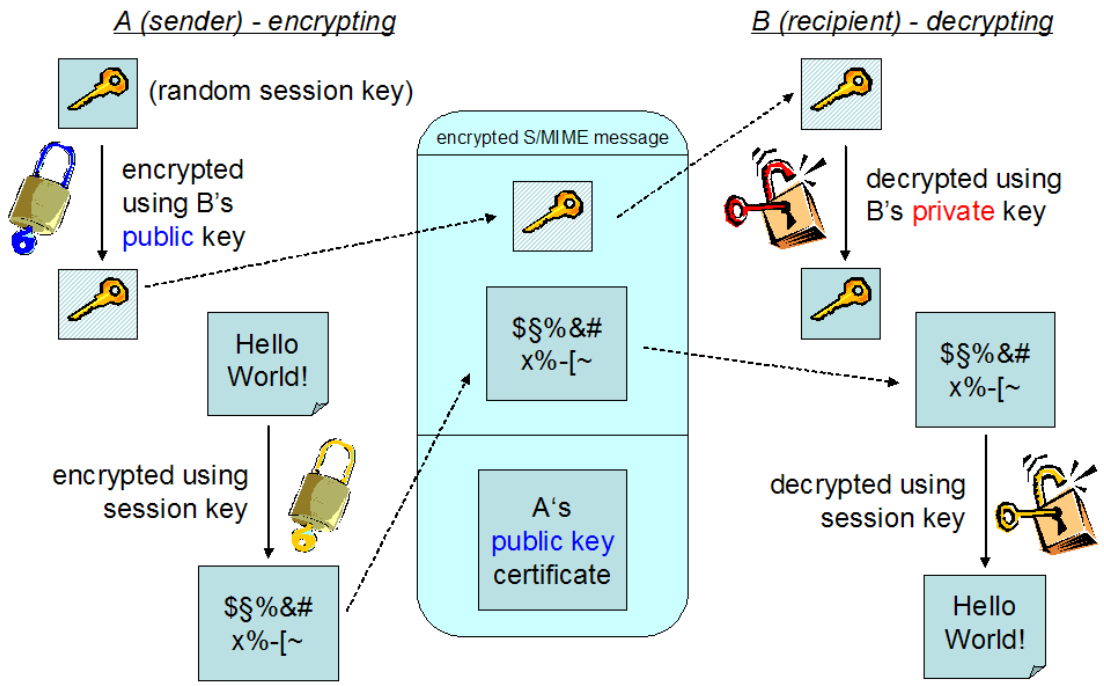

Figure 2: Encrypted Mail

- The temporary session key is being encrypted using B's public key. Therefore, only B can retrieve the session key and thus decrypt the original message.

The following steps are taken in order to create an encrypted message:

1. The user writes the message as clear-text.
2. A random session key is being created (tripleDES[6] or RC2[7])
3. The message is being encrypted using the random session key.
4. For every recipient, the session key is being encrypted using the recipient's public key (DH[8] or RSA[5]).

One of the drawbacks of encrypting all e-mail correspondence is that automatic virus scanning at mail gateways is no longer possible because the mail gateway cannot read the contents of the mail.

1.3 Cryptographic Algorithms

hash functions S/MIME-compatible e-mail software must support SHA-1[2] (Secure Hash Standard) and should support MD5[3] (Message Digest Algorithm) in order to provide backward-compatibility with MD5-digested S/MIME v2 messages.

Encrypted e-mail	
encrypted message	encrypted with the session key
encrypted session key	encrypted with the recipient's public key—can only be decrypted using the recipient's private key
algorithm identifier	to tell the receiver's software which decryption algorithm to use
sender's public key	to enable the recipient to encrypt his response

Table 2: Contents of an encrypted e-mail

digital signatures To encrypt the message digest, the S/MIME client must support DSS[4] (Digital Signature Standard) and should support RSA[5] (Rivest, Shamir, Adleman).

content encryption To encrypt the message content (symmetrically, using a random session key), tripleDES[6] must be supported and RC2[7] (40-bit, considered insecure[2]) should also be supported for compatibility reasons. Earlier S/MIME versions only required support for 40-bit RC2 encryption in order to be compatible with US export regulations.

key encryption To encrypt the session key, DH[8] (Diffie-Hellmann) must be supported, RSA[5] should also be implemented.

1.4 Key Management

S/MIME uses X.509v3 certificates to determine whether a public key used to verify a signature is trustworthy. A *certificate* (or *digital ID*) basically consists of a public key and personal information (name, e-mail address, country, ...). The certificate is signed by a *certification authority* (CA), thereby claiming that the public key really belongs to that person.

S/MIME client software includes certain root certificates, which are automatically trusted. A root certificate usually belongs to a CA and is self signed. Every public key certificate that has been signed by one of these root certificates is considered trusted.

Example. B's mail client contains root certificates of VeriSign, TC Trustcenter and Deutsche Telekom. When a signed message arrives from person A including A's public key that has been signed by VeriSign, the message would be considered authentic if the message digest verification using A's public key succeeds. If A's public key had been signed by another CA whose root certificate is not included in B's mail client's root certificate list (e.g. Thawte or a-sign), the user would get a warning message that the signature of the certificate could not be verified and therefore the sender cannot be trusted.

[2]RC2 cracking screensaver: `http://www.counterpane.com/smime.html`

The same X.509 certificates can also be used to authenticate web servers and -clients (SSL[3]) or to digitally sign software available for download.

CAs maintain CRLs[9] (Certificate Revocation Lists). When validating the certification chains of digital signatures, S/MIME clients are supposed to check whether one of the certificates has been revoked by accessing the CRLs.

2 E-Mail Standards

S/MIME extends the well-known MIME standard, which enables e-mails containing multiple contents (e.g. mixed text/HTML-mails or attachments).

2.1 RFC 822

The first[4] Internet e-mail specification which is nowadays still in use was RFC 822[10].

According to RFC 822, a message consists of a *header* followed by a blank line and the message text as plain ASCII text, called the *body*.

Example. RFC 822 compliant mail

```
From: "Donald Duck" <donald@moserware.at>
To: "Scrooge McDuck" <scrooge@moserware.at>
Subject: RFC 822-compliant message
Date: Sun, 6 Jan 2002 01:38:36 +0100

Dear Uncle Scrooge!

This is an RFC 822-compliant message.

Sincerely,
    Donald
```

However, RFC 822 messages have certain drawbacks. No 8-bit national language characters are supported, because messages are limited to 7-bit ASCII. To send binary data (e.g. non-text files), these files have to be encoded into 7-bit ASCII and included into the message text using tools such as uuencode.

2.2 MIME

A solution to these and other problems is MIME (Multipurpose Internet Mail Extensions), described in RFCs 2045 through 2049. MIME allows to include multiple contents of different types in the message body (text, HTML, images, attachments, ...).

[3] URLs starting with `https://`

[4] There are actually even older standards for text messages on the Arpanet, the Internet's predecessor, e.g. RFC 733.

MIME also specifies transfer encodings that can be used to convert 8-bit data into a 7-bit ASCII stream: *Quoted printable* encodes only non-7-bit characters, thereby leaving the rest of the text human-readable, whereas *base64*[5] encoding works by translating three 8-bit characters into four 6-bit characters.

MIME specifies certain *content types* that specify the type of content used in this section of the mail (= MIME entity). There are special *multipart* content types, which contain multiple other MIME entities themselves (and can, of course, include other multipart entities). A list of all default content types is included in RFC 2046[11].

Example. MIME mail

```
From: "Donald Duck" <donald@moserware.at>
To: "Scrooge McDuck" <scrooge@moserware.at>
Subject: MIME message
Date: Sun, 6 Jan 2002 01:38:36 +0100
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="ABCDE"

--ABCDE
Content-Type: text/plain; charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable

Dear Uncle Scrooge!

At the end of a letter, the Germans usually say:
"Mit freundlichen Gr=FC=DFen"

Sincerely,
    Donald

--ABCDE
Content-Type: image/jpeg; name="picture.jpg"
Content-Transfer-Encoding: base64

/9j/4AAQSkZJRgABAQEASABIAAD/2wBDAA0JCgsKCA0L
LSwzOko+MzZGNywtQFdBRkxOUlNSMj5aYVpQYEpRUk//
T09PT09PT09PT09PT09PT09PT09PT09PT09PT09P
[...]

--ABCDE--
```

[5] See Section 5.2 of RFC 1521.

3 S/MIME

S/MIME (Secure/Multipurpose Internet Mail Extensions) is a specification for secure electronic mail providing authentication (digital signatures) and confidentiality (encryption). S/MIME is not a particular software product but a standard designed to be implemented by various e-mail vendors, so that any two S/MIME-supporting mail clients can communicate securely.

Originally developed by RSA Data Security[6], S/MIME version 3 is now being maintained by the S/MIME Working Group of the Internet Engineering Task Force[7].

3.1 Message Format

S/MIME specifies additional MIME content types to be used for encryption and digital signatures. A MIME entity (which can be the complete message or subparts of the message) is being wrapped into an encrypted or signed CMS[8] (Cryptographic Message Syntax) object. The data structures used by CMS are described by ASN.1[9] (Abstract Syntax Notation 1).

The CMS object is usually base64 encoded and with content-type *application/pkcs7-mime*. The additional parameter "smime-type" specifies whether the message has been encrypted or signed.

If the message contains a clear-text part (i.e. a clear-signed message), the clear-text part and the CMS object are combined within a *multipart/signed* content and the CMS object is of content-type *application/pkcs7-signature*.

3.2 Examples

Original Message

```
From: "Donald Duck" <donald@moserware.at>
To: "Scrooge McDuck" <scrooge@moserware.at>
Subject: message
Date: Sun, 6 Jan 2002 01:38:36 +0100
MIME-Version: 1.0
Content-Type: text/plain

This is my plain text letter.
```

This message is neither encrypted nor signed and can be read and modified freely.

[6] `http://www.rsasecurity.com`

[7] `http://www.ietf.org/html.charters/smime-charter.html`

[8] The CMS is defined in RFC 2315, the Public Key Cryptographic Standard #7, and is also used with other protocols than S/MIME, for example SET (Secure Electronic Transaction).

[9] ISO/IEC 8824

S/MIME Clear-signed Message

```
From: "Donald Duck" <donald@moserware.at>
To: "Scrooge McDuck" <scrooge@moserware.at>
Subject: S/MIME message
Date: Sun, 6 Jan 2002 01:38:36 +0100
MIME-Version: 1.0
Content-Type: multipart/signed;
    protocol="application/x-pkcs7-signature";
    micalg=SHA1; boundary="ABCDE"

--ABCDE
Content-Type: text/plain

This is my plain text letter.

--ABCDE
Content-Type: application/x-pkcs7-signature
Content-Transfer-Encoding: base64

MIAGCSqGSIb3DQEHAqCAMIACAQExCzAJBgUrDgMCGgUAMI
ggJfoAMCAQICDnW7AAAAAjGqNhNz5/78MA0GCSqGSIb3DQ
MA4GA1UECBMHSGFtYnVyZzEQMA4GA1UEBxMHSGFtYnVyZz
[...]

--ABCDE--
```

This `application/x-pkcs7-signature` part (a CMS object) contains the sender's public key certificate, algorithm identifiers, the encrypted message digest, and can contain additional public key certificates of the CA that signed the sender's public key certificate. (See Table 1.)

The message text is contained as plain text and can therefore be read by any mail client. However, modifications would be detected because the new modified message digest would differ from the message digest included in the signature.

S/MIME Opaque-signed Message

```
From: "Donald Duck" <donald@moserware.at>
To: "Scrooge McDuck" <scrooge@moserware.at>
Subject: S/MIME message
Date: Sun, 6 Jan 2002 01:38:36 +0100
MIME-Version: 1.0
Content-Type: application/x-pkcs7-mime;
    smime-type=signed-data
Content-Transfer-Encoding: base64
```

```
MIAGCSqGSIb3DQEHAqCAMIACAQExCzAJBgUrDgMCGgUAMI
bTogIkhlaW5yaWNoIE1vc2VyIiA8aC5tb3Nlci5qdW5AbW
c2VyLmp1bkBtb3NlcndhcmUuYXQ+DQpTdWJqZWN0OiBNSU
[...]
```

This message contains the same data as the previous, clear-signed mail. The only difference is that the original message text is also included in the base64 encoded CMS object. This prevents the original message text (and thus the message digest) from being altered by message transfer agents (e.g. to change the encoding).

The message text can only be read by S/MIME-compatible mail clients who know how to remove the base64 encoding and extract the text from the CMS object. However, it is important to notice that the message text is not encrypted and can therefore be read by S/MIME-compatible monitoring software on its way through the Internet.

The message is digitally signed; therefore, modifications of the message text will be detected.

S/MIME Encrypted Message

```
From: "Donald Duck" <donald@moserware.at>
To: "Scrooge McDuck" <scrooge@moserware.at>
Subject: S/MIME message
Date: Sun, 6 Jan 2002 01:38:36 +0100
MIME-Version: 1.0
Content-Type: application/x-pkcs7-mime;
    smime-type=enveloped-data
Content-Transfer-Encoding: base64

MIAGCSqGSIb3DQEHA6CAMIACAQAxggJUMIIBJgIBADCBzz
BAgTBOhhbWJ1cmcxEDAOBgNVBAcTBOhhbWJ1cmcxOjA4Bg
ciBTZWN1cml0eSBpbiBEYXRhIE5ldHdvcmtzIEdtYkgxIj
[...]
```

This message contains the (symmetrically) encrypted message text, the (asymmetrically) encrypted session key, the algorithm identifier and the sender's public key. (See Table 2.)

Due to the message being encrypted, the message cannot be read by anyone but the intended recipient (i.e. the owner of the private key corresponding to the public key used to encrypt the session key). However, the encrypted message text can be replaced by another encrypted message text, because the public key used to encrypt the message text is usually available to everyone.

S/MIME Encrypted and Signed Message

An encrypted and signed message is first being signed and then encrypted. Therefore, an encrypted and signed message looks exactly like the above example of an encrypted message. Another consequence is that only the recipient knows that the message has been digitally signed.

Being encrypted, the message can not be read by any unauthorized person. The encrypted message text (which is actually an encrypted signed message) can still be replaced by another encrypted message text; however, the digital signature (which is included in the encrypted part of the message whose contents the malicious user cannot read) is lost during that process.

Although this is a common method of signing and encrypting messages, it is not the only one. Signed-only as well as encrypted-only messages are MIME entities and can therefore be signed or encrypted again and again. It is up to the mail client or, if the mail client offers that much flexibility, up to the user to decide whether to encrypt and then sign or to sign and encrypt afterwards.

3.3 Other S/MIME Message Formats

S/MIME also defines a MIME type for entities only containing a public key certificate (Content-Type: application/pkcs7-mime, smime-type: certs-only).

3.4 Enhanced Security Services (ESS)

There are a few optional security service extensions for S/MIME version 3 specified in RFC 2634[12].

Signed Receipts A special flag can be set for signed messages to request the return of a signed receipt. This signed receipt proves that the message has been received by the recipient and the digital signature has been successfully verified.

A signed receipt is created by signing the complete original message (including the original signature) and returning this new signature.

Security Labels Security label attributes included in a signed message specify the security classification of the signed content. They can be based on the X.411 recommendation (unmarked, unclassified, restricted, confidential, secret, top-secret) or on an organization's own security policy. Based on this information, the mail agent can decide whether or not to show the contents of the message to the user.

Mail List Management When sending encrypted messages to a large number of recipients, the mail client would have to encrypt the session key using every recipient's public key. With the mail list management extension, the user would only encrypt the message once, using the Mailing List Agent's public key. The Mailing List Agent would then encrypt and forward the message to every intended recipient.

S/MIME Mail List Management also includes Expansion Attributes to prevent mail loops between mailing list.

4 The Future

With OpenPGP[10], a similar standard is available with the major differences being the certificate handling (S/MIME: X.509 - hierarchical, OpenPGP: based on PGP - web of trust) and the message format (S/MIME: CMS/PKCS #7, OpenPGP: based on PGP).

The big advantage of PGP—no need for a certification authority—which makes it very convenient for private use, is also its biggest drawback, because for business-to-business transactions and e-commerce, hierarchically signed certificates are usually preferred.

At the moment, S/MIME is being supported by popular e-mail software such as Microsoft Outlook, Outlook Express and Netscape Messenger. With the recent signature laws providing a legal infrastructure for signed e-mail correspondence within the European Union and increased e-mail surveillance providing the need for encryption, it is obvious that secure e-mail processing will get more attention in the future than it gets now.

References

[1] Ramsdell, B.: *S/MIME Version 3 Message Specification*, RFC 2633, June 1999, `ftp://ftp.isi.edu/in-notes/rfc2633.txt`

[2] NIST: *Secure Hash Standard*, NIST FIPS PUB 180-1, April 1995, `http://www.itl.nist.gov/fipspubs/fip180-1.htm`

[3] Rivest, R.: *The MD5 Message Digest Algorithm*, RFC 1321, April 1992, `ftp://ftp.isi.edu/in-notes/rfc1321.txt`

[4] NIST: *Digital Signature Standard*, NIST FIPS PUB 186, May 1994, `http://www.itl.nist.gov/fipspubs/fip186.htm`

[5] Kaliski, B.: *PKCS #1: RSA Encryption Version 2.0*, RFC 2437, October 1998, `ftp://ftp.isi.edu/in-notes/rfc2437.txt`

[6] ANSI: *Triple Data Encryption Algorithm Modes of Operation*, ANSI X9.52-1998, 1998

[7] Rivest, R.: *A Description of the RC2 (r) Encryption Algorithm*, RFC 2268, March 1998, `ftp://ftp.isi.edu/in-notes/rfc2268.txt`

[8] Rescorla, E.: *Diffie-Hellman Key Agreement Method*, RFC 2631, June 1999, `ftp://ftp.isi.edu/in-notes/rfc2631.txt`

[9] Housley, R., Ford, W., Polk, W. and D. Solo: *Internet X.509 Public Key Infrastructure Certificate and CRL Profile*, RFC 2459, January 1999, `ftp://ftp.isi.edu/in-notes/rfc2459.txt`

[10] `http://www.imc.org/smime-pgpmime.html`

[10] Crocker, D.: *Standard for ARPA Internet Text Messages*, RFC 822, August 1982, `ftp://ftp.isi.edu/in-notes/rfc822.txt`

[11] Freed, N. and N. Borenstein: *Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types*, RFC 2046, November 1996, `ftp://ftp.isi.edu/in-notes/rfc2046.txt`

[12] Hoffman, P.: *Enhanced Security Services for S/MIME*, RFC 2634, June 1999, `ftp://ftp.isi.edu/in-notes/rfc2634.txt`