Network Working Group   T. Harding
Request for Comments: 3335   Cyclone Commerce
Category: Standards Track   R. Drummond
                                                   Drummond Group
                                                          C. Shih
                                                    Gartner Group
                                                   September 2002

                    MIME-based Secure Peer-to-Peer
                Business Data Interchange over the Internet

Status of this Memo

Copyright Notice

Abstract

   This document describes how to exchange structured business data
   securely using SMTP transport for Electronic Data Interchange, (EDI -
   either the American Standards Committee X12 or UN/EDIFACT, Electronic
   Data Interchange for Administration, Commerce and Transport), XML or
   other data used for business to business data interchange. The data
   is packaged using standard MIME content-types. Authentication and
   privacy are obtained by using Cryptographic Message Syntax (S/MIME)
   or OpenPGP security body parts. Authenticated acknowledgements make
   use of multipart/signed replies to the original SMTP message.

Table of Contents

1.0 Introduction

   Previous work on Internet EDI focused on specifying MIME content
   types for EDI data ([2] RFC 1767). This document expands on RFC 1767
   to specify use of a comprehensive set of data security features,
   specifically data privacy, data integrity/authenticity, non-
   repudiation of origin and non-repudiation of receipt. This document
   also recognizes contemporary RFCs and is attempting to "re-invent" as
   little as possible. While this document focuses specifically on EDI
   data, any other data type is also supported.

   With an enhancement in the area of "receipts", as described below
   (5.2), secure Internet MIME based EDI can be accomplished by using
   and complying with the following RFCs:

      -RFC 821 SMTP
      -RFC 822 Text Message Formats
      -RFC 1767 EDI Content Type
      -RFC 1847 Security Multiparts for MIME
      -RFC 1892 Multipart/Report
      -RFC 2015, 3156, 2440 MIME/PGP

      -RFC 2045 to 2049 MIME RFCs
      -RFC 2298 Message Disposition Notification
      -RFC 2630, 2633 S/MIME v3 Specification

   Our intent here is to define clearly and precisely how these are used
   together, and what is required by user agents to be compliant with
   this document.

   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 RFC 2119.

2.0 Overview

2.1 Purpose of a Security Guideline for MIME EDI

   The purpose of these specifications is to ensure interoperability
   between EDI user agents, invoking some or all of the commonly
   expected security features. This document is also NOT limited to
   strict EDI use, but applies to any electronic commerce application
   where business data needs to be exchanged over the Internet in a
   secure manner.

2.2 Definitions

2.2.1 Terms

   EDI   Electronic Data Interchange

   EC   Electronic Commerce

   Receipt   The functional message that is sent from a
                       receiver to a sender to acknowledge
                       receipt of an EDI/EC interchange.

   Signed Receipt   Same as above, but with a digital
                       signature.

   Message Disposition The Internet messaging format used to
   Notification   convey a receipt. This term is used
                       interchangeably with receipt. A signed
                       MDN is a signed receipt.

   Non-repudiation of NRR is a "legal event" that occurs when
   Receipt (NRR)   the original sender of an EDI/EC
                       interchange has verified the signed
                       receipt coming back from the receiver.
                       NRR IS NOT a functional or a technical
                       message.

   PGP/MIME   Digital envelope security based on the
                       Pretty Good Privacy (PGP) standard
                       (Zimmerman), integrated with MIME Security
                       Multiparts [6].

   S/MIME   A format and protocol for adding
                       Cryptographic signature and/or encryption
                       services to Internet MIME messages.

2.2.2 The secure transmission loop

   This document's focus is on the formats and protocols for exchanging
   EDI content that has had security applied to it using the Internet's
   messaging environment.

   The "secure transmission loop" for EDI involves one organization
   sending a signed and encrypted EDI interchange to another
   organization, requesting a signed receipt, followed later by the
   receiving organization sending this signed receipt back to the
   sending organization. In other words, the following transpires:

     -The organization sending EDI/EC data signs and encrypts the data
      using either PGP/MIME or S/MIME. In addition, the message will
      request a signed receipt to be returned to the sender of the
      message.

     -The receiving organization decrypts the message and verifies the
      signature, resulting in verified integrity of the data and
      authenticity of the sender.

     -The receiving organization then returns a signed receipt to the
      sending organization in the form of a message disposition
      notification message. This signed receipt will contain the hash
      of the signature from the received message, indicating to the
      sender that the received message was verified and/or decrypted
      properly.

   The above describes functionality which, if implemented, would
   satisfy all security requirements. This specification, however,
   leaves full flexibility for users to decide the degree to which they
   want to deploy those security features with their trading partners.

2.2.3 Definition of receipts

   The term used for both the functional activity and message for
   acknowledging receipt of an EDI/EC interchange is receipt, or signed
   receipt. The first term is used if the acknowledgment is for an
   interchange resulting in a receipt which is NOT signed. The second
   term is used if the acknowledgment is for an interchange resulting in
   a receipt which IS signed. The method used to request a receipt or a
   signed receipt is defined in RFC 2298, "An Extensible Message Format
   for Message Disposition Notifications".

   The "rule" is:

     - If a receipt is requested, explicitly specifying that the receipt
       be signed, then the receipt MUST be returned with a signature.

        - If a receipt is requested, explicitly specifying that the receipt
          be signed, but the recipient cannot support the requested
          protocol format or requested MIC algorithms, then a receipt,
          either signed or unsigned SHOULD be returned.

        - If a signature is not explicitly requested, or if the signed
          receipt request parameter is not recognized by the UA, a receipt
          may or may not be returned. This behavior is consistent with the
          MDN RFC 2298.

   A term often used in combination with receipts is "Non-Repudiation of
   Receipt (NRR). NRR refers to a legal event which occurs only when
   the original sender of an interchange has verified the signed receipt
   coming back from recipient of the message. Note that NRR is not
   possible without signatures.

2.3 Assumptions

2.3.1 EDI Process Assumptions

   -Encrypted object is an EDI Interchange
    This specification assumes that a typical EDI interchange is the
    lowest level object that will be subject to security services.

    In ANSI X12, this means anything between, and including segments ISA
    and IEA. In EDIFACT, this means anything between, and including,
    segments UNA/UNB and UNZ. In other words, the EDI interchanges
    including envelope segments remain intact and unreadable during
    secure transport.

   -EDI envelope headers are encrypted
    Congruent with the above statement, EDI envelope headers are NOT
    visible in the MIME package. In order to optimize routing from
    existing commercial EDI networks (called Value Added Networks or
    VANs) to the Internet, work may need to be done in the future to
    define ways to pull out some of the envelope information to make
    them visible; however, this specification does not go into any
    detail on this.

   -X12.58 and UN/EDIFACT security considerations
    The most common EDI standards bodies, ANSI X12 and EDIFACT, have
    defined internal provisions for security. X12.58 is the security
    mechanism for ANSI X12 and AUTACK provides security for EDIFACT.
    This specification DOES NOT dictate use or non-use of these security
    standards. They are both fully compatible, though possibly
    redundant, with this specification.

2.3.2 Flexibility Assumptions

   -Encrypted or unencrypted data

    This specification allows for EDI message exchange where the EDI
    data can either be un-protected or protected by means of encryption.

   -Signed or unsigned data

    This specification allows for EDI message exchange with or without
    digital signature of the original EDI transmission.

   -Use of receipt or not

    This specification allows for EDI message transmission with or
    without a request for receipt notification. If a signed receipt
    notification is requested however, a mic value is REQUIRED as part
    of the returned receipt, unless an error condition occurs in which a
    mic value cannot be returned. In error cases, an un-signed receipt
    or MDN SHOULD be returned with the correct "disposition modifier"
    error value.

   -Formatting choices

    This specification defines the use of two methods for formatting EDI
    contents that have security applied to it:

    -PGP/MIME
    -S/MIME

    This specification relies on the guidelines set forth in RFC
    2015/3156/2440, as reflected in [4] "MIME Security with Pretty Good
    Privacy" (PGP); OpenPGP Message Format, and RFC 2633/2630 [8]
    "S/MIME Version 3 Message Specification; Cryptographic Message
    Syntax". PGP/MIME or S/MIME as defined in this Applicability
    statement.

   -Hash function, message digest choices

    When a signature is used, it is RECOMMENDED that the SHA1 hash
    algorithm be used for all outgoing messages, and that both MD5 and
    SHA1 be supported for incoming messages.

    In summary, the following eight permutations are possible in any
    given trading relationship:

    (1) Sender sends unencrypted data, does NOT request a receipt.

   (2) Sender sends unencrypted data, requests a signed or unsigned
       receipt. The receiver sends back the signed or unsigned
       receipt.

   (3) Sender sends encrypted data, does NOT request a receipt.

   (4) Sender sends encrypted data, requests a signed or unsigned
       receipt. The receiver sends back the signed or unsigned
       receipt.

   (5) Sender sends signed data, does NOT request a signed or unsigned
       receipt.

   (6) Sender sends signed data, requests a signed or unsigned receipt.
       Receiver sends back the signed or unsigned receipt.

   (7) Sender sends encrypted and signed data, does NOT request a
       signed or unsigned receipt.

   (8) Sender sends encrypted and signed data, requests a signed or
       unsigned receipt. Receiver sends back the signed or unsigned
       receipt.

   NOTE: Users can choose any of the eight possibilities, but only
   example (8), when a signed receipt is requested, offers the whole
   suite of security features described in the "Secure transmission
   loop" above.

3.0 Referenced RFCs and Their Contribution

3.1 RFC 821 SMTP [7]

   This is the core mail transfer standard that all MTAs need to adhere
   to.

3.2 RFC 822 Text Message Format [3]

   Defines message header fields and the parts making up a message.

3.3 RFC 1847 MIME Security Multiparts [6]

   This document defines security multiparts for MIME:
   multipart/encrypted and multipart/signed.

3.4 RFC 1892 Multipart/report [9]

   This RFC defines the use of the multipart/report content type,
   something that the MDN RFC 2298 builds upon.

3.5 RFC 1767 EDI Content [2]

   This RFC defines the use of content type "application" for ANSI X12
   (application/EDI-X12), EDIFACT (application/EDIFACT) and mutually
   defined EDI (application/EDI-Consent).

3.6 RFC 2015, 3156, 2440 PGP/MIME [4]

   These RFCs define the use of content types "multipart/encrypted",
   "multipart/signed", "application/pgp encrypted" and
   "application/pgp-signature" for defining MIME PGP content.

3.7 RFC 2045, 2046, and 2049 MIME [1]

   These are the basic MIME standards, upon which all MIME related RFCs
   build, including this one. Key contributions include definition of
   "content type", "sub-type" and "multipart", as well as encoding
   guidelines, which establishes 7-bit US-ASCII as the canonical
   character set to be used in Internet messaging.

3.8 RFC 2298 Message Disposition Notification [5]

   This Internet RFC defines how a message disposition notification
   (MDN) is requested, and the format and syntax of the MDN. The MDN is
   the basis upon which receipts and signed receipts are defined in this
   specification.

3.9 RFC 2633 and 2630 S/MIME Version 3 Message Specifications [8]

   This specification describes how MIME shall carry CMS Objects.

4.0 Structure of an EDI MIME Message - Applicability

4.1 Introduction

   The structures below are described hierarchically in terms of which
   RFC's are applied to form the specific structure. For details of how
   to code in compliance with all RFC's involved, turn directly to the
   RFC's referenced.

   Also, these structures describe the initial transmission only.
   Receipts, and requests for receipts are handled in section 5.

4.2 Structure of an EDI MIME Message - PGP/MIME

4.2.1 No Encryption, No Signature

```
   -RFC822/2045
     -RFC1767 (application/EDIxxxx or /xml)
```

4.2.2 No Encryption, Signature

```
   -RFC822/2045
     -RFC1847 (multipart/signed)
       -RFC1767 (application/EDIxxxx or /xml)
       -RFC2015/2440/3156 (application/pgp-signature)
```

4.2.3 Encryption, No Signature

```
   -RFC822/2045
     -RFC1847 (multipart/encrypted)
       -RFC2015/2440/3156 (application/pgp-encrypted)
         -"Version: 1"
       -RFC2015/2440/3156 (application/octet-stream)
         -RFC1767 (application/EDIxxxx or /xml) (encrypted)
```

4.2.4 Encryption, Signature

```
   -RFC822/2045
     -RFC1847 (multipart/encrypted)
       -RFC2015/2440/3156 (application/pgp-encrypted)
         -"Version: 1"
       -RFC2015/2440/3156 (application/octet-stream)
         -RFC1847 (multipart/signed)(encrypted)
           -RFC1767 (application/EDIxxxx or /xml)(encrypted)
           -RFC2015/2440/3156 (application/pgp-signature)(encrypted)
```

4.3 Structure of an EDI MIME Message - S/MIME

4.3.1 No Encryption, No Signature

```
   -RFC822/2045
     -RFC1767 (application/EDIxxxx or /xml)
```

4.3.2 No Encryption, Signature

```
   -RFC822/2045
     -RFC1847 (multipart/signed)
       -RFC1767 (application/EDIxxxx or /xml)
       -RFC2633 (application/pkcs7-signature)
```

4.3.3 Encryption, No Signature

```
-RFC822/2045
  -RFC2633 (application/pkcs7-mime)
    -RFC1767 (application/EDIxxxx or /xml) (encrypted)
```

4.3.4 Encryption, Signature

```
-RFC822/2045
  -RFC2633 (application/pkcs7-mime)
    -RFC1847 (multipart/signed) (encrypted)
      -RFC1767 (application/EDIxxxx or /xml) (encrypted)
      -RFC2633 (application/pkcs7-signature) (encrypted)
```

5.0 Receipts

5.1 Introduction

   In order to support non-repudiation of receipt (NRR), a signed
   receipt, based on digitally signing a message disposition
   notification, is to be implemented by a receiving trading partner's
   UA (User Agent). The message disposition notification, specified by
   RFC 2298 is digitally signed by a receiving trading partner as part
   of a multipart/signed MIME message.

   The following support for signed receipts is REQUIRED:

   1) The ability to create a multipart/report; where the report-type =
      disposition-notification.

   2) The ability to calculate a message integrity check (MIC) on the
      received message. The calculated MIC value will be returned to
      the sender of the message inside the signed receipt.

   4) The ability to create a multipart/signed content with the message
      disposition notification as the first body part, and the signature
      as the second body part.

   5) The ability to return the signed receipt to the sending trading
      partner.

   The signed receipt is used to notify a sending trading partner that
   requested the signed receipt that:

   1) The receiving trading partner acknowledges receipt of the sent EDI
      Interchange.

2) If the sent message was signed, then the receiving trading partner
   has authenticated the sender of the EDI Interchange.

3) If the sent message was signed, then the receiving trading partner
   has verified the integrity of the sent EDI Interchange.

Regardless of whether the EDI Interchange was sent in S/MIME or
PGP/MIME format, the receiving trading partner's UA MUST provide the
following basic processing:

1) If the sent EDI Interchange is encrypted, then the encrypted
   symmetric key and initialization vector (if applicable) is
   decrypted using the receiver's private key.

2) The decrypted symmetric encryption key is then used to decrypt the
   EDI Interchange.

3) The receiving trading partner authenticates signatures in a
   message using the sender's public key. The authentication
   algorithm performs the following:

   a) The message integrity check (MIC or Message Digest), is
      decrypted using the sender's public key.

   b) A MIC on the signed contents (the MIME header and encoded EDI
      object, as per RFC 1767) in the message received is calculated
      using the same one-way hash function that the sending trading
      partner used.

   c) The MIC extracted from the message that was sent, and the MIC
      calculated using the same one-way hash function that the
      sending trading partner used is compared for equality.

4) The receiving trading partner formats the MDN and sets the
   calculated MIC into the "Received-content-MIC" extension field.

5) The receiving trading partner creates a multipart/signed MIME
   message according to RFC 1847.

6) The MDN is the first part of the multipart/signed message, and the
   digital signature is created over this MDN, including its MIME
   headers.

   7) The second part of the multipart/signed message contains the
      digital signature. The "protocol" option specified in the second
      part of the multipart/signed is as follows:

      S/MIME: protocol = "application/pkcs-7-signature"

      PGP/MIME: protocol = "application/pgp-signature"

   8) The signature information is formatted according to S/MIME or
      PGP/MIME specifications.

   The EDI Interchange and the RFC 1767 MIME EDI content header, can
   actually be part of a multi-part MIME content-type. When the EDI
   Interchange is part of a multi-part MIME content-type, the MIC MUST
   be calculated across the entire multi-part content, including the
   MIME headers.

   The signed MDN, when received by the sender of the EDI Interchange
   can be used by the sender:

   1) As an acknowledgment that the EDI Interchange sent, was delivered
      and acknowledged by the receiving trading partner. The receiver
      does this by returning the original message id of the sent message
      in the MDN portion of the signed receipt.

   2) As an acknowledgment that the integrity of the EDI Interchange was
      verified by the receiving trading partner. The receiver does this
      by returning the calculated MIC of the received EDI Interchange
      (and 1767 MIME headers) in the "Received-content-MIC" field of the
      signed MDN.

   3) As an acknowledgment that the receiving trading partner has
      authenticated the sender of the EDI Interchange.

   4) As a non-repudiation of receipt when the signed MDN is
      successfully verified by the sender with the receiving trading
      partner's public key and the returned mic value inside the MDN is
      the same as the digest of the original message.

5.2 Requesting a Signed Receipt

   Message Disposition Notifications are requested as per RFC 2298,

   "An Extensible Message Format for Message Disposition Notification".
   A request that the receiving user agent issue a message disposition
   notification is made by placing the following header into the message
   to be sent:

```
MDN-request-header = "Disposition-notification-to" ":"
                        mail-address
```

The mail-address field is specified as an RFC 822 user@domain
address, and is the return address for the message disposition
notification.

In addition to requesting a message disposition notification, a
message disposition notification that is digitally signed, or what
has been referred to as a signed receipt, can be requested by placing
the following in the message header following the "Disposition-
Notification-To" line.

```
Disposition-notification-options =
      "Disposition-Notification-Options" ":"
      disposition-notification-parameters
```

where

```
  disposition-notification-parameters =
                    parameter *(";" parameter)
```

where

```
  parameter = attribute "=" importance ", " 1#value"
```

where

```
  importance = "required" | "optional"
```

So the Disposition-notification-options string could be:

```
  signed-receipt-protocol=optional, <protocol symbol>;
  signed-receipt-micalg=optional, <micalg1>, <micalg2>,...;
```

The currently supported values for <protocol symbol> are
"pkcs7-signature", for the S/MIME detached signature format, or
"pgp-signature", for the pgp signature format.

The currently supported values for MIC algorithm values are:

```
Algorithm   Value
used

MD5         md5
SHA-1       sha1
```

(Historical Note: Some early implementations of EDIINT emitted and
expected "rsa-md5" and "rsa-sha1" for the micalg parameter.)
Receiving agents SHOULD be able to recover gracefully from a micalg
parameter value that they do not recognize.

An example of a formatted options line would be as follows:

Disposition-notification-options:
  signed-receipt-protocol=optional, pkcs7-signature;
  signed-receipt-micalg=optional, sha1, md5

The semantics of the "signed-receipt-protocol" parameter is as
follows:

1) The "signed-receipt-protocol" parameter is used to request a
   signed receipt from the recipient trading partner. The
   "signed-receipt-protocol" parameter also specifies the format in
   which the signed receipt should be returned to the requester.

   The "signed-receipt-micalg" parameter is a list of MIC algorithms
   preferred by the requester for use in signing the returned
   receipt. The list of MIC algorithms should be honored by the
   recipient from left to right.

   Both the "signed-receipt-protocol" and the "signed-receipt-micalg"
   option parameters are REQUIRED when requesting a signed receipt.

2) The "importance" attribute of "Optional" is defined in the MDN RFC
   2298 and has the following meaning:

   Parameters with an importance of "Optional" permit a UA that does
   not understand the particular options parameter to still generate
   a MDN in response to a request for a MDN. A UA that does not
   understand the "signed-receipt-protocol" parameter, or the
   "signed-receipt-micalg" will obviously not return a signed
   receipt.

   The importance of "Optional" is used for the signed receipt
   parameters because it is RECOMMENDED that an MDN be returned to
   the requesting trading partner even if the recipient could not
   sign it.

   The returned MDN will contain information on the disposition of
   the message as well as why the MDN could not be signed. See the
   Disposition field in section 5.3 for more information.

Within an EDI trading relationship, if a signed receipt is
expected and is not returned, then the validity of the transaction
is up to the trading partners to resolve. In general, if a signed
receipt is required in the trading relationship and is not
received, the transaction will likely not be considered valid.

5.2.1 Additional Signed Receipt Considerations

The "rules" stated in Section 2.2.3 for signed receipts are as
follows:

1) When a receipt is requested, explicitly specifying that the
   receipt be signed, then the receipt MUST be returned with a
   signature.

2) When a receipt is requested, explicitly specifying that the
   receipt be signed, but the recipient cannot support either the
   requested protocol format, or requested MIC algorithms, then
   either a signed or unsigned receipt SHOULD be returned.

3) When a signature is not explicitly requested, or if the signed
   receipt request parameter is not recognized by the UA, then no
   receipt, an unsigned receipt, or a signed receipt MAY be returned
   by the recipient.

NOTE: For Internet EDI, it is RECOMMENDED that when a signature is
not explicitly requested, or if parameters are not recognized, that
the UA send back at a minimum, an unsigned receipt. If a signed
receipt however was always returned as a policy, whether requested or
not, then any false unsigned receipts can be repudiated.

When a request for a signed receipt is made, but there is an error in
processing the contents of the message, a signed receipt MUST still
be returned. The request for a signed receipt SHALL still be
honored, though the transaction itself may not be valid. The reason
for why the contents could not be processed MUST be set in the
"disposition-field".

When a request for a signed receipt is made, the
"Received-content-MIC" MUST always be returned to the requester.
The"Received-content-MIC" MUST be calculated as follows:

- For any signed messages, the MIC to be returned is calculated on
  the RFC1767 MIME header and content. Canonicalization as specified
  in RFC 1848 MUST be performed before the MIC is calculated, since
  the sender requesting the signed receipt was also REQUIRED to
  canonicalize.

- For encrypted, unsigned messages, the MIC to be returned is
  calculated on the decrypted RFC 1767 MIME header and content. The
  content after decryption MUST be canonicalized before the MIC is
  calculated.

- For unsigned, unencrypted messages, the MIC MUST be calculated over
  the message contents prior to Content-Transfer-Encoding and without
  the MIME or any other RFC 822 headers, since these are sometimes
  altered or reordered by MTAs.

5.3 Message Disposition Notification Format

   The format of a message disposition notification is specified in RFC
   2298. For use in Internet EDI, the following format will be used:

   - content-type - per RFC 1892 and the RFC 2298 specification

   - reporting-ua-field - per RFC 2298 specification

   - MDN-gateway-field - per RFC 2298 specification

   - original-recipient-field - per RFC 2298 specification

   - final-recipient-field - per RFC 2298 specification

   - original-message-id-field - per RFC 2298 specification

   - disposition-field - the following "disposition-mode" values SHOULD
                          be used for Internet EDI:

     "automatic-action" - The disposition described by the disposition
                          type was a result of an automatic action,
                          rather than an explicit instruction by the
                          user for this message.

     "manual-action"   - The disposition described by the disposition
                          type was a result of an explicit instruction
                          by the user rather than some sort of
                          automatically performed action.

     "MDN-sent-automatically" - The MDN was sent because the UA had
                                previously been configured to do so.

     "MDN-sent-manually" - The user explicitly gave permission for this
                           particular MDN to be sent.
                           "MDN-sent-manually" is meaningful with
                           "manual-action", but not with
                           "automatic-action".

- disposition-field - the following "disposition-type" values SHOULD
                    be used for Internet EDI:

   "processed" - The message has been processed in some manner (e.g.,
                 printed, faxed, forwarded, gatewayed) without being
                 displayed to the user. The user may or may not see
                 the message later.

   "failed" - A failure occurred that prevented the proper generation
               of an MDN. More information about the cause of the
               failure may be contained in a Failure field. The
               "failed" disposition type is not to be used for the
               situation in which there is some problem in processing
               the message other than interpreting the request for an
               MDN. The "processed" or other disposition type with
               appropriate disposition modifiers is to be used in such
               situations.

- disposition-field - the following "disposition-modifier" values
                    SHOULD be used for Internet EDI:

   "error" - An error of some sort occurred that prevented successful
              processing of the message. Further information is
              contained in an Error field.

   "warning" - The message was successfully processed but some sort of
               exceptional condition occurred. Further Information is
               contained in a Warning field.

5.3.1 Message Disposition Notification Extensions

   The following "extension field" will be added in order to support
   signed receipts for RFC 1767 MIME content type and multipart MIME
   content types that include the RFC 1767 MIME content type. The
   extension field" defined below follows the "disposition-field" in the
   MDN.

   The "Received-content-MIC" extension field is set when the integrity
   of the received message is verified. The MIC is the base64 encoded
   quantity computed over the received message with a hash function.
   For details of "what" the "Received-content-MIC" should be calculated
   over, see Section 5.2.1. The algorithm used to calculate the
   "Received-content-MIC" value MUST be the same as the "micalg" value
   used by the sender in the multipart/signed message. When no
   signature is received, or the mic-alg parameter is not supported then
   it is RECOMMENDED that the SHA1 algorithm be used to calculate the
   MIC on the received message or message contents.

This field is set only when the contents of the message are processed
successfully. This field is used in conjunction with the recipient's
signature on the MDN in order for the sender to verify "non-
repudiation of receipt".

- extension field = "Received-content-MIC" ":" MIC

  where:

  <MIC> = <base64MicValue> "," <micalg>

  <base64MicValue> = the result of one way hash function, base64
                     encoded.

  < micalg> = the micalg value defined in RFC1847, an IANA
              registered MIC algorithm ID token.

5.3.2 Disposition Mode, Type, and Modifier Use

   Guidelines for use of the "disposition-mode", "disposition-type", and
   "disposition-modifier" fields within Internet EDI are discussed in
   this section. The "disposition-mode", "disposition-type', and
   "disposition-modifier' fields are described in detail in RFC 2298.
   The "disposition-mode', "disposition-type" and "disposition-modifier"
   values SHOULD be used as follows:

5.3.2.1 Successful Processing

   When the request for a receipt or signed receipt, and the received
   message contents are successfully processed by the receiving EDI UA,
   a receipt or MDN SHOULD be returned with the "disposition-type" set
   to there is no explicit way for a user to control the sending of the
   MDN, then the first part of the "disposition-mode" should be set to
   "automatic-action". When the MDN is being sent under user
   configurable control, then the first part of the "disposition-mode"
   should be set to "manual-action". Since a request for a signed
   receipt should always be honored, the user MUST not be allowed to
   configure the UA to not send a signed receipt when the sender
   requests one.

   The second part of the "disposition-mode" is set to "MDN-sent-
   manually" if the user gave explicit permission for the MDN to be
   sent. Again, the user MUST not be allowed to explicitly refuse to
   send a signed receipt when the sender requests one. The second part
   of the "disposition-mode" is set to "MDN-sent-automatically" whenever
   the EDI UA sends the MDN automatically, regardless of whether the
   sending was under a user's, administrator's, or under software
   control.

Since EDI content is generally handled automatically by the EDI UA, a
request for a receipt or signed receipt will generally return the
following in the "disposition-field":

   Disposition: automatic-action/MDN-sent-automatically; processed

Note this specification does not restrict the use of the
"disposition-mode" to just automatic actions. Manual actions are
valid as long as it is kept in mind that a request for a signed
receipt MUST be honored.

5.3.2.2 Unprocessed Content

The request for a signed receipt requires the use of two
"disposition-notification-options", which specify the protocol format
of the returned signed receipt, and the MIC algorithm used to
calculate the mic over the message contents. The "disposition-field"
values that should be used in the case where the message content is
being rejected or ignored, for instance if the EDI UA determines that
a signed receipt cannot be returned because it does not support the
requested protocol format, so the EDI UA chooses not to process the
message contents itself, should be specified in the MDN
"disposition-field" as follows:

Disposition: "disposition-mode";
  failed/Failure: unsupported format

The syntax of the "failed" "disposition-type" is general, allowing
the sending of any textual information along with the "failed"
"disposition-type". For use in Internet EDI, the following "failed"
values are defined:

"Failure: unsupported format" "Failure: unsupported MIC-algorithms"

5.3.2.3 Content Processing Errors

When errors occur processing the received message content, the
"disposition-field" should be set to the "processed" "disposition-
type" value and the "error" "disposition-modifier" value. For use in
Internet EDI, the following "error" "disposition-modifier" values are
defined:

"Error: decryption-failed" - the receiver could not decrypt the
                             message contents.

"Error: authentication-failed" - the receiver could not authenticate
                                 the sender.

"Error: integrity-check-failed" - the receiver could not verify
                                content integrity.

"Error: unexpected-processing-error" - a catch-all for any additional
                                     processing errors.

An example of how the "disposition-field" would look when content
processing errors are detected is as follows:

Disposition: "disposition-mode";
  processed/Error: decryption-failed

5.3.2.4 Content Processing Warnings

Situations arise in EDI where even if a trading partner cannot be
authenticated correctly, the trading partners still agree to continue
processing the EDI transactions. Transaction reconciliation is done
between the trading partners at a later time. In the content
processing warning situations as described above, the "disposition-
field" SHOULD be set to the "processed" "disposition-type" value, and
the "warning" "disposition-modifier" value. For use in Internet EDI,
the following "warning" "disposition-modifier" values are defined:

"Warning: authentication-failed, processing continued"

An example of how the "disposition-field" would look when content
processing warnings are detected is as follows:

Disposition: "disposition-mode"; processed/Warning:
            authentication-failed, processing continued

5.4 Message Disposition Notification Processing

5.4.1 Large File Processing

Large EDI Interchanges sent via SMTP can be automatically fragmented
by some message transfer agents. A subtype of message/partial, is
defined in RFC 2045 [1] to allow large objects to be delivered as
separate pieces of mail and to be automatically reassembled by the
receiving user agent. Using message/partial, can help alleviate
fragmentation of large messages by different message transfer agents,
but does not completely eliminate the problem. It is still possible
that a piece of a partial message, upon re-assembly, may prove to
contain a partial message as well. This is allowed by the Internet
standards, and it is the responsibility of the user agent to
reassemble the fragmented pieces.

It is RECOMMENDED that the size of the EDI Interchange sent via SMTP
be configurable so that if fragmentation is needed, then
message/partial can be used to send the large EDI Interchange in
smaller pieces. RFC 2045 [1] defines the use of Content-Type:
message/partial.

Note: Support of the message/partial content type for use in Internet
EDI is OPTIONAL and in the absence of knowledge that the recipient
supports partial it SHOULD NOT be used.

The receiving UA is required to re-assemble the original message
before sending the message disposition notification to the original
sender of the message. A message disposition notification is used to
specify the disposition of the entire message that was sent, and
should not be returned by a processing UA until the entire message is
received, even if the received message requires re-assembling.

5.4.2 Example

The following is an example of a signed receipt returned by a UA
after successfully processing a MIME EDI content type. The sending
trading partner has requested a return signed receipt.

This example follows the S/MIME application/pkcs-7-signature format.

NOTE: This example is provided as an illustration only, and is not
considered part of the protocol specification. If an example
conflicts with the protocol definitions specified above or in the
other referenced RFCs, the example is wrong.

```
    To: <recipient email>
    Subject:
    From: <sender email>
    Date: <date>
    Mime-Version: 1.0
    Content-Type: multipart/signed; boundary="separator";
      micalg=sha1; protocol="application/pkcs7-signature"

    --separator
  & Content-Type:  multipart/report; report-type=disposition
  &   notification;  boundary="xxxxx"
  &
  & --xxxxx
  & Content-Type: text/plain
  &
  & The message sent to Recipient <Recipient@cyclonesoftware.com>
  & has been received, the EDI Interchange was successfully
  & decrypted and its integrity was verified.  In addition, the
```

```
& sender of the message, Sender <Edi_Sender@cyclonesoftware.com>
& was authenticated as the originator of the message.  There is
& no guarantee however that the EDI Interchange was
& syntactically correct, or was received by the EDI
& application.
&
& --xxxxx
& Content-Type:  message/disposition-notification
&
& Reporting-UA: Interchange.cyclonesoftware.com (CI 2.2)
& Original-Recipient: rfc822; Edi_Recipient@cyclonesoftware.com
& Final-Recipient: rfc822;  Edi_Recipient@cyclonesoftware.com
& Original-Message-ID: <17759920005.12345@cyclonesoftware.com >
& Disposition: automatic-action/MDN-sent-automatically; processed
& Received-content-MIC: Q2hlY2sgSW50Xwdyaxriq, sha1
&
& --xxxxx
& Content-Type: message/rfc822
&
& To: <recipient email>
& Subject:
&
&  [additional header fields go here]
&
& --xxxxx--

  --separator
  Content-Type: application/pkcs7-signature; name=smime.p7s;
  Content-Transfer-Encoding: base64
  Content-Disposition: attachment; filename=smime.p7s

  MIIHygYJKoZIhvcNAQcDoIIHuzCCB7cCAQAxgfIwge8CAQAwg
  ZgwgYMxFjAUBgNVBAMTDVRlcnJ5IEhhcmRpbmcxEDAOBgNVBA
  oTB0NZQ0xPTkUxDDAKBgNVBAsTA04vQTEQMA4GA1UEBxMHU=

  --separator--
```

Notes:

-The lines preceded with "&" is what the signature is calculated
 over.

 (For details on how to prepare the multipart/signed with protocol =
 "application/pkcs7-signature" see the "S/MIME Message Specification,
 PKCS Security Services for MIME".)

   Note: As specified by RFC 1892 [9], returning the original or
   portions of the original message in the third body part of the
   multipart/report is not required. This is an optional body part. It
   is RECOMMENDED that the received headers from the original message be
   placed in the third body part, as they can be helpful in tracking
   problems.

   Also note that the textual first body part of the multipart/report
   can be used to include a more detailed explanation of the error
   conditions reported by the disposition headers. The first body part
   of the multipart/report when used in this way, allows a person to
   better diagnose a problem in detail.

6.0 Public Key Certificate Handling

6.1 Near Term Approach

   In the near term, the exchange of public keys and certification of
   these keys must be handled as part of the process of establishing a
   trading partnership. The UA and/or EDI application interface must
   maintain a database of public keys used for encryption or signatures,
   in addition to the mapping between EDI trading partner ID and RFC 822
   [3] email address. The procedures for establishing a trading
   partnership and configuring the secure EDI messaging system might
   vary among trading partners and software packages.

   For systems which make use of X.509 certificates, it is RECOMMENDED
   that trading partners self-certify each other if an agreed upon
   certification authority is not used. It is highly RECOMMENDED that
   when trading partners are using S/MIME, that they also exchange
   public key certificates using the recommendations specified in the
   S/MIME Version 3 Message Specification. The message formats and
   S/MIME conformance requirements for certificate exchange are
   specified in this document.

   This applicability statement does NOT require the use of a
   certification authority. The use of a certification authority is
   therefore OPTIONAL.

6.2 Long Term Approach

   In the long term, additional Internet-EDI standards may be developed
   to simplify the process of establishing a trading partnership,
   including the third party authentication of trading partners, as well
   as attributes of the trading relationship.

7.0 Security Considerations

   This entire document is concerned with secure transport of business
   to business data, and considers both privacy and authentication
   issues.

   Extracted from S/MIME Version 2 Message Specification:

   40-bit encryption is considered weak by most cryptographers. Using
   weak cryptography offers little actual security over sending plain
   text. However, other features of S/MIME, such as the specification
   of tripleDES or AES and the ability to announce stronger
   cryptographic capabilities to parties with whom you communicate,
   allow senders to create messages that use strong encryption. Using
   weak cryptography is never recommended unless the only alternative is
   no cryptography. When feasible, sending and receiving agents should
   inform senders and recipients the relative cryptographic strength of
   messages.

   Extracted from S/MIME Version 2 Certificate Handling:

   When processing certificates, there are many situations where the
   processing might fail. Because the processing may be done by a user
   agent, a security gateway, or other program, there is no single way
   to handle such failures. Just because the methods to handle the
   failures has not been listed, however, the reader should not assume
   that they are not important. The opposite is true: if a certificate
   is not provably valid and associated with the message, the processing
   software should take immediate and noticeable steps to inform the end
   user about it.

   Some of the many places where signature and certificate checking
   might fail include:

   - no certificate chain leads to a trusted CA
   - no ability to check the CRL for a certificate
   - an invalid CRL was received
   - the CRL being checked is expired
   - the certificate is expired
   - the certificate has been revoked

   There are certainly other instances where a certificate may be
   invalid, and it is the responsibility of the processing software to
   check them all thoroughly, and to decide what to do if the check
   fails.

8.0 Acknowledgments

   Many thanks go out to the previous authors of the MIME-based Secure
   EDI IETF Draft: Mats Jansson.

   The authors would like to extend special thanks to Carl Hage, Jun
   Ding, Dale Moberg, and Karen Rosenthal for providing the team with
   valuable, and very thorough feedback. Without participants like
   those cited above, these efforts become hard to complete in a way
   useful to the users and implementers of the technology.

   In addition, the authors would like to thank Harald Alvestrand, Jim
   Galvin, and Roger Fajman for their guidance and input.

9.0 References

   [1] Borenstein, N. and N. Freed, "Multipurpose Internet Mail
        Extensions (MIME) Part One: Format of Internet Message Bodies",
        RFC 2045, November 1996.

        Borenstein, N. and N. Freed, "Multipurpose Internet Mail
        Extensions (MIME) Part Two: Media Types", RFC 2046, November
        1996.

        Borenstein, N. and N. Freed, "Multipurpose Internet Mail
        Extensions (MIME) Part Five: Conformance Criteria and Examples",
        RFC 2049, November 1996.

   [2] Crocker, D., "MIME Encapsulation of EDI Objects", RFC 1767,
        March 1995.

   [3] Resnick, P., "Internet Message Format", RFC 2822, April 2001.

   [4] Elkins, M., "MIME Security With Pretty Good Privacy (PGP)", RFC
        2015, October 1996.

        Callas, J., Donnerhacke, L., Finney, H. and R.Thayer "OpenPGP
        Message Format", RFC 2440, November 1998.

        Elkins, M., Del Torto, D., Levien, R. and T. Roessler "MIME
        Security with OpenPGP", RFC 3156, August 2001.

   [5] Fajman, R., "An Extensible Message Format for Message
        Disposition Notifications", RFC 2298, March 1998.

   [6] Galvin, J., Murphy, S., Crocker, S. and N. Freed, "Security
        Multiparts for MIME: Multipart/Signed and Multipart/Encrypted",
        RFC 1847, October 1995.

   [7] Klensin, J., "Simple Mail Transfer Protocol", RFC 2821, April
        1982.

   [8] Ramsdell, B., "S/MIME Version 3 Message Specification;
        Cryptographic Message Syntax", RFC 2633, June 1999.

        Housley, R., "Cryptographic Message Syntax", RFC 2630, June
        1999.

   [9] Vaudreuil, G., "The Multipart/Report Content Type for the
        Reporting of Mail System Administrative Messages", RFC 1892,
        January 1996.

Appendix IANA Registration Form

A.1 IANA registration of the signed-receipt-protocol content
    disposition parameter

        Parameter-name: signed-receipt-protocol
        Syntax: See section 5.2 of this document
        Specification: See section 5.2 of this document

A.2 IANA registration of the signed-receipt-micalg content
    disposition parameter

        Parameter-name: signed-receipt-micalg
        Syntax: See section 5.2 of this document
        Specification: See section 5.2 of this document

A.3 IANA registration of the Received-content-MIC MDN extension
    field name

        Extension field name: Received-content-MIC
        Syntax: See section 5.3.1 of this document
        Specification: See section 5.3.1 of this document

Authors' Addresses

    Terry Harding
    Cyclone Commerce
    8388 E. Hartford Drive
    Scottsdale, Arizona 85255, USA

    EMail: tharding@cyclonecommerce.com

    Chuck Shih
    Gartner Group
    251 River Oaks Parkway
    San Jose, CA 95134-1913 USA

    EMail: chuck.shih@gartner.com

    Rik Drummond
    Drummond Group
    P.O. Box 101567
    Ft. Worth, TX 76105 USA

    EMail: rik@drummondgroup.com

Full Copyright Statement

Acknowledgement