AS2 Interoperability Resolved Issues Report

Provides a list of issues resolved over the course of multiple AS2 Interoperability Tests through AS2-1Q14

June 20, 2014

Prepared & Facilitated by:
Drummond Group Inc.
www.drummondgroup.com

Drummond group

Table of Contents

Drummond group

Interoperability Issues

During the course of interoperability tests, interoperability issues were discovered or questioned and then resolved through the debugging stage of the test. All products from a given test comply with the corresponding resolved issues. These issues are listed below to assist in resolving any supply-chain trading problem which may occur between products-with-version from this test and AS2 products-with-version from outside the test, including backward versions of these test products.

Interoperability Issues Resolved or Affirmed AS2-1Q14

1. Participant was expecting Message-ID in MDN however it was not present. Message-ID in MDN is not required, only the Original-Message-ID is required. Participant changed code to allow processing of the AS2 message when Message-ID not present.
2. Participant product was unable to send messages when there was space in AS2 Identifier. Spaces in AS2 Identifier are permissible as long as the identifier value is enclosed in quotes. Product was modified to allow handling of spaces in AS2 Identifier
3. Participant product was not returning 200 OK on AS2 messages with Async MDN requests. The 200 OK needs to go immediately as soon the AS2 message is received. The product was modified to make sure 200 OK is sent immediately on receiving AS2 messages with Async MDN requests.
4. Participant product was failing to process AS2 messages as Message Id received from sending participant was composed of more than 100 characters long (the receiving product had set its limit to 100 characters only). As per RFC 4130 (AS2 specification), the Message-Id can be as long as 998 with 255 max suggested. The receiving product was modified to allowing handling up to 255 characters for the Message ID.
5. Participant product received HEAD requests at asynchronous MDN URL, which was not expected, and returned a non-zero content-length. This caused recipient to post only part of the MDN receipt, which could not be processed. HEAD requests are legal for the asynchronous MDN post. Participant product now responds with a 0 content length for HEAD requests to the Async MDN URL.
6. Participant product received a Restart request when product was still calculating how much had been received from the partner which would cause a local resource conflict and fail the test. Participant now returns a 500 error if product cannot process a Restart request because the resource associated with the ETag is locked.
7. One participant changed configuration of SSL cipher suites to allow SSL connectivity to two other participants.

Drummond group

Interoperability Issues Resolved or Affirmed AS2-3Q13

1. Participant was generating incorrect Disposition Notification when testing MDN Conformance scenarios. MDN Conformance tool generates invalid Encrypted, Signed and Compressed AS2 messages. Participant was generating the following Dispositions for the MDN Conformance Tests:

   Wrong was:

   MDN-K.1: Disposition:

   automatic-action/MDN-sent-automatically; error: authentication-failed

   MDN-K.2: Disposition:

   automatic-action/MDN-sent-automatically; error: decryption-failed

   MDN-K.3: Disposition:

   automatic-action/MDN-sent-automatically; error: decompression-failed

   Corrected to:

   MDN-K.1: Disposition:

   automatic-action/MDN-sent-automatically; processed/error: authentication-failed

   MDN-K.2: Disposition:

   automatic-action/MDN-sent-automatically; processed/error: decryption-failed

   MDN-K.3: Disposition:

   automatic-action/MDN-sent-automatically; processed/error: decompression-failed

2. Participant was waiting to send the HTTP 200 until the received message was completely processed (decrypt, check signature, decompress). This took almost 5 minutes for test AS2-J which requests an Async MDN. The 5 minute processing time caused a timeout for the HTTP connection of the sending partners. This was fixed by returning the HTTP 200 OK right after an AS2 message has been completely received but only if an Asynchronous MDN is requested. The unwrapping of the message is done after the HTTP 200 OK has been returned.
3. For FN-MDN testing, participant was getting a URL containing a query string in the “receipt-delivery-option” header which is allowed. Participant was unable to properly create the URL from that header value in order to return the MDN. Participant updated their third party libraries to resolve the issue.
4. Participant implementing SHA-2 support changed the SHA-1 value to sha-1. The value sha1 should also be supported on the receiving side for backwards compatibility.
5. For multiple attachments, participant was not sending the “type” parameter which must be used as part of the out boundary content-type. Please refer to the Multiple-Attachment RFC for this requirement. Participant added the “type” parameter.
6. Participant interpreted requirement of RFC 2822 for Message ID to mean that an @ sign is required between left and right parts. At least one participant does not use the

@ symbol to separate the right from the left which is OK. Participant relaxed check to not require @ symbol for Message IDs.

7. Participant was sending folded headers. Based on an earlier consensus, MIME folded headers are not allowed in AS2 Interop. The participant changed their code in order that AS2 message includes only unfolded header. Folded headers are not allowed in AS2 Interop by consensus.
8. Sending side was not sending a Content-Transfer-Encoding header for the signed part of the message. On the receiving side, javax.mail.internet.MimeMessage by default adds CTE header when the saveChanges() method is called. Due to this the receiving side MIC calculation did not matching with the sending side's MIC calculation.
9. The Message ID for Filename Preservation (FN) messages contained characters that are illegal for file naming but are legal Message ID characters. The Receiving side changed FN parsing code to filter out illegal file name characters when Message IDs are used to name files.

Interoperability Issues Resolved or Affirmed AS2-1Q13

1. The latest version of the sending participants Crypto API uses RFC 5751 which has micalg=sha-1 by default when signing data. The receiving participant required the older RFC 3851 micalg=sha1. The sending participant modified their code to accommodate the receiving participant as described in the following note. All other participants accepted either sha1 or sha-1. Note from Crypto API vendor: between RFC 3851 and RFC 5751 the values used in the micalg parameter for signed messages changed. We will accept both, but the default is now to use RFC 5751. In the event you are dealing with an RFC 3851 based system you will also need to use a constructor that sets the micalgs table and call it with RFC3851_MICALGS.
2. During AS2 Restart testing, after the entire message that requested a synchronous MDN had been sent(after the InSitu network error simulations had completed), participant was not properly detecting the MDN content in the received synchronous response and sent the entire POST again causing the receiving participant to report a duplicate document error. The sending participant corrected logic so that synchronous responses are now properly detected.
3. During FN-MDN tests, sending participant when sending an unsigned and unencrypted messages, and when the Filename Preservation MDN Responses option was set to "Reject Payload, Return Error", the receiving participant AS2 server would hang for 2-3 minutes before replying with the synchronous response. AS2 server was attempting to read and flush the stream that had already been read causing it to wait until the internal timeout period was reached before returning. The receiving participant removed the attempt to read and flush the already read stream.
4. Participant was treating the header Content-Type as case-sensitive which caused Interop issues. The participant changed code to look for the header with case-insensitivity to resolve the issue.
5. Chunked test cases were failing as participant was sending both Transfer-Encoding and Content-Length headers. (The HTTP Spec 4.4 does not mandate it "If a message is received with both a Transfer-Encoding header field and a Content-Length header field,

the latter MUST be ignored", however not all participants ignored the Content-Length header). Sending Participant changed Chunked messages so as to have these two headers mutually exclusive and not send the Content-Length when sending Chunked Transfer Encoded messages.

6. There was trailing whitespace following the semicolon in the chunk size, which is allowed to separate the chunk size from any chunked extensions. Participant modified code to ignore white space and other participant elected not to include it.
7. Participant acting as the client was sending a generic/basic HTTP Authorization header in its AS2 message. Normally, if a receiving server does not require authentication, it disregards this optional header as not applicable. However, the receiving participant (server) was processing this header and attempting to authenticate the HTTP Authorization header in their HTTP server and responding with an "HTTP 401: Authentication invalid". The participant acting as the client configured the receiving trading partner as not requiring authentication and thus did not send the generic HTTP Authorization header. No other participants were sending an Authorization header.

   Additional commentary:

   Typically, Authorization header is sent in response to 401 from the Server. However, in this case client was sending the Authorization regardless if the server is requesting client-side authentication. The client expected the server to ignore it, and not go into an authorization mode or to be configured to ignore all authorization headers.

   http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html

   ### 14.8 Authorization

   ```
   A user agent that wishes to authenticate itself with a server--
   usually, but not necessarily, after receiving a 401 response—does
   so by including an Authorization request-header field with the
   request.  The Authorization field value consists of credentials
   containing the authentication information of the user agent for
   the realm of the resource being requested.
       Authorization  = "Authorization" ":" credentials
   ```

   Also see:
   http://tools.ietf.org/rfc/rfc2617.txt
   HTTP Authentication: Basic and Digest Access Authentication

   The receiving server in question reported having received **Authorization null:***** and did not recognize it as generic/basic authorization request header and thus did not ignore it.

Interoperability Issues Resolved or Affirmed AS2-3Q12

1. Participant was sending SHA-1 instead of SHA1. Per consensus, it was agreed that for backwards compatibility products send SHA1 (without hyphen) but can receive and process SHA-1 or SHA1.

2. MIC verification mismatches were occurring in the case of where the messages were not signed, but the message was requesting signed MDN with algorithm different than SHA1. Participant, changed code to use the algorithm specified in the Disposition-Notification-Options header to calculate and verify the MIC value.
3. Participant encountered connectivity problems during the SSL handshake, which was caused by the usage of elliptic curve algorithms. It was resolved by disabling the elliptic curve algorithms in Java 1.7 by setting the property com.sun.net.ssl.enableECC=false
4. Participant was tagging all tests as Transfer Encoded: Chunked. Participant modified their code to include a valid content length so as to disable the default of Transfer Encoded:Chunked being set within the new HTTPClient 4 API.
5. Participant experienced SSL connectivity issues for SSL test cases tests (C,G,H and I) with two other participants. The exception being returned was: SslPeerAuthenticationException. To resolve these exceptions from being thrown JVM parameter –Dcom.sun.net.ssl.enableECC=false was implemented when using JDK1.7.0_07. This parameter disables the use of elliptical curve algorithms.
6. Participant was getting MIC value mismatches when receiving MDNs from a participant. To resolve this participant receiving the MDN's added parsing of the Received-Content-MIC string into separate tokens (i.e., MIC value and hash algorithm) instead of comparing the entire string as one entity. The receiving participant was expecting a space after the comma that separates the MIC value from the hash algorithm (and while all other participants do except for one) it is not required.
7. When processing the received MDN, the SHA-512 hash values exceed the 76 character threshold that causes a header value to wrap across lines causing a NoSuchElementException. The logic that unwraps wrapped lines failed because a space was expected between the tokens separating the MIC value and the hash algorithm in the Received-Content-MIC string of the MDN. Participant added logic to no longer require that space after the comma.
8. During AS2 Restart testing, a failed response to the HEAD request was causing a restart at byte 0. To resolve this, if the previous HEAD requests (during the same message transfer) had succeeded then the participant now assumes that the other participant supports HEAD requests. For this particular HEAD request the failure was due to a network issue and so now participant retries the HEAD request instead of just giving up and restarting back at the beginning.
9. Participant was receiving an MDN where the other participant used MD5 for MIC (Message Integrity Check) but receiving participant was expecting SHA1. The sending participant reconfigured to use the trading partner relationship to use SHA1 and then the test cases passed.
10. Participant was not saving the AS2 Message file during AS2 Restart when the inbound message did not contain a Content-Length or Content-Range header. This only occurred during the initial send of a message with Content-Transfer-Encoding: chunked. Participant made code changes to properly handle this case.
11. When parsing the Disposition-Notification-Options header for the `signed-receipt-micalg=optional,<micalg1>,<micalg2>,...;` participant was not using the first

micalg in the list. Participant made code changes to properly parse the micalg list to use the first algorithm in the list.

12. Participant was throwing an SSL exception error if the client connecting was using a Diffie Hellman key exchange. The sending partner restricted available ciphers to non Diffie Hellman ciphers. This was accomplished by disabling the use of Elliptic Curve ciphers.
13. The sending participant's encrypted-only messages were failing on the receiving side due to an RSA algorithm lookup failure by the receiving participant. The sending participant's encrypted messages had the following message header (sample):

    ```
    "Content-Type:multipart/signed; protocol="application/pkcs7-
    signature"; micalg=sha-1; boundary="----
    =_Part_80_524427828.1347454680685""
    ```

    Note that the value of "micalg" was "sha-1" and not "sha1". The receiving participant decryption routines failed due to "sha-1" algorithm not found. The Sending participant changed their header processing code to include the accepted "sha1" algorithm.
14. Participant was experiencing random failures while attempting to send AS2 messages or MDNs to receiving participant (HTTP/HTTPS POST failure). This was caused by receiving participant returning code of HTTP 102 (server busy) error. Although part of the AS2 Reliability specification, this is a feature that is not implemented by non-AS2 Reliability or all AS2 Reliability participants. HTTP 102 (to indicate server busy) was removed as a return code during processing of the incoming AS2 message.
15. Participant was using BER encoding in ASN.1 however receiving participant did not support BER encoding. Participant added support for BER.
16. Receiving Participant does not negotiate SSL version later than SSL 3.1/TLS1.0. Sending Participant made code change to send the SSL version that is supported by the other trading partners when the other participant does not negotiate SSL versions.
17. For SHA-2 test cases, participant resolved signature verification issues by correctly writing the SHA-384 and SHA-512 hash algorithm in the signature.
18. For CEM testing, participant's CEM requests contained extra characters right after the XML content. Participant changed code so as to not send the extra characters in the CEM XML Request messages.

Interoperability Issues Resolved or Affirmed AS2-1Q12

1. The sending participant detected a MIC mismatch on the returned MDN. The sending participant was using an earlier version of the MAIL API - version 1.4.1. Sending participant upgraded to version 1.4.5 (latest version) and it resolved this issue. In another scenario, a different receiving participant upgraded to Java Mail API from 1.4.1 to 1.4.5 to resolve a similar MIC mismatch with this sending participant.
2. A participant was sending MIME folded headers. Based on an earlier consensus, MIME folded headers are not allowed in AS2 Interop. The participant changed their code in order that AS2 message includes only unfolded header.

3. Participant was generating an encrypted AS2 message which another participant could not decrypt. The encrypted AS2 message was using the Bouncy Castle SmimeEnvelopedGenerator API. Participant changed code to use CMSEnvelopedDataGenerator API instead and this issue was resolved.
4. Participant was failing to verify a signed AS2 message. If the Content-Transfer-Encoding doesn't exist in the signed part, it is considered to be binary by default in AS2, while the default value is 7bits in HTTP. Bouncy Castle API was using 7bits Content-Transfer-Encoding by default and failed to verify. Participant explicitly indicated the Content-Transfer-Encoding as binary when verifying the AS2 message using Bouncy Castle API and this issue was resolved.
5. Participant was expecting the value of **Final-Recipient** header in MDN to match value of **AS2-To** header in original AS2 message. However, the value of Final-Recipient header in MDN doesn't have to match value of AS2-To header in original AS2 message. Participant changed code to not fail test case if the two values don't match.
6. Participant was not returning the Synchronous HTTP 200 response back before sending Asynchronous MDN. Synchronous HTTP 200 responses must be sent back before Asynchronous MDN are sent. Participant changed code to send Synchronous HTTP 200 response back before sending Asynchronous MDN.
7. Participant was formatting the **Disposition** field value in the MDN as **automatic-action/MDN-sent-automatically ; processed** which one participant was unable to parse. Participant generating the MDN changed their code to remove the space after "automatically" and receiving participant allowed the extra space after "automatically" for added flexibility.
8. Participant rejected another Participant's certificate containing an IA5String in the OrganizationalUnit name. IA5String is not allowed as a NAME in the subject name unless it is an email address. New digital certificates were generated.
9. Participant was configured to use SSL 3.2. Another participant's product supports SSL 3.1 but not SSL 3.2. Per the AS2 test plan SSL 3.1 meets the minimum requirements for SSL. The participant using SSL 3.2 changed to use SSL 3.1 at the global level.
10. At least one participant's product does not support TLS 1.1 / SSL 3.2 which is now the default trading partner SSL configuration in another product. This caused SSL handshake errors. The trading partner configuration setting was changed to use "TLS 1.0 / SSL 3.1"instead SSL 3.2 for this specific participant. The expected behavior is that during SSL negotiation, when the server does not support the protocol version, it should be able to negotiate down to a version that it supports and not cause a handshake error, however negotiation was not supported.
11. Participant was using AS2 Async MDN URL over 100 characters. One other product did not support Async MDN URL's over 100 characters. Participant changed their Async MDN URL to less than 100 characters.
12. For AS2 messages that are not signed, but the MDN is requested to be signed using SHA-2, the default SHA-1 algorithm was being used for the MIC calculation/verification. This was resolved by using the algorithm from the Disposition-Notification-Options header versus defaulting to SHA-1 for MIC calculation/verification.

13. Participant was returning the wrong MIC value in the MDN for test cases G and J. Participant resolved by making code changes to better deal with processing (unpackaging) signed/compressed messages which are base64 encoded. The processing issue did not show itself when messages were NOT base64 encoded.
14. Participant was sending messages with the AS2 Restart ETag header. Other participant was unable to parse the ETag header used for AS2 Restart. Participant made a code change to allow processing of the ETag header.

Interoperability Issues Resolved or Affirmed AS2-3Q11

1. When sending a non-signed, non-compressed and non-encrypted message participant was delivering a malformed Content_Type (/application/application/EDIFACT) vs (/application /EDIFACT). Corrected by removing the extra "/application" string.
2. SSL connectivity could not be established between two participants due to one side not listing IAIK and BouncyCastle as JCE Providers.
3. After Receiving participant returned synchronous MDN over an SSL connection, the connection was not closed down properly. The Sending participant would eventually consider the returned MDN transfer to be incomplete and failed the test case.

   The Sending participant was not returning a close_notify alert which the Receiving participant was waiting on before it replied with a TCP FIN. However, the Sending participant was waiting on the TCP FIN before sending its close_notify. This created a deadlock which kept the connection from closing.

   To resolve this issue, Receiving participant added a timeout when waiting for the close_notify alert from Sending participant. After the timeout, the Sending participant now sends a TCP FIN after a 3 minutes time out. Thus allowing the Sender to properly shutdown the connection and receive the MDN.

   The Sending participant did not experience this issue with any other participant, nor did Receiving participant experience this issue with any other participant. This issue was unique between these two participants and only for the Synchronous MDN over SSL test case.
4. Sending Participant was expecting Message-ID in returned MDN, however, none was present. However per RFC 4130, Message-ID in MDN is not required, only the Original-Message-ID is required. Participant changed code to not fail AS2 messages where Message-ID is not present in returned MDN.
5. Participant was sending folded headers however due to prior Consensus Item, folded headers are not allowed in AS2 Interop's. Participant was using BouncyCastle which folded headers so headers were unfolded prior to sending.
6. Participant modified configuration between SSL Client and SSL Server to allow SSL Client to support SSLv2Hello with SSLv3. This resolved SSL connectivity between Participant and 3 other participants.
7. Participant was sending HTTP response code OK in quotes, the two CRLF's then another 200 "OK" as follows:

```
HTTP/1.1 200 “OK”

200 “OK”
```

This caused connectivity issue with one other participant. Offender changed code to send as follows without quotes:

```
HTTP/1.1 200 OK
```

8. Participant was replying with Asynchronous MDN with unquoted AS2 ID’s values in the Original-Recipient and Final-Recipient. The AS2 ID values needed to be enveloped with double-quotes (“) as the AS2 ID’s contained spaces.
9. Participant was incorrectly specifying Content-Transfer-Encoded=quoted-printable. This caused the payload to be formatted with CRLF at the end of each logical line, causing payload mismatches between Sender and Receiver. Participant changed to Content-Transfer-Encoding to binary.
10. Participant, testing out of mainland China, was experiencing connectivity issues when connecting with participant using HTTP port 4444. It was determined that most Internet Service Providers in China shut off access to port 4444 due to virus, trojan horses, worms, etc. exploiting this port. Participant changed to a port other than 4444.
11. Participant added quotes to AS2-To and AS2-From values as some AS2 Id’s had spaces in them.
12. Participant was not properly setting the content-type based on the payload type. Participant added payload type detection to properly set the content-type.
13. Receiving Participant was not closing the connection on the large payload test case (J) which requires the Recipient to reply with Async MDN. Participant changed code to close connection before processing AS2 Message. In addition, participant optimized code for better performance when receiving large payloads. Signature verification was optimized.
14. Participant was expecting HTTP response code of 204 for AS2 messages with Async MDN’s, however most participants return 200. Participant changed code to accept 2xx response codes.
15. Message-ID was not properly formatted according to unique@host format: See RFC 4130 Section 5.3.3. Message-Id and Original-Message-Id which states:

```
Message-Id and Original-Message-Id is formatted as defined in RFC
2822 [9]:

       "<" id-left "@" id-right ">"        (RFC 2822, 3.6.4)
```

Interoperability Issues Resolved or Affirmed AS2-1Q11

1. One participant was requiring the “Date” field to be present however the “Date” field is not required. The participant removed this requirement on it having to be required. RFC 4130 states: Other headers, especially "Subject" and "Date", SHOULD be supplied; Note that RFC states SHOULD versus MUST for the Date field.
2. During CEM testing, participant was using a Certificate which had not been accepted yet. The spec does states you must not use a cert that has not been accepted. This issue was resolved.

Interoperability Issues Resolved or Affirmed AS2-3Q10

1. One participant upgraded from IAIK-CMS 3.2 to IAIK-CMS 4.1 which caused the signature verification to fail with only one participant. IAIK resolved the issue and provided the following description of why the problem had occurred:

```
The reason for the signature verification failure is that the
DigestInfo Digest AlgorithmID encoding handling during signature
verification was changed to recognize the parameter encoding
used in the Digest AlgorithmID. For SHA-1 there are two ways to
encode the parameters: entirely omitting the parameter field
or encoding it as an ASN.1 NULL object. An application should
be able to handle both alternatives, depending on what is used
in the SignedData/SignerInfo to be verified.

For that reason, the same encoding is used in the digestAlgorithm ID
of the SignedData/SignerInfo digestAlgorithm field to be verified.
A raw signature engine is used and it is updated with the encoded
DigestInfo object.

However, the SignedData provided by the participant omitted the
parameters component in the digestAlgorithm(s) field, but included
it when encoding the digestAlgorithm of the DigestInfo when
calculating
the signature, which is not a customary methodology.
```

2. One participant experienced a handshake failure while trying to connect with another participant over their SSL port. While processing the SSL certificate, as a DER encoded object, DER decoding failed with:

```
    java.security.cert.CertificateException: Error parsing DER
data: Invalid printable string character `42'.
```

   The above error was caused by an asterisk (ANSI 42) in the "Issued to" field, for example: *.x.company.com. The asterisk in the domain name (Issued to) field indicates that the partner was accepting a potential security risk by allowing one SSL certificate to be used for any web site within the partner's domain. Only one participant was not able to process this SSL certificate and experienced this handshake failure. This participant, in discussion with their 3rd Party Toolkit vendor, agreed to modify their implementation to allow the asterisk in the Issued To field in a future Interop. For this

Interop, the SSL certificate was modified to remove the asterisk in the Issued To field and make it specific.

3. One participant was generating Async MDNs with a MIME boundary in the MDN which was preventing another participant from processing the MDN. There was an extra character before the boundary delimiter: …--MDNboundary as the boundary when it should be just –MDNboundary. For example:

```
…--MDNboundary
Content-Type: message/disposition-notification

 "other headers"

…--MDNboundary--
```

4. Participant had outgoing AS2 messages with AS2-Version=1.2, and the EDIINT-Features header=AS2-Reliability, however outgoing MDN's did not. Review and consensus agreed to the following:

> In section 3 of the EDIINT Features header draft, it says "The EDIINT Features header MUST be present in all messages transmitted by the user agent and not just messages which utilize the feature." The authors intent and interpretation of this is that an MDN is part of these "all messages". The Feature Header is an effort to move from the AS2-Version header means of communicating implemented functionality to a more flexible means. Thus, if you are already using AS2-Version in your MDN and supporting EDIINT Feature header, you should have EDIINT Feature header in your MDN.

5. An originating Participant was assigned an AS2 ID similar to: company [test]. A Recipient participant returned an incorrect AS2 ID: "company â test ã". The â is hex E2 and ã is hex E3. They should have been [(hex 5B) and] (hex 5D). The Recipient participant modified their ASCII Table to resolve the issue.
6. Originator Participant was not able to process a Recipients MDNs because the MDN boundary was incorrect. The boundary line –MDNboundary must be preceded by 0D0A (\r\n); those characters are part of the boundary. –MDNboundary before the message/disposition-notification was preceded by 0D85. Consequently, that boundary and the terminating boundary were not recognized, and the MDN was not valid. There were other instances of 0D85 in the text/plain part of the MDN. Those didn't matter, but they were probably intended to be 0D0A as well.
7. Participant was not including a "Content-Transfer-Encoding" header which is OK. Participant was referencing RFC2045 page 14 which states: "Content-Transfer-Encoding: 7BIT" is assumed if the Content-Transfer-Encoding header field is not present. However, 7bit encoding is default for AS1 (SMTP) but for AS2 (HTTP) it is binary if the content-transfer-encoding header is not present. From AS2 spec:

> 4.1.1 Content-Transfer-Encoding not used in HTTP transport Because HTTP, unlike SMTP, does not have an early history involving 7-bit restriction, there is no need to use the Content Transfer Encodings of MIME [1]. This difference is discussed in [11] section 19.4.4.

So the problem was that binary data was being sent by the Originator, while the Recipient incorrectly assumed 7bit by default according RFC2045.

Interoperability Issues Resolved or Affirmed AS2-1Q10

1. There was a Signature validation failure because the sending participant does not provide a content transfer encoding header for the signature block of a signed transaction. The receiving participant was using Bouncy Castle which defaults to base64 rather than binary. The receiving participant modified their client code to check for the header and when not present explicitly define it as binary to Bouncy Castle.

2. Participant enhanced their AS2 identifier parsing. The AS2 identifiers are varied and introduce spaces, and other legal characters as well as the use of the “rfc822” identifier all of which must be handled. Participant modified their AS2 identifier parser to handle all of these permutations.

3. There was an issue in receiving an Async MDN. Sending participant was receiving an Async MDN before receiving back the “200 ok” for the original AS2 message; the recipient participant modified their side and now closes the initial connection prior to returning the Async MDN.

4. MDN validation was failing because the original and final sender AS2 ID did not match; Participant had misinterpreted RFC-4130 section 5.3.2 as a “MUST” rather than a “SHOULD” constraint. Participant modified their code to only generate a warning message to the log file rather than a failed MDN.

5. Recipient was not seeing a “Disposition-Notification-To” header and so for example in test case A, was not returning a synchronous MDN. The reason was that the sender’s AS2-From value ends with a semi-colon and this same character is used by receiving side’s header parser to determine when a header line is continued to the next line. The sending side included the AS2-From as part of the subject line (and was not encapsulated in quotes). The receiving side, instead of interpreting the Subject line as an independent header of its own included the Disposition-Notification-To header line too. An example is shown below:

```
Message-ID: <1d655d88-1e3a-427a-ae9d-a00d199411b1@211.75.166.48>
Subject: participant_as_id;
Disposition-Notification-To: ignored@example.com
```

6. Receiving participant was using a version of Jetty web server which had a bug and did not allow Test Case I (SSL) to pass. Participant upgraded the Jetty web server to a later version (v6.1.23) which resolved the connectivity issue.
7. Participant had configured their SSL (HTTPS) port to 80 which caused two other participants to fail all SSL test cases. Participant changed port for HTTPS from 80 to 443 which resolved the connectivity issues
8. Participant could not process other participants SSL certificates due to their chain validation restrictions. However, when the SSL certificates were provided as a PKCS#7 file (.p7b), they were able to perform SSL tests succesfully.
9. Participant could not complete the SSL handshake with another participant. One participants cipher stack contained an entry for AES (specifically, "TLS_RSA_WITH_AES_128_CBC_SHA"), which caused issues with the other participants processor. Participant had to remove this cipher from their cipher stack so that they could proceed with the remaining SSL tests. This occurred after the initial connectivity and debug test rounds had completed successfully.

Interoperability Issues Resolved or Affirmed AS2-3Q09

1. MDN Receipts from some participant were not parsed successfully because the MDN Receipt was split into two packets along the CRLF CRLF split between headers and body as follows:

```
Packet 1: Headers, CR, LF, CR
Packet 2: LF Body
```

2. Several participants were sending TIFF files with content-types of application/octet-stream instead of image/tiff
3. One participant was checking the content-type of the MDN with case sensitivity, which lead to error-messages that no MDN has been received.
4. One participant was checking for the existence of a Message-ID in the MDN's which is optional with MDN's. The participant, acting as originator, was returning an ERROR 500 when a recipient posted an ASYNC MDN to them.
5. One participant was setting the content-length-header in the HTTP-response of an AS2-request with synchronous MDN's to two bytes too low leading to the effect that the trailing CR/NL after the last MIME-boundary was missing. All except one participant was ignoring this. Sending the correct length solved the problem.
6. Four products had problem processing AS2 message with an AS2 Identifier which had exclamation mark at end as assigned to one participant. For Example:

```
AS2 Identifier:  a b c !
```

7. Participant was starting to parse the HTTP-response the moment a CR was received and didn't wait until the NL was transferred but was interpreting it as part of the response-body. Together with the above explained effect that the trailing CRNL was

missing, the read of the body stopped with one '-' missing, leading to the (in that case correct) error-message that there was no valid MIME-boundary at the end.

8. Participant was failing FN test cases because it expected the filename in the Content-ID where it should have been using the attribute filename in Content-Disposition
9. One participant was sending messages and MDN's with extra CRFL at end. They were also sending messages and MDN's with Base64 encoding containing LF instead of CRFL. Participant resolved by removing the extraneous CRLF's and to their base64 encoding (CRLF at end of lines).
10. Sending participant was receiving "failure to receive HTTP response" when sending to one participant. The sending participant increased an internal connection limit and the tests ran successfully
11. Participant had a problem sending Test Case I test to a participant. The recipient was timing out in the DMZ waiting for their message. Receiving participant changed a timeout value internally from 5 minutes to 30 minutes and the test succeeded
12. Participant was sending messages with non-conformant date format in their headers which caused jetty exceptions on receiving AS2 server. Sender changed the format of their date to conform to RFC
13. Sending participant was failing all outbound AS2 messages despite Recipient getting message, processing and sending back good MDN's. Sender complained about Received-Content-MIC not having a space before the digest algorithm id. Sender agreed to relax check that sender should be flexible in allowing for no space there. Researching specifications did not indicate space was required.
14. Default basic authentication caused problems with one participant. When sending to that participant, the sending participant received an access denial, "HTTP/1.0 401 [ISS.0084.9001] Invalid credentials" WWW-Authenticate: Basic realm="xxx" Connection: Close, Authorization required message. The sender was doing a 'basic authentication' step just before sending the AS2 message; the sender changed their configuration to remove basic authentication for that one recipient.
15. Participant reported failures related to non-empty MIME epilogue in signed AS2 Tests involving signed payloads from originator. The problem was that the originator was generating a non-empty MIME epilogue (another CRLF after the final MIME separator and its CRLF) in the signed message. The resolution was for originator to change their code to stop generating the non-empty MIME epilogue (the HTTP RFC clearly disallows them).
16. This issue revolved around tests with encrypted payloads (with no compression or signing). The problem was that originator was base64-encoding the payload at the MIME layer before encrypting it and recipient was not removing this encoding. The resolution was for recipient to change their code to perform the proper decoding after decryption.
17. This issue revolved around LF versus CRLF in signed AS2. Tests involving signed payloads. The problem was that originator was generating lines ending only in LF in the base64-encoded signature. The resolution was for originator to change their code to generate CRLF for these lines instead of LF.

18. This issue revolved around no space after the key word “processed” as in ;processed, in the MDN disposition. The problem was that the MDN from the recipient contained no space between ";" and "processed" in the MDN disposition (causing originator to treat it as a negative MDN). The resolution was for originator to change their code to treat this MDN as a positive MDN by allowing a no spaced after the colon and before key word “processed”.
19. This issue revolved around a tab character included in the Disposition-Notification-Options header. The problem was that the Disposition-Notification-Options header sent by the originator contained a tab character after the semi-colon which caused the recipient to reject the AS2 message. The resolution was for recipient to change their code to accept tabs in this header.
20. A participant was rejecting a sender’s AS2 message with the error: " 500: HTTP/1.1 500 Cannot convert date: Tue Sep 29 10:18:18 EDT 2009". This date format is the Java default format, i.e., "Tue Sep 29 11:21:09 EDT 2009". The sender changed their date format to conform to RFC 822, for example to "Tue, 29 Sep 2009 11:21:09 EDT", which recipient then accept as valid.
21. This issue revolves around SSL connectivity between different SSL toolkits. While doing a Client Handshake whenever session-resume is requested by the Client and the version of Client Hello does not match the version of previous handshake, an SSLException is generated. This is happening with one participant’s SSL Client using Java 6 APIs with two other participants SSL Servers: one using IAIK and the other Entrust Toolkit for Java (which currently only supports JDK 1.4 and 1.5)

    The issue was resolved by the client participant using the Java 6 APIs setting explicitly the session timeout and session cache size. This solved the failed test cases with JDK 1.6 being in use, so there is was no need for the client participant to be restricted to using JDK 1.5. The setting of above settings is described in http://java.sun.com/javase/6/docs/api/javax/net/ssl/SSLSessionContext.html

    The timeout was set to 1 second and cache size was set to 1 to make sure that there is no session-caching happening. This is not the final solution as the JDK 1.6 participant is working with their toolkit vendors to resolve the interoperability issue with the other toolkits. In summary, the Sun Java API should not send TLSv1.2-messages and mark them as TLSv1.0.

Interoperability Issues Resolved or Affirmed AS2-1Q09

1. Participant was not accepting uppercase ‘SHA1’ for specifying the digest algorithm in the AS2-received-content-MIC-field and in the signed-receipt-micalg as they believed that it must be lowercase ‘sha1”. This value is case insensitive and participant changed code to allow any case.
2. Participant was specifying micalg=SHA instead of micalg=SHA1 in the content-type header of signed messages. Participant resolved by modifying their security package.
3. Participant expected Message-ID in MDN’s however they are optional and not required. Participant modified code so as not to require Message-ID’s.

4. Payloads with indefinite lengths should padded at the end with EOC (x'00'), and nothing else. One participant was padding with zeroes (x'00') but was followed by a linefeed character. This caused the test case to fail. Participant modified code to remove the final linefeed character (x'0A').
5. Participant was incorrectly using the From field of the AS2 MDN instead of the AS2-From as the value of the final-recipient.
6. Participant was sending an AS2 message with bad date format and one recipient responded with a 500 error response code with the error "Cannot convert date: Tue,07Apr200912:35:27GMT." and then shutdown the connection. Participant resolved by adding spaces to the date value where appropriate.
7. One Participant resolved security retrieval information when # character assigned to Participants AS2 Identifier.
8. Participant was not including Disposition Notification field as required by the test case and thus the test case failed because MDN was never returned.
9. Sending "expect : 100 continue" caused issued with other participant http server. Participant disabled sending "expect: 100 continue"
10. Participant was not adding quotes to the AS2-From, AS2-To MDN field values as expected when these values contained spaces.
11. Participant removed linefeed at end of encrypted data which caused other participant parser to fail, and at the end of mulitpart (e.g. mdn) data which caused other partner to fail when verifying signature.
12. There was an incompatibility between curl (as a client) and JSSE (as a server) in https. This causes failure because of TLS Extension "Ticket Session" sent by curl and not understood (TLS 1.1) or rejected silently (TLS 1.0) by JSSE. An upgrade to both sides was the solution (curl 7.9.14/openssl 0.9.8j and Java 1.6.0u12).

    RFC 2246 that TLS 1.0 with extensions should be supported:

    "Forward compatibility note: In the interests of forward compatibility, it is permitted for a client hello message to include extra data after the compression methods. This data must be included in the handshake hashes, but must otherwise be ignored. This is the only handshake message for which this is legal; for all other messages, the amount of data in the message must match the description of the message precisely."
13. One participant was using a key size of RSA 4096 bits in their public key but other participant does not support that key size. Per the AS2 test plan, "Testing will assume 128 bit and 1024/2048 bit keys and 24 byte 3DES unless consensus for larger keys is reached with all participants". Participant changed change certificate to agreed key size.
14. Participant was returned "Signature verification fails when verifying signature on compressed payload" for test case G and J. Participant modified default content-transfer-encoding on the body part when there is none specified. Participant was using Bouncy Castle as their S/MIME library. If content-transfer-encoding is not specified on the body part, Bouncy Castle uses 7-bit encoding as a default value. Participant modified code to set "binary" as the default content-transfer-encoding. NOTE: This

issue also occurred as issue No.3 as reported for AS2-1Q08" in this “AS2 Interoperability Issues Report".

15. Participant was not providing the Content-Length header in the MDN returned. This caused the message digest to be calculated incorrectly and thus fail. Participant modified code to add Content-Length.
16. Participant was not returning the proper Disposition for MDN Conformance test case K.3. Participant could getting a processing error:

    2009-04-17 08:32:46,137 Process[PID: 646, Desc.: No description] ERROR [..] indefinite length primitive encoding encountered

    Participants newer version of their S/MIME toolkit was causing the issue as newer version threw different exception: (IllegalStateException). Participant changed code to catch new exception code. Proper MDN response should have been:

    Disposition: automatic-action/MDN-sent-automatically; processed/error: decompression-failed

17. First participant sent a Bravo CEM request to a second participant and their Bravo cert was in a “pending” state. Second participant then sent a Bravo CEM request to first participant before accepting their Bravo cert request. First participant immediately accepted Bravo cert (and as per the spec, as soon as a partner accepts the new signing cert, they must be prepared to accept messages signed with the new certificate). Second participant then sent a response to their Bravo CEM request signed with new Bravo cert (since first participant had accepted it) but the MDN returned from that request contained an authentication error since first participant had not yet updated the second participants profile with the new Bravo certificate.

Drummond group

Interoperability Issues Resolved or Affirmed AS2-3Q08

1. Participant was not providing filename in the Content-Disposition for Filename Preservation test cases.
2. Participant was sending MDN's with folded headers and reconfigured to not send folded headers in MDN's.
3. Participant was calculating incorrect MIC digest value on test cases F, H, I, and FNP-F. The sender was not including "multipart" in the Content-Type.
4. Participant was assigned AS2 Identifier with angled brackets (< >) as part of the identifier however one participant did not support processing angled brackets; angled brackets are allowed and participant added support for allowing angled brackets.
5. Participant was not calculating correctly the MIC on test cases with Multiple Attachments for test cases MA-D, MA-E.
6. Participant was unable to properly decompress a compressed message.
7. Participant was failing test cases because the data is using indefinite length encoding. When indefinite length encoding is used, the data was getting terminated by end of content bytes, which is 00 00. In the data, these bytes are found before actual end of data; hence the parser was not reading the data beyond the end of content bytes. (Test Cases: D, G, J, VFNP-D, VFNP-G and VFNP-J)
8. For MA test cases, participant was missing the required "Type" parm in the multi-part/related content-type header. Content-Type: multipart/related however it is required (RFC 2387). Also, participant content-type value of the first attachment which was an XML file had a content-type of multipart-related/plain it should have been application/xml. Finally, the second attachment did not have a Content-Type header.
9. Participant was sending `Content-Disposition` headers with additional double-quotes around the filename attribute, for FNP MA test cases, for example:

```
Content-Disposition: attachment; name=""T234-ma_test_data_1.xml"";
filename=""T234-ma_test_data_1.xml""
```

10. Participant was not including Content-Type header for the second attachment of the MA test cases. Recipient was requiring a Content-Type and modified code to default to "text/plain" per the specifications; however attachment was actually a PDF file. Sending participant modified code to add Content-Type header. Below is example of missing Content-Type;

```
--boundarySA==
   Content-Id: <mime-part-1>
   Content-Disposition: attachment; name="z_ma_test_data_2.pdf";
   filename="z_ma_test_data_2.pdf"
```

Interoperability Issues Resolved or Affirmed AS2-1Q08

1. Participant was receiving 400 bad requests when sending Async MDN's to two participants. The problem was that the initiating participant was sending the "length" twice i.e., in the transport header and also in MDN headers. Hence the receiving partner replied with a HTTP response code 400 - bad request".
2. One participant was sending folded headers in MDN's. Based on earlier consensus items, folded headers are not allowed because some Microsoft IIS does not process folded headers. Also, the Date header was incorrectly inserted before the protocol and boundary parameters.

   Wrong:

   ```
   Content-Type: multipart/signed; micalg=sha1;
   Date: Wed, 02 Apr 2008 12:55:01 UTC
         protocol="application/pkcs7-signature";
         boundary="----=_Part_361_1392923398.1207140899485"
   ```

   Correct:

   ```
   Content-Type: multipart/signed; micalg=sha1; protocol="application/pkcs7-signature"; boundary="----=_Part_361_1392923398.1207140899485"
   ```

3. Bouncy Castle 1.38 assumes that if Content Transfer Encoding is not specified in the body part that the encoding is 7-bit instead of binary.

   Participant was getting:

   "org.bouncycastle.cms.CMSException: invalid signature format in message: content hash found in signed attributes different"

   The Recipient participant modified the default to be "binary", for example:

   ```
   BodyPart bodyPart = mime.getBodyPart(0);
   if (bodyPart.getHeader("content-transfer-encoding") == null) {
       signedParser = new SMIMESignedParser(receivedMsg, "binary");
   } else {
       signedParser = new SMIMESignedParser(receivedMsg);
   }
   ....
   ```

4. Participant was receiving Content-Length with leading zeros but the participant did not have support for leading zeros.

   Example:

   ```
   Content-Length: 000000000000977
   ```

   RFC2616 says the content length is formatted as 1*DIGIT. This means that at least 1 digit must be present and it can repeat. The value of DIGIT is 0-9. Therefore the following values 000000000000977 and 977 would satisfy the rule and should be considered equivalent.

   See: 14.13 Content-Length

   http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.13

   See: BNF notation

   http://www.apps.ietf.org/rfc/rfc2234.html

5. One Participant's outer Content-Type header was missing the micalg (and protocol) element. Receiving participant was reporting and error: "Unknown MIC algorithm: null".
6. One participant during persistent HTTP connections was not resetting the Async MDN flag, hence they were sending HTTP 200 response with html message body (in addition to regular MDN) for test cases requesting Sync MDN after Async MDN test case was executed.
7. One participant assumed that the Message-ID in Async MDN's was required however it is optional.
8. Participant was acknowledging receiving AS2 message with a 200 OK, but not closing the connection until after the Async MDN Receipt was returned.

Interoperability Issues Resolved or Affirmed AS2-3Q07

1. One participant was using the character "#" in PrintableString field in their certificate but it is not allowed.
2. One participant's certificate was using Extended KeyUsage extension marked as critical but this field is not supported as critical.
3. One participants SSL certificate was invalid due to invalid encoding of AuthorityKeyIdentifierExtension.
4. One participant was calculating incorrect MIC on base64 encoded MIME part of AS2 message
5. One participant enabled base64 encoding at transport level but some participants failed to process AS2 message. Earlier consensus determined that base64 encoding at transport level is not allowed.
6. One participant was not adding quotes to MDN AS2-To field value in as required for AS2 identifiers which include spaces.
7. One participant reported incorrect Disposition for K.1 Test Case. Reported processed" only, no reason given and evaluated positive when it should have been evaluated negative.
8. One participant reported incorrect Disposition value of for K.2 Test Case: automatic-action/MDN-sent-automatically; processed/error: decryption-failed;/processed/error: decryption-failed;
9. One participant reported incorrect Disposition on K.3 Test Case. Reported as "Processed", should have reported: Processed/Error: decompression-failed
10. HTTPs SSL handshake issue resolved by increasing response timeout.
11. One participant introduced new SSL layer – issues reported and resolved.
12. One participant could not support MA payloads if one of the attachments was not XML or EDI. MA payloads contained were PDF and TIFF.
13. One participant was calculating incorrect MIC for MA AS2 message.
14. One participant was not terminating properly the MIME inner boundary on MA AS2 messages.
15. One participant was enveloping valid MIME content with an extraneous boundary marker prior to encryption, for encrypted-only MA AS2 messages.

Interoperability Issues Resolved or Affirmed AS2-1Q07

1. One participant issued certificates with a Country Code using three characters (USA) vs. two (US). Two characters are required, three characters caused Interop issues with some participants.
2. One participant issued certificates using IAString vs. the correct DirectoryString type for organizational unit name field which caused Interop issues.
3. AS2 Identifiers containing embedded spaces MUST be enclosed in double quotes. When the AS2 Identifier is not enclosed in double quotes, the agreed rule is to parse up to the first blank space, however this caused Interop issues. The solution was to enclose the AS2 Identifier with double quotes.
4. The ability to calculate a message integrity check (MIC) on the received message and return it to the sender of the message inside the signed receipt (MDN) is a basic requirement of AS2. The MIC should be calculated over the signed data. One participant was uncompressing the signed data first, then calculating the MIC, thus generating an incorrect value and causing Interop issues. The solution was to calculate the MIC over the signed payload.
5. Several participants were sending folded headers in the MDNs. However, an earlier Consensus item stated that headers should not be folded. The participants unfolded their headers to resolve Interop issues.
6. One participant was encoding their data as 8bit (used in SMTP), however, binary transfer encoding is the default over HTTP. The participant switched to binary encoding to resolve Interop issues as a result of using 8bit encoding.
7. The Message-ID in the MDN is not required. One participant was failing test cases because of the misunderstanding that the Message-ID was required in the MDN's. This continues to be a source for misinterpretation. The Original-Message-ID, however, is required in the MDNs.
8. AS2-From and AS2-To do not have to be UPPER CASE. One participant was failing test cases because they misunderstood that the AS2 portion must be UPPERCASE. MIME/HTTP headers are not case sensitive.

Interoperability Issues Resolved or Affirmed AS2-3Q06

1. Certificates and security toolkit related errors continued to be observed in this test round. Certificates using unusual fields or extensions could create problems within supply-chains. Not all possible certificate fields or extensions were tested against every AS2 product's toolkit, and potential issues could still exist due to certain certificate fields and extensions. Also, it is becoming apparent that not every version of security toolkits is interoperable with every other version.
2. MDN were rejected by at least one participant, based on misunderstanding that Message-ID header is not required. The Message-ID is optional, and the Original-Message-ID is required.
3. Certificates with serialNumber equal to zero could not be processed by at least one participant's security toolkit. Therefore, the consensus from previous Interop's was amended to further constrain serial number to be non-negative integers, greater than, but not including zero.
4. Folded-Headers once again caused Interop issues in this round. Folded-Headers, although allowed in the standards, cause Interop issues with some participants, and thus are not allowed. Several participants were using folded headers, and at least one participant could not accept MDN messages with folded headers (that is a CR LF in the string value of a header).
5. Participant was missing a CRLF for the MIME boundary, or not following CR with a LF, or adding an extra CR LF, causing receiving applications to fail the test. See RFC 2045-2049.
6. Participant was missing the report-type, as in: Content-Type: multipart/report; report-type=disposition-notification; for in the MDN header, however, it was required.
7. Although not required, the To: and From: headers, if used, should follow the MIME header formatting rules. At least one participant was not enclosing the values with quotes when they were required. Same issue appeared for AS2-To and AS2-From headers.
8. Base64 encoding the entire HTTP body was being used. Note however, that Content Transfer Encoding (CTE) of MIME body parts within the AS2 message is allowed. Consensus was arrived that if the MIME bodies were already encrypted and or compressed, CTE was neither necessary nor practical for performance reasons. Participants agreed to remove Base64 encoding over the entire HTTP body, which helped resolve Interop issues, and theoretically improved processing performance of messages. Performance metrics are not measured in Interop testing
9. It was agreed that HTTP/1.0 servers are required to close HTTP connections, and it is not the responsibility of HTTP clients. At least one participant was relying on a timeout for the connection to close on the server-side, or for the HTTP client to close the connection. When the HTTP /1.0 Server waited for the HTTP /1.1 Client to close the connection and the Client waited for the Server to close the connection but the Server did not close then the Client timed-out and perceived it as an aborted connection and

flagged the test failed. The HTTP 1.0 Servers must close the connection based on the HTTP/1.0 specification.

10. Certificates needed to have two-character country code. This was in the list "Interoperability Issues Resolved or Affirmed from previous Test Rounds", but it occurred in this round as well.
11. A participant had a certificate organizational unit name specified as "R&D" and it was encoded as an IA5String. This is supposed to be a DirectoryString. The participant discovered that their certificate generation tool used to create their certificates was using the outdated IA5String encoding for some of the elements within the Subject and or Issuer name fields.
12. At least one participant found an issue with LF of CRLF being removed on the outgoing payload data which was causing payload mismatches (the sent payload did not match the received).

Interoperability Issues Resolved or Affirmed AS2-1Q06

1. Certificates and security toolkit related errors continued to be observed in this test round. Certificates using unusual fields or extensions could create problems within supply-chains. Not all possible certificate fields or extensions were tested against every AS2 product's toolkit, and potential issues could still exist due to certain certificate fields and extensions.

   For example, there was an issue with a participant SSL certificate was not properly formatted (for DER encoding). It caused another participant to reject it during the SSL handshake. The participant's certificate had an explicit default value in the version identifier. The certificate had a 0 in this field, when it should be version 1 (everyone else's certificates had version 1).
2. Also, it was discovered that security toolkit versions are not always interoperable from version to version, and this Interop revealed and helped resolve these incompatibility in the security toolkits. However, security toolkits were not exhaustively tested for interoperability.

3. The AS2 specification requires that human-readable portion of MDN must contain "Final-Recipient". Please see http://www.ietf.org/rfc/rfc4130.txt section 7.4.2. A participant was not sending "Final-Recipient" however it was required.
4. A participant was sending "folded headers" in the MDN's and this caused an error because at least one other participant did not process "folded headers". In previous Interop's, it was a consensus item to not fold the headers.

   Example:

   Content-Type: multipart/report; report-type=disposition-notification; boundary="----=_Part_1139974138134"

5. An AS2 server was attempting to connect to port 80 instead of 443 when the URL was provided without a port, for example: https://hostname/ The correct port to connect to should be 443, (the default port for SSL when port is not specified). The default port for non-ssl is port 80. Please see: http://rfc.net/rfc2616.html
6. MDN conformance testing revealed that one participant's MDN disposition text says, "Disposition: automatic-action/MDN-sent-automatically; processed". It should have returned error text "processed/error: authentication-failed" or "processed/error: integrity-check-failed".

Interoperability Issues Resolved or Affirmed from previous Test Rounds

1. Some products could not accept certain characters or certain strings of AS2 identifiers. Two specific issues were: 1) having a space (" ") at the third location, e.g. "AS 2", and 2) identifiers containing a comma (","). While these conflicts were very rare and not associated with every participant, supply-chain implementers of these products should avoid identifiers with this syntax and discuss with their AS2 vendor any potential AS2 Identifier issues.
2. Trailing long white spaces (LWS) at the end of HTTP headers is not permitted. Leading LWS is allowed within HTTP (RFC2616) but not clear if trailing LWS is or is not.
3. The value "RSA-SHA1" was used by some participants for the MIC algorithm of the digital signature. It is a valid value and should be considered equal to that of the more common "SHA1" value. "RSA-SHA1" is a legacy value from an earlier S/MIME implementation.
4. Field names in MDNs, such as Original-Message-ID, are case-insensitive. According to RFC2298, section 3.1.1, "field names are case-insensitive, so the names of notification fields may be spelled in any combination of upper and lower case letters." As well, it is permissible to have a white space character (" ") before the message-id value of the Original-Message-ID field in the MDN. Thus, the two examples below are considered identical:
5. Original-Message-ID:<123foo@example>
6. original-message-id: <123foo@example>
7. The Message-ID header is not required in MDNs.
8. Chunked encoding for HTTP 1.1 requests and responses is acceptable for AS2. Rules for implementing, supporting and understanding chunked encoding can be found in the HTTP 1.1 standard, RFC2616.
9. Some products require valid EDI/XML documents on inbound messages and will generate MDNs with errors if they are invalid. This includes both valid formatting and/or recognized identifiers.
10. Certificate serial numbers must not be negative, per RFC3280. While some AS2 systems accept negative serial numbers, other systems cannot accept negative values.

11. Certificates are uniquely identified through their Issuer name and their serial number. As with negative serial numbers, certain AS2 systems will reject duplicate certificates, but others can accept them.
12. Some products utilizing the open source OpenSSL experienced problems in SSL transactions. The cause was due to the sending of empty fragments in the transaction which caused some trading partner products to corrupt the inbound document. The solution was to modify configuration flags within OpenSSL.
13. HTTP Content-length header is not necessarily required on MDN. The HTTP standard specifies the use and requirement of this header, and the AS2 draft is being updated to refer back to the HTTP standard for the use of content-length.
14. MIME Folded headers continue to cause problems with several products due to their associated web server. Folded headers were not used during the test and should be avoided in actual implementation.
15. The use of quotation marks on AS2 System Identifiers should not be used for atomic names. Also, the use of quotation marks on AS2 System Identifiers must be consistent for both the payload messages as well as for the MDNs. That is, if quotation marks are used in the payload message, they also must be present in MDNs.
16. A 204 (No content) HTTP response would be acceptable in an HTTP response of an async MDN request. This should be accepted (assuming the response has no body). From the latest version (13) of the AS2 draft, section 7.6, notice the comment of the response being "in the 200 range." HTTP RFC2616 states that if a 204 is returned, there is to be no message body and the message is terminated by the first empty line after the header fields. So, the 204 will work as long as there are only HTTP headers in the response.
17. If certificates use the country attribute, the country attribute may only contain two characters. For example, "C=USA" is invalid and instead should be listed as "C=US".
18. Encrypted messages can contain multiple RecipientInfo structures within the CMS data, including one describing the originator. Refer to RFC 2630 Section 6 for more details.
19. Consensus was reached that AS2 messages with EDI payloads should identify the content-type either as application/EDI-X12 or application/EDIFACT and NOT application/EDI-CONSENT.
20. The Message-ID is not required in Asynch MDN's because the AS2 standard states it SHOULD be contained, that is, it is not required. Asynch MDN's should not be rejected if MDN's do not contain Message-ID because it is not required. It is recommended that it be present. Please refer to the meanings of SHOULD and MUST.

About Drummond Group Inc.

Drummond Group Inc. (DGI) is the trusted interoperability test lab offering global testing services through the product life cycle. Auditing, QA, conformance testing, custom software test lab services, and consulting are offered in addition to interoperability testing. Founded in 1999, DGI has tested over a thousand international software products used in vertical industries such as automotive, consumer product goods, healthcare, energy, financial services, government, petroleum, pharmaceutical and retail. For more information, please visit www.drummondgroup.com or email: info2@drummondgroup.com