Ontario Gas Distribution Access Rule (GDAR)

Transport System Test, Point to Point Protocol

Draft
Version 0.2

Published by:

Ontario GDAR EBT Working Group

NOTICE OF DISCLAIMER

The information provided is directed solely to professionals who have the appropriate degree of experience to understand and interpret its contents in accordance with generally accepted engineering or other professional standards and applicable regulations. No recommendation as to products or vendors is made nor should be implied.

This document contains information contained in the OEB public protocol document entitled ‘Ontario GDAR EBT Transport Protocol Between Points’.

Table of Contents

Revision History

VERSION	DATE	AUTHOR	COMMENTS
0.1	June 1, 2006	S Atkins	Initial Draft
0.2	July 20, 2006	S. Atkins	Revisions from Protocol WG Meeting

1. Test Outline

This document will outline the procedure for conducting the GDAR Point to point protocol test suite. This document is paired with a spreadsheet (GDAR_P2P_TestMatrix_0.1.xls) containing the test scenarios and possible outcomes. The intention is that the spreadsheet should be used to perform the actual tests and tabulate results while this document is used as an ultimate reference for the testing procedure.

1.1 Audience

This document is intended for technical personnel already familiar with the GDAR Point to Point transport protocol who will be involved in the performance or evaluation of GDAR protocol testing. Persons responsible for performing these tests are assumed to be familiar with relevant aspects of public key cryptography, HTTP, SSL, X.509 Certificates, Internet communication, and any other technologies or specifications represented in the GDAR Point to Point protocol document.

2. Pre Conditions for Testing

2.1 Technical Checklist

The following test scenarios assume that the tester has the ability to do the following

- Make arbitrary HTTP requests with and without SSL
- Perform GDAR complaint Point-to-Point Upload commands
- Maintain a GDAR Point available over the internet
- Creation and manipulation of GDAR complaint PGP keys
- Ability to generate schematically valid GDAR Documents containing all transaction types and non-static document and transaction header information (Document reference number, transaction reference number)
- Creation and manipulation of SSL certificates

The point being tested is expected to have fully implemented the GDAR Point to Point protocol as defined by the OEB GDAR Working group.

Testing partners should agree before hand on what licence numbers will be used during testing to represent each end. Existing licences should be used when possible, new or potential vendors participating in GDAR should either use their licence number which is pending approval, or a mutually agreed upon string exactly 10 characters in length.

2.2 Pre-Testing Checklist

The following list should be fully checked by both parties involved in a round of testing prior to testing commencement

- Creation and exchange by each party of a GDAR compliant PGP key which is then signed and incorporated into each others key ring facility
- Creation and importing of a valid SSL certificate with correct parameters. The certificate used for testing may be 'self-signed' and if so, should be manually trusted by each party.
- Exchange of security credentials mentioned above with all testing partners.
- The point(s) being tested should be available over the internet for a reasonable timeframe to perform testing. Both parties should agree to the exact timeframe.
- A Set of schematically valid and invalid GDAR documents. A simple document generator mechanism is recommended for repeated tests amongst various testing partners.
- Ability to use common HTTP testing tools such as Curl, wget, wput, telnet, ping and any other related technologies which can deal with low level HTTP requests.

3. Test Scenarios

3.1 Scenario Format

The following test scenarios are elaborations upon the individual tests as contained in the spreadsheet which accompanied this document. Only tests labelled 'Protocol' should be performed between testing partners during official protocol testing. The remainder of the tests, labelled 'Diagnostic' should be used for internal and pre-market testing. Diagnostic tests can be performed between two testing partners if

The following test descriptions are given by titles in the following format.

<Test Number>, <Test Title>, <Matrix Column>

Diagnostic | ***Protocol****: Is this test used for partner-to-partner testing (Protocol), or is it meant to be internal to the organization*
Description: *A brief statement about the rationale for this test*
Procedure: *How the test should be performed. In all cases, the point being tested should not have a procedure extend beyond having implemented the full GDAR Point to Point specification*
Expected Pass Result: *The 'pass' condition. The number in brackets corresponds to the result ID in the test matrix.*

<Test Number>: *A unique ID identifying this test even if its position in the matrix is modified.*
<Test Title>*: Brief description of the specific test*
<Matrix Column>*: Location in the test matrix spreadsheet which accompanies this document.*

3.2 Point Connectivity/Secure Transmission

PC3, Connect to Point via https://<Pointname>/<point_uri>:443

Protocol
Description: Check for basic SSL connectivity with HTTPS client (browser).
Procedure: Using an SSL capable browser, navigate to the URL of the point
Expected Pass Result: Connection followed by Certificate challenge (1)

PC1, telnet to <Point Name> 443

Diagnostic
Description: Telnet to the port will test for basic connectivity.
Procedure: Telnet to the HTTPS URL (port 443) of the testing partner.
Expected Pass Result: Connection followed by server forced disconnect (1)

PC2, telnet to <Point Name> 80

Diagnostic
Description: Telnet to the HTTP port to test that unsecured HTTP access is disabled.
Procedure: Telnet to the HTTPS URL (port 80) of the testing partner.
Expected Pass Result: Unable to connect (2)

PC4, Connect to Point via https://<Pointname>.com/polint_uri:<portnum>

Diagnostic

Description: Check that only port 443 is available for SSL communication
Procedure: Using an SSL capable browser, navigate to the URL of the point on a port other than 443 or 80.
Expected Pass Result: Unable to Connect (2)

PC5, Connect to Point via http://<Pointname>.com

Diagnostic

Description: Check that HTTP on port 80 is not available for uploads
Procedure: Using an SSL capable browser, navigate to the URL of the point on port 80
Expected Pass Result: Unable to Connect (2)

3.3 Security and Security Protocol

SSP1, Client/Point have valid PGP Keys

Protocol

Description: Test for basic PGP abilities by verifying that the recipient point can decrypt data from the testing sender.

Procedure: Upload a file (of any kind) to the point using the agreed upon keys for testing via the GDAR point protocol. Confirm that the recipient point was able to decrypt the file.

Expected Pass Result: Confirm the decryption by challenging the recipient as to the contents of the file (1)

SSP2, Invalid Hash Algorithm

Diagnostic

Description: Test that only SHA-2 documents are accepted by the point

Procedure: Upload a file to the point encrypted with the proper PGP keys, but specify SHA-1 hash algorithm.

Expected Pass Result: Confirm that an FA Accept is not received by the testing point, and that the recipient point correctly rejected the incoming document. (25)

SSP3 Compression not for encryption

Diagnostic

Description: Test that uncompressed encrypted documents are not accepted by the point

Procedure: Upload an encrypted but uncompressed valid document to the point

Expected Pass Result: Confirm that an FA Accept is not received by the testing point, and that the recipient point correctly rejected the incoming document. (25)

3.4 Message Protocol

MP1, Send properly formatted HTTP request with document

Protocol

Description: Tests basic Upload-FA capability
Procedure: Upload a valid GDAR document containing 1 transaction to the point. Receive a corresponding FA Accept
Expected Pass Result: 200 OK, FA Accept (23)

MP2, Send properly formatted HTTP request with empty body

Diagnostic

Description: Tests that the point HTTP layer checks for non-null document payload
Procedure: Issue a GDAR Upload to the point but with an empty document payload
Expected Pass Result: HTTP 400 Bad Request (3)

MP3, Invalid request – invalid http version

Diagnostic

Description: Tests that only HTTP version 1.1 connections are accepted
Procedure: Upload a valid GDAR document with 1 transaction but issue an HTTP version other than 1.1
Expected Pass Result: HTTP 400 Bad Request (3)

MP4, Incorrect Date in HTTP header

Diagnostic

Description: Tests that HTTP date format checks are in place
Procedure: Upload a valid GDAR document with 1 transaction but issue an HTTP date that is inconsistent with the testing timeframe
Expected Pass Result: HTTP 400 Bad Request (3)

MP5, No entity body but include Content-Length

Diagnostic

Description: Tests that HTTP Content-Length mismatches are detected when there is no payload
Procedure: Issue a valid GDAR Upload request with a content length greater than the request size, and no document payload
Expected Pass Result: HTTP 400 (Bad Request) (3)

MP6 Entity body but no Content-Language

Diagnostic

Description: Tests that Content-Language headers in the protocol are respected
Procedure: Upload a valid GDAR document containing 1 transaction but exclude the Content-Language header
Expected Pass Result: HTTP 400 (Bad Request) (3)

MP7 Entity body but no Content-Length

Diagnostic

Description: Tests that the Content-Length header is respected when parsing HTTP body data
Procedure: Upload a valid GDAR document containing 1 transaction but exclude the Content-length header
Expected Pass Result: HTTP 400 (Bad Request) (3)

MP8 Entity body but no Content-Type

Diagnostic

Description: Tests that the Content-Type header is respected when parsing HTTP body data
Procedure: Upload a valid GDAR document containing 1 transaction but exclude the Content-Type header
Expected Pass Result: HTTP 400 (Bad Request) (3)

MP9 Entity body but Content-Length incorrect (1)

Diagnostic
Description: Tests that the Content-Length header is respected when parsing HTTP body data
Procedure: Upload a valid GDAR document containing 1 transaction but issue a Content-Length less than the actual
Expected Pass Result: HTTP 400 (Bad Request) (3)

MP10 Entity body but Content-Type incorrect (2)

Diagnostic
Description: Tests that the Content-Length header is respected when parsing HTTP body data
Procedure: Upload a valid GDAR document containing 1 transaction but issue a Content-Length greater than the actual
Expected Pass Result: HTTP 400 (Bad Request) (3)

MP11 Invalid request – invalid http

Diagnostic
Description: Tests that only the HTTP Post method
Procedure: Upload a valid GDAR document containing 1 transaction to the point but use the HTTP GET Method
Expected Pass Result: HTTP 501 Not Implemented (8)

MP12 Include Host HTTP line with incorrect host

Diagnostic
Description: Tests that the HTTP Host header is respected
Procedure: Upload a valid GDAR document containing 1 transaction but have the HTTP Host header be something other than the actual
Expected Pass Result: HTTP 400 Bad Request (3)

MP13 time out request

Diagnostic
Description: Tests that the point can detect and handle an idle connection
Procedure: Issue a valid GDAR document upload but pause the data stream and wait for timeout
Expected Pass Result: Connection followed by forced server close due to idle connection (1)

3.5 Certificate and Key Security

CKS6 Send valid document

Protocol

Description: Tests the basic upload and decryption functionality of the point
Procedure: Upload a valid GDAR document containing 1 transaction and encrypted according to the agreed upon parameters
Expected Pass Result: 200 OK, FA Accept

CKS7 Market Test Multiple IDs

Protocol

Description: Tests that PGP key name mapping to XML sender is checked
Procedure: Upload a valid GDAR document containing 1 transaction but modify the XML sender to be someone other than the actual sender.
Expected Pass Result: 200 OK, FA Reject

CKS1, Valid Certificate check

Diagnostic

Description: Tests that the server checks for a client certificate
Procedure: Upload a valid GDAR document containing 1 transaction but do not include a client certificate
Expected Pass Result: Connect Failure (2)

CKS2, Invalid Certificate - Bad Common Name (CN)

Diagnostic

Description: Tests that the server checks DNS entries for invalid common names
Procedure: Upload a valid GDAR document containing 1 transaction but issue the upload from a DNS entry other than the one in the CN field of the client certificate being used for testing
Expected Pass Result: Connect Failure (2)

CKS3, Unknown Certificate

Diagnostic

Description: Tests that only trusted client certificates are accepted
Procedure: Upload a valid GDAR document containing 1 transaction but issue a client certificate that is self signed and is not the one used for normal testing
Expected Pass Result: Connect Failure (2)

CKS4 Bad Decryption

Diagnostic

Description: Tests that badly decrypted documents are detected and handled correctly
Procedure: Upload a valid GDAR document containing 1 transaction but truncate the encrypted output at 80% such that decryption will not be possible
Expected Pass Result: 200 OK, offline acknowledgment (25)

CKS5 Fail Invalid Key

Diagnostic

Description: Tests that PGP keys other than the agreed upon participant keys are handled
Procedure: Upload a valid GDAR document containing 1 transaction but encrypt the document with a key other than the one exchanged prior to testing
Expected Pass Result: 200 OK, offline acknowledgment (25)

CKS8 Invalid Recipient

Diagnostic

Description: Tests that XML Recipient is checked to ensure it is the receiving point
Procedure: Upload a valid GDAR document containing 1 transaction but have the XML recipient be someone other than the actual recipient Point
Expected Pass Result: 200 OK, FA Reject

CKS9 Invalid Trading agreement

Diagnostic

Description: Tests that the recipient point is able to respect trading partner agreements
Procedure: Instruct the recipient point to invalid the testing trading partner agreement. Upload a valid GDAR document containing 1 transaction. Ensure that the FA Reject received has the correct reason.
Expected Pass Result: 200 OK, FA Reject

3.6 Upload Requests

UR2 Partial bad Document

Protocol

Description: Tests that a document containing some good and some bad transactions is handled correctly

Procedure: Upload a valid GDAR document containing 5 transactions but one of them is schematically invalid.

Expected Pass Result: HTTP 200 OK, FA Reject (17)

UR3 Multiple Transaction Document

Protocol

Description: Tests that a document containing multiple transactions of the same type is accepted

Procedure: Upload a valid GDAR document containing 10 transactions of the same kind.

Expected Pass Result: HTTP 200 OK, FA Accept (23)

UR4 Totally Bad Document

Protocol

Description: Tests that a document consisting completely of bad transactions is handled correctly

Procedure: Upload a GDAR document containing 1 transaction but have that transaction be schematically invalid.

Expected Pass Result: HTTP 200 OK, FA Reject (17)

UR5 Invalid XML

Protocol

Description: Tests that non-well formed XML is handled correctly

Procedure: Upload a GDAR document that is not well-formed (i.e. incorrect matching begin-end tags)

Expected Pass Result: HTTP 200 OK, offline acknowledgment (25)

UR6 Duplicate Document Reference Number

Protocol

Description: Tests that duplicate document reference numbers are rejected

Procedure: Upload a sequential series of documents all containing the same document reference number.

Expected Pass Result: HTTP 200 OK, FA Accept (exactly 1 total), FA Reject (Multiple) (23 and 17)

UR7 Duplicate Transaction Reference Number

Protocol

Description: Tests that duplicate transaction

Procedure: Upload a GDAR document that contains several transactions all with the same transaction reference number.

Expected Pass Result: HTTP 200 OK, FA Reject (17)

UR1 Unencrypted Upload

Diagnostic

Description: Tests that an octet-stream part is present

Procedure: Upload a valid GDAR document containing 1 transaction but do not encrypt the body. All other HTTP headers (including content-length) should be correct.

Expected Pass Result: HTTP 400 Bad Request (3)

3.7 Transaction Level

TL1, Multiple Transaction Types

Protocol

Description: Tests that heterogeneous GDAR documents are handled correctly
Procedure: Upload a valid GDAR document containing 3 transactions of 3 different types each (9 transactions total)
Expected Pass Result: HTTP 200 OK, FA Accept (23)

TL2, Multiple Transaction Types (2)

Protocol

Description: Tests that heterogeneous GDAR documents are handled correctly
Procedure: Upload a valid GDAR document containing 3 transactions of every type (108 transactions total)
Expected Pass Result: HTTP 200 OK, FA Accept (23)

TL3, Multiple Transaction Types (3)

Protocol

Description: Tests that heterogeneous GDAR documents are handled correctly
Procedure: Upload a valid GDAR document containing 1 valid and 1 invalid transaction of each kind (72 transactions total)
Expected Pass Result: HTTP 200 OK, FA Reject (23)

TL4, Document Header – bad Version

Protocol

Description: Tests that the GDAR document version is respected
Procedure: Upload a valid GDAR document containing 1 valid transaction but have the document header specify a schematically invalid document version (i.e, 10.1)
Expected Pass Result: HTTP 200 OK, FA Reject

TL5, Document Header – Invalid Instance

Protocol

Description: Tests that the GDAR document header schema is respected
Procedure: Upload a valid GDAR document containing 1 valid transaction, but have the document header specify a document reference number that is 40 characters long
Expected Pass Result: HTTP 200 OK, FA Reject

TL6, Document Directory – Invalid Instance

Protocol

Description: Tests that the GDAR Market Participant Directory schema is respected
Procedure: Upload a valid GDAR document containing 1 valid transactions, but have the MPD section contain 2 senders
Expected Pass Result: HTTP 200 OK, FA Reject

TL7, Schema Coverage (1)

Protocol

Description: Tests that the complete schema is implemented
Procedure: Upload a sequence of valid GDAR Documents each containing 1 valid transactions, each document containing a different transaction type. Ensure all transaction types are sent. Uploads sequence should include some concurrency (i.e., Uploads must not be exclusively sequential)
Expected Pass Result: HTTP 200 OK, FA Accept (Multiple)

TL8 Schema Coverage (2)

Protocol

Description: Tests that the complete schema is implemented

Procedure: Upload a sequence of valid GDAR Documents each containing 1 schematically invalid transaction, each document containing a different transaction type. Ensure all transaction types are sent. Uploads sequence should include some concurrency (i.e., Uploads must not be exclusively sequential)

Expected Pass Result: HTTP 200 OK, FA Reject (Multiple)

3.8 Polling Tests

PT1, Point checks 'facility' regularly and uploads pending FA Responses in a timely fashion

Protocol

Description: Tests that the recipient point is able to automatically return an FA for an Upload

Procedure: Upload a valid GDAR document containing 1 valid transaction. Ensure that an FA is pushed back to the recipient point within an agreed upon timeframe and within the scope of the GDAR Point to Point

Expected Pass Result: HTTP 200 OK (Multiple)

3.9 Load Tests

LT1, Maximum Document Size (1)

Diagnostic

Description: Tests that the recipient point can process a document near the maximum GDAR document size

Procedure: Upload a valid GDAR document containing multiple valid transactions nearly the maximum document size (but not over)

Expected Pass Result: HTTP 200 OK, FA Accept

LT2, Maximum Document Size (2)

Diagnostic

Description: Tests that the recipient point can process a document near the maximum GDAR document size

Procedure: Upload a valid GDAR document containing 1 valid transaction nearly the maximum document size (but not over)

Expected Pass Result: HTTP 200 OK, FA Accept

LT3 Maximum Document Size (3)

Diagnostic

Description: Tests that the recipient point can process a document near the maximum GDAR document size

Procedure: Upload a valid GDAR document containing multiple valid transactions and multiple invalid transactions nearly the maximum document size (but not over)

Expected Pass Result: HTTP 200 OK, FA Reject

LT4 Maximum Document Size (4)

Diagnostic

Description: Tests that the recipient point can process a document over the maximum GDAR document size

Procedure: Upload a valid GDAR document containing1 invalid transaction nearly the maximum document size (but not over)

Expected Pass Result: HTTP 200 OK, FA Reject

LT5 Maximum Document Size (5)

Diagnostic

Description: Tests that the recipient point can process a document over the maximum GDAR document size

Procedure: Upload a valid GDAR document containing multiple invalid transaction nearly the maximum document size (but not over)

Expected Pass Result: HTTP 200 OK, FA Reject