# Ontario Gas Distribution Access Rule (GDAR)

# EBT Transport Protocol

# Between Points

**Draft**
**Version 0.~~1~~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.

## Contents

## 1. Executive Summary

This document defines the Internet Data Transport protocol and rules for EBT transactions for communication between Points and other Points as defined by the Ontario Energy Board's GDAR EBT Working Group for the gas marketplace in Ontario, Canada.

All Market Participants active in Ontario's gas market must support the EBT Standards for communicating, but may also provide other channels (e.g. web screens instead of XML files) for performing transactions so long as the same business rules and timelines apply. That is, a participant should be neither advantaged nor disadvantaged by the channel they choose to submit their transactions.

## 2. Revision History

| Author | Version | Date | Description |
|---|---|---|---|
| Tom Stark | 0.1 | February 7, 2006 | Initial version based on draft Ontario EBT Protocol Between Points for electricity and Ontario EBT Data Transport Protocol for electricity |
| Tom Stark | 0.2 | February 21, 2006 | Revised to reflect discussions in Transport Protocol Subgroup on February 10, 2006 and February 17, 2006 |
| | | | |
| | | | |
| | | | |
| | | | |

# 3. Introduction

This document defines the Internet Data Transport protocol and rules for EBT transactions for communication between sending Points and recipient Points as defined by the Ontario Energy Board's GDAR EBT Working Group for the gas marketplace in Ontario, Canada.

This document assumes the reader is familiar with the Ontario GDAR EBT Standards Document, Public Key encryption, and HTTP.

## *3.1 Scope*

The GDAR EBT Transport Protocol Between Points defines the technical and functional standard for EBT messaging between Sending Points and Recipient Points in the Ontario gas market. This includes Point to Point messages and the responsibilities of Points.

This document consists of the following parts:

- Transport Level protocol and connections, which in turn consists of three parts:
  - The Security Protocol defines the architecture to ensure EBT message integrity;
  - The Message Protocol defines an overall framework for expressing the different types of exchanged messages and structure of these messages; and
  - The Delivery Protocol defines the logical flow of EBT messages.
- The delivery of Functional Acknowledgements;
- Time synchronisation and time stamping of transactions; and
- XML Parsers.

## *3.2 Definitions*

This document uses the following definitions:

| | |
|---|---|
| **CA** | Certification Authority. |
| **EBT Message** | A transport-protocol object that includes HTTP information and encapsulates one PIPE Document. |
| **EBT Document** | One XML instance document consisting of a PIPE Document, which in turn is made up of one or more EBT transactions. |
| **HTTP** | HyperText Transfer Protocol. |
| **HTML** | HyperText Markup Language. |

| | |
|---|---|
| **PIPE** | Partner Interface Protocol for Energy. |
| **Sending Point** | Point that sends a message |
| **Recipient Point** | Point that receives a message |
| **XML** | eXtensible Markup Language. |

### *3.3 Guiding Principles for this Protocol*

The following principles were used as guidelines to develop the Ontario GDAR EBT Transport Protocol Between Points:

- The rules for EBT communication between Vendors and Distributors are defined in the GDAR EBT Standards Document. All solutions must conform to these standards.
- Each Recipient Point must provide an inbound ~~mailbox~~ facility for Sending Points to upload EBTs to. All Points must upload their outbound EBTs to the appropriate destination ~~mailboxes~~.
- In Point to Point communication, the Market Participant Point is always the originator or end destination of the EBT; all other EBTs not specific to that Point are rejected.

### *3.4 References*

For additional information, please see the following document:

- "GDAR Electronic Business Transactions (EBT) Standards Document" for the Gas Marketing Industry (Business Rules Document).

# 4. Transport Level Protocol

This section defines the transport level issues for Point to Point connections.

## *4.1 Basic Transport Level Protocol*

This Transport Protocol encompasses the:

- security protocol,
- message protocol and
- delivery protocol

described below.

~~A Point must have a unique identifier (a self-selected unique identifier within the EBT Message standard) and a user id and password for each Point in the market that it needs to connect to.~~

Note that the basic Transport Level Protocol is end to end and therefore the EBT message will be encrypted by a Sending Point for its Recipient Point. For example, a Point sending an EBT message to another Point will encrypt the EBT message with the destination Point's Public Key and sign the message with its own Private Key.

## *4.2 Point to Point Push Communication Model*

Each Sending Point connects to a Recipient Point in order to deliver an EBT to its destination. A Recipient Point must maintain a~~n inbound mailbox~~facility in order for other Sending Points to have a place to push EBT messages to. ~~A Sending Point must also deliver outbound messages to other Point's mailboxes.~~

~~The Sending Point will not push documents more frequently than every 15 minutes.~~ A Recipient Point must be able to handle at least one concurrent connection for each trading partner who has implemented the Point to Point protocol and possibly multiple connections for some trading partners. The number of concurrent connections allowed will be determined through discussion between the two parties. If the maximum number of concurrent connections is exceeded between two parties, service may be denied by the Recipient Point~~, and for each Hub (if any)~~.

## *4.3 Store and Forward Storage Requirements*

Upon a successful EBT upload, the Recipient and Sending Points must each maintain an archive for a period of seven years of the encrypted ~~sent~~ EBTs sent and received~~in accordance with the OEB mandated retention time period~~. Original encryption/signature keys must be retained and associated with the archived messages. This archive can be used for dispute resolution.

~~It is up to the discretion of the Recipient Point what level of inbound message archiving and logging they wish to maintain to protect their interests.~~

### *4.4 Failures*

The following is ~~the minimum~~ suggested ~~recommendation~~ for handling Point to Point communication failures:

1. A Protocol Failure ~~occurs~~ is defined as any time a sender cannot connect to a minimum of one concurrent connection to the recipient's server. For example, if connection to a recipient server fails, or posting a file fails, this is a Protocol Failure.
2. An Exchange Failure ~~occurs~~ is defined as when a sender has had ~~continual~~ repeated Protocol Failures over a minimum ~~two-hour~~15 minute period. ~~Each entity is required to try at least 3 times over the two-hour period before flagging an Exchange Failure.~~
3. The Sender is encouraged to inform the Recipient of Exchange Failures. E-mail is the recommended mechanism to notify the intended recipient of Protocol and Exchange Failures, although any other method of communication may also be used. ~~This~~ Communication will assist in resolving and documenting problems.

### *4.5 Security Protocol*

Secure transmission of EBT messages requires:

- Encryption of the EBT document such that only the intended recipient may read it.
- Digital signing of the EBT document such that the sender identity and data integrity may be verified.
- A secure connection between Market Participant senders and recipients~~,~~ which~~, together with logon identification and password for the sender,~~ authenticates and validates the identity of the party. Note that this ~~does not~~ requires client or browser installed public key certificates. It ~~does~~ also requires the recipient's HTTP secure server to have a valid server side X.509 public key certificate.

#### 4.5.1 Encryption and Signature

Software and parameters compliant with the OpenPGP Internet draft RFC2440 shall be used to encrypt and sign the EBT document prior to sending, ensuring that the contents of the document may not be tampered with and may only be decrypted by the intended recipient.

##### 4.5.1.1 Rules

The following rules are in effect for security:

- Market Participant senders and Market Participant recipients must generate a Public/Private Key-pair with a ~~1024~~2048-bit key length.
- Market Participants must have only one Public Key active at any time.

##### 4.5.1.2 PGP Parameters and Options

The following PGP parameters and options must be used:

- ~~Signing Private Key must be DSA at 1024 bits~~
- Encrypting Public and Private Key must be El Gamal (ELG-E) at ~~1~~2~~2~~4~~2~~8 bits
- Digital Signatures must be computed using Digital Signature Algorithm (DSA) FIPS 186

**Formatted:** Bullets and Numbering

- ~~Cipher~~ Symmetric Encryption Algorithm must be ~~Triple D~~AES ~~(3DES)~~ 256
- Key expiration must be set at 2 years
- User ID must be in ~~format~~ form of "~~n~~Name (~~o~~Organization) <email address>"
- Message Digest Algorithm / Hash must be SHA-2
- Compression must be used: ZIP parameter option ~~is preferred,~~ in which compressed packets are compressed with RFC19~~5~~41 DEFLATE or compress-algo 1 option in Open PGP
- Public Keys sent for key exchange between Points must be encrypted in the ASCII armor mode (armor option in Open PGP, Text Output mode in PGP 8)
- All documents must be encrypted in the ASCII armor mode (armor in Open PGP, Text Output mode in PGP 8)

**Formatted:** Bullets and Numbering

### 4.5.1.3 Public Key Availability

The Public Key should be registered with the Massachusetts Institute of Technology worldwide PGP public key-server. ~~Although Market Participants are required to renew PGP keys every 2 years, it is required that legacy PGP keys must be kept for a minimum of 7 years.~~

**Comment [TS1]:** GI 765 – v3.0

### 4.5.1.4 Usage

All EBT documents passing between a participant sender and a recipient must be encrypted using the Public Key of the recipient and signed with the Private Key of the sender.

#### 4.5.1.4.1 Usage Examples

Participant A sends a document intended for participant B.

1. Sender looks up Recipient Public Key in the Sender's directory.
2. Sender encrypts document using Recipient Public Key and signs document using Sender Private Key.
3. Recipient decrypts document using Recipient Private Key and verifies signature.
4. Recipient looks up Recipient Public Key in the Recipient's directory.
5. Recipient encrypts document using Recipient Public Key and signs document using Recipient Private Key.
6. Recipient decrypts document using Recipient Private Key and verifies signature.

### 4.5.1.5 References

| | |
|---|---|
| PGP homepage. | http://www.pgp.com |
| PGP international homepage. | http://www.pgpi.com |
| MIT public PGP key server. | http://pgp.mit.edu |
| OpenPGP IETF standards | http://www.ietf.org/ids.by.wg/openpgp.html |
| Repository for free PGP | http://www.gnupg.org |

Information and software.

### 4.5.2 Secure Transmission

The HTTPS protocol shall be used to provide a secure connection between the recipient's server and a sender's client computer, delivering an encrypted stream of data~~. This is necessary to protect the sender's logon ID and password~~.

HTTPS will utilize VeriSign Digital Ids to verify the identity of the recipient's ~~S~~server and the sender's client and to encrypt the communication session.

#### 4.5.2.1 Rules

The following rules are in effect for the HTTPS connection:

- The recipients and senders must have a valid VeriSign X.509 class 3 Digital ID installed on their server.
- ~~The senders are not required to have a client-side Digital ID.~~
- EBT Documents must always be transferred to a recipient's server using the HTTPS protocol.
- ~~The Recipients in a session must present a valid Digital ID.~~

Formatted: Bullets and Numbering

#### 4.5.2.2 HTTPS connection

- port 443

#### 4.5.2.3 VeriSign Digital ID Class

- Global Server ID.

#### 4.5.2.4 HTTPS Session encryption type

- 128-bit

#### 4.5.2.5 References

VeriSign homepage. http://www.verisign.com

### *4.6 Message Protocol*

The Message Protocol defines the format of EBT messages exchanged between Market Participants over the Internet. Using the Hypertext Transfer Protocol (HTTP), EBT messages in the form of data files will be sent over the Internet. This section outlines the HTTP request and response specifications.

NOTE: HTTP will be used within the HTTPS secure connection protocol.

### 4.6.1 Definitions

HTTP-Version HTTP/1.1

Formatted: Font: Not Bold

### 4.6.2 References

HTTP/1.1 Standards Document *Hypertext Transfer Protocol – HTTP/1.1*
*Fielding, et. al*

### 4.6.3 Message Format

EBT message exchange between Market Participants must follow the HTTP/1.1 protocol.

All interactions will involve an HTTP request followed by a corresponding HTTP response.

#### 4.6.3.1 General Message Format

Refer to the HTTP/1.1 Standards Document.

#### 4.6.3.2 Required Header Fields

Every HTTP message exchanged must contain the following general headers:

- Date
- Connection

##### 4.6.3.2.1 Date

The Date field is an HTTP Date that is created by the origin client (Market Participant) in a request or by the recipient server in a response.

The Date format is according to RFC 1123 and includes a 24-hour clock with a time zone of GMT. For example:

Date: Sun, 06, Nov 1994 08:49:37 GMT

##### 4.6.3.2.2 Connection

The Connection field in HTTP/1.1 should ~~always~~ not be of type 'Keep-Alive'
The Connection format is:

Connection: Keep-Alive (??)

#### 4.6.3.3 Entity-Header Fields

Entity-Header fields are only included if the HTTP message contains an entity-body. Every HTTP message containing an entity-body must include the following mandatory entity-header fields:

- Content-Language
- Content-Length
- Content-Type.

##### 4.6.3.3.1 Content-Language

The Content-Language is:

Content-Language: en, fr

#### 4.6.3.3.2 Content-Length

The Content-Length is:

Content-Length: x

where x is the size of the entity-body in bytes and must be greater than 0.

#### 4.6.3.3.3 Content-Type

The Content-Type for is:

Content-Type: multipart/form-data; boundary=EBTpart~~;~~

### 4.6.3.4 Message-Body / Entity-Body

The Message-Body is the Entity-Body and may be either:

- an encrypted EBT document with sender and security information; or
- an XML HTTP response;
- ~~an XML HTTP response with a directory listing; or~~
- ~~empty.~~

Formatted: Bullets and Numbering

#### 4.6.3.4.1 Rules

The following rule~~s are~~is in effect for the message protocol:

- ~~If the message-body is empty, entity-headers must not be included.~~
- The plaintext EBT document must not be larger than 500Mb prior to encryption and compression. The recipient has the option of returning a FunctionalAcknowledgement of type DocReject if the size is exceeded.

## 4.6.4 HTTP Requests

Senders send HTTP requests to a recipient to upload an EBT document.

### 4.6.4.1 Request-Line

The Request-Line follows the following format:

Method SP Request-URI SP HTTP-Version CRLF

#### 4.6.4.1.1 Method

The Method indicates the “method to be performed on the resource identified by the Request-URI” and is case-sensitive.

Points only allow the POST method; all other methods will be responded to by a 501 HTTP response, indicating that the method is Not Implemented.

EBT documents are uploaded to the Recipient using the POST method. The entity-body is multipart containing the sender, security information and the encrypted EBT document to be uploaded.

NOTE: This EBT document must have a unique filename (written in the POST header information) which shall be the PIPE Document Reference Number assigned by the sender. Including the Market Participant unique ID as part of the document filename will ensure that the name is different from other market participant documents.

#### 4.6.4.1.2 Request-URI

The Request-URI identifies the ~~mailbox~~ location on the Recipient Point where the request will be delivered. Each Recipient's server must supply this URI to each Market Participant sender.

### 4.6.4.2 Request-Header Fields

The following is a list of mandatory request-header fields included with every HTTP request to the Recipient's server:

- Host

#### 4.6.4.2.1 Host

The Host is:

Host: www.ontarioRecipientServer.com: ~~x~~

where www.ontarioRecipientServer.com is the domain of a particular Recipient Point ~~and x is the port number~~.

## 4.6.5 HTTP Responses

A Recipient Point will send an HTTP response for every request it receives from a sender. These responses are not the Functional Acknowledgements to PIPE documents, but server-generated messages to acknowledge the receipt of an HTTP request and to supply any data as the entity-body.

Errors reported by the Recipient Point must be those from the transport response schema. These error responses being returned by the application server at the Recipient must include an XML response that can be validated against the transport response schema. This XML response must correspond to the information being returned in the HTTP error code and message.

If the application server at the Recipient wishes to return an error code that is not part of the transport response schema, it should use the base error code for the class of error that it is reporting (i.e., 400 or 500).

Errors being returned before the request makes it to the application server at the Recipient (i.e., from firewalls) are not required to include the XML response suffix and can therefore use other standard error codes as required.

#### 4.6.5.1 Status-Line

The Status-Line contains the protocol version, numeric status code, and an associated textual phrase.

HTTP-Version `SP` Status-Code `SP` Reason-Phrase `CRLF`

#### 4.6.5.2 Status Code and Reason Phrase

An EBT Point must support the following Status-Codes and Reason Phrases:

- 200 OK
- 400 Bad Request
- 403 Forbidden
- 408 Request Time-Out
- 500 Internal Server Error
- 501 Not Implemented
- 505 HTTP Version not supported

##### 4.6.5.2.1 200 OK

The 200 OK status code indicates that the request has succeeded.

For an Upload request, the entity body contains XML with the HTTP response. The "200 OK" status code only indicates that the document has been transferred and not that decryption has succeeded, or any other checking has taken place without error. This response includes the ~~new document id and~~ name of the document that was uploaded within the request. ~~The new document id may, or may not, be related to the name of the document~~.

Errors that are not reportable via a Functional Acknowledgement may be detected after successfully uploading a document. Typical errors of this type include problems decrypting the document and documents with headers that are mangled. In such cases, the receiving party will report the problem to the sending party via e-mail, FAX or telephone conversation. In such cases, the following types of information must be reported:

- Error type;
- File name; and
- Date and time of the error.

##### 4.6.5.2.2 400 Bad Request

The 400 Bad Request status code indicates that the Recipient could not understand the request due to malformed syntax.

The Sender should make modifications to the request before re-submitting it.

All messages where the Sender has apparently made an error in the request (4xx response) that are not explicitly included in this document will be responded to with this status code.

#### 4.6.5.2.3 403 Forbidden

The 403 Forbidden status code indicates that the Recipient Point will not fulfill the request.

#### 4.6.5.2.4 408 Request Time-Out

After achieving a secure connection through HTTPS the Recipient Point will wait a predefined amount of time for an HTTP request. If no request is received then the session will time-out. The initial suggested timeout is 90 seconds.

#### 4.6.5.2.5 500 Internal Server Error

The 500 Internal Server Error indicates that the Recipient Point encountered an unexpected condition that prevented it from fulfilling the request. No explanation need be given as to the server error.

#### 4.6.5.2.6 501 Not Implemented

The 501 Not Implemented status code indicates that the method in an HTTP request was not a POST and that the requesting party has not followed the proper Message Protocol.

#### 4.6.5.2.7 505 HTTP Version Not Supported

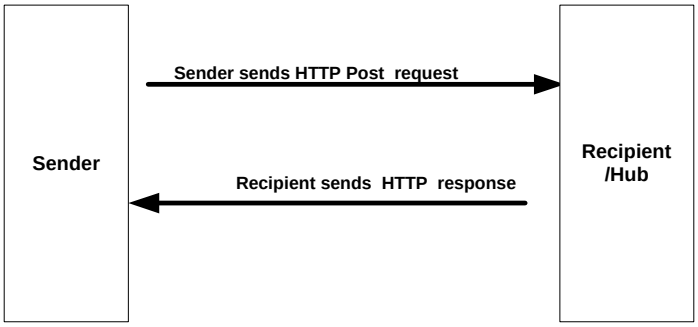
The HTTP Version must be specified as HTTP/1.1.
No other versions are supported.

### *4.7 Delivery Protocol*

The Delivery Protocol defines how to exchange EBT documents using the HTTPS transport for PGP encrypted XML within the Ontario gas market. There is one possible request / response message scenario:

- A Sending Point uploads an EBT document to a Recipient Point.

The following represents the flow of data:




General Rules
For the delivery protocol, the following rules apply:

- After sending any HTTP request, Market Participant Sending Points should not send another HTTP request to the same Recipient Point for **5** seconds.

### 4.7.1 Entity-Body Common Elements

Each multipart section of the entity body will begin with the boundary string:

EBTpart

Formatted: Bullets and Numbering

#### ~~4.7.1.1 Sender~~

~~Content-Disposition: form-data; name="sender"~~
~~12345678~~

~~where 12345678 is the Sender's unique identifier of the entity that issued the HTTP request.~~

Formatted: Bullets and Numbering

#### ~~4.7.1.2 User_id~~

~~Content-Disposition: form-data; name="user_id"~~
~~User~~

~~where User is the User's allocated User_id.~~

~~The User_id should be a string with a minimum length of 4 characters, maximum length of 16 characters, consisting only of alphanumeric characters.~~

~~The User_id and its associated password is used to authenticate that the sender of the request is a valid trading partner known to the Recipient Point and is allowed to use the requested service. This information is passed with every request.~~

~~**4.7.1.3 user_password**~~

Formatted: Bullets and Numbering

~~Content-Disposition: form-data; name="user_password"~~
~~`Password`~~

~~where `Password` is the User's password.~~

~~The password should be a string, minimum length 8 characters, maximum length 16 characters, consisting only of alphanumeric characters.~~

Formatted: Heading 4

Formatted: Bullets and Numbering

**~~4.7.1.3.1~~4.7.1.1 request_type**

Content-Disposition: form-data; name="request_type"
`Request`

Where request is a string and may only be:

- Upload
- ~~Unknown~~

Formatted: Bullets and Numbering

The 'Upload' request is used when uploading an EBT document file to a Recipient Point.

~~The 'Unknown' request is not sent as part of a request. It is to be used within a response where the original request does not equal the valid request type~~For all other requests a 400 error will be returned.

## 4.7.2 Upload Request

### 4.7.2.1 Overview

A Market Participant uploads an EBT document to a Recipient Point. See Appendix A for a sample message exchange.

### 4.7.2.2 Description

A Market Participant sends an EBT document within an HTTP request using the HTTP POST method to a Recipient Point.

The Recipient Point replies with an appropriate HTTP response, verifying receipt of the request and indicating whether it was successfully received or whether there were any errors during the transfer.

### 4.7.2.3 Rules

The following rules apply:

- Market Participants must only send EBT documents with themselves as the Sender in the ~~TradingPartnerDirectory~~ MarketParticipantDirectory node of the EBT document.

#### 4.7.2.4 Entity-body - ebt_Document

Formatted: Normal

See Appendix A for an example.

~~The EBT document part of the entity body has the following format:~~

~~Content-Disposition: form-data; name="ebt_document";~~
~~filename="transaction.xml"~~
~~Content-Type: application/octet-stream~~

~~MIME-Version: 1.0~~
~~Content-Type: multipart/encrypted; boundary="--";~~
~~protocol="application/pgp-encrypted"~~

~~--~~
~~Content-Type: pgp-encrypted~~

~~Version: 1~~

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

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

~~hIwDY32hYGCE8MkBA/wOu7d45aUxF4Q0RKJprD3v5Z9K1YcRJ2fve8~~
~~7lMlDlx4OjeW4GDdBfLbJE7VUpp13N19GL8e/AqbyyjHH4aS0YoTk1~~
~~0QQ9nnRvjY8nZL3MPXSZg9VGQxFeGqzykzmykU6A26MSMexR4ApeeO~~
~~N6xzZWfo+0yOqAq6lb46wsvldZ96YA~~
~~AABH78hyX7YX4uT1tNCWEIIBoqqvCeIMpp7UQ2IzBrXg6GtukS8Nxb~~
~~ukLeamqVW31yt21DYOjuLzcMNe/JNsD9vDVCvOOG3OCi8=~~
~~=zzaA~~
~~---END PGP MESSAGE---~~
~~--~~
~~--EBTpart--~~

~~where transaction.xml is the Sender allocated name of the file.~~
~~This filename should be a string, maximum length 255 characters.~~

Formatted: Bullets and Numbering

## ~~4.8~~ Certificates

Formatted: Heading 3

### 4.8.1 Overview

Digital Certificates compliant with the X.509 specification will be used by all clients and servers to secure transport-level (HTTPS/SSL) encryption. Digital Certificates work by binding a public key with an organizational identity to ensure that communication takes place only between valid Market Participants.

In addition to any and all standard checks performed by SSL engines, it is the responsibility of a Point in both the client and server case to ensure that all connections made comply with the following:

- Are SSL encrypted (HTTPS protocol) i.e., not clear text HTTP
- Contain a certificate signed by VeriSign, class 3 certified
- The SSL key has a 128-bit modulus (key strength)
- Are valid for the current time as embedded in the certificate
- Have a common name field (CN) matching the URL of a known valid Market Participant

Formatted: Bullets and Numbering

Connections not meeting all of the above conditions should be immediately terminated. Reasonable measures must be in place to prevent denial-of-service (DOS) attacks when repeated invalid connection attempts are made.

Note that server gated cryptography (SGC), sometimes referred to as a 'Global Server ID' is obsolete. This was a means to ensure that high-grade (128-bit) security was used during periods of tight export controls on encryption. Nevertheless, VeriSign currently incorporates SGC into their class 3 product offerings. See section 4.8.2 for more details.

### 4.8.2 Certificate Fields

Formatted: Heading 3

Formatted: Bullets and Numbering

In more detail, the structure of all certificate fields must conform to the following field-value pairs. Additional non-critical field-value pairs may be present. Comments are enclosed in parentheses.

| Field | Value |
|---|---|
| Version | 3 |
| Serial Number | *{Exact serial number, certificate dependant. Example below}*<br>09:34:23:72:E2:3A:EF:46:7C:83:2D:07:F8:DC:22:BA |
| Algorithm ID | PKCS #1 SHA-1 With RSA Encryption |
| Issuer | OU = www.verisign.com/CPS Incorp.by Ref. LIABILITY LTD.(c)97 VeriSign<br>OU = VeriSign International Server CA - Class 3<br>OU = VeriSign, Inc.<br>O = VeriSign Trust Network |

| | |
|---|---|
| Subject | {***Example values are given for CN, L,O, and OU***}<br>CN = *www.green-valley-gas.com*<br>OU = Member, VeriSign Trust Network<br>OU = {*May contain addition OU entries for informational purposes*}<br>OU = *Production Operations*<br>O = *Green Valley Gas Distribution*<br>L = *Green Valley*<br>ST = Ontario<br>C = CA |
| Public Key Algorithm | PKCS #1 RSA Encryption |
| Subject Public Key | {*Exact public key, usually displayed as a Hexadecimal representation in browsers*} |
| Validity Not Before | {*Example. Must not be before use in production systems*}<br>12/27/2005 19:00:00 PM<br>(12/28/2005 0:00:00 AM GMT) |
| Validity Not After | {*Example. Must not be after use in production systems*}<br>1/16/2007 18:59:59 PM<br>(1/16/2007 23:59:59 PM GMT) |
| Certificate Signature Algorithm | PKCS #1 MD5 With RSA Encryption |
| Certificate Signature Value | {*Exact certificate, usually displayed as a Hexadecimal representation in browsers*} |

Formatted: Heading 3

Formatted: Bullets and Numbering

### 4.8.3 References

VeriSign SSL certificate generation and support page

- http://www.verisign.com/support/ssl-certificates-support/page_dev019431.html

IETF X.509 specification

- http://www.ietf.org/html.charters/pkix-charter.html

Cambridge SSL reference and SGC notes

- http://www-uxsup.csx.cam.ac.uk/~jw35/courses/using_https/html/x773.html

Formatted: Bullets and Numbering

Formatted: Bullets and Numbering

Formatted: Bullets and Numbering

# 5. Functional Acknowledgements

This section describes the requirements for Functional Acknowledgements between Sending Points and Recipient Points. Functional Acknowledgements are used to acknowledge the delivery of PIPE Documents. They identify invalid XML formatting of PIPE Documents and good and bad formatting of PIP transactions. A more complete description of Functional Acknowledgements can be found in the "GDAR Electronic Business Transactions (EBT) Standards Document" (Business Rules Document).

## *5.1 Operation of Functional Acknowledgements*

### 5.1.1 XML Validation

The Recipient Point ~~will~~ shall XML-validate each EBT Message that it receives. After checking for errors, the Recipient Point will send a Functional Acknowledgement back to the Sending Point as described in the "GDAR Electronic Business Transactions (EBT) Standards Document" (Business Rules Document).

### 5.1.2 Point Functional Acknowledgement Processing

The following lists the rules for a Recipient Point to send Functional Acknowledgements to a Sending Point:

- If the Market Participant Recipient in the EBT document is the Recipient Point then the FunctionalAcknowledgement should be of type DocAccept if the document passes XML validation and consists of all good PIPs; otherwise the FA should be of type DocReject.
- If the Market Participant Recipient in the EBT document is not the Recipient Point then the FunctionalAcknowledgement should be of type DocReject.

The Recipient Point may send a DocReject Functional Acknowledgement if it detects other errors within the EBT document such as duplicate PIPE Document reference numbers ~~and duplicate Transaction reference numbers~~ from a Sending Point.

- If the Sending Point is unable to upload the EBT Message ~~by loading it into the destination Mailbox of~~ to the Recipient Point, then it must report the problem back to the Recipient Point. Since all Points are validating based on public schemas at the OEB web site, this should only be due to network failures. This reporting is to be done via e-mail or some other mutually agreed upon method between the Points.
- When the Sending Point receives a Doc~~ument~~Reject Functional Acknowledgement, within four hours it will ~~notify the Recipient Point and~~ trigger a process to resolve the issue. Best efforts should be made to resolve the issue.

### 5.1.3 ~~Trading Partner~~Market Participant Validation

The Recipient Point will send a DocReject Functional Acknowledgement if it determines that no ~~trading partner~~Service ~~a~~Agreement is in place. If the Sender ID does not match the certificate, a FunctionalAcknowledgement of type DocReject shall be returned. If a Service Agreement is revoked, the Certificate should be immediately revoked also.

### 5.2 ~~Availability~~ Archive Requirements

For the purposes of dispute resolution, each Point must archive all encrypted EBT ~~Messages~~Documents (with the appropriate keys) that it processes and the~~ir~~ corresponding Functional Acknowledgement~~s~~ for each Document.

### 5.3 XML Parsers

This section describes a Point's requirements for XML Parsers.

Each party is free to select and use their own XML validating parser. If two parties disagree on the XML validity of a PIPE Document, final resolution will be determined by examination of the XML according to the schemas as published on the OEB web site. Validity will be determined using the current active specification of the XML Schema Definition Language as listed on the World Wide Web Consortium (W3C) web site.

### 5.4 Detection of Other Errors

Where the error reporting tools and mechanisms of the protocol (i.e., HTTPS responses and Functional Acknowledgements) do not provide a mechanism for communicating the type of error back to the sender, the issue will be escalated based on business arrangements between the Market Participants. At a minimum, the problem will be escalated via a telephone call to the operations staff of the originator of the transaction.

Such errors include:

- Decryption errors;
- Signature errors;
- Receipt of a DocReject Functional Acknowledgement from a Recipient Point; and
- Lack of a Functional Acknowledgement from the Recipient Point within the required four hours.

# 6. Time Synchronisation and Time Stamping

This section describes the Point requirements for time synchronisation and the time stamping of transactions. In a networked environment such as the EBT marketplace and since the arrival times for transactions are used for dispute resolution, an accurate consistent time stamp is imperative.

## *6.1 Time Synchronisation*

Each Point will maintain an accurate and consistent time by connecting through the NTP protocol to a service with an accuracy of at least that provided by a stratum-2 server[1]. There are various stratum-2 timeserver sources available, including free servers and subscription servers.

## *6.2 Time Stamping of ~~Transactions~~Documents*

Each Point will date and time-stamp each EBT Document when it receives it.
The Point should use ~~Eastern Standard Time with no Daylight Savings Time change~~GMT as specified in the "GDAR Electronic Business Transactions (EBT) Standards Document".

[1] Stratum-1 servers connect to GPS or atomic clocks. Stratum-2 servers obtain their reference from stratum-1 servers. Startum-2 does not define an accuracy of the clock, but instead defines the number of hops to a source time-base. The resulting accuracy is dependent on the Internet and the number of routers between the various timeservers. By connecting to a stratum-2 timeserver, the point becomes stratum-3. Stratum-3 timeservers are expected to have a short-term drift of less than $3.7 \times 10^{-7}$ in 24 hours.

# 7. Requirements of a Point

## *7.1 Routing of Messages*

In Point to Point communication, the Market Participant Point is always the originator or end destination of the EBT. All other EBTs not specific to the Recipient Point will be rejected. All Points are expected to communicate directly.

## *7.2 Availability Requirements*

Because the processing of EBT messages by Points is critical to the success of the marketplace, Points must commit to remain online connected to the Internet ~~with no more than 15 minutes of downtime per day~~continuously. In the event of system failures, Market Participants shall make best efforts to restore service as soon as possible.

~~Where there are connectivity problems, it is suggested that a Point retry 3 times, waiting 15 minutes between retries before implementing the escalation process for handling failures. The initiator will log each transmission.~~

A suggested maintenance window exists between midnight Saturdays (EST) and noon Sundays (EST) when Point servers can be shut down for longer periods.

Different availability requirements for specific Points may apply if mutually agreed to by the two parties.

## *7.3 Market Participant Information*

In order to send an EBT message to the correct Point, each Point must maintain Market Participant addressing information within their systems. The communications parameters to be exchanged between the Points include URL, ~~User ID, Password~~Certificates, unique identifier, ~~trading partner~~Service ~~a~~Agreement contract information, and the e-mail address. In other words, the Sending Point needs to know the URL and~~,~~ ~~User ID, and Password~~Certificate to push EBTs to the Recipient Point, and vice versa. The Public Key for PGP encryption/digital signature verification should be registered with the Massachusetts Institute of Technology worldwide PGP public key-server.

~~It is the responsibility of the initiating party to contact the second party to initiate communication set up.~~

## *7.4 Routing Information Messages*

Only the Upload request type will be allowed for Point to Point communication:

```
Content-Disposition: form-data; name="request_type"

Upload
```

A Point will respond to the upload request based on the process and the response schema defined in this document.

For the format of an HTTP Upload Request, see Appendix A - “HTTP Upload Request”. For the format of a HTTP Upload successful HTTP Response see Appendix B - “HTTP Upload Response”.

## Appendix A -- HTTP Upload Request

It is not the intention that the following sample be the basis for any programming effort. Compliance to IETF relevant RFC's and W3C standards should be consulted for syntactical references.

The following is an example of a Point to Point HTTP Upload Request:

```
POST /RecipientServer/mailbox HTTP/1.1
Date: Tue, 20 Dec 2000 08:12:31 GMT
Connection: Keep-Alive
Host: www.ontarioRecipientServer.com
Content-Language: en, fr
Content-Type: multipart/form-data; boundary=EBTpart~~;~~
Content-Length: 3222

--EBTpart
Content-Disposition: form-data; name=”sender”

12345678

~~--EBTpart~~
~~Content-Disposition: form-data; name=”user_id”~~

~~aUser~~

~~--EBTpart~~
~~Content-Disposition: form-data; name=”user_password”~~

~~aPassword~~

--EBTpart
Content-Disposition: form-data; name=”request_type”

Upload

--EBTpart
Content-Disposition: form-data; name=”ebt_document”;
filename=”transaction.xml”
Content-Type: application/octet-stream

MIME-Version: 1.0
Content-Type: multipart/encrypted; boundary=”=--”;
protocol="application/pgp-encrypted"

--=--
Content-Type: pgp-encrypted

Version: 1

--=--
Content-Type: application/octet-stream
```

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

hIwDY32hYGCE8MkBA/wOu7d45aUxF4Q0RKJprD3v5Z9K1YcRJ2fve87lMlDlx4OjeW
4GDdBfLbJE7VUpp13N19GL8e/AqbyyjHH4aS0YoTk10QQ9nnRvjY8nZL3MPXSZg9VG
QxFeGqzykzmykU6A26MSMexR4ApeeON6xzZWfo+0yOqAq6lb46wsvldZ96YA
AABH78hyX7YX4uT1tNCWEIIBoqqvCeIMpp7UQ2IzBrXg6GtukS8NxbukLeamqVW31y
t21DYOjuLzcMNe/JNsD9vDVCvOOG3OCi8=
=zzaA
-----END PGP MESSAGE-----
--=----
--EBTpart--
```

## Appendix B -- HTTP Response

It is not the intention that the following sample be the basis for any programming effort. Compliance to IETF relevant RFC's and W3C standards should be consulted for syntactical references.

The following is an example of a Point to Point HTTP Upload Response:

```
HTTP/1.1 200 OK
Date: Tue, 20 Dec 2000 08:12:31 GMT
Host: www.ontarioRecipientServer.com
Content-Type: text/XML
Content-Length: 1234

<?xml version="1.0" encoding="UTF-8"?>
<RESPONSE xmlns:xsi=" http://www.w3.org/2001/XMLSchema"
xsi:noNamespaceSchemaLocation = "
http://www.oeb.gov.on.ca/Response.xsd">
  <HTTP_RESPONSE>
    <STATUS_CODE>200</STATUS_CODE>
    <REASON_PHRASE>OK</REASON_PHRASE>
    <REQUEST_TYPE>Upload</REQUEST_TYPE>
    <TIMESTAMP>Tue, 20 Dec 2000 08:12:31 GMT</TIMESTAMP>
  </HTTP_RESPONSE>
  <UPLOAD new_doc_id="12345678" doc_name="abcdefg.xml"/>
</RESPONSE>
```

# Appendix C -- Sample HTTP Responses

## C.1 200 OK. Upload

```
HTTP/1.1 200 OK
Date: Tue, 20 Dec 2000 08:12:31 GMT
Connection: Keep-Alive
Host: www.ontarioRecipientServer.com
Content-Type: text/XML
Content-Length: 1234

<?xml version=”1.0” encoding=”UTF-8”?>
<RESPONSE xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="http://www.oeb.gov.on.ca/Response.
xsd">
  <HTTP_RESPONSE>
    <STATUS_CODE>200</STATUS_CODE>
    <REASON_PHRASE>OK</REASON_PHRASE>
    <REQUEST_TYPE>Upload</REQUEST_TYPE>
    <TIMESTAMP>Tue, 20 Dec 2000 08:12:31 GMT</TIMESTAMP>
  </HTTP_RESPONSE>
  <UPLOAD new_doc_id=”12345678” doc_name=”abcdefg.xml”/>
</RESPONSE>
```

## C.2 400 Bad Request

```
HTTP/1.1 400 Bad Request
Date: Tue, 20 Dec 2000 08:12:31 GMT
Connection: Keep-Alive
Host: www.ontarioRecipientServer.com
Content-Type: text/XML
Content-Length: 1234

<?xml version=”1.0” encoding=”UTF-8”?>
<RESPONSE xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://www.oeb.gov.on.ca/Response.x
sd">
  <HTTP_RESPONSE>
    <STATUS_CODE>400</STATUS_CODE>
    <REASON_PHRASE>Bad_Request</REASON_PHRASE>
    <REQUEST_TYPE>Upload</REQUEST_TYPE>
    <TIMESTAMP>Tue, 20 Dec 2000 08:12:31 GMT</TIMESTAMP>
  </HTTP_RESPONSE>
</RESPONSE>
```

## C.3 403 Forbidden

```
HTTP/1.1 403 Forbidden
Date: Tue, 20 Dec 2000 08:12:31 GMT
Connection: Keep-Alive
Host: www.ontarioRecipientServer.com
Content-Type: text/XML
Content-Length: 1234

<?xml version=”1.0” encoding=”UTF-8”?>
```

```
<RESPONSE xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://www.oeb.gov.on.ca/Response.x
sd">
  <HTTP_RESPONSE>
    <STATUS_CODE>403</STATUS_CODE>
    <REASON_PHRASE>Forbidden</REASON_PHRASE>
    <REQUEST_TYPE>Upload</REQUEST_TYPE>
    <TIMESTAMP>Tue, 20 Dec 2000 08:12:31 GMT</TIMESTAMP>
  </HTTP_RESPONSE>
</RESPONSE>
```

## C.4 408 Request Time-Out

```
HTTP/1.1 408 Request Time-Out
Date: Tue, 20 Dec 2000 08:12:31 GMT
Connection: Keep-Alive
Host: www.ontarioRecipientServer.com
Content-Type: text/XML
Content-Length: 1234

<?xml version=”1.0” encoding=”UTF-8”?>
<RESPONSE xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://www.oeb.gov.on.ca/Response.x
sd">
  <HTTP_RESPONSE>
    <STATUS_CODE>408</STATUS_CODE>
    <REASON_PHRASE>Request_Time-Out</REASON_PHRASE>
    <REQUEST_TYPE>Upload</REQUEST_TYPE>
    <TIMESTAMP>Tue, 20 Dec 2000 08:12:31 GMT</TIMESTAMP>
  </HTTP_RESPONSE>
</RESPONSE>
```

## C.5 500 Internal Server Error

```
HTTP/1.1 500 Internal Server Error
Date: Tue, 20 Dec 2000 08:12:31 GMT
Connection: Keep-Alive
Host: www.ontarioRecipientServer.com
Content-Type: text/XML
Content-Length: 1234

<?xml version=”1.0” encoding=”UTF-8”?>
<RESPONSE xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://www.oeb.gov.on.ca/Response.x
sd">
  <HTTP_RESPONSE>
    <STATUS_CODE>500</STATUS_CODE>
    <REASON_PHRASE>Internal_Server_Error</REASON_PHRASE>
    <REQUEST_TYPE>Upload</REQUEST_TYPE>
    <TIMESTAMP>Tue, 20 Dec 2000 08:12:31 GMT</TIMESTAMP>
  </HTTP_RESPONSE>
</RESPONSE>
```

## C.6 501 Not Implemented

```
HTTP/1.1 501 Not Implemented
Date: Tue, 20 Dec 2000 08:12:31 GMT
Connection: Keep-Alive
```

```
Host: www.ontarioRecipientServer.com
Content-Type: text/XML
Content-Length: 1234

<?xml version="1.0" encoding="UTF-8"?>
<RESPONSE xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://www.oeb.gov.on.ca/Response.x
sd">
  <HTTP_RESPONSE>
    <STATUS_CODE>501</STATUS_CODE>
    <REASON_PHRASE>Not_Implemented</REASON_PHRASE>
    <REQUEST_TYPE>Download</REQUEST_TYPE>
    <TIMESTAMP>Tue, 20 Dec 2000 08:12:31 GMT</TIMESTAMP>
  </HTTP_RESPONSE>
</RESPONSE>
```

## C.7 505 HTTP Version Not Supported

```
HTTP/1.1 505 HTTP Version Not Supported
Date: Tue, 20 Dec 2000 08:12:31 GMT
Connection: Keep-Alive
Host: www.ontarioRecipientServer.com
Content-Type: text/XML
Content-Length: 1234

<?xml version="1.0" encoding="UTF-8"?>
<RESPONSE xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://www.oeb.gov.on.ca/Response.x
sd">
  <HTTP_RESPONSE>
    <STATUS_CODE>505</STATUS_CODE>
    <REASON_PHRASE>HTTP_Version_Not_Supported</REASON_PHRASE>
    <REQUEST_TYPE>Upload</REQUEST_TYPE>
    <TIMESTAMP>Tue, 20 Dec 2000 08:12:31 GMT</TIMESTAMP>
  </HTTP_RESPONSE>
</RESPONSE>
```

# Appendix D -- EBT Public Key Management

This appendix describes the details of the manual management of keys for Points. This appendix should be viewed in the light of "best practices." Further, the intent of this appendix is to provide guidelines by which Market Participants may conduct the management of keys.

## D.1 Background

In the transfer of data to and from Points there are inherent risks of security. Primary among these risks is the possibility of Public or Private Keys falling into unauthorized hands. The suggested infrastructure contained in this appendix addresses the management of keys in order to safe-guard the data transfer process. The person or persons responsible for safe guarding Public Keys should be kept to a minimum and shall be referenced in this appendix as a security unit (Key Manager Unit, KMU).

## D.2 Scope

The suggested EBT Key Management Protocol defines the technical and functional standard for manually managing Public and Private Keys between EBT senders and EBT recipients in the Ontario gas market.

The management categories included in this appendix are:

- Public Key / Private Key Creation / Public Key protection
- Public Key distribution
- Public Key Storage (PKCS #12)
- Retiring / Revoking Public keys
- Private Key security / Private Key Storage
- Expiring keys and Keys that expire
- Window key management (between expired or revoked keys)
- ~~Spoke~~ Point Proactive expiration (window anticipation)
- Use of Expired or Revoked keys

The intricacies of X.509 certificates, encryption and hashing algorithms, non-repudiation (digital signatures), Storage only repositories – Certificate Servers (Key Servers), Public Key Infrastructures (PKI), and Certification Authorities are beyond the scope of this appendix.

Further, the following sections take precedence in the case of a conflict between this appendix content and the GDAR Electronic Business Transactions (EBT) Standards Document.

## D.3 References

For additional information, please see the following document:

- "GDAR Electronic Business Transactions (EBT) Standards Document" (Business Rules Document), Ontario GDAR EBT Working Group.

## D.4 Point Key Management

The Public Key storage and distribution process is currently manual. The use of certificates and key servers are a future implementation option. In lieu of the aforementioned scenario, it is suggested that the following approach be defined that will ensure:

- A secure and auditable Public Key exchange with the sender providing proof of possession of the Private Key and maintaining an audit trail for expiration and revocation purposes.
- Proper policing and management (import/export, expiration, revocation) of the keys.
- Public human contact or contacts (KMU) to be responsible for the keys.

## D.5 Public Key / Private Key Creation / Public Key protection

Each Market Participant is responsible for generating and policing his own Public/Private key pairs. An individual employed by the Market Participant's business entity shall be publicly designated as the Key Manager for that entity. The generated key pair should meet the requirements set forth in the Transport Protocol Between Points document (OpenPGP). The Public Key of the Market Participant will be "self-signed" before distribution.

## D.6 Public Key distribution

As a best practice, the Public Keys should be distributed via e-mail or other method agreeable to the sender and recipient of the Public Key. It is advised that the Public Key be distributed solely by the unit (KMU) responsible for and in possession of the Private Key. Upon receipt of a Public Key the recipient will confirm authenticity of the Public Key by contacting the sender/owner of the Public Key via telephone and requesting a reading of the Public Key's fingerprint. If the fingerprint of the Public Key received matches the fingerprint of that sent, the recipient should sign the Public Key in the PGP Public Key Ring. A Key Distribution Log will be kept by the Key Management Unit and made available to other Market Participants at their request. Making the distribution log available should be done so that everyone is apprised who has a copy of the Public Key. The log shall contain:

1. The date sent
2. Method of distribution
3. Key ID(s)

4. Fingerprint(s)
5. Size or CRC of the extracted key(s)
6. The name of the person to whom the key was sent
7. The date they received the key(s) and confirmed authenticity
8. The production activation date/time of the key(s).

## D.7 Public Key Storage

Public Keys should be stored on securable electronic media such as a floppy disk, magnetic tape, or CD-ROM. When the electronic media is connected or installed to/on a computer, it should only be accessible by the Key Manager Unit and only when they are at the console of the computer.

## D.8 Revoking Public keys

When a Public Key is revoked a new Public/Private Key pair should be created at that time. The Market Participant is responsible for distributing the revoked key and the new key via the agreed distribution mechanism. They should refer to their Key Distribution Log to ensure that all parties who received the original/revoked key will be sent and do receive a copy of the revoked and new keys. If a Key Management entity of the KMU leaves employment of the Market Participant's business entity the Public Key should be revoked.

## D.9 Private Key Security / Private Key Storage

As a best practice, the Key Management Unit is the only entity authorized to handle the Private Key on any type of electronic media. When the Private Key is installed on a computer attached to a network, proper physical and system security should be in place to ensure that the Private Key file is only accessible by the KMU and only when they are physically located at the computer's console. It is recommended that a dedicated user account be setup for this purpose, and that any automated encryption and decryption programs that make use of the Private Key for decryption and signing purposes also be executed under the context of this user account. The user account should have no network logon privileges. When the Private Key file is stored on removable electronic media, that media should be stored in a physically secure location.

## D.10 Expiring keys and Keys that expire

Keys will be set to expire every 2 years per the Transport Protocol. Certificates are being left out for now, so no further action is necessary.

## D.11 Proactive/windowed key expiration

New keys should be generated and distributed prior to the old key expiring, in order to allow for the distribution timeframe and to prepare automated systems so they do not report errors. It is recommended that the new key be distributed at least 7 and no more than 28 business days prior to the key expiration date. This will allow ample time for the key to be received and installed before it starts being used. The EBT sender will communicate an exact time and file identification to the EBT recipient when a new Public Key is activated and first used.

## D.12 Use of Expired or Revoked keys

Expired and revoked keys shall not be used to encrypt or sign any documents. They may be used at the operator's discretion to decrypt or verify signature on archived documents