Canada Customs and Revenue Agency | Agence des douanes et du revenu du Canada

# CUSTOMS INTERNET GATEWAY

# PARTICIPANT REQUIREMENTS DOCUMENT

November 2003

## Table of Contents


1. INTRODUCTION ..... 3

2. PUBLIC KEY INFRASTRUCTURE (PKI) ..... 6

2.1 WHAT IS A PKI? ..... 6
2.2 PKI REGISTRATION PROCESS ..... 6
2.3 CERTIFICATE REVOCATION ..... 9
2.4 CERTIFICATE RECOVERY ..... 10

3. CLIENT TEST PROCEDURES ..... 11

3.1 INTRODUCTION ..... 11
3.2 TESTING REQUIREMENTS ..... 11
3.2.1 INTERNAL CLIENT TESTING ..... 11
3.2.2 INTERNET SERVICE SUPPLIER NETWORK TESTING ..... 11
3.2.3 INTERNET/CLIENT COMMUNICATION SERVICES ..... 11
3.3 TEST CASES: CUSTOMS EDI APPLICATIONS ..... 12
3.3.1 CADEX AND CUSDEC ..... 12
3.3.2 ACROSS ..... 12
3.3.3 RELEASE NOTIFICATION SYSTEM ..... 13
3.4 LINES OF COMMUNICATION ..... 13
3.4.1 REPORTING AND RESOLVING PROBLEMS ..... 13

ANNEX A: TECHNICAL (PROTOCOL) SPECIFICATIONS ..... 15

ANNEX B – PKI SOFTWARE REQUIREMENTS ..... 48

ANNEX C – PKI/INTERNET TECHNICAL REQUIREMENTS ..... 50

# 1. Introduction

The Canada Customs and Revenue Agency (CCRA) has developed the Customs Internet Gateway to provide customs importers and brokers with a way to send and receive CADEX, CUSDEC, ACROSS, and RNS data over the Internet. The CCRA has adopted a Public Key Infrastructure (PKI) to provide for the security and integrity of the data transmitted.

The Customs Internet Gateway process involves the following:

- **Sending data to the CCRA**

The client prepares accounting and/or release transactions. The client's application software formats the data in either the CADEX or EDIFACT message format. The transactions are encrypted and digitally signed by PKI software at the client's facility(see Annex E - PKI/Internet Technical Requirements "Target searchbases" for details). The transactions are sent through the Internet using the HTTP protocol. The transactions are received in the Customs Internet Gateway. An acknowledgement message for the sender is generated when the Customs Internet Gateway receives the transaction. The transaction is decrypted and the digital signature verified. The transaction is submitted for processing by the CADEX or ACROSS system. CADEX or ACROSS may generate a transaction-specific acknowledgement or error message as defined in the Participant Requirements Documents (PRDs) that apply.

- **Receiving data from the CCRA**

The ACROSS and CADEX applications process client data. The Customs Electronic Commerce Platform (CECP) formats the data in either the CADEX or EDIFACT message format. The transactions are encrypted and digitally signed by the Customs Internet Gateway. Customs places outbound messages in the clients' queues. Periodically, the client's application queries the CCRA for messages. The client's application receives transactions using the HTTP protocol. The client's application decrypts the file and verifies the CCRA's digital signature. The decrypted CADEX or EDIFACT message is submitted for processing by the client's business system.

The Internet connection will support sending/receiving the same messages (CADEX, CUSDEC, ACROSS, and RNS messages) that are currently sent through either a CADEX line or a value added network (VAN).

This Customs Internet Gateway Participant Requirements Document (PRD) includes all of the information required for an existing Customs EDI client to begin making use of the Internet alternative. New clients that are not already taking advantage of the EDI options offered by the CCRA can get more information, as well as copies of the appropriate Participant Requirements Documents, by contacting:

Manager
Electronic Commerce Unit, Client Services Division
Operational Policy and Coordination Directorate, Customs Branch
Canada Customs and Revenue Agency
15th Floor, 191 Laurier Avenue West,
Ottawa Ontario
K1A 0L5
Telephone: (888) 957-7224
Fax: (613) 952-9979
Email: ecu.uce@ccra-adrc.gc.ca

The CCRA will also offer participants the option of maintaining a backup communication method to transmit EDI data in the event of a system outage. This will apply to situations when either the participant or the CCRA experiences communication problems for a period of 4 hours and beyond. If you wish to maintain a backup communication plan, please advise the Electronic Commerce Unit of your

intention when you apply to transmit data over the Internet. An ECU Representative will be in contact with you to obtain the details. The type of information that will be asked is:

- The back up method of communication that you intend to use (i.e. CADEX line, the Value Added Network);
- If you are a current EDI participant, will you maintain your existing EDI profile such as CADEX, ACROSS, RNS, CUSDEC;
- The applications you want set up on a back up system, i.e. RNS, CADEX B3, ACROSS;
- The format you will you be sending your EDI data, i.e. CADEX or Edifact; and
- Your mailbox identification, this will apply to participants using a Value Added Network as their secondary communication method.

You will also be required to communicate with your software provider to advise that you are switching communication methods.

The CCRA has elected to manage a Certification Authority for issuing PKI certificates to its business partners. The CCRA PKI is based on the Entrust product suite. Participants will be registered and issued a CCRA PKI certificate. However, participants will have to purchase the software package required to access the certificate (i.e. Entrust Entelligence) and the user installation guide. **The certificate belongs to the CCRA and is to be used for CCRA business only.**

This document is divided into two additional sections. Section 2 includes a description of PKI and instructions on how clients can register as PKI subscribers with the CCRA.

Section 3 outlines the client test procedures for EDI clients who currently conduct electronic business with the CCRA in the production environment. New clients should contact the Manager of the Electronic Commerce Unit. All client test requirements/details are fully elaborated in PRDs that have been previously published by the CCRA for the CADEX, CUSDEC, ACROSS and RNS applications. Section 3 also outlines whom clients should contact if they have difficulty using the Customs Internet Gateway.

Annex A contains a general outline of the technical (protocol) specifications for the general syntax and coding constructs needed to communicate with CCRA Customs commercial applications over the Internet. **Once a client has registered to participate in the Customs Internet Gateway project they may obtain more detailed information on these specifications by contacting:**

Manager
Electronic Commerce Unit, Client Services Division
Operational Policy and Coordination Directorate, Customs Branch
Canada Customs and Revenue Agency
15th Floor, 191 Laurier Avenue West,
Ottawa Ontario
K1A 0L5

Telephone: (888) 957-7224
Fax: (613) 952-9979
Email: ecu.uce@ccra-adrc.gc.ca

**It is absolutely critical that these specifications not be shared with anyone outside of your organization. Once the CCRA has provided the detailed specifications to the organization, it is the organization's responsibility to safeguard this information and ensure it is not distributed to outside parties.**

Clients who want to program Internet connectivity to the CCRA for their own systems should follow these specifications. Clients who currently deal with a software supplier should check with their supplier to determine what additional system requirements and costs there may be for establishing the Internet

connection to the CCRA. Detailed system specifications are available from the ECU for clients who want to program this connectivity for their own systems. **It is important to note that this interface connectivity must be undertaken prior to PKI registration.**

If you have questions about this PRD, contact the Manager of the Electronic Commerce Unit at the address above.

# 2. Public Key Infrastructure (PKI)

## 2.1 What is a PKI?

A PKI is an automated system that manages the generation, maintenance, and delivery of encryption and digital signature keys. Together, encryption and digital signature keys provide:

- **Confidentiality** – Data is obscured and protected from view or access by unauthorized individuals.
- **Integrity** – The verifier of a digital signature can easily determine whether or not digitally signed data has been altered since it was signed.
- **Authentication** – Users can securely identify themselves to other users and servers on a network without sending secret information (such as passwords) over the network.
- **Non-repudiation** – Users who digitally sign data cannot successfully deny signing that data.
- **Access control** – Data can only be accessed in a comprehensible form by those specifically identified when data was encrypted.

Both key types, encryption and digital signature, have two related components: a public-key component that is accessible to all users, and a private-key component that must be secured from access by others. The public key and other identification information are stored in a digital certificate that is digitally signed by a Certification Authority (CA). The digital signature of the CA on the digital certificate binds the identity of the end-entity with its public-key. It also guarantees that the public key has not been tampered with.

To provide a level of assurance or trust in the CA, certain policies and procedures must be followed. One of the main issues is the registration process, which involves how a client is identified and authenticated before a digital certificate is issued.

## 2.2 PKI Registration Process

To take advantage of this new Internet Gateway project, participants must become CCRA PKI subscribers. To register as a subscriber, all participants in the project must be properly identified and authenticated, adhere to terms and conditions regulating the use of the CCRA digital certificates, and acquire the PKI client software.

The registration process will take place on-line from the CCRA PKI registration Web site, except for signing and returning the required organization and subscriber agreements. **The employee (who we will refer to as the "designated representative") who will, at the end of the process, manage the digital certificate(s) must do the registration.**

All participants, to the satisfaction of the CCRA, must complete the following steps before they may take advantage of the new Internet Gateway project.

**Step 1**

The designated representative of a client such as, but not limited to, a customs broker/importer (who we will refer to as the "organization") who is interested in participating in the program will review the PRD. After reviewing this information, the designated representative must visit the secure on-line registration Web site and provide the following information to begin the process:

The designated representative will download and complete the PKI agreements which include the Organization Agreement, Appendix A to the Organization Agreement, the Subscriber Agreement and the Subscriber Application Form (RC129). These documents can be found at:
https://reg-pki-ext.ccra-adrc.gc.ca/cig/requirements.do

**Organization agreement** – a senior official of the organization who has the authority (the organizational authority) to bind the organization must sign this document. This agreement details the terms and conditions relating to PKI participation, as well as the responsibilities of the organization. This agreement also identifies those personnel appointed to receive and manage digital certificates

**Appendix A to the Organization Agreement** – a senior official of the organization who has the authority to bind the organization must complete this document. This appendix identifies those personnel appointed to receive and manage digital certificates. This appendix must include the Business Number (BN) for the organization as well as the email address of the primary contact

**Subscriber agreement** – This document must be signed by the designated representative(s) appointed by the organization agreement to receive and manage digital certificates. This agreement details the terms and conditions for use and custody of a digital certificate.

**Subscriber Application Form (RC129)** – this form must be completed by the employee (the "designated representative") who will, at the end of the process, manage the digital certificate(s). The designated representative must complete Section 1 of the RC129 and must include his/her email address. The Organizational Authority must complete Section 2 of the RC 129. The information contained on this form is protected and is retained in CCRA PPU 165 Public Key Infrastructure for External Clients.

Once the above-mentioned agreements are completed and signed they must be returned to the CCRA representative. In addition, the organization agreement, when signed on behalf of a corporation, must be accompanied by a certified copy of the corporation's signing by-law, as well as a certified copy of a certificate of incumbency, or, when signed on behalf of a partnership, must be accompanied by a certified copy of proof of name registration.

The PKI agreements and accompanying documents can be sent by mail or courier only to:

Manager
Electronic Commerce Unit, Client Services Division
Operational Policy and Coordination Directorate, Customs Branch
Canada Customs and Revenue Agency
15th Floor, 191 Laurier Avenue West
Ottawa ON K1A 0L5

Telephone: (888) 957-7224
Fax: (613) 952-9979
Email: ecu.uce@ccra-adrc.gc.ca

Once the CCRA representative receives the above-noted documents, he or she will ensure they are complete. The senior official who signed the corporate agreement may be contacted if additional information is needed to complete the process of authenticating the designated representative and the organization. When the CCRA representative is satisfied that all conditions have been met, an email will be sent to the designated representative to invite them to complete the application for a digital certificate.

## Step 2

**Important note**: To complete this step you will need a 128-bit encryption enabled Web browser and Internet connectivity. Please see "Annex E – Technical Requirements" for details.

The designated representative will receive notification via email that includes his ten-character registration password to access the secure on-line registration Web site. After reviewing this information, the designated representative must visit the secure on-line registration Web site, click "Log In" on the left menu and provide the following information to access the site:

- The organization's CCRA nine-digit Business Number (BN)
- The ten-character registration password provided in the notification email

Once in the site, the designated representative should choose "Initial Registration" from the left menu.

**A.** On the initial registration page, the designated representative will review the following

**Organization information**

- Name of the organization (full legal name)
- The organization's CCRA nine-digit Business Number (BN)
- Full organization mailing address, including postal code
- Email address

If the information is correct, the designated representative will input his email address.

**B.** On the contact information page, the designated representative will provide the following

**Personal information**

- Designated representative (full name)
- Mailing address (if different from organization address)
- Email address
- Telephone number
- One personalized secret and their associated hint — this should be something that is easy to remember but relatively unknown to other people (This personal secret will be used to authenticate the designated representative when he or she returns to the CCRA PKI registration Web site.) **Both the Secret and the Hint must be a minimum of eight characters.**
  For example:
  - ⇒ Secret = SOCRATES
  - ⇒ Hint = my first pet
- The number of devices being registered

**C.** On the contact information page, the designated representative will provide the following

**Device information**

- The name of the device(s) on which the certificate profile will reside. The name chosen should be unique and meaningful; typically, this will be the same name you have assigned the computer.

Once the above information has been properly entered, a 16-digit Registration Number will be generated and assigned to each certificate request. This 16-digit Registration Number will appear on the screen, and must printed for future use and reference.

The information completed on the CCRA PKI registration Web site must match the completed PKI Agreements and accompanying documents.

**Step 3**

While step 1 and 2 are underway, the process of acquiring and installing the client cryptographic software can be carried out. This is a commercial transaction between the organization and the software vendor. It is a prerequisite to the final registration step for obtaining a CCRA digital certificate.

Please see “Annex D – PKI Software Requirements” for details.

**Step 4**

**Important note**: The client cryptographic software must be installed and functional on the system to be used to return to the CCRA PKI registration site for the final step of obtaining the digital certificate.

An email will be sent upon approval of the device registration. There will be an embedded URL to the CCRA PKI registration site. The designated representative can simply click on the URL to link to the secure CCRA PKI registration Web site.

The designated representative will be prompted to enter the BN of the organization and the registration password in order to access the site.

Once in the site, the designated representative should choose “Create or Recover Profile” from the left menu.

The BN and Organization name will be shown. The designated representative will enter the 16-digit Registration Number for the device.

The Device Name, Contact Name and Hint will be shown. The designated representative will enter the Secret (minimum eight characters).

Once the representative has been identified, the automated certificate issuance process will be triggered and a client profile will be created on the system being used to access the Web site. The profile can then subsequently be moved to the target device (system) that will be performing the CCRA transactions such as CADEX, CUSDEC, ACROSS, RNS, Marine and CSA.

## 2.3 Certificate Revocation

There are certain circumstances when the organization must immediately request that the CCRA revoke their certificate(s). These circumstances are:

a) If the password, token or private keys of the Subscriber have been, or if it is suspected they may have been compromised or are insecure in any way;
b) If any of the information contained in the certificate(s), or the identification and authentication information has been changed or altered, or is otherwise no longer accurate or complete; or
c) If the device(s) holding the certificate is lost or stolen, or is no longer authorized or designated for the Customs Internet Gateway project.

The certificate revocation request should be sent to the CCRA client representative via a digitally signed email. If this is not possible then the designated representative must contact the CCRA client representative by phone. The CCRA client representative will also need to contact the senior official in order to establish proper authentication and authority prior to the revocation of the certificate(s).

The organization may request a new certificate at anytime by going through the online registration process outlined in section 2.2.

## 2.4 Certificate Recovery

To ensure continued trust, the CA will perform periodic key and profile updates. This function is completed automatically with little or no user intervention. However, certain circumstances will require a subscriber to request and perform a manual or forced update. The most common conditions are:

a) A lost or forgotten password;
b) Profile corruption or deletion (typically due to hardware or software failure); or
c) Accidental deletion.

In these situations, the designated representative will have to return to the PKI registration site and select the recovery option versus the create option.

The designated representative will be prompted to enter the BN and registration password to access the PKI Registration site and follow the steps listed in Step 4 of Section 2.2, PKI Registration Process, above. Once the representative has been identified, the automated key recovery process will be triggered and the designated representative will be prompted for a new profile name and password.

# 3. Client Test Procedures

### 3.1 Introduction

This section provides information on the testing stages for transmitting data to the Canada Customs and Revenue Agency (CCRA) via the Internet. **Please note that these requirements are specifically for clients that are currently in the production environment.** Specific communication tests required for each individual Customs EDI application are outlined below.

**For new clients transmitting via the Internet, the conditions outlined within the specific EDI systems' Participant Requirements Document (PRD) must be followed.** All participants should be aware of these to ensure that they are accounted for within their development plans. You can get copies of the appropriate PRD for the EDI application you are interested in by contacting:

Manager
Electronic Commerce Unit, Client Services Division
Operational Policy and Coordination Directorate, Customs Branch
Canada Customs and Revenue Agency
15th Floor, 191 Laurier Avenue West
Ottawa ON K1A 0L5

Telephone: (888) 957-7224
Fax: (613) 952-9979
Email: ecu.uce@ccra-adrc.gc.ca

### 3.2 Testing requirements

#### 3.2.1 Internal client testing

Participants will be responsible for all aspects of internal testing of their Internet interface components. This should include testing of:

- All customized application code used to process the input and output records;
- Any necessary error handling and exception reporting routines; and
- The processing of acknowledgement messages to and from the CCRA.

#### 3.2.2 Internet service supplier network testing

The network-testing component will be the responsibility of participants in conjunction with their Internet service provider (ISP). The ISP, to help clients in this testing phase, should provide direct support. It is not necessary for the CCRA to be involved at this level of testing. Once successful communication has been established with the service provider, clients can advise their client representative in the Electronic Commerce Unit that they are ready to proceed with connectivity to the CCRA.

#### 3.2.3 Internet/Client communication services

The responsibility for the interface to the Internet service provider (ISP) rests with the supplier of the service, your ISP. As such, any problems related to the connection and/or transmission of data over the Internet is to be resolved by the network provider.

The following is the basic process to be followed to establish an Internet connection:

- Determine which Internet service provider your organization will use to interface with the CCRA.

- Advise your ISP that you will be transmitting encrypted messages to the CCRA.
- Establish a communications interface of appropriate hardware and software.
- Acquire and install the Entrust software.
- Configure and test all components and interface routines with the ISP.
- If your company is already a client with an ISP, then you need to meet the requirements identified in Annex D. In addition, you will need to install the encryption program and set up the trading partner arrangement with the CCRA. Transmission of data is possible at this point.

**You have to ensure that you are able to receive an acknowledgement of receipt from the CCRA for every message you transmit to the CCRA.**

### 3.3 Test cases: Customs EDI applications

The following test cases, according to application, must be completed before transmitting data to the CCRA over the Internet. Please contact the Electronic Commerce Unit at (888) 957-7224 to arrange a testing date. **Clients will be asked to confirm that they have received acknowledgement of receipt of transmission by the CCRA for all test cases transmitted.**

#### 3.3.1 CADEX and CUSDEC

a) Transmit five entries that you would typically use in production. This will test both the acceptance of your entries and the generation of the K84 Accounting Statement.

b) Query the classification and the exchange rate files. This will test that the correct information is being sent.

c) Provide the test representative in the Electronic Commerce Unit with four unused transaction numbers that will be used to create release records. This will test the generation of the Release Notification Report for CADEX clients and the Automatic Release Notification Messages for CUSDEC clients. It will also generate the Overdue Release Notification Report for both CADEX and CUSDEC clients.

d) Optional – Provide the test representative of the Electronic Commerce Unit with one or a range of classification numbers to test the classification download.

#### 3.3.2 ACROSS

a) Transmit an original PARS with Appraisal Quality data to follow.
Change the transaction.
Cancel the transaction.

b) Transmit an original RMD with Appraisal Quality data.
Change the transaction.
Cancel the transaction.

c) Transmit an original RMD without the Appraisal Quality Data. The test representative will create a Y51 and contact the client to ensure a code number 6 or the text has been received.

d) Transmit an original RMD with Appraisal Quality Data. The test representative will refer the transaction for examination and contact the client to ensure that a code number 5 or the text has been received.

e) Transmit an RMD with appraisal quality data with three invoices (Purchase Orders) with five lines of description each.

### 3.3.3 Release notification system

#### 3.3.3.1 Automatic release notification system

Provide five transaction numbers and five cargo control numbers (CCNs). The test representative will create five release records. This test will ensure that the client is receiving the automatic release notification message.

#### 3.3.3.2 Arrivals

a) Transmit an arrival (631) with a CCN (to be provided by the client) using release office 0677. This will test that the client receives an error code 6, meaning invalid customs office.

b) Transmit an arrival (631) with transaction number (to be provided by the client) using release office 0395. This will test that the client receives an error code 11, meaning arrival by transaction number not permitted.

c) Transmit an arrival (631) with a CCN (to be provided by client) using release office 0395, and a future date. This will test that the client receives an error code 4, meaning arrival date is future dated.

d) Transmit an arrival (631) with a CCN (to be provided by the client) using release office 0398 and current date. This will test that the client receives a GIS code 4, meaning the goods are released.

#### 3.3.3.3 Status query

a) Transmit status query (998) with a CCN (provided by the client) using release office 0395 and current date. This will test that the client receives a GIS code 9 (accepted, awaiting arrival of goods) or code 4 (goods released) depending on the type of release.

b) Transmit status query (998) with a transaction number (provided by the client) using release office 0395 and current date. This will test that the client receives a code 02, meaning the transaction number is not on file.

c) Transmit status query (998) with a CCN (to be provided by the client) using release office 0395 and future date. This will test that the client receives an error code 01, meaning the CCN is not on file.

d) Transmit status query (998) with transaction number (to be provided by the client) using release office 0667 and current date. This will test that the client receives a GIS code 9 (accepted, awaiting arrival of goods) or code 4 (goods released) regardless of invalid office code.

## 3.4 Lines of Communication

### 3.4.1 Reporting and resolving problems

Each testing participant in this project will be assigned a client representative from the Electronic Commerce Unit (ECU), Client Services Division. The ECU client representative will be the first point of contact for all problems and queries related to transmitting data on the Internet, except for the initial network-testing phase.

Depending on the nature and circumstances of the problem, the client representative will determine if the problem is within the communication structure for the CCRA. If not, you will be requested to seek assistance from your ISP. For technical/application problems, the client representative will assign it to the appropriate area within the CCRA's Information Technology Branch.

The following outlines the basic procedures to be followed for reporting problems at various stages of the project:

**Initial ISP network testing**
Participants will address network/communications problems directly with the Internet service provider as defined by the vendor. Serious and/or ongoing network problems should be referred to the client representative, Client Service Division.

**Application testing**
Participants who encounter problems during this phase will contact their assigned client representative. This individual will carry out the initial investigation of the problem, and if unable to resolve it, will enter a problem in the CCRA problem management system for assignment to the appropriate support section. The CCRA's Client Services Division will assume overall responsibility for tracking and escalating problems for testing and production.

**Internet production**
Network problems should be reported to the ISP for initial problem determination and resolution. **All operational or application problems should be referred to the Electronic Commerce Unit at (888) 957-7224 (Email: ecu.uce@ccra-adrc.gc.ca) for prompt assistance by a client representative.**

**Implementation**
Once all testing requirements have been successfully completed, clients should schedule an implementation date to the production environment with their client representative.

# ANNEX A: Technical (Protocol) Specifications

# Electronic Commerce Services Protocol Specification

Customs Commercial Applications -
Secure File Transport via the Internet using the Hypertext Transfer Protocol (HTTP) and the EDIINT draft standards.

# Revision History

| Revision Number | Summary of Revision |
|---|---|
| 00.0327 | Original document |
| 00.0606 | • The header definition table in Section 7.1 referring to the HTTP version in the HTTP command previously indicated a setting to "HTTP/1.1", referencing RFC 2616.<br><br>**This has been changed to "HTTP/1.0" and now references RFC 1945.**<br><br>• Section 7.10 defines return code 204 as indicating no data is queued at the host for delivery.<br><br>**This has been changed to return code 205.**<br><br>• Section 7.9 has added an illustration of a multiple message download from the response queue.<br><br>• Section 7.10 now illustrates the format of a message indicating there is no data queued for the recipient. |
| 00.0703 | • Section 7.10 defines return code 204 as indicating no data is queued at the host for delivery. In revision 0606 this was changed to return code 205. Changes to the HTTP server have resulted in **return code 204** being returned to indicate no data queued. Therefore the original specification for no data queued will remain in effect.<br><br>• All URL's now reflect the OPS environment.<br><br>• Furthermore, due to HTTP 1.1's mandatory "chunking" and other features not relevant to the CCRA implementation, HTTP 1.0 will be considered the reference specification for the implementation. As such all references to HTTP 1.1 have been modified to reflect this change.<br><br>• Appendix A, CPP Source Code, has been removed. |

## Table of Contents


1. SCOPE ..... ERROR! BOOKMARK NOT DEFINED.

2. REFERENCES ..... ERROR! BOOKMARK NOT DEFINED.

3. DEFINITIONS ..... ERROR! BOOKMARK NOT DEFINED.

4. SYMBOLS AND ABBREVIATIONS ..... ERROR! BOOKMARK NOT DEFINED.

5. GENERAL OVERVIEW ..... ERROR! BOOKMARK NOT DEFINED.

6. CANADA CUSTOMS AND REVENUE AGENCY ENVIRONMENTERROR! BOOKMARK NOT DEFINED.

- THE THREE TIER MODEL ..... ERROR! BOOKMARK NOT DEFINED.
- TIER TWO - THE APPLICATION SERVER ..... ERROR! BOOKMARK NOT DEFINED.
- ADDRESSING WITHIN TIER TWO ..... **ERROR! BOOKMARK NOT DEFINED.**
- SPECIFIC TIER 2 ADDRESSES WITHIN CCRA ..... ERROR! BOOKMARK NOT DEFINED.
- TIER THREE - THE DATA SOURCE ..... ERROR! BOOKMARK NOT DEFINED.

7. LEXICAL DEFINITIONS ..... ERROR! BOOKMARK NOT DEFINED.

- THE MESSAGE HEADER ..... ERROR! BOOKMARK NOT DEFINED.
- THE MESSAGE BODY ..... ERROR! BOOKMARK NOT DEFINED.
- THE MULTIPART CONTENT-TYPE ..... ERROR! BOOKMARK NOT DEFINED.
- EDI-INT'S S/MIME : SIGNATURE AND ENCRYPTION (MULTIPART/SIGNED)ERROR! BOOKMARK NOT DEFINED.
- CONTENTS OF THE BOUNDARY ..... ERROR! BOOKMARK NOT DEFINED.
- SECURING THE EDI INTERCHANGE MESSAGE. ..... ERROR! BOOKMARK NOT DEFINED.
- MESSAGE DELIVERY NOTIFICATION AND THE SECURE MESSAGE LOOP.ERROR! BOOKMARK NOT DEFINED.
- REQUESTING A DOWNLOAD OF TRANSACTION RESPONSES. ..... ERROR! BOOKMARK NOT DEFINED.
- THE RESPONSE MESSAGE. ..... ERROR! BOOKMARK NOT DEFINED.
- RESPONSE CODES ..... ERROR! BOOKMARK NOT DEFINED.

8. ENCRYPTION OVERVIEW ..... ERROR! BOOKMARK NOT DEFINED.

- SINGLE KEY ENCRYPTION ..... ERROR! BOOKMARK NOT DEFINED.
- PUBLIC KEY ENCRYPTION ..... ERROR! BOOKMARK NOT DEFINED.
- DIGITAL SIGNATURES ..... ERROR! BOOKMARK NOT DEFINED.
- ENTRUST IMPLEMENTATION ..... ERROR! BOOKMARK NOT DEFINED.

9. PASSWORDS ..... ERROR! BOOKMARK NOT DEFINED.

# 1. Scope

This specification describes the general syntax and coding constructs required to communicate with Canada Customs and Revenue Agency Customs Commercial Applications over the Internet using:

- The PKCS#7 Cryptographic Message Syntax Standard in conjunction with the
- EDI over the Internet draft standards proposals as well as,
- The Hypertext Transfer Protocol (HTTP)

PKCS#7 support is provided through the use of Entrust Technologies' Entrust Toolkit.

Although this specification is intended to be non-specific in its content, in areas where remaining generic would have defeated the purpose of the description (e.g. coding constructs):

- The computer language used was C++,
- For application name, "CADEX" was used.

# 1. References


| | |
|---|---|
| PKCS # 7 | RSA Laboratories, PKCS #7 Cryptographic Message Syntax Standard, Version 1.5, November 1993 |
| RFC 1945 | Hypertext Transfer Protocol, HTTP1.0 T. Berners-Lee et al; May 1996 |
| RFC 822 | Standard For The Format of ARPA Internet Text Messages, David H. Crocker, Dept Of EE, University of Delaware, August 1982 |
| RFC 1521 | MIME (Multipurpose Internet Mail Extensions Part One, - Standards Track N. Borenstien, Bellcore, N. Freed, Innosoft, September 1993 |
| Draft-ietf-ediint-req-08.txt | Requirements for Inter-operable Internet EDI - Draft Proposal - T Harding, Cyclone Software, R. Drummond, Drummond Group, C. Shih, Gartner Group, September 1999 |
| Draft-ietf-as1-11.txt | Mime-based Secure EDI - Draft Proposal - T Harding, Cyclone Software, R. Drummond, Drummond Group, C. Shih, Gartner Group, September 1999 |
| Draft-ietf-ediint-as2-06.txt | HTTP Transport for Secure EDI – Draft Proposal – D. Moberg, D. Brooks, R. Drummond, October 1999 |
| RFC 1847 | Security Multiparts for MIME : J. Galvin, S. Murphy, Trusted Information Systems, S. Crocker, CyberCash, Inc., N. Freed, Innosoft International, Inc., October 1995 |
| RFC 2298 | An Extensible Message Format for Message Disposition Notification : R. Fajman, National Institutes of Health, March 1998 |
| RFC 1848 | MIME Object Security Services- Standards Track, October 1995 |
| Software Release 5.01 | Entrust/Toolkit C++ Edition - Programmers Guide/Programmers Reference, Software Release 5.01 |

# 1. Definitions

| | |
|---|---|
| CRLF | "The term CRLF, in this document, refers to the sequence of the two ASCII characters CR (13) and LF (10) which, taken together, in this order, denote a line break in RFC822 mail" [1]. |

# 1. Symbols and Abbreviations

| Abbreviation | Definition |
|---|---|
| HTTP | Hypertext Transfer Protocol |
| IETF | Internet Engineering Task Force |
| MIME | Multipurpose Internet Mail Extensions |
| S/MIME | Secure Multipurpose Internet Mail Extensions |
| PKCS | Public Key Cryptography Standards |
| CCRA | Canada Customs and Revenue Agency |
| RFC | Request for Comment |
| URI | Uniform Resource Identifier |
| MDN | Message Delivery Notification |
| MIC | Message Integrity Check |

| Symbol | Definition |
|---|---|
| | no symbols are defined or referred to in this document |

# 1. General Overview

This document is intended as a technical specification. As such it is assumed that the reader is comfortable in the technical aspects of:

- The Hypertext Transfer Protocol (HTTP) standard,
- The Multipurpose Internet Mail Extensions (MIME) standard, as well as it's secure counterpart (S/MIME),
- X.500 and Lightweight Directory Access Protocol terminology,
- The Public Key Cryptography Standard (PKCS) and,
- And Internet messaging in general.

The reader is also encouraged to obtain and read the standards used and quoted throughout this specification. IETF RFC's and Draft proposals can be obtained at www.ietf.org/home.html, while RSA documentation can be obtained at www.rsasecurity.com/index.html. Entrust related information (including the toolkits and related documentation) can be found at www.entrust.com.

[1] RFC1521 Section 2.

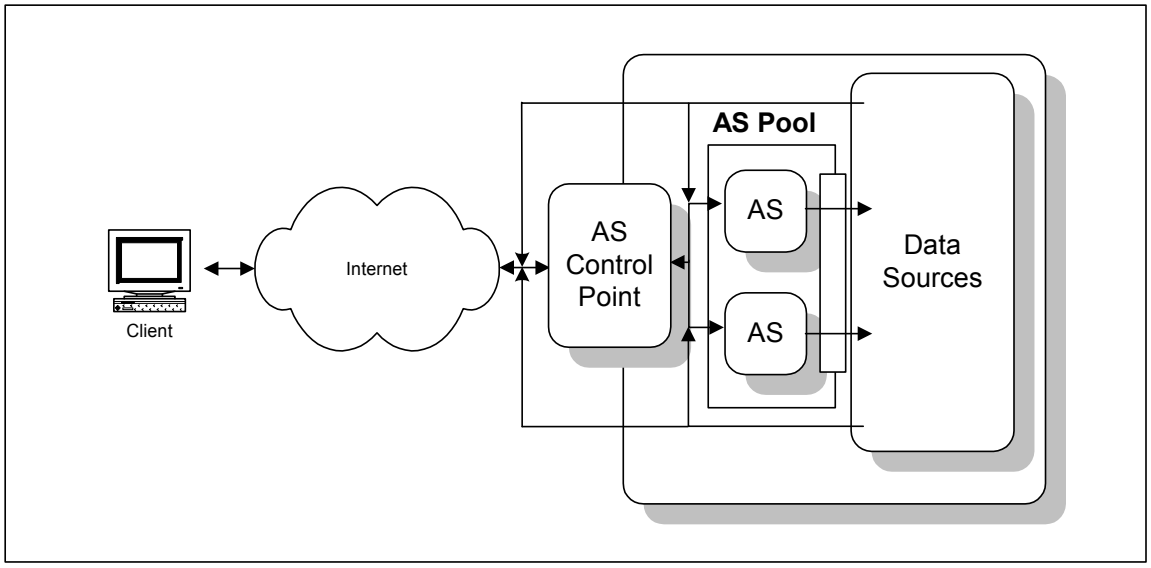
With regard to third party reference to this specification: this specification is not intended to prescribe or impose the internal formats or algorithms, specific system features, and/or characteristics of a user interface used to access, manipulate or create the defined Internet messages.

These specifications are provided on an "AS-IS" basis with no warranty whatsoever expressed or implied.

# 1. Canada Customs and Revenue Agency Environment

## The Three Tier Model

The architecture in which secure file transfers take place between clients and Canada Customs and Revenue Agency is known as the three-tier model. Tier one represents the client's application, whose specifications are contained herein. Tier two represents the application servers (AS), and tier three represents the data source(s). Tiers two and three reside within the Canada Customs and Revenue Agency network infrastructure. The following illustration depicts the three-tier model:




## Tier Two - The Application Server

Computing platforms that represent the second tier are known as Application Servers (AS). These servers are responsible for implementing the business logic of a particular application. The number of application servers in the second tier can be dynamically altered (or pooled) in response to a varying workload. This is accomplished by assigning one Application server to act as a control point, a focal point for client requests. The AS control point (ASCP) in turn routes the request to the AS tasked to reply to the request. Note that reply to the request is not (necessarily) routed back through the control point to the client, resulting in significant performance gains to the requestor.

## Addressing Within Tier Two

Addressing within the second tier is represented as a traditional Internet Protocol host address/data address fashion. An example of this pairing might be:

myServerName /PathTo/Resource

where

| | |
|---|---|
| myServerName | Represents the host address; the destination address of the request. In the illustration above this address would represent the Application Server Control Point. |
| /PathTo/Resource | Represents the data address. Note that in the case of our illustration above, the data address may resolve to the Application Server Control Point, or, the ASCP may determine that another AS within the pool represents this data address and forward the request accordingly. |

### Specific Tier 2 Addresses within CCRA

For the OPS environment:

This section is left blank purposely.

Further information on addressing is contained in Section 7.1, "The Message Header".

### Tier Three - The Data Source

Tier three represents the data source(s) for the application. This may include entities such as databases or messaging systems intended to forward client requests to legacy applications. The responsibility for moving data to and from tier three lies with the tier two entities.

# 1. Lexical Definitions

The following section is intended to familiarize the reader with lexical definitions of the data files that are to be sent to, and will be received from Canada Customs and Revenue Agency. All definitions are based upon international standards (or proposals) of either the International Standards Organization (ISO) or the Internet Engineering Task Force (IETF). As a result, any system that processes data to these specifications can expect to benefit from all of the advantages of "open standards". Wherever appropriate, annotation to the relevant standard will be made.

These standards can also be used to form the basis of communication to other systems that follow the formatting standards for Internet text messaging. Ultimately the network communication described herein conforms to the Internet draft proposal for:

- Inter-operable Internet EDI, described in document "draft-ietf-ediint-req-08.txt";
- MIME-based Secure EDI, described in document "draft-ietf-ediint-as1-11.txt"; and
- HTTP Transport for Secure EDI, described in document "draft-ietf-ediint-as2-06.txt";

Throughout this section an Internet message will be referred to as comprising of two distinct elements, the header and the body. The term message will be used to refer to the entire entity; its header and all of its body parts.

At the highest level, the message header can be defined as comprised of individual header fields that represent information required only once within the message.

The body of the Internet message is formally defined as "simply a sequence of lines containing ASCII characters."[2] This definition will suffice for now, however, as this section progresses, a more sophisticated definition for the message body will emerge to meet the specification's requirements.

The remainder of this section further defines the message header and body.

[2] RFC 822, Section 3.1

## The Message Header

Within the scope of this specification, RFC's 822 and 1945 define the message header. An Internet message is illustrated and defined below:

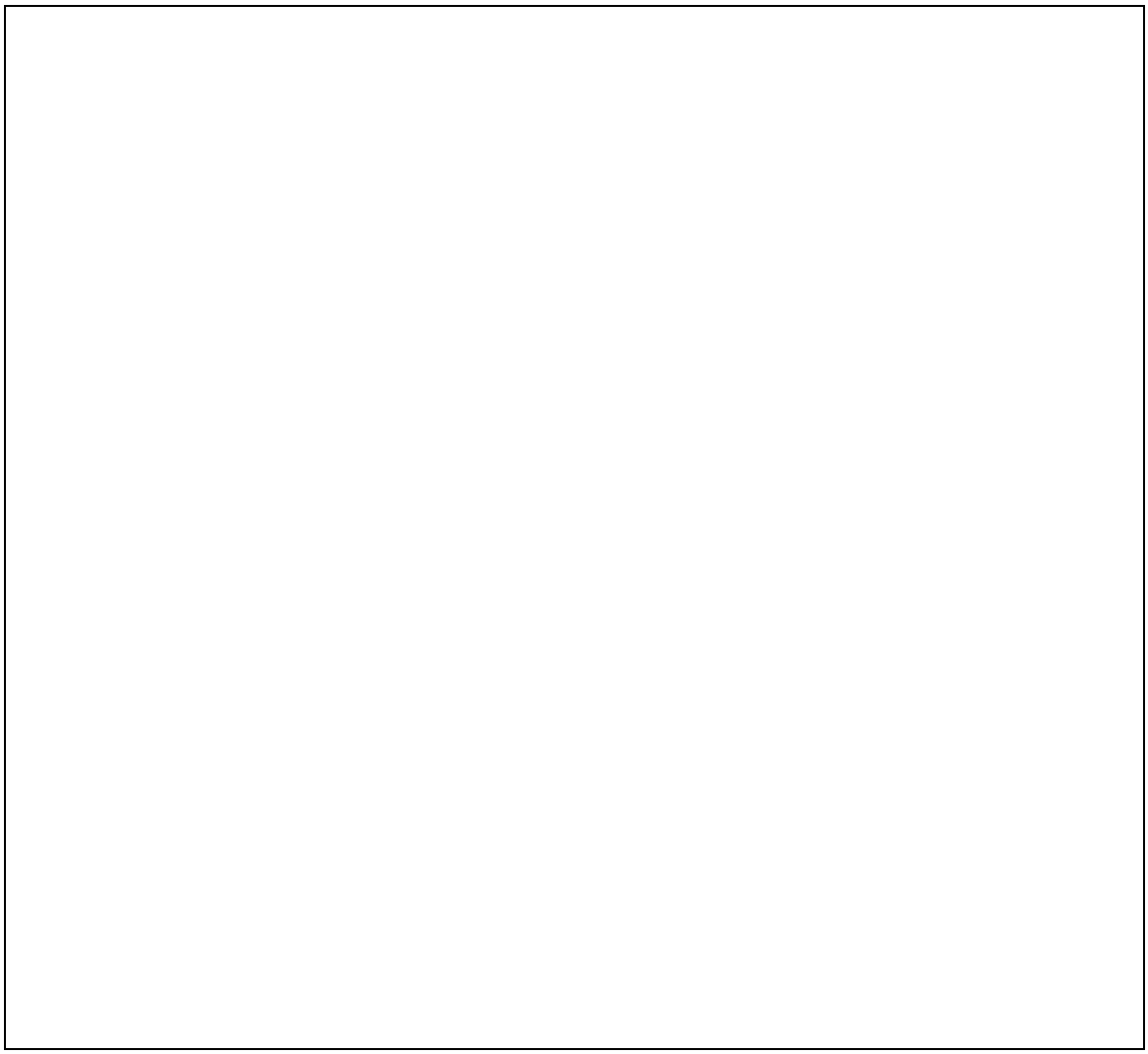
The EDIINT Message Header.

| Request Header Field Name | Sub Fields | Description | Reference |
|---|---|---|---|
| HTTP Command[3] | POST | Indicates to the destination server that the entire message body should be passed as data to the data address indicated. | RFC1945 Section 5 / 8.3 |
| | Data address | The URI of a resource residing within Canada Customs and Revenue Agency. This is a required address provided to the client by Canada Customs and Revenue Agency. | RFC1945 Section 5.1.2 |
| | HTTP Version | Currently set to "HTTP/1.0" | RFC1945 Section 5.1 |
| From | None | Contains an Internet email address representing the client's system administrator. | RFC1945 Section 10.8 |
| Message-ID | None | A unique sequence number associated with the Interchange | |
| Mime-Version | None | Currently set to "1.0" | RFC1945 Section D.2.7 |
| Date | None | The date/time stamp of Message generation. This must be expressed in Greenwich Mean Time (GMT) | RFC1945 Section 10.6 |
| Accept | None | Used to specify the media types acceptable for the response. At a minimum this must be defined as "application/pkcs7-mime; text/plain" | RFC1945, Section D.2.1 |
| Content-Type | | Used to specify the media type of the body being sent to the recipient. This must be set to "application/pkcs7-mime". | Draft-ietf-ediint-req-08 |
| Disposition-Notification-to | None | See Section 7.7 "Message Delivery Notification and the Secure Message Loop" | Draft-ietf-ediint-as1-11 |
| Disposition-Notification-Options | None | See Section 7.7 "Message Delivery Notification and the Secure Message Loop". | Draft-ietf-ediint-as1-11 |
| Receipt-Delivery-Option | None | See Section 7.7 "Message Delivery Notification and the Secure Message Loop". | Draft-ietf-ediint-as2-06 |
| Content-Length | None | Specifies the length of the entire body in decimal. | RFC1945 Section 10.4 |

[3] whose subfields are delimited by whitespace (0x20).

| Request Header Field Name | Sub Fields | Description | Reference |
|---|---|---|---|
| “null” line | None | a CRLF pair indicating the end of the message header. | RFC822 Section 3.1 |

An example follows:



Notes:

- With the exception of the HTTP command header field, all header fields delimit the header field name with its value using the colon character. For example:

  Content-Type: application/pkcs7-mime;

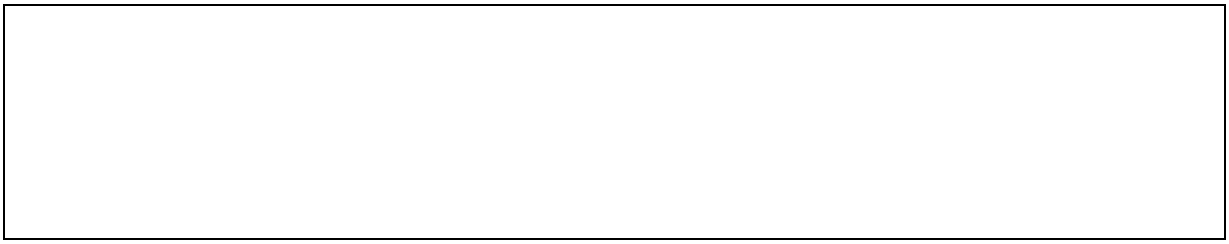
- In the example above, all entries within the message header are terminated with a CRLF pair.

### The Message Body

Within the scope of this specification, RFC's 822, 1521 and the EDI-INT draft define the Internet message body as well as the granularity of definition. In other words, RFC 822 defines the Internet message at its highest level (header only) whereas the EDI-INT Draft defines the specifics of messages with content-types "multipart/signed".

In RFC 1521, the specification of a multipart content-type field in a message header states that the message's body will contain one or more body parts, each of which will be proceeded by an encapsulating boundary (which itself is preceded and followed by a CRLF pair). Additionally, the last body part will be followed by a closing boundary. A message body boundary is further explained in later sections.

A body entry itself (after the encapsulating boundary) consists of a header area, a blank line and a "body" area. Thus the MIME body is semantically similar to the message header format described in RFC 822. An example of the body entry:



The EDI-INT draft further defines that each body entry defined by media type "multipart/signed" is expected to contain two body parts, the first having a content-type of "application/<EDI standard>, the second a content-type of "application/pkcs7-signature".

Notes:

- All header fields delimit the header field name with its value using the colon character. For example:

  Content-Type: application/edi-consent;

- Parameter definitions are delimited using the semi-colon character (";")

- In the example above, all entries within the message header are terminated with a CRLF pair.

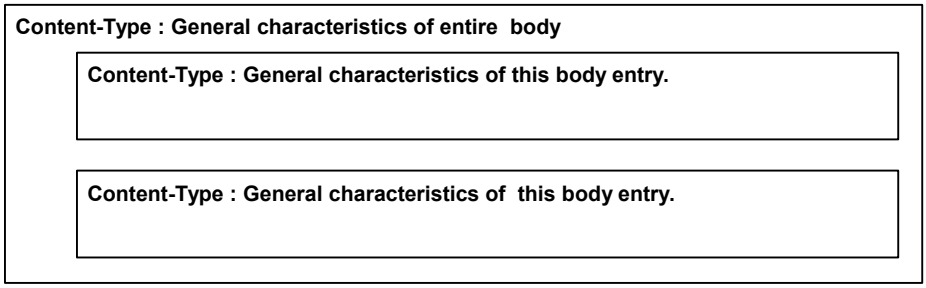
| | |
|---|---|
| Warning to Implementers | Throughout this last section, a great deal of data has been depicted that may be thought of as being static in nature. Be aware that this is ***not*** the case. HTTP headers and their contents may change over the course of time due to system changes or to increase efficiency. Therefore, the reader is advised to implement the preceding data areas in such a manner that they can be dynamically changed (i.e. without program recompilation). |

## The Multipart Content-Type

Up to this point in the document, the Internet message has been defined as being composed of two parts; the header and the body. However in order to meet the specification requirement of transporting encrypted files within an internet message, a more sophisticated message body must be implemented.

This refined message body is known as a multipart message body and is defined as part of the Multipurpose Internet Mail Specification or MIME specification. The MIME specification is found in the IETF's RFC 1521[4]. Subsequent RFCs [5] define the secure variant of the MIME.

However, to be precise, the RFC1521 specification (beginning in Section 7.2), in a case where multiple body parts are to be carried within a single message, states that the "Multipart" content type must appear as request header field in the message's header. The initial Content-type header defines the general characteristics of the following body parts. These body parts in turn can define their own content types. For example:




A content-type request header field can also further define a sub-field known as the "boundary" field.

It is the boundary field that is used to delimit the individual body parts within the Internet message. In use, the boundary field is known as the "encapsulation boundary". This is due to the fact that prior to use it is altered from its original definition state. The final body part is terminated with a "closing boundary", again, an alteration of the original encapsulation boundary. An example follows:

[4] One of several RFCs defining MIME, e.g. 2045 to 2049.
[5] RFC 2630, 2633 Secure/Mime V3 specification, RFC1847 Security Multiparts for MIME

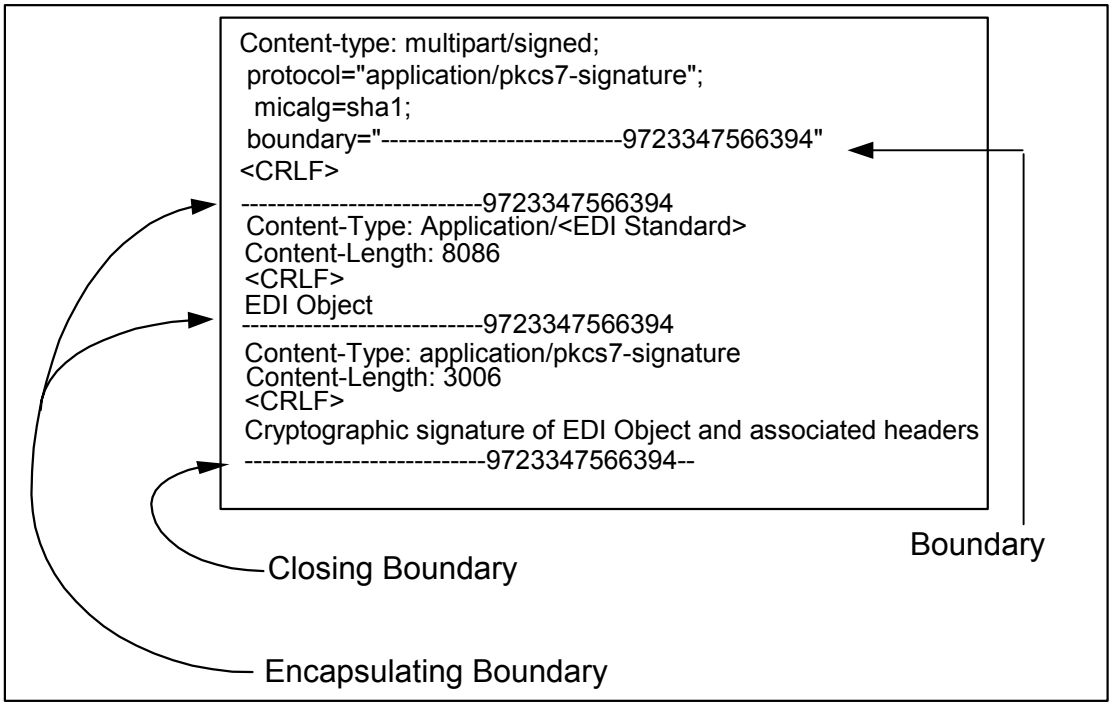



Figure 1 Multipart Content Type Semantics

The **boundary** is defined as sub-field, or parameter, of the Content-type header request field. From the standard's perspective the boundary is simply a random string of characters to be used to "index" the body parts. The standard's only mandatory requirement of the boundary is that the boundary string does not appear within the body it delimits.

In the example above the boundary definition is enclosed in double quotes. Although the specification does not require this grammar in the boundary definition, it does recommend the use of double quotes to avoid illegal Content-Type definitions. In fact, for reasons that become clear in later paragraphs, within this specification *the enclosure of the boundary definition within double quotes is considered **mandatory***.

As per RFC1521, the boundary cannot be greater than 70 characters in length.

The **encapsulating boundary** is defined as line composed of two hyphens ("-" or decimal 45) followed by the boundary line.

"Note that the encapsulation boundary must occur at the beginning of a line, i.e., following a CRLF, and that the initial CRLF is considered to be attached to the encapsulation boundary rather than part of the preceding part. The boundary must be followed immediately either by another CRLF and the header fields for the next part, or by two CRLFs, in which case there are no header fields for the next part (and it is therefore assumed to be of Content-Type text/plain).

NOTE: The CRLF preceding the encapsulation line is conceptually attached to the boundary so that it is possible to have a part that does not end with a CRLF (line break). Body parts that must be considered to end with line breaks, therefore, must have two CRLFs preceding the encapsulation line, the first of which is part of the preceding body part, and the second of which is part of the encapsulation boundary."[6]

[6] RFC1521, Section 7.2.1.

The **closing boundary** also known as the distinguished delimiter, is defined as a line composed of the encapsulating boundary followed by two hyphens ("-" or decimal 45). The closing boundary indicates that no further body parts follow.

### EDI-INT's S/MIME : Signature and Encryption (Multipart/Signed)

Building on the general multi-part MIME message definition above, the EDI-INT proposal defines its multi-part message as follows:

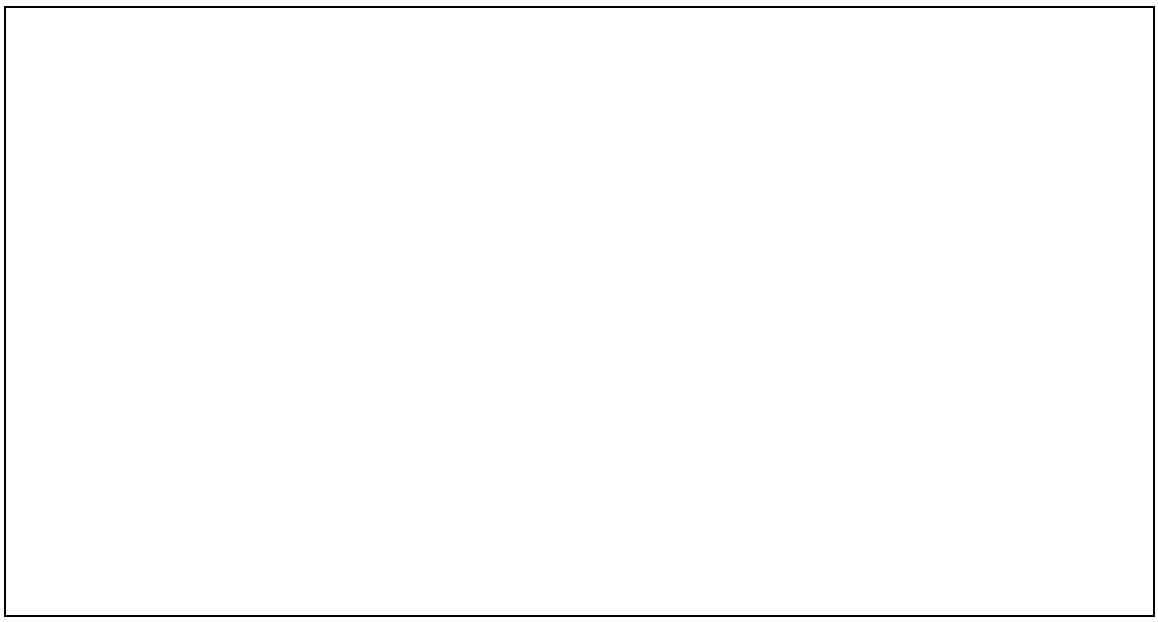


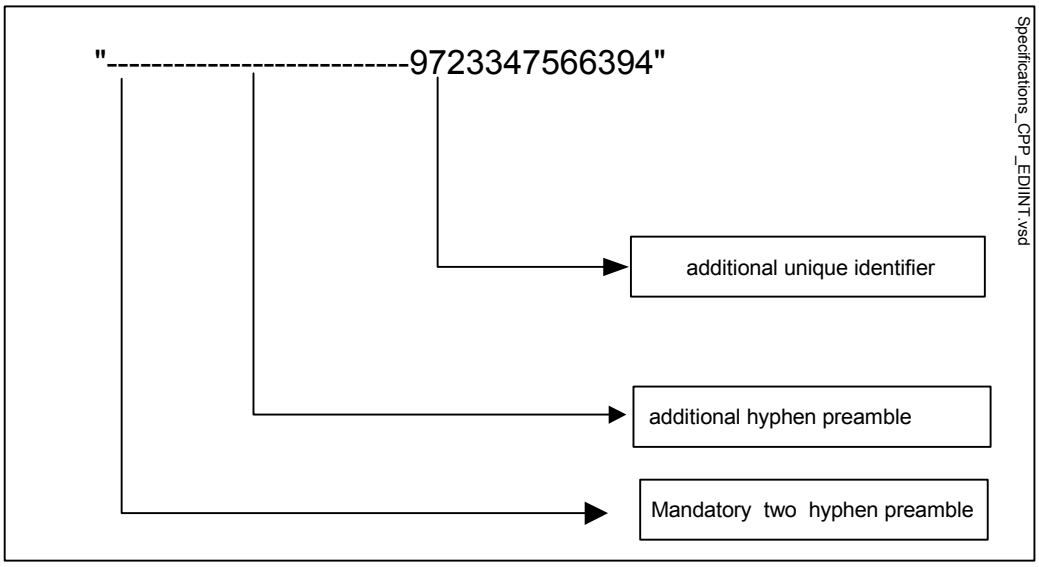
Figure 2 EDI-INT Message Body.

The previous illustration depicts three distinct body parts (content-types);

| Content-Type | Sub Field(s) | Description |
|---|---|---|
| Multipart/signed | | Describes the attributes applying to the entire message body. |
| | Protocol | Defined as "application/pkcs7-signature". Although the EDI-INT draft also defines other protocols that can be used, for CCRA communication "application/pkcs7-signature" is mandatory. |
| | Micalg | Algorithm used to calculate the MIC (Message Integrity Check) value. This value must be set to "**sha1**". |
| | Boundary | Boundary to be used to delimit body parts. |
| Application/<EDI Standard> | None applicable | Defines the body part containing the EDI object being transported. Expected values for <EDI Standard> are either "**edi-edifact** "or "**edi-consent**". The value "edi-edifact" is to be used to identify that the transaction within the body is in EDIFACT format. The value "edi-consent" is to be used to identify that the transaction within the body is in CADEX proprietary format. |
| Application/pkcs7-signature | None applicable | Defines the body part containing the digital signature of the EDI object being transported. |

## Contents of the Boundary

Referring back to figure 2, although it would appear that a random character string is used within the boundary, this is not the case. Specifically the boundary is broken down as follows:




In our example above the unique identifier is the long integer representation of the time and date stamp. Although this is a recommended convention to follow a particular implementation may require further uniqueness perhaps including workstation-id, process-id and/or thread-id.

## Securing the EDI interchange message.

The EDI-INT draft proposal secures the EDI interchange using the signature/envelope technique. This technique is assembled as follows:

a) A multipart/signed MIME message is created as per RFC 1847 indicating the appropriate protocol, micalg and boundary.

b) The first body part is assembled containing the EDI Interchange object and a Content-Type of “Application/<EDI Standard>”. **NOTE:** in the case of CADEX, the EDI object is simply a transaction that is being sent today to Canada Customs and Revenue Agency. As well see Section 7.4 for acceptable <EDI Standard> values.

b) The digital signature of the body part containing the EDI Interchange object is calculated. **This includes the body part’s MIME headers. ****

c) The digital signature becomes the data payload of the second (required) body part for the message bearing the Content-Type of “application/pkcs7-signature.

d) The above two body parts (EDI Object and Signature) form the basis for the higher-level body part having a Content-Type of “multipart/signed” (defined in a, above). Additionally this header defines a protocol of type “application /pkcs7-signature”, a MIC algorithm definition and a boundary definition.

e) The above body parts are encrypted and appended to the Internet Message Header.

| | |
|---|---|
| ** Warning to Implementers | As defined by RFC 1848 Section 2.1, Digital Signature Service, Canonicalization of a bodypart must be performed **BEFORE** either signing or verifying the body part. Simply stated, this means that end-of-line delimiters must be represented as a <CR><LF> pair before the data is signed or verified. This applies to the bodypart headers as well as the bodypart contents.<br><br>"… The application of the digital signature service requires that the same line delimiter be used by both the originator and the recipient. This document specifies that the two character sequence "<CR><LF>" must be used as the line delimiter. …"<br><br>Implementers should not that in many operating system and programming languages, the internal representation of end-of-line may not be <CR><LF>. Thus, some conversion may be required before application or verification of the digital signature |

The following figure illustrates a complete EDI interchange/Internet message.

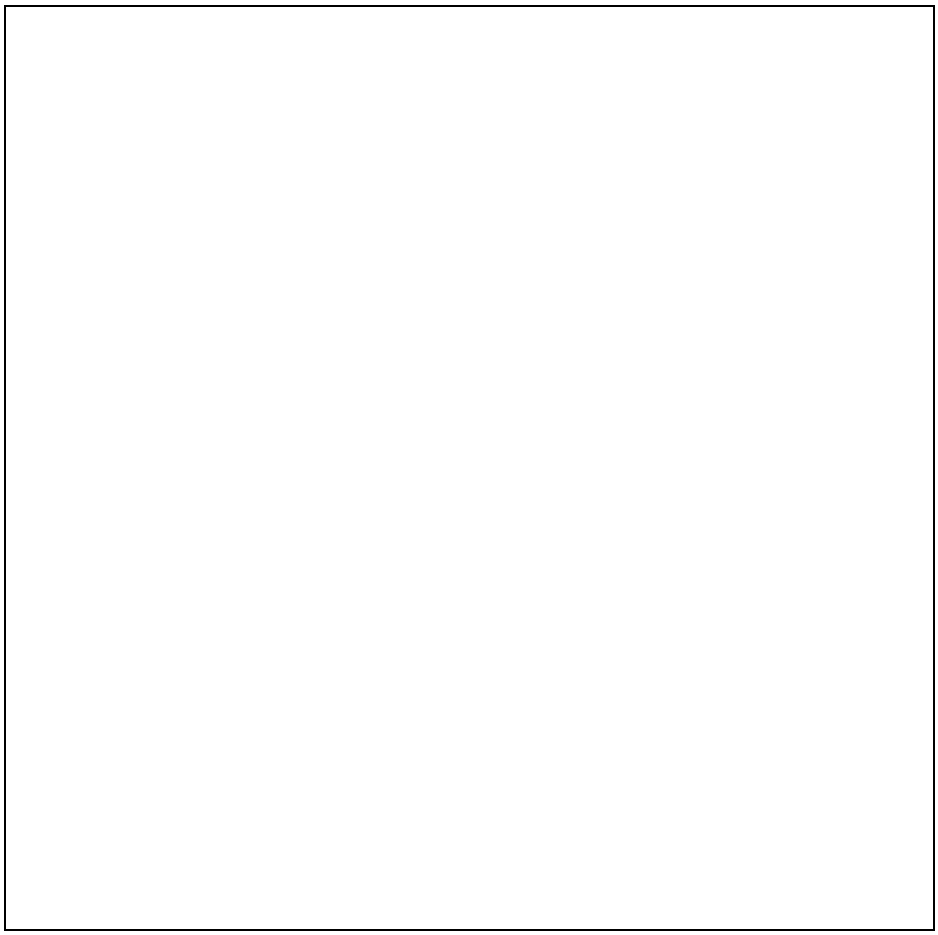


Figure 3 The EDIINT Internet Message

**Message Delivery Notification and the Secure Message Loop.**

Internet draft document "draft-ediint-as1-11.txt" expands upon RFC 2298, Message Delivery Notification (MDN), as it applies to EDI Interchange over the Internet as well as the concept of the "secure transmission loop". To quote this document:

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

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

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

-The receiving organization then returns a signed receipt to

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

Clearly, the secure message loop (as well as the MDN) plays an integral role in the EDIINT draft proposal and the paragraph above summarizes its operation precisely. So precisely, in fact, the paragraph serves a perfect springboard for the analysis of the secure message loop.

"The organization sending EDI/EC data signs and encrypts the data using either PGP/MIME or S/MIME."

Refers to the S/MIME based Internet message described in the previous sections of this document and illustrated in its entirety in Figure 3.

"In addition, the message will request a signed receipt to be returned to the sender of the message."

A signed receipt is requested by including the following three request headers in the HTTP Message header:

Disposition-notification-to: <YourAdministrator@YourCompany.com>
Disposition-notification-options :
signed-receipt-protocol=required, pkcs7-signature;
signed-receipt-micalg=required, sha1
Receipt-delivery-option: http://ciguat.ccra-adrc.gc.ca/CigWasOp/CigWas.CigWasGet0

The "draft-ietf-ediint-as1-11.txt" further defines these headers and their associated options.

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

Specifically,

1) the message body is decrypted by the recipient revealing a multipart/signed MIME message containing two body parts.
2) the recipient authenticates the signature contained in the second body part by:
    - decrypting the sent MIC using the senders public key.
    - a MIC is calculated on the first body part as per RFC 1767.
    - the two MICs are compared for equality.

If equal, the signature has been verified.

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

Specifically,

1) the recipient formats an MDN and sets the "Received-content-MIC" to the value of the calculated MIC.
2) the recipient creates a multipart/signed MIME message as per RFC 1847
3) the MDN becomes the first body part of the multipart/signed MIME message.
4) the second body part contains the digital signature of the first body part *including all of it associated headers.*

The following illustration depicts the Message Delivery Notification MIME message.

Of particular interest is the location of the Received-content-MIC and the Disposition fields.

Contents of the Received-content-MIC is explained above.

A Disposition of "**automatic-action/MDN-sent-automatically; processed**" indicates the MDN process was an "automatic-action", that the user did not give explicit permission for the MDN to be generated, it was "sent-automatically" and message being referenced was " processed" successfully.

Failure events can occur at any given stage of a messages' receipt processing. These events will result in the following dispositions being issued.

| Event | Disposition |
|---|---|
| Content Processing Errors | automatic-action/MDN-sent-automatically; processed/ Error: decryption-failed<br>automatic-action/MDN-sent-automatically; processed/Error: authentication-failed<br>automatic-action/MDN-sent-automatically; processed/Error: integrity-check-failed<br>automatic-action/MDN-sent-automatically; processed /Error: unexpected-processing-error |

An example MDN follows:

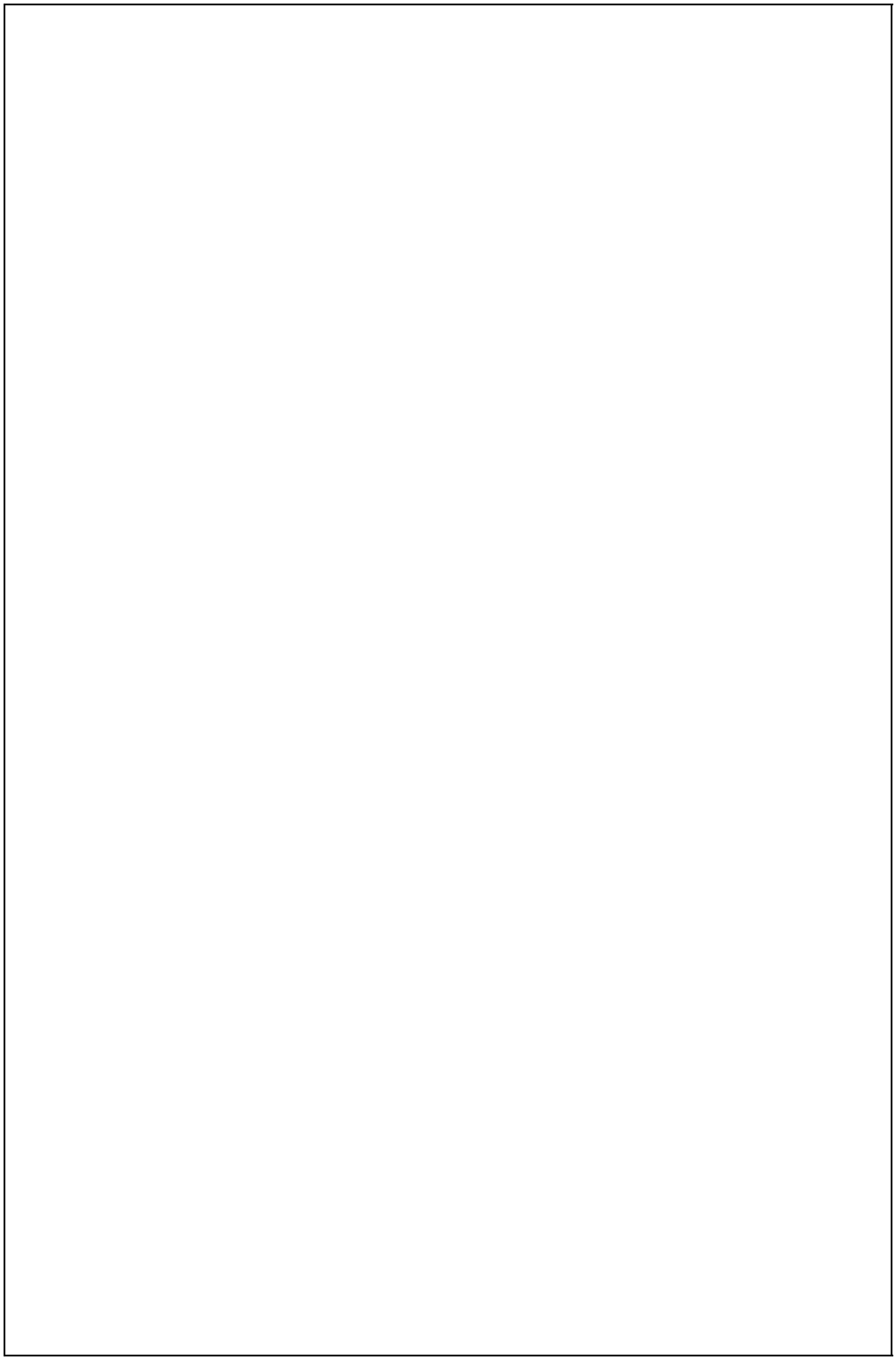


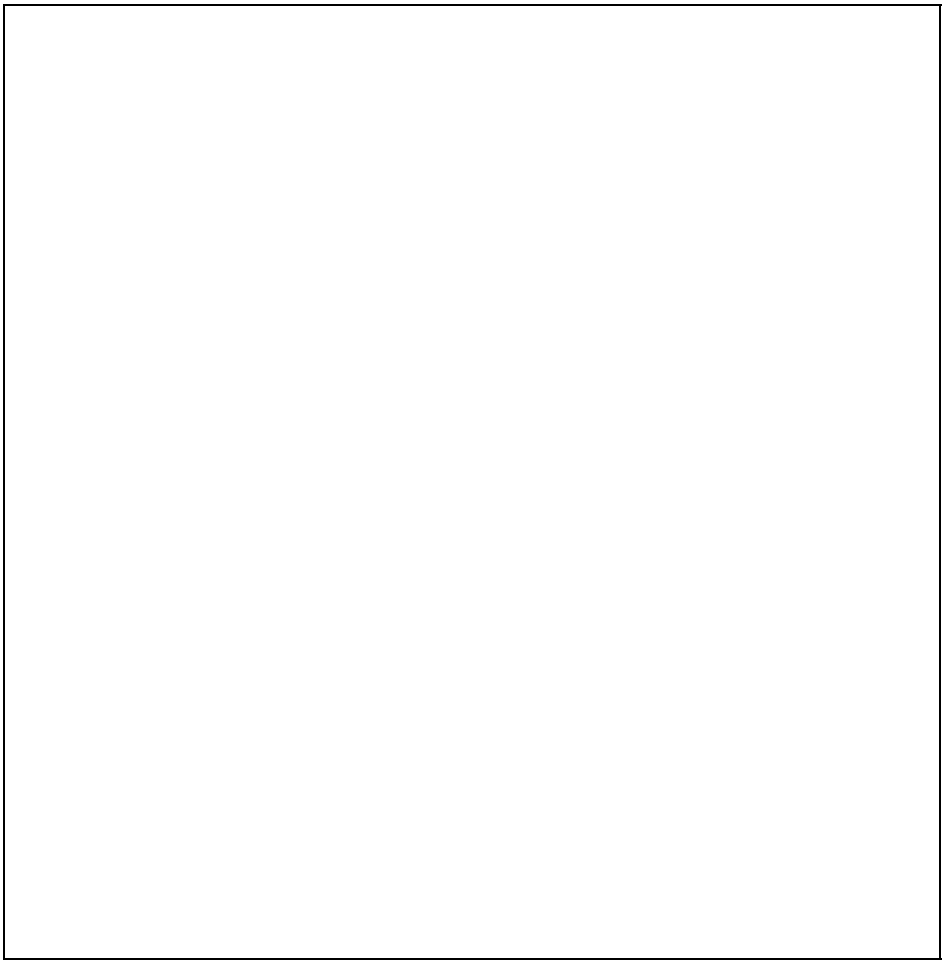
Figure 4 The Message Delivery Notification Message

Further information regarding EDI object content processing errors and /or warnings reflected in the disposition field can be found in documents “MIME-based Secure EDI”, “An Extensible Message Format for Message Disposition Notification” as well as the “draft-ietf-ediint-as1-11.txt” document, section 5.3.

An MDN will be issued for each EDI message that is sent to CCRA for processing. CCRA will also request MDNs when it sends EDI transactions to its trading partners. Trading partners should use the same general format when returning MDN’s to CCRA. Note that, as per the specification (RFC 2298, Section 2.1), MDNs are ***not*** acknowledged with MDNs.

## Requesting A Download of Transaction Responses.

In order to receive any transaction response(s) that may be queued at the host, a particular Internet message must be sent indicating the data address of the service responsible for downloading responses. An example of such a message follows:



Note that this message is simply a variant of the “Content-Type: Application/pkcs7-mime” message used for EDI object submission. Where the message differs from an EDI object submission message is :

| | |
|---|---|
| Data Address | The URI of the resource tasked with processing the Response Request will differ from the resource responsible for processing EDI object submissions. |
| Response Request | The Response Request Object *may* be any valid Edifact or Cadex formatted transaction (Note: the transaction will not be processed as an EDI transaction however. It is simply used as a means of requesting a download). |

As a result of the above submission, the CCRA transaction download service, named CigWasGet0 and located in CigWasOp/CigWas, will:

a) validate the sender of request;

b) and upon successful validation send to the requestor all transactions destined for the requesting client broker.

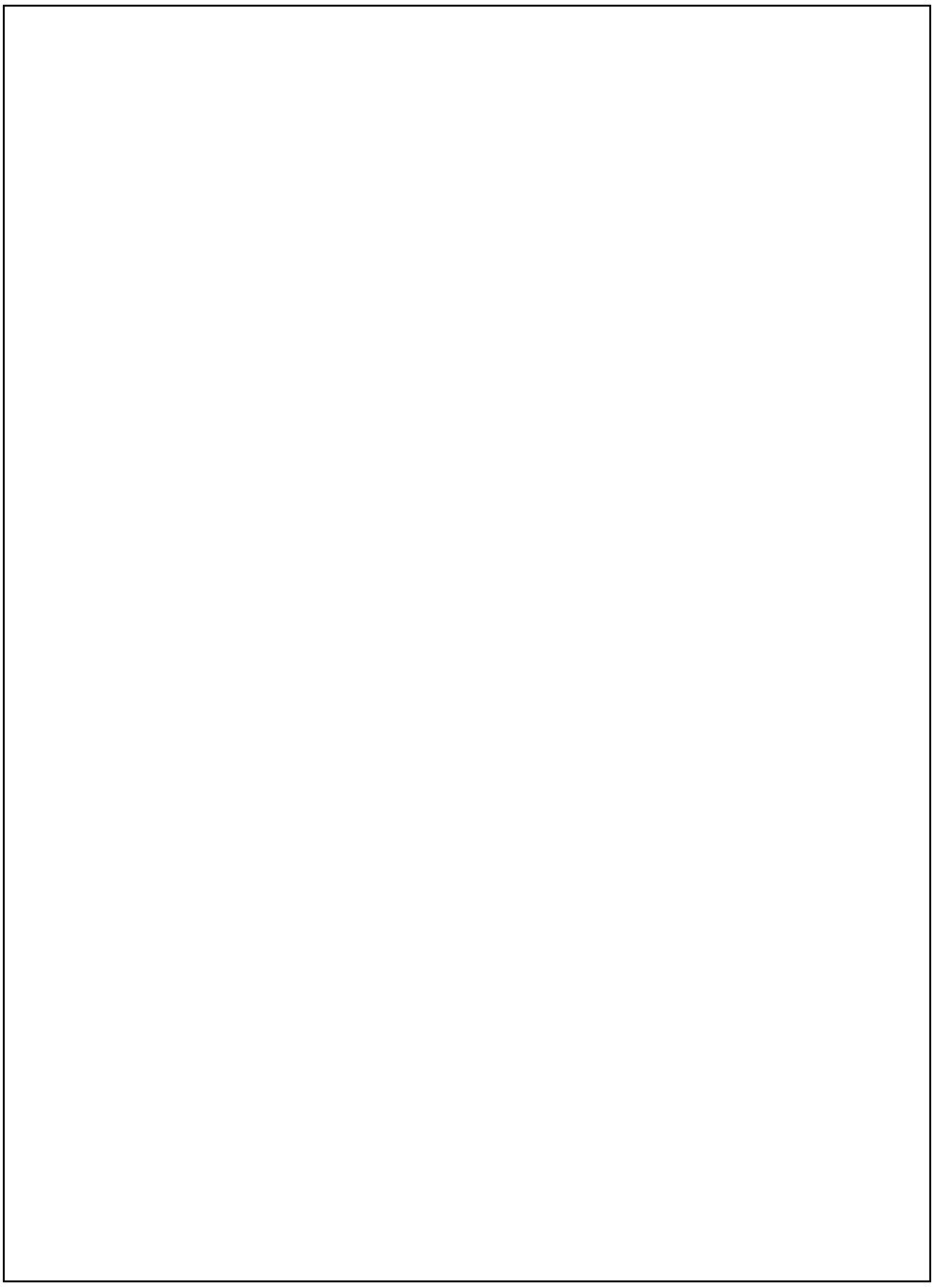
Also note that a "request to download" transaction must ***not*** contain a request for MDN response, i.e. the Disposition-Notification-To:, Disposition-Notification-Options: and Receipt-Delivery-Options: headers are omitted

## The Response Message.

Once the Trading Partner has requested a download using the technique described above, all EDI and MDN transactions awaiting delivery will be prepared and formatted for transmission to the Trading Partner.

A multipart format is used so that individual transactions can be distinguished from one another.

An illustrative example of a typical response message is illustrated below.



The shaded areas represent EDI and MDN transactions.

| Response Header Field Name | Sub Fields | Description | Reference |
|---|---|---|---|
| HTTP Response[7] | HTTP Version | Currently set to "HTTP/1.0". | RFC1945 Section 3.1 |
| | Status Code | See Section 7.10 of this document. | RFC1945. Section 9 |
| | Reason Phrase | See Section 7.10 of this document. | RFC1945 Section 9 |
| Date | None | The Date/Time at which the message was originated. | RFC1945 Section 10.6 |
| Server | None | Contains information about the software used by the origin server to handle the request. | RFC1945 Section 10.14 |
| Pragma | None | Used to indicate to intermediary Proxy Servers that the response should not be cached. | RFC1945 Section 10.12 |
| Cache-Control | None | Used to indicate to intermediary Proxy Servers that the response should not be cached. *This is included for compliancy purposes.* | RFC2616 Section 14.9.1 |
| Expires | None | Used to indicate to intermediary Proxy Servers that the response should not be cached. | RFC1945 Section 10.7 |
| Connection | None | Allows the sender to request options that will apply to the particular connection. Currently set to "close". | |
| Mime-Version | None | Currently set to "1.0". | |
| Content-Type | | Used to specify the media type of the body being sent to the recipient. This is set to "multipart/mixed". | Draft-ietf-ediint-as2-06 |
| Boundary | None | Boundary to be used to delimit body parts. | RFC1521 |
| Content-Length | None | Specifies the length of the entire body in decimal. | RFC1945Section 10.4 |
| MDN Response Fields | | See Section 7.1 of this document. | |
| EDI Response Fields | | See Section 7.1 of this document. | |

[7] whose subfields are delimited by whitespace (0x20).

## Response Codes

The HTTP 1.0 specification (RFC1945) defines the status codes that can be delivered to a client in response to a request. In general these status codes are categorized as :

1xx - Informational codes,

2xx - Successful codes,

3xx - Redirection codes,

4xx - Client Error codes and,

5xx - Server Error codes.

The following section summarizes the HTTP response codes the client can expect to receive from the Canada Customs and Revenue Agency server and their meanings as they apply to the current server implementation[8]. Note that these code meanings may change and/or may be added to in future implementations.

| Status Code | Meaning |
|---|---|
| 200 | **OK** ; the request has succeeded.<br>In the context of sending a transaction this indicates the transaction was received, validated successfully and has been queued for execution.<br>In the context of receiving a transaction this code indicates that the users request was received, validated successfully and is returning with responses that have been queued at the host. |
| 204 | **No Content**; in response to a download request, this status indicates that the request was received and processed successfully, however there were no responses to be sent at this time. |
| 301 | **Moved Permanently**; indicating that the resource has permanently moved to a new location, and that future references should use a new URI (CCRA data address) with their requests. Contact Canada Customs and Revenue Agency for the new address. |
| 400 | **Bad Request**; indicating the request sent by the client was syntactically incorrect. Further, specific information relating to the error could not be determined. |
| 403 | **Forbidden**; indicating the server understood the request but refused to fulfill it. This error indicates the client's environment profile did not include access to the requested resource. |
| 404 | **Not Found**; indicating that:<br>1. the requested resource is not available,<br>2. the data address provided was malformed or unknown. |
| 500 | The HTTP server is unavailable. |
| 503 | **Service Unavailable**; indicating that the HTTP |

[8] in other words, whereas these codes have generic meaning within the specification, this section describes, in addition to their generic meaning, their specific meaning within RC's server implementation.

server or one of it's components is *temporarily* unable to handle the request.

All other return codes should be interpreted as error unless otherwise indicated in subsequent CCRA publications and/or bulletins.

| | |
|---|---|
| Warning to Implementers | Throughout this last section, a great deal of data has been depicted that may be thought of as being static in nature. Be aware that this is ***not*** the case. HTTP status codes or their contents may change over the course of time due to system changes or to increase efficiency. Therefore, the implementers are advised to implement the preceding data areas in such a manner that they can be dynamically changed (i.e. without programme recompilation). |

An example of a "no data queued" response from the server follows:

```
HTTP/1.0 204 ok
Date: Mon, 05 Jun 2000 19:04:40 GMT
Server: Server Specific information
Pragma: No-cache
Cache-Control: no-cache
Expires: Thu, 01 Jan 1970 00:00:00 GMT
Connection: close
Content-Type: text/plain
```

# 1. Encryption Overview

To protect a message, an originator transforms a message (called plaintext) into a human unrecognizable form, called ciphertext. This process is known as *encryption*. The ciphertext is then transmitted to the receiver by whatever means is preferred. Upon receipt, the message recipient converts the ciphertext into its original format. This is known as decryption.

Should the message be intercepted, the intruder only has access to the unintelligible ciphertext. A good encryption scheme makes it difficult for the intruder to re-compute the original plaintext, even if they know the process used to encrypt the plaintext.

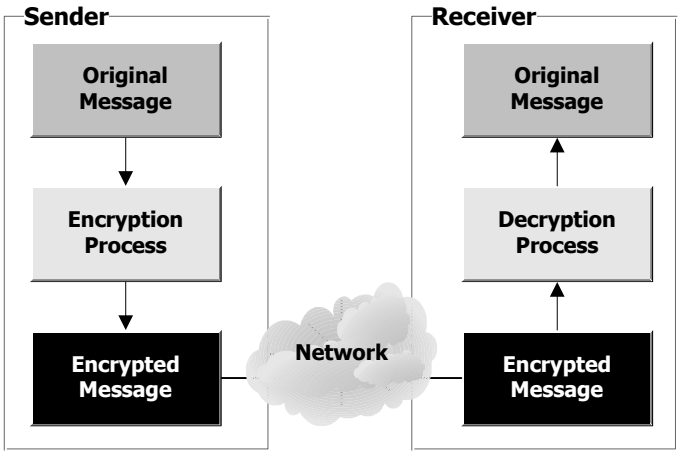



Figure 5 - The Encryption/Decryption Process

Before sending an encrypted message, the originator must have two pieces of information that the receiver must agree to:

- The method (or algorithm) that is to be used to convert the original text into ciphertext.
- The mathematical value (or key) to be used to do the transformation.

Once these have been agreed upon, then the originator can proceed with the encryption. The target receives the ciphertext, decrypts it, and interprets the message as normal.

### Single Key Encryption

The earliest form of encryption used a single key to determine the ciphertext, and to re-determine the original plaintext. This encryption method is known as symmetric encryption. The problem with this form of encryption is that it makes it imperative that the key remains secret. If the key becomes known to outside parties, then the message cannot be trusted. The issue would then be how to agree upon a new key. The parties would need to agree to a new key, but the exchange of this information would also need to be done in private.

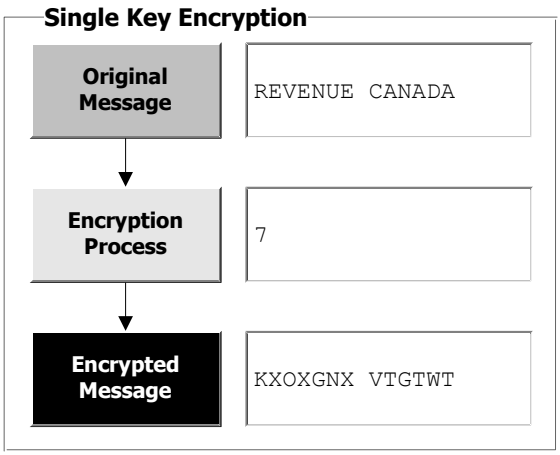



Figure 6 - Single Key Encryption

Consider the diagram shown in Figure 5 in the following example. There is a need for a message to be sent across an insecure, or public network such as the Internet. To make it more difficult for an intruder to determine the contents of this message, the sender and receiver decide to add a random value to each byte of the message. The resulting ciphertext looks nothing like the original message, the key is random (any number would do), and the process can be reversed to determine the original message (just subtract the key). The drawback of this example is that if an intruder knows the algorithm used, then all they would need to do is run the ciphertext through all 10 possible keys, and they would eventually arrive at the proper original message. The algorithms used by Entrust are much more robust, and use keys of much greater length (currently they are up to 2048 bits). If an intruder wanted to break a key of 2048 bits in length, and had access to a computer that could check 1 million possible keys in a second, it would still take them $1.0248 \cdot 10^{604}$ years to check every possible key. A billion computers that could check a billion keys per second would still require $1.0248 \cdot 10^{592}$ years.

## Public Key Encryption

Public key encryption, also known as asymmetric encryption, involves using a pair of keys to transmit information. Every user has two keys, a public key and a private key. The public key is made available in a direct way to anyone who wishes to encrypt information to the user, who will use his or her private key to decrypt the package. The process is such that the sender is assured that only the person with the correct private key can decrypt the message, any other key will only produce unreadable text.

## Digital Signatures

A digital signature is an electronic means of validating the integrity of a given piece of data. For example, suppose someone received a document electronically. How does the receiver know if some third party corrupted the file? Even if encryption was used, the receiver needs to ensure that the file was not changed by any means. This is especially important when dealing with electronic commerce applications.

The integrity issue is dealt with by creating an electronic signature of the data. This is often referred to as a hash or digest of the original plaintext. A hash function takes as input data of any length, and produces a hash or digest (hence the term) of a finite length, usually 128 or 160 bits in length. This hash can represent the larger data if it has the following properties:

- *Consistent*
  The same input files will always product the same output (i.e. hash).

- *Unpredictable*
  Given a particular hash, it will be practically impossible to reverse the hash process and produce the original message.

- *Volatile*

This may seem like a contradiction with the first property, but it is essential that a slight change in the input message will produce a drastic change in the hash. This reduces the possibility of a change in one bit of the data being ignored by the hash function and producing the same hash.

An example of a hash is shown in the table below. Notice how the original input messages are a) quite different as a result of hashing, and b) how even a slight change in the input message produces a quite different hash. These hashes were generated using the *Secure Hash Algorithm*, as defined by NIST, the US National Institute of Standards and Technology (the de facto Hashing algorithm standard in the computing industry):

| **Message** | **Message Digest** (using base16 to represent the bits) |
|---|---|
| Today is February 01, 1998 | ADE4F5EE 81BCC565 6F225414 CA9615DD E09D1754$_{16}$ |
| Today is February 02, 1998 | F6181A1E BBBCD329 60B9CF9A 138649A3 5E4F138C$_{16}$ |

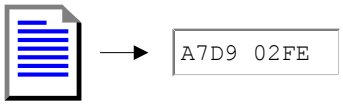
## Entrust Implementation

The Entrust/PKI solution uses all of the above processes to provide secure data communications. There are several infrastructure components that are required to make a public key infrastructure work. First, each user needs a means of sharing their public keys with the public. In Canada Customs and Revenue Agency, a certificate authority and directory are available for this purpose. Each user has four keys:

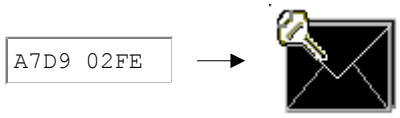
- A private and public encryption key
- A private and public signing key

The following illustrations explain how Entrust uses these keys to secure electronic data. The assumption made in these diagrams is that both the sender and receiver have access to each other's public keys, and trust the source that provided them.

1. *Signing the Data*
   The first part in the process is to create a hash of the data to be sent. This is also known as Message Integrity Check or MIC. Using the appropriate hashing algorithm, a unique identifier is created, which will be used to (in effect) sign the message. (Note: the convention for representing a hash is to use hexadecimal format).




2. *Encrypting the Digest*
   Now the information that will be used to validate the original data needs to be protected before being transmitted. For this, the sender will use their private signing key to encrypt the hash.

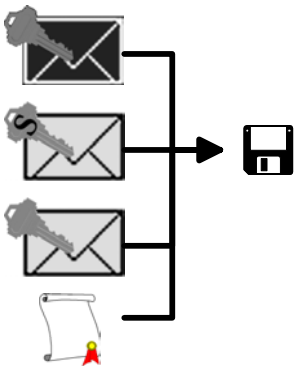



3. *Encrypting the Data*
   Now the data can be encrypted. For this, the Entrust system will create a new unique one-time symmetric key, and encrypt the original data with it. A symmetric encryption process is used since it is generally much

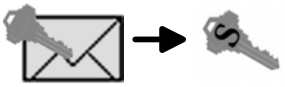
faster than an asymmetric algorithm.

4. *Encrypting the Unique Key*
   Now that the data is secure, the unique key needs to be sent to the receiver, so that they can decrypt the message. For that, the sender will locate the receiver's public encryption key, and use that to encrypt the key.

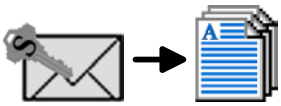
5. *Deliver the Encrypted Information*
   The final piece to be included is the sender's public certificate information. This will tell the receiver where the sender's public information can be found. Then the data is sent over to the receiver, using whatever medium is desired. This can be done using several methods, including diskette, email, FTP, HTTP etc. Please note that this is all contained in one file, and not three, as the diagram might suggest. If the original file is called Document.doc, then the encrypted file would be named `Document.ent`. One benefit of using Entrust is that it can compress the encrypted data, which generally results in the encrypted information being smaller that the original format. This results in lowering the transfer time and reducing bandwidth.



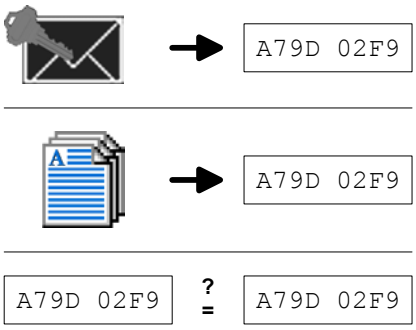
6. *Decrypting the Unique Key*
   Now the receiver needs to decipher the message. The first task is to determine the unique key used by the sender to encrypt the data. For this, the receiver will get his or her own private encryption key.



7. *Decrypting the data*
   Using this symmetric key, the receiver will decrypt the data. It is now readable, but the receiver is not completely sure the data can be trusted.



8. *Validating the Data*
   To guarantee the sender of the data, and the data itself, the receiver will first locate the sender's public signing key, and decrypt the hash value that was send over. Then the receiver will compute the hash of the original data. If the two hash values are identical, then the receiver can be assured of whom the sender was, and that the data was not corrupted.



# 9. Passwords

A password is required to access the contents of the Entrust Profile. The Profile in turn allows access to such sensitive material as encryption and signing keys. It is therefore imperative that the access password remains secret.

It is recommended that access to the user password be minimized in its use.

It is recommended that once a password has been used within programme code (i.e. EntrustProfile.logon()) that the data area containing the password be overwritten prior to deallocated or loss of scope (in the case of C++) and/or "garbage collected" (in the case of java).

It is ***not*** recommended that these specifications be used to create a "without human intervention" recoverable application. In other words, a human operator must be present to enter the password needed to start the application.

Notes:

# Annex B – PKI SOFTWARE REQUIREMENTS

# PKI SOFTWARE REQUIREMENTS

**Introduction**

To participate in the CCRA Customs Internet Gateway (CIG) project and complete the registration process, you must purchase a certificate issued by the CCRA CA.

To receive your certificate from the CCRA CA, you must first purchase the right to use that certificate and the client-side cryptographic software that will allow you to retrieve that certificate through a Key Generation Process.

The CCRA runs a version 5.0 Entrust Authority®, the requirement for the client-side software is Entrust Entelligence® version 5.0 or a 100% compatible Entrust® enabled client developed per specifications set forth in Appendix A of this document.

**Technical prerequisites**

**Software**

The target installation system for the version 5.0 Entrust Entelligence® client software must be loaded with one of the following operating systems:

Microsoft®Windows®95 (with Service Pack 1 or higher)
Microsoft®Windows®98
Microsoft®Windows®NT 4.0 (Service Pack 4 or higher)
Microsoft®Windows®2000
MacIntosh Power PC, System 7.1 or higher

**Hardware**

The following are the minimum workstation specifications for running the version 5.0 Entrust Entelligence® client:

Pentium class personal computer
32 Mb of RAM
At least 12Mb of available disk space
A CDROM reader
TCP/IP protocol stack

**Acquisition**

Please contact the manufacturer or an authorized reseller for terms, pricing and availability of the client software. You should indicate that you are a participant in the CCRA Customs Internet Gateway project.

# Annex C – PKI/Internet Technical Requirements

# PKI/Internet Technical Requirements

## Workstation

### Software

The Web Browser loaded on the workstation must support 128-bit strength encryption such as:

Microsoft®Internet Explorer 4.0, or higher
Netscape®Communicator 4.5, or higher

## Internet Connectivity

The CCRA Automated Registration Authority, Public Directory, and Certificate Authority are accessible by Internet only. As a result, an Internet connection is required. This connection can be an on-demand connection through an Internet service provider (ISP) or through a local area network infrastructure. The following technical details should be shared with local information technology resources or the ISP to ensure proper connectivity.

### Firewall

The following firewall configuration rules should be communicated to local technical resources and applied to the local firewall. Communications with the CCRA PKI will not be possible without implementation of these rules.

### Port 389 – LDAP

The Lightweight Directory Access Protocol (LDAP) version 3 is a universal data retrieval protocol. LDAP can be used as a common front-end for programmers to retrieve data from various sources, such as directories and databases. CCRA PKI uses an X500 directory as a public repository of certificates and certificate revocation lists (CRLs). This repository is accessed by the Entrust client to retrieve both the most-up-to-date certificates and the current CRLs to ensure that the established trust relationship is still valid. The directory access occurs every time a client starts up.
**Firewall rule:** Allow outgoing TCP connections to 207.245.213.12 only.

### Port 709 – Entrust KMS

The Entrust Key Management Service (KMS) is the protocol used by the Certificate Authority (CA) for client CA communications. After the initial profile creation, clients rarely need to communicate with the CA, except if there is a problem with their existing profile that needs to be rectified. Some examples of problems are: recovering the profile due to corruption, a forgotten password, certificates that have expired, and keys that need updating.
**Firewall rule:** Allow outgoing TCP connections to 207.245.213.141 only.

### Port 829 – PKIX-CMP

PKIX is a set of standards designed to allow interoperability between different PKI products. Entrust version 5 supports this protocol, and it is hoped that all CCRA clients will be migrated to version 5. If all clients at your location are version 5, apply the rule below and delete rule for port 709.
**Firewall rule:** Allow outgoing TCP connections to 207.245.213.141 only.

### Risk assessment

Opening any port on a firewall represents a certain amount of risk, since it represents additional target ports for potential attacks, mainly the “denial of service” type. There are no specific vulnerabilities identified with the ports detailed below. Properly configured firewall rules as per recommendations below will limit this risk by allowing outbound connections requests only (i.e. connection requests to these ports originating from the Internet will be denied).

Firewalls operate so that when a permitted internal outbound connection request through a selected port is granted, the returning inbound data connection will only be authorized if the source of the inbound connection corresponds to the target destination specified in the original outbound message.

### Target searchbases

Much like the operation of selecting an addressee on an email to allow it to reach the intended destination, a target certificate must be specified to enable the encryption and digital signature process specifically for the party the message is secured for.

The Customs Internet Gateway uses two target destination certificates, the first for Ops testing and the second for production transactions. They are:

For Ops testing;

**cn=Ops CCRA-ADRC GW+serialNumber=100063,ou=EQUIP,ou=EXTERN,ou=CCRA-ADRC,o=GC,c=CA**

For production;

**cn=Prod CCRA-ADRC GW+serialNumber=100062,ou=EQUIP,ou=EXTERN,ou=CCRA-ADRC,o=GC,c=CA**