The Public Health Information Network Messaging System (PHINMS) sends and receives sensitive data over the internet to the public health information systems using Electronic Business Extensible Markup Language (ebXML) technology securely.
The created ebXML message must be compliant with the ebXML Messaging Specification version 2.0. The ebXML Messaging Specification encompasses a set of services and protocols that allows partners to electronically request services from other participating partners. CDC has put forward a technical profile to describe exactly how partners are to exchange data within the Public Health Information Network.

The CDC profile describes a compliant superset of ebMS 2.0. Thus, all messages must be compliant with ebMS 2.0 as well as:

- All messaging is carried over HTTPs (HTTP with SSL)
- SSL Client Authentication is required for every message
- Messages have one single payload
- Payloads are required to be encrypted in a manner compliant to XML Encryption
- All messaging is synchronous, syncReplyMode is mshSignalsOnly
- Messages may be digitally signed via XML Digital Signature, specific tests should be performed to exercise the interoperability of messages that are both encrypted and signed.

The CDC PHIN profile requires the use of HTTPS. The CDC PHIN profile also requires the use of SSL Client Authentication, itself an optional feature of SSL/TLS. Additionally, the ebXML specification allows for multiple payloads whereas the CDC PHIN profile allows single payload of any type.

The CDC PHIN profile requires payload encryption, and describes details for applying encryption to a single XML document payload. ebMS 2.0 allows for both synchronous and asynchronous message exchange patterns. At this time, CDC PHIN profile only allows for Synchronous messaging. In other words, all replies including Acknowledgement of Receipt are sent immediately in an HTTP reply within the same HTTP session as the original message.

Secure Sockets Layer (SSL)

Secure Sockets Layer (SSL) provides encryption throughout the system. Connections are secured by SSL between the client's Web browser and the Web server within the SDN. Additional SSL connections secure message traffic between (1) the Web server and the application server, and (2) the application server and the database servers.

The PHIN MS Sender and Receiver use Secure Sockets Layer (SSL) between web-browser clients and the web server that accepts data from users. Additional SSL sessions secure data between the web server and the application server, and the application server and the database server. Each of these SSL sessions uses the same type of encryption used by all major financial services and electronic commerce sites today. From a user's perspective, then, confidential information is encrypted from the time it leaves the PC to the time it is stored in the central database.

Ensure all the correct ports, which may be 5088 (default local host port), 443 (Secure Socket Layer (SSL) – Hyper Text Transfer Protocol over Secure Sockets Layer (HTTPS)), and 389 (Lightweight Directory Access Protocol (LDAP)) are open on the firewall.

The PHINMS default port numbers are 6087 for the Database, 5088 for HTTP, and 5089 for HTTPS.

CDC's Secure Data Network (SDN)

Digital Certificates are digital identity of a person, computer or organization. It is a binary file which is used for Authentication, Encryption, and Signature etc.
Typically Digital certs are issued by Organizations or Root Certificate Authorities.

PHINMS recommends 1024 bit encryption strength both on Client certificate (Sender) and SSL certificates (IIS Proxy Server).

1. **Session Management**

   The SDN provides session management capabilities using specific software. The software provides CDC with a centralized security infrastructure for managing user authentication and access to Web applications. One of the important policy-based controls enforced by the software helps mitigate this risk. Within the SDN, the idle timeout is set to 15 minutes. Therefore, sessions where no activity[1] takes place for 15 minutes are terminated and further system activity is prevented (until another login occurs).

2. **Intrusion Detection**

   Most enterprise firewalls act as filters to determine which traffic can enter a network and which cannot. While the firewall examines incoming traffic, it does so only to prevent unauthorized traffic from entering the network; it does not look at the intent of the traffic. If traffic arrives from a single address on each port, the firewall will allow the authorized traffic (blocking the rest), but it may not recognize that this traffic pattern is consistent with port scanning—often the first step in an attack. Intrusion detection software, on the other hand, is designed to recognize unusual traffic patterns and respond, typically alerting administrators and allowing them to take corrective action.

   The SDN uses intrusion detection software. This software provides Web intrusion prevention against application-level breaches by identifying legitimate requests and permitting only those actions to take place. By preventing breaches and subsequently alerting administrators to any type of application manipulation through the browser, expected application behavior is maintained.

3. **Application Vulnerability Testing**

   Often known as buffer overflow attacks, application vulnerabilities have cost businesses and personal users billions of dollars. One step in preventing these attacks is pre-deployment application vulnerability testing. Within the CDC, the SDN provides application vulnerability testing using specific software. This software program detects security vulnerabilities automatically as an integrated component of an enterprise security process review.

Digital Certificate and Challenge Phrase

A digital certificate and challenge phrase are used to validate users before providing access to the SDN. After validation, access is granted to PHIN MS activity and role assignments.

In order for offenders to circumvent these security mechanisms, they would have to possess a valid digital certificate for which they knew the associated challenge phrase and discover a valid username and password combination for the PHIN MS application.

In addition, if a person leaves an organization, that digital certificate should be removed (de-activated), and a new one should be installed by the new user. Certificates expire yearly. Each user must apply for a new digital certificate each year.

There are many vendors which issues digital certificates like Client certificate and SSL server certificates. Some of them are Verisign, Thawte, Entrust, Equifax, Geotrust, etc. PHINMS (CDC) doesn't recommend one over other. You can also you the existing certificate and also can use self signed certificate.

There are several signs of certificate problem we come across while dealing with certificate. Some of the most common issues:

- SSL and Client Certificate expiration
- Encrypting with a wrong public key on the sender side
- Issue with Sender not able to trust the receiver's SSL cert chain
- Wrong private key password typed using the sender or receiver's console

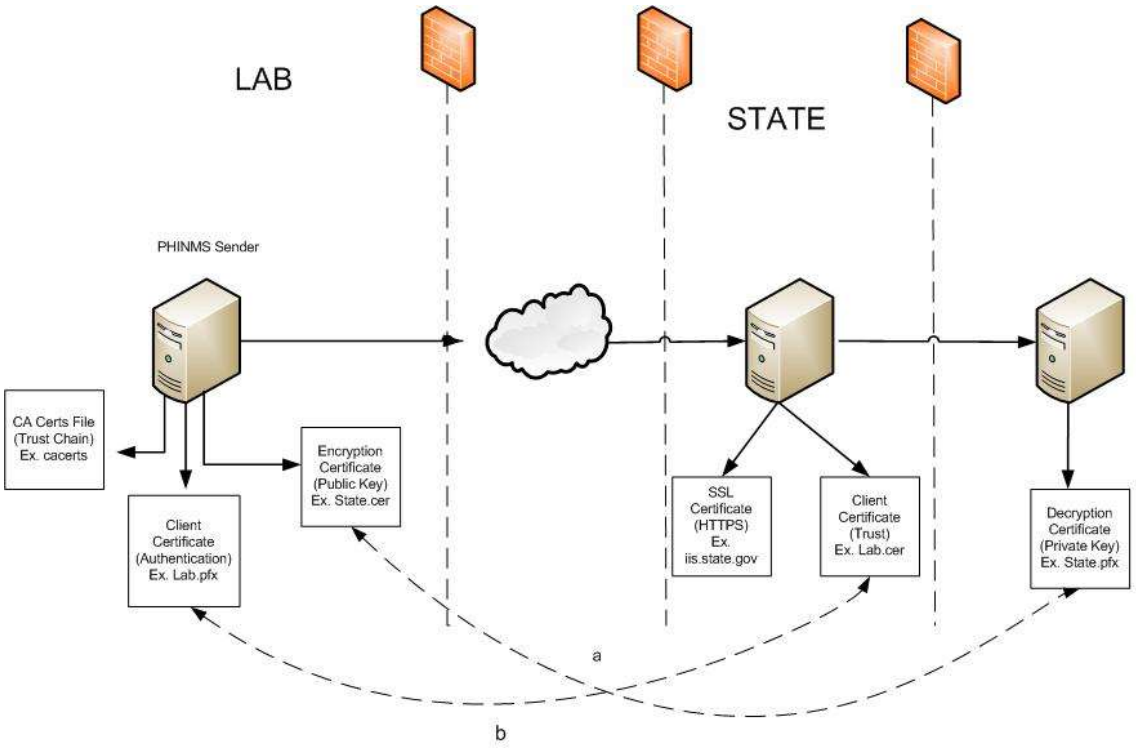
Easy way to troubleshoot a problem is to reading and understanding the log files.

PHIN Messaging Certificate

Collaboration Protocol Agreement (CPA). The receiver uses the CPA to confirm that the Sender is a valid trading partner. PHINMS creates a CPA file for each route listed on the Route Map tab of the Sender Configuration panel. The PHINMS Administrator must send the PHINMS Helpdesk the CPA files for each route specifying either the CDC Production Receiver or the CDC Staging Receiver. Only after the PHIN helpdesk has received the CPA file and applied it to the PHINMS Receiver can there be a successful transmission of messages from the Sender to the Receiver to CDC.

A PartyID is required for each organization and every organization sending and receiving messages to the CDC. A PartyID uniquely identifies a PHINMS installation, also called an instance or node. The PartyID is included with every message informing the recipient of the originator.

Setting up the PHINMS software requires the PartyID which is permanent and not required to be stored for later use. The PartyID is stored as long as the PHINMS instance for sending messages to partners is being used by the PHINMS application. The PHINMS application will need to be reinstalled if the PartyID needs to be changed.



Exchanging Data between Public Health Partners

Sending Partner

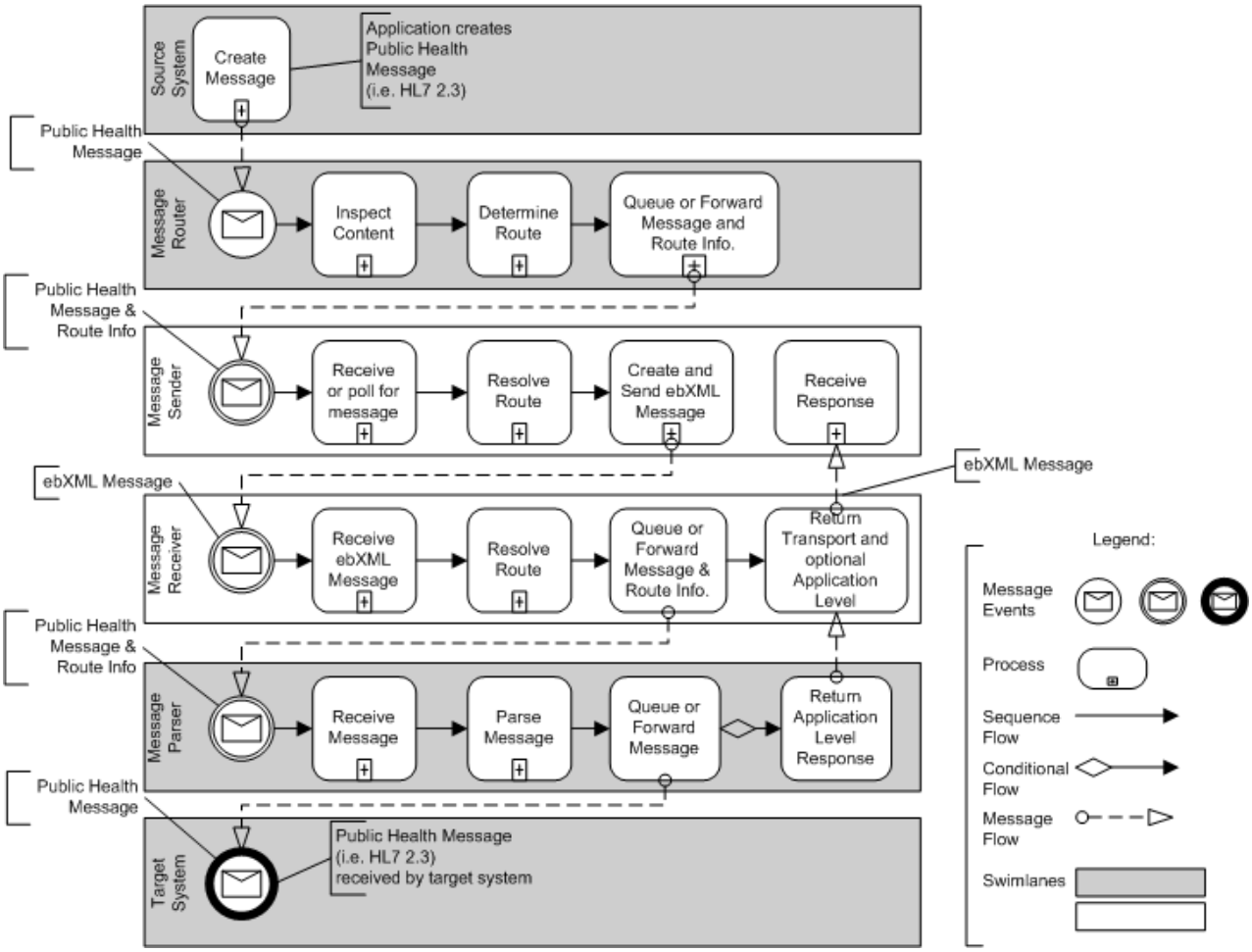
Forward the below items to the Receiving Partner:

- Public key of Sender's SDN certificate *****
- Root and Intermediate Certificate of the Sender's SDN Certificate (base 64)*****
- CPA file

Receiving Partner:

Forward the below items to the Sending Partner:

- Receiver's Party ID
- Host name of the IIS Web Server
- Public key of Receiver's IIS Web Server SSL certificate (base 64)*
- Public key of Receiver's SDN certificate (base 64)*



All PHIN Secure Message processing occurs in the Message Sender and Message Receiver swim-lanes. The Source System, Message Router, Message Parser and Target System swimlanes and the processes that occur within them are not within scope of PHIN Secure Messaging integration point. PHIN Secure Messaging starts upon acceptance of the public health (HL7) message and associated routing information from a Message Router. PHIN Secure Message processing ends upon receipt of the Public Health Message and associated routing information by the receiver-side Message Parser.

Message Sender and Message Receiver processing is based on the ebXML version 2.0 messaging specification. The ebXML specification extends XML based Simple Object Access Protocol (SOAP) specifications with essential capabilities to securely and reliably send messages over the Internet. EbXML's flexible enveloping technique allows payloads of any format type to be securely transported over the Internet.

Message-Send components must receive or accept a new message from the Message Router. The processes should retrieve the message by polling the Message Router or the process should be capable of accepting a new message through queuing technologies such as Java Message Services (JMS), Microsoft Message Queue (MSMQ) or similar commercial queuing implementation.

The Resolve Route process determines the receiving endpoint for a message based on route information provided by the Message Router process. The Resolve Route process maps "the route" to a Collaboration Protocol Agreement (CPA). The CPA is an electronic ebXML compliant file that specifies the conditions under which the sender and receiver conduct transactions. The file includes information such as Internet protocols (HTTP, HTTPS), reliability and security settings, as well as the endpoint address (URL).

The Create ebXML Message processes wrap message payloads in ebXML compliant envelope. The envelope is an XML document that contains the receiver's Service/Action as well as reliable delivery attributes such as "once-and-only-once delivery". The XML document is sent as the first component of a multi-part MIME message with the payload in the subsequent parts of the multipart message. The message payload consists of an attachment which in the case of public health, may consist of electronic lab report message of notifiable disease, an outbreak management message, a case notification message, a lab request message or other.

XML Encryption is identified as the preferred method for persistent encryption of message content for ebXML. At the time specification was drafted XML Encryption was itself a draft, and for this reason ebMS 2.0 does not specify detailed methods or give examples for how to encrypt a message payload. The CDC PHIN profile requires payload encryption, and describes details for applying encryption to a single XML document payload. ebMS 2.0 allows for both synchronous and asynchronous message exchange patterns. At this time, CDC PHIN profile only allows for Synchronous messaging. In other words, all replies including Acknowledgement of Receipt are sent immediately in an HTTP reply within the same HTTP session as the original message.

Once the message is retrieved from the queue, the process maps the specified route to a Collaboration Protocol Agreement (CPA). The CPA is an ebXML compliant file that specifies the conditions under which the sender and receiver conduct transactions and includes information such as endpoints (URLs), protocols and security settings. The process uses information contained in the CPA as well as routing attributes passed with the message to construct the ebXML envelope. The process creates the XML envelope by

- Generating a unique Conversation ID
- Including the Service/Action to call on the receiver
- Inserting Quality Service attributes such as OnceAndOnlyOnce delivery.

All ebXML messages must reference the ConversationId element. The ConversationId element identifies the conversation the message belongs to. Starting a conversation means sending the initial message in a sequence of transactions that comprise a complete business process. Once a conversation is started, sender and receiver continue a conversation by sending and receiving messages based on settings specified in the CPA. Qualities of Service attributes specify exactly how to handle error situations that typically occur over unreliable networks like the Internet. These attributes and supporting processes provide a strategy for retrying failed messages and persisting messages while waiting for the transmission to complete. If partners are digitally

signing data, the message first must be digitally signed via XML Digital Signature before being encrypted via XMLEncryption. XML Encryption should be used for the transmission of sensitive data.

Digitally Sign
To guarantee message integrity and authenticity, the sender can digitally sign the message. A digitally signed message provides a level of confidence that information has not be tampered with in route (message integrity) and that the sender actually sent the message (non-repudiation). The signature is created by performing an operation on the public health message such that the receiver of the message can confirm that a sender digitally signed the information and that the information has not subsequently changed in transit.

Using PHINMS, the message is digitally signed with the sender's private key. The signed message is sent to the receiver along with the sender's public key. The receiver then uses the sender's public key to determine if the received message is valid. In the CDC ebMS profile, payloads are not canonicalized during the digital signing process. This means that transformations applied to payloads are "implementation dependent." allowing for the interoperable validation of signature digests.

In the CDC ebMS profile, the KeyInfo element specified by XML DSIG may be ignored. In other words, the assumption is that the Digital Certificate used to verify a message signature will be exchanged out of band and will be known beforehand by the message receiver.

Detailed File and Database Design

PHIN MS

TransportQ Database Fields

FIELD NAME	DESCRIPTION SOURCE OPTION
recordId	Unique ID of the record in the table and the table's primary key. Auto Generated Mandatory
messageId	Application level message identifier. Application Optional payloadFile File name of the payload file of an outgoing message relative to a local directory such as myinputs.txt. Application Optional
payloadFile	File name of the payload file of an outgoing message relative to a local directory such as myinputs.txt. Application Optional
payloadContent	Used only when the payloadFile field is not specified. Populates the contents of a file within the table. Application Optional
destinationFilename	The name of the payload file when it is stored on the Receiver/handler. Application Optional
routeInfo	Points to the routemap table which points to the message route. Maps to a CPA, a configuration file which maps to the uniform resource locator (URL) of the Message Receiver. Application Mandatory
service	ebXML service name. Application Mandatory
action	ebXML action. Application Mandatory arguments Arguments specified by the Message Sender. Application Optional
messageCreationTime	Time when record was created, in UTC format. Sender Optional
messageRecipient	Recipient's ID specified by the Sender in the TransportQ_out. Sender Optional

processingStatus	Initial value of the status of record created queued. Sender Optional
applicationStatus	Status of the application. Sender Optional encryption The value is Yes if payload is encrypted and No if it is not. Application Mandatory
signature	If Yes, XML signature is applied to the payload. Application Mandatory
publicKeyLdapAddress	LDAP address of the LDAP directory server. Application Optional
publicKeyLdapBaseDN	LDAP Base Distinguished Name of the public key such as o=. Application Optional
publicKeyLdapDN	LDAP Distinguished Name of the public key such as cn=. Application Optional certificateURL URL of a recipeint's public key certificate. Application Optional
transportStatus	Transport level status. Sender Optional
transportErrorCode Error	code describing the transport failure. Sender Optional applicationErrorCode The error code returned by the service/action in a synchronous manner. Sender Optional
applicationResponse	The synchronous response returned by the service/action. Sender Optional
messageSentTime	Time when the message was sent, in UTC format. Sender Optional
messageReceivedTime	Time when the message was received, in UTC format. Sender Optional
responseMessageId	Message ID of the response message in the rout-not-read scenario. Sender Optional
responseArguments	Used in the Route-not-Read scenario to convey arguments being sent by a Message Sender to a receiving client. Sender Optional
responseLocalFile	The response to a poll type request which may contain a payload file in the Route-not-Read scenario Sender Optional
responseFilename	The response file name in the Route-not-Read scenario. Sender Optional
responseContent	Used when the sender.xml configuration file in the Message Sender specifies the response payload should be written into a database field instead of to a disk. Sender Optional
responseMessageOrigin	The PartyID of the party originating the message in the Route-not-Read scenario. Sender Optional
responseMessageSignature	The PartyID of the party signing the message in the Route-not-Read scenario. Sender Optional priority An integer indicating the request's priority. Application Optional

PHINMS Receiver's

WorkerQ Database Fields

FIELD NAME	DESCRIPTION SOURCE OPTION
recordId	Unique ID of the record in the table and the table's primary key. Receiver Mandatory
messageId	Application level message identifier. Sender Optional payloadName File name of the payload, specified by the Message Sender. Sender Optional
payloadBinaryContent	Image field written to by the Receiver servlet. Sender * Optional
payloadTextContent	Text field populated if textPayload=true in the servicemap entry. Sender * Optional
localFilename	File written to disk instead of a database when payloadToDisk =true. Receiver Mandatory

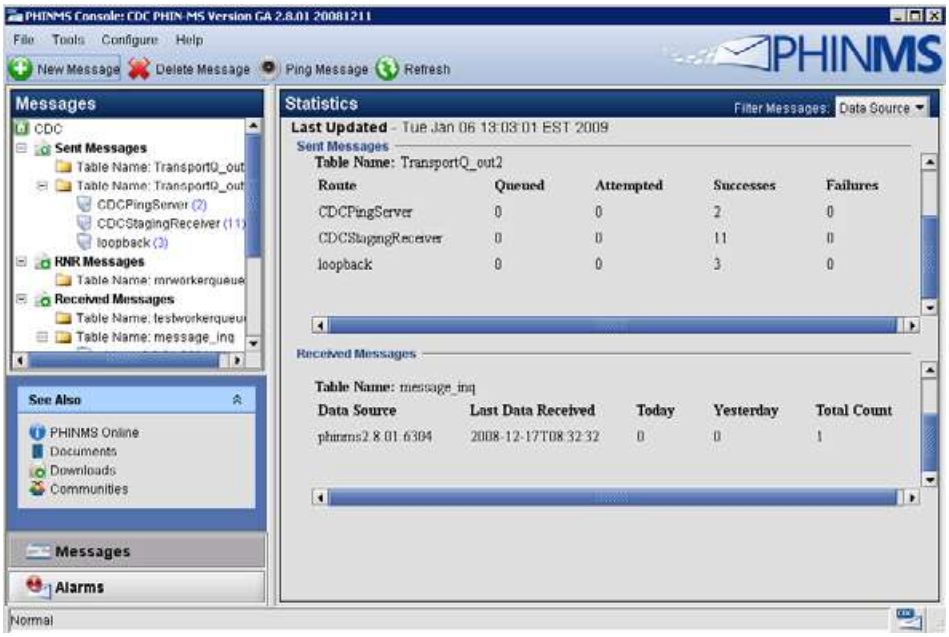
service	ebXML service name. Sender Mandatory
action	ebXML action. Sender Mandatory
arguments	Arguments specified by the Message Sender. Sender Optional
fromPartyId	PartyID of the Message Sender. Sender Optional
messageRecipient	Recipient's ID specified by the Sender in the TransportQ_out. Sender Optional
errorCode	Error code. Receiver Optional
errorMessage	Error message. Receiver Optional
processingStatus	Initial value of the status of record created queued. Receiver Optional
applicationStatus	Status of the application. Receiver Optional encryption The value is Yes if payload stored in worker Q is encrypted and No if it is not. Receiver Mandatory
receivedTime	Time when payload was received, in UTC format. Receiver Optional
lastUpdateTime	Time when record was last updated, in UTC format. Receiver Optional
processId	Identifies the process processing the record. Receiver Optional

* The two fields identified are mutually exclusive. The payload coming from the Sender is placed into one of the two fields by the Receiver depending on the receiver's configuration value for the textPayload = true/false field.

Detailed System Integration of CDC Tools Design

PHIN MS Console

- Must login with proper username and password



PHIN MMS Dashboard Console

- provides integrated management of the MSS modules
- http://localhost:8080/MSSDashboard with proper username and password
- Message Management
- Subscription Management
- Vocabulary Management
- Message Activity Monitor