



---

# **Security White Paper**

# **Public Health Information Network Messaging System (PHINMS)**

**Version: 1.1**

**Prepared by:  
U.S. Department of Health & Human Services**

**Date: April 16, 2008**



## ABSTRACT

The Public Health Information Network Messaging System (PHINMS) is the Centers for Disease Control and Prevention's (CDC) implementation of the Electronic Business Extensible Markup Language (ebXML) 2.0 messaging standards. This system was developed for the purpose of secure and reliable messaging over the Internet. This software has been widely deployed by CDC and its public health partners; including state health departments, local health departments, and healthcare providers. PHINMS is designed to leverage X.509 Digital Certificates issued by the public key infrastructures, but does not require a single, universal Public Key Infrastructure (PKI). This white paper discusses some of the security aspects of PHINMS.



**REVISION HISTORY**

<b>VERSION #</b>	<b>IMPLEMENTER</b>	<b>DATE</b>	<b>EXPLANATION</b>
1.0	Raja Kailar	03-05-2005	Co-Authored PHINMS Security White Paper.
1.1	Wendy Fama	04-16-2008	Added enhanced figures, corrected grammar, and removed duplicate sentences.

---

**TABLE OF CONTENTS**

**1.0 Introduction.....7**

**2.0 Security Considerations.....8**

    2.1 Trust <sup>1</sup> .....8

    2.2 Identification, Authentication, and Authorization .....8

    2.3 Authentication Factors .....10

    2.4 Confidentiality .....11

    2.5 Integrity and Non-Repudiation .....11

    2.6 Access Control .....11

    2.7 Public Key Infrastructure .....12

    2.8 Firewalls.....13

    2.9 Authentication Interoperability.....15

**3.0 Summary.....16**

    3.1 References.....16

**LIST OF FIGURES**

Figure 1. Security Credential Authentication ..... 9  
Figure 2. PHINMS Architecture ..... 10  
Figure 3. Message Routing ..... 12  
Figure 4. Parties with Internet Access ..... 13  
Figure 5. One Party Behind Firewall ..... 14  
Figure 6. Intermediary Server ..... 14  
Figure 7. Security Interoperability ..... 15

---

## ACRONYM LIST

B2B	Business to Business
CA	Collaboration Agreements
CDC	Centers for Disease Control and Prevention
DMZ	De-Militarized Zone
ebXML	Electronic Business Extensible Markup Language
E-SIGN	Electronic Signatures in Global and National Commerce Act
GPEA	Government Paperwork Elimination Act
HSM	Hardware-Based Security Modules
J2EE	Java 2 Platform, Enterprise Edition
JDBC	Java Database Connectivity
LDAP	Lightweight Directory Access Protocol
LRN	Laboratory Response Network
NEDDS	National Electronic Disease Surveillance System
NHSN	National Health Safety Network
ODBC	Open Database Connectivity
OMB	Office of Management and Budget
PHIN	Public Health Information Network
PHINMS	Public Health Information Network Messaging System
PKI	Public Key Infrastructure
SOAP	Simple Object Access Protocol
SSL	Secure Socket Layer
URL	Uniform Resource Locator
XML	Extensible Markup Language
XMLDSIG	Extensible Markup Language Digital Signature
XMLENC	Extensible Markup Language Encryption

## 1.0 INTRODUCTION

The Public Health Information Network Messaging System (PHINMS) is a Center for Disease Control and Prevention (CDC) developed implementation of existing standards for the secure and reliable transmittal of messages across the Internet.

The PHINMS relies on Electronic Business Extensible Markup Language (ebXML), Extensible Markup Language (XML) Extensible Markup Language Encryption (XMLENC), Extensible Markup Language Digital Signature (XMLDSIG), Simple Object Access Protocol (SOAP) and other standards. PHINMS is the primary message transport system for the National Electronic Disease Surveillance System (NEDSS), the Laboratory Response Network (LRN), National Health Safety Network (NHSN), and various other public health preparedness programs within CDC.

PHINMS is message data (payload) independent; hence it can be used to transport any type of data (e.g., text, binary) by design.

PHINMS is operating system neutral since it is implemented using Java and Java 2 Platform, Enterprise Edition (J2EE) standards.

It provides language neutral, queue-based interfaces for sending and receiving messages. The preferred queue implementation is an Open Database Connectivity (ODBC)/Java Database Connectivity (JDBC) compliant database table, but support for queues based on XML file descriptors also exists. PHINMS supports peer-to-peer messaging, as well as messaging via a third party using a send and poll model.

Message data security is accomplished using a combination of encryption, end-point authentication, and access control techniques. Transport reliability is accomplished using message integrity verifications; transmissions retries, and duplicate detections on message receipt.

Since PHINMS is used to transport sensitive data over public un-trusted networks (e.g., Internet), it is important to make sure end-points trust each other, are able to identify and authenticate each other, and communication channels preserve data confidentiality and integrity. Access to data sent and received should be controlled.

The balance of this paper will focus on some of the security considerations which went into the design and implementation of PHINMS.

## 2.0 SECURITY CONSIDERATIONS

Several security considerations went into the design, implementation, and deployment of PHINMS. This section provides a brief description of the security considerations.

### 2.1 Trust<sup>1</sup>

Secure messaging over public un-trusted networks requires messaging parties to be able to identify, authenticate, and trust each other. Real-world trust relationships need to be established between messaging organizations. This may include written agreements on service levels, liabilities, etc., pursuant to Office of Management and Budget (OMB) guidance on the Government Paperwork Elimination Act (GPEA) as well as the Electronic Signatures in Global and National Commerce Act (E-SIGN). Business processes for creating and handling messages at each end of the messaging pipe need to be put in place. Once trust and business relationships are established in real-world terms, Collaboration Agreements (CA) can be setup for message transport and processing. This includes setting up relationships to trust certification authorities and the identity of the sending and receiving components (e.g., using access control lists).

In the centralized trust scenario, a central node performs identity binding and security credentialing, and all nodes establish trust relationships with a central node. In this case, assuming  $n$  nodes, only  $O(n)$  trust relations are needed; however, in a heterogeneous environment where trust is de-centralized, with  $n$  nodes, each node may need to establish a trust relationship and security credentials with every other node, and in the worst case scenario  $O(n^2)$  trust relationships may be needed. Since messaging nodes typically belong to autonomous organizations and realms, establishing a globally accepted central identity and trust authority may not be politically acceptable. In a purely Public Key Infrastructure (PKI)-based authentication framework, a trust bridge such as the Federal Bridge CA could be used to address this problem; however, while PHINMS supports PKI-based authentication, it also supports other modes of authentication, such as basic or custom authentication.

PHINMS is designed to support both centralized and de-centralized trust models. Decisions on identity binding and security credentialing are made by the deploying organizations. Decisions on trusting the identity and security credentials are made mutually between messaging parties at the time when electronic collaborations are created.

### 2.2 Identification, Authentication, and Authorization

Identification and authentication in messaging is a difficult topic and is one which is far from mature. Since the message is typically sent by a process and not necessarily triggered by an individual, the authentication dialog must be scriptable. Meaning the sending application must be able to negotiate the exchange of credentials without human intervention. This is only possible for certain security tokens (e.g., hardware-based one time passwords and biometric identities do not lend themselves to this kind of scripted authentication exchange).

---

<sup>1</sup> Trust in this context is more generic than what is involved in PKI-based certificate chain validation. It may involve other (non-PKI) authentication mechanisms (e.g., basic or form based authentication).

PHINMS supports automated authentication dialogs for client certificate-based authentication over Secure Socket Layer (SSL), basic authentication, and form based authentication. The method used for mutual and automated identification and authentication between messaging parties is part of the electronic agreement between them, and should be established upfront, after the real-world trust relationship has been established.

Each messaging node in the Public Health Information Network (PHIN) is identified by a globally unique identifier. A messaging node (i.e., PHINMS Sender) may contain one or more security credentials allowing it to conduct automated authentication dialogs with other messaging nodes shown in Figure 1. In the absence of a universally trusted authority to issue security identities and credentials, potentially, a different security identity and set of credentials may be needed for the purpose of authenticating to each message destination. The security credentials may include client certificates (Key Stores), passwords, etc. Managing these security credentials can be a daunting task for the messaging administrator in the face of expiring certificates, password renewals etc. Certificates can be issued with an expiration period of years, whereas passwords typically must be changed every 90 days, so the problem with the latter is far more daunting.

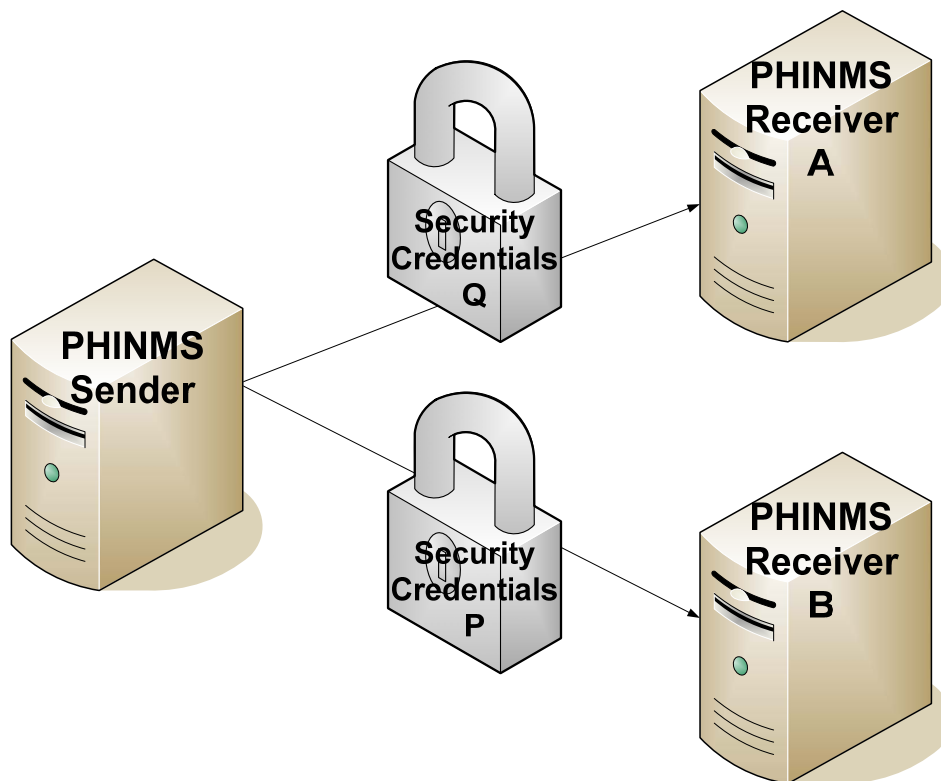


Figure 1. Security Credential Authentication

The recommended architecture for PHINMS messaging is one where the PHINMS Receiver components are protected from direct access from the Internet, by web-server proxies shown in Figure 2.

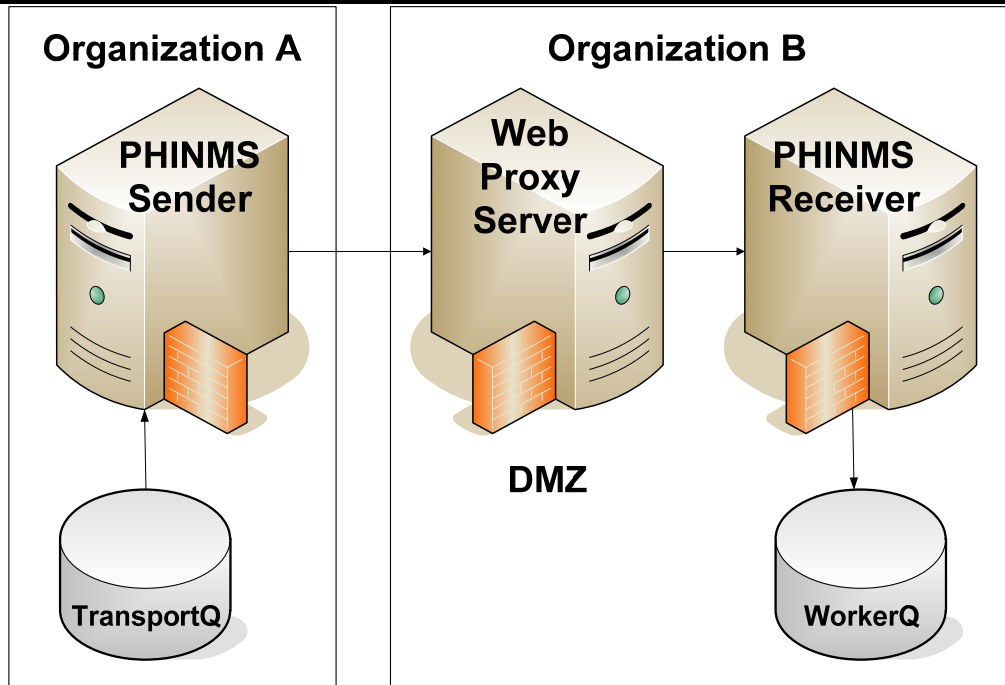


Figure 2. PHINMS Architecture

The web-server proxies typically reside in the organization’s De-Militarized Zone (DMZ), and mediates all inbound traffic for the PHINMS Receiver server, authenticating the sending process. SSL with client-certificate based authentication is the preferred method of authentication for PHINMS, since it is a well established standard and is widely implemented by web-server proxies.

Once the message sender is authenticated, it is the responsibility of the receiving organization’s web-server proxy to ensure that an authenticated sender only gains access to the receiver Uniform Resource Locator (URL). At this time, PHINMS does not provide support for attribute certificates which can be used for authorization decisions. Authorization information is stored on the receiver server, and enforced by the web-server proxy based on the authenticated identity of the PHINMS Senders.

### 2.3 Authentication Factors

Generally, two-factor authentication<sup>2</sup> is considered stronger and more secure than single-factor authentication for interactive authentication dialogs over the Internet. The case of Business to Business (B2B) automated security dialog; however, the security value of two-factor authentication is significantly diminished, since there is no real user behind the authentication dialog.

<sup>2</sup> Authentication mechanisms typically use secrets such as a user password, a user hardware token, and a user thumbprint known as authentication factors. A combination of two of the three factors is used for strong authentication.

All user factors required for the authentication dialogs would need to be pre-configured into the software which initiates the authentication handshake. Further, at the time of this writing, there are no published and accepted Internet standards for two-factor authentication in B2B transactions.

While it is possible to use hardware-based security modules (HSM) to emulate additional authentication factors for B2B exchanges, such mechanisms require additional hardware and management complexity.

## **2.4 Confidentiality**

Since communication is over un-trusted public networks, protecting its confidentiality is important. PHINMS uses payload level asymmetric encryption for end-to-end persistent confidentiality. The XML encryption standard is XMLENC and used for encrypting the payload.

The case of stored and forwarded messaging, data is protected from being read by intermediaries with asymmetric encryption using the public key of the message recipient to encrypt a random symmetric key, which in turn encrypts the data. Additionally, communication is typically conducted over a SSL channel, ensuring the message meta-data is also protected. The channel between the web-server proxy and the application server is also over SSL to ensure end-to-end confidentiality.

## **2.5 Integrity and Non-Repudiation**

PHINMS supports the use of XMLDSIG for message integrity and non-repudiation of message data. Signing certificates can be sent as part of the signature meta-data facilitating verification of the signature, alternatively, signing certificates can be statically pre-configured at the receiving node. Additionally, communication is typically conducted over SSL with client-certificate based authentication, which provides further message integrity and non-repudiation assurances.

## **2.6 Access Control**

The ebXML messaging standard supports message labels called Service and Action. These XML tags are part of the message envelope, and can be mapped to a service on the receiving node.

In the PHINMS implementation, messages received using the receiver servers are stored in database tables (queues) based on Service and Action tags shown in Figure 3. The queues are the equivalent of an application inbox and each application can only access its own inbox.

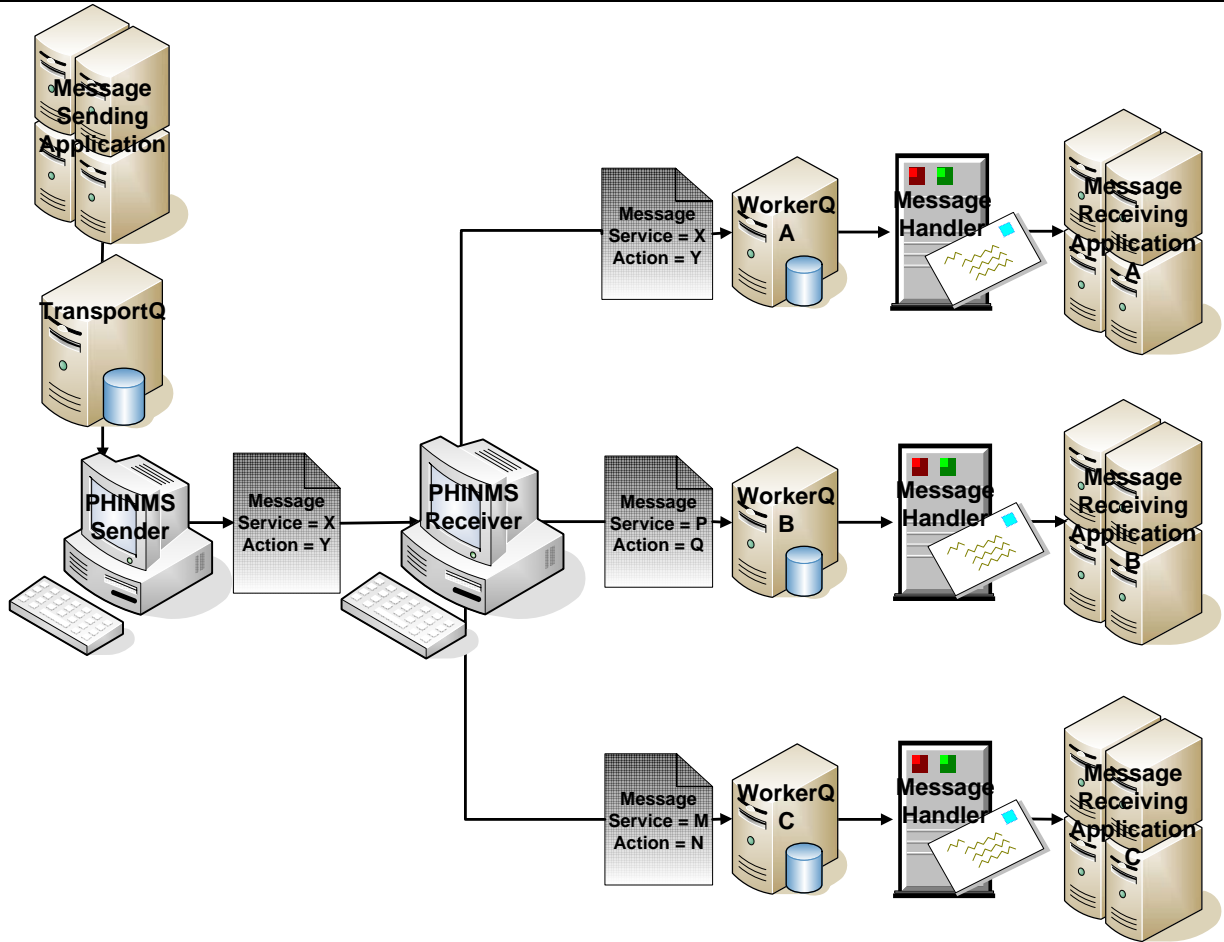


Figure 3. Message Routing

## 2.7 Public Key Infrastructure

PHINMS is designed to leverage a PKI but does not require a universal PKI. A PHINMS sending client can use a client certificate issued by one certification authority (CA) to authenticate to a PHINMS Receiver server and use a client certificate issued by a different CA to authenticate to a different PHINMS Receiver server. Currently, PKI trust relations are statically defined at the time when collaboration is established and configured between messaging entities. This is sometimes called the Certificate Trust List model. Ideally, public key certificates are published in a Lightweight Directory Access Protocol (LDAP) directory which does not need to be centralized. PHINMS also supports a web-service interface to publish and retrieve certificates. As a third alternative, encryption public key certificates can be distributed out of band and pre-configured at the message sending nodes. Public key certificates can be published in de-centralized LDAP directories as well.

## 2.8 Firewalls

Though firewalls are necessary for the protection of resources within an enterprise, they complicate matters for a messaging system trying to send messages across enterprise boundaries. PHINMS uses two independent pieces of code, a client capable of sending messages and receiving real time (synchronous) responses and a server receiver which receives messages at any time. These two components may be used in three possible scenarios. These examples assume the parties are in different organizations with separate firewalls. The three scenarios are as follows:

1. both parties are located outside their respective firewalls (i.e., in their DMZ),
2. one party is outside the firewall and the other is inside a firewall, or
3. both parties are inside their respective firewalls.

In the case where both parties are located outside their respective firewalls, messages may be sent and received at any time and acknowledgements sent either synchronously or asynchronously. This requires both parties have sending and receiving components installed shown in Figure 4.

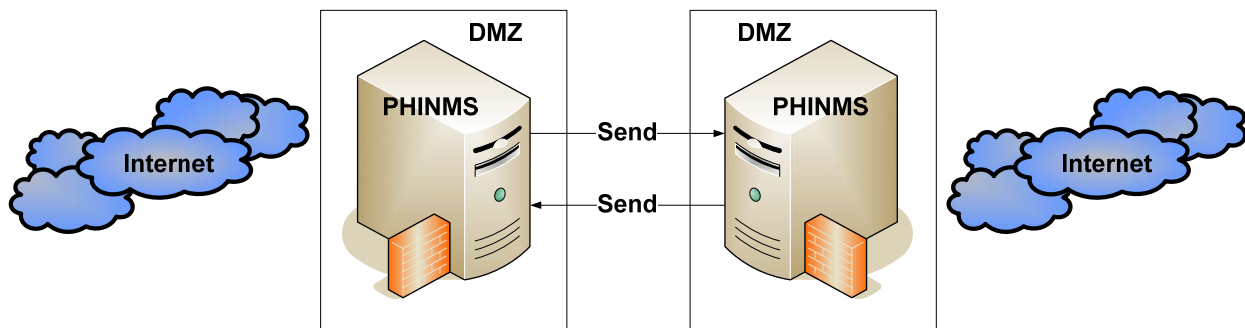


Figure 4. Parties with Internet Access

A situation where one party is behind a firewall and the other party has a server receiver located in the public Internet; message sending options are slightly reduced. The client piece behind the firewall can send data much like a typical browser to a receiver and receive synchronous acknowledgements back.

The client which sits behind a firewall messages cannot be received as firewalls typically block this type of push of information; however, it can poll for messages shown in Figure 5.

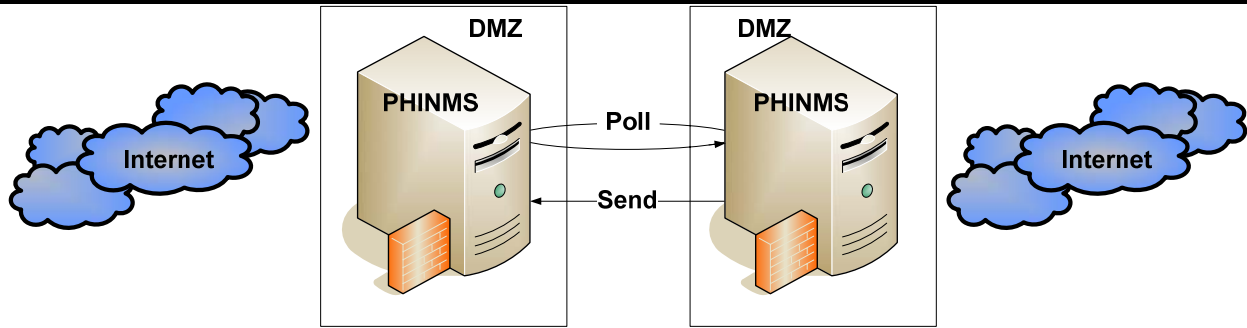


Figure 5. One Party Behind Firewall

Typically the client can reside on a workstation capable of hosting a Java application.

The case where both parties are behind firewalls, a third-party server with Internet presence is required to broker the exchange shown in Figure 6.

**Example:** Party A is located behind a firewall in enterprise 1 and wants to send a message to party B in enterprise 2 also behind a firewall. Party A must send a message to an intermediary server on the Internet with a service action stating the server should hold the message in a queue for B. When B polls the intermediary server, it will find the message from A in its queue and be able to retrieve the message.

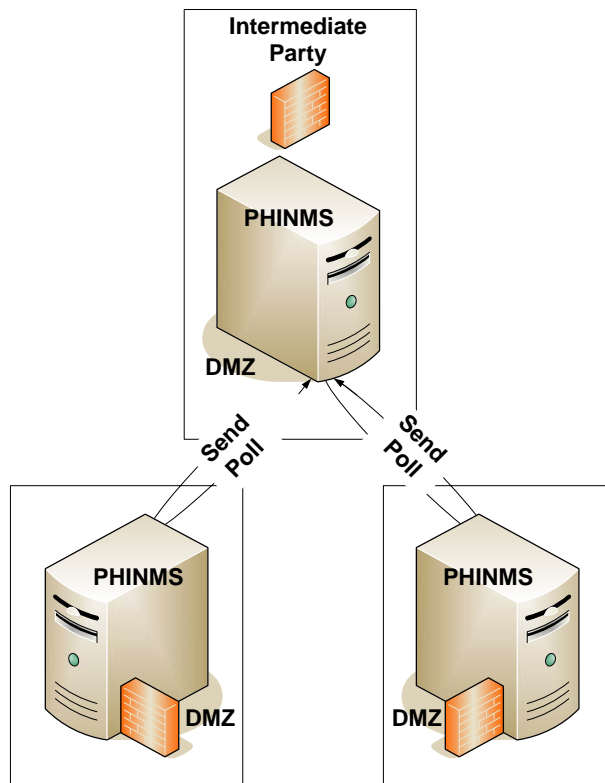


Figure 6. Intermediary Server

## 2.9 Authentication Interoperability

The ebXML messaging standard specifies the structure and semantics of message meta-data and addressing information, but for the most part, leaves the messaging security (identification and authentication) aspects to the implementers.

The security mechanisms also need to interoperate in addition to the message structure and semantics shown in Figure 7. XMLDSIGs can be used to support message non-repudiation (the strength of which is dependent upon legal elements which transcend the technology), but using them may not be sufficient for authentication, since digital signatures can be replayed<sup>3</sup>.

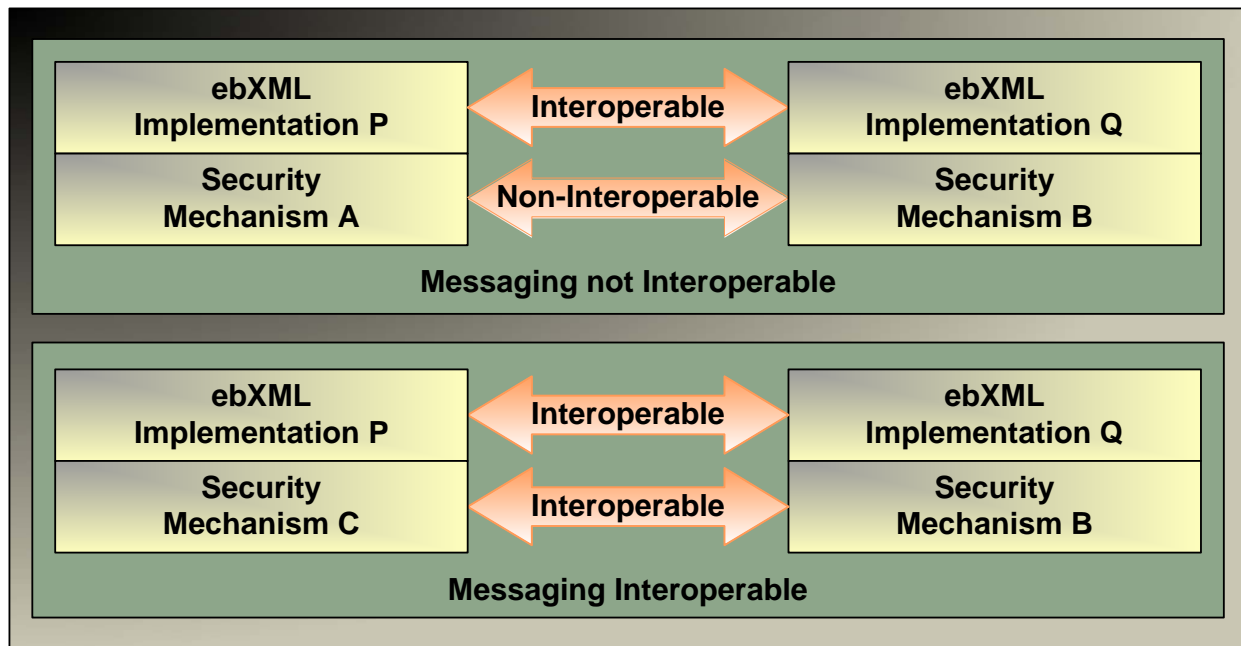


Figure 7. Security Interoperability

When used, XML digital signatures should be combined with a handshaking protocol such as SSL, which mitigates the threat of replay attacks and provide freshness assurances. The alternative is to use SSL with client certificate-based authentication. This provides per-link assurance of identity and authentication, as well as confidentiality. Since SSL is the most widely accepted standard, this is the recommended mode of authentication for PHINMS.

<sup>3</sup> A digital signature does not necessarily provide evidence unless it is cryptographically bound to a token, requiring time synchronization or none based handshakes. Without adequate assurances, use of DSIG in authentication may not be adequate for some applications.

### 3.0 SUMMARY

The security design, implementation and deployment considerations of CDC's PHINMS were discussed herein.

### 3.1 References

1. [ebXML] Message Service Specification - ebXML  
Version 2.0, OASIS ebXML Messaging Services Technical Committee  
(<http://www.ebxml.org/specs/ebMS2.pdf>)
2. [LRN] Laboratory Response Network Partners in Preparedness  
<http://www.bt.cdc.gov/lrn/>
3. [NEDSS] National Electronic Disease Surveillance System, The Surveillance and Monitoring Component for the Public Health Information Network  
[www.cdc.gov/nedss/](http://www.cdc.gov/nedss/)
4. [NHSN] National Healthcare Safety Network  
<http://www.cdc.gov/ncidod/hip/NNIS/members/nhsn.htm>
5. [SOAP] SOAP Version 1.2 Part 0: Primer  
<http://www.w3.org/TR/2003/REC-soap12-part0-20030624/>
6. [XMLENC] XML Encryption Requirements  
[www.w3.org/TR/xml-encryption-req](http://www.w3.org/TR/xml-encryption-req)
7. [XMLDSIG] XML-Signature Syntax Processing  
[www.w3.org/TR/xmlsig-code](http://www.w3.org/TR/xmlsig-code)