A DEMONSTRATION OF SECURE HEALTH TRANSPORT USING DIRECT MESSAGING TO ENABLE PROVIDER TO PATIENT HEALTH INFORMATION EXCHANGE

By

William B. Rice

A CAPSTONE

Presented to the Department of Medical Informatics & Clinical Epidemiology
and the Oregon Health & Science University
School of Medicine
in partial fulfillment of
the requirements for the degree of
Master of Biomedical Informatics

May 2012

School of Medicine
Oregon Health & Science University
CERTIFICATE OF APPROVAL

This is to certify that the Master's Capstone Project of

William B. Rice

has been approved

_____

Judith R. Logan, MD, MS
Capstone Advisor

Table of Contents

Figures

Acknowledgments

I would like to thank many people who provided support and inspiration throughout my education and academic pursuits. I am very grateful to all of these people for contributing their perspectives, insights, experiences and wisdom.

Thank you to the faculty, staff, and students at Oregon Health & Science University. I would like to thank Diane Doctor and Andrea Ilg for their patience with my seemingly endless questions. I would especially like to thank Dr. Judith R. Logan for giving me direction and guidance as my capstone advisor.

Thank you to all my friends, mentors and former colleagues at Vanderbilt Informatics and Vanderbilt Center for Better Health. I would like to especially thank Dr. Mark Frisse for his leadership and friendship.

Thank you to my parents for always giving me every opportunity. And lastly, thank you to my wife, Jennifer Rice, for her support and patience through it all.

Abstract

In 2010, the Office of the National Coordinator for Health Information Technology launched the Direct Project to expand the specifications of the nationwide health information network to create specifications and service descriptions that enable simple, secure point-to-point electronic messages between health care participants. Using Direct Messaging for electronic push of information between two healthcare providers or between a provider and a patient not only improves the patient experience, it also enables them to meet the exchange requirements of the Center for Medicare and Medicaid Services Electronic Health Record (EHR) incentive program. Although considerable progress is being made in launching Direct Messaging implementations, much about the role of a Health Information Service Provider (HISP) and some of a HISP's future functionality remain untested in the market place. The goal of this project was to demonstrate interoperable health information exchange using Direct Messaging specifications. To do this, a HISP was created using a .NET reference implementation provided by the Direct Project. Messages were then exchanged to demonstrate a provider sending a summary clinical document to a patient with a Microsoft HealthVault personal health record. A step-by-step instruction manual was created to guide users through a typical workflow scenario. While the technical implementation of the HISP was challenging, this demonstration illustrates that still harder problems remain to be solved for Direct Messaging to be widely adopted, including mechanisms for establishing trust relationships between Direct Messaging providers.

Acronyms

ARRA: The American Recovery and Reinvestment Act of 2009, commonly referred to as the Stimulus or The Recovery Act, is an economic stimulus package enacted by the 111th United States Congress in February 2009 and signed into law on February 17, 2009, by President Barack Obama

CA: Certificate authority, an entity that issues digital certificates.

CCD: Continuity of Care Document, an XML-based markup standard intended to specify the encoding, structure and semantics of a patient summary clinical document for exchange.

CERT: Cryptographic public keys are frequently published, and their authenticity is demonstrated by certificates. A CERT resource record is defined so that certificates and related certificate revocation lists can be stored in the Domain Name System (See DNS).

DNS: The Domain Name System (DNS) is a hierarchical distributed naming system for computers, services, or any resource connected to the Internet or a private network[1].

EHR: Electronic Health Record, a computerized system for recording, storing, producing, and using electronic patient medical and health information.

HIE: Health Information Exchange, the mobilization of healthcare information electronically across organizations within a region, community or hospital system[2].

---

[1] Wikipedia, The Free Encyclopedia. Domain Name System [Internet]. cited 2012 April. Available from: http://en.wikipedia.org/wiki/Domain_Name_System

[2] Wikipedia, The Free Encyclopedia. Health Information Exchange [Internet]. 2011 [updated 2011 May 9; cited 2012 April]. Available from: http://en.wikipedia.org/wiki/Health_information_exchange.

HISP: Health Information Service Provider, similar in concept to an Internet Service Provider, is responsible for the management of security and transport for Direct messaging.

HIT: Health information technology provides the umbrella framework to describe the comprehensive management of health information across computerized systems and its secure exchange between consumers, providers, government and quality entities, and insurers[3].

IHE: Integrating the Healthcare Enterprise, a group of healthcare industry stakeholders that promotes and defines coordination of established standards to provide meaningful and effective information exchange.

IETF: Internet Engineering Task Force, an international community that creates and maintains protocol standards that influence the Internet architecture

IMAP: Internet message access protocol is one of the most prevalent Internet standard protocols for e-mail retrieval[4].

ISP: An Internet service provider is an organization that provides access to the Internet[5].

MU: Meaningful Use, defined in the Final Rule from the Centers for Medicare & Medicaid Services published in July, 2010 under the ARRA Health Information Technology for Economic and Clinical Health Act provisions.

---

[3] Wikipedia, The Free Encyclopedia. Health information technology [Internet]. cited 2012 May. Available from: http://en.wikipedia.org/wiki/Health_information_technology

[4] Wikipedia, The Free Encyclopedia. Internet Message Access Protocol [Internet]. cited 2012 May. Available from: http://en.wikipedia.org/wiki/Imap

[5] Wikipedia, The Free Encyclopedia. Internet service provider [Internet]. cited 2012 May. Available from: http://en.wikipedia.org/wiki/Internet_Service_Provider

MIME: Multipurpose Internet Mail Extensions, an Internet standard that extends email to support content beyond simple ASCII plaintext data.

MVC: Model–View–Controller, a design pattern for computer user interfaces that divides an application into three areas of responsibility.

MX: Mail exchanger record, a type of resource record in the Domain Name System that specifies a mail server responsible for accepting email messages.

NwHIN: Nationwide Health Information Network, a set of standards, services and policies that enable secure health information exchange over the Internet.

ONC: Office of the National Coordinator for Health Information Technology, a division of the Office of the Secretary, within the U.S. Department of Health and Human Services. It is primarily focused on coordination of nationwide efforts to implement and use health information technology and the electronic exchange of health information.

PCP: Primary Care Physician is a physician or medical doctor who provides both the first contact for a person with an undiagnosed health concern as well as continuing care of varied medical conditions[6].

PHR: Personal health record, a health record where health data and information related to the care of a patient is maintained by the patient[7].

---

[6] Wikipedia, The Free Encyclopedia. Primary care physician [Internet]. cited 2012 May. Available from: http://en.wikipedia.org/wiki/Primary_care_physician

[7] Wikipedia, The Free Encyclopedia. Personal health record [Internet]. cited 2012 May. Available from: http://en.wikipedia.org/wiki/Personal_health_record

PKI: Public-key infrastructure, a set of hardware, software, people, policies, and procedures needed to create, manage, distribute, use, store, and revoke digital certificates.

POP/POP3: Post Office Protocol, version 3, is an application-layer Internet standard protocol used by local e-mail clients to retrieve e-mail from a remote server

PPACA: Patient Protection and Affordable Care Act, a US federal statute signed into law by President Barack Obama on March 23, 2010. This legislation makes sweeping changes to the US health care system.

REST: Representational state transfer is a style of software architecture for distributed systems commonly found on the Internet[8].

SDK: software development kit, a set of software development tools that allows for the creation of applications for a certain software package[9].

S/MIME: Secure/Multipurpose Internet Mail Extensions, an Internet standard for securing MIME data. S/MIME provides privacy and data security through encryption; and authentication, integrity assurance, and non-repudiation of origin through signing.

SMTP: Simple Mail Transport Protocol, an industry standard for transporting email.

---

[8] Wikipedia, The Free Encyclopedia. Representational state transfer [Internet]. cited 2012 May. Available from: http://en.wikipedia.org/wiki/Representational_state_transfer

[9] Wikipedia, The Free Encyclopedia. Software development kit [Internet]. cited 2012 May. Available from: http://en.wikipedia.org/wiki/Software_Development_Kit

SOAP: Simple Object Access Protocol, is a protocol specification for exchanging structured information in the implementation of Web Services in computer networks[10].

TLS: Transport Layer Security, a cryptographic protocol that provides communication security over the Internet.

UI: User Interface is the point of interaction between user(s) of a system and the system.

X.509: A digital certificates standard defined by IETF for asserting that an entity is who it purports to be.

XDM: Cross-Enterprise Document Media Interchange, an IHE integration profile, a specification for the exchange of electronic health record documents on portable media. XDM provides an option for zipped file transfer over email, which is very relevant to the Direct Project specifications.

WCF: Windows Communication Foundation is an application programming interface in the .NET Framework for building connected, service-oriented applications[11].

---

[10] Wikipedia, The Free Encyclopedia. Simple Object Access Protocol [Internet]. cited 2012 May. Available from: http://en.wikipedia.org/wiki/SOAP

[11] Wikipedia, The Free Encyclopedia. Windows Communication Foundation [Internet]. cited 2012 May. Available from: http://en.wikipedia.org/wiki/Windows_Communication_Foundation

Introduction

The modernization of our nation's health care information technology (HIT) infrastructure remains a top policy priority among state and federal governments. This priority reflects the belief among policymakers that HIT plays an essential role in improving healthcare quality and provides opportunities to reduce the costs of healthcare. Yet, to fully leverage the benefits of HIT, providers must not only adopt certified electronic health records (EHRs) within their organizations, but also share clinical data electronically to allow physician access to a patient's clinical data across sites of care (1).

Health information exchange (HIE), defined as the mobilization of healthcare information electronically across organizations within a region, community or hospital system (2), enables the collection of patient clinical data across sites of care to provide more complete and timely information for treatment. HIE improves coordination of care when patients see several providers and receive care in more than one care setting, as well as supporting quality improvement and reporting, public health activities, and clinical research.

HIE is often used to describe not only the activity of sharing clinical data, but also as the type of organizations that are formed to provide the health data sharing services. HIE, as an activity, provides the capability to electronically move clinical information among disparate health care information systems. The goal of an HIE organization is to enhance the quality and safety of patient care to benefit patients and the healthcare system. These two ideas are complimentary such that healthcare quality is improved inherently by the functionality and activity of sharing clinical information. At the same time, HIE activity is not exclusively provided by these types of HIE organizations. Common forms of HIE

organizations include regional, local, or state nonprofit or government-sponsored exchange networks that would broadly support all providers in a community. Today many other approaches are emerging, including local models advanced by newly formed accountable care organizations, exchange options offered by electronic health records vendors, and services provided by national exchange networks (3). It is clear that as needs increase to meet the demand for HIE, there will be a variety of exchange networks, services, and architectures to support different business models, local conditions, and provider requirements.

HIE-based activities are critical components for success with recent federal legislation in the American Recovery and Reinvestment Act of 2009 (ARRA) (1) and the Patient Protection and Affordable Care Act of 2010 (PPACA) (4) health reform initiatives. ARRA contains significant financial incentives for clinicians to implement certified electronic health record (EHR) systems. The legislation requires clinicians to demonstrate that they are using the certified EHR technology in a meaningful way (1). The government has made it clear that the key to Meaningful Use is the ability for information to follow patients, wherever and whenever they seek care, in a private and secure manner so that teams of doctors, nurses, and care managers can provide coordinated, effective, and efficient care (5). Meaningful Use requirements encompass critical aspects of health information exchange, including sharing important information with other providers and patients and reporting quality information and public health results. In other words, physician and hospital EHR systems must be able to exchange health information with EHR systems in other practices, hospitals, labs or other locations. This priority reflects the belief among policymakers that HIT plays an essential role in

improving healthcare quality and provides opportunities to reduce the costs of healthcare. Due to this importance, there has been a heightened focus on evolving technical architectures to enable HIE activities.

The Office of the National Coordinator for Health Information Technology (ONC) in the U.S. Department of Health and Human Services has supported development of nationwide health information networks (NwHINs), including standards, services and policies to support nationwide exchange of health information since 2004 (6). One of ONC's priorities has been a focus on facilitating development of the standards, services and policies needed for interoperable HIE across the nation. In 2010, ONC launched the Direct Project to expand the specifications for NwHINs to help providers begin to electronically transmit information to meet the limited health information exchange requirements of Stage 1 Meaningful Use (7). Informally known as "Direct," it is often described as a "push" model – somewhat like secure email – in which a message can be sent as long as the receiving person's email address is known (8). Direct complements the current specifications in the NwHIN by providing standards and specifications for a transport mechanism that allows participants to send encrypted information directly to known and trusted recipients over the Internet.

The goal of this project was to demonstrate interoperable health information exchange using Direct Messaging specifications. To do this, a Health Information Service Provider was established using a .NET reference implementation provided by the Direct Project. Direct Messages and clinical information were then exchanged demonstrating a provider sending a summary clinical document to a patient with a Direct enabled personal health record.

Background: The Direct Project

Established as an open government initiative and modeled after open-source approach for collaboration, in 2010 the Direct Project invited private companies and public sector entities to work together, on a volunteer basis, to collaborate on developing standards and services required to enable secure, directed health information exchange (7). The Direct Project focuses on the technical standards and services necessary to securely push content from a sender to a receiver and not the actual content exchanged. In particular, the Direct Project is intended to solve simple direct secure electronic transport supporting health information exchange. For example, a primary care physician who is referring a patient to a specialist can use Direct Messaging to provide a clinical summary document of that patient to the specialist and to receive a summary of the consultation.

Simply put, the Direct Project created a set of specifications and standards referred as "Direct Messaging" (9) that specifies a standards-based method for participants to send authenticated, encrypted health information directly to known, trusted recipients over the internet. At its core, it defines a security and transport protocol for exchanging information independent of exchange payload content. Combined with pre-negotiated, structured payloads between endpoints, innovative workflows can be implemented. Direct Messaging may be used by health care providers to come into compliance with some of the requirements for the Meaningful Use of EHRs necessary to qualify for Medicare and Medicaid incentive payments (1). Use of Direct Messaging is not required for Meaningful Use but is an option in meeting the requirements related to health information exchange.

Direct Messaging

Direct Messaging is defined by the Direct Project in the Applicability Statement for Secure Health Transport, which describes how to use Simple Mail Transfer Protocol (SMTP), Secure/Multipurpose Internet Mail Extensions (S/MIME) and X.509 certificates (9). Participants in the exchange are identified using standard e-mail addresses along with associated X.509 certificates. The data is packaged using standard Multipurpose Internet Mail Extensions (MIME) content types. Authentication and privacy are enforced by using Cryptographic Message Syntax (S/MIME), and confirmation delivery is performed using encrypted and signed Message Disposition Notification. Lastly, certificate discovery is typically accomplished through the use of the Domain Name System (DNS), although other services can be used but are less common.

Health Information Service Providers

Before reviewing the components of Direct, a key term and concept needs to be defined. In Direct, an entity that powers Direct exchange is called a Health Information Service Provider (HISP). The term HISP has been used by the Direct Project both to describe a function (the management of security and transport for directed exchange) and an organizational model (an organization that performs HISP functions on behalf of the sending or receiving organization or individual) (10). A HISP is similar in concept to an Internet Service Provider (ISP). As such, a HISP is responsible for delivering Direct Messages from a sender to a receiver via the internet. HISPs encrypt, authenticate, and run trust verification activities to ensure patient health information is secure.

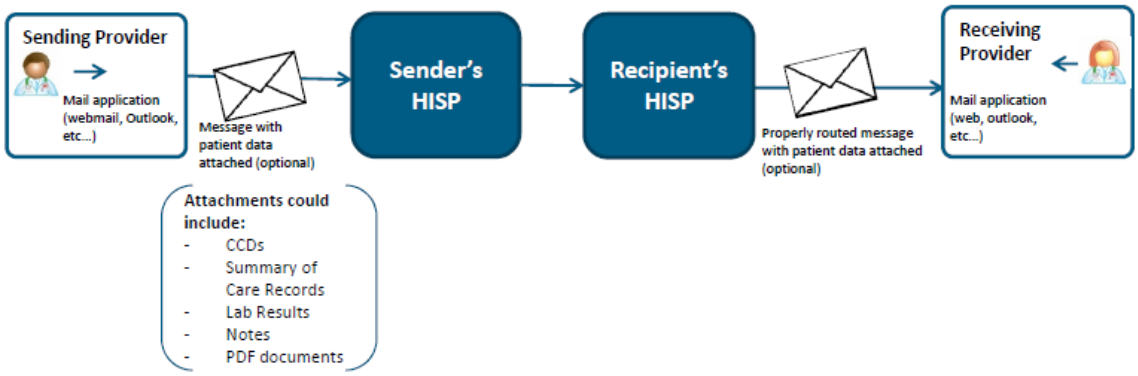


Figure 1: A HISP to HISP abstract model

The Direct Project workgroups developed an extensive, but non-exhaustive, list of possible deployment models that detail the various architectural components and their responsibilities for performing the necessary tasks for Direct Messaging (9,11). While Direct can be deployed using a federated architecture of, it is recognized that there are a number of benefits for an organization that performs the necessary functions that enable Direct Messaging for its subscribers. The role of a HISP can be filled by a variety of organizations including providers, payers, EHR vendors, personal health record (PHR) vendors, health information exchanges, and third-party entities.

A HISP is a new class of entity established by the Direct Project whose role is to provide Direct Messaging services to its subscribers. A HISP may be a separate business or technical entity from the sender or receiver, depending on the deployment option chosen (11). Between the sender and receiver of Direct Messages, one will usually find two HISPs; one HISP for the sender and one for the receiver (Figure 1), although in some instances, it is conceivable that senders and receivers may share the same HISP.

It is important to understand that a HISP performs additional functions aimed at overall security and trust of the email service. These services may include account set up and management, registry services, and, most importantly, handling of the security and trust aspects of Direct Messaging and exchange between senders and receivers. In general, a HISP's duties include:

- Ability to assign unique Direct addresses to individuals or organizations
    - Ability to associate X.509 certificates with full Direct address or Health Domain Names
    - Issue certificates as a Certificate Authority (CA) or obtain the certificates from a trusted third-party CA
- Provide an "edge" or "on-ramp" protocol or application/protocol combination to the end user, for sending and receiving messages and attachments
- Package message content using MIME and, optionally, Cross-Enterprise Document Media Interchange (XDM)
- Secure the confidentiality and integrity of the content by handling it through S/MIME encryption and signatures
- Ensure the authenticity of the sender and receiver via X.509 certificates
- Route messages between HISPs

Having reviewed the technical functions that a HISP performs, we will now review the technology architecture and its components that enable Direct Messaging services. Although Direct focuses on standards and services rather than requirements for architecture, the components described are typical within a HISP deployment.

Direct Components

An early philosophy of the Direct Project was to build on top of existing standards already ubiquitously deployed. These were driving factors that lead to the final Direct specification, defined in the Applicability Statement for Secure Health Transport (9). Direct is a set of standards that are delivered by coupling together a series of services and functions that can be grouped into an extensible set of components (Figure 2). Components include the following:

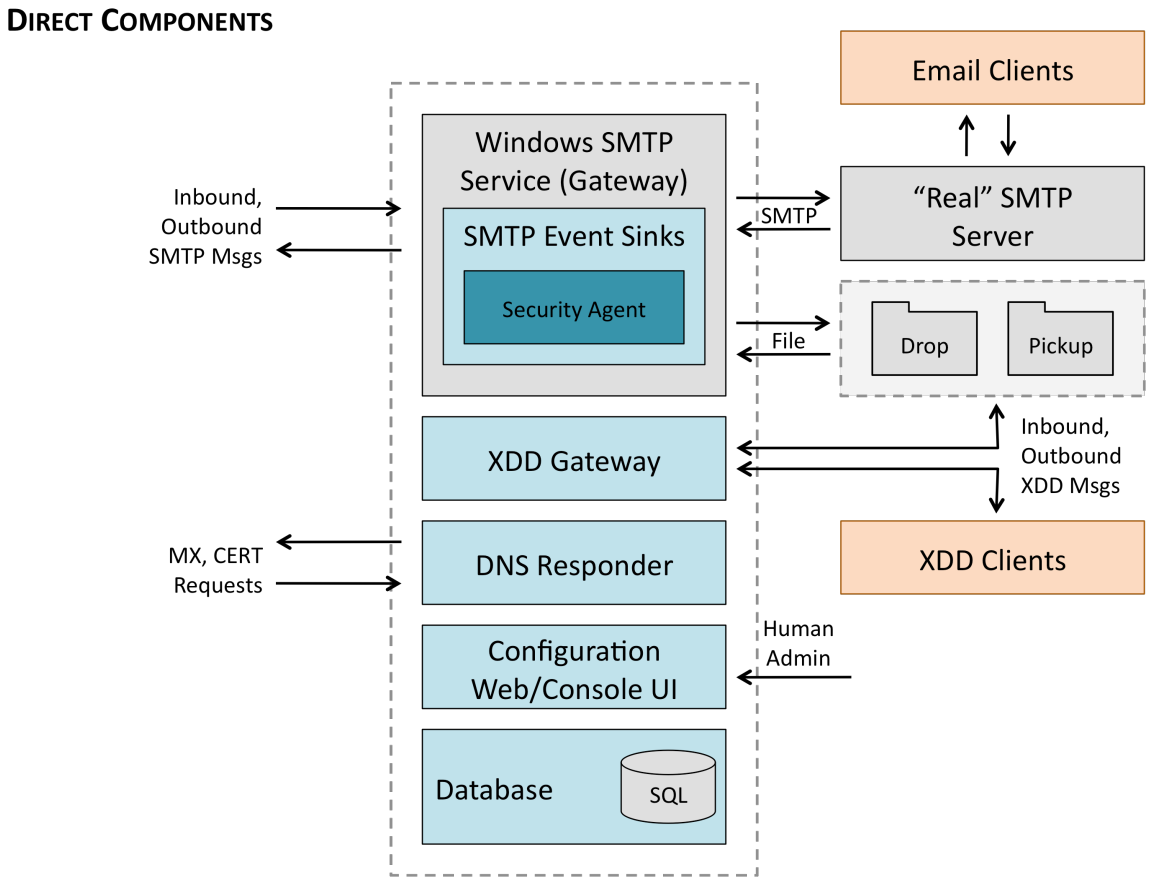


Figure 2: A typical architectural diagram of the Direct Project

- An SMTP gateway that inserts the agent code in front of any standard SMTP email server, handles error conditions, and can drop/pickup messages from file system folders as a loosely-coupled integration point for non-SMTP systems.
- The Security Agent that performs S/MIME encoding and decoding, and enforces the NwHIN Direct Security & Trust requirements.
- A configuration system that holds account, preference and certificate information in a database, and exposes web services for retrieving and manipulating configurations.
- A web-based configuration user interface (UI) that sits on top of the default configuration system.
- A DNS responder service for mail exchanger record (MX) and CERT record distribution.
- An audit logging system that accepts audit events for storage.
- Miscellaneous standard server utility code such as an audit logging system, and an XDD gateway that enables communication between other NwHIN nodes and the NwHIN Direct SMTP backbone.

SMTP Gateway

In the early Direct Project workgroups, there was much debate regarding the underlying technology that would ultimately drive the information exchange. After reviewing several proposals there was consensus for using SMTP with Multipurpose Internet Mail Extensions (MIME) attachments as described by the Internet Engineering Task Force (IETF) draft standard for Internet Message Format RFC5322 (12). MIME is an Internet standard that extends the format of email. Virtually all Internet email is transmitted via

SMTP in MIME format. Internet email is so closely associated with the SMTP and MIME standards that it is sometimes called SMTP/MIME email (13).

Direct Project wanted to build on existing standards, and SMTP is a ubiquitous and mature standard for message transport. An important attribute of the original Direct Project proposal was universal addressing. Addressing refers to the source and destination endpoints of a message and how they are named. Universal means that an endpoint name is unique across the entire namespace of a protocol. An email address is an example of a universal address. Internet addresses as described in RFC5322 are globally unique endpoints (12). Generally, each Direct participant that subscribes to HISP services is assigned an email address. Message routing to an internet address over SMTP is already built into almost every SMTP server using DNS standards.

SMTP is also more or less agnostic to the content of the payload carried in the message. RFC5322 gives some structure and meaning to the payload, but is still flexible enough to allow almost any type of content to be packaged. This is an important attribute of Direct, as it does not limit the type of content that can be exchanged from one participant to another.

There are, of course, limitations regarding the type of applications that can be built on top of Direct. SMTP is not fit for every use case. SMTP is an asynchronous protocol which adds complexity to ensuring quality of service. This means that there is not a guarantee that a message will be successfully delivered to its final destination after it is leaves the source endpoint. Additional work is currently being done to provide further guidance for HISP responsibilities in terms of quality of service. Some use cases utilizing Direct will

absolutely require certain levels of message delivery assurance or negative acknowledgement.

Direct does leave room for other protocols such as SOAP to be used as the backbone transport. This has both historical and forward-looking implications and potentially requires more complex configuration and pre-negotiated protocol and address routing (9). The Direct Project, however, requires HISPs to support SMTP as a backbone transport protocol option to provide a common transport standard across all HISPs. In the default configuration, the Direct SMTP gateway is configured to sit directly between a "real" SMTP server and the Internet. The Direct SMTP gateway server accepts all incoming mail for local domains, passes it through the "ProcessingIncomingMessage" method of the security agent, and relays the resulting message to the real server. The real server accepts outgoing mail from local clients and relays it to the gateway, which passes it through the "ProcessOutgoingMessage" method of the security agent and then relays it out to the Internet for delivery.

Security Agent

The core value proposition of Direct is securely transporting authenticated messages between mutually trusted parties, and implicitly describes a component called the security agent. This agent is responsible for implementing security and trust specifications. Because Direct uses MIME messages as its payload over SMTP, it needs a way to secure the message but remain in compliance with MIME standards. Fortunately, a MIME extension called S/MIME exists and is defined by IETF draft standard RFC5751 (14). S/MIME has attributes that cover both security and message authenticity.

Direct specifications state that all messages on the backbone protocol have a MIME content type of application/pkcs7-mime – the content type of an encrypted S/MIME message (9). The security agent is responsible for encrypting and decrypting all outgoing and incoming messages, respectively, using Public Key Infrastructure (PKI) and X.509 certificates. For the purpose of S/MIME, certificates contain the keys that are used to encrypt and decrypt messages.

A X.509 public key certificate (also known as a digital certificate or identity certificate) is an electronic document which uses a digital signature to bind a public key with an identity — information such as the name of a person or an organization, their address, and other identity information. In cryptography, X.509 is a standard for PKI. PKI assumes a strict hierarchical system of CAs for issuing the certificates. X.509 and is a model that specifies formats for public key certificates, certificate revocation lists, attribute certificates, and a certification path validation algorithm (15).

PKI is essential to understanding the foundation for interoperable Direct Messaging among parties who wish to exchange messages. One's identity is bound to one's public key through a digital certificate. This digital certificate is an electronic document that has information about the user, such as the organization, address, and information about how your public key may be used. A user also has a private key that is kept secret. Everyone who uses PKI technology has two keys, one they distribute publicly and the other they keep hidden. The two keys are mathematically related in such a way that permits encryption with one key (the public key) and decipherment or decryption only with the other key (the private key).

With PKI in place, User A wants to send User B an encrypted message, to protect the privacy of its contents. If User A has User B's public key, then User A can use it to encrypt the message to send to User B. Because of the mathematical correspondence involved, only User B private key, and no one else's, can decrypt the message.

Every endpoint in Direct is associated with one or more X.509 certificates. When a message is addressed to a recipient, the message is encrypted using the recipient's public key contained in the associated certificate using S/MIME standards. This is an oversimplification of the process, but the result is an S/MIME-encrypted message envelope that contains the original message. The encrypted payload is then sent to the recipient's HISP using SMTP over public networks. When the recipient's HISP receives the message, the recipient's certificate is obtained along with its corresponding private key. The message is then decrypted, and the original message is extracted. The asymmetric attributes of the public and private keys ensure that only the recipient's private key can decrypt the message.

What differentiates Direct Messaging from a regular mail service like Google's Gmail or Outlook email is the inclusion of PKI, the additional layer of security that makes it possible to authenticate and encrypt the messages sent and received in order to keep them private. Almost every public SMTP server supports transport over a non-encrypted channel. Because the line is unencrypted, however, the message payload itself must be encrypted to protect against eavesdropping and ensure message integrity, and S/MIME provides both of these functions.

Configuration Database and User Interface

The configuration service uses a simple database that implements a security model that permits both system administrators and specific account owners to manage their configurations. The configuration service allows the HISP to register other HISP administrators, register domains, register domain users and register their addresses. These domain and certificate records are delivered to requesting HISPs through a DNS service described in detail below. However, the configuration database stores the DNS settings associated for each domain hosted within the HISP. These are important configurations but arguably the most important configuration database service is to register domain and user PKI certificates hosted on the HISP as well as domains and certificates that are trusted from other HISPs.

The database can be administered using a traditional SQL command interface. Additionally, the Direct reference implementation also supplies a web-based configuration UI tool. The configuration UI is a simple model, view, controller (MVC) web site used by administrators (both system administrators and domain/account/address administrators) to manually configure and administer the underlying configuration database. The MVC presents a simple UI application with a graphical view of the database model below the application (Figure 3). This simple UI application allows the administrator to issue SQL commands though the UI interface rather than through a SQL command line interface.

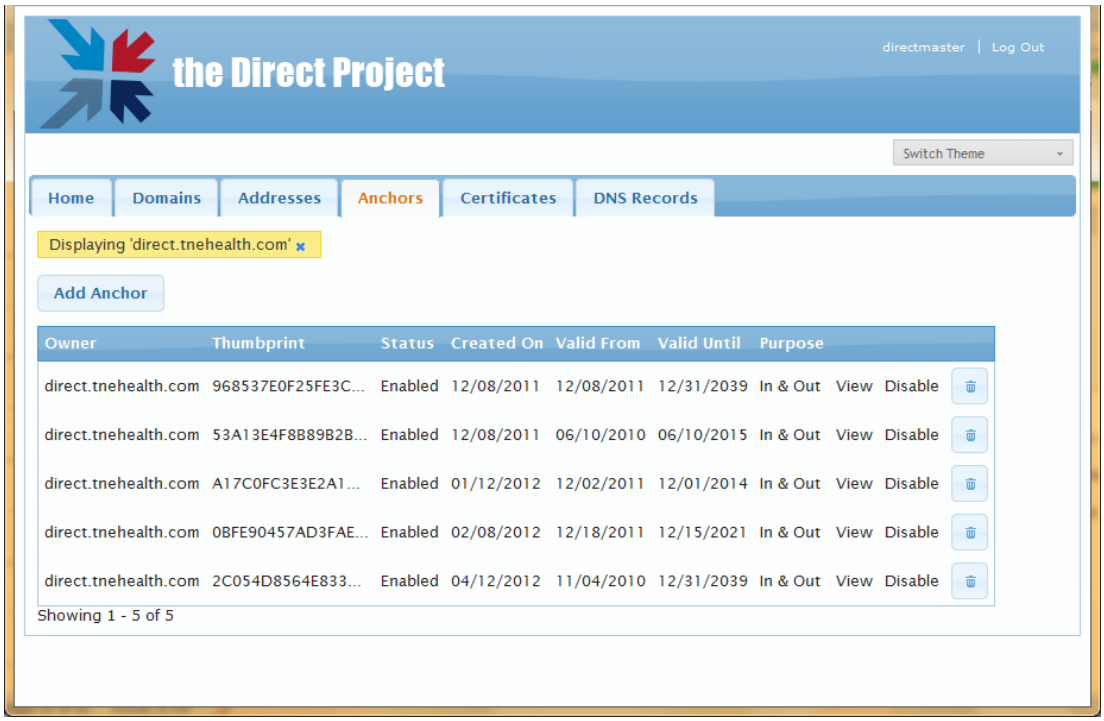


Figure 3: Direct Configuration UI

By including a web-based configuration UI tool with the Direct reference implementations, a broad group of users may begin implementation of Direct who might not otherwise be familiar with or comfortable administrating a system using command-line SQL statements for creating or maintaining the configuration settings.

Domain Name System

For the security model to work there must be a method for certificates to be “discovered" for encryption, decryption and signature operations. PKI can be difficult to implement, and certificate discovery is just one small piece of the puzzle. There are two use cases of certificate discovery in Direct: private and public discovery. Private discovery refers to accessing a certificate, along with its private key, and is only applicable to addresses maintained by a HISP. It is up to the HISP to implement proper protection of private keys

and the methods to access them. Direct implements a few certificate resolvers for discovering private certificates.

Public discovery refers to finding certificates that are not managed by the HISP. Because Direct wanted to use existing and ubiquitous standards, DNS using CERT-type records was originally selected as the preferred method. The Direct implementation includes a simple DNS responder that answers MX and CERT requests using information from the configuration service (16).

While the components that enable Direct are technically complex, Direct Messaging is simple in concept and carries many benefits. At the most basic level, the solution provides a secure email platform designed for the exchange of health information between providers. Since unsecure email carries with it numerous risks of information being compromised during transmission, or being accessed by unauthorized users, providers cannot use regular email to exchange patient health information. Direct Messaging help providers communicate more securely, which will ultimately help provide better care for patients.

Building a Direct Messaging Gateway

The main purpose of this project was to demonstrate Direct health information exchange. To do so required building a functional HISP. The steps described below are only intended to provide a high-level review of this process, not to serve as an instruction manual for how build a Direct enabled gateway (Figure 4). After building and testing this gateway, it was possible to demonstrate a use case that would not only illustrate the technical abilities of the HISP, but also demonstrate real-world scenarios of how Direct Messaging could be used.

Direct Reference Implementation

In order to achieve the goal for Direct Messaging, volunteer members of The Direct Project set out to create an open-source reference implementation and associated libraries implementing the Direct specifications. Participants in the Direct Project collaboratively authored two reference software implementations (Java and .NET) of the Direct

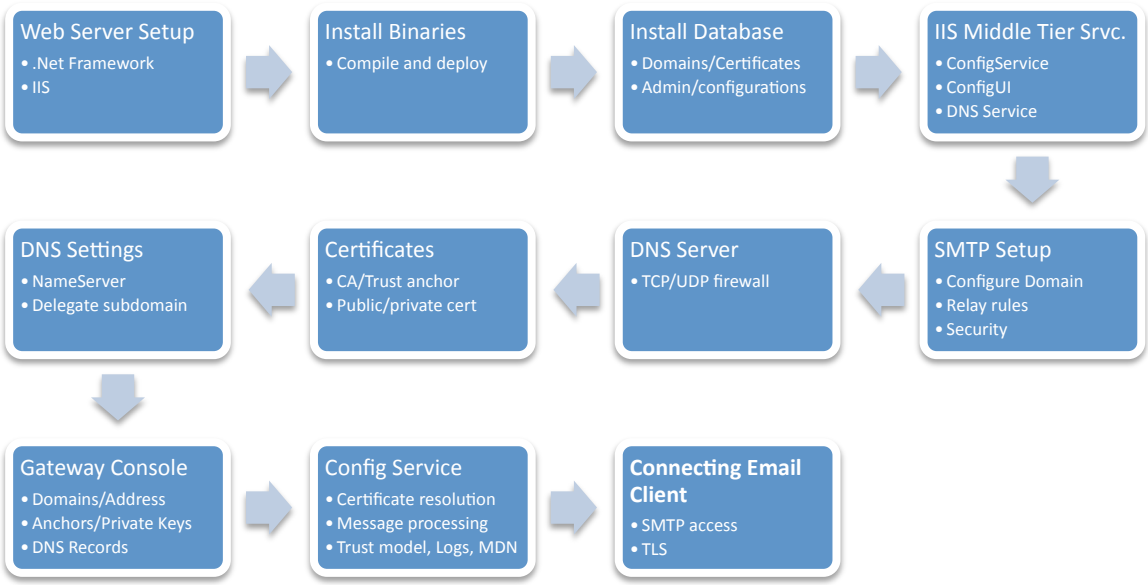


Figure 4: Steps for Installing the Gateway

specifications, so as to serve the needs of the pilot projects, and to accelerate adoption in the health IT marketplace (16). Participants contributed content, ideas, and specifications under a Creative Commons Attribution license. The reference implementations are intended to serve both as an "out of the box" system for providing Direct Messaging services, and as an extensible set of components that can be integrated into an existing environment.

To demonstrate health information exchange using Direct specifications, I began by building a Direct-compliant message gateway, otherwise known as a HISP. I chose to use the .NET reference implementation as a starting point based on my personal familiarity with Windows-based server technologies. It should be noted that the Java reference implementation provides equal functionality to the .NET version and neither reference implementation provides benefits or disadvantages over the other code base. Additionally, although the Direct Project did create two reference implementations, developers could program a Direct Messaging gateway using programming systems besides .NET or Java as long as the services and functionality adhere to transport standards defined in the Direct Messaging applicability statement (9).

Step 1: Server Deployment

Before installing the binaries for the .Net reference implementation, I needed a server that met the .NET Direct system requirements, including:

- Windows Server 2008 (64-Bit)
- Windows Large Message Hot Fix (to fix an issue with large messages)
- .Net Framework 3.5+ with SP1

- SMTP Server
- IIS 7x
- Windows communication foundation (WCF) with HTTP Activation
- SQL Server Express, SQL Server 2008 or equivalent SQL Database
- Outbound Firewall Ports:
    - TCP Port 25 used by SMTP Server to send outbound mail
    - TCP Port 53 used by the Gateway to resolve a mail recipient's CERT records from DNS.
    - UDP Port 53 used to resolve standard MX, NS and ANAME records
- Inbound Firewall Ports:
    - TCP Port 25 used by SMTP Server to receive incoming mail
    - TCP Port 53 primarily used to receive and respond to requests for CERT records
    - UDP Port 53 Respond to requests for DNS records like MX, NS, ANAME and SOA

To meet these requirements, I chose to use the Amazon Elastic Compute Cloud (Amazon EC2) running Microsoft Windows Server 2008. Amazon EC2 is a web service that provides resizable computing capacity in the cloud without the operational burden of on-premises server software. The use of Amazon EC2 is not required as part of the implementation nor is it use in this demonstration intended to endorse this service over other cloud-based server offerings.

Once the cloud-based Windows Server 2008 was initialized, I was able to ensure the proper configuration including IIS configurations, WCF services, and firewall port exceptions. Familiarity and experience with Windows Server is highly encouraged.

Step 2: Install .Net Binaries

Once the Windows server was properly configured, the next steps were to download and deploy the Direct reference implementation components. The code is maintained within the Google Code Repository, a free collaborative development environment for open source projects. Once the files were downloaded, using a command line tool to issue the install commands, the binaries were compiled and deployed onto the server. Again, familiarity with .NET, Visual Basic and application deployment is highly encouraged for this step.

Step 3: Install Configuration Database

With the server properly configured and the code deployed to the server, the next step was initialization of the Direct configuration database. Fortunately, the Amazon EC2 Windows Server image comes preconfigured with Microsoft SQL Server 2008 Express database. With the SQL Server database already installed, this step involved the creation of the tables, naming the attributes, defining the data types, and establishing primary and foreign keys (Figure 5). The .NET reference implementation was written using Microsoft SQL Server so the reference implementation ships with batch scripts that can assist automating the creation process.

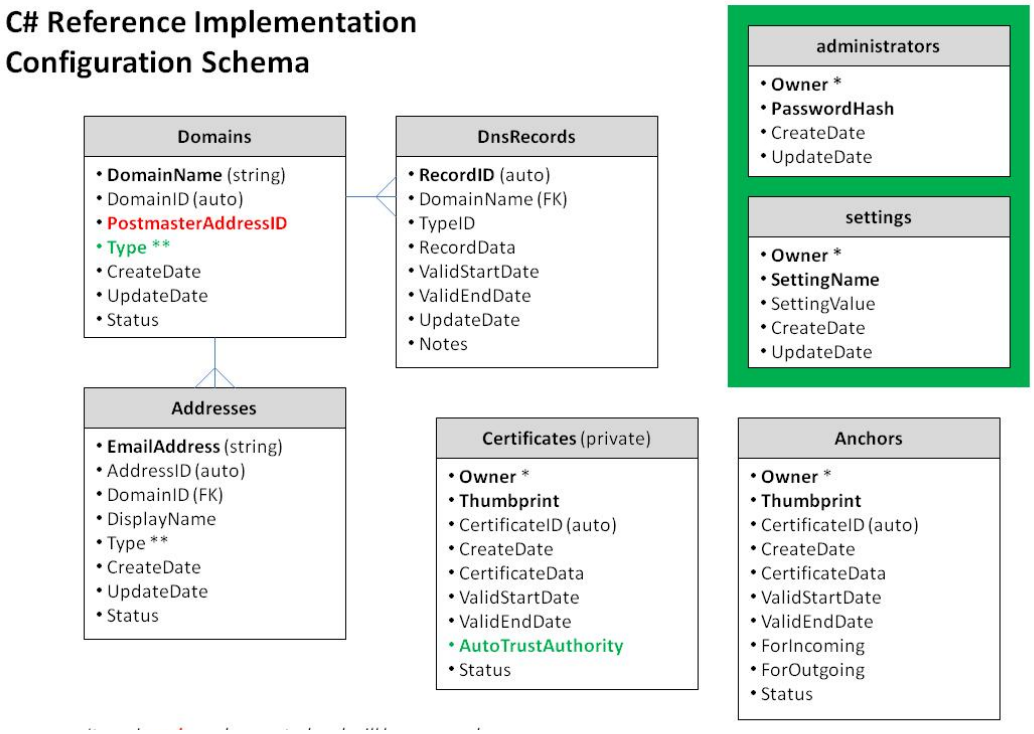


Figure 5: .NET Configuration Database Schema

Step 4: Web Services

At this point, Direct and IIS web service end points were configured and tested. This process connected the Direct configuration application services with the IIS web services. For example, connections were tested to ensure database connection string and DNS web services were functional. Familiarity and experience with application and web services is highly encouraged for this step.

Step 5: Managing Domains

Next, the domain name was set up for this installation. To do this, an email domain was needed to represent Direct addresses. For instance, my domain is named to represent electronic health (eHealth) in Tennessee. I used the domain name “TNeHealth” for Direct addresses in my organization. Anyone can acquire a new domain from a domain registrar service like Go Daddy.

It is also possible to use a sub-domain of a domain you already own. While any well formatted domain name can be used, the Direct Project workgroups recommend using a sub-domain prefixed with “Direct”. Using specific domains for Direct secure health information transport will identify common (unsecure) email address as distinct from Direct address (encrypted and secured). For instance, since I own TNeHealth.com, I used “Direct.TneHealth.com” for Direct addresses. This approach requires delegation of the sub-domain from your root DNS server to your Direct instance’s DNS server. This process is beyond the scope of this document, as it will vary depending on your domain infrastructure.

Step 6: Creating Certificates

Secure email X.509 certificates were now obtained and installed for the domain. There are many commercial suppliers for certificates through third-party “Certificate authority” security organizations such as Symantec or Entrust. Optionally, personally self-signed certificates can be used. However, since these certificates are not signed by an approved CA, the certificate will not automatically be trusted by other computers or people unless they add the self-signed certificate to their list of certificate authorities. Personally self-signed certificates are generally only useful for testing or for exchanging information with people you already know and trust. Either method will work with Direct and the choice of commercial versus self-signed digital certificates is determined more by policy than by technology.

Microsoft Windows Software Development Kit (SDK) provides a tool “Makecert.exe” that developers can use to generate their own email certificates. Opening a command

window to run the Makecert application, I was able to create the CA as well as the organizational identity certificate. These are then stored in the Windows certificate store.

Step 7: Configurations

As a final step in the server deployment, operational preferences were registered in the

```
<?xml version="1.0" encoding="utf-8" ?>
<!--
  SAMPLE CONFIG FILE
  Compliant with schema parsed by: NHINDirect.ScriptAgent.SmtpAgentSettings
  Customize as per your needs
-->
<SmtpAgentConfig>
  <Domain>redmond.hsgincubator.com</Domain>
  <Log>
    <DirectoryPath>C:\inetpub\logs</DirectoryPath>
    <NamePrefix>gateway__${date:format=yyyyMMdd}</NamePrefix>
    <RolloverFrequency>Day</RolloverFrequency>
    <Level>Debug</Level>
  </Log>
  <InternalMessage>
    <PickupFolder>C:\inetpub\mailroot\pickup</PickupFolder>
    <EnableRelay>false</EnableRelay>
  </InternalMessage>
  <Notifications>
    <AutoResponse>false</AutoResponse>
    <AlwaysAck>true</AlwaysAck>
    <Text>Message Delivery Notification</Text>
  </Notifications>
  <PrivateCerts>
    <MachineResolver>
      <Name>NHINDPrivate</Name>
    </MachineResolver>
  </PrivateCerts>
  <PublicCerts>
    <!--
    <DnsResolver>
      <ServerIP>127.0.0.1</ServerIP>
      <Timeout>5000</Timeout>
    </DnsResolver>
    -->
    <MachineResolver>
      <Name>NHINDExternal</Name>
    </MachineResolver>
  </PublicCerts>
  <Anchors>
    <MachineResolver>
      <Incoming>
        <Name>NHINDAnchors</Name>
      </Incoming>
```

Figure 6: Sample Direct Configuration File

Direct configuration services. Preferences and message process rules are registered with the Direct services that describe certain operations of the HISP (Figure 6). For example, gateway rules for certificate resolution must be established, as well as for message processing, audit logs, and message disposition notices.

Step 8: Connecting the Email Client

With the HISP configured, there must be a way for users to send and receive messages. There are various deployment models that could be considered. For this HISP, I chose to use a cloud-based email client linked to a full service HISP deployment model (Figure 7). This model allows the end system, the email client, to operate independently of the HISP where the end system outsources all the functions to a "full service" HISP. Microsoft Exchange Online 365 is a hosted messaging solution that delivers the capabilities of Microsoft Exchange Server as a cloud-based service. Exchange Online 365 allowed my deployment to take advantage of email client capabilities without the operational burden of on-premises application software. The use of Exchange Online 365 for this demonstration is not intended to endorse this service over other email services.

In this configuration, I placed the HISP in an Exchange Online instance. This way, incoming messages always pass through the HISP for decryption processing before they're handed to Exchange Online. The HISP enforces the Direct security model and forwards messages that pass the security and trust agent validation. Messages that fail validation are rejected by the HISP and are not forwarded to Exchange Online. Similarly, when messages are sent out of our HISP to another HISP, they always pass through the encryption processing. This ensures the messages are signed and encrypted using Direct protocols before it leaves the HISP. By design, all the difficult parts of the deployment of

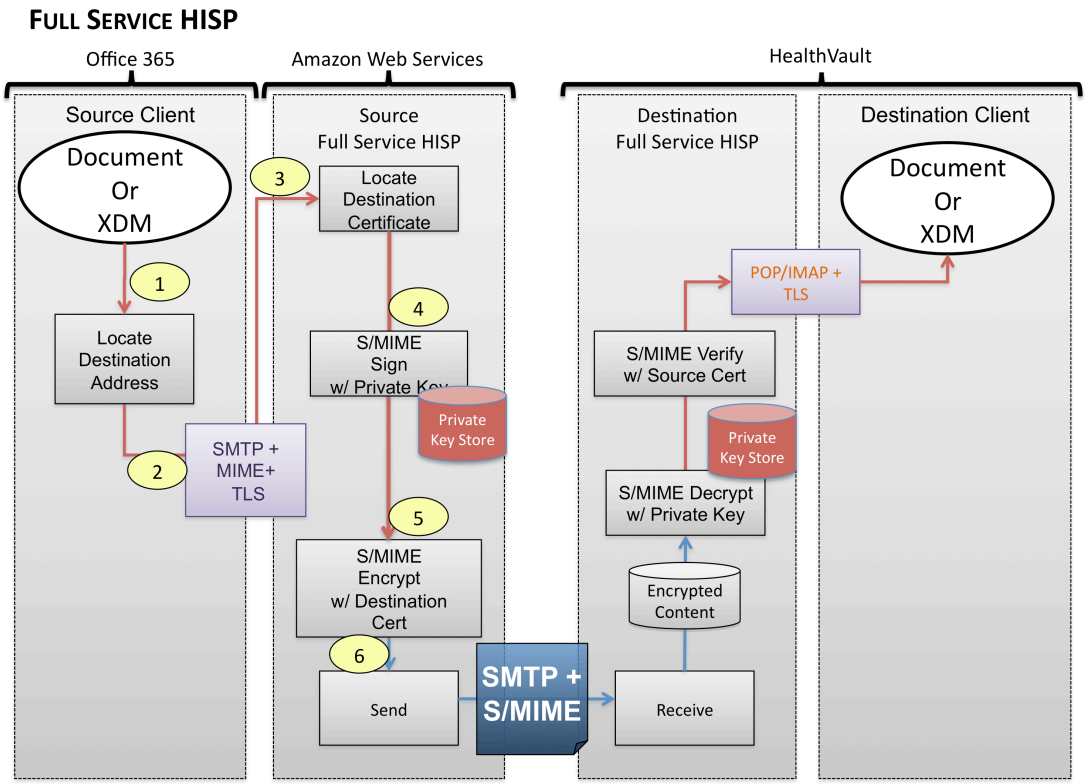


Figure 7: Email Client with Full Service HISP

public and private keys and the complexity of security are carried out behind the scenes by the external HISP service.

The email client must be configured to connect to the SMTP server hosted by the full service HISP in a manner that uses transport layer security (TLS). All other encryption is handled by the HISP. To configure the email client:

- Obtain the SMTP and POP3 or IMAP domains. Configure the client to use these domains for sending and receiving email.
- Configure the domains to use SSL for both incoming and outgoing messages. This will secure the information transmitted between your email client and the HISP. This step is required to protect personal health information passed in your

message from the point it leaves a computer to the point it reaches the HISP's email servers.

- Configure the HISP connectors to ensure that Exchange Online only accepts mail that has passed security validation.

The Direct Messaging specifications are common to all deployment models and this commonality allows the sender and receiver to choose very different deployment models. It is important to recognize that any deployment model can be used to send and any deployment model can be used to receive email. The deployment model of the sender is independent of the deployment model of the receiver. The adherence to the Direct Messaging specification assures that any deployment model can communicate to any other deployment model and, in fact, the receiver won't be able to tell which deployment model the sender is using.

Use Case: Provider Sends Patient Health Information to the Patient

During the development process, The Direct Project workgroups created an extensive but non-exhaustive list of use cases intended to capture what a user does or needs to do as part of his or her job function. These user cases helped define the functional criteria the Direct standards must provide. When these services are used by providers and organizations to transport and share qualifying clinical content, the combination of content and Direct Project-specified transport standards may satisfy some Stage 1 Meaningful Use (MU) requirements. In fact, the use cases were prioritized to ensure that Direct will support MU requirements (17).

Included in the use cases was the case of a provider using Direct Messaging for sending a clinical summary of an office visit to a patient, which supports certain Stage 1 MU requirements. The use case did not detail implementation specifics but instead focused on certain high level assumptions. It was envisioned that whatever Direct specifications were developed, its functionality must support the requirements of this use case.

To complete the objective of demonstrating Direct Messaging, I chose to implement this use case to showcase secure health transport using Direct Messaging. To illustrate the scenario, I first developed a flowchart that includes the user actions and decision (Figure 8). There are almost an endless number of possible actions that a provider can take based upon a variety of user settings, workflow preferences, and other unknown variables.

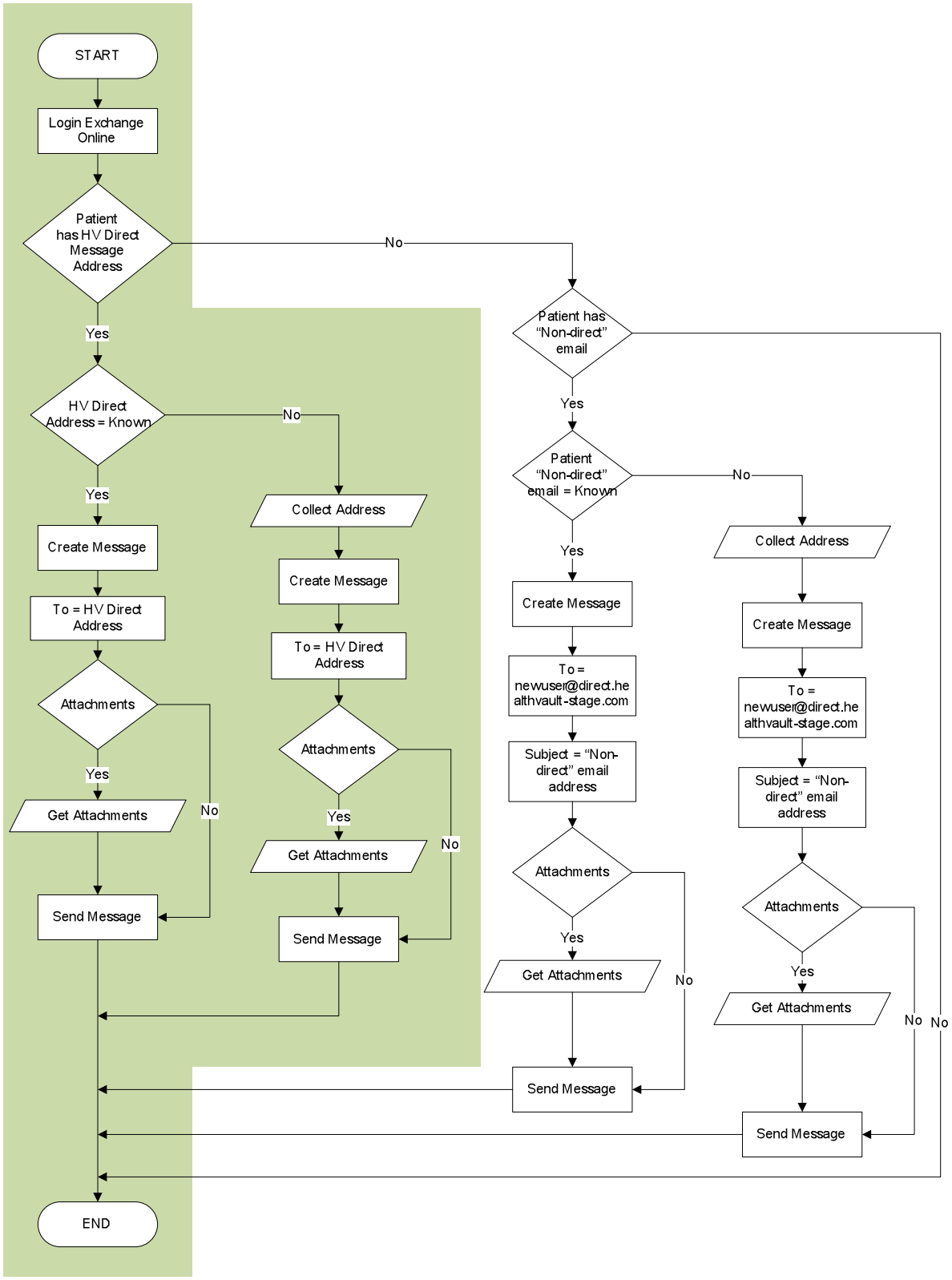


Figure 8: Provider Sending the Direct Message Flowchart

Alternative workflows were documented but the primary use case was based on typical or default settings. I chose to use Microsoft's HealthVault as a PHR to for demonstrating Direct Messaging interoperability. It should be noted that other PHR services have or are planning to incorporate Direct Messaging into their PHR platforms. The use of HealthVault for this demonstration is not intended to endorse this PHR service over other PHR services. A non-inclusive listing of personal health record companies that are planning to enable Direct to can be found on The Direct Project website (18).

In early 2011, HealthVault included Direct services in the PHR allowing a Direct-enabled clinical partner to send Direct Messages to patients who have a HealthVault account. HealthVault is an online PHR service that offers patients a central place to store and share personal health information. HealthVault Message Center allows patients to receive Direct Messages through their HealthVault accounts from participating providers. Every HealthVault account is given a Direct address to accept Direct emails from healthcare providers for the purposes of receiving health information from their providers into their HealthVault PHR account. Using Direct, an encrypted copy of a patient's clinical information is transmitted electronically to an email address the patient creates in HealthVault.

Through this functionality, a provider can create a copy of an individual's clinical information to be encrypted and electronically transmitted to a patient's new email address created within Microsoft HealthVault. Once received by the patient, it is automatically saved to the patient's HealthVault account as part of their longitudinal personal health record (Figure 9).

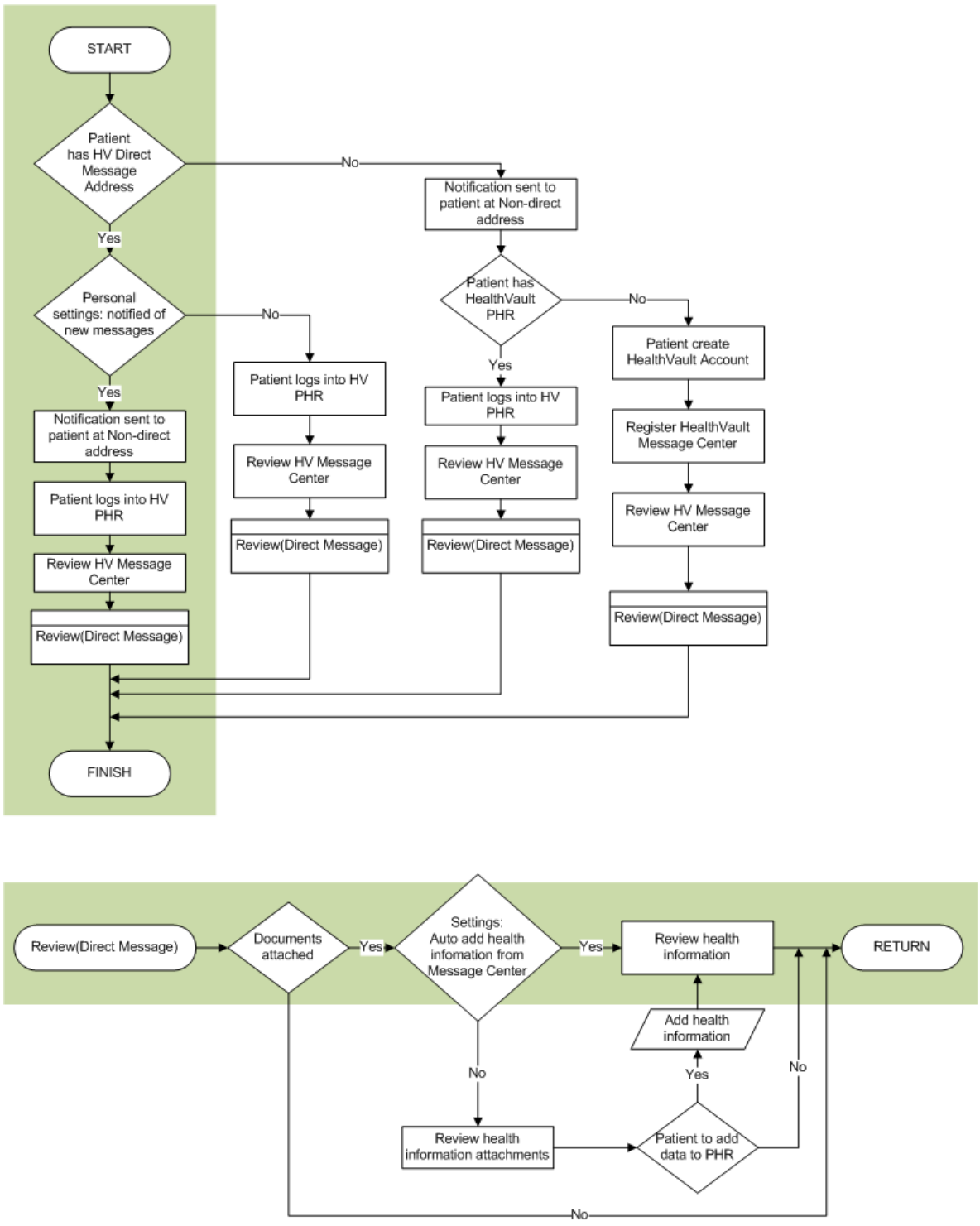


Figure 9: Patient Receiving the Direct Message Flowchart

User Story

Related to the use case, I developed a user story defining interactions between "actors" to demonstrate a provider sending a clinical summary of an office visit to a patient. Detailed step-by-step instructions for the current implementation can be found in Appendix A. The general steps involved are as follows.

- A primary care provider (PCP) wishes to update the patient's record with their personal clinical information. The PCP creates a Direct Message to the patient. The message includes the summary of care record (CCD) containing pertinent health information about the patient. The PCP sends the message to the address provided by the patient.
    - Step #1: Log into TNeHealth Direct Messaging client
    - Step #2: Create a message addressed to the patient's Direct email
    - Step #3: Add attachments of any type (OPTIONAL)
    - Step #4: Review the final message and click "Send"

The steps for a patient to receive a Direct Message in their HealthVault PHR are also described. There are almost an endless number of possible actions that a patient can undertake based upon a variety of user settings and preferences. Therefore the use case was based on typical or default HealthVault PHR settings. In general, the steps for the patient are as follows:

- The patient receives an email through their non-Direct ISP that their PCP has updated their PHR. The patient verifies that they own the Direct address,

authenticates into their HealthVault PHR and views their updated health information.

- Step #1: Patient will receive a notification email at their “non-Direct” email account.
- Step #2: Patient will login to HealthVault to review the Direct Message
- Step #3: Review the email message from the provider
- Step #4: Review clinical data in HealthVault patient health record

Once the message is sent (by the PCP actor) and received (by the patient actor) it helps to review exactly what the HISP performed. The HISP uses the following simplified algorithm when sending messages:

- The message is validated for all proper MIME content.
- The sender's private key is discovered, and a message signature is generated.
- The recipient's or recipients' certificates are discovered and the message and signature are encrypted into an S/MIME envelope.
- The encrypted message is sent to the recipients' HISP(s) using the message gateway.

Conversely, HealthVault uses the following simplified algorithm when receiving messages:

- The encrypted message is received by the message gateway and handed off to the security and trust agent.
- The recipient's or recipients' private keys are discovered, and the message is decrypted.

- The sender's certificate is discovered, and the message signature is verified.
- The sender's certificate is checked against the trust store to ensure the sender is from a trusted HISP.

A failure in any of these steps results in the message being discarded by either the HISP or the HealthVault PHR, and an appropriate action is taken. Reasons for the rejection could include:

- Malformed/non-S/MIME compliant messages
- Encryption/Decryption failed
- Signature creation/verification failed
- Certificate Resolution failed - such as over DNS
- Certificate from sender is not trusted by recipient
- Certificate is from an un-trusted source
- Target user address does not exist (in Config System)

In testing mode, the HISP can capture messages in the raw before and after the gateway processes them. This is very useful for debugging, but a HISP should disable this option on production since capturing these raw messages would expose the message and the contents to the HISP. However, by accessing the copies of these messages, we can confirm that the messages were processed and encrypted to meet the Direct specification. Before the HISP performs the message processing, it is clear that that the subject and the message of the email is in plain text and not encrypted (Figure 10).

Figure 10: Raw Message Sample (Not Encrypted)

After the HISP performs the message processing, it is clear that the message has been encrypted using S/MIME and the entirety of the email message is encrypted (Figure 11).

The reference implementation is a fully working model and its extensibility provides for any number of conceivable configurations. The reference implementation is not intended to be the final design that ultimately goes into your finished solution. It is tweaked and

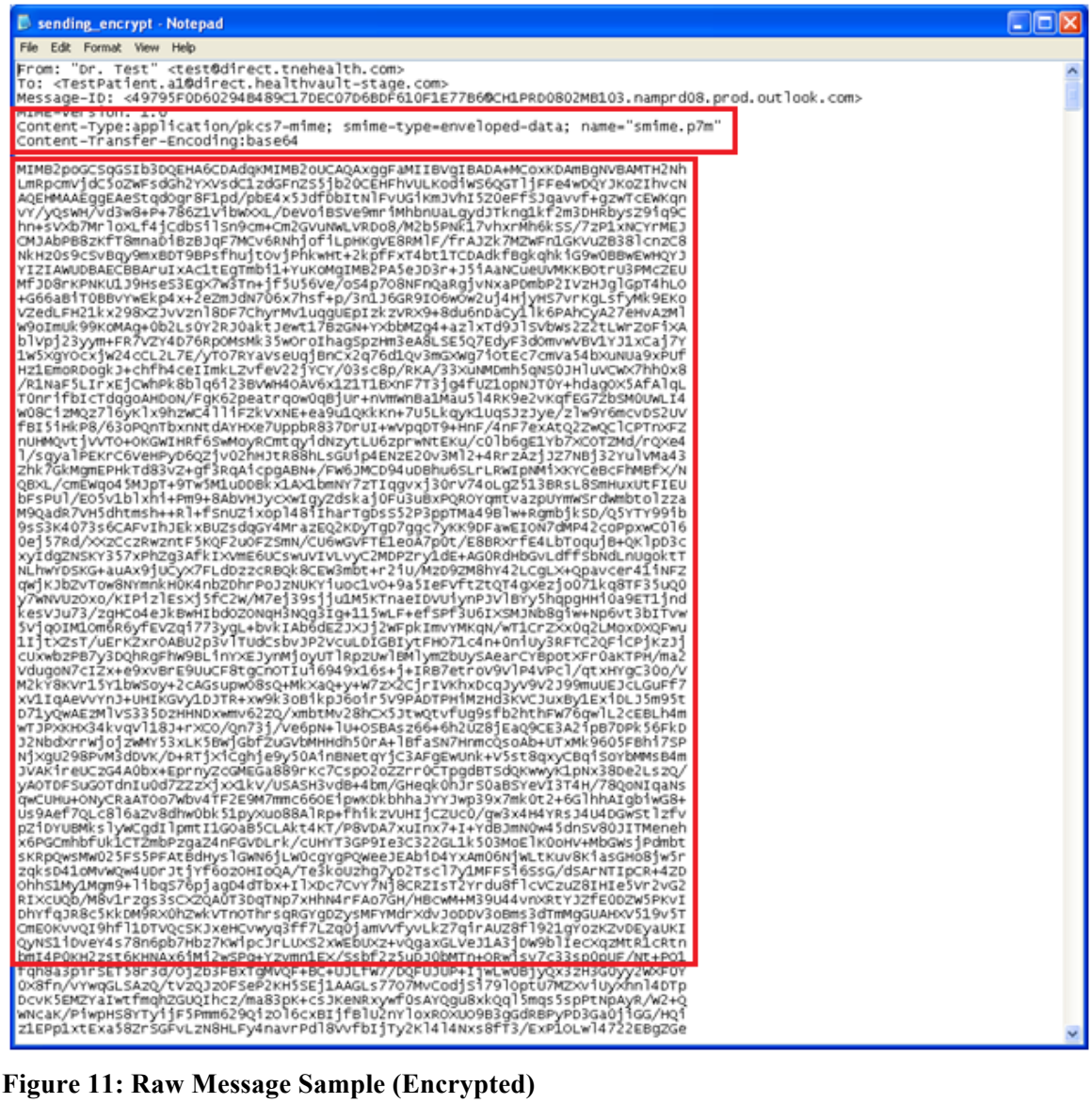


Figure 11: Raw Message Sample (Encrypted)

extended with custom modules, or some modules may be removed. The end product is a customized set of services and functions that meets the specific implementation needs.

The reference implementation comes with a standard deployment model and software components such as the security agent, the messaging gateway, a certificate store and a simple web or command line tool for configuration. However, the model does not meet the requirements of an industry-class production system, such as high availability, failover, scalability and disaster recovery. The main protocol supported is POP/SMTP,

and although this may work well for email clients, innovative edge clients and workflows may need other protocols such as REST or SOAP.

In addition, the reference implementation does not meet the security policy requirements emerging from the various governance agencies. For example, the private certificate store in the reference implementation does not meet auditing requirements for access to private keys. The auditing subsystem does not write audit events to a storage mechanism with proper access controls. For a HISP to become fully compliant with industry best practices, certificate policies and required operational procedures, investment in infrastructure and some software development is necessary.

Policy Implications for Direct

As I have demonstrated, the Direct Project seeks to benefit patients and providers by improving the transport of health information, making it faster, more secure, and less expensive. The Direct Project focuses on the technical standards and services necessary to securely push content from a sender to a receiver. However, in the real world, these technical standards should be implemented within a strong policy framework.

When a healthcare delivery organization or clinician decides to exchange data using Direct, there are a number of questions surrounding technical, operations and privacy policies that must be answered. These questions are raised by the exchange of clinical data using mechanisms provided by Direct, but their answers must be agreed upon by the users who exchange the data in order to ensure efficient, secure, and acceptable workflows for clinical care.

A Model for Trust

Trust is important to the confidence that the providers and patients will have in the privacy and security of Direct exchange. The Direct components describe a security agent that is responsible for encrypting and decrypting all outgoing and incoming messages. Arguably the most important aspect of the security agent is the trust model. The S/MIME encryption algorithms ensure a message is securely transported from one location to another without being compromised or tampered with, but what value is a message if you do not trust its sender?

Users may be able to attest to their identity assigned to them by their HISP using a certificate, but how does a receiving participant know that they can trust the entity that is

asserts the identity? In other words, if I receive a message from Dr. A, how do I know the public certificate that signed the message actually represents Dr. A, or that Dr. A is actually trustworthy? The basic uncertainty involves how two HISPs know to trust one another at a level sufficient to carry out the exchange of public keys linked to the digital certificates deployed, or in more general terms, how a HISP becomes trustworthy to other HISPs.

As a rule of thumb, only HISPs that follow and can prove to abide by good certificate practices and identity proofing procedures should be deemed trustworthy (10). A HISP should only allow the creation of addresses for participants that they deem trustworthy. This leads to the subject of PKI, identity proofing and certificate authority.

Public Key Infrastructure

PKI provides the foundation for interoperable trust among parties who wish to exchange online. PKI is more than just technology related to digital certificates, it is the architecture, organization, techniques, policies, practices, and procedures that collectively support the implementation and operation of a certificate-based public key cryptographic system. Also included in PKI are the policies and other contract agreements among parties that document the operating rules, procedural policies, and liabilities of the parties operating within the particular PKI.

In the discussion of the security agent, I explained that a digital certificate binds a public key to identity information for an individual or organization. Individuals and organizations use the digital certificates to identify themselves in electronic transactions. When an individual digitally signs a transaction using the certificate, the relying party can

verify the individual's signature and query the CA to ensure that the certificate is valid. If both are valid, the relying party can trust that the individual signing the message is who they say they are.

Identity Proofing

The identity verification issue is very important to signal the trustworthiness of the HISP and its users to other HISPs. How do any of the parties relying on the authenticity and validity of the information in these digital certificates know when to trust each other, and perhaps when not to trust someone? Different HISPs may have different identification policies and will, therefore, be trusted differently by other HISPs. One HISP might insist on seeing a driver's license or passport, while another might want the certificate request form to be notarized, or include the fingerprints or other biometric proof-of-identity of the requesting party.

Identity proofing is the process of assuring that an entity really is what or who it says it is. The proofing of Direct users before issuing a digital certificate and Direct address is a very important core competency of a HISP. This aims to ensure that only legitimate users of Direct are given access. Providers and healthcare organizations should be verified through multiple approaches (i.e. in-person, electronic two-factor or a Notary). Based on their verification, some or all of their information should be verified against an independent source (such as a state licensing board or a federal database) before a Direct address is issued. The basis for the trust is the digital certificate issued by a trusted third party, the CA.

Certificate Authorities

It is very important to understand the role CAs play in issuance and management of identity credentials such as digital certificates, and how policy decisions made by these parties can make it easier or more difficult for Direct exchange to proceed on the HISP-HISP level. Different CAs have different identification policies and will, therefore, be trusted differently by other CAs. One CA might insist on seeing a driver's license or passport, while another might want the certificate request form to be notarized, or include the fingerprints or other biometric proof-of-identity of the requesting party. CAs can publish identification requirements and standards so that other verifying CAs can attach the appropriate level of confidence in the certified name-key bindings. CAs with lower levels of identification requirements produce certificates with lower levels of identity assurance. CAs can be considered to be of high, medium, and low assurance, and the assurance level is often a way to select one CA over another. Additionally, CAs must have operational procedures for renewing and revoking certificates.

Overall, the trust model provides a great deal of flexibility in determining trust relationships between HISPs and/or individual participants. Because messages are signed using the sender's X509 certificate, PKI algorithms "filter" messages based on entities called trust anchors. Every X509 certificate is issued by a CA, and a CA can be used to validate the authenticity of the issuer of an X509 certificate. In the simplest case, a CA is a trust anchor. If a HISP trusts a particular trust anchor, then all certificates created by that anchor are considered to be trusted. In order for subscribers of different HISPs to be assured of seamless, secure sending and reception of messages, trust must be a known and transparent feature. Unless HISPs trust one another, Direct exchange is not

interoperable. HISPs must know not just the technical specifications required for Direct exchange, they must know how to trust one another in an automated fashion.

Discussion and Conclusion

Electronic health information exchange addresses a critical need in the US health care system to have information follow patients to support patient care. The Office of the National Coordinator for Health Information Technology has led the process of establishing the essential building blocks of NwHIN that will support multiple models of health information exchange. ONC led the Direct Project to create a set of specifications and standards that specifies a simple, secure, scalable, and standards-based method for participants to send authenticated, encrypted health information directly to known, trusted recipients over the Internet.

In order to demonstrate interoperable health information exchange using Direct Project specifications, a fully functional HISP, was created in this project. Even with a "jump start" from the Direct Project reference implementation, this was not a clear and straightforward process. Once operational, framed within the context of a real-world use case, I was able to demonstrate secure health transport using Direct Messaging to enable provider to patient health information exchange. In addition to exploring technical specifications of Direct by deploying a fully functional HISP, this project also served as a vehicle to explore various policy issues related to security and identity proofing inherent in the PKI model used within Direct Messaging.

Although considerable progress is being made in launching Direct implementations across the country, much about the role of a HISP and some of a HISP's future functionality remains untested in the marketplace. This demonstration can serve as a tool

to help raise industry awareness and understanding about Direct, in hopes that best practices that will be adopted universally to assure Directed exchange scales.

It is believed by many that consumers with access to their own health information can improve the effectiveness and coordination of their own healthcare by sharing information with other providers, identifying potential medical errors, correcting inaccurate health and billing information, and making more-informed decisions (3). This provider and patient use case demonstrates not only the technical feasibility of using Direct Messaging, but also illustrates the opportunity to put necessary tools in the hands of patients by giving them ready and secure access to their own electronic health information, which they can use and share to improve their health and make better health care decisions in partnership with providers.

Bibliography

1. American Recovery and Reinvestment Act of 2009, Pub L. 111-5, 11th Cong., 1st Sess., (January 6, 2009).

2. Wikipedia, The Free Encyclopedia. Health Information Exchange [Internet]. 2011 [updated 2011 May 9; cited 2012 April]. Available from: http://en.wikipedia.org/wiki/Health_information_exchange.

3. Williams C, Mostashari F, Mertz K, Hogin E, Atwal P. From the Office of the National Coordinator: the strategy for advancing the exchange of health information. Health Aff. 31, no.3 (2012):527-536.

4. The Patient Protection and Affordable Care Act, Pub L. 111-148, 111th Cong., (Mar 23, 2010).

5. Robert Wood Johnson Foundation. Health Information Technology in the United States: Moving Toward Meaningful Use; 2010.

6. U.S. Department of Health & Human Services, The Office of the National Coordinator for Health Information Technology. The Nationwide Health Information Network [Internet]. 2012 [updated 2011 March 8, cited 2012 May]. Available from:. http://healthit.hhs.gov/portal/server.pt?open=512&mode=2&objID=3340.

7. U.S. Department of Health & Human Services, The Office of the National Coordinator for Health Information Technology. Direct Project [Internet]. 2012 January. Available from:. http://healthit.hhs.gov/portal/server.pt/community/healthit_hhs_gov__direct_project/3338

8. HIMSS Electronic Health Record Association. Supporting a Robust Health Information Exchange Strategy with a Pragmatic Transport Framework; 2011. Available from: http://www.himssehra.org/docs/20110629_EHRA_TransportFramework_Final%20.pdf.

9. The Direct Project wiki. Applicability Statement for Secure Health Transport [Internet]. 2011 [cited 2012 January]. Available from: http://wiki.directproject.org/Applicability+Statement+for+Secure+Health+Transport.

10. The Direct Project wiki. The Direct Project Overview [Internet]. 2011 [cited 2012 March]. Available from: http://wiki.directproject.org/The+Direct+Project+Overview.

11. The Direct Project wiki. Deployment Models [Internet]. 2011 [cited 2012 May]. Available from: http://wiki.directproject.org/Deployment+Models.

12. Internet Message Format. P. Resnick, Ed. October 2008. (Format: TXT)(Status: DRAFT STANDARD).

13. Wikipedia, The Free Encyclopedia. MIME [Internet]. March 24, 2012 [cited 2012 April]. Available from: http://en.wikipedia.org/wiki/MIME.

14. Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification. B. Ramsdell, S. Turner. January 2010. (Format: TXT)(Status: PROPOSED STANDARD).

15. Wikipedia, The Free Encyclopedia. X.509 [Internet]. April 12, 2012 [cited 2012 April]. Available from: http://en.wikipedia.org/wiki/X.509.

16. The Direct Project wiki. Reference Implementation Workgroup [Internet]. 2011 [cited 2012 January]. Available from: http://wiki.directproject.org/Reference+Implementation+Workgroup.

17. The Direct Project wiki. User Stories [Internet]. 2011 [cited 2012 April]. Available from: http://wiki.directproject.org/User+Stories.

18. The Direct Project wiki. Ecosystem [Internet]. 2011 [cited 2012 January]. Available from: http://wiki.directproject.org/Ecosystem.

Appendix: Step-by-step Direct Messaging Demonstration Instructions

Story: Provider Sends Patient Health Information to the Patient

A primary care provider (PCP) wishes to update the patient's record with their personal clinical information. The PCP creates a Direct Message to the patient. The message includes the summary of care record (CCD) containing pertinent health information about the patient. The PCP sends the message to the address provided by the patient.

Actor (Provider): The PCP has validated the patient's identity and has the patient's Health Internet Address.

Actor (Patient): The patient verifies that they own the Direct address, authenticates into HealthVault personal health record and views their updated health information.

Details & Assumptions

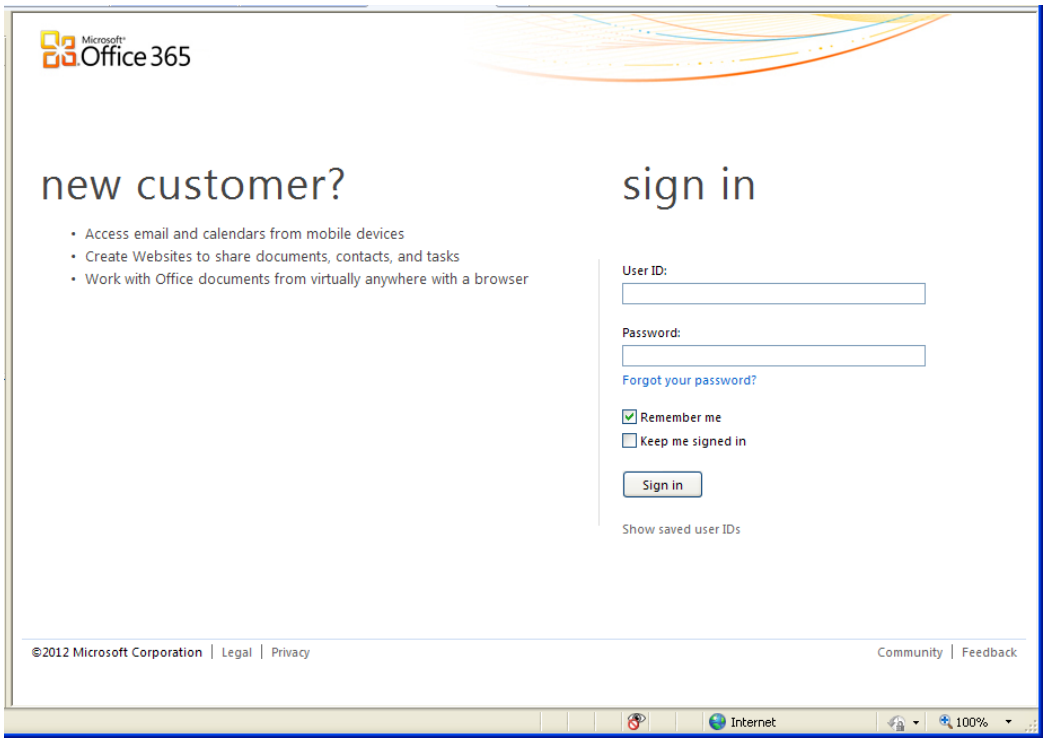
- Actor (Provider): Create TNeHealth.com Direct messaging account
    - Typically, before a user account is provisioned, it is assumed that the user would have authenticated through a multi-factor validation process before being provisioned with a Direct Message account. For the demonstration, we will assume an approved user validation and a user account will be created. The user will receive a "New user" email with instructions to sign in and change the temporary password.
    - Log into TNeHealth Direct messaging client (Microsoft Office 365 Exchange Online)
        - http://mail.office365.com
        - You will receive an email from Microsoft Online Services indicating that "A user account has been created". Please follow the login instructions to begin using your direct.tnehealth.com Direct Message account.
            - Enter your user name and corresponding temporary password.
            - Follow the instructions on the sign-in page to create a new password.
- Actor (Patient): Patient's HealthVault Direct address and account.
    - Patient HealthVault Direct address
        - Name: Test Patient
        - Direct email address: TestPatient.a1@direct.healthvault-stage.com
    - Patient HealthVault personal health record account
        - HealthVault message center (Pre-production environment)
        - http://direct.healthvault-stage.com
        - Username: testpatient.a1@gmail.com
        - Password: patienta1

- Actor (Patient): Patient non-Direct email account
  - Google account
    - http://mail.google.com
  - Username: testpatient.a1@gmail.com
  - Password: patienta1
- Actor (Provider): Access to clinical document(s)
  - Certified EHR capable to create/export patient summary record (CCD)
    - John Halamka, the CIO of Boston based Beth Israel Deaconess Hospital posted his entire lifelong medical history in continuity care document (CCD) that can be downloaded and used as a sample (http://services.bidmc.org/geekdoctor/johnhalamkaccddocument.xml)

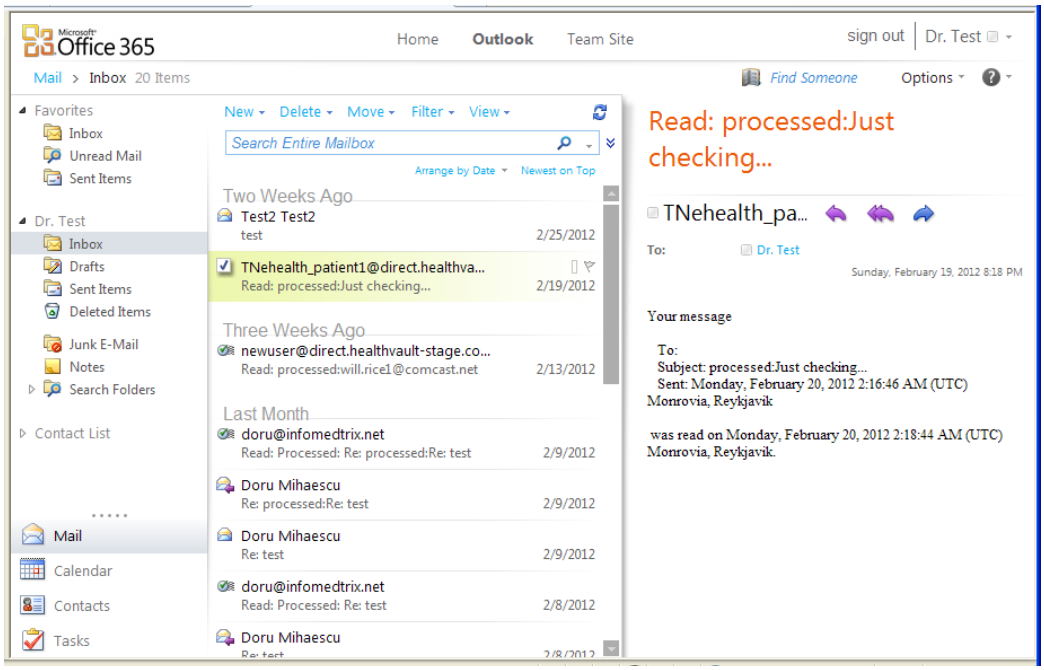
Demonstration (Provider Sending the Direct Message)

Step #1

- Log into TNeHealth Direct messaging client (Microsoft Office 365 Exchange Online)
  - http://mail.office365.com



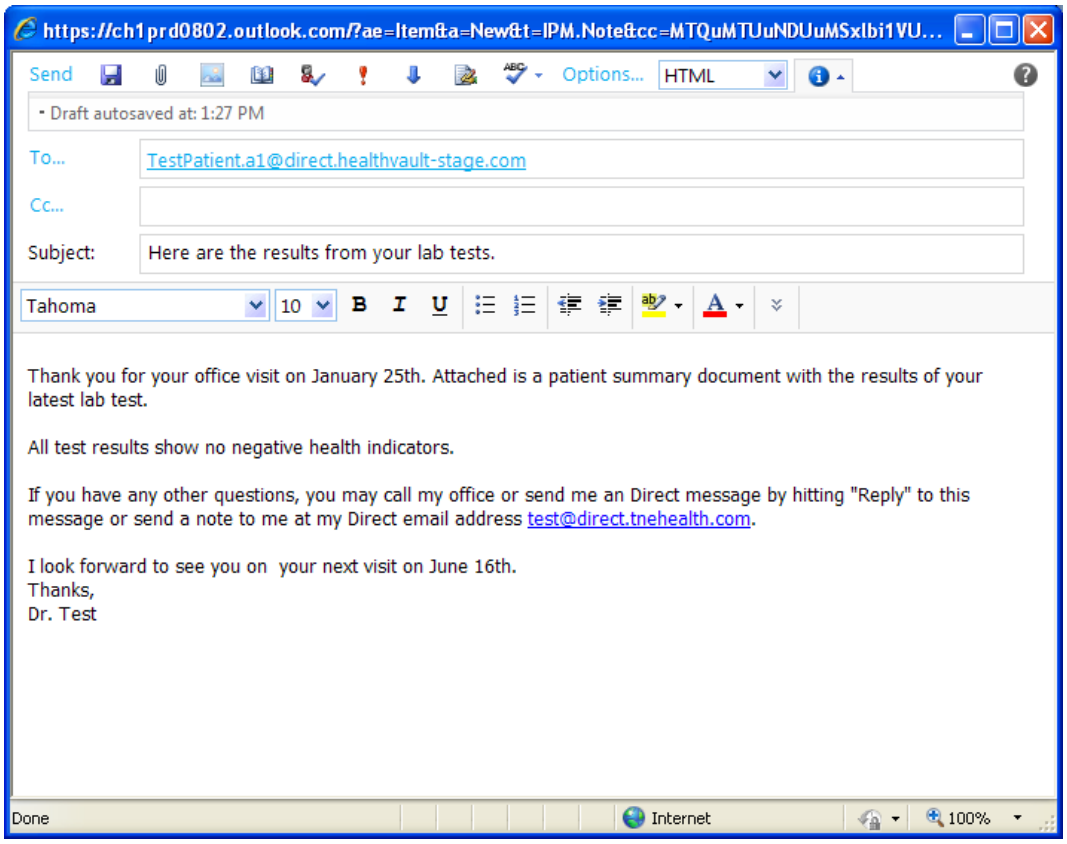
Login page



Direct messaging email client

Step #2

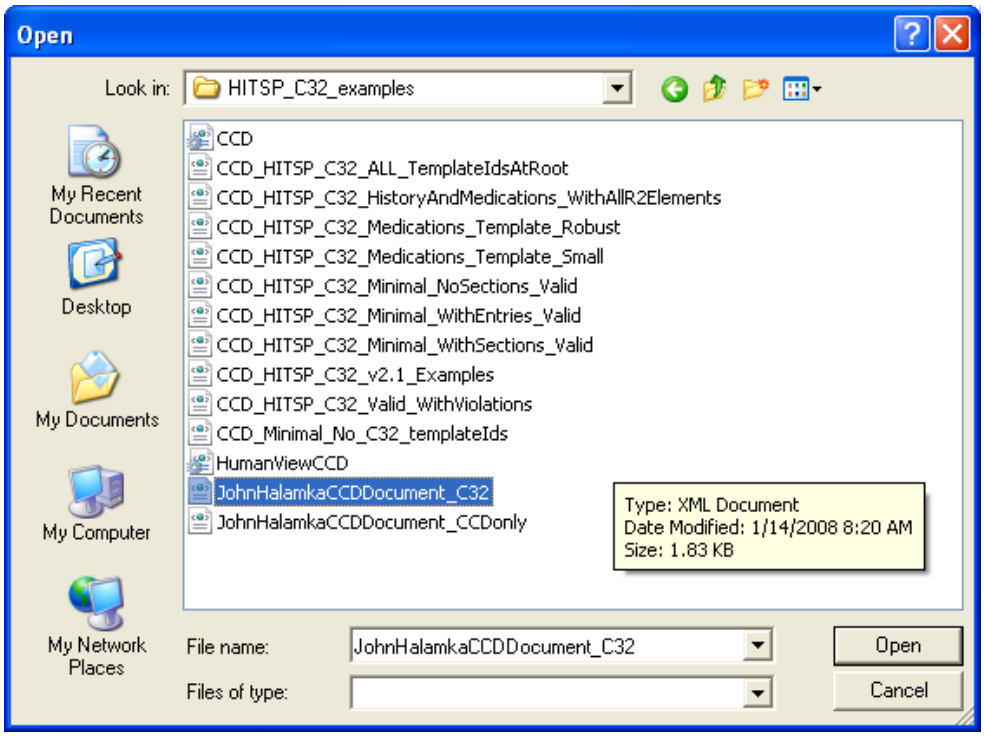
- Type a message addressed to the patient's Direct email.
    - How do you get a patient's direct address? Just ask the patient for their Direct address like you would ask for their phone number or mailing address. Usually, this is best done during the registration process, but it could happen anywhere in the workflow. If you already have a patient management system, it is likely that you already capture a personal "Non-Direct" email address for the patient. You can record a new demographic field to hold the Direct address.
- Address message to Test Patient HealthVault Direct Message address
    - Select "New > Message"
    - In the "New Message" window, use the Direct email address in the "To:" field
        - To: TestPatient.a1@direct.healthvault-stage.com
    - OPTIONAL: include a title or related message in the "Subject:" field
        - "Subject: Here are the results from your lab tests."
    - OPTIONAL: include a letter or related message in the "Body"



Message to patient

Step #3

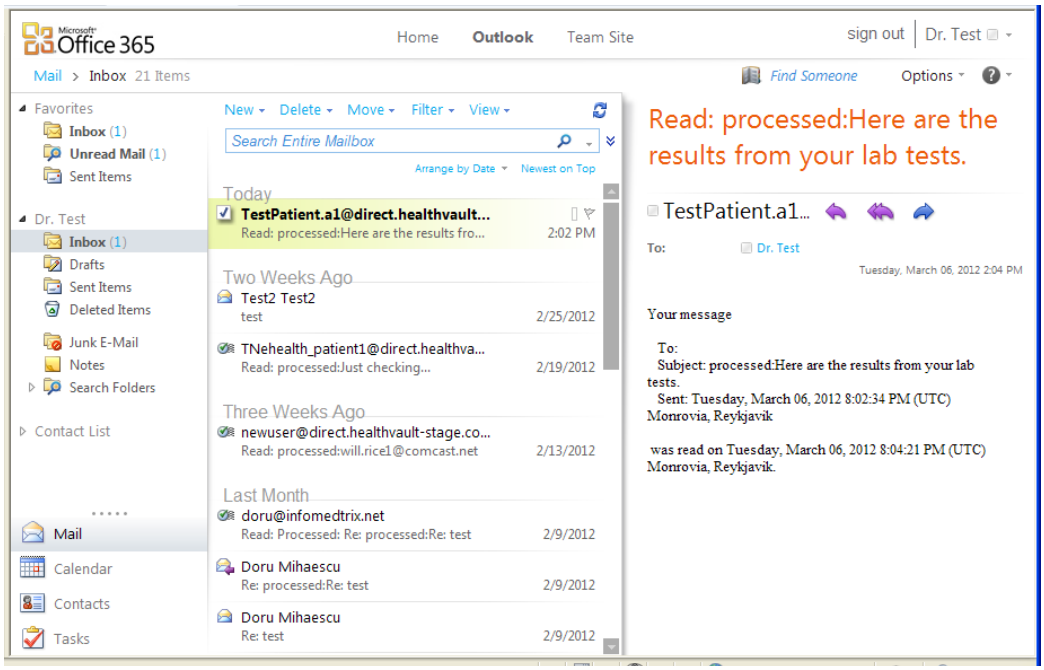
- OPTIONAL: Add attachments of any type
  - Contents can be structured or unstructured such as:
    - HL7 lab results
    - CCD – continuity of care document
    - CCR – continuity of care record
    - JPEG, PDF, TIFF
  - Click the “Add Attachment” Paperclip Icon
  - Locate the document(s) and click “Open” to attach to the Direct Message



Attaching a sample CCD document to the message

Step #4

- Review the final message and click "Send"
- By design, all the hard parts of the Direct deployment of public and private keys are carried out for Direct exchange subscribers behind the scenes by their Health Information Service Provider (HISP). The prime advantage of this deployment model is to move the complexity of security to an external service, the HISP. Your message went through the TNeHealth.com HISP:
    - Parsed the outgoing test message
    - Resolved the sender's (test@direct.tnehealth.com) private key
    - Resolved the anchors the sender trusts
    - Resolved the recipient's (TestPatient.a1@direct.healthvault-stage.com) public X509 Certificate
    - Verified that the recipient's certificate is both valid and trusted - i.e. issued by an anchor the sender trusts.
    - Created appropriate S/MIME envelopes
    - Signed the message with the sender's private key
    - Encrypted the message with the recipient's certificate
    - Produced a new, secure email message
- If successful, you should receive a Message Disposition Notification (MDN) indicating that the message was successfully processed and delivered



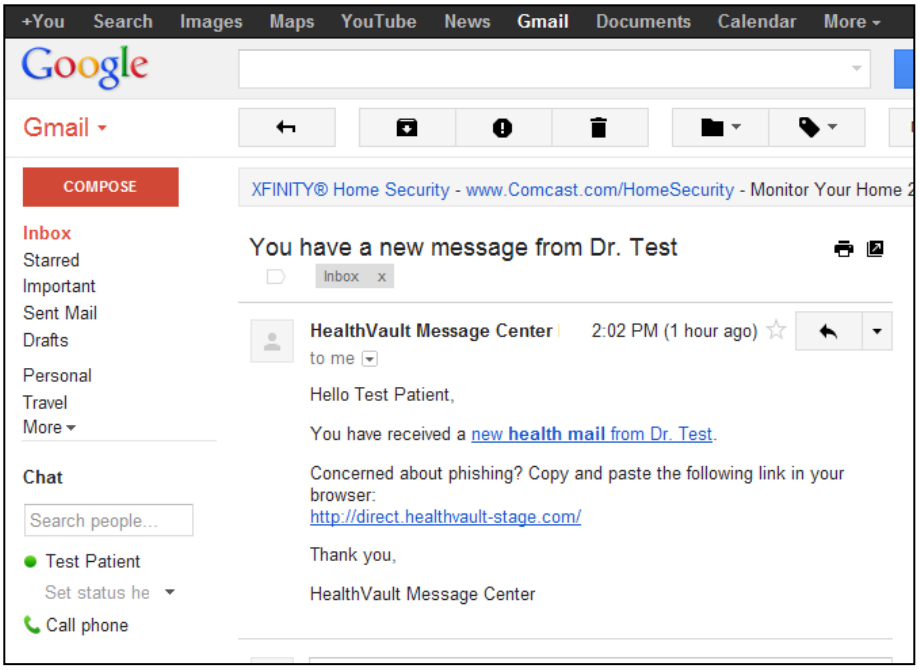
Successful Message Disposition Notification (MDN)

Demonstration (Patient Receiving the Direct Message)

There are almost an endless number of possible actions that a patient would undertake based upon a variety of user setting and preferences options. The steps in this demonstration and in the flowchart represent the actions based on the default settings when the patient first established the HealthVault personal health record account.

Step #1

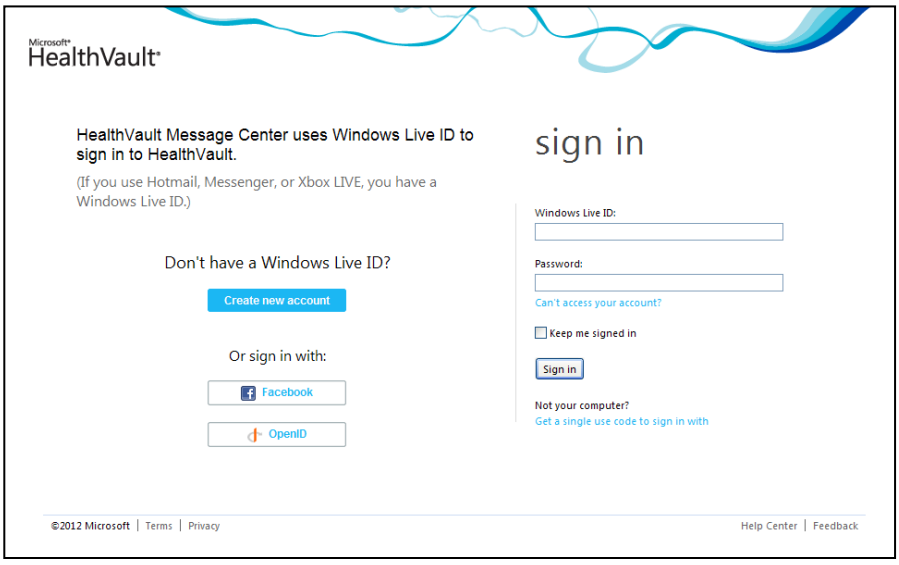
- Patient will receive a notification email at their "Non-Direct" email account.
  - http://mail.google.com
  - Username: testpatient.a1@gmail.com
  - Password: patienta1



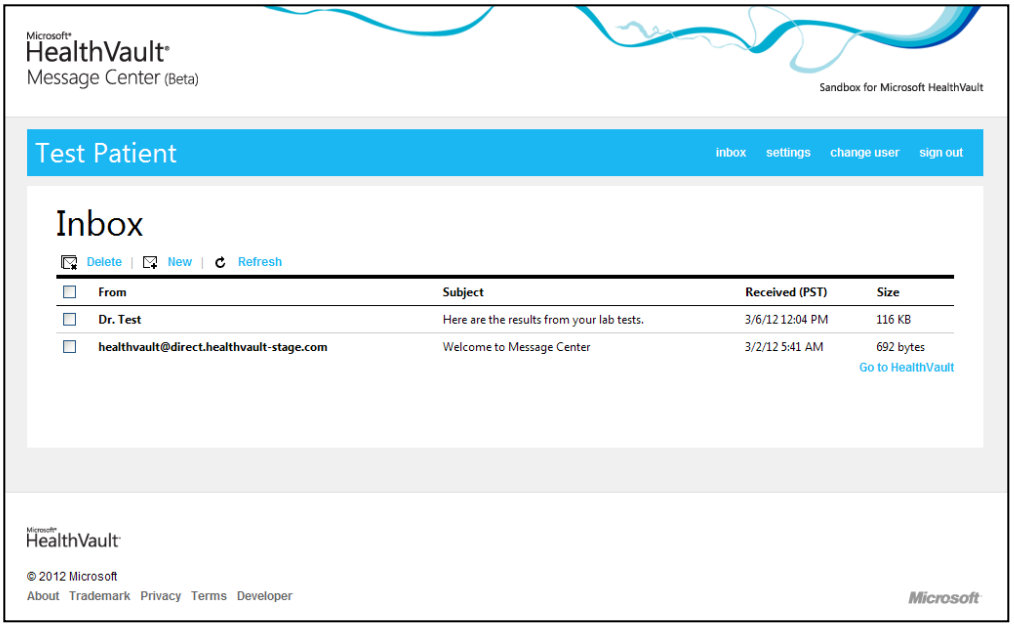
Notification email from HealthVault

Step #2

- Patient will login to HealthVault to review the Direct Message
  - Click the link in the email or access the HealthVault message center (Pre-production environment)
  - http://direct.healthvault-stage.com
  - Username: testpatient.a1@gmail.com
  - Password: patienta1



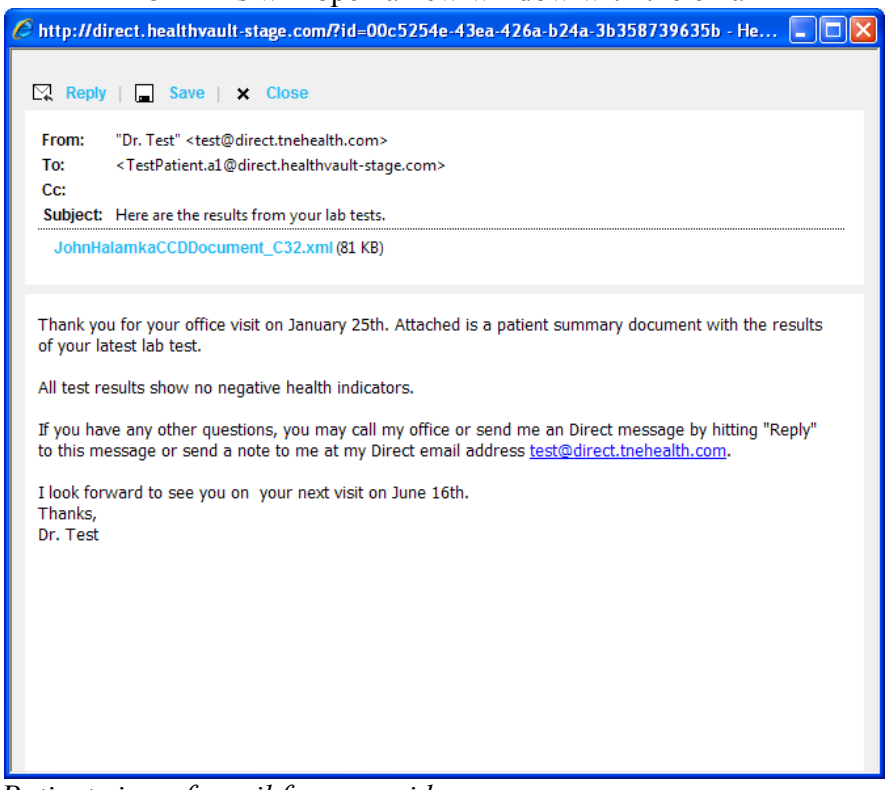
HealthVault message center login screen



HealthVault message center

Step #3

- Review the email message from the provider you wish to view
  - Click anywhere on the row with the email
  - This will open a new window with the email

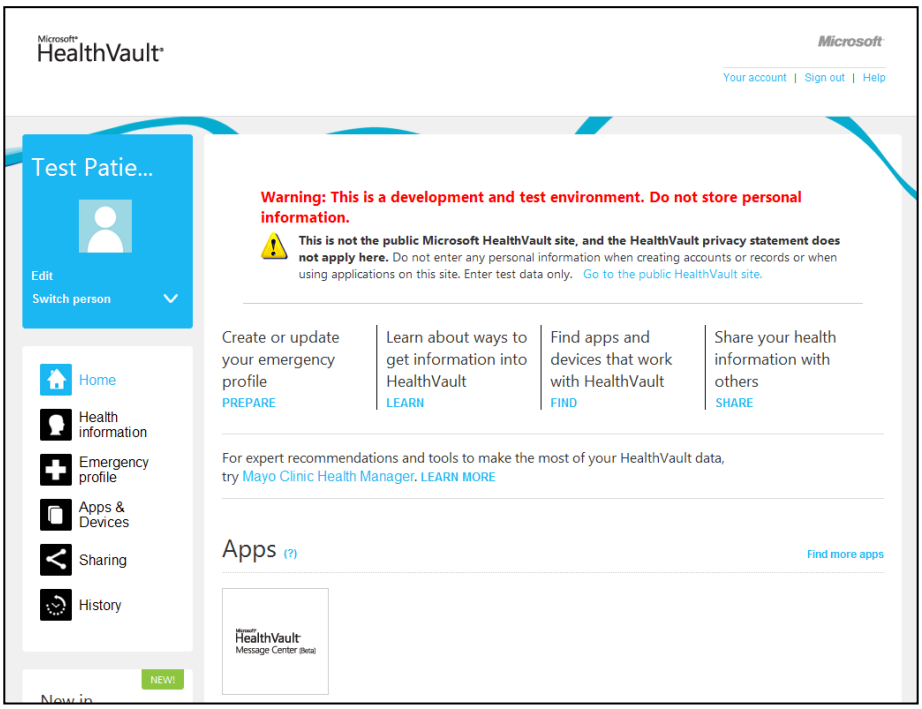


Patient view of email from provider

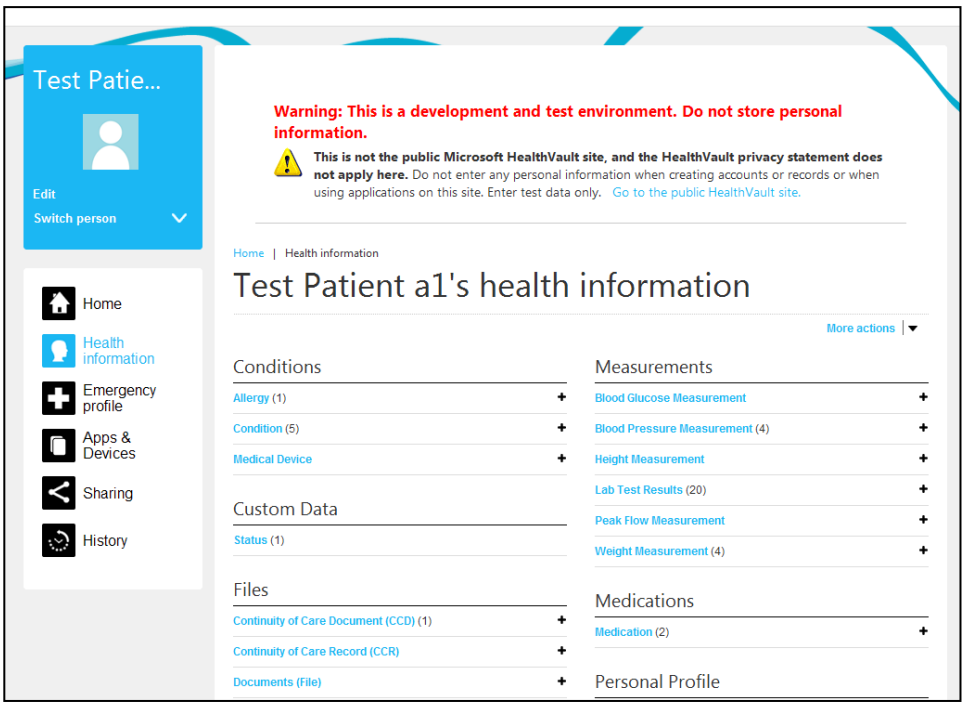
Step #4

Reviewing the email does indicate that a clinical document was attached to the email. The document that the provider attached in Step #3 is attached to the patient email. While user setting may vary, by default, the message center will detect that a clinical document was received and will auto-reconcile the clinical data from the document into the HealthVault patient health record

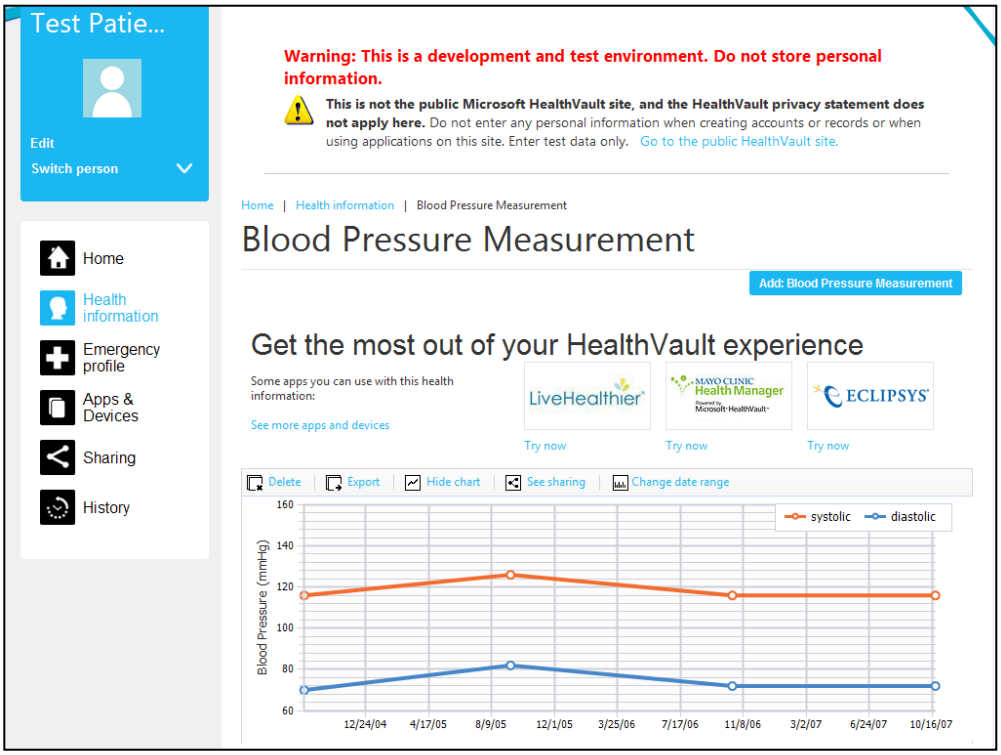
- Review clinical data in HealthVault patient health record
    - Close the email to return to the message center
    - Click the link "Go to HealthVault" located at the bottom right of the message center
        - This will open a new window to the HealthVault home page (Image: HealthVault Home page)
    - Click the "Health Information" button from the navigation
    - Click on any link to review the data (Image: Test Patient Health Record Information)
        - Patient can easily identify what data is present by the number presented by each information type
    - Click "Blood Pressure Measurement (4)"
        - Patient can review the data recorded for Blood Pressure (Image: Blood Pressure Measurements with chart)
        - OPTIONAL: Patient can add/edit/delete blood pressure measurements
        - OPTIONAL: Patient can chart the blood pressure measurements graphically
            - Click the "See Chart" link



HealthVault home page



Test Patient health record information



Blood Pressure Measurements (With Chart)

Step #5

The personal health record now contains helpful information that can empower the patient to better track and manage their own health.

- Patient can continue to review different health information
    - Click the “Health Information” button from the navigation to return to information page
- Patient can return to the message center
    - Click the “Home” button from the navigation to return to the home page
    - Click the “Open” link next to the HealthVault Message Center App to return to the messages
- Patient can sign out
    - Click the “Sign out” link at the top of the page.