EMR Certification

eHealth_hub - Authentication Specification

March 31, 2020

Version 1.0

Shared health Soins communs Manitoba

Table of Contents

List of Tables

List of Figures

1 Introduction

The purpose of this document is to provide an overview of eHealth_hub - Authentication. The document will describe the high-level details and authentication requirements used by all eHealth_hub services.

1.1 Glossary

The following table defines key business terms and acronyms used throughout this document.

Table 1: Terms and Acronyms

TERM OR ACRONYM	DEFINITION
Certificate Authority (CA)	Certificate Authority (CA) is an entity that issues digital certificates.
Digital Certificate	Small data file that digitally binds a cryptographic key to an organization's details.
Digital Signature	A mathematical scheme for verifying the authenticity of digital messages or documents. It is a type of electronic signature that encrypts documents with digital codes to achieve security.
eHealth_hub	A Shared Health business service that coordinates electronic delivery of information between systems to authorized health-care providers who are using a Manitoba Certified EMR.
EMR instance	A single distinct EMR entity that uses network to securely connect and access authorized eHealth_hub services.
Health Information Access Layer (HIAL)	Shared Health's core integration service which provides the environment and capabilities necessary to establish interoperability between systems. For example, HIAL is a Shared Health Service that enables Business Services such as eHealth_hub.
Manitoba Certified EMR Product (EMR)	An Electronic Medical Record product that has achieved Certification in Manitoba, and is referred to throughout this document as "EMR".
Public Key Infrastructure (PKI)	PKI enables secure communication on public networks through public and private keys provided by Certificate Authority.
Simple Object Access Protocol (SOAP)	SOAP is a messaging protocol that allows programs that run on disparate operating systems (such as Windows and Linux) to communicate using Hypertext Transfer Protocol (HTTP) and its Extensible Markup Language (XML).

Transport Layer Security (TLS)	TLS is a cryptographic protocol used for secure communication over a network.

1.2 Business Objectives & Benefits Summary

The objective of the eHealth_hub – Authentication specification is to provide a central location that describes the authentication requirements and supporting information necessary to interact with any of the eHealth_hub components. All eHealth_hub component specifications will contain a reference to this document.

2 eHealth_hub - Authentication Overview

eHealth_hub is a Shared Health business service that coordinates electronic delivery of information between systems to authorized health-care providers who are using a Manitoba Certified EMR.

All transactions between an EMR and eHealth_hub require the use of a secure mutual authentication connection for sharing information. This secure connection is achieved through the use of digital certificates. The same certificates are used for authentication and encryption.

Each EMR instance will require a unique client certificate, issued by Shared Health, to provide its own identity and authenticity to the eHealth_hub service. Equally, the eHealth_hub service will use its own unique certificate to provide identity and authenticate to the EMR instance. Note that certificates, while typically issued for each client instance, may also be issued on a per logical instance basis.

Once authentication is established, an EMR instance will require provisioning in order to access eHealth_hub service components.

2.1 Bi-Directional or Mutual Authentication

Bi-Directional or mutual authentication is a technique used to authenticate and establish a secure connection between a client and a server and vice versa at the same time. To establish a secure connection, the mutual authentication between the client and eHealth_hub service is achieved by the use of public key technology based on certificates from a trusted PKI Certificate Authority.

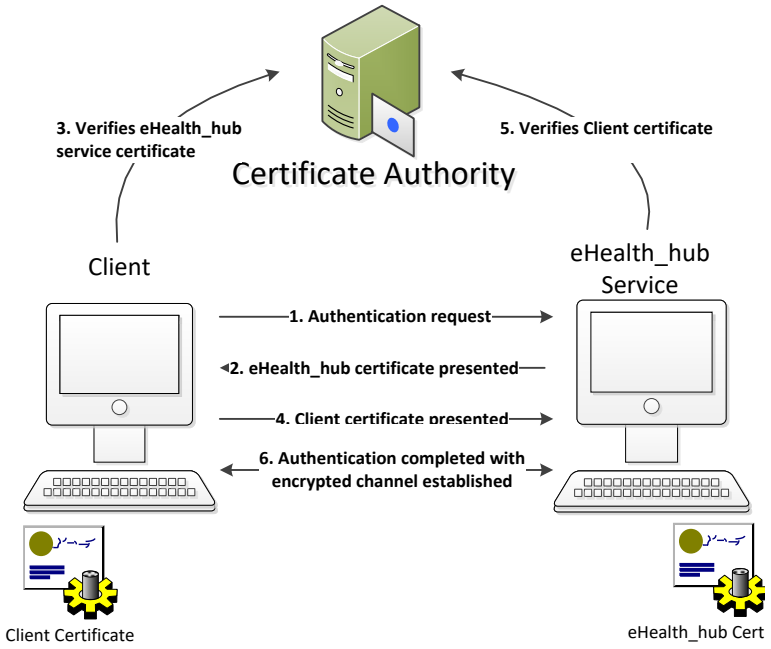

FIGURE 1: eHealth_hub - Authentication

3 Authentication Use Case

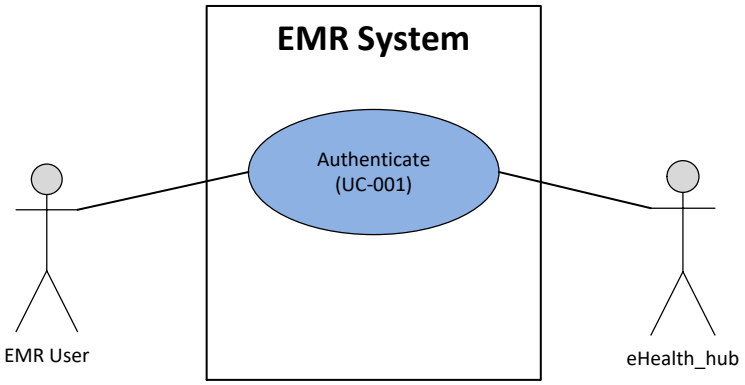

FIGURE 2: eHealth_hub - Authentication Use Case

3.1 Authenticate (UC-001)

This use case describes how a request is authenticated to use the eHealth_hub.

PRIMARY ACTORS
• eHealth_hub • EMR • EMR User
TRIGGER(S)
• EMR has a need to communicate with an eHealth_hub service through either an EMR User or an automated process
PRE-CONDITIONS / INPUTS
• EMR has been provisioned to use an eHealth_hub service
POST-CONDITIONS / OUTPUTS
• Successful authentication with an established secure connection or an authentication failure
MAIN SUCCESS SCENARIO (BASIC FLOW)
1. EMR sends an authentication request. 2. eHealth_hub service: a) Authenticates certificate received from the EMR b) Validates the EMR c) Validates that the common name on the certificate matches the setup of the service within the eHealth_hub service 3. EMR receives a successful authentication response.
EXCEPTION FLOWS
E1 - The certificate received from the EMR is invalid (e.g. expired or not signed) *Start Exception Flow at:* • *Basic Flow – After Step 2* 1. EMR receives an error and contacts Shared Health to troubleshoot/investigate.

4 Requirements

This section includes requirements and guidelines for configuring the authentication of the client instance to integrate the eHealth_hub services.

4.1 Requirement Column Definition

For ease of review and understanding, requirements are documented in a manner consistent with previous EMR Certification documentation. For each requirement, the following information is provided:

- **ID** – a unique identifier assigned to the requirement by Manitoba
- **Requirement** – a concise statement describing the requirement
- **Guidelines** – these additional instructions constitute part of the requirement and are relevant to implementation of the requirement in the EMR product. As such, these guidelines form part of the assessment criteria and are included in the planned product assessment.
- **Additional Notes** – relevant information or examples intended to give additional context to the requirement and to improve understanding
- **Status** – each requirement is clearly identified as:
    - New (not included in previous specifications);
    - Updated (modification to intent of the requirement from a previous version); or
    - Previous (unchanged from last issuance of core requirements).
- **Assessment** – the method of assessment is stated in the “Assessment” column for each requirement. All requirements will be assessed using the following method:
    - Assertion – Vendors will make an assertion (Yes or No) based on their self-assessment of the product’s ability to meet the requirement. Manitoba may choose to audit Vendor assertions as part of the certification process, as authorized within the Agreement
    - System Integration Testing – this most comprehensive assessment method requires an end-to-end test of key functions such as interoperability between the EMR product and other systems (e.g. eChart Manitoba Launch or eHealth_hub)

4.2 Authentication Requirements

Table 2: Authentication Requirements

ID	REQUIREMENT	GUIDELINES	ADDITIONAL NOTES	STATUS	ASSESSMENT
AUTH-001	Uses HTTPS using TCP/IP port 443 and TLS 1.2 protocol when invoking any of the eHealth_hub service methods.			Previous	System Integration Testing
AUTH-002	Supports AES 256 cipher encryption and SHA-256 message digest used for authentication.	See 5.3.2 Algorithms section for the preferred settings.		Previous	System Integration Testing
AUTH-003	Uses client-side certificate with each installation.	The client-side certificate is issued by Shared Health for each given client instance and must be used to make authenticated signed requests to the eHealth_hub service. In addition to using the client-side certificate to sign requests the eHealth_hub service public key must also be used.	Client-side certificates are: • Issued for a two-year period and subsequently reissued as part of the registration continuance process • Used by the eHealth_hub service for the validation of the client-instance as well in encryption and signing	Previous	System Integration Testing

			Shared Health is the certificate authority. A sample test certificate is provided in the eHealth_hub Assessment Information Addendums specific to each component.		
AUTH-004	Maintains that client-side certificate is non-transferable.	The client-side certificate for each given client instance is not transferable to other sites, machines or people.		Previous	Assertion
AUTH-005	Uses the client-side certificate for Digital Signatures and decrypting encrypted parts of service responses.			Previous	System Integration Testing
AUTH-006	Encrypts the message body for all eHealth_hub service requests, using the server public certificate received during mutual authentication.	Encryption is only required for message body and not the headers. See 5.3.6 Encryption and Decryption of Messages section for supported algorithms and methods. See 5.3.3 WS-Security Settings section for the applicable	All eHealth_hub service request messages must be encrypted with the eHealth_hub service public key received during mutual authentication and signed with client instance private key.	Previous	System Integration Testing

		settings.			
AUTH-007	Signs all SOAP requests using the client private certificate before sending the request via this interface.	See 5.3.5 Signing and Verification of Signatures section for supported algorithms and methods.		Previous	System Integration Testing
AUTH-008	Decrypts the SOAP body for all web service responses (that are received in this interface) using the client private certificate.	The error message responses are not encrypted. See 5.3.3 WS-Security Settings section for the applicable settings.		Previous	System Integration Testing
AUTH-009	Verifies the signatures on all received web service responses using the server public certificate before decrypting the SOAP body.	Timestamps associated with signed requests will only be valid for a configured interval, allowing for the potential resubmission of failed messages. See 5.3.5 Signing and Verification of Signatures section for verification methods.		Previous	System Integration Testing

5 Appendix A: Web Service Security

5.1 Mutual TLS Authentication

Mutual TLS authenticated communication uses TCP/IP port 443. The EMR must be able to communicate outbound on this port.

5.2 Certificate Governance

Shared Health is the certificate authority and custodian of the certificates. Certificates will be issued for a given EMR instance as part of the registration and certification process ensuring compliance with terms and conditions of use for the eHealth_hub service.

Certificates will be issued for an appropriate period of time and will be reissued as part of the registration continuance process. Certificates are non-transferable to other sites, machines or people.

5.3 WS-Security Overview

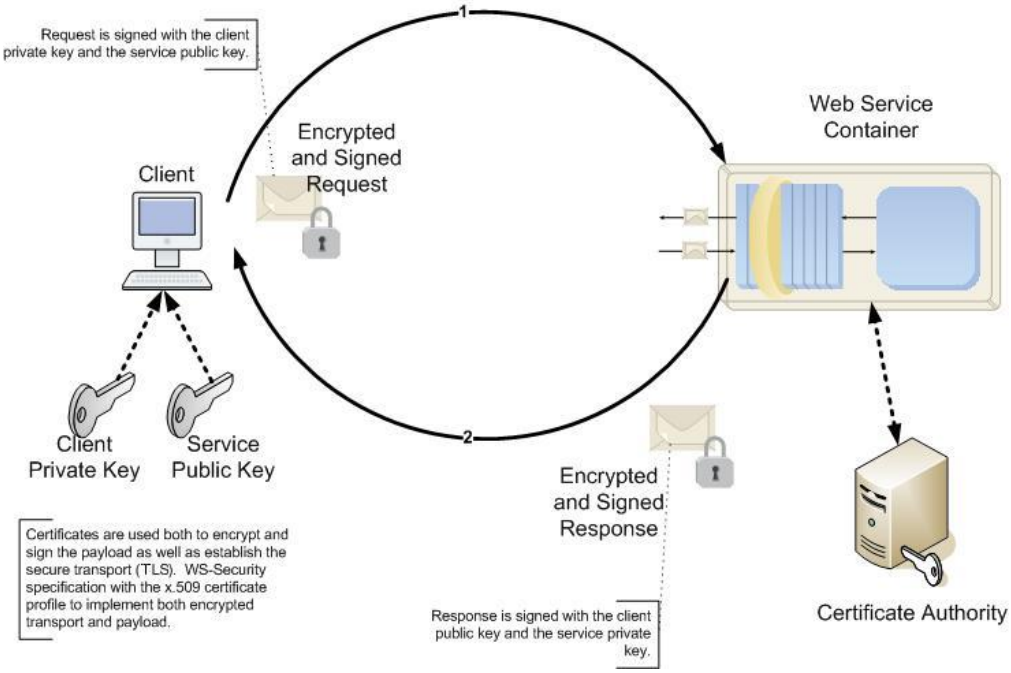

FIGURE 3: eHealth_hub - Authentication Overview

5.3.1 Mutual Auth.TLS

Stakeholder transactions are sent over the HTTPS protocol – HTTP over a connection secured by TLS. Stakeholders must enforce TLS mutual authentication. This means that both ends of the secure connection have authenticated the other party using public key technology.

To enable TLS mutual authentication, one needs:

1. An application framework that supports TLS 1.2. Most operating systems and application frameworks support this, including .Net, Java, etc.
2. A certificate (and corresponding private key) issued by the Shared Health Public Key Infrastructure (PKI). This certificate will be obtained as part of the Shared Health registration/renewal and enrolment process.
3. To configure application or integration engines to support TLS client authentication. Often this means configuring the toolkit/Application Programming Interface, before attempting the TLS connection, to use the Shared Health issued certificate.
4. A list of supported cryptographic algorithms for use with TLS. This list is sent to the server when the connection is being initiated. This list is sometimes explicitly set or is set based on a default set in the framework or operating system.
5. To configure the stakeholder's Point of Service (POS) being used to trust the Shared Health CA certificate. This will be used during the TLS initial handshake.

5.3.2 Algorithms

An Advanced Encryption Standard (AES) cipher with key sizes of 256 bits is utilized by Shared Health and should be supported. The SHA-256 message digest must be supported.

The following table describes supported TLS configuration in order of preference.

Table 3: Algorithms

TLS Characteristics	Preferred Setting
Key Exchange and Authentication	RSA/RSA
Key Size	2048 bit
Hash	SHA-256
Cipher Support	AES
Encryption	AES 256

5.3.3 WS–Security Settings

The following settings must be applied when WS-Security is applied to stakeholder requests:

Table 4: WS-Security Settings

SHORT NAME	VALUE
SOAP Actor/User Roles	Must be blank
WS-Sec Policy	Must utilize SOAP v.1.1
Signed Parts	SOAP Body: http://schemas.xmlsoap.org/soap/envelope/
WS-Security Profile	X.509 Certificate Token Profile

5.3.4 Use of Certificates

Certificates are required for TLS encryption, for WS-Security Signatures and for SOAP Body encryption. All signed parts of the message must use the same certificate.

5.3.5 Signing and Verification of Signatures

All soap transactions must be signed on posting of requests and signed responses verified on receipt.

5.3.5.1 Key Identifier Type

WS-Security provides for an infinite number of ways to validate a user. Supported identifier types:

- Binary Security Token
- Issuer Name and Serial Number
- Subject Key Identifier
- X.509 Certificate

5.3.5.2 Digital Signature

A digital signature or digital signature scheme is a mathematical mechanism for demonstrating the authenticity of a digital message or document. A valid digital signature gives a recipient assurance that the message was created by a known sender and that it was not altered in transit.

Supported signing algorithms:

- http://www.w3.org/2001/04/xmldsig-more#rsa-sha256
- http://www.w3.org/2001/04/xmldsig-more#rsa-sha384
- http://www.w3.org/2001/04/xmldsig-more#rsa-sha512

5.3.5.3 Canonicalization

Canonicalization, also sometimes referred to as standardization or normalization, is a process for converting data that has more than one possible representation into a "standard", "normal",

or canonical form. It may be abbreviated c14n, where 14 represents the number of letters between the C and the N. Canonicalization removes whitespace within tags, uses particular character encodings, sorts namespace references and eliminates redundant ones, removes XML and DOCTYPE declarations, and transforms relative URIs into absolute URIs.

Supported canonicalization algorithms:

- http://www.w3.org/2001/10/xml-exc-c14n#
- http://www.w3.org/2001/10/xml-exc-c14n#WithComments

5.3.5.4 Digital Digest (Hash or Message Authentication Coded (MAC))

A cryptographic hash function is a deterministic procedure that takes an arbitrary block of data and returns a fixed-size bit string, the (cryptographic) hash value, such that an accidental or intentional change to the data will change the hash value. The data to be encoded is often called the "message," and the hash value is sometimes called the message digest or simply digest.[1]

Supported digest algorithms:

- http://www.w3.org/2001/04/xmldsig-more#rsa-sha256
- http://www.w3.org/2001/04/xmldsig-more#rsa-sha384
- http://www.w3.org/2001/04/xmldsig-more#rsa-sha512

5.3.6 Encrypting and Decrypting of Messages

All message bodies must be encrypted on posting of requests and decrypted responses verified on receipt.

5.3.6.1 Key Identifier Type

WS-Security provides for an infinite number of ways to validate a user. Supported identifier types:

- Binary Security Token
- Issuer Name and Serial Number
- Subject Key Identifier
- X.509 Certificate

[1] https://www.oasis-open.org/committees/wss/

5.3.6.2 *Symmetric Encoding Algorithm*

Symmetric-key Algorithms are algorithms that use the same keys for both encryption and decryption. The keys are meant to represent shared secret between parties to ensure private information link. It is required that parties that are trying to communicate have access to the secret key.

Supported encryption algorithms:

- http://www.w3.org/2001/04/xmlenc#aes128-cbc
- http://www.w3.org/2001/04/xmlenc#aes192-cbc
- http://www.w3.org/2001/04/xmlenc#aes256-cbc
- http://www.w3.org/2001/04/xmlenc#tripledes-cbc

5.3.6.3 *Key Encryption Algorithm*

Supported encryption algorithms:

- http://www.w3.org/2001/04/xmlenc#rsa-1_5
- http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgf1p

5.3.6.4 *Create Encrypted Key*

Encrypt the symmetric key into an Encrypted Key.

6 Appendix E: Release Notes

Version 1.0 March 31, 2020

- Initial release
- eHealth_hub Authentication information has been modularized
- All eHealth_hub component specifications will contain a reference to this authentication specification