Security Module

The ebXML Message Service, by its very nature, presents certain security risks. A Message Service may be at risk by means of:

- Unauthorized access
- Data integrity and/or confidentiality attacks (e.g. through man-in-the-middle attacks)
- Denial-of-Service and spoofing

Each security risk is described in detail in the ebXML Technical Architecture Risk Assessment Technical Report [ebRISK].

Each of these security risks may be addressed in whole, or in part, by the application of one, or a combination, of the countermeasures described in this section. This specification describes a set of profiles, or combinations of selected countermeasures, selected to address key risks based upon commonly available technologies. Each of the specified profiles includes a description of the risks that are not addressed.

Application of countermeasures SHOULD be balanced against an assessment of the inherent risks and the value of the asset(s) that might be placed at risk.

Security Element

Web Services Security [WSS10] or [WSS11] can be utilized to secure an ebMS message. Web Services Security provides three mechanisms to secure messages: ability to send security tokens as part of a message, message integrity and message confidentiality.

Zero or one Security elements per target, belonging to the Web Services Security-defined namespace, MAY be present as a child of the SOAP Header. The Security element MUST be namespace qualified in accordance with Web Services Security. The structure and content of the Security element MUST conform to the Web Services Security specification [WSS10] or [WSS11] and the Web Services Security SOAP Messages with Attachments Profile [SOAPATTACH].

To promote interoperability the Security element MUST conform to the WS-I Basic Security Profile Version 1.0 [WSIBSP10], and WS-I Attachments Profile Version 1.0 [WSIAP10].

> Note
> An MSH implementation may elect to leverage WSS 1.0 and/or or WSS 1.1. Note that the security of attachment defined in WSS 1.1 is not only applicable to SOAP 1.1; security of attachment is orthogonal to the SOAP version, even though all examples in the WSS 1.1 specification depict only the SOAP 1.1 variant when securing attachments. In other words, an MSH may secure a SOAP 1.2 with Attachments message in the same way a SOAP 1.1 with Attachment can be secured in WSS 1.1. Refer to Section for complete details of the ebMS SOAP binding.

This specification outlines the use of Web Services Security x.509 Certificate Token Profile [???] and the Web Services Security Username Token Profile [???]. An MSH implementation MAY choose to support other Web Services Security Profiles.

Signing Messages

Signing of ebMS Messages is defined in Web Services Security [WSS10] and [WSS11]. Support for Web Services Security x.509 Certificate Token Profile [???] is REQUIRED to sign a message.

It is REQUIRED that compliant MSH implementations support Detached Signatures as defined by the XML Signature Specification [XMLDSIG].

An MSH implementation MAY support Enveloped Signatures as defined by the XML Signature Specification. Enveloped Signatures add an additional level of security in preventing the addition of XML elements to the SOAP Header. The use of Enveloped Signatures may limit the ability of intermediaries to process messages.

To ensure the integrity of the user-specified payload data and ebMS message headers it is RECOMMENDED that the entire eb:Messaging Container Element and the SOAP Body be included in the signature.

Signing SOAP with Attachments Messages

Application payloads that are are built in conformance with the [SOAPATTACH] specification may be signed. To sign a SOAP with Attachment message the Security element must be built in accordance to WSS 1.1.

It is REQUIRED that compliant MSH implementations support the Attachment-Content-Only transform. It is RECOMMENDED that compliant MSH implementations support the Attachment-Complete transform.

To ensure the integrity of the user-specified payload data and ebMS headers it is RECOMMENDED that the entire eb:Messaging Container Element, and all MIME Body parts of included payloads are included in the signature.

Encrypting Messages

Encryption of ebMS Messages is defined in Web Services Security [WSS10] and [WSS11]. Support for Web Services Security x.509 Certificate Token Profile is REQUIRED to encrypt message.

An MSH Implementation may encrypt the eb:Messaging Container Element. The eb:PartyInfo section may be used to aid in message routing before decryption has occurred. It is RECOMMENDED that the eb:PartyInfo section not be encrypted. To ensure the confidentiality of the user-specified payload data it is RECOMMENDED that the SOAP Body is encrypted.

Encrypting SOAP with Attachments Messages

Application payloads that are are built in conformance with the [SOAPATTACH] specification may be encrypted. To encrypt a SOAP with Attachment message the Security element must be built in accordance to WSS 1.1. To ensure the confidentiality of the user-specified payload data it is RECOMMENDED that the MIME Body parts of included payloads are encrypted,

Signing and Encrypting Messages

When both signature and encryption are required of the MSH, sign first and then encrypt.

UsernameToken Authentication

In constrained environments where management of XML digital signatures is not possible, an authentication alternative that is based on Web Services Security Username Token Profile [???] MUST be supported.

Support for wsse:PasswordText type passwords is REQUIRED. The value of the wsse:UserName element is an implementation issue. The “user” may represent the MSH itself, or may represent a party using the MSH. In the latter case, there is no requirement that this user name be identical to some eb:From/PartyId value.

Security Policy Errors

A responding MSH MAY respond with an error if a received ebMS message does not meet the security policy of the responding MSH. For example, a security policy might indicate that messages with unsigned parts of the SOAP Body or eb:Messaging Container element are unauthorized for further processing. If a responding MSH receives a message with unsigned data within the SOAP Body and error MAY be returned to the initiating MSH.