CHAPTER 12

XML Encryption and XML Signature

This chapter describes how to secure traffic using encryption and XML signature. It covers these topics:

- Using XML Encryption, page 12-127
- Using XML Signature, page 12-129

Using XML Encryption

The ACE XML Gateway supports XML Encryption. XML Encryption is a specification recommended by the W3C for securing XML content. It allows specific parts of a message to be encrypted without affecting other parts. You can set up decryption of XML content in incoming messages and encryption of content in outgoing messages.

Encryption and decryption rely upon the use of cryptographic keys. In decryption, the Gateway uses its own private key to decipher content that was encrypted with its public key at the message source. Conversely, the Gateway encrypts content using the public key of the intended recipient, who uses the corresponding private key to decipher the content. The keys can be in various formats, including PKCS #12, PEM, and DER.

To enable encryption controls in a virtual service, you must first specify its traffic as XML data (for SOAP service definitions, this is the default). If the outgoing message is specified as XML, the handler editor provides the option to encrypt the root element of the message or any element under the root element. The configuration editors provide tools you can use to choose which elements to encrypt.

SOAP messages present more options for encryption. For a SOAP document-style message, you can configure handlers to encrypt the entire document, or to encrypt specific elements of it. For a SOAP RPC-style message you can configure handlers to encrypt either the entire RPC message, or specific SOAP-RPC arguments.

When working with service definitions that handle XML encrypted messages with sensitive or confidential content, it's important to consider the effects of message logging.

When the ACE XML Gateway receives an encrypted SOAP message, it decrypts the message in order to identify its SOAPAction and match the message to its destination and perform other message processing tasks. If the handler is set for message logging, the text of the decrypted message may appear in the message traffic log. For this and other reasons, it's important to carefully control access to your logs. In general, message logging should be disabled in a production environment. Alternatively, you can apply an XSLT to messages before they are logged to conceal or remove sensitive information before it is logged.

Encrypting XML Messages

The following instructions describe how to encrypt outgoing responses for a handler or basic virtual service object. You can also configure encryption for service descriptors, in which case the outgoing request is encrypted. The procedure is similar to that described here.

To set up encryption of outgoing responses:

Step 1 While logged on to the console as an `Administrator` user or as a `Privileged` user with the `Routing` role, click **Virtual Services** link in the navigation menu.

Step 2 Click the name of the virtual service object for which you want to configure XML encryption.

As mentioned, for XML encryption controls to be enabled for the service definition, its message specification must indicate that it is XML data. It cannot be raw byte data, for example, which is the default for non-SOAP HTTP service definitions.

The **Response Message Specification** pane indicates how the message content is treated, whether as XML or as raw byte. If necessary, change message-body handling settings by:

a. Clicking the **Edit** link in the heading of the **Response Message Specification** subsection of the Outgoing Response section of the page.

b. Use the editor's controls to specify that the handler treat the bodies of outgoing response messages as XML.

c. Click **Save Changes**.

Step 3 In the service definition settings page, specify content encryption by clicking the **Add Encryption List** or the **Enable** link in the **XML Encryption pane** of the message processing section.

Step 4 In the XML Encryption configuration page, use the following controls to specify how encryption occurs:

- **For SOAP Role**. The SOAP role of the intended consumer of the encrypted data, if any.
- **Transport with Public Key**. The public key to use to encrypt the outgoing response, from these options:
  - The public key attribute of the consumer that sent the original request message
  - The public key used to sign the original request message.
  - A public key set by an extension created with the ACE XML Gateway SDK. This is only available if any extensions are on the ACE XML Manager. This ability is useful if an extension performs client authentication and it has access to the user's public key, which can then be used in message processing.
  - Any public key that a **Consumer Certificate Resource** provides to the ACE XML Manager. The **Upload** button allows you to add as a named Consumer Certificate Resource an XML certificate or keypair from the local filesystem or by URL.
- **Encryption Algorithm**. The algorithm to use to encrypt the message: 3DES, AES-256, AES-192, AES-128
- **Transport Cipher**. The cipher used to encrypt transported packets: RSA-PKCS#1 or RSA-OAEP.
- **Encryption Type**. Whether to encrypt the entire elements specified (including XML tag) or only the contents of elements.
- **XML Argument** (for HTTP messages). Whether to extract elements from a message body or from an argument.
- **Encrypt Elements**. An XPath expression that selects elements to encrypt. To encrypt multiple elements, click **Add New XPath Row**.

You can create as many XPath expressions as necessary to select elements to encrypt. For SOAP services, if the expression matches multiple elements in the message, all are encrypted. For HTTP post body, only the first element is matched.

Step 5 Click **Save Changes** when finished to commit your changes to the working policy.

Using XML Signature

A digital signature is a cryptographic value that enables a recipient to verify the source and validity of an incoming message. XML Signature defines an XML syntax for digital signatures.

When you enable SOAP header processing for a particular virtual service, the ACE XML Gateway validates XML signatures in incoming messages received at the interface defined by the object. If a signature does not match the element that is signed, the message is rejected.

Signature validity may not alone ensure message integrity—the signature could have been generated using any certificate, including one issued by an untrusted source. If using XML Signature as part of your implementation strategy, you should also specify which Certificate Authorities you want to be trusted, and direct the ACE XML Gateway to accept only signatures generated with certificates issued by those trusted CA.

Enabling header processing causes signatures to be validated if present in an incoming message (and causes messages with invalid signatures to be blocked), but it doesn't require a message to have a signature.

The final step in configuring XML Signature, therefore, is specifying the elements of the incoming message that must be signed. In the policy configuration, you can require a signature covering one or more of:

- the message timestamp (a common practice in Web service implementations).
- the first element below the SOAP body.
- a particular element you specify by XPath. Each XPath expression you specify must resolve to a signed XML element whose signature must be valid for the ACE XML Gateway to accept the message.

The following section describes how to set up an XML signature requirement.

Verifying XML Signatures

To set up an XML signature requirement for messages received at a particular consumer or backend service interface:

Step 1 In the **Virtual Services** browser, click the name of the virtual service object that receives messages for which you would like signature verification.

Step 2 In the settings page for the object, click the **enable** link next to **SOAP Header Processing**. This is in either the **Request Message Specification** section (for a handler or basic virtual service) or the **Response Message Specification** section (for a service descriptor or virtual service).

Step 3 Select the **Process header elements for SOAP role** checkbox.

If you save and deploy the changes up to this point, all signatures will be checked for validity against their embedded certificate. However, no restriction will be imposed on the signature in terms of the certificate used to generate it, nor will a signature be required in incoming messages. To specify that messages must have signatures, follow the next steps.

Step 4 Specify the elements that need to be covered by a signature, from one or more of these options:

- **Require that the Timestamp element is covered by a valid signature**. Requires a signed `Timestamp` element in incoming messages. The ACE XML Gateway blocks messages that do not have a `Timestamp` element, have an unsigned `Timestamp` element, or have an invalid signature covering the `Timestamp` element (that is, either computationally invalid or generated by an untrusted certificate).
- **Require that the SOAP method is covered by a valid signature**. Requires the SOAP method element to carry a verifiable signature. The SOAP method element is the first sub-element of the SOAP body element.
- **Require that nodes at each of these XPaths are covered by a valid signature**. Lets you specify other elements that need to be signed. Specify the XPath to the element by clicking the **Add new XPath** button and entering the XPath to the element, such as //*[local-name()='retrieveQuote']

If you save and deploy after specifying the elements required to be signed, the ACE XML Gateway will require incoming messages to have valid signatures covering the element you specify. The signature must be generated by a trusted certificate, that is, by a certificate issued by any of the Trusted Certificate Authorities you have specified in the policy.

Note For information on adding trusted Certificate Authorities, see "Uploading a Certificate Authority Resource" section on page 28-286.

Instead of accepting signatures generated by certificate of any trusted CA, you can specify a specific certificate to be matched (rather than one embedded in the signature) or a particular subset of trusted CAs issuing an embedded certificate, as described next.

Step 5 To specify certificate requirements for generated signatures, select the option **Verify all XML signatures; only accept signatures that can be verified using** and from the menu choose either:

- **embedded certificates signed by any of the uploaded Certificate Authorities**. Choose this item to accept signatures covered by certificates signed by any of the trusted CAs appearing on the **Trusted Certificate Authorities** page of web console.

  Note Enabling this option alone does not provide for verification. You must also specify the element to be covered by the signature, either `Timestamp` element, the SOAP method element, or an element specified by XPath reference.

- **embedded certificates signed by any one of the following Certificate Authorities**. Choose this item to specify a subset of the CAs appearing on the **Trusted Certificate Authorities** page of web console.

  When this option is enabled, select the CAs from the text box that appears below the option.

  Note Enabling this option alone does not provide for verification. You must also specify the element to be covered by the signature, either `Timestamp` element, the SOAP method element, or an element specified by XPath reference.

- **the public key from the following certificate, ignoring any embedded certificates**. Use a specified certificate's public key to verify all signatures, ignoring any certificates that may be embedded in the message. To specify the certificate with the key that the ACE XML Gateway uses to verify the signature, pick an item from the menu. You can choose the certificate used to authenticate the client, a public key set by an extension (created with the ACE XML Gateway SDK), or a specific consumer certificate loaded to the policy.

Step 6 Click **Save Changes** to commit your changes to the working policy.

Once the changes are deployed, the ACE XML Gateway verifies signatures as specified by your configuration.

Signing Outgoing Messages

For a SOAP virtual service, you can have the ACE XML Gateway sign elements with XML signatures as follows:

Step 1 In the **Virtual Services** browser, click on the name of the virtual service object for which you want to perform signing.

Step 2 In the virtual service information page, click the **enable** link next to XML Signing on either the Request Message Specification or the Response Message Specification sections.

Step 3 Choose the role of the recipient that should process the signature from the **For SOAP Role** menu.

- **no role**. Not intended for any particular role.
- **custom**. The header is intended for a custom SOAP role you specify. Type the name of any valid SOAP role into the field that appears beneath the For SOAP Role menu when you choose this item.
- **next**. The header is intended for the next SOAP role that processes the message.
- **ultimateReceiver**. The header is intended for the ultimate receiver of the message.

SOAP roles are attribute values that identify the intended recipients of particular headers in a message. If a receiver or an intermediary that relays SOAP messages has a SOAP role assigned to it, and it finds headers in a SOAP message assigned to that role, then it processes those headers.

Step 4 Choose a private key used for the signature from the Private Key menu. If necessary, click **Upload** to load the key file in the policy.

Step 5 Choose the encryption algorithm used for the signature from the **Algorithm** menu.

Step 6 Next to **Sign Elements**, choose whether you want to sign the entire element or only elements specified by the XPaths you enter. This option varies depending on the protocol of the service. In general, however, choose to sign the entire body (or RPC method) or specific elements by identifying them with XPath. If you add more than one element to be signed, note that each element is signed with the same private key. To sign RPC SOAP headers, specify them as XPath elements to be signed.

Step 7 When finished, click **Save Changes**.

After the changes are deployed, the ACE XML Gateway processes and generates XML Signatures as configured.