XML Signature

Beginning with Version 2.3.1, both the Request Group and Response Group structures were enhanced to support the XML Signature structure, based on the W3C XML Signature Standards (http://www.w3.org/TR/xmldsig-core/). The XML Signature is used in the MISMO Transaction Envelope structures to maintain the data integrity and authenticity of the original data. It can be applied to either all or just a portion of the transaction contents.

For example, a Credit Reporting Vendor might want to apply an XML Signature to the contents of the Credit Response transaction to create a "tamper evident seal" around the credit report data that was delivered to the lender. If there was ever a need to verify that the data contents were unmodified from the original report, the XML Signature can be validated to make sure that it still matches the data contents.

The XML Signature provides two features that are important for the security of the data:

- **Authentication –** The XML Signature, when properly implemented, includes a digital certificate from the sender of the transaction document. This certificate can be validated by the receiver to verify that the document was created by the sender and not some other source. MISMO recommends the use of digital certificates that have been issued by a SISAC-approved certificate provider.

  SISAC, the Secure Identity Services Accreditation Corporation, has been established by the Mortgage Bankers Association to verify that approved certificate providers meet its strict requirements (see www.SISAC.org). A properly issued and managed digital certificate provides a high degree of certainty regarding the identity of the lender, service provider or other entity that is generating the XML Signature.

- **Tamper Evidence –** When the XML Signature is applied to a transaction document by the sender, the receiver will be able to verify whether or not the data in the document has been changed since the XML Signature was first created. The XML Signature **does not prevent** tampering of the data, it only provides a way of detecting if the data has been changed.

Even though portions of the XML Signature data are encrypted, it is important to note that **the XML Signature by itself does not encrypt the transaction data**. This is a common misconception about the XML Signature. The encryption of the transactional data can be handled in a number of ways. There is an XML Encryption standard, but at the writing of this document, MISMO has not made any recommendations about the encryption of transactional data.

Signature Element Structure

The sample below shows a simplified Response Group envelope that contains the **Signature** element. The **Reference** element's **URI** attribute contains the **CreditResponseID** value of the **CREDIT RESPONSE** element, designating that this is the portion of the credit response transaction to be "signed".

Sample RESPONSE Envelope with <Signature> Element

```
<RESPONSE_GROUP MISMOVersionID="2.3.1">
   <RESPONDING_PARTY _Name="ABC Credit Services"/>
   <RESPOND_TO_PARTY _Name="MFC Mortgage"/>
   <RESPONSE ResponseDateTime="2005-04-18T13:01:58">
      <RESPONSE_DATA>
         <CREDIT_RESPONSE MISMOVersionID="2.3.1"
                           CreditResponseID="CResp02472">
           ... CREDIT RESPONSE DATA GOES HERE ...
         </CREDIT_RESPONSE>
      </RESPONSE_DATA>
   </RESPONSE>
   <Signature>
      <SignedInfo>
         <CanonicalizationMethod Algorithm="xml-exc-c14n#"/>
         <SignatureMethod Algorithm="xmldsig#rsa-sha1"/>
         <Reference URI="#CResp02472">
           <DigestMethod Algorithm="xmldsig#sha1"/>
           <DigestValue>SMn9GmIXglfuzGbmrLr4xU3</DigestValue>
         </Reference>
      </SignedInfo>
      <SignatureValue>jESGhs92msJS7S8snwXhwK7</SignatureValue>
      <KeyInfo>
         <X509Data>
           ... X.509 Digital Certificate data goes here ...
         </X509Data>
      </KeyInfo>
   </Signature>
</RESPONSE_GROUP>
```

Generating the XML Signature

Although many companies may use pre-built software or a web service to generate the XML Signature block, it is helpful to have a basic understanding of the steps involved in creating the **Signature** element. The steps below are a simplified for demonstration purposes.

1 – Identify the area(s) to be signed

In the previous sample, the **CREDIT RESPONSE** element was identified, using its **CreditResponseID** attribute value. This value is entered into the **Reference** element's **URI** attribute that is part of the **Signature** element. More than one **Reference** element may be included in a **Signature** element.

2 – Transform areas to be signed to Canonical XML

This step puts the XML data into a common normalized format, defined by the **CanonicalizationMethod**. This allows the Digest Values, which are created in the next step and in the Signature Validation process, to match each other.

3 – Calculate the Digest Value of Area(s) to be Signed

The data in the area to be signed is reduced to a unique **DigestValue**, using a method specified in **DigestMethod**.

4 – Calculate the Signature Value

An encrypted **SignatureValue** is now calculated on all of the data in the **SignedInfo** element.

5 – Store the Key Certificate

The **KeyInfo** element contains the certificate data that will be used to validate the signature.

Sample Signature Element Structure

```
     <Signature>
        <SignedInfo>
2       <CanonicalizationMethod Algorithm="xml-exc-cl4n#"/>
           <SignatureMethod Algorithm="xmldsig#rsa-sha1"/>
1          <Reference URI="#CResp02472">
              <DigestMethod Algorithm="xmldsig#sha1"/>
3             <DigestValue>SMn9GmIXglfuzGbmrLr4xU3</DigestValue>
           </Reference>
        </SignedInfo>
4       <SignatureValue>jESGhs92msJS7S8snwXhwK7</SignatureValue>
5       <KeyInfo>
           <X509Data>
              ... X.509 Digital Certificate data goes here ...
           </X509Data>
        </KeyInfo>
     </Signature>
```

Validating the XML Signature

When a party receives a document signed with an XML Signature, they can validate the signed portion of the document to determine whether or not any of the XML data in the signed area has been changed. The party that receives the document re-generates the **DigestValue** data values using the same steps used to generate original data values. If the original Digest Values match the new calculated values then the data has not changed.

Because validating the XML Signature involves the use of properly issued digital certificate, this process also provides reliable identification of the sender of the transaction.