XML Signatures for Interactive XML Documents

by

John Boyer
STSM, IBM Victoria Software Lab

What is an XML Signature?

- A digital signature is an information packet encodes an encrypted “fingerprint” of some content.
  - The encryption associates the digital signing identity of a person with the fingerprint of the content.
  - The content cannot be altered as there is no way to create a new encrypted fingerprint without the signer identity
- An XML signature is a digital signature that is encoded in a standard XML markup defined by the W3C

What does XML signature markup look like?

```
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
  <SignedInfo>
    <CanonicalizationMethod Algorithm=“http://www.w3.org/TR/2001/REC-xml-c14n-20010315”/>
    <SignatureMethod Algorithm=“http://www.w3.org/2001/04/xmldsig-more#rsa-sha256”/>
    <Reference URI? >...</Reference> +
  </SignedInfo>

  <SignatureValue>3H3K9TigFCzVDT4//wbZpAHr0wEAAA==</SignatureValue>

  (<KeyInfo>)?
  (<Object Id?>)*
</Signature>
```

What can an XML Signature Sign?

- The content signed by an XML signature can consist of multiple resources, including
  - Binary data
  - XML data
  - Presentational templates for data
- The XML Signature creates a “Reference” to each resource that comprises the content to be signed.
- A resource may be in the same document as the XML signature or it may be externally located.

What does a “Reference” look like?

- **Before signing**

```
<Reference URI="http://www.example.org/logo.jpg">
   <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
</Reference>
```

- **After signing**

```
<Reference URI="http://www.example.org/logo.jpg">
   <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
   <DigestValue>3H3K9TigFCzVDT4//wbZpAHr0wEAAA==</DigestValue>
</Reference>
```

But what if I want to work offline?

- First, the URI does not have to be dereferenced on the network if the application has a cached copy.
- Second, there are alternatives to **detached** signatures (i.e. those with non-empty URI References.
- A signature can also be an **enveloping** signature-- the content being signed is placed inside the XML signature element.
- A signature can be **enveloped**, which means it can be placed within a resource that it signs.
-

How are References in “Attached” Signatures done?

- Enveloping signatures are less common because the XML root element changes (so no further info here)
- The **enveloped** signature can be written with either an empty URI attribute or no URI attribute.
- An empty URI attribute refers to the XML document containing the XML signature.
- A Reference with no URI has application-defined meaning that is important to **XForms integration**.

How does being in an XForm affect an XML signature?

- Signing a document is an act performed by an end-user.
- In XForms, end-user changes are stored in a data layer called an “XForms instance”
- In XForms, instances are treated as separate XML documents, not part of the document containing the XForm.
- Therefore, <Reference URI= “”>... refers to the XML data of an XForms instance, not the document containing the XForm.

What does XForms markup look like?

```
<xf:model xmlns:xf="http://www.w3.org/2002/xforms">

   <xf:instance xmlns="">
      <instance>
         <a>2</a>
         <b>5</b>
         <c>7</c>
      </instance>
   </xf:instance>

   <xf:bind id="A" nodeset="a"/>
   <xf:bind id="B" nodeset="b"/>

   <xf:bind id="C" nodeset="c"
                   calculate="../a + ../b" />
</xf:model>
```

Where does XForms markup go in a document?

```
<office:document-content …>
    …
    <xf:model>
        …
        <xf:bind id="C" nodeset="c"
                        calculate=../a + ../b" />
        …
    </xf:model>
    …
    <form:form>
        <form:text xf:bind="C" form:id="Sum" … >
            <form:properties> … </form:properties>
        </form:text>
        …
    </form:form>
    …
```

The rest of the document picture

```
<office:document-content …>
    …
    <form:form>
        <form:text xf:bind="C" form:id="Sum_Ctl" … >
            <form:properties> … </form:properties>
        </form:text>
        …
    </form:form>
    …
    <text:p text:style-name="P20">
        <text:span>The sum is </text:span>
        <draw:control draw:control="Sum_Ctl" … />
    </text:p>
    …
</office:document-content>
```

Where does the XML signature go in a document?

```
<office:document-content …>
    …
    <xf:model>
        …
        <xf:instance>
            ...
                <ds:Signature ...>
                    ...
                    <ds:Reference ...>...</ds:Reference>
                    ...
                </ds:Signature>
            ...
        </xf:instance>
        …
    </xf:model>
    …
```

I just want to sign the transaction data, so why do I care about the "separate document" issue?

- First, some documents are "semi-structured" so user input contributes to structured data and unstructured document content, both of which must be signed.
- Second, even for purely structured data applications, a digital signature that signs only data is useless from a security standpoint
- "What you see is what you sign" is a principle expressed by the XML signature standard. XML signature must cover the data AND the presentational markup in the document containing the XForm.

How do I sign the presentation document *AND* the data?

- For XML Signatures in inline XForms instances, the Reference with no URI can be defined to refer to the containing document, and a workaround for the enveloped signature transform must be added:

```
<Reference>
  <Transforms>
    <Transform Algorithm= "http://www/w3/org/2002/06/xmldsig-filter2">
      <XPath xmlns= "http://www/w3/org/2002/06/xmldsig-filter2" Filter= "subtract">
          Absolute XPath to containing <dsig:Signature> element
      </Xpath>
     ...
    </Transform>
  ...
```

What are transforms and filters?

- Transforms provide a method for altering the referenced resource before taking its “digest”
- One reason to alter the referenced resource is to **filter** out an enveloped signature.
- Affixing the Reference digest values and the signature value modifies the document that the signature is signing
- An XPath Filter 2 transform was used in lieu of an enveloped signature transform because the latter is slower and is not useful in the general case.

Why can't I use here(), and why is it bad that I can't?

- Enveloped transform defined in terms of here(), which is defined only for “same document” references
- If XML signatures are in repeated content, then ID-based and absolute References are problematic
- URI-less Reference has app-specific context, so it needs to allow app-specific definition of here()
- Also, clarify that URI-less Reference can indicate a nodeset (not just octet stream), that Transforms can apply to it, and that the hook to specify the app-specific context is **required** to implement.

Why else would you filter a resource being signed?

- You may want to omit the “wizard, guided interview” pages from a form and only sign the “contract view” pages.
- You may have a multiple signer scenario or any scenario where some part of the document must remain interactive after the signature is affixed.

How should I filter a document I want to sign?

- XPath Filter 2 provides **subtract**, union and intersect filters.
- The **union** and **intersect** filters *almost* always leave ***gaping security holes*** in your application.
- The **subtract** filter is the ***most secure filter*** to use because it makes you say exactly what has to remain interactive after the signature is affixed.
- Obviously, a signature is designed to prevent unauthorized changes, and the subtract filter states the changes that *are* authorized by the signature.

What if I want to subtract more than one thing from the document?

- In the two-signer scenario, the first signature needs to subtract itself and a second signature.
- Sometimes you may want to subtract the enveloped signature, an office-use-only section and an approval signature.
- You would just use multiple XPath elements in the same XPath Filter 2 Transform.
- Need better W3C test suite, e.g. Current Java 6 fails on the subtract filter (need updated apache lib).
- Need to deprecate XPath Filter 1 for performance

How to actuate signing, validating?

- Activation can start with xf:trigger
- The UI binding can go to the ds:SignatureValue, allowing us to take advantage of required MIP
- The xf:label can refer to ../ds:KeyInfo/ds:KeyName
- However, seem to need event driven host language participation to provide UI for sign versus validate.
  - Choose signing certificate, augment metadata
  - Show core validation results as well as certificate validation information, metadata, enhanced validation with metadata

Any other security concerns?

- The XML Signatures standard became a W3C Recommendation in early 2002
- Originally specified with MD5 and SHA1, but now at least SHA256 is preferred.
- Unfortunately, platform technologies like MS CryptoAPI do not make this available.
- It would help if XML Sec WG could do the “community” work to get this fixed.
- Market pressure in EU

Backup

If parts of a document are subtracted, can they be used to attack security?

- What you see is what you sign. An unsigned piece of markup can **obscure** or **unobscure** signed content that the signer saw.
- We augment core signature generation and validation with on-the-glass geometric tests called the **Overlap and Layout Tests**
- Unsigned content cannot overlap signed content during signing or validation
- Signed content cannot change how it overlaps other signed content between signing and validation

How do you protect the geometry tests from attack?

- The overlap test is done on both sign and validate, which is sufficient.
- The signed content does not change and it is never overlapped by unsigned content.
- The layout test adds layout information generated during signing as “metadata”
- Layout is not tested on pages that are not signed.
- A signature “Object” is created to hold the layout information metadata, and this Object is “unioned” into the signature in some way.