WASP Mini Tutorial

Preliminary, subject to change without notice.

WASP - Web Activated Signature Protocol, is an effort creating a standard for browser-based signatures. The applications for web-signatures include e-government services (C2G), on-line banking and many other systems that are building on the "thin client" paradigm. This tutorial assumes basic familiarity with XML, and XML signatures as well as with standard web-technologies.

The following is a minimal description on how the WASP signature framework is to be used in a web-application.

"Hello signature world!"

Assume that you want a user to sign the text "Hello signature world!". To do that the requesting service (a web-server application), may as a minimum create a `SignatureRequest` object like below:

```
<?xml version="1.0" encoding="UTF-8"?>
<SignatureRequest ID="_10d16a170d97ddd1a7024f7d9ee"
                  SubmitURL="https://example.com/submit"
                  ServerTime="2007-08-26T17:48:01Z"
                  xmlns="http://xmlns.webpki.org/wasp/1.0/core#">
  <SignatureProfiles>
    <pr:ProfileData xmlns:pr="http://xmlns.webpki.org/wasp/1.0/prof/xds#"/>
  </SignatureProfiles>
  <DocumentReferences>
    <MainDocument ContentID="cid:d0@example.com" MimeType="text/plain"/>
  </DocumentReferences>
  <DocumentData>
    <Text ContentID="cid:d0@example.com">Hello signature world!</Text>
  </DocumentData>
</SignatureRequest>
```

When this structure is returned (as the response of an arbitrary HTTP GET or POST operation invoked by the *user*), to a compliant browser using the WASP-specific MIME-type `application/xbpp+xml`, the browser should *automatically* respond with a signature dialog or similar showing the text to be signed:

In the example above, the core WASP XML DSig profile was requested.

If the user carries out the signature process, a `SignatureResponse` object like the following is POSTed to the `SubmitURL` specified in the `SignatureRequest` object:

```
<?xml version="1.0" encoding="UTF-8"?>
<SignatureResponse xmlns="http://xmlns.webpki.org/wasp/1.0/core#">
  <pr:SignedData ID="_10d16a170d97ddd1a7024f7d9ee"
                 SubmitURL="https://example.com/submit"
                 RequestURL="https://example.com/signreq"
                 ServerTime="2007-08-26T17:48:01Z"
                 ClientTime="2007-08-26T19:53:50+02:00"
                 ServerCertificateSHA1="I29nPn9XxbvEOlS7cTJ6APfBBp4="
                 xmlns:pr="http://xmlns.webpki.org/wasp/1.0/prof/xds#">
    <DocumentReferences>
      <MainDocument ContentID="cid:d0@example.com" MimeType="text/plain"/>
    </DocumentReferences>
    <DocumentSignatures CanonicalizationAlgorithm="http://xmlns.webpki.org/wasp/1.0/core#cn"
                        DigestAlgorithm="http://www.w3.org/2000/09/xmldsig#sha1">
      <Digest ContentID="cid:d0@example.com">uutNNW7SZPtWdgNHzevaNLRDNR4=</Digest>
    </DocumentSignatures>
    <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
      <ds:SignedInfo>
        <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
        <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
        <ds:Reference URI="#_10d16a170d97ddd1a7024f7d9ee">
          <ds:Transforms>
            <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
            <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
          </ds:Transforms>
          <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
          <ds:DigestValue>I29nPn9XxbvEOlS7cTJ6APfBBp4=</ds:DigestValue>
        </ds:Reference>
      </ds:SignedInfo>
      <ds:SignatureValue>AeSIfXfsJKjDk Removed Base64 Data WA=</ds:SignatureValue>
      <ds:KeyInfo>
        <ds:X509Data>
          <ds:X509IssuerSerial>
            <ds:X509IssuerName>CN=Demo Sub CA,O=example.com,C=US</ds:X509IssuerName>
            <ds:X509SerialNumber>1153346562390</ds:X509SerialNumber>
          </ds:X509IssuerSerial>
          <!-- Signer DN: "CN=Marion Anderson, serialNumber=19750710-1518" -->
          <ds:X509Certificate>MIIB7TCC Removed Base64 Data YC/s==</ds:X509Certificate>
        </ds:X509Data>
      </ds:KeyInfo>
    </ds:Signature>
  </pr:SignedData>
</SignatureResponse>
```

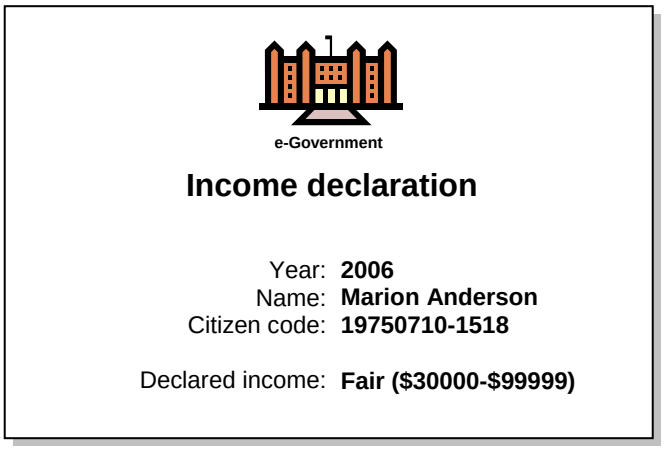
Essentially this is it. Note that the signature (using the selected sample signature profile NB), is *detached* from the actual (raw) document data, while document meta-data and hashes of the document data objects are included. In case you would like to keep the document data and signature in one place, the sample profile (see XML Schemas), allows you to do that. The last lines of the signature container would then look like this:

```
  </pr:SignedData>
  <DocumentData>
    <Text ContentID="cid:d0@example.com">Hello signature world!</Text>
  </DocumentData>
</SignatureResponse>
```

Note that the POSTed signature object is like any other user-originated browser-to-server invocation, which means that the server should preferably respond with a page showing a note to the user that the signed operation has been successfully received (and presumably also validated).

Using HTML

Since plain-text documents are rather constraining for displaying complex data, most service providers will probably want to use HTML forms. Assume that the following signature request display is to be coded in HTML:



A suitable `SignatureRequest` object could then be like the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<SignatureRequest ID="_10d15ead54bc8359c0fae1ae0f7"
                  SubmitURL="https://taxes.mygov.org/SignDeclaration"
                  CancelURL="https://taxes.mygov.org/IncomeDeclaration?INCOME=6"
                  ServerTime="2007-08-26T19:38:32+02:00"
                  xmlns="http://xmlns.webpki.org/wasp/1.0/core#">
  <SignatureProfiles>
    <pr:ProfileData xmlns:pr="http://xmlns.webpki.org/wasp/1.0/prof/xds#"/>
  </SignatureProfiles>
  <DocumentReferences>
    <MainDocument ContentID="cid:d0@mygov.org" MimeType="text/html"/>
    <EmbeddedObject ContentID="cid:d1@mygov.org" MimeType="text/css"/>
    <EmbeddedObject ContentID="cid:d2@mygov.org" MimeType="image/gif"/>
  </DocumentReferences>
  <DocumentData>
    <Text ContentID="cid:d0@mygov.org"><![CDATA[<html><head><link rel="stylesheet" type="text/css"
href="cid:d1@mygov.org"></head><body scroll="no"><table border="0" cellpadding="0" cellspacing="0"
width="100%" height="100%"><tr><td align="center"><table cellpadding="0" cellspacing="0" border="0">
<tr><td align="center"><img src="cid:d2@mygov.org"></td></tr><tr><td height="15"></td></tr><tr>
<td align="center" class="headline">Income declaration</td></tr><tr><td height="20"></td></tr><tr>
<td align="center"><table cellpadding="0" cellspacing="5"><tr><td align="right">Year: </td><td>
<b>2006</b></td></tr><tr><td align="right">Name: </td><td><b>Marion Anderson</b></td></tr>
<tr><td align="right">Citizen code: </td><td><b>19750710-1518</b></td></tr><tr>
<td colspan="2" height="5"></td></tr><tr><td align="right">Declared income: </td><td>
<b>Fair ($30000-$99999)</b></td></tr></table></td></tr></table></td></tr></table></body></html>]]></Text>
    <Text ContentID="cid:d1@mygov.org">body {font-size: 8pt; color: #000000; font-family: Verdana, Arial;
background-color: #ffffff}
```

```
td {font-size: 8pt; font-family: Verdana, Arial}
.headline {font-weight: bolder; font-size: 11pt; font-family: Verdana, Arial}</Text>
      <Binary ContentID="cid:d2@mygov.org">R0lGOD Removed Base64 Data 67LZLQggAOw==</Binary>
   </DocumentData>
</SignatureRequest>
```

Note that all document objects to be signed, must be *declared* (see `DocumentReferences`), and use Content-ID URIs such as featured in HTML mail. The signature application (plugin etc.) does the translation between "cid" URIs and local storage used during the signature process.

Next Step(s)

For more thorough information and finding out other features of the WASP scheme, the interested reader should take peek in **wasp-core-schema.pdf** as well as in a number of associated application notes.

Anders Rundgren (editor)
September 2007