

Re: XmlDsig Countersignature DigestValue

Source: <http://www.derkeiler.com/Newsgroups/microsoft.public.dotnet.security/2007-03/msg00094.html>

- *From:* "Iguana" <szewczyk@xxxxxxxxxxxxxxxx>
 - *Date:* 27 Mar 2007 04:31:18 -0700
-

Hi!

I have create signatures with csharp (vc 2005) and net 2.0.
I think preserwe white spaces is not a problem in my code – this works fine (with my code I verify signature have generated in java – verification works good).

What I do exacly in my code:

read from xml document all tag <Signature ... </Signature> and put this to new XmlDocument.

```
// get signature to countersign
XmlNodeList signs =
existingXmlDocument.GetElementsByTagName("Signature",
SignedXml.XmlDsigNamespaceUrl);
XmlElement el = signs[0]; // in my test code I have only one signature
to countersign
SignedXml sig = new SignedXml();
sig.LoadXml((XmlElement)el);

XmlDocument doc = new XmlDocument(); //new empty xmlDocument – without
header and any attributes
doc.PreserveWhitespace = true;

// load obj – sognature to countersign to new created XmlDocument
System.Security.Cryptography.Xml.DataObject obj = new
System.Security.Cryptography.Xml.DataObject();

obj.LoadXml(sig.GetXml());
doc.LoadXml(obj.GetXml().OuterXml); // this load to new created
XmlDocument signature xml text

Transform t1 = new
System.Security.Cryptography.Xml.XmlDsigC14NTransform(); // my
reference have not transforms – only SignedINfo have connonicalization
transform
t1.LoadInput(doc);
System.IO.Stream s1 = (System.IO.Stream)t1.GetOutput();

// calculate hash after transform
SHA1 sha1 = SHA1.Create();
```

Re: Xmlldsig Countersignature DigestValue

```
MessageBox.Show(string.Format("{0}",  
Convert.ToBase64String(sha1.ComputeHash(s1))));
```

This is my first test
After fall, I add new transform:

```
Transform t2 = (Transform)CryptoConfig.CreateFromName("http://  
www.w3.org/2001/10/xml-exc-c14n#WithComments");  
t2.LoadInput(t1); // transform on transformed signature  
System.IO.Stream s2 = (System.IO.Stream)t2.GetOutput();
```

```
MessageBox.Show(string.Format("{0}",  
Convert.ToBase64String(sha1.ComputeHash(s2))));
```

This is what i do.
DigestValue is wrong (in code with two transformation – I have
DigestValue on t1 and t2 the same always!)
I have no more idea... but must calculate this DigestValue before i
call SignedXml.ComputeSignature and show DigestValue to my application
user.
Any other idea?
Iguana

.