

RE: how to specify private key to generate signature

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework/2007-02/msg00244.html>

- *From:* stcheng@xxxxxxxxxxxxxxxxxxxxxx (Steven Cheng[MSFT])
 - *Date:* Fri, 09 Feb 2007 08:55:28 GMT
-

Hello Benny,

From your description, you're going to use the DSA provider component in

..net framework to perform digital signing on message, and you're wondering how to supply the private key for the provider ,correct?

Based on my experience, private key for digital signing are contained in CSP Containers. Generally, when you get a certificate(for encrypting and signing), you will get public/private key pair. And you can specify a container name when requesting the certificate from CA. After the certificate is installed on your computer(in user or computer store), you can get the private key either from the certificate(if private key is not exported) or from the CSP container(if you've specify a container name or know an existing key container). Then, in .net code, you can either retrieve a DSA Provider from the certificate's PrivateKey property or use CspParameters(specify key container name and providertype) to create a DSA Provider. Here are two code snippet demonstrate this:

```
=====get DSA provider from certificate=====
private void btnSign_Click(object sender, EventArgs e)
{
    string tp;

    //Thumbprint of the certificate
    tp = "90 6e 56 71 da 31 ac bc 9e 3e 8b b2 08 5f 6e a9 ec f1 21
7a";

    X509Store store = new X509Store(StoreName.My,
StoreLocation.CurrentUser);
    store.Open(OpenFlags.ReadOnly);

    X509Certificate2Collection certs =
store.Certificates.Find(X509FindType.FindByThumbprint, tp, false);
```

RE: how to specify private key to generate signature

```
X509Certificate2 dsouser = certs[0];

store.Close();

DSACryptoServiceProvider dsa = dsouser.PrivateKey as
DSACryptoServiceProvider;

if (dsa != null)
{
string plaintext = "123456";

byte[] hashedtext =
SHA1.Create().ComputeHash(Encoding.UTF8.GetBytes(plaintext));

byte[] signature = dsa.CreateSignature(hashedtext);

MessageBox.Show("signature: " +
Convert.ToBase64String(signature));

MessageBox.Show("Signature Invalid: " +
dsa.VerifySignature(hashedtext, signature));

}
}
=====

=====create DSA provider from key container=====
private void btnSignCSP_Click(object sender, EventArgs e)
{

CspParameters csp = new CspParameters();

csp.KeyContainerName = "SC-TEST-CERT2";

//csp.ProviderName = "Microsoft Enhanced DSS and Diffie-Hellman
Cryptographic Provider";

csp.ProviderType = 13;

DSACryptoServiceProvider dsa = new
DSACryptoServiceProvider(csp) ;// dsouser.PrivateKey as
DSACryptoServiceProvider;
```

RE: how to specify private key to generate signature

```
if (dsa != null)
{
string plaintext = "123456";
byte[] hashedtext =
SHA1.Create().ComputeHash(Encoding.UTF8.GetBytes(plaintext));

byte[] signature = dsa.CreateSignature(hashedtext);

MessageBox.Show("signature: " +
Convert.ToBase64String(signature));

MessageBox.Show("Signature Invalid: " +
dsa.VerifySignature(hashedtext, signature));

}

}

}
```

=====

For test, you can setup a CA on windows server machine to issue some test certificate(make sure you choose the DSS provider). Here are two good web articles introducing the cryptography programming in .net framework:

#Cryptography in Microsoft.NET Part I: Encryption
<http://www.c-sharpcorner.com/UploadFile/gparamasivam/CryptEncryption11282005061028AM/CryptEncryption.aspx>

#Cryptography in Microsoft.NET Part II: Digital Envelop and Digital Signatures
<http://www.c-sharpcorner.com/UploadFile/Gowri%20S%20Paramasivam/Cryptography211242005003308AM/Cryptography2.aspx?ArticleID=cf900b08-35ed-4524-aeff-6806265a4196>

Hope this helps.

Sincerely,

Steven Cheng

Microsoft MSDN Online Support Lead

=====

Get notification to my posts through email? Please refer to
<http://msdn.microsoft.com/subscriptions/managednewsgroups/default.aspx#notif>

RE: how to specify private key to generate signature

RE: how to specify private key to generate signature

ications.

Note: The MSDN Managed Newsgroup support offering is for non-urgent issues where an initial response from the community or a Microsoft Support Engineer within 1 business day is acceptable. Please note that each follow up response may take approximately 2 business days as the support professional working with you may need further investigation to reach the most efficient resolution. The offering is not appropriate for situations that require urgent, real-time or phone-based interactions or complex project analysis and dump analysis issues. Issues of this nature are best handled working with a dedicated Microsoft Support Engineer by contacting Microsoft Customer Support Services (CSS) at <http://msdn.microsoft.com/subscriptions/support/default.aspx>.

=====

This posting is provided "AS IS" with no warranties, and confers no rights.

.