

RE: CryptSignMessage fails with unknown cryptographic algorithm

Source:

<http://www.derkeiler.com/Newsgroups/microsoft.public.platformsdk.security/2007-12/msg00056.html>

- *From:* Kenneth <Kenneth@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Mon, 31 Dec 2007 09:01:01 -0800
-

Hello,

I am experiencing a similar problem when I attempt to validate a detached signature. When using CAPICOM.Verify, an error message saying " unknown cryptographic algorithm" is displayed. I used a ANS1 editor to verify the signature is formed properly, so I am just not sure what could be wrong.

I really appreciate any help you may provide

"Arnoud Lems" wrote:

Hi,

I'm using a call to CryptSignMessage to sign a certain message, however the call to obtain the size of the returned signed message blob fails. GetLastError reveals that the call failed because of an "Unknown cryptographic algorithm". Can anyone tell me why this error occurs?

The code below shows what I have so far.

```
using System;
using System.Text;
using System.Runtime.InteropServices;
using System.ComponentModel;

#region Structs
[StructLayout(LayoutKind.Explicit, Size=20)]
public struct CERT_CONTEXT
{
    [FieldOffset(0)]
    public uint dwCertEncodingType;

    [FieldOffset(4)]
    public IntPtr pbCertEncoded;

    [FieldOffset(8)]
```

RE: CryptSignMessage fails with unkown cryptographic algorithm

```
public uint cbCertEncoded;

[FieldOffset(12)]
public IntPtr pCertInfo;

[FieldOffset(16)]
public IntPtr hCertStore;
}

[StructLayout(LayoutKind.Explicit, Size=8)]
public struct CRYPT_OBJID_BLOB
{
[FieldOffset(0)]
public uint cbData;

[FieldOffset(4)]
public IntPtr pbData;
}

[StructLayout(LayoutKind.Explicit, Size=12)]
public struct CRYPT_ALGORITHM_IDENTIFIER
{
[FieldOffset(0)]
public IntPtr pszObjId;

[FieldOffset(4)]
public CRYPT_OBJID_BLOB Parameters;
}

[StructLayout(LayoutKind.Explicit, Size=84)]
public struct _CRYPT_SIGN_MESSAGE_PARA
{
[FieldOffset(0)]
public uint cbSize;

[FieldOffset(4)]
public uint dwMsgEncodingType;

[FieldOffset(8)]
public unsafe IntPtr pSigningCert;

[FieldOffset(12)]
public CRYPT_ALGORITHM_IDENTIFIER HashAlgorithm;

[FieldOffset(24)]
public IntPtr pvHashAuxInfo;

[FieldOffset(28)]
public uint cMsgCert;

[FieldOffset(32)]
```

RE: CryptSignMessage fails with unkown cryptographic algorithm

```
public IntPtr rgpMsgCert;

[FieldOffset(36)]
public uint cMsgCrl;

[FieldOffset(40)]
public IntPtr rgpMsgCrl;

[FieldOffset(44)]
public uint cAuthAttr;

[FieldOffset(48)]
public IntPtr rgAuthAttr;

[FieldOffset(52)]
public uint cUnauthAttr;

[FieldOffset(56)]
public IntPtr rgUnauthAttr;

[FieldOffset(60)]
public uint dwFlags;

[FieldOffset(64)]
public uint dwInnerContentType;

[FieldOffset(68)]
public CRYPT_ALGORITHM_IDENTIFIER HashEncryptionAlgorithm;

[FieldOffset(80)]
public IntPtr pvHashEncryptionAuxInfo;

}

[StructLayout(LayoutKind.Explicit, Size=20)]
public struct _CRYPT_VERIFY_MESSAGE_PARA
{
[FieldOffset(0)]
public uint cbSize;

[FieldOffset(4)]
public uint dwMsgAndCertEncodingType;

[FieldOffset(8)]
public uint hCryptProv;

[FieldOffset(12)]
public IntPtr pfnGetSignerCertificate; //fn ptr->delegate:
GeneratedDelegate9
```

RE: CryptSignMessage fails with unknown cryptographic algorithm

```
[FieldOffset(16)]
public IntPtr pvGetArg;
}
#endregion

public class Win32Alt
{
    public const uint PKCS_7_ASN_ENCODING = 0x00010000;
    public const uint X509_ASN_ENCODING = 0x00000001;
    public const uint CERT_SYSTEM_STORE_CURRENT_USER = 0x00010000;
    public const uint CERT_STORE_READONLY_FLAG = 0x00008000;
    public const uint CERT_STORE_OPEN_EXISTING_FLAG = 0x00004000;
    public const uint CERT_FIND_SUBJECT_STR = 0x00080007;
    public const string CERT_STORE_NAME = "MY";
    public const int CERT_STORE_PROV_SYSTEM = 10;
    public const uint RSA_CSP_PUBLICKEYBLOB = 19;
    public const int AT_KEYEXCHANGE = 1; //keyspec values
    public const int AT_SIGNATURE = 2;
    public const string szOID_RSA_MD5 = "1.2.840.113549.1.1.4";

    [DllImport(@"crypt32.dll", EntryPoint="CryptVerifyMessageSignature",
    CallingConvention=CallingConvention.StdCall, SetLastError=true)]
    public static extern int CryptVerifyMessageSignature(
    IntPtr pVerifyPara,
    int dwSignerIndex,
    byte[] pbSignedBlob,
    int cbSignedBlob,
    byte[] pbDecoded,
    ref int pcbDecoded,
    IntPtr ppSignerCert);

    [DllImport(@"crypt32.dll", EntryPoint="CryptSignMessage",
    CallingConvention=CallingConvention.StdCall, SetLastError=true)]
    public static extern int CryptSignMessage(
    IntPtr pSignPara,
    int fDetachedSignature,
    uint cToBeSigned,
    IntPtr rgpbToBeSigned, //pointer naar een array van bufferpointers
    uint[] rgcbToBeSigned, //pointer naar array van buffer grotes
    byte[] pbSignedBlob,
    ref int pcbSignedBlob);

    [DllImport("Crypt32.dll", EntryPoint="CertOpenStore")]
    public static extern IntPtr CertOpenStore
    (
    IntPtr lpszStoreProvider,
    UInt32 dwEncodingType,
    IntPtr hCryptProv,
    UInt32 dwFlags,
    byte[] pvPara
    );
}
```

RE: CryptSignMessage fails with unknown cryptographic algorithm

```
[DllImport("Crypt32.dll", EntryPoint="CertFindCertificateInStore")]
public static extern IntPtr CertFindCertificateInStore
(
    IntPtr hCertStore,
    UInt32 dwCertEncodingType,
    UInt32 dwFindFlags,
    UInt32 dwFindType,
    IntPtr pvFindPara,
    IntPtr pPrevCertContext
);

[DllImport("crypt32.dll", SetLastError=true)]
public static extern bool CertCloseStore(
    IntPtr hCertStore,
    uint dwFlags) ;

}

/// <summary>
/// Summary description for Class1.
/// </summary>
class App
{
    /// <summary>
    /// The main entry point for the application.
    /// </summary>

    private static void showWin32Error(int errorcode)
    {
        Win32Exception myEx=new Win32Exception(errorcode);
        Console.WriteLine("Error message: {0} (Code: 0x{1:X})", myEx.Message,
        myEx.ErrorCode);
    }

    [STAThread]
    static void Main(string[] args)
    {
        const string certId = "WSE2QuickStartClient";

        uint ENCODING_TYPE = Win32Alt.PKCS_7_ASN_ENCODING |
        Win32Alt.X509_ASN_ENCODING ;

        string Message = "Hello world";

        IntPtr hStoreHandle = Win32Alt.CertOpenStore ((IntPtr)
        Win32Alt.CERT_STORE_PROV_SYSTEM, 0, IntPtr.Zero,
        Win32Alt.CERT_SYSTEM_STORE_CURRENT_USER,
        Encoding.Unicode.GetBytes(Win32Alt.CERT_STORE_NAME));
```

RE: CryptSignMessage fails with unknown cryptographic algorithm

```
IntPtr SIGNER_NAME = Marshal.StringToBSTR(certId);
IntPtr pCtx = IntPtr.Zero;
IntPtr pCertContext;
CERT_CONTEXT SignerCert;

if (hStoreHandle != IntPtr.Zero)
{
    pCertContext = Win32Alt.CertFindCertificateInStore(hStoreHandle,
    ENCODING_TYPE, 0, Win32Alt.CERT_FIND_SUBJECT_STR, SIGNER_NAME, pCtx);
    SignerCert = (CERT_CONTEXT)Marshal.PtrToStructure(pCertContext,
    typeof(CERT_CONTEXT));

    _CRYPT_SIGN_MESSAGE_PARA SignPara = new _CRYPT_SIGN_MESSAGE_PARA();

    SignPara.cbSize =
    (uint)Marshal.SizeOf(typeof(_CRYPT_SIGN_MESSAGE_PARA));
    SignPara.dwMsgEncodingType = ENCODING_TYPE;
    SignPara.pSigningCert = pCertContext;
    SignPara.HashAlgorithm.pszObjId =
    Marshal.StringToBSTR(Win32Alt.szOID_RSA_MD5);
    SignPara.HashAlgorithm.Parameters.cbData = 0;
    SignPara.HashAlgorithm.Parameters.pbData = IntPtr.Zero;
    SignPara.pvHashAuxInfo = IntPtr.Zero;
    SignPara.cMsgCrl = 0;
    SignPara.rgpMsgCrl = IntPtr.Zero;
    SignPara.cAuthAttr = 0;
    SignPara.rgAuthAttr = IntPtr.Zero;
    SignPara.cUnauthAttr = 0;
    SignPara.rgAuthAttr = IntPtr.Zero;
    SignPara.dwFlags = 0;
    SignPara.rgAuthAttr = IntPtr.Zero;
    SignPara.dwInnerContentType = 0;
    SignPara.pvHashEncryptionAuxInfo = IntPtr.Zero;

    SignPara.rgpMsgCert =
    Marshal.AllocCoTaskMem(Marshal.SizeOf(typeof(IntPtr)));
    Marshal.StructureToPtr(pCertContext, SignPara.rgpMsgCert, false);

    SignPara.cMsgCert = 1;

    char[] Data = Message.ToCharArray();

    IntPtr pData = Marshal.AllocCoTaskMem(Data.Length + 1);
    Marshal.Copy(Data, 0, pData, Data.Length);

    IntPtr ppData = Marshal.AllocCoTaskMem(Marshal.SizeOf(typeof(IntPtr)));

    Marshal.WriteIntPtr(ppData, pData);

    IntPtr pDataSize = Marshal.AllocCoTaskMem(4);
    Marshal.WriteInt32(pDataSize, Data.Length);
```

RE: CryptSignMessage fails with unknown cryptographic algorithm

```
uint[] BlobGrootte = new uint[] {(uint)Data.Length};

int SignedDataSize = 0;

IntPtr pSignPara =
Marshal.AllocCoTaskMem(Marshal.SizeOf(typeof(_CRYPT_SIGN_MESSAGE_PARA)));
Marshal.StructureToPtr(SignPara, pSignPara, false);

if(Win32Alt.CryptSignMessage(pSignPara, 0, 1, ppData, BlobGrootte,
null, ref SignedDataSize) == 0)
{
showWin32Error(Marshal.GetLastWin32Error());
}

Marshal.FreeCoTaskMem(pData);
Marshal.FreeCoTaskMem(ppData);
Marshal.FreeCoTaskMem(pDataSize);

}
```