kamstrup

Kryptering af en KEM- eller KEM2-fil

1 Indholdsfortegnelse

Samplekoden i dette dokument er baseret på C# og .NetCore.

1 KEM2-filen

KEM2-filen er, ligesom en KEM-fil, et filformat til alle formål, der er beskrevet i dette dokument, men med den egenskab, at den signeres af Kamstrup med sin private nøgle.

Ved at verificere denne digitale signatur sikres både autenticiteten af dokumentet, der modtages fra Kamstrup, og integriteten af selve filen. For at verificere signaturen skal du gemme certifikatet med den offentlige nøgle i den personlige mappe i Local Machine Certificate Manager. Du kan downloade certifikatet her:

https://eks.kamstrup.com/certificates/Eks.Signing.PublicKey.cer

For at have en gyldig certifikationssti skal du også installere certifikatet i Trusted Root Certification Authorities store:

https://eks.kamstrup.com/certificates/Eks.Signing.HCWSSRootCA.cer

https://eks.kamstrup.com/certificates/Eks.Signing.CA.crt

Og det følgende certifikat skal installeres i mappen Intermediate Certification Authorities:

https://eks.kamstrup.com/certificates/Eks.Signing.C1.crt

Se afsnittet "Validere signaturen" for informationer om, hvordan du anvender certifikatet med den offentlige nøgle til validering af signaturen.

2 Downloade KEM-filen

Download filen fra Encryption Key Service-portalen. Vælg KEM, og indtast adgangskoden for at sikre filen.

For flere informationer om, hvordan du downloader en KEM-fil, see superuservejledningen.

3 Udtrække data

Downloadfilen skal se ud som i eksemplet nedenfor og ende på "**.zip.kem**" eller "**zip.kem2**".

```
20210407_1119DownloadMeters.zip.kem
20210812_1131_selected_DownloadDevices.zip.kem2
```

Slette udvidelse

Omdøb filen, og slet udvidelsen "**.kem**" eller "**.kem2**" for at konvertere den til en ZIP-fil.

Unzippe

Outputmappen vil se ud som filerne nedenfor:

- 363A6D7848DF4882B0991674191A9B84.kem
- meter_information_file.xsd

Skemafil

Filen, der ender på "**.xsd**" indeholder skemaet for dataene, når de er blevet dekrypterede. Den kan anvendes til at validere, at outputtet er i det forventede format.

Vær opmærksom på, at skemaet for nogle enheder kan se anderledes ud (device_information_file.xsd).

Datafil

Filen, der ender på "**.kem**" indeholder dataene. Dette er en XML-fil, som skal se ud som følger:

```
<?xml version="1.0" encoding="utf-8"?><EncryptedData
Type="http://www.w3.org/2001/04/xmlenc#aes128-cbc"
xmlns="http://www.w3.org/2001/04/xmlenc#">

<EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc" />
     <CipherData>
           <CipherValue>
               Nd7AyPuTPjsRgvS0l3ExFLqmsAgKDzOQYBQToY23/4g7ArcUF3RXEuYhwY2Uskdlq
               knRyvmZSuTlrNqctiafLB5c9pgpEZ0KydL1Sm26OmhsZjMzg2qhcLPoA3DV/aA97s
               aotIlxYk3Pq2DmpkJtA8Mw4Kh57TirNOcFB3mhTN4V3GjwevmNib2nZ22EvU0V17i
               GHCQPNMl3VN5lK5Ev8QnNT4Mjtlr3d0P7iWkO3w7H35lYPdoDZzaZiCluCTxKUZMW
               TGTRRcUHgvDifSmCXhmpZS7f4AA+3S9x2LUB6vdHlQYh5FxQT/tsy/mLVphzDYXEE
               Ej+TLG7Y3g59T0WVf6h+x8J0Rh5SGWfsV58F6+276cDkfMX7A8KanU6owlW6MFg/s
               5s1RyK397mjDw9UJh9bIJuTc5Yeq6D8I3K5c5EyF+MYDGUo+SxwXbrX/j0UJkmua/
               w4yhzBSvccF0yGQXS4QpVEK8yq+dRHvtpac8ZCXRS2OLP+eZt8qCQJDmpJ/M02kgx
               UAo9Ocd30EBokXLxuwjOqxeZ/ZNTG3vFtn89dMhW+QtL8k8UL1hBkvaCClApLoi38
               RVCFd/D9LIGJBDtsa20eu9miHX5CraLaLzsoXYXE8ijSU5u2we4N3mSGBzJWam/2i
               JkHIxrGpRYqovNDHT7BHd/EVwGlzbekbVueWm8fjcnkXGY5GtxnciuIUJ4rvfvdmq
               tcFXOR5iAx0RK25Y0/I6mmj3+S0f24hGatmzNfVWmNqFXmybO717i+sQMfyflTli/
               RY6Eeb/0G3qABv+NIWJP+8p2CSDCC59LmrsDWFtDxsEBn+RQnTMT+vh4VDPOFEuM1
               SNlvvHozlwsIV0jdQxw1muc7yVzeaHZoGIJe+fculeB4VLsaGxZVOHnd3R6iwYc1v
               z7r1KiDXlh/F0zORtaAowoqglPNfQ=
           </CipherValue>
     </CipherData>
</EncryptedData>
```

Konvertere fil til data

De krypterede data er værdien af indholdet med tagget "**CipherValue**" i datafilen. Indlæs filen som XML, søg efter tagget, og konverter den til byte array.

```
var encryptedXmlDocuent = new XmlDocument();
encryptedXmlDocuent.Load(kemFilePath);
var encryptedElement = encryptedXmlDocuent.GetElementsByTagName("CipherValue")[0]
as
XmlElement;
var encryptedData = Convert.FromBase64String(encryptedElement.InnerText);
```

4 Dekryptere data

Sikker adgangskode

Du skal konvertere adgangskoden, som du indtastede, da du downloadede filen, til en SecureString.

```
SecureString pwd = plainPwd.ConvertToSecureString();
public static SecureString ConvertToSecureString(this string unsecurePassword)
{
    if (unsecurePassword == null)
    {
        throw new ArgumentNullException("unsecurePassword");
    }

    var securePassword = new SecureString();
    foreach (var c in unsecurePassword)
        securePassword.AppendChar(c);
    securePassword.MakeReadOnly();
    return securePassword;
}
```

Dekryptere

Med nøglen og de krypterede data kan dekrypteringsværktøjet dekryptere og konvertere dataene.

Dette er markeret som usikkert. Derfor skal selve projektet markeres som tilladt for at kunne anvende en usikker kode.

```
public static unsafe byte[] GetDecryptedData(SecureString key, byte[] textToDe-
crypt)
{
    const int keyLength = 0x10;
    byte[] decryptedText;
    var keyAs16Bytes = new byte[keyLength];

    //validations

    var unmanagedBytes = Marshal.SecureStringToGlobalAllocAnsi(key);
    try
    {
        byte* byteArray = (byte*)unmanagedBytes;
        var pEnd = byteArray;
        while (*pEnd++ != 0){}
        var length = (int)((pEnd - byteArray) - 1);
        for (var i = 0; i < length; ++i)
        {
            keyAs16Bytes[i] = *(byteArray + i);
        }

        using (var alg = new RijndaelManaged())
        {
            //User cipher block chaning
            alg.Mode = CipherMode.CBC;
            alg.Padding = PaddingMode.PKCS7;

            // 128 bit key
            alg.KeySize = 0x80;
            alg.BlockSize = 0x80;

            // Secret key
            alg.Key = keyAs16Bytes;
            // Initialation Vector
            alg.IV = keyAs16Bytes;

            decryptedText = alg.CreateDecryptor().TransformFinalBlock(textToDe-
crypt, 0, textToDecrypt.Length);
        }
    }
    catch (CryptographicException e)
    {
        throw new Exception(“CryptographicException while decrypting document.”,
          e);
    }
    finally
    {
        Marshal.ZeroFreeGlobalAllocAnsi(unmanagedBytes);
        Array.Clear(keyAs16Bytes, 0, keyLength);
    }

    return decryptedText;
}
```

Valideringer

I det følgende vises nogle af de anbefalede kontroller, som kan udføres før dekrypteringen for at undgå yderligere undtagelser.

```
   if (textToDecrypt == null || textToDecrypt.Length == 0)
    {
        throw new Exception("No encrypted data");
    }

    if (key == null || key.Length == 0)
    {
        throw new Exception("Key cannot be null or empty");
    }

    if (key.Length > keyLength)
    {
       throw new Exception("Key size is wrong. The key can not be larger than " + key-
Length);
  }
```

Output

Ved at konvertere outputtet til en string, bliver det nemmere at læse den, og det er muligt at oprette et XML-dokument.

```
decrtyptedText = Encoding.UTF8.GetString(decryptedData);
```

```
<MetersInOrder orderid="" schemaVersion="2.0">
    <Meter>
        <MeterNo>78127978</MeterNo>
        <SerialNo>78127978</SerialNo>
        <EncKeys>
            <DEK>00000000000000000000000000000004</DEK>
        </EncKeys>
        <MeterName>MC602</MeterName>
        <ConsumptionType>Cooling</ConsumptionType>
        <ConfigNo>510002424003</ConfigNo>
        <ProgramNo>44458458</ProgramNo>
        <TypeNo>602A00000075DA</TypeNo>
        <VendorId>KAM</VendorId>
    </Meter>
    <Meter>
        <MeterNo>78177775</MeterNo>
        <SerialNo>78177775</SerialNo>
        <EncKeys>
            <DEK>00000000000000000000000000000004</DEK>
        </EncKeys>
        <MeterName>MC602</MeterName>
        <ConsumptionType>Heat</ConsumptionType>
        <ConfigNo>217002424003</ConfigNo>
        <ProgramNo>34451451</ProgramNo>
        <TypeNo>602C03870A1212</TypeNo>
        <VendorId>KAM</VendorId>
    </Meter>
</MetersInOrder>
```

5 Validere signaturen

Hente certifikatet

Først skal du gennemgå maskincertifikatlageret for at finde certifikatet med den offentlige nøgle. For at gøre dette skal du blot kende certifikatets thumbprint for at kunne genkende det (i dette tilfælde "14d7305cadf3c982a390c-94cc6e0054091b63531") og derefter loope X509Store for at hente og gemme det som en X509Certificate-klasse. Begge klasser er fra namespacet System.Security.Cryptography.X509Certificates.

```
var certThumbprint = "14d7305cadf3c982a390c94cc6e0054091b63531";

    X509Certificate2 cert = null;
    using (var store = new X509Store(StoreLocation.LocalMachine))
            {
                store.Open(OpenFlags.ReadOnly);

                foreach (var certificate in store.Certificates)
                {
                    if (certificate.Thumbprint == certThumbprint.ToUpper())
                    {
                        cert = certificate;
                        break;
                    }
                }
                store.Close();
            }
```

Anvende certifikatet til validering af signaturen

Ved dekryptering af en KEM2-fil vil den dekrypterede tekst være anderledes:

Du skal tage SignedXml-klassen fra namespacet SignatureSystem.Security.Cryptography.Xml, indlæse "Signature"-tagget for denne klasse og kalde CheckSignature-metoden med den offentlige nøgle fra X509Certificate-klassen.

```
var decryptedXmlDocument = new XmlDocument();
decryptedXmlDocument.LoadXml(decryptedText);

var signedXml = new SignedXml(decryptedXmlDocument);
signedXml.LoadXml((XmlElement)decryptedXmlDocument.GetElementsByTagName("Signature")[0]);
var verifySignature = signedXml.CheckSignature((RSA)cert.PublicKey.Key);
```

Kamstrup • 55123206_B1_DA_10.2021

Kamstrup A/S
Industrivej 28, Stilling
DK-8660 Skanderborg
T: +45 89 93 10 00
F: +45 89 93 10 01
info@kamstrup.com
kamstrup.com