

Decryptionfailed to bring original text back....

Source: <http://www.derkeiler.com/Newsgroups/microsoft.public.dotnet.security/2006-08/msg00039.html>

- *From:* den 2005 <den2005@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Wed, 19 Jul 2006 01:52:02 -0700
-

Hi everybody,

I am not sure where to put this in this forum. So, I posted this at several topics. I created a class library that has two public methods Encrypt() and Decrypt(). I reference this dll to a window application. I used DESCryptoServiceProvider Algorithm to encrypt and decrypt then with same Key and IV. But unable to decrypt it back to original text. This project I plan to use all algorithm and Hash. This is Phase One. There is no problem ingenerating the Key and IV and at both encrypt and decrypt they are the same. Can anyone spot the mistake and know how to correct this? Thanks.

[code]

```
//Generate a Key
private static void GenerateDESKey(DESCryptoServiceProvider
desProv,int keySize,bool maxKeySize)
{
if (Key == null)
{
if (keySize != 0)
desProv.KeySize = keySize;
else
{
KeySizes[] keySizeSets = desProv.LegalKeySizes;
int len = keySizeSets.Length;

for (int x = 0; x < len; x++)
{
if (maxKeySize)
keySize = keySizeSets[0].MaxSize;
else
keySize = keySizeSets[0].MinSize;
}
}
desProv.KeySize = keySize;
desProv.GenerateKey();
Key = desProv.Key;
}
}
```

Decryptionfailed to bring original text back....

```
//Generate a IV
private static void GenerateDESIV(DESCryptoServiceProvider desProv)
{
if (IV == null)
{
desProv.GenerateIV();
IV = desProv.IV;
}
}

//Encrypting String Data passed as parameter and returns it
public string Encrypt(string strData,int keySize, bool bMaxSize)
{
string strEncrypt = string.Empty;
try
{
//Variable Telling if Crypto or Managed object is selected
MemoryStream memStream = new MemoryStream();

CryptoStream cryptStream;
UnicodeEncoding byteConvert = new UnicodeEncoding();
byte[] byteData = byteConvert.GetBytes(strData);

byte[] encryptedData = { };

if (this.CRYPTOCLASS == Algorithm.DES.ToString())
{
this.CreateDESCrypto();
if (des != null)
{
//Generate Cryptographic Key and saved it
GenerateDESKey(des, keySize, bMaxSize);
//Generate Cryptographic IV and saved it
GenerateDESIV(des);

transform = des.CreateEncryptor((byte[])Key.Clone(),
(byte[])IV.Clone());
}
}
...
//Use the created algorithm object to encrypt data
cryptStream = new CryptoStream(memStream, transform,
CryptoStreamMode.Write);

cryptStream.Write(byteData, 0, byteData.Length);
cryptStream.FlushFinalBlock();

encryptedData = memStream.ToArray();
```

Decryptionfailed to bring original text back....

Decryptionfailed to bring original text back....

```
memStream.Close();
cryptStream.Close();
transform.Dispose();

//Call to dispose data
this.DisposeActiveObjects();
//Convert encrypted bytes[] back to string
strEncrypt = Convert.ToBase64String(encryptedData);

}
catch (Exception ex)
{
this.WriteAppendLogFile(" , Encrypt() " + ex.ToString());
}
return strEncrypt;
}

//Decrypting String Data passed as parameter and returns it
public string Decrypt(string strEncrypt)
{
string strData = string.Empty;
try
{
//Variable Telling if Crypto or Managed object is selected

//Check if Key and IV is still has data
if (Key == null || IV == null)
{
return "Cryptographic Key and IV cannot be null.";
}
MemoryStream memStream;
CryptoStream cryptStream;
byte[] encryptedData = Convert.FromBase64String(strEncrypt);
byte[] decryptedData = new Byte[encryptedData.Length];

if (this.CRYPTOCLASS == Algorithm.DES.ToString())
{
this.CreateDESCrypto();
transform = des.CreateDecryptor((byte[])
Key.Clone(),(byte[])IV.Clone());

}
.....

//Use the created algorithm object to encrypt data
memStream = new MemoryStream(encryptedData);
cryptStream = new CryptoStream(memStream, transform,
CryptoStreamMode.Read);

cryptStream.Read(decryptedData, 0, decryptedData.Length);
```

Decryptionfailed to bring original text back....

Decryptionfailed to bring original text back....

```
memStream.Close();
cryptStream.Close();
transform.Dispose();

//Call to dispose data
this.DisposeActiveObjects();
//Convert encrypted bytes[] back to string
//strEncrypt = Convert.ToBase64String(decryptedData);
strData = Encoding.ASCII.GetString(decryptedData);
}
catch (Exception ex)
{
this.WriteAppendLogFile(" , Decrypt() " + ex.ToString());
}
return strData;
}
[/code]
```

den2005

--
MCP Year 2005, Philippines

.