

Length of the data to decrypt is invalid

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.aspnet/2006-06/msg03957.html>

- *From:* "Hannibal111111" <Hannibal111111@xxxxxxxxxxxx>
 - *Date:* 27 Jun 2006 13:42:30 -0700
-

I found this code on a site for doing string encryption/decryption. The string will encrypt fine, but I get this error when I try to decrypt. Any idea why? I posted the code below.

The error actually points to this line of code in byte[] decrypt function:

```
cs.FlushFinalBlock();
```

```
public static byte[] encrypt(byte[] clearData, byte[] Key, byte[] IV)
{
    // Create a MemoryStream to accept the encrypted bytes
    MemoryStream ms = new MemoryStream();
```

```
    // Create a symmetric algorithm.
    // We are going to use Rijndael because it is strong and
    // available on all platforms.
    // You can use other algorithms, to do so substitute the
    // next line with something like
    // TripleDES alg = TripleDES.Create();
    Rijndael alg = Rijndael.Create();
```

```
    // Now set the key and the IV.
    // We need the IV (Initialization Vector) because
    // the algorithm is operating in its default
    // mode called CBC (Cipher Block Chaining).
    // The IV is XORed with the first block (8 byte)
    // of the data before it is encrypted, and then each
    // encrypted block is XORed with the
    // following block of plaintext.
    // This is done to make encryption more secure.
```

```
    // There is also a mode called ECB which does not need an IV,
    // but it is much less secure.
    alg.Key = Key;
    alg.IV = IV;
```

```
    // Create a CryptoStream through which we are going to be
```

Length of the data to decrypt is invalid

```
// pumping our data.
// CryptoStreamMode.Write means that we are going to be
// writing data to the stream and the output will be written
// in the MemoryStream we have provided.
CryptoStream cs = new CryptoStream(ms,
alg.CreateEncryptor(), CryptoStreamMode.Write);

// Write the data and make it do the encryption
cs.Write(clearData, 0, clearData.Length);

// Close the crypto stream (or do FlushFinalBlock).
// This will tell it that we have done our encryption and
// there is no more data coming in,
// and it is now a good time to apply the padding and
// finalize the encryption process.
cs.Close();

// Now get the encrypted data from the MemoryStream.
// Some people make a mistake of using GetBuffer() here,
// which is not the right way.
byte[] encryptedData = ms.ToArray();

return encryptedData;
}

/// <summary>
/// Encrypt a string into a string using a password
/// </summary>
/// <param name="clearText"></param>
```