

# How do I Use DPAPI to Encrypt and Decrypt Data (C#/VB.NET)?

**Source:**

<http://www.derkeiler.com/Newsgroups/microsoft.public.dotnet.framework.aspnet.security/2005-03/0196.html>

---

*anoniako\_at\_hotmail.com*

**Date:** 03/17/05

Date: 16 Mar 2005 15:54:45 -0800

## How To: Use DPAPI to Encrypt and Decrypt Data (C#/VB.NET)

The code below demonstrates how to call Data Protection API (DPAPI) functions `CryptProtectData` and `CryptUnprotectData` to encrypt and decrypt data. The code sample is provided in C# and Visual Basic.NET.

### Introduction

DPAPI functions encrypt and decrypt data using the Triple-DES algorithm. In addition to encryption and decryption, the API handles key generation and protection. DPAPI can generate two types of encryption keys: user- or machine-specific (these key types are commonly referred to as user store and machine store). User store and machine store are mutually exclusive; this means that you cannot combine a user-specific key with machine-specific key in one DPAPI call.

### DPAPI with user store

When making DPAPI calls with user-specific keys, encryption and decryption must be performed by the same user (i.e. the identity under which the application runs). Applications calling DPAPI functions with user-specific keys must run with loaded user profiles of Windows® domain or local accounts; they cannot use the profiles of the built-in system accounts, such as `LocalSystem`, `ASPNET`, `IUSR_MachineName`, etc. The user profile must be created on the system where DPAPI calls are made, which normally requires the user to log on to the system interactively at least once. **IMPORTANT:** ASP.NET applications and other programs running under the built-in system accounts, such as Windows® services running as `LocalSystem`, cannot use DPAPI with user-specific keys.

### DPAPI with machine store

When making DPAPI calls with machine-specific keys, encryption and decryption can be performed by any user or application as long as both operations are executed on the same computer. Any application – including ASP.NET – can use DPAPI with machine-specific keys. It is

worth noting that this option is not secure, because it allows a malicious application installed on a system to decrypt any data encrypted by other applications on the same system using DPAPI with machine-specific keys.

#### Secondary entropy

When an application calls the DPAPI encryption function, it can specify an optional secondary entropy ("secret" bytes) which will have to be provided by an application attempting to decrypt data. The secondary entropy can be used with either user- or machine-specific keys. It must be protected.

#### Data description

When an application calls the DPAPI encrypti

```
' computer where data were encrypted to perform decryption.
```

```
' Note: If optional entropy is specified, it will be required  
' for decryption.
```

```
' </param>
```

```
' <param name="plainTextBytes">
```

```
' Plaintext data to be encrypted.
```

```
' </param>
```

```
' <param name="entropyBytes">
```

```
' Optional entropy which – if specified – will be required to  
' perform decryption.
```

```
' </param>
```

```
' <param name="description">
```

```
' Optional description of data to be encrypted. If this value is  
' specified, it will be stored along with encrypted data and  
' returned as a separate value during decryption.
```

```
' </param>
```

```
' <returns>
```

```
' Encrypted value.
```

```
' </returns>
```

```
Public Shared Function Encrypt _
```

```
( _
```

```
    ByVal keyType As KeyType, _
```

```
    ByVal plainTextBytes As Byte(), _
```

```
    ByVal entropyBytes As Byte(), _
```

```
    ByVal description As String _
```

```
) As Byte()
```

```
    ' Make sure that parameters are valid.
```

```
    If plainTextBytes Is Nothing Then
```

```
        plainTextBytes = New Byte(0){}
```

```
    End If
```

```
    If entropyBytes Is Nothing Then
```

```
        entropyBytes = New Byte(0){}
```

```
    End If
```

```
    If description Is Nothing Then
```

```
        description = String.Empty
```

```
End If

' Create BLOBs to hold data.
Dim plainTextBlob As DATA_BLOB = New DATA_BLOB
Dim cipherTextBlob As DATA_BLOB = New DATA_BLOB
Dim entropyBlob As DATA_BLOB = New DATA_BLOB

' We only need prompt structure because it is a required
' parameter.
Dim prompt As _
    CRYPTPROTECT_PROMPTSTRUCT = New
CRYPTPROTECT_PROMPTSTRUCT
InitPrompt(prompt)

Try
    ' Convert plaintext bytes into a BLOB structure.
    Try
        InitBLOB(plainTextBytes, plainTextBlob)
    Catch ex As Exception
        Throw New Exception("Cannot init
```