CryptUnprotectData function (dpapi.h) - Win32 apps

By GrantMeStrength

Archived: 2026-04-05 18:41:16 UTC

The **CryptUnprotectData** function decrypts and does an integrity check of the data in a DATA_BLOB structure. Usually, the only user who can decrypt the data is a user with the same logon credentials as the user who encrypted the data. In addition, the encryption and decryption must be done on the same computer. For information about exceptions, see the Remarks section of CryptProtectData.

Syntax

```
DPAPI_IMP BOOL CryptUnprotectData(
  [in]            DATA_BLOB                 *pDataIn,
  [out, optional] LPWSTR                    *ppszDataDescr,
  [in, optional]  DATA_BLOB                 *pOptionalEntropy,
                  PVOID                     pvReserved,
  [in, optional]  CRYPTPROTECT_PROMPTSTRUCT *pPromptStruct,
  [in]            DWORD                     dwFlags,
  [out]           DATA_BLOB                 *pDataOut
);
```

Parameters

`[in] pDataIn`

A pointer to a DATA_BLOB structure that holds the encrypted data. The **DATA_BLOB** structure's **cbData** member holds the length of the **pbData** member's byte string that contains the text to be encrypted.

`[out, optional] ppszDataDescr`

A pointer to a string-readable description of the encrypted data included with the encrypted data. This parameter can be set to **NULL**. When you have finished using *ppszDataDescr*, free it by calling the LocalFree function.

`[in, optional] pOptionalEntropy`

A pointer to a DATA_BLOB structure that contains a password or other additional entropy used when the data was encrypted. This parameter can be set to **NULL**; however, if an optional entropy **DATA_BLOB** structure was used in the encryption phase, that same **DATA_BLOB** structure must be used for the decryption phase. For information about protecting passwords, see Handling Passwords.

`pvReserved`

This parameter is reserved for future use and must be set to **NULL**.

`[in, optional] pPromptStruct`

A pointer to a CRYPTPROTECT_PROMPTSTRUCT structure that provides information about where and when prompts are to be displayed and what the content of those prompts should be. This parameter can be set to **NULL**.

`[in] dwFlags`

A **DWORD** value that specifies options for this function. This parameter can be zero, in which case no option is set, or the following flag.

Value	Meaning
CRYPTPROTECT_UI_FORBIDDEN	This flag is used for remote situations where the user interface (UI) is not an option. When this flag is set and UI is specified for either the protect or unprotect operation, the operation fails and GetLastError returns the ERROR_PASSWORD_RESTRICTION code.
CRYPTPROTECT_VERIFY_PROTECTION	This flag verifies the protection of a protected BLOB. If the default protection level configured of the host is higher than the current protection level for the BLOB, the function returns **CRYPT_I_NEW_PROTECTION_REQUIRED** to advise the caller to again protect the plaintext contained in the BLOB.

`[out] pDataOut`

A pointer to a DATA_BLOB structure where the function stores the decrypted data. When you have finished using the **DATA_BLOB** structure, free its **pbData** member by calling the LocalFree function.

Return value

If the function succeeds, the function returns **TRUE**.

If the function fails, it returns **FALSE**.

The CryptProtectData function creates a session key when the data is encrypted. That key is derived again and used to decrypt the data BLOB.

The Message Authentication Code (MAC) hash added to the encrypted data is used to detect whether the encrypted data was altered in any way. However, the specific error code returned when tampering is detected may vary depending on the nature of the corruption. The function may return ERROR_INVALID_DATA, ERROR_INVALID_PARAMETER, or in some cases may succeed with corrupted output. Applications should not

rely on a specific error code to detect data tampering. For robust tamper detection, consider implementing additional integrity checks at the application level.

When you have finished using the DATA_BLOB structure, free its **pbData** member by calling the LocalFree function. Any *ppszDataDescr* that is not **NULL** must also be freed by using **LocalFree**.

When you have finished using sensitive information, clear it from memory by calling the SecureZeroMemory function.

Examples

The following example shows decrypting encrypted data in a DATA_BLOB structure. This function does the decryption by using a session key that the function creates by using the user's logon credentials. For another example that uses this function, see Example C Program: Using CryptProtectData.

```
 // Decrypt data from DATA_BLOB DataOut to DATA_BLOB DataVerify.

//-------------------------------------------------------------------
// Declare and initialize variables.

DATA_BLOB DataOut;
DATA_BLOB DataVerify;
LPWSTR pDescrOut =  NULL;
//-------------------------------------------------------------------
// The buffer DataOut would be created using the CryptProtectData
// function. If may have been read in from a file.

//-------------------------------------------------------------------
//   Begin unprotect phase.

if (CryptUnprotectData(
        &DataOut,
        &pDescrOut,
        NULL,                 // Optional entropy
        NULL,                 // Reserved
        NULL,                 // Here, the optional
                              // prompt structure is not
                              // used.
        0,
        &DataVerify))
{
     printf("The decrypted data is: %s\n", DataVerify.pbData);
     printf("The description of the data was: %s\n",pDescrOut);
     LocalFree(DataVerify.pbData);
     LocalFree(pDescrOut);
}
```

```
else
{
    printf("Decryption error!");
}
```

Requirements

Requirement	Value
Minimum supported client	Windows XP [desktop apps | UWP apps]
Minimum supported server	Windows Server 2003 [desktop apps | UWP apps]
Target Platform	Windows
Header	dpapi.h
Library	Crypt32.lib
DLL	Crypt32.dll

See also

CryptProtectData

CryptUnprotectMemory

Data Encryption and Decryption Functions

LocalFree

Microsoft Base Cryptographic Provider
