最近使用Microsoft CryptoAPI的过程中，发现了其中两个函数的一些问题，做此笔记。

**BOOL WINAPI** **CryptEncrypt(**

**HCRYPTKEY** *hKey***,**

**HCRYPTHASH** *hHash***,**

**BOOL** *Final***,**

**DWORD** *dwFlags***,**

**BYTE\*** *pbData***,**

**DWORD\*** *pdwDataLen***,**

**DWORD** *dwBufLen*

**);**

需要注意其中的三个参数：

一、BOOL Final

*Final*

[in] Boolean value that specifies whether this is the last section in a series being encrypted.

*Final* is set to TRUE for the last or only block and to FALSE if there are more blocks to be encrypted.

For more information, see Remarks.

说的比较清楚，若只有一个分组的数据需要加密或者为最后一个分组，则Final为TRUE。

二、DWORD\* pdwDataLen

*pdwDataLen*

[in, out] Pointer to a **DWORD** value that contains the length of the data buffer. Upon input, the **DWORD** value is set to the number of bytes to be encrypted. Upon return, the **DWORD**value contains the number of bytes needed to hold the encrypted data.

在输入时，pdwDataLen为需要加密的分组长度，如使用DES对64位数据进行加密，那么输入时

pdwDataLen为8；在输出时，pdwDataLen为保存密文所需要的字节数。

If the buffer allocated for *pbData* is not large enough to hold the encrypted data,GetLastError returns ERROR\_MORE\_DATA and stores the required buffer size, in bytes, in the **DWORD** value pointed to by *pdwDataLen*.

如果pbData，即需要加密的数据的缓冲区长度不够保存密文所需要的字节数，那么GetLastError返回

ERROR\_MORE\_DATA，并且pdwDataLen为需要的字节数。所以为了正确的加密，需要将待加密数据的

缓冲区长度设得更长一点，如上例，不妨设为16个字节。

If *pbData* is NULL, no error is returned, and the function stores the size of the encrypted data, in bytes, in the **DWORD** value pointed to by *pdwDataLen*. This lets an application unambiguously determine the correct buffer size.

If a *block cipher* is used, this data length must be a multiple of the block size unless this is the final section of data to be encrypted and the *Final* parameter is set to TRUE.

三、DWORD dwBufLen

*dwBufLen*

[in] **DWORD** value that specifies the length, in bytes, of the input *pbData* buffer.

Note that, depending on the algorithm used, the encrypted text can be larger than the original plaintext. In this case, the *pbData* buffer needs to be large enough to contain the encrypted text and any padding.

As a rule, if a *stream cipher* is used, the *ciphertext* is the same size as the plaintext. If a block cipher is used, the ciphertext is up to a block length larger than the plaintext.

表示pbData缓冲区的长度。需要注意的是，取决于所使用的[**算法**](http://lib.csdn.net/base/31)，密文的长度可能会大于明文的长度。

 在这种情况下，pbData缓冲区的长度需要足够大以保证可以存储密密文及填充数据。

如果使用流密码，那么密文和明文的长度是相同的；如果使用分组密码，那么密文比明文多一个分组的

长度。

同样，CryptDecrypt的几个参数类似于上面的解释，也需要注意。

*The*[*Microsoft Enhanced Cryptographic Provider*](ms-help://MS.MSDNQTR.v80.en/MS.MSDN.v80/MS.WIN32COM.v10.en/seccrypto/security/microsoft_enhanced_cryptographic_provider.htm)*supports direct encryption with RSA public keys and decryption with RSA private keys. The encryption uses PKCS #1 Type 2 padding. On decryption, this padding is verified. The length of plaintext data that can be encrypted with a call to****CryptEncrypt****with an RSA key is the length of the key modulus minus eleven bytes. The eleven bytes is the chosen minimum for PKCS #1 padding. The ciphertext is returned in little-endian format.*

在使用RSA密钥，调用CryptEncrypt对明文进行加密时，明文的长度最大为模的长度减去11个字节。

另外，还有一个函数的用法：NetGetJoinInformation

需要有如下代码方能正常使用：

#include <lm.h>

#include <lmjoin.h>

#pragma comment(lib,"NetAPI32.lib")

LPWSTR lpServer;

PNETSETUP\_JOIN\_STATUS buftype;

buftype=new NETSETUP\_JOIN\_STATUS;

NetGetJoinInformation(NULL,&lpServer,buftype);

if(lpServer!=NULL)

{

     NetApiBufferFree(lpServer);

}

引用buftype时，只需要(\*buftype)即可。

顶

1