AES 加密工具程式

說明書

一、 前言

AES (Advanced Encryption System)是目前非常安全且運算效能相對快速的對稱式演算法, 以下摘自 Wikipedia 網站的說明:

(WikiPedia) In cryptography, the Advanced Encryption Standard (AES), also known as Rijndael, is a block cipher adopted as an encryption standard by the U.S. government. It is expected to be used worldwide and analysed extensively, as was the case with its predecessor, the Data Encryption Standard (DES). AES was announced by National Institute of Standards and Technology (NIST) as U.S. FIPS PUB 197 (FIPS 197) in November 26, 2001 after a 5-year standardization process (see Advanced Encryption Standard process for more details). It became effective as a standard May 26, 2002. As of 2006, AES is one of the most popular algorithms used in symmetric key cryptography.

AES 發明人爲 Rijndael, 而經 NIST 規範的標準 AES 演算法其金鑰長度可爲 128, 192 or 256 bits, 區塊大小則固定爲 16 bytes; 此外, 區塊串接模式(Chaining Mode)可爲 ECB, CBC 等, 由於每次的 AES 加密資料必需爲區塊大小, 亦即資料長度必需爲 16 的倍數, 不足 16 bytes 的部份應補(Padding)爲 16 bytes, 而 Padding 的方式又有多種可能. 造成一般應用系統開發人員要了解其細節將會非常困難.

本加密工具程式提供最容易使用的 API 供應用系統開發人員使用. 包括:

1. 三個基本的 AES 加密/解密/Hash 函數, 可供需要有上述參數彈性的開發人員使用
2. 兩個檔案加解密的函數 (Hash & non-Hash)
3. 一個專供兆豐金控(Mega International Commercial Bank, 原中國國際商銀 ICBC)全球金融網(Global E-Banking)加密傳輸檔案的函數

本加密工具程式內容包括一個包含上述 API 函數的 DLL 檔案 (aestool.dll), 本文件說明, 以及 VC 與 VB 的範例程式碼.

安裝方式: 只要將 aestool.dll 檔案複製到您程式目錄或 Windows 目錄(建議)下即可.

二、 API 使用說明

1. 全球金融網檔案加密函數

micb_ebanking_file_encryption(char far* infile, char far* outfile, char far *key, int action)

函數名稱: **micb_ebanking_file_encryption**
輸入參數:
infile – 要加密/解密的檔案名稱
outfile – 輸出的檔案名稱
key – 加密金鑰字串, 16 碼文數字串
action: 整數 1 代表要加密, 整數 2 代表要解密
返回值: 0 代表成功, 非 0 值代表失敗, 請參考 Return Code 一節之說明
說明: 本函數簡化各種 AES 演算法的參數, 只固定使用下列最常用的值
Keysize=16
Blocksize=16
OperationMode=CBC
PaddingMode=Zero(NULL)
使用本函數時, 記得在全球金融網站上傳檔的設定上選用以上之值
範例說明: 參考 VB 範例一節

```
rc = micb_ebanking_file_encryption("megabank_darfon.txt", "megabank_darfon.enc",
        "1234567890ABCDEF", 1)
if (rc=0) then
   msgbox("加密 megabank_darfon.txt 檔案成功, 產生加密檔 megabank_darfon.enc")
else
  msgbox("加密失敗, Return Code=" & str(rc))
endif
```

附註 1: 產生的檔案爲 Read-Only, 如要重覆寫入相同檔案, 請將其移除後再執行, 否則將得到檔案開啓失敗之訊息.
附註 2: 另有 ebanking_file_encryption()函數, 供不需要在加密檔案前加 32 bytes hash value 使用.

2. AES 檔案加密函數

ap_aes_file_encryption(char far* infile, char far* outfile, unsigned char far *keyvalue, int keystrtype, int keylen, int mode)

函數名稱: **ap_aes_file_decryption**
輸入參數:

infile – 要加密的檔案名稱
outfile – 輸出的檔案名稱
keyvalue – 加密金鑰, 文數字串或 16 進位碼(Hex String)
keystrtype – 整數 1 代表 keyvalue 爲文數字串, 整數 2 代表 keyvalue 爲 16 進位碼
keylen: 金鑰長度, 必需爲 16, 24, 32; 若 keystrvalue=1 則 keyvalue 字串長度應等於 keylen; 若 keystrvalue=2, 則 keyvalue 字串長度應等於 keylen 的兩倍
mode: Chaining Mode, 整數 0 代表 ECB, 整數 1 代表 CBC

返回值: 0 代表成功, 非 0 值代表失敗, 請參考 Return Code 一節之說明
說明: 此函數除加密輸入檔案內容外, 在輸出檔案最前面內容加上加密參數的 Hash Value, 此加密參數字串格式爲：
”RIJNDAEL”+”加密金鑰(密碼)”+Blocksizes+Key sizes+Block mode+Padding mode; Hash 演算法使用 SHA-256. 所以輸出的檔案大小會大於原來輸入的檔案大小 (其原因爲, 加密過的資料大小一定爲 16 的倍數, 再加上此 Hash Value)
範例:

ap_aes_file_encryption(“mydata.txt”, “mydata.enc”, “1234567890123456”,1, 16,1)
此呼叫將檔案”mydata.txt”加密後存放爲檔案”mydata.enc”, 金鑰長度爲 16 bytes, 金鑰值爲”1234567890123456”, 使用 CBC Chaining mode

附註 1: 產生的檔案爲 Read-Only, 如要重覆寫入相同檔案, 請將其移除後再執行, 否則將得到檔案開啓失敗之訊息.
附註 2: 另有 ap_aes_file_encryption_nohash ()函數, 供不需要在加密檔案前加 32 bytes hash value 使用.

3. AES 檔案解密函數

ap_aes_file_decryption(char far* infile, char far* outfile, unsigned char far *keyvalue, int keystrtype, int keylen, int mode)

函數名稱: **ap_aes_file_decryption**
輸入參數:

infile – 要解密的檔案名稱
outfile – 輸出的檔案名稱
keyvalue – 加密金鑰, 文數字串或 16 進位碼(Hex String)
keystrtype – 整數 1 代表 keyvalue 爲文數字串, 整數 2 代表 keyvalue 爲 16 進位碼
keylen: 金鑰長度, 必需爲 16, 24, 32; 若 keystrvalue=1 則 keyvalue 字串長度應等於 keylen; 若 keystrvalue=2, 則 keyvalue 字串長度應等於 keylen 的兩倍
mode: Chaining Mode, 整數 0 代表 ECB, 整數 1 代表 CBC

返回值: 0 代表成功, 非 0 值代表失敗, 請參考 Return Code 一節之說明
說明: 此函數先驗證輸入檔案內容的前面 Hash Value 是否無誤, 然後再將其餘檔案內容解密, 在存入輸出檔案, 此函數不將原來 Padding 成 16 倍數的 Padding Character(NULL)刪除, 故其檔案長度可能大於原始未加密的檔案
範例:

ap_aes_file_encryption(“mydata.enct”, “mydata.txt”, “1234567890123456”,1, 16,1)
此呼叫將檔案”mydata.txt”解密後存放爲檔案”mydata.txt”, 金鑰長度爲 16 bytes, 金鑰值爲”1234567890123456”, 使用 CBC Chaining mode

附註 1: 產生的檔案爲 Read-Only, 如要重覆寫入相同檔案, 請將其移除後再執行, 否則將得到檔案開啓失敗之訊息.
附註 2: 另有 ap_aes_file_decryption_nohash ()函數, 供不需要在加密檔案前加 32 bytes hash value 使用

4. AES 資料加密函數

ap_aes_encrypt(unsigned char far* InData, int inLen, unsigned char far* OutData, unsigned char far* UserKey, int KeyLen, unsigned char far* icv, int cmode)

函數名稱: **ap_aes_encryption**
輸入參數:
InData – 要加密的資料
inLen – 要加密的資料長度
OutData – (Output) 加過密後的資料, 使用此函數時必須確認此變數有足夠長度容納加密後的資料 (加密後的資料長度將爲 16 的倍數)
UserKey-- 加密金鑰
KeyLen: 金鑰長度, 必需爲 16, 24, 32
icv: 若爲 CBC 模式, 可指定此 16 byte 的值, 否則給 NULL 即可
mode: Chaining Mode, 整數 0 代表 ECB, 整數 1 代表 CBC
返回值: 正整數代表加密後的資料長度, 非 0 負值代表失敗, 請參考 Return Code 一節之說明
範例說明: 參考 C 程式範例

5. AES 資料解密函數

ap_aes_decrypt(unsigned char far* InData, int inLen, unsigned char far* OutData, unsigned char far* UserKey, int KeyLen, unsigned char far* icv, int cmode)

函數名稱: **ap_aes_decryption**
輸入參數:
InData – 要解密的資料
inLen – 要解密的資料長度, 應爲 16 的倍數
OutData – (Output) 解完密後的資料, 使用此函數時必須確認此變數有足夠長度容納解密後的資料, 本函數不刪除 Padding Char, 故解密後的資料長度應等於原來輸入的長度
UserKey-- 加密金鑰
KeyLen: 金鑰長度, 必須爲 16, 24, 32
icv: 若爲 CBC 模式, 可指定此 16 byte 的值, 否則給 NULL 即可
mode: Chaining Mode, 整數 0 代表 ECB, 整數 1 代表 CBC
返回值: 正整數代表解密後的資料長度, 非 0 負值代表失敗, 請參考 Return Code 一節之說明
範例說明: 參考 C 程式範例

6. SHA-256 Hash 函數

ap_aes_sha256(unsigned char far* InData, int inLen, unsigned char far* OutData)

函數名稱: **ap_aes_sha256**
輸入參數:
InData – 要 Hash(湊雜)的資料
inLen – 輸入的資料長度
OutData – (Output) Hash 值, 使用此函數時必須確認此變數有足夠長度, 無論輸入資料多長, 其 Hash 值均爲 32 bytes
返回值: 整數. 目前暫定永遠成功
範例說明: 參考 C 程式範例

7. VB 程式使用說明

請在 VB 程式的模組檔裡宣告使用 aestool.dll 裡的上述函數, 例如:
Declare Function micb_ebanking_file_encryption Lib "aestool.dll" (ByVal infile As String, ByVal outfile As String, ByVal key As String, ByVal action As Long) As Long

請注意 Call by Value 與 Call by Reference 的問題, 及 VB 與 C 在資料格式(Data type)上的差異性.

詳細資料請參考 VB 相關文件

8. C 程式使用說明

請將 aestool.lib 連結到你的 C 專案裡

宣告引用 DLL 函數, 例如:

```
extern "C" _declspec(dllimport) int __stdcall ap_aes_encrypt(unsigned char *
        InData, int inLen, unsigned char * OutData,    unsigned char * UserKey,
        int KeyLen, unsigned char *icv, int cmode);
extern "C" _declspec(dllimport) int __stdcall ap_aes_decrypt(unsigned char *
        InData, int inLen, unsigned char * OutData,    unsigned char * UserKey,
        int KeyLen, unsigned char *icv, int cmode);
```

```
extern "C" _declspec(dllexport) int __stdcall ap_aes_sha256(unsigned char *
        InData, int inLen, unsigned char * OutData);
```

詳情請參考 C Compiler 相關文件

9. 使用的常數

```
//
// Constants used in DLL and external calling parameter
//
#define KEYLEN16 16
#define KEYLEN24 24
#define KEYLEN32 32
#define ECB 0
#define CBC 1
#define ASCSTR 1
#define HEXSTR 2

#define AESNAME "RIJNDAEL"
#define BLOCKSIZE 16
#define HASHSIZE 32
```

10. 函數傳回值與錯誤碼

```
//
//    Return codes from DLL functions
//
#define RT_AES_OK                    0          // Sucessful
#define RT_AES_INVALID_MODE    -100     // Invalid Cipher Mode, only CBC and ECB supported
#define RT_AES_INVALID_KEYLEN   -101     // Invalid Key Len
#define RT_ARE_INVALID_STRTYPE  -102     // Invalide String Type
#define RT_ARE_INVALID_DATALEN -103     // Data length error
#define RT_AES_FILEOPEN_ERROR     -200     // Open file fail
#define RT_AES_MEMALLOC_ERROR     -201     // Memory Allocation Error
#define RT_AES_HASHVAULE_ERROR    -202     // Invalid Hash value
#define RT_AES_INVALIDACTION       -203     // Action parameter error
#define RT_AES_SETENCRYPTKEY_ERROR -300    // Internal Error, Please contact AsiaPeak
#define RT_AES_SETDECRYPTKEY_ERROR -301    // Internal Error, Please contact AsiaPeak
```

三、範例說明

1. VB6 範例

\v4 目錄下的 VB 範例使用 micb_ebanking_file_encryption 函數來加密指定的檔案. 此函數足夠用在 ICBC 全球金融網.上傳檔案. 請在該網頁選擇:
操作模式=CBC
填滿模式=ZEROES
金鑰長度=16
區塊長度=16
然後輸入您的金鑰值

```
Declare Function micb_ebanking_file_encryption Lib "aestool.dll" (ByVal infile As String, ByVal
        outfile As String, ByVal key As String, ByVal action As Long) As Long

Private Sub cmdEncrypt_Click()
    Dim rc As Long
    rc = micb_ebanking_file_encryption("megabank_darfon.txt", "megabank_darfon.enc",
            "1234567890ABCDEF", 1)
    MsgBox ("micb_ebanking_file_encryption RC=" & Str(rc))
End Sub
```

\v5 目錄是一個不需要 Hash 值驗證的 AES 加密檔案範例

2. VC6 範例

\t4 目錄下的 VC 範例使用
ap_aes_encrypt
ap_aes_decrypt
ap_aes_sha256
三個函數來完成檔案加密，產生的加密檔案也可上傳 ICBC 全球金融網.

\t5 目錄下的 VC 測試範例則直接使用
ap_aes_file_encryption
這個函數來加密檔案