上海市电子商务安全证书管理中心有限公司

证书管理器
Ver 2.0
API 函数接口编程说明

一、获取证书管理器中证书列表

1、int CMB_InitialDataBase(struct dbcontext *contxt);
建立证书管理器数据库连接，并初始化数据库环境

2、int CMB_GetUserCertificateLists(struct dbcontext *contxt,
int*CertMgrCertId,
char*commonName,
char*ViaName,
char*CretSerialNo,
char*status,
char *PEMCert
char*CertPcertIDList,
char*CertPcertNameList,
int * CertPLen);
反复执行，每执行一次，返回一条记录，直至返回非零

3、int CMB_CloseDataBase(struct dbcontext *);
断开数据库连接，并释放数据库环境

编程示例：

```
#include <ShecaMgrCert.h>

char * UserName, *ViaName, *CertPCertIdList, *CertPNameList, * Status, *PemCert, *CertSerialNo;
char CertPName[64],CertViaName[64],CertPCerId[8];
int CertMgrCertid,CertPNum;
int i;

UserName = ( char * )malloc(64);
ViaName = ( char * )malloc(64);
CertPCertIdList = ( char * )malloc(128);
CertPNameList = ( char * )malloc(8*64);
sStatus   = ( char * )malloc(8);
```

```
PEMCert  = ( char * )malloc(2048);
CertSerial=( char * )malloc(48);
struct dbcontext db;


if(CMB_InitialDataBase(&db)!=0)
{
      free(UserName);
      free(ViaName);
      free(CertSerial);
      free(CertPCertIdList);
      free(CertNameList);
      free(Status);
      free(PEMCert);
}

while(CMB_GetUserCertificateLists(&db,&CertMgrCertId,UserName,ViaName,CertSerial,Stat
us,
PEMCert,CertPCertIdList,CertPNameList,&CertPNum)==0)
      {
       ……..
// Notice:
//CertMgrCertId : 从证书管理器中取得的证书标识号.考虑到区别用户级证书和根级证书的
// 区别，用户级证书的标识号=证书标识号 + 10000, 根级证书的标识号=证书标识号.
// 故对于用户级证书, CertMgrCertId 可能为 10028, 证书标识号 CertId 应为 28.
// 故对于根级证书, CertMgrCertId 可能为 28, 证书标识号 CertId 也为 28.
//CerPCertIdlist : 证书链中根级证书标识号列表,每个标识号用八个字符表示，如
00001024000102300001022，子级根证书在前，父级根证书在后。
//CertPNameList : 证书链中根级证书名称列表, 每个名称用 64 个字符表示，其中含字符串
结束符（'\0'）; 如  SHECA'\0'…….Beijing CA'\0'……..BeijingBOCC SubCA'\0'…….
//CertPNum : 证书链中根级证书的个数。


      for(i=CertPNum;i>=1;i--)
      {
            strncpy(CertPCertId,&CertPCertIdList[8*(i-1)],8);
            CertId[8]='\0';
            strncpy(CertPName,&CertPNameList[64*(i-1)],64);
      }
     //Notice:
     //若当前证书没有别名，ViaName 项为字符"0"，有的话，返回证书别名。
      if(strcmp(ViaName,"0")==0)
      {
        //Current Cerificate No ViaName;
      }
      else
       {
          //Current Cetificate With ViaName
        }
  }
```

```
CMB_CloseDataBase(&db);

free(UserName);
free(CertPCertIdlist);
free(ViaName);
free(CertSerial);
free(Status);
free(PEMCert);
```

二、添加证书：由以下六个函数完成

1. int CMB_GetUserCertChain(unsigned short certificatedevicetype,
char* certificatedeviceparameter,
char* CertChainCetIdList,
char* CertChainNameList,
int* CertChainNums)

在证书管理器数据库中匹配用户证书对应的证书链。
返回相关信息（证书链中证书标识号列表，证书链中证书名称列表, 证书链中证书数目）

2. int CMB_AddUserCertificate(unsigned short certificatedevicetype,
char* certificatedeviceparameter,
char* certificatepassword,
unsigned short privatekeydevicetype,
char* privatekeydeviceparameter,
char* privatekeypassword,
int ParentCertId,
int*CertMgrCertId,
char * CertCommonName);

添加他人证书(无私钥)，可从他人发来的签名文件中获取其证书，也可从 Sheca 的 LDAP 服务器获取指定用户的证书。
返回相关信息（证书标识号，证书名称）

3. int CMB_AddUserCertAndKey(unsigned short certificatedevicetype,
char* certificatedeviceparameter,
char* certificatepassword,
unsigned short privatekeydevicetype
char* privatekeydeviceparameter,
char* privatekeypassword,
int ParentCertId,
int*CertMgrCertId,

char * CertCommonName);

将自己的证书和私钥添进数据库。

返回相关信息（证书标识号，证书名称）

编程示例：

```
#include “ShecaCertMgr.h”

char CertPCertIdList[64],CertPNameList[8*64];
char test[20],CertName[64];
int  retCode,PriKeyFlag,CertPNum,CertId,ParentCertId=0;
retCode=CMB_GetUserCertChain(1,”a:\UserCert.der”,
                                    CertPCertIdList,CertPNameList,&CertPNum);
if (retCode==0)
{
            AfxMessageBox("匹配证书链成功. ");
            strncpy(test,&CertPCertIdList[8*(CertPNum-1)],8);
            test[8]='\0';
            ParentCertId=atoi(test);
}
else {
            AfxMessageBox("匹配证书链出错. ");
            ParentCertId=0;
     }
*****************************
if (PriKeyFlag==1)
{
   //Add User Cert And Key Together
    intReturn = CMB_AddUserCertAndKey(
              1,”A:\UserCert.der”,”sheca”,
              1, “A:\UserKey.key”,”sheca”,
             ParentCertId,&CertId,CertName)
}
else
{
   //Add User Cetrificate Only
     intReturn = CMB_AddUserCertificate(
                      1,”A:\UserCert.der’,”sheca”,
                      1,NULL,NULL,
                       ParentCertId,&CertId,CertName);
}

*****************************
```

4. int CMB_InsertCertificate(char* certpem,bool overwrite);

插入证书，无返回信息

5. int CMB_InsertCertAndKey(char * serialno,
char * username,
char * email,
char* certpem,
char* skpem,
char*medium,
char* path,
bool overwrite);

插入证书与私钥，无返回信息

三、证书的导出、导入

1、CMB_ExportUserCertificate(char *CertId,int CertType,char *filename);
从证书管理器中导出 sheca 格式用户证书.

编程示例：
//导出标识号为 122 的用户证书到 A:\UserCert.der
CMB_ExportUserCertificate("122",0,"A:\UserCert.der");
//导出标识号为 12 的根级证书到 A:\UserCert.der
CMB_ExportUserCertificate("12",1,"A:\UserCert.der");

2、int int CMB_p12CertImport(char* filepath,char*password,
int *CertId, char * username,
char * CertPCertIdList,
char * CertPNameList,
int * CertPNum);

导入 p12 格式用户证书和私钥到证书管理器。
返回证书链信息。

3、CMB_p12CertExport(int CertId,char* filename,char* password);
从证书管理器中导出 p12 格式用户证书和私钥

四　证书的管理

1. int CMB_GetCertById(char * CertId,int CertType, char * PemCert);
从证书管理器中取得用户证书.

编程示例：
char UserPemCert[2048];

//取得标识号为 122 的用户 Pem 编码证书

CMB_GetCertById(“122”,0,UserPemCert);
//导出标识号为 12 的根级 Pem 编码证书 r
CMB_GetCertById(“12”,1,UserPemCert);

2. int CMB_GetCertId(char* PemCert,int certtype,
int * CertId,int * ParentCertId);
从证书管理器中查找用户证书对应的证书标识号和父证书标识号

编程示例：
char UserPemCert[2048];
int CertID, ParentCertId,

//By Some Way, User Got Pem Certificate in UserPemCert[2048]

//取得用户 Pem 编码证书的标识号
CMB_GetCertId(UserPemCert,0,&CertId,&ParentCertId);
//导出根级 Pem 编码证书的标识号
CMB_GetCertId(UsrePemCert,1,&CertId,&ParentCertId);

3. int CMB_GetCertStatus(int CertId, int CertType,
char * VerifyClass, char *CertStatus);
从证书管理器中取得证书状态。

4. int CMB_SetCertStatus(int CertId, int CertType,
char * VerifyClass, char *CertStatus);
设置证书状态。

5. int CMB_GetCertViaName(char *CertId[input], char *ViaName[input]);
获取证书别名

6. int CMB_SetCertViaName(char *CertId[input], char *ViaName[input]);
设置证书别名

7. int CMB_DeleteCertificate(char * CertId, int CetrType);
从证书管理器中删去用户证书。

编程示例：

//删除标识号为 122 的用户证书,若存在私钥，连同私钥一起删除。
CMB_DeleteCertificate(“122”,0);
//删除标识号为 12 的根级证书 r
CMB_DeleteCertificate(“12”,1);

8. int CMB_UpdateUserCertificate(int CertId,
int CertType,
int CertCheckMode,
char* CertStatus);

更新证书管理器中的用户证书.

编程示例：

```
//更新标识号为 122 的用户证书, 更新模式为 1（证书链完整性，证书有效期，
//CRL 废除校验）
CMB_UpdateUserCertificate(“122”,0,1,CertStatus);
//更新标识号为 12 的根级证书, 更新模式为 3（证书链完整性，证书有效期，
//CRL 废除,OCSP 查询校验）
CMB_UpdateUserCertificate(“12”,1,3,CertStatus);

//NOTICE:
// CertCheckMode, 证书更新模式,常值如下：
0：证书链完整性和证书有效期校验
1：证书链完整性，证书有效期和 CRL 校验
2：证书链完整性，证书有效期和 OCSP 校验
3：证书链完整性，证书有效期，CRL 和 OCSP 校验
//返回的 CertStatus 字符串,有效字段为前六个字符。分别是：
CertStatus[0]: 该 CertStatus 字段的有效性。 ‘0’, 有效，’1’无效
CertStatus[1]: 证书链完整性状态, ‘0’, 有效，’1’无效
CertStatus[2]: 证书有效期状态, ‘0’, 有效，’1’无效
CertStatus[3]: CRL 废除状态, ‘0’, 有效，’1’无效
CertStatus[4]: OCSP 查询状态, ‘0’, 有效，’1’无效
CertStatus[5]: 私钥存在状态, ‘0’, 有效，’1’无效
```

为取得当前证书的有效状态值，需接合 更新模式(CertCheckMode),从 CertStatus 字符串中读取相应的状态值。

9. int CMB_ChangePasswd(char *CertId, char *oldpassword,
char *newpassword);

对证书管理器中的私钥进行密码修改。

五 加密、解密、签名、验证

进行加密、签名验证前，必须将装入对方的证书；

进行解密和签名前，可以通过 CMB_LoadUserPrivateKey()函数将私钥装入内存。

```
int CMB_LoadUserPrivateKey(char* CertId,
                           char* privatekeypassword);
```

解密和签名操作完毕，可以调用 CMB_ClearUserPrivateKey()清空内存私钥。

```
int CMB_ClearUserPrivateKey();
```

注意：

证书常用加密、解密、签名、验证 API 接口同证书管理器 Ver1.0, 现对证书管理器 Ver2.0 新增功能作一说明。

1.
```
int CMB_VerifyCertByCRL(unsigned char *DerCert,
                        unsigned short DerCertLen,
                        int isRoot);
```

   证书 CRL 教验。

   返回 0 证书 CRL 教验通过.其它不通过。

2.
```
int CMB_ VerifyCertByOCSP (unsigned char *DerCert,
                           unsigned short DerCertLen,
                           unsigned char *ParentDerCert,
                           unsigned short ParentDerCertLen,
                           int isRoot)
```

   证书 OCSP 教验。

   返回 0 证书 ocsp 教验通过.其它不通过。