```
            // Put the cert in the store!
            if (!CertAddCertificateContextToStore(hCertStore,   pCertContext,
              CERT_STORE_ADD_REPLACE_EXISTING, //or CERT_STORE_ADD_NEW
              NULL)) {
               lResult = GetLastError();
               __leave;
            }
        }
        __finally {
            // Don't forget to free resources, if allocated.
            if (pCertContext != NULL) {
                CertFreeCertificateContext(pCertContext);
            }
            if (hCertStore != NULL) {
                CertCloseStore(hCertStore, CERT_CLOSE_STORE_FORCE_FLAG);
            }
        }
        return lResult;
    }
```

12.5.3　使用私钥

从上节可知，在 Windows 环境下，智能卡作为密码设备，可以通过 CryptoAPI 或 CNG 接口操作。

12.1.3 节和 12.4.3 节都给出了使用私钥的例子，它们对智能卡也是适用的，可参考这些使用私钥的例子。

12.6　国密接口

12.6.1　国密接口简介

《智能 IC 卡及智能密码钥匙密码应用接口规范》简称为国密接口规范。此规范规定了基于 PKI 密码体制的智能 IC 卡及智能密码钥匙密码应用接口，描述了密码应用接口的函数、数据类型、参数的定义和设备的安全要求。

与 CryptoAPI、CNG 等相比，国密接口既支持 SM2 密码算法，又支持 RSA 算法，SM2 算法是国家密码局颁发的非对称密码算法。

智能 IC 卡及智能密码钥匙密码应用接口位于智能 IC 卡及智能密码钥匙应用程序与设备之间，如图 12-11 所示。

一个设备中存在设备认证密钥和多个应用，应用之间相互独立。设备的逻辑结构如图 12-12 所示。

应用由管理员 PIN、用户 PIN、文件和容器组成，可以存在多个文件和多个容器。每个应用维护各自的与管理员 PIN 和用户 PIN 相关的权限状态。应用的逻辑结构如图 12-13 所示。

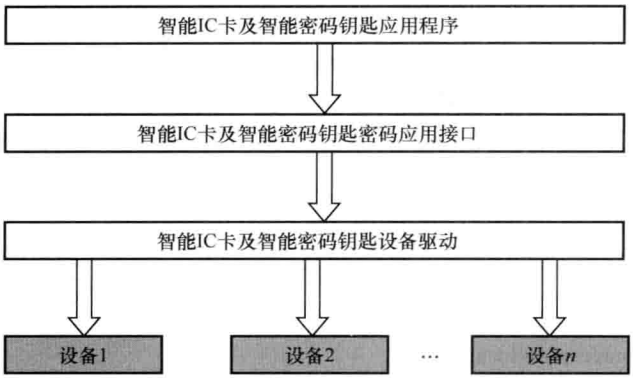

图 12-11　接口在应用层次关系中的位置

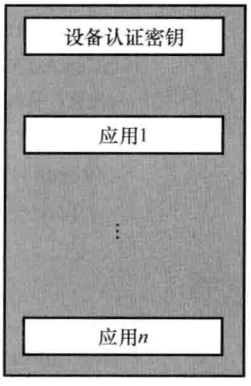

图 12-12　设备逻辑结构

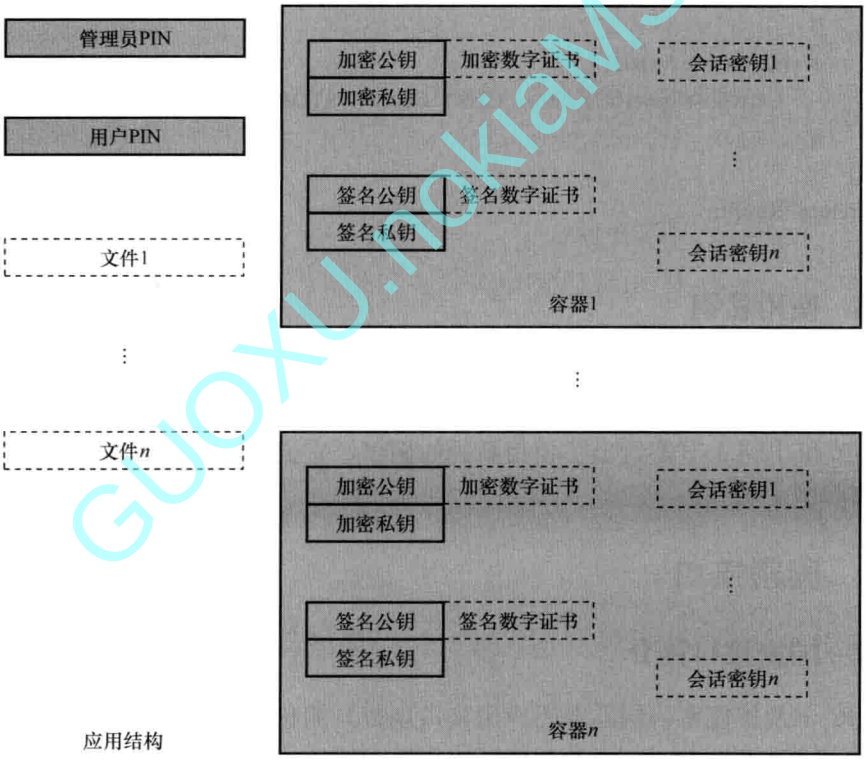

图 12-13　应用逻辑结构图

国密接口提供的接口函数分为“设备管理、访问控制、应用管理、文件管理、容器管理、密码服务”6 个部分。

设备管理函数主要完成设备的插拔事件处理、枚举设备、连接设备、断开连接、获取设备状态、设置设备标签、获取设备信息、锁定设备、解锁设备和设备命令传输等操作。

访问控制函数主要完成设备认证、PIN 码管理和安全状态管理等操作。

应用管理函数主要完成应用的创建、枚举、删除、打开、关闭等操作。

文件管理函数用以满足用户扩展开发的需要，包括创建文件、删除文件、枚举文件、获取文件信息、文件读写等操作。

容器管理函数包括创建、删除、枚举、打开和关闭容器的操作。

密码服务函数提供对称算法运算、非对称算法运算、密码哈希运算、密钥管理、消息鉴别码计算等功能。

12.6.2　使用证书

12.6.2.1　函数说明

1．连接设备

函数原型	ULONG DEVAPI SKF_ConnectDev (LPSTR szName, DEVHANDLE *phDev)	
功能描述	通过设备名称连接设备，返回设备的句柄	
参数	szName	[IN] 设备名称
	phDev	[OUT] 返回设备操作句柄
返回值	SAR_OK：　成功 其他：　错误码	

2．获取设备信息

函数原型	ULONG DEVAPI SKF_GetDevInfo (DEVHANDLE hDev, DEVINFO *pDevInfo)	
功能描述	获取设备的一些特征信息，包括设备标签、厂商信息、支持的算法等	
参数	hDev	[IN] 连接设备时返回的设备句柄
	pDevInfo	[OUT] 返回设备信息
返回值	SAR_OK：　成功 其他：　错误码	

3．打开应用

函数原型	ULONG DEVAPI SKF_OpenApplication(DEVHANDLE hDev, LPSTR szAppName, HAPPLICATION *phApplication)	
功能描述	打开指定的应用	
参数	hDev	[IN] 连接设备时返回的设备句柄
	szAppName	[IN] 应用名称
	phApplication	[OUT] 应用的句柄
返回值	SAR_OK：　成功 其他：　错误码	

4．打开容器

函数原型	ULONG DEVAPI SKF_OpenContainer(HAPPLICATION hApplication,LPSTR szContainerName, HCONTAINER *phContainer)	
功能描述	获取容器句柄	
参数	hApplication	[IN] 应用句柄
	szContainerName	[IN] 容器的名称
	phContainer	[OUT] 返回所打开容器的句柄
返回值	SAR_OK：　成功 其他：　错误码	

5. 导出数字证书

函数原型	ULONG DEVAPI SKF_ExportCertificate(HCONTAINER hContainer, BOOL bSignFlag, BYTE* pbCert, ULONG *pulCertLen)	
功能描述	从容器内导出数字证书	
参数	hContainer	[IN] 容器句柄
	bSignFlag	[IN] TRUE 表示签名证书，FALSE 表示加密证书
	pbCert	[OUT] 指向证书内容缓冲区，如果此参数为 NULL，pulCertLen 表示返回数据所需要缓冲区的长度；如果此参数不为 NULL，返回数字证书内容
	pulCertLen	[IN/OUT] 输入时表示 pbCert 缓冲区的长度，输出时表示证书内容的长度
返回值	SAR_OK：成功 其他：错误码	

12.6.2.2 示例程序

应用程序在获取证书时，必须经过如下步骤：①连接设备；②获取设备信息（可选）；③打开应用程序句柄；④打开容器句柄；⑤导出证书。

导出容器中证书示例代码如下：

```
ULONG       rv = 0;
DEVHANDLE hDev = NULL;
DEVINFO     DevInfo;
HAPPLICATION hApplication = NULL;
LPSTR       szAppName="My-Application";
LPSTR       containerName = "test";
HANDLE      hCon = NULL;
BYTE        bufCert[4096];
ULONG       ulCertLen = sizeof(bufCert);
do {
    // 打开设备句柄
    rv = SKF_ConnectDev(“xxx 设备名称”, &hDev);
    if (rv) {
        printf("call SKF_ConnectDev error.\n");
        break;
    }
    // 获得设备的一些信息，可以进行展示
    rv = SKF_GetDevInfo(hDev, &DevInfo);
    if (rv) {
        printf("call SKF_GetDevInfo error.\n");
        break;
    }
    // 根据名称打开应用句柄，应用名称为 szAppName 中存储的“My-Application”
```

```
        rv = SKF_OpenApplication(hDev, szAppName, &hApplication);
        if (rv) {
            printf("call SKF_OpenApplication error.\n");
            break;
        }
        // 打开应用中容器，容器名称为 test
        rv = SKF_OpenContainer(hApplication, “test”, &hCon);
        if (rv) {
            printf("call SKF_OpenContainer error\n");
            break;
        }
        // 导出容器中的签名证书，证书存储空间要预先分配
        rv = SKF_ExportCertificate(hCon, TRUE, bufCert, &ulCertLen);
        if (rv) {
            printf("call SKF_ExportCertificate error\n");
            break;
        }
    } while(0);
    // 关闭打开句柄，以防止资源泄露
    if (hCon) {      SKF_CloseContainer(hCon); }
    if (hApplication) {      rv = SKF_CloseApplication(hApplication);  }
    if (hDev) {      SKF_DisconnectDev(hDev);  }
```

由于国密接口中没提供证书操作的接口函数，需要解析证书、获取证书中信息、验证证书有效性等功能，需要其他开发包，例如使用 CryptoAPI、OpenSSL 等。

12.6.3　使用私钥

12.6.3.1　函数说明

1. 校验 PIN

函数原型	ULONG DEVAPI SKF_VerifyPIN (HAPPLICATION hApplication, ULONG ulPINType, LPSTR szPIN, ULONG *pulRetryCount)	
功能描述	校验 PIN 码。校验成功后，会获得相应的权限，如果 PIN 码错误，会返回 PIN 码的重试次数，当重试次数为 0 时表示 PIN 码已经锁死	
参数	hApplication	[IN] 应用句柄
	ulPINType	[IN] PIN 类型
	szPIN	[IN] PIN 值
	pulRetryCount	[OUT] 出错后返回的重试次数
返回值	SAR_OK：　成功 其他：　错误码	