摘要：本文通过对CSP结构的剖析，详细说明了在WINDOWS操作系统下建立特定的CSP所需要的步骤和开发方法，对CSP开发者有很好的帮助。

关键词：CSP　CRYPTOAPI　CYRPTOSPI

The Method to Create A CSP Program

加密服务程序CSP的建立方法

姚世军　（郑州解放军信息工程大学测绘学院　450052）

1 CSP简介

CSP（cryptographic service provider）是微软提供的加密应用程序接口CryptoAPI所需的独立软件模块，它完成加密算法的具体实现。CSP是由实现CryptoSPI（系统程序接口）中的函数的动态链接库组成。大多数CSP包括了所有函数的实现；也有些CSP主要实现在由WINDOWS服务控制管理程序管理的基于WINDOWS服务程序的功能；也有用硬件实现的，如智能卡和安全协处理器。

本文定义了CSP接口，描述了CSP编制者必须要采用的步骤和要建立自己的CSP必须满足的需求，并通过一个CSP模板例子介绍了CSP程序的结构。

2 CSP的结构

虽然每个CSP的结构不完全相同，但下面的内容适用于所有CSP。应用程序并不直接与CSP通信，而是调用操作系统动态连接库（Advapi32.dll和Crypt32.dl）中的CryptoAPI函数；操作系统分析这些函数调用并通过CryptoSPI将这些调用传递给相应的CSP，其关系如图1。

CSP的编程者必须知道操作系统传给CSP函数的参数顺序、特性和意义，并返回操作系统所希望的值。

应用程序通过句柄来引用CSP中的数据对象。被句柄引用的对象包括密钥容器、散列对象、会话密钥对和公钥密钥对等。应用程序用来访问数据对象的句柄与CSP中使用的句柄是不同的。CSP要实现所有与CryptoAPI中函数相对应的CryptoSPI函数，所有这些函数都必须用关键字WINAPI说明。

完全用软件实现的CSP以加密方式把密钥对存储在注册表中。硬件CSP将密钥对存储在永久的硬件中。密钥对以逻辑数据对象(密钥容器)存储。CSP为每个使用该CSP的客户维护一个密钥容器。每个密钥容器可以存储CSP支持的每种类型的密钥对。任何时间可以有多个密钥容器是打开的。对CryptoSPI函数的调用是通过参数来指定所用的密钥容器。

3 编写CSP

在编写CSP之前首先选择加密算法和相应的数据格式及它们的实现方法，然后才能建立CSP。建立的过程如下：

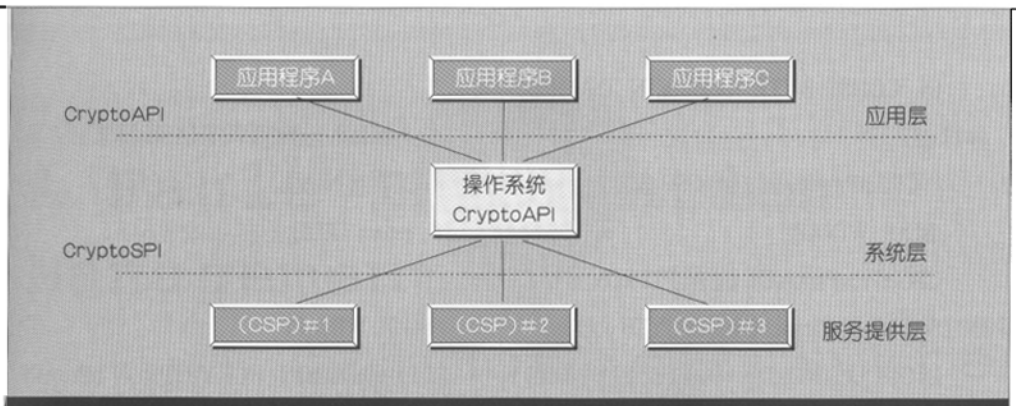

图1

3.1 建立CSP动态链接库

可以像建立任何其他DLL一样建立CSP的DLL。对于硬件CSP，DLL中包括写智能卡的设备驱动程序或能在卡上运行的嵌入式代码。

每个自定义CSP 的DLL必须包括所有CryptoSPI函数。如果CSP不支持特定的函数，对该函数的调用会返回E_NOTIMPL错误代码。自定义的CSP可以有多个DLL。

3.2 写CSP注册程序以建立适当的注册项

建立CSP要提供安装程序，CSP的安装程序至少有下列内容：将CSP DLL拷贝到PATH环境变量指定的目录；建立适当的注册项等。

3.2.1 注册CSP

用下列注册键注册CSP，

HKEY_LOCAL_MACHINE>SOFTWARE>Microsoft>Cryptography>Defaults>Provider

在Provider键下有每个CSP的名字，其结构类似如下：

<CSP name>

Image Path:REG_SZ:<CSP DLL name>

Signature:REG_BINARY:<digital signature>

Type:REG_DWORD:<CSP type>

<CSP name>是CSP的名字，<CSP DLL name>是CSP动态链接库的名字，<digital signature>是CSP DLL的数字签名文件，它可以是由SIGN.EXE生成的测试签字，也可以是从微软发布的得到，<CSP type>是十进制的三位数，例如：如果CSP类型是25，密钥的类型是025。

3.2.2 设置缺省的CSP

为每个类型的CSP指定缺省CSP。当程序调用CryptAcquireContext只提供CSP类型，将使用缺省的CSP。如果必须用自定义的CSP，设置程序可以将它的CSP安装成机器缺省CSP。

3.2.3 定义用户缺省CSP

可以为每个CSP类型指定一个用户缺省CSP。应用程序调用CRYPTOAPI中的函数CryptAcquireContext时只提供CSP类型时，就使用用户缺省CSP。

3.3 对CSP签名

每次建立CSP DLL时，必须进行签名。这包括为测试而建立的每个CSP模块。在Windows2000以前，签名是放在注册表中。Windows2000以后引进了在CSP DLL的资源表中的数字签名。为了支持同一个CSP二进制在各种Windows上运行，CSP用注册表和资源表两种方式签名。

为了支持新的签名格式，CSP必须提供144字节的空间，签名将放在该位置。签名应用程序(Sign.exe)对CSP DLL进行签名。给定DLL文件，该程序将产生一个签名文件，它的内容将在建立缺省CSP时放在注册表中。Sign.exe有三个参数：Sign {slv} <文件名> <签名文件>

如果要生成签名文件，第一个参数必须是S；如果签名文件已经存在，只是对DLL文件进行验证时第一个参数用V。第二个参数必须是DLL文件的绝对路径名。第三个参数必须是签名文件的绝对路径名。

3.4 测试CSP

为了与商业出口限制一致，Microsoft公司必须数字签名CSP，以便它能运行在微软的操作系统上。用微软的CSPDK来帮助进行CSP的测试。CSP测试包括下列步骤：

(1) 用SIGN.EXE程序对CSP进行签名，以产生调试文件.SIG。

(2) 用为CSP写的安装程序安装CSP。

(3) 运行调用CryptoAPI的测试代码，以测试CSP对这些函数的实现。

4 CSP程序结构框架

如上所述，CSP程序是一个实现加密算法的动态连接库，这个DLL中通常要实现CryptoSPI中的所有或部分函数。现在有许多CSP提供商，微软也提供了几个基本的CSP，CSP程序的结构框架如下所示。

```
#ifndef WIN32_LEAN_AND_MEAN
#define WIN32_LEAN_AND_MEAN
#endif
#undef UNICODE
#include <windows.h>
#include <wincrypt.h>
#include <cspdk.h>
HINSTANCE g_hModule = NULL;
BOOL WINAPI
DllMain(
 HINSTANCE hinstDLL, // DLL模块句柄
 DWORD fdwReason,
 LPVOID lpvReserved)
{ if (fdwReason == DLL_PROCESS_ATTACH)
  { DisableThreadLibraryCalls(hinstDLL);
    g_hModule = hinstDLL;
  }
  return TRUE;
}
/* CPAcquireContext函数定义--获得CSP的环境句柄 */
BOOL WINAPI
CPAcquireContext(OUT HCRYPTPROV *phProv,IN LPCSTR szContainer,
  IN DWORD dwFlags,IN PVTableProvStruc pVTable)
{
  *phProv = (HCRYPTPROV)NULL; // 用自定义结构替换NULL
  return TRUE;
}
/* CPReleaseContext -- 释放句柄 */
BOOL WINAPI
CPReleaseContext(IN HCRYPTPROV hProv, IN DWORD dwFlags)
```

```
{   return TRUE;
}
/* CPGenKey 生成加密密钥 */
BOOL WINAPI
CPGenKey(IN  HCRYPTPROV hProv,IN ALG_ID Algid,
    IN DWORD dwFlags, OUT HCRYPTKEY *phKey)
{  *phKey = (HCRYPTKEY)NULL;  // 用自定义结构替换 NULL
    return TRUE;
}
/* CPDeriveKey 从基数据中生成密钥 */
BOOL WINAPI
CPDeriveKey(IN  HCRYPTPROV hProv,IN ALG_ID Algid,
    IN  HCRYPTHASH hHash, IN  DWORD dwFlags, OUT HCRYPTKEY *phKey)
{  *phKey = (HCRYPTKEY)NULL;  // 用自定义结构替换 NULL
    return TRUE;
}
/* CPDestroyKey  删除密钥 */
BOOL WINAPI
CPDestroyKey(IN HCRYPTPROV hProv,IN HCRYPTKEY hKey)
/* CPSetKeyParam 设置密钥参数 */
BOOL WINAPI
CPSetKeyParam(IN  HCRYPTPROV hProv, IN  HCRYPTKEY hKey,
    IN  DWORD dwParam,IN  CONST BYTE *pbData, IN  DWORD dwFlags)
/* CPGetKeyParam 得到密钥参数 */
BOOL WINAPI
CPGetKeyParam(IN  HCRYPTPROV hProv, IN  HCRYPTKEY hKey,
    IN  DWORD dwParam, OUT LPBYTE pbData,IN OUT LPDWORD pcbDataLen,
    IN  DWORD dwFlags)
/* CPSetProvParam 设置 CSP 参数 */
BOOL WINAPI
CPSetProvParam(IN  HCRYPTPROV hProv, IN  DWORD dwParam,
    IN  CONST BYTE *pbData,IN  DWORD dwFlags)
/* CPGetProvParam 获得 CSP 参数 */
BOOL WINAPI
CPGetProvParam(IN  HCRYPTPROV hProv, IN  DWORD dwParam,  OUT LPBYTE pbData,
    IN OUT LPDWORD pcbDataLen, IN DWORD dwFlags)
/* CPEncrypt 加密数据
* 参数: IN  hProv  -  CSP 用户的句柄; IN  hKey  -  密钥句柄
*   IN  hHash  -  散列句柄;   IN  Final  -  数据块结束标志
*   IN  dwFlags  -  标志值;   IN OUT pbData  -  加密的数据
*   IN OUT pdwDataLen  -  加密数据长度的指针; IN dwBufLen  -  数据缓冲区大小 *
/ BOOL WINAPI
CPEncrypt(IN  HCRYPTPROV hProv,IN HCRYPTKEY hKey,
    IN HCRYPTHASH hHash, IN BOOL fFinal, IN  DWORD dwFlags,
        IN OUT LPBYTE pbData,IN OUT LPDWORD pcbDataLen,IN  DWORD cbBufLen)
{
  /* 具体加密算法的实现 */
}
/* CPDecrypt 数据解密  参数同加密函数 */
BOOL WINAPI
CPDecrypt(IN  HCRYPTPROV hProv,IN HCRYPTKEY hKey,IN  HCRYPTHASH hHash,
    IN BOOL fFinal,IN  DWORD dwFlags,IN OUT LPBYTE pbData,
    IN OUT LPDWORD pcbDataLen)
{
  /* 解密算法的实现 */
}
/* */
BOOL WINAPI
CPCreateHash(IN  HCRYPTPROV hProv, IN ALG_ID Algid,IN  HCRYPTKEY hKey,
    IN DWORD dwFlags,OUT HCRYPTHASH *phHash)
{
  *phHash = (HCRYPTHASH)NULL;  // 用散列结来替换 NULL.
  return TRUE;
}
BOOL WINAPI
CPHashData(IN  HCRYPTPROV hProv,IN HCRYPTHASH hHash,
    IN  CONST BYTE *pbData,IN  DWORD cbDataLen,IN  DWORD dwFlags)
{
  /* 散列算法的具体实现 */
}
```

下面各函数也是CSP中应实现的，因篇幅所限，仅列出它们的名称及功能，读者可根据其功能编制相应的程序：CPHashData(数据进行散列)、CPCreateHash(建立散列算法)、CPImportKey(倒入密钥到CSP中)、CPExportKey(从CSP倒出密钥)、CPGetHashParam (获得散列算法参数)、CPSetHashParam(置散列算法参数)、CPHashSessionKey(散列密钥)、CPSignHash (散列建立数字签名)、CPDestroyHash(删除散列对象)、CPVerifySignature(验证数字签名)、CPGenRandom (填充缓冲区)、CPGetUserKey(获得永久用户密钥)、CPDuplicateHash(复制散列句柄)、CPDuplicateKey (复制密钥)。

加密服务程序CSP是通过微软CryptoAPI进行加密的关键，它支持对称密钥、公开密钥、散列算法等多种加密算法。编写一个通用灵活的CSP，将会为电子商务的安全或数据传输提供重要保证。

参 考 文 献

1 Ben Forla等著，杜大鹏等译，《WINDOW2000开发人员指南》，水利水电出版社，2001年12月。
2 Ian Mclean 著，吴世忠等译，《WINDOWS 2000安全技术》，机械工业出版社，2001年6月。
3 Microsoft CPSDK