四川省社会保障卡信息管理系统
多合一高拍接口规范

文档编号：	生效日期：	受控编号：
密级：秘密	版次：V1.0	修改状态：可编辑
编制： 社保卡系统建设项目组	审核：	批准：

四川久远银海软件股份有限公司

2018 年 1 月 29 日

编号	修订日期	版本号	修订描述	说明
01	2018 年 1 月 29 日	1.0	C	

说明：[C]-创建；[M]-修改；[A]-增加；[D]-删除；

目录

1 引言

本规范是《四川省社会保障卡信息管理系统多合一高拍接口规范》关于调用多合一规范的解释信息。

2 适用范围

本规范适用于四川省社会保障卡信息管理系统。其使用对象主要是各级人力资源和社会保障部门以及社会保障卡应用相关的卡片应用系统的开发、集成和维护等机构。

3 参考标准

《社会保障卡(个人)卡规范》（LDB0021-2000）

社会保障卡文件结构和数据项（V2.0）

社会保障卡医保结算流程(人社信息函[2012]38 号)

社保 PSAM 应用指南（V2.0）

人社信息函[2016]38 号-关于印发社会保障卡读写终端接口规范的通知

4 定义

以下定义适用于本规范。

4.1 应用城市代码

使用地行政区划代码遵守 GB/T2260 的规定。

5 缩略语和符号表示

以下缩略语和符号表示适用于本规范。

FLAG　　传递接口命令

Data　　传递送往接口程序的获取或修改的信息

ErrMsg　错误信息

6 接口描述

6.1　接口说明

说明：

1、对于应用系统只需要调用应用厂商提供的 JS 函数类配合厂家提供的驱动，就可以访问设备，设备接口具有统一的规范。

2、厂家提供 JS 函数中，不能使用 OCX 控件方式访问，由于需要兼容多浏览器版本，只能通过 JS 函数调用。

3、要求不同厂商提供的 JS 函数类（.jS）符合本规范，对于不同终端厂商提供的 JS 库，更换设备时只需将厂商提供 JS 库重写配置并安装厂家驱动。

6.2　接口 JS 函数封装说明

该 JS 为标准的 javascript 代码，要适合于 IE8 以上的 IE 浏览器，chrome，firefox 等主流浏览器。

6.3　JS 函数文件名

接 口 要 求	满足条件
基 于 规 范	《社会保障卡读写终端规范》（LD/T 33—2015）
版　　本	JS 函数类
命　　名	厂家代码+GPYHS.js 如：银海（YH），厂家代码加 gpyhs.js
支 持 系 统	WindowsXP 及以上的 32 位和 64 位操作系统
认 证 类 型	基于 PSAM 卡/加密机均支持

操 作 流 程	每个/组函数中应包含预处理流程。在函数开始时进行打开设备、卡上电操作，卡上电时应先判断是否插入接触卡，若是则后续命令向接触卡发送，否则后续命令向非接触卡发送。在函数正常结束或出现异常时应进行卡下电、关闭设备操作。

6.4 实现的功能

高拍仪部分：

主，副 摄像头的拍照、旋转、PDF 的合并、上传、二维码识别、连拍等。

读卡器部分：

实现全国社会保障一卡通，现制定社保卡跨省应用技术方案。本方案主要规范社保卡预处理、鉴权、外部认证、医保结算等操作的基本环节，统一各地社保卡读写终端接口、用卡流程；规范用卡业务系统接入持卡库的接口，通过部持卡库提供跨省用卡时的鉴权和认证服务；针对不同的跨省用卡场景（如异地就医结算用卡、其他异地业务用卡、异地卡服务等），规范通过部持卡库连接部级加密机进行用卡认证的流程。

身份证部分：

实现身份证的信息读取。

7 高拍仪接口定义

7.1 打开主摄像头

函数名称	SS_CMR_OpenAB
函数功能	打开主摄像头
函数原型	SS_CMR_OpenAB(int iCamNo,int mainX,int mainY,int mwidth,int mheight, int sideX,int sideY,int widthside,int heightside);

参数说明	参数	参数含义
	[IN]iCamNo	摄像头号: 0: 桌面摄像头 1: 柜员摄像头 2: 客户摄像头
	[IN] mainX	主摄像头 X 坐标
	[IN] mainY	主摄像头 Y 坐标
	[IN] mwidth	主摄像头宽度
	[IN] mheight	主摄像头高度
	[IN] sideX	副摄像头 X 坐标
	[IN] sideY	副摄像头 Y 坐标
	[IN] widthside	副摄像头宽度
	[IN] heightside	副摄像头高度
详细说明	1、在打开主摄像头的同时，将主、副摄像头画面的位置和大小也同时设置好； 2、如果不启用副摄像头，将 sideX、sideY、widthside、heightside 这四个参数设置为 0。	

7.2 关闭主摄像头

函数名称	SS_CMR_CloseAB	
函数功能	关闭主摄像头	
函数原型	SS_CMR_CloseAB(int iCamNo)	
参数说明	参数	参数含义
	[IN]iCamNo	摄像头号: 0: 桌面摄像头

		1: 柜员摄像头 2: 客户摄像头
详细说明	1、必须在打开摄像头的前提下，才能调用这个函数进行关闭； 2、在关闭主摄像头的时候，副摄像头也随之关闭。	

7.3 打开副摄像头

函数名称	SS_CMR_OpenSideAB	
函数功能	用于打开副摄像头号	
函数原型	SS_CMR_OpenSideAB(int iCamNoSide);	
参数说明	参数	参数含义
	[IN]iCamNoSide	摄像头号: 0: 桌面摄像头 1: 柜员摄像头 2: 客户摄像头
详细说明	1、不能和主摄像头同时选择同一个摄像头号； 2、副摄像头必须在主摄像头打开的前提下使用，不能单独是用副摄像头；	

7.4 关闭副摄像头

函数名称	SS_CMR_CloseSideAB	
函数功能	关闭副摄像头	
函数原型	SS_CMR_CloseSideAB(int iCamNoSide);	
参数说明	参数	参数含义
	[IN]iCamNoSide	摄像头号: 0: 桌面摄像头 1: 柜员摄像头 2: 客户摄像头

详细说明	1、不能和主摄像头同时选择同一个摄像头号； 2、副摄像头必须在主摄像头打开的前提下使用，不能单独是用副摄像头；

7.5 获取摄像头的状态

函数名称	SS_CMR_GetStatus	
函数功能	用于获取当前摄像头的状态	
函数原型	SS_CMR_GetStatusAB(int iCamNo, char *status);	
参数说明	参数	参数含义
	[IN]iCamNo	摄像头号: 0: 桌面摄像头 1: 柜员摄像头 2: 客户摄像头
	[OUT] status	设备状态值 1：设备处于打开状态 2：设备正在忙 3：设备处于硬件故障状态 4：设备处于关闭状态 9999：其他状态

7.6 拍照保存

函数名称	SS_CMR_PhotoAB	
函数功能	拍照并保存	
函数原型	SS_CMR_PhotoAB(int iCamNo,char *fileAddr);	
参数说明	参数	参数含义
	[IN]iCamNo	摄像头号: 0: 桌面摄像头 1: 柜员摄像头 2: 客户摄像头
	[IN]fileAddr	要保存到的图片保存路径，后缀支持：jpg、bmp、png
详细说明	fileAddr 必须是文件路径，包括路径和文件名。	

7.7 放大

函数名称	SS_CMR_ZoomInAB	
函数功能	放大	
函数原型	SS_CMR_ZoomInAB(int iCamNo);	
参数说明	参数	参数含义
	[IN]iCamNo	摄像头号: 0: 桌面摄像头 1: 柜员摄像头 2: 客户摄像头
详细说明	必须在打开摄像头的前提下，才能使用。	

8.8 缩小

函数名称	SS_CMR_ZoomOutAB	
函数功能	缩小	
函数原型	SS_CMR_ZoomOutAB(int iCamNo)	
参数说明	参数	参数含义
	[IN]iCamNo	摄像头号: 0: 桌面摄像头 1: 柜员摄像头 2: 客户摄像头
详细说明	必须在打开摄像头的前提下，才能使用。	

7.9 左旋

函数名称	SS_CMR_RotateLeftAB
函数功能	左旋

函数原型	SS_CMR_RotateLeftAB(int iCamNo)	
参数说明	参数	参数含义
	[IN] iCamNo	摄像头号: 0：桌面摄像头 1：柜员摄像头 2：客户摄像头
详细说明	必须在打开摄像头的前提下，才能使用。	

7.10 右旋

函数名称	SS_CMR_RotateRight	
函数功能	右旋	
函数原型	SS_CMR_RotateRight(int iCamNo);	
参数说明	参数	参数含义
	[IN]iCamNo	摄像头号: 0：桌面摄像头 1：柜员摄像头 2：客户摄像头
详细说明	必须在打开摄像头的前提下，才能使用。	

7.11 开始录像

函数名称	SS_CMR_VideoStartAB	
函数功能	开始录像	
函数原型	SS_CMR_VideoStartAB(int iCamNo,char *fileAddr);	
参数说明	参数	参数含义
	[IN]iCamNo	摄像头号: 0：桌面摄像头 1：柜员摄像头

		2: 客户摄像头
	[IN]fileAddr	要保存到的录像保存路径
详细说明	1、必须在打开摄像头的前提下，才能使用。	

3.12 停止录像

函数名称	SS_CMR_VideoStopAB	
函数功能	开始录像	
函数原型	SS_CMR_VideoStartAB(int iCamNo,char *fileAddr);	
参数说明	参数	参数含义
	[IN]iCamNo	摄像头号: 0: 桌面摄像头 1: 柜员摄像头 2: 客户摄像头
	[IN]fileAddr	要保存到的录像保存路径
详细说明	1、此接口必须在已调用了开始录像接口后使用	

7.13 开始定时抓拍

函数名称	SS_CMR_StartTimerCapAB	
函数功能	开始定时抓拍	
函数原型	SS_CMR_StartTimerCapAB(int iCamNo, int nTime, char *pImagePath);	
参数说明	参数	参数含义
	[IN]iCamNo	摄像头号:

		0: 桌面摄像头 1: 柜员摄像头 2: 客户摄像头
	[IN]nTime	定时抓拍的时间间隔，以秒为单位。
	[IN]pImagePath	照片保存路径
详细说明	必须在打开摄像头的前提下，才能使用；	

7.14 关闭定时抓拍

函数名称	SS_CMR_StopTimerCapAB	
函数功能	关闭定时抓拍	
函数原型	int SS_CMR_StopTimerCapAB(int iCamNo);	
参数说明	参数	参数含义
	[IN] int iCamNo	摄像头号: 0: 桌面摄像头 1: 柜员摄像头 2: 客户摄像头
详细说明	1、必须在打开摄像头的前提下，才能使用。 2、必须在开启定时抓拍的前提下，才能使用。	

7.15 设置视频属性

函数名称	SS_CMR_ShowVideoPropAB	
函数功能	设置视频属性	
函数原型	SS_CMR_ShowVideoPropAB(int iCamNo);	
参数说明	参数	参数含义
	[IN]iCamNo	摄像头号: 0: 桌面摄像头

		1: 柜员摄像头 2: 客户摄像头
详细说明	必须在打开摄像头的前提下，才能使用。	

7.16 设置图片属性

函数名称	SS_CMR_ParaSetAB	
函数功能	设置视频属性	
函数原型	SS_CMR_ShowVideoPropAB(int iCamNo);	
参数说明	参数	参数含义
	[IN]iCamNo	摄像头号: 0: 桌面摄像头 1: 柜员摄像头 2: 客户摄像头
详细说明	必须在打开摄像头的前提下，才能使用。	

7.17 设置剪裁模式

函数名称	SS_CMR_SetCutModeAB	
函数功能	设置剪裁模式	
函数原型	SS_CMR_SetCutModeAB(int iCamNo,int CutMode);	
参数说明	参数	参数含义
	[IN]iCamNo	摄像头号: 0: 桌面摄像头 1: 柜员摄像头 2: 客户摄像头
	[IN] CutMode	裁剪模式 0：不裁切，默认 1：画框裁切

		2：固定矩形框 3：自动裁切
详细说明	1、必须在打开摄像头的前提下，才能使用； 2、此处默认选择 0 模式下。	

7.18 添加水印

函数名称	SS_CMR_WaterMarkAddAB	
函数功能	添加水印	
函数原型	int SS_CMR_WaterMarkAddAB(int iCamNo)	
参数说明	参数	参数含义
	[IN] iCamNo	摄像头编号: 0: 桌面摄像头 1: 柜员摄像头 2: 客户摄像头
详细说明	1、必须在打开摄像头的前提下，才能使用；	

7.19 设置水印

函数名称	SS_CMR_WaterMarkSetAB	
函数功能	设置水印（取消水印）	
函数原型	int SS_CMR_WaterMarkSet(int iCamNo);	
参数说明	参数	参数含义
	[IN] iCamNo	摄像头编号: 0: 桌面摄像头 1: 柜员摄像头 2: 客户摄像头

特殊说明	1、必须在打开摄像头的前提下，才能使用；

7.20 返回图片 base64

函数名称	SS_Base64StringAB	
函数功能	返回图片的 base64	
函数原型	int SS_Base64StringAB(int iCamNo);	
参数说明	参数	参数含义
	[IN] iCamNo	摄像头编号: 0: 桌面摄像头 1: 柜员摄像头 2: 客户摄像头
	[OUT] info	返回照片相片的 BASE64 编码
特殊说明	1、必须在打开摄像头的前提下，才能使用；	

7.21 条码识别

函数名称	SS_CMR_ReadBarcodeAB	
函数功能	条码识别	
函数原型	SS_CMR_ReadBarcodeAB(int iCamNo, char * info);	
参数说明	参数	参数含义
	[IN]iCamNo	摄像头号: 0: 桌面摄像头 1: 柜员摄像头 2: 客户摄像头

	[OUT] info	条码信息，包含了条码的位置、类 型、条码值，此接口可识别同一画面中的多个条码，目前支持所有主流一维码、QR 二维码。从上往下，从左到右顺序返回 如： 条码 A|二维码 B|
详细说明	1、Info 既是出参又是入参。当入参时，是二维码图片的路径；当出参时，是带着条码信息。 2、必须先对证件拍照，选择拍好的图片进行识别。	

输出函数接口说明

7.22 输出信息

函数名称	showmessage	
函数功能	输出信息	
函数原型	showmessage（char *meg）;	
参数说明	参数	参数含义
	[OUT] meg[]	1、msg[0] 函数名 例：“iReadSIEF05； 2、msg[1] 函数的出参(用“|”分割) 例 ：“123456|3|2.00|987654321|20170101|20171230|123456789|”； 3、msg[2] 函数返回值，成功返回 0， 失败返回非 0
特殊说明	用来展示函数调用的结果。 开发厂商根据定义的 JS 封装好。	

社保卡接口定义

根据社会保障卡（以下简称社保卡）全国通用的要求，在《社会保障卡读写终端规范》（LD/T 33—2015）的基础上，进一步统一社保

卡读写终端与业务系统及其客户端的接口，规定了“读基本信息”、“通用读卡”、“通用写卡”、“PIN 校验”、“PIN 修改”、“PIN 重置”、“PIN 解锁”、“消费交易”、“读消费交易记录”等 9 个接口的函数定义、参数及处理流程，包括 9 个基于 PSAM 卡认证的函数和 6 组基于社保卡持卡人员基础信息库（以下简称持卡库）连接加密机认证的函数，适用于社保卡读写终端动态库及读写终端所关联的相关业务系统接口的研发、集成及维护。

交易列表

分类	函数名	说明
基于 PSAM 卡认证	iReadCardBas	读基本信息
	iReadCard	通用读卡
	iWriteCard	通用写卡
	iVerifyPIN	PIN 校验
	iChangePIN	PIN 修改
	iReloadPIN	PIN 重置
	iUnblockPIN	PIN 解锁
	iDoDebit	消费交易
	iReadDebit	读消费交易记录
基于加密机认证	iReadCardBas_HSM_Step1	基于加密机的读基本信息（步骤一）
	iReadCardBas_HSM_Step2	基于加密机的读基本信息（步骤二）
	iReadCard_HSM_Step1	基于加密机的通用读卡（步骤一）
	iReadCard_HSM_Step2	基于加密机的通用读卡（步骤二）
	iWriteCard_HSM_Step1	基于加密机的通用写卡（步骤一）
	iWriteCard_HSM_Step2	基于加密机的通用写卡（步骤二）
	iReloadPIN_HSM_Step1	基于加密机的 PIN 重置（步骤一）
	iReloadPIN_HSM_Step2	基于加密机的 PIN 重置（步骤二）
	iReloadPIN_HSM_Step3	基于加密机的 PIN 重置（步骤三）
	iUnblockPIN_HSM_Step1	基于加密机的 PIN 解锁（步骤一）
	iUnblockPIN_HSM_Step2	基于加密机的 PIN 解锁（步骤二）
	iUnblockPIN_HSM_Step3	基于加密机的 PIN 解锁（步骤三）
	iDoDebit_HSM_Step1	基于加密机的消费交易（步骤一）
	iDoDebit_HSM_Step2	基于加密机的消费交易（步骤二）
公共接口	iReadICCardNum	读取银行卡号
	iReadIdentityCard	读取身份证号码

1 “读基本信息”函数

1.1 iReadCardBas “读基本信息”

1.1.1 函数定义

读基本信息接口函数定义见表 1。在进行任何社保卡操作前，应先调用此函数。

表 1 读基本信息接口函数定义

函数名称	**读基本信息**					
函数名	**iReadCardBas(iType)**					
原语法	long **iReadCardBas**(int iType, char* pOutInfo)					
功能描述	选择社保卡社会保障系统环境后，通过 PSAM 卡对社保卡进行内部认证，通过后将卡内的基本信息读出返回。					
参数说明	序号	参数	输入/输出	类型	长度（十进制）	含义
	1	iType	IN	整数	4	操作卡的类型
	2	pOutInfo	OUT	字符串	1024	读出数据或返回错误信息
返回值	0 表示成功；非 0 表示失败。					

1.1.2 参数说明

（1） 输入参数 iType

表示执行本函数时操作卡的类型，定义如下：1-接触式操作卡；2-非接触式操作卡；3-自动寻卡，接触式操作卡优先；4-自动寻卡，非接触式操作卡优先。

（2） 输出参数 pOutInfo

当函数执行成功时，该输出参数为读出的社保卡基本信息各数据项，依次为：发卡地区行政区划代码（卡识别码前 6 位）、社会保障号码、卡号、卡识别码、姓名、卡复位信息（仅取历史字节）、规范版本、发卡日期、卡有效期、终端机编号、终端设备号。各数据项之间以“|”分割，且最后一个数据项以“|”结尾。例如：639900|1111111981010111 10|X00000019|639900D15600000500BF7C7A48FB4966|张三|00814E43238697159900BF7C7A|1.00|20101001|20201001|410100813475|终端设备号|。

当函数执行失败时，该输出参数为错误信息描述。

注：当没有终端设备号时，终端设备号返回空字符串。

1.1.3 处理流程

（1） 判断输入参数有效性；

（2） 调用 SSSE32.DLL 动态库中的“打开设备”函数对设备进行初始化，获得设备句柄；

（3） 调用 SSSE32.DLL 动态库中的“卡上电”函数，分别对社保卡和 PSAM

卡进行上电复位，若无 PSAM 卡，返回错误代码“-2201”（具体参见附录 A）；

（4） 选择社保卡社会保障系统环境，同时根据卡片返回的应答信息确定算法环境，根据算法环境标识选择 PSAM 卡相应算法密钥文件，具体方法详见 10.1；

（5） 若 PSAM 卡算法不支持（即 PSAM 卡内没有 SSF33 算法或 SM4 算法），返回错误代码“-2202”；若为 3.0 卡（要求详见 10.2），但 PSAM 卡内没有 RK_{SSSE} 密钥，返回错误代码“-2203”（具体参见附录 A）；

（6） 执行内部认证指令，验证卡片有效性；若为 3.0 卡，再执行 RK_{SSSE} 密钥外部认证；其中密钥分散因子提取具体方法详见 10.3；

（7） 依次读取各项数据；读取“卡号”时只读前 9 位（详见 10.4）；2.0 卡或 3.0 卡需读出“姓名扩展”，与“姓名”拼接后返回完整姓名（详见 10.2）；

（8） 读取 PSAM 卡终端机编号和终端设备号；

（9） 调用 SSSE32.DLL 动态库中的“关闭设备”函数关闭设备，释放端口；

（10） 以上操作过程中出现异常时，均应先调用 SSSE32.DLL 动态库中的“关闭设备”函数关闭设备，释放端口，然后退出函数，返回错误代码和错误信息描述。

注：业务系统判断到五种情况的错误代码（详见 10.5）后，调用基于加密机的读基本信息函数继续进行读基本信息操作。

1.2 iReadCardBas_HSM_Step1 “基于加密机的读基本信息（步骤一）”

1.2.1 函数定义

基于加密机的读基本信息（步骤一）接口函数定义见表 2。

表 2 基于加密机的读基本信息（步骤一）接口函数定义

函数名称	**基于加密机的读基本信息（步骤一）**					
函数	**iReadCardBas_HSM_Step1**（int iType）					
语法	long **iReadCardBas_HSM_Step1**(int iType, char* pOutInfo)					
功能描述	选择社会保障系统环境后，返回内部认证和外部认证所需信息。					
参数说明	序号	参数	输入/输出	类型	长度（十进制）	含义
	1	iType	IN	整数	4	操作卡的类型
	2	pOutInfo	OUT	字符串	1024	返回认证信息或错误信息
返回值	0 表示成功；非 0 表示失败。					

1.2.2 参数说明

（1） 输入参数 iType

定义同 1.1.2（1）。

（2） 输出参数 pOutInfo

当函数执行成功时，该输出参数为读出的社保卡内部认证和外部认证的计算数据，依次为：发卡地区行政区划代码（卡识别码前 6 位）、卡复位信息（仅取历史字节）、算法标识、卡识别码、内部认证过程因子、内部认证鉴别所需的原始信息、外部认证过程因子、外部认证鉴别所需的原始信息、终端机具编码。其中外部认证相关数据项全部不为空或全部为空。各数据项之间以“|”分割，且最后一个数据项以“|”结尾。

当函数执行失败时，该输出参数为错误信息描述。

注：当外部认证相关数据项为空时，表示不做外部认证。

1.2.3 处理流程

（1） 判断输入参数有效性；

（2） 调用 SSSE32.DLL 动态库中的“打开设备”函数对设备进行初始化，获得设备句柄；

（3） 调用 SSSE32.DLL 动态库中的“卡上电”函数，对社保卡进行上电复位；

（4） 选择社保卡社会保障系统环境，同时根据卡片返回的应答信息确定算法环境，具体方法详见 10.1；

（5） 从卡片取 2 个随机数执行内部认证计算，获取卡片返回的内部认证鉴别数据；

（6） 判断是否为 3.0 卡，若是则从卡片取 2 个随机数用于外部认证计算，否则输出参数的最后两项（即外部认证过程因子、外部认证鉴别所需的原始信息）应全部为空；

（7） 返回内部认证和外部认证所需信息；

（8） 以上操作过程中出现异常时，均应先调用 SSSE32.DLL 动态库中的“关闭设备”函数关闭设备，释放端口，然后退出函数，返回错误代码和错误信息描述。

1.3 iReadCardBas_HSM_Step2 “基于加密机的读基本信息（步骤二）”

1.3.1 函数定义

基于加密机的读基本信息（步骤二）接口函数定义见表 3。

表 3 基于加密机的读基本信息（步骤二）接口函数定义

函数名称	基于加密机的读基本信息（步骤二）
函数	**iReadCardBas_HSM_Step2**(char *pKey)

语法	long **iReadCardBas_HSM_Step2**(char *pKey, char* pOutInfo)					
功能描述	根据加密机返回的内部认证和外部认证结果数据对社保卡进行内部认证和外部认证，通过后将卡内的基本信息读出返回。					
参数说明	序号	参数	输入/输出	类型	长度（十进制）	含义
	1	pKey	IN	字符串	128	加密机返回的内部认证和外部认证结果数据
	2	pOutInfo	OUT	字符串	1024	读出数据或返回错误信息
返回值	0 表示成功；非 0 表示失败。					

1.3.2 参数说明

（1） 输入参数 pKey

加密机返回的内部认证和外部认证结果数据，依次为：内部认证结果数据（即内部认证鉴别数据（16 位）和内部认证鉴别所需的原始信息（16 位）拼接组成）、外部认证结果数据（即外部认证鉴别数据（16 位）和外部认证鉴别所需的原始信息（16 位）拼接组成）。各数据项之间以“|”分割，且最后一个数据项以“|”结尾。

注：如果不做外部认证，则后面一个参数都为空字符串。

（2） 输出参数 pOutInfo

定义同 1.1.2（2）。

注：当没有 PSAM 卡时，终端机编号返回 12 个 0，即 6 个 0x00 对应的字符。

当没有终端设备号时，终端设备号返回空字符串。

1.3.3 处理流程

（1） 判断输入参数有效性；

（2） 比对卡片和加密机返回的内部认证鉴别数据，验证卡片有效性；

（3） 判断输入参数中是否含有外部认证鉴别数据，如果有，执行外部认证；

（4） 依次读取各项数据，详见 1.1.3（7）；

（5） 读取 PSAM 卡终端机编号和终端设备号；

（6） 调用 SSSE32.DLL 中的“关闭设备”函数关闭设备，释放端口；

（7） 以上操作过程中出现异常时，均应先调用 SSSE32.DLL 动态库中的“关闭设备”函数关闭设备，释放端口，然后退出函数，返回错误代码和错误信息描述。

2 “通用读卡”函数

2.1 iReadCard “通用读卡”

2.1.1 函数定义

通用读卡接口函数定义见表 4。

表 4 通用读卡接口函数定义

函数名称	通用读卡					
函数	**iReadCardAB(Type,KeyType,Buffdata,BuffAddr);**					
语法	long **iReadCard** (int iType, int iAuthType, char* pCardInfo, char* pFileAddr, char* pOutInfo)					
功能描述	根据所需读取的信息进行认证后读出卡内指定文件的信息。					
参数说明	序号	参数	输入/输出	类型	长度（十进制）	含义
	1	iType	IN	整数	4	操作卡的类型
	2	iAuthType	IN	整数	4	认证方式
	3	pCardInfo	IN	字符串	128	卡基本信息
	4	pFileAddr	IN	字符串	1024	文件名及数据项
	5	pOutInfo	OUT	字符串	1024*20	读出数据或返回错误信息
返回值	0 表示成功；非 0 表示失败。					

2.1.2 参数说明

（1） 输入参数 iType

定义同 1.1.2（1）。

（2） 输入参数 iAuthType

当文件的读控制受 PIN 或 RK 密钥保护时，该参数用于指定读控制认证方式，定义如下：1-PIN 校验；2-RK 密钥认证。此参数只在文件的读控制权限为“PIN 或 RK”时有效。

（3） 输入参数 pCardInfo

该参数用于传入卡的基本信息，依次为：卡识别码、卡号。各数据项之间以“|”分割，且最后一个数据项以“|”结尾。

（4） 输入参数 pFileAddr

该参数用于指定需要读出的文件和文件下的数据项。不同规范版本的卡内数据文件结构说明详见 10.2。

文件名由 ADF 的文件标识符和 AEF 的文件标识符组成，如 SSSEEF05、DF01EF06。文件名及各数据项之间以“|”分隔，且最后一个数据项以“|”结尾。数据项以记录标识符表示，若同一数据项由多条记录组成，则在数据项后加“:”再加记录号表示。不同文件之间以“$”分隔，且最后应以“$”结束。例如读出 2.0 卡的学位信息 2 表示为：DF01EF15|40:2|$；读出国家/地区代码、学历、学位信息 2 表示为：DF01EF0A|37|$DF01EF15|2A|40:2|$。

当所要读出的文件为循环文件时，如果只指定文件名，函数将读出该文件下的所有记录数据；如果同时给出指定文件名和记录号，函数将读出该文件下的记录号所对应的记录数据。每条记录之间以“|”分隔，每条记录里面的数据项之间以“^”分隔，最后一个数据项以“^”结尾，最后一条记录以“|”结尾。

当所要读出的文件为透明文件时，只需指定文件名，函数将读出该文件下的

所有文件数据。

（5） 输出参数 pOutInfo

当函数执行成功时，该输出参数为读出的由输入参数指定的各数据项，其格式与输入参数 pFileAddr 严格对应且分隔符完全一致。例如读出国家/地区代码的输出参数表示为：DF01EF0A|CHN|$。

当函数执行失败时，该输出参数为错误信息描述。

2.1.3 处理流程

（1） 判断输入参数有效性；

（2） 调用 SSSE32.DLL 动态库中的“打开设备”函数初始化设备，获得设备句柄；

（3） 调用 SSSE32.DLL 动态库中的“卡上电”函数，对社保卡进行上电复位；当需要密钥认证时，对 PSAM 卡进行上电复位；

（4） 选择社保卡社会保障系统环境，同时根据卡片返回的应答信息确定算法环境，根据算法环境标识选择 PSAM 卡相应算法密钥文件，具体方法详见 10.1；

（5） 读取社保卡的卡识别码、卡号，与输入参数 pCardInfo 一一核对，核对成功后进行下一步操作；核对不成功则报错退出；

（6） 根据输入参数 pFileAddr 判断，如果所要读取的信息所在文件可自由读取，则依次读出所需数据。如果文件读控制受 PIN 保护，则先校验 PIN，校验通过后依次读出所需数据。如果文件读控制受 PIN **或**密钥保护，则根据输入参数 iAuthType 选择通过 PIN 校验或密钥外部认证方式，校验或认证通过则依次读出所需数据，校验或认证失败则报错退出。如果文件读控制受 PIN **和**密钥双重保护，则先通过外部认证指令对密钥进行认证，认证失败则报错退出，认证通过则校验 PIN，校验失败则报错退出，校验通过则依次读出所需数据；

（7） 调用 SSSE32.DLL 中的“关闭设备”函数关闭设备，释放端口；

（8） 以上操作过程中出现异常时，均应先调用 SSSE32.DLL 动态库中的“关闭设备”函数关闭设备，释放端口，然后退出函数，返回错误代码和错误信息描述。

注：当所在文件可自由读取或读控制由 PIN 保护时，调用本接口函数。当文件读控制仅受密钥保护、PIN **和**密钥双重保护、PIN **或**密钥保护选择采用密钥保护方式时，若 PSAM 卡算法或密钥不支持（详见 10.5），则调用基于加密机的通用读卡函数进行通用读卡操作。

2.2 iReadCard_HSM_Step1 “基于加密机的通用读卡（步骤一）”

2.2.1 函数定义

基于加密机的通用读卡（步骤一）接口函数定义见表 5。

表 5 基于加密机的通用读卡（步骤一）接口函数定义

函数名称	**基于加密机的通用读卡（步骤一）**					
函数	**iReadCard_HSM_Step1(Type,CardInfo,BuffAddr);**					
语法	long **iReadCard_HSM_Step1**(int iType, char* pCardInfo, char* pFileAddr, char* pOutInfo)					
功能描述	根据所需读取的信息确定需要认证的密钥，并返回认证所需信息。					
参数说明	序号	参数	输入/输出	类型	长度（十进制）	含义
	1	iType	IN	整数	4	操作卡的类型
	2	pCardInfo	IN	字符串	128	卡基本信息
	3	pFileAddr	IN	字符串	1024	文件名及数据项
	4	pOutInfo	OUT	字符串	1024	返回认证信息或错误信息
返回值	0 表示成功；非 0 表示失败。					

2.2.2 参数说明

（1） 输入参数 iType

定义同 1.1.2（1）。

（2） 输入参数 pCardInfo

定义同 2.1.2（3）。

（3） 输入参数 pFileAddr

定义同 2.1.2（4）。

本函数只允许对一个文件进行操作。若传入多个文件则只对第一个文件进行操作，后续内容将被忽略。

（4） 输出参数 pOutInfo

当函数执行成功时，该输出参数为需要计算的认证信息，依次为：算法标识、外部认证密钥地址（参见附录 B）、外部认证过程因子（从卡片获得的随机数）、外部认证鉴别所需的原始信息（从卡片获得的随机数）。各数据项之间以“|”分割，且最后一个数据项以“|”结尾。

当函数执行失败时，该输出参数为错误信息描述。

2.2.3 处理流程

（1） 判断输入参数有效性；

（2） 调用 SSSE32.DLL 动态库中的“打开设备”函数初始化设备，获得设备句柄；

（3） 调用 SSSE32.DLL 动态库中的“卡上电”函数，对社保卡进行上电复位；

（4） 选择社保卡社会保障系统环境，同时根据卡片返回的应答信息确定算法标识，具体方法详见 10.1；

（5） 读取社保卡的卡识别码、卡号，与输入参数 pCardInfo 一一核对，核对成功后进行下一步操作；核对不成功则报错退出；

（6） 根据输入参数 pFileAddr 判断，如果文件读控制受密钥保护、PIN **或**密钥保护，则从卡内取出两个随机数并返回。如果文件读控制受 PIN **和**密钥双重保护，则先校验 PIN，校验通过则从卡内取出两个随机数并返回，校验失败则报错退出；

（7） 以上操作过程中出现异常时，均应先调用 SSSE32.DLL 动态库中的“关闭设备”函数关闭设备，释放端口，然后退出函数，返回错误代码和错误信息描述。

注：当所在文件可自由读取或读控制仅受 PIN 保护时，调用本接口函数将返回错误代码“-2204”（具体参见附录 A），则业务系统继续调用基于 PSAM 卡的通用读卡函数进行通用读卡操作。

2.3 iReadCard_HSM_Step2 “基于加密机的通用读卡（步骤二）”

2.3.1 函数定义

基于加密机的通用读卡（步骤二）接口函数定义见表 6。

表 6 基于加密机的通用读卡（步骤二）接口函数定义

函数名称	**基于加密机的通用读卡（步骤二）**					
函数	**ReadCard_HSM_Step2(**pKey**);**					
语法	long **iReadCard_HSM_Step2**(char* pKey, char* pOutInfo)					
功能描述	根据加密机返回的结果数据对社保卡进行外部认证，通过后读出卡内指定文件的信息。					
参数说明	序号	参数	输入/输出	类型	长度（十进制）	含义
	1	pKey	IN	字符串	32	加密机返回的结果数据
	2	pOutInfo	OUT	字符串	1024*20	读出数据或返回错误信息
返回值	0 表示成功；非 0 表示失败。					

2.3.2 参数说明

（1） 输入参数 pKey

该参数用于传入由加密机返回的结果数据，由鉴别数据（过程因子分散后加密原始信息的密文）和鉴别所需的原始信息拼接组成，总长度为 32 位。

（2） 输出参数 pOutInfo

当函数执行成功时，该输出参数为读出的由 iReadCard_HSM_Step1 函数输入参数 pFileAddr 指定的各数据项，其格式严格对应且分隔符完全一致。

当函数执行失败时，该输出参数为错误信息描述。

2.3.3 处理流程

（1） 判断输入参数有效性；

（2） 通过外部认证指令对密钥进行认证，认证通过则依次读出所需数据，认证失败则报错退出；

（3） 调用 SSSE32.DLL 中的“关闭设备”函数关闭设备，释放端口；

（4） 以上操作过程中出现异常时，均应先调用 SSSE32.DLL 动态库中的“关闭设备”函数关闭设备，释放端口，然后退出函数，返回错误代码和错误信息描述。

3 “通用写卡”函数

3.1 iWriteCard “通用写卡”

3.1.1 函数定义

通用写卡接口函数定义见表 7。

表 7 通用写卡接口函数定义

函数名称	**通用写卡**					
函数	**iWriteCard(Type,CardInfo,BuffAddr,WriteData);**					
语法	long **iWriteCard** (int iType, char* pCardInfo, char* pFileAddr, char* pWriteData, char* pOutInfo)					
功能描述	根据所需写入的信息进行外部认证后写入指定文件的信息。					
参数说明	序号	参数	输入/输出	类型	长度（十进制）	含义
	1	iType	IN	整数	4	操作卡的类型
	2	pCardInfo	IN	字符串	128	卡基本信息
	3	pFileAddr	IN	字符串	1024	文件名及数据项
	4	pWriteData	IN	字符串	1024*20	写入数据项信息
	5	pOutInfo	OUT	字符串	1024	返回空字符串或错误信息
返回值	0 表示成功；非 0 表示失败。					

3.1.2 参数说明

（1） 输入参数 iType

定义同 1.1.2（1）。

（2） 输入参数 pCardInfo

定义同 2.1.2（3）。

（3） 输入参数 pFileAddr

该参数用于指定拟写入的文件和文件下的数据项。不同规范版本的卡内数据文件结构说明详见 10.2。

文件名由 ADF 的文件标识符和 AEF 的文件标识符组成，如 SSSEEF05、

DF01EF06。文件名及各数据项之间以“|”分隔, 且最后一个数据项以“|”结尾。数据项以记录标识符表示，若同一数据项由多条记录组成，则在数据项后加“:”再加记录号表示。不同文件之间以“$”分隔，且最后应以“$”结束。例如写入2.0卡就业状态，表示为：DF01EF07|29|$；写入国家/地区代码，表示为：DF01EF0A|37|$。

当拟写入的文件为循环文件时，只需指定文件名，函数将新增记录；当拟写入的文件为透明文件时，只需指定文件名，函数将更新全部文件数据。

（4） 输入参数 pWriteData

该参数用于传入拟写入的数据项信息。其格式与输入参数 pFileAddr 严格对应且分隔符完全一致。例如写入 2.0 卡就业状态为 1，表示为：DF01EF07|1|$；写入国家/地区代码为 CHN，表示为：DF01EF0A|CHN|$。

（5） 输出参数 pOutInfo

当函数执行成功时，该输出参数为空字符串。

当函数执行失败时，该输出参数为错误信息描述。

3.1.3 处理流程

（1） 判断输入参数有效性；

（2） 调用 SSSE32.DLL 动态库中的“打开设备”函数初始化设备，获得设备句柄；

（3） 调用 SSSE32.DLL 动态库中的“卡上电”函数，分别对社保卡和 PSAM 卡进行上电复位；

（4） 选择社保卡社会保障系统环境，同时根据卡片返回的应答信息确定算法环境，根据算法环境标识选择 PSAM 卡相应算法密钥文件，具体方法详见 10.1；

（5） 读取社保卡的卡识别码、卡号，与输入参数 pCardInfo 一一核对，核对成功后进行下一步操作；核对不成功则报错退出；

（6） 根据输入参数 pFileAddr 判断，如果拟写入的信息所在文件不允许写入（或受 COS 保护，如交易记录文件）则报错退出；如果文件写控制受密钥保护，则通过外部认证指令对密钥进行认证，认证通过则依次写入各项数据，认证失败则退出；

（7） 调用 SSSE32.DLL 中的“关闭设备”函数关闭设备，释放端口。

（8） 以上操作过程中出现异常时，均应先调用 SSSE32.DLL 动态库中的“关闭设备”函数关闭设备，释放端口，然后退出函数，返回错误代码和错误信息描述。

注：业务系统判断到五种情况的错误代码（详见 10.5）后，调用基于加密机的通用写卡

函数继续进行通用写卡操作。

3.2 iWriteCard_HSM_Step1 “基于加密机的通用写卡（步骤一）”

3.2.1 函数定义

基于加密机的通用写卡（步骤一）接口函数定义见表 8。

表 8 基于加密机的通用写卡（步骤一）接口函数定义

函数名称	**基于加密机的通用写卡（步骤一）**					
函数	**iWriteCard_HSM_Step1(Type,CardInfo,pFileAddr);**					
语法	long **iWriteCard_HSM_Step1**(int iType, char* pCardInfo, char* pFileAddr, char* pOutInfo)					
功能描述	根据所需写入的信息确定需要认证的密钥，并返回认证所需信息。					
参数说明	序号	参数	输入/输出	类型	长度（十进制）	含义
	1	iType	IN	整数	4	操作卡的类型
	2	pCardInfo	IN	字符串	128	卡基本信息
	3	pFileAddr	IN	字符串	1024	文件名及数据项
	4	pOutInfo	OUT	字符串	1024	返回认证信息或错误信息
返回值	0 表示成功；非 0 表示失败。					

3.2.2 参数说明

（1） 输入参数 iType

定义同 1.1.2（1）。

（2） 输入参数 pCardInfo

定义同 2.1.2（3）。

（3） 输入参数 pFileAddr

定义同 3.1.2（3）。

本函数只允许对一个文件进行操作。若传入多个文件则只对第一个文件进行操作，后续内容将被忽略。

（4） 输出参数 pOutInfo

当函数执行成功时，该输出参数为需要计算的认证信息，依次为：算法标识、外部认证密钥地址（参见附录 B）、外部认证过程因子（从卡片获得的随机数）、外部认证鉴别所需的原始信息（从卡片获得的随机数）。各数据项之间以“|”分割，且最后一个数据项以“|”结尾。

当函数执行失败时，该输出参数为错误信息描述。

3.2.3 处理流程

（1） 判断输入参数有效性；

（2） 调用 SSSE32.DLL 动态库中的“打开设备”函数初始化设备，获得设备句柄；

（3） 调用 SSSE32.DLL 动态库中的“卡上电”函数，对社保卡进行上电复位；

（4） 选择社保卡社会保障系统环境，同时根据卡片返回的应答信息确定算法标识，具体方法详见 10.1；

（5） 读取社保卡的卡识别码、卡号，与输入参数 pCardInfo 一一核对，核对成功后进行下一步操作；核对不成功则报错退出；

（6） 根据输入参数 pFileAddr 判断，如果拟写入的信息所在文件不允许写入（或受 COS 保护，如交易记录文件）则报错退出；如果文件写控制受密钥保护，则从卡内取出两个随机数并返回；

（7） 以上操作过程中出现异常时，均应先调用 SSSE32.DLL 动态库中的“关闭设备”函数关闭设备，释放端口，然后退出函数，返回错误代码和错误信息描述。

3.3 iWriteCard_HSM_Step2 “基于加密机的通用写卡（步骤二）”

3.3.1 函数定义

基于加密机的通用写卡（步骤二）接口函数定义见表 9。

表 9 基于加密机的通用写卡（步骤二）接口函数定义

函数名称	**基于加密机的通用写卡（步骤二）**					
函数	**iWriteCard_HSM_Step2(**pKey,pWriteData**);**					
语法	long **iWriteCard_HSM_Step2**(char* pKey, char* pWriteData, char* pOutInfo)					
功能描述	根据加密机返回的结果数据对社保卡进行外部认证，通过后写入卡内指定文件的信息。					
参数说明	序号	参数	输入/输出	类型	长度（十进制）	含义
	1	pKey	IN	字符串	32	加密机返回的结果数据
	2	pWriteData	IN	字符串	1024*20	写入数据项信息
	3	pOutInfo	OUT	字符串	1024	返回空字符串或错误信息
返回值	0 表示成功；非 0 表示失败。					

3.3.2 参数说明

1 输入参数 pKey

定义同 2.3.2（1）。

2 输入参数 pWriteData

该参数用于传入要写入的数据项信息，其格式与 iWriteCard_HSM_Step1 函数输入参数 pFileAddr 严格对应且分隔符完全一致。

3 输出参数 pOutInfo

定义同 3.1.2（5）。

3.3.3 处理流程

（1） 判断输入参数有效性；

（2） 通过外部认证指令对密钥进行认证，认证通过则依次写入所需数据，认证

失败则报错退出；

（3） 调用 SSSE32.DLL 中的“关闭设备”函数关闭设备，释放端口；

（4） 以上操作过程中出现异常时，均应先调用 SSSE32.DLL 动态库中的“关闭设备”函数关闭设备，释放端口，然后退出函数，返回错误代码和错误信息描述。

4 “PIN 校验”函数

4.1 iVerifyPIN “PIN 校验”

4.1.1 函数定义

PIN 校验接口函数定义见表 10。

表 10 PIN 校验接口函数定义

函数名称	**PIN 校验**					
函数	**iVerifyPIN(Type)**					
语法	long **iVerifyPIN**(int iType, char* pOutInfo)					
功能描述	校验 PIN。					
参数说明	序号	参数	输入/输出	类型	长度（十进制）	含义
	1	iType	IN	整数	4	操作卡的类型
	2	pOutInfo	OUT	字符串	512	返回空字符串或错误信息
返回值	0 表示成功；非 0 表示失败。					

4.1.2 参数说明

（1） 输入参数 iType

定义同 1.1.2（1）。

（2） 输出参数 pOutInfo

定义同 3.1.2（5）。

4.1.3 处理流程

（1） 判断输入参数有效性；

（2） 调用 SSSE32.DLL 动态库中的“打开设备”函数初始化设备，获得设备句柄；

（3） 调用 SSSE32.DLL 动态库中的“卡上电”函数，对社保卡进行上电复位；

（4） 选择社保卡社会保障系统环境；

（5） 启动密码键盘，语音和显示同时提示持卡人“请输入密码”，获取密码键盘上持卡人输入的 PIN（有效长度为 4-16 位数字），并以此密码进行 PIN 校验。若 PIN 校验失败，将剩余可尝试次数通过输出参数 pOutInfo 返回；

（6） 调用 SSSE32.DLL 中的“关闭设备”函数关闭设备，释放端口；

（7） 以上操作过程中出现异常时，均应先调用 SSSE32.DLL 动态库中的“关闭设备”函数关闭设备，释放端口，然后退出函数，返回错误代码和错误信息描述。

5 “PIN 修改”函数

5.1 iChangePIN “PIN 修改”

5.1.1 函数定义

PIN 修改接口函数定义见表 11。

表 11 PIN 修改接口函数定义

函数名称	**PIN 修改**					
函数	**iChangePIN(Type);**					
语法	long **iChangePIN**(int iType, char* pOutInfo)					
功能描述	修改 PIN。					
参数说明	序号	参数	输入/输出	类型	长度（十进制）	含义
	1	iType	IN	整数	4	操作卡的类型
	2	pOutInfo	OUT	字符串	512	返回空字符串或错误信息
返回值	0 表示成功；非 0 表示失败。					

5.1.2 参数说明

（1） 输入参数 iType

定义同 1.1.2（1）。

（2） 输出参数 pOutInfo

定义同 3.1.2（5）。

5.1.3 处理流程

（1） 判断输入参数有效性；

（2） 调用 SSSE32.DLL 动态库中的“打开设备”函数初始化设备，获得设备句柄；

（3） 调用 SSSE32.DLL 动态库中的“卡上电”函数，对社保卡进行上电复位；

（4） 选择社保卡社会保障系统环境；

（5） 启动密码键盘，语音和显示同时依次提示持卡人“请输入原密码”、“请输入新密码”、“请再次输入新密码”，分别获取密码键盘上持卡人输入的原 PIN 和新 PIN（有效长度为 4-16 位数字），判断两次输入的新密码是否一致，并以此新密码进行 PIN 修改；

（6） 调用 SSSE32.DLL 中的“关闭设备”函数关闭设备，释放端口；

（7） 以上操作过程中出现异常时，均应先调用 SSSE32.DLL 动态库中的“关闭设备”函数关闭设备，释放端口，然后退出函数，返回错误代码和错误信

息描述。

6 “PIN 重置”函数

6.1 iReloadPIN “PIN 重置”

6.1.1 函数定义

PIN 重置接口函数定义见表 12。

表 12 PIN 重置接口函数定义

函数名称	**PIN 重置**					
	iReloadPIN(Type,pCardInfo**);**					
语法	long **iReloadPIN**(int iType, char* pCardInfo, char* pOutInfo)					
功能描述	重置 PIN。					
参数说明	序号	参数	输入/输出	类型	长度（十进制）	含义
	1	iType	IN	整数	4	操作卡的类型
	2	pCardInfo	IN	字符串	128	卡基本信息
	3	pOutInfo	OUT	字符串	512	返回空字符串或错误信息
返回值	0 表示成功；非 0 表示失败。					

6.1.2 参数说明

（1） 输入参数 iType

定义同 1.1.2（1）。

（2） 输入参数 pCardInfo

定义同 2.1.2（3）。

（3） 输出参数 pOutInfo

定义同 3.1.2（5）。

6.1.3 处理流程

（1） 判断输入参数有效性；

（2） 调用 SSSE32.DLL 动态库中的“打开设备”函数初始化设备，获得设备句柄；

（3） 调用 SSSE32.DLL 动态库中的“卡上电”函数，分别对社保卡和 PSAM 卡进行上电复位；

（4） 选择社保卡社会保障系统环境，同时根据卡片返回的应答信息确定算法环境，根据算法环境标识选择 PSAM 卡相应算法密钥文件，具体方法详见 10.1；

（5） 读取社保卡的卡识别码、卡号，与输入参数 pCardInfo 一一核对，核对成功后进行下一步操作；核对不成功则报错退出；

（6） 启动密码键盘，语音和显示同时依次提示持卡人“请输入新密码”、“请

再次输入新密码”，分别获取密码键盘上持卡人输入的新 PIN（有效长度为 4-16 位数字），判断两次输入的新密码是否一致，并以此新密码进行 PIN 重置操作；

（7） 调用 SSSE32.DLL 中的“关闭设备”函数关闭设备，释放端口；

（8） 以上操作过程中出现异常时，均应先调用 SSSE32.DLL 动态库中的“关闭设备”函数关闭设备，释放端口，然后退出函数，返回错误代码和错误信息描述。

注：业务系统判断到五种情况的错误代码（详见 10.5）后，调用基于加密机的 PIN 重置函数继续进行 PIN 重置操作。

6.2 iReloadPIN_HSM_Step1 “基于加密机的 PIN 重置（步骤一）”

6.2.1 函数定义

基于加密机的 PIN 重置（步骤一）接口函数定义见表 13。

表 13 基于加密机的 PIN 重置（步骤一）接口函数定义

函数名称	**基于加密机的 PIN 重置（步骤一）**					
函数	**iReloadPIN_HSM_Step1(Type,CardInfo);**					
语法	long **iReloadPIN_HSM_Step1**(int iType, char* pCardInfo, char* pOutInfo)					
功能描述	获取新 PIN，返回所需的认证信息。					
参数说明	序号	参数	输入/输出	类型	长度（十进制）	含义
	1	iType	IN	整数	4	操作卡的类型
	2	pCardInfo	IN	字符串	128	卡基本信息
	3	pOutInfo	OUT	字符串	512	返回认证信息或错误信息
返回值	0 表示成功；非 0 表示失败。					

6.2.2 参数说明

（1） 输入参数 iType

定义同 1.1.2（1）。

（2） 输入参数 pCardInfo

定义同 2.1.2（3）。

（3） 输出参数 pOutInfo

定义同 3.2.2（4）。

6.2.3 处理流程

（1） 判断输入参数有效性；

（2） 调用 SSSE32.DLL 动态库中的 “打开设备”函数初始化设备，获得设备句柄；

（3） 调用 SSSE32.DLL 动态库中的“卡上电”函数，对社保卡进行上电复位；

（4） 选择社保卡社会保障系统环境，同时根据卡片返回的应答信息确定算法标

识，具体方法详见 10.1；

（5） 读取社保卡的卡识别码、卡号，与输入参数 pCardInfo 一一核对，核对成功后进行下一步操作；核对不成功则报错退出；

（6） 获取认证相关信息后返回；

（7） 以上操作过程中出现异常时，均应先调用 SSSE32.DLL 动态库中的“关闭设备”函数关闭设备，释放端口，然后退出函数，返回错误代码和错误信息描述。

6.3 iReloadPIN_HSM_Step2 “基于加密机的 PIN 重置（步骤二）”

6.3.1 函数定义

基于加密机的 PIN 重置（步骤二）接口函数定义见表 14。

表 14 基于加密机的 PIN 重置（步骤二）接口函数定义

函数名称	**基于加密机的 PIN 重置（步骤二）**					
函数	**iReloadPIN_HSM_Step2(**pKey**);**					
语法	long **iReloadPIN_HSM_Step2**(char* pKey, char* pOutInfo)					
功能描述	进行认证，返回安全报文计算数据。					
参数说明	序号	参数	输入/输出	类型	长度（十进制）	含义
	1	pKey	IN	字符串	32	加密机返回的结果数据
	2	pOutInfo	OUT	字符串	1024	返回安全报文计算数据或错误信息
返回值	0 表示成功；非 0 表示失败。					

6.3.2 参数说明

（1） 输入参数 pKey

定义同 2.3.2（1）。

（2） 输出参数 pOutInfo

当函数执行成功时，该输出参数为需要由加密机计算的安全报文数据，依次为：算法标识、安全报文计算密钥地址（参见附录 B）、安全报文计算过程因子（从卡片获得的随机数）、APDU 命令头、APDU 命令明文数据（新 PIN）。各数据项之间以“|”分割，且最后一个数据项以“|”结尾。

当函数执行失败时，该输出参数为错误信息描述。

6.3.3 处理流程

（1） 判断输入参数有效性；

（2） 进行外部认证；

（3） 启动密码键盘，语音和显示同时依次提示持卡人“请输入新密码”、“请再次输入新密码”，分别获取密码键盘上持卡人输入的新 PIN（有效长度为 4-16 位数字），判断两次输入的新密码是否一致；

（4） 通过后产生安全报文计算数据；

（5） 以上操作过程中出现异常时，均应先调用 SSSE32.DLL 动态库中的“关闭设备”函数关闭设备，释放端口，然后退出函数，返回错误代码和错误信息描述。

6.4 iReloadPIN_HSM_Step3 “基于加密机的 PIN 重置（步骤三）”

6.4.1 函数定义

基于加密机的 PIN 重置（步骤三）接口函数定义见表 15。

表 15 基于加密机的 PIN 重置（步骤三）接口函数定义

函数名称	**基于加密机的 PIN 重置（步骤三）**					
函数	**iReloadPIN_HSM_Step3(**pKey**);**					
语法	long **iReloadPIN_HSM_Step3**(char* pKey, char* pOutInfo)					
功能描述	完成 PIN 重置。					
参数说明	序号	参数	输入/输出	类型	长度（十进制）	含义
	1	pKey	IN	字符串	50	安全报文数据
	2	pOutInfo	OUT	字符串	1024	返回空字符串或错误信息
返回值	0 表示成功；非 0 表示失败。					

6.4.2 参数说明

（1） 输入参数 pKey

该参数用于传入由加密机计算的安全报文数据，由命令头、加密数据和 MAC 拼接组成，总长度为 34 位（DES 算法）或 50 位（SSF33/SM4 算法）。

（2） 输出参数 pOutInfo

定义同 3.1.2（5）。

6.4.3 处理流程

（1） 判断输入参数有效性；

（2） 进行 PIN 重置操作；

（3） 调用 SSSE32.DLL 中的“关闭设备”函数关闭设备，释放端口；

（4） 以上操作过程中出现异常时，均应先调用 SSSE32.DLL 动态库中的“关闭设备”函数关闭设备，释放端口，然后退出函数，返回错误代码和错误信息描述。

7 “PIN 解锁”函数

7.1 iUnblockPIN “PIN 解锁”

7.1.1 函数定义

PIN 解锁接口函数定义见表 16。

表 16 PIN 解锁接口函数定义

函数名称	**PIN 解锁**					
函数	**iUnblockPIN(Type,CardInfo);**					
语法	long **iUnblockPIN**(int iType, char* pCardInfo, char* pOutInfo)					
功能描述	解锁 PIN。					
参数说明	序号	参数	输入/输出	类型	长度（十进制）	含义
	1	iType	IN	整数	4	操作卡的类型
	2	pCardInfo	IN	字符串	128	卡基本信息
	3	pOutInfo	OUT	字符串	512	返回空字符串或错误信息
返回值	0 表示成功；非 0 表示失败。					

7.1.2 参数说明

- 输入参数 iType
  定义同 1.1.2（1）。
- 输入参数 pCardInfo
  定义同 2.1.2（3）。
- 输出参数 pOutInfo
  定义同 3.1.2（5）。

7.1.3 处理流程

（1） 判断输入参数有效性；

（2） 调用 SSSE32.DLL 动态库中的 “打开设备”函数初始化设备，获得设备句柄；

（3） 调用 SSSE32.DLL 动态库中的“卡上电”函数，分别对社保卡和 PSAM 卡进行上电复位；

（4） 选择社保卡社会保障系统环境，同时根据卡片返回的应答信息确定算法环境，根据算法环境标识选择 PSAM 卡相应算法密钥文件，具体方法详见 10.1；

（5） 读取社保卡的卡识别码、卡号，与输入参数 pCardInfo 一一核对，核对成功后进行下一步操作；核对不成功则报错退出；

（6） 进行 PIN 解锁操作；

（7） 调用 SSSE32.DLL 中的“关闭设备”函数关闭设备，释放端口；

（8） 以上操作过程中出现异常时，均应先调用 SSSE32.DLL 动态库中的“关闭设备”函数关闭设备，释放端口，然后退出函数，返回错误代码和错误信息描述。

注：业务系统判断到五种情况的错误代码（详见 10.5）后，调用基于加密机的 PIN 解锁函数继续进行 PIN 解锁操作。

7.2 iUnblockPIN_HSM_Step1 “基于加密机的 PIN 解锁（步骤一）”

7.2.1 函数定义

基于加密机的 PIN 解锁（步骤一）接口函数定义见表 17。

表 17 基于加密机的 PIN 解锁（步骤一）接口函数定义

函数名称	基于加密机的 PIN 解锁（步骤一）					
函数	**iUnblockPIN_HSM_Step1(Type,CardInfo);**					
语法	long **iUnblockPIN_HSM_Step1**(int iType, char* pCardInfo, char* pOutInfo)					
功能描述	获得所需的认证信息。					
参数说明	序号	参数	输入/输出	类型	长度（十进制）	含义
	1	iType	IN	整数	4	操作卡的类型
	2	pCardInfo	IN	字符串	128	卡基本信息
	3	pOutInfo	OUT	字符串	512	返回认证信息或错误信息
返回值	0 表示成功；非 0 表示失败。					

7.2.2 参数说明

- 输入参数 iType
  定义同 1.1.2（1）。
- 输入参数 pCardInfo
  定义同 2.1.2（3）。
- 输出参数 pOutInfo
  定义同 3.2.2（4）。

7.2.3 处理流程

（1） 判断输入参数有效性；

（2） 调用 SSSE32.DLL 动态库中的 “打开设备”函数初始化设备，获得设备句柄；

（3） 调用 SSSE32.DLL 动态库中的“卡上电”函数，对社保卡进行上电复位；

（4） 选择社保卡社会保障系统环境，同时根据卡片返回的应答信息确定算法标识，具体方法详见 10.1；

（5） 读取社保卡的卡识别码、卡号，与输入参数 pCardInfo 一一核对，核对成功后进行下一步操作；核对不成功则报错退出；

（6） 获取认证相关信息后返回；

（7） 以上操作过程中出现异常时，均应先调用 SSSE32.DLL 动态库中的“关闭设备”函数关闭设备，释放端口，然后退出函数，返回错误代码和错误信息描述。

7.3 iUnblockPIN_HSM_Step2 “基于加密机的 PIN 解锁（步骤二）”

7.3.1 函数定义

基于加密机的 PIN 解锁（步骤二）接口函数定义见表 18。

表 18 基于加密机的 PIN 解锁（步骤二）接口函数定义

函数名称	**基于加密机的 PIN 解锁（步骤二）**					
函数	**iUnblockPIN_HSM_Step2(**pKey**);**					
语法	long **iUnblockPIN_HSM_Step2**(char* pKey, char* pOutInfo)					
功能描述	进行认证，返回安全报文计算数据。					
参数说明	序号	参数	输入/输出	类型	长度（十进制）	含义
	1	pKey	IN	字符串	32	加密机返回的结果数据
	2	pOutInfo	OUT	字符串	1024	返回安全报文计算数据或错误信息
返回值	0 表示成功；非 0 表示失败。					

7.3.2 参数说明

（1） 输入参数 pKey

定义同 2.3.2（1）。

（2） 输出参数 pOutInfo

当函数执行成功时，该输出参数为需要由加密机计算的安全报文数据，依次为：算法标识、安全报文计算密钥地址（参见附录 B）、安全报文计算过程因子（从卡片获得的随机数）、APDU 命令头、APDU 命令明文数据（空字符串）。各数据项之间以“|”分割，且最后一个数据项以“|”结尾。

当函数执行失败时，该输出参数为错误信息描述。

7.3.3 处理流程

（1） 判断输入参数有效性；

（2） 进行外部认证，通过后产生安全报文计算数据；

（3） 以上操作过程中出现异常时，均应先调用 SSSE32.DLL 动态库中的“关闭设备”函数关闭设备，释放端口，然后退出函数，返回错误代码和错误信息描述。

7.4 iUnblockPIN_HSM_Step3 “基于加密机的 PIN 解锁（步骤三）”

7.4.1 函数定义

基于加密机的 PIN 解锁（步骤三）接口函数定义见表 19。

表 19 基于加密机的 PIN 解锁（步骤三）接口函数定义

函数名称	**基于加密机的 PIN 解锁（步骤三）**
函数	**iUnblockPIN_HSM_Step3(**pKey**);**

语法	long **iUnblockPIN_HSM_Step3**(char* pKey, char* pOutInfo)					
功能描述	完成 PIN 解锁。					
参数说明	序号	参数	输入/输出	类型	长度（十进制）	含义
	1	pKey	IN	字符串	18	安全报文数据
	2	pOutInfo	OUT	字符串	1024	返回空字符串或错误信息
返回值	0 表示成功；非 0 表示失败。					

7.4.2 参数说明

（1） 输入参数 pKey

该参数用于传入由加密机计算的安全报文数据，由命令头和 MAC 拼接组成，总长度为 18 位。

（2） 输出参数 pOutInfo

定义同 3.1.2（5）。

7.4.3 处理流程

（1） 判断输入参数有效性；

（2） 进行 PIN 解锁操作；

（3） 调用 SSSE32.DLL 中的“关闭设备”函数关闭设备，释放端口；

（4） 以上操作过程中出现异常时，均应先调用 SSSE32.DLL 动态库中的“关闭设备”函数关闭设备，释放端口，然后退出函数，返回错误代码和错误信息描述。

8 “消费交易”函数

8.1 iDoDebit “消费交易”

8.1.1 函数定义

消费交易接口函数定义见表 20。

表 20 消费交易接口函数定义

函数名称	**消费交易**					
函数	**iDoDebitAB(Type,pCardInfo,pPayInfo);**					
语法	long **iDoDebit**(int iType, char* pCardInfo, char* pPayInfo, char* pOutInfo)					
功能描述	执行社保卡消费交易并写入消费记录。					
参数说明	序号	参数	输入/输出	类型	长度（十进制）	含义
	1	iType	IN	整数	4	操作卡的类型
	2	pCardInfo	IN	字符串	128	卡基本信息
	3	pPayInfo	IN	字符串	512	消费信息
	4	pOutInfo	OUT	字符串	512	返回交易验证数据或错误信息
返回值	0 表示成功；非 0 表示失败。					

8.1.2 参数说明

（1） 输入参数 iType

定义同 1.1.2（1）。

（2） 输入参数 pCardInfo

定义同 2.1.2（3）。

（3） 输入参数 pPayInfo

该参数用于传入消费相关信息，依次为：本次消费总金额(小于 42949672.95 的小数，小数点后保留两位)、个人账户交易金额和统筹基金支付金额相加的总金额（小于 42949672.95 的小数，小数点后保留两位）、交易时间（格式为 YYYYMMDDHHMMSS）。各数据项之间以“|”分割，且最后一个数据项以“|”结尾。

（4） 输出参数 pOutInfo

当函数执行成功时，该输出参数为交易验证码及相关信息，依次为：算法标识、密钥地址（参见附录 B）、交易金额（转换成十六进制向卡片发送命令时的后两个金额拼接组成）、交易类型、终端机编号、终端交易序号、交易时间（格式为 YYYYMMDDHHMMSS）、交易验证码（TAC）。各数据项之间以“|”分割，且最后一个数据项以“|”结尾。

当函数执行失败时，该输出参数为错误信息描述。

8.1.3 处理流程

（1） 判断输入参数有效性；

（2） 调用 SSSE32.DLL 动态库中的 “打开设备”函数初始化设备，获得设备句柄；

（3） 调用 SSSE32.DLL 动态库中的“卡上电”函数，分别对社保卡和 PSAM 卡进行上电复位；

（4） 选择社保卡社会保障系统环境，同时根据卡片返回的应答信息确定算法环境，根据算法环境标识选择 PSAM 卡相应算法密钥文件，具体方法详见 10.1；

（5） 读取社保卡的卡识别码、卡号，与输入参数 pCardInfo 一一核对，核对成功后进行下一步操作；核对不成功则报错退出；

（6） 启动密码键盘，语音和显示同时提示持卡人“请输入密码”，获取密码键盘上持卡人输入的 PIN（有效长度为 4-16 位数字），并以此密码进行 PIN 校验；

（7） 执行医疗消费初始化命令，由 PSAM 卡计算出 MAC1 后执行医疗消费命令，再对 PSAM 卡进行 MAC2 验证，最后产生交易验证码及相关验证计

算数据；

（8） 调用 SSSE32.DLL 中的“关闭设备”函数关闭设备，释放端口；

（9） 以上操作过程中出现异常时，均应先调用 SSSE32.DLL 动态库中的“关闭设备”函数关闭设备，释放端口，然后退出函数，返回错误代码和错误信息描述。

注：业务系统判断到五种情况的错误代码（详见 10.5）后，调用基于加密机的消费交易接口继续进行消费交易操作。

8.2 iDoDebit_HSM _Step1“基于加密机的消费交易（步骤一）”

8.2.1 函数定义

基于加密机的消费交易（步骤一）接口函数定义见表 21。

表 21 基于加密机的消费交易（步骤一）接口函数定义

函数名称	**基于加密机的消费交易（步骤一）**					
函数	**iDoDebit_HSM_Step1(Type,CardInfo,pPayInfo);**					
语法	long **iDoDebit_HSM_Step1**(int iType, char* pCardInfo, char* pPayInfo, char* pOutInfo)					
功能描述	执行社保卡消费交易初始化命令并返回交易认证相关数据					
参数说明	序号	参数	输入/输出	类型	长度（十进制）	含义
	1	iType	IN	整数	4	操作卡的类型
	2	pCardInfo	IN	字符串	128	卡基本信息
	3	pPayInfo	IN	字符串	512	消费信息
	4	pOutInfo	OUT	字符串	512	返回交易认证数据或错误信息
返回值	0 表示成功；非 0 表示失败。					

8.2.2 参数说明

（1） 输入参数 iType

定义同 1.1.2（1）。

（2） 输入参数 pCardInfo

定义同 2.1.2（3）。

（3） 输入参数 pPayInfo

定义同 8.1.2（3）。

（4） 输出参数 pOutInfo

当函数执行成功时，该输出参数为用于计算 MAC1 的相关交易认证数据，依次为：算法标识、密钥地址（参见附录 B）、伪随机数、医疗消费交易序号、交易金额（转换成十六进制向卡片发送命令时的后两个金额拼接组成）、交易类型、终端机编号、交易时间（格式为 YYYYMMDDHHMMSS）。各数据项之间以“|”分割，且最后一个数据项以“|”结尾。

当函数执行失败时，该输出参数为错误信息描述。

注：当没有 PSAM 卡时，终端机编号返回 12 个 0，即 6 个 0x00 对应的字符。

8.2.3 处理流程

（1） 判断输入参数有效性；

（2） 调用 SSSE32.DLL 动态库中的 “打开设备”函数初始化设备，获得设备句柄；

（3） 调用 SSSE32.DLL 动态库中的“卡上电”函数，对社保卡进行上电复位；

（4） 选择社保卡社会保障系统环境，同时根据卡片返回的应答信息确定算法标识，具体方法详见 10.1；

（5） 读取社保卡的卡识别码、卡号，与输入参数 pCardInfo 一一核对，核对成功后进行下一步操作；核对不成功则报错退出；

（6） 启动密码键盘，语音和显示同时提示持卡人“请输入密码”，获取密码键盘上持卡人输入的 PIN（有效长度为 4-16 位数字），并以此密码进行 PIN 校验；

（7） 执行医疗消费初始化命令，返回用于计算 MAC1 的相关交易认证数据；

（8） 以上操作过程中出现异常时，均应先调用 SSSE32.DLL 动态库中的“关闭设备”函数关闭设备，释放端口，然后退出函数，返回错误代码和错误信息描述。

8.3 iDoDebit_HSM _Step2“基于加密机的消费交易（步骤二）”

8.3.1 函数定义

基于加密机的消费交易（步骤二）接口函数定义见表 22。

表 22 基于加密机的消费交易（步骤二）接口函数定义

函数名称	**基于加密机的消费交易（步骤二）**					
函数	**iDoDebit_HSM_Step2AB(pKey);**					
语法	long **iDoDebit_ HSM_Step2** (char* pKey, char* pOutInfo)					
功能描述	完成消费交易写入消费记录					
参数说明	序号	参数	输入/输出	类型	长度（十进制）	含义
	1	pKey	IN	字符串	128	交易认证数据
	2	pOutInfo	OUT	字符串	1024	返回交易验证数据或错误信息
返回值	0 表示成功；非 0 表示失败。					

8.3.2 参数说明

（1） 输入参数 pKey

该参数用于传入由加密机计算的交易认证数据，依次为：终端交易序号、交易时间、MAC1。各数据项之间以“|”分割，且最后一个数据项以“|”结尾。

（2） 输出参数 pOutInfo

当函数执行成功时，该输出参数为交易验证码及相关信息，依次为：MAC2、算法标识、密钥地址（参见附录 B）、交易金额（转换成十六进制向卡片发送命令时的后两个金额拼接组成）、交易类型、终端机编号、终端交易序号、交易时间（格式为 YYYYMMDDHHMMSS）、交易验证码（TAC）。各数据项之间以“|”分割，且最后一个数据项以“|”结尾。

当函数执行失败时，该输出参数为错误信息描述。

8.3.3 处理流程

（1） 判断输入参数有效性；

（2） 执行医疗消费命令，产生交易验证码及相关验证计算数据；

（3） 调用 SSSE32.DLL 中的“关闭设备”函数关闭设备，释放端口；

（4） 以上操作过程中出现异常时，均应先调用 SSSE32.DLL 动态库中的“关闭设备”函数关闭设备，释放端口，然后退出函数，返回错误代码和错误信息描述。

9 “读消费交易记录”函数

9.1 iReadDebitRecord “读消费交易记录”

9.1.1 函数定义

读消费交易记录接口函数定义见表 23。

表 23 读取消费交易记录接口函数定义

函数名称	**读消费交易记录**					
函数	**iReadDebitRecord(iType);**					
语法	long **iReadDebitRecord**(int iType, char* pOutInfo)					
功能描述	读取消费交易记录。					
参数说明	序号	参数	输入/输出	类型	长度（十进制）	含义
	1	iType	IN	整数	4	操作卡的类型
	2	pOutInfo	OUT	字符串	2048	返回交易记录或错误信息
返回值	0 表示成功；非 0 表示失败。					

9.1.2 参数说明

◆ 输入参数 iType

定义同 1.1.2（1）。

◆ 输出参数 pOutInfo

当函数执行成功时，该输出参数为读出的交易记录，每条记录由交易序号、终端机编号、交易时间（格式为 YYYYMMDDHHMMSS）、本次消费总金额、个人账户交易金额和统筹基金支付金额相加的总金额组成。每条记录之间以“|”

分隔，每条记录里面的数据项之间以“^”分隔，最后一个数据项以“^”结尾，最后一条记录以“|”结尾。

当函数执行失败时，该输出参数为错误信息描述。

9.1.3 处理流程

（1） 判断输入参数有效性；

（2） 调用 SSSE32.DLL 动态库中的“打开设备”函数初始化设备，获得设备句柄；

（3） 选择社保卡社会保障系统环境；

（4） 启动密码键盘，语音和显示同时提示持卡人“请输入密码”，获取密码键盘上持卡人输入的 PIN（有效长度为 4-16 位数字），并以此密码进行 PIN 校验；

（5） 循环读取社保卡内存储的消费交易记录，按格式组织后返回；

（6） 调用 SSSE32.DLL 中的“关闭设备”函数关闭设备，释放端口；

（7） 以上操作过程中出现异常时，均应先调用 SSSE32.DLL 动态库中的“关闭设备”函数关闭设备，释放端口，然后退出函数，返回错误代码和错误信息描述。

10 “读银行卡号的函数”函数

10.1 iReadICCardNum“读取银行卡号”

10.1.1 函数定义

函数名称	读银行卡号					
函数	iReadICCardNum()					
语法	long iReadICCardNum(char* pCardNum, char *pErrMsg)					
功能描述	读取消费交易记录。					
参数说明	序号	参数	输入/输出	类型	长度（十进制）	含义
	1	pCardNum	OUT	整数	19	银行卡号
	2	pErrMsg	OUT	字符串	200	错误信息
返回值	0 表示成功；非 0 表示失败。					

11“读取身份证号码”函数

11.1 iReadIdentityCard“读取身份证信息”

11.1.1 函数定义

函数名称	读取身份证信息

函数	iReadIdentityCard()					
语法	long iReadIdentityCard(char *pOutInfo,char *pErrMsg);					
功能描述	读取身份证信息。					
参数说明	序号	参数	输入/输出	类型	长度（十进制）	含义
	1	pOutInfo	OUT	整数	2048	身份证信息
		pErrMsg	OUT	字符串	200	错误信息
返回值	0 表示成功；非 0 表示失败。					

pOutInfo 参数包含的内容为:

姓名|性别|民族|出生日期|地址|身份证号码|签发机关|签发日期|有效期限|照片(base64)|

10 关键问题说明

10.1 算法环境选择

使用“SELECT”命令通过文件名方式选择社保卡社会保障系统环境，同时根据卡片返回的应答信息确定算法环境，由“9F0C”带出的“86”模板代表算法，“01”为 DES 算法，“02”为 SSF33 算法，“03”为 SM4 算法；如无此模板，需根据发卡地区行政区划代码（卡识别码前 6 位）的前 2 位编码的不同数值进行选择，如果是“11”，则为 SSF33 算法，否则为 DES 算法；之后再根据算法环境标识选择相应算法密钥文件。

10.2 规范版本判断方法

卡内规范版本数据项共 4 位，格式如 1.0x、2.0x、3.0x 等。用卡时，只根据前 3 位进行规范版本判断。规范版本为“1.0x”、“2.0x”、“3.0x”的社保卡，分别简称为 1.0 卡、2.0 卡、3.0 卡，如卡内版本为空时，默认为 1.0x。

1.0 卡的卡内数据文件结构参见《社会保障（个人）卡规范》（LB 002—2000）。2.0 卡的卡内数据文件结构参见《社会保障卡规范 第 6 部分：应用数据结构》（LD/T 32.6—2015）。

2.0 卡或 3.0 卡读个人基本信息时需读出“姓名扩展”，与“姓名”拼接后返回完整姓名。3.0 卡的 SSSEEF06 个人基本信息文件增加了读控制密钥“RK_{SSSE}”，读取该文件信息前需进行外部认证。

10.3 密钥分散因子提取方法

提取密钥分散因子的方法如下：根据发卡地区行政区划代码（卡识别码前 6 位）的前 2 位编码的不同数值进行选择，如果是“33”，则将卡识别码的前 2 个字节展开为 4 字节的 ASCII 码（如：3301 展开为“33 33 30 31”）后补“30 30 73 78”，形成 8 个字节的二级分散因子；否则，将卡识别码的前 3 个字节展开

为 6 字节的 ASCII 码（如：650100 展开为“36 35 30 31 30 30”），后补“73 78”，形成 8 个字节的二级分散因子。

10.4 卡号读取长度

“卡号”标准长度应为 9 位。读取“卡号”时只读前 9 位。

10.5 PSAM 卡认证和加密机认证选择

PSAM 卡认证是首选模式。当出现以下五种情况：1）无 PSAM 卡；2）PSAM 卡算法不支持（即 PSAM 卡内没有 SSF33 算法或 SM4 算法）；3）PSAM 卡内没有 RK_{SSSE} 密钥（3.0 卡读个人基本信息需要 RK_{SSSE} 密钥外部认证）；4）PSAM 卡内未找到引用数据、未找到密钥；5）PSAM 卡密钥级别不够，分别返回特定错误代码“-2201”、“-2202”、“-2203”、“-27272”、“-24”（具体参见附录 A）。业务系统判断到此五种错误代码后，调用基于加密机的函数接口继续进行相应操作。业务系统调用读写终端接口流程具体参见附录 C。

附录 A

常见错误信息

常见错误信息见表 A.1。

表 A.1 常见错误信息

返回值	错误信息描述
-1	卡类型不对
-2	无卡
-3	有卡未上电
-4	卡无应答
-5	加载动态库错
-11	读卡器连接错
-12	未建立连接
-13	（动态库）不支持该命令
-14	（发给动态库的）命令参数错
-15	信息校验和出错
-20	卡识别码格式错
-21	内部认证失败（用户卡不合法）
-22	传入数据与卡内不符
-23	传入数据不合法
-24	PSAM 卡密钥级别不够
-31	用户取消密码输入
-32	密码输入操作超时
-33	输入密码长度错
-34	两次输入密码不一致
-35（预留）	初始密码不能交易
-36（预留）	不能改为初始密码
-41	运算数据含非法字符
-42	运算数据长度错
-51	PIN 校验失败，剩余次数 N 次（根据卡返回信息）
-52	PIN 锁定
-2201	无 PSAM 卡
-2202	PSAM 卡算法不支持(即 PSAM 卡内没有 SSF33 算法或 SM4 算法）
-2203	PSAM 卡内没有 RK_{SSSE} 密钥（3.0 卡读个人基本信息需要 RK_{SSSE} 密钥外部认证）
-2204	不需要加密机认证
-25536、-25537、-25538	外部认证失败，剩余可尝试次数 2、1、0 次
-26368	Lc/Le 不正确
-26881	命令不接受（无效状态）
-27009	命令与文件结构不相符、当前文件非所需文件
-27010	不满足安全条件
-27011	密钥锁定（算法锁定）鉴别方法锁定
-27012	引用数据无效、随机数无效
-27013	不满足使用条件、应用被锁定、应用未选择、余额上溢

-27016	安全报文数据项不正确、MAC 不正确
-27264	数据域参数不正确
-27265	不支持该功能、卡中无 MF、卡被锁定、应用锁定
-27266	未找到文件、文件标识相重、SFI 不正确
-27267	未找到记录
-27272	未找到引用数据、未找到密钥
-37634	MAC 无效
-37635	应用已被永久锁定、卡片锁定
-37891	PSAM 卡不支持消费交易
-37894	所需 MAC（或/和 TAC）不可用
其他	未知错误

附录 B

密钥逻辑地址

B.1 部级密钥逻辑地址

部级密钥逻辑地址见表 B.1。

表 B.1 部级密钥逻辑地址

地址	类别	标识	长度	组数	名称	适用的应用范围
004C	07	0A	16	01	RK_{SSSE}	指纹和相片信息读取
004D	07	0A	16	02	RK_{SSSE}	指纹和相片信息读取
004E	07	0A	16	03	RK_{SSSE}	指纹和相片信息读取
004F	07	2E	16	01	$RK1_{DF01}$	公共应用信息读取
0050	07	2E	16	02	$RK1_{DF01}$	公共应用信息读取
0051	07	2E	16	03	$RK1_{DF01}$	公共应用信息读取
0052	07	2F	16	01	$UK6_{DF01}$	学历信息更新
0053	07	2F	16	02	$UK6_{DF01}$	学历信息更新
0054	07	2F	16	03	$UK6_{DF01}$	学历信息更新
0055	07	30	16	01	$UK7_{DF01}$	预留信息 1 更新
0056	07	30	16	02	$UK7_{DF01}$	预留信息 1 更新
0057	07	30	16	03	$UK7_{DF01}$	预留信息 1 更新
0058	07	31	16	01	$UK8_{DF01}$	预留信息 2 更新
0059	07	31	16	02	$UK8_{DF01}$	预留信息 2 更新
005A	07	31	16	03	$UK8_{DF01}$	预留信息 2 更新
005B	07	32	16	01	$UK9_{DF01}$	预留信息 3 更新
005C	07	32	16	02	$UK9_{DF01}$	预留信息 3 更新
005D	07	32	16	03	$UK9_{DF01}$	预留信息 3 更新
005E	07	33	16	01	UKA_{DF01}	预留信息 4 更新
005F	07	33	16	02	UKA_{DF01}	预留信息 4 更新
0060	07	33	16	03	UKA_{DF01}	预留信息 4 更新
0061	07	34	16	01	UKB_{DF01}	预留信息 5 更新
0062	07	34	16	02	UKB_{DF01}	预留信息 5 更新
0063	07	34	16	03	UKB_{DF01}	预留信息 5 更新
0064	07	35	16	01	$RK1_{DF02}$	就业与失业信息读取
0065	07	35	16	02	$RK1_{DF02}$	就业与失业信息读取
0066	07	35	16	03	$RK1_{DF02}$	就业与失业信息读取
0067	07	36	16	01	$UK6_{DF02}$	就业扶持政策享受信息更新
0068	07	36	16	02	$UK6_{DF02}$	就业扶持政策享受信息更新
0069	07	36	16	03	$UK6_{DF02}$	就业扶持政策享受信息更新
006A	07	37	16	01	$UK4_{DF03}$	工伤保险信息更新
006B	07	37	16	02	$UK4_{DF03}$	工伤保险信息更新

006C	07	37	16	03	$UK4_{DF03}$	工伤保险信息更新
006D	07	38	16	01	$UK5_{DF03}$	生育保险信息更新
006E	07	38	16	02	$UK5_{DF03}$	生育保险信息更新
006F	07	38	16	03	$UK5_{DF03}$	生育保险信息更新
0070	07	39	16	01	$UK6_{DF03}$	工伤认定信息更新
0071	07	39	16	02	$UK6_{DF03}$	工伤认定信息更新
0072	07	39	16	03	$UK6_{DF03}$	工伤认定信息更新
0073	07	3A	16	01	$UK7_{DF03}$	供养亲属信息更新
0074	07	3A	16	02	$UK7_{DF03}$	供养亲属信息更新
0075	07	3A	16	03	$UK7_{DF03}$	供养亲属信息更新
0076	07	3B	16	01	$UK8_{DF03}$	参保凭证信息更新
0077	07	3B	16	02	$UK8_{DF03}$	参保凭证信息更新
0078	07	3B	16	03	$UK8_{DF03}$	参保凭证信息更新
0079	08	3C	08	01	STK_{DF07}	人事与人才应用维护
007A	07	3D	16	01	$RK1_{DF07}$	荣誉信息读取
007B	07	3D	16	02	$RK1_{DF07}$	荣誉信息读取
007C	07	3D	16	03	$RK1_{DF07}$	荣誉信息读取
007D	07	3E	16	01	$UK1_{DF07}$	荣誉信息更新
007E	07	3E	16	02	$UK1_{DF07}$	荣誉信息更新
007F	07	3E	16	03	$UK1_{DF07}$	荣誉信息更新
0080	07	3F	16	01	$RK2_{DF07}$	专家信息读取
0081	07	3F	16	02	$RK2_{DF07}$	专家信息读取
0082	07	3F	16	03	$RK2_{DF07}$	专家信息读取
0083	07	B0	16	01	$UK2_{DF07}$	专家信息更新
0084	07	B0	16	02	$UK2_{DF07}$	专家信息更新
0085	07	B0	16	03	$UK2_{DF07}$	专家信息更新
0086	07	B1	16	01	$RK3_{DF07}$	军队转业干部信息读取
0087	07	B1	16	02	$RK3_{DF07}$	军队转业干部信息读取
0088	07	B1	16	03	$RK3_{DF07}$	军队转业干部信息读取
0089	07	B2	16	01	$UK3_{DF07}$	军队转业干部信息更新
008A	07	B2	16	02	$UK3_{DF07}$	军队转业干部信息更新
008B	07	B2	16	03	$UK3_{DF07}$	军队转业干部信息更新
008C	08	B3	08	01	STK_{DF05}	生命与健康应用维护
008D	07	B4	16	01	$UK1_{DF05}$	生命与健康信息更新
008E	07	B4	16	02	$UK1_{DF05}$	生命与健康信息更新
008F	07	B4	16	03	$UK1_{DF05}$	生命与健康信息更新
0170	07	08	16	01	IRK	应用提供者（内部认证）
0171	07	08	16	02	IRK	应用提供者（内部认证）
0172	07	08	16	03	IRK	应用提供者（内部认证）
0173	08	09	08	01	STK_{SSSE}	发卡方应用维护
0174	08	20	08	01	STK_{DF01}	公共应用维护
0175	08	23	08	01	STK_{DF04}	社会保险 2 应用维护
0176	07	24	16	01	$UK3_{DF01}$	婚姻状况信息更新

0177	07	24	16	02	$UK3_{DF01}$	婚姻状况信息更新
0178	07	24	16	03	$UK3_{DF01}$	婚姻状况信息更新
0179	07	25	16	01	$UK4_{DF01}$	通讯信息更新
017A	07	25	16	02	$UK4_{DF01}$	通讯信息更新
017B	07	25	16	03	$UK4_{DF01}$	通讯信息更新
017C	07	2A	16	01	$UK2_{DF04}$	医疗保险临时脱网结算信息更新
017D	07	2A	16	02	$UK2_{DF04}$	医疗保险临时脱网结算信息更新
017E	07	2A	16	03	$UK2_{DF04}$	医疗保险临时脱网结算信息更新
017F	02	01	16	01	DPK	医疗保险医疗费用结算交易 版本 1
0180	02	01	16	02	DPK	医疗保险医疗费用结算交易 版本 1
0181	02	01	16	03	DPK	医疗保险医疗费用结算交易 版本 1
0182	07	2D	16	01	$RK1_{DF04}$	医疗保险和医疗费用结算信息读取
0183	07	2D	16	02	$RK1_{DF04}$	医疗保险和医疗费用结算信息读取
0184	07	2D	16	03	$RK1_{DF04}$	医疗保险和医疗费用结算信息读取
0185	08	21	08	01	STK_{DF02}	就业与失业应用维护
0186	08	22	08	01	STK_{DF03}	社会保险 1 应用维护
0187	07	26	16	01	$UK1_{DF02}$	职业和专业技能信息更新
0188	07	26	16	02	$UK1_{DF02}$	职业和专业技能信息更新
0189	07	26	16	03	$UK1_{DF02}$	职业和专业技能信息更新
018A	07	27	16	01	$UK4_{DF02}$	就业创业证信息更新
018B	07	27	16	02	$UK4_{DF02}$	就业创业证信息更新
018C	07	27	16	03	$UK4_{DF02}$	就业创业证信息更新
018D	07	28	16	01	$UK5_{DF02}$	就业援助对象认定信息更新
018E	07	28	16	02	$UK5_{DF02}$	就业援助对象认定信息更新
018F	07	28	16	03	$UK5_{DF02}$	就业援助对象认定信息更新
0190	07	29	16	01	$UK2_{DF03}$	劳动能力鉴定信息更新
0191	07	29	16	02	$UK2_{DF03}$	劳动能力鉴定信息更新
0192	07	29	16	03	$UK2_{DF03}$	劳动能力鉴定信息更新
0193	07	2B	16	01	$RK1_{DF03}$	养老、工伤、生育保险信息读取
0194	07	2B	16	02	$RK1_{DF03}$	养老、工伤、生育保险信息读取
0195	07	2B	16	03	$RK1_{DF03}$	养老、工伤、生育保险信息读取
0196	07	2C	16	01	$RK2_{DF03}$	失业保险信息读取
0197	07	2C	16	02	$RK2_{DF03}$	失业保险信息读取
0198	07	2C	16	03	$RK2_{DF03}$	失业保险信息读取
0199	02	02	16	01	DPK	医疗保险医疗费用结算交易 版本 2
019A	02	02	16	02	DPK	医疗保险医疗费用结算交易 版本 2
019B	02	02	16	03	DPK	医疗保险医疗费用结算交易 版本 2
019C	02	03	16	01	DPK	医疗保险医疗费用结算交易 版本 3
019D	02	03	16	02	DPK	医疗保险医疗费用结算交易 版本 3
019E	02	03	16	03	DPK	医疗保险医疗费用结算交易 版本 3

B.2 省级密钥逻辑地址

省级密钥逻辑地址见表 B.2。

表 B.2 省级密钥逻辑地址

地址	类别	标识	长度	组数	名称	适用的应用范围
0090	08	10	16	01	PUK	发卡方（PIN 解锁）
0091	08	11	16	01	BK	发卡方（卡锁定）
0092	08	40	16	01	LK_{DF03}	社会保险 1 应用（应用锁定）
0093	08	41	16	01	LK_{DF04}	社会保险 2 应用（应用锁定）
0094	08	12	16	01	UK_{SSSE}	发卡方和个人基本信息更新
0095	08	12	16	02	UK_{SSSE}	发卡方和个人基本信息更新
0096	08	12	16	03	UK_{SSSE}	发卡方和个人基本信息更新
0097	08	42	16	01	$UK1_{DF01}$	户籍信息更新
0098	08	42	16	02	$UK1_{DF01}$	户籍信息更新
0099	08	42	16	03	$UK1_{DF01}$	户籍信息更新
009A	08	43	16	01	$UK2_{DF01}$	个人状况信息更新
009B	08	43	16	02	$UK2_{DF01}$	个人状况信息更新
009C	08	43	16	03	$UK2_{DF01}$	个人状况信息更新
009D	08	44	16	01	$UK5_{DF01}$	国家/地区及政治面貌信息更新
009E	08	44	16	02	$UK5_{DF01}$	国家/地区及政治面貌信息更新
009F	08	44	16	03	$UK5_{DF01}$	国家/地区及政治面貌信息更新
00A0	08	45	16	01	$UK2_{DF02}$	就业状况信息更新
00A1	08	45	16	02	$UK2_{DF02}$	就业状况信息更新
00A2	08	45	16	03	$UK2_{DF02}$	就业状况信息更新
00A3	08	46	16	01	$UK3_{DF02}$	就业记录更新
00A4	08	46	16	02	$UK3_{DF02}$	就业记录更新
00A5	08	46	16	03	$UK3_{DF02}$	就业记录更新
00A6	08	47	16	01	$UK1_{DF03}$	失业保险信息更新
00A7	08	47	16	02	$UK1_{DF03}$	失业保险信息更新
00A8	08	47	16	03	$UK1_{DF03}$	失业保险信息更新
00A9	08	48	16	01	$UK3_{DF03}$	养老保险信息更新
00AA	08	48	16	02	$UK3_{DF03}$	养老保险信息更新
00AB	08	48	16	03	$UK3_{DF03}$	养老保险信息更新
00AC	08	49	16	01	$UK1_{DF04}$	医疗、工伤、生育保险基本信息更新
00AD	08	49	16	02	$UK1_{DF04}$	医疗、工伤、生育保险基本信息更新
00AE	08	49	16	03	$UK1_{DF04}$	医疗、工伤、生育保险基本信息更新
00AF	08	4A	16	01	DSK	更新年度起始日期
00B0	08	4A	16	02	DSK	更新年度起始日期
00B1	08	4A	16	03	DSK	更新年度起始日期
00B2	08	4B	16	01	DLK	医疗保险账户划入交易
00B3	08	4B	16	02	DLK	医疗保险账户划入交易
00B4	08	4B	16	03	DLK	医疗保险账户划入交易
00B5	08	4C	16	01	DTK	医疗保险交易

00B6	08	4C	16	02	DTK	医疗保险交易
00B7	08	4C	16	03	DTK	医疗保险交易

附录 C

业务系统调用读写终端接口流程

本附录给出业务系统调用读写终端接口的基本流程，有几点说明如下：

（1）所有操作的第一步均应调用 “读基本信息”函数进行读基本信息操作，业务系统如果判断到五种情况的错误代码（详见 10.5），则继续调用“基于加密机的读基本信息”函数进行读基本信息操作。

（2）业务系统应缓存读取的基本信息留作后用。

（3）在通用读卡和通用写卡接口函数中，业务系统应根据不同“规范版本”的文件结构（详见 10.2）确定读写卡调用的函数和输入参数。

C.1 读基本信息

读基本信息流程如图 C.1 所示。

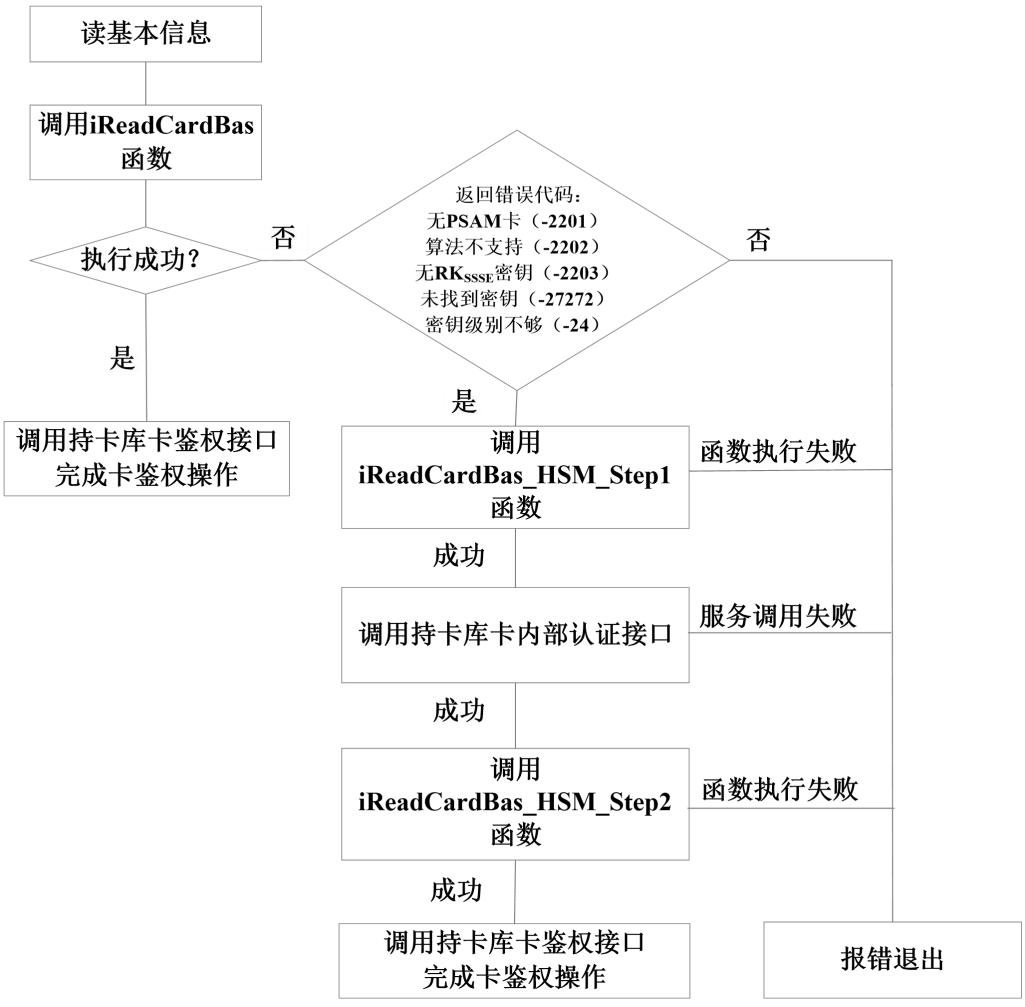

图 C.1 读基本信息流程

具体流程如下：

（1）调用 iReadCardBas 函数，判断函数执行结果，若执行成功，则返回基本信息，调用持卡库的卡鉴权服务接口，完成卡鉴权操作；若执行失败，返回错误代码并进行判断，当判断到五种情况的错误代码（详见 10.5）后，则调用基于加密机的 iReadCardBas_HSM_Step1 函数，否则报错退出；

（2）若 iReadCardBas_HSM_Step1 函数执行成功，则分析返回数据内容，组织持卡库报文，调用持卡库的卡内部认证服务接口，否则报错退出；

（3）若卡内部认证服务接口调用成功，则调用基于加密机的 iReadCardBas_HSM_Step2 函数，否则报错退出；

（4）若 iReadCardBas_HSM_Step2 函数执行成功，则返回基本信息，调用持卡库的卡鉴权服务接口，完成卡鉴权操作，否则报错退出。

C.2 通用读卡

通用读卡流程如图 C.2 所示。

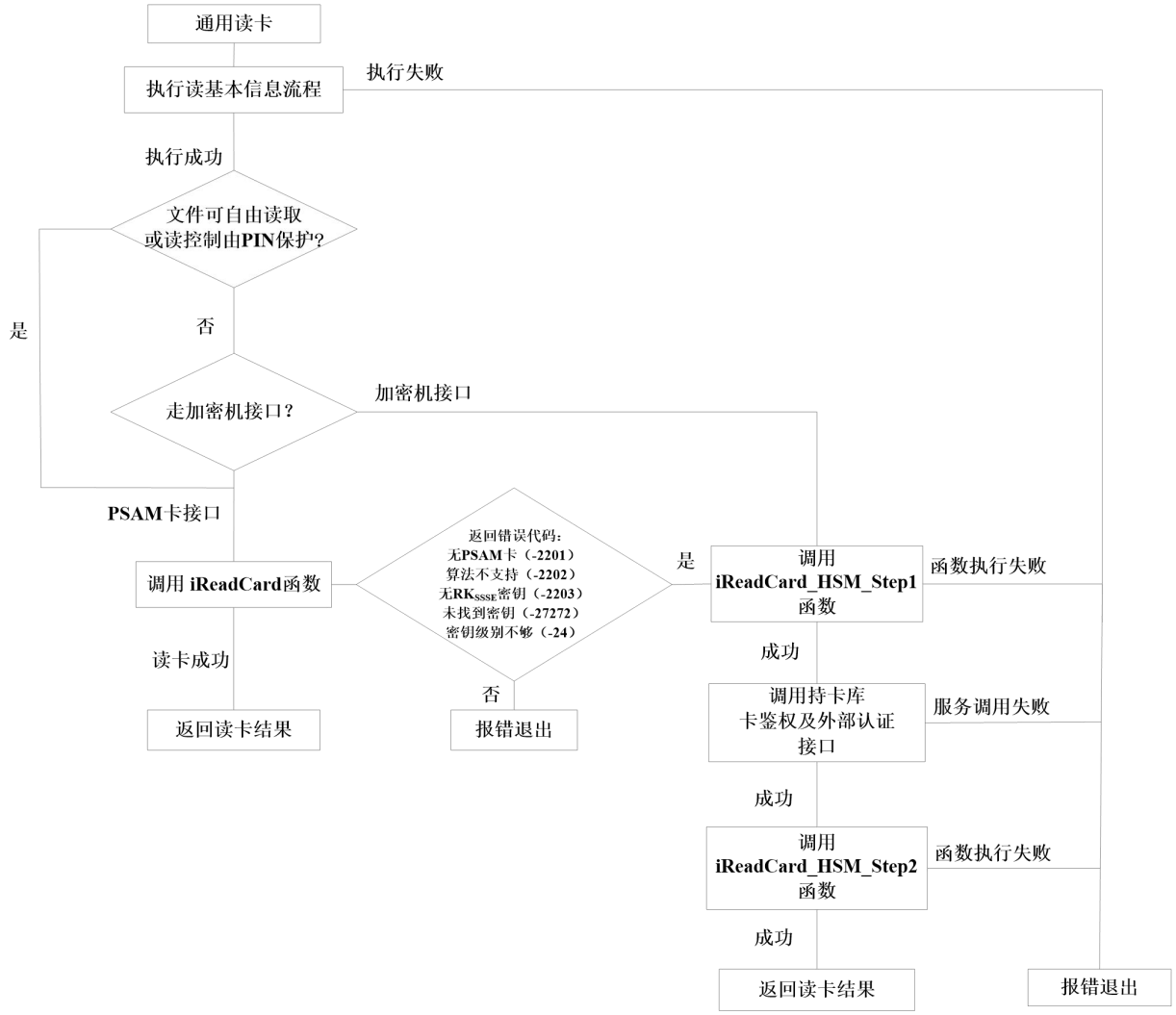

图 C.2 通用读卡流程

具体流程如下：

（1）执行读基本信息流程，若执行失败，则报错退出；若执行成功，则判断文件读控制权限；

（2）若文件可自由读取或读控制由 PIN 保护，则走 PSAM 卡接口；若不是，判断是否走加密机接口；

（3）若走 PSAM 卡接口，则调用 iReadCard 函数，判断函数执行结果；若执行成功，则读出所需数据，返回读卡结果；若执行失败，返回错误代码并进行判断，当判断到五种情况的错误代码（详见 10.5）后，则调用基于加密机的 iReadCard_HSM_Step1 函数，否则其他错误代码时，则报错退出；

（4）若 iReadCard_HSM_Step1 函数执行成功，则分析返回数据内容，组织持卡库报文，调用持卡库的卡鉴权及外部认证服务接口；否则报错退出，若此函数返回错误代码“-2204”（具体参见附录 A）且所读文件可自由读取或读控制仅受 PIN 保护，则业务系统继续调用 iReadCard 函数进行通用读卡操作；

（5）若卡鉴权及外部认证服务接口调用成功，则调用基于加密机的 iReadCard_HSM_Step2 函数，否则报错退出；

（6）若 iReadCard_HSM_Step2 函数执行成功，则读出所需数据，返回读卡结果，否则报错退出。

C.3 通用写卡

通用写卡流程如图 C.3 所示。

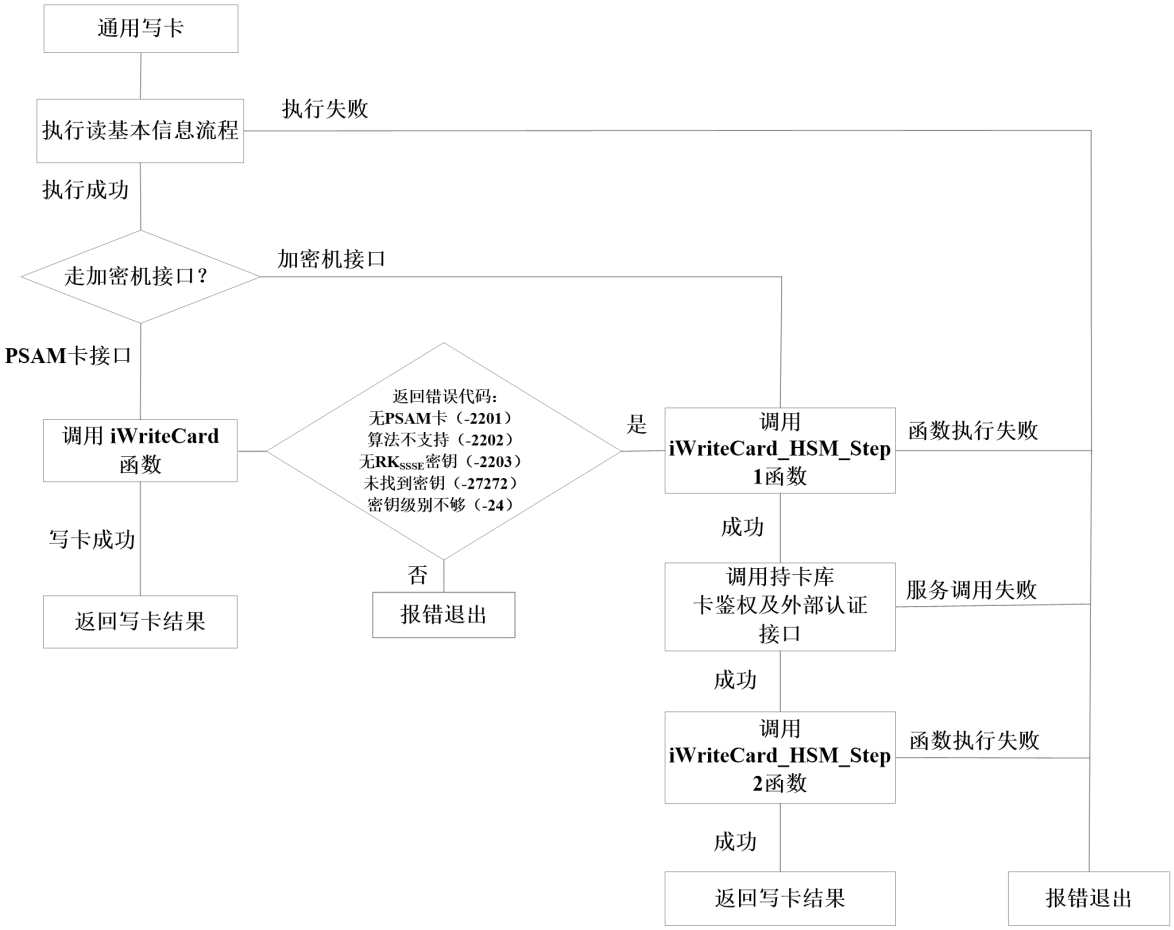

图 C.3 通用写卡流程

具体流程如下：

（1）执行读基本信息流程，若执行失败，则报错退出；若执行成功，则判断是否走加密机接口；

（2）若走 PSAM 卡接口，则调用 iWriteCard 函数，判断函数执行结果；若执行成功，则写入数据，返回写卡结果；若执行失败，返回错误代码并进行判断，当判断到五种情况的错误代码（详见 10.5）后，则调用基于加密机的 iWriteCard_HSM_Step1 函数，否则其他错误代码时，则报错退出；

（3）若 iWriteCard_HSM_Step1 函数执行成功，则分析返回数据内容，组织持卡库报文，调用持卡库的卡鉴权及外部认证服务接口，否则报错退出；

（4）若卡鉴权及外部认证服务接口调用成功，则调用基于加密机的 iWriteCard_HSM_Step2 函数，否则报错退出；

（5）若 iWriteCard_HSM_Step2 函数执行成功，则写入数据，返回写卡结果，否则报错退出。

C.4 PIN 重置

PIN 重置流程如图 C.4 所示。

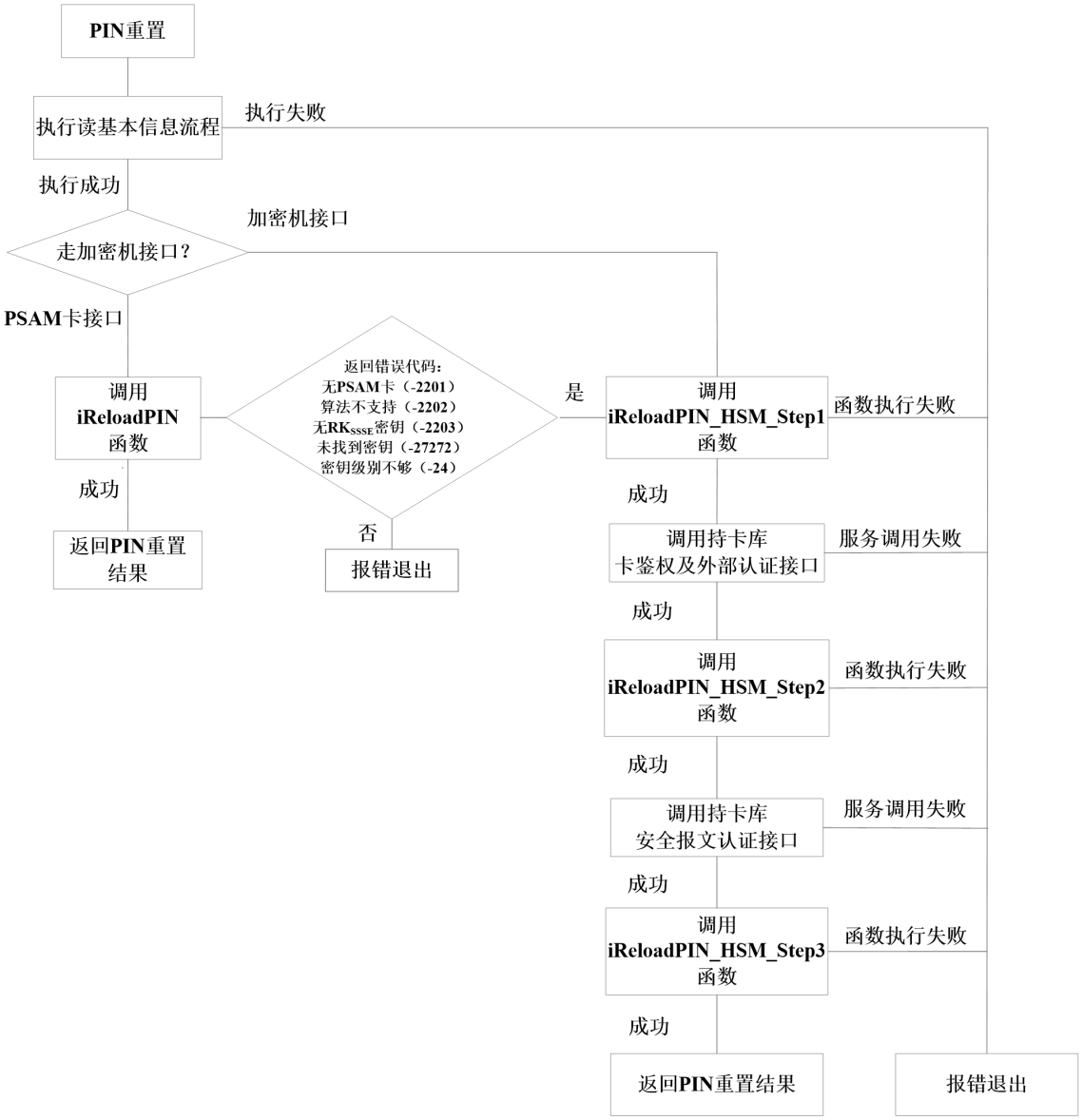

图 C.4 PIN 重置流程

具体流程如下：

（1）执行读基本信息流程，若执行失败，则报错退出；若执行成功，则判断是否走加密机接口；

（2）若走 PSAM 卡接口，则调用 iReloadPIN 函数，判断函数执行结果；若执行成功，则进行 PIN 重置；若执行失败，返回错误代码并进行判断，当判断到五种情况的错误代码（详见 10.5）后，则调用基于加密机的 iReloadPIN_HSM_Step1 函数，否则其他错误代码时，则报错退出；

（3）若 iReloadPIN_HSM_Step1 函数执行成功，则分析返回数据内容，组织持卡库报文，调用持卡库的卡鉴权及外部认证服务接口，否则报错退出；

（4）若卡鉴权及外部认证服务接口调用成功，则调用基于加密机的 iReloadPIN_HSM_Step2 函数，否则报错退出；

（5）若 iReloadPIN_HSM_Step2 函数执行成功，则分析返回数据内容，组织持卡库报文，调用持卡库的安全报文认证服务接口，否则报错退出；

（6）若安全报文认证服务接口调用成功，则调用基于加密机的 iReloadPIN_HSM_Step3 函数，否则报错退出；

（7）若 iReloadPIN_HSM_Step3 函数执行成功，则进行 PIN 重置，否则报错退出。

C.5 PIN 解锁

PIN 解锁流程如图 C.5 所示。

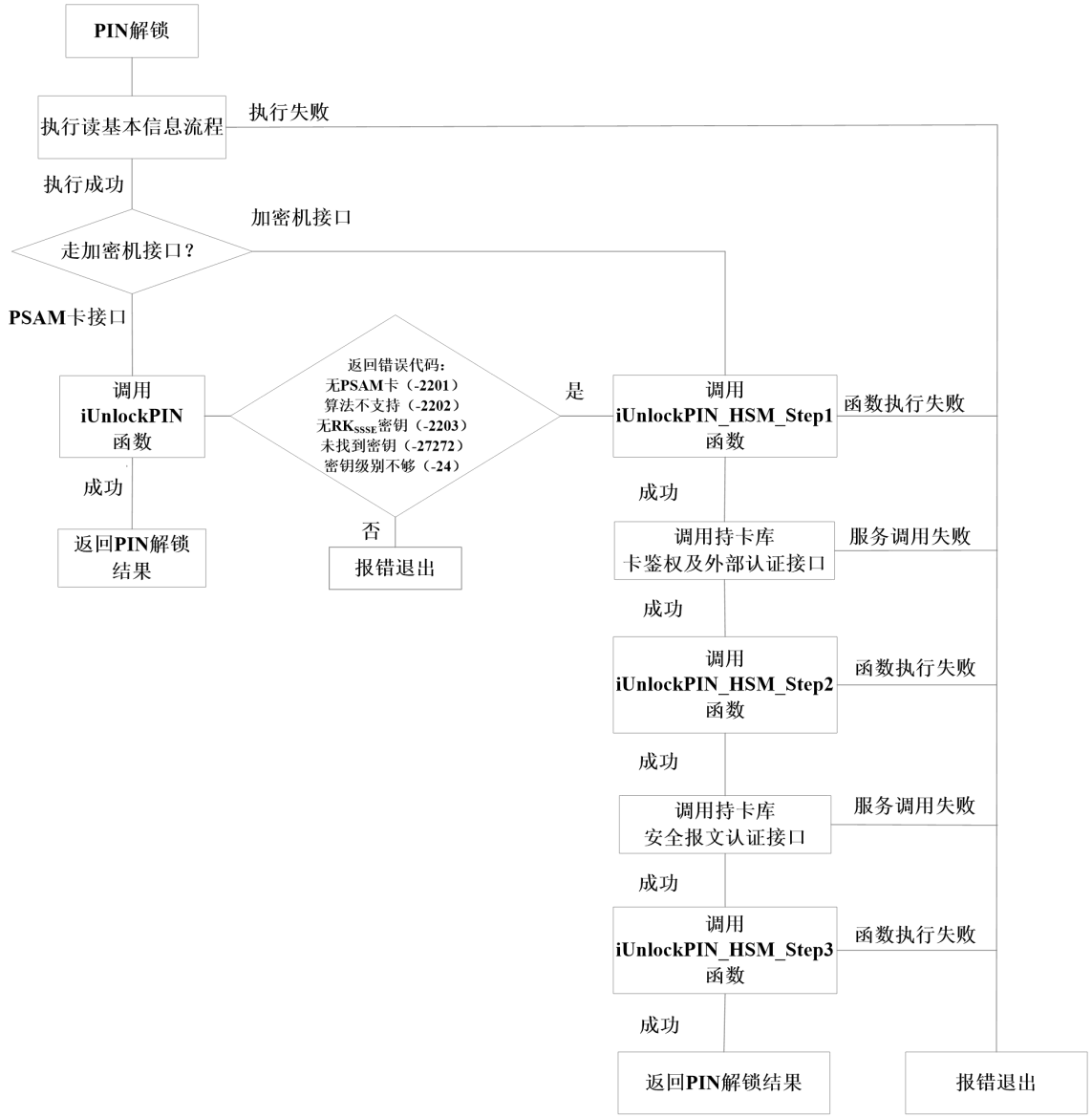

图 C.5 PIN 解锁流程

具体流程如下：

（1）执行读基本信息流程，若执行失败，则报错退出；若执行成功，则判断是否走加密机接口；

（2）若走 PSAM 卡接口，则调用 iUnlockPIN 函数，判断函数执行结果；若执行成功，则进行 PIN 解锁；若执行失败，返回错误代码并进行判断，当判断到五种情况的错误代码（详见 10.5）后，则调用基于加密机的 iUnlockPIN_HSM_Step1 函数，否则其他错误代码时，则报错退出；

（3）若 iUnlockPIN_HSM_Step1 函数执行成功，则分析返回数据内容，组织持卡库报文，调用持卡库的卡鉴权及外部认证服务接口，否则报错退出；

（4）若卡鉴权及外部认证服务接口调用成功，则调用基于加密机的 iUnlockPIN_HSM_Step2 函数，否则报错退出；

（5）若 iUnlockPIN_HSM_Step2 函数执行成功，则分析返回数据内容，组织持卡库报文，调用持卡库的安全报文认证服务接口，否则报错退出；

（6）若安全报文认证服务接口调用成功，则调用基于加密机的 iUnlockPIN_HSM_Step3 函数，否则报错退出；

（7）若 iUnlockPIN_HSM_Step3 函数执行成功，则进行 PIN 解锁，否则报错退出。

C.6 消费交易

消费交易流程如图 C.6 所示。

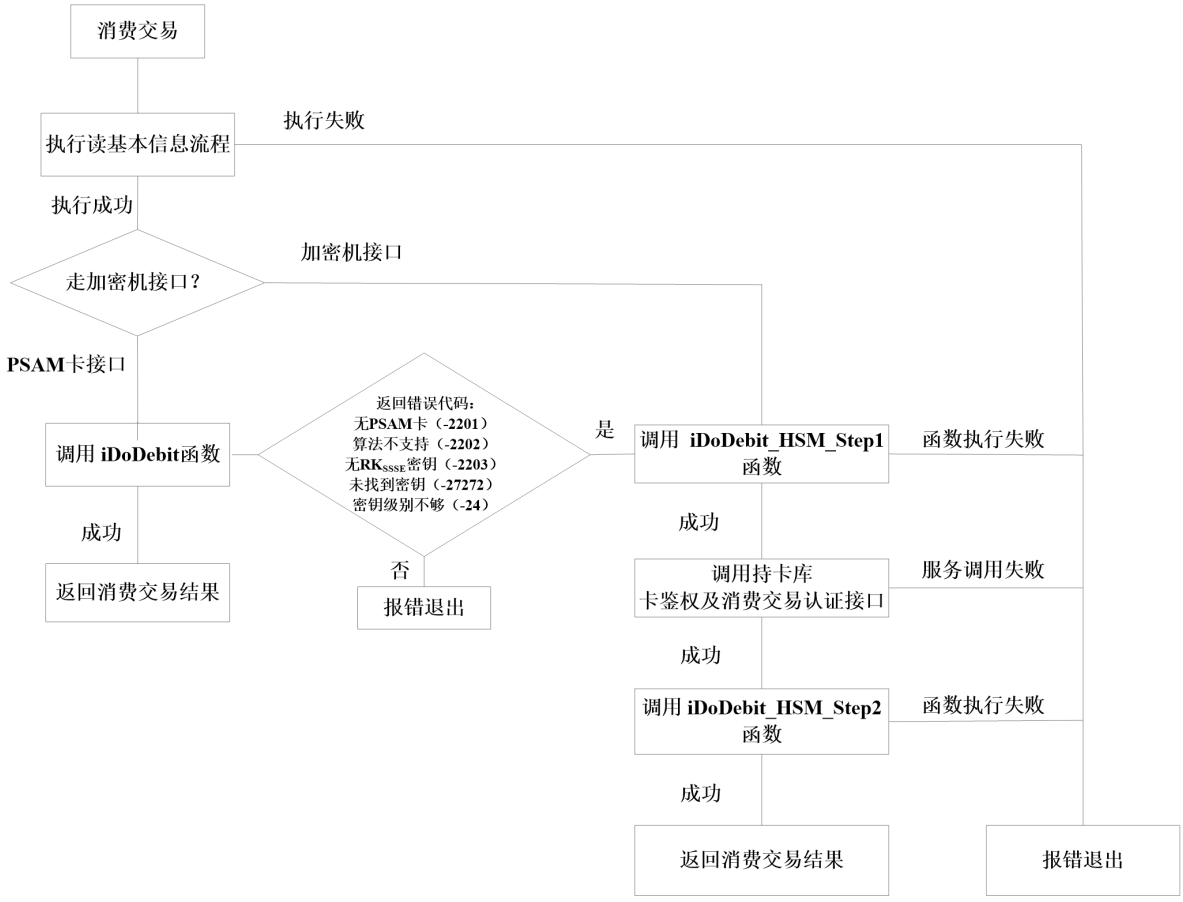

图 C.6 消费交易流程

具体流程如下：

（1）执行读基本信息流程，若执行失败，则报错退出；若执行成功，则判断是否走加密机接口；

（2）若走 PSAM 卡接口，则调用 iDoDebit 函数，判断函数执行结果；若执行成功，则进行消费交易，返回消费交易结果；若执行失败，返回错误代码并进行判断，当判断到五种情况的等错误代码（详见 10.5）后，则调用基于加密机的 iDoDebit_HSM_Step1 函数，否则其他错误代码时，则报错退出；

（3）若 iDoDebit_HSM_Step1 函数执行成功，则分析返回数据内容，组织持卡库报文，调用持卡库的卡鉴权及消费交易认证服务接口，否则报错退出；

（4）若卡鉴权及消费交易认证服务接口调用成功，则调用基于加密机的 iDoDebit_HSM_Step2 函数，否则报错退出；

（5）若 iDoDebit_HSM_Step2 函数执行成功，则进行消费交易，返回消费交易结果，否则报错退出。

C.7 消费交易结算验证

消费交易结算验证流程如图 C.7 所示。

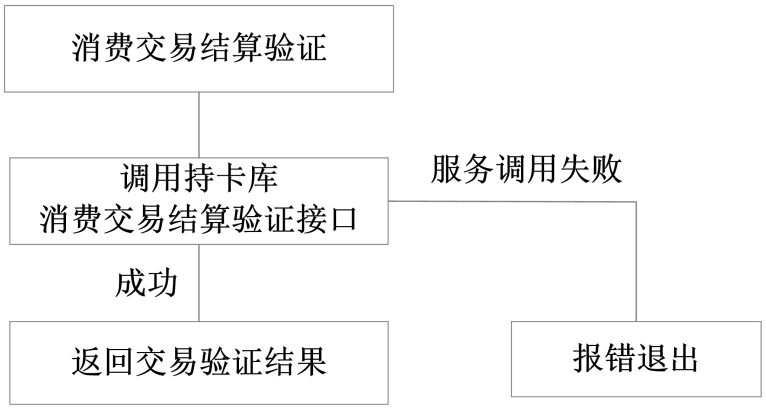

图 C.7 消费交易结算验证流程

具体流程如下：

根据保存的交易记录，组织持卡库报文，调用持卡库的消费交易结算验证服务接口，若调用成功，则取得返回结果，并根据返回结果判定是否交易验证成功，否则报错退出。