ICS 03.220.20
R 80

团 体 标 准

T/ITS XXXX-XXXX

智能交通运输系统车载安全密码设备应用接口规范

（征求意见稿）

（本草案完成时间：2021 年 8 月）

在提交反馈意见时，请将您知道的相关专利连同支持性文件一并附上。

发布 实施

中 国 智 能 交 通 产 业 联 盟 发 布

目　　次

前　言

本文件规定了车载安全密码设备应用接口的功能要求、接口规范要求、以及设备安全要求。本文件的目的是为智能交通运输系统体系下的车载安全密码设备制定统一的应用接口规范，通过该接口调用密码设备提供基础密码服务。

本文件参考了 PKI 体系的国密规范《GB/T 0018-2012 公钥密码基础设施应用技术体系 密码设备应用接口规范》和《GM/T 0016-2012 智能密码钥匙 密码应用接口规范》。

智能交通运输系统车载安全密码设备应用接口规范

1 范围

本文件规定了车载安全密码设备应用接口的功能要求、接口规范要求、以及设备安全要求。本文件的目的是为智能交通运输系统体系下的车载安全密码设备制定统一的应用接口规范，通过该接口调用密码设备提供基础密码服务。

本文件适用于车载安全密码设备的研制、使用和检测。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中注日期的引用文件，仅该日期对应的版本适用于本文件；不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GM/T 0016-2012 智能密码钥匙 密码应用接口规范

GB/T 0018-2012 公钥密码基础设施应用技术体系 密码设备应用接口规范

3 术语和定义、缩略语

3.1 术语和定义

下列术语和定义适用于本文件。

3.1.1

算法标识 arithmetic identifier

用于标明算法机制的数字化信息。

3.1.2

非对称密码算法/公钥密码算法 asymmetric cryptographic algorithm/public key cryptographic algorithm

加解密使用不同密钥的密码算法。其中一个密钥（公钥）可以公开，另一个密钥（私钥）必须保密，且由公钥求解私钥是计算不可行的。

3.1.3

解密 decipherment/decryption

加密过程对应的逆过程。

3.1.4

设备密钥 device key pair

存储在设备内部的用于设备管理的非对称密钥对，包含签名密钥对和加密密钥对。

3.1.5

加密 encipherment/encryption

对数据进行密码变换以产生密文的过程。

3.1.6

密钥加密密钥 key-encrypting key

又称二级密钥(Secondary Key)或密钥传送密钥(key Transport key)，用于对密钥进行加解密。

4 结构模型

典型的密码应用从云端到车端的调用模型见图 1。

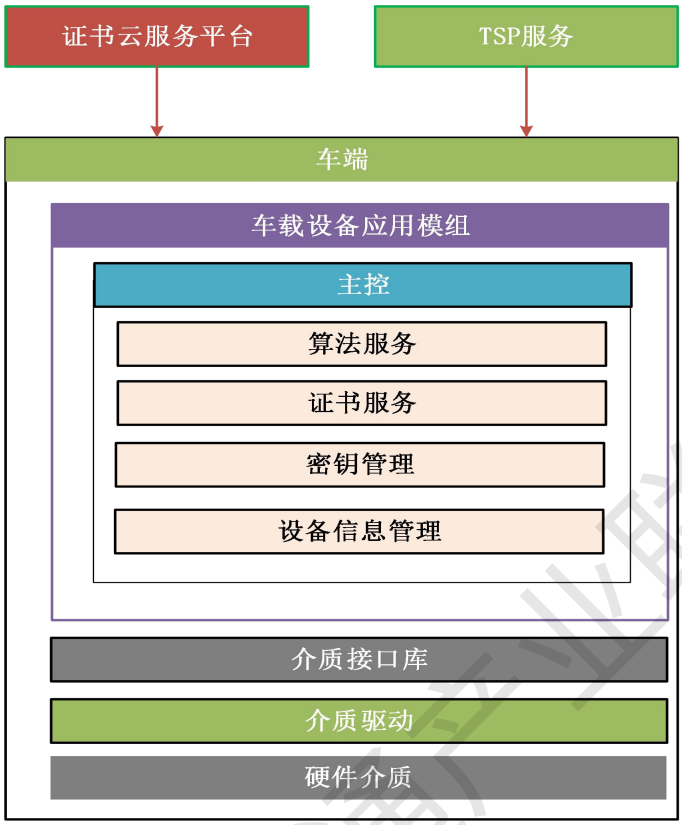

图 1　密码设备应用调用模型

5　数据类型定义

5.1 算法标识

5.1.1 分组密码算法标识

分组密码算法标识包含密码算法的类型和加密模式。

分组密码算法标识的编码规则为：从低位到高位，第 0 位到第 7 位按位表示分组密码算法工作模式，第 8 位到第 31 位按位表示分组密码算法类型，分组密码算法的标识如表 1 所示。

表 1 分组密码算法标识表

标签	标识符	描述
SGD_SM4_ECB	0x00000401	SM4 算法 ECB 加密模式
SGD_SM4_CBC	0x00000402	SM4 算法 CBC 加密模式
SGD_SM4_CFB	0x00000404	SM4 算法 CFB 加密模式
SGD_SM4_OFB	0x00000408	SM4 算法 OFB 加密模式

表 1 （续）

表 1（第 2 页/共 2 页）		
SGD_SM4_MAC	0x00000410	SM4 算法 MAC 运算
SGD_SM4_CCM	0x00000416	SM4 算法 CCM 加密模式
AES_CCM	0x00002016	AES 算法 CCM 加密模式

5.1.2 非对称算法标识

非对称密码算法标识仅定义了密码算法的类型，在使用非对称算法进行数字签名运算时，可将非对称密码算法标识符与密码杂凑算法标识符进行“或”运算后使用，如“RSA with SHA1”可表示为 SGD_RSA | SGD_SHA1，即 0x00010002，“|”表示“或”运算。

非对称密码算法标识的编码规则为：从低位到高位，第 0 位到第 7 位为 0，第 8 位到第 15 位按位表示非对称密码算法的算法协议，如果所表示的非对称算法没有相应的算法协议则为 0，第 16 位到第 31 位按位表示非对称密码算法类型，非对称密码算法的标识如表 2 所示。

表 2 非对称密码算法标识表

标签	标识符	描述
SGD_RSA	0x00010000	RSA 算法
SGD_SM2_1	0x00020100	椭圆曲线签名算法
SGD_SM2_2	0x00020200	椭圆曲线密钥交换协议
SGD_SM2_3	0x00020400	椭圆曲线加密算法
0x00000400-0x800000xx		为其他非对称密码算法预留

5.1.3 杂凑算法标识

密码杂凑算法标识符可以在进行密码杂凑运算或计算 MAC 时应用，也可以与非对称密码算法标识符进行“或”运算后使用，表示签名运算前对数据进行密码杂凑运算的算法类型。

密码杂凑算法标识的编码规则为：从低位到高位，第 0 位到第 7 位表示密码杂凑算法，第 8 位到第 31 位为 0，密码杂凑算法的标识如表 3 所示。

表 3 杂凑算法标识表

标签	标识符	描述
SGD_SM3	0x00000001	SM3 杂凑算法
SGD_SHA1	0x00000002	SHA1 杂凑算法
SGD_SHA256	0x00000004	SHA256 杂凑算法
SGD_SHA384	0x00000008	SHA384 杂凑算法
0x0000010-0x000000FF		为其杂凑算法预留

5.2 基本数据类型

这里对安全密码设备中用到的数据类型做统一的定义描述

表 4 数据类型标签表

类型名称	描述	定义
INT8	有符号 8 位整数	
INT16	有符号 16 位整数	
INT32	有符号 32 位整数	
UINT8	无符号 8 位整数	
UINT16	无符号 16 位整数	
UINT32	无符号 32 位整数	
BOOL	布尔类型，取值为 TRUE 或 FALSE	
BYTE	字节类型，无符号 8 位整数	typedef UINT8 BYTE
CHAR	字符类型，无符号 8 位整数	typedef UINT8 CHAR
SHORT	短整数，有符号 16 位	typedef INT16 SHORT
USHORT	无符号 16 位整数	typedef UINT16 USHORT
LONG	长整数，有符号 32 位整数	typedef INT32 LONG
ULONG	长整数，无符号 32 位整数	typedef UINT32 ULONG
UINT	无符号 32 位整数	typedef UINT32 UINT
WORD	字类型，无符号 16 位整数	typedef UINT16 WORD

表 4 （续）

表 4 （第 2 页/共 2 页）		
DWORD	双字类型，无符号 32 位整数	typedef UINT32 DWORD
FLAGS	标志类型，无符号 32 位整数	typedef UINT32 FLAGS
LPSTR	8 位字符串指针，按照 UTF8 格式存储及交换	typedef CHAR * LPSTR
HANDLE	句柄，指向任意数据对象的起始地址	typedef void * HANDLE
DEVHANDLE	设备句柄	typedef HANDLE DEVHANDLE
HAPPLICATION	应用句柄	typedef HANDLE HAPPLICATION
HCONTAINER	容器句柄	typedef HANDLE HCONTAINER
SCHAR	字符类型，有符号 8 位字符	
UCHAR	字符类型，无符号 8 位整数	

5.3 常量定义

数据常量标识定义了在规范中用到的常量的取值。

表 5-1 常量定义

常量名	取值	描述
TRUE	0x00000001	布尔值为真
FLASE	0x00000000	布尔值为假

表 5-2 常量定义（扩展）

常量名	取值	描述
DEVAPI	__stdcall	__stdcall 函数调用方式
ADMIN_TYPE	0	管理员 PIN 类型
USER_TYPE	1	用户 PIN 类型

5.4 复合数据类型

定义了在规范中用到的比较复杂的数据类型，如：版本信息、设备信息、公钥数据结构、私钥数据

结构、非对称密文数据结构、签名数据结构、加密密钥对保护数据结构、密钥衍生相关数据结构等

5.4.1 版本信息

类型定义：

```
typedef struct
{
    SCHAR ucHwVersion[32];
    SCHAR ucFwVersion[32];
    SCHAR ucSwVersion[32];
}Version_t;
```

数据项描述参见表 6

表 6 版本信息描述

数据项	类型	含义	备注
ucHwVersion	SCHAR 数组	密码设备硬件版本号	以'\0'为结束符的 ASCII 字符串
ucFwVersion	SCHAR 数组	密码设备固件版本号	以'\0'为结束符的 ASCII 字符串
ucSwVersion	SCHAR 数组	密码设备应用接口版本号	以'\0'为结束符的 ASCII 字符串

5.4.2 设备信息

类型定义：

```
typedef struct
{
    Version_t Version;
    SCHAR scManufacturer[64];
    SCHAR scIssuer[64];
    SCHAR scLable[32];
    SCHAR scSerialNumber[32];
    BYTE  byReserved[64];
}DevInfo_t;
```

数据项描述参见表 7

表 7 设备信息描述

数据项	类型	含义	备注
Version	Version_t	版本号	
scManufacturer	SCHAR 数组	设备厂商信息	以'\0'为结束符的 ASCII 字符串
scIssuer	SCHAR 数组	发行厂商信息	以'\0'为结束符的 ASCII 字符串
scLable	SCHAR 数组	设备标签	以'\0'为结束符的 ASCII 字符串
scSerialNumber	SCHAR 数组	序列号	以'\0'为结束符的 ASCII 字符串
byReserved	BYTE 数组	保留扩展	

5.4.3 RSA 公钥数据结构

类型定义：

```
typedef struct
{
    UINT32 u32BitLen;
    UCHAR  ucModule[256];
    UINT32 u32PublicExPonent;
}RsaPublicKey_t;
```

数据项描述参见表 8

表 8 RSA 公钥结构信息描述

数据项	类型	含义	备注
u32BitLen	UINT32	公钥模长 bit	必须是 8 的倍数
ucModule	UCHAR 数组	模数 n = p*q	实际长度为 u32BitLen/8 字节
u32PublicExPonent	UINT32	公开密钥 e	一般为 0x00010001

5.4.4 RSA 私钥数据结构

类型定义：

```
typedef struct
{
    UINT32 u32BitLen;
    UCHAR   ucPrime1[128];
    UCHAR   ucPrime2[128];

    UCHAR   ucPrime1ExPonent[128];
    UCHAR   ucPrime2ExPonent[128];
    UCHAR   ucCoefficient[128];
}RsaPrivateKey_t;
```

数据项描述参见表 9

表 9 RSA 私钥结构信息描述

数据项	类型	含义	备注
u32BitLen	UINT32	公钥模长 bit	必须是 8 的倍数
ucPrime1	UCHAR 数组	素数因子 p	实际长度为 u32BitLen/8 字节
ucPrime2	UCHAR 数组	素数因子 q	实际长度为 u32BitLen/16 字节
ucPrime1ExPonent	UCHAR 数组	p 的 crt 幂	实际长度为 u32BitLen/16 字节
ucPrime2ExPonent	UCHAR 数组	q 的 crt 幂	实际长度为 u32BitLen/16 字节
ucCoefficient	UCHAR 数组	q 的逆模 p	实际长度为 u32BitLen/16 字节

5.4.5 ECC 公钥数据结构

类型定义：

```
typedef struct
{
    UCHAR   ucPublicKeyX[ECC_XCOORDINATE_LEN];
    UCHAR   ucPublicKeyY[ECC_YCOORDINATE_LEN];
}EccPubKey_t;
```

数据项描述参见表 10

表 10 ECC 公钥结构信息描述

数据项	类型	含义	备注
ucPublicKeyX	UCHAR 数组	公钥 x 分量	#define ECC_XCOORDINATE_LEN 32
ucPublicKeyY	UCHAR 数组	公钥 y 分量	#define ECC_YCOORDINATE_LEN 32

5.4.6 私钥数据结构

类型定义：

```
typedef struct
{
    UCHAR   ucPrivateKey[ECC_PRIVATE_KEY_LEN];
}EccPriKey_t;
```

数据项描述参见表 11

表 11 ECC 私钥结构信息描述

数据项	类型	含义	备注
ucPrivateKey	UCHAR 数组	私钥	#define ECC_PRIVATE_KEY_LEN 32

5.4.7 ECC 签名数据结构

类型定义：

```
typedef struct
{
    UCHAR   ucSignatureR[ECC_XCOORDINATE_LEN];
    UCHAR   ucSignatureS[ECC_XCOORDINATE_LEN];
}EccSignature_t;
```

数据项描述参见表 12

表 12 ECC 签名结构信息描述

数据项	类型	含义	备注

ucSignatureR	UCHAR 数组	签名结果的 r 部分	
ucSignatureS	UCHAR 数组	签名结果的 s 部分	

5. 4. 8 非对称加密密钥对数据结构

类型定义：

ECC 公钥加密只用来加密对称密钥，加密后的密文结构如下：

curve | EccPubKey | 密文 | 明文 HASH(32B)

数据项描述参见表 13

表 13 证书类型描述

数据项	类型	意义	备注
curve		ECC 算法	0-SM2；1-nistP256；2-brainpoolP256r1
EccPubKey	EccPubKey_t	ECC 公钥	长度为 64 字节
密文		明文加密后的密文	长度与明文长度相同
明文 HASH		对明文的 HASH	长度为 32 字节

5.4.9 非对称签名数据结构

类型定义：

```
typedef struct Struct_ECCSIGNATUREBLOB{
    BYTE r[ECC_MAX_XCOORDINATE_BITS_LEN/8];
    BYTE s[ECC_MAX_XCOORDINATE_BITS_LEN/8];
} ECCSIGNATUREBLOB, *PECCSIGNATUREBLOB;
```

注：ECC_MAX_MODULUS_BITS_LEN 为 ECC 算法模数的最大长度

数据项描述参见表 13：

表 14 ECC 签名数据结构

数据项	类型	含义	备注
r	BYTE 数组	签名结果的 r 部分	
s	BYTE 数组	签名结果的 s 部分	

5. 4. 10 证书类型数据结构

类型定义：

```
typedef enum
{
    ITS_智能交通_CA = 1,
    ITS_智能交通_E_CERT,
    ITS_BASE_SM2_CERT,
    ITS_BASE_RSA_CERT,
}ITSCertType_e;
```

数据项描述参见表 14

表 14 证书类型描述

数据项	类型	含义	备注
ITS_智能交通_CA	enum	智能交通 CA 证书	IEEE1609.2 或 GB/T37376 证书
ITS_智能交通_EP_CERT	enum	智能交通车载终端证书	IEEE1609.2 或 GB/T37376 证书
ITS_BASE_SM2_CERT	enum	基础的 SM2 证书	格式为 x.509 证书
ITS_BASE_RSA_CERT	enum	基础的 RSA 证书	格式为 x.509 证书

5.5 容器索引定义

这里设计容器的概念是为了证书以及密钥的统一管理，包括容器的创建、打开、关闭、删除，证书的导入导出等操作。

容器由车载安全密码设备内部创建并管理，每个容器可以存储对称密钥、非对称密钥对、证书等，容器对外由索引号标识，一个索引标识一个容器，表 15 为容器索引定义表的示例。

密码设备可以自行规划密钥、证书等存储的索引范围。

如：CA 证书索引，0-29，代表 CA 证书可存储到 0-29 号的容器内。

表 15 容器索引定义表

容器索引号	非对称密钥（是否设置）	对称密钥（是否设置）	CA 证书（是否设置）	终端证书（是否设置）	……
0	是	是	是	是	……
1	是	是	是	是	……

……	……	……	……	……	……

6 接口函数

6.1 设备信息管理

6.1.1 设备状态查询

函数原型　INT32 ITSGetDevState(UINT32* pu32State)

功能描述　读取密码设备状态

参数　　pu32State　　[OUT] 0-空闲；1-运算中；2-故障

返回值　　0：成功

　　其他：错误码

6.1.2 设备信息查询

函数原型　INT32 ITSGetDevInfo(DevInfo_t *ptDevInfo)

功能描述　读取密码设备信息

参数　　ptDevInfo　　[OUT] 设备信息

返回值　　0：成功

　　其他：错误码

6.2 访问控制

6.2.1 设备认证

函数原型　INT32 ITSDevAuth(UCHAR *pucAuthData, UINT32 u32Len)

功能描述　设备认证是设备对应用程序的认证，认证过程参见 7.2

参数　　pucAuthData　　[IN] 认证数据

　　u32Len　　[IN] 认证数据的长度

返回值　　0：成功

　　其他：错误码

6.2.2 修改设备认证密钥

函数原型　INT32 ITSChangeDevAuthKey(UCHAR *pucKeyValue, UINT32 u32KeyLen)

功能描述　更改设备认证密钥

参数　　pucKeyValue　　[IN] 密钥值

　　u32KeyLen　　[IN] 密钥长度

返回值　　0：成功

　　其他：错误码

备注　　权限要求：设备认证成功后才能使用

6.2.3 校验 PIN

函数原型　INT32 ITSVerifyPin(SCHAR *pscPin, UINT32 u32PinLen, UINT32 *pu32RetryCount)

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 码的剩余重试次数，当剩余次数为 0 时，表示 PIN 码已经锁死

参数　　pucPin　　[IN] PIN 码

　　u32PinLen　　[IN] PIN 码长度

　　pu32RetryCount　[OUT] 出错后返回的重试次数

返回值　　0：成功

　　其他：错误码

6.2.4 修改 PIN

函数原型　INT32 ITSChangePin(UINT32 u32OldPinLen, SCHAR* pscOldPin, UINT32 u32NewPinLen, SCHAR * pscNewPin, UINT32 *pu32RetryCount)

功能描述　调用该函数可以修改 PIN 码，如果原 PIN 码错误导致验证失败，会返回 PIN 码的剩余重试次数，当剩余次数为 0 时，表示 PIN 码已经锁死

参数　　u32OldPinLen　　[IN] 旧 PIN 码长度

　　pscOldPin　　[IN] 旧 PIN 码

　　u32NewPinLen　　[IN] 新 PIN 码长度

　　pscNewPin　　[IN] 新 PIN 码

　　pu32RetryCount　[OUT] 出错后返回的重试次数

返回值　0：成功

其他：错误码

6.2.5 重置 PIN

函数原型　INT32 ITSReloadPin(UINT32 u32PinLen, SCHAR* pscPin, UINT32 u32MasterKeyLen, UCHAR * pucMasterKey, UINT32 *pu32RetryCount)

功能描述　当用户 PIN 码锁死后，调用该函数来解锁 PIN 码，解锁后，用户的 PIN 码被设置成新值，用户 PIN 码的重试次数也恢复到原值

参数　u32PinLen　[IN] 新 PIN 码长度

pscPin　[IN] 新 PIN 码

u32MasterKeyLen [IN] 密码设备主控密钥长度

pucMasterKey　[IN] 密码设备主控密钥

pu32RetryCount　[OUT] 主控密钥错误后返回的重试次数

返回值　0：成功

其他：错误码

备注　禁止主控密钥以明文形式传入密码设备，可在接口内部加密传输

6.2.6 更新主控密钥

函数原型　INT32 ITSUpdateMasterKey(UINT32 u32OldMasterKeyLen, UCHAR * pucOldMasterKey, UINT32 u32NewMasterKeyLen, UCHAR * pucNewMasterKey)

功能描述　此接口用来更新主控密钥

参数　u32OldMasterKeyLen [IN] 旧的主控密钥长度

pucOldMasterKey　[IN] 旧的主控密钥

u32NewMasterKeyLen [IN] 新的主控密钥长度

pucNewMasterKey　[IN] 新的主控密钥

返回值　0：成功

其他：错误码

备注　密码设备初始主控密钥为 16 字节全 0，禁止主控密钥以明文形式传入密码设备，可在接口内部加密传输

6.3 证书管理

6.3.1 证书导入

函数原型　INT32 ITSImportCert(UINT32 u32Index, ITSCertType_e eCertType, UINT32 u32Len, UCHAR * pucCert)

功能描述　向密码设备导入证书，证书类型见 5.4.10

参数　u32Index [IN] 容器索引

eCertType [IN] 证书类型

u32Len [IN] 证书长度

pucCert [IN] 证书

返回值　0：成功

其他：错误码

6.3.2 证书导出

函数原型　INT32 ITSExportCert(UINT32 u32Index, ITSCertType_e eCertType, UCHAR* pucCertData, UINT32 u32DataLen, UINT32* pu32CertLen)

功能描述　从密码设备导出证书，证书类型见 5.4.10

参数　u32Index [IN] 容器索引

eCertType [IN] 证书类型

pucCertData [OUT] 存取证书的缓冲区指针

u32DataLen [IN] 存取证书的缓冲区长度

pu32CertLen [OUT] 导出的证书长度指针

返回值　0：成功

其他：错误码

6.3.3 证书删除

函数原型　INT32 ITSDeleteCert(UINT32 u32Index, ITSCertType_e eCertType)

功能描述　删除密码设备中的证书，证书类型见 5.4.10

参数　　u32Index　[IN] 容器索引

　　eCertType　[IN] 证书类型

返回值　0：成功。

　　其他：错误码

6.4 通用密码服务接口

定义说明密码服务函数提供对称算法运算、非对称算法运算、密码杂凑运算、密钥管理（生成、导出）等功能

6.4.1 生成随机数

函数原型　INT32 ITSGetRandom(UINT32 u32RandomLen, UCHAR* pucRandom)

功能描述　产生指定长度的随机数

参数　　u32RandomLen　[IN] 随机数长度

　　pucRandom　[OUT] 返回的随机数

返回值　0：成功

　　其他：错误码

6.4.2 生成 ECC 密钥对

函数原型　INT32 ITSGenerateEccKey(UINT32 u32Index, UINT32 u32AlgId)

功能描述　生成 ECC 密钥对，存储在密码设备内

参数　　u32Index　[IN] 密钥对索引

　　u32AlgId　[IN] 算法标识，0-SM2；1-nistP256；2-brainpoolP256r1

返回值　0：成功

　　其他：错误码

6.4.3 导出 ECC 公钥对

函数原型　INT32 ITSExportEccKey(UINT32 u32Index, UINT32 u32AlgId, EccPubKey_t *ptPubKey)

功能描述　从密码设备导出 ECC 公钥

参数　　u32Index　[IN] 密钥对索引

u32AlgId　[IN] 算法标识，0-SM2；1-nistP256；2-brainpoolP256r1

PtPubKey　[OUT] ECC 公钥数据结构

返回值　　0：成功

其他：错误码

6.4.4 ECC 公钥加密

函数原型　　INT32 ITSEccPubKeyEncrypt(UINT32 u32AlgId, EccPubKey_t *ptPubKey UCHAR* pucData, UINT32 u32DataLen, UCHAR* pucDataOutput, UINT32 u32OutputLen, UINT32* pu32OutDataLen)

功能描述　使用 ECC 公钥对数据进行加密运算

参数　　u32AlgId　　[IN] 算法标识，0-SM2；1-nistP256；2-brainpoolP256r1

PtPubKey　　[IN] ECC 公钥数据结构

pucData　　[IN] 输入的待加密数据。

u32DataLen　[IN] 输入的待加密数据长度

pucDataOutput [OUT] 缓冲区指针，用于存放输出的数据

u32OutputLen　[IN] 缓冲区长度

pu32OutDataLen[OUT] 输出的数据长度

返回值　　0：成功

其他：错误码

6.4.5 ECC 私钥解密

函数原型　INT32 ITSEccPriKeyDecryptIndex(UINT32 u32AlgId, UINT32 u32Index, UCHAR* pucData, UINT32 u32DataLen, UCHAR* pucDataOutput, UINT32 u32OutputLen, UINT32* pu32OutDataLen)

功能描述　使用密码设备内部 ECC 私钥对数据进行解密运算。

参数　　u32AlgId　　[IN] 算法标识，0-SM2；1-nistP256；2-brainpoolP256r1

u32Index [IN] ECC 密钥对索引

pucData [IN] 输入的待解密数据

u32DataLen [IN] 输入的待解密数据长度

pucDataOutput [OUT] 缓冲区指针，用于存放输出的数据

u32OutputLen [IN] 缓冲区长度

pu32OutDataLen[OUT] 输出的数据长度

返回值 0：成功

其他：错误码

6.4.6 ECC 密钥签名

函数原型 INT32 ITSEccPriKeySignIndex(UINT32 u32AlgId, UINT32 u32Index,

UCHAR* pucData, UINT32 u32DataLen, EccSignature_t *ptEccSignature)

功能描述 使用密码设备内部 ECC 私钥对数据进行签名运算

参数 u32AlgId [IN] 算法标识，0-SM2；1-nistP256；2-brainpoolP256r1

u32Index [IN] ECC 密钥对索引

pucData [IN] 输入的待签名数据

u32DataLen [IN] 输入的待签名的数据长度

ptEccSignature[OUT] 签名值

返回值 0：成功

其他：错误码

6.4.7 ECC 签名验证

函数原型 INT32 ITSEccPubKeyVerify(UINT32 u32AlgId, EccPubKey_t *ptPubKey,

UCHAR* pucData, UINT32 u32DataLen, EccSignature_t *ptEccSignature)

功能描述 使用 ECC 公钥对数据进行签名验证

参数 u32AlgId [IN] 算法标识，0-SM2；1-nistP256；2-brainpoolP256r1

ptPubKey [IN] ECC 公钥

pucData [IN] 输入的待验证数据

u32DataLen [IN] 输入的待验证的数据长度

ptEccSignature[IN] 签名值

返回值 0：成功

其他：错误码

6.4.8 导入对称密钥

函数原型 uint32 ImportSymmKey(uint32 nAlg, uint32 nKeyIndex, uint32 nLock, uint32 nKeyLen, uint8* szKey, uint8* szMac);

功能描述 导出对称密钥

参数 nAlg [IN] 导入密钥算法: 0-ALL; 1-DES; 2-AES; 3-SM4

nKeyIndex [IN] 密钥存储的索引号

nLock [IN] 密钥是否锁定，不可修改: 0-不锁定; 1-锁定

nKeyLen [IN] 密钥的长度

szKey [IN] 密钥密文，使用 KMC 加密密钥明文得到

（明文:2 字节长度+密钥明文+8000 补位，不强制补位）

szMac [IN] 校验值，使用 KMC 对密钥信息计算的 MAC 值，4 字节

返回值 0： 成功

其他：错误码

6.4.9 删除对称密钥

函数原型 uint32 DeleteSymmKey(uint32 nKeyIndex);

功能描述 删除对称密钥

参数 nKeyIndex [IN] 密钥存储的索引号

返回值 0： 成功

其他：错误码

6.4.10 导出对称密钥

函数原型 uint32 ExportSymmKey(uint32 nKeyIndex, uint32 nKekIndex, uint32* nAlg, uint32* nKeyLen, uint8* szKey, uint8* szCV);

功能描述 对称密钥导出

参数

	nKeyIndex	[IN] 密钥存储的索引号
	nKekIndex	[IN] 保护密钥索引号
	nAlg	[OUT] 密钥算法: 0-ALL; 1-DES; 2-AES; 3-SM4
	nKeyLen	[OUT] 密钥长度
	szKey	[OUT] 密钥密文，使用保护密钥加密密钥明文得到（明文:2 字节长度+密钥明文+8000 补位，不强制补位）
	szCV	[OUT] 密钥的校验值

返回值 0： 成功

其他：错误码

6.4.11 SM4 ECB 对称运算

函数原型 uint32 SM4ECBIndex(uint32 nMode, uint32 nKeyIndex, uint32 nDatalen, uint8* szData, uint32* nOutlen, uint8* szOutData);

功能描述 SM4 使用内部密钥,加解密 ECB 模式

参数

	nMode	[IN] 运算模式：0-加密；1-解密
	nKeyIndex	[IN] 运算密钥索引
	nDatalen	[IN] 待运算数据长度
	szData	[IN] 待运算数据，需要外部补位
	nOutlen	[OUT] 运算结果长度
	szOutData	[OUT] 运算结果

返回值 0： 成功

其他：错误码

6.4.12 SM4 CBC 对称运算

函数原型	uint32 SM4CBCIndex(uint32 nMode, uint32 nKeyIndex, uint8* szIV, uint32 nDatalen, uint8* szData,uint32* nOutlen, uint8* szOutData);	
功能描述	SM4 使用内部密钥，加解密 CBC 模式	
参数	nMode	[IN] 运算模式：0-加密；1-解密
	nKeyIndex	[IN] 运算密钥索引
	szIV	[IN] 初始化向量
	nDatalen	[IN] 待运算数据长度
	szData	[IN] 待运算数据，需要外部补位
	nOutlen	[OUT] 运算结果长度
	szOutData	[OUT] 运算结果
返回值	0： 成功	
	其他：错误码	

6.4.13 SM4 CCM 对称运算

函数原型	uint32 SM4CCMIndex(uint32 nMode, uint32 nKeyIndex, uint32 nIVLen, uint8* szIV,uint32 nDatalen, uint8* szData, uint32* pnOutlen, uint8* szOutData);	
功能描述	SM4 使用内部密钥，加解密 CCM 模式	
参数	nMode	[IN] 运算模式：0-加密；1-解密
	nKeyIndex	[IN] 运算密钥索引
	nIVLen	[IN] 初始化向量长度
	szIV	[IN] 初始化向量
	nDatalen	[IN] 待运算数据长度
	szData	[IN] 待运算数据，如果 nMode = 1，数据格式：密文||标签(16 字节)

nOutlen	[OUT] 运算结果长度
szOutData	[OUT] 运算结果，如果 nMode = 0，数据格式：密文||标签(16 字节)

返回值　0： 成功

其他：错误码

6.5 智能交通车辆服务接口

基于智能交通服务的特殊性，这里为了智能交通证书以及智能交通密钥的统一管理，定义说明了智能交通 CA 证书和终端证书的导入、导出、删除，终端证书密钥的产生、签名等操作。智能交通服务系列函数如表 15 所示：

表 15 智能交通服务系列函数描述

函数名称	功能
智能交通 ECertGenKeyIndex	智能交通-终端证书密钥对产生
智能交通 ImportECert	智能交通-终端证书导入
智能交通 ExportECert	智能交通-终端证书导出
智能交通 DeleteECert	智能交通-终端证书删除
智能交通 ECertSignIndex	智能交通-终端证书私钥 SM2 签名
智能交通 ECertSM2SignIndexWithIDA	智能交通-终端证书私钥 SM2 签名带 IDA
智能交通 ECertSM2SKDecryptIndex	智能交通-终端证书 SM2 私钥解密
智能交通 VerifyKeyPairIndex	智能交通-验证内部密钥对合法性
智能交通 GetPK	智能交通-获取终端证书的密钥对公钥
智能交通 ECertSignIndexExp	智能交通-终端证书衍生私钥签名
智能交通 ECertSM2SignIndexExpWithIDA	智能交通-终端证书衍生私钥 SM2 签名带 IDA
智能交通 ECertSKDeriveIndex	智能交通-终端证书私钥衍生并存储
智能交通 Verify	智能交通-验证签名(支持压缩公钥)

6.5.1 智能交通-终端证书密钥对产生

函数原型　uint32 智能交通 ECertGenKeyIndex(uint32 index, uint8 alg, uint8* szX,uint8* szY);

功能描述　智能交通终端证书密钥对产生，密钥对内部存储

参数　index　[IN] 证书索引

alg　[IN] 算法: 0-SM2; 1-nistP256;2-brainpoolP256r1

nistP256 和 brainpoolP256r1 非必实现算法

szX　[OUT] 公钥 X，定长 32 字节

szY　[OUT] 公钥 Y，定长 32 字节

返回值　0： 成功

其他：错误码

6.5.2 智能交通-终端证书私钥签名

函数原型　uint32 智能交通 ECertSignIndex(uint32 index, uint8* hash, uint8 *r, uint8* s);

功能描述　使用终端私钥签名

参数　index　[IN] 证书索引

hash　[IN] 待签名的 HASH 数据，定长 32 字节

r　[OUT] 签名 R，定长 32 字节

s　[OUT] 签名 S，定长 32 字节

返回值　0： 成功

其他：错误码

6.5.3 智能交通-终端证书私钥签名

函数原型　uint32 智能交通 ECertSM2SignIndexWithIDA(uint32 index, uint32 len, uint8* data,

uint8 *x, uint8* y, uint8 *r, uint8* s);

功能描述　使用终端私钥签名 SM2 算法，数据先执行 SM3-E 哈希

参数　index　[IN] 证书索引

	len	[IN] 待签名数据长度
	data	[IN] 待签名数据
	x	[IN] 参与签名的公钥 X
	y	[IN] 参与签名的公钥 Y
	r	[OUT] 签名 R，定长 32 字节
	s	[OUT] 签名 S，定长 32 字节
返回值	0： 成功	
	其他：错误码	

6.5.4 智能交通-终端证书私钥解密

函数原型 uint32 智能交通 ECertSM2SKDecryptIndex(uint32 index, uint32 nDatalen, uint8* szData,uint32* nOutlen, uint8* szOutData);

功能描述 智能交通-终端证书使用内部私钥 SM2 解密

参数	index	[IN] 证书索引
	nDatalen	[IN] 待运算数据长度
	szData	[IN] 待运算数据
	nOutlen	[OUT] 输出数据长度
	szOutData	[OUT] 输出数据
返回值	0： 成功	
	其他：错误码	

6.5.5 智能交通-获取终端证书的密钥对公钥

函数原型 uint32 智能交通 GetPK(uint32 index, uint8* szX, uint8* szY);

功能描述 智能交通-获取密钥对公钥

参数	index	[IN] 证书索引
	szX	[OUT] 公钥 X，定长 32 字节

	szY	[OUT] 公钥 Y，定长 32 字节

返回值　0：成功

其他：错误码

6.5.6 智能交通-终端证书衍生私钥签名

函数原型　uint32 智能交通 ECertSignIndexExp(uint32 index, uint8* hash, uint8* exp, uint8* c, uint8 *r, uint8* s);

功能描述　使用终端衍生私钥签名 SM2 算法，直接对输入的 HASH 签名

车载硬件设备内部先执行 Private key = a + exp + c，然后再使用衍生私钥签名

参数	index	[IN] 证书索引
	hash	[IN] 待签名的 HASH 数据，定长 32 字节
	exp	[IN] 衍生因子，定长 32 字节
	c	[IN] 服务器返回的私钥因子，定长 32 字节
	r	[OUT] 签名 R，定长 32 字节
	s	[OUT] 签名 S，定长 32 字节

返回值　0：成功

其他：错误码

6.5.7 智能交通-终端证书衍生私钥 SM2 签名带 IDA

函数原型　uint32 智能交通 ECertSM2SignIndexExpWithIDA(uint32 index, uint32 len, uint8* data, uint8 *x, uint8* y, uint8* exp, uint8* c, uint8 *r, uint8* s);

功能描述　使用终端衍生私钥签名 SM2 算法，数据先执行 SM3-E 哈希

车载硬件设备内部先执行 Private key = a + exp + c，然后再使用衍生私钥签名

参数	index	[IN] 证书索引
	len	[IN] 待签名数据长度

	data	[IN] 待签名数据
	x	[IN] 参与签名的公钥 X
	y	[IN] 参与签名的公钥 Y
	exp	[IN] 衍生因子，定长 32 字节
	c	[IN] 服务器返回的私钥因子，定长 32 字节
	r	[OUT] 签名 R，定长 32 字节
	s	[OUT] 签名 S，定长 32 字节
返回值	0： 成功	
	其他：错误码	

6.5.8 智能交通-终端证书私钥衍生并存储

函数原型 uint32 智能交通 ECertSKDeriveIndex(uint32 index1, uint32 index2, uint8* f, uint8* e, uint8 *r, uint8* x, uint8* y);

功能描述 内部私钥衍生后产生新的密钥对，并存入新的索引中

衍生计算公式：k_new = (k_seed + f)* e + r

参数	index1	[IN] 待衍生的私钥索引
	index2	[IN] 待存储的私钥索引
	f	[IN] 计算因子 f
	e	[IN] 计算因子 e
	r	[IN] 计算因子 r
	x	[OUT] 签名 R，定长 32 字节
	y	[OUT] 签名 S，定长 32 字节
返回值	0： 成功	
	其他：错误码	

6.5.9 智能交通-验证签名(支持压缩公钥)

函数原型	uint32 智能交通 Verify(uint32 nHashFlag, uint32 nAlg, uint32 nPKType , uint8* szPk, uint8* szR, uint8* szS, uint32 nIDALen, uint8* szIDA, uint32 nDataLen, uint8* szData);	
功能描述	智能交通签名验证，使用压缩公钥，支持 IDA 哈希	
参数	nHashFlag	[IN] 哈希标志: 0-不哈希；1-先哈希后验证签名
	nAlg	[IN] 算法标识：0-SM2; 1-nistP256; 2-brainpoolP256r1
	nPKType	[IN] 公钥类型: 0-压缩公钥(公钥长度 33 字节,格式: ODD_FLAG||X) 1-非压缩公钥(公钥长度 64 字节，格式 X||Y)
	szPk	[IN] 公钥(根据 nPKType，公钥格式不同)
	szR	[IN] 签名 R
	szS	[IN] 签名 S
	nIDALen	[IN] IDA 的长度，如果为 0，表示无 IDA 参与 HASH
	szIDA	[IN] IDA 数据
	nDataLen	[IN] 待验证签名数据长度
	szData	[IN] 待验证签名数据
返回值	0： 成功 其他：错误码	

7 安全设备的安全要求

7.1 安全设备使用阶段

安全安全设备须具备硬件层面的产品生命周期状态管理功能，并配合固件系统实现安全设备全生命周期状态管理机制

7.2 权限管理

7.2.1 权限分类

权限分为安全设备权限和用户权限。

安全设备权限：密码设备出厂时预置主控密钥，车辆激活时需更新安全设备主控密钥，用于敏感数据的访问控制和用户 PIN 码的控制。

用户权限：用户 PIN 码验证通过后，获得用户权限，用户权限只作用于接口的调用。

7.2.2 权限使用

权限的使用应遵循以下要求：

a) 安全设备权限用于更新认证密钥，更新主控密钥，重置用户 PIN 码和导入外部密钥。

b) 调用相关 API 接口需要用户权限。

c) 相关 API 接口调用是否需要用户权限在设计时指定。

d) 用户 PIN 码的修改需要用户权限，用户 PIN 码的重置需要安全设备权限。

7.2.3 PIN 码安全要求

PIN 码的使用应遵循以下要求：

a) PIN 码长度不少于 6 个字节。

b) PIN 码在安全设备和本接口之间的传输过程中应采取保护措施，防止 PIN 码泄露。

c) PIN 码在安全设备中应安全存储，不可从设备中导出。

7.3 密钥安全要求

密钥应遵循以下安全要求：

a) 安全设备内产生的随机数应为真随机数，应符合随机性检测的要求。

b) 安全设备内产生的会话密钥应使用随机数。

c) 安全设备内的密钥应具备有效的密钥保护机制防止解剖、探测和读取。

d) 安全设备内的密钥应按权限要求使用。

e) 除公钥外的密钥不能以明文形式出现在安全设备外。

f) 签名私钥必须在安全设备中产生。

g) 安全安全设备失效时必须销毁该安全设备内所有的密钥。

7.4 安全设备抗攻击要求

安全设备应具有以下抗攻击要求：

a) 安全设备需要达到《GM/T0008 安全设备密码检测准则》的安全防护要求。

b) 安全设备硬件电路需要具备主动保护的屏蔽层，并有完整的物理攻击检测报警逻辑电路。

c) 安全设备内部非易失性存储器需要有硬件实现的加密存储机制，对敏感数据（如密钥、根证书等）必须密文存储，并具备对敏感数据的访问控制机制。

d) 安全设备密码算法硬件单元必须具备专业的抗攻击能力。须能够防御通过侧信道攻击（Side Channel Attack）和故障攻击（Fault Attack）方式的各类密码 IP 攻击手段。在密码 IP 设计实现中应至少采用以下技术手段：

- 掩码(Masking)和盲化(Blinding)技术；
- 乱序执行技术（confusion）；
- 随机插入"伪操作"（Perudo Cycle）；
- 操作特征平衡设计(Crypto-Operation Equllibrium)；
- 运算校验技术；
- 内部寄存器数据加扰等。

e) 安全设备真随机数发生器（TRNG）电路必须进行冗余设计，应至少具备 4 个独立的物理发生源。

T/ITS XXXX- XXXX

中国智能交通产业联盟

标准

智能交通运输系统车载安全密码设备应用接口规范

T/ITS XXXX- XXXX

北京市海淀区西土城路 8 号（100088）

中国智能交通产业联盟印刷

网址：http://www.c-its.org.cn

2021 年 X 月第一版　2021 年 X 月第一次印刷