GY

中.华 人 民 共 和 国 广 播 电 视 行 业 标 准

GY/T 303.6—2018

智能电视操作系统
第 6 部分：可信执行环境接口

Smart TV operating system—
Part 6: Trusted execution environment interface

2018–07–06 发布　　2018 –07–06 实施

国家广播电视总局科技司　发布

目　　次

前言 II
1　范围 3
2　规范性引用文件 3
3　缩略语 3
4　概述 4
5　DCAS TEE 接口 4
6　DRM TEE 接口 5
7　融合 CA TEE 接口 15
8　北斗 TEE 接口 17

前　言

GY/T 303《智能电视操作系统》已经或计划发布以下部分：

——第1部分：功能与架构；

——第2部分：安全；

——第3部分：应用编程接口；

——第4部分：硬件抽象接口；

——第5部分：功能组件接口；

——第6部分：可信执行环境接口；

——第7部分：符合性测试。

本部分为GY/T 303的第6部分。

本部分按照GB/T 1.1—2009给出的规则起草。

请注意本部分的某些内容可能涉及专利。本部分的发布机构不承担识别这些专利的责任。

本部分由全国广播电影电视标准化技术委员会（SAC/TC 239）归口。

本部分起草单位：国家新闻出版广电总局广播科学研究院、国家广播电视网工程技术研究中心、华为技术有限公司、深圳市海思半导体有限公司、北京永新视博数字电视技术有限公司、上海兆芯集成电路有限公司、晨星软件研发（深圳）有限公司、北京数码视讯科技股份有限公司、杭州国芯科技股份有限公司、湖南国科微电子股份有限公司、上海高清数字科技产业有限公司。

本部分主要起草人： 盛志凡、刘金晓、郭晓霞、邹峰、解伟、严海峰、张晶、熊彬、郭永伟、郑力铮、钱义力、曾付山、刘晓翔、吴迪、赵良福、谢嵘、方中华、李望舒、黄新军、叶丰、王旭升、杨勋、王强。

智能电视操作系统
第6部分：可信执行环境接口

1 范围

GY/T 303的本部分规定了智能电视操作系统可信执行环境接口相关技术要求。

本部分适用于智能电视操作系统可信执行环境硬件抽象接口的研发、生产、测试和应用。

2 规范性引用文件

下列文件对于本部分的应用是必不可少的。凡是注日期的引用文件，仅所注日期的版本适用于本部分。凡是不注日期的引用文件，其最新版本（包括所有的修改单）适用于本部分。

GY/T 308—2017 单向可下载条件接收系统技术规范

3 缩略语

下列缩略语适用于本部分。

AES 高级加密标准（Advanced Encryption Standard）
ATR 复位应答（Answer To Reset）
CA 条件接收（Conditional Access）
CW 控制字（Control Word）
CBC 密码分组链接模式（Cipher Block Chaining ）
CRC 循环冗余校验(Cyclic Redundancy Check)
CTR 计算器模式（Counter）
CFB 密码反馈模式（Cipher FeedBack ）
DCAS 可下载条件接收系统（Downloadable Conditional Access System）
DES 数据加密标准（Data Encryption Standard）
DMA 直接内存存取（Direct Memory Access）
DRM 数字版权管理（Digital Rights Management）
DSD 可分离式安全设备（Detachable Security Device）
ECB 电码本模式（Electronic Codebook Book ）
HAL 硬件抽象层（Hardware Abstract Layer）
HSM 硬件安全模块（Hardware Security Module）
IV 初始向量（Initialization Vector）
KLAD 层级密钥（Keyladder）
OFB 输出反馈模式（Output FeedBack ）
Soc 芯片级系统（System on Chip）
TApp 可信应用（Trusted Application）
TEE 可信执行环境（Trusted Execution Environment）
TDES 三重数据加密标准（Triple DES）

TVOS　智能电视操作系统（Smart TV Operating System）

4　概述

TVOS TEE部分由安全操作系统、TVOS TEE硬件抽象层（TEE HAL）和可信应用(TApp)构成。

TVOS TEE硬件抽象层实现对TVOS TEE硬件平台能力的抽象封装，对同一类型的硬件设备采用统一的抽象封装模型，为上层可信应用对TEE硬件平台能力的访问和控制提供统一的调用接口。TVOS可信执行环境接口应包括全球平台组织基础应用编程接口和TVOS自有TEE应用编程接口等两大类，如图1所示。

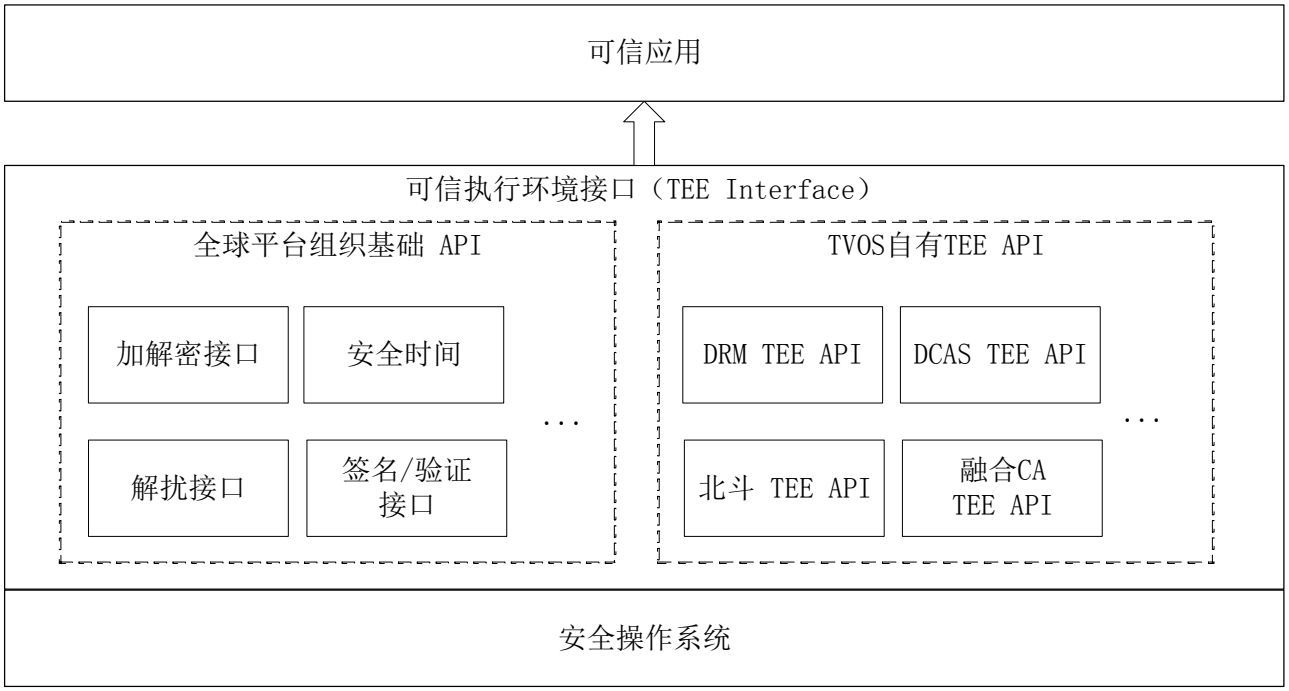

图1　TVOS 可信执行环境接口

TVOS可信执行环境接口支持全球平台组织编程接口，为上层TApp提供基础的加解密、解扰、安全时间、签名/验证等接口。

TVOS可信执行环境接口同时针对广播电视媒体业务需要，扩展了TVOS自有TEE应用编程接口，包括DCAS TEE接口模块、DRM TEE接口模块、融合CA TEE接口模块、北斗TEE接口模块等。

5　DCAS TEE 接口

5.1　安全芯片层级密钥驱动接口

安全芯片层级密钥驱动接口包括层级密钥初始化、层级密钥反初始化、读取安全芯片标识、通过SoC层级密钥计算挑战应答结果、设置解扰参数及密钥、触发解扰、停止解扰等接口，具体接口函数遵循GY/T 308—2017中B.3的规定。

5.2　硬件安全模块应用程序接口

硬件安全模块应用程序接口包括获取硬件安全模块软件版本号、获取硬件安全模块基本信息、获取硬件安全模块诊断信息、获取硬件安全模块存储容量及读写能力信息、获取最近一次硬件安全模块被激活的时间、获取激活信息中CA厂商专属的数据、生成激活请求数据、设置硬件安全模块消息、与硬件安全模块

建立安全认证通道、读取硬件安全模块数据、写入数据到硬件安全模块、读取硬件安全模块存储的位置信息、读取硬件安全模块公共存储区、向硬件安全模块公共存储区写入数据、设置硬件安全模块中重加密算法、硬件安全模块控制字CW、关闭硬件安全模块安全认证通道等接口，具体接口函数遵循GY/T 308—2017中B.4的规定。

5.3 加解密及签名校验接口

加解密及签名校验接口包括验证SM2数字签名、SM3散列、SM2加密、CRC校验、SM4加密、SM4解密等接口，具体接口函数遵循GY/T 308—2017中B.6.1的规定。

6 DRM TEE 接口

6.1 接口概述

DRM TEE接口包括初始化加解密硬件、去初始化加解密硬件、创建加解密句柄、销毁加解密句柄、配置加解密控制信息、配置加解密控制信息扩展、解密、获取加解密控制信息等接口。DRM TEE常量列表如表1所示，数据结构列表如表2所示，接口列表如表3所示。

表1 DRM TEE常量列表

常量	说明
TEE_CIPHER_WORK_MODE_E	加解密算法的工作模式
TEE_CIPHER_ALG_E	加解密算法
TEE_CIPHER_KEY_LENGTH_E	密钥长度
TEE_CIPHER_BIT_WIDTH_E	加解密位宽
TEE_CIPHER_TYPE_E	加解密类型选择
TEE_CIPHER_SM1_ROUND_E	SM1计算的轮数

表2 DRM TEE数据结构列表

数据结构	说明
TEE_CIPHER_CTRL_CHANGE_FLAG_S	加解密控制参数变更标志结构
TEE_CIPHER_CTRL_S	加解密类型结构
TEE_CIPHER_CTRL_S	加解密控制信息结构
TEE_CIPHER_CTRL_AES_S	AES加解密控制信息结构
TEE_CIPHER_CTRL_AES_CCM_GCM_S	AES CCM/GCM 加解密控制信息结构
TEE_CIPHER_CTRL_DES_S	DES加解密控制信息结构
TEE_CIPHER_CTRL_3DES_S	3DES加解密控制信息结构
TEE_CIPHER_CTRL_SM1_S	SM1加解密控制信息结构
TEE_CIPHER_CTRL_SM4_S	SM4加解密控制信息结构
TEE_CIPHER_CTRL_EX_S	加解密控制信息扩展结构

表3 DRM TEE接口列表

接口	说明
TEE_CIPHER_Init	初始化加解密硬件
TEE_CIPHER_DeInit	去初始化加解密硬件
TEE_CIPHER_CreateHandle	创建加解密句柄
TEE_CIPHER_DestroyHandle	销毁加解密句柄
TEE_CIPHER_ConfigHandle	配置加解密控制信息
TEE_CIPHER_ConfigHandleEx	配置加解密控制信息的扩展接口
TEE_CIPHER_Decrypt	解密接口
TEE_CIPHER_GetHandleConfig	获取加解密控制信息

6.2 常量定义

6.2.1 加解密算法的工作模式

原型：

```
typedef enum _TEE_CIPHER_WORK_MODE_E
{
    TEE_CIPHER_WORK_MODE_ECB,
    TEE_CIPHER_WORK_MODE_CBC,
    TEE_CIPHER_WORK_MODE_CFB,
    TEE_CIPHER_WORK_MODE_OFB,
    TEE_CIPHER_WORK_MODE_CTR,
    TEE_CIPHER_WORK_MODE_CCM,
    TEE_CIPHER_WORK_MODE_GCM,
    TEE_CIPHER_WORK_MODE_CBC_CTS,
    TEE_CIPHER_WORK_MODE_BUTT,
    TEE_CIPHER_WORK_MODE_INVALID  = 0xffffffff,
}TEE_CIPHER_WORK_MODE_E;
```

描述：

加解密算法的工作模式

成员：

TEE_CIPHER_WORK_MODE_ECB：ECB模式

TEE_CIPHER_WORK_MODE_CBC：CBC模式

TEE_CIPHER_WORK_MODE_CFB：CFB模式

TEE_CIPHER_WORK_MODE_OFB：OFB模式

TEE_CIPHER_WORK_MODE_CTR：CTR模式

TEE_CIPHER_WORK_MODE_CCM：CTR模式

TEE_CIPHER_WORK_MODE_GCM：GCM模式

TEE_CIPHER_WORK_MODE_CBC_CTS：CBC_CTS模式

TEE_CIPHER_WORK_MODE_BUTT：加解密算法的工作模式枚举定义最大值

TEE_CIPHER_WORK_MODE_INVALID：字节对齐边界值

6.2.2 加解密算法

原型：

```
typedef enum _TEE_CIPHER_ALG_E
{
    TEE_CIPHER_ALG_DES          = 0x0,
    TEE_CIPHER_ALG_3DES         = 0x1,
    TEE_CIPHER_ALG_AES          = 0x2,
    TEE_CIPHER_ALG_SM1          = 0x3,
    TEE_CIPHER_ALG_SM4          = 0x4,
    TEE_CIPHER_ALG_DMA          = 0x5,
    TEE_CIPHER_ALG_BUTT         = 0x6,
    TEE_CIPHER_ALG_INVALID      = 0xffffffff,
}TEE_CIPHER_ALG_E;
```

描述：

加解密算法

成员：

TEE_CIPHER_ALG_DES：DES算法

TEE_CIPHER_ALG_3DES：3DES算法

TEE_CIPHER_ALG_AES：AES算法

TEE_CIPHER_ALG_SM1：SM1算法

TEE_CIPHER_ALG_SM4：SM4算法

TEE_CIPHER_ALG_DMA：DMA拷贝

TEE_CIPHER_ALG_BUTT：加解密算法枚举定义最大值

TEE_CIPHER_ALG_INVALID：字节对齐边界值

6.2.3 密钥长度

原型：

```
typedef enum _TEE_CIPHER_KEY_LENGTH_E
{
    TEE_CIPHER_KEY_AES_128BIT    = 0x0,
    TEE_CIPHER_KEY_AES_192BIT    = 0x1,
    TEE_CIPHER_KEY_AES_256BIT    = 0x2,
    TEE_CIPHER_KEY_DES_3KEY      = 0x2,
    TEE_CIPHER_KEY_DES_2KEY      = 0x3,
    TEE_CIPHER_KEY_DEFAULT       = 0x0,
    TEE_CIPHER_KEY_INVALID       = 0xffffffff,
}TEE_CIPHER_KEY_LENGTH_E;
```

描述：

密钥长度常量定义

成员：

TEE_CIPHER_KEY_AES_128BIT：AES运算方式下采用128bit密钥长度

TEE_CIPHER_KEY_AES_192BIT：AES运算方式下采用192bit密钥长度

TEE_CIPHER_KEY_AES_256BIT：AES运算方式下采用256bit密钥长度

TEE_CIPHER_KEY_DES_3KEY：DES运算方式下采用3个key

TEE_CIPHER_KEY_DES_2KEY：DES运算方式下采用2个key

TEE_CIPHER_KEY_DEFAULT：默认Key长度，DES-8, SM1-48, SM4-16
TEE_CIPHER_KEY_INVALID：字节对齐边界值

6.2.4 加解密位宽

原型：

```
typedef enum _TEE_CIPHER_BIT_WIDTH_E
{
    TEE_CIPHER_BIT_WIDTH_64BIT   = 0x0,
    TEE_CIPHER_BIT_WIDTH_8BIT    = 0x1,
    TEE_CIPHER_BIT_WIDTH_1BIT    = 0x2,
    TEE_CIPHER_BIT_WIDTH_128BIT  = 0x3,
    TEE_CIPHER_BIT_WIDTH_INVALID = 0xffffffff,
}TEE_CIPHER_BIT_WIDTH_E;
```

描述：

加解密位宽

成员：

TEE_CIPHER_BIT_WIDTH_64BIT：64bit位宽
TEE_CIPHER_BIT_WIDTH_8BIT：8bit位宽
TEE_CIPHER_BIT_WIDTH_1BIT：1bit位宽
TEE_CIPHER_BIT_WIDTH_128BIT：128bit位宽
TEE_CIPHER_BIT_WIDTH_INVALID：字节对齐边界值

6.2.5 加解密类型选择

原型：

```
typedef enum
{
    TEE_CIPHER_TYPE_NORMAL  = 0x0,
    TEE_CIPHER_TYPE_COPY_AVOID,
    TEE_CIPHER_TYPE_BUTT,
    TEE_CIPHER_TYPE_INVALID = 0xffffffff,
}TEE_CIPHER_TYPE_E;
```

描述：

加解密类型选择

成员：

TEE_CIPHER_TYPE_NORMAL：常规模式
TEE_CIPHER_TYPE_COPY_AVOID：免拷贝模式
TEE_CIPHER_TYPE_BUTT：加解密类型枚举定义最大值
TEE_CIPHER_TYPE_INVALID：字节对齐边界值

6.2.6 SM1 计算的轮数

原型：

```
typedef enum _TEE_CIPHER_SM1_ROUND_E
{
    TEE_CIPHER_SM1_ROUND_08 = 0x00,
```

```
    TEE_CIPHER_SM1_ROUND_10 = 0x01,
    TEE_CIPHER_SM1_ROUND_12 = 0x02,
    TEE_CIPHER_SM1_ROUND_14 = 0x03,
    TEE_CIPHER_SM1_ROUND_BUTT,
    TEE_CIPHER_SM1_ROUND_INVALID  = 0xffffffff,
}TEE_CIPHER_SM1_ROUND_E;
```

描述：

SM1计算的轮数

成员：

TEE_CIPHER_SM1_ROUND_08：SM1计算的轮数为8

TEE_CIPHER_SM1_ROUND_10：SM1计算的轮数为10

TEE_CIPHER_SM1_ROUND_12：SM1计算的轮数为12

TEE_CIPHER_SM1_ROUND_14：SM1计算的轮数为14

TEE_CIPHER_SM1_ROUND_BUTT：SM1计算轮数枚举的最大值

TEE_CIPHER_SM1_ROUND_INVALID：字节对齐边界值

6.3 数据结构定义

6.3.1 加解密控制参数变更标志结构

原型：

```
typedef struct _TEE_CIPHER_CTRL_CHANGE_FLAG_S
{
    uint32_t   bit1IV:4;
    uint32_t   bitsResv:28;
}TEE_CIPHER_CTRL_CHANGE_FLAG_S;
```

描述：

加解密控制参数变更标志

成员：

bit1IV：IV向量变更标志位，0表示不设置；1表示只设置第一个包；2表示每个包都设置

bitsResv：预留位

6.3.2 加解密类型结构

原型：

```
typedef struct
{
    TEE_CIPHER_TYPE_E enCipherType;
}TEE_CIPHER_ATTS_S;
```

描述：

加解密类型结构

成员：

enCipherType：参见6.1.5 TEE_CIPHER_TYPE_E的定义

6.3.3 加解密控制信息结构

原型：

```
typedef struct TEE_CIPHER_CTRL_S
{
    uint32_t u32Key[8];
    uint32_t u32IV[4];
    TEE_CIPHER_ALG_E enAlg;
    TEE_CIPHER_BIT_WIDTH_E enBitWidth;
    TEE_CIPHER_WORK_MODE_E enWorkMode;
    TEE_CIPHER_KEY_LENGTH_E enKeyLen;
    TEE_CIPHER_CTRL_CHANGE_FLAG_S stChangeFlags;
} TEE_CIPHER_CTRL_S;
```

描述：

加解密控制信息结构

成员：

u32Key[8]：输入密钥

u32IV[4]：初始向量

enAlg：加解密算法

enBitWidth：加密或解密的位宽

enWorkMode ：加解密算法的工作模式

enKeyLen：密钥长度

stChangeFlags：控制信息变更选项，选项中没有标识的项默认全部变更

6.3.4 AES 加解密控制信息结构

原型：

```
typedef struct _TEE_CIPHER_CTRL_AES_S
{
    uint32_t u32EvenKey[8];
    uint32_t u32OddKey[8];
    uint32_t u32IV[4];
    TEE_CIPHER_BIT_WIDTH_E enBitWidth;
    TEE_CIPHER_KEY_LENGTH_E enKeyLen;
    TEE_CIPHER_CTRL_CHANGE_FLAG_S stChangeFlags;
} TEE_CIPHER_CTRL_AES_S;
```

描述：

AES加解密控制信息结构

成员：

u32EvenKey[8]：输入偶密钥，默认使用偶密钥

u32OddKey[8]：输入奇密钥，只对多包加解密有效

u32IV[4]：初始向量

enBitWidth：加密或解密的位宽

enKeyLen：密钥长度

stChangeFlags：控制信息变更选项，选项中没有标识的项默认全部变更

6.3.5 AES CCM/GCM 加解密控制信息结构

原型：

```
typedef struct TEE_CIPHER_CTRL_AES_CCM_GCM_S
{
    uint32_t u32Key[8];
    uint32_t u32IV[4];
    TEE_CIPHER_KEY_LENGTH_E enKeyLen;
    uint32_t u32IVLen;
    uint32_t u32TagLen;
    uint32_t u32ALen;
    uint32_t u32APhyAddr;
}TEE_CIPHER_CTRL_AES_CCM_GCM_S;
```

描述：

AES CCM/GCM 加解密控制信息结构

成员：

u32Key[8]: 输入密钥

u32IV[4]: 初始向量

enKeyLen: 密钥长度

u32IVLen: CCM/GCM的IV长度，CCM的取值范围{7, 8, 9, 10, 11, 12, 13}，GCM的取值范围[1-16]

u32TagLen: CCM的TAG长度，取值范围{4,6,8,10,12,14,16}

u32ALen: CCM/GCM的关联数据长度

u32APhyAddr: CCM/GCM的关联数据长度

6.3.6 DES 加解密控制信息结构

原型：

```
typedef struct _TEE_CIPHER_CTRL_DES_S
{
    uint32_t u32Key[2];
    uint32_t u32IV[2];
    TEE_CIPHER_BIT_WIDTH_E enBitWidth;
    TEE_CIPHER_CTRL_CHANGE_FLAG_S stChangeFlags;
}TEE_CIPHER_CTRL_DES_S;
```

描述：

DES加解密控制信息结构

成员：

u32Key[2]:输入偶密钥

u32IV[2]:初始向量

enBitWidth:加密或解密的位宽

stChangeFlags:控制信息变更选项，选项中没有标识的项默认全部变更

6.3.7 3DES 加解密控制信息结构

原型：

```
typedef struct _TEE_CIPHER_CTRL_3DES_S
{
    uint32_t u32Key[6];
    uint32_t u32IV[2];
```

```
        TEE_CIPHER_BIT_WIDTH_E enBitWidth;
        TEE_CIPHER_KEY_LENGTH_E enKeyLen;
        TEE_CIPHER_CTRL_CHANGE_FLAG_S stChangeFlags;
    }TEE_CIPHER_CTRL_3DES_S;
```

描述：

3DES加解密控制信息结构

成员：

u32Key[6]:输入偶密钥

u32IV[2]:初始向量

enBitWidth:加密或解密的位宽

enKeyLen:密钥长度

stChangeFlags:控制信息变更选项，选项中没有标识的项默认全部变更

6.3.8 SM1 加解密控制信息结构

原型：

```
typedef struct _TEE_CIPHER_CTRL_SM1_S
{
    uint32_t u32EK[4];
    uint32_t u32AK[4];
    uint32_t u32SK[4];
    uint32_t u32IV[4];
    TEE_CIPHER_BIT_WIDTH_E enBitWidth;
    TEE_CIPHER_SM1_ROUND_E enSm1Round;
    TEE_CIPHER_CTRL_CHANGE_FLAG_S stChangeFlags;
}TEE_CIPHER_CTRL_SM1_S;
```

描述：

SM1加解密控制信息结构

成员：

u32EK[4]: 输入密钥EK

u32AK[4]: 输入密钥AK

u32SK[4]: 输入密钥SK

u32IV[4]: 初始向量

enBitWidth: 加密或解密的位宽

enSm1Round: SM1计算的轮数配置

stChangeFlags: 控制信息变更选项，选项中没有标识的项默认全部变更

6.3.9 SM4 加解密控制信息结构

原型：

```
typedef struct _TEE_CIPHER_CTRL_SM4_S
{
    uint32_t u32Key[4];
    uint32_t u32IV[4];
    TEE_CIPHER_CTRL_CHANGE_FLAG_S stChangeFlags;
} TEE_CIPHER_CTRL_SM4_S;
```

描述：

SM4加解密控制信息结构

成员：

u32Key[4]: 输入密钥

u32IV[4]: 初始向量

stChangeFlags: 控制信息变更选项，选项中没有标识的项默认全部变更

6.3.10 加解密控制信息扩展结构

原型：

```
typedef struct _TEE_CIPHER_CTRL_EX_S
{
    TEE_CIPHER_ALG_E enAlg;
    TEE_CIPHER_WORK_MODE_E enWorkMode;
    void *pParam;
} TEE_CIPHER_CTRL_EX_S
```

描述：

加解密控制信息扩展结构

成员：

enAlg: 加解密算法

enWorkMode:加解密算法的工作模式

pParam:加解密算法的控制参数，其中，对于AES, 指针应指向 TEE_CIPHER_CTRL_AES_S；对于AES_CCM 或 AES_GCM, 指针应指向 TEE_CIPHER_CTRL_AES_CCM_GCM_S；对于 DES, 指针应指向TEE_CIPHER_CTRL_DES_S;对于3DES, 指针应指向 TEE_CIPHER_CTRL_3DES_S;对于SM1, 指针应指向TEE_CIPHER_CTRL_SM1_S;对于SM4, 指针应指向 TEE_CIPHER_CTRL_SM4_S

6.4 接口定义

6.4.1 初始化加解密硬件

原型：

```
int32_t TEE_CIPHER_Init()
```

功能：

初始化加解密硬件

参数：

无

返回：

TEE_SUCCESS:成功

其他：失败

6.4.2 去初始化加解密硬件

原型：

```
int32_t TEE_CIPHER_DeInit()
```

功能：

去初始化加解密硬件

参数：

无

返回：

TEE_SUCCESS:成功

其他：失败

6.4.3 创建加解密句柄

原型：

```
int32_t TEE_CIPHER_CreateHandle(uint32_t* phCipher,
        const TEE_CIPHER_ATTS_S *pstCipherAttr);
```

功能：

创建加解密句柄

参数：

stCipherAttr：创建加解密句柄时的配置参数

phCipher：加解密句柄

返回：

TEE_SUCCESS:成功

其他：失败

6.4.4 销毁加解密句柄

原型：

```
int32_t TEE_CIPHER_DestroyHandle(uint32_t hCipher);
```

功能：

销毁加解密句柄

参数：

hCipher：加解密句柄

返回：

TEE_SUCCESS:成功

其他：失败

6.4.5 配置加解密控制信息

原型：

```
int32_t TEE_CIPHER_ConfigHandle(uint32_t hCipher, TEE_CIPHER_CTRL_S* pstCtrl);
```

功能：

配置加解密控制信息

参数：

hCipher：加解密句柄

pstCtrl：配置加解密的控制信息参数

返回：

TEE_SUCCESS:成功

其他：失败

6.4.6 配置加解密控制信息的扩展接口

原型：

```
int32_t TEE_CIPHER_ConfigHandleEx(uint32_t hCipher,
```

TEE_CIPHER_CTRL_EX_S* pstExCtrl)

功能：

配置加解密控制信息的扩展接口

参数：

hCipher：加解密句柄

pstExCtrl：配置加解密的控制信息参数

返回：

TEE_SUCCESS:成功

其他：失败

6.4.7 解密接口

原型：

```
int32_t TEE_CIPHER_Decrypt(uint32_t hCipher,
        uint32_t u32SrcPhyAddr,
        uint32_t u32DestPhyAddr,
        uint32_t u32ByteLength)
```

功能：

进行解密操作。TDES模式下解密的数据长度应当是8的倍数，AES下应当是16的倍数。此外，解密数据长度不能长于0xFFFFF。本次操作完成后，此次操作的向量运算结果会作用于下一次操作，如果要清除向量，需要在下次解密操作之前调用TEE_CIPHER_ConfigHandle重新配置IV(需要设置pstCtrl->stChangeFlags.bit1IV为1)

参数：

hCipher：加解密句柄

u32SrcPhyAddr：待解密数据的源物理地址

u32DestPhyAddr：解密后数据的目的物理地址

u32ByteLength：待解密的数据长度

返回：

TEE_SUCCESS:成功

其他：失败

6.4.8 获取加解密控制信息

原型：

```
int32_t TEE_CIPHER_GetHandleConfig(uint32_t hCipher,
        TEE_CIPHER_CTRL_S* pstCtrl)
```

功能：

获取加解密控制信息

参数：

hCipher：加解密句柄

pstCtrl：加解密的控制信息

返回：

TEE_SUCCESS:成功

其他：失败

7 融合 CA TEE 接口

7.1 接口概述

融合CA是指同时支持DCAS和传统CA终端功能特性的TVOS功能。融合CA TEE接口包括智能卡复位、智能卡通讯等接口。融合CA TEE接口列表如表4所示。

表4 融合CA TEE接口列表

接口	说明	备注
TEE_DSD_SCReset	智能卡复位	
TEE_DSD_SCPBRun	智能卡通讯	TApp通过本接口调用终端的读卡器驱动，进而向智能卡发送指令，获取返回数据。

7.2 常量定义

无。

7.3 数据结构定义

无。

7.4 接口定义

7.4.1 智能卡复位

原型：

```
bool TEE_DSD_SCReset(uint8_t* atr,  uint8_t* len)
```

功能：

TApp要求终端复位智能卡。终端在调用该函数时应设置ATR长度不小于33字节，在获取到正确的ATR后返回实际长度。不允许TEE HAL主动调用，否则易导致TApp异常，出现节目无法观看或机卡对应丢失等错误现象。

参数：

atr：返回ATR的数据的空间

len：返回ATR的长度

返回：

true：复位成功

false：复位失败

7.4.2 智能卡通讯

原型：

```
bool TEE_DSD_SCPBRun(const uint8_t * command, uint16_t commandLen,
        uint8_t * reply, uint16_t * replyLen)
```

功能：

TApp通过本接口调用终端的读卡器驱动，进而向智能卡发送指令，获取返回数据。

参数：

command：待发送的一串命令字节

commandLen：待发送命令字节的长度

reply：返回的数据，TApp分配足够空间，终端驱动将卡返回的结果写在这里

replyLen：返回数据的长度

返回：

true：通讯成功，表示已收到智能卡返回的状态码，包括0x6XXX和0x9XXX

false：通讯失败，表示没有收到卡返回的状态码返回

8 北斗 TEE 接口

北斗TEE接口包括获取北斗软件版本号、获取位置信息、获取定位卫星信号参数、计算两点距离等信息的接口，具体接口函数遵循GY/T 308—2017中B.5的规定。

中 华 人 民 共 和 国
广播电视行业标准
智能电视操作系统
第 6 部分：可信执行环境接口
GY/T 303.6－2018
*
国家新闻出版广电总局广播电视规划院出版发行
责任编辑：王佳梅
查询网址：www.abp2003.cn
北京复兴门外大街二号
联系电话：（010）86093424　　86092923
邮政编码：100866