# 检测客户端接口说明

## 修订记录

| 修订日期 | 修订说明 | 备注 |
|---|---|---|
| 2013/11/26 | V13.0.0.1 | 电能表全检函数、公钥下抽检函数 |
| 2013/12/19 | V13.0.0.2 | a) 对函数说明错误的地方进行了矫正；<br>b) 对新旧电能表兼容性进行了调整和说明；<br>c) 优化了错误代码；<br>d) 增加了对芯片发行信息的验证；<br>e) 红色字体请重点注意； |
| 2013/12/27 | V13.0.0.3 | a) 增加了终端检测函数 |
| 2014/01/27 | V14.1.27.1 | a) 私钥下清零用普通操作员即可，不需授权； |
| 2015/09/11 | V15.0.0.1 | a) 单地址 MAC、组地址 MAC、系统广播 MAC 增加了文件传输 |
| 2016/05/13 | V16.5.13.1 | a) 增加了巡检仪终端检测函数 |
| 2016/10/18 | V16.10.18.1 | b) 增加了面向对象相关检测函数，电能表分散因子更改为表号 |
| 2016/11/21 | V16.11.21.1 | c) 密钥恢复增加了权限，需要密钥恢复 key 才能进行。 |

# 目录

删除的内容: 89

# 1. 前言

使用环境

(1). 服务器版，用于各省电力公司计量中心，使用过程中需要外接检测操作员 USBkey 登录检测服务器，接口函数在 WinSocketServer.dll 中，该 dll 可以自行更改名称后调用。

(2). 直连密码机版，用于厂家电能表和终端生产测试，使用时需要厂家 USBKEY，直连厂家检测密码机，接口函数在 SoketApi.dll 中,该 dll 可以自行更改名称后调用。

# 2. 通用函数

## 2.1. 获取登录服务器权限函数

- 功能描述：

  用于获取登录服务器的权限，兼容 09 版电能表使用的函数。

- 函　　数：

  Int OpenUsbkey();

- 参数说明：

- 函数返回：

  返回 0　　成功

  其他　　失败

## 2.2. 登录服务器函数

- 功能描述：

  用于登录服务器，兼容 09 版电能表使用的函数。

- 函　　数：

```
Int  LgServer(
    char*ip,
    u_short port,
    int  nPwdLen,
    unsigned char *pPwd
);
```

- 参数说明：

| | |
|---|---|
| ip | 字符型 |
| port | 服务器端口号,短整型； |
| nPwdLen | 密码长度,整型； |
| pPwd | USBKEY 密码,无符号字符型。 |

- 函数返回：

| | |
|---|---|
| 返回 0 | 成功 |
| 其他 | 失败 |

## 2. 3. 断开与服务器连接函数

- 功能描述：

断开与服务器连接，兼容 09 版电能表使用的函数。

- 函　　数：

```
Int LgoutServer();
```

- 参数说明：

无

- 函数返回：

| | |
|---|---|
| 返回 0 | 成功 |
| 其他 | 失败 |

## 2.4. 释放服务器登录权限函数

- 功能描述：

  释放服务器登录权限，兼容 09 版电能表使用的函数。

- 函　　数：

  Int ClseUsbkey()

- 参数说明：

  无

- 函数返回：

  返回 0　　成功

  其他　　失败

## 2.5. 创建随机数函数

- 功能描述：

  用于产生随机数，也可以不调用本函数自己产生随机数，新增。

- 函　　数：

  Int CreateRand (

  Int　InRandLen

  char *OutRand1

  )

- 参数说明：

  InRandLen　　8 或 16，表示输出的随机数长度；

  OutRand1　　8 字节或 16 字节随机数；

- 函数返回：

  返回 0　　成功

  其他　　失败

## 2.6. 连接密码机（服务器）函数

- 功能描述：

  连接密码机，用于连接服务器或密码机，新增。

- 函　　数：

```
int   ConnectDevice (
  char *PutIP,
  char *PutPort,
  char *PutCtime
);
```

- 参数说明：

  PutIP　　ip 地址字符型

  PutPort　　密码机端口号,短整型；

  PutCtime　　字符型，单位秒；

- 函数返回：

  等于 0　　成功

  其他　　失败

## 2.7. 断开密码机（服务器）函数

- 功能描述：

  连接密码机，用于断开服务器或密码机连接，新增。

- 函　　数：

```
int   CloseDevice ();
```

- 参数说明：

  无

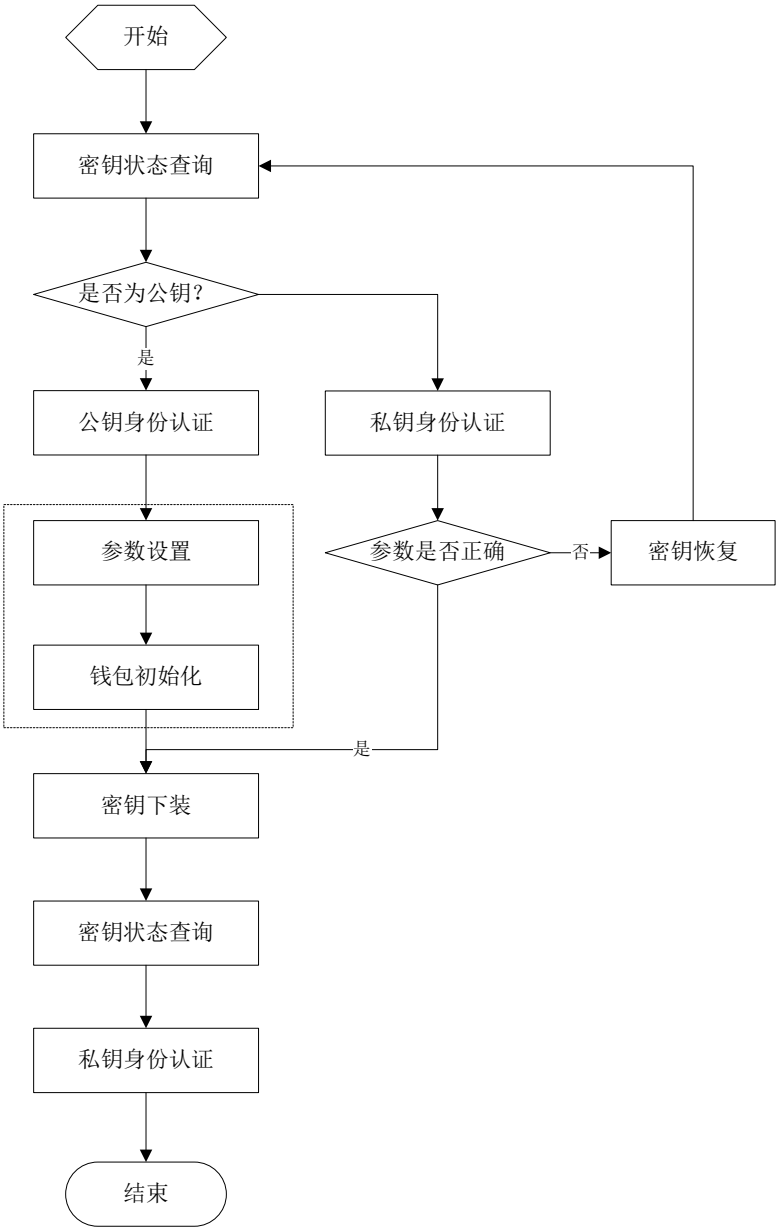
- 函数返回：

| | |
|---|---|
| 等于 0 | 成功 |
| 其他 | 失败 |

## 3. 电能表检测

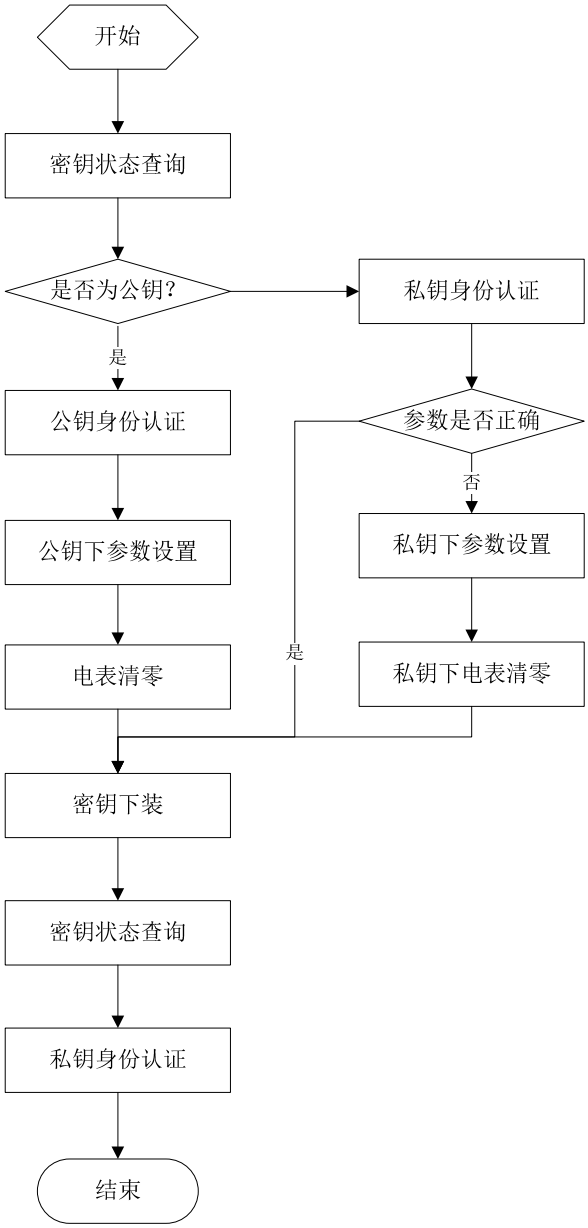
## 3.1. 全检流程

### 3.1.1. 本地表




注：电价设置和钱包初始化可以通过参数预置卡实现。

### 3.1.2. 远程表




## 3.2. 接口说明

特殊授权指对普通操作员 USBKey 进行特殊处理以达到授权目的，经授权后的用户才能取得对应的操作权限。

特殊授权的 USBKey 需要专人负责，制定严格的安全管理制度。

### 3.2.1. 2009 版规范电能表函数

本节函数主要用于兼容 2009 版规范电能表已开发的软件。

#### 3.2.1.1. 身份认证函数

- 功能描述：

从密码机获取随机数以及密文,用于远程身份认证，公钥下函数内部分散因子默认为“0000000000000001”，只用于 2009 版规范电能表。

- 函　　数：

```
Int IdentityAuthentication(
    int Flag,
    char *PutDiv,
    char *OutRand,
    char *OutEndata,
    char *NameId
);
```

- 参数说明：

| | |
|---|---|
| Flag | 整型, 0:公钥状态;1,私钥状态 |
| PutDiv | 输入的分散因子,字符型,长度 16, “0000”+表号 |
| OutRand | 输出的随机数 1,字符型,长度 16 |
| OutEndata | 输出的密文,字符型,长度 16 |
| NameId | 厂家名称，可以为空，主要用于兼容 2009 版电表身份认证 |

- 函数返回：

| | |
|---|---|
| 返回 0 | 成功 |

其他　　失败

#### 3.2.1.2. 控制命令加密函数

- 功能描述：

用户获取控制命令密文。

- 函　　数：

```
Int UserControl(
    int Flag,
    char *PutRand,
    char *PutDiv,
    char *PutEsamNo,
    char *PutData,
    char *OutEndata
);
```

- 参数说明：

| | |
|---|---|
| Flag | 整型, 0:公钥状态;1,私钥状态 |
| PutRand | 输入的随机数 2,字符型,长度 8 |
| PutDiv | 输入的分散因子,字符型,长度 16,“0000”+表号 |
| PutEsamNo | 输入的 ESAM 序列号,复位信息的后 8 字节,字符型,长度 16 |
| PutData | 跳闸或合闸控制命令明文,字符型 |
| OutEndata | 输出的密文,字符型 |

- 函数返回：

返回 0　　成功

其他　　失败

#### 3.2.1.3. 一类参数 MAC 计算函数

- 功能描述：

用于电能表一类参数 MAC 计算函数，。

- 函　　数：

```
Int ParameterUpdate (int Flag,
    char *PutRand,
    char *PutDiv,
    char *PutApdu,
    char *PutData,
    char *OutEndata
);
```

- 参数说明：

| | |
|---|---|
| Flag | 整型, 0:公钥状态;1,私钥状态 |
| PutRand | 输入的随机数 2,字符型,长度 8 |
| PutDiv | 输入的分散因子,字符型,长度 16, “0000”+表号 |
| PutApdu | 一类参数设置的写 Esam 命令头,字符型,长度 10 |
| PutData | 输入的一类参数明文,字符型 |
| OutEndata | 输出的明文数据+MAC 数据,字符型,长度为明文数据长度+8 |

- 函数返回：

| | |
|---|---|
| 返回 0 | 成功 |
| 其他 | 失败 |

#### 3. 2. 1. 4. 二类参数加密函数

- 功能描述：

用于远程二类参数设置加密。

- 函　　数：

```
Int ParameterElseUpdate (
    int Flag,
    char *PutRand,
    char *PutDiv,
```

char *PutApdu,

char *PutData,

char *OutEndata

);

- 参数说明：

| | |
|---|---|
| Flag | 整型, 0:公钥状态;1,私钥状态(需要特殊授权) |
| PutRand | 输入的随机数 2,字符型,长度 8 |
| PutDiv | 输入的分散因子,字符型,长度 16, “0000”+表号 |
| PutApdu | 输入的指令数据,字符型,长度 10 |
| PutData | 输入的二类参数明文,字符型 |
| OutEndata | 输出的密文,字符型 |

- 函数返回：

| | |
|---|---|
| 返回 0 | 成功 |
| 其他 | 失败 |

#### 3. 2. 1. 5. 第一套费率电价设置函数

- 功能描述：

用于第一套费率参数 MAC 计算。

- 函　　数：

Int ParameterUpdate1 (

int Flag,

char *PutRand,

char *PutDiv,

char *PutApdu,

char *PutData,

char *OutEndata

);

- 参数说明：

| | |
|---|---|
| Flag | 整型, 0:公钥状态;1,私钥状态 |
| PutRand | 输入的随机数 2,字符型,长度 8 |
| PutDiv | 输入的分散因子,字符型,长度 16,“0000”+表号 |
| PutApdu | 输入的指令数据,字符型,长度 10 |
| PutData | 输入的第一套费率参数明文,字符型 |
| OutEndata | 输出的明文数据长度+MAC,字符型,长度为明文数据长度+8 |

- 函数返回:

| | |
|---|---|
| 返回 0 | 成功 |
| 其他 | 失败 |

#### 3. 2. 1. 6. 第二套费率电价设置函数

- 功能描述:

该函数用于 2009 版规范中的第二套费率电价设置，也用于 2013 版规范中的备用套电价参数 MAC 计算。

- 函　　数:

```
Int ParameterUpdate2 (
    int Flag,
    char *PutRand,
    char *PutDiv,
    char *PutApdu,
    char *PutData,
    char *OutEndata
);
```

- 参数说明:

| | |
|---|---|
| Flag | 整型, 0:公钥状态;1,私钥状态 |
| PutRand | 输入的随机数 2,字符型,长度 8 |
| PutDiv | 输入的分散因子,字符型,长度 16,“0000”+表号 |
| PutApdu | 输入的指令数据,字符型,长度 10 |

| | |
|---|---|
| PutData | 输入的第二套费率参数或当前套电价参数明文,字符型 |
| OutEndata | 输出的明文数据长度+MAC,字符型,长度为明文数据长度+8 |

- 函数返回:

| | |
|---|---|
| 返回 0 | 成功 |
| 其他 | 失败 |

#### 3.2.1.7. 密钥清零函数

- 功能描述:

获取远程密钥和主控密钥的密钥信息和密钥密文。

- 函　　数:

```
Int ClearKeyInfo (
    int Counter,
    char *PutRand,
    char *PutDiv,
    char *PutEsamNo,
    char * PutKeyinfo1,
    char *OutKey1,
    char *OutKeyinfo1,
);
```

- 参数说明:

| | |
|---|---|
| Flag | 整型,为 0 公钥下清零;为 1 时私钥下清零 |
| PutRand | 输入的随机数 2,字符型,长度 16 |
| PutDiv | 输入的分散因子,字符型,长度 16,“0000”+表号 |
| PutEsamNo | 输入的 ESAM 序列号,复位信息的后 8 字节,字符型,长度 16 |
| PutKeyinfo1 | 输入的主控密钥密钥信息明文,字符型 |
| OutKey1 | 输出的主控密钥密文,字符型,长度 64 |
| OutKeyinfo1 | 输出的主控密钥信息,字符型,长度 8 |

- 函数返回:

| | |
|---|---|
| 返回 0 | 成功 |
| 其他 | 失败 |

#### 3. 2. 1. 8. 获取远程密钥函数

- 功能描述：

  获取远程密钥和主控密钥的密钥信息和密钥密文。

- 函　　数：

```
Int KeyUpdate (
    int Counter,
    char *PutRand,
    char *PutDiv,
    char *PutEsamNo,
    char * PutKeyinfo1,
    char * PutKeyinfo2,
    char * PutKeyinfo3,
    char * PutKeyinfo4,
    char *OutKey1,
    char *OutKeyinfo1,
    char *OutKey2,
    char *OutKeyinfo2,
    char *OutKey3,
    char *OutKeyinfo3,
    char *OutKey4,
    char *OutKeyinfo4
);
```

- 参数说明：

| | |
|---|---|
| Flag | 整型,为 0 时修改;为 1 时恢复,需特殊授权 |
| PutRand | 输入的随机数 2,字符型,长度 16 |

| | |
|---|---|
| PutDiv | 输入的分散因子,字符型,长度 16,“0000”+表号 |
| PutEsamNo | 输入的 ESAM 序列号,复位信息的后 8 字节,字符型,长度 16 |
| PutKeyinfo1 | 输入的主控密钥密钥信息明文,字符型 |
| PutKeyinfo2 | 输入的远程控制密钥信息明文,字符型 |
| PutKeyinfo3 | 输入的二类参数设置密钥信息明文,字符型 |
| PutKeyinfo4 | 输入的远程身份认证密钥信息明文,字符型 |
| OutKey1 | 输出的主控密钥密文,字符型,长度 64 |
| OutKeyinfo1 | 输出的主控密钥信息 MAC,字符型,长度 8 |
| OutKey2 | 输出的远程控制密钥密文,字符型,长度 64 |
| OutKeyinfo2 | 输出的远程控制密钥信息 MAC,字符型,长度 8 |
| OutKey3 | 输出的二类参数设置密钥密文,字符型,长度 64 |
| OutKeyinfo3 | 输出的二类参数设置密钥信息 MAC,字符型,长度 8 |
| OutKey4 | 输出的远程身份认证密钥密文,字符型,长度 64 |
| OutKeyinfo4 | 输出的远程身份认证密钥信息 MAC,字符型,长度 8 |

- 函数返回:

| | |
|---|---|
| 返回 0 | 成功 |
| 其他 | 失败 |

### 3.2.2. 2013 版规范电能表函数

#### 3.2.2.1. 身份认证函数

- 功能描述：

  从密码机获取随机数以及密文,用于远程身份认证。

- 函　　数：

```
Int Meter_Formal_IdentityAuthentication(
    int Flag,
    char *PutDiv,
    char *OutRand,
    char *OutEndata,
);
```

- 参数说明：

| | |
|---|---|
| Flag | 整型, 0:公钥状态;1,私钥状态 |
| PutDiv | 输入的分散因子,字符型,长度 16, “0000”+表号 |
| OutRand | 输出的随机数 1,字符型,长度 16 |
| OutEndata | 输出的密文,字符型,长度 16 |

- 函数返回：

| | |
|---|---|
| 返回 0 | 成功 |
| 其他 | 失败 |

#### 3.2.2.2. 控制命令加密函数

- 功能描述：

  用户获取控制命令密文。

- 函　　数：

```
Int Meter_Formal_UserControl(
```

int Flag,

char *PutRand,

char *PutDiv,

char *PutEsamNo,

char *PutData,

char *OutEndata

);

- 参数说明：

| | |
|---|---|
| Flag | 整型, 0:公钥状态;1,私钥状态(需要特殊授权) |
| PutRand | 输入的随机数 2,字符型,长度 8 |
| PutDiv | 输入的分散因子,字符型,长度 16,“0000”+表号 |
| PutEsamNo | 输入的 ESAM 序列号,复位信息的后 8 字节,字符型,长度 16 |
| PutData | 跳闸或合闸控制命令明文,字符型 |
| OutEndata | 输出的密文,字符型 |

- 函数返回：

| | |
|---|---|
| 返回 0 | 成功 |
| 其他 | 失败 |

#### 3. 2. 2. 3. 一类参数 MAC 计算函数

- 功能描述：

用于电能表一类参数 MAC 计算函数，。

- 函　　数：

Int Meter_Formal_ParameterUpdate (int Flag,

char *PutRand,

char *PutDiv,

char *PutApdu,

char *PutData,

char *OutEndata

);

- 参数说明：

| | |
|---|---|
| Flag | 整型, 0:公钥状态;1,私钥状态(需要特殊授权) |
| PutRand | 输入的随机数 2,字符型,长度 8 |
| PutDiv | 输入的分散因子,字符型,长度 16, “0000”+表号 |
| PutApdu | 一类参数设置的写 Esam 命令头,字符型,长度 10 |
| PutData | 输入的一类参数明文,字符型 |
| OutEndata | 输出的明文数据+MAC 数据,字符型,长度明文数据长度+8 |

- 函数返回：

| | |
|---|---|
| 返回 0 | 成功 |
| 其他 | 失败 |

#### 3. 2. 2. 4. 二类参数加密函数

- 功能描述：

用于远程二类参数设置加密。

- 函　　数：

```
Int Meter_Formal_ParameterElseUpdate (
    int Flag,
    char *PutRand,
    char *PutDiv,
    char *PutApdu,
    char *PutData,
    char *OutEndata
);
```

- 参数说明：

| | |
|---|---|
| Flag | 整型, 0:公钥状态;1,私钥状态(需要特殊授权) |
| PutRand | 输入的随机数 2,字符型,长度 8 |
| PutDiv | 输入的分散因子,字符型,长度 16, “0000”+表号 |

PutApdu 输入的指令数据,字符型,长度 10

PutData 输入的二类参数明文,字符型

OutEndata 输出的密文,字符型

- 函数返回:

返回 0 成功

其他 失败

#### 3. 2. 2. 5. 第一套费率电价设置函数

- 功能描述:

只用于第一套费率参数 MAC 计算,只用于 2009 版规范电能表。

- 函 数:

```
Int Meter_Formal_ParameterUpdate1 (
    int Flag,
    char *PutRand,
    char *PutDiv,
    char *PutApdu,
    char *PutData,
    char *OutEndata
);
```

- 参数说明:

Flag 整型, 0:公钥状态;1,私钥状态(需要特殊授权)

PutRand 输入的随机数 2,字符型,长度 8

PutDiv 输入的分散因子,字符型,长度 16,“0000”+表号

PutApdu 输入的指令数据,字符型,长度 10

PutData 输入的第一套费率参数明文,字符型

OutEndata 输出的明文数据长度+MAC,字符型,长度为明文数据长度+8

- 函数返回:

返回 0 成功

其他　　失败

### 3. 2. 2. 6. 第二套费率电价设置函数

- 功能描述：

该函数用于 2009 版规范中的第二套费率电价设置，也用于 2013 版规范中的备用套电价参数 MAC 计算。

- 函　　数：

```
Int Meter_Formal_ParameterUpdate2 (
    int Flag,
    char *PutRand,
    char *PutDiv,
    char *PutApdu,
    char *PutData,
    char *OutEndata
);
```

- 参数说明：

| | |
|---|---|
| Flag | 整型, 0:公钥状态;1,私钥状态(需要特殊授权) |
| PutRand | 输入的随机数 2,字符型,长度 8 |
| PutDiv | 输入的分散因子,字符型,长度 16,“0000”+表号 |
| PutApdu | 输入的指令数据,字符型,长度 10 |
| PutData | 输入的第二套费率参数或当前套电价参数明文,字符型 |
| OutEndata | 输出的明文数据长度+MAC,字符型,长度为明文数据长度+8 |

- 函数返回：

返回 0　　成功

其他　　失败

### 3. 2. 2. 7. 钱包初始化

- 功能描述：

  用于本地费控电能表钱包数据 MAC 计算

- 函　　数：

```
int    Meter_Formal_InintPurse(
        int    Flag,
        char *PutRand,
        char*PutDiv,
        char*PutData,
        char *OutData
);
```

- 参数说明：

| | |
|---|---|
| Flag | 整型, 0:公钥状态 |
| PutRand | 随机数 2,电表身份认证成功返回, 4 字节 |
| PutDiv | 分散因子,8 字节,“0000”+表号 |
| PutData | 输入的数据明文,包含预置金额 |
| OutData | 输出的数据,预置金额+MAC1+”00000000”+MAC2 |

- 函数返回:

| | |
|---|---|
| 返回 0 | 成功 |
| 其他 | 失败,见错误代码表 |

### 3. 2. 2. 8. 电表清零函数

- 功能描述

  用于远程费控电能表清零。

- 函　　数

```
int    Meter_Formal_DataClear1(
```

int Flag,

char *PutRand,

char *PutDiv,

char *PutData,

char *Outdata

);

- 参数说明

| | |
|---|---|
| Flag | 整型,0:公钥状态;1,私钥状态 |
| PutRand | 随机数 2,电表身份认证成功返回, 4 字节 |
| PutDiv | 分散因子,8 字节,“0000”+表号 |
| PutData | 入参,清零数据 |
| OutData | 20 字节密文； |

- 函数返回:

| | |
|---|---|
| 返回 0 | 成功 |
| 其他 | 失败,见错误代码表 |

#### 3. 2. 2. 9. 事件或需量清零函数

- 功能描述

用于电能表事件或需量清零。

- 函　　数

int Meter_Formal_DataClear2(

int Flag,

char *PutRand,

char *PutDiv,

char *PutData,

char *Outdata

);

- 参数说明

| | |
|---|---|
| Flag | 整型,0:公钥状态;1,私钥状态; |
| PutRand | 随机数 2,电表身份认证成功返回, 4 字节 |
| PutDiv | 分散因子,8 字节,“0000”+表号 |
| PutData | 入参,清零数据 |
| OutData | 20 字节密文； |

- 函数返回:

| | |
|---|---|
| 返回 0 | 成功 |
| 其他 | 失败,见错误代码表 |

#### 3. 2. 2. 10. 红外认证函数

- 功能描述：

用于获取红外认证密文和随机数 2;红外认证前先进行红外查询。

- 函　　数：

```
int    Meter_Formal_InfraredAuth(
        int    Flag,
        char *PutDiv,
        char *PutEsamNo,
        char *PutRand1,
        char *PutRand1Endata,
        char *PutRand2,
        char *OutRand2Endata
);
```

- 参数说明：

| | |
|---|---|
| Flag | 0:  公钥状态 |
| PutDiv | 8 字节分散因子,“0000”+表号 |
| PutEsamNo | 8 字节 ESAM 序列号,电能表红外查询命令返回 |

| | |
|---|---|
| PutRand1 | 8 字节随机数 1,创建随机数函数返回 |
| InRand1Endata | 8 字节随机数 1 密文,电能表红外查询命令返回 |
| PutRand2 | 8 字节随机数 2,电能表红外查询命令返回 |
| OutRand2Endata | 返回 8 字节随机数 2 密文 |

- 函数返回:

| | |
|---|---|
| 返回 0 | 成功 |
| 其他 | 失败,见错误代码表 |

#### 3. 2. 2. 11. 数据回抄

- 功能描述

用于数据回抄 MAC 校验。

- 函　　数

```
int    Meter_Formal_MacCheck (
        int    Flag,
        char *PutRand,
        char *PutDiv ,
        char *PutApdu,
        char *PutData,
        char *PutMac
);
```

- 参数说明

| | |
|---|---|
| Flag | 0: 公钥状态;1,私钥状态; |
| PutRand | 随机数 1 的高 4 字节 |
| PutDiv | 分散因子,8 字节,“0000”+表号 |
| PutApdu | 命令头,5 字节（04D686+起始地址+Len,Len 为数据长度+0x0C） |
| PutData | 数据回抄返回的数据 |
| PutMac | 4 字节数据回抄返回的 MAC |

函数返回：

返回 0　　成功

其他　　失败,见错误代码表

### 3.2.2.12. 密钥更新函数

功能描述：

用于电能表远程密钥更新,2013 标准电能表密钥更新本地表和远程表都采用通信方式完成，共 20 条密钥，需 5 次调用本函数，所得密钥分 5 次下发给电能表。密钥更新需要先抄读芯片发行信息文件数据。

函　　数

```
int     Meter_Formal_KeyUpdateV2 (
        int    PutKeySum ,
        char *PutKeystate,
        char *PutKeyid,
        char *PutRand ,
        char *PutDiv,
        char *PutEsamNo,
        char *PutChipInfor,
        char *OutData
);
```

参数说明：

| | |
|---|---|
| PutKeySum | 密钥总条数,固定为 20 |
| PutKeystate | 密钥状态,”01”,密钥下装;”00”,密钥恢复(需要特殊授权) |
| PutKeyid | 密钥编号,0x00-0x13,每次最多输出 4 条密钥"00010203" |
| PutRand | 随机数 2,电表身份认证成功返回, 4 字节 |
| PutDiv | 8 字节分散因子,“0000”+表号 |
| PutEsamNo | 8 字节 ESAM 序列号 |
| PutChipInfor | 芯片发行信息文件(001A 文件)数据,通过 078001FF 命令从电表 ESAM |

| | |
|---|---|
| | 抄读所得 005AH 字节数据（不含 MAC）； |
| OutData | 出参,4*（4 字节密钥信息+32 字节密钥密文）+4 字节 MAC |

- 函数返回:

| | |
|---|---|
| 返回 0 | 成功 |
| 其他 | 失败,见错误代码表 |

# 4. 终端检测

## 4.1. 检测步骤

### 4.1.1. 终端检测和密钥下装步骤

终端密钥下装、证书更新、功能测试步骤如下：

(1). 读取终端是否启用硬件加密模式，如果否，则进行（AFN=04 H，F5 终端消息认证参数设置），启用硬件加密模式；

(2). 进行（AFN=06H，F11 获取终端 ESAM 信息），获取对称密钥版本、证书状态和离线计数器。

1) 对称密钥版本为“0000000000000000”是第一套密钥，所有函数中 PutState 应为“00”，否则是第二套密钥，PutState 应为“01”；
2) 证书状态为“00”是测试证书，为“01 ”是正式证书；
3) 离线计数器设置默认为 5000（十进制），小于 500 则设为 5000 或更大；

(3). 进行证书切换

后续所有测试都必须在正式证书状态下进行，如果证书状态处于测试状态，则需要将其切换到正式状态。

1) 进行（AFN=06H，F17 内部认证请求）；
2) 调用内外部认证函数（Terminal_Formal_ExternalAuth）；
3) 进行（AFN=06H，F18 外部认证请求）；
4) 调用证书切换函数（Terminal_Formal_CertificateStateChange），进行（AFN=06H，F19 证书状态切换），将证书状态切换到正式证书状态（证书状态可以从测试切换到正式，也可以从正式切换到正式）；

5) 进行（AFN=06H，F11 获取终端 ESAM 信息），获取证书状态和离线计数器，确保证书处于正式状态，否则重新进行状态切换或判定不合格；

(4). 会话协商

1) 调用获取随机数函数 Terminal_Formal_GetR1，获取随机数 R1；

2) 调用会话初始化函数 Terminal_Formal_SessionInitRec，进行（AFN=06H，F12 会话初始化）；

3) 调用会话建立函数 Terminal_Formal_SessionKeyConsult，进行（AFN=06H，F13 会话协商）；

4) 调用会话协商验证函数 Terminal_Formal_SessionConsultVerify，进行会话协商确认；

(5). 对称密钥更新

1) 调用对称密钥更新函数 Terminal_Formal_SymmetricKeyUpdate，进行（AFN=06H，F14 对称密钥更新）；

2) 进行（AFN=06H，F11 获取终端 ESAM 信息），获取对称密钥版本，确保对称密钥版本为第二套，否则重新进行对称密钥更新或判定不合格；

(6). 证书更新：调用证书更新函数 Terminal_Formal_CACertificateUpdate，进行（AFN=06H，F16 CA 证书更新）；

(7). 会话协商：重复第（4）步骤；

(8). 协商恢复测试

1) 调用获取随机数函数 Terminal_Formal_GetR1，获取随机数 R1；

2) 调用会话恢复函数 Terminal_Formal_SessionInitRec，进行（AFN=06H，F12 会话恢复）；

3) 调用会话恢复验证函数 Terminal_Formal_SessionRecoveryVerify，进行会话恢复验证；

(9). 功能测试

1) 单地址任务测试：调用单地址数据 MAC 计算函数 Terminal_Formal_MACVerify，进行 MAC 计算，MAC 计算主要针对 AFNType 为“01”（复位）、“04”（设参）、“05”（控制）、“10”（数据转发）、“0F”（文件传输）；

2) 组广播功能测试：调用组广播数据 MAC 计算函数 Terminal_Formal_GroupBroadcast，进行 MAC 计算，MAC 计算主要针对 AFNType 为“01”（复位）、“04”（设参）、“05”（控制）、“10”（数据转发）、“0F”（文件传输）；

3) 系统广播功能测试：调用系统广播数据 MAC 计算函数 Terminal_Formal_SystemBroadcast，进行 MAC 计算，MAC 计算主要针对 AFNType 为“01”（复位）、“04”（设参）、“05”（控

制）、“10”（数据转发）、“0F”（文件传输）；

(10). 批量任务测试

1) 转加密授权：调用转加密授权函数 Terminal_Formal_ChangeDataAuthorize，进行（AFN=10H，F21 转加密授权）；

2) 批量身份认证测试

a) 调用获取电能表密钥密文函数 Terminal_Formal_GetCipherMeterKey，获取身份认证密文；

b) 调用单地址数据 MAC 计算函数 Terminal_Formal_MACVerify，进行（AFN=10H，F12 批量下发身份认证任务）；

3) 批量对时任务测试

a) 调用任务数据加密函数 Terminal_Formal_EncTaskData，获取对时任务密文；

b) 调用获取电能表密钥密文函数 Terminal_Formal_GetCipherMeterKey，获取对时任务密钥密文；

c) 调用单地址数据 MAC 计算函数 Terminal_Formal_MACVerify，进行（AFN=10H，F12 批量下发对时任务）；

(11). 设置离线计数器：如果离线计数器小于 500，调用离线计数器函数 Terminal_Formal_SetOfflineCounter，进行（AFN=06H，F20 置离线计数器），将离线计数器设置为 5000 或更大；

(12). 确认安全状态

1) 进行（AFN=06H，F11 获取终端 ESAM 信息），获取对称密钥版本、证书状态和离线计数器。对称密钥版本为非“00000000000000000”，证书状态为“01 ”是正式证书；离线计数器设置为 5000 或更大；

2) 读取终端是否启用硬件加密模式，根据要求关闭或设置硬件加密模式（AFN=04H，F5 终端消息认证参数设置）；

说明：其中是密钥下装和证书更新的必要步骤，（8）、（9）、（10）主要针对部分功能进行测试。

### 4.1.2. 巡检仪终端密钥检测和密钥下装步骤

巡检仪终端密钥下装、证书更新、功能测试步骤如下：

(1). 读取终端是否启用硬件加密模式，如果否，则进行（AFN=04H，F5 终端消息认证参数设置），

启用硬件加密模式；

(2). 进行（AFN=06H，**F100 获取终端安全认证信息**），获取芯片序列号、证书序列号、离线计数器、芯片证书状态、对称密钥版本、随机数和 MAC；

1) 调用终端查询 MAC 验证函数 CT_Terminal_Formal_CalVerifyCTTESAMMac 验证 MAC；

2) 对称密钥版本为“0000000000000000”是第一套密钥，所有函数中 PutState 应为“00”，否则是第二套密钥，PutState 应为“01”；

3) 证书状态为“00”是测试证书，为“01 ”是正式证书；

4) 离线计数器设置默认为 5000（十进制），小于 500 则设为 5000 或更大；

(3). 进行证书切换

后续所有测试都必须在正式证书状态下进行，如果证书状态处于测试状态，则需要将其切换到正式状态。

1) 进行（AFN=06H，F17 内部认证请求）；

2) 调用内外部认证函数（Terminal_Formal_ExternalAuth）；

3) 进行（AFN=06H，F18 外部认证请求）；

4) 调用证书切换函数（Terminal_Formal_CertificateStateChange），进行（AFN=06H，F19 证书状态切换），将证书状态切换到正式证书状态（证书状态可以从测试切换到正式，也可以从正式切换到正式）；

5) 进行（AFN=06H，**F100 获取终端安全认证信息**），获取证书状态和离线计数器，确保证书处于正式状态，否则重新进行状态切换或判定不合格；

(4). 会话协商

1) 调用获取随机数函数 Terminal_Formal_GetR1，获取随机数 R1；

2) 调用会话初始化函数 Terminal_Formal_SessionInitRec，进行（AFN=06H，F12 会话初始化）；

3) 调用会话建立函数 Terminal_Formal_SessionKeyConsult，进行（AFN=06H，F13 会话协商）；

4) 调用会话协商验证函数 Terminal_Formal_SessionConsultVerify，进行会话协商确认；

(5). 对称密钥更新

1) 调用对称密钥更新函数 **Terminal_Formal_SymmetricKeyUpdateCT**，进行（AFN=06H，F102 对称密钥更新）；

2) 进行（AFN=06H，F11 获取终端 ESAM 信息），获取对称密钥版本，确保对称密钥版本为

第二套，否则重新进行对称密钥更新或判定不合格；

(6). 证书更新：调用证书更新函数 Terminal_Formal_CACertificateUpdate，进行（AFN=06H，F16 CA 证书更新）；

(7). 会话协商：重复第（4）步骤；

**(8). 回路状态巡检仪密钥下装和功能测试**

1) 进行（AFN=06H，F101 获取 CT 模块安全认证信息），获取 CT 模块 ESAM 序列号、CT 模块当前密钥状态、双向认证状态、CT 随机数和 MAC；

2) 调用巡检仪查询 MAC 验证函数 CT_Terminal_Formal_CalVerifyCTESAMMac 验证 MAC；

3) 如果密钥状态为“00000000”，InFlag=0，则调用函数 CT_Terminal_Formal_GetCTESAMKey 进行（AFN=06H，F102 CT 模块 ESAM 密钥更新），更新回路状态巡检仪密钥；如果为“0000007F”，则无需更新；

4) 进行（AFN=06H，F101 获取 CT 模块安全认证信息），获取 CT 模块 ESAM 序列号、CT 模块当前密钥状态、双向认证状态、CT 随机数和 MAC；

5) 如果当前密钥状态为“0000007F”，则 InFlag=1，调用巡检仪查询 MAC 验证函数 CT_Terminal_Formal_CalVerifyCTESAMMac 验证 MAC，成功则密钥更新完成；

6) 调用函数 CT_Terminal_Formal_CalCTESAMMac，进行（AFN=14F，F2，CT 变比参数设置）；

7) 进行（AFN=1A，F2，CT 变比参数查询），调用函数 CT_Terminal_Formal_CalVerifyCTESAMMac 验证 MAC；

8) 进行（AFN=1A，F100，主站读取 CT 模块状态；AFN=0C ，F50，电流回路状态查询；AFN=0A，F153，电流回来使能参数查询），调用函数 CT_Terminal_Formal_CalVerifyCTTESAMMac 验证 MAC；进行（AFN=0A，F154，电流变比参数查询），调用函数 CT_Terminal_Formal_CalVerifyCTESAMMac 验证 MAC；

(9). 终端协商恢复测试

1) 调用获取随机数函数 Terminal_Formal_GetR1，获取随机数 R1；

2) 调用会话恢复函数 Terminal_Formal_SessionInitRec，进行（AFN=06H，F12 会话恢复）；

3) 调用会话恢复验证函数 Terminal_Formal_SessionRecoveryVerify，进行会话恢复验证；

(10). 终端功能测试

1) 单地址任务测试：调用单地址数据 MAC 计算函数 Terminal_Formal_MACVerify，进行 MAC 计算，MAC 计算主要针对 AFNType 为“01”（复位）、“04”（设参）、“05”（控制）、“10”（数据转发）、“0F”（文件传输）；

2) 组广播功能测试：调用组广播数据 MAC 计算函数 Terminal_Formal_GroupBroadcast，进行 MAC 计算，MAC 计算主要针对 AFNType 为“01”（复位）、“04”（设参）、“05”（控制）、“10”（数据转发）、“0F”（文件传输）；

3) 系统广播功能测试：调用系统广播数据 MAC 计算函数 Terminal_Formal_SystemBroadcast，进行 MAC 计算，MAC 计算主要针对 AFNType 为“01”（复位）、“04”（设参）、“05”（控制）、“10”（数据转发）、“0F”（文件传输）；

(11). 终端批量任务测试

1) 转加密授权：调用转加密授权函数 Terminal_Formal_ChangeDataAuthorize，进行（AFN=10F，F21 转加密授权）；

2) 批量身份认证测试

a) 调用获取电能表密钥密文函数 Terminal_Formal_GetCipherMeterKey，获取身份认证密文；

b) 调用单地址数据 MAC 计算函数 Terminal_Formal_MACVerify，进行（AFN=10F，F12 批量下发身份认证任务）；

3) 批量对时任务测试

a) 调用任务数据加密函数 Terminal_Formal_EncTaskData，获取对时任务密文；

b) 调用获取电能表密钥密文函数 Terminal_Formal_GetCipherMeterKey，获取对时任务密钥密文；

c) 调用单地址数据 MAC 计算函数 Terminal_Formal_MACVerify，进行（AFN=10F，F12 批量下发对时任务）；

(12). 设置离线计数器：如果离线计数器小于 500，调用离线计数器函数 Terminal_Formal_SetOfflineCounter，进行（AFN=06H，F20 置离线计数器），将离线计数器设置为 5000 或更大；

(13). 确认安全状态

1) 进行（AFN=06H，F11 获取终端 ESAM 信息），获取对称密钥版本、证书状态和离线计数器。对称密钥版本为非“00000000000000000”，证书状态为“01 ”是正式证书；离线计数器设置为 5000 或更大；

2) 读取终端是否启用硬件加密模式，根据要求关闭或设置硬件加密模式（AFN=04H，F5 终端消息认证参数设置）；

说明：其中是密钥下装和证书更新的必要步骤，（9）、（10）、（11）主要针对部分功能进行测试。

## 4.2. 采集终端函数

此终端检测函数内部协商产生的会话密钥都使用全局变量，开发程序比较方便，但无法支持多线程调用。

如果需要多线程调用有两种方法：

方法一：每个线程复制一个新的动态库并更改名称，即保证每个线程使用不同的动态库。

方法二：使用另外一套动态库，该动态库采用会话密钥外部输入输出方式，但开发检测程序需要通过写文件或存数据库的方式把每块终端协商产生的会话密钥外部存储，以供后续函数使用。

### 4.2.1. 登录服务器函数

- 功能描述：

用于登录服务器，后续函数调用必须先调用此函数。

服务器连接如果在 10 分钟内没有活动，会自动断开，建议每个连接增加一个心跳功能，通过调用电表身份认证函数保证连接处于活动状态。

- 函　　数：

```
ConnectDevice(
    char *PutIP,
    char *PutPort,
    char *PutCtime
);
```

- 参数说明：

| | |
|---|---|
| PutIP | 字符型 |
| PutPort | 服务器端口号,短整型； |
| PutCtime | 字符型，单位秒； |

- 函数返回：

  返回 0　　成功

  其他　　失败

#### 4.2.2. 断开服务器连接函数

- 功能描述：

  用于断开服务器连接。

- 函　　数：

  CloseDevice(

  );

- 参数说明：

  无；

- 函数返回：

  返回 0　　成功

  其他　　失败

#### 4.2.3. 获取随机数

- 功能描述：

用于获取随机数 1。

- 函　　数：

```
int Terminal_Formal _GetR1 (
    char *OutR1
);
```

- 参数说明：

OutR1　　16 字节随机数；

- 函数返回：

返回 0　　成功

其他　　失败，见错误代码表

### 4.2.4. 会话初始化或恢复

- 功能描述：

用于会话初始化或会话恢复。

- 函　　数：

```
int   Terminal_Formal_SessionInitRec(
      char *PutState,
      char *PutTESAMNo,
      char* PutVersionNum,
      char* PutSessionID ,
      char *PutR1,
      char* OutMasterCertificate,
      char *OutEncR1,
      char *OutMac,
      char *OutSign1
);
```

- 参数说明：

| | |
|---|---|
| PutState | 对称密钥状态，　00--第一套密钥，01--第二套密钥； |
| PutTESAMNo | TESAM 序列号，8 字节； |
| PutVersionNum | 版本号，1 字节； |
| PutSessionID | 会话 ID；1 字节；00--新建注册，01--恢复 |
| PutR1 | 随机数 1，字符型，16 字节； |
| OutMasterCertificate | 主站证书(大于 1K，小于 2K)； |
| OutEncR1 | 随机数 1 密文，16 字节； |
| OutMac | Mac，4 字节； |

OutSign1　　签名 64 字节；

函数返回：

返回 0　　成功

其他　　失败，见错误代码表

### 4.2.5. 会话协商

功能描述：

用于会话密钥协商。

函　　数：

```
int   Terminal_Formal_ SessionKeyConsult (
      char *PutState,
      char *PutTESAMNo,
      char * PutVersionNum,
      char * PutSessionID ,
      char * PutCRLCertificateNo ,
      char *PutMasterCertificateNo
      char * PutTerminalCertificate,
      char * PutEncR2,
      char *PutSign2,
      char *PutR1,
      char* OutEncM1,
      char * OutSign3,
      char * OutMac2,
      char * OutSign4
);
```

参数说明：

PutState　　芯片状态，　00--第一套密钥，测试证书；01--第二套密钥，正式证书；11--第二套密钥，正式证书交易状态（正式　证书更新后

使用）。调用此函数时，对称密钥版本全’0’时，证书状态必须是”00”， 对称密钥版本非全’0’时，证书状态必须是”01”，否则须将证书状态切换到对应的状态）。

| | |
|---|---|
| PutTESAMNo | TESAM 序列号，8 字节； |
| PutVersionNum | 1 字节版本； |
| PutSessionID | 1 字节会话 ID； |
| PutCRLCertificateNo | 16 字节 CRL 证书序列号； |
| PutMasterCertificateNo | 16 字节主站证书序列号； |
| PutTerminalCertificate | 终端证书； |
| PutEncR2 | 16 字节 R2（随机书 2）密文； |
| PutSign2 | 64 字节签名； |
| PutR1 | 16 字节随机数 1； |
| OutEncM1 | 会话密钥密文,113 字节； |
| OutSignData | 主站证书验证码，97 字节； |
| OutMac2 | Mac2，字节； |
| OutSign3 | 签名，64 字节； |

- 函数返回：

| | |
|---|---|
| 返回 0 | 成功 |
| 其他 | 失败，见错误代码表 |

### 4.2.6. 会话协商验证

- 功能描述：

用于会话密钥协商验证。

- 函　　数：

```
int   Terminal_Formal_SessionConsultVerify (
      char* PutR3,
```

char *PutMac3

);

- 参数说明：

| | |
|---|---|
| PutR3 | 16 字节随机数 3； |
| PutMac3 | 4 字节 MAC； |

- 函数返回：

| | |
|---|---|
| 返回 0 | 成功 |
| 其他 | 失败，见错误代码表 |

### 4.2.7. 会话恢复验证

- 功能描述：

用于会话密钥恢复验证。

- 函　　数：

```
int   Terminal_Formal_SessionRecoveryVerify (
      char *PutState,
      char *PutTESAMNo,
      char * PutVersionNum,
      char * PutSessionID ,
       char * PutEncR2 ,
       char * PutR3,
      char * PutMac
);
```

- 参数说明：

| | |
|---|---|
| PutState | 对称密钥状态， 00--第一套密钥，01--第二套密钥； |
| PutTESAMNo | TESAM 序列号，8 字节； |
| PutVersionNum | 1 字节版本； |
| PutSessionID | 1 字节会话 ID； |

| | |
|---|---|
| PutEncR2 | 16 字节 R2（随机书 2）密文； |
| PutR3 | 16 字节随机数 3； |
| PutMac | 4 字节 MAC； |

函数返回：

| | |
|---|---|
| 返回 0 | 成功 |
| 其他 | 失败，见错误代码表 |

### 4.2.8. 单地址数据 MAC 计算

功能描述：

用于单地址数据 MAC 计算。

函　　数：

```
int   Terminal_Formal_MACVerify (
      char *PutState,
      char *PutTESAMNo,
      char *PutFnType,
      char *PutData,
      char *OutMac
);
```

参数说明：

| | |
|---|---|
| PutState | 对称密钥状态， 00--第一套密钥，01--第二套密钥； |
| PutTESAMNoTESAM | 序列号，字符型，8 字节； |
| PutFnType | 命令类型，1 字节：“01”，复位；“04”，设参；“05”，控制；“10”数据转发；“0F”，文件传输，TESAM 所用密钥为“24”； |
| PutData | 计算 MAC 的数据； |
| OutMac | Mac，4 字节； |

函数返回：

| | |
|---|---|
| 返回 0 | 成功 |

其他　　失败，见错误代码表

### 4.2.9. 内外部认证

- 功能描述：

用于内外部认证。

- 函　　数：

```
int   Terminal_Formal_ExternalAuth (
      char *PutState,
      char * PutTESAMNo,
      char * PutR4,
      char * PutEncR4,
      char * PutR5,
      char *OutEncR5
);
```

- 参数说明：

| | |
|---|---|
| PutState | 对称密钥状态 00--第一套密钥，01—第二套密钥； |
| PutTESAMNo | TESAM 序列号，8 字节； |
| PutR4 | 随机数 4，16 字节，可通过产生随机数函数产生； |
| PutEncR4 | 随机数 4 密文，终端返回； |
| PutR5 | 随机数 5，终端返回 16 字节； |
| OutEncR5 | 随机数 5 密文，16 字节； |

- 函数返回：

返回 0　　成功

其他　　失败，见错误代码表

### 4.2.10. 证书状态切换

- 功能描述：

用于证书状态切换。

- 函　　数：

```
int   Terminal_Formal_CertificateStateChange (
      char *PutState,
      char *PutTESAMNo,
      char *PutCertificateState,
      char *PutR6,
      char *OutEncR6,
      char *OutMac
);
```

- 参数说明：

| | |
|---|---|
| PutState | 对称密钥状态 00--第一套密钥，01—第二套密钥； |
| PutTESAMNo | TESAM 序列号，8 字节； |
| PutCertificateState | 证书状态，00—切换到测试证书，01—切换到正式证书； |
| PutR6 | 随机数 6，16 字节； |
| OutEncR6 | 随机数 6 密文，16 字节； |
| OutMac | Mac，4 字节； |

- 函数返回：

| | |
|---|---|
| 返回 0 | 成功 |
| 其他 | 失败，见错误代码表 |

### 4.2.11. 设置离线计数器

- 功能描述：

用于证书状态切换。

- 函　　数：

```
int   Terminal_Formal_SetOfflineCounter (
      char *PutState,
      char *PutTESAMNo,
      char *PutCounter,
      char *OutEncCounter
);
```

- 参数说明：

| | |
|---|---|
| PutState | 对称密钥状态 00--第一套密钥，01—第二套密钥； |
| PutTESAMNo | TESAM 序列号，8 字节； |
| PutCounter | 离线计数器数值，4 字节； |
| OutEncCounter | 密文数据，20 字节； |

- 函数返回：

| | |
|---|---|
| 返回 0 | 成功 |
| 其他 | 失败，见错误代码表 |

### 4.2.12. 转加密授权

- 功能描述：

用于转加密授权。

- 函　　数：

```
int   Terminal_Formal_ChangeDataAuthorize (
      char *OutData
);
```

- 参数说明：

| | |
|---|---|
| OutData | 32 字节转加密授权数据； |

函数返回：

返回 0　　成功

其他　　失败，见错误代码表

### 4.2.13. 获取电表密钥密文

功能描述：

用于转加密授权。

函　　数：

```
int   Terminal_Formal_GetCipherMeterKey (
      char *PutMeterState,
      char *PutMeterNo,
      int     PutTaskType,
      char * OutMeterEncKey
);
```

参数说明：

| | |
|---|---|
| PutMeterState | 电表密钥状态，1 字节，0--公开密钥；01—交易密钥； |
| PutMeterNo | “0000”+电表表号，共 8 字节； |
| PutTaskType | 任务类型：0，身份认证任务；1，对时任务；2，红外认证； |
| OutMeterEncKey | 对应任务和表号的密钥密文，32 字节，如果多表，调用多次； |

函数返回：

返回 0　　成功

其他　　失败，见错误代码表

### 4.2.14. 任务数据加密函数

功能描述：

用于任务数据加密。

- 函　　数：

```
int   Terminal_Formal_EncTaskData (
      int   PutInDataType,
      char *PutTaskData,
      char * OutTaskData
);
```

- 参数说明：

| | |
|---|---|
| PutInDataType | 输入数据类型，对时任务当前为 0； |
| PutTaskData | 任务数据，字节数小于 2k； |
| OutTaskData | 输出的任务密文； |

- 函数返回：

| | |
|---|---|
| 返回 0 | 成功 |
| 其他 | 失败，见错误代码表 |

### 4.2.15. 组广播数据 MAC 计算

- 功能描述：

用于组广播数据 MAC 计算。

- 函　　数：

```
int   Terminal_Formal_GroupBroadcast (
      char *PutState,
      char *PutTESAMNo,
      char *PutFnType,
      int    PutOutDataType,
      char *PutGroupAdrass,
      char *PutMtime,
```

```
    char *PutBroadcastData,
    char *OutMac
);
```

- 参数说明：

| | |
|---|---|
| PutState | 对称密钥状态 00--第一套密钥，01—第二套密钥； |
| PutTESAMNo | TESAM 序列号，8 字节； |
| PutFnType | 命令类型，1 字节：“01”，复位；“04”，设参；“05”，控制；“10”，数据转发；“0F”，文件传输，TESAM 所用密钥为“14”，向量为“1E”； |
| PutOutDataType | 输出数据类型：0，输出为明文数据的 MAC；1，输出为密文数据；2，输出数据为密文+MAC； |
| PutGroupAdrass | 组地址，2 字节； |
| PutMtime | 6 字节，默认“130202224622”； |
| PutBroadcastData | 广播数据； |
| OutMac | 4 字节 Mac； |

- 函数返回：

| | |
|---|---|
| 返回 0 | 成功 |
| 其他 | 失败，见错误代码表 |

### 4.2.16. 系统广播数据 MAC 计算

- 功能描述：

用于系统广播数据 MAC 计算。

- 函　　数：

```
int   Terminal_Formal_SystemBroadcast (
    char *PutState,
    char *PutTESAMNo,
    char *PutFnType,
    int     PutOutDataType,
```

char *PutGroupAdrass,

char *PutMtime,

char *PutBroadcastData,

char *OutMac

);

参数说明：

| | |
|---|---|
| PutState | 对称密钥状态 00--第一套密钥，01—第二套密钥； |
| PutTESAMNo | TESAM 序列号，8 字节； |
| PutFnType | 命令类型，1 字节：“01”，复位；“04”，设参；“05”，控制；“10”，数据转发；“0F”，文件传输，TESAM 所用密钥为“15”，向量为“1F”； |
| PutOutDataType | 输出数据类型：0，输出为明文数据的 MAC；1，输出为密文数据；2，输出数据为密文+MAC； |
| PutGroupAdrass | 组地址，2 字节； |
| PutMtime | 6 字节，默认“130202224622”； |
| PutBroadcastData | 广播数据； |
| OutMac | 4 字节 Mac； |

函数返回：

| | |
|---|---|
| 返回 0 | 成功 |
| 其他 | 失败，见错误代码表 |

### 4.2.17. 对称密钥修改

功能描述：

用于不带巡检仪功能终端的对称密钥更新，如果是巡检仪终端对称密钥更新，则使用函数 3.3.1。

函　　数：

int　Terminal_Formal_SymmetricKeyUpdate (

char *PutState,

char *PutTESAMNo,

char *OutKeyNum,

char *OutEncKeyData

);

- 参数说明：

| | |
|---|---|
| PutState | 00--修改到第一套密钥；01--修改到第二套密钥； |
| PutTESAMNo | TESAM 序列号，8 字节； |
| OutKeyNum | 当前更新的密钥总条数**(33H)**，16 进制字符串； |
| OutEncKeyData | 密钥密文，长度为 32* OutKeyNum； |

- 函数返回：

| | |
|---|---|
| 返回 0 | 成功 |
| 其他 | 失败，见错误代码表 |

### 4.2.18. 证书更新

- 功能描述：

用于证书更新。

- 函　　数：

int　Terminal_Formal_CACertificateUpdate (

char *PutCertificateState,

char * PutCertificateType,

char *OutEncCertificateData

);

- 参数说明：

| | |
|---|---|
| PutCertificateState | 00--修改测试证书；01--修改正式证书到交易状态；<br>11--恢复正式证书到初始状态； |

PutCertificateType　　证书类型，1 字节； 01--CRL 证书，其他保留；

OutEncCertificateData　　证书密文，长度小于 2k；

函数返回：

返回 0　　成功

其他　　失败，见错误代码表

## 4.3. 回路状态巡检仪函数

巡检仪终端在原有终端上增加了回路状态巡检仪，即此章节函数为带巡检仪功能终端新增的专有函数。

### 4.3.1. 巡检仪终端对称密钥更新函数

功能描述：

用于巡检仪终端对称密钥更新（AFN=06H，F14），区别于 3.2.17，即不带巡检仪功能使用 3.2.17 中的函数。

函　　数：

```
int   Terminal_Formal_SymmetricKeyUpdateCT (
      char *PutState,
      char *PutTESAMNo,
      char *OutKeyNum,
      char *OutEncKeyData
);
```

参数说明：

PutState　　00--修改到第一套密钥；01--修改到第二套密钥；

PutTESAMNo　　TESAM 序列号，8 字节；

OutKeyNum　　当前更新的密钥总条数（3BH），16 进制字符串；

OutEncKeyData　　密钥密文，长度为 32* OutKeyNum；

- 函数返回：

返回 0　　成功

其他　　失败，见错误代码表

### 4.3.2. 回路状态巡检仪密钥更新函数

- 功能描述：

用于获取巡检仪终端中回来状态巡检仪对称密钥（AFN=06H，F102）。

- 函　　数：

```
int    CT_Terminal_Formal_GetCTESAMKey (
            int    InFlag,
            char *InKeyid,
            char *nCTTESAMNo,
            char *InCTESAMNo,
            char *InRand,
            char *OutKey,
            char *OutMac
);
```

- 参数说明：

| | |
|---|---|
| InFlag | 0: 密钥恢复到初始状态；1，密钥修改到正式状态 |
| InKeyid | 密钥索引，目前共 8 条密钥，分两包更新，每包不超过 4 条，如分别传输入密钥"00010203","04050607" |
| InCTTESAMNo | 8 字节终端 ESAM 序列号(从终端获取) |
| InCTESAMNo | 8 字节 CTESAM 序列号 |
| InRand | 8 字节随机数； |
| OutKey | （4+32）*N 字节密钥密文; |
| OutMac | 4 字节 MAC; |

- 函数返回：

返回 0　　成功

其他　　失败,见错误代码表

### 4.3.3. 主站对向巡检仪下发报文的 MAC 计算函数

- 功能描述：

用于回路巡检仪 CT 变比参数设置（AFN=14F，F2）和回路巡检仪程序升级（AFN=0FH，文件标识=08H）时巡检仪数据单元 MAC（MAC1）计算。

- 函　　数：

```
int    CT_Terminal_Formal_CalCTESAMMac (
    int InFlag,
    char *InCTESAMNo,
    char *InRand,
    char *InData,
    char *OutMac
);
```

- 参数说明：

| | |
|---|---|
| InFlag | CT 模块安全认证信息中的密钥状态，如果为“00000000”，则 InFlag =0，否则 InFlag =0； |
| InCTESAMNo | 8 字节 CTESAM 序列号 |
| InRand | 8 字节随机数 |
| InData | 明文数据 |
| OutMac | 4 字节 MAC |

- 函数返回：

返回 0　　成功

其他　　失败,见错误代码表

### 4.3.4. 主站对向终端下发报文的 MAC 计算函数

- 功能描述：

用于巡检仪终端使能控制参数设置（AFN=14F，F1）、巡检仪程序远程升级（AFN=0FH，文件标识=08H）时终端数据单元 MAC（MAC2）计算。

- 函　　数：

```
int    CT_Terminal_Formal_CalCTTESAMMac (
    char *PutState,
    char *InCTTESAMNo,
    char *InRand,
    char *InData,
    char *OutMac
);
```

- 参数说明：

| | |
|---|---|
| PutState | 对称密钥状态 00--第一套密钥，01—第二套密钥； |
| InCTTESAMNo | 8 字节 CTTESAM 序列号 |
| InRand | 8 字节随机数 |
| InData | 明文数据 |
| OutMac | 4 字节 MAC |

- 函数返回：

| | |
|---|---|
| 返回 0 | 成功 |
| 其他 | 失败,见错误代码表 |

### 4.3.5. 巡检仪查询数据的 MAC 验证函数

- 功能描述：

用于验证获取 CT 模块安全信息（AFN=06，F101）、CT 变比参数查询（AFN=1A，F2）时 MAC 验证。

- 函　　数：

```
int    CT_Terminal_Formal_CalVerifyCTESAMMac (
    int InFlag,
    char *InCTESAMNo,
    char *InRand,
    char *InData,
    char *InMac
);
```

- 参数说明：

| | |
|---|---|
| InFlag | CT 模块安全认证信息中的密钥状态，如果为“00000000”，则 InFlag =0，否则 InFlag =0； |
| InCTESAMNo | 8 字节 CTESAM 序列号 |
| InRand | 8 字节随机数 |
| InData | 明文数据 |
| InMac | 4 字节 MAC |

- 函数返回：

| | |
|---|---|
| 返回 0 | 成功 |
| 其他 | 失败,见错误代码表 |

### 4.3.6. 终端查询数据的 MAC 验证函数

- 功能描述：

用于获取终端安全认证信息（AFN=06，F100）、终端使能控制参数查询（AFN=1A，F1）、主站读取 CT 模块状态（AFN=1A，F100）时 MAC 验证。

- 函　　数：

```
int    CT_Terminal_Formal_CalVerifyCTTESAMMac (
    char *PutState,
```

```
    char *InCTTESAMNo,
    char *InRand,
    char *InData,
    char *InMac
);
```

- 参数说明：

| | |
|---|---|
| PutState | 对称密钥状态 00--第一套密钥，01—第二套密钥 |
| InCTTESAMNo | 8 字节 CTTESAM 序列号 |
| InRand | 8 字节随机数 |
| InData | 明文数据 |
| InMac | 4 字节 MAC |

- 函数返回：

| | |
|---|---|
| 返回 0 | 成功 |
| 其他 | 失败,见错误代码表 |

# 5. 超高频标签

## 5.1. 密钥更新函数

- 功能描述：

  用于获取互感器标签正式密钥。

- 函　　数

```
int     RDID_Formal_RFIDChangeKey (
        char *PutKeyState,
        char *PutDiv,
        char *OutKey)
);
```

- 参数说明：

| | |
|---|---|
| PutKeystate | 密钥状态,”00”,初始密钥；”01”,更新后的密钥； |
| PutDiv | 8 字节分散因子，此处用应用序列号； |
| OutKey | 出参，16 字节。 |

- 函数返回:

| | |
|---|---|
| 返回 0 | 成功 |
| 其他 | 失败,见错误代码表 |

## 5.2. 数据加密和 MAC 计算函数

- 功能描述：

  用于对互感器参数信息进行加密和 MAC 计算。

- 函　　数

```
int     RDID_Formal_RFIDEncrptData (
        char *PutKeyState,
```

char *PutDiv,

char *PutData,

char *PutOPInfor,

char *OutEncData

char *OutMAC2)

);

- 参数说明：

| | |
|---|---|
| PutKeystate | 密钥状态,默认”01”； |
| PutDiv | 8 字节分散因子，此处用应用序列号； |
| PutData | 互感器参数明文数据（二次负荷上限值、电流比、准确度等级、额定一次电流扩大倍数、仪表保安系数）； |
| PutOPInfor | 检定人员信息（检定人员、检定结果、检定时间、预留位）； |
| OutEncData | 出参，16 字节密文数据； |
| OutMAC2 | 4 字节 MAC。 |

- 函数返回：

| | |
|---|---|
| 返回 0 | 成功 |
| 其他 | 失败,见错误代码表 |

## 5. 3. 数据解密和 MAC 验证函数

- 功能描述：

用于对标签中的互感器参数信息密文数据解密和 MAC 验证。

- 函　　数

int　RDID_Formal_RFIDDisEncrptData (

char *PutKeyState,

char *PutDiv,

char *PutEncData,

```
    char *PutOPInfor,
    char *PutMAC,
    char *OutData)
);
```

- 参数说明：

| | |
|---|---|
| PutKeystate | 密钥状态,默认”01”； |
| PutDiv | 8 字节分散因子，此处用应用序列号； |
| PutEncData | 16 字节互感器参数密文数据； |
| PutOPInfor | 检定人员信息（检定人员、检定结果、检定时间和保留数据） |
| PutMAC | 4 字节 MAC； |
| OutData | 出参，互感器参数明文数据。 |

- 函数返回：

| | |
|---|---|
| 返回 0 | 成功 |
| 其他 | 失败,见错误代码表 |

## 5.4. 数据校验函数

- 功能描述：

用于对标签中的数据进行 MAC 校验。

- 函　　数

```
int    RDID_Formal_RFIDCheckData  (
         char *InKeyState,
         int    InDataNo,
         char *PutDiv,
         char *PutData,
         char *PutMac
);
```

- 参数说明：

| | |
|---|---|
| PutKeystate | 密钥状态，默认”01”,密钥下装； |
| PutDataNo | MAC 编号： InDataNo=1， MAC1； InDataNo=2， MAC3； InDataNo=3，MAC3；InDataNo=4，MAC4 |
| PutDiv | 8 字节分散因子，此处用应用序列号； |
| PutData | 明文数据； |
| PutMac | 4 字节 MAC。 |

- 函数返回：

| | |
|---|---|
| 返回 0 | 成功 |
| 其他 | 失败,见错误代码表 |

# 6. 面向对象

## 6.1. 数据字典

| 名称 | 意义 | 长度（Byte） | 备注 |
|---|---|---|---|
| i* | 代表入参，整型 | | 整型无长度 Byte |
| c* | 代表入参，char* | | 有长度 Byte |
| cOut* | 代表出参，char* | | 有长度 Byte |
| iKeyState | 对称密钥状态：<br>0，出厂密钥；<br>1，正式密钥 | | 1、根据对称密钥版本判断，密钥版本是“7FFFFFFF**********”，则 iKeyState=1，否则 iKeyState=0；(*代表值不确定，终端和电能表有区别)<br>2、采集主站调用时 iKeyState=1，所读取到的密钥版本必须是“7FFFFFFF**********”，否则代表设备没有进行密钥下装； |
| cTESAMID | TESAM 序列号 | 8 | 终端安全芯片序列号，从芯片中读取，系统须存储 |
| cESAMID | ESAM 序列号 | 8 | 电表安全芯片序列号，从芯片中读取，系统须存储 |

| | | | |
|---|---|---|---|
| cMeterNo | 电表表表号 | 8 | 不足 8 字节，前补”0”； |
| cASCTR | 应用会话计数器 | 4 | 可从芯片读取，系统须存储，每次会话必须必上次大 1 |
| cOutRSTCTR | 主动上报计数器 | 4 | 主动上报验证成功后函数返回，系统须存储 |
| cOutSessionKey | 会话密钥 | 176 | 建立会话成功后密码机产生的会话密钥，系统须存储 |
| cSessionKey | 会话密钥 | 176 | 建立会话成功后系统存储的会话密钥 |
| cTerminalCa | 终端证书 | N〉1000 | 可从芯片读取，系统须存储，终端会话时使用 |
| iOperateMode | 操作模式 | | iOperateMode =1：明文+Mac<br>iOperateMode= 2：密文<br>iOperateMode= 3：密文+MAC |
| cOutSID | 安全标识 | 4 | |
| cOutAddData, | SID 的附加数据 | 不小于 2 | |
| cOutData | 数据 | | 明文或密文数据 |
| cOutMAC | 数据校验码 | 4 | 根据 iOperateMode 确定是否存在 |
| 会话协商 | | | (1). 建立应用连接时进行；<br>(2). 终端采用数字签名连接认证机制，电能表采用对称密码连接认证机制。<br>(3). 抄读数据、主动上报和广播不进行会话协商也可以加解密数据，即无需建立应用连接； |

## 6.2. 主站调用接口测试流程

测试流程主要用于全性能检测，需要分别对出厂密钥下的功能和正式密钥下的功能进行检测。

(1). 读取设备信息(COS 版本，芯片序列号、对称密钥版本、证书状态、证书、表号等)；

根据对称密钥版本确定 iKeyState 值，版本=“7FFFFFFF**********”，则 iKeyState=1，否则 iKeyState=0，(*代表值不确定，终端和电能表有区别，所以无需对后续的每个字符进行判断)；

(2). 建立应用连接

1) 调用会话协商机函数（Obj_*_Formal_InitSession），产生密文 1 和客户机签名 1，组织建立应用连接报文并下发；
2) 解析应用连接返回报文，获取密文 2 和客户机签名 2；
3) 调用会话协商验证函数(Obj_*_Formal_VerifySession)，完成会话协商验证，产生会话密钥；
4) 密钥协商成功，可以进行密钥更新、证书更新、设置和操作安全传输的数据加密；

(3). 密钥恢复（此过程，无需调用芯片初始化函数）

1) 调用密钥更新函数（Obj_*_Formal_GetTrmKeyData，iKeyState=0），获取密钥恢复数据；
2) 调用安全传输加密函数（Obj_*_ Formal_GetSessionData）对密钥恢复数据进行加密，并组织密钥更新报文下发；
3) 调用安全传输解密函数（Obj_*_ Formal_VerifyMeterData），进行设置返回数据解密；
4) 初始化成功后，设备对称密钥处于测试密钥状态，版本为（00000000000000000000000000000000）；

(4). 设置或操作类安全传输

1) 调用安全传输加密函数（Obj_*_ Formal_GetSessionData），进行设置或操作类安全传输的数据加密，组织并下发；
2) 调用安全传输解密函数（Obj_*_Formal_VerifyMeterData），进行设置或操作类安全传输的返回数据解密；

(5). 设置 ESAM 参数（本地表具有功能）

1) 调用设置 ESAM 参数函数（Obj_Meter_Formal_SetESAMData），获取 ESAM 加密数据；表号设置只能在 iKeyState=0 时进行；
2) 调用安全传输加密函数（Obj_Meter_Formal_GetSessionData）对 ESAM 加密数据进行加密，并组织密钥更新报文下发；
3) 调用安全传输解密函数（Obj_Meter_Formal_VerifyMeterData），进行返回数据解密；

(6). 钱包操作（本地表具有功能）

1) 调用钱包操作函数（Obj_Meter_Formal_GetPurseData），进行钱包操作数据加密；钱包初始化只能在 iKeyState=0 时进行，其他操作只能在 iKeyState=1 时进行；

2) 调用安全传输加密函数（Obj_Meter_Formal_GetSessionData）对 ESAM 数据进行加密，并组织密钥更新报文下发；

3) 调用安全传输解密函数（Obj_Meter_Formal_VerifyMeterData），进行返回数据解密；

(7). 主动上报（无需建立应用连接）

1) 接收上报报文，调用上报数据验证函数（Obj_*_Formal_VerifyReportData）解密数据；

2) 调用上报数据返回报文加密函数（Obj_*_Formal_GetResponseData）对上报返回数据加密，组织报文并下发；

(8). 抄读数据（无需建立应用连接）

1) 明文+随机数方式下发抄读报文；

2) 调用抄读数据验证函数（Obj_*_Formal_VerifyReadData），解密验证抄读数据；

(9). 广播（无需建立应用连接）

1) 调用广播数据加密函数（Obj_*_Formal_GetGrpBrdCstData）组织报文并下发。

(10). 密钥更新

1) 调用密钥更新函数（Obj_*_Formal_GetTrmKeyData，iKeyState=1），获取密钥更新数据；

2) 调用安全传输加密函数（Obj_*_Formal_GetSessionData）对密钥恢复数据进行加密，并组织密钥更新报文下发；

3) 调用安全传输解密函数（Obj_*_Formal_VerifyMeterData），进行返回数据解密；

(11). 证书更新（终端有此项）

1) 调用获取证书信息函数（Obj_Terminal_Formal_GetCACertificateData，iKeyState=1），获取证书更新数据；

2) 调用安全传输加密函数（Obj_*_Formal_GetSessionData）对证书信息进行加密，并组织密钥更新报文下发；

3) 调用安全传输解密函数（Obj_*_Formal_VerifyMeterData），进行返回数据解密；

(12). 建立应用连接验证

1) 调用会话协商机函数（Obj_*_Formal_InitSession，iKeyState=1），产生密文 1 和客户机签名 1，组织建立应用连接报文并下发；

2) 解析应用连接返回报文，获取密文 2 和客户机签名 2；

3) 调用会话协商验证函数(Obj_*_Formal_VerifySession，iKeyState=1)，完成会话协商验证，产生会话密钥；

## 6.3. 密钥更新流程

### 6.3.1. 密钥更新流程

密钥更新流程主要用于各省计量中心全检密钥下装，其中(4)、(5)、(6)不是必要步骤，根据参数设置需要进行。

(1). 读取设备信息(COS 版本，芯片序列号、对称密钥版本、证书状态、证书、表号等)；

(2). 根据对称密钥版本确定 iKeyState 值，对称密钥版本=”7FFFFFFF**********”iKeyState=1，对称密钥版本=”00000000000000000000000000000000”， iKeyState=0，(*代表值不确定，终端和电能表有区别，所以无需对后续的每个字符进行判断)；

(3). 建立应用连接

1) 调用会话协商机函数（Obj_*_Formal_InitSession），产生密文 1 和客户机签名 1，组织建立应用连接报文并下发；

2) 解析应用连接返回报文，获取密文 2 和客户机签名 2；

3) 调用会话协商验证函数(Obj_*_Formal_VerifySession)，完成会话协商验证，产生会话密钥；

4) 密钥协商成功，可以进行密钥更新、证书更新、设置和操作安全传输的数据加密；

(4). 设置或操作类安全传输（根据需要进行参数设置）

1) 调用安全传输加密函数（Obj_*_ Formal_GetSessionData），进行设置或操作类安全传输的数据加密，组织并下发；

2) 调用安全传输解密函数（Obj_*_Formal_VerifyMeterData），进行设置或操作类安全传输的返回数据解密；

(5). 设置 ESAM 参数（根据需要进行参数设置）

1) 调用设置 ESAM 参数函数（Obj_Meter_Formal_SetESAMData），获取 ESAM 加密数据；表号设置只能在 iKeyState=0 时进行；

2) 调用安全传输加密函数（Obj_Meter_Formal_ GetSessionData）对 ESAM 数据进行加密，并组织密钥更新报文下发；

3) 调用安全传输解密函数（Obj_Meter_ Formal_ VerifyMeterData），进行返回数据解密；

(6). 钱包操作（本地表功能）

1) 调用钱包操作函数（Obj_Meter_Formal_GetPurseData），进行钱包初始化；

2) 调用安全传输加密函数（Obj_Meter_Formal_GetSessionData）对 ESAM 数据进行加密，并组织密钥更新报文下发；

3) 调用安全传输解密函数（Obj_Meter_Formal_ VerifyMeterData），进行返回数据解密；

(7). 密钥更新

1) 调用密钥更新函数（Obj_*_Formal_GetTrmKeyData，iKeyState=1），获取密钥更新数据；

2) 调用安全传输加密函数（Obj_*_Formal_GetSessionData）对密钥更新数据进行加密，并组织密钥更新报文下发；

3) 调用安全传输解密函数（Obj_*_Formal_VerifyMeterData），进行返回数据解密；

(8). 证书更新（终端有此项）

1) 调用获取证书信息函数（Obj_Terminal_Formal_GetCACertificateData，iKeyState=1），获取证书更新数据；

2) 调用安全传输加密函数（Obj_*_Formal_GetSessionData）对证书信息数据进行加密，并组织密钥更新报文下发；

3) 调用安全传输解密函数（Obj_*_Formal_VerifyMeterData），进行返回数据解密；

(9). 读取设备信息(COS 版本，芯片序列号、对称密钥版本、证书状态、电能表表号等)，对称密钥版本=”7FFFFFFF**********”、证书状态=”01”，密钥下装成功，(*代表值不确定，终端和电能表有区别，所以无需对后续的每个字符进行判断)；

(10). 建立应用连接验证

1) 调用会话协商机函数（Obj_*_Formal_InitSession，iKeyState=1），产生密文 1 和客户机签名 1，组织建立应用连接报文并下发；

2) 解析应用连接返回报文，获取密文 2 和客户机签名 2；

3) 调用会话协商验证函数(Obj_*_Formal_VerifySession，iKeyState=1)，完成会话协商验证，产生会话密钥；

(11). 私钥下清零（见第(4)步）

### 6.3.2. 密钥更新注意事项

(1). 密钥更新完成后，需要读取密钥版本，然后进行私钥下应用连接，即第（9）和第（10）步；

(2). 电能表密钥更新成功与否不能完全根据电能表屏幕上的出厂状态（小房子）判断，因为目前试点厂家做法没有统一；

(3). 密钥恢复需要密钥恢复 key 才能进行；

(4). 主站对返回帧需要严格验证和解密，如果验证不过或解密不成功，则需要查找原因，不能判定合格；

(5). 密钥下装程序正式电能表或终端生产前，必须通过采集主站联调确认；如果是本地表，还需要通过售电系统进行确认；

## 6.4. 终端远程动态库接口说明

### 6.4.1. 主站会话协商

功能描述：

数字签名连接认证机制，用于主站与设备进行会话协商时产生密文和签名数据，该过程在建立应用连接时完成。

函　　数：

```
int    Obj_Terminal_Formal_InitSession (
   int      iKeyState,
   char * cTESAMID,
   char * cASCTR,
   char * cFLG,
   char * cMasterCert,
   char * cOutRandHost,
   char * cOutSessionInit,
   char * cOutSign
);
```

参数说明：

iKeyState

cTESAMID

cASCTR

cFLG　　应用密钥产生标识，1Byte，默认”01”；

cMasterCert　　主站证书；

cOutRandHost　　主站随机数（16Byte）；

cOutSessionInit　　会话协商数据（32Byte），建立应用连接中的密文 1；

cOutSign　　协商数据签名(64Byte) ，建立应用连接中的客户机签名 1；

- 函数返回：

返回 0　　成功

其他　　失败

### 6.4.2. 主站会话协商验证

- 功能描述：

数字签名连接认证机制，用于主站验证设备会话协商时返回的数据，验证成功主站产生会话密钥。

- 函　　数：

```
int    Obj_Terminal_Formal_VerifySession (
   int      iKeyState,
   char * cTESAMID,
   char * cRandHost,
   char * cSessionData,
   char * cSign,
   char * cTerminalCert,
   char * cOutSessionKey
);
```

- 参数说明：

iKeyState

cTESAMID

cRandHost　　主站随机数 R1（16Byte）；

cSessionData　　终端返回的应用会话协商数据(48Byte)，对应建立应用连接中的

密文 2；

cSign 终端返回的应用会话协商数据签名(64Byte)，对应建立应用连接中的签名数据 2；

cTerminalCert 终端证书

cOutSessionKey 会话密钥

- 函数返回：

返回 0 成功

其他 失败

### 6.4.3. 抄读数据验证

- 功能描述：

主站验证设备返回的抄读数据，具体指抄读终端返回的数据。

- 函　　数：

```
int   Obj_Terminal_Formal_VerifyReadData(
   int     iKeyState,
   int     iOperateMode,
   char * cTESAMID,
   char * cRandHost,
   char * cReadData,
   char * cMac,
   char * cOutData
);
```

- 参数说明：

iKeyState=1

iOperateMode

cTESAMID

cRandHost 主站随机数(16Byte)

cReadData 抄读数据

cMacMAC　数据

cOutData　明文抄读数据，iOperateMode=1，为空

- 函数返回：

返回 0　　成功

其他　　失败

### 6.4.4. 上报数据验证

- 功能描述：

设备主动上报数据时，主站验证数据的合法性。

- 函　　数：

```
int   Obj_Terminal_Formal_VerifyReportData(
  int     iKeyState,
  int     iOperateMode,
  char * cTESAMID,
  char * cRandT,
  char * cReportData,
  char * cMac,
  char * cOutData,
  char *cOutRSTCTR
);
```

- 参数说明：

iKeyState=1

iOperateMode

cTESAMID

cRandT　　终端随机数(12B)

cReportData　　上报数据

cMac　　MAC 数据

cOutData　　明文数据，iOperateMode=1，为空

cOutRSTCTR　　主动上报随机数

- 函数返回：

返回 0　　成功

其他　　失败

### 6.4.5. 上报数据返回报文加密

- 功能描述：

用于设备主动上报主站返回帧数据加密计算。

- 函　　数：

```
int    Obj_Terminal_Formal_GetResponseData (
   int      iKeyState,
   int      iOperateMode,
   char * TESAMID,
   char * RandHost,
   char * cReportData,
   char * OutSID,
   char * OutAttachData,
   char * cOutData,
   char * ucOutMac
  );
```

- 参数说明：

iKeyState=1

iOperateMode

cTESAMID

cRandT　　上报随机数，12Byte

cReportData　　上报数据

cOutSID

cOutAttachData

ucOutData

ucOutMac

- 函数返回：

返回 0　　成功

其他　　失败

### 6.4.6. 安全传输加密

- 功能描述：

用于对具体业务数据进行数据加密计算。

- 函　　数：

```
int    Obj_Terminal_Formal_GetSessionData(
   int      iOperateMode,
   char * cTESAMID,
   char * cSessionKey,
   int      cTaskType,
   char * cTaskData,
   char * cOutSID,
   char * cOutAttachData,
   char * cOutData ,
   char * cOutMAC
);
```

- 参数说明：

iOperateMode；

cTESAMID

cSessionKey　　会话密钥

cTaskType　　参数类型：

4，安全模式参数，会话时效；
7，拉闸；
8，文件传输；

3，除上述操作外的数据加密。

cTaskData　　数据明文；NByte

cOutSID

cOutAttachData

cOutData

cOutMAC

- 函数返回：

返回 0　　成功

其他　　失败

### 6.4.7. 安全传输解密

- 功能描述：

用于验证终端返回帧数据解密验证。

- 函　　数：

```
int    Obj_Terminal_Formal_VerifyTerminalData (
   int      iKeyState,
   int      iOperateMode,
   char * cTESAMID,
   char * cSessionKey,
   char * cTaskData,
   char * cMac,
   char * cOutData
);
```

- 参数说明：

iKeyState=1

iOperateMode

cTESAMID

cSessionKey

| | |
|---|---|
| cTaskData | 数据 |
| cMac | MAC 数据 |
| cOutData | 数据明文 |

- 函数返回：

| | |
|---|---|
| 返回 0 | 成功 |
| 其他 | 失败 |

### 6.4.8. 广播数据加密

- 功能描述：

用于广播数据加密计算。

- 函　　数：

```
int   Obj_Terminal_Formal_GetGrpBrdCstData(
   int     iKeyState,
   int     iOperateMode ，
   char * cTESAMID,
   char * cBrdCstAddr,
   char * AGSEQ,
   char * cBrdCstData,
   char * cOutSID,
   char * cOutAttachData,
   char * cOutData,
   char * cOutMac
);
```

- 参数说明：

iKeyState=1

iOperateMode

cTESAMID

ccBrdCstAddr　广播地址；

删除的内容: char * cGrpKID,　.

删除的内容: DIV

删除的内容: cGrpKID．．．业务名称，按照 01-10 参数设置．

删除的内容: DIV

删除的内容: ．．分散因子，组地址

删除的内容: 8Byte

| | |
|---|---|
| AGSEQ | 广播应用通信序列号，4Byte |
| cBrdCstData | 广播数据明文； |
| cOutSID | |
| cOutAttachData | |
| cOutData | |
| cOutMac | |

删除的内容:

删除的内容: NBytePutCtime . 字符型，单位秒；

- 函数返回：

| | |
|---|---|
| 返回 0 | 成功 |
| 其他 | 失败 |

### 6.4.9. 终端对称密钥更新

- 功能描述：

用于对称密钥更新。

- 函　　数：

```
int Obj_Terminal_Formal_GetTrmKeyData(
   int    iKeyState,
   char * cTESAMID,
   char * cSessionKey,
   char * cTerminalAddress,
   char *cKeyType,
   char * cOutSID,
   char * cOutAttachData,
   char * cOutTrmKeyData ,
   char * cOutMAC
);
```

- 参数说明：

| | |
|---|---|
| iKeyState | 密钥更新的目标状态，1，代表更新到私钥，0 代表恢复到初始密钥； |

cTESAMID

cSessionKey

cTerminalAddress　　终端地址(8 Bytes)

cKeyType　　密钥类型，00 应用密钥，01 链路密钥

cOutSID

cOutAttachData

cOutData

cOutMAC

删除的内容: iOperateMode .

函数返回:

返回 0　　成功

其他　　失败

### 6.4.10. 终端对称密钥初始化

功能描述:

用于对终端密钥进行初始化，会话计数器次数为 1 时，须先对密钥进行初始化。

函　　数:

```
int Obj_Terminal_Formal_ InitTrmKeyData (
   int    iKeyState,
   char * cTESAMID,
   char * cSessionKey,
   char * cTerminalAddress,
   char *cKeyType,
   char * cOutSID,
   char * cOutAttachData,
   char * cOutTrmKeyData ,
   char * cOutMAC
);
```

参数说明：

iKeyState=0

cTESAMID

cSessionKey

cTerminalAddress　　终端地址(8 Bytes)

cKeyType　　密钥类型，00 应用密钥

cOutSID

cOutAttachData

cOutData

cOutMAC

删除的内容: iOperateMode .

函数返回：

返回 0　　成功

其他　　失败

### 6.4.11. 获取证书信息

功能描述：

用于对终端通密钥进行初始化，会话计数器次数为 1 时，须先对密钥进行初始化。

函　　数：

```
int Obj_Terminal_Formal_ GetCACertificateData (
  int   iKeyState,
  char * cTESAMID,
  char * cSessionKey,
  char * cCerType,
  char * cOutSID,
  char * cOutAttachData,
  char * cOutCertificateData,
```

char * cOutMAC

);

- 参数说明：

iKeyState

cTESAMID　　此处可为空

cSessionKey

cCerType　　证书类型，“01”

cOutSID

cOutCertificateData　　证书数据

cOutData

cOutMAC

- 函数返回：

返回 0　　成功

其他　　失败

## 6.5. 电能表远程动态库接口说明

### 6.5.1. 主站会话协商

- 功能描述：

对称密码连接认证机制，用于主站与设备进行会话协商时产生密文 1 和客户机签名 1，该过程在建立应用连接时完成。

- 函　　数：

int　Obj_ Meter _Formal_InitSession (

int　iKeyState,

char * cDiv,

char * cASCTR,

char * cFLG,

删除的内容: cESAMID

```
  char * cOutRandHost,
  char * cOutSessionInit,
  char * cOutSign
);
```

删除的内容: char * cMasterCert, .

参数说明：

| | |
|---|---|
| iKeyState | |
| cDiv | 分散因子（8Byte），iKeyState=0，cDiv 为芯片序列号；<br>iKeyState=1，cDiv 为表号； |
| cASCTR | |
| cFLG | 应用密钥产生标识，1Byte，默认”01”; |
| cOutRandHost | 主站随机数（16Byte） |
| cOutSessionInit | 会话协商数据，建立应用连接中的密文 1； |
| cOutSign | 协商数据签名(4Byte)，建立应用连接中的客户机签名 1； |

删除的内容: cESAMID .

函数返回：

| | |
|---|---|
| 返回 0 | 成功 |
| 其他 | 失败 |

### 6.5.2. 主站会话协商验证

功能描述：

对称密码连接认证机制，用于主站验证设备会话协商时返回的数据，验证成功主站产生会话密钥。

```
int    Obj_Meter_Formal_VerifySession (
  int     iKeyState,
  char * cDiv,
  char * cRandHost,
  char * cSessionData,
  char * cSign,
  char * cOutSessionKey
```

删除的内容: cESAMID

);

- 参数说明：

iKeyState

cDiv　　分散因子（8Byte），iKeyState=0，cDiv 为芯片序列号；

　　iKeyState=1，cDiv 为表号，表号不足 8 字节前补 0；

cRandHost　　主站随机数 R1（16Byte）

cSessionData　　终端返回的会话协商数据(48Byte)，建立应用连接中的密文 2；

cSign　　终端返回的会话协商数据签名(4Byte) ，建立应用连接中的客户机签名 2；

cOutSessionKey　会话密钥

删除的内容: cESAMID

- 函数返回：

返回 0　　成功

其他　　失败

### 6.5.3. 抄读数据验证

- 功能描述：

主站验证设备返回的抄读数据，具体指抄读电能表返回的数据。

- 函　　数：

```
int   Obj_Meter_Formal_VerifyReadData(
  int     iKeyState,
  int     iOperateMode,
  char * cMeterNo,
  char * cRandHost,
  char * cReadData,
  char * cMac,
  char * cOutData
);
```

删除的内容: cESAMID

参数说明：

iKeyState

iOperateMode

cMeterNo

cRandHost　主站随机数(16Byte)

cReadData　抄读数据

cMacMAC　数据

cOutData　明文抄读数据，iOperateMode=1，为空

函数返回：

返回 0　　成功

其他　　失败

### 6.5.4. 上报数据验证

功能描述：

设备主动上报数据时，主站验证数据的合法性。

函　　数：

```
int   Obj_Meter_Formal_VerifyReportData(
  int    iKeyState,
  int    iOperateMode,
  char * cMeterNo,
  char * cRandT,
  char * cReportData,
  char * cMac,
  char * cOutData,
  char * cOutRSPCTR
);
```

参数说明：

删除的内容: =1

删除的内容: cESAMID .

删除的内容: cESAMID

iKeyState=1

iOperateMode

cMeterNo

删除的内容: cESAMID

| | |
|---|---|
| cRandT | 终端随机数(12B) |
| cReportData | 上报数据 |
| cMac | MAC 数据 |
| cOutData | 明文数据，iOperateMode=1，为空 |
| cOutRSPCTR | 主动上报随机数 |

函数返回:

| | |
|---|---|
| 返回 0 | 成功 |
| 其他 | 失败 |

### 6.5.5. 上报数据返回报文加密

功能描述:

用于设备主动上报主站返回帧数据加密计算。

函　　数:

```
int   Obj_Meter_Formal_GetResponseData (
   int     iKeyState,
   int     iOperateMode,
   char * cMeterNo,
   char * RandHost,
   char * cReportData,
   char * OutSID,
   char * OutAttachData,
   char * cOutData,
   char * ucOutMac
 );
```

删除的内容: cESAMID

参数说明：

iKeyState=1

iOperateMode

cMeterNo

cRandT　　上报随机数，12Byte

cReportData　　上报数据

cOutSID

cOutAttachData

ucOutData

ucOutMac

删除的内容: cESAMID

函数返回：

返回 0　　成功

其他　　失败

### 6.5.6. 安全传输加密

功能描述：

用于对具体业务数据进行数据加密计算。

函　　数：

```
int    Obj_Meter_Formal_GetSessionData(
   int      iOperateMode,
   char * cESAMID,
   char * cSessionKey,
   int      cTaskType,
   char * cTaskData,
   char * cOutSID,
   char * cOutAttachData,
   char * cOutData ,
   char * cOutMAC
```

);

- 参数说明：

iOperateMode；

cESAMID　　此处可为空

cSessionKey　　会话密钥

cTaskType　　参数类型：

4，安全模式参数、会话时效门限；
5，电价、电价切换时间、费率时段、对时；
8，拉闸任务；
3，除上述操作外的数据加密，密钥更新、写 ESAM 操作和钱包操作数据下发通过此函数进行安全计算。

cTaskData　　数据明文；NByte

cOutSID

cOutAttachData

cOutData

cOutMAC

- 函数返回：

返回 0　　成功

其他　　失败

### 6.5.7. 安全传输解密

- 功能描述：

用于电能表返回帧数据解密验证。

- 函　　数：

```
int    Obj_Meter_Formal_VerifyMeterData (
   int      iKeyState,
   int      iOperateMode,
   char * cESAMID,
   char * cSessionKey,
   char * cTaskData,
```

char * cMac,

char * cOutData

);

- 参数说明：

iKeyState=1

iOperateMode

cESAMID 此处可为空

cSessionKey

cTaskData 数据

cMac MAC 数据

cOutData 数据明文

- 函数返回：

返回 0 成功

其他 失败

### 6.5.8. 广播数据加密

- 功能描述：

用于广播数据加密计算。

- 函 数：

int Obj_Meter_Formal_GetGrpBrdCstData(

int iKeyState,

int iOperateMode,

char * cESAMID,

char * cBrdCstAddr,

char * AGSEQ,

char * cBrdCstData,

char * cOutSID,

删除的内容: char * cGrpKID, .

删除的内容: DIV

char * cOutAttachData,

char * cOutData,

char * cOutMac

);

- 参数说明：

iKeyState=1

iOperateMode

cESAMID 此处可为空

cBrdCstAddr 广播地址

AGSEQ 广播应用通信序列号，4Byte

cBrdCstData 广播数据明文；

cOutSID

cOutAttachData

cOutData

cOutMac

- 函数返回：

返回 0 成功

其他 失败

### 6.5.9. 设置 ESAM 参数

- 功能描述：

用于设置表号、当前套电价文件、备用套电价文件、ESAM 存储标识。

- 函　　数：

int Obj_Meter_Formal_SetESAMData (

int InKeyState,

int InOperateMode,

char * cESAMID,

char * cSessionKey,

删除的内容: cGrpKID．．．业务名称，按照 01-10 参数设置．

删除的内容: DIV

删除的内容: ．分散因子，组地址

删除的内容: ； 8Byte

删除的内容:

```
    char * cMeterNo,
    char * cESAMRand,
    char * cData,
    char * OutSID,
    char * OutAddData,
    char * OutData,
    char * OutMAC
);
```

- 参数说明：

| | |
|---|---|
| iKeyState=1 | |
| iOperateMode | |
| cESAMID | 此处可为空 |
| cSessionKey | |
| cMeterNo | 表号(8Byte)，不够 8Byte 前面填充 0 |
| cESAMRand | |
| cData | 4ByteOAD + 1Byte 内容 LEN + 内容 |
| cOutSID | |
| cOutAttachData | |
| cOutData | |
| cOutMAC | |

- 函数返回：

| | |
|---|---|
| 返回 0 | 成功 |
| 其他 | 失败 |

### 6.5.10. 钱包操作

- 功能描述：

用于设置表号、当前套电价文件、备用套电价文件、ESAM 存储标识。

- 函　　数：

```
int    Obj_Meter_Formal_GetPurseData (
   int      iOperateMode,
   char * cESAMID,
   char * cSessionKey,
   int      cTaskType,
   char * cTaskData,
   char * cOutSID,
   char * cOutAttachData,
   char * cOutData ,
   char * cOutMAC
);
```

- 参数说明：

| | |
|---|---|
| iOperateMode | |
| cESAMID | 此处可为空 |
| cSessionKey | |
| cTaskType | 任务序编号， 9，钱包初始化；10，钱包充值；11，钱包退费 |
| cTaskData | 数据明文，包含预置金额，4Byte cOutSID |
| cOutAttachData | |
| cOutData | |
| cOutMAC | |

- 函数返回：

| | |
|---|---|
| 返回 0 | 成功 |
| 其他 | 失败 |

### 6.5.11. 电能表对称密钥更新

- 功能描述：

  用于对称密钥更新。

- 函　　数：

```
int Obj_ Meter _Formal_GetTrmKeyData(
   int    iKeyState,
   char * cESAMID,
   char * cSessionKey,
   char * cMeterNo,
   char *cKeyType,
   char * cOutSID,
   char * cOutAttachData,
   char * cOutTrmKeyData ,
   char * cOutMAC
);
```

- 参数说明：

| | |
|---|---|
| iKeyState | 密钥更新的目标状态，1，代表更新到私钥，0 代表恢复到初始密钥； |
| cESAMID | |
| cSessionKey | |
| cMeterNo | 表号(8Bytes) |
| cKeyType | 密钥类型，此处为 00，应用密钥 |
| cOutSID | |
| cOutAttachData | |
| cOutData | |
| cOutMAC | |

删除的内容: iOperateMode .

- 函数返回：

| | |
|---|---|
| 返回 0 | 成功 |
| 其他 | 失败 |

### 6.5.12. 电表对称密钥初始化

- 功能描述：

用于对电能表对称密钥进行初始化，须先对密钥进行初始化，此函数暂时不需要使用。

## 函　　数：

```
int Obj_ Meter _Formal_ InitTrmKeyData (
   int    iKeyState,
   char * cESAMID,
   char * cSessionKey,
   char * cMeterNo,
   char *cKeyType,
   char * cOutSID,
   char * cOutAttachData,
   char * cOutTrmKeyData ,
   char * cOutMAC
);
```

## 参数说明：

| | |
|---|---|
| iKeyState | |
| cESAMID | |
| cSessionKey | |
| cMeterNo | 表号(8Bytes)，不足前补 0； |
| cKeyType | 密钥类型，00 应用密钥 |
| cOutSID | |
| cOutAttachData | |
| cOutData | |
| cOutMAC | |

删除的内容: iOperateMode .

## 函数返回：

| | |
|---|---|
| 返回 0 | 成功 |
| 其他 | 失败 |

# 7. 错误码说明

| 错误码 | 错误描述 | 错误码 | 错误描述 |
|---|---|---|---|
| 48 | 无设备或设备无效 | 56 | 创建 socket 句柄失败 |
| 57 | 连接服务器失败 | 64 | 客户端发送数据失败 |
| 65 | 客户端接收数据失败 | 100 | 打开设备失败 |
| 160 | 连接密码机失败 | 161 | 操作权限不够 |
| 162 | USBKey 不是操作员 | 163 | 服务器发送数据失败 |
| 164 | 服务端接收报文失败 | 165 | 密码机加密数据失败 |
| 166 | 密码机导出密钥失败 | 167 | 密码机计算 MAC 失败 |
| 168 | 服务器已断开连 | 169 | 数据无效 |
| 170 | 密码机收发报文错误 | 171 | 密码机故障 |
| 172 | 数据库出错 | 700-712 | 客户端导出密钥失败 |
| 800-810 | 计算 MAC 失败 | 900-910 | 加密数据失败 |
| 1000-1010 | 数据长度错 | 1100 | 系统认证错误 |
| 1107 | USBKey 权限不正确 | 1108-1111 | 操作 USBKey 失败 |
| 1206 | 签名数据错误 | 45 | 密码机密钥错 |