

服务器密码机 应用编程开发手册（C） V1.1

目录

1	API 提供形式	3
2	数据结构定义	3
2.1	宏定义	3
2.2	结构定义	4
3	设备管理类接口	5
3.1	SDF_OpenDevice	5
3.2	SDF_OpenDevice_ex	6
3.3	SDF_OpenDeviceWithPath	6
3.4	SDF_CloseDevice	6
3.5	SDF_OpenSession	7
3.6	SDF_CloseSession	7
3.7	SDF_GetDeviceInfo	7
3.8	SDF_GenerateRandom	8
3.9	SDF_GetPrivateKeyAccessRight	8
3.10	SDF_ReleasePrivateKeyAccessRight	9
4	密钥管理类函数	9
4.1	SDF_GenerateKeyPair_ECC	10
4.2	SDF_ExportSignPublicKey_ECC	10
4.3	SDF_ExportEncPublicKey_ECC	11
4.4	SDF_GenerateKeyWithEPK_ECC	11
4.5	SDF_GenerateKeyWithIPK_ECC	12
4.6	SDF_ImportKeyWithISK_ECC	12
4.7	SDF_GenerateAgreementDataWithECC	13
4.8	SDF_GenerateKeyWithECC	14
4.9	SDF_GenerateAgreementDataAndKeyWithECC	14
4.10	SDF_ExchangeDigitEnvelopeBaseOnECC	15
4.11	SDF_GenerateKeyWithKEK	16
4.12	SDF_ImportKeyWithKEK	16

4.13	SDF_ImportKey.....	17
4.14	SDF_DestroyKey.....	18
5	非对称运算类接口.....	18
5.1	SDF_ExternalSign_ECC.....	18
5.2	SDF_ExternalVerify_ECC.....	19
5.3	SDF_InternalSign_ECC.....	20
5.4	SDF_InternalVerify_ECC.....	20
5.5	SDF_ExternalEncrypt_ECC.....	21
5.6	SDF_ExternalDecrypt_ECC.....	21
6	对称算法运算类接口.....	22
6.1	SDF_Encrypt.....	22
6.2	SDF_Decrypt.....	23
6.3	SDF_CalculateMAC.....	24
7	杂凑运算类接口.....	24
7.1	SDF_HashInit.....	25
7.2	SDF_HashUpdate.....	25
7.3	SDF_HashFinal.....	26
8	用户文件操作类接口.....	26
8.1	SDF_CreateFile.....	26
8.2	SDF_ReadFile.....	27
8.3	SDF_WriteFile.....	27
8.4	SDF_DeleteFile.....	28
	附录 A 常见返回码注释.....	30

1 API提供形式

密码服务一体机（以下简称密码机）提供以下六类应用程序接口(API)函数：

- 设备管理类接口
- 密钥管理类接口
- 非对称运算类接口
- 对称运算类接口
- 杂凑运算类接口
- 用户文件操作类接口

密码机的应用程序接口(API)将以 API 头文件+动态库的形式提供给应用开发。针对不同的操作系统平台(windows、Linux 等)提供各自独立的库文件。应用程序通过与库文件及预定义的头文件联编后即可实现对 API 的调用。支持 C、C++等高级语言编程。

密码机应用程序接口(API)头文件及链接库定义如下：

- 头文件
sdf_cryptoapi.h: 描述应用程序接口(API)原型
- 链接库
在 Windows 环境:
动态库: sdfapi_x64.dll
在 Linux 环境:
动态库: libsdfapi_x64.so
- 配置文件
Linux : /etc/sdt_hsmcrypt.conf
Windows: sdt_hsmcrypt.conf

2 数据结构定义

2.1 宏定义

```
#define MAX_USERNAME_LEN      32    //用户名长度
#define MAX_USERPASSWD_LEN    32    //用户口令长度
#define ECCref_MAX_BITS       512
```

```
#define ECCref_MAX_LEN ((ECCref_MAX_BITS+7) / 8)
```

```
#define ECCCipher_MAX_LEN 1440
```

2.2 结构定义

```
typedef struct DeviceInfo_st {           //设备信息
    unsigned char IssuerName[40];        //设备生产厂商名称
    unsigned char DeviceName[16];        //设备型号
    unsigned char DeviceSerial[16];      //设备编号
    unsigned int DeviceVersion;           //密码设备内部软件的版本号
    unsigned int StandardVersion;         //密码设备支持的接口规范版本号
    unsigned int AsymAlgAbility[2];       //前 4 字节表示支持的算法，后 4 字
节表示算法的最大模长
    unsigned int SymAlgAbility;           //所有支持的对称算法
    unsigned int HashAlgAbility;          //所有支持的杂凑算法
    unsigned int BufferSize;              //支持的最大文件存储空间
} DEVICEINFO;

typedef struct ECCrefPublicKey_st {       //ECC 公钥结构
    unsigned int bits;                   //密钥位长
    unsigned char x[ECCref_MAX_LEN];     //公钥 x 坐标
    unsigned char y[ECCref_MAX_LEN];     //公钥 y 坐标
} ECCrefPublicKey;

typedef struct ECCrefPrivateKey_st {      //ECC 私钥结构
    unsigned int bits;                   //密钥位长
    unsigned char K[ECCref_MAX_LEN];     //私钥
} ECCrefPrivateKey;

typedef struct ECCCipher_st {             //ECC 密文结构
    unsigned char x[ECCref_MAX_LEN];     //X 分量
    unsigned char y[ECCref_MAX_LEN];     //Y 分量
    unsigned char M[32];                  //明文的杂凑值
    unsigned int L;                       //密文数据长度
    unsigned char C[ECCCipher_MAX_LEN];  //密文数据
} ECCCipher;

typedef struct ECCSignature_st {          //签名结构
    unsigned char r[ECCref_MAX_LEN];     //签名的 r 部分
    unsigned char s[ECCref_MAX_LEN];     //签名的 s 部分
} ECCSignature;
```

```
typedef struct SDF_ENVELOPEDKEYBLOB { // ECC 加密密钥对保护结构
    unsigned long Version;           //版本号
    unsigned long ulSymmAlgID;       //对称算法 ID
    ECCCipher ECCCipherBlob;        //对称密钥密文
    ECCrefPublicKey PubKey;          //加密公钥
    unsigned char cbEncryptedPrivKey[64]; //加密密钥对的私钥密文
} EnvelopedKeyBlob, *PEnvelopedKeyBlob;
```

3 设备管理类接口

设备管理类接口包含以下具体函数，各函数返回值见附录 A 错误代码定义：

- A. 打开密码设备：SDF_OpenDevice
- B. 打开密码设备（IP/端口方式）：SDF_OpenDevice_ex
- C. 打开密码设备（配置路径方式）：SDF_OpenDeviceWithPath
- D. 关闭密码设备，并释放相关资源：SDF_CloseDevice
- E. 创建与密码设备的会话：SDF_OpenSession
- F. 关闭与密码设备已建立的会话，并释放相关资源：SDF_CloseSession
- G. 获取设备信息：SDF_GetDeviceInfo
- H. 获取指定长度的随机数：SDF_GenerateRandom
- I. 获取密码设备内部存储的指定索引私钥的使用授权：
SDF_GetPrivateKeyAccessRight
- J. 释放密码设备存储的指定索引私钥的使用授权：
SDF_ReleasePrivateKeyAccessRight

3.1 SDF_OpenDevice

功能：

打开密码设备。

函数原型：

```
int SDF_OpenDevice(
    void **phDeviceHandle
);
```

参数说明：

- phDeviceHandle：输出：返回设备句柄。

返回值说明：

0	成功。
非 0	失败，返回错误代码。返回码的取值范围参见附录 A。

3.2 SDF_OpenDevice_ex

功能：

打开密码设备。

函数原型：

```
int SDF_OpenDevice_ex(
    void **phDeviceHandle, char *ip, int port
);
```

参数说明：

- ip: 输入：密码机 IP 地址。
- port: 输入：密码机服务端口。
- phDeviceHandle: 输出：返回设备句柄。

返回值说明：

0	成功。
非 0	失败，返回错误代码。返回码的取值范围参见附录 A。

3.3 SDF_OpenDeviceWithPath

功能：

打开密码设备。

函数原型：

```
int SDF_OpenDeviceWithPath (
    char *pInFileName, void **phDeviceHandle
);
```

参数说明：

- pInFileName: 输入：密码机配置文件名称（含路径）。
- phDeviceHandle: 输出：返回设备句柄。

返回值说明：

0	成功。
非 0	失败，返回错误代码。返回码的取值范围参见附录 A。

3.4 SDF_CloseDevice

功能：

关闭密码设备，并释放相关资源。

函数原型：

```
int SDF_CloseDevice(
```

```
void *hDeviceHandle
```

```
);
```

参数说明:

- hDeviceHandle: 输入: 已打开的设备句柄。

返回值说明:

- | | |
|-----|----------------------------|
| 0 | 成功。 |
| 非 0 | 失败, 返回错误代码。返回码的取值范围参见附录 A。 |

3.5 SDF_OpenSession**功能:**

创建与密码设备的会话。

函数原型:

```
int SDF_OpenSession(
    void *hDeviceHandle,
    void **phSessionHandle
);
```

参数说明:

- hDeviceHandle: 输入: 已打开的设备句柄。
- phSessionHandle: 输出: 返回与密码设备建立的新会话句柄。

返回值说明:

- | | |
|-----|----------------------------|
| 0 | 成功。 |
| 非 0 | 失败, 返回错误代码。返回码的取值范围参见附录 A。 |

3.6 SDF_CloseSession**功能:**

关闭与密码设备已建立的会话, 并释放相关资源。

函数原型:

```
int SDF_CloseSession(
    void *hSessionHandle
);
```

参数说明:

- hSessionHandle: 输入: 会话句柄。

返回值说明:

- | | |
|-----|----------------------------|
| 0 | 成功。 |
| 非 0 | 失败, 返回错误代码。返回码的取值范围参见附录 A。 |

3.7 SDF_GetDeviceInfo**功能:**

获取设备信息。

函数原型:

```
int SDF_GetDeviceInfo (  
    void *hSessionHandle,  
    DEVICEINFO *pstDeviceInfo  
);
```

参数说明:

- hSessionHandle: 输入: 会话句柄。
- pstDeviceInfo: 输出: 设备信息。

返回值说明:

- 0 成功。
- 非 0 失败, 返回错误代码。返回码的取值范围参见附录 A。

3.8 SDF_GenerateRandom

功能:

获取指定长度的随机数。

函数原型:

```
int SDF_GenerateRandom (  
    void *hSessionHandle,  
    unsigned int uiLength,  
    unsigned char *pucRandom  
);
```

参数说明:

- hSessionHandle: 输入: 会话句柄。
- uiLength: 输入: 随机数长度。
- pucRandom: 输出: 随机数首地址。

返回值说明:

- 0 成功。
- 非 0 失败, 返回错误代码。返回码的取值范围参见附录 A。

3.9 SDF_GetPrivateKeyAccessRight

功能:

获取密码设备内部存储的指定索引私钥的使用授权。

函数原型:

```
int SDF_GetPrivateKeyAccessRight (  
    void *hSessionHandle,  
    unsigned int uiKeyIndex,  
    unsigned char *pucPassword,  
    unsigned int uiPwdLength  
);
```

参数说明:

- `hSessionHandle`: 输入: 会话句柄。
- `uiKeyIndex`: 输入: 私钥索引号 1-10000。
- `pucPassword`: 输入: 私钥访问口令。
- `uiPwdLength`: 输入: 私钥访问口令长度, 8-16。

返回值说明:

- 0 成功。
- 非 0 失败, 返回错误代码。返回码的取值范围参见附录 A。

3.10 SDF_ReleasePrivateKeyAccessRight**功能:**

释放密码设备存储的指定索引私钥的使用授权。

函数原型:

```
int SDF_ReleasePrivateKeyAccessRight (
    void *hSessionHandle,
    unsigned int uiKeyIndex
);
```

参数说明:

- `hSessionHandle`: 输入: 会话句柄。
- `uiKeyIndex`: 输入: 私钥索引号 1-10000。

返回值说明:

- 0 成功。
- 非 0 失败, 返回错误代码。返回码的取值范围参见附录 A。

4 密钥管理类函数

对称密钥管理类接口包含以下具体函数, 各函数返回值见附录 A 错误代码定义:

- 请求密码设备产生指定类型和模长的 ECC 密钥对:
`SDF_GenerateKeyPair_ECC`
- 导出 ECC 签名公钥: `SDF_ExportSignPublicKey_ECC`
- 导出 ECC 加密公钥: `SDF_ExportEncPublicKey_ECC`
- 生成会话密钥并用外部 ECC 公钥加密输出, 同时返回密钥句柄:
`SDF_GenerateKeyWithEPK_ECC`
- 生成会话密钥并用内部 ECC 公钥加密输出, 同时返回密钥句柄:
`SDF_GenerateKeyWithIPK_ECC`
- 导入会话密钥并用内部 ECC 加密私钥解密, 同时返回密钥句柄:

SDF_ImportKeyWithISK_ECC

- G. 生成密钥协商参数并输出: **SDF_GenerateAgreementDataWithECC**
- H. 使用 ECC 密钥协商算法, 使用自身协商句柄和响应方的协商参数计算会话密钥, 同时返回会话密钥句柄: **SDF_GenerateKeyWithECC**
- I. 使用 ECC 密钥协商算法, 产生协商参数并计算会话密钥, 同时返回产生的协商参数和和密钥句柄: **SDF_GenerateAgreementDataAndKeyWithECC**
- J. 基于 ECC 算法的数字信封转换: **SDF_ExchangeDigitEnvelopeBaseOnECC**
- K. 生成会话密钥并用密钥加密密钥加密输出, 同时返回密钥句柄: **SDF_GenerateKeyWithKEK**
- L. 导入会话密钥并用密钥加密密钥解密, 同时返回会话密钥句柄: **SDF_ImportKeyWithKEK**
- M. 导入明文会话密钥, 同时返回密钥句柄: **SDF_ImportKey**
- N. 销毁会话密钥, 并释放为密钥句柄分配的内存等资源: **SDF_DestroyKey**

4.1 SDF_GenerateKeyPair_ECC**功能:**

请求密码设备产生指定类型和模长的 ECC 密钥对。

函数原型:

```
int SDF_GenerateKeyPair_ECC(
    void *hSessionHandle,
    unsigned int uiAlgID,
    unsigned int uiKeyBits,
    ECCrefPublicKey *pucPublicKey,
    ECCrefPrivateKey *pucPrivateKey
);
```

参数说明:

- **hSessionHandle:** 输入: 会话句柄。
- **uiAlgID:** 输入: 指定算法标识。SGD_SM2、SGD_SM2_1、SGD_SM2_2、SGD_SM2_3
- **uiKeyBits:** 输入: 指定密钥长度, 仅支持 256。
- **pucPublicKey:** 输出: ECC 公钥结构。
- **pucPrivateKey:** 输出: ECC 私钥结构。

返回值说明:

- 0 成功。
- 非 0 失败, 返回错误代码。返回码的取值范围参见附录 A。

4.2 SDF_ExportSignPublicKey_ECC**功能:**

导出 ECC 签名公钥。

函数原型:

```
int SDF_ExportSignPublicKey_ECC(
    void *hSessionHandle,
    unsigned int uiKeyIndex,
    ECCrefPublicKey *pucPublicKey
);
```

参数说明:

- hSessionHandle: 输入: 会话句柄。
- uiKeyIndex: 输入: ECC 密钥索引号 1-10000。
- pucPublicKey: 输出: ECC 公钥结构。

返回值说明:

- 0 成功。
- 非 0 失败, 返回错误代码。返回码的取值范围参见附录 A。

4.3 SDF_ExportEncPublicKey_ECC

功能:

导出 ECC 加密公钥。

函数原型:

```
int SDF_ExportEncPublicKey_ECC(
    void *hSessionHandle,
    unsigned int uiKeyIndex,
    ECCrefPublicKey *pucPublicKey
);
```

参数说明:

- hSessionHandle: 输入: 会话句柄。
- uiKeyIndex: 输入: ECC 密钥索引号 1-10000。
- pucPublicKey: 输出: ECC 公钥结构。

返回值说明:

- 0 成功。
- 非 0 失败, 返回错误代码。返回码的取值范围参见附录 A。

4.4 SDF_GenerateKeyWithEPK_ECC

功能:

生成会话密钥并用外部 ECC 公钥加密输出, 同时返回密钥句柄。

函数原型:

```
int SDF_GenerateKeyWithEPK_ECC (
    void *hSessionHandle,
    unsigned int uiKeyBits,
    unsigned int uiAlgID,
```

```

    ECCrefPublicKey *pucPublicKey,
    ECCCipher *pucKey,
    void **phKeyHandle
);

```

参数说明:

- hSessionHandle: 输入: 会话句柄。
- uiKeyBits: 输入: 指定产生的会话密钥长度(128)。
- uiAlgID: 输入: 外部 ECC 公钥的算法标识。SGD_SM2、SGD_SM2_3
- pucPublicKey: 输入: 输入的外部 ECC 公钥结构。Bits=256
- pucKey: 输出: 缓冲区指针, 用于存放返回的密钥密文。
- phKeyHandle: 输出: 返回的密钥句柄。

返回值说明:

- | | |
|-----|----------------------------|
| 0 | 成功。 |
| 非 0 | 失败, 返回错误代码。返回码的取值范围参见附录 A。 |

4.5 SDF_GenerateKeyWithIPK_ECC**功能:**

生成会话密钥并用内部 ECC 公钥加密输出, 同时返回密钥句柄。

函数原型:

```

int SDF_GenerateKeyWithIPK_ECC (
    void *hSessionHandle,
    unsigned int uiIPKIndex,
    unsigned int uiKeyBits,
    ECCCipher *pucKey,
    void **phKeyHandle
);

```

参数说明:

- hSessionHandle: 输入: 会话句柄。
- uiIPKIndex: 输入: 密码设备内部存储公钥的索引值 1-10000。
- uiKeyBits: 输入: 指定产生的会话密钥长度(128)。
- pucKey: 输出: 缓冲区指针, 用于存放返回的密钥密文。
- phKeyHandle: 输出: 返回的密钥句柄。

返回值说明:

- | | |
|-----|----------------------------|
| 0 | 成功。 |
| 非 0 | 失败, 返回错误代码。返回码的取值范围参见附录 A。 |

4.6 SDF_ImportKeyWithISK_ECC**功能:**

导入会话密钥并用内部 ECC 加密私钥解密, 同时返回密钥句柄。

函数原型:

```
int SDF_ImportKeyWithISK_ECC (
    void *hSessionHandle,
    unsigned int uiISKIndex,
    ECCCipher *pucKey,
    void **phKeyHandle
);
```

参数说明:

- hSessionHandle: 输入: 会话句柄。
- uiISKIndex: 输入: 密码设备内部存储加密私钥的索引值, 对应于加密时的公钥。1-10000
- pucKey: 输入: 缓冲区指针, 用于存放返回的密钥密文。
- phKeyHandle: 输出: 返回的密钥句柄。

返回值说明:

- 0 成功。
- 非 0 失败, 返回错误代码。返回码的取值范围参见附录 A。

4.7 SDF_GenerateAgreementDataWithECC**功能:**

生成密钥协商参数并输出。

函数原型:

```
int SDF_GenerateAgreementDataWithECC (
    void *hSessionHandle,
    unsigned int uiISKIndex,
    unsigned int uiKeyBits,
    unsigned char *pucSponsorID,
    unsigned int uiSponsorIDLength,
    ECCrefPublicKey *pucSponsorPublicKey,
    ECCrefPublicKey *pucSponsorTmpPublicKey,
    void **phAgreementHandle
);
```

参数说明:

- hSessionHandle: 输入: 会话句柄。
- uiISKIndex: 输入: 密码设备内部存储加密私钥的索引值, 该私钥用于参与密钥协商。1-10000
- uiKeyBits: 输入: 指定产生的会话密钥长度(128)。
- pucSponsorID: 输入: 参与密钥协商的发起方 ID 值。
- uiSponsorIDLength: 输入: 发起方 ID 长度。取值范围 1-32
- pucSponsorPublicKey: 输出: 返回的发起方 ECC 公钥结构。
- pucSponsorTmpPublicKey: 输出: 返回的发起方临时 ECC 公钥结构。

- `phAgreementHandle`: 输出：返回的协商句柄，用于计算协商密钥。

返回值说明：

- 0 成功。
- 非 0 失败，返回错误代码。返回码的取值范围参见附录 A。

4.8 SDF_GenerateKeyWithECC**功能：**

使用 ECC 密钥协商算法，使用自身协商句柄和响应方的协商参数计算会话密钥，同时返回会话密钥句柄。

函数原型：

```
int SDF_GenerateKeyWithECC (
    void *hSessionHandle,
    unsigned char *pucResponseID,
    unsigned int uiResponseIDLength,
    ECCrefPublicKey *pucResponsePublicKey,
    ECCrefPublicKey *pucResponseTmpPublicKey,
    void *hAgreementHandle,
    void **phKeyHandle
);
```

参数说明：

- `hSessionHandle`: 输入：会话句柄。
- `pucResponseID`: 输入：外部输入的响应方 ID 值。
- `uiResponseIDLength`: 输入：外部输入的响应方 ID 长度。取值范围 1-32
- `pucResponsePublicKey`: 输入：外部输入的响应方 ECC 公钥结构。
- `pucResponseTmpPublicKey`: 输入：外部输入的响应方临时 ECC 公钥结构。
- `hAgreementHandle`: 输入：协商句柄，用于计算协商密钥。
- `phKeyHandle`: 输出：返回的密钥句柄。

返回值说明：

- 0 成功。
- 非 0 失败，返回错误代码。返回码的取值范围参见附录 A。

4.9 SDF_GenerateAgreementDataAndKeyWithECC**功能：**

使用 ECC 密钥协商算法，产生协商参数并计算会话密钥，同时返回产生的协商参数和和密钥句柄。

函数原型：

```
int SDF_GenerateAgreementDataAndKeyWithECC (
    void *hSessionHandle,
    unsigned int uiISKIndex,
    unsigned int uiKeyBits,
```

```

    unsigned char *pucResponseID,
    unsigned int uiResponseIDLength,
    unsigned char *pucSponsorID,
    unsigned int uiSponsorIDLength,
    ECCrefPublicKey *pucSponsorPublicKey,
    ECCrefPublicKey *pucSponsorTmpPublicKey,
    ECCrefPublicKey *pucResponsePublicKey,
    ECCrefPublicKey *pucResponseTmpPublicKey,
    void **phKeyHandle
);

```

参数说明:

- hSessionHandle: 输入: 会话句柄。
- uiISKIndex: 输入: 密码设备内部存储加密私钥的索引值, 该私钥用于参与密钥协商。1-10000
- uiKeyBits: 输入: 协商后要求输出的密钥长度(128)。
- pucResponseID: 输入: 响应方 ID 值。
- uiResponseIDLength: 输入: 响应方 ID 长度。取值范围 1-32
- pucSponsorID: 输入: 发起方 ID 值。
- uiSponsorIDLength: 输入: 发起方 ID 长度。取值范围 1-32
- pucSponsorPublicKey, 输入: 外部输入的发起方 ECC 公钥结构
- pucSponsorTmpPublicKey, 输入: 外部输入的发起方临时 ECC 公钥结构
- pucResponsePublicKey 输出: 返回的响应方 ECC 公钥结构
- pucResponseTmpPublicKey 输出: 返回的响应方临时 ECC 公钥结构
- phKeyHandle 输出: 返回的密钥句柄

返回值说明:

- 0 成功。
- 非 0 失败, 返回错误代码。返回码的取值范围参见附录 A。

4.10 SDF_ExchangeDigitEnvelopeBaseOnECC

功能:

基于 ECC 算法的数字信封转换。

函数原型:

```

int SDF_ExchangeDigitEnvelopeBaseOnECC(
    void *hSessionHandle,
    unsigned int uiKeyIndex,
    unsigned int uiAlgID,
    ECCrefPublicKey *pucPublicKey,
    ECCCipher *pucEncDataIn,
    ECCCipher *pucEncDataOut
);

```


参数说明:

- hSessionHandle[in]: 输入: 会话句柄
- uiKeyIndex[in]: 输入: 密码设备存储的 ECC 密钥对索引值 1-10000
- uiAlgID[in]: 输入: 外部 ECC 公钥的算法标识 SGD_SM2、SGD_SM2_3
- pucPublicKey[in]: 输入: 外部 ECC 公钥结构
- pucEncDataIn[in]: 输入: 缓冲区指针, 用于存放输入的会话密钥密文
- pucEncDataOut[out]: 输出: 缓冲区指针, 用于存放输出的会话密钥密文

返回值说明:

- 0 成功。
- 非 0 失败, 返回错误代码。返回码的取值范围参见附录 A。

4.11 SDF_GenerateKeyWithKEK**功能:**

生成会话密钥并用密钥加密密钥加密输出, 同时返回密钥句柄。

函数原型:

```
int SDF_GenerateKeyWithKEK( void *hSessionHandle,
    unsigned int uiKeyBits,
    unsigned int uiAlgID,
    unsigned int uiKEKIndex,
    unsigned char *pucKey,
    unsigned int *puiKeyLength,
    void **phKeyHandle
);
```

参数说明:

- hSessionHandle[in]: 输入: 会话句柄
- uiKeyBits [in]: 输入: 指定产生的会话密钥长度 128
- uiAlgID[in]: 输入: 对称加密算法标识 SGD_SM4
- uiKEKIndex [in]: 输入: 密码设备存储的密钥加密密钥索引值 1-10000
- pucKey [out]: 输出: 缓冲区指针, 用于存放返回的会话密钥密文
- puiKeyLength [out]: 输出: 返回的密钥密文长度
- phKeyHandle [out]: 输出: 返回的密钥句柄

返回值说明:

- 0 成功。
- 非 0 失败, 返回错误代码。返回码的取值范围参见附录 A。

4.12 SDF_ImportKeyWithKEK**功能:**

导入会话密钥并用密钥加密密钥解密, 同时返回会话密钥句柄。

函数原型:

```
int SDF_ImportKeyWithKEK( void *hSessionHandle,
    unsigned int uiAlgID,
    unsigned int uiKEKIndex,
    unsigned char *pucKey,
    unsigned int uiKeyLength,
    void **phKeyHandle
);
```

参数说明:

- hSessionHandle[in]: 输入: 会话句柄
- uiAlgID [in]: 输入: 对称加密算法标识 取值 SGD_SM4
- uiKEKIndex [in]: 输入: 内部密钥加密密钥索引值 1-10000
- pucKey [in]: 输入: 存放输入的会话密钥密文
- uiKeyLength[in] 输入: 输入的密钥密文长度取值 16
- phKeyHandle[out] 输出: 返回的密钥句柄

返回值说明:

- 0 成功。
- 非 0 失败, 返回错误代码。返回码的取值范围参见附录 A。

4.13 SDF_ImportKey**功能:**

导入明文会话密钥, 同时返回密钥句柄。

函数原型:

```
int SDF_ImportKey(
    void *hSessionHandle,
    unsigned char *pucKey,
    unsigned int uiKeyLength,
    void **phKeyHandle
);
```

参数说明:

- hSessionHandle[in]: 输入: 会话句柄
- pucKey [in]: 输入: 缓冲区指针, 用于存放输入的密钥明文
- uiKeyLength [in]: 输入: 输入的密钥明文长度=16
- phKeyHandle [out]: 输出: 返回的密钥句柄

返回值说明:

- 0 成功。
- 非 0 失败, 返回错误代码。返回码的取值范围参见附录 A。

4.14 SDF_DestroyKey

功能:

销毁会话密钥，并释放为密钥句柄分配的内存等资源。

函数原型:

```
int SDF_DestroyKey (
    void *hSessionHandle,
    void *hKeyHandle
);
```

参数说明:

- hSessionHandle[in]: 输入：会话句柄
- hKeyHandle [in]: 输入：会话密钥句柄

返回值说明:

- 0 成功。
- 非 0 失败，返回错误代码。返回码的取值范围参见附录 A。

5 非对称运算类接口

非对称算法运算类接口包含以下具体函数，各函数返回值见附录 A 错误代码定义：

- A. 使用外部 ECC 私钥对数据进行签名运算：SDF_ExternalSign_ECC
- B. 使用外部 ECC 公钥对数据进行验证签名运算：SDF_ExternalVerify_ECC
- C. 使用内部 ECC 私钥对数据进行签名运算：SDF_InternalSign_ECC
- D. 使用内部 ECC 公钥对 ECC 签名值进行验证运算：SDF_InternalVerify_ECC
- E. 使用外部 ECC 公钥对数据进行加密运算：SDF_ExternalEncrypt_ECC
- F. 使用外部 ECC 私钥对数据进行解密运算：SDF_ExternalDecrypt_ECC
- G. 使用内部 ECC 私钥对数据进行解密运算：SDT_InternalDecrypt_ECC

5.1 SDF_ExternalSign_ECC

功能:

使用外部 ECC 私钥对数据进行签名运算。

函数原型:

```
int SDF_ExternalSign_ECC(void *hSessionHandle,
    unsigned int uiAlgID,
    ECCrefPrivateKey *pucPrivateKey,
    unsigned char *pucData,
```

```

    unsigned int    uiDataLength,
    ECCSignature *pucSignature
);

```

参数说明:

- hSessionHandle[in]: 输入: 会话句柄
- uiAlgID[in]: 输入: 算法标识, 指定使用的 ECC 算法 SGD_SM2、SGD_SM2_1
- pucPrivateKey [in]: 输入: 外部 ECC 私钥结构
- pucData [in]: 输入: 缓冲区指针, 用于存放外部输入的数据
- uiDataLength [in]: 输入: 输入的数据长度, 必须是 32 位摘要值
- pucSignature [out]: 输出: 缓冲区指针, 用于存放输出的签名值数据

返回值说明:

- 0 成功。
- 非 0 失败, 返回错误代码。返回码的取值范围参见附录 A。

5.2 SDF_ExternalVerify_ECC

功能:

使用外部 ECC 公钥对数据进行验证签名运算。

函数原型:

```

int SDF_ExternalVerify_ECC(
    void *hSessionHandle,
    unsigned int uiAlgID,
    ECCrefPublicKey *pucPublicKey,
    unsigned char *pucDataInput,
    unsigned int uiInputLength,
    ECCSignature *pucSignature
);

```

参数说明:

- hSessionHandle[in]: 输入: 会话句柄
- uiAlgID[in]: 输入: 算法标识, 指定使用的 ECC 算法 SGD_SM2、SGD_SM2_1
- pucPublicKey[in]: 输入: 外部 ECC 公钥结构
- pucDataInput[in]: 输入: 缓冲区指针, 用于存放外部输入的数据
- uiInputLength[in]: 输入: 输入的数据长度, 必须是 32 位摘要值
- pucSignature[in]: 输入: 缓冲区指针, 用于存放输入的签名值数据

返回值说明:

- 0 成功。
- 非 0 失败, 返回错误代码。返回码的取值范围参见附录 A。

5.3 SDF_InternalSign_ECC

功能:

使用内部 ECC 私钥对数据进行签名运算。

函数原型:

```
int SDF_InternalSign_ECC(
    void *hSessionHandle,
    unsigned int    uiISKIndex,
    unsigned char *pucData,
    unsigned int    uiDataLength,
    ECCSignature *pucSignature
);
```

参数说明:

- hSessionHandle[in]: 输入: 会话句柄
- uiISKIndex[in]: 输入: 密码设备存储的ECC签名私钥的索引值 1-10000
- pucData[in]: 输入: 缓冲区指针, 用于存放外部输入的数据
- uiDataLength[in]: 输入: 输入的数据长度, 必须是 32 位摘要值
- pucSignature[out]: 输出: 缓冲区指针, 用于存放输出的签名值数据

返回值说明:

- 0 成功。
- 非 0 失败, 返回错误代码。返回码的取值范围参见附录 A。

5.4 SDF_InternalVerify_ECC

功能:

使用内部 ECC 公钥对 ECC 签名值进行验证运算。

函数原型:

```
int SDF_InternalVerify_ECC(
    void *hSessionHandle,
    unsigned int    uiISKIndex,
    unsigned char *pucData,
    unsigned int    uiDataLength,
    ECCSignature *pucSignature
);
```

参数说明:

- hSessionHandle[in]: 输入: 会话句柄
- uiISKIndex[in]: 输入: 密码设备存储的ECC签名私钥的索引值 1-10000
- pucData[in]: 输入: 缓冲区指针, 用于存放外部输入的数据
- uiDataLength[in]: 输入: 输入的数据长度, 必须是 32 位摘要值
- pucSignature[in]: 输入: 缓冲区指针, 用于存放输出的签名值数据

返回值说明:

- | | |
|-----|---------------------------|
| 0 | 成功。 |
| 非 0 | 失败，返回错误代码。返回码的取值范围参见附录 A。 |

5.5 SDF_ExternalEncrypt_ECC**功能:**

使用外部 ECC 公钥对数据进行加密运算。

函数原型:

```
int SDF_ExternalEncrypt_ECC(
    void *hSessionHandle,
    unsigned int uiAlgID,
    ECCrefPublicKey *pucPublicKey,
    unsigned char *pucData,
    unsigned int uiDataLength,
    ECCCipher *pucEncData
);
```

参数说明:

- | | |
|-----------------------|--|
| ● hSessionHandle[in]: | 输入: 会话句柄 |
| ● uiAlgID[in]: | 输入: 算法标识, 指定使用的 ECC 算法 SGD_SM2、SGD_SM2_3 |
| ● pucPublicKey[in]: | 输入: 外部 ECC 公钥结构 新定义的结构 |
| ● pucData[in]: | 输入: 缓冲区指针, 用于存放外部输入的数据 |
| ● uiDataLength[in]: | 输入: 输入的数据长度, 1-1440 |
| ● pucEncData[out]: | 输出: 缓冲区指针, 用于存放输出的数据密文 |

返回值说明:

- | | |
|-----|---------------------------|
| 0 | 成功。 |
| 非 0 | 失败，返回错误代码。返回码的取值范围参见附录 A。 |

5.6 SDF_ExternalDecrypt_ECC**功能:**

使用外部 ECC 私钥对数据进行解密运算。

函数原型:

```
int SDF_ExternalDecrypt_ECC (
    void *hSessionHandle,
    unsigned int uiAlgID,
    ECCrefPrivateKey *pucPrivateKey,
    ECCCipher *pucEncData,
    Unsigned char *pucData,
```

```

        unsigned int *puiDataLength
    );

```

参数说明:

- hSessionHandle 输入: 会话句柄
- uiAlgID 输入: 算法标识,指定使用的 ECC 算法 SGD_SM2、SGD_SM2_3
- pucPrivateKey 输入: 外部 ECC 私钥结构
- pucEncData 输入: 缓冲区指针,用于存放输入的数据密文
- pucData 输出: 缓冲区指针,用于存放输出的数据明文
- puiDataLength 输出: 输出的数据明文长度

返回值说明:

- 0 成功。
- 非 0 失败, 返回错误代码。返回码的取值范围参见附录 A。

6 对称算法运算类接口

对称算法运算类接口包含以下具体函数, 各函数返回值见附录 A 错误代码定义:

- A. 使用指定的密钥句柄和 IV 对数据进行对称加密运算: SDF_Encrypt
- B. 使用指定的密钥句柄和 IV 对数据进行对称解密运算: SDF_Decrypt
- C. 使用指定的密钥句柄和 IV 对数据进行 MAC 运算: SDF_CalculateMAC

6.1 SDF_Encrypt

功能:

使用指定的密钥句柄和 IV 对数据进行对称加密运算。

函数原型:

```

int SDF_Encrypt(
    void *hSessionHandle,
    void *hKeyHandle,
    unsigned int uiAlgID,
    unsigned char *pucIV,
    unsigned char *pucData,
    unsigned int uiDataLength,
    unsigned char *pucEncData,
    unsigned int *puiEncDataLength
);

```

参数说明:

- hSessionHandle 输入: 会话句柄
- hKeyHandle 输入: 指定的密钥句柄
- uiAlgID 输入: 算法标识, 指定对称加密算法 SM4
ECB/CBC/OFB/CFB 的组合 eg.SGD_SM4_ECB
- pucIV 输入+输出: 缓冲区指针, 用于存放输入和返回的 IV 数据
- pucData 输入: 缓冲区指针, 用于存放输入的数据明文
- uiDataLength 输入: 输入的数据明文长度, 16 字节整数倍(CFB 和 OFB
模式可以不是 16 的整数倍)
- pucEncData 输出: 缓冲区指针, 用于存放输出的数据密文
- puiEncDataLength 输出: 输出的数据密文长度

返回值说明:

- 0 成功。
- 非 0 失败, 返回错误代码。返回码的取值范围参见附录 A。

6.2 SDF_Decrypt**功能:**

使用指定的密钥句柄和 IV 对数据进行对称解密运算。

函数原型:

```
int SDF_Decrypt (
    void *hSessionHandle,
    void *hKeyHandle,
    unsigned int uiAlgID,
    unsigned char *pucIV,
    unsigned char *pucEncData,
    unsigned int uiEncDataLength,
    unsigned char *pucData,
    unsigned int *puiDataLength
);
```

参数说明:

- hSessionHandle 输入: 会话句柄
- hKeyHandle 输入: 指定的密钥句柄
- uiAlgID 输入: 算法标识, 指定对称加密算法 SM4
ECB/CBC/OFB/CFB 的组合 eg.SGD_SM4_ECB
- pucIV 输入+输出: 缓冲区指针, 用于存放输入和返回的 IV 数据
- pucEncData 输入: 缓冲区指针, 用于存放输入的数据密文
- uiEncDataLength 输入: 输入的数据密文长度, 16 字节整数倍(CFB 和 OFB
模式可以不是 16 的整数倍)
- pucData 输出: 缓冲区指针, 用于存放输出的数据明文
- puiDataLength 输出: 输出的数据明文长度

返回值说明:

0	成功。
非 0	失败，返回错误代码。返回码的取值范围参见附录 A。

6.3 SDF_CalculateMAC**功能:**

使用指定的密钥句柄和 IV 对数据进行 MAC 运算。

函数原型:

```
int SDF_CalculateMAC(
    void *hSessionHandle,
    void *hKeyHandle,
    unsigned int uiAlgID,
    unsigned char *pucIV,
    unsigned char *pucData,
    unsigned int uiDataLength,
    unsigned char *pucMAC,
    unsigned int *puiMACLength
);
```

参数说明:

● hSessionHandle	输入：会话句柄
● hKeyHandle	输入：指定的密钥句柄
● uiAlgID	输入：算法标识，指定 MAC 加密算法 SGD_SM3
● pucIV	输入+输出：缓冲区指针，用于存放输入和返回的 IV 数据
● pucData	输入：输入数据，
● uiDataLength	输入：输入数据长度 1-1440，16 字节整数倍
● pucMAC	输出：MAC 值
● puiMACLength	输出：MAC 值长度，8 字节

返回值说明:

0	成功。
非 0	失败，返回错误代码。返回码的取值范围参见附录 A。

7 杂凑运算类接口

杂凑运算类接口包含以下具体函数，各函数返回值见附录 A 错误代码定义：

- A. 杂凑运算初始化接口： SDF_HashInit
- B. 多包杂凑运算接口： SDF_HashUpdate
- C. 杂凑运算结束接口： SDF_HashFinal

7.1 SDF_HashInit

功能：

杂凑运算初始化。

函数原型：

```
int SDF_HashInit(
    void *hSessionHandle,
    unsigned int uiAlgID,
    ECCrefPublicKey *pucPublicKey,
    unsigned char *pucID,
    unsigned int uiIDLength
);
```

参数说明：

- hSessionHandle 输入：会话句柄
- uiAlgID 输入：指定杂凑算法标识
- pucPublicKey 输入：签名者公钥。当 uiAlgID 为 SGD_SM3 时有效
- pucID 输入：签名者的 ID 值，当 uiAlgID 为 SGD_SM3 时有效
- uiIDLength 输入：签名者 ID 的长度(1-32 字符)，当 uiAlgID 为 SGD_SM3 时有效

返回值说明：

- 0 成功。
- 非 0 失败，返回错误代码。返回码的取值范围参见附录 A。

7.2 SDF_HashUpdate

功能：

多包杂凑运算。

函数原型：

```
int SDF_HashUpdate(
    void *hSessionHandle,
    unsigned char *pucData,
    unsigned int uiDataLength
);
```

参数说明：

- hSessionHandle 输入：会话句柄
- pucData 输入：缓冲区指针，用于存放输入的数据明文
- uiDataLength 输入：输入的数据明文长度

返回值说明：

- 0 成功。
- 非 0 失败，返回错误代码。返回码的取值范围参见附录 A。

7.3 SDF_HashFinal

功能:

杂凑运算结束。

函数原型:

```
int SDF_HashFinal(
    void *hSessionHandle,
    unsigned char *pucHash,
    unsigned int *puiHashLength
);
```

参数说明:

- hSessionHandle 输入: 会话句柄
- pucHash 输出: 缓冲区指针, 用于存放输出的杂凑数据
- puiHashLength 输出: 返回的杂凑数据长度

返回值说明:

- 0 成功。
- 非 0 失败, 返回错误代码。返回码的取值范围参见附录 A。

8 用户文件操作类接口

用户文件类接口包含以下具体函数, 各函数返回值见附录 A 错误代码定义:

- A. 在设备创建指定文件名的文件: SDF_CreateFile
- B. 读取设备中指定文件名文件的内容: SDF_ReadFile
- C. 向设备中指定文件名的文件写入内容: SDF_WriteFile
- D. 从设备删除指定文件名的文件: SDF_DeleteFile

8.1 SDF_CreateFile

功能:

在设备创建指定文件名的文件。

函数原型:

```
int SDF_CreateFile(
    void *hSessionHandle,
    unsigned char *pucFileName,
    unsigned int uiNameLen,
    unsigned int uiFileSize
);
```

参数说明:

- hSessionHandle 输入: 会话句柄
- pucFileName 输入: 指向文件名存储缓冲区
- uiNameLen 输入: 文件名长度 1-64
- uiFileSize 输入: 创建文件的长度

返回值说明:

- 0 成功。
- 非 0 失败, 返回错误代码。返回码的取值范围参见附录 A。

8.2 SDF_ReadFile**功能:**

读取设备中指定文件名文件的内容。

函数原型:

```
int SDF_ReadFile(
    void *hSessionHandle,
    unsigned char *pucFileName,
    unsigned int uiNameLen,
    unsigned int uiOffset,
    unsigned int *puiFileLength,
    unsigned char *pucBuffer
);
```

参数说明:

- hSessionHandle 输入: 会话句柄
- pucFileName 输入: 指向文件名存储缓冲区
- uiNameLen 输入: 文件名长度 1-64
- uiOffset 输入: 读取文件的偏移地址
- puiFileLength 输入+输出: 请求读取的长度; 实际读取的长度。
- pucBuffer 输出: 输出缓冲区指针, 存储读取的文件内容

返回值说明:

- 0 成功。
- 非 0 失败, 返回错误代码。返回码的取值范围参见附录 A。

8.3 SDF_WriteFile**功能:**

向设备中指定文件名的文件写入内容。

函数原型:

```
int SDF_WriteFile (
    void *hSessionHandle,
    unsigned char *pucFileName,
    unsigned int uiNameLen,
    unsigned int uiOffset,
    unsigned int uiFileLength,
    unsigned char *pucBuffer
);
```

参数说明:

- hSessionHandle 输入: 会话句柄
- pucFileName 输入: 指向文件名存储缓冲区
- uiNameLen 输入: 文件名长度 1-64
- uiOffset 输入: 写入文件的偏移地址
- puiFileLength 输入: 指定写入文件内容的长度 (单次写入长度最大不得超过 8192)
- pucBuffer 输入: 缓冲区指针, 存储写入的文件数据

返回值说明:

- 0 成功。
- 非 0 失败, 返回错误代码。返回码的取值范围参见附录 A。

8.4 SDF_DeleteFile

功能:

从设备删除指定文件名的文件。

函数原型:

```
int SDF_DeleteFile(
    void *hSessionHandle,
    unsigned char *pucFileName,
    unsigned int uiNameLen
);
```

参数说明:

- hSessionHandle 输入: 会话句柄
- pucFileName 输入: 指向文件名存储缓冲区
- uiNameLen 输入: 文件名长度 1-64

返回值说明:

- 0 成功。
- 非 0 失败, 返回错误代码。返回码的取值范围参见附录 A。

附录A 常见返回码注释

错误码定义	值	描述
SDR_OK	0x0	操作成功
SDR_BASE	0x01000000	错误码基础值
SDR_UNKNOWERR	(SDR_BASE + 0x00000001)	未知错误
SDR_NOTSUPPORT	(SDR_BASE + 0x00000002)	不支持的接口调用
SDR_COMMFAIL	(SDR_BASE + 0x00000003)	与设备通信错误
SDR_HARDFAIL	(SDR_BASE + 0x00000004)	运算模块无响应
SDR_OPENDEVICE	(SDR_BASE + 0x00000005)	打开设备失败
SDR_OPENSESSION	(SDR_BASE + 0x00000006)	创建会话失败
SDR_PARDENY	(SDR_BASE + 0x00000007)	无私钥使用权限
SDR_KEYNOTEXIST	(SDR_BASE + 0x00000008)	不存在的密钥调用
SDR_ALGNOTSUPPORT	(SDR_BASE + 0x00000009)	不支持的算法调用
SDR_ALGMODNOTSUPPORT	(SDR_BASE + 0x0000000A)	不支持的算法模式调用
SDR_PKOPERR	(SDR_BASE + 0x0000000B)	公钥运算失败
SDR_SKOPERR	(SDR_BASE + 0x0000000C)	私钥运算失败
SDR_SIGNERR	(SDR_BASE + 0x0000000D)	签名运算失败
SDR_VERIFYERR	(SDR_BASE + 0x0000000E)	验证签名失败
SDR_SYMOPERR	(SDR_BASE + 0x0000000F)	对称算法运算失败
SDR_STEPERR	(SDR_BASE + 0x00000010)	多步运算步骤错误
SDR_FILESIZEERR	(SDR_BASE + 0x00000011)	文件长度超出限制
SDR_FILENOEXIST	(SDR_BASE + 0x00000012)	指定的文件不存在
SDR_FILEOFSERR	(SDR_BASE + 0x00000013)	文件起始位置错误
SDR_KEYTYPEERR	(SDR_BASE + 0x00000014)	密钥类型错误
SDR_KEYERR	(SDR_BASE + 0x00000015)	密钥错误
SDR_ENCDATAERR	(SDR_BASE + 0x00000016)	ECC 加密数据错误
SDR_RANDERR	(SDR_BASE + 0x00000017)	随机数产生错误

SDR_PRKRERR	(SDR_BASE + 0x00000018)	私钥使用权限获取失败
SDR_MACERR	(SDR_BASE + 0x00000019)	MAC 运算失败
SDR_FILEEXISTS	(SDR_BASE + 0x0000001A)	指定文件已存在
SDR_FILEWERR	(SDR_BASE + 0x0000001B)	文件写入失败
SDR_NOBUFFER	(SDR_BASE + 0x0000001C)	存储空间不足
SDR_INARGERR	(SDR_BASE + 0x0000001D)	输入参数错误
SDR_OUTARGERR	(SDR_BASE + 0x0000001E)	输出参数错误
SDR_HASHERR	(SDR_BASE + 0x0000001F)	杂凑运算错误
SDR_SESSHANDLE	(SDR_BASE + 0x00000020)	会话句柄错
SDR_KEYHANDLE	(SDR_BASE + 0x00000021)	密钥句柄错
SDR_DEVSTATE	(SDR_BASE + 0x00000022)	设备状态错

附录B 配置文件

[client]

CONN_TIMEOUT=3000 //连接超时时间

RW_TIMEOUT=3000 //读写超时时间

[server]

IP1=xxx.xxx.xxx.xxx //密码机 IP 地址

PORT1=9190 //密码机服务端口

SRVCOUNT=1 //密码服务数量