

灵锐 I

函数参考

2012年9月

Version 1.0.0.0

许可协议

同意本许可协议的所有条款及此处包含的任何补充或特殊的许可条款是获得本产品许可的必要条件。如果您不同意此协议的所有条款，请在三天内将产品退还北京深思洛克软件技术股份有限公司。您对本软件的使用将表明您同意接受本协议中条款的约束。

1. 授予您使用许可权。您可以为了备份的目的而复制磁盘中的软件，可以为了将本产品集成到您的软件的目的，根据本产品的文档说明将我们提供的软件合并进您的程序中。
2. 除已按上述第一条被授权外，不可以复制、修改、逆向工程、分解或重组该产品的全部或部分，不可向他人销售、租借、许可、转让、分发全部或部分本产品或本协议授予的权利。
3. 保证在自产品交给您之日起的 12 个月内，在正常使用情况下，产品不会出现实质性的材料上和生产制造上的缺陷。北京深思洛克软件技术股份有限公司的全部责任和您能获得的全部补救措施为：可选择尝试更换或修理或其他补救措施。
4. 除了上述对本产品的原始购买者所提供的有限保证之外，不向任何人作任何其他的保证。对北京深思洛克软件技术股份有限公司的产品、性能或服务亦没有明示的或暗示的或其他任何形式的保证，包括但不限于商品的适销性和对特定用途的适用性。
5. 任何情况下，无论如何引起及依据何种责任理论，均不负担任何因使用或不能使用本产品造成的损失责任，包括：丢失数据、损失利润及其他特别的、偶然的、附随的、继发的或间接的损失。
6. 所有的产品，包括灵锐 I 设备、软件、文档、与本产品一并附送的其他材料及您制作的备份的所有权与版权均属于北京深思洛克软件技术股份有限公司。
7. 违反上述条款时，本协议的授权将自动终止。

“灵锐”是北京深思洛克软件技术股份有限公司的注册商标。

本文所涉及的其他产品和公司名称可能是各自相应所有者的商标。

联系深思洛克

公司名称：北京深思洛克软件技术股份有限公司

办公地点：北京市海淀区中关村南大街甲 6 号铸诚大厦 B 座 1201 室

邮 编：100086

电 话：+86-10-51581366

传 真：+86-10-51581365

电子邮件：sense@sense.com.cn

网 址：<http://www.sense.com.cn>

阅读指南

【手册目标】

本手册主要对北京深思洛克软件技术股份有限公司开发的灵锐 I 加密锁的使用进行说明，由于实际情况千变万化，本手册很难一次做到面面俱到，需要逐渐完善。

【手册约定】



手册中出现该标志的地方表示需要您引起高度重视，否则可能会引发严重的后果。



手册中出现该标志的地方表示需要您特别引起注意的内容。

目 录

第 1 章 灵锐 I 接口函数参考.....	1
1.1 LIV_open.....	1
1.2 LIV_close.....	2
1.3 LIV_passwd.....	3
1.4 LIV_read.....	4
1.5 LIV_write.....	5
1.6 LIV_encrypt.....	6
1.7 LIV_decrypt.....	7
1.8 LIV_set_passwd.....	8
1.9 LIV_change_passwd.....	9
1.10 LIV_get_hardware_info.....	10
1.11 LIV_get_software_info.....	11
1.12 LIV_get_version_info.....	12
1.13 LIV_hmac.....	13
1.14 LIV_hmac_software.....	14
1.15 LIV_update.....	15
1.16 LIV_set_key.....	16
1.17 LIV_gen_update_pkg.....	17
1.18 接口函数错误代码.....	18

第 1 章 灵锐 I 接口函数参考

1.1 LIV_open

功能:

根据开发商编号和索引，打开符合条件的设备。

声明:

```
int LIVAPI LIV_open(  
    IN    int    vendor,  
    IN    int    index,  
    OUT   int    *handle  
);
```

参数:

vendor	[in]	开发商编号 (0=所有)
index	[in]	设备索引 (0=第一个, 依次类推)
handle	[out]	返回的设备句柄

返回值:

1. 未找到设备，函数返回 LIV_FIND_DEVICE_FAILED。
2. 找到相应设备但不能打开，函数返回 LIV_OPEN_DEVICE_FAILED。
3. 找到相应设备并成功打开，函数返回 LIV_SUCCESS，handle 返回设备句柄。

备注:

该函数打开设备的方式为独占打开。

1.2 LIV_close

功能:

关闭已经打开的设备。

声明:

```
int LIVAPI LIV_close(  
    IN int handle  
);
```

参数:

handle [in] 打开的设备句柄

返回值:

如果关闭相应设备成功，函数返回 LIV_SUCCESS，否则返回错误代码。

备注:

1. 设备不能重复关闭。
2. 关闭设备的同时将清除密码的校验状态。

1.3 LIV_passwd

功能:

校验密码。

声明:

```
int LIVAPI LIV_passwd(  
    IN int handle,  
    IN int type,  
    IN unsigned char *passwd  
);
```

参数:

handle	[in] 打开的设备句柄
type	[in] 密码类型 (0: 管理密码; 1: 普通用户密码; 2: 认证密码)
passwd	[in] 密码 (8 字节)

返回值:

1. 密码错误, 函数返回 LIV_INVALID_PASSWORD。
2. 校验密码成功, 函数返回 LIV_SUCCESS。

备注:

成功校验密码后即可获得相应的权限。

1.4 LIV_read

功能:

读取指定区块的数据。

声明:

```
int LIVAPI LIV_read(  
    IN    int          handle,  
    IN    int          block,  
    OUT   unsigned char *buffer  
);
```

参数:

handle	[in]	打开的设备句柄
block	[in]	要读取的区块号 (0~3)
buffer	[out]	读取数据缓冲区 (必须大于等于 512 字节)

返回值:

如果读取成功, 函数返回 LIV_SUCCESS, 否则返回错误代码。

备注:

需要先校验密码, 任何权限均可。

1.5 LIV_write

功能:

向指定区块内写入数据。

声明:

```
int LIVAPI LIV_write(  
    IN int handle,  
    IN int block,  
    IN unsigned char *buffer  
);
```

参数:

handle	[in] 打开的设备句柄
block	[in] 要写入的区块号 (0~3)
buffer	[in] 写入数据缓冲区 (必须大于等于 512 字节)

返回值:

如果写入成功，函数返回 LIV_SUCCESS，否则返回错误代码。

备注:

需要先校验密码，写区块 0 任何权限均可，写区块 1 到区块 3 需要管理权限。

1.6 LIV_encrypt

功能:

使用设备内置的 AES 算法加密数据。

声明:

```
int LIVAPI LIV_encrypt(  
    IN    int          handle,  
    IN    unsigned char *plaintext,  
    OUT   unsigned char *ciphertext  
);
```

参数:

handle	[in]	打开的设备句柄
plaintext	[in]	要加密的明文数据 (16 字节)
ciphertext	[out]	加密后的密文数据 (16 字节)

返回值:

如果加密成功, 函数返回 LIV_SUCCESS, 否则返回错误代码。

备注:

需要普通用户权限。

1.7 LIV_decrypt

功能:

使用设备内置的 AES 算法解密数据。

声明:

```
int LIVAPI LIV_decrypt(  
    IN    int          handle,  
    IN    unsigned char *ciphertext,  
    OUT   unsigned char *plaintext  
);
```

参数:

handle	[in]	打开的设备句柄
ciphertext	[in]	要解密的密文数据（16 字节）
plaintext	[out]	解密后的明文数据（16 字节）

返回值:

如果写入成功，函数返回 LIV_SUCCESS, 否则返回错误代码。

备注:

需要普通用户权限。

1.8 LIV_set_passwd

功能:

设置新密码。

声明:

```
int LIVAPI LIV_set_passwd(  
    IN int handle,  
    IN int type,  
    IN unsigned char *passwd,  
    IN int retries  
);
```

参数:

handle	[in]	打开的设备句柄
type	[in]	密码类型 (0: 管理密码; 1: 普通用户密码; 2: 认证密码)
passwd	[in]	新密码 (8 字节)
retries	[in]	密码的最大尝试次数 (1~15), -1 表示无尝试次数

返回值:

如果设置成功, 函数返回 LIV_SUCCESS, 否则返回错误代码。

备注:

1. 需要管理权限。
2. 当灵锐 I 设备的硬件版本低于 V1.1.0 (不含) 时, 参数 “retries” 设置的尝试次数只对认证密码有效。当设备的硬件版本高于 V1.1.0 (含) 时, 参数 “retries” 设置的尝试次数对三种密码均有效。

1.9 LIV_change_passwd

功能:

修改认证密码。

声明:

```
int LIVAPI LIV_change_passwd(  
    IN int handle,  
    IN int type,  
    IN unsigned char *oldpasswd,  
    IN unsigned char *newpasswd  
);
```

参数:

handle	[in] 打开的设备句柄
type	[in] 密码类型 (2: 认证密码)
oldpasswd	[in] 旧密码 (8 字节)
newpasswd	[in] 新密码 (8 字节)

返回值:

如果修改成功, 函数返回 LIV_SUCCESS, 否则返回错误代码。

备注:

1. 需要认证权限。
2. 该函数只能修改认证密码, 管理密码和普通用户密码需要使用 [LIV_set_passwd](#) 函数进行修改。

1.10 LIV_get_hardware_info

功能:

获取设备的硬件信息。

声明:

```
int LIVAPI LIV_get_hardware_info(
    IN    int          handle,
    OUT   LIV_hardware_info *info
);
```

参数:

handle [in] 打开的设备句柄
info [out] 存储硬件信息的结构体指针

返回值:

如果获取成功，函数返回 LIV_SUCCESS，否则返回错误代码。

备注:

1. 不需要权限。
2. LIV_hardware_info 结构体用于存放设备的硬件信息，其定义如下：

```
typedef struct {
    int    developerNumber;
    char   serialNumber[8];
    int    setDate;
    int    reservation;
}LIV_hardware_info;
```

结构体成员变量说明如下：

developerNumber	开发商编号
serialNumber	全球唯一序列号
setDate	生产日期
reservation	保留

1.11 LIV_get_software_info

功能:

获取库版本信息。

声明:

```
int LIVAPI LIV_get_software_info(  
    OUT LIV_software_info *info  
);
```

参数:

info [out] 存储库版本信息的结构体指针

返回值:

如果获取成功，函数返回 LIV_SUCCESS，否则返回错误代码。

备注:

1. 可以无锁操作。

2. LIV_software_info 结构体用于存放库版本的信息，其定义如下：

```
typedef struct {  
    int version;        // 软件版本  
    int reservation;    // 保留  
} LIV_software_info;
```

结构体成员变量说明如下：

version	软件版本
reservation	保留

1.12 LIV_get_version_info

功能:

获取硬件相关版本信息。

声明:

```
int LIVAPI LIV_get_version_info(
    IN    int          handle,
    OUT   LIV_version_info *versioninfo
);
```

参数:

handle [in] 打开的设备
versioninfo [out] 硬件相关版本信息

返回值:

如果获取成功，函数返回 LIV_SUCCESS，否则返回错误代码。

备注:

1. 不需要权限。
2. LIV_version_info 结构体用于存放硬件相关版本的信息，其定义如下：

```
typedef struct {
    char    reserved[4];    //保留
    char    version[3];     //硬件相关版本
} LIV_version_info;
```

结构体成员变量说明如下：

reserved	保留
version	硬件相关版本

1.13 LIV_hmac

功能:

硬件计算 HMAC 值。

声明:

```
int LIVAPI LIV_hmac(
    IN    int          iHandle,
    IN    unsigned char *pucText,
    IN    int          iTextlen,
    OUT   unsigned char *pucDigest
);
```

参数:

iHandle	[in]	打开的设备句柄
pucText	[in]	要计算 hmac 值的数据
iTextlen	[in]	pucText 的长度（按字节，>=0）
pucDigest	[out]	hmac 值（20 字节）

返回值:

如果计算成功，函数返回 LIV_SUCCESS，否则返回错误代码。

备注:

需要认证权限。

1.14 LIV_hmac_software

功能:

软件计算 HMAC 值。

声明:

```
int LIVAPI LIV_hmac_software(  
    IN    unsigned char    *pucText,  
    IN    int               iTextlen,  
    IN    unsigned char    *pucKey  
    OUT   unsigned char    *pucDigest  
);
```

参数:

pucText	[in]	要计算 hmac 值的数据
iTextlen	[in]	pucText 的长度（按字节，>=0）
pucKey	[in]	hmac 算法的 Key（20 字节）
pucDigest	[out]	hmac 值（20 字节）

返回值:

如果计算成功，函数返回 LIV_SUCCESS，否则返回错误代码。

备注:

1. 可以无锁操作。
2. 调用此函数进行认证时，参数“pucKey”的值必须与锁内设置的身份认证密钥相同。

1.15 LIV_update

功能:

使用远程升级包升级指定设备。

声明:

```
int LIVAPI LIV_update(  
    IN int handle,  
    IN unsigned char *buffer  
);
```

参数:

handle	[in] 打开的设备句柄
buffer	[in] 升级包缓冲区

返回值:

如果升级成功，函数返回 LIV_SUCCESS，否则返回错误代码。

备注:

1. 需要普通用户权限。
2. 在升级过程中请不要拔掉锁。

1.16 LIV_set_key

功能:

设置密钥。

声明:

```
int LIVAPI LIV_set_key(  
    IN int handle,  
    IN int type,  
    IN unsigned char *key  
);
```

参数:

handle	[in] 打开的设备句柄
type	[in] 密钥类型 (0: 远程升级密钥; 1: 认证密钥)
key	[in] 密钥 (20 字节)

返回值:

如果设置成功，函数返回 LIV_SUCCESS，否则返回错误代码。

备注:

需要管理权限。

1.17 LIV_gen_update_pkg

功能:

生成远程升级包。

声明:

```
int LIVAPI LIV_gen_update_pkg(  
    IN    unsigned char    *serial,  
    IN    int              block,  
    IN    unsigned char    *buffer,  
    IN    unsigned char    *key,  
    OUT   unsigned char    *uptPkg  
);
```

参数:

serial	[in]	待升级锁的设备序列号
block	[in]	需要升级的区块号（1-3 区块）
buffer	[in]	升级内容
key	[in]	远程升级密钥
uptPkg	[out]	生成的升级包（大于 549 字节）

返回值:

如果生成成功，函数返回 LIV_SUCCESS，否则返回错误代码。

备注:

可以无锁操作。

1.18 接口函数错误代码

表 1-1 接口函数错误代码

错误代码	值	说明
LIV_SUCCESS	0	函数执行成功
LIV_OPEN_DEVICE_FAILED	1	打开设备失败
LIV_FIND_DEVICE_FAILED	2	未找到符合条件的设备
LIV_INVALID_PARAMETER	3	参数错误
LIV_INVALID_BLOCK_NUMBER	4	区块号错误
LIV_HARDWARE_COMMUNICATE_ERROR	5	通讯错误
LIV_INVALID_PASSWORD	6	密码错误
LIV_ACCESS_DENIED	7	没有权限
LIV_ALREADY_OPENED	8	设备已经打开
LIV_ALLOCATE_MEMORY_FAILED	9	内存分配失败
LIV_INVALID_UPDATE_PACKAGE	10	不合法的升级包
LIV_SYN_ERROR	11	线程同步错误
LIV_OTHER_ERROR	12	其它未知错误
LIV_OPERATION_NOT_SUPPORTED	13	不支持的操作
LIV_DEVICE_BLOCKED	14	设备被锁