

ET99 多功能锁 API 开发手册

V1.0 版

版权所有© 2006 坚石诚信科技有限公司

<http://www.jansh.com.cn>

北京坚石诚信科技有限公司（以下简称坚石）尽最大努力使这篇文章中的内容完善且正确。坚石对于由这篇文档导致的任何形式的直接或间接损失不负有责任。这篇文章的内容会跟随产品的升级而有所变化。

修改记录：

日期	版本	修改
2012 年 12 月	1.0	第一版

坚石诚信科技有限公司

软件开发协议

坚石诚信科技有限公司（以下简称坚石）的所有产品，包括但不限于：开发工具包，磁盘，光盘，硬件设备和文档，以及未来的所有订单都受本协议的制约。

1. 许可使用

您可以将本软件合并、连接到您的计算机程序中，但其目的只是如使用手册中描述的那样保护您的程序或进行网络身份认证。您可以以备份为目的复制合理数量的拷贝。

2. 禁止使用

除在条款 1 中特别允许的之外，不得复制、反向工程、反汇编、反编译、修改、增加、改进软件、硬件和产品的其它部分。禁止对软件 and 产品的任何部分进行反向工程，禁止推导软件的源代码。禁止使用产品中的磁盘或光盘来传播、存储非本产品的原始内容的任何信息或由坚石提供的产品的任何升级。禁止将软件放在公共服务器上传播。

3. 有限担保

坚石保证在自产品发给您之日起的 12 个月内，在正常的使用情况下，硬件和软件存储介质没有重大的工艺和材料上的缺陷。

4. 修理限度

当根据本协议提出索赔时，坚石唯一的责任就是根据实际情况，免费进行替换或维修。坚石对被替换下来的任何产品部件都享有所有权。

保修索赔单必须在担保期内写好，在发生故障 14 天内连同令人信服的证据交给坚石诚信科技公司。往返运费需由客户承担。

除了在本协议中保证的担保之外,坚石诚信科技公司不再提供特别的或隐含的担保,也不再对本协议中所描述的产品负其它责任,包括它们的质量,性能和对某一特定目的的适应性。

5. 责任限度

不管因为什么原因,不管是因合同中的规定还是由于刑事的原因,包括疏忽的原因,而使您及任何一方受到了损失,由我方产品所造成的损失或该产品是起诉的原因或与起诉有间接关系,坚石诚信科技公司对您及任何一方所承担的全部责任不超出您购买该产品所支付的货款。在任何情况下,坚石诚信科技公司对于由于您不履行责任所导致的损失,或对于数据、利润、储蓄或其它的后续的和偶然的损失,即使坚石诚信科技公司被建议有这种损失的可能性,或您根据第 3 方的索赔而提出的任何索赔均不负责任。

6. 协议终止

当您不能遵守本协议所规定的条款时,将终止您的许可和本协议。但条款 2, 3, 4, 5 将继续有效。

目 录

第一章 概述	1
第二章 API 函数接口	2
2.1 查找 ET99	2
2.2 打开锁	2
2.3 关闭设备	3
2.4 读存储区	4
2.5 写存储区	5
2.6 产生产品标识.....	6
2.7 产生随机数	7
2.8 产生超级用户 PIN 码.....	8
2.9 重置普通用户 PIN 码.....	9
2.10 设置密钥	10
2.11 纯软件 HMAC_MD5 接口.....	11
2.12 计算 HMAC_MD5.....	12
2.13 验证 PIN 码.....	13
2.14 修改用户 PIN 码.....	14
2.15 重置安全状态.....	15
2.16 获得硬件序列号.....	16
2.17 配置设备	17
2.18 打开 LED 灯.....	18
2.19 关闭 LED 灯.....	18
第三章 常量定义	20
3.1 接口函数的返回值.....	20
3.2 PIN 码标志	20
3.3 数据区的读写标志.....	21
3.4 常量 PID	21

第一章 概述

该文档详细讲述 API 函数的接口定义以及 API 中的常量定义，有关 ET99 多功能锁的详细介绍，请参阅《ET99 多功能锁用户手册》。

第二章 API 函数接口

2.1 查找 ET99

```
et_FindToken(  
    unsigned char* pid,  
    int * count  
)
```

功能说明：

查找计算机上指定 pid 的 ET99 个数。

参数：

pid: [in]产品标识, 为固定长度 8 个字节的字符串；

count: [out]还回的设备个数；

返回值：

ET_SUCCESS: 执行成功,Count 为查找到的 ET99 的数目。

ET_UNIT_NOT_FOUND: 没有可以用的硬件，此时 Count 值为 0。

2.2 打开锁

```
et_OpenToken(  
    ET_HANDLE* hHandle,  
    unsigned char*pid,  
    int ind
```


)

功能说明:

打开指定 PID 的硬件,由 `index` 指定打开硬件的索引,`index` 应该小于等于找到的 Token 数目。进入匿名用户状态。

参数:

`hHandle`: [out]打开设备的句柄, 返回给用户, 供以后的函数调用;
`pid`: [in]输入的硬件设备的 `pid`, 为固定长度 8 个字节的字符串;
`index`: [in]打开第 `index` 个硬件设备。

返回值:

`ET_SUCCESS`: 执行成功。
`ET_UNIT_NOT_FOUND`: 打开指定的设备失败。

2.3 关闭设备

```
et_CloseToken(
    ET_HANDLE hHandle
)
```

功能说明:

关闭指定的设备。

参数:

`hHandle`: [in] 设备句柄。

返回值:

ET_SUCCESS: 关闭成功。

ET_COMMUNICATIONS_ERROR: 没有打开设备。

2.4 读存储区

```
et_Read(  
    ET_HANDLE hHandle,  
    WORD offset,  
    int Len,  
    unsigned char* pucReadBuf  
)
```

功能说明:

从指定的位置, 读取指定的数据到指定的 BUFFER 中。此函数调用需要有 User 权限, 且调用以后不改变安全状态。

参数:

hHandle: [in]设备句柄

Offset: [in]偏移量

Len: [in]长度, 每次操作不能超过 60 字节, 如果超过则需要分块操作。

pucReadBuf: [out]读出的数据存放此缓存区中, 调用者保证缓冲区大小至少是 Len, 否则可能产生系统存取异常。

返回值:

ET_SUCCESS: 表示成功。

ET_INVALID_PARAMETER: 无效的参数。

ET_NOT_SET_PID: 没有设置 PID。

ET_ACCESS_DENY: 权限不够。

ET_COMMUNICATIONS_ERROR: 没有打开设备。

分块操作: 假如要写入\读取120个字节的数字, 那么可以使用循环操作进行分块操作,
第1次: 从偏移位置为“0”的地址起始写入\读取到第59的位置, 这样前60个字节就
完成写入\读取的操作。

第2次: 从偏移位置为60的地址开始写入\读取, 完成后60字节的操作。到此120字节便
已全部完成写入\读取

以此类推……………

第i次: ……………

第i+1次: ……………

↓

第i+n次: ……………

2.5 写存储区

```
et_Write(
    ET_HANDLE hHandle,
    WORD offset,
    int Len,
    unsigned char* pucWriteBuf
)
```

功能说明:

将 buf 中, Length 长的数据写到指定的偏移。有存取权限控制。匿名状态不可用, 且在普通用户状态时还需要检查设备的配置。不改变安全状态。

参数:

hHandle: [in]设备句柄;

Offset: [in]偏移;

Len: [in]长度, 不能超过 60, 如果超过则需要写多次;

pucWriteBuf: [in]等写入的数据缓存区指针;

返回值:

ET_SUCCESS: 表示成功。

ET_HARD_ERROR: 硬件错误

ET_INVALID_PARAMETER: 无效的参数。

ET_NOT_SET_PID: 没有设置 PID。

ET_ACCESS_DENY: 权限不够。

ET_COMMUNICATIONS_ERROR: 没有打开设备。

2.6 产生产品标识

```
et_GenPID(  
    ET_HANDLE hHandle,  
    int SeedLen,  
    unsigned char* pucSeed,  
    unsigned char* pid  
)
```

功能说明:

根据参数中指定的种子, 产生产品标识。种子长度不能超过 51 个字节。必须在超级用户状态下才能用, 调用以后不改变安全状态。

参数:

hHandle: [in]设备句柄;
pucSeed: [in]种子;
SeedLen: [in]种子长度, 小于等于 51。数字、英文、汉字都可
pid: [out]产生的产品标识, 为固定长度 8 个字节的字符串;

返回值:

ET_SUCCESS: 表示成功;
ET_HARD_ERROR: 硬件错误
ET_INVALID_PARAMETER: 无效的参数;
ET_ACCESS_DENY: 权限不够, 需要先验证 SOPIN。
ET_COMMUNICATIONS_ERROR: 没有打开设备。

2.7 产生随机数

```
et_GenRandom(  
    ET_HANDLE hHandle,  
    unsigned char* pucRandBuf  
)
```

功能说明:

产生 16 字节的随机数, 放到参数指定的 BUF 中。调用者需要保护 BUF 至少 16 字节, 否则会产生系统的存取异常。该函数在匿名状态不可用, 且在函数调用以后, 安全状态不变。

参数:

hHandle: [in]设备句柄

pucRandBuf: [out]等写入的数据缓存区指针

返回值:

ET_SUCCESS: 表示成功。

ET_NOT_SET_PID: 没有设置 PID

ET_INVALID_PARAMETER: 无效的参数。

ET_ACCESS_DENY: 权限不够。

ET_COMMUNICATIONS_ERROR: 没有打开设备。

2.8 产生超级用户 PIN 码

```
et_GenSOPIN(  
    ET_HANDLE hHandle,  
    int SeedLen,  
    unsigned char* pucSeed,  
    unsigned char* pucNewSoPIN  
)
```

功能说明:

根据参数中指定的种子, 产生超级密码, 放到参数指定的存储区 newSOPIN 中。种子码的长度不能超过 51 个字节。调用者需要保证 newSOPIN 至少 16 字节, 否则会产生系统的存取异常。只能在超级用户状态下调用, 调用成功后, 安全状态返回到匿名状态。因为此时 SOPIN 已经改变, 以前所有校验过的状态应该失效, 如果需要超级用户状态, 再使用新的 SOPIN, 重新校验。

参数:

hHandle: [in]设备句柄

pucSeed: [in]产生超级用户密码需要的种子。

SeedLen: [in]种子长度, 小于等于 51

pucNewSoPIN: [out]用于存放产生的超级用户密码的缓冲区指针, 至少可容纳 16 字节。

返回值:

ET_SUCCESS: 表示成功。

ET_HARD_ERROR: 硬件错误

ET_INVALID_PARAMETER: 无效的参数。

ET_ACCESS_DENY: 权限不够。

ET_COMMUNICATIONS_ERROR: 没有打开设备。

2.9 重置普通用户 PIN 码

et_ResetPIN(

```
    ET_HANDLE hHandle,  
    unsigned char* pucSoPIN  
)
```

功能说明:

重新设置普通用户密码为 16 个 ‘F’, 相当于解锁。命令执行成功后, 当前安全状态变成超级用户状态。

参数:

hHandle: [in]设备句柄

pucSoPIN: [in]超级用户密码, 16 字节。

返回值:

ET_SUCCESS: 表示成功。

ET_HARD_ERROR: 硬件错误

ET_NOT_SET_PID: 没有设置 PID

ET_INVALID_PARAMETER: 无效的参数。

ET_COMMUNICATIONS_ERROR: 没有打开设备。

注意:

如果验证超级 PIN 码错误, 并且错误值在 0xF0 和 ET_PIN_ERR_MAX(0xFF) 之间, 我们可以通过错误码&ET_PIN_ERR_MASK(0x0F)得到剩余重试次数。如果还回 0xF0 表示已经被锁死, 如果还回 0xFF 表示验证出错, 且 pin 永远不被锁死。

2.10 设置密钥

```
et_SetKey(  
    ET_HANDLE hHandle,  
    int Keyid,  
    unsigned char* pucKeyBuf  
)
```

功能说明:

更新参数指定的密钥, 此密钥是用于计算 HMAC—MD5 的。其中 KEY 的获得, 是通过一个纯软件接口 HMAC_MD5 (), 参见相应说明。匿名状态不可用, 且在普通用户状态时还需要检查设备配置。不改变安全状态。

参数:

hHandle: [in]设备句柄

Keyid: [in]密钥指示, 取值范围 (1—8)

pucKeyBuf: [in]KEY 缓存区指针, KEY 固定为 32 字节。

返回值:

ET_SUCCESS: 表示成功;

ET_NOT_SET_PID: 没有设置 PID

ET_HARD_ERROR: 硬件错误

ET_INVALID_PARAMETER: 无效的参数;

ET_COMMUNICATIONS_ERROR: 没有打开设备。

2.11 纯软件 HMAC_MD5 接口

MD5_HMAC(

```
    unsigned char * pucText,  
    unsigned long   ulText_Len,  
    unsigned char * pucKey,  
    unsigned long   ulKey_Len,  
    unsigned char * pucToenKey,  
    unsigned char * pucDigest  
)
```

功能说明:

标准 HMAC_MD5 的软件实现, 参照 RFC2104 标准。可用于产生 TOKEN 硬件需要的 KEY。产生的计算结果, 可以与其它系统 (如服务器端系统) 进行对比, 以验证算法的正确性。匿名状态可用不改变安全状态。

参数:

pucText:[in]等处理的数据缓存区指针, 大于 0 小于等于 51 个字节

ulText_Len : [in]数据长度, 大于 0 小于等于 51

pucKey: [in]密钥, 按标准 RFC2104, 长度可以任意

ulKey_Len: [in]密钥长度

pucToenKey: [out]硬件计算需要的 KEY, 固定 32 字节。

PucDigest: [out]计算结果, 固定 16 字节。

返回值:

ET_SUCCESS: 表示成功。

ET_INVALID_PARAMETER: 无效的参数。

2.12 计算 HMAC_MD5

```
et_HMAC_MD5(  
    ET_HANDLE hHandle,  
    int keyID,  
    int textLen,  
    unsigned char* pucText,  
    unsigned char *digest  
)
```

功能说明:

利用硬件计算 HMAC-MD5 , pid 为出厂时, 还回错误。权限等同于 KEY 的读权限。不改变安全状态。

参数:

hHandle: [in]设备句柄

keyID: [in]密钥指示, 范围 (1—8)
pucText: [in]待计算的数据, 大于 0 小于等于 51 个字节
textLen: [in]数据长度, 大于 0 小于等于 51
digest: [out]散列结果的数据指针, 固定长度 16 个字节。

返回值:

ET_SUCCESS: 表示成功;
ET_NOT_SET_PID: 没有设置 PID;
ET_INVALID_PARAMETER: 无效的参数;
ET_COMMUNICATIONS_ERROR: 没有打开设备。、

2.13 验证 PIN 码

```
et_Verify(  
    ET_HANDLE hHandle,  
    int Flags,  
    unsigned char* pucPIN  
)
```

功能说明:

验证密码, 以获得相应的安全状态, 不受安全状态限制, 验证成功以后, 进入相应的安全状态。

参数:

hHandle: [in]设备句柄;
Flags: [in]验证 PIN 的类型, 见下表;
pucPIN: [in] PIN 码, 固定长度 16 个字节。

Flag	意义
ET_VERIFY_USER_PIN	验证的是普通用户PIN码，如果验证通过，则进入普通用户状态。
ET_VERIFY_SO_PIN	验证的是超级用户PIN码，如果验证通过，则进入超级用户状态。

返回值：

ET_SUCCESS：表示成功；

ET_INVALID_PARAMETER：无效的参数；

ET_COMMUNICATIONS_ERROR：没有打开设备。

注意：

如果验证 USER PIN 码或者超 SO PIN 错误，如果验证超级 PIN 码错误，并且错误值在 0xF0 和 ET_PIN_ERR_MAX（0xFF）之间,我们可以通过错误码 &ET_PIN_ERR_MASK(0x0F)得到剩余重试次数。如果还回 0xF0 表示已经被锁死，如果还回 0xFF 表示验证出错，且 pin 永远不被锁死。

2.14 修改用户 PIN 码

```
et_ChangeUserPIN(  
    ET_HANDLE hHandle,  
    unsigned char* pucOldPIN,  
    unsigned char* pucNewPIN  
)
```

功能说明：

修改普通用户密码，从 pucOldPIN，改为 pucNewPIN。普通用户密码长度固定为 16 字节。此命令可以在匿名状态下进行，命令执行成功后，进入普通用户状

态。

参数:

hHandle: [in]设备句柄

pucOldPIN: [in]原来的密码, 长度固定为 16 字节

pucNewPIN: [in]新密码, 长度固定为 16 字节

返回值:

ET_SUCCESS: 表示成功。

ET_HARD_ERROR: 硬件错误

ET_NOT_SET_PID: 没有设置 PID

ET_INVALID_PARAMETER: 无效的参数。

ET_COMMUNICATIONS_ERROR: 没有打开设备。

注意:

如果验证超级 PIN 码错误, 并且错误值在 0xF0 和 ET_PIN_ERR_MAX(0xFF) 之间, 我们可以通过错误码 & ET_PIN_ERR_MASK(0x0F) 得到剩余重试次数。如果还回 0xF0 表示已经被锁死, 如果还回 0xFF 表示验证出错, 且 pin 永远不被锁死。

2.15 重置安全状态

```
et_ResetSecurityState(  
    ET_HANDLE hHandle  
)
```

功能说明:

重置安全状态, 回到匿名用户状态。

参数:

hHandle: [in]设备句柄

返回值:

ET_SUCCESS: 表示成功;

ET_COMMUNICATIONS_ERROR: 没有打开设备。

2.16 获得硬件序列号

```
et_GetSN(  
    ET_HANDLE hHandle,  
    unsigned char* pucSN  
)
```

功能说明:

获得硬件序列号。可以在匿名状态下进行。不改变安全状态。

参数:

hHandle: [in]设备句柄

pucSN: [out]用于存放获得的序列号, 长度固定为 8 字节

返回值:

ET_SUCCESS: 表示成功;

ET_INVALID_PARAMETER: 无效的参数;

ET_COMMUNICATIONS_ERROR: 没有打开设备。

2.17 配置设备

```
et_SetupToken(  
    ET_HANDLE hHandle,  
    BYTE bSoPINRetries,  
    BYTE bUserPINRetries,  
    BYTE bUserReadOnly,  
    BYTE bReserved  
)
```

功能说明：

对硬件进行配置。必须在超级用户状态下进行。不改变安全状态。

参数：

hHandle: [in]设备句柄

bSoPINRetries: [in]超级 PIN 码的重试次数，范围 0—15，0 表示永远不被锁死；

bUserPINRetries: [in]用户 PIN 码的重试次数，范围 0—15，0 表示永远不被锁死；

bUserReadOnly: [in]读写/只读标注，如下表；

bReserved: [in]保留字，必须为 0。

bUserReadOnly	意义
ET_USER_WRITE_READ	可读写
ET_USER_READ_ONLY	只读

返回值：

ET_SUCCESS: 表示成功。

ET_HARD_ERROR: 硬件错误。

ET_INVALID_PARAMETER: 无效的参数。

ET_ACCESS_DENY: 权限不够。

ET_COMMUNICATIONS_ERROR: 没有打开设备。

2.18 打开 LED 灯

```
et_TurnOnLED(  
    ET_HANDLE hHandle  
)
```

功能说明:

打开 LED 灯, 使其变亮。匿名状态不可用, 不改变安全状态。设备加电后, LED 灯是常亮的。

参数:

hHandle: [in]设备句柄

返回值:

ET_SUCCESS: 表示成功。

ET_ACCESS_DENY: 权限不够。

ET_COMMUNICATIONS_ERROR: 没有打开设备。

2.19 关闭 LED 灯

```
et_TurnOffLED(  
    ET_HANDLE hHandle  
)
```


功能说明:

关闭 LED 灯，使其熄灭。匿名状态不可用，不改变安全状态。设备加电后，LED 灯是常亮的。

参数:

hHandle: [in]设备句柄

返回值:

ET_SUCCESS: 表示成功。

ET_ACCESS_DENY: 权限不够。

ET_COMMUNICATIONS_ERROR: 没有打开设备。

第三章 常量定义

3.1 接口函数的返回值

常量	值	意义
ET_SUCCESS	0x00	函数执行成功
ET_ACCESS_DENY	0x01	访问被拒绝，权限不够
ET_COMMUNICATIONS_ERROR	0x02	通讯错误，没有打开设备
ET_INVALID_PARAMETER	0x03	无效的参数，参数出错
ET_NOT_SET_PID	0x04	没有设置 PID
ET_UNIT_NOT_FOUND	0x05	打开指定的设备失败
ET_HARD_ERROR	0x06	硬件错误
ET_UNKNOWN_ERROR	0x07	未知错误

另外，还有两个常量定义，帮助用户在验证 PIN 码失败时，用来计算 PIN 码重试次数。

ET_PIN_ERR_MASK	0x0F	验证 PIN 码掩码
ET_PIN_ERR_MAX	0xFF	验证 PIN 码错误且永远不锁死

3.2 PIN 码标志

常量	值	意义
ET_VERIFY_USERPIN	0	表示验证普通用户 pin
ET_VERIFY_SOPIN	1	表示验证超级用户 pin

3.3 数据区的读写标志

常量	值	意义
ET_USER_WRITE_READ	0	表示数据区可读写
ET_USER_READ_ONLY	1	表示数据区只允许读

3.4 常量 PID

CONST_PID “FFFFFFFF”