高端 POS 平台 PINPAD 模块应用开发指南

（v0.4）

福建联迪商用设备有限公司

Contents

1. 引言

1.1. 修订履历

日期	版本	修订内容	修订人
2015-09-15	0.1	创建	彭波涛
2016-12-21	0.2	升级完善	彭波涛
2016-12-22	0.3	根据培训和评审结果进行完善	彭波涛
2017-05-11	0.4	修正一些文字错误	彭波涛

1.2. 适用范围

本文档提供联迪高端 POS 平台 PINPAD 模块的应用开发指南，包括各种安全机制介绍、用法说明、安全开发注意事项。

采用该 PINPAD 模块的产品包括 EPT-AND、EPT-HQL、APOS 等平台的所有产品系列。

说明：本文档所有引用的 API 原型均来自于 NDK 接口层，如果应用做了再次封装（例如 SDK 接口层），应该将其替换成 SDK 中对应的接口原型。

1.3. 名词与术语

缩写	描述
AES	Advanced Encryption Standard
FK	Fixed Key
DUKPT	Derived Unique Key Per Transaction
MK/SK	Master Key and Session Key
PEK	Pin Encryption Key
MAK	MAC Key
TDK	Track Data Encryption Key
DEK	Data Encryption Key
DUKPT IK	DUKPT Initial Key
DUKPT FK	DUKPT Future Key
N/A	Not Applicable
PED	PIN Entry Device
PIN	Personal Identification Number
RSA	Rivest Shamir Adelman Algorithm
SHA	Secure Hash Algorithm
TDES	Triple Data Encryption Standard

IPP	Internal or Embedded PED of POS terminal
EPP	External PED of POS terminal
KAP	Key Arrangement Pool. It is a logical isolated key area for key storing.

1.4. 参考文档

[1] EPT-AND 应用开发指南.chm
[2] EPT-HQL 应用开发指南.chm
[3] PED 设备初始密钥装载指南.pdf
[4] PCI PTS 规范
[5] UPTS2.0 规范
[6] 《银联卡密码算法使用与密钥管理规范》（QCUP 058-2013）

2. PINPAD 模块

2.1. 功能简述

PINPAD 模块主要提供 PIN 输入及加密保护、密钥管理等功能，并提供相应的 API 给上层应用开发使用。

2.1.1. 主要特色

LANDI 高端 POS 平台相比于其他平台的 PINPAD 模块，主要存在如下的区别：

1、 新增了多密钥区隔离和认证的安全机制；
在我司旧平台的 POS 中，多个收单机构共用 100 组终端主密钥 TMK，各个收单机构相互协商好各自使用哪一组 MK，在这种情况下，对应用程序的安全性要求较高，如果没有相关的审核和签名机制，就存在多密钥误用和混用的安全风险。高端 POS 平台针对此情况，提供了多密钥区的机制，以便应用程序更加安全、合理的开发和使用多应用。

2、 对明文下载主密钥接口进行了完善，一方面提高使用的方面性，另一方面针对存在的安全风险进行了控制；

3、 结合 Android 平台自身的优势，对部分接口的使用方式进行了改变，从而从整体上提高系统的性能。

4、 对部分存在安全风险的接口进行了完善

5、 基于对 PCI/TR-31 等安全规范的充分理解，对 PINPAD 的 API 原型进行了较大的调整，相关接口设计更加安全，同时可扩展性更强。

6、 支持细化权限管理，根据应用开发者的角色不同，提供不同的权限供管理者使用。

2.1.2. 功能接口

本模块提供如下的 API：

- EA_pinpad_ucGetAllPinpadsInfo()
- EA_pinpad_ucOpenDevice()
- EA_pinpad_vCloseDevice()
- EA_pinpad_ucSetPinpadCfg()
- EA_pinpad_ucResetPinpad()
- EA_pinpad_ucFormatPinpad()
- EA_pinpad_ucGetPinpadInfo()
- EA_pinpad_ucSetPinpadSerialNum()
- EA_pinpad_ucDispPinpad()
- EA_pinpad_ucLocatePinpadLcd()
- EA_pinpad_ucPinpadBeep()
- EA_pinpad_ucGetRandom()
- EA_pinpad_ucAuthEncKeyKCV()
- EA_pinpad_ucLoadEncKey()
- EA_pinpad_ucGenerateKey()
- EA_pinpad_ucCheckKey()
- EA_pinpad_ucLoadDukptInitialKeyFromKap()
- EA_pinpad_ucDeleteKey()
- EA_pinpad_ucGetKcv()

- EA_pinpad_ucMacInit()
- EA_pinpad_ucMacLoadData()
- EA_pinpad_ucMacGenerate()
- EA_pinpad_ucMacVerify()
- EA_pinpad_ucCalculateDes()
- EA_pinpad_ucEncryptMagTrackData()
- EA_pinpad_ucGetExistentKapIds()
- EA_pinpad_ucCreateKap()
- EA_pinpad_ucDeleteKap()
- EA_pinpad_ucDeleteAllKaps()
- EA_pinpad_ucEraseAllKeysWithinKap()
- EA_pinpad_ucFormatKap()
- EA_pinpad_ucSetKapCfg()
- EA_pinpad_ucGetKapMode()
- EA_pinpad_ucGetKapInfo()
- EA_pinpad_ucGetExistentKeyIdsInKeySystem()
- EA_pinpad_ucStartPinEntry()
- EA_pinpad_ucStartOfflinePin()
- EA_pinpad_ucVerifyOfflinePin()
- EA_pinpad_ucCancelPinEntry()
- EA_pinpad_ucInjectPinEntryFuncKey()
- EA_pinpad_ucGetPinEntryInfo()
- EA_pinpad_ucStartPinEntryAndVerifyWithIcCard()

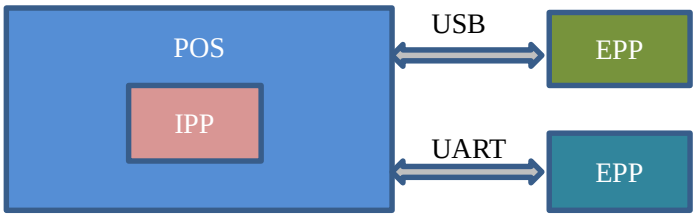
详见 NDK 应用开发指南的 PINPAD API 部分。

2.2. 主要机制介绍

2.2.1. PINPAD 设备与句柄（PINPAD Device）

终端将每个 PINPAD 设备看作一个虚拟的 PINPAD Device，通过设备句柄进行统一管理，使用设备前必须打开设备获得设备句柄，句柄相当于设备的使用权。

终端支持多个 PINPAD 设备，包括内置 PINPAD 和一种外置 PINPAD（可以是 USB 接口或者串口）。在使用 PINPAD 设备之前，用户必须通过相关 API 明确指定要访问哪种设备。



注意：由于终端可以支持多种外置密码键盘，因此到底使用哪一种是在开机阶段自动设别的。因此使用者必须将外置密码键盘插在 POS 主机上开机，才能确保 POS 机正确识别和使用。

相关 API:

如下 API 必须用到：

EA_pinpad_ucGetAllPinpadsInfo: 用于列举出检测到的所有可用的 PINPAD 设备
EA_pinpad_ucOpenDevice: 用于打开一种 PINPAD 设备并获得设备操作句柄
EA_pinpad_vCloseDevice: 用于关闭一种 PINPAD 设备并释放操作句柄

如下的 API 可选使用：

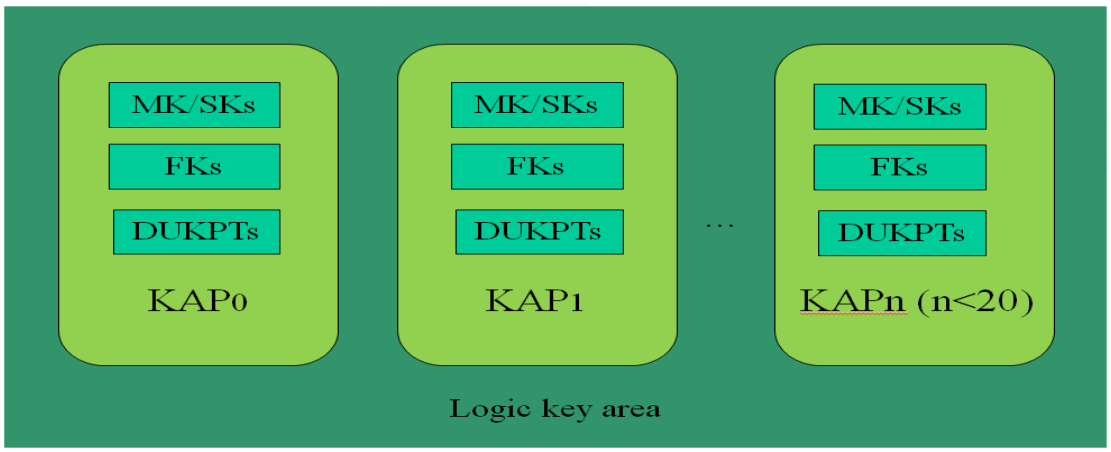
EA_pinpad_ucSetPinpadCfg：配置 PINPAD
EA_pinpad_ucResetPinpad: 复位 PINPAD 设备（仅逻辑复位）
EA_pinpad_ucFormatPinpad: 格式化 PINPAD（恢复到出厂状态）
EA_pinpad_ucGetPinpadInfo: 获取 PINPAD 设备的信息
EA_pinpad_ucSetPinpadSerialNum: 设置 PINPAD 设备的序列号
EA_pinpad_ucGetRandom: 从 PINPAD 设备读取一组随机数

2.2.2. 密钥区（KAP）

KAP: Key Arrangement Pool，密钥区.一种逻辑上隔离的密钥存储区域。每个用户可以使用一个或者多个 KAP

每款 PINPAD 设备最多可以同时支持 20 个 KAP，即每台 PINPAD 设备最多可以允许 20 个收单应用同时使用而不会冲突。

密钥存储区的结果示意图如下:



KAP 描述符（KAP Identifier）

引入如下的数据结构来唯一标识每个 KAP:

```
typedef struct ET_KapId {
```

```
    ET_RegionId tRegionId;
    ET_KapNum tKapNum;
} ET_KapId;

typedef ushort ET_RegionId;
typedef ushort ET_KapNum;
```

tRegionId：密钥区（Region）的索引，用于标识某个密钥使用者的管理机构，例如银行、第三方支付机构等。tRegionId=0，表示该管理机构未在联迪注册，应用无独立身份（未在联迪申请应用根证书，或者申请了未使用），而是由联迪进行统一托管，即联迪充当多应用管理者的身份。

tRegionId=0xFFFF，则代表所有 Region 的意思。在 APP 签名时，应用管理者可以为该 APP 指定可访问的 KAP ID 列表，如果其中某个 KAP ID 的 tRegionId=0xFFFF，则表示允许该 APP 访问所有 RegionId 的 KAP。正常情况下，一个 APP 的 RegionId 是和其签名 UKEY 关联的，并不能随意设置。只有部分特殊的 APP（例如厂商开发的中间件 APP），有权访问所有 KAP，这时候签名后台才会将其 RegionId 配置为 0xFFFF。

注意：tRegionId 由联迪统一分配，与客户申请的应用签名卡/UKEY 绑定，不同签名卡/UKEY 分配不同的 tRegionId，确保不同客户的密钥区的隔离。

tKapNum：密钥区组内号，用于标识某一个收单机构。每个 KAP 是一个安全上隔离和独立的密钥区，不同 KAP 有不同的访问权限要求。如果一个机构内部存在多个独立的收单业务和后台，则每个收单业务和后台都应该看作一个独立的收单机构，需确保他们之间密钥管理的隔离。

tKapNum=0，表示特定 RegionId 下的共享 KAP。对于该 KAP，只要是同一个 Region 的应用，则都可以共享访问。举例来说：如果一个 APP 签名时授予的可访问 KAP ID=0xAAAA0001，实际是表示该 APP 可以同时访问 0xAAAA0001 的私有 KAP 以及 0xAAAA0000 的组内共享 KAP 的意思。

tKapNum=0xFFFF，则表示特定 RegionId 下的所有 KAP。如果一个 APP 签名时被赋予的 KapNum=0xFFFF，则表示该 APP 可以访问该 Region 下的所有 KAP。举例来说，如果一个 APP 签名时授予的可访问 KAP ID=0xAAAAFFFF，实际是表示该 APP 可以访问取值范围为 0xAAAA0000~0xAAAAFFFF 的所有 KAP。

注意：

1） 在国内，tKapNum 由应用管理机构统一分配，以智能 POS 生态系统为例，由运营管理平台为各个收单机构分配不同的 tKapNum
2） 在国外，tKapNum 目前仍然由联迪统一分配。每个区域的安全管理员在申请 UKEY 时应该为不同收单机构申请不同的 tKapNum，由联迪将其与该 UKEY 进行关联绑定。

对于普通收单应用来说，建议应用只将密钥保存于自己的私有 KAP 内，确保只有自己可以访问。从而实现不同 APP 之间访问的 KAP 是相互独立和隔离的。

针对常见的一个应用需要用到多个 MK 的情况，该如何区分使用一个还是多 KAP 呢，建议遵循如下的原则：

1） CASE 1：如果一个应用有用到两个独立的密钥体系，各自有自己的初始密钥 TMK1 和 TMK2，且 TMK1 和 TMK2 都可以分别下载各自的 PIN KEY 来做 PIN 加密，则按照 PCI 和银联等安全规范的要求，TMK1、TMK2 需要隔离，他们对应的后台需看作两个独立的收单机构，分别使用不同的 KAP。
2） CASE 2：如果一个应用有用到两个密钥 MK1 和 MK2，但是他们同属于同一个密钥层次，共用

同一个根 TMK，且 MK1/MK2 是通过密文方式下载，则仍然看作同一个收单机构。

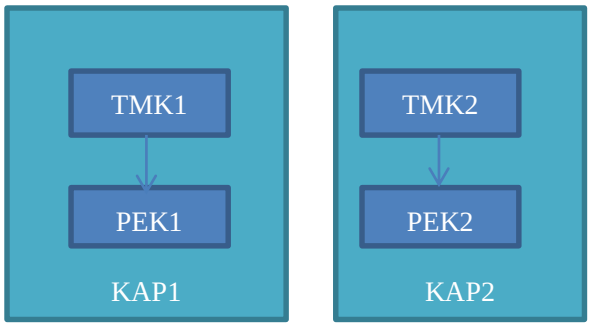
3） CASE 3：如果一个应用有用到两个独立的密钥层次，各自有自己的初始密钥 TMK1 和 TMK2，但是其中只有 TMK1 可用于下载 PIN KEY，另一个 MK 用于普通数据功能，则 TMK2 不看做一个独立的收单机构，则允许二者共用同一个 KAP 或多个 KAP。一般有几种做法：

a) 将 TMK1 和 TMK2 都存放于私有 KAP。
b) 将 TMK1 存放于私有 KAP，将 TMK2 存放于共享 KAP。
c) 将 TMK2 存放于私有 KAP，将 TMK1 存放于共享 KAP。
d) 将 TMK1/TMK2 都存放于共享 KAP。

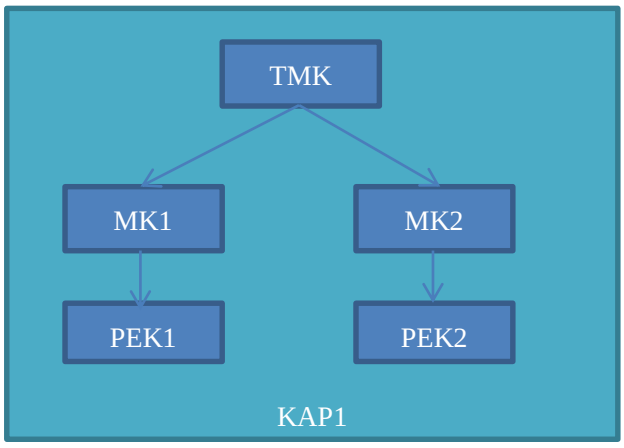
方案 a)和 b)都是符合安全要求的，只不过方案 a）的弊端是 TMK1/TMK2 要同时下载，一个私有 KAP 内已经下载过 TMK，后续就不允许通过 API 调用方式随意追加新的 TMK，除非是采用经过认证的方式（例如通过 LKI 母 POS 下载密钥，或者通过符合双向认证的 RKIS 下载密钥）下载新的 TMK，或者将 TMK1 擦除后再重新同时下载 TMK1 和 TMK2。
方案 c)不推荐，将重要的密钥放在共享区，不重要的放在私有区，是本末倒置的做法；
方案 d)也不推荐，因为共享区的密钥可以被同组 APP 共享访问，如非万不得已不推荐这种用法。

根据 PCI 以及银联等安全规范的要求，不同收单机构的密钥要求安全隔离，防止误用和混用。联迪 POS 的设计规则，是建议每个收单机构使用一个独立的 KAP，确保各各自密钥的安全隔离。针对上述三种情况，对密钥区（KAP）的使用要求分别如下：

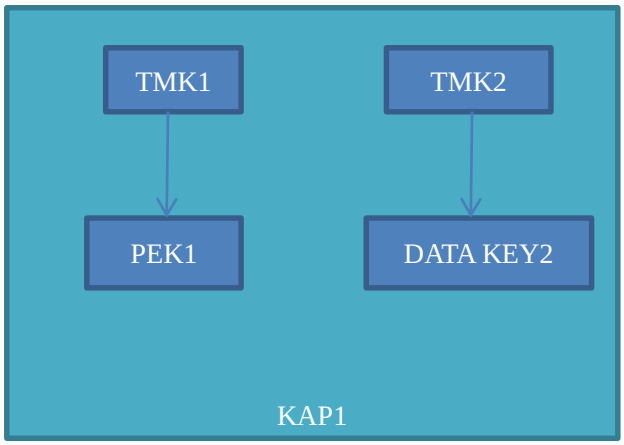
CASE 1：两个独立的密钥树，都可用于收单，则应该使用不同的私有 KAP



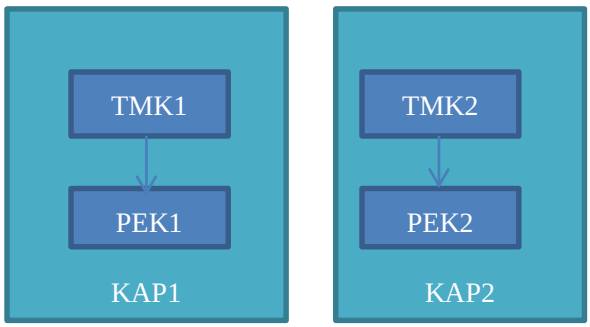
CASE 2：同一密钥树，可以使用同一个私有的 KAP



CASE 3：两个独立的密钥树，只有一个可以支持 PIN KEY，则允许使用同一个 KAP



也可以使用多个 KAP：



KAP 管理

KAP 的使用规则

KAP ID		KAP 性质	访问权限要求	明文初始密钥下载限制	多应用密钥隔离
RegionId	KapNum				
=0	0	全局共享密钥区	无要求，任何应用均可访问该 KAP	无，密钥区可随时添加新的初始明文密钥	无隔离
	[1~15]			有，首次下载密钥后，后续不允许未授权方式添加新的密钥，除非整个密钥区所有密钥清空或者格式化	无隔离

	[16~0xffff]	由联迪统一分配，目前预留			
[1,0xFFFE]	= 0x0000	每个用户组内部的共享密钥区	同组可访问，即只要是该用户组内用户的应用（使用同一套应用根证书），则都可以访问该密钥区	无，密钥区可随时添加新的初始明文密钥	组之间隔离
	!= 0x0000	每个用户独立的私有密钥区	由用户组管理者授权，只有拥有对应权限的应用才可访问。	有，首次下载密钥后，后续不允许未授权方式添加新的密钥，除非整个密钥区所有密钥清空或者格式化	组内隔离
=0xFFFF		作为被访问的 KAP 时，不允许有该取值； 作为可访问的 KAP 列表时，表示可以访问所有 Region			

注意：

1、 联迪 POS 机不仅提供安全隔离的私有 KAP，同时也提供共享 KAP，以满足某些应用的特殊要求：

2、 为了保证满足 PCI 以及银联相关安全规范的要求，要求每个收单机构将密钥下载到各自私有的 KAP 内，不要使用共享 KAP（包括全局共享和组内共享）。如果使用共享 KAP，则其他应用也能够对其进行访问或者覆盖，可能造成多个收单机构的密钥混用和误用，由此导致的不符合安全规范以及产生的安全风险，联迪不承担相关责任。收单机构有义务承担自己的密钥管理责任，保证自己的密钥不被其他非法机构利用。收单应用的开发者，代表着收单机构的利益，有义务正确合理的使用 KAP；本着“谁签名谁负责”的原则，应用程序签名者需承担应用程序代码审核和确认的作用，应用程序不规范导致的风险，由应用程序签名者进行负责。

KAP 的分配与管理

KAP 主要用于在多应用时，提供多个密钥区，实现各个应用的密钥隔离。KAP 的组成包括两部分：

1） 组号 tRegionId：由联迪统一分配，联迪为每个客户（可以是一个收单机构，也可以是多个收单机构的总机构）颁发唯一的应用根证书，用于唯一标识该客户的身份，根证书的 ID 被用作 KAP 的组号 tRegionId。后续所有安全隔离机制，都要依赖客户身份做隔离。联迪的分配规则确保了不同客户使用不同的根证书，确保不同客户之间的安全隔离。

2） 组内号 tKapNum：由联迪或客户指定。绝大多数情况下，每个收单机构会申请独立的签名卡，拥有一个合法客户身份，在该身份下只有一个收单业务和后台，因此 tKapNum 取默认值 1；少数情况下，一个客户下有多个收单业务，是以总机构的身份申请的签名卡，这时候总机构需要为各个子收单机构应用配置可访问的 tKapNum，总机构需要保证不同子收单机构使用不同的私有 KAP。

1） 如果多个应用分别属于不同的收单机构（第三方支付企业/银行等），则通过不同的签名卡区分和隔离；

---- 比如银商和拉卡拉，属于不同的机构。

2） 如果多个应用虽然属于同一个收单机构（第三方支付企业/银行等），但是属于不同的业务部门，没有统一的管理者，各自使用不同的签名卡/工具、不同的密钥下载设备/工具等，则建议把他们看做不同的子机构，同样通过不同的签名卡区分和隔离；

----比如银商的传统 POS 收单业务和全民付业务，可能也要看做两个独立的业务部门。

3） 如果多个应用虽然属于统一个收单机构（第三方支付企业/银行等），但是有统一的管理者，而且希望使用相同的签名卡/工具、相同的密钥下载设备/工具，则建议把他们看做一个机构，只发一套签名卡，内部的 KAP ID 由其管理者统一分配。

KAP 使用注意事项

1、 多个应用程序，建议尽量不要使用同一个密钥区，否则可能出现密钥下载时相互覆盖和冲突的情况；如果一定要使用，建议使用 GSK（全局共享 KAP）或者组内共享 KAP，但是要注意协商号各自的 MK ID 的分配，不然也会导致冲突；

2、 最安全的做法，是每个应用或所属机构，要申请一张单独的根证书，这样才能保证该应用具有独立的身份，使用的密钥区与其他应用可以相互隔离，防止其他非法应用往自己的密钥区中随意添加密钥。

相关 API:

EA_pinpad_ucGetExistentKapIds: 获取已经存在的 KAP 列表
EA_pinpad_ucCreateKap: 创建 KAP
EA_pinpad_ucDeleteKap: 删除 KAP
EA_pinpad_ucDeleteAllKaps: 删除所有 KAP
EA_pinpad_ucFormatKap: 格式化 KAP
EA_pinpad_ucSetKapCfg: 配置 KAP
EA_pinpad_ucGetKapInfo: 获取某个 KAP 的信息

上述 KAP 管理相关的 API，通常由密钥下载机构使用，其在下载初始密钥时，创建对应的 KAP，并下载好初始明文密钥，后续交易过程只需要使用密钥，通常不需要再对 KAP 进行操作，因此应用管理者根据不同的角色分配不同的权限（例如密钥下载程序开发者需要提供 KAP 管理操作的权限，普通应用开发者不需要 KAP 管理权限。详见权限管理章节）。

2.2.3. 密钥（Keys）

密钥标识符（Key Identifier）

数据结构 ET_KeyHandle 被用于唯一标识和描述一个 Key:

```
typedef struct ET_KeyHandle {
    ET_KapId tKapId;
    ET_KeySystem tKeySystem;
```

```
    ET_KeyId tKeyId;
} ET_KeyHandle;
```

密钥体系（Key System）

每个 KAP 可以存储 FK、MK/SK、DUKPT 三种密钥体系的密钥。

数据结构 ET_KeySystem 被用于描述一中密钥体系:

```
typedef char ET_KeySystem;
#define EM_pinpad_KM_MKSK ( (ET_KeySystem)(0) )
#define EM_pinpad_KM_DUKPT ( (ET_KeySystem)(1) )
#define EM_pinpad_KM_FIXED_KEY ( (ET_KeySystem)(2) )
```

密钥索引号（Key ID）

```
typedef ushort ET_KeyId;
#define EM_pinpad_MAX_KEY_ID ( (ET_KeyId)(0x00ff) )
```

注意：

1、 这里的索引号是同一种密钥体系内的密钥索引号，不同密钥体系内的索引号可以重复，取值范围均是[0，EM_pinpad_MAX_KEY_ID]

2、 请注意密钥索引号与终端支持的密钥组数不是同一种概念，二者不具有必然联系，很多应用开发者容易混淆二者。密钥组数是指一个 PINPAD 内部可以支持的密钥存储单位的个数，类似一个建筑内的房间的个数；而密钥索引号只是一个密钥的编号，相当于一个房间的房间号。房间号不一定是从 1 开始递增到房间总数为止的。一个 KAP 内，应用可以任意指定索引号（只要不超过规定的最大值 0xFFFF），因此密钥索引号是有可能大于密钥组数的。比如一个 PINPAD 最多可以支持 250 组密钥，但是密钥索引号是可以大于 256 的（比如 1000），因为密钥索引号的编号是可以跳跃的。因为密钥组数是相对整个 PINPAD 存储空间而言的，而密钥索引号只是一种密钥体系中的每种密钥的唯一编码。 就好比一个建筑，房间号可以是 3206，但是并不意味着有超过 3206 间房一样的道理。

密钥配置（Key Configuration）

数据结构 ET_KeyCfg 被用于描述密钥配置：

```
typedef struct ET_KeyCfg {

    ET_KeyUsage tKeyUsage;
    ET_KeyAlgorithm tKeyAlgorithm;
    ET_ModeOfUse tModeOfUse;
```

```
    ET_KeyVersionNumber tVerNum;
    ET_KeyExportability tExportability;

} ET_KeyCfg;
```

密钥用途（Key Usage/Purpose）

密钥用途的定义，是参考 TR-31 规范，定义如下：

```
typedef char ET_KeyUsage;

#define EM_pinpad_KU_BDK ( (ET_KeyUsage) (0x00) ) //    Corresponding to “B0” defined in TR-31

#define EM_pinpad_KU_DUKPT_INIT_KEY ( (ET_KeyUsage) (0x01) )// "B1". DUKPT init key.

#define EM_pinpad_KU_CVK ( (ET_KeyUsage) (0x02) )// "C0". CVK : Card Verification Key.

#define EM_pinpad_KU_DATA_ENCRYPTION ( (ET_KeyUsage) (0x03) )// "D0". Data encryption Key.

#define EM_pinpad_KU_EMV_IMK_APP_CRYPTOGRAMS ( (ET_KeyUsage) (0x04) )// "E0". IMK :
Issuer Master Key.

#define EM_pinpad_KU_EMV_IMK_SECURE_MSG_FOR_CONFIDENTIALITY ( (ET_KeyUsage)
(0x05) )// "E1". IMK : Issuer Master Key.

#define EM_pinpad_KU_EMV_IMK_SECURE_MSG_FOR_INTEFRITY ( (ET_KeyUsage) (0x06) )//
"E2"

#define EM_pinpad_KU_EMV_IMK_DATA_AUTHENTICATION_CODE ( (ET_KeyUsage) (0x07) )//
"E3"

#define EM_pinpad_KU_EMV_IMK_DYNAMIC_NUM ( (ET_KeyUsage) (0x08) )// "E4"

#define EM_pinpad_KU_EMV_IMK_CARD_PERSONALIZATION ( (ET_KeyUsage) (0x09) )// "E5"

#define EM_pinpad_KU_EMV_IMK_OTHER ( (ET_KeyUsage) (0x0a) )// "E6"

#define EM_pinpad_KU_IV ( (ET_KeyUsage) (0x0b) )// "I0". IV : Initializatio Vector.

#define EM_pinpad_KU_KEY_ENC_OR_WRAP ( (ET_KeyUsage) (0x0c) )// "K0". including MK.

#define EM_pinpad_KU_KBPK ( (ET_KeyUsage) (0x0d) )// "K1". KBPK : TR-31 Key Block Protection
Key.

#define EM_pinpad_KU_ISO_16609_MAC_ALGORITHM_1 ( (ET_KeyUsage) (0x0e) )// "M0"
```

#define EM_pinpad_KU_ISO_9797_1_MAC_ALGORITHM_1 ((ET_KeyUsage) (0x0f))// "M1"

#define EM_pinpad_KU_ISO_9797_1_MAC_ALGORITHM_2 ((ET_KeyUsage) (0x10))// "M2"

#define EM_pinpad_KU_ISO_9797_1_MAC_ALGORITHM_3 ((ET_KeyUsage) (0x11))// "M3". MAC Key.

#define EM_pinpad_KU_ISO_9797_1_MAC_ALGORITHM_4 ((ET_KeyUsage) (0x12))// "M4"

#define EM_pinpad_KU_ISO_9797_1_MAC_ALGORITHM_5 ((ET_KeyUsage) (0x13))// "M5"

#define EM_pinpad_KU_PIN_ENCRYPTION ((ET_KeyUsage) (0x14))// "P0". PIN Key, WK.

#define EM_pinpad_KU_KPV ((ET_KeyUsage) (0x15))// "V0". PIN verification, KPV, other algorithm.

#define EM_pinpad_KU_PIN_VERIFICATION_IBM_3624 ((ET_KeyUsage) (0x16))// "V1"

#define EM_pinpad_KU_PIN_VERIFICATION_VISA_PVV ((ET_KeyUsage) (0x17))// "V2"

如下的密钥用途是扩展出来的(未包含在 TR-31 中)：

#define EM_pinpad_KU_TRACK_DATA_ENCRYPTION ((ET_KeyUsage) (0x20)) 磁道加密
#define EM_pinpad_KU_LAKALA_TMK ((ET_KeyUsage) (0x30)) 拉卡拉专用 MK

密钥算法（Key Algorithm）

typedef char ET_KeyAlgorithm;

#define EM_pinpad_KA_AES ((ET_KeyAlgorithm)('A'))
#define EM_pinpad_KA_DEA ((ET_KeyAlgorithm)('D'))
#define EM_pinpad_KA_ELLIPTIC_CURVE ((ET_KeyAlgorithm)('E'))
#define EM_pinpad_KA_HMAC_SHA_1 ((ET_KeyAlgorithm)('H'))
#define EM_pinpad_KA_RSA ((ET_KeyAlgorithm)('R'))
#define EM_pinpad_KA_DSA ((ET_KeyAlgorithm)('S'))
#define EM_pinpad_KA_TDEA ((ET_KeyAlgorithm)('T')) // Triple DEA
#define EM_pinpad_KA_SMS4 ((ET_KeyAlgorithm)('M')) // SMS4

LANDI 终端现在只支持 DES/TDES、AES、SM4 算法，其他算法预留将来扩展。注意：海外版本不支持 SM4 算法，所有相关操作均返回失败。

密钥使用模式（Key Mode of Use）

typedef char ET_ModeOfUse

#define EM_pinpad_MOU_ENC_DEC_WRAP_UNWRAP ((ET_ModeOfUse)('B')) // Both Encrypt & Decrypt / Wrap &Unwrap. "KUM" : Key Use Mode.

#define EM_pinpad_MOU_GENERATE_AND_VERIFY ((ET_ModeOfUse)('C')) // Both Generate & Verify

#define EM_pinpad_MOU_DEC_OR_UNWRAP_ONLY ((ET_ModeOfUse)('D')) // Decrypt / Unwrap Only

#define EM_pinpad_MOU_ENC_OR_WRAP_ONLY ((ET_ModeOfUse)('E')) // Encrypt / Wrap Only

#define EM_pinpad_MOU_GENERATE_ONLY ((ET_ModeOfUse)('G')) // Generate Only

#define EM_pinpad_MOU_NO_RESTRICTIONS ((ET_ModeOfUse)('N')) // No special restrictions (other than restrictions implied by the Key Usage)

#define EM_pinpad_MOU_SIGNATURE_ONLY ((ET_ModeOfUse)('S')) // Signature Only

#define EM_pinpad_MOU_VERIFY_ONLY ((ET_ModeOfUse)('V')) // Verify Only

#define EM_pinpad_MOU_DERIVE_KEYS ((ET_ModeOfUse)('X')) // Key used to derive other key(s)

密钥管理（Key Management）

密钥注入（Key Injection）

明文密钥注入:

LANDI 提供 LKI 解决方案用于安全的注入明文初始密钥。更多信息请联系技术支持。

加密密钥注入:

提供如下 API 用于下载加密密钥：

EA_pinpad_ucLoadEncKey: Load encrypted keys.

密钥生成（Key Generation）

EA_pinpad_ucGenerateKey: 根据指定算法产生过程密钥.

注：目前大多数情况下，密钥是由收单机构生成并注入到设备中的；只有少部分客户会由 PED 内部自动生成密钥。

密钥检查（Key Check:）

EA_pinpad_ucCheckKey: 检查密钥是否可用.

KCV:

EA_pinpad_ucGetKcv: 用于计算一组密钥的 KCV，输出的 KCV 不超过 4 字节（国内）或者 3 字节（海外）;

EA_pinpad_ucAuthEncKeyKCV：装载密文密钥前检查 KCV（此接口不会对外输出 KCV，因此不受 KCV 字节限制）

密钥删除（Key Deletion:）

EA_pinpad_ucDeleteKey: 删除一组密钥

EA_pinpad_ucEraseAllKeysWithinKap: 删除某个 KAP 内所有密钥

密钥使用（Key Using）

MAC:

EA_pinpad_ucMacInit: Init MAC parameters

EA_pinpad_ucMacLoadData: Load data to be calculated with MAC algorithm

EA_pinpad_ucMacGenerate: Generate MAC using loaded data

EA_pinpad_ucMacVerify: Compare the generated MAC data with the appointed MAC data

Before using the APIs, a MAK(MAC key) must be loaded first by invoking EA_pinpad_ucLoadEncKey or injection in plaint-text (for Fixed key only).

Data Encryption:

EA_pinpad_ucCalculateDes: Do des/tdes encryption/decryption with appointed key.

Before using the API, a DATA KEY must be loaded first by invoking EA_pinpad_ucLoadEncKey or injection in plaint-text (for Fixed key only).

Account data Encryption:

EA_pinpad_ucEncryptMagTrackData: Encrypt the account data with TDK

Before using the API, a TDK(Track data encryption key) must be loaded first by invoking EA_pinpad_ucLoadEncKey or injection in plaint-text (for Fixed key only).

PIN Encryption:

EA_pinpad_ucStartPinEntry: Start PIN entry and the encrypted PIN BLOCK will returned through the call-back parameter.

Before using the API, a PEK(PIN encryption key) must be loaded first by invoking EA_pinpad_ucLoadEncKey or injection in plaint-text (for Fixed key only).

2.2.4. PIN 采集和处理

和应用交互方式：

终端提供异步 PIN 输入和处理接口，即应用在需要输入 PIN 时只需要启动 PIN 输入，之后异步监听 PIN 事件（包括 PIN 输入过程的按键、PIN 加密/校验结果上送等）.

PINPAD 支持物理按键或者触屏两种 PIN 输入方式，最终使用哪种需根据终端类型决定，通常的策略是：

1、 如果终端支持物理按键，则只能使用物理键盘输入 PIN，禁止使用触屏输入 PIN；
2、 只有纯触屏终端，才支持触屏 PIN 输入。

针对触屏 PIN 输入过程，由系统和底层程序定义 UI 界面风格，有一个专门的系统服务负责 PIN 输入过程的 UI 交互，应用程序如果不满意默认的 PIN 输入交互界面，可以向该系统服务申请定制 PIN 输入交互界面风格，具体接口请参看“PIN 输入过程 UI 交互模块”的编程指南。

无论哪种 PIN 输入过程，每次 PIN 按键事件，都会通知给应用程序，应用程序可以根据实际情况进行处理，如果是非触屏方式，应用需要负责 UI 的实现（例如 PIN 输入过程的回显提示等）。

注意事项：

1、 PIN 输入过程，禁止应用和焦点切换（系统应用除外），如果发生切换则会自动取消 PIN 输入过程；
2、 终端提供 API 供应用程序随时取消 PIN 输入过程；
3、 当 PINPAD 设备被关闭时，驱动会自动退出 PIN 输入过程；

脱机 PIN 处理

如果是脱机 PIN，终端会在 PIN 输入完成后自动送到 IC 卡中完成校验，应用程序无法接触到 PIN 明文。

应用程序需要将要做校验的 IC 卡信息、公钥（如果是脱机密文 PIN）、IC 卡 PIN 校验 APDU 类型等信息传递给 PINPAD 驱动，以便完成正确的校验。

相关 API:

EA_pinpad_ucStartPinEntry: 启动 PIN 输入
EA_pinpad_ucCancelPinEntry:取消 PIN 输入
EA_pinpad_ucInjectPinEntryFuncKey: PIN 输入过程插入一个控制键（确认、取消）
EA_pinpad_ucGetPinEntryInfo: 获取 PIN 输入和处理结果
EA_pinpad_ucStartPinEntryAndVerifyWithIcCard: 脱机 PIN 输入和校验

2.2.5. PINPAD 权限管理

2.2.6. 多应用密钥区隔离

多应用的密钥隔离，是指一个 KAP 内的密钥操作，不会影响到其他 KAP，相互之间是各自独立而不互相影响的。此套隔离机制，体现在 PED 的 API 接口上，每个密钥相关的操作，都引入了一个 KAP ID 属性来标识。

多密钥区的认证，指的是访问某一个密钥区时，需要对其访问者拥有的权限进行检查，只有具备合法权限的 APP 才允许访问。对 APP 的检查，是通过数字签名技术验证 APP 目标文件的完整和合法性，然后从中提取 APP 的权限（可访问的 KAP ID 列表）来实现的。

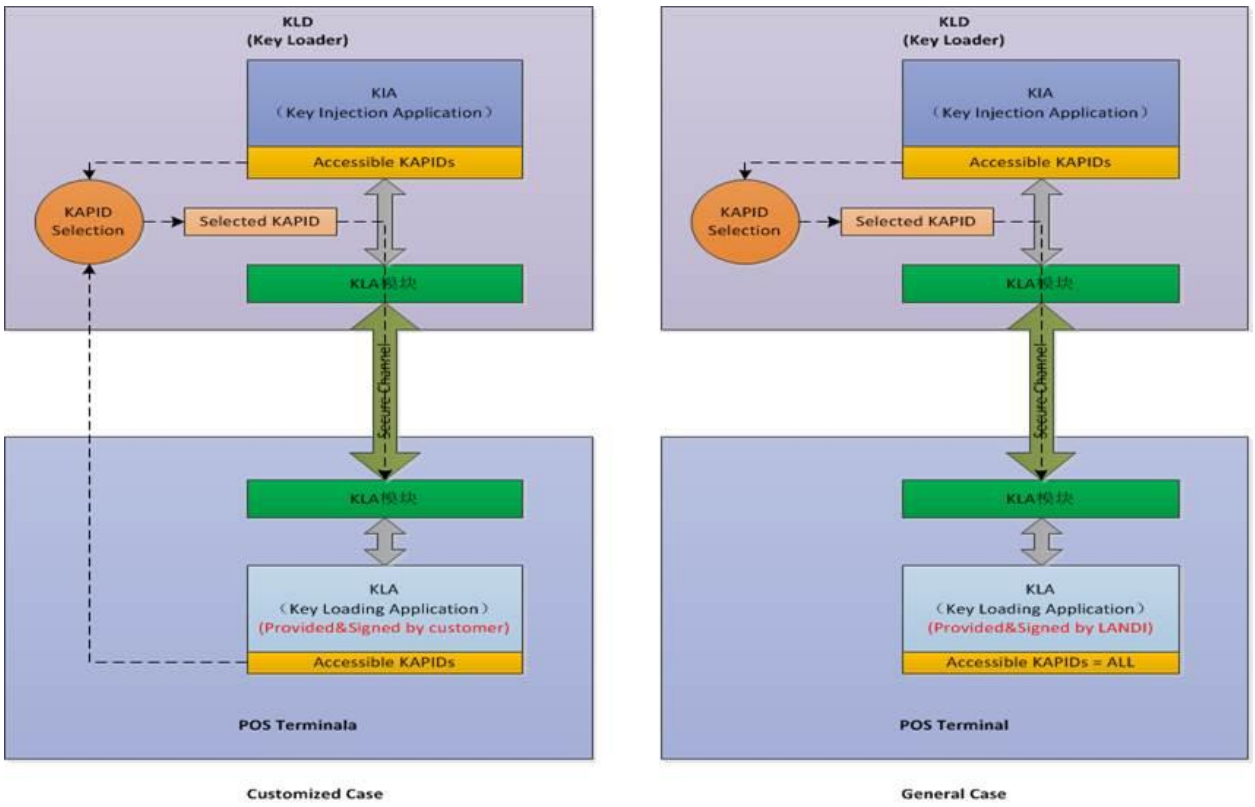
PED 提供的密钥区密钥操作接口参数中，需要指明要操作的目标 KAP ID，实际执行时会根据 APP 的权限（可访问的 KAP ID 列表）与要操作的对象（目标 KAP ID）进行比对，从而决定是否允许操作。

注意：多密钥区机制，提供的目的是为了方便应用程序更加合理、安全的使用密钥，单凭此机制，无法单独保证多个密钥区隔离和认证的安全性，其安全性同时还依赖于其他系统安全机制的使用，对这些机制的使用策略是否合理、以及其他一些管理措施是否到位等方面：

1、 应用程序需要保证采用正确的策略，来合理使用这些机制：
   如果 PED 提供了多密钥区机制，而应用程序仍然只使用一个密钥区，将所有密钥全部放在一个密钥区内，则安全风险和单密钥区机制是等效的。
2、 依赖于应用程序权限审核和授权机制的安全性
   多密钥区密钥访问时的身份认证，是需要依赖系统的数字签名技术来保证应用程序身份和权限的绑定的。如果应用程序权限审核和管理上不到位，导致应用程序获得非法权限，也会导致多密钥访问时认证的有效性。

为了实施真正的多收单机构应用隔离，应用程序应该遵循如下的安全开发指南：

1） 应用开发人员：
   a) 下载密钥的 POS 应用或者母 POS 应用，需将密钥保存于私钥的 KAP 内；
   b) 推荐使用 LKI 母 POS 来下载密钥。LKI 母 POS 支持多应用隔离机制，会将母 POS 的 APP 中允许访问的 KAP IDs 和目标 POS 应用可访问的 KAP IDs 进行比对，确保只有二者共同允许的 KAP ID 才能被访问。



2） UKEY 申请和管理人员： 如果一个管理者可以为多个收单机构申请签名 UKEY，则在申请签名 UKEY 时，需要明确制定不同过的收单机构，以便为不同机构分配不同的私有 KAP；

2.3. API 详细描述

参看 “EPT-AND 应用开发指南.chm”.

2.4. 常用功能开发指南

PINPAD 模块详细的 API 说明，请参看“EPT-AND 应用开发指南.chm”。本章节重点描述最常见的支付交易该如何编程。

2.4.1. 设备开启与关闭

用法说明：

详见“PINPAD 设备与句柄（PINPAD Device）”。

相关 API：

EA_pinpad_ucGetAllPinpadsInfo: 用于列举出检测到的所有可用的 PINPAD 设备
EA_pinpad_ucOpenDevice: 用于打开一种 PINPAD 设备并获得设备操作句柄
EA_pinpad_vCloseDevice: 用于关闭一种 PINPAD 设备并释放操作句柄
更多 API 用法，请参看《EPT-AND 应用开发指南.chm》

2.4.2. 下载明文主密钥

安全要求：

根据 PCI 以及银联的安全规范，初始明文密钥的下载，必须**同时**满足如下的几点安全要求：

1、 所有提供下载明文 KEY 功能的程序，都应该当做固件来对待，其代码需要经过严格的审核，然后通过数字签名机制保护起来防止篡改。开发该类程序的人员/团队，有责任按照合理的安全要求进行编程，否则由于未按照安全要求编程，或者故意实现恶意和非法功能，导致明文密钥信息泄露的，收单行（一般是明文 KEY 的所有者）有权追究该开发人员/团队的责任。
2、 KEY 明文只能存储于安全区，或者短暂的出现在缓冲区中（之后需要立即清空缓冲区）。因此，所有提供下载明文 KEY 的程序，都不能偷偷缓存 KEY 的明文，使用完毕后需要立即清空缓冲区。
3、 手输明文密钥的界面，需要符合信息拆分、双重控制的原则；
4、 明文密钥的下载，只能在安全环境进行；
5、 对于接收明文密钥的设备（例如 PED），下载明文主密钥的操作需要授权，如果无法授权，则需要清空该密钥体系下已经存在的所有密钥。

密钥下载过程的认证机制：

针对上述安全要求的第 5 点，要求 PED 接收明文密钥之前，需要对其来源的合法性继续认证，如果无法认证，需要删除 PED 中已经存在的合法密钥。

根据此要求，PED 提供两种安全的明文密钥下载保护机制：

（1） 下载明文密钥之前，要求先清空同一密钥区内已经存在的所有合法密钥；
（2） 下载明文密钥，使用口令方式进行授权和认证，此方式下不会影响 PED 中已经存在的密钥。
（3） 下载明文密钥，使用 LKI 解决方案，通过专用 Key Loader 进行密钥下载，下载前 PED 会对 Key Loader 合法性进行验证。

上述机制是三者选其一即可。

鉴于上述安全要求的复杂性，关于初始明文密钥下载方案，请联系安全认证团队单独讨论。联迪可以根据客户需求，提供安全便捷的密钥下载方式，包括：

1、 使用 LKI 母 POS，用于本地密钥注入；
2、 厂商预置安全的密钥下载应用，提供手输密钥功能；
3、 使用 RKIS 系统进行远程密钥注入；
4、 根据客户需求定制开发。

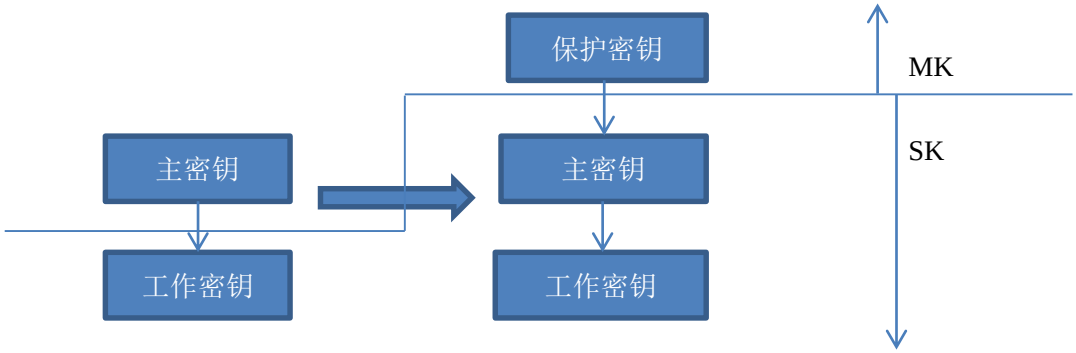
2.4.3. 下载工作密钥

用法说明：
工作密钥主要是对国内常用的 MK/SK 密钥管理体系下 SK 的一种习惯性称呼，在直连 POS 应用规范中，工作密钥是指直接用来进行 PIN 加密、MAC 计算、磁道数据加密等的密钥。

如果是 DUKPT 密钥体系中，工作密钥都是交易时自动生成，不存在“下载”一说；如果是 FK 密钥体系，就只有一级密钥结构，FK 可以看做就是工作密钥，直接用于 PIN 加密/MAC 计算/磁道加密等用途。

由于 MK/SK 密钥体系是非常灵活的，可以扩展成二层、三层甚至任意层次，因此理论上只有最根一级的密钥才是 MK，其他所有下级密钥都是 SK（过程/会话密钥）。 国内常用的“主密钥”和“工作密钥”等叫法是一种习惯性叫法，主要还是基于传统的银行收单系统的二级或者三级密钥层级。但是随着新兴市场的发展和变化，“主密钥”和“工作密钥”的叫法已经很难适用于所有场合，容易引起歧义，举例来说：

1）传统银行中，最初是“主密钥”+“工作密钥”二级结构，由于某些原因，演变成了三级密钥体系，即：在原来的主密钥上层增加了一种保护密钥，主密钥是通过加密后下载的。因此新的密钥体系中，严格说来，保护密钥才是 MK，下二级密钥都是 SK，但是国内习惯性将第二级的密钥仍然也叫做主密钥。



2）有些客户，例如拉卡拉，其工作密钥不是直接用于 PIN 加密等，而是需要先用随机数分散得到一个新的密钥，再用新的密钥进行 PIN 加密等。这种情况下，他们也叫作“工作密钥”，概念上容易混乱。

正是基于上述原因，在联迪的 PINPAD 模块提供的接口中，并不局限于“主密钥”和”工作密钥”一说，而是只有两种密钥：一种是以明文方式下载的初始密钥（即 MK ，对应于上述的三层体系中的保护密钥或二层体系中的主密钥），另一种是以密文方式下载的密钥（即 SK，可能是二层体系中的工作密钥，也可能是三层体系中的第二层的主密钥）。这种设计理论上可以满足任意 MK/SK 的变种，不必受限于传统 POS 的“主密钥”+“工作密钥”的用法。

对一种密钥的唯一标识，是通过密钥的索引号、密钥用途等信息来标识的，无论是哪种称谓，最终如何使用该密钥取决于下载密钥时为该密钥指定的用途，这些用途包括：

```
typedef char ET_KeyUsage;

#define EM_pinpad_KU_BDK ( (ET_KeyUsage) (0x00) )  //  DUKPT 中的 BDK

#define EM_pinpad_KU_DUKPT_INIT_KEY ( (ET_KeyUsage) (0x01) )//  DUKPT 的 IK

#define EM_pinpad_KU_DATA_ENCRYPTION ( (ET_KeyUsage) (0x03) )// 数据加密

#define EM_pinpad_KU_KEY_ENC_OR_WRAP ( (ET_KeyUsage) (0x0c) )// 密钥加密和打包.

#define EM_pinpad_KU_KBPK ( (ET_KeyUsage) (0x0d) )// TR-31 中的 KBPK

#define EM_pinpad_KU_ISO_9797_1_MAC_ALGORITHM_1 ( (ET_KeyUsage) (0x0f) )// MAC 密钥

#define EM_pinpad_KU_ISO_9797_1_MAC_ALGORITHM_2 ( (ET_KeyUsage) (0x10) )// MAC 密钥

#define EM_pinpad_KU_ISO_9797_1_MAC_ALGORITHM_3 ( (ET_KeyUsage) (0x11) )// MAC 密钥

#define EM_pinpad_KU_ISO_9797_1_MAC_ALGORITHM_4 ( (ET_KeyUsage) (0x12) )// MAC 密钥

#define EM_pinpad_KU_ISO_9797_1_MAC_ALGORITHM_5 ( (ET_KeyUsage) (0x13) )// MAC 密钥

#define EM_pinpad_KU_PIN_ENCRYPTION ( (ET_KeyUsage) (0x14) )// PIN 加密密钥

#define EM_pinpad_KU_TRACK_DATA_ENCRYPTION      ( (ET_KeyUsage) (0x20) ) // 磁道数据加密

#define EM_pinpad_KU_LAKALA_TMK                 ( (ET_KeyUsage) (0x30) ) // 拉卡拉 TMK(用于分散出
PEK/MAK/TDK)
```

这些密钥与通常的“主密钥”“工作密钥”之间的对应关系推荐设置如下：

密钥体系	密钥类型	对应的 KeyUsage	对应的 ModeOfKeyUse
MK/SK	主密钥传输保护密钥	EM_pinpad_KU_KEY_ENC_OR_WRAP	EM_pinpad_MOU_DEC_OR_UNWRAP_ONLY
	主密钥	EM_pinpad_KU_KEY_ENC_OR_WRAP	EM_pinpad_MOU_DEC_OR_UNWRAP_ONLY
	工作密钥	EM_pinpad_KU_PIN_ENCRYPTION	EM_pinpad_MOU_ENC_OR_WRAP_ONLY

	（PEK）		
	工作密钥（MAK）	EM_pinpad_KU_ISO_9797_1_MAC_ALGORITHM_3	EM_pinpad_MOU_GENERATE_AND_VERIFY
	工作密钥（TDK）	EM_pinpad_KU_TRACK_DATA_ENCRYPTION	EM_pinpad_MOU_ENC_OR_WRAP_ONLY
	工作密钥（DEK）	EM_pinpad_KU_DATA_ENCRYPTION	EM_pinpad_MOU_ENC_DEC_WRAP_UNWRAP/ EM_pinpad_MOU_ENC_OR_WRAP_ONLY/ EM_pinpad_MOU_DEC_OR_UNWRAP_ONLY
	拉卡拉 TMK	EM_pinpad_KU_LAKALA_TMK	EM_pinpad_MOU_DERIVE_KEYS
	拉卡拉 PEK	EM_pinpad_KU_PIN_ENCRYPTION	EM_pinpad_MOU_ENC_OR_WRAP_ONLY
	拉卡拉 MAK	EM_pinpad_KU_ISO_9797_1_MAC_ALGORITHM_3	EM_pinpad_MOU_GENERATE_AND_VERIFY
	拉卡拉 TDK	EM_pinpad_KU_TRACK_DATA_ENCRYPTION	EM_pinpad_MOU_ENC_OR_WRAP_ONLY
DUKPT	DUKPT BDK	EM_pinpad_KU_BDK	EM_pinpad_MOU_DERIVE_KEYS
	DUKPT IK/FK	EM_pinpad_KU_DUKPT_INIT_KEY	EM_pinpad_MOU_DERIVE_KEYS
	FK 衍生出来的 PEK	EM_pinpad_KU_PIN_ENCRYPTION	EM_pinpad_MOU_ENC_OR_WRAP_ONLY
	FK 衍生出来的 MAK	EM_pinpad_KU_ISO_9797_1_MAC_ALGORITHM_3	EM_pinpad_MOU_GENERATE_AND_VERIFY
	FK 衍生出来的 TDK	EM_pinpad_KU_TRACK_DATA_ENCRYPTION	EM_pinpad_MOU_ENC_OR_WRAP_ONLY
FK	PEK	EM_pinpad_KU_PIN_ENCRYPTION	EM_pinpad_MOU_ENC_OR_WRAP_ONLY
	MAK	EM_pinpad_KU_ISO_9797_1_MAC_ALGORITHM_3	EM_pinpad_MOU_GENERATE_AND_VERIFY
	TDK	EM_pinpad_KU_TRACK_DATA_ENCRYPTION	EM_pinpad_MOU_ENC_OR_WRAP_ONLY
	DEK	EM_pinpad_KU_DATA_ENCRYPTION	EM_pinpad_MOU_ENC_DEC_WRAP_UNWRAP/ EM_pinpad_MOU_ENC_OR_WRAP_ONLY/ EM_pinpad_MOU_DEC_OR_UNWRAP_ONLY

上述密钥表中，MK/SK 中的初始密钥、DUKPT 的 BDK 以及 FK 所有密钥，都需要以明文方式下载，其他密钥可以通过密文下载或者自动生成。

相关 API：

EA_pinpad_ucLoadEncKey: Load encrypted keys

更多 API 用法，请参看《EPT-AND 应用开发指南.chm》

注意事项：

1、 按照 PCI 要求，不同用途的密钥不能有相同的取值，因此编程时请注意此问题，如果不同用途的工作密钥取值相同，有可能会返回 EM_pinpad_SAME_KEY_VALUE_DETECTED

2、 EA_pinpad_ucLoadEncKey 同时可以用来下载密文的主密钥(传统叫法)。

3、 关于密钥索引号和密钥组数

a) 关于密钥索引号：

使用该接口下载密钥时，注意密钥索引号不要冲突。密钥索引号是在统一密钥体系下的所有密钥的统一编号，请注意不同的 MK 下的 SK 的 ID 不要相互冲突。例如已有 Key ID = 0 的 MK1 和 KEY ID = 1 的 SK1；则新增 KEY ID=3 的 MK2 时，如果其下级 SK2 也是 KEY ID =1，则会将原有的 SK1 覆盖掉，并不会因为 MK ID 的不同而自动区分。也就是

说，Key ID 只是密钥存储位置的一种编号，不会自动体现任何 MK 与 SK 的关联关系，应用程序需要做好 ID 号的统一分配和管控。如果一个密钥区存在多个密钥层级，尤其要注意这种问题。

b) 关于密钥组数：

PINPAD 的密钥存储空间是动态分配的，PINPAD 可支持的最大密钥组数是伴随不同产品、配置而不同的。 理论上，一个 KAP 内可以使用的密钥组数只要不超过 PINPAD 的总容量，就都是允许的；但是如果一个 KAP 使用了过多的密钥，则其他 KAP 可使用的密钥数量就变少了。可使用的密钥组数，并不等于“KAP 数目*KAP 内支持的最大密钥组数”。举例来说，如果 PINPAD 最多支持 250 组密钥，则每个 KAP 可以使用的密钥组数最多可以到 250 组，PINPAD 如果可以支持 20 个 KAP，意思是这 20 个 KAP 共用 250 组密钥，而不是每个 KAP 可以单独使用 250 组密钥。

c) 密钥组数和密钥索引号的关系：

密钥索引号和密钥组数是两种概念，二者不具有必然联系。 很多应用开发者容易混淆二者。密钥组数是指一个 PINPAD 内部可以支持的存储的组数，类似一个建筑内的房间的个数；而密钥索引号只是一个密钥的编号，相当于一个房间的房间号。房间号不一定是从 1 开始递增到房间总数为止的。一个 KAP 内，应用可以任意指定索引号（只要不超过规定的最大值 0xFFFF），因此密钥索引号是有可能大于密钥组数的。比如一个 PINPAD 最多可以支持 250 组密钥，但是密钥索引号是可以大于 256 的（比如 1000），因为密钥索引号的编号是可以跳跃的。因为密钥组数是相对整个 PINPAD 存储空间而言的，而密钥索引号只是一种密钥体系中的每种密钥的唯一编码。 就好比一个建筑，房间号可以是 3206，但是并不意味着有超过 3206 间房一样的道理。

2.4.4. 联机 PIN 输入和加密

用法说明：

底层驱动提供异步接口实现 PIN 输入和加密功能，应用启动 PIN 输入后，每当有 PIN 相关事件（PIN 按键事件或者输入结束事件）发生时，PINPAD 的接口都会以回调的方式通知给应用程序，应用程序可以根据此事件实现各种人机交互功能（例如 PIN 输入的回显提示、返回值处理等）。

整个 PIN 输入过程，所有 PIN 数字按键，都以“*”上报给应用，应用无法接触到任何明文 PIN 数值，但是可以获知有效 PIN 按键的个数等信息。

由于不同客户有不同的偏好，众口难调，PIN 输入过程的 UI 设计，需要应用程序根据客户的风格自行实现。针对支持触屏 PIN 输入的设备，底层会提供一种默认的 UI 风格，应用如果需要修改风格或者有其他定制需求，可以通过专门的接口来实现定制风格需求。

相关 API：

EA_pinpad_ucStartPinEntry: Start PIN Entry and encrypt PIN BLOCK

EA_pinpad_ucStartTouchScreenPinEntry: Start PIN Entry and encrypt PIN BLOCK for touch screen PINPAD.

更多 API 用法，请参看《EPT-AND 应用开发指南.chm》

注意事项：

1、 基于物理按键输入 PIN 的设备，应用程序需要自己实现 UI 界面；

2、 关于“取消”、“清除”、“确认”等按键的处理规则：

a) “取消”键用于删除所有已经输入的 PIN 数字，如果 PIN 输入个数为零，则直接返回；

因此如果是输入到一般，第一次按取消键是清空所有数字，第二次按取消键是返回；

部分客户如果不希望支持取消退出功能，可以设置 ET_PinEntryReactionMode 参数，来禁用“按取消键退出 PIN 输入功能”。

b) “清除”键相当于“退格/删除”键，用于删除前一个 PIN 数字，每按一次会删除一个数字，直到已输个数为零时无法删除。

c) “确定”键用于结束 PIN 输入。对于支持“无 PIN 输入”的情况（“允许的 PIN 长度列表”中必须包含 0x00 这个数值），按确认键可以直接返回。按确认键时，如果已经输入的 PIN 个数不是允许的个数（即该数值不再参数“允许的 PIN 长度列表”范围之内），则确认键被当做无效按键处理，按下后没有任何响应（等效于没有按）。

举例来说，如果“允许的 PIN 长度列表”的值为：”\x00\x06”，则表示可以不输入 PIN 直接返回，也可以按满 6 位 PIN 后返回，如果是其他长度（例如 4 个 PIN 数字）下按确认键，则没有任何反映。

如果“允许的 PIN 长度列表”的值为：”\x04\x06”，则表示允许的 PIN 长度只有 4 和 6 两种取值，不支持“不输入 PIN 直接返回”，可以按满 4 位或 6 位 PIN 后返回，如果是其他长度（例如 5 个 PIN 数字）下按确认键，则没有任何反映。

d) PIN 输入过程还支持“PIN 输入自动完成功能”：即当允许的 PIN 长度只有一个取值时，如果输入的数字长度达到这个值，则不需要按确认键就可以自动结束 PIN 输入。

部分客户如果不希望支持该功能，可以设置 ET_PinEntryReactionMode 参数，来禁用“PIN 输入自动完成功能”。

3、 启动 PIN 输入时，必须传入 **8 字节的压缩卡号（由 16 位卡号每 2 位压缩得到）**，底层 PINPAD 驱动处理时会将 PIN 数字和卡号一起进行加密，加密格式支持 ANSI X9.8\ISO9564 规定的 Format0/3/4 等格式。目前国内大部分使用的是 Format0，PINPAD 底层对卡号和 PIN 加密的过程如下：

a) 将 8 字节压缩的卡号还原成 16 字节 PAN，截取中间 12 字节（去掉最前面 3 字节和最后面 1 个字节），然后前面填充 4 字节 0x00 得到 16 字节数据，然后压缩成 8 字节的 BLOCK1

示例：

16 位卡号：1234567890123456

压缩的 8 字节卡号（BCD）：\x12\x34\x56\x78\x90\x12\x34\x56

还原的 16 字节卡号：\x01\x02\x03\x04\x05\x06\x07\x08\x09\x00\x01\x02\x03\x04\x05\x06

截取后的 PAN：\x04\x05\x06\x07\x08\x09\x00\x01\x02\x03\x04\x05

填充后的 PAN：\x00\x00\x00\x00 \x04\x05\x06\x07\x08\x09\x00\x01\x02\x03\x04\x05

压缩后：\x00\x00\x45\x67\x89\x01\x23\x45

b) 将 PIN 按照 Format0 规则进行填充得到 BLOCK2

示例：

明文 PIN：123456

填充后的 PIN：\x06\x12\x34\x56\xFF\xFF\xFF\xFF

c) 将 BLOCK1 和 BLOCK2 逐位异或得到最终的 PIN BLOCK：

PIN BLOCK = BLOCK1 ^ BLOCK2

示例：

明文 PIN：123456

16 位卡号：1234567890123456

BLOCK 1: *\x00\x00\x45\x67\x89\x01\x23\x45*

BLOCK 2: *\x06\x12\x34\x56\xFF\xFF\xFF\xFF*

PIN BLOCK: *\x06\x12\x71\x31\x76\xfe\xdc\xba*

4、 为了防止 PIN 穷举攻击，推荐 PIN BLOCK 使用 Format3 或者 Format4 等格式，采用随机填充模式，而不要使用 Format0 这种固定数值（0xFF）的填充模式。

2.4.5. 脱机 PIN 输入和校验

用法说明：

PINPAD 模块提供专用的接口实现脱机 PIN 的输入和校验功能。使用该接口，可以在 PINPAD 内部安全的完成 PIN 输入和校验，而不会将脱机 PIN 明文泄露给应用程序。当 PIN 启动输入以后，会自动缓存输入的 PIN，当输入结束时会自动使用 API 参数传递的信息将 PIN 送至 IC 卡完成校验。

使用此接口，除了输入和联机 PIN 接口类似的参数外，还需要传入脱机 PIN 校验相关的信息，包括：

1） 要校验的卡做号

2） PIN 格式（明文 PIN 还是密文 PIN）

3） 挑战随机数

4） IC 卡公钥

5） 校验 PIN 的 APDU 报文类型

相关 API：

EA_pinpad_ucStartPinEntryAndVerifyWithIcCard：脱机 PIN 输入和校验（物理键盘 PIN 输入）

EA_pinpad_ucStartTouchScreenPinEntryAndVerifyWithIcCard:脱机 PIN 输入和校验(触屏 PIN 输入）

EA_pinpad_ucStartOfflinePin:启动脱机 PIN 输入（PIN 输入完毕后经过加密后缓存区安全区域，需要在规定时间(1 分钟)内调用 EA_pinpad_ucVerifyOfflinePin 将缓存的 PIN 送到 IC 中完成校验）

EA_pinpad_ucVerifyOfflinePin：校验脱机 PIN

更多 API 用法，请参看《EPT-AND 应用开发指南.chm》

注意事项：

1、 请应用程序务必使用此类接口完成脱机 PIN 校验，而不要自己实现 PIN 输入功能。应用自行实现的各种脱机 PIN 校验功能，不一定能够满足 PCI 等安全规范的要求，因此不受联迪的安全保护，出了任何安全事故联迪也不承担任何风险责任。

2、 关于脱机 PIN 的使用，如果应用架构上是直接使用最新的 EMV 轻内核提供的接口，则 EMV 轻内核模块已经完成脱机 PIN 所有处理流程，应用程序可不必关心上述细节实现过程。

2.4.6. 磁道数据加密

用法说明：

PINPAD 模块提供专用的接口实现磁道数据加密。应用程序可以调用此接口实现对账号信息的加密并送至收端机构后台。

相关 API：

EA_pinpad_ucEncryptMagTrackData: Encrypt the account data with TDK

更多 API 用法，请参看《EPT-AND 应用开发指南.chm》

注意事项：

1、 按照 PCI 等安全规范的要求，磁道密钥不允许和 PIN 加密密钥有相同的取值，应用编程时请注意此点，如果出现找不到密钥时，请注意检查是否因为此原因之前密钥根本就未下载成功。

2、 针对账户数据加密，如果使用的是 MK/SK 或者 FK 密钥体系，为了保证安全，请确保使用不低于 24 字节的 TDES 密钥，否则不符合 PCI 等安全规范要求。如果使用的是 DUKPT 密钥体系中，由于该体系是每笔交易一个密钥，因此不受到此限制。

2.4.7. 计算 MAC

用法说明：

PINPAD 模块提供专用的接口实现 MAC 计算功能，支持 ISO9797 MAC 算法 1（基于单 DES 密钥，即 ANSI X9.9）和 ISO9797 MAC 算法 3（基于 TDES 密钥，即 ANSI X9.19）。

PINPAD 模块支持批量数据分段计算 MAC 的功能（填充模式 2 无法支持），用户可以先通过 EA_pinpad_ucMacInit 初始化算法，然后调用 EA_pinpad_ucMacLoadData 分组多次载入数据，最后调用 EA_pinpad_ucMacGenerate 或者 EA_pinpad_ucMacVerify 实现最终的 MAC 输出或者校验。

相关 API：

EA_pinpad_ucMacInit: Init MAC parameters

EA_pinpad_ucMacLoadData: Load data to be calculated with MAC algorithm

EA_pinpad_ucMacGenerate: Generate MAC using loaded data

EA_pinpad_ucMacVerify: Compare the generated MAC data with the appointed MAC data

更多 API 用法，请参看《EPT-AND 应用开发指南.chm》

注意事项：

1、 按照 PCI 等安全规范的要求，MAC 密钥不允许和 PIN 加密密钥有相同的取值，应用编程时请注意此点，如果出现找不到密钥时，请注意检查是否因为此原因之前密钥根本就未下载成功。

2、 下载 MAC 密钥时，可以设置 MAC 密钥的用途为仅生成 MAC、仅校验或二者皆可。使用相关 API 时，也会检查对应的 MAC 密钥用途是否匹配，例如 EA_pinpad_ucMacGenerate 时，MAC 密钥用途不能是仅校验，同样使用 EA_pinpad_ucMacVerify 时，MAC 密钥用途也不能是仅生成。

2.4.8. 报文加解密

用法说明：

PINPAD 模块提供专用的接口实现普通的数据 DES 加解密运算功能，可以通过此接口实现报文加解密功能。

使用此接口，需要先下载一组 DES/TDES 密钥，然后使用时和算法库的 DES/TDES 用法类似，可

以支持加密、解密、TECB、TCBC 等功能。

相关 API：

EA_pinpad_ucCalculateDes: Do des/tdes encryption/decryption with appointed key.

更多 API 用法，请参看《EPT-AND 应用开发指南.chm》

注意事项：

1、 按照 PCI 等安全规范的要求，普通数据加解密密钥不允许和 PIN 加密密钥有相同的取值，应用编程时请注意此点，如果出现找不到密钥时，请注意检查是否因为此原因之前密钥根本就未下载成功。

2.5. 安全开发指南

注意：如下的安全开发指南非常重要，请应用程序开发者务必认真阅读。如下的安全开发指南，致力于指导和协助应用开发者提供安全合理使用 POS 设备的方式，应用开发者有责任遵循本指南合理开发应用程序，以保证收单机构和持卡人的合法利益。

如因未遵循如下安全要求并私自使用其他方式实现支付交易功能的，相关安全责任由应用开发机构自行承担，应用程序签名审核机构务必确保应用程序编程的正确性，否则因此给收单机构或者持卡人造成利益损失的，POS 终端厂商不承担任何安全责任。联迪仅针对遵循下述安全开发要求的行为提供安全承诺和保证，任何其他不遵循下述安全要求和擅自使用其他方式实现的交易行为，均属于未授权行为，不受联迪保护。

1、 请确保使用正确的 API 来完成相关加密功能，如果未正确使用相关 API，可能导致该应用不符合 PCI/银联等安全规范的相关要求。包括：

a) 联机 PIN 输入和加密：
应用开发者务必使用 EA_pinpad_ucStartPinEntry()来启动联机 PIN 输入和加密功能；不得使用其他方式实现 PIN 输入和加密（例如软加密方式）。

b) 脱机 PIN 输入和验证
应用开发者务必使用 EA_pinpad_ucStartPinEntryAndVerifyWithIcCard 或者 EA_pinpad_ucStartOfflinePin+EA_pinpad_ucVerifyOfflinePin 来启动脱机 PIN 输入和加校验功能；不得使用其他方式实现 PIN 输入和验证（例如自行通过触屏输入 PIN，或者使用自己封装的 PIN 输入界面，均不符合要求）

c) 账户数据加密
- 应用开发者务必对持卡人账户数据进行加密，如因收单机构后台不支持账户数据加密的，收单机构需充分评估由此带来的风险并保证持卡人的合法权益，联迪不为此承担相关风险。应用开发者作为收单机构的委托开发者，承担着维护收单机构利益和保证持卡人数据安全的责任，针对收单机构不支持持卡人账户数据加密的情况，有义务告知收单机构此做法存在的风险。
- 应用开发者务必使用 EA_pinpad_ucEncryptMagTrackData 实现持卡人账户数据加密，不得使用除此以外的其他未授权方式加密（例如使用软加密，或者自行使用其他方式实现）。

2、 终端上的应用软件应尽量少地保留用户敏感数据

限制数据存储量和保留时间，以达到恰好能满足业务、法律,行业规定和管理规定需要的程度。事实上，如无必要，终端上就不应该保留用户敏感数据，如银行卡磁道信息、卡号、银行卡密码、CVN2等（即使经过加密）。

3、 不得存储敏感数据。

如果整个交易系统没有明确要求，终端上的应用程序在身份认证结束后就不应该存储敏感数据，如银行卡磁道信息、卡号、银行卡密码、CVN2 等（即使经过加密）。

4、 应用程序中应仅含有需要的功能操作，没有有意或者无意地隐藏后门。后门包括但不限于下列情形：

d) 未经收单机构和终端产品使用者同意，软件自动联网，与未经收单机构或者使用者确认的网络端口通信，传递终端产品上的任何信息；
e) 未经收单机构和终端产品使用者同意，擅自开启监听端口；
f) 未经收单机构和终端产品使用者同意，利用 ICMP 协议当作后门进行非法通信，传递 POS 上的任何信息；
g) 未经收单机构和终端产品使用者同意，利用类似 libpcap 的工具构造私有的网络数据报文进行非法通信；
h) 未经收单机构和终端产品使用者同意，植入网页后门，线程后门等，传递 POS 上的任何信息；
i) 未经收单机构和终端产品使用者同意，转储任何输入事件或 framebuffer；
j) 未经收单机构和终端产品使用者同意，监听系统的各种输入输出设备；
k) 未经收单机构和终端产品使用者同意，修改（包括提升和降低）终端产品中任何软件的访问权限；
l) 未经收单机构和终端产品使用者同意，隐含了最高访问权限的线程；
m) 未经收单机构和终端产品使用者同意，未调用乙方终端 SDK 提供的安全 PIN 输入函数，而是另外模拟 PIN 输入的界面和操作。这种模拟 PIN 输入的界面和操作可能会误导用户输入 PIN，从而造成客户 PIN 的泄露。

5、 终端界面上一旦进入到输入 PIN 的状态下，就认为之前的交易内容已经确定，程序上不能再提供修改交易内容（包括金额，商品内容等）。

6、 应用程序重写库函数（如何适用）

由于 memset、memcmp 函数在编译时有可能被编译器优化，留下安全隐患，安全建议：

a) 在编译底层代码时，需加上编译以下选项：
LOCAL_CFLAGS += **-fno-dse**
b) 必须对这些函数进行重新实现。实现代码如下：

```
uchar EI_ucMemcmp(void *pvInput1, void *pvInput2, uint uiLen)
{
    uchar *   pucInput1 = (uchar*)pvInput1;
    uchar *   pucInput2 = (uchar*)pvInput2;
    uchar   ucRet;
    volatile uint   i;

    ucRet = EM_SUCCESS;
    for(i=0;i<uiLen;i++)
```

```
    {
        if(pucInput1[i] != pucInput2[i])
        {
            ucRet = EM_ERROR;
        }
    }
    return ucRet;
}

void EA_vMemset(void *s, uchar ucC, uint uiLen)
{
    uchar *   pucInput1 = (uchar*)s;
    volatile uint   i;
    for(i=0;i<uiLen;i++)
    {
        pucInput1[i] = ucC;
    }
    pucInput1 = NULL;
    ucC = 0;
}
```

7、 敏感信息缓存清除

敏感信息包括，明文或明文的磁道数据、账户信息、PINBLOCK、密钥、口令等信息，在敏感操作完成后，必须将敏感变量以及敏感数据的密文的缓存数据清 0，包括全局变量，以及函数中的局部变量。

8、 账户数据使用

经过认证的应用允许获取账户数据，但是使用账户数据必须遵从以下要求:

a) 账户数据不可明文显示或输出。
b) PAN 显示或输出，必须经过截断处理，最多显示前 6 位和后 4 位，其余位用“*”替代。
c) 在使用用户数据时，必须进行加密处理。
d) 在交易完成时，必须将缓存数据清 0。
e) 在交易超时发生时，必须将缓存数据清 0，建议超时时间小于 2 分钟。

注：缓存数据包括：明文或密文的卡数据。

		Data Element	Storage Permitted	Render Stored Account Data Unreadable per Requirement 3.4
Account Data	Cardholder Data	Primary Account Number (PAN)	Yes	Yes
		Cardholder Name	Yes	No
		Service Code	Yes	No
		Expiration Date	Yes	No
	Sensitive Authentication Data[1]	Full Magnetic Stripe Data[2]	No	Cannot store per Requirement 3.2
		CAV2/CVC2/CVV2/CID	No	Cannot store per Requirement 3.2
		PIN/PIN Block	No	Cannot store per Requirement 3.2

9、 PIN 输入与非 PIN 输入界面分离

PIN 输入和非 PIN 输入必须不能在同一个界面，避免在同一个界面，用户将 PIN 输入到非 PIN 输入框内，导致敏感信息泄露。

10、 密钥注入

密钥注入过程，需遵循如下《PED 设备初始密钥装载指南.pdf》的要求。

PED设备初始密钥装载指南.pdf