Advanced Card Systems Ltd.
Card & Reader Technologies

ACR3x
移动读卡器

参考手册 V1.04

info@acs.com.hk
www.acs.com.hk

版本历史

发布日期	修订说明	版本号
2014-06-16	• 初始发布	1.00
2014-09-26	• 更新 2.0 节 – 特性 • 增加 9.0 节 – 非接触卡片命令	1.01
2015-02-25	• 更新 10.0 节 – 敏感数据导入方法	1.02
2015-07-27	• 更新 2.0 节 – 特性	1.03
2015-05-26	• 更新 2.0 节 – 特性 • 更新 9.2.1 节 – 加载认证密钥	1.04

目录

图目录

表目录

1.0. 介绍

ACR3x 移动读卡器是移动设备与磁条卡/接触式卡/非接触式卡之间的通信接口。不同类型的卡片采用不同的命令和通信协议，ACR3x 移动读卡器则在移动设备与卡片之间搭建起一个统一的接口。

ACR3x 通过 3.5mm 音频接口与移动设备相连接。这样就可以通过读卡器上的解码器读取卡片信息，再将其发送给智能手机或平板电脑等移动设备。另外为了增强安全性，卡片信息在发送至后台服务器之前，还会通过 AES-128 加密算法进行加密。

本文档介绍了 ACR3x 的软硬件设计情况，同时列出了 ACR3x 与移动设备通信时用到的命令。

1.1. 术语定义

缩写	说明
ACS 密钥	用来执行主复位认证的密钥。此密钥被硬编码在固件中，不能通过命令报文进行修改。另外它只能由 ACS 妥善保存。
AES	高级加密标准
AES 密钥	用于通过 AES-128 CBC 模式加密磁条卡磁道数据的密钥，不能由客户进行修改。
客户 ID	由客户设定的 10 字节标识码，可以由客户进行修改。
客户主密钥	此密钥由客户保存，用于在导入 AES 密钥、新的客户主密钥、客户 ID 和 DUKPT 选项，以及执行 DUKPT 初始化之前与 ACR3x 进行认证。此密钥可由客户进行修改。
设备 ID	ACR3x 的 MCU 的唯一标识码（8 字节）。客户可以使用此 ID 生成客户 ID 或 DUKPT 初始化数据。此 ID 由 MCU 生产商硬编码在 MCU 中，任何情况下都不能修改。
主复位	相当于恢复出厂设置。执行主复位后，存储在闪存中的所有数据都会被擦除，恢复为默认值。
主复位过程密钥	为了执行主复位操作而进行相互认证后生成的唯一密钥。
过程密钥	为了导入敏感数据而进行相互认证后生成的唯一密钥。
TDES	三重数据加密标准

表1 ：术语定义

2.0. 特性

2.1. ACR31

- 3.5 mm 音频插孔接口
- 电源：
  - CR2016 电池供电
- 磁条卡读卡器：
  - 可读取两个磁道的卡片数据
  - 支持双向读取
  - 支持 AES-128 加密算法
  - 支持 DUKPT 密钥管理系统
  - 支持 ISO 7810/7811 磁条卡
  - 支持高矫顽力和低矫顽力磁条卡
  - 支持 JIS1 和 JIS2
- 支持 Android™ 2.3 及以上版本 [1]
- 支持 iOS 5.0 及以上版本 [2]
- 符合下列标准：
  - CE
  - FCC
  - VCCI
  - RoHS 2
  - REACH

[1] *使用 ACS 定义的 Android 库*

[2] *使用 ACS 定义的 iOS 库*

注： *关于支持的设备列表，请访问 www.acs.com.hk。*

2.2. ACR32

- 3.5 mm 音频插孔接口
- USB 全速接口
- 电源：
  - 锂离子电池供电（可通过 Micro-USB 端口充电）
  - USB 供电（连 PC 模式）
- 即插即用 – 支持 CCID 标准，具有高度的灵活性
- 智能卡读写器：
  - 非接触接口：
    - 支持 ISO 7816 A 类、B 类和 C 类（5 V、3 V、1.8 V）卡
    - 支持符合 T=0 和 T=1 协议的微处理器卡
    - 支持各类存储卡
    - 支持 PPS（协议和参数选择）
    - 具有短路保护功能
- 磁条卡读卡器：
  - 可读取两个磁道的卡片数据
  - 支持双向读取
  - 支持 AES-128 加密算法
  - 支持 DUKPT 密钥管理系统
  - 支持 ISO 7810/7811 磁条卡
  - 支持高矫顽力和低矫顽力磁条卡
  - 支持 JIS1 和 JIS2
- 应用程序编程接口：
  - 支持 PC/SC
  - 支持 CT-API（通过 PC/SC 上一层的封装）
- 支持 Android™ 2.0 及以上版本 [3]
- 支持 iOS 5.0 及以上版本 [4]
- 符合下列标准：
  - EN 60950/IEC 60950
  - ISO 7816
  - ISO 7811
  - EMV™ Level 1 (接触式)
  - PC/SC
  - CCID
  - CE
  - FCC
  - RoHS 2
  - REACH
  - VCCI (日本)
  - KC (韩国)
  - Microsoft® WHQL

---

[3] *使用 ACS 定义的 Android 库；不适用 PC/SC 和 CCID 支持*
[4] *使用 ACS 定义的 iOS 库；不适用 PC/SC 和 CCID 支持*
注： *关于支持的设备列表，请访问 www.acs.com.hk。*

2.3. ACR35

- 3.5 mm 音频插孔接口
- 电源：
  - 锂离子电池供电（可通过 Micro-USB 端口充电）
- 智能卡读写器：
  - 非接触接口：
    - 内置天线用于读写非接触式标签，读取智能卡的距离可达 30 mm（视标签的类型而定）
    - 支持 ISO 14443 第 4 部分 A 类和 B 类卡、MIFARE®卡、FeliCa 卡和全部 4 种 NFC（ISO/IEC 18092）标签 [5]
    - 内建防冲突特性（任何时候都只能访问 1 张标签）
    - NFC 支持：
      - 卡片读/写模式
- 磁条卡读卡器：
  - 可读取两个磁道的卡片数据
  - 支持双向读取
  - 支持 AES-128 加密算法
  - 支持 DUKPT 密钥管理系统
  - 支持 ISO 7810/7811 磁条卡
  - 支持高矫顽力和低矫顽力磁条卡
  - 支持 JIS1 和 JIS2
- 支持 Android™ 2.0 及以上版本 [6]
- 支持 iOS 5.0 及以上版本 [7]
- 符合下列标准：
  - EN 6095/IEC 60950
  - ISO 14443
  - ISO 18092
  - ISO 7811
  - CE
  - FCC
  - RoHS 2
  - REACH
  - VCCI (日本)

---

[5] *不包括 Topaz。如需了解更多信息，请联系 ACS。*
[6] *使用 ACS 定义的 Android 库*
[7] *使用 ACS 定义的 iOS 库*
注： *关于支持的设备列表，请访问 www.acs.com.hk。*

3.0. 支持的卡片

3.1. 磁条卡

ACR3x 支持 ISO 7810/7811 高矫顽力和低矫顽力磁条卡。

3.2. 接触式智能卡

3.2.1. MCU 卡

ACR32 是一款符合 PC/SC 标准的智能卡读写器。它支持 ISO 7816 A 类、B 类和 C 类（5 V、3 V 和 1.8 V）智能卡。还可以读写所有符合 T=0 或 T=1 协议的 MCU 卡。

若卡片产生的 ATR 指定了专用的操作模式（TA2 存在；TA2 中的 b5 位必须为 0），但 ACR32 不支持该特定模式，则 ACR32 会将卡片置为协商模式。如果卡片不能被置为协商模式，ACR32 会拒绝读写该卡。

若卡片产生的 ATR 指定了协商模式（TA2 不存在时）和通信参数，而不是默认参数，则 ACR32 读卡器将执行 PPS 并尝试使用卡片在 ATR 中指定的通信参数。如果卡片不接受 PPS，读卡器会使用默认参数（F=372，D=1）。

对于上述参数的含义，请参照 ISO 7816-3 标准。

3.2.2. 存储卡

ACR32 支持多种类型的存储卡，例如：

- 符合 I2C 总线协议（空白存储卡）、且每页最大容量为 128 字节的存储卡，包括：
  - Atmel®：AT24C01/02/04/08/16/32/64/128/256/512/1024
  - SGS-Thomson：ST14C02C、ST14C04C
  - Gemplus：GFM1K、GFM2K、GFM4K、GFM8K
- 具有安全记忆体 IC 以及密码和认证功能的存储卡，包括：
  - Atmel®：AT88SC153 和 AT88SC1608
- 具有 1 千字节 EEPROM 智能存储空间以及写保护功能的存储卡，包括：
  - Infineon®：SLE4418、SLE4428、SLE5518 和 SLE5528
- 具有 256 字节 EEPROM 智能存储空间以及写保护功能的存储卡，包括：
  - Infineon®：SLE4432、SLE4442、SLE5532 和 SLE5542
- ‘104’型 EEPROM 不可重置标记计数卡，包括：
  - Infineon®：SLE4406、SLE4436、SLE5536 和 SLE6636
- 具有 416 位 EEPROM 智能存储空间以及内部 PIN 检查功能的存储卡，包括：
  - Infineon®：SLE4404
- 包含应用区域的逻辑加密卡，包括：
  - Atmel®：AT88SC101、AT88SC102 和 AT88SC1003

3.3. 非接触智能卡

ACR35 支持多种非接触式卡和标签，例如：

- ISO 14443 A 类卡
- ISO 14443 B 类卡
- ISO/IEC 18092（NFC）卡
- MIFARE® Classic 1K/4K 卡
- FeliCa
- MIFARE Ultralight®
- MIFARE Ultralight® C
- MIFARE® DESFire® EV1

4.0. 系统框图

4.1. ACR31

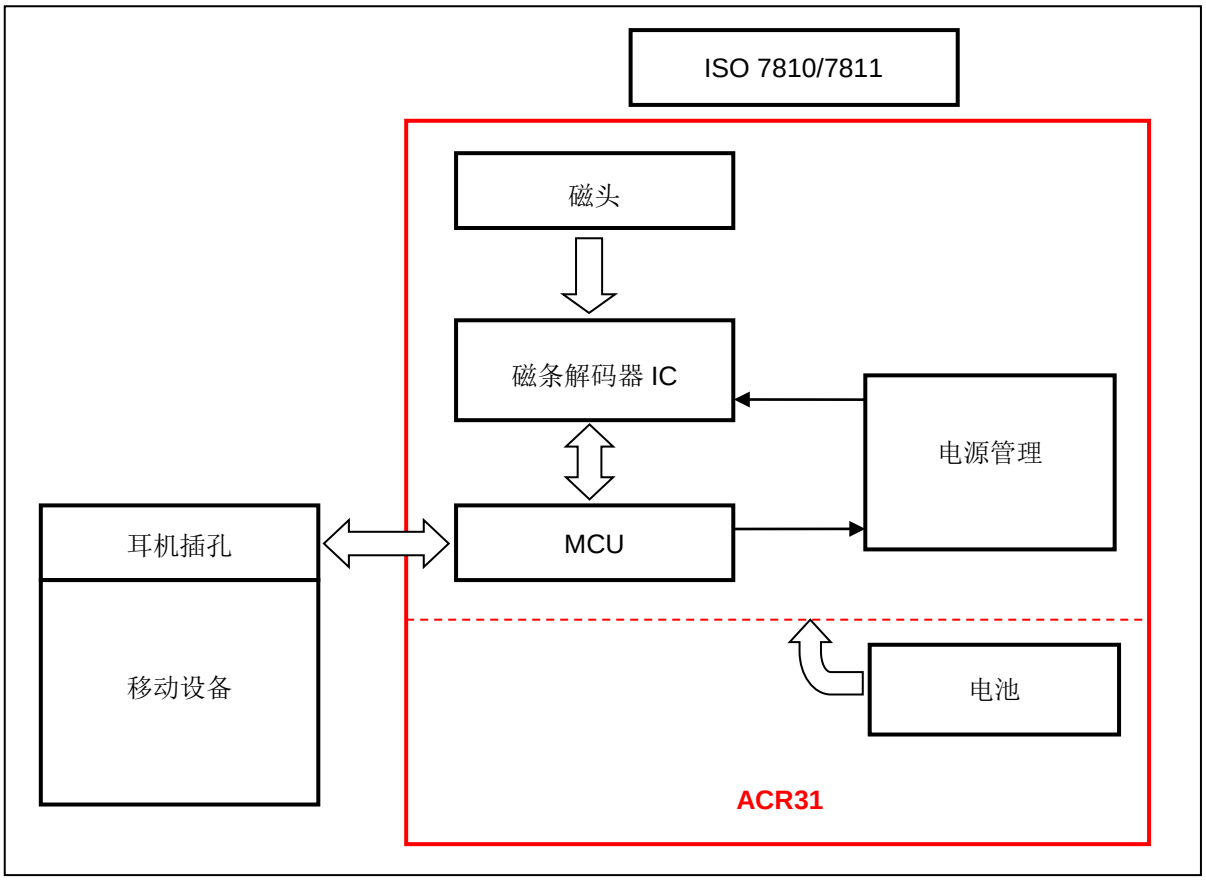

图1 ： ACR31 架构图

4.2. ACR32

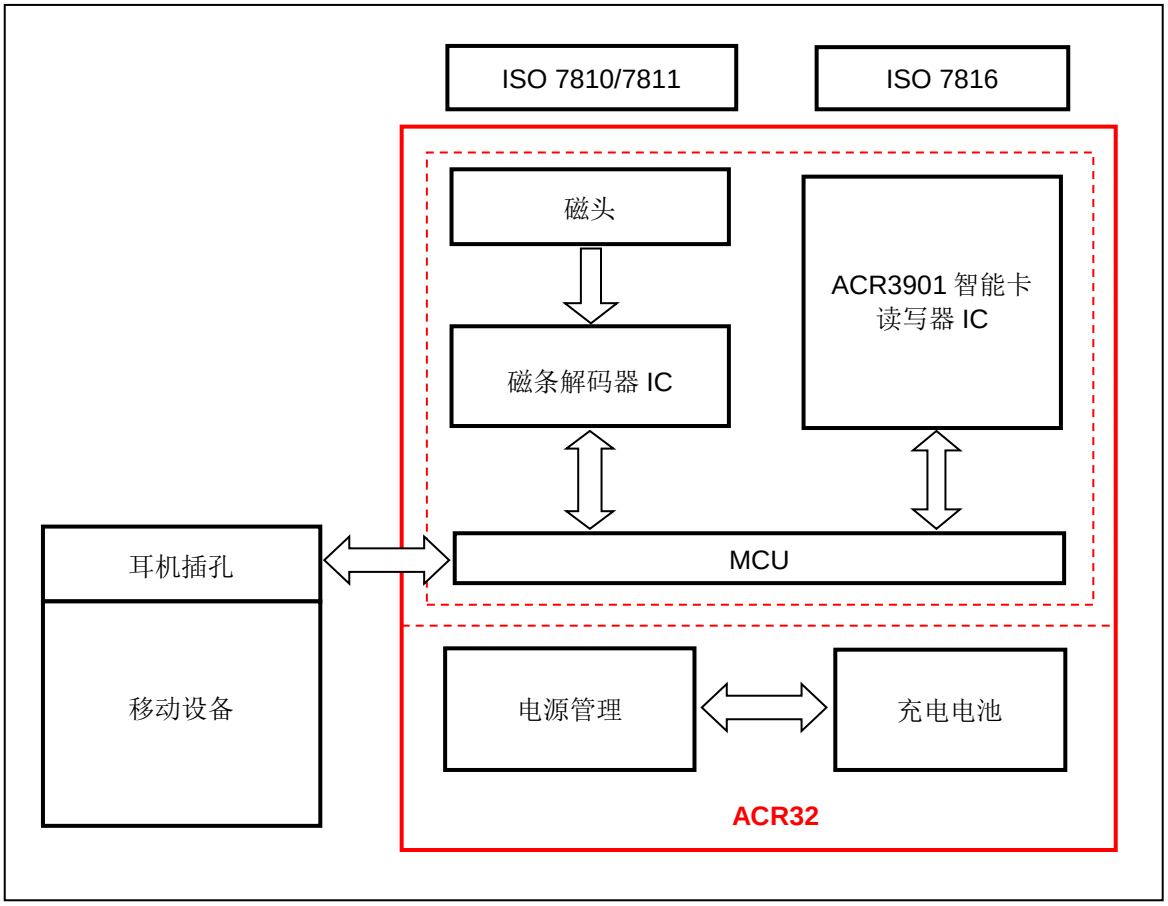

图2 ： ACR32 架构图

4.3. ACR35

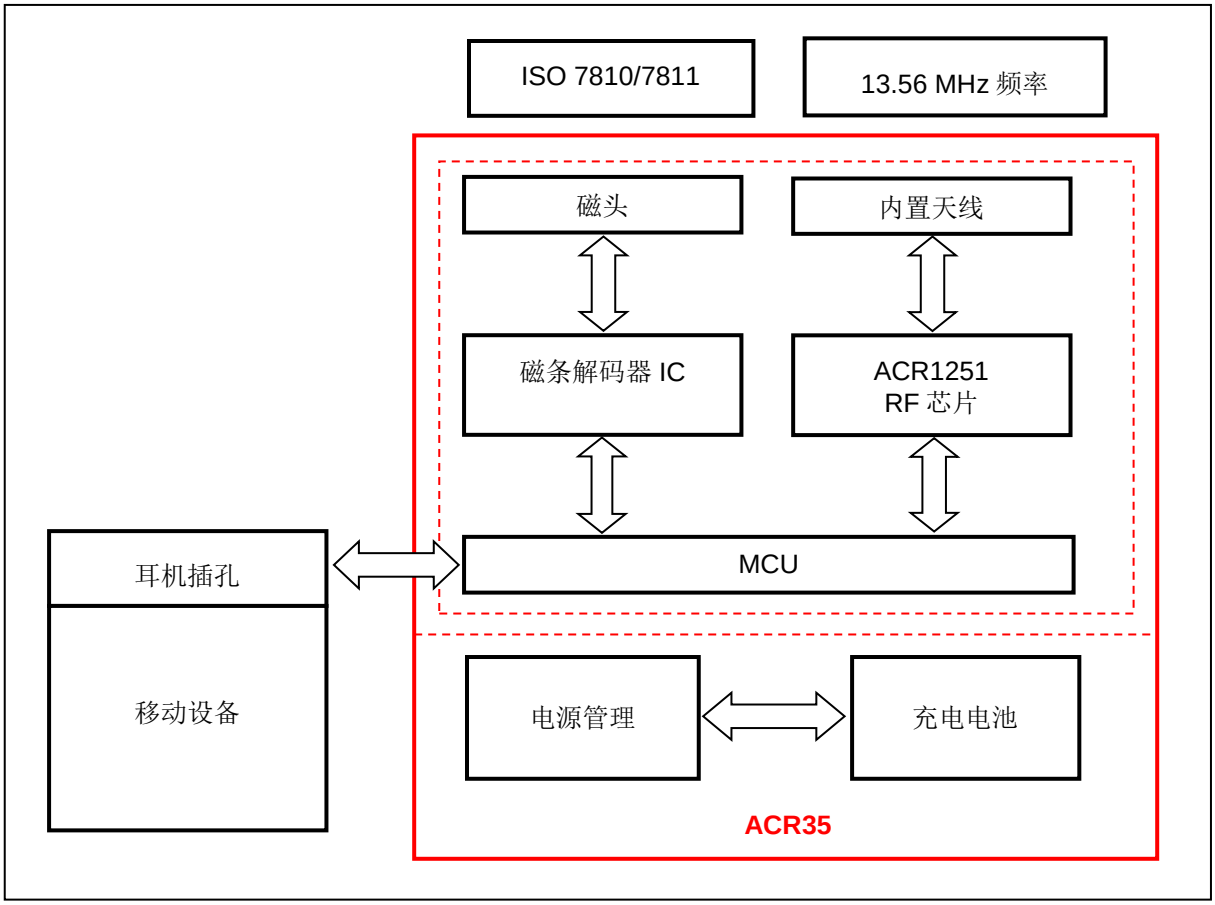

图3 ： ACR35 架构图

5.0. 硬件设计

5.1. 电池

ACR31 使用容量为 90 毫安的 CR2016 电池供电。而 ACR32 和 ACR35 则使用锂离子充电电池，电池容量为 200 毫安时。

5.2. LED 状态指示灯

不同颜色的 LED 代表了 ACR32 和 ACR35 的不同状态，其中：

- **绿色 LED** – 工作中
- **红色 LED** – 电池状态

5.3. Micro USB 接口

ACR32 和 ACR35 可通过 Micro USB 作为电池充电端口与电脑连接。另外利用此端口，ACR32 还可以用作连 PC 读写器。

5.4. 音频通道

5.4.1. 通信参数

ACR3x 通过音频通道与移动设备建立连接。

引脚	信号	功能
1	左	向 ACR3x 传输数据
2	右	唤醒设备信号
3	GND	GND
4	MIC	向智能手机传输数据

表2 ：3.5mm 音频插孔配线

5.5. 磁条卡接口

ACR3x 可以读写所有符合 ISO 7810/7811 标准的磁条卡。ISO 7810 标准规定了各类卡的物理特性，而 ISO 7811 标准则规定了各类识别卡的记录技术。

高矫顽力（Hi-Co）磁条卡通常为黑色磁条，大多以 2750 奥斯特的较高磁场强度进行编码。因此暴露于外部磁场时，编码于磁条中的数据抗意外擦除能力越强，卡片更加耐用。在磁头上刷卡时，高矫顽力磁条卡可以产生更大的信号脉冲，更容易检测和解码。

低矫顽力（Lo-Co）磁条卡一般为棕色磁条，大多以 300 奥斯特的较低磁场强度进行编码。从磁头刷过时，产生的信号脉冲比高矫顽力磁条卡小。因此信噪比（S/N）相对较低，易受噪音干扰。需要更复杂的硬件支持以及信号处理算法才能正确解码信号。

由于高磁卡和低磁卡的磁场不同，在设计时可采用具有自动增益控制的磁条解码 IC 来满足这两种卡片的需求。

5.6. 接触式智能卡接口

ACR32 与插入的智能卡之间的接口符合 ISO 7816-3 标准协议，并进行了某些限制或提升来增强 ACR32 的实用功能。

5.6.1. 智能卡电源 VCC（C1）

插入的智能卡的电流消耗不得大于 50 mA。

5.6.2. 编程电压 VPP（C6）

根据 ISO7816-3 的规定，由智能卡上的触点 C6（VPP）为智能卡提供编程电压。但由于市面上的智能卡大多数基于 EEPROM，不需要为其提供外部编程电压，ACR32 的触点 C6（VPP）已被实现为普通的控制信号。此触点的电气规格与 RST 信号（触点 C2）的规格相同

5.6.3. 卡片类型选择

每次激活插入的卡片前，处于控制地位的电脑都要向 ACR32 发送适当的命令来选择卡片类型。这些卡片包括存储卡和基于 MCU 的卡。

对于基于 MCU 的卡片来说，读写器允许从 T=0 或 T=1 中选择首选的协议。但是只有当插入读写器的卡片对这两种协议类型都支持时，读写器才可以通过协议与参数选择（PPS）接受并执行这样的选择。当基于 MCU 的卡仅支持一种协议类型（T=0 或 T=1）时，读写器会自动采用该协议类型，而不管应用程序选择哪一种。

5.6.4. 微控制器卡接口

基于微控制器的智能卡只使用触点 C1（VCC）、C2（RST)、C3（CLK）、C5（GND）和 C7（I/O）。时钟信号（C3）的频率为 4 MHz。

5.6.5. 卡片插拔保护

ACR32 提供了一种机制来保护在上电状态下被突然拔出的卡片。卡片移出时，卡片的电源以及 ACR32 与卡之间的信号线路会立即取消激活。但是作为惯例，只应在断电后才从读卡器移出卡片，这样可以避免电气损伤。

注：*ACR32 本身不会接通向卡片的供电。必须由处于控制地位的计算机通过发送给读写器的命令来执行.*

6.0. 通信协议

ACR3x 做为从属设备，几乎所有的操作都由移动设备发起。移动设备发送的命令以连续的命令请求-响应交换的形式执行。另外，新的请求报文应当在收到前一个请求的响应报文后再发送。

ACR3x 通过音频插孔接口与移动设备进行通信。信道是双方向的，读卡器通过音频插口的 MIC 脚向移动设备发送数据，而移动设备则通过右声道向读卡器发送命令。

不工作时，ACR3x 会进入深度休眠模式。从音频插孔的左声道收到移动设备的唤醒信号后，ACR3x 从休眠模式恢复并向移动设备返回一个确认信号。之后 ACR3x 会在超时时限内等待刷磁条卡。从刷过的磁条卡成功获得数据后，ACR3x 会对收到的卡片数据进行 AES-128 加密，然后把加密过的数据发送给移动设备。若读卡器在超时时间内未获得刷卡数据或命令消息，则会向移动设备发送相应的状态消息。之后 ACR3x 会返回深度休眠模式以节约电量。

ACR3x 与移动设备之间的通信协议采用直接信号馈入之前，从 ACR3x 收到的信号会通过一个 DC 偏移消除滤波器。待传输的数据采用曼彻斯特编码方案（符合 IEEE 802.3）进行编码，采用的时钟频率设为 10 kHZ。在曼彻斯特编码方案中，数据传输速度总是与时钟速度一致，可以实现的最大波特率约为 10 Kbps。

移动设备和 ACR3x 上的信号解释均基于对相应的输入波形的采样。采样频率至少应为曼彻斯特编码方案使用的时钟频率(尼奎斯特速率)的两倍。信号采样完成后，通过确定逻辑零点交叉时间可以接收到编码在信号中的数据。

7.0. 应用程序编程接口

请参考 ACR3x Android 库或 ACR3x iOS 库中的 HTML 文件。

您可以在 ACS 网站下载这些库。

8.0. 接触式卡命令

本节将对 ACR32 的存储卡命令进行介绍。

8.1. 存储卡 – 1、2、4、8 和 16 kilobit I2C 卡

8.1.1. SELECT_CARD_TYPE

此命令用于对选定的插入读写器的卡片进行上电/下电，同时进行卡片复位操作。

***注：** 只有使用 SCardConnect( ) API 建立逻辑智能卡读写器通信之后才可以使用此命令。对于 SCardConnect( ) API 的详细说明参见 PC/SC 规范。*

命令格式

Pseudo-APDU					
CLA	**INS**	**P1**	**P2**	**Lc**	**Card Type**
FFh	A4h	00h	00h	01h	01h

响应数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

8.1.2. SELECT_PAGE_SIZE

此命令会选择用于读取智能卡的页面大小。默认值是 8 字节页写。当卡片被移出，或者当读写器被下电时会重置为默认值。

命令格式

Pseudo-APDU					
CLA	**INS**	**P1**	**P2**	**Lc**	**Page Size**
FFh	01h	00h	00h	01h	

其中：

Page size = 03h：8 字节页写

= 04h：16 字节页写

= 05h：32 字节页写

= 06h：64 字节页写

= 07h：128 字节页写

响应数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

8.1.3. READ_MEMORY_CARD

命令格式

Pseudo-APDU				
CLA	INS	Byte Address		MEM_L
		MSB	LSB	
FFh	B0h			

其中：

Byte Address 存储卡的内存地址位置

MEM_L 待从存储卡读取的数据的长度

响应数据格式

BYTE 1	...	BYTE N	SW1	SW2

其中：

BYTE x 从存储卡中读取的数据

SW1 SW2 = 90 00h（未发生错误）

8.1.4. WRITE_MEMORY_CARD

命令格式

Pseudo-APDU							
CLA	INS	Byte Address		MEM_L	Byte 1	....	Byte n
		MSB	LSB				
FFh	D0h						

其中：

Byte Address 存储卡的内存地址位置

MEM_L 要写入存储卡的数据的长度。

Byte x 要写入存储卡的数据

响应数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

8.2. 存储卡 – 32、64、128、256、512 和 1024 kilobit I2C 卡

8.2.1. SELECT_CARD_TYPE

此命令用于对插入读写器的选定的卡片进行上电/下电，同时进行卡片复位操作。

注： *只有使用 SCardConnect( ) API 建立逻辑智能卡读写器通信之后才可以使用此命令。对于 SCardConnect( ) API 的详细说明参见 PC/SC 规范。*

命令格式

Pseudo-APDU					
CLA	**INS**	**P1**	**P2**	**Lc**	**Card Type**
FFh	A4h	00h	00h	01h	02h

响应数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

8.2.2. SELECT_PAGE_SIZE

此命令会选择用于读取智能卡的页面大小。默认值是 8 字节页写。当卡片被移出，或者当读写器被下电时会重置为默认值。

命令格式

Pseudo-APDU					
CLA	**INS**	**P1**	**P2**	**Lc**	**Page size**
FFh	01h	00h	00h	01h	

其中：

Data 待发送给卡片的 TPDU

Page size = 03h：8 字节页写

= 04h：16 字节页写

= 05h：32 字节页写

= 06h：64 字节页写

= 07h：128 字节页写

响应数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

8.2.3. READ_MEMORY_CARD

命令格式

Pseudo-APDU				
CLA	INS	Byte Address		MEM_L
		MSB	LSB	
FFh				

其中：

INS = B0h：32、64、128、256 和 512 kilobit IIC 卡
= 1011 000*b：1024 kilobit IIC 卡，
其中 * 表示 17 位地址的 MSB。

Byte Address 存储卡的内存地址位置

MEM_L 要从存储卡读取的数据的长度

响应数据格式

BYTE 1	...	BYTE N	SW1	SW2

其中：

BYTE x 从存储卡中读取的数据

SW1 SW2 = 90 00h（未发生错误）

8.2.4. WRITE_MEMORY_CARD

命令格式

Pseudo-APDU							
CLA	INS	Byte Address		MEM_L	Byte 1	....	Byte n
		MSB	LSB				
FFh							

其中：

INS = D0h：32、64、128、256 和 512 kilobit IIC 卡
= 1101 000*b：1024 kilobit IIC 卡，

	其中 * 表示 17 位地址的 MSB。
Byte Address	存储卡的内存地址位置
MEM_L	要写入存储卡的数据的长度
Byte x	要写入存储卡的数据

响应数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

8.3. 存储卡 – Atmel® AT88SC153

8.3.1. SELECT_CARD_TYPE

此命令用于对插入读写器的选定的卡片进行上电/下电，同时进行卡片复位操作。还将选择页面大小为 8 字节页写。

注： *只有使用 SCardConnect( ) API 建立逻辑智能卡读写器通信之后才可以使用此命令。对于 SCardConnect( ) API 的详细说明参见 PC/SC 规范。*

命令格式

Pseudo-APDU					
CLA	**INS**	**P1**	**P2**	**Lc**	**Card Type**
FFh	A4h	00h	00h	01h	03h

响应数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

8.3.2. READ_MEMORY_CARD

命令格式

Pseudo-APDU				
CLA	**INS**	**P1**	**Byte Address**	**MEM_L**
FFh		00h		

其中：

INS = B0h：读取 00b 区
= B1h：读取 01b 区
= B2h：读取 10b 区
= B3h：读取 11b 区
= B4h：读取标识位

Byte Address 存储卡的内存地址位置

MEM_L 待从存储卡读取的数据的长度

响应数据格式

BYTE 1	...	BYTE N	SW1	SW2

其中：

BYTE x 从存储卡中读取的数据

SW1 SW2 = 90 00h（未发生错误）

8.3.3. WRITE_MEMORY_CARD

命令格式

Pseudo-APDU							
CLA	**INS**	**P1**	**Byte Address**	**MEM_L**	**Byte 1**	**....**	**Byte n**
FFh		00h					

其中：

INS = D0h：写入 00b 区
= D1h：写入 01b 区
= D2h：写入 10b 区
= D3h：写入 11b 区
= D4h：写入标识位

Byte Address 存储卡的内存地址位置

MEM_L 要写入存储卡的数据的长度

MEM_D 待写入存储卡的数据

响应数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

8.3.4. VERIFY_PASSWORD

命令格式

Pseudo-APDU							
CLA	**INS**	**P1**	**P2**	**Lc**	**Pw(0)**	**Pw(1)**	**Pw(2)**
FFh	20h	00h		03h			

其中：

Pw(0),Pw(1),Pw(2) 待发送给存储卡的密码

P2 = 0000 00rpb

其中的“rp”位指明待比较的密码

r = 0：写密码

r = 1：读密码

p：密码集编号，

rp = 01：安全密码。

响应数据格式

SW1	**SW2 ErrorCnt**
90h	

其中：

SW1 = 90h

SW2 (ErrorCnt) 错误计数器。FFh 表示验证正确，00h 表示密码被锁定（或超过最大重试次数）。其它值表示当前验证失败。

8.3.5. INITIALIZE_AUTHENTICATION

命令格式

Pseudo-APDU								
CLA	**INS**	**P1**	**P2**	**Lc**	**Q(0)**	**Q(1)**	**…**	**Q(7)**
FFh	84h	00h	00h	08h				

其中：

Q(0),Q(1)…Q(7) 主机随机数，8 个字节

响应数据格式

SW1	**SW2**

其中：

SW1 SW2 = 90 00h（未发生错误）

8.3.6. VERIFY_AUTHENTICATION

命令格式

Pseudo-APDU								
CLA	**INS**	**P1**	**P2**	**Lc**	**Ch(0)**	**Ch(1)**	**…**	**Ch(7)**
FFh	82h	00h	00h	08h				

其中：

Ch(0),Ch(1)…Ch(7) 主机挑战数，8 个字节

响应数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

8.4. 存储卡 – Atmel® AT88C1608

8.4.1. SELECT_CARD_TYPE

此命令用于对插入读写器的选定的卡片进行上电/下电，同时进行卡片复位操作。还将选择页面大小为 16 字节页写。

注：只有使用 SCardConnect( ) API 建立逻辑智能卡读写器通信之后才可以使用此命令。对于 SCardConnect( ) API 的详细说明参见 PC/SC 规范。

命令格式

Pseudo-APDU					
CLA	**INS**	**P1**	**P2**	**Lc**	**Card Type**
FFh	A4h	00h	00h	01h	04h

响应数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

8.4.2. READ_MEMORY_CARD

命令格式

Pseudo-APDU				
CLA	**INS**	**Zone Address**	**Byte Address**	**MEM_L**
FFh				

其中：

INS = B0h：读取用户区
= B1h：读取配置区或读取标识位

Zone Address = 0000 $0A_{10}A_9A_8$b，其中 A_{10} 是区域地址的 MSB
= 读标识位时无关

Byte Address = $A_7A_6A_5A_4$ $A_3A_2A_1A_0$b 是存储卡的内存地址位置
= 1000 0000b 读取标识位

MEM_L 待从存储卡读取的数据的长度

响应数据格式

BYTE 1	...	BYTE N	SW1	SW2

其中：

BYTE x 从存储卡中读取的数据

SW1 SW2 = 90 00h（未发生错误）

8.4.3. WRITE_MEMORY_CARD

命令格式

Pseudo-APDU							
CLA	**INS**	**Zone Address**	**Byte Address**	**MEM_L**	**Byte 1**	**...**	**Byte n**
FFh							

其中：

INS = D0h：写用户区

= D1h：写配置区或写标识位

Zone Address = 0000 $0A_{10}A_9A_8$b，其中 A_{10} 是区域地址的 MSB

= 写标识位时无关

Byte Address = $A_7A_6A_5A_4$ $A_3A_2A_1A_0$b 是存储卡的内存地址位置

= 1000 0000b：写标识位

MEM_L 要写入存储卡的数据的长度

Byte x 要写入存储卡的数据

响应数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

8.4.4. VERIFY_PASSWORD

命令格式

Pseudo-APDU								
CLA	**INS**	**P1**	**P2**	**Lc**	**数据**			
FFh	20h	00h	00h	04h	RP	Pw(0)	Pw(1)	Pw(2)

其中：

Pw(0),Pw(1),Pw(2) 待发送给存储卡的密码

RP = 0000 $rp_2p_1p_0b$

其中“$rp_2p_1p_0$”位指明待比较的密码

r = 0 ： 写密码

r = 1 ： 读密码

$p_2p_1p_0$ ： 密码集编号。

($rp_2p_1p_0$ = 0111：安全密码）

响应数据格式

SW1	SW2 ErrorCnt
90h	

其中：

SW1 = 90h

SW2 (ErrorCnt) = 错误计数器。FFh 表示验证正确，00h 表示密码被锁定（或超过最大重试次数）。其它值表示当前验证失败。

8.4.5. INITIALIZE_AUTHENTICATION

命令格式

Pseudo-APDU								
CLA	**INS**	**P1**	**P2**	**Lc**	**Q(0)**	**Q(1)**	**…**	**Q(7)**
FFh	84h	00h	00h	08h				

其中：

Byte Address 存储卡的内存地址位置

Q(0),Q(1)…Q(7) 主机随机数，8 个字节

响应数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

8.4.6. VERIFY_AUTHENTICATION

命令格式

Pseudo-APDU								
CLA	**INS**	**P1**	**P2**	**Lc**	**Q1(0)**	**Q1(1)**	**…**	**Q1(7)**
FFh	82h	00h	00h	08h				

其中：

Byte Address 存储卡的内存地址位置

Q1(0),Q1(1)…Q1(7) 主机挑战数，8 个字节

响应数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

8.5. 存储卡 – SLE4418/SLE4428/SLE5518/SLE5528

8.5.1. SELECT_CARD_TYPE

此命令用于对插入读写器的选定的卡片进行上电/下电，同时进行卡片复位操作。

注：*只有使用 SCardConnect( ) API 建立逻辑智能卡读写器通信之后才可以使用此命令。对于 SCardConnect( ) API 的详细说明参见 PC/SC 规范。*

命令格式

Pseudo-APDU					
CLA	**INS**	**P1**	**P2**	**Lc**	**Card Type**
FFh	A4h	00h	00h	01h	05h

响应数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

8.5.2. READ_MEMORY_CARD

命令格式

Pseudo-APDU				
CLA	**INS**	**Byte Address**		**MEM_L**
		MSB	**LSB**	
FFh	B0h			

其中：

MSB Byte Address = 0000 00A_9A_8b 是存储卡的内存地址位置

LSB Byte Address = $A_7A_6A_5A_4$ $A_3A_2A_1A_0$b 是存储卡的内存地址位置

MEM_L 待从存储卡读取的数据的长度

响应数据格式

BYTE 1	…	BYTE N	SW1	SW2

其中：

BYTE x 从存储卡中读取的数据

SW1 SW2 = 90 00h（未发生错误）

8.5.3. READ_PRESENTATION_ERROR_COUNTER_MEMORY_CARD (SLE4428 和 SLE5528)

此命令用于读取密码输入错误计数器。

命令格式

Pseudo-APDU				
CLA	**INS**	**P1**	**P2**	**MEM_L**
FFh	B1h	00h	00h	03h

响应数据格式

ERRCNT	DUMMY 1	DUMMY 2	SW1	SW2

其中：

ERRCNT 错误计数器。FFh 表示最后一次验证正确。00h 表示密码被锁定（超过最大重试次数）。其它值表示最后一次验证失败。

DUMMY 从卡片读取的 2 个字节的虚拟数据

SW1 SW2 = 90 00h（未发生错误）

8.5.4. READ_PROTECTION_BIT

命令格式

Pseudo-APDU				
CLA	**INS**	**Byte Address**		**MEM_L**
		MSB	**LSB**	
FFh	B2h			

其中：

MSB Byte Address = 0000 00A_9A_8b 是存储卡的内存地址位置

LSB Byte Address = $A_7A_6A_5A_4$ $A_3A_2A_1A_0$b 是存储卡的内存地址位置

MEM_L 要从卡片中读取的保护位的长度，位数是 8 的倍数。最大值为 32。

MEM_L = 1 + INT((位数 - 1)/8)

例如，要读取始于内存 0010h 的 8 个保护位，应当运行下面的私有 APDU：

FF B2 00 10 01h

响应数据格式

PROT 1	…	PROT L	SW1	SW2

其中：

PROT y 含有保护位的字节

SW1 SW2 = 90 00h（未发生错误）

在 PROT 字节中，保护位的排列如下：

PROT 1								PROT 2								…							
P8	P7	P6	P5	P4	P3	P2	P1	P16	P15	P14	P13	P12	P11	P10	P9	..	..	..	..	..	..	P18	P17

Px 是响应数据中 BYTE x 的保护位

‘0’字节被写保护

‘1’字节可以被写入

8.5.5. WRITE_MEMORY_CARD

命令格式

Pseudo-APDU							
CLA	INS	Byte Address		MEM_L	Byte 1	....	Byte N
		MSB	LSB				
FFh	D0h						

其中：

MSB Byte Address = 0000 00A_9A_8b 是存储卡的内存地址位置

LSB Byte Address = $A_7A_6A_5A_4$ $A_3A_2A_1A_0$b 是存储卡的内存地址位置

MEM_L 要写入存储卡的数据的长度

Byte x 要写入存储卡的数据

响应数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

8.5.6. WRITE_PROTECTION_MEMORY_CARD

命令中指定的每个字节用于在卡片中与存储在特定地址位置中的字节做比较。如果数据相符，则相应的保护位就会不可逆地被设定为“0”。

命令格式

Pseudo-APDU							
CLA	INS	Byte Address		MEM_L	Byte 1	....	Byte N
		MSB	LSB				
FFh	D1h						

其中：

MSB Byte Address = 0000 00A_9A_8b 是存储卡的内存地址位置

LSB Byte Address = $A_7A_6A_5A_4$ $A_3A_2A_1A_0$b 是存储卡的内存地址位置

MEM_L 要写入存储卡的数据的长度

Byte x 要与卡片内始于 Byte Address 的数据做比较的 Byte 值。BYTE 1 与在 Byte Address 的数据比较；BYTE N 与在（Byte Address + N -1）的数据比较。

响应数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

8.5.7. PRESENT_CODE_MEMORY_CARD (SLE4428 和 SLE5528)

此命令用于向存储卡提交密码，使能够对 SLE4428 和 SLE5528 进行写操作。执行以下操作：

1. 搜索密码输入错误计数器中值为‘1’的位，然后将该位写为‘0’。
2. 向卡片提交指定的密码。
3. 尝试擦除密码输入错误计数器。

命令格式

Pseudo-APDU						
CLA	INS	P1	P2	MEM_L	CODE	
					Byte 1	Byte 2
FFh	20h	00h	00h	02h		

其中：

CODE 2 个字节的密码（PIN）

响应数据格式

SW1	SW2 ErrorCnt
90h	

其中：

SW1 = 90h

SW2 (ErrorCnt) = 错误计数器。FFh 表示校验成功。00h 表示密码被锁定（或超过最大重试次数）。其它值表示当前验证失败。

8.6. 存储卡 – SLE4432/SLE4442/SLE5532/SLE5542

8.6.1. SELECT_CARD_TYPE

此命令用于对插入读写器的选定的卡片进行上电/下电，同时进行卡片复位操作。

注：*只有使用 SCardConnect( ) API 建立逻辑智能卡读写器通信之后才可以使用此命令。对于 SCardConnect( ) API 的详细说明参见 PC/SC 规范。*

命令格式

Pseudo-APDU					
CLA	**INS**	**P1**	**P2**	**Lc**	**Card Type**
FFh	A4h	00h	00h	01h	06h

响应数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

8.6.2. READ_MEMORY_CARD

命令格式

Pseudo-APDU				
CLA	**INS**	**P1**	**Byte Address**	**MEM_L**
FFh	B0h	00h		

其中：

Byte Address = $A_7A_6A_5A_4$ $A_3A_2A_1A_0$b 是存储卡的内存地址位置

MEM_L 待从存储卡读取的数据的长度

响应数据格式

BYTE 1	…	BYTE N	SW1	SW2

其中：

BYTE x 从存储卡中读取的数据

SW1 SW2 = 90 00h（未发生错误）

8.6.3. READ_PRESENTATION_ERROR_COUNTER_MEMORY_CARD (SLE4442 和 SLE5542)

此命令用于读取密码输入错误计数器。

命令格式

Pseudo-APDU				
CLA	**INS**	**P1**	**P2**	**MEM_L**
FFh	B1h	00h	00h	04h

响应数据格式

ERRCNT	DUMMY 1	DUMMY 2	DUMMY 3	SW1	SW2

其中：

ERRCNT 错误计数器。07h 表示最后一次验证正确。00h 表示密码被锁定（超过最大重试次数）。其它值表示最后一次验证失败。

DUMMY 从卡片读取的 3 个字节的虚拟数据

SW1 SW2 = 90 00h（未发生错误）

8.6.4. READ_PROTECTION_BITS

此命令用于读取前 32 个字节的保护位。

命令格式

Pseudo-APDU				
CLA	**INS**	**P1**	**P2**	**MEM_L**
FFh	B2h	00h	00h	04h

响应数据格式

PROT 1	PROT 2	PROT 3	PROT 4	SW1	SW2

其中：

PROT y 含有保护位的字节

SW1 SW2 = 90 00h（未发生错误）

在 PROT 字节中，保护位的排列如下：

PROT 1								PROT 2								...							
P8	P7	P6	P5	P4	P3	P2	P1	P16	P15	P14	P13	P12	P11	P10	P9	..	..	..	..	..	..	P18	P17

其中：

Px 是响应数据中 BYTE x 的保护位

‘0’字节被写保护

‘1’字节可以被写入

8.6.5. WRITE_MEMORY_CARD

命令格式

Pseudo-APDU							
CLA	**INS**	**P1**	**Byte Address**	**MEM_L**	**Byte 1**	**....**	**Byte N**
FFh	D0h	00h					

其中：

Byte Address = $A_7A_6A_5A_4\ A_3A_2A_1A_0$b 是存储卡的内存地址位置

MEM_L 要写入存储卡的数据的长度

Byte x 要写入存储卡的数据

响应数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

8.6.6. WRITE_PROTECTION_MEMORY_CARD

命令指定的每一个字节均在卡片内部与存储在特定地址中的字节进行对比，若数据相符，则相应的保护位就会被不可逆转的设定为‘0’。

命令格式

Pseudo-APDU							
CLA	**INS**	**P1**	**Byte Address**	**MEM_L**	**Byte 1**	**....**	**Byte N**
FFh	D1h	00h					

其中：

Byte Address = $000A_4\ A_3A_2A_1A_0$b (00h - 1Fh) 是存储卡的保护内存地址位置

MEM_L 要写入存储卡的数据的长度

Byte x 要与卡片内始于 Byte Address 的数据做比较的 Byte 值。BYTE 1 与在 Byte Address 的数据比较；BYTE N 与在（Byte Address + N -1）的数据比较。

响应数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

8.6.7. PRESENT_CODE_MEMORY_CARD (SLE4442 和 SLE5542)

此命令用于向存储卡提交密码，使能够对 SLE4442 和 SLE5542 进行写操作。执行以下操作：

1. 搜索密码输入错误计数器中值为‘1’的位，然后将该位写为‘0’。
2. 向卡片提交指定的密码。
3. 尝试擦除密码输入错误计数器。

命令格式

Pseudo-APDU							
CLA	INS	P1	P2	MEM_L	CODE		
					Byte 1	Byte 2	Byte 3
FFh	20h	00h	00h	03h			

其中：

CODE 3 个字节的密码（PIN）

响应数据格式

SW1	SW2 ErrorCnt
90h	

其中：

SW1 = 90h

SW2 (ErrorCnt) = 错误计数器。07h 表示验证正确。00h 表示密码被锁定（超过最大重试次数）。其它值表示当前验证失败。

8.6.8. CHANGE_CODE_MEMORY_CARD (SLE4442 和 SLE5542)

此命令用于将特定数据作为新密码写入卡片。

执行此命令之前，需要先使用 *PRESENT_CODE* 命令向卡片提交当前密码。

命令格式

Pseudo-APDU							
CLA	INS	P1	P2	MEM_L	CODE		
					Byte 1	Byte 2	Byte 3
FFh	D2h	00h	01h	03h			

响应数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

8.7. 存储卡 – SLE4406/SLE4436/SLE5536/SLE6636

8.7.1. SELECT_CARD_TYPE

此命令用于对插入读写器的选定的卡片进行上电/下电，同时进行卡片复位操作。

注：*只有使用 SCardConnect( ) API 建立逻辑智能卡读写器通信之后才可以使用此命令。对于 SCardConnect( ) API 的详细说明参见 PC/SC 规范。*

命令格式

Pseudo-APDU					
CLA	**INS**	**P1**	**P2**	**Lc**	**Card Type**
FFh	A4h	00h	00h	01h	07h

响应数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

8.7.2. READ_MEMORY_CARD

命令格式

Pseudo-APDU				
CLA	**INS**	**P1**	**Byte Address**	**MEM_L**
FFh	B0h	00h		

其中：

Byte Address = 存储卡的内存地址位置

MEM_L 待从存储卡读取的数据的长度

响应数据格式

BYTE 1	…	BYTE N	SW1	SW2

其中：

BYTE x 从存储卡中读取的数据

SW1 SW2 = 90 00h（未发生错误）

8.7.3. WRITE_ONE_BYTE_MEMORY_CARD

此命令用于向所插入卡片的特定地址写一个字节。该字节从 LSB 开始写入卡片，也就是说，卡片地址 bit 0 被视为 byte 0 的 LSB。

此类卡片有四种不同的写入模式，通过命令数据域内的标志加以区分。

a) **Write**

命令中指定的字节值被写入特定的地址，可用于向卡片写入个人化信息和计数器值。

b) **Write with carry**

命令中指定的字节值被写入特定的地址，且命令被送至卡片来擦除下一个低位计数器。此模式仅适用于卡内计数器值的更新。

c) **Write with backup enabled** (SLE4436, SLE5536 and SLE6636 only)

命令中指定的字节值被写入特定的地址，可用于向卡片写入个人化信息和计数器值。同时启用备份位，保护数据免受卡片插拔导致的损失。

d) **Write with carry and backup enabled** (SLE4436, SLE5536 and SLE6636 only)

命令中指定的字节值被写入特定的地址，且命令被送至卡片来擦除下一个低位计数器。此模式仅适用于卡内计数器值的更新。同时启用备份位，保护数据免受卡片插拔导致的损失。

在这四种模式下，指定地址上的字节在写操作前不会被擦除，所以存储位只能由“1”设为“0”。

SLE4436 卡和 SLE5536 卡的备份模式可以在写操作中被启用或禁用。

命令格式

Pseudo-APDU						
CLA	**INS**	**P1**	**Byte Address**	**MEM_L**	**MODE**	**BYTE**
FFh	D0h	00h		02h		

其中：

Byte Address = 存储卡的内存地址位置

MODE 指定写入模式和备份选项

00h：Write

01h：Write with carry

02h：Write with backup enabled (SLE4436, SLE5536 and SLE6636 only)

03h：Write with carry and with backup enabled (SLE4436, SLE5536 and SLE6636 only)

BYTE 待写入卡片的字节值

响应数据格式

SW1	**SW2**

其中：

SW1 SW2 = 90 00h（未发生错误）

8.7.4. PRESENT_CODE_MEMORY_CARD

此命令用于向存储卡提交密码来启用卡片个人化模式，执行的操作如下：

1. 搜索密码输入错误计数器中值为‘1’的位，然后将该位写为‘0’。
2. 向卡片提交指定的密码。

密码提交后，ACR3901x 不会尝试擦除密码输入计数器，必须由应用软件通过单独的‘*Write with carry*’命令来进行。

命令格式

Pseudo-APDU								
CLA	INS	P1	P2	MEM_L	CODE			
					ADDR	Byte 1	Byte 2	Byte 3
FFh	20h	00h	00h	04h	09h			

其中：

ADDR 卡片中输入计数器的字节地址

CODE 3 个字节的密码（PIN）

响应数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

8.7.5. AUTHENTICATE_MEMORY_CARD (SLE4436、SLE5536 和 SLE6636)

此命令用于从 SLE5536 卡或 SLE6636 卡读取一个卡片认证证书，ACR3901x 执行的操作如下：

1. 根据命令在卡片中选择 Key 1 或 Key 2。
2. 将命令中指定的随机数提交给卡片。
3. 为卡片计算出的每位认证数据生成指定数量的时钟脉冲。
4. 从卡片中读取 16 位的认证数据。
5. 将卡片复位回正常的操作模式。

认证的过程分为两步：步骤 1 是将认证证书发送至卡片。步骤 2 是取回卡片计算出的 2 个字节的认证数据。

步骤 1：向卡片发送认证证书

命令格式

Pseudo-APDU											
CLA	INS	P1	P2	MEM_L	CODE						
					KEY	CLK_CNT	Byte 1	Byte 2	……	Byte 5	Byte 6
FFh	84h	00h	00h	08h							

其中：

KEY 用于计算认证证书的密钥：

00h：Key 1，不带密码块链接

01h：Key 2，不带密码块链接

80h：key 1，带密码块链接 (仅限 SLE5536 和 SLE6636）

81h：key 2，带密码块链接 (仅限 SLE5536 和 SLE6636）

CLK_CNT 待提供给卡片的时钟脉冲的个数，卡片将该脉冲用于计算认证证书的每个位。通常为 160（A0）。

BYTE 1...6 卡片随机数据

响应数据格式

SW1	SW2
61h	02h

其中：

SW1 SW2 = 61 02h (未发生错误），表示两个字节的认证数据准备就绪。可以通过 *GET_RESPONSE* 命令获取认证数据。

步骤 2：取回认证数据（Get_Response）

命令格式

Pseudo-APDU				
CLA	INS	P1	P2	MEM_L
FFh	C0h	00h	00h	02h

响应数据格式

CERT		SW1	SW2

其中：

CERT 卡片计算出的 16 位的认证数据。BYTE 1 的 LSB 是从卡片中读取的第一个认证位。

SW1 SW2 = 90 00h（未发生错误）

8.8. 存储卡 – SLE 4404

8.8.1. SELECT_CARD_TYPE

此命令用于对插入读写器的选定的卡片进行上电/下电，同时进行卡片复位操作。

注：*只有使用 SCardConnect( ) API 建立逻辑智能卡读写器通信之后才可以使用此命令。对于 SCardConnect( ) API 的详细说明参见 PC/SC 规范。*

命令格式

Pseudo-APDU					
CLA	**INS**	**P1**	**P2**	**Lc**	**Card Type**
FFh	A4h	00h	00h	01h	08h

响应数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

8.8.2. READ_MEMORY_CARD

命令格式

Pseudo-APDU				
CLA	**INS**	**P1**	**Byte Address**	**MEM_L**
FFh	B0h	00h		

其中：

Byte Address = 存储卡的内存地址位置

MEM_L 待从存储卡读取的数据的长度

响应数据格式

BYTE 1	…	BYTE N	SW1	SW2

其中：

BYTE x 从存储卡中读取的数据

SW1 SW2 = 90 00h（未发生错误）

8.8.3. WRITE_MEMORY_CARD

此命令用于向所插入卡片的特定地址写入数据。该字节从 LSB 开始写入卡片，也就是说，卡片地址 bit 0 被视为 byte 0 的 LSB。

指定地址上的字节在写操作前不会被擦除，所以存储位只能由‘1’设为‘0’。

命令格式

Pseudo-APDU							
CLA	**INS**	**P1**	**Byte Address**	**MEM_L**	**Byte 1**	**…**	**Byte N**
FFh	D0h	00h					

其中：

Byte Address = 存储卡的内存地址位置

MEM_L 要写入存储卡的数据的长度

BYTE 待写入卡片的字节值

响应数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

8.8.4. ERASE_SCRATCH_PAD_MEMORY_CARD

此命令用于擦除所插入卡片的暂存存储器的数据。暂存存储器内所有的存储位都会被设定为状态“1”。

要擦除错误计数器或者用户区，请使用 *VERIFY_USER_CODE* 命令，如 **8.8.5 节**所示。

命令格式

Pseudo-APDU				
CLA	**INS**	**P1**	**Byte Address**	**MEM_L**
FFh	D2h	00h		00h

其中：

Byte Address 暂存存储区的内存字节地址位置

通常为 02h

响应数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

8.8.5. VERIFY_USER_CODE

此命令用于向插入的卡片提交用户密码（2 个字节）。用户密码旨在使卡的内存能够被访问。

执行的操作如下：

1. 向卡片提交指定的密码。
2. 搜索密码输入错误计数器中值为‘1’的位，然后将该位写为‘0’。
3. 擦除密码输入错误计数器。提交的密码验证正确后，用户错误计数器可被擦除。

命令格式

Pseudo-APDU						
CLA	INS	Error Counter LEN	Byte Address	MEM_L	CODE	
					Byte 1	Byte 2
FFh	20h	04h	08h	02h		

其中：

Error Counter LEN 密码输入错误计数器的长度，单位为比特

Byte Address 卡片中密钥的字节地址

CODE 2 字节的用户密码

响应数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

= 63 00h（如果不再有重试的机会）

注： *收到响应 SW1 SW2 = 90 00h 后，应当再次读取用户错误计数器，检查 VERIFY_USER_CODE 是否正确。如果用户错误计数器被擦除并且等于‘FFh’，证明先前的验证成功。*

8.8.6. VERIFY_MEMORY_CODE

此命令用于向插入的卡片提交存储密码（4 个字节）。该存储密码可授权用户重新载入用户内存及用户密码。

执行的操作如下：

1. 向卡片提交指定的密码。
2. 搜索密码输入错误计数器中值为‘1’的位，然后将该位写为‘0’。
3. 擦除密码输入错误计数器。请注意，存储错误计数器的内容不能被擦除。

命令格式

Pseudo-APDU								
CLA	INS	Error Counter LEN	Byte Address	MEM_L	CODE			
					Byte 1	Byte 2	Byte 3	Byte 4
FFh	20h	40h	28h	04h				

其中：

Error Counter LEN 密码输入错误计数器的长度，单位为比特

Byte Address 卡片中密钥的字节地址

CODE 4 个字节的存储密码

响应数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

= 63 00h（如果不再有重试的机会）

注：*收到响应 SW1 SW2 = 90 00h 后，应当再次读取应用区，检查 VERIFY_MEMORY_CODE 是否正确。如果应用区域的全部数据都被擦除并且等于‘FFh’，证明先前的验证成功。*

8.9. 存储卡 – AT88SC101/AT88SC102/AT88SC1003

8.9.1. SELECT_CARD_TYPE

此命令用于对插入读写器的选定的卡片进行上电/下电，同时进行卡片复位操作。

注：*只有使用 SCardConnect( ) API 建立逻辑智能卡读写器通信之后才可以使用此命令。对于 SCardConnect( ) API 的详细说明参见 PC/SC 规范。*

命令格式

Pseudo-APDU					
CLA	**INS**	**P1**	**P2**	**Lc**	**Card Type**
FFh	A4h	00h	00h	01h	09h

响应数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

8.9.2. READ_MEMORY_CARD

命令格式

Pseudo-APDU				
CLA	**INS**	**P1**	**Byte Address**	**MEM_L**
FFh	B0h	00h		

其中：

Byte Address = 存储卡的内存地址位置

MEM_L 待从存储卡读取的数据的长度

响应数据格式

BYTE 1	…	BYTE N	SW1	SW2

其中：

BYTE x 从存储卡中读取的数据

SW1 SW2 = 90 00h（未发生错误）

8.9.3. WRITE_MEMORY_CARD

此命令用于向所插入卡片的特定地址写入数据。该字节从 LSB 开始写入卡片，也就是说，卡片地址 bit 0 被视为 byte 0 的 LSB。

指定地址上的字节在写操作前不会被擦除，所以存储位只能由‘1’设为‘0’。

命令格式

Pseudo-APDU							
CLA	**INS**	**P1**	**Byte Address**	**MEM_L**	**Byte 1**	**....**	**Byte N**
FFh	D0h	00h					

其中：

Byte Address 存储卡的内存地址位置

MEM_L 要写入存储卡的数据的长度

BYTE 待写入卡片的字节值

响应数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

8.9.4. ERASE_NON_APPLICATION_ZONE

此命令用于擦除存储在非应用区的数据。EEPROM 内存由 16 位字构成。即使只擦除单独的一个位，内存中的整个字都会被 ERASE 操作所清除。因此对某个字中的任何位执行 ERASE 操作，都会将该字的全部 16 位清除为状态‘1’。

要擦除错误计数器或是在应用区域存储的数据，请参考以下命令：

1. **8.9.5 节**定义的 *ERASE_APPLICATION_ZONE_WITH_ERASE* 命令
2. **8.9.6 节**定义的 *ERASE_APPLICATION_ZONE_WITH_WRITE_AND_ERASE* 命令
3. **8.9.7 节**定义的 *VERIFY_SECURITY_CODE* 命令

命令格式

Pseudo-APDU				
CLA	**INS**	**P1**	**Byte Address**	**MEM_L**
FFh	D2h	00h		00h

其中：

Byte Address 待擦除的字的内存字节地址位置

响应数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

8.9.5. ERASE_APPLICATION_ZONE_WITH_ERASE

此命令可用于下列情况：

1. AT88SC101：擦除应用区域中的数据，EC 功能禁用。
2. AT88SC102：擦除应用区域 1 中的数据。
3. AT88SC102：擦除应用区域 2 中的数据，EC2 功能禁用。
4. AT88SC1003：擦除应用区域 1 中的数据。
5. AT88SC1003：擦除应用区域 2 中的数据，EC2 功能禁用。
6. AT88SC1003：擦除应用区域 3 中的数据。

此命令执行以下操作：

1. 向卡片提交指定的密码
   a. 擦除密码输入错误计数器。提交的密码验证正确后，相应的应用区域中的数据可以被擦除。

命令格式

Pseudo-APDU								
CLA	INS	Error Counter LEN	Byte Address	MEM_L	CODE			
					Byte 1	Byte 2	…	Byte N
FFh	20h	00h						

其中：

Error Counter LEN 密码输入错误计数器的长度，单位为比特。值始终是 00h。

Byte Address 卡片中的应用区密钥的字节地址。正确值请参阅下表：

	Byte Address	LEN
AT88SC101：擦除应用区域，EC 功能禁用	96h	04h
AT88SC102：擦除应用区域 1	56h	06h
AT88SC102：擦除应用区域 2，EC2 功能禁用	9Ch	04h
AT88SC1003：擦除应用区域 1	36h	06h

	Byte Address	LEN
AT88SC1003：擦除应用区域 2，EC2 功能禁用	5Ch	04h
AT88SC1003：擦除应用区域 3	C0h	06h

MEM_L 擦除密钥的长度。正确值请参阅上表。

CODE 擦除密钥的 N 个字节

响应数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

注： *收到响应 SW1 SW2 = 90 00h 后，应当再次读取应用区的数据，检查 ERASE_APPLICATION_ZONE_WITH_ERASE 是否正确。如果应用区域的全部数据都被擦除并且等于'FFh'，则说明先前的验证成功。*

8.9.6. ERASE_APPLICATION_ZONE_WITH_WRITE_AND_ERASE

此命令可用于下列情况：

1. AT88SC101：擦除应用区域中的数据，EC 功能启用。
2. AT88SC102：擦除应用区域 2 中的数据，EC2 功能启用。
3. AT88SC1003：擦除应用区域 2 中的数据，EC2 功能启用。

EC 或 EC2 功能启用后（即：ECEN 或 EC2EN 标识位没有被破坏并处于"1"状态），会执行以下操作：

1. 向卡片提交指定的密码。
2. 搜索密码输入错误计数器中值为'1'的位，然后将该位写为'0'。
3. 擦除密码输入错误计数器。提交的密码验证正确后，相应的应用区域中的数据可以被擦除。

命令格式

Pseudo-APDU								
CLA	INS	Error Counter LEN	Byte Address	MEM_L	CODE			
					Byte 1	Byte 2	Byte 3	Byte 4
FFh	20h	80h		04h				

其中：

Error Counter LEN 密码输入错误计数器的长度，单位为比特值始终是 80h。

Byte Address 卡片中的应用区密钥的字节地址

	Byte Address
AT88SC101	96h
AT88SC102	9Ch
AT88SC1003	5Ch

CODE 4 个字节的擦除密钥

响应数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

= 63 00h（如果不再有重试的机会）

注：收到响应 SW1 SW2 = 90 00h 后，应当再次读取应用区的数据，检查 ERASE_APPLICATION_ZONE_WITH_WRITE_AND_ERASE 是否正确。如果应用区域的全部数据都被擦除并且等于'FFh'，则说明先前的验证成功。

8.9.7. VERIFY_SECURITY_CODE

此命令用于向插入的卡片提交安全密码（2 个字节）。安全密码旨在使卡的内存能够被访问。

执行的操作如下：

1. 向卡片提交指定的密码。
2. 搜索密码输入错误计数器中值为'1'的位，然后将该位写为'0'。
3. 擦除密码输入错误计数器。提交的密码验证正确后，安全密码尝试计数器可被擦除。

命令格式

Pseudo-APDU						
CLA	INS	Error Counter LEN	Byte Address	MEM_L	CODE	
					Byte 1	Byte 2
FFh	20h	08h	0Ah	02h		

其中：

Error Counter LEN 密码输入错误计数器的长度，单位为比特

Byte Address 卡片中密钥的字节地址

CODE 2 个字节的安全密码

响应数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

= 63 00h（如果不再有重试的机会）

注：*收到响应 SW1 SW2 = 90 00h 后，可重新读取安全密码尝试计数器（SCAC），检查 VERIFY_USER_CODE 是否正确。如果 SCAC 被擦除并且等于'FFh'，证明先前的验证成功。*

8.9.8. BLOWN_FUSE

此命令用于更改所插入卡片的标识位。标识位可以是 EC_EN 标识位、EC2EN 标识位、发行商标识位或生产商标识位。

注：*更改标识位是一个不可逆的过程。*

命令格式

Pseudo-APDU								
CLA	INS	Error Counter LEN	Byte Address	MEM_L	CODE			
					Fuse Bit Addr (High)	Fuse Bit Addr (Low)	State of FUS Pin	State of RST Pin
FFh	05h	00h	00h	04h			01h	00h 或 01h

其中：

Fuse Bit Addr (2 个字节) 标识位的位地址。正确值请参阅下表：

State of FUS Pin FUS pin 的状态。始终是 01h。

State of RST Pin RST pin 的状态。正确值请参阅下表。

		Fuse Bit Addr (High)	Fuse Bit Addr (Low)	State of RST Pin
AT88SC101	生产商标识位	05h	80h	01h
	EC_EN 标识位	05h	C9h	01h
	发行商标识位	05h	E0h	01h
AT88SC102	生产商标识位	05h	B0h	01h
	EC2EN 标识位	05h	F9h	01h
	发行商标识位	06h	10h	01h
AT88SC1003	生产商标识位	03h	F8h	00h
	EC2EN 标识位	03h	FCh	00h
	发行商标识位	03h	E0h	00h

响应数据格式

SW1	SW2

其中：

SW1 SW2 = 90 00h（未发生错误）

9.0. 非接触式卡命令

本节介绍 ACR35 的非接触卡命令。

9.1. 非接触接口私有 APDU 指令

9.1.1. Get Data

此命令用于获取 PICC 卡的序列号或 ATS。

GET UID 的 APDU 结构（5 个字节）

命令	CLA	INS	P1	P2	Le
Get Data	FFh	CAh	00h 01h	00h	00h （最大长度）

若 P1=00h，响应格式为获取 UID（UID + 2 个字节）

响应	响应数据域					
结果	UID (LSB)	…	…	UID (MSB)	SW1	SW2

如果 P1 = 01h，则获取 ISO14443 A 类卡的 ATS（ATS + 2 个字节）

响应	响应数据域		
结果	ATS	SW1	SW2

响应状态码

结果	SW1 SW2	含义
成功	90 00h	操作成功完成
警告	62 82h	UID/ATS 的末尾先于 Le 字节到达（Le 大于 UID 的长度）
错误	6C XX	长度错误（错误的 Le：‘XX’表示确切的数字），如果 Le 小于 UID 的长度
错误	63 00h	操作失败
错误	6A 81h	不支持此功能

例如：

//获取 PICC 卡的序列号

UINT8 GET_UID[5]={FF, CA, 00, 00, 00h};

//获取 ISO14443 A 类非接触式卡的 ATS

UINT8 GET_ATS[5]={FF, CA, 01, 00, 00h};

9.2. MIFARE Classic 1K/4K 存储卡的 PICC 命令（T=CL 模拟）

9.2.1. Load Authentication Keys

此命令用于向读写器加载认证密钥。该认证密钥用于验证 MIFARE Classic 1K/4K 存储卡的特定扇区。读写器提供了两种认证密钥位置：易失密钥位置和非易失密钥位置。

Load Authentication Keys 的 APDU 结构（11 个字节）

命令	CLA	INS	P1	P2	Lc	命令数据域
Load Authentication Keys	FFh	82h	密钥结构	密钥号	06h	密钥（6 个字节）

其中：

密钥结构 1 个字节。

00h = 密钥被载入读写器的易失性存储器。

其它 = 保留。

密钥号 1 个字节。

00h – 01h = 用于存储密钥的非易失存储器。密钥被永久地存在读写器中，即使读写器与电脑断开连接也存储在其中。读写器的非易失存储器内可以存储最多 2 个密钥。

注：默认值为 FF FF FF FF FF FFh.

密钥 6 个字节。载入读写器的密钥值。

例如：FF FF FF FF FF FFh。

Load Authentication Keys 的响应结构（2 个字节）

响应	响应数据域	
结果	SW1	SW2

Load Authentication Keys 命令的响应状态码

结果	SW1 SW2	含义
成功	90 00h	操作成功完成。
错误	63 00h	操作失败。

例如：

// 向易失存储器位置 00h 加载密钥 {FF FF FF FF FF FFh}。

APDU = {FF 82 00 00 06 FF FF FF FF FF FFh}

9.2.2. Authentication for MIFARE Classic 1K/4K

此命令使用存储在 ACR3x 中的密钥来验证 MIFARE Classic 1K/4K 卡（PICC）。其中会用到两种认证密钥：TYPE_A 和 TYPE_B。

Load Authentication Keys 的 APDU 结构（10 个字节）

命令	CLA	INS	P1	P2	Lc	命令数据域
Authentication	FFh	86h	00h	00h	05h	认证数据字节

认证数据字节（5 个字节）

字节 1	字节 2	字节 3	字节 4	字节 5
版本 01h	00h	块号	密钥类型	密钥号

其中：

块号 1 个字节

待验证的存储块。

一张 MIFARE 1K 卡分为 16 个扇区，每个扇区包含 4 个连续的块。

例如：扇区 00h 包含块{00h、01h、02h 和 03h}；扇区 01h 包含块{04h、05h、06h 和 07h}；最后一个扇区 0Fh 包含块{3Ch、3Dh、3Eh 和 3Fh}。

验证通过后，读取同一个扇区内的其他块不需要再进行验证。*

详情请参考 MIFARE 1K/4K 卡标准。

***注：** 一旦该块被成功验证，即可访问属于同一扇区的所有块。*

密钥类型 1 个字节。

60h = 密钥被用作 TYPE A 密钥进行验证

61h = 密钥被用作 TYPE B 密钥进行验证

密钥号 1 个字节。

00h ~ 01h = 用于存储密钥的易失存储器读写器与电脑断开连接的时候，密钥会被删除。易失密钥有两个。可以用作不同会话的过程密钥。

Load Authentication Keys 的响应结构（2 个字节）

响应	响应数据域	
结果	SW1	SW2

Load Authentication Keys 命令的响应状态码

结果	SW1 SW2	含义
成功	90 00h	操作成功完成
错误	63 00h	操作失败

扇区 (16 个扇区，每个扇区包含 4 个连续的块)	数据块 (3 个块，每块 16 个字节)	尾部块 (1 个块，16 个字节)	
扇区 0	00 ~ 02h	03h	
扇区 1	04 ~ 06h	07h	
…			1 KB
…			
扇区 14	38 ~ 0Ah	3Bh	
扇区 15	3C ~ 3E	3Fh	

表3 ：MIFARE CLASSIC 1K 卡内存结构

扇区 (32 个扇区，每个扇区包含 4 个连续的块)	数据块 (3 个块，每块 16 个字节)	尾部块 (1 个块，16 个字节)	
扇区 0	00 ~ 02h	03h	
扇区 1	04 ~ 06h	07h	
…			2 KB
…			
扇区 30	78 ~ 7Ah	7Bh	
扇区 31	7C ~ 7Eh	7Fh	

扇区 (8 个扇区，每个扇区包含 16 个连续的块)	数据块 (15 个块，每块 16 个字节)	尾部块 (1 个块，16 个字节)	
扇区 32	80 ~ 8Eh	8Fh	
扇区 33	90 ~ 9Eh	9Fh	
…			2 KB
…			
扇区 38	E0 ~ EEh	EFh	
扇区 39	F0 ~ FEh	FFh	

表4 ：MIFARE CLASSIC 4K 卡内存结构

例如：

// 要使用{TYPE A，密钥号 00h}验证块 04h。

// PC/SC V2.01，弃用

APDU = {FF 88 00 04 60 00h};

同样，

// 要使用{TYPE A，密钥号 00h}验证块 04h。

// PC/SC V2.07

APDU = {FF 86 00 00 05 01 00 04 60 00h}

注：*MIFARE Ultralight 不需要进行验证，其内存可以自由访问。*

字节号	0	1	2	3	页
序列号	SN0	SN1	SN2	BCC0	0
序列号	SN3	SN4	SN5	SN6	1
内部/锁	BCC1	Internal	Lock0	Lock1	2
OTP	OPT0	OPT1	OTP2	OTP3	3
数据读/写	Data0	Data1	Data2	Data3	4
数据读/写	Data4	Data5	Data6	Data7	5
数据读/写	Data8	Data9	Data10	Data11	6
数据读/写	Data12	Data13	Data14	Data15	7
数据读/写	Data16	Data17	Data18	Data19	8
数据读/写	Data20	Data21	Data22	Data23	9
数据读/写	Data24	Data25	Data26	Data27	10
数据读/写	Data28	Data29	Data30	Data31	11
数据读/写	Data32	Data33	Data34	Data35	12
数据读/写	Data36	Data37	Data38	Data39	13
数据读/写	Data40	Data41	Data42	Data43	14
数据读/写	Data44	Data45	Data46	Data47	15

512 位
或者
64 个字节

表5 ：MIFARE Ultralight 卡内存结构

9.2.3. Read Binary Blocks

Read Binary Blocks 命令用于从 PICC 卡中取回多个数据块。执行 Read Binary Blocks 命令前，必须先对数据块/尾部块进行验证。

Read Binary 命令的 APDU 结构（5 个字节）

命令	CLA	INS	P1	P2	Le
Read Binary Blocks	FFh	B0h	00h	块号	待读取的字节数

其中：

块号 1 个字节。起始块。

待读取的字节数 1 个字节。

MIFARE 1K/4K 卡的待读字节的长度应该是 16 字节的倍数；MIFARE Ultralight 卡应该是 4 字节的倍数。

MIFARE 1K 卡的待读字节最大为 48（多块模式；3 个连续的块）。

MIFARE 4K 卡的待读字节最大为 240（多块模式；15 个连续的块）。

MIFARE Ultralight 卡的待读字节最大为 16。

例 1： 10h（16 个字节）。仅起始块（单块模式）

例 2： 40h（64 个字节）。从起始块至起始+3 块。（多块模式）

注：*出于安全原因，多块模式仅用于访问数据块。尾部块不能在多块模式下被访问，请使用单块模式对其进行访问。*

Read Binary Block 的响应结构（4/16 的倍数 + 2 个字节）

响应	响应数据域		
结果	数据（4/16 字节的倍数）	SW1	SW2

Read Binary Block 命令的响应状态码

结果	SW1 SW2	含义
成功	90 00h	操作成功完成
错误	63 00h	操作失败。

例如：

// 从二进制块 04h 中读取 16 个字节（MIFARE 1K 或 4K）

APDU = {FF B0 00 04 10}

从二进制块 80h 开始读取 240 个字节（MIFARE 4K）

// 块 80 至块 8Eh（15 个块）

APDU = {FF B0 00 80 F0}

9.2.4. Update Binary Blocks

Update Binary Blocks 命令用于向 PICC 卡写入多个数据块。执行 Update Binary Blocks 命令前，必须先对数据块/尾部块进行验证。

Update Binary 命令的 APDU 结构（16 的倍数 + 5 个字节）

命令	CLA	INS	P1	P2	Le	命令数据域
Update Binary Blocks	FFh	D6h	00h	块号	待更新的字节数	块数据（16 字节的倍数）

其中：

块号 1 个字节。待更新的起始块

待更新的字节数 1 个字节。MIFARE 1K/4K 卡的待更新字节的长度应该是 16 字节的倍数；MIFARE Ultralight 卡应该是 4 字节的倍数。

MIFARE 1K 卡的待更新字节最大为 48（多块模式；3 个连续的块）。

MIFARE 4K 卡的待更新字节最大为 240（多块模式；15 个连续的块）。

MIFARE Ultralight 卡的待读字节数最大为 16。

例 1： 10h（16 个字节）。仅起始块。（单块模式）

例 2： 30h（48 个字节）。从起始块至起始+2 块。（多块模式）

注： 出于安全原因，多块模式仅用于访问数据块。尾部块不能在多块模式下被访问，请使用单块模式对其进行访问。

块数据 16 的倍数 + 2 个字节（或 6 个字节）

待写入二进制块的数据。

Update Binary Block 命令的响应状态码

结果	SW1 SW2	含义
成功	90 00h	操作成功完成。
错误	63 00h	操作失败。

例如：

// 将 MIFARE 1K 或 4K 卡中的二进制块 04h 的数据更新为 {00 01 … 0F}

APDU = {FF D6 00 04 10 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F}

将 MIFARE Ultralight 中的二进制块 04 的数据更新为{00 01 02 03}

APDU = {FF D6 00 04 04 00 01 02 03}

9.2.5. Value Block Operation (INC, DEC, STORE)

Value Block Operation 命令用于进行数值操作（例如：增加值块的值等）。

Value Block Operation 的 APDU 结构（10 个字节）

命令	CLA	INS	P1	P2	Le	命令数据域	
Value Block Operation	FFh	D7h	00h	块号	05h	VB_OP	VB_Value （4 个字节） {MSB ..LSB}

其中：

块号 1 个字节。待操作的值块。

VB_OP 1 个字节。

00h = 将 VB_Value 存入该块，然后该块变为一个值块。

01h = 使值块的值增加 VB_Value。仅适用于对值块的操作。

02h = 使值块的值减少 VB_Value。仅适用于对值块的操作。

VB_Value 4 个字节。用于算数运算的数值，是一个有符号长整数。

例 1： Decimal –4 = {FFh, FFh, FFh, FCh}

VB_Value			
MSB			LSB
FFh	FFh	FFh	FCh

例 2： Decimal 1 = {00h, 00h, 00h, 01h}

VB_Value			
MSB			LSB
00h	00h	00h	01h

Value Block Operation 的响应结构（2 个字节）

响应	响应数据域	
结果	SW1	SW2

Value Block Operation 响应码

结果	SW1 SW2	含义
成功	90 00h	操作成功完成。
错误	63 00h	操作失败。

9.2.6. Read Value Block

Read Value Block 命令用于获取值块中的数值，仅适用于对值块的操作。

Read Value Block 命令的 APDU 结构（5 个字节）

命令	CLA	INS	P1	P2	Le
Read Value Block	FFh	B1h	00h	块号	04h

其中：

块号 1 个字节。待访问的值块。

Read Value Block 的响应结构（4 + 2 个字节）

响应	响应数据域		
结果	值 {MSB … LSB}	SW1	SW2

其中：

值 4 个字节。卡片返回的数值，是一个有符号长整数。

例 1： Decimal –4 = {FFh, FFh, FFh, FCh}

值			
MSB			LSB
FFh	FFh	FFh	FCh

例 2： Decimal 1 = {00h, 00h, 00h, 01h}

值			
MSB			LSB
00h	00h	00h	01h

Read Value Block 命令的响应状态码

结果	SW1 SW2	含义
成功	90 00h	操作成功完成。
错误	63 00h	操作失败。

9.2.7. Copy Value Block

Copy Value Block 命令用于将数值从一个值块复制到另外一个值块。

Copy Value Block 的 APDU 结构（7 个字节）

命令	CLA	INS	P1	P2	Lc	命令数据域	
Value Block Operation	FFh	D7h	00h	源块号	02h	03h	目标块号

其中：

源块号 1 个字节。
源值块中的值会被复制到目标值块。

目标块号 1 个字节。
要恢复的值块。
源值块和目标值块必须位于同一个扇区。

Copy Value Block 的响应报文结构（2 个字节）

响应	响应数据域	
结果	SW1	SW2

Copy Value Block 命令的响应状态码

结果	SW1 SW2	含义
成功	90 00h	操作成功完成。
错误	63 00h	操作失败。

例如：

//将数值 1 存入值块 05h
APDU = {FF D7 00 05 05 00 00 00 00 01}

//从值块 05h 读取一个数值
APDU = {FF B1 00 05 04}

//将数值从值块 05h 复制到值块 06h
APDU = {FF D7 00 05 04 02 03 06}

//使值块 05h 的值增加 5
APDU = {FF D7 00 05 05 01 00 00 00 05}

9.2.8. 访问符合 PC/SC 标准的标签（ISO14443-4)

基本上，所有符合 ISO14443-4 标准的卡片（PICC 卡）都可以理解 ISO 7816-4 规定的 APDU。ACR35 读卡器与符合 ISO14443-4 标准的卡片进行通信时，只需要对 ISO 7816-4 规定的 APDU 和响应进行转换。ACR35 会在内部处理 ISO14443 第 1-4 部分协议。

MIFARE 1K、4K、Mini 和 Ultralight 标签是通过 T=CL 模拟进行支持的。

ISO 7816-4 规定的 APDU 报文结构

命令	CLA	INS	P1	P2	Lc	命令数据域	Le
ISO7816 第 4 部分规定的命令					命令数据域的长度		期望返回的响应数据的长度

ISO 7816-4 规定的响应报文结构（数据+2 个字节）

响应	响应数据域		
结果	响应数据	SW1	SW2

ISO 7816-4 响应状态码

结果	SW1 SW2	含义
成功	90 00h	操作成功完成。
错误	63 00h	操作失败。

典型的操作顺序为：

1. 出示标签，并连接 PICC 界面。
2. 读取/更新标签的存储内容。

需要：

1. 与标签建立连接。

   标签的 ATR 为 3B 88 80 01 00 00 00 00 33 81 81 00 3Ah.

   其中，

   ATQB 应用数据 = 00 00 00 00，ATQB 协议信息 = 33 81 81。这是一个 ISO 14443-4 Type B 标签。

2. 发送 APDU，取随机数

   << 00 84 00 00 08h

   >> 1A F7 F3 1B CD 2B A9 58h [90 00h]

   注：*对于 ISO 14443-4 Type A 标签来说，可以通过 APDU“FF CA 01 00 00h”来获取 ATS。*

例如：

//要从 ISO14443-4 B 类 PICC 中读取 8 个字节

APDU = {80 B2 80 00 08}

CLA	INS	P1	P2	Lc	命令数据域	Le
80h	B2h	80h	00h	无	无	08h

应答：00 01 02 03 04 05 06 07 [$9000]

9.2.9. 访问 FeliCa 标签

访问 FeliCa 标签的命令与访问 PC/SC 标签和 MIFARE 卡的命令有所不同。这些命令符合 FeliCa 规范，加了一个命令头。

FeliCa 命令格式

命令	CLA	INS	P1	P2	Lc	命令数据域
Felica Command	FFh	00h	00h	00h	命令数据域的长度	FeliCa 命令（开始于长度字节）

FeliCa 的响应结构（数据+2 个字节）

响应	响应数据域
结果	响应数据

以读取内存块为例：

1. 与 FeliCa 建立连接。

   ATR = 3B 8F 80 01 80 4F 0C A0 00 00 03 06 11 00 3B 00 00 00 00 42h

   其中，11 00 3Bh = FeliCa

2. 读取 FeliCa IDM。

   CMD = FF CA 00 00 00h

   RES = [IDM (8bytes)] 90 00h

   例如：FeliCa IDM = 01 01 06 01 CB 09 57 03h

3. FeliCa 命令访问。

   例如：“读取”内存块。

   CMD = FF 00 00 00 10 10 06 01 01 06 01 CB 09 57 03 01 09 01 01 80 00h

   其中：

   Felica 命令 = 10 06 01 01 06 01 CB 09 57 03 01 09 01 01 80 00h

   IDM = 01 01 06 01 CB 09 57 03h

RES = 内存块数据

10.0.敏感数据导入方法

本节将介绍敏感数据的导入方法，例如在更安全的环境中将客户主密钥、DUKPT 初始 PIN 加密密钥、AES 加密密钥和客户 ID 导入 ACR3x。

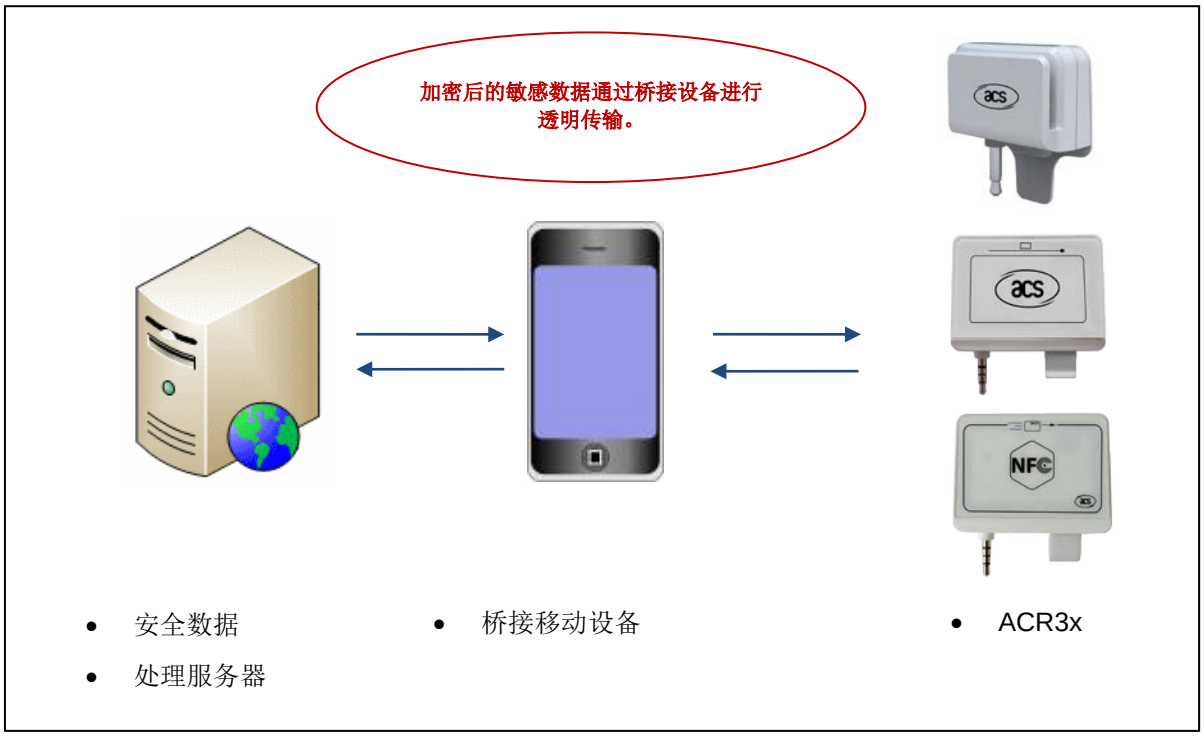

图4 ：敏感数据导入模式

上图涉及到三个实体，分别是安全数据处理服务器、桥接移动设备和 ACR3x。安全数据处理服务器负责接收和生成针对 ACR3x 的加密敏感数据，移动设备仅用作数据处理服务器和 ACR3x 之间的消息桥接通道。服务器和 ACR3x 之间的数据不需要在移动设备上进行处理（除非需要将数据重新打包成适用于通过音频通道向 ACR3x 发送的数据帧的情况）。

10.1. 认证

敏感数据被导入 ACR3x 之前，数据处理服务器（移动设备同时连接至服务器）必须通过 ACR3x 的认证才能获得许可来修改 ACR3x 中的机密数据。ACR3x 中用到了相互认证。

认证请求总是由数据处理服务器或桥接设备发起，然后会触发 ACR3x 返回一序列 16 字节的随机数（RND_A[0:15]）。随机数从 ACR3x 发出之前，要先采用 AES-128 CBC 加密模式通过当前存储在 ACR3x 中的客户主密钥进行加密。桥接设备必须将加密的随机数传递给数据处理服务器，之后服务器使用其内部正在使用的客户主密钥对数据进行 AES-128 CBC 加密模式解密（服务器中的客户主密钥应与 ACR3x 中的密钥相同，并且应由客户妥善保管）。此后，解密后的 16 字节 ACR3x 随机数会被填充到数据服务处理器生成的另外一个 16 字节随机数（RND_B[0:15]）的后面。最后形成 32 个字节的随机数（RND_C[0:31]），即：

RND_C[0:31] = RND_B[0:15] + RND_A[0:15],

此 32 字节随机数将通过服务器中使用的客户主密钥进行加密操作，最后输出的数据使用认证响应报文通过桥接设备发送给 ACR3x。

ACR3X 接收到认证响应报文后，报文数据使用自己的客户主密钥进行解密操作，转换成普通的 32 字节随机数。理论上讲，前面 16 个字节的随机数应当等于 RND_B[0:15]，是由数据处理服务器生成的，而另外 16 个字节应当等于 RND_A[0:15]，原本是由 ACR3x 生成的。

ACR3x 首先比较 RND_A[0:15]是否与原本的数据相同。若相同，则数据处理服务器通过了 ACR3x 的认证。然后 ACR3x 使用客户主密钥对获得的 RND_B[0:15]进行加密，通过桥接设备在认证响应应答报文中返回给数据处理服务器。

收到认证响应应答报文后，数据处理服务器会解密报文中的数据，检查 16 个字节的随机数是否与最初生成的 RND_B[0:15]完全相同。如果相同，则 ACR3x 通过了服务器的认证。此时，整个认证过程已经完成，敏感数据可以导入 ACR3x。

认证成功后，ACR3x 和数据处理服务器中都会生成一个 16 字节的过程密钥。此过程密钥（SK[0:15]）是将 RND_A 的前 8 个字节填充在 RND_B 的前 8 个字节后，也就是：

SK[0:15] = RND_B[0:7] + RND_A[0:7]

所有从安全数据处理服务器发出的敏感数据都要使用这个过程密钥以 AES-128 CBC 加密模式进行加密。这样即便是桥接的移动设备捕获到加密后的数据，在不清楚客户主密钥的情况下也很难获取原始的敏感数据。

为了更好的进行说明，请参考下图（图中省去了桥接设备，以便更简单明了的进行说明）：

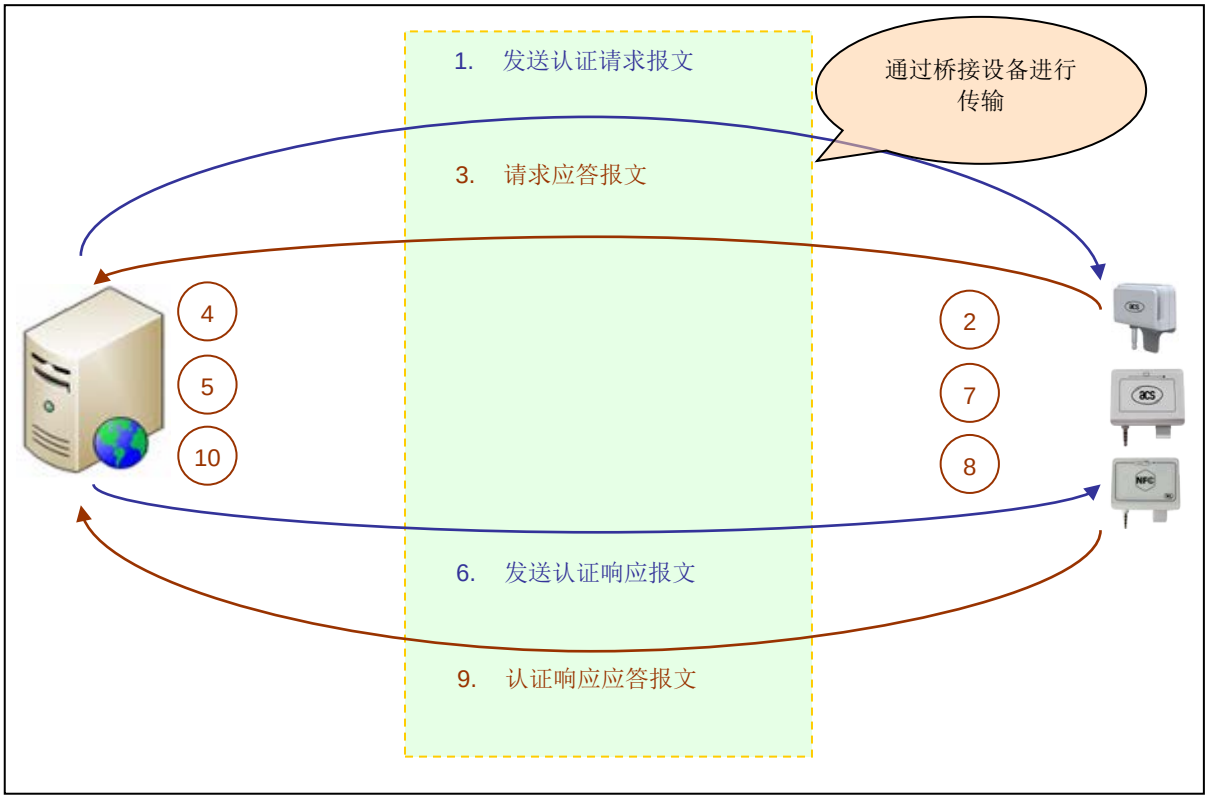

图5 ：认证步骤

上述各步骤汇总如下：

1. 数据处理服务器/桥接设备发出一条认证请求报文，向 ACR3x 发起认证请求。
2. 接收到认证请求报文后，ACR3x 会生成一个 16 字节的随机数（RND_A[0:15]）。这 16 字节的随机数要通过 ACR3x 当前使用的客户主密钥进行加密。
3. 加密后的 RND_A[0:15]通过认证响应应答报文发送至数据处理服务器。
4. 数据处理服务器对接收到的数据进行解密操作，还原出 RND_A[0:15]。
5. 数据处理服务器生成另外一个 16 字节的随机数（RND_B[0:15]）。RND_A[0:15]填充在 RND_B[0:15]的后面，形成一个 32 字节的随机数（RND_C[0:31] = RND_B[0:15] + RND_A[0:15]）。所有 32 字节的随机数通过服务器当前使用的客户主密钥进行加密处理。
6. 加密操作输出的最终数据通过认证响应报文传递给 ACR3x。
7. 之后在 ACR3x 中，收到的数据经过解密处理，还原出 32 字节的随机数。ACR3x 查看得到的 RND_A[0:15]是否与最初的随机数相同。若不同，则认证过程终止。
8. ACR3x 使用客户主密钥对得出的 RND_B[0:15]进行加密。与此同时，将 RND_A 的前 8 字节填充至 RND_B 的前 8 字节后方，得出 16 字节的过程密钥。
9. 加密后的 RND_B[0:15]将通过认证响应报文发送至数据处理服务器。
10. 数据处理服务器对收到的报文数据进行解密，然后比较解密后的数据是否与原始的 RND_B[0:15]相同。若不同，则认证过程终止。否则认证完成，将 RND_A 的前 8 字节填充至 RND_B 的前 8 字节后方，得出过程密钥。

10.2. 导入客户主密钥

在工厂生产过程中，ACR3x 的闪存应被重置为一些默认值。客户主密钥应全部置 0。

如需修改客户主密钥，数据处理服务器和 ACR3x 必须先使用旧的密钥进行认证。认证成功后，数据处理服务器可以向 ACR3x 发送设置主密钥的命令报文，报文中含有客户主密钥。新的客户主密钥必须使用当前过程密钥进行加密。新的客户主密钥成功载入 ACR3x 后，之前由 ACR3x 建立的认证将终止。进一步导入新的敏感数据之前，服务器要使用新的客户主密钥执行新认证请求。

10.3. 导入 AES 密钥

在工厂生产过程中，ACR3x 的闪存中会导入一个默认的 ACS AES 密钥。客户可以在认证成功后将此密钥更改为任意值。

若 DUKPT 被禁用，则此 AES 密钥可用于加密磁条卡磁道数据。新的 AES 密钥立即生效，不会对当前认证过程产生任何影响。

10.4. DUKPT 初始化

DUKPT 密钥管理算法正常工作之前，必须执行一些初始化过程。

首先，数据处理服务器必须向 ACR31 提供 10 字节的初始密钥序列号（IKSN）和 16 字节的初始 PIN 加密密钥（IPEK）。DUKPT 密钥管理引擎将使用这两组数据初始化未来的密钥表及其它设置。DUKPT 引擎的加密计数器会自动重置。

DUKPT 初始化完成后，DUKPT 选项启用，用于加密磁条卡磁道数据的密钥将由 DUKPT 算法生成。每次成功刷卡后，都会从 DUKPT 请求提供唯一的加密密钥，而不是在每次交易时使用固定的 AES 密钥。

需要注意的是，如果刷卡数据出现错误，则不会从 DUKPT 引擎请求密钥。另外在响应报文中，磁道数据会全部置为 0，只有错误代码显示检测到的卡片数据错误类型。刷卡失败后不从 DUKPT 请求密钥是为了让移动设备应用提示用户重新刷卡，而不会将无用的数据发送给后台服务器，另外还维持与服务器更同步的加密计数器。

11.0.卡片数据加密

每次刷卡都会向移动设备自动发送一个响应报文。报文中封装的磁道数据要使用 AES-128 CBC 加密模式进行加密（初始向量为 16 字节的 0）。

如果启用了 DUKPT 选项，则每次成功刷卡后，用于进行磁道数据加密的密钥将由 DUKPT 密钥管理算法生成。这样每次成功的交易都使用不同的密钥进行磁道数据加密。

若禁用了 DUKPT，则会使用 AES 密钥进行磁道数据加密。您可以使用 **10.0 节**介绍的敏感数据导入方法来修改 AES 密钥。ACR3x 在出货时会预先导入一个默认的 AES 密钥。默认的 AES 密钥为：

4E 61 74 68 61 6E 2E 4C 69 20 54 65 64 79 20h

要注意的是，如果在刷卡时出现数据错误，则磁道数据字段会填充为 0，只在报文中报告错误代码。

12.0.AES-128 CBC 加密测试向量

下表为 ACR3x 所使用的 AES-128 CBC 模式加密提供了几个测试向量。

原始数据：	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00h
初始向量：	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00h
密钥：	10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00h
加密结果：	6B 1E 2F FF E8 A1 14 00 9D 8F E2 2F 6D B5 F8 76h
原始数据：	69 88 44 21 13 84 0A 10 00 0C 02 22 11 88 00 0E 12 84 00 B1 40 80 80 11 31 02 45 20 20 28 E4 00h
初始向量：	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00h
密钥：	10 29 02 14 03 90 53 09 12 08 20 20 02 C0 9A 80h
加密结果：	EF 14 C9 C9 F3 48 96 5B 18 36 0A 2F 81 1A 93 C7 E2 FF F3 61 04 B8 D4 5E 13 F7 26 FE 2A 94 2B 69h
原始数据：	80 83 11 13 09 D1 11 30 0E 00 0A 49 04 00 26 99 C0 58 D1 7A 45 CD 17 10 30 00 22 08 10 4C 41 51h
初始向量：	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00h
密钥：	9A 04 2A 10 21 00 06 20 10 84 20 01 00 00 22 1Ch
加密结果：	56 85 B9 6B A1 B2 09 AB 58 71 58 B5 E0 30 42 71 64 62 51 FA 55 94 52 BC 78 33 24 FB 15 F5 33 62h
原始数据：	41 01 06 02 21 A8 C4 40 08 00 44 11 11 88 0D 09 10 81 92 10 01 20 20 2E 20 C4 05 81 58 08 18 86h
初始向量：	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00h
密钥：	30 13 30 C8 91 53 49 44 0E 29 98 42 84 17 00 D0h
加密结果：	ED 0F 2E BC 7D EA 58 C4 AB E8 72 91 87 74 2F C3 B1 8B 66 4F F5 E5 3F 8B BD A9 63 40 F8 0D 11 97h

13.0.TDES ECB 加密测试向量

下表为 ACR3x 所使用的 TDES ECB 模式加密提供了几个测试向量。

原始数据：	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00h
密钥：	10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00h
加密结果：	49 F9 E7 A6 0C 40 6D BF 49 F9 E7 A6 0C 40 6D BFh
原始数据：	02 50 88 82 22 21 13 C4 42 00 08 44 60 24 8A 04h
密钥：	00 C0 08 28 8E 28 16 10 01 80 50 4D 72 00 28 88h
加密结果：	8E BF 16 AA B4 59 AA C0 13 DB 32 E5 1D 04 BD 66h
原始数据：	61 10 88 19 42 31 01 26 42 02 74 24 00 07 0C 82h
密钥：	00 10 80 42 09 20 13 24 82 22 24 89 62 08 09 90h
加密结果：	74 57 DF 51 3B 04 7A F2 2B 26 C4 BF 81 6B 4D 58h

附录A. 磁道数据错误代码

Bit 7 MSB	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0 LSB
MSB	0	0	0	0	LRC 错误	结束码错误	起始码错误

注：

1. *b7-b1 位为错误代码*
2. *无误 = 0*

附录B. 系统错误代码

下表列举了 ACR3x 所有的系统错误代码及其相应说明。

错误代码	状态
00h	ERROR_SUCCESS
FFh	ERROR_INVALID_CMD
FEh	ERROR_INVALID_PARAM
FDh	ERROR_INVALID_CHECKSUM
FCh	ERROR_INVALID_STARTBYTE
FBh	ERROR_UNKNOWN
FAh	ECODE_DUKPT_CEASE_OPERATION
F9h	ECODE_DUKPT_DATA_CORRUPTED
F8h	ECODE_FLASH_DATA_CORRTPTED
F7h	ECODE_VERIFICATION_FAILED

表6 ： 系统错误代码