华大公用事业非接触智能卡

用户参考手册

编　　号：　**HED-ICC-13.7.2.01-TD044_090818**
支持算法：　**DES & SM1**
COS 版本：　**MCOS 3.6.0.1**

北京中电华大电子设计有限责任公司

二 OO 九年八月

文件修改履历

序号	日期	修改范围	新版本号	批准人	修改单号
1	2009-8-19	初稿	V1.0	刘宏梅	

目录

声明

本文档的版权属北京中电华大电子设计有限责任公司所有。任何未经授权对本文档进行复印、印刷、出版发行的行为，都将被视为是对北京中电华大电子设计有限责任公司版权的侵害。北京中电华大电子设计有限责任公司保留对此行为诉诸法律的权力。

北京中电华大电子设计有限责任公司保留未经通知用户对本手册内容进行修改的权利。

1 编写目的

华大公用事业非接触智能卡配合华大自主开发的 MCOS，支持金融环境和建设应用环境，适用于金融领域和建设领域中的应用；通过此用户参考手册可以帮助用户了解华大公用事业非接触智能卡的性能，熟悉使用华大公用事业非接触智能卡，配合应用的开发。

2 适用范围

此用户参考手册适用于利用华大公用事业非接触智能卡进行应用设计与开发的人员使用。

3 参考资料

- ISO 7816-1: Identification cards - Integrated circuit(s) cards with contacts – Physical characteristics-1987/07/01
- ISO 7816-2: Identification cards - Integrated circuit(s) cards with contacts – Dimensions and location of the contacts-1998/05/15
- ISO 7816-3: Identification cards - Integrated circuit(s) cards with contacts – Electronic signals and transmission protocols-1989
- ISO 7816-4: Identification cards - Integrated circuit(s) cards with contacts – Inter-industry commands for interchange-1994/07/08
- ISO 7816-5: Identification cards - Integrated circuit(s) cards with contacts – Numbering system and registration procedure for application identifiers-1992/09/24
- ISO 7816-6: Identification cards - Integrated circuit(s) cards with contacts – Inter-industry data elements-1994/07/08
- EMV’96 Integrated Circuit Card Specification for Payment System
- EMV’96 Integrated Circuit Card Application Specification for Payment System
- JR/T0025.1—2005《中国金融集成电路（IC）卡规范－第 1 部分：电子钱包/电子存折卡片规范》（PBOC2.0）
- JR/T0025.2—2005《中国金融集成电路（IC）卡规范－第 2 部分：电子钱包/电子存折应用规范》（PBOC2.0）
- JR/T0025.2—2005《中国金融集成电路（IC）卡规范－第 8 部分：与应用无关的非接触式规范》（PBOC2.0）
- JR/T0025.2—2005《中国金融集成电路（IC）卡规范－第 9 部分：电子钱包扩展应用指南》（PBOC2.0）
- 《建设事业非接触式 CPU 卡芯片技术要求》（报批稿）
- 《建设事业 CPU 卡操作系统技术要求》（报批稿）

4 定义

接口设备 Interface Device

终端上插入IC卡的部分，包括其中的机械和电气部分。

终端 Terminal

为完成交易而在交易点安装的设备，用于同IC卡的连接。

命令 Command

终端向IC卡发出的一条信息，该信息启动一个操作或一个应答。

响应 Response

IC卡处理完成收到的命令报文后，返回给终端的报文。

报文 Message

由终端向卡或卡向终端发出的，不含传输控制字符的字节串。

明文 Plaintext

没有加密的信息。

密文 Ciphertext

通过密码系统产生的不可理解的文字或信号。

密钥 Key

控制加密转换操作的符号序列。

加密算法 Cryptographic Algorithm

为了隐藏或揭露信息内容而变换数据的算法。

认证机构 Certification Authority

利用公开密钥和其他相关数据为所有者提供可靠校验的第三方机构。

对称加密技术 Symmetric Cryptographic Technique

发送方和接收方使用相同保密密钥进行数据变换的加密技术。

DES 算法

DES是一个对称算法，加密和解密用的是同一算法。DES的安全性依赖于所用的密钥。

SM1 算法

SM1算法是以128位分组为单位进行运算的对称密钥算法，密钥长度为16字节。

保密密钥 Secret Key

对称加密技术中仅供指定实体所用的密钥。

数据完整性 Data Integrity

数据不受未经许可的方法变更或破坏的属性。

电子钱包 Electronic Purse

一种为方便持卡人进行小额消费而设计的IC卡应用，它支持圈存、消费等交易。除圈存外，使用电子钱包进行的其他交易均不记录明细，且均无需提交个人密码（PIN）。

电子存折 Electronic Deposit

一种为持卡人进行消费、取现等交易而设计的使用个人密码（PIN）保护的金融IC卡应用，它支持圈存、圈提、消费、取现等交易。

圈存 Load

持卡人将其在银行相应账户上的资金划转到电子存折或电子钱包中。

圈提 Unload

持卡人将其在电子存折中的部分或全部资金划回到其在银行的相应账户上。

5 缩略语与符号

ADF	应用数据文件（Application Definition File）
AEF	应用基本文件（Application Elementary File）
AID	应用标识符（Application Identifier）
An	字母数字型（Alphanumeric）
ans	字母数字及特殊字符型（Alphanumeric Special）
APDU	应用协议数据单元（Application Protocol Data Unit）
ATS	复位应答（Answer to Reset）
b	二进制（Binary）
CLA	命令类别（Chip Card Payment Service）
CLK	时钟（Clock）
cn	压缩数字（Compressed Numeric）
DDF	目录数据文件（Directory Definition File）
DF	专用文件（Dedicated File）
DIR	目录（Directory）
EF	基本文件（Elementary File）
FCI	文件控制信息（File Control Information）
f	频率（Frequency）
GND	地(Ground)
IFS	信息域（Information Field）
INS	命令报文的指令字节（Instruction Byte of Command Message）
I/O	输入/输出（Input/Output）
Lc	终端发出的命令数据的实际长度（Exatct Length of Data Sent）
Le	响应数据中的最大期望长度（Maximum Length of Data Expected）
MAC	报文鉴别代码（Message Authentication Code）
MF	主控文件（Mater File）
N	数字型（Numeric）
O	可选型（Optional）
P1	参数 1（Parameter 1）
P2	参数 2（Parameter 2）
P3	参数 3（Parameter 3）
PCD	接近式耦合设备（读写器）(Proximity Coupling Device)
PICC	接近式卡(Proximity Card)
PIN	个人密码（Personal Identification Number）
RFU	保留为将来所用（Reserved for Future Use）
SW1	状态码 1（Status Word One）
SW2	状态码 2（Status Word Two）

TLV	标签、长度、值（Tag Length Value）
TAC	交易验证码（Transaction Authorization Cryptogram）
VCC	电源电压（Supply Voltage）
V_{PP}	Vpp 触点上的测量电压（Programming Voltage Message VCC Contact）

6 内容概述

本手册各部分内容包括：

- MCOS/PBOC/SSC 简介

  介绍华大公用事业非接触智能卡以及 MCOS/PBOC/SSC 的特性和所支持的文件结构特点。

- 安全管理

  描述了安全管理的基本概念，安全管理的实现方法，以及使用安全报文时命令的传送情况。

- 命令与响应

  详细介绍了华大公用事业非接触智能卡支持的各种基本命令和专有命令的使用要求，以及命令执行的返回信息。

- 卡片个人化

  简单介绍了，在应用中进行卡片个人化的流程。

- 交易流程

  介绍华大公用事业非接触智能卡所支持的各种交易流程。

7 产品介绍

7.1 特性

华大公用事业非接触智能卡是华大自主开发，拥有自主产权的一款高性能 CPU 卡，配合自主开发的高安全性的 CPU 卡操作系统 MCOS。

华大公用事业非接触智能卡具有以下特性：

- 符合规范《建设事业 CPU 卡操作系统技术要求》
- 符合 JR/T0025《中国金融集成电路（IC）卡规范》（PBOC2.0）第 1、2、9 部分
- 支持电子钱包/存折应用、扩展电子钱包应用
- 符合射频 ISO/IEC 14443 TYPE A 通讯协议
- 用户空间为 8K
- 支持 DES、3DES 和 SM1 算法
- 具有文件系统管理功能
- 具有安全管理功能
- 支持数据镜像保护功能
- 支持快速个人化
- 携带 DES、SM1 算法协处理器
- 传输波特率 106kbps
- 载波频率为 13.56MHz
- 工作场强为 1.5A/m~7.5A/m
- 10 年以上数据保持时间，10 万次以上重复擦写
- 工作温度：-40℃~85℃

7.2 复位应答（ATS）

IC 卡上电检测到处于非接触通讯方式后，等待接收 REQA 或 WUPA 命令，如果接收到 REQA 或 WUPA 命令，则 IC 卡返回 ATQA。

接收到 ATQA 后，读卡器发出防冲突和选卡指令，当 IC 卡被选中时，IC 卡返回 SAK；读卡器接收到 SAK 后，如果 IC 卡支持 ISO/IEC 14443-4，则读卡器发送 RATS 命令给 IC 卡，IC 卡接收到 RATS 命令后，返回一个 ATS。

PICC向PCD发送的ATS，结构如下：

XX(‘05’~‘14’)	‘78’	‘00’	‘C0’	‘02’	历史字节（0~15字节）

XX：ATS数据长度。

‘78’：‘7’，TA1，TB1，TC1被发送；‘8’，FSCI，表明PICC可以接收的最大帧大小为256字节。

‘00’：接收和发送使用相同速率，仅支持106kb/s。

‘C0’：’C’为FWI，FWT = (256 x 16 / fc) x 2^{FWI}，’0’为SFGI，SFGT = (256 x 16 / fc) x 2^{SFGI}。

‘02’：NAD 不支持，CID 支持。

历史字节：用户可定制。

7.3 通讯协议

7.3.1 射频接口概述

卡进入射频工作场并获得足够能量后，首先要完成防冲突处理流程，然后进入应用处理流程，处理应用层的命令。

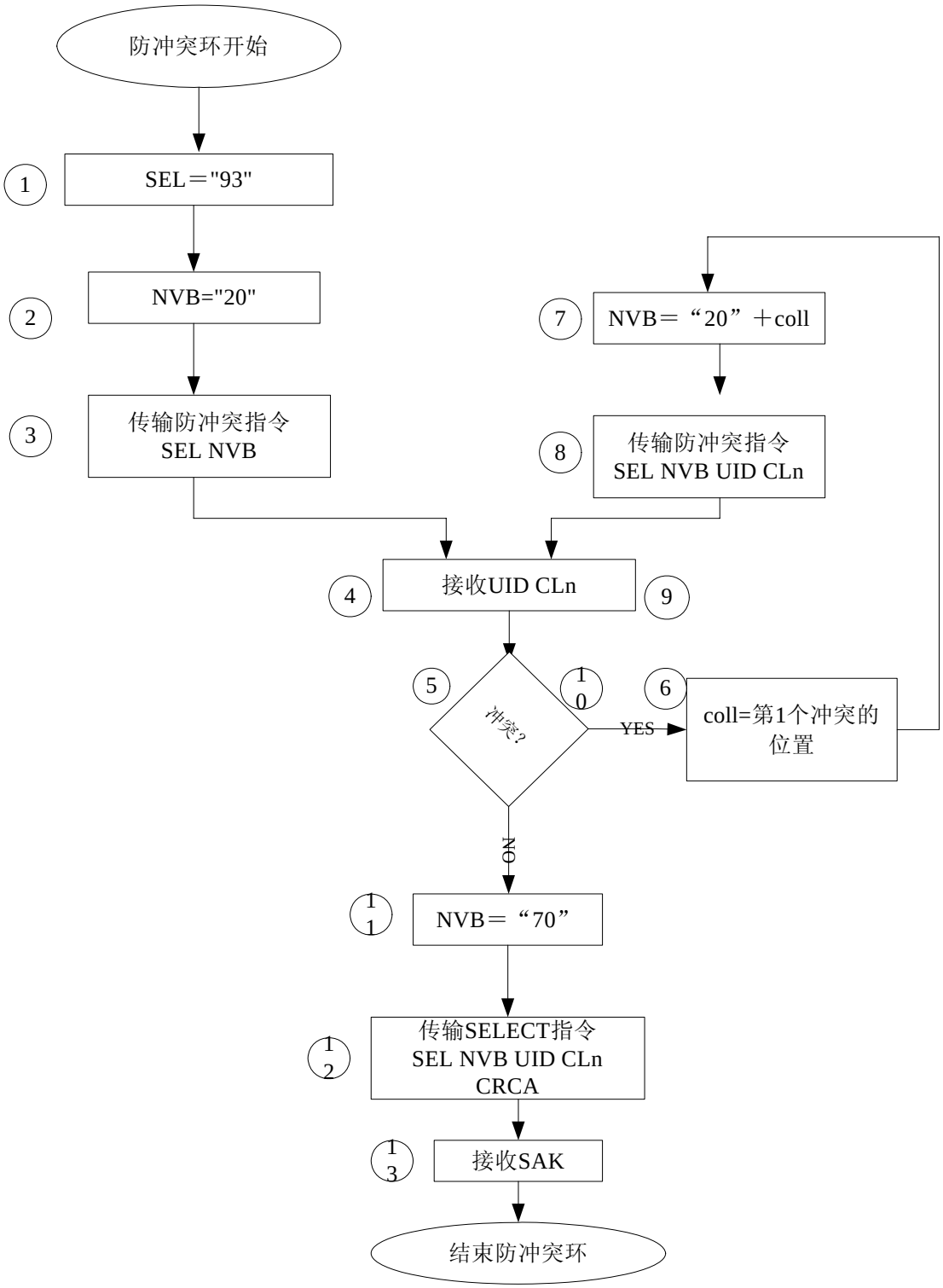
7.3.1.1 防冲突处理流程

卡感应到最小场强即进入 IDLE 状态，这时，卡被上电，只能响应 REQA/WUPA 命令。收到正确的 REQA/WUPA 命令后即进入 READY 状态，执行防冲突循环操作。

防冲突操作的步骤如下：

步骤 1：PCD 设置 SEL 为‘93’，即表示第一级防冲突。
步骤 2：PCD 设置 NVB 值为‘20’。
注：该值定义了该PCD将不发送UID CLn的任何部分。因此该命令迫使工作场内的所有PICC以其完整的UID CLn表示响应。
步骤 3：PCD 发送 SEL 和 NVB。
步骤 4：工作场内的所有 PICC 应使用它们的完整的 UID CLn 响应。
步骤 5：假设场内的 PICC 拥有唯一序列号，那么，如果一个以上的 PICC 响应，则冲突发生。如果没有冲突发生，则步骤 6 到步骤 10 可被跳过。
步骤 6：PCD 应识别出第一个冲突的位置。
步骤 7：PCD 分配了带有值的 NVB，该值规定了 UID CLn 有效 bit。这些有效位应是 PCD 在冲突出现之前所接收到的 UID CLn 的一部分再加上(0)b 或(1)b。典型的实现是增加(1)b。
步骤 8：PCD 发送 SEL 和 NVB，后随有效位。
步骤 9：只有 PICC 的 UID CLn 中的一部分等于 PCD 所发送的有效 bit 时，PICC 才应发送其 UID CLn 的其余部分。
步骤 10：如果出现进一步的冲突，则重复步骤 6~9，多重复的次数为 32 次。
步骤 11：如果不再出现冲突，则 PCD 设置 NVB 为‘70’。
注：该值定义了 PCD 将发送完整的 UID CLn。
步骤 12：PCD 发送 SEL 和 NVB，后面紧随 40bits 的 UID，和 CRC_A 校验和。
步骤 13：如果 PICC 的 UID CLn 与 40bits 匹配，则该 PICC 以其 SAK 表示响应。

下面是防冲突流程图：


防冲突环开始
1
SEL＝"93"
2
NVB="20"
3
传输防冲突指令
SEL NVB
7
NVB＝“20”＋coll
8
传输防冲突指令
SEL NVB UID CLn
4
接收UID CLn
9
5
10
冲突?
YES
6
coll=第1个冲突的
位置
NO
11
NVB＝“70”
12
传输SELECT指令
SEL NVB UID CLn
CRCA
13
接收SAK
结束防冲突环

7.3.1.2 应用层处理流程

PICC 防冲突完成进入 ACTIVE 状态，如果 PICC 返回的 SAK 表明 PICC 支持 14443-4 的功能，PCD 会发送 ATS 命令，PICC 在完成正常处理后，进入了应用层处理流程，此时，只响应应用层命令和 DESELECT 命令。

应用层处理流程中，卡与读卡器的交互由三种数据块组成，分别是 I-BLOCK、R-BLOCK、 S-BLOCK。其中：

I-BLOCK 用来传输应用层的信息。

R-BLOCK 用来传输命令的应答(ACK /NAK)。

S-BLOCK 用来交换控制信息，包括两种类型：

其一，延长等候时间，包括 1 字节 INF。

其二，DESELECT 命令。

PCD 与 PICC 之间的通讯有一定的规则，见下面：

交互过程中，第一块总是由 PCD 发出。

当 PICC 收到的 I-BLOCK 指示有后续块时，返回 R(ACK)块。

S-BLOCK 只能成对出现。卡可发 S-BLOCK(WTX)取代 I-BLOCK 或 R-BLOCK 应答,来请求延长等候时间。

如果收到 I-BLOCK（不管块号是多少），PICC 都应先翻转块号，再发送数据。

如果收到 R(ACK)，块号与 PICC 不同，则 PICC 应该先翻转块号，再发送块。

如果收到 I-BLOCK，且 I-BLOCK 表明不是链传输，则 PICC 应返回 I-BLOCK 应答。

如果收到 R(ACK)或 R(NAK)，块号与 PICC 的当前块号相等，则 PICC 上一次的块被重发。

如果收到 R(NAK)，块号与 PICC 的当前块号不同，则 PICC 返回 R(ACK)。

如果收到 R(ACK)，块号与 PICC 不同，且 PICC 处于链接状态，则 PICC 应继续发送链接块。

当通讯错误和协议错误发生时，PICC 不做任何恢复，返回到接收模式，等待 PCD 发送命令。

除链传输和 S（DESELECT）命令之外，当出现超时和无效数据块时，PCD 返回 R(NAK)。

链传输时，如果出现超时和无效数据块时，PCD 返回 R(ACK)。

当 S（DESELECT）命令没有返回时，PCD 可继续发 S（DESELECT）或忽略掉此 PICC。

如果收到 R(ACK)块时，块号与 PCD 当前块号不同，则 PCD 重发上一次的

I-BLOCK。

如果收到 R(ACK) 数据块时，块号与 PCD 当前块号相同，则 PCD 继续传输。

PCD 和 PICC 分别有自己当前的块号，规则如下：

对于 PCD: 初始块号为 0。

当收到的 R-BLOCK 或 I-BLOCK 为当前块号，则翻转块号。

对于 PICC：初识块号为 1。

当收到 I-BLOCK 时，翻转块号。

当收到 R-BLOCK(ACK)时，如果块号不为 PICC 当前块号，翻转块号。

7.3.2 射频通讯格式

7.3.2.1 防冲突命令格式

ANTICMD N Bytes	CRC 2 Bytes

ANTICMD：防冲突流程中的命令， $N > 0$。

7.3.2.2 防冲突命令返回数据格式

DATA	CRC_ B

DATA：响应数据。

7.3.2.3 应用层命令格式

协议头		命令应用数据单元	协议尾
PCB 1 Byte	CID 1 Byte（可选）	见表格 1	CRC 2 bytes

协议头		命令应用数据单元	协议尾
PCB 1 Byte	CID 1 Byte（可选）	见表格 1	CRC 2 bytes

表格 1 —— 命令应用数据单元

命令可以分为两种格式

格式	命令组成
2	CLA INS P1 P2 Le
3	CLA INS P1 P2 Lc Data

PCB： 协议控制字节
CID： 分配的 ID 号
CRC： 帧校验码。

7.3.2.4 应用层返回数据格式

协议头		响应数据单元	协议尾
PCB 1 Byte	CID 1 Byte（可选）	见表格 2	CRC 2 bytes

应答体	应答尾部	
响应数据体	SW1	SW2

表格 2 —— 响应应用数据单元

PCB： 协议控制字节
CID： 卡分配的识别号
CRC ： 帧校验码。

7.3.3 命令描述

7.3.3.1 REQA/ WUPA

7.3.3.1.1 功能说明

初始化卡片的通信，探询读卡器工作范围内的卡片，使其进入碰撞选卡状态，响应返回卡片的类型。

REQA：使处于 IDLE 状态的卡片进入 READY 状态并返回卡的类型号，编码为‘26’，PCD 通过该命令来探测场内是否有 TYPEA 类型的卡片存在。

WUPA：使处于 HALT/IDLE 状态的卡片进入 READY 状态并返回卡的类型号，编码为‘52’，用于后继的防冲突和选择进程。

7.3.3.1.2 命令格式

b7	b6	b5	b4	b3	b2	b1	Meaning
0	1	0	0	1	1	0	‘26’ = REQA
1	0	1	0	0	1	0	‘52’ = WUPA

响应数据为 ATQA。ATQA 编码规则如下：

MSB								LSB							
b16	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1
RFU				Proprietary coding				UID Size		RFU	Bit frame anticollision				

b08～b07：‘00’。

b05～b01：下表中的任何一种情况。

b05	b04	b03	b02	b01
1	0	0	0	0
0	1	0	0	0
0	0	1	0	0
0	0	0	1	0
0	0	0	0	1

所有的 RFU 位均为 0。

ATQA 固定为 0x01，表明支持“UID size: single，indicate bit frame anticollision”。

7.3.3.2 ANTICOLLISION/SELECT 命令

7.3.3.2.1 功能说明

这两个命令是在防碰撞的过程中使用，命令包括：

- 命令编码：SEL（1 字节）
- 命令参数：传输的有效位数 NVB（1 字节）
- 根据 NVB 确定的 0-40 位的 UID

根据 ISO/IEC 14443-3 中的规定，SEL 表示使用的碰撞帧的类型和 UID 的级联级别，由于只采用 40 位的 UID，所以 SEL 的编码为‘93’；

NVB 表示由 PCD 发送的命令有效位数，如果根据 NVB 不能确定 40 位的 UID，这时该命令叫做 ANTICOLLISION 命令，PICC 将返回未确定的 UID 位，且仍处于防碰撞的状态；如果根据 NVB 能够传输 40 位的 UID，则把该命令叫做 SELECT 命令，PICC 在接收到完整的 40 位 UID 后进入 AUTHENTICATION 状态，并返回 SAK。NVB 的长度为 1 个字节，高四位表示字节计数，其值由包括 SEL 和 NVB 在内的所有被 PCD 发送的有效位数除 8 取商得到；低四位表示位计数，其值由所有被 PCD 发送的有效位数模 8 得到。编码如下：

7.3.3.2.2 命令格式

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
1	0	0	1	0	0	1	1	‘93’: Select cascade level 1
1	0	0	1	0	1	0	1	‘95’: Select cascade level 2
1	0	0	1	0	1	1	1	‘97’: Select cascade level 3
1	0	0	1	other values except those here above				RFU

本 COS 仅支持’93’。

NVB 格式：

b8	b7	b6	b5	Meaning
0	0	1	0	Byte count = 2
0	0	1	1	Byte count = 3
0	1	0	0	Byte count = 4
0	1	0	1	Byte count = 5
0	1	1	0	Byte count = 6
0	1	1	1	Byte count = 7

b4	b3	b2	b1	Meaning
0	0	0	0	bit count = 0
0	0	0	1	bit count = 1
0	0	1	0	bit count = 2
0	0	1	1	bit count = 3
0	1	0	0	bit count = 4
0	1	0	1	bit count = 5
0	1	1	0	bit count = 6
0	1	1	1	bit count = 7

响应数据格式编码如下：

1^{st} byte		2^{nd}, 3^{rd} bytes	
SAK		CRC_A	
(1 byte)		(2 bytes)	
MSB	LSB	MSB	LSB

SAK 编码：

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
X	X	X	X	X	1	X	x	级联位被设置：UID没有结束
X	X	1	X	X	0	X	X	UID 结束，PICC兼容IISO/IEC 14443-4
X	X	0	X	X	0	X	X	UID 结束，PICC不兼容ISO/IEC 14443-4

SAK：0x20。

7.3.3.3 HALTA 命令

7.3.3.3.1 功能说明

使卡处于暂停状态，响应此命令后，卡只对 WUPA 命令有响应。

7.3.3.3.2 命令格式

‘50’	‘00’	CRC_A

响应数据：

无。

7.3.3.4 RATS 命令

7.3.3.4.1 功能说明

通过该指令，完成防冲突的卡片可以在后续与读卡器的通信中，应用 ISO/IEC14443-4 通信方式。

7.3.3.4.2 命令格式

Byte1	Byte2	Byte3	Byte4
E0	Parameter	CRC1	CRC2

Parameter

b8	b7	b6	b5	b4	b3	b2	b1
FSDI				CID			

FSDI：定义了 PCD 能够接收的最大帧能力。

CID：定义了目标 PICC 的逻辑号，编码为 0~14。

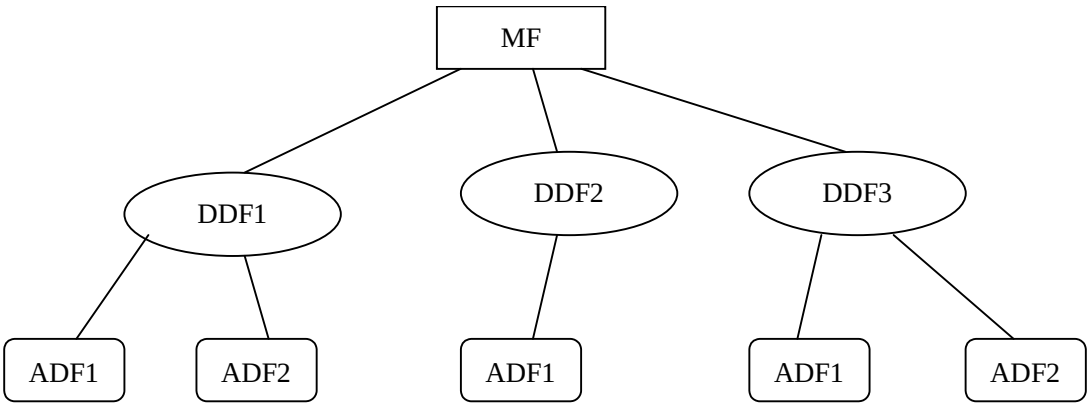
响应数据为 ATS。ATS 表明了 PICC 的能力，所以部分是固定的，部分是用户配置的。

数据项	值	说明
TL	0x05+K	
T0	0x78	TA、TB、TC 同时存在，FSC（卡支持的最大帧长度）为 256 字节
TA	0x00	仅仅支持 106Kb/s 的通信速率
TB	0xC0	高半字节为 FWI，低半字节为 SFGI。
TC	0x02	支持 CID，不支持 NAD
T_1~T_K	历史字节	历史字节由用户配置。长度 K 的范围为 0<=K<=9，使得 ATS 帧长度在 16 以内。

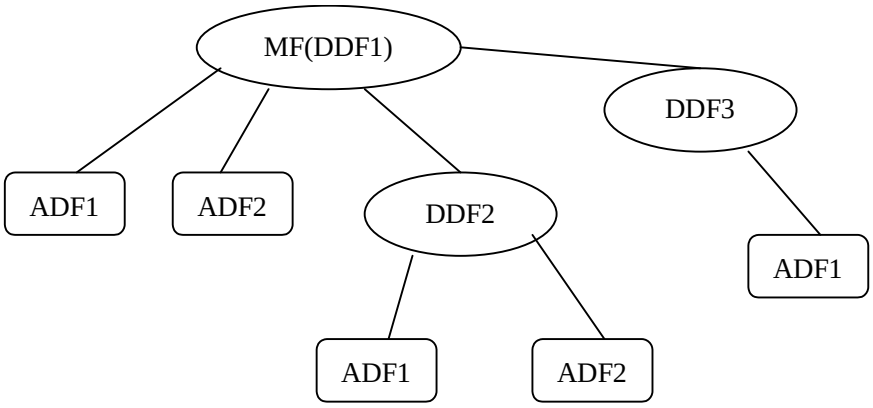
7.4 一般性说明

本卡所支持的应用系统的组织架构如下：

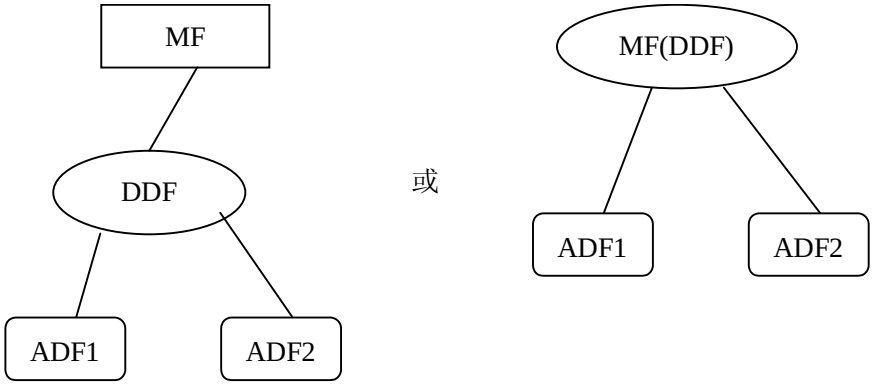
1．多应用平衡架构



2．多应用主次架构



3．单应用架构



7.4.1 文件系统概述

华大公用事业非接触智能卡允许在可用空间内建立自己的文件系统。支持多层目录结构。在同层目录中文件不能有相同的 ID（标识符）。

三种专用文件（DF）类型：

MF： 根目录，是整个文件系统的根，同属应用环境类。MF 下可以有 EF（基本文件）、SF（安全文件）、DDF(目录文件)、ADF（应用文件）；

DDF： 目录文件，用于定义一个应用环境，它是应用的集合。DDF 下可以有 ADF、子 DDF、EF 和 SF 等结构。

ADF： 应用文件，用于定义具体应用。ADF 下可以有 EF 和 SF。

四种基本文件（EF）类型：

透明文件： 文件数据是通过连续空间中的字节地址进行存取。

记录文件： 数据以固定长度格式记录在文件中，文件内最多可以容纳 255 条记录。以明文方式存取时，最大记录长度可以为 255 个字节。记录文件有以下几种形式：

1、线性定长记录文件： 同一文件内的所有记录都是等长度的。

2、线性变长记录文件： 同一文件内的每个记录的长度可以不相等。支持 TLV 和 V 两种记录格式。

3、循环定长记录文件： 同一文件内的所有记录都是等长度的。支持对文件中的记录循环存取。

三种交易文件类型：

交易文件：该类文件为特定格式文件。通过交易命令对这类文件进行操作。
交易文件有以下几种形式：

扩展电子钱包文件： 完成圈存、消费、复合消费等交易。执行圈存交易前须提交用户口令。

电子存折文件： 完成圈存、消费等交易。执行交易命令前须提交用户口令。适用于金融环境。

电子钱包文件： 完成圈存和消费交易。执行圈存交易前须提交用户口令。适用于金融环境。

安全文件类型（SF）：

安全文件：该文件只能写入不能读出。文件内可存放密钥或口令。

7.5 文件结构

下面定义了公用事业卡支持的文件类型。所有文件的标识符（FID）不能为‘0000’和‘FFFF’。

7.5.1 MF 文件

MF 是根目录，是整个文件系统的根，同属应用环境类。可从任何应用内进入根目录 MF。在执行 FREEZE MF 命令之前，卡处于应用开发状态，可以重新建立 MF，为开发应用提供了调试的途径。

MF 文件头信息中的主要参数说明如下：

File-ID（2 字节）：　文件标识符。固定为’3F00’

App-Type（1 字节）：　环境类型。基本命令的算法由高 4 字节来决定，交易命令的算法由密钥属性来决定。

高四位　‘0000’表示该环境采用 DES 算法
　　‘0010’表示该环境采用 SM1 算法

低四位　‘0000’表示该应用为金融应用
　　‘1000’表示该应用为建设事业应用

RFU 保留字节（1 字节）：　固定为’00’。

ATS-SFI（1 字节）：　ATS 文件的短文件标识符。卡上电复位后，ATS 文件的内容，作为复位应答信息（ATS）的历史字节被发送。最大 15 个字节，该文件的类型为透明文件。如 ATS-SFI =’00’时，表示卡内不设置 ATS 文件，卡会给出缺省的 ATS 内容。

DIR-SFI（1 字节）：　目录文件的短文件标识符。该文件的类型为线性记录文件。DIR-SFI =’00’时，表示 MF 下不设置目录文件。

FCI-SFI（1 字节）：　FCI 文件的短文件标识符。选择应用时，FCI 文件的内容作为 SELECT FILE 命令的响应数据从卡内送出。FCI 文件的类型为透明文件。FCI-SFI =’00’时，表示 MF 下不设置 FCI 文件。

ACw（1 字节）：　MK 控制属性，控制 MK 重装方式。
注：MK：MF 的主控密钥（卡片主控密钥/环境主控密钥）。短标识符为’00’（KID=’00’）。在 MF 下建立文件时，须认证此密钥。)

Write	

EPL	0	CER	CIPH	KT	0	0	0

bit7 EPL，当 EPL='1'时，电子钱包消费交易记录明细。
bit6 保留'0'
bit5 CER，校验码（MAC）。当 CER='1'时，WRITE KEY 命令的数据域中要附有校验码（MAC）。
bit4 CIPH，数据加密。当 CIPH='1'时，WRITE KEY 命令的数据域为密文。
bit3 KT，标识 MF-Key 的长度，KT='0'时 MK 的长度为 8 字节，KT='1'时 MK 的长度为 16 字节
bit2~bit0 保留'0'

RLD-KID（1 字节）：MK 的重装密钥的短标识符。重装 MK 时，加密数据和计算 MAC 所用密钥的短标识符。该密钥的用途为传输密钥或主控密钥。

BLK-KID（1 字节）：环境锁定密钥的短标识符。执行 CARD BLOCK 命令时，计算 MAC 所用密钥的短标识符。该密钥的用途为传输密钥或主控密钥。

Limit-Value（1 字节）：密钥认证限制数。MK 的尝试次数，最大可设置为 15 次，Limit-Value 减为 0 时，密钥被锁定。Limit-Value 设置为 0 值时，限制数为无限大。

MF-Name（1-16 字节）：MF 的应用名称（可选）

7.5.2 DDF 文件

DDF 文件用于创建一个应用环境。在该环境下可建立若干与应用环境相关的 ADF、EF 和 SF 等，也可建立 DDF。在 DDF 下，只能选择其范围内直接的 DDF、ADF、EF 和 SF 文件。

DDF 文件头信息中的主要参数说明如下：

File-ID（2 字节）：文件标识符。

LNG（2 字节）：文件体空间。

App-Type（1 字节）：环境类型。基本命令的算法由高 4 字节来决定，交易命令的算法由密钥属性来决定。
高四位 '0000'表示该环境采用 DES 算法。

‘0010’表示该环境采用 SM1 算法。

低四位 ‘0000’表示该应用为金融应用。

‘1000’表示该应用为建设事业应用。

RFU 保留字节（2 字节）：固定为’0000’。

DIR-SFI（1 字节）：目录文件的短文件标识符。该文件的类型为线性记录文件。DIR-SFI =’00’时，表示 DDF 下不设置目录文件。

FCI-SFI（1 字节）：FCI 文件的短文件标识符。选择 DDF 时，FCI 文件的内容作为 SELECT FILE 命令的响应数据从卡内送出。FCI 文件的类型为透明文件。FCI-SFI =’00’时，表示 DDF 下不设置 FCI 文件。

ACw（1 字节）：DDF 主控密钥控制属性，控制 DDF 主控密钥重装方式。
注：MK：DDF 的主控密钥（环境主控密钥）。短标识符为’00’（KID=’00’）。在 DDF 下建立文件时，须认证此密钥。

Write							
EPL	0	CER	CIPH	KT	KP1	0	KACT

bit7 EPL，1’钱包消费记明细，‘0’不记。

bit6 保留’0’。

bit5 CER，校验码（MAC）。当 CER=’1’时，WRITE KEY 命令的数据域中要附有校验码（MAC）。

bit4 CIPH，数据加密。当 CIPH=’1’时，WRITE KEY 命令的数据域为密文。

bit3 KT，标识 MK 的长度，KT=’0’时 MK 的长度为 8 字节，KT=’1’时 MK 的长度为 16 字节。

bit2 KP1，标识 RLD-KID 所指密钥的位置，KP1=’0’时，RLD-KID 所指密钥在当前 DDF 下，KP1=’1’时，RLD-KID 所指密钥在父目录下。

bit1 保留’0’。

bit0 MK 的有效性。

KACT=’0’表示不存在 MK。可通过父目录下的主控密钥按 ACw 定义的方式建立 MK。一旦建立 MK 成功 COS 将自动维护此位并置’1’。如果不建立 MK，当卡复位后，在该应用环境下不可建立任何形式的应用和文件。

KACT=’1’表示 MK 已经存在。缺省值为全’0’。密钥长度由 KT 设定。

RLD-KID（1 字节）：MK 的重装密钥的短标识符。重装 MK 时，加密数据和计算 MAC 所用密钥的短标识符。该密钥的用途为传输密钥或主控密钥。

BLK-KID（1 字节）：环境锁定密钥的短标识符。执行 CARD BLOCK 命令时，计算 MAC 所用密钥的短标识符。该密钥的用途为传输密钥或主控密钥。

Limit-Value（1 字节）：密钥认证限制数。MK 的尝试次数，最大可设置为 15 次，Limit-Value 减为 0 时，密钥被锁定。Limit-Value 设置为 0 值时，限制数为无限大。

DDF-Name（1-16 字节）：DDF 的应用名称（可选）。

7.5.3 ADF 文件

ADF 文件用于定义一个具体的应用。在该应用下可建立多个 EF 和 SF。

ADF 文件头信息中的主要参数说明如下：

File-ID（2 字节）：文件标识符。

LNG（2 字节）：文件体空间。

App-Type（1 字节）：应用类型。基本命令的算法由高 4 字节来决定，交易命令的算法由密钥属性来决定。

高四位	‘0000’表示该环境采用 DES 算法。 ‘0010’表示该环境采用 SM1 算法。
低四位	‘0000’表示该应用符合金融应用。 ‘1000’表示该应用符合建设事业应用。

RFU 保留字节（3 字节）：固定为’000000’。

FCI-SFI（1 字节）：FCI 文件的短文件标识符。选择 ADF 时，FCI 文件的内容作为 SELECT FILE 命令的响应数据从卡内送出。FCI 文件的类型为透明文件。FCI-SFI =’00’时，表示卡 ADF 下不设置 FCI 文件。

ACw（1 字节）：MK 控制属性，控制 MK 重装方式。
注：MK：ADF 的主控密钥（应用主控密钥）。短标识符为’00’（KID=’00’）。在 ADF 下建立文件时，须认证此密钥。

Write	

EPL	AV2	CER	CIPH	KT	KP1	KP2	KACT

bit7 EPL，‘1’钱包消费记明细，‘0’不记。

bit6 AV2，‘1’PBOC2.0 版本信息位于 FCI 专用信息之前，‘0’位于之后。

bit5 CER，校验码（MAC）。当 CER=’1’时，WRITE KEY 命令的数据域中要附有校验码（MAC）。

bit4 CIPH，数据加密。当 CIPH=’1’时，WRITE KEY 命令的数据域为密文。

bit3 KT，标识 MK 的长度，KT=’0’时 MK 的长度为 8 字节，KT=’1’时 MK 的长度为 16 字节。

bit2 KP1，标识 RLD-KID 所指密钥的位置，KP1=’0’时，RLD-KID 所指密钥在当前 ADF 下，KP1=’1’时，RLD-KID 所指密钥在父目录下。

bit1 KP2，标识 BLK-KID 所指密钥的位置，KP2=’0’时，BLK-KID 所指密钥在当前 ADF 下，KP2=’1’时，BLK-KID 所指密钥在父目录下。

bit0 MK 的有效性。

KACT=’0’表示不存在 MK。可通过父目录下的主控密钥按 ACw 定义的方式建立 MK，一旦建立 MK 成功 COS 将自动维护此位并置’1’。如果不建立 MK。当卡复位后，在该应用环境下不可建立任何形式的应用和文件。

KACT=’1’表示 MK 已经存在。缺省值为全’0’。密钥长度由 KT 设定。

RLD-KID（1 字节）：MK 的重装密钥的短标识符。重装 MK 时，加密数据和计算 MAC 所用密钥的短标识符。该密钥的用途为传输密钥或主控密钥。

BLK-KID（1 字节）：应用锁定密钥的短标识符。在 ADF 下，执行 APPLICATION BLOCK 命令时，计算 MAC 所用密钥的短标识符。该密钥的用途为传输密钥或主控密钥。

Limit-Value（1 字节）：密钥认证限制数。MK 的尝试次数，最大可设置为 15 次，Limit-Value 减为 0 时，密钥被锁定。Limit-Value 设置为 0 值时，限制数为无限大。

ADF-Name（1-16 字节）：ADF 的应用名称（可选）。

7.5.4 透明文件

透明文件的文件体是一个连续的区域，以字节为存取单元。

透明文件头信息中的主要参数说明如下：

File-ID（2 字节）：　文件标识符。以 SFI 访问时，文件标识符的低五位有效，只能访问到第一个文件。

LNG（2 字节）：　文件体空间。

RFU 保留字节（1 字节）：　固定为'00'。

ACr（1 字节）：　文件的读控制属性。

READ							
PMK	CMK	CER	CIPH	0	0	PINL	MPIN

bit7　PMK，认证当前环境主控密钥（MK）。PMK='1'时，在执行读命令前，必须通过当前环境（MF/DDF）主控密钥（MK）的认证。

bit6　CMK，认证当前应用主控密钥（MK）。CMK='1'时，执行读命令前，需要通过当前应用主控密钥（MK）的认证。在 MF/DDF 下执行读命令，该位无意义。

bit5　CER，校验码（MAC）。当 CER='1'时，读命令的数据域中要附有校验码（MAC）。

bit4　CIPH，数据加密。当 CIPH='1'时，从卡内读出的数据为密文。

bit3　保留'0'

bit2　保留'0'

bit1　PINL，PIN 权限和读权限的逻辑关系。（即 bit0 与 Read-Right 的逻辑关系）PINL='0'时，为'与'的关系；PINL='1'时，为'或'的关系。

bit0　MPIN，认证 PIN。MPIN='1'时，在执行读命令前，需要通过 PIN 的认证。

ACw（1 字节）：　文件的写控制属性。

UPDATE							
PMK	CMK	CER	CIPH	DISA	DISU	PINL	MPIN

bit7	PMK，认证当前环境主控密钥（MK）。PMK=‘1’时，在执行写命令前，必须通过当前环境（MF/DDF）主控密钥（MK）的认证。
bit6	CMK，认证当前应用主控密钥（MK）。CMK=‘1’时，执行写命令前，需要通过当前应用主控密钥（MK）的认证。在 MF/DDF 下执行读命令，该位无意义。
bit5	CER，校验码（MAC）。当 CER=’1’时，写命令的数据域中要附有校验码（MAC）。
bit4	CIPH，数据加密。当 CIPH=’1’时，写命令的数据域为密文。
bit3	DISA，禁止添加。DISA=’1’时，禁止向文件添加数据。
bit2	DISU，禁止修改。DISU=’1’时，禁止修改文件内的数据。
bit1	PINL，PIN 权限和写权限的逻辑关系。（即 bit0 与 Write-Right 的逻辑关系）PINL=’0’时，为‘与’的关系；PINL=’1'时，为‘或’的关系。
bit0	MPIN，认证 PIN。MPIN=’1’时，在执行写命令前，需要通过 PIN 的认证。

Read-Right（2 字节）：文件的读权限。和 ACr 一起控制文件的读操作。高字节为全局读权限，低字节为局部读权限。

Write-Right（2 字节）：文件的写权限。和 ACw 一起控制文件的写操作。高字节为全局写权限，低字节为局部写权限。

RT-KID（1 字节）：读密钥的短标识符。执行读命令时，加密数据和计算校验码（MAC）所用密钥的短标识符。该密钥的用途为传输密钥或主控密钥。

WT-KID（1 字节）：写密钥的短标识符。执行写命令时，加密数据和计算校验码（MAC）所用密钥的短标识符。该密钥的用途为传输密钥或主控密钥。

7.5.5 记录文件

记录文件中每条记录以固定长度记录数据。记录文件既可按记录号方式访问，也可按标签方式（TAG）访问。

7.5.5.1 线性定长记录文件

线性定长记录文件中的每条记录长度一致。

线性定长记录文件头信息中需要以下参数：

File-ID（2 字节）：　文件标识符。以 SFI 访问时，文件标识符的低五位有效，只能访问到第一个文件。

RL（1 字节）：　每条记录的长度。

RN（1 字节）：　文件中可容纳的最大记录个数。

RE（1 字节）：　预开记录个数。在建立文件时，用户可设定一初始值，其值小于等于 RN。预开记录后，文件中将存在相应数量的记录。RE=’00’表示文件内无记录存在。

RFU 保留字节（2 字节）：　固定为‘0000H’。

ACr（1 字节）：　文件的读控制属性。

READ							
PMK	CMK	CER	CIPH	0	0	PINL	MPIN

bit7　PMK，认证当前环境主控密钥（MK）。PMK=‘1’时，在执行读命令前，必须通过当前环境（MF/DDF）主控密钥（MK）的认证。

bit6　CMK，认证当前应用主控密钥（MK）。CMK=‘1’时，执行读命令前，需要通过当前应用主控密钥（MK）的认证。在 MF/DDF 下执行读命令，该位无意义。

bit5　CER，校验码（MAC）。当 CER=’1’时，读命令的数据域中要附有校验码（MAC）。

bit4　CIPH，数据加密。当 CIPH=‘1’时，从卡内读出的数据为密文。

bit3　保留’0’。

bit2　保留’0’。

bit1　PINL，PIN 权限和读权限的逻辑关系。（即 bit0 与 Read-Right 的逻辑关系）PINL=‘0’时，为‘与’的关系；PINL=‘1’时，为‘或’的关系。

bit0　　MPIN，认证 PIN。MPIN=‘1’时，在执行读命令前，需要通过 PIN 的认证。

ACw（1 字节）：　　文件的写控制属性。

UPDATE							
PMK	CMK	CER	CIPH	DISA	DISU	PINL	MPIN

bit7　　PMK，认证当前环境主控密钥（MK）。PMK=‘1’时，在执行写命令前，必须通过当前环境（MF/DDF）主控密钥（MK）的认证。

bit6　　CMK，认证当前应用主控密钥（MK）。CMK=‘1’时，执行写命令前，需要通过当前应用主控密钥（MK）的认证。在 MF/DDF 下执行写命令，该位无意义。

bit5　　CER，校验码（MAC）。当 CER=’1’时，写命令的数据域中要附有校验码（MAC）。

bit4　　CIPH，数据加密。当 CIPH=’1’时，写命令的数据域为密文。

bit3　　DISA，禁止添加。DISA=’1’时，禁止向文件添加数据。

bit2　　DISU，禁止修改。DISU=’1’时，禁止修改文件内的数据。

bit1　　PINL，PIN 权限和写权限的逻辑关系。（即 bit0 与 Write-Right 的逻辑关系）PINL=’0’时，为‘与’的关系；PINL=’1'时，为‘或’的关系。

bit0　　MPIN，认证 PIN。MPIN=’1’时，在执行写命令前，需要通过 PIN 的认证。

Read-Right（2 字节）：　　文件的读权限。和 ACr 一起控制文件的读操作。高字节为全局读权限，低字节为局部读权限。

Write-Right（2 字节）：　　文件的写权限。和 ACw 一起控制文件的写操作。高字节为全局写权限，低字节为局部写权限。

RT-KID（1 字节）：　　读密钥的短标识符。执行读命令时，加密数据和计算校验码（MAC）所用密钥的短标识符。该密钥的用途为传输密钥或主控密钥。

WT-KID（1 字节）：　　写密钥的短标识符。执行写命令时，加密数据和计算校验码（MAC）所用密钥的短标识符。该密钥的用途为传输密钥或

主控密钥。

7.5.5.2 线性变长记录文件

线性变长记录文件中的记录长度可以不一致。记录一旦建立添加，其长度不可改变。

线性变长记录文件头信息中需要以下参数：

File-ID（2 字节）：　文件标识符。以 SFI 访问时，文件标识符的低五位有效，只能访问到第一个文件。

LNG（2 字节）：　文件体空间。

RFU 保留字节（1 字节）：　固定为‘0000H’。

ACr（1 字节）：　文件的读控制属性。

READ							
PMK	CMK	CER	CIPH	0	0	PINL	MPIN

bit7	PMK，认证当前环境主控密钥（MK）。PMK=‘1’时，在执行读命令前，必须通过当前环境（MF/DDF）主控密钥（MK）的认证。
bit6	CMK，认证当前应用主控密钥（MK）。CMK=‘1’时，执行读命令前，需要通过当前应用主控密钥（MK）的认证。在 MF/DDF 下执行读命令，该位无意义。
bit5	CER，校验码（MAC）。当 CER=’1’时，读命令的数据域中要附有校验码（MAC）。
bit4	CIPH，数据加密。当 CIPH=‘1’时，从卡内读出的数据为密文。
bit3	保留’0’。
bit2	保留’0’。
bit1	PINL，PIN 权限和读权限的逻辑关系。（即 bit0 与 Read-Right 的逻辑关系）PINL=‘0’时，为‘与’的关系；PINL=‘1’时，为‘或’的关系。
bit0	MPIN，认证 PIN。MPIN=‘1’时，在执行读命令前，需要通过 PIN 的认证。

ACw（1 字节）：　文件的写控制属性。

UPDATE							
PMK	CMK	CER	CIPH	DISA	DISU	PINL	MPIN

bit7	PMK，认证当前环境主控密钥（MK）。PMK=‘1’时，在执行写命令前，必须通过当前环境（MF/DDF）主控密钥（MK）的认证。
bit6	CMK，认证当前应用主控密钥（MK）。CMK=‘1’时，执行写命令前，需要通过当前应用主控密钥（MK）的认证。在 MF/DDF 下执行写命令，该位无意义。
bit5	CER，校验码（MAC）。当 CER=’1’时，写命令的数据域中要附有校验码（MAC）。
bit4	CIPH，数据加密。当 CIPH=’1’时，写命令的数据域为密文。
bit3	DISA，禁止添加。DISA=’1’时，禁止向文件添加数据。
bit2	DISU，禁止修改。DISU=’1’时，禁止修改文件内的数据。
bit1	PINL，PIN 权限和写权限的逻辑关系。（即 bit0 与 Write-Right 的逻辑关系）PINL=’0’时，为‘与’的关系；PINL=’1'时，为‘或’的关系。
bit0	MPIN，认证 PIN。MPIN=’1’时，在执行写命令前，需要通过 PIN 的认证。

Read-Right（2 字节）：　文件的读权限。和 ACr 一起控制文件的读操作。高字节为全局读权限，低字节为局部读权限。

Write-Right（2 字节）：　文件的写权限。和 ACw 一起控制文件的写操作。高字节为全局写权限，低字节为局部写权限。

RT-KID（1 字节）：　读密钥的短标识符。执行读命令时，加密数据和计算校验码（MAC）所用密钥的短标识符。该密钥的用途为传输密钥或主控密钥。

WT-KID（1 字节）：　写密钥的短标识符。执行写命令时，加密数据和计算校验码（MAC）所用密钥的短标识符。该密钥的用途为传输密钥或主控密钥。

7.5.5.3 循环定长记录文件

循环定长记录文件中的每条记录长度一致。循环定长记录文件支持循环存取记录数据。

最新添加的一条记录，记录号为 1；上一条添加的记录， 记录号为 2；依此类推。

循环定长记录文件头信息中需要以下参数：

File-ID（2 字节）：　文件标识符。以 SFI 访问时，文件标识符的低五位有效，只能访问到第一个文件。

RL（1 字节）：　每条记录的长度。

RN（1 字节）：　文件中可容纳的最大记录个数。

RE（1 字节）：　预开记录个数。在建立文件时，用户可设定一初始值，其值小于等于 RN。预开记录后，文件中将存在相应数量的记录。RE=’00’表示文件内无记录存在。

RFU（1 字节）：　’00H’ 。

ACr（1 字节）：　文件的读控制属性。

READ							
PMK	CMK	CER	CIPH	0	0	PINL	MPIN

bit7　PMK，认证当前环境主控密钥（MK）。PMK=‘1’时，在执行读命令前，必须通过当前环境（MF/DDF）主控密钥（MK）的认证。

bit6　CMK，认证当前应用主控密钥（MK）。CMK=‘1’时，执行读命令前，需要通过当前应用主控密钥（MK）的认证。在 MF/DDF 下执行读命令，该位无意义。

bit5　CER，校验码（MAC）。当 CER=’1’时，读命令的数据域中要附有校验码（MAC）。

bit4　CIPH，数据加密。当 CIPH=‘1’时，从卡内读出的数据为密文。

bit3　保留’0’。

bit2　保留’0’。

bit1　PINL，PIN 权限和读权限的逻辑关系。
（即 bit0 与 Read-Right 的逻辑关系）
PINL=‘0’时，为‘与’的关系；PINL=‘1’时，为‘或’的关系。

bit0　MPIN，认证 PIN。MPIN=‘1’时，在执行读命令前，需要通过 PIN 的认证。

ACw（1 字节）：　文件的写控制属性

UPDATE							
PMK	CMK	CER	CIPH	DISA	DISU	PINL	MPIN

bit7　PMK，认证当前环境主控密钥（MK）。PMK=‘1’时，在执行写命令前，必须通过当前环境（MF/DDF）主控密钥（MK）的认证。

bit6　CMK，认证当前应用主控密钥（MK）。CMK=‘1’时，执行写命令前，需要通过当前应用主控密钥（MK）的认证。在 MF/DDF 下执行写命令，该位无意义。

bit5　CER，校验码（MAC）。当 CER=’1’时，写命令的数据域中要附有校验码（MAC）。

bit4　CIPH，数据加密。当 CIPH=’1’时，写命令的数据域为密文。

bit3　DISA，禁止添加。DISA=’1’时，禁止向文件添加数据。

bit2　DISU，禁止修改。DISU=’1’时，禁止修改文件内的数据。

bit1　PINL，PIN 权限和写权限的逻辑关系。
（即 bit0 与 Write-Right 的逻辑关系）
PINL=’0’时，为‘与’的关系；PINL=’1'时，为‘或’的关系。

bit0　MPIN，认证 PIN。MPIN=’1’时，在执行写命令前，需要通过 PIN 的认证。

Read-Right（2 字节）：　文件的读权限。和 ACr 一起控制文件的读操作。高字节为全局读权限，低字节为局部读权限。

Write-Right（2 字节）：　文件的写权限。和 ACw 一起控制文件的写操作。高字节为全局写权限，低字节为局部写权限。

RT-KID（1 字节）：　读密钥的短标识符。执行读命令时，加密数据和计算校验码

（MAC）所用密钥的短标识符。该密钥的用途为传输密钥或主控密钥。

WT-KID（1 字节）：　写密钥的短标识符。执行写命令时，加密数据和计算校验码（MAC）所用密钥的短标识符。该密钥的用途为传输密钥或主控密钥。

7.5.6 交易文件

交易文件是一种特殊的文件结构。文件体为固定长度。只有交易命令才能对其操作。

7.5.6.1 电子钱包(EP)文件

建立电子钱包(EP)文件时所用参数说明如下：

File-ID（2 字节）：　文件标识符。不能为‘0000H’和‘FFFFH’。

Bala-Limit（4 字节）：　余额上限。持卡人所能持有的最高金额值。

电子钱包文件是专用文件。只能建立在 ADF 下。文件体为固定长度。只有交易命令才能对其操作。

与电子钱包文件相关的 KEY 和文件有：

DPKep：　消费密钥。其标识符固定为‘02xxH’。其中‘xx’为密钥索引号。密钥长度 16 个字节。

DLKep：　圈存密钥。其标识符固定为‘09xxH’。其中‘xx’为密钥索引号。密钥长度 16 个字节。

DTKep：　交易认证密钥（TAK）。其标识符固定为‘0CxxH’。其中‘xx’为密钥索引号。密钥长度 16 个字节。

List：　交易记录明细文件。为循环定长记录文件，其标识符固定为‘0018H’。 文件体最小为记录长度 23 个字节，记录个数 10 条。

7.5.6.2 电子存折(ED)文件

建立电子存折(ED)文件时所用参数说明如下：

File-ID（2 字节）：　文件标识符。不能为‘0000H’和‘FFFFH’。

Bala-Limit（4 字节）：　余额上限。持卡人所能持有的最高金额值。

电子存折文件是专用文件。只能建立在 ADF 下。文件体为固定长度。只有交易命令才能对其操作。

与电子存折文件相关的 KEY 和文件有：

DPKed： 消费/取现密钥。其标识符固定为‘02xxH’。其中‘xx’为密钥索引号。密钥长度 16 个字节。

DLKed： 圈存密钥。其标识符固定为‘09xxH’。其中‘xx’为密钥索引号。密钥长度 16 个字节。

DULKed： 圈提密钥。其标识符固定为‘0AxxH’。其中‘xx’为密钥索引号。密钥长度 16 个字节。

DUKed： 修改透支限额密钥。其标识符固定为‘0BxxH’。其中‘xx’为密钥索引号。密钥长度 16 个字节。

DTKed： 交易认证密钥（TAK）。其标识符固定为‘0CxxH’。其中‘xx’为密钥索引号。密钥长度 16 个字节。

List： 交易记录明细文件。为循环定长记录文件，其标识符固定为‘0018H’。文件体最小为记录长度 23 个字节，记录个数 10 条。

7.5.6.3 扩展电子钱包文件

扩展电子钱包文件只能建立在 ADF 下。其大小固定为 75 个字节。支持消费、圈存、复合消费、灰锁、联机解扣操作，支持消费透支。同时需要注意的是，扩展电子钱包不可以与电子钱包同时存在于相同的 ADF 下。

建立扩展电子钱包文件时所用参数说明如下：

File-ID（2 字节）： 文件标识符。不能为‘0000H’和‘FFFFH’。

Bala-Limit（4 字节）： 余额上限。持卡人所能持有的最高金额值。

over-amount（4 字节）： 透支额度。

Load-List-SFI（1 字节）： 圈存交易明细 SFI。

扩展电子钱包文件是专用文件。只能建立在 ADF 下。文件体都是固定的。只有交易命令才能对其操作。

与电子钱包文件相关的 KEY 和文件有：

DPKed： 消费/取现密钥。其标识符固定为‘02xxH’。其中‘xx’为密钥索引号。密钥长度 16 个字节。

DLKed： 圈存密钥。其标识符固定为‘09xxH’。其中‘xx’为密钥索引号。密钥长度 16 个字节。

DULKed： 圈提密钥。其标识符固定为‘0AxxH’。其中‘xx’为密钥索引号。密钥长度 16 个字节。

DUKed： 修改透支限额密钥。其标识符固定为‘0BxxH’。其中‘xx’为密钥索引号。密钥长度 16 个字节。

DTKed： 交易认证密钥（TAK）。其标识符固定为‘0CxxH’。其中‘xx’为密钥索引号。密钥长度 16 个字节。

List： 交易记录明细文件。为循环定长记录文件，其标识符固定为‘0018H’。文件体最小为记录长度 23 个字节，记录个数 10 条。

复合消费专用文件： 复合消费专用文件。为变长记录文件，其标识符固定为‘0019H’。文件体最大为 256 字节。

7.5.7 安全文件（内部）

安全文件只能写入不能读出。一个安全文件可存有多个密钥和口令。密钥和口令可通过 WRITE KEY 命令写入卡内。

标识符为‘00’的密钥定义为主控密钥。一个 DF 下只能有一个主控密钥。主控密钥的建立是随 DF 一起建立的，可通过 WRITE KEY 命令更新主控密钥值。详细内容请参照 WRITE KEY 命令。

建立安全文件时所用参数说明如下：

File-ID（2 字节）： 文件标识符。

LNG（2 字节）： 文件体空间。

WT-KID（1 字节）： 重装密钥的短标识符。执行修改密钥命令时，加密数据和计算校验数据所用 KEY 的短标识符。该密钥的用途为传输密钥，或主控密钥。

ACw（1 字节）： 文件的写控制属性。

Reload				Append			
PMK	CMK	CER	CIPH	0	0	CER	CIPH

bit7 PMK，认证父目录主控密钥（MK）。PMK=‘1’时，在 ADF 下执行修改 KEY 命令前，需要通过父目录主控密钥（MK）的认证。如果在 MF/DDF 下执行修改 KEY 命令，则该位无意义。

bit6	CMK，认证当前主控密钥（MK）。CMK='1'时，执行修改 KEY 命令前，需要通过当前主控密钥（MK）的认证。
bit5	CER，校验码。当 CER='1'时，修改 KEY 命令的数据域中要附有校验码（MAC）数据。
bit4	CIPH，数据加密。当 CIPH='1'时，修改 KEY 命令的数据域为密文。
Bit3~bit2	保留'0'。
bit1	CER，校验码。当 CER='1'时，新建 KEY 命令的数据域中要附有校验码（MAC）数据。用于计算校验码的 KEY 固定为当前主控密钥。
bit0	CIPH，数据加密。当 CIPH='1'时，新建 KEY 命令的数据域为密文。用于加密数据的 KEY 固定为当前主控密钥。

Write-Right（2 字节）：密钥的修改权限。和 ACw 一起控制密钥的修改操作。高字节为全局修改密钥权限，低字节为局部修改密钥权限。详细说明参见"安全管理"一章。

密钥分类定义如下：

密钥类型	适用范围
类型 1	内部认证密钥、外部认证密钥
类型 2	传输密钥
类型 3	交易密钥
类型 4	PIN

密钥数据信息格式：

字节数 / 类型	1	1	1	1	2	1	1	8 或 16
类型 1	用途	标识	00	算法	Access-Right	Limit	SSB	Key-Data
类型 2	用途	标识	00	算法	Access-Right	Limit	00	Key-Data
类型 3	用途	索引	版本	算法	Access-Right	00	00	Key-Data
类型 4（主 PIN）	用途	00	UBK	RLK	Access-Right	Limit	00	PIN（8）
类型 4（非主 PIN）	用途	标识	UBK	RLK	Access-Right	Limit	SSB	PIN（8）

注：

1、 PIN 的有效长度 2~6 个字节。

2、对于类型 4 情况下的 PIN，数据信息格式中的 UBK 指的是解锁 PIN 密钥的标识符；RLK 指的是重装 PIN 密钥的标识符。

3、对于类型 4，PIN 为主 PIN 时，标识为'00' ，SSB 为'00'。

参数说明：

标识： 密钥的标识符。取值范围在'00H'—'7FH'之间。标识为'00H'的密钥定义为主控密钥（MK）。

用途： 密钥用途定义如下：

用途	说明	密钥类型	相关命令
00	外部认证密钥	1	外部认证
01	传输密钥	2	数据传输
02	消费密钥	3	交易命令
09	圈存密钥	3	交易命令
0A	圈提密钥	3	交易命令
0B	修改透支密钥	3	交易命令
0C	交易认证密钥（TAC）	3	交易命令
1C	内部认证密钥	1	内部认证
1F	口令	4	PIN 认证

MF、DDF、ADF 算法环境	
DES 算法	**SM1 算法**
DES 算法用途为 00 的密钥	SM1 算法用途为 00 的密钥
DES 算法用途为 01 的密钥	SM1 算法用途为 01 的密钥
DES 算法用途为 02 的密钥	SM1 算法用途为 02 的密钥
DES 算法用途为 09 的密钥	SM1 算法用途为 09 的密钥
DES 算法用途为 0A 的密钥	SM1 算法用途为 0A 的密钥
DES 算法用途为 0B 的密钥	SM1 算法用途为 0B 的密钥
DES 算法用途为 0C 的密钥	SM1 算法用途为 0C 的密钥
DES 算法用途为 1C 的密钥	SM1 算法用途为 1C 的密钥
SM1 算法用途为 02 的密钥	DES 算法用途为 02 的密钥
SM1 算法用途为 09 的密钥	DES 算法用途为 09 的密钥
SM1 算法用途为 0A 的密钥	DES 算法用途为 0A 的密钥
SM1 算法用途为 0B 的密钥	DES 算法用途为 0B 的密钥
SM1 算法用途为 0C 的密钥	DES 算法用途为 0C 的密钥

索引：　　密钥的引用序列号。

版本：　　密钥的版本序号。

算法：　　安全算法。’00’为 3DES 算法；’01’为单 DES 算法；‘03’SM1 算法。

Access-Right：　密钥的使用权限。高字节为全局使用权限，低字节为局部使用权限。详细说明参见“安全管理”一章。

Limit：　　密钥认证限制数。连续认证失败的次数，最大设置为 15 次，Limit 减为 0 时，密钥和 PIN 被锁定或应用被永久锁定。Limit 设置为 0 值时，认证限制数为无限大。

SSB：　　安全级别。详细说明参见“安全管理”一章。

Key-Data：　密钥数据。其有效长度为 8 字节或 16 字节。密钥的长度取决于加密算法。采用 3DES 算法的密钥长度为 16 字节，采用单 DES 算法的密钥长度为 8 字节。

PIN：　　口令数据，其有效长度为 2~6 字节。第一次新建口令时，PIN 值为 8 字节
其组成是：PIN 值=有效字节+填充数据’FF’（6~2 字节）。而更 PIN 时只输入有效值（2~6），不需要填充数据’FF’。
例如：PIN 值为‘1234’
新建口令：PIN 值=‘1234FFFFFFFFFFFF’
更新口令：PIN 值=‘1234’

UBK：　　解锁密钥的短标识符。执行口令解锁命令时，加密数据和计算校验数据所用 KEY 的短标识符。该密钥的用途为传输密钥，或主控密钥。

RLK：　　重装密钥的短标识符。执行口令重装命令时，加密数据和计算校验数据所用 KEY 的短标识符。该密钥的用途为传输密钥，或主控密钥。

8 安全管理

8.1 安全状态

安全状态是指卡当前所处的一种安全级别，卡的环境目录（DDF）和当前应用目录（ADF）分别具有 16 种不同的安全级别。在卡内用两个寄存器（16 位，每一位对应一个级别）表示整个环境的安全状态，称为全局安全状态字；两个寄存器（16 位，每一位对应一个级别）表示当前应用的安全状态，称为局部安全状态字。如果当前目录为 DDF 或 MF，局部安全状态无意义。四个寄存器的初始值为 0。

内存中安全状态字中的安全级别状态，是通过对 KEY/PIN 进行认证，认证成功后，将 KEY/PIN 记录中的安全级别字节（SSB）映射到相应的安全状态字来求得的，若密钥在 DDF 或 MF 下，则映射到全局安全状态字；若密钥在 ADF 下，则映射到局部安全状态字。SSB 的高 4 位表示安全级别区间的下限（1~15）， 低 4 位表示安全级别区间的上限（1~15），该字节值为‘XY’，表示认证成功后可以获得 X 至 Y 区间内的安全级别。映射的方法是：根据 SSB 指定的安全级别区间，对相应安全状态字中 X 至 Y 之间的所有位置 1。例如：SSB=‘46’，那么认证通过后，寄存器的第 4、5、6 位置‘1’；若 SSB=‘AD’，则认证通过后，寄存器的第 10、11、12、13 位置‘1’。

若 SSB 为‘00’，表示对安全状态没有影响。

全局安全状态在当前 DDF 或 MF 整个工作期间有效，直到卡被重新复位或选择新的 DDF。局部安全状态只在一个 ADF 下有效，当从一个 ADF 变换到另一个 ADF（包括重新选择当前的 ADF）时，局部安全状态的内容被复位。

8.2 文件访问权限

文件访问权限包括文件的 Read-Right 和 Write-Right。Right 定义了文件操作条件，2 个字节表示，高字节对应环境目录（DDF 或 MF）下的安全状态（或称全局安全状态），低字节对应当前 ADF 的安全状态（或称局部安全状态）。每个字节的高 4 位表示安全状态区间的下限（1~15）， 低 4 位表示安全状态区间的上限（1~15），该字节值为‘XY’， $X \le Y$ 表示应获得相应安全状态字中 X 至 Y 区间内的安全级别；该字节值为‘0Y’，相应操作不受限制；若 $X \ge Y$，表示相应操作被禁止。

综上所述，判别访问权限是否满足的方法是：

1. 如果 Right 字节为‘0X’格式，则满足访问条件；
2. 如果 Right 字节 $X > Y$，则无法满足访问条件，访问被禁止；
3. 如果 Right 字节 $X \le Y$，则检查安全状态字中 X 至 Y 区间内是否有‘1’存在，若有则满足访问条件；若无则不满足访问条件。

8.3 数据交换模式

在卡的操作权限得到满足后，还要使用文件指定的数据交换模式才能正确地读写数据。

终端与卡之间的数据交换有四种模式：数据可以是明文、密文、明文加校验和密文加校验。命令根据被存取对象的存取模式控制字 ACx 采用相应的数据交换模式（如“READ BINARY”要根据当前透明文件的存取条件信息 ACwr 中指定的数据交换模式将数据读出）。

针对密文、明文加校验和密文加校验的数据交换模式，数据必须由 8/16 个字节组成一个数据块， 并以数据块为单位对数据加密或产生校验码 MAC。

安全数据交换的目的是保证数据的可靠性、数据完整性和对发送方的认证。数据完整性和对发送方的认证通过使用校验码来实现。数据的可靠性通过对数据域的加密来得到保证。

安全报文传送格式符合 ISO 7816-4 的规定，当 CLA 字节的后半字节等于十六进制‘4’时，表明对发送方命令数据要采用安全报文传送。对基本文件操作的命令报文数据是否使用安全报文传送取决于对 ACx 中 CER 和 CIPH 的设置。当 CER 和 CIPH 中的任一项为‘1’时，表明使用安全报文传送。

8.3.1 明文模式

如果对数据传输的安全性、完整性以及对发送方的认证都没有要求，可以采用明文模式。数据交换中的明文模式就是命令报文的数据域中和响应报文的数据域中的数据不经过任何形式的变换处理直接传送。

8.3.2 加密模式

如果侧重于数据在传输中的安全性，可以采用加密模式。数据交换中的加密模式就是命令报文的数据域中和响应报文的数据域中的数据先经过加密变换，然后再放在相应的数据域中传送。数据是如何加密的，请看数据加密相关章节。

8.3.3 校验模式

如果侧重于数据在传输中的完整性和对数据发送方进行认证，可以采用校验模式。校验模式就是对命令报文的所有内容或响应报文的所有内容使用一个算法进行加密得到一个 4 字节的校验码（MAC），然后把它放在命令报文或响应报文的数据域中发送。有关校验码的计算，请看 MAC 计算相关章节。

8.3.4 加密校验模式

如果既要求数据在传输中的安全性又要求数据在传输中的完整性和对数据发送方进行认证，可以采用加密校验模式。加密校验模式就是首先对命令报文数据域中或响应报文数据域中的数据进行加密；接着把命令报文数据域中或响应报文数据域中的明文数据替换为加密数据，再对命令报文的所有内容或响应报文的所有内容使

用一个算法进行加密得到一个 4 字节的校验码（MAC）；最后把报文数据域中或响应报文数据域中的数据替换为加密数据，再把校验码紧接在加密数据之后发送。数据域中的数据是怎样被加密的以及命令头和加密后的数据或加密后的响应报文数据是怎样作为输入数据产生校验码（MAC）的，请看加密数据和 MAC 计算相关章节。

8.3.5 模式控制字 Acx

文件的读、修改和添加等操作采用的具体模式是由文件头中的模式控制字 ACx 决定的。并且一旦确定某个存取功能的存取模式，以后将无法修改。对 ACx 的定义参见“文件结构”一节。当 ACx 中的 CER=‘1’时，表示安全报文传送必须使用校验模式；当 ACx 中的 CIPH=‘1’时，表示安全报文传送必须使用加密模式。

8.4 安全计算

安全计算包括了华大公用事业非接触智能卡中涉及的各种安全算法。它们有：密钥分散计算，过程密钥计算、安全鉴别数据、校验码（MAC）的计算、数据加密和解密计算等。

校验码（MAC）总是命令或命令响应数据域中最后一个数据元素。规定 MAC 的长度为 4 个字节。当命令的数据域中要求带有 MAC 时，即命令安全报文传送，命令头中 CLA 字节的低半字节必须为十六进制数‘4’。

8.4.1 DES 算法在金融环境中的安全管理

8.4.1.1 密钥分散的计算方法

8.4.1.1.1 单倍长密钥的分散方法

密钥分散通过分散因子产生子密钥。

分散因子为 8 字节，将一个单倍长的主密钥MK，对分散数据进行处理，推导出一个单倍长的子密钥DK。方法是用主密钥MK对分散因子（8 字节）进行DEA计算产生子密钥DK。如图 8-1。

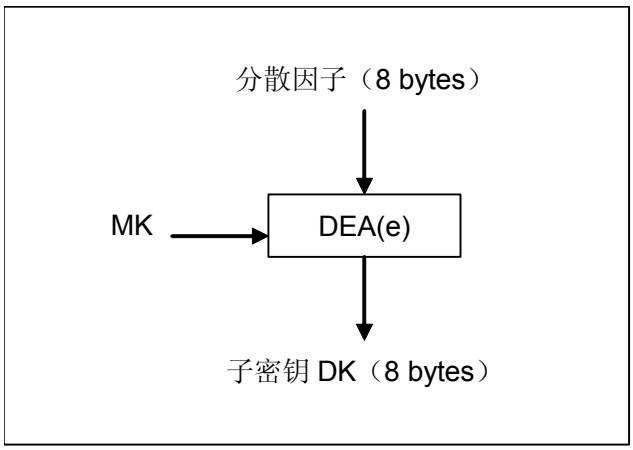


图 8-1 单倍长密钥的分散

8.4.1.1.2 双倍长密钥的分散方法

密钥分散通过分散因子产生子密钥。

分散因子为 8 字节，将一个双倍长的主密钥 MK，对分散数据进行处理，推导出一个双倍长的子密钥 DK（DK=DK_L+DK_R）。

推导 DK 左半部分 DK_L 的方法是：

第一步： 将分散因子作为输入数据；
第二步： 将 MK 作为加密密钥；
第三步： 用 MK 对输入数据进行 3DEA 运算。

推导 DK 右半部分 DK_R 的方法是：

第一步： 将分散因子求反，作为输入数据；
第二步： 将 MK 作为加密密钥；
第三步： 用 MK 对输入数据进行 3DEA 运算。

将左右两部分连接在一起，产生双倍长子密钥，如图 8-2。

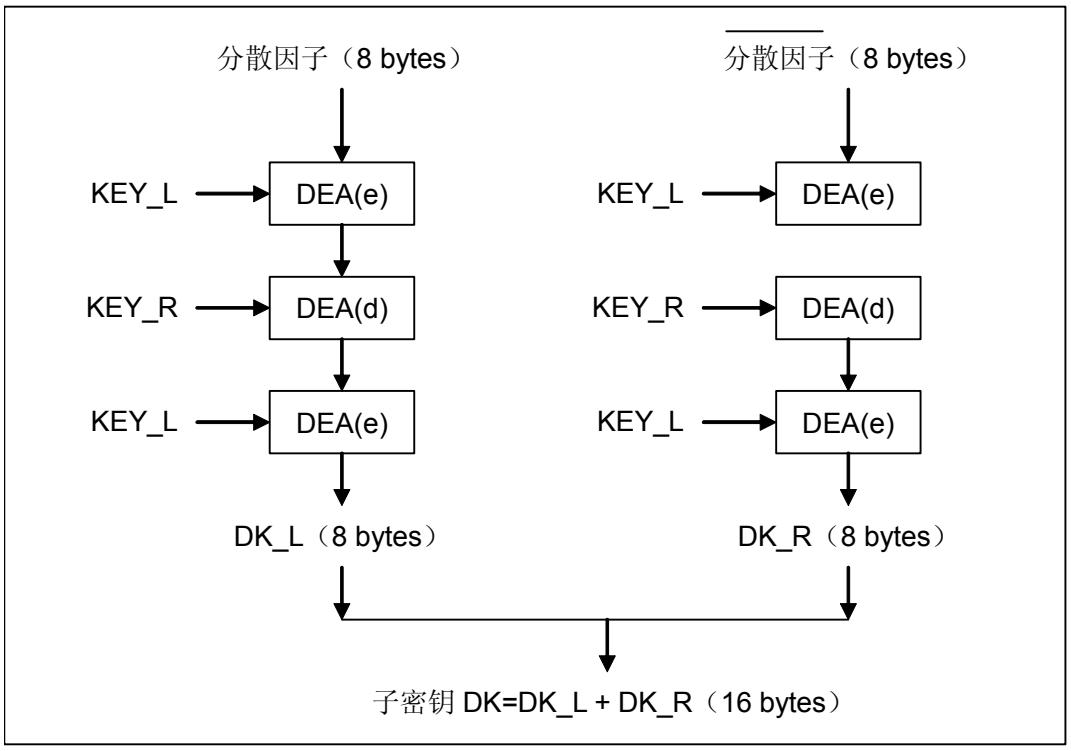


图 8-2 双倍长密钥的分散

8.4.1.2 过程密钥的计算方法

8.4.1.2.1 过程密钥的计算方法 1

该方法来源于PBOC。

该方法是通过指定密钥对过程密钥输入因子（8 字节）进行 3DEA或DEA计算产生过程密钥（8 字节）。如图 8-3和图 8-4。

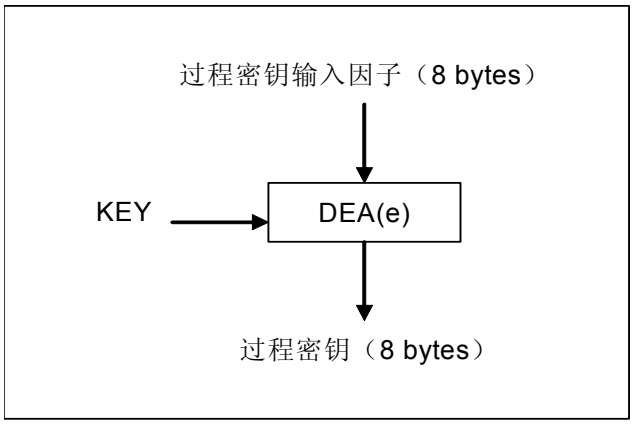


图 8-3 单倍长密钥产生过程密钥

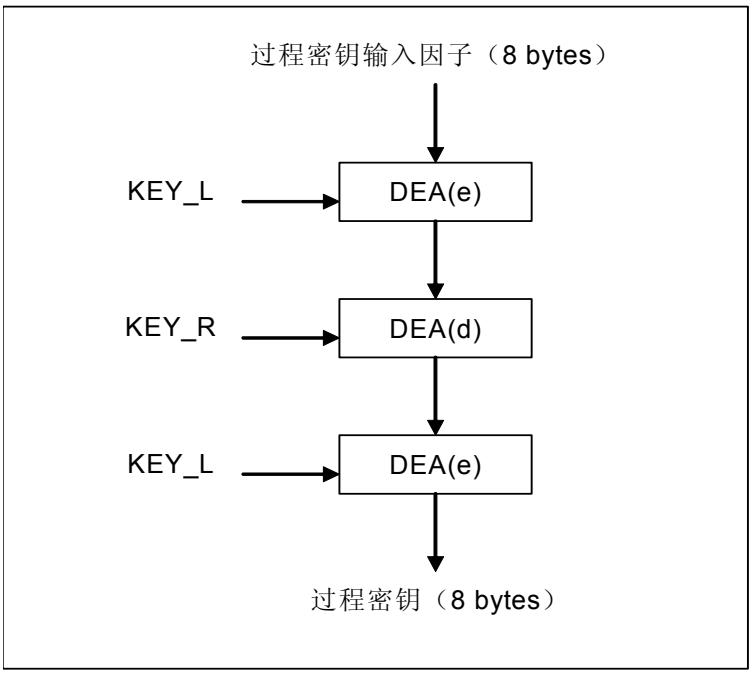


图 8-4 双倍长密钥产生过程密钥

8.4.1.2.2 过程密钥的计算方法 2

该方法来源于 PBOC 标准。

该方法是通过对指定的双倍长密钥进行左右异或计算来产生单倍长过程密钥。如图 8-5。

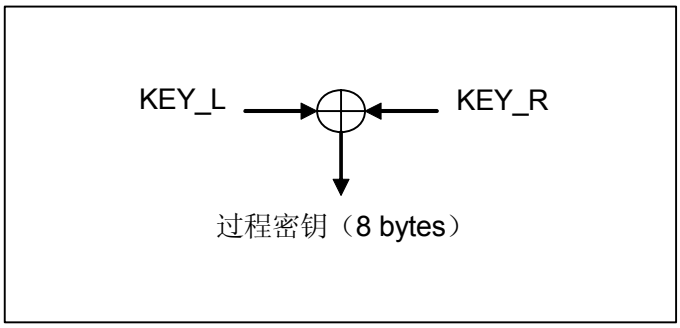


图 8-5 过程密钥产生

8.4.1.3 鉴别数据的计算方法

该方法来源于 PBOC 标准。

该方法是通过指定的密钥（单倍长或双倍长）对鉴别数据输入因子（8 字节）进行DEA计算产生鉴别数据（8 字节），供IC卡或接口设备进行验证。如图 8-6和图 8-7。

按照如下方式使用 DEA 加密方式产生鉴别数据：

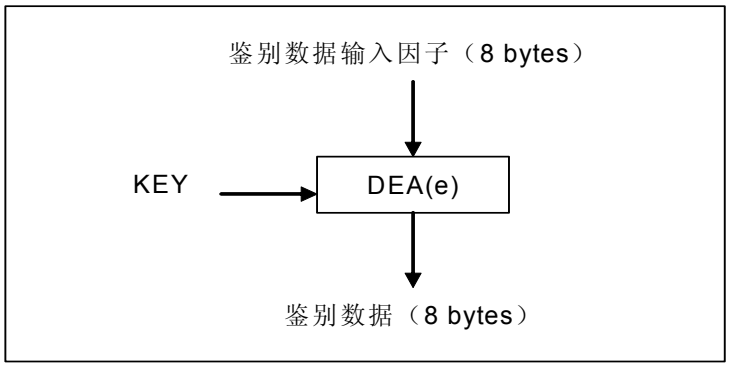


图 8-6 单倍长密钥的鉴别数据的计算

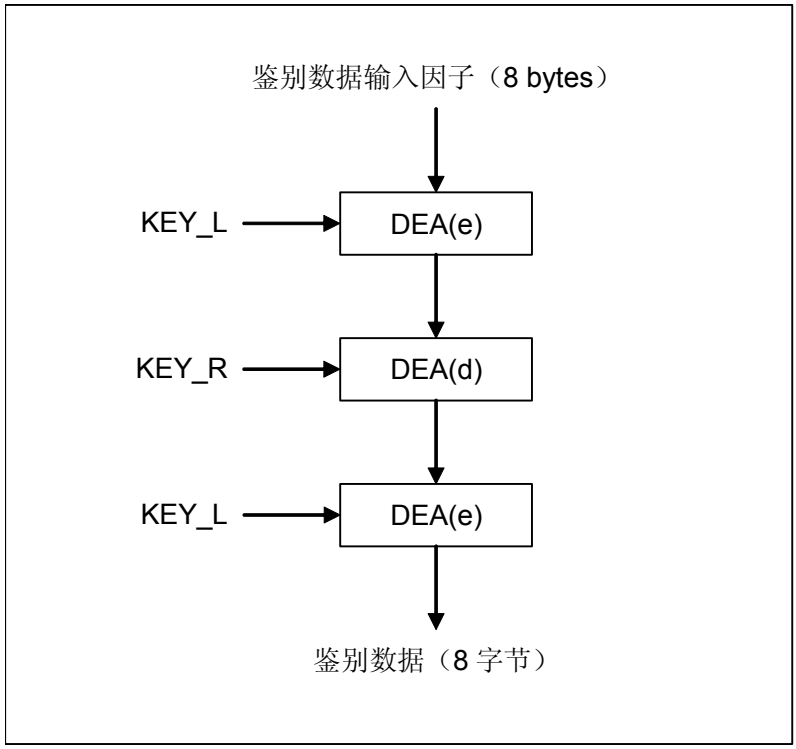


图 8-7 双倍长密钥的鉴别数据的计算

8.4.1.4 MAC 的计算方法

8.4.1.4.1 命令安全报文中的 MAC

该方法来源于 PBOC 标准。

命令安全报文中的 MAC 是使用命令的所有元素（包括命令头和命令数据域中的数据）来产生的。以保证命令连同数据能够正确完整地传送，并对发送方进行认证。

按照如下方式使用 DEA 加密方式产生 MAC：

第一步： 终端通过向 IC 卡发 GET CHALLENGE 命令获得一个 4 字节随机数，后补’00 00 00 00’作为初始值；或取一个 8 字节随机数作为初始值。

第二步： 将 5 字节命令头（CLA，INS，P1，P2，Lc）和命令数据域中的明文或密文数据连接在一起形成数据块。注意，这里的 Lc 应是数据长度加上将计算出的 MAC 的长度（4 字节）后得到的实际长度。

第三步： 将该数据块分成 8 字节为单位的数据块， 表示为 BLOCK1、BLOCK2、BLOCK3、BLOCK4 等。最后的数据块有可能是 1~8 个字节。

第四步： 如果最后的数据块的长度是 8 字节的话，则在该数据块之后再加一个完整的 8 字节数据块‘80 00 00 00 00 00 00 00’， 转到第五步。

如果最后的数据块的长度不足 8 字节，则在其后加入 16 进制数‘80’，如果达到 8 字节长度，则转到第五步；否则接着在其后加入 16 进制数‘00’

直到长度达到 8 字节。

第五步： 按照图 8-8和图 8-9所述的算法对这些数据块使用指定密钥进行加密来产生MAC。

第六步： 最终取计算结果（高 4 字节）作为 MAC。

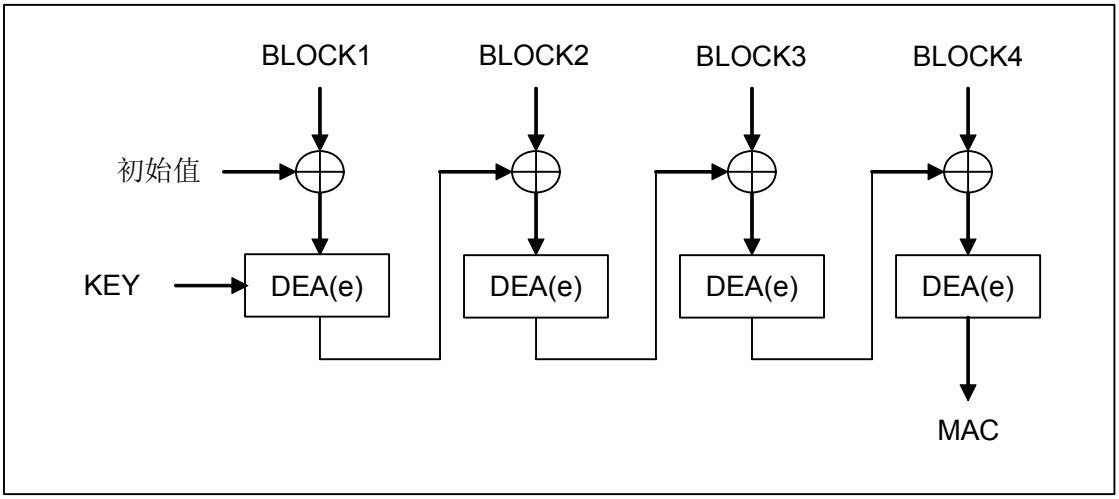


图 8-8 安全报文中单倍长密钥 MAC 计算

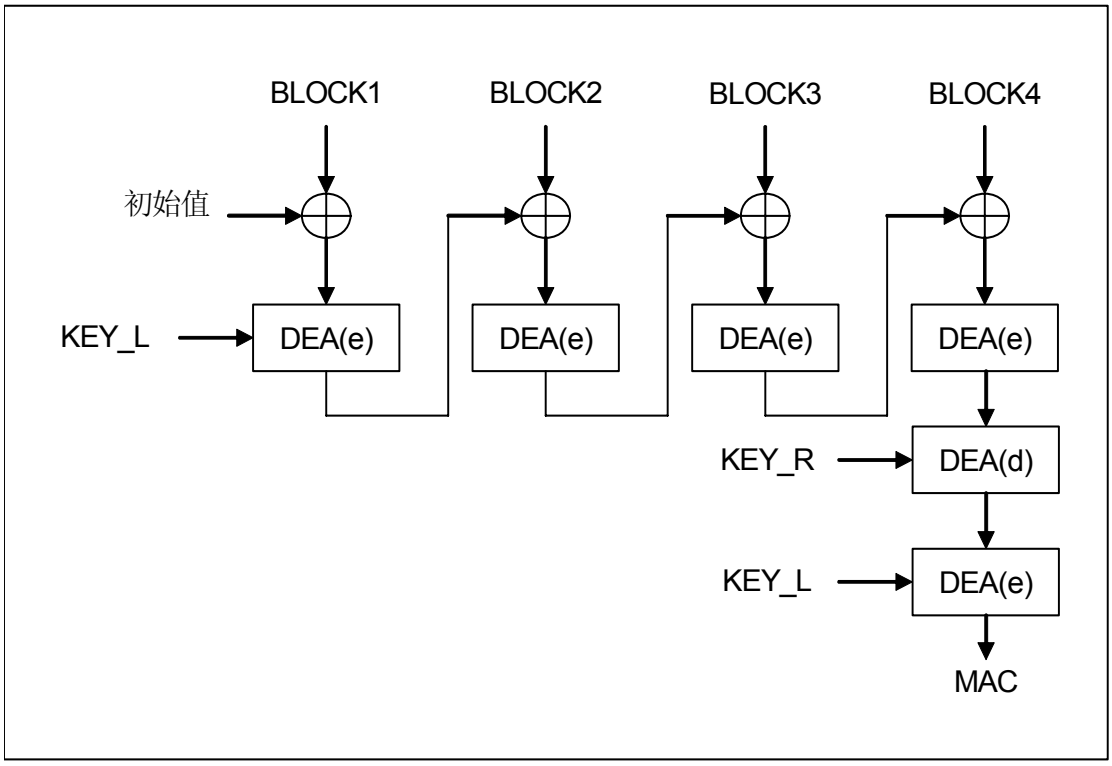


图 8-9 安全报文中双倍长密钥 MAC 算法

8.4.1.4.2 交易中的 MAC/TAC/GMAC /GTAC

交易中的 MAC/TAC/GMAC/GTAC 计算使用此方法。计算方法分二步完成。先用指定密钥产生过程密钥（计算 MAC 时请参看 3.4.1.1.2.1 节过程密钥计算；计算 TAC 时请参看 3.4.1.1.2.2 节过程密钥计算），再用过程密钥计算 MAC/TAC/GMAC/GTAC。

ED/EP 交易中的 MAC/TAC/GMAC/GTAC 是使用不同交易指定的数据元序列来产生的。从而保证交易的安全性。按照如下方式使用过程密钥 DEA 算法产生 MAC/TAC/GMAC/GTAC：

第一步： 将一个 8 字节长的初始值设定为 16 进制数‘00 00 00 00 00 00 00 00’。
第二步： 将所有输入数据按指定顺序连接成一个数据块。
第三步： 将该数据块分成 8 字节为单位的数据块，表示为 BLOCK1、BLOCK2、BLOCK3、BLOCK4 等。最后的数据块有可能是 1~8 个字节。
第四步： 如果最后的数据块的长度是 8 字节的话，则在该数据块之后再加一个完整的 8 字节数据块‘80 00 00 00 00 00 00 00’，转到第五步。
如果最后的数据块的长度不足 8 字节，则在其后加入 16 进制数‘80’，如果达到 8 字节长度，则转到第五步；否则在其后加入 16 进制数‘00’直到长度达到 8 字节。
第五步： 按照图 8-10所述的算法对这些数据块使用过程密钥（单倍长度）进行加密来产生MAC/TAC/GMAC/GTAC。
第六步： 最终取计算结果（高 4 字节）作为 MAC/TAC/GMAC/GTAC。

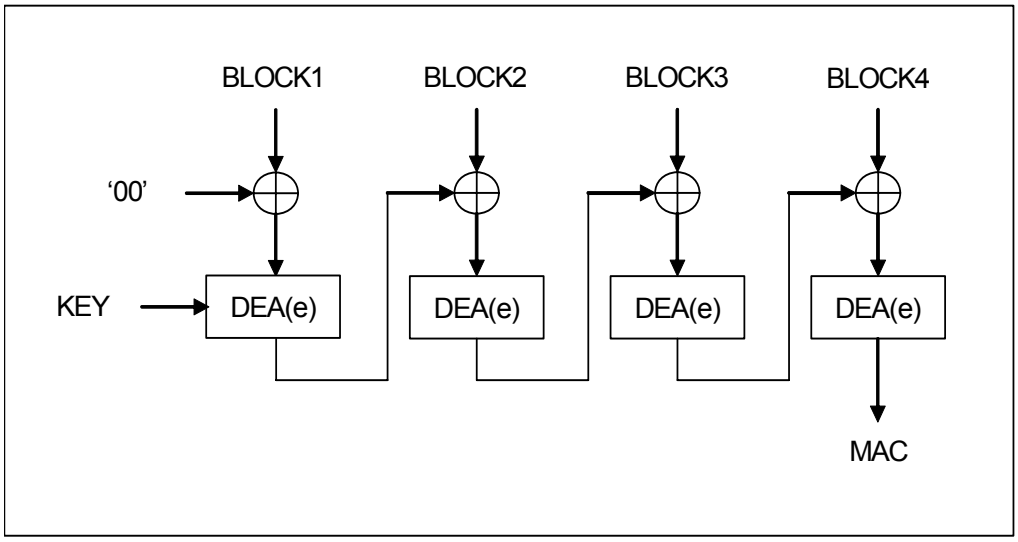


图 8-10 ED/EP 交易中的 MAC/TAC/GMAC/GTAC 算法

8.4.1.5 数据加密的计算方法

第一步： 用 LD（1 字节）表示明文数据的长度，在明文数据前加上 LD 产生新的数据块。

第二步： 将该数据块分成 8 字节为单位的数据块，表示为 BLOCK1、BLOCK2、BLOCK3、BLOCK4 等。最后的数据块有可能是 1~8 个字节。

第三步： 如果最后（或唯一）的数据块的长度是 8 字节的话，转到第四步；如果不足 8 字节，则在其后加入 16 进制数‘80’，如果达到 8 字节长度，则转到第四步；否则在其后加入 16 进制数‘00’直到长度达到 8 字节。

第四步： 按照图 8-11和图 8-12所述的算法使用指定密钥对每一个数据块进行加密。

第五步： 计算结束后，所有加密后的数据块依照原顺序连接在一起。

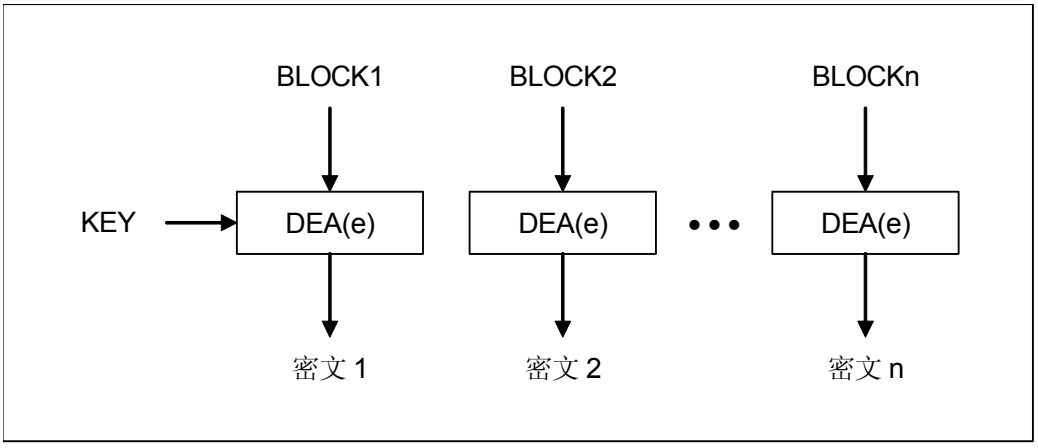


图 8-11 单倍长密钥 DEA 数据加密算法

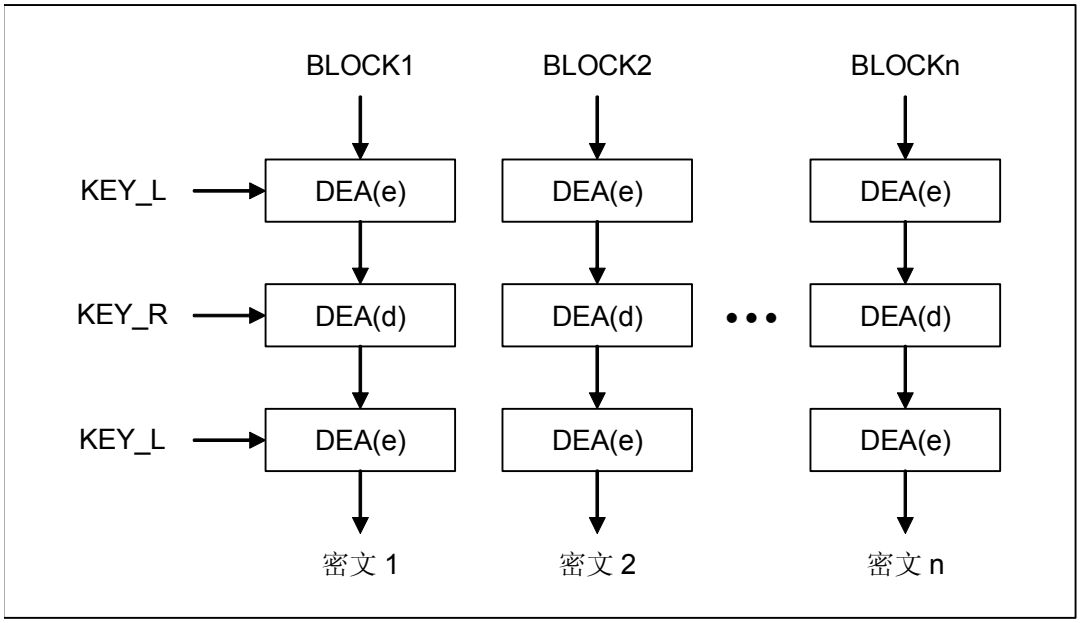


图 8-12 双倍长密钥 DEA 数据加密算法

8.4.1.6 数据解密的计算方法

数据解密则采用相反的过程，如图 8-13和图 8-14。

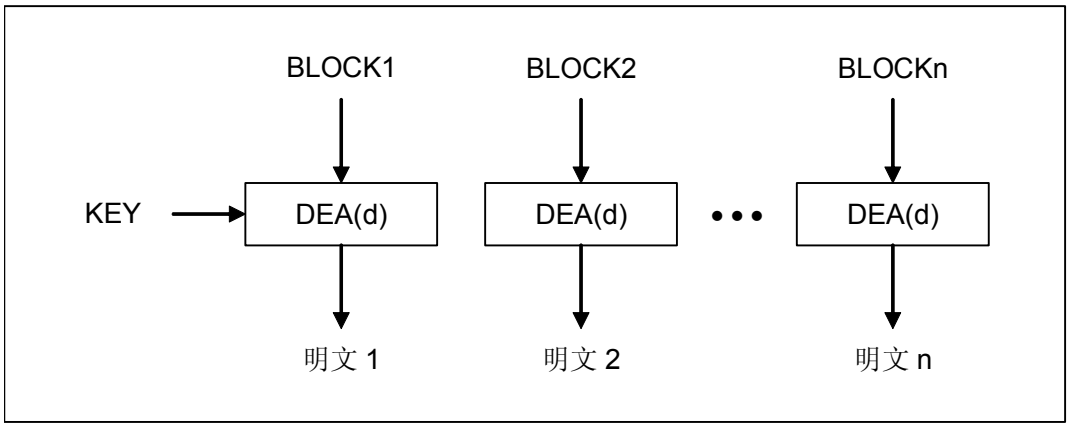


图 8-13 单倍长密钥 DEA 数据解密算法

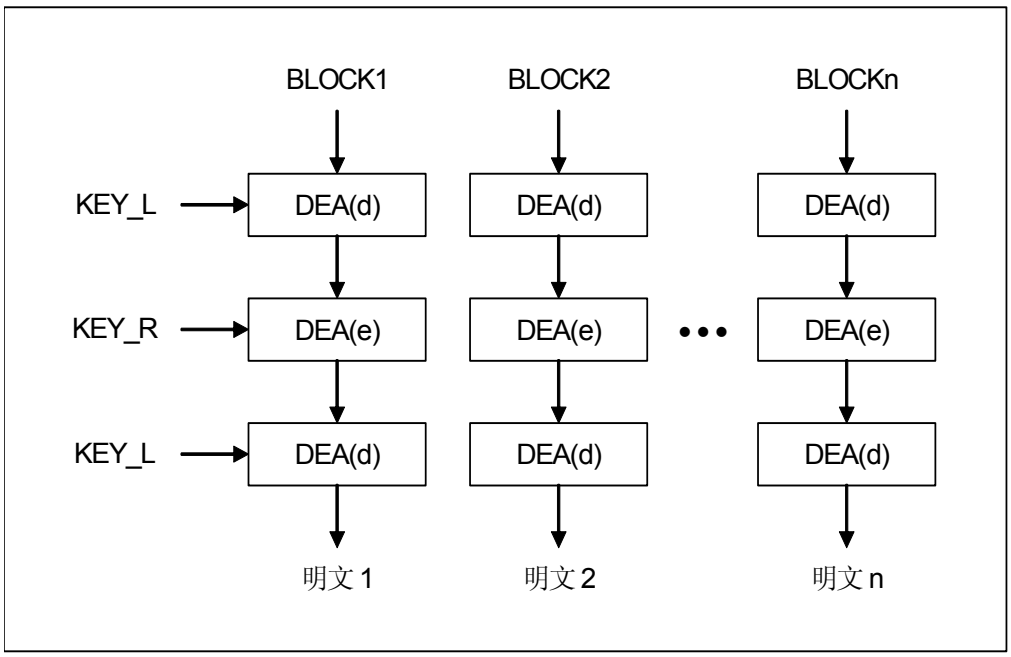


图 8-14 双倍长密钥 DEA 数据解密算法

8.4.2 SM1 算法在金融环境中的安全管理应用

SM1 是以 128 位分组为单位进行运算的对称密钥算法，密钥长度为 16 字节。

8.4.2.1 密钥分散的计算方法

将应用序列号的最右 16 个数字作为分散因子（8 字节）

用指定的分散因子加上分散因子求反值作为输入数据，执行SM1(e)计算，产生的 16 字节结果作为子密钥。见图 8-15。

密钥分散通过分散因子产生子密钥。

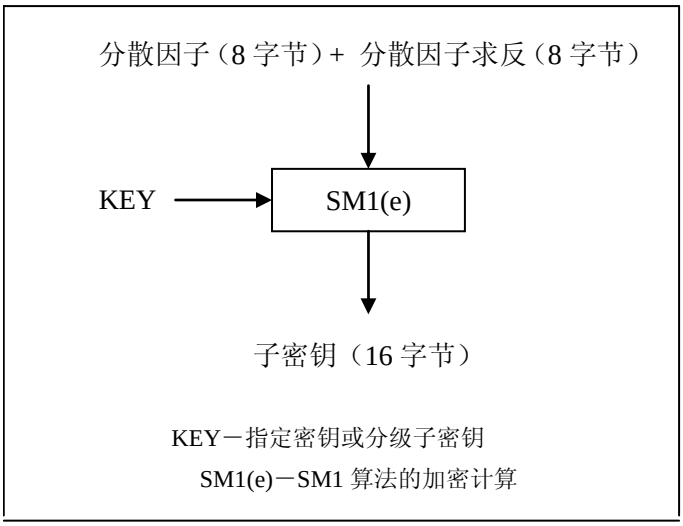


图 8-15 密钥分散计算方法

8.4.2.2 过程密钥的计算方法

过程密钥输入因子由 8 字节交易数据元合成因子补 8 字节“0000000000000000”达到 16 字节构成。 见图 8-16。

通过对过程密钥输入因子做 SM1(e)运算来产生过程密钥。

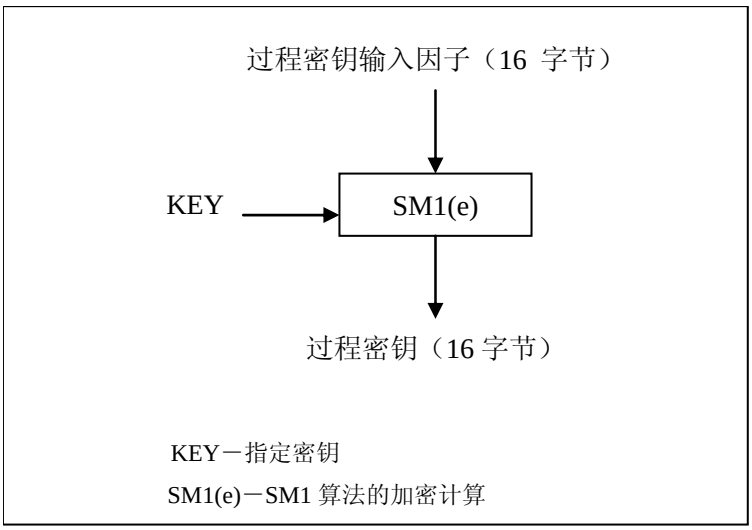


图 8-16 过程密钥的计算方法

另，对于灰锁交易，过程密钥的产生分两步：

——先用交易密钥按上述方法生成中间密钥；

输入数据：伪随机数（ICC） || 电子钱包脱机交易序号 || 终端交易序号的最右两个字节 || 8 字节'00'；

——再用中间密钥按上述算法生成过程密钥；

输入数据：终端随机数 || '80000000' || 8 字节'00'。

8.4.2.3 鉴别数据的计算方法

鉴别数据的计算方法是通过对鉴别数据输入因子（16 字节）做 SM1(e)运算来产生鉴别数据，供接口设备对 IC 卡进行内部验证或外部认证。

鉴别数据输入因子由 4 字节随机数补 12 字节"000000000000000000000000"达到 16 字节构成；或由 8 字节随机数补 8 字节"0000000000000000"达到 16 字节，或取 16 随机数构成。这里KEY是指定密钥。见图 8-17。

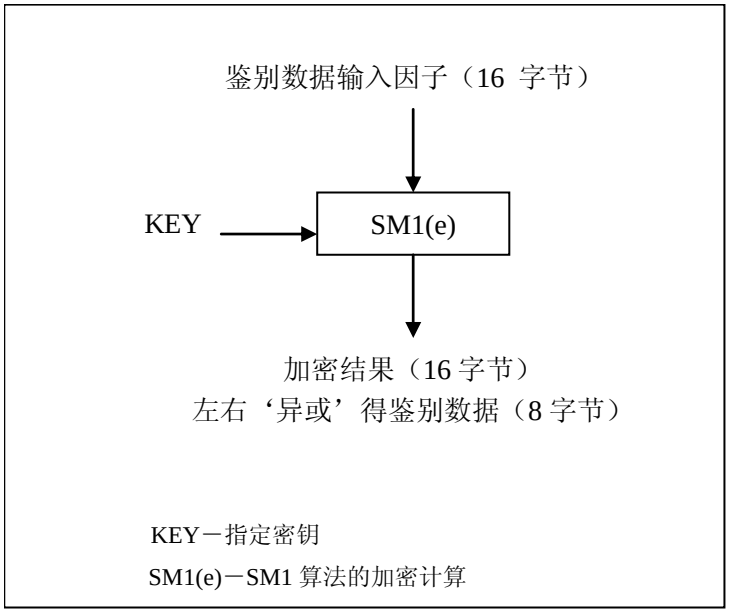


图 8-17 鉴别数据的计算方法

8.4.2.4 MAC 的计算方法

8.4.2.4.1 命令安全报文中的 MAC

命令安全报文中的MAC是使用命令的所有元素（包括命令头和命令数据域中的数据）来产生的。以保证命令连同数据能够正确完整地传送，并对发送方进行认证。按照图 8-18所示做SM1(e)运算产生MAC：

第一步： 取 4 字节随机数补 12 字节“000000000000000000000000”达到 16 字节作为初始值；或 8 字节随机数补 8 字节’00 00 00 00 00 00 00 00’达到 16 字节，或取 16 随机数

第二步： 将 5 字节命令头（CLA，INS，P1，P2，Lc）和命令数据域中的明文或密文数据连接在一起形成数据块。注意，这里的 Lc 应是数据长度加上将计算出的 MAC 的长度（4 字节）后得到的实际长度，Lc 的值不小于 4。

第三步： 将该数据块分成 16 字节为单位的数据块， 表示为 BLOCK1、BLOCK2 … … BLOCKn 等。最后的数据块有可能是 1~16 个字节。

第四步： 如果最后的数据块的长度是 16 字节的话，则在该数据块之后再加一个完整的 16 字节数据块‘80 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00’， 转到第五步。

如果最后的数据块的长度不足 16 字节，则在其后加入 16 进制数‘80’，如果达到 16 字节长度，则转到第五步；否则接着在其后加入 16 进制数‘00’直到长度达到 16 字节。

第五步：　　按照图 8-18所述步骤，对这些数据块使用指定密钥进行加密。

第六步：　　将 16 字节运算结果按 4 字节分块做异或运算，最终结果作为 MAC。

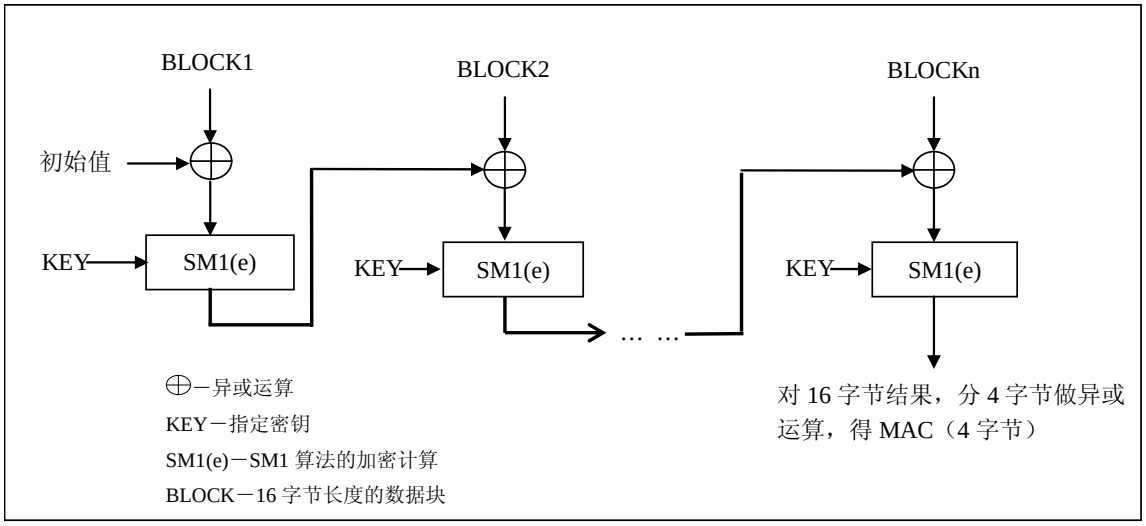


图 8-18 安全报文中的 MAC 计算

8.4.2.4.2 交易中 MAC/GMAC/TAC/GTAC 的计算方法

计算 TAC/GTAC 的密钥采用指定 DTK 密钥。

计算 MAC/GMAC 的密钥则采用过程密钥，过程密钥的计算方法请参考过程密钥的计算。

第一步：　　取 16 字节“00”作为初始值。

第二步：　　将 5 字节命令头（CLA，INS，P1，P2，Lc）和命令数据域中的明文或密文数据连接在一起形成数据块。注意，这里的 Lc 应是数据长度加上将计算出的 MAC 的长度（4 字节）后得到的实际长度，Lc 的值不小于 4。

第三步：　　将该数据块分成 16 字节为单位的数据块， 表示为 BLOCK1、BLOCK2 … … BLOCKn 等。最后的数据块有可能是 1~16 个字节。

第四步：　　如果最后的数据块的长度是 16 字节的话，则在该数据块之后再加一个完整的 16 字节数据块‘80 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00’， 转到第五步。

如果最后的数据块的长度不足 16 字节，则在其后加入 16 进制数‘80’，如果达到 16 字节长度，则转到第五步；否则接着在其后加入 16 进制数‘00’直到长度达到 16 字节。

第五步：　　按照图 8-18所述步骤，对这些数据块使用过程密钥或者DTK进行加密。

第六步：　　将 16 字节运算结果按 4 字节分块做异或运算，最终结果作为 MAC。

8.4.2.5 数据加密的计算方法

安全报文传送中数据的可靠性通过对数据域的加密来得到保证。为保证命令中明文数据的保密性，可按照图 8-19所示做SM1(e)运算，对数据进行加密：

第一步：　　用 LD（1 字节）表示明文数据的长度，在明文数据前加上 LD 产生新的数据块，LD 的值不小于 1。

第二步：　　将该数据块分成 16 字节为单位的数据块，表示为 PLAIN1、PLAIN2 … … PLAINn 等。最后的数据块有可能是 1~16 个字节。

第三步：　　如果最后（或唯一）的数据块的长度是 16 字节的话，转到第四步；如果不足 16 字节，则在其后加入 16 进制数‘80’，如果达到 16 字节长度，则转到第四步；否则在其后加入 16 进制数‘00’直到长度达到 16 字节。

第四步：　　按照图 8-19所述的算法对每一个数据块进行加密。

第五步：　　计算结束后，所有加密后的数据块依照原顺序连接在一起。

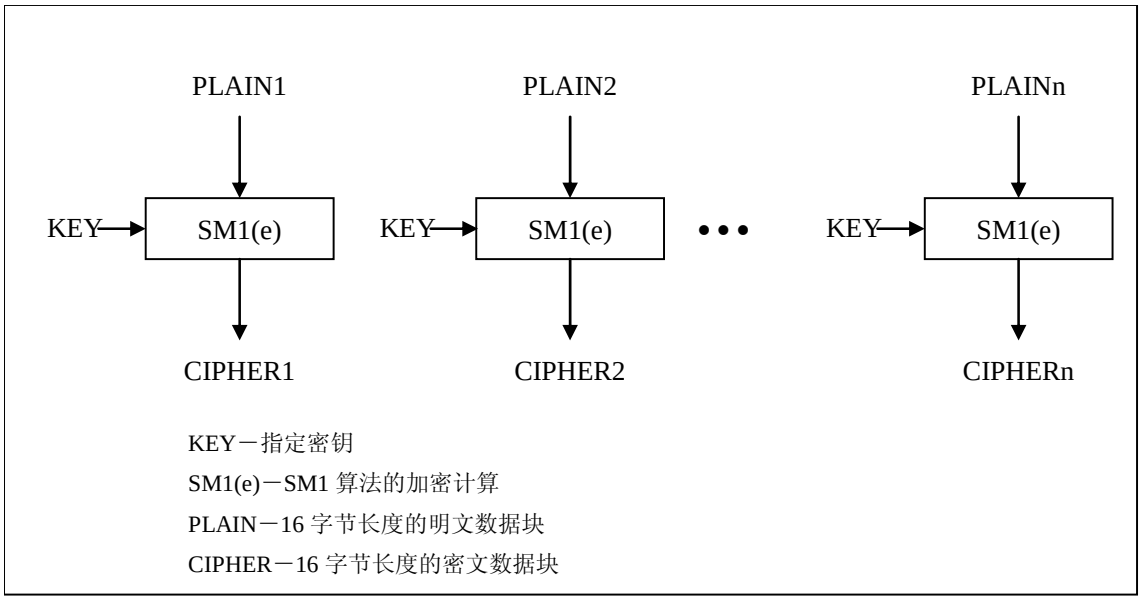


图 8-19 数据加密

8.4.2.6 数据解密的计算方法

数据解密与数据加密采用相反的过程，如图 8-20。

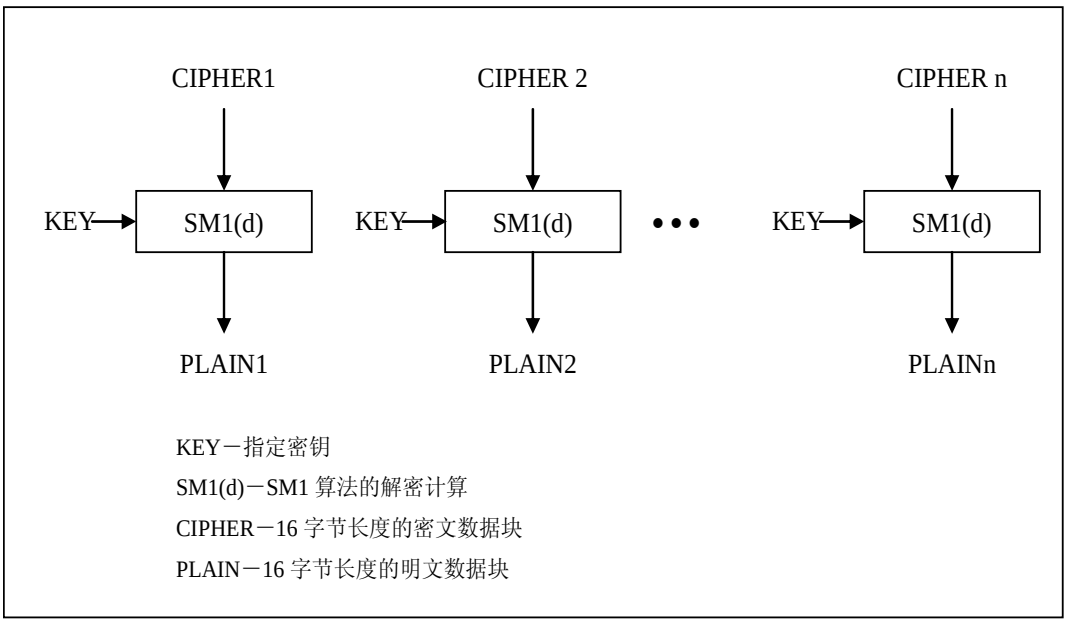


图 8-20 数据解密

9 命令

9.1 命令与响应的格式

9.1.1 命令格式

命令由“命令头”和“命令体”组成

命令头				命令体		
CLA	INS	P1	P2	Lc	DATA	Le

命令可分为四种情况：

格式	命令组成
CASE 1	CLA INS P1 P2
CASE 2	CLA INS P1 P2 Le
CASE 3	CLA INS P1 P2 Lc Data
CASE 4	CLA INS P1 P2 Lc Data Le

9.1.2 响应格式

响应的格式：

数据	状态字	
DATA	SW1	SW2

DATA ：响应数据
SW1、SW2：卡片执行命令的返回值

9.2 COS 支持的命令集

9.2.1 基本命令

9.2.1.1 APPEND RECORD 命令

9.2.1.1.1 命令描述

APPEND RECORD 命令用于向记录文件中添加新记录。对循环记录文件，可无限添加记录；对其他记录文件，只能添加到文件的最后一条记录。

9.2.1.1.2 使用条件和安全

在个人化状态下，APPEND RECORD 命令的执行不需要满足文件的添加条件，只需以明文方式写入；在应用状态下，APPEND RECORD 命令的执行必须满足被选文件的添加条件和添加属性。

9.2.1.1.3 命令格式

代码	数　值								
CLA	‘00’或‘04’								
INS	‘E2’								
P1	‘00’								
P2	b8	b7	b6	b5	b4	b3	b2	b1	说　明
	0	0	0	0	0	0	0	0	当前的 EF 文件
	x	x	x	x	x	0	0	0	用 SFI 方式
Lc	DATA 域数据长度 **DES**： **明文方式**：　‘00’< Lc ≤‘FF’ **加密方式**：　‘08’≤ Lc ≤‘F8’（模 8） **校验方式**：　‘04’< Lc ≤‘FE’ **校验加密方式**：‘0C’≤ Lc ≤‘FC’（模 8 + 4） **SM1**： **明文方式**：　‘00’< Lc ≤‘FF’ **加密方式**：　‘10’≤ Lc ≤‘F0’（模 16） **校验方式**：　‘04’< Lc ≤‘FE’ **校验加密方式**：‘14’≤ Lc ≤‘F4’（模 16 + 4）								
DATA	明文方式：　新数据 加密方式：　被加密的新数据 校验方式：　新数据||MAC 校验加密方式：被加密的新数据||MAC								
Le	不存在								

当文件 ACw 中的 CER 和/或 CIPH 为‘1’时，必须使用安全报文传送；即 CLA 的后半字节等于十六进制‘4’。 若 CER=‘1’，则发送命令前，先要执行 GET

CHALLENGE 命令，向卡申请一随机数作为 MAC 计算的初值。

9.2.1.1.4 响应信息

响应信息中可能出现的状态码有：

SW1	SW2	说　明
90	00	命令执行成功
65	81	写 EEPROM 失败
67	00	Lc 长度错误
69	81	当前文件不是记录文件
69	82	不满足安全状态
69	83	认证密钥锁定
69	84	引用数据无效（未申请随机数）
69	85	使用条件不满足
69	86	没有选择当前文件
69	88	安全信息（MAC 和加密）数据错误
6A	81	功能不支持
6A	82	未找到文件
6A	84	记录空间已满
6A	85	Lc 与 TLV 结构不匹配
6A	86	P1、P2 参数错
6A	88	未找到密钥数据
6D	00	命令不存在
6E	00	CLA 错
93	03	应用永久锁定

9.2.1.1.5 命令详解

1、DES 算法：

条件 1：　文件标识符：’EF21’(线性定长记录文件)
　　通过 SFI 方式访问(注意 P2 参数)
　　明文方式写入
　　最大记录长度为’FF’

操作：　向记录文件中添加一条新的长度为’08’个字节的记录。
　　‘1806000000000000’为记录内容 DataIn

命令：　‘00E2000808’ + DataIn

响应：　‘9000’

条件 2：　文件标识符：’EF22’ (线性定长记录文件)
　　通过 SFI 方式访问(注意 P2 参数)
　　明文+MAC 方式写入(DES 算法)
　　本例的记录长度为’F3’

操作：　向记录文件中添加一条新的长度为’F3’个字节的记录。
‘0000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000
000000000000000’为记录内容 DataIn

步骤 1：　取 4 字节随机数 Random+’00000000’作为 MAC 计算初始值 InitialData。

命令：　‘00840000 04’

响应：　Random

步骤 2：　用应用维护密钥 DAMK（本例取 16 字节’00’）计算 MAC
初始值 InitialData= Random 4 字节
命令头 HeadData= ‘04E20010F7’
MAC 计算数据为：MACData = InitialData + HeadData + DataIn
3DES MAC（DAMK，MACData，MAC）
MAC=’ A030FDE6’

步骤 3：　明文 + MAC 写入数据

命令：　‘04E20010F7’ + DataIn + MAC

响应：　‘9000’

条件 3：　文件标识符：’EF23’ (线性定长记录文件)
通过 SFI 方式访问(注意 P2 参数)
密文方式写入(DES 算法)
本例的记录长度为’6F’

操作：　向记录文件中添加一条新的长度为’EF’个字节的记录。
‘0000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000’为记录内容
DataIn

步骤1：　用应用维护密钥 DAMK（本例取 16 字节’00’）计算密文
3DES Encryption（DAMK，(Ld+DataIn)，EncData）
密文 EncData =
‘E02BA426582B63318CA64DE9C1B123A78CA64DE9C1B123A78CA64
DE9C1B123A78CA64DE9C1B123A78CA64DE9C1B123A78CA64DE9C1
B123A78CA64DE9C1B123A78CA64DE9C1B123A78CA64DE9C1B123A
78CA64DE9C1B123A78CA64DE9C1B123A78CA64DE9C1B123A78CA6
4DE9C1B123A7’

步骤2：　密文写入数据

命令： ‘04E2001870’ + EncData
响应： ‘9000’

条件 4： 文件标识符：’EF24’ (线性定长记录文件)
通过 SFI 方式访问(注意 P2 参数)
密文+MAC 方式写入(DES 算法)
本例使用的记录长度为’6F’
操作： 向记录文件中添加一条新的长度为’6F’个字节的记录。
‘00’为记录内容 DataIn
步骤 1： 用应用维护密钥 DAMK（本例取 16 字节’00’）计算密文
3DES Encryption（DAMK，(Ld+DataIn)，EncData）
密文 EncData=
‘E02BA426582B63318CA64DE9C1B123A78CA64DE9C1B123A78CA64DE9C1B123A78CA64DE9C1B123A78CA64DE9C1B123A78CA64DE9C1B123A78CA64DE9C1B123A78CA64DE9C1B123A78CA64DE9C1B123A78CA64DE9C1B123A78CA64DE9C1B123A78CA64DE9C1B123A78CA64DE9C1B123A78CA64DE9C1B123A7’
步骤 2： 取 8 字节随机数 Random，作为 MAC 计算初始值 InitialData。
命令： ‘00840000 08’
响应： Random
步骤 3： 用应用维护密钥计算 MAC
初始值 InitialData= Random’ 4 字节
命令头 HeadData= ‘04E20020F4’
MAC 计算数据为：MACData = InitialData + HeadData + EncData
3DES MAC（DAMK，MACData，MAC）
MAC=’ 3ECED41B’
步骤 4： 密文+MAC 写入数据
命令： ‘04E20020F4’ + EncData + MAC
响应： ‘9000’

2、SM1 算法：
条件 1： 文件标识符：’EF21’(线性定长记录文件)
通过 SFI 方式访问(注意 P2 参数)
明文方式写入
本例的记录长度为’08’
操作： 向记录文件中添加一条新的长度为’08’个字节的记录。
‘1806000000000000’为记录内容 DataIn

命令：　‘00E2000808’ + DataIn
响应：　‘9000’

条件 2：　文件标识符：’EF22’ (线性定长记录文件)
通过 SFI 方式访问(注意 P2 参数)
明文+MAC 方式写入(SM1 算法)
本例的记录长度为’F3’

操作：　向记录文件中添加一条新的长度为’F3’个字节的记录。
‘0000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000
000000000000000’为记录内容 DataIn

步骤 1：　取 8 字节随机数 Random+’0000000000000000’作为 MAC 计算初始值 InitialData。
命令：　‘00840000 08’
响应：　Random
步骤 2：　用应用维护密钥 DAMK（本例取 16 字节’00’）计算 MAC
初始值 InitialData= Random 8 字节+’0000000000000000’
命令头 HeadData= ‘04E20010F7’
MAC 计算数据为：MACData = InitialData + HeadData + DataIn
3DES MAC（DAMK，MACData，MAC）
MAC=’ E8FB70D0’
步骤 3：　明文 + MAC 写入数据
命令：　‘04E20010F7’ + DataIn + MAC
响应：　‘9000’

条件 3：　文件标识符：’EF23’ (线性定长记录文件)
通过 SFI 方式访问(注意 P2 参数)
密文方式写入(SM1 算法)
本例的记录长度为’EF’

操作：　向记录文件中添加一条新的长度为’EF’个字节的记录。
‘0000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000

000’为记录内容 DataIn

步骤 1： 用应用维护密钥 DAMK（本例取 16 字节’00’）计算密文
3DES Encryption（DAMK，(Ld+DataIn)，EncData）
密文 EncData =
‘AD668986227F2F878EDDAC4677349EC5CEE479B4B459804539FF5D4842015644CEE479B4B459804539FF5D4842015644CEE479B4B459804539FF5D4842015644CEE479B4B459804539FF5D4842015644CEE479B4B459804539FF5D4842015644CEE479B4B459804539FF5D4842015644CEE479B4B459804539FF5D4842015644CEE479B4B459804539FF5D4842015644CEE479B4B459804539FF5D4842015644CEE479B4B459804539FF5D4842015644CEE479B4B459804539FF5D4842015644CEE479B4B459804539FF5D4842015644CEE479B4B459804539FF5D4842015644’

步骤 2： 密文写入数据
命令： ‘04E20018F0’ + EncData
响应： ‘9000’

条件 4： 文件标识符：’EF24’ (线性定长记录文件)
通过 SFI 方式访问(注意 P2 参数)
密文+MAC 方式写入(SM1 算法)
本例使用的记录长度为’EF’

操作： 向记录文件中添加一条新的长度为’EF’个字节的记录。
‘00080’为记录内容 DataIn

步骤 1： 用应用维护密钥 DAMK（本例取 16 字节’00’）计算密文
3DES Encryption（DAMK，(Ld+DataIn)，EncData）
密文 EncData=
‘AD668986227F2F878EDDAC4677349EC5CEE479B4B459804539FF5D4842015644CEE479B4B459804539FF5D4842015644CEE479B4B459804539FF5D4842015644CEE479B4B459804539FF5D4842015644CEE479B4B459804539FF5D4842015644CEE479B4B459804539FF5D4842015644CEE479B4B459804539FF5D4842015644CEE479B4B459804539FF5D4842015644CEE479B4B459804539FF5D4842015644CEE479

B4B459804539FF5D4842015644CEE479B4B459804539FF5D4842015644CEE479B4B459804539FF5D4842015644’

步骤2： 取 8 字节随机数 Random，作为 MAC 计算初始值 InitialData。

命令： ‘00840000 08’

响应： Random

步骤3： 用应用维护密钥计算 MAC

初始值 InitialData= Random’ 4 字节

命令头 HeadData= ‘04E20020F4’

MAC 计算数据为：MACData = InitialData + HeadData + EncData

3DES MAC（DAMK，MACData，MAC）

MAC=’A04EE666’

步骤4： 密文+MAC 写入数据

命令： ‘04E20020F4’ + EncData + MAC

响应： ‘9000’

9.2.1.2 APPLICATION BLOCK 命令

9.2.1.2.1 命令描述

APPLICATION BLOCK 命令执行成功后，锁定当前有效应用。锁定后的应用仍可以被选择（成功选择应用后，返回 SW1_SW2='6A81'），但被锁定应用下的文件是不可访问的，任何试图对文件的访问都将返回 SW1SW2='6A81'。锁定后的应用可通过 Get Response 命令得到 FCI 信息。如果应用被永久锁定返回 SW1SW2='9303'。

9.2.1.2.2 使用条件和安全

APPLICATION BLOCK 命令的执行采用校验模式。计算校验码使用的 KEY 为 ADF 文件中的 BLK-KID 密钥。

9.2.1.2.3 命令格式

代码	数　值
CLA	'84'
INS	'1E'
P1	'00'
P2	'00'锁定后可用 APPLICATION　UNBLOCK 命令解锁 '01'永久锁定应用
Lc	'04'
DATA	信息认证码（MAC）
Le	不存在

9.2.1.2.4 响应信息

响应信息中可能出现的状态码有：

SW1	SW2	说　明
90	00	命令执行成功
62	81	回送数据可能出错
62	83	选择文件无效
64	00	状态标志位未变
65	81	写 EEPROM 失败
67	00	Lc 长度错误
69	00	无信息提供
69	82	不满足安全状态
69	84	引用数据无效（未申请随机数）
69	85	使用条件不满足
69	88	安全信息（MAC）数据错误

6A	81	功能不支持
6A	86	P1、P2 参数错
6A	88	未找到密钥数据
6D	00	命令不存在
6E	00	CLA 错
93	03	应用永久锁定
93	03	应用永久锁定

9.2.1.2.5 命令详解

1、DES 算法

操作： 锁定当前 ADF。

步骤 1： 取 4 字节随机数 Random，后补’00000000’作为 MAC 计算初始值。

命令： ‘00840000 04’

响应： Random=’B5B0C549’

步骤 2： 用当前 ADF 下 BLK-KID 指向密钥 BLK（本例取：16 字节’00’）计算 MAC。

初始值 InitialData=’B5B0C54900000000’ 8 字节

命令头 HeadData= ‘841E000104’

MAC 计算数据为：MACData = InitialData + HeadData

3DES MAC（BLK，MACData，MAC）

MAC=’34BE5E04’

步骤 3： 锁定当前 ADF。

命令： ‘841E000104’ + MAC

响应： 9000

2、 SM1 算法

操作： 锁定当前 ADF。

步骤 1： 取 8 字节随机数 Random。

命令： ‘00840000 08’

响应： Random = ’ 3CD9B04AC54E19A2’

步骤 2： 用当前 ADF 下 BLK-KID 指向密钥 BLK（本例取：16 字节’00’）计算 MAC。

初始值 InitialData= Random+’000000000000000’ 8 字节

命令头 HeadData= ‘841E000004’

MAC 计算数据为：MACData = InitialData + HeadData

SM1 MAC（BLK，MACData，MAC）

MAC=’ 282167E3’

步骤 3： 永久锁定当前 ADF

命令： ‘841E000104’ + MAC

响应： 9000

9.2.1.3 APPLICATION UNBLOCK 命令

9.2.1.3.1 命令描述

APPLICATION UNBLOCK 命令执行成功后，解锁当前锁定的应用。

9.2.1.3.2 使用条件和安全

APPLICATION UNBLOCK 命令的执行必须采用命令校验方式。使用 ADF 文件头中 BLK-KID 所指定的密钥。该密钥必须满足可使用条件。

9.2.1.3.3 命令格式

代码	数　值
CLA	‘84’
INS	‘18’
P1	‘00’
P2	‘00’
Lc	‘04’
DATA	信息认证码（MAC）
Le	不存在

9.2.1.3.4 响应信息

响应信息中可能出现的状态码有：

SW1	SW2	说　明
90	00	命令执行成功
64	00	状态标志位未变
65	81	写 EEPROM 失败
67	00	Lc 长度错误
69	82	不满足安全状态
69	83	认证密钥锁定
69	84	引用数据无效（未申请随机数）
69	85	使用条件不满足
69	88	安全信息（MAC）数据错误
6A	81	功能不支持
6A	86	P1、P2 参数错
6D	00	命令不存在
6E	00	CLA 错
93	03	应用永久锁定

9.2.1.3.5 命令详解

1、DES 算法

操作： 解锁当前 ADF。
步骤 1： 取 4 字节随机数 Random，后补’00000000’作为 MAC 计算初始值。
命令： ‘00840000 04’
响应： Random=’D3272021’
步骤 2： 用当前 ADF 下 BLK-KID 指向密钥 BLK（本例取：16 字节’00’）计算 MAC。
初始值 InitialData=’ DFBD968E00000000’ 8 字节
命令头 HeadData= ‘8418000004’
MAC 计算数据为：MACData = InitialData + HeadData
3DES MAC（BLK，MACData，MAC）
MAC=’FB4A4741’
步骤 3： 解锁当前 ADF。
命令： ‘8418000004’ + MAC
响应： 9000

2、SM1 算法

操作： 解锁当前 ADF。
步骤 1： 取 8 字节随机数 Random，后补’0000000000000000’作为 MAC 计算初始值。
命令： ‘00840000 08’
响应： Random=’ 5C9428E3FED70C23’
步骤 2： 用当前 ADF 下 BLK-KID 指向密钥 BLK（本例取：16 字节’00’）计算 MAC。
初始值 InitialData=’ Random’ 8 字节+’0000000000000000’
命令头 HeadData= ‘8418000004’
MAC 计算数据为：MACData = InitialData + HeadData
SM1 MAC（BLK，MACData，MAC）
MAC=’ BAB74861’
步骤 3： 解锁当前 ADF。
命令： ‘8418000004’ + MAC
响应： 9000

9.2.1.4 CARD BLOCK 命令

9.2.1.4.1 命令描述

成功执行 CARD BLOCK 命令后，卡被锁定。除 GET INFO 命令外，卡拒绝执行任何命令，返回状态信息'6A81'。

9.2.1.4.2 使用条件和安全

命令的执行必须采用校验模式。使用 MF、DDF 或 ADF 文件中 BLK-KID 所指定的密钥计算校验码。该命令必须在 DDF 下执行。

9.2.1.4.3 命令格式

代码	数　值
CLA	'84'
INS	'16'
P1	'00'
P2	'00'
Lc	'04'
DATA	信息认证码（MAC）
Le	不存在

9.2.1.4.4 响应信息

响应信息中可能返回的状态码有：

SW1	SW2	说　明
90	00	命令执行成功
64	00	状态标志位未变
65	81	写 EEPROM 失败
67	00	Lc 长度错误
69	84	引用数据无效（未申请随机数）
69	85	使用条件不满足
69	88	安全信息（MAC）数据错误
6A	81	功能不支持
6A	86	P1、P2 参数错
6A	88	未找到密钥数据
6D	00	命令不存在
6E	00	CLA 错

9.2.1.4.5 命令详解

1、DES 算法

操作：　卡片锁定。

步骤 1： 取 4 字节随机数 Random，后补’00000000’作为 MAC 计算初始值。
命令： ‘00840000 04’
响应： Random = ’C2F02FFC’
步骤 2： 用当前 MF 或 DDF 下 BLK-KID 指向密钥 BLK（本例取：16 字节’00’）计算 MAC
初始值 InitialData=’ C2F02FFC00000000’ 8 字节
命令头 HeadData= ‘8416000004’
MAC 计算数据为：MACData = InitialData + HeadData
3DES MAC（BLK，MACData，MAC）
MAC=’00CE33E6’
步骤 3： 卡片锁定。
命令： ‘8416000004’ + MAC
响应： 9000

2、SM1 算法

操作： 卡片锁定。
步骤 1： 取 8 字节随机数 Random，作为过程密钥产生因子。
命令： ‘00840000 04’
响应： Random = ’ 27CE5EAE’
步骤 2： 用当前 MF 或 DDF 下 BLK-KID 指向密钥 BLK（本例取：16 字节’00’）计算 MAC
初始值 InitialData= Random 4 字节+’00000000’
命令头 HeadData= ‘8416000004’
MAC 计算数据为：MACData = InitialData + HeadData
SM1 MAC（BLK，MACData，MAC）
MAC=’ 6E838F47’
步骤 3： 卡片锁定。
命令： ‘8416000004’ + MAC
响应： 9000

9.2.1.5 EXTERNAL AUTHENTICATE 命令

9.2.1.5.1 命令描述

EXTERNAL AUTHENTICATE 命令的目的是 IC 卡中的应用验证外部接口设备的有效性，使接口设备对 IC 卡获得某种操作授权。

接口设备提供的认证数据应按以下规则产生：

金融应用环境	1、Lc = ‘08’ 2、用 GET CHALLENGE 命令向 IC 卡申请一组随机数。 3、用指定密钥对随机数作加密计算，产生认证数据。参见“安全计算”一节。

9.2.1.5.2 使用条件和安全

EXTERNAL AUTHENTICATE 命令所使用的密钥（由 P2 参数指定）必须满足密钥的访问权限。密钥验证失败计数器减一。当计数器减为‘0’值时，密钥被锁定。

9.2.1.5.3 命令格式

代码	数　值								
CLA	‘00’								
INS	‘82’								
P1	‘00’								
P2	b8	b7	b6	b5	b4	b3	b2	b1	说　明
	0	-	-	-	-	-	-	-	MF 或 DDF 下的密钥
	1	-	-	-	-	-	-	-	ADF 下的密钥
	-	x	x	x	x	x	x	x	密钥标识
	0	0	0	0	0	0	0	0	当前 DF 下的 MK
Lc	‘08’								
DATA	认证数据（8 字节）								
Le	不存在								

9.2.1.5.4 响应信息

响应信息中可能出现的状态码有：

SW1	SW2	说　明
90	00	命令执行成功
63	Cx	认证失败，还可认证 x 次

65	81	写 EEPROM 失败
67	00	Lc 长度错误
69	82	不满足安全状态
69	83	认证密钥锁定
69	84	引用数据无效（未申请随机数）
69	85	使用条件不满足
6A	81	功能不支持
6A	86	P1、P2 参数错
6A	88	未找到密钥数据
6D	00	命令不存在
6E	00	CLA 错
93	03	应用永久锁定

9.2.1.5.5 命令详解

1、DES 算法

操作： 认证 MF 的主控密钥

步骤 1： 取 8 字节随机数 Random，作为鉴别数据产生因子。

命令： ‘00840000 08’

响应： Random=’ 8F8D5AEA85880901’

步骤 2： 终端用与 MF 主控密钥 MF_MK 相同的密钥（本例取：16 字节’00’）对随机数加密，产生鉴别数据。

3DES Encryption（MF_MK，Random，VeriData）

鉴别数据计算结果为：VeriData = ’ 82FE8A38C35A59DF’

步骤 3： 外部认证 MF 主控密钥

命令： ‘0082000008’ + VeriData

响应： ‘9000’

2、SM1 算法

操作： 认证 MF 的主控密钥

步骤 1： 取 8 字节随机数 Random，作为鉴别数据产生因子。

命令： ‘00840000 08’

响应： Random=’ 5ECF5C58E8EFF667’

步骤 4： 终端用与 MF 主控密钥 MF_MK 相同的密钥（本例取：16 字节’00’）对 Random 补位后的结果加密计算鉴别数据 VeriData

DataIn：Random +’0000000000000000’

SM1 Encryption（MF_MK，DataIn，DataOut）

DataOut=’ 151CDF0AABECCEE47E059B635B358846’

VeriData=Xor（DataOut，8）左右异或运算

VeriData=’AFA3AEDC72C9E4FF’

步骤 5： 外部认证 MF 主控密钥

命令： ‘0082000008’ + VeriData

响应： ‘9000’

9.2.1.6 GET CHALLENGE 命令

9.2.1.6.1 命令描述

GET CHALLENGE 命令从 IC 卡中获取一组随机数，用于相关命令的安全认证。

9.2.1.6.2 使用条件和安全

GET CHALLENGE 命令可无条件使用。

9.2.1.6.3 命令格式

代码	数　值
CLA	‘00’
INS	‘84’
P1	‘00’
P2	‘00’
Lc	不存在
DATA	不存在
Le	‘04’或‘08’或‘10’

9.2.1.6.4 响应信息

响应信息中的数据：

说 明	长度（字节）
随机数	4 或 8 或 16

响应信息中可能返回的状态码有：

SW1	SW2	说　明
90	00	命令执行成功
67	00	Le 长度错误
6A	81	功能不支持
6A	86	P1、P2 参数错
6D	00	命令不存在
6E	00	CLA 错

9.2.1.6.5 命令详解

在任何情况下，直接执行此命令即可。

9.2.1.7 GET RESPONSE 命令

9.2.1.7.1 命令描述

GET REPONSE 命令从 IC 卡中向接口设备传送 APDU 的数据。

9.2.1.7.2 使用条件和安全

GET REPONSE 命令无使用条件限制。

9.2.1.7.3 命令格式

代码	数　值
CLA	‘00’
INS	‘C0’
P1	‘00’
P2	‘00’
Lc	不存在
DATA	不存在
Le	响应的最大数据长度

9.2.1.7.4 响应信息

响应信息中的数据：

说　明	长度（字节）
响应数据	X

响应信息中可能返回的状态码有：

SW1	SW2	说　明
90	00	命令执行成功
61	xx	还有 xx 字节需要返回
62	81	回送数据可能有错
67	00	Lc 或 Le 长度错误
6A	86	P1、P2 参数错
6C	xx	长度错误，‘xx’表示实际长度
6D	00	命令不存在
6E	00	CLA 错
6F	00	数据无效

9.2.1.7.5 命令详解

在任何情况下，直接执行此命令即可。

9.2.1.8 INTERNAL AUTHENTICATE 命令

9.2.1.8.1 命令描述

INTERNAL AUTHENTICATE 命令的目的是 IC 卡向外部接口设备提供认证数据，以使接口设备对 IC 卡进行认证。

认证数据按以下规则产生：IC 卡对接口设备提供的数据作加密计算，同时将产生的认证数据回送给接口设备。

金融应用环境	1、Lc = ‘08’ 2、用指定密钥对随机数作加密计算，产生认证数据。参见“安全计算”一节。

9.2.1.8.2 使用条件和安全

INTERNAL AUTHENTICATE 命令所使用的密钥（由 P2 参数指定）必须满足密钥的访问权限。

9.2.1.8.3 命令格式

代码	数 值								
CLA	‘00’								
INS	‘88’								
P1	‘00’								
P2	b8	b7	b6	b5	b4	b3	b2	b1	说 明
	0	-	-	-	-	-	-	-	MF 或 DDF 下的密钥
	1	-	-	-	-	-	-	-	ADF 下的密钥
	-	x	x	x	x	x	x	x	密钥标识
	0	0	0	0	0	0	0	0	主内部认证密钥标识
Lc	‘08’								
DATA	输入数据（8 字节）								
Le	‘08’（认证数据）								

9.2.1.8.4 响应信息

响应信息中的数据：

说 明	长 度（字节）
认证数据	8

响应信息中可能返回的状态码有：

SW1	SW2	说 明
90	00	命令执行成功

62	81	回送数据可能有错
64	00	标志状态位未变
67	00	Lc 长度错误
69	82	不满足安全状态
69	85	使用条件不满足
6A	81	功能不支持
6A	86	P1、P2 参数错
6A	88	未找到密钥数据
6D	00	命令不存在
6E	00	CLA 错
93	03	应用永久锁定

9.2.1.8.5 命令详解

1、DES 算法

操作： 发出内部认证命令

条件： 内部认证密钥标识’01’，内部认证密钥值，本例取 16 字节’00’，输入数据为’1122334455667788’

命令： ‘0088000108’ + ’1122334455667788’

响应： ‘CD72DFC6E6D040A4’

2、SM1 算法

操作： 发出内部认证命令

条件： MF 下，主内部认证密钥标识为’00’，密钥值本例取 16 字节’11’，输入数据为’1306BCEA8FF617F9’。

命令： ‘0088000308’ + ‘1306BCEA8FF617F9’

响应： ‘C1EFE951CDD9F3CA’

9.2.1.9 PIN CHANGE/UNLOCK 命令

9.2.1.9.1 命令描述

PIN CHANGE/UNLOCK 命令为使用者提供了更改 PIN 和解锁 PIN 的功能。执行命令前，应先执行 GET CHALLENGE 命令。

9.2.1.9.2 使用条件和安全

更改和解锁 PIN 时，命令的执行必须满足 PIN 的访问权限。如果更改 PIN，命令格式为密文校验模式。如果解锁 PIN，命令格式为校验模式。

对于金融主 PIN 解锁请参看 PIN UNLOCK 命令。

对于金融主 PIN 重装请参看 RELOAD PIN 命令。

9.2.1.9.3 命令格式

代码	数 值								
CLA	‘84’								
INS	‘24’								
P1	‘00’								
P2	b8	b7	b6	b5	b4	b3	b2	b1	说 明
	0	-	-	-	-	-	-	-	MF 或 DDF 下的 PIN
	1	-	-	-	-	-	-	-	ADF 下的 PIN
	-	x	x	x	x	x	x	-	PIN 的 KID
								0	解锁 PIN，尝试计数器重置，但不修改 PIN
								1	更改 PIN，尝试计数器重置，同时修改 PIN
	0	0	0	0	0	0	0	-	MPIN
Lc	**校验方式：**’04’ **密文校验方式：**’0C’ (DES 算法)或者’14’（SM1 算法）								
DATA	—其他应用 若 P2=‘xxxxxxx0’，MAC 若 P2=‘xxxxxxx1’，新 PIN 数据密文||MAC P2=‘01’，使用 DPUK 对 PIN 数据加密。 —PBOC 应用 若 P2=‘00’，旧 PIN 密文||MAC								
Le	不存在								

9.2.1.9.4 响应信息

响应信息中可能返回的状态码有：

SW1	SW2	说　明
90	00	命令执行成功
65	81	写 EEPROM 失败
67	00	Lc 长度错误
69	82	不满足安全状态
69	83	认证密钥锁定
69	84	引用数据无效（未申请随机数）
69	85	使用条件不满足
69	88	安全信息（MAC 和加密）数据错误
6A	81	功能不支持
6A	83	未找到 PIN
6A	86	P1、P2 参数错
6A	88	未找到密钥数据
6D	00	命令不存在
6E	00	CLA 错
93	03	应用永久锁定

9.2.1.9.5 命令详解

1、DES 算法

操作 1：　对 MF 的主 PIN 改写

条件：　MF 下主 PIN 已经锁定，即认证主 PIN 时返回’6983’。

步骤 1：　用 MF 主 PIN 信息中 UBK-KID 指向密钥 UBK（本例取：UBK，16 字节’00’）对旧 PIN 明文（本例取’0000’）加密，产生旧 PIN 密文。
3DES Encryption（UBK，’0000’， EncPIN）
过程密钥计算结果为：EncPIN = ’ C5D6090EFE1729BC’

步骤 2：　取 4 字节随机数 Random，后补’00000000’作为 MAC 计算初始值。

命令：　‘00840000 04’

响应：　Random=’72174890’

步骤 3：　用 MF 主 PIN 信息中 UBK-KID 指向密钥 UBK（本例取：UBK，16 字节’00’）计算 MAC。
初始值 InitialData=’ 7217489000000000’ 8 字节
命令头 HeadData= ‘842400010C’
MAC 计算数据为：MACData = InitialData + HeadData + EncPIN
3DES MAC（UBK，MACData，MAC）
MAC=’ 2C393066’

步骤 4：　修改 MF 下主 PIN

命令： ‘842400010C’ + ’C5D6090EFE1729BC’ + ‘2C393066’
响应： ‘9000’

操作 2： 对 MF 的个人 PIN 修改，PIN_ID=’01’
条件： MF 下主 PIN 已经锁定，即认证主 PIN 时返回’6983’。
步骤 1： 用 MF 主 PIN 信息中 UBK-KID 指向密钥 UBK（本例取：UBK，16 字节’00’）对旧 PIN 明文（本例取’2222’）加密，产生旧 PIN 密文。
3DES Encryption（UBK，’2222’，EncPIN）
过程密钥计算结果为：EncPIN = ’ 092F659EE37D9AAE’
步骤 2： 取 4 字节随机数 Random，后补’00000000’作为 MAC 计算初始值。
命令： ‘00840000 04’
响应： Random=’ 235FC22’
步骤 3： 用 MF 主 PIN 信息中 UBK-KID 指向密钥 UBK（本例取：UBK，16 字节’00’）计算 MAC。
初始值 InitialData=’ 235FC2200000000’ 8 字节
命令头 HeadData= ‘842400030C’
MAC 计算数据为：MACData = InitialData + HeadData + EncPIN
3DES MAC（UBK，MACData，MAC）
MAC=’ 5B77D063’
步骤 4： 修改 MF 下个人 PIN
命令： ‘842400030C’ + ’ 092F659EE37D9AAE’ + ‘5B77D063’
响应： ‘9000’

2、SM1 算法
操作 1： 对 MF 的主 PIN 改写
条件： MF 下主 PIN 已经锁定，即认证主 PIN 时返回’6983’。
步骤 1： 用 MF 主 PIN 信息中 UBK-KID 指向密钥 UBK（本例取：UBK，16 字节’11’）对旧 PIN 明文（本例取’1234’）加密，产生旧 PIN 密文。
SM1 Encryption（UBK，’1234’， EncPIN）
过程密钥计算结果为：EncPIN =
‘1A5A87D7FE196E8974E59D5D303A71A’
步骤 2： 取 8 字节随机数 Random，后补’0000000000000000’作为 MAC 计算初始值。
命令： ‘00840000 08’
响应： Random=’ 18A600247A56ADBF’
步骤 3： 用 MF 主 PIN 信息中 UBK-KID 指向密钥 UBK（本例取：UBK，16 字节’11’）计算 MAC。
初始值 InitialData= Random+’0000000000000000’ 8 字节
命令头 HeadData= ‘8424000014’

MAC 计算数据为：MACData = InitialData + HeadData + EncPIN
3DES MAC（UBK，MACData，MAC）
MAC=’8DCC0D45’

步骤 4： 修改 MF 下主 PIN
命令： ‘8424000114’ + ’1A5A87D7FE196E8974E59D5D303A71A’+‘8DCC0D45’
响应： ‘9000’

操作 2： 对 MF 的个人 PIN 修改，PIN_ID=’01’
条件： MF 下主 PIN 已经锁定，即认证主 PIN 时返回’6983’。
步骤 1： 用 MF 主 PIN 信息中 UBK-KID 指向密钥 UBK（本例取：UBK，16 字节’11’）对旧 PIN 明文（本例取’0000’）加密，产生旧 PIN 密文。
SM1 Encryption（UBK，’0000’，EncPIN）
过程密钥计算结果为：EncPIN = ’
782DE85F855C41876BB87052CEC9C17A’
步骤 2： 取 8 字节随机数 Random，后补’0000000000000000’作为 MAC 计算初始值。
命令： ‘00840000 08’
响应： Random=’ 8B0C179403372102’
步骤 3： 用 MF 主 PIN 信息中 UBK-KID 指向密钥 UBK（本例取：UBK，16 字节’11’）计算 MAC。
初始值 InitialData= Random+’0000000000000000’ 8 字节
命令头 HeadData= ‘8424000314’
MAC 计算数据为：MACData = InitialData + HeadData + EncPIN
SM1 MAC（UBK，MACData，MAC）
MAC=’ B9F0AD19’
步骤 4： 修改 MF 下个人 PIN
命令： ‘842400030C’ + ’ 782DE85F855C41876BB87052CEC9C17A’+
‘B9F0AD19’
响应： ‘9000’

9.2.1.10 READ BINARY 命令

9.2.1.10.1 命令描述

READ BINARY 命令用于读出透明文件的内容。

9.2.1.10.2 使用条件和安全

在个人化状态下，READ BINARY 命令的执行不需要满足文件的读条件和读属性，以明文方式读出；在应用状态下，READ BINARY 命令的执行必须满足相应文件的读条件和读属性。

9.2.1.10.3 命令格式

代码	数　值								
CLA	‘00’或‘04’								
INS	‘B0’								
P1	b8	b7	b6	b5	b4	b3	b2	b1	说　明
	0	x	x	x	x	x	x	x	当前文件高位地址
	1	0	0	x	x	x	x	x	通过 SFI 方式访问
P2	若 P1 的 b8=0，P2 为文件的低位地址 若 P1 的 b8=1，P2 为文件地址								
Lc	1）不存在——明文方式 2）‘04’——校验方式								
DATA	1）不存在 2）MAC								
Le	期望返回的数据长度								

可能的命令/响应有：

CER	CIPH	命令	响应
0	0	00 B0 P1 P2 Le	明文数据 || SW1 SW2
0	1	04 B0 P1 P2 Le	密文数据 || SW1 SW2
1	0	04 B0 P1 P2 Lc MAC Le	明文数据 || SW1 SW2
1	1	04 B0 P1 P2 Lc MAC Le	密文数据 || SW1 SW2

9.2.1.10.4 响应信息

响应信息中的数据为明文或密文数据。

响应信息中可能返回的状态码有：

SW1	SW2	说　明
90	00	命令执行成功

61	xx	还有 xx 字节要返回
62	81	部分回送的数据可能有错
62	82	文件长度<Le
65	81	写 EEPROM 失败
67	00	Lc 长度错误
69	81	当前文件不是透明文件
69	82	不满足安全状态
69	83	认证密钥锁定
69	84	引用数据无效（未申请随机数）
69	85	使用条件不满足
69	86	没有选择当前文件
69	88	安全信息（MAC 和加密）数据错误
6A	81	功能不支持
6A	82	未找到文件
6A	86	P1、P2 参数错
6A	88	未找到密钥数据
6B	00	起始地址超出范围
6C	xx	Le 长度错误。‘xx’表示实际长度
6D	00	命令不存在
6E	00	CLA 错
93	03	应用永久锁定

9.2.1.10.5 命令详解

1、DES 算法

条件 1：　文件标识符：’EF11’(透明文件)
　　　　　通过 Select File 方式访问
　　　　　明文方式读取
　　　　　最大记录长度为’FF’

操作：　读取透明文件中长度为’FF’个字节的数据。

步骤 1：　选择’EF11’

命令：　‘00A4000002EF11’

响应：　‘9000’

步骤 1：　明文写入数据

命令：　‘00B00000 FF’

响应：　‘00000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000

0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000’

条件 2：文件标识符：’EF12’ (透明文件)
通过 Select File 方式访问
MAC 方式读取(DES 算法)
最大数据长度为’FA’

操作：读取透明文件中长度为’FF’个字节的数据。

步骤 1：选择’EF12’

命令：‘00A4000002EF12’

响应：‘9000’

步骤 2：取 8 字节随机数 Random，作为过程密钥产生因子。

命令：‘00840000 08’

响应：Random = ’90051916EEDD261A’

步骤 3：用当前 DF 主控密钥 MK（本例取：16 字节’00’）计算 MAC
初始值 InitialData=’0000000000000000’ 8 字节
命令头 HeadData= ‘04B0000004’
MAC 计算数据为：MACData = InitialData + HeadData
DES MAC（MK，MACData，MAC）
MAC=’ 58FD32C1’

步骤 4：MAC 读取数据

命令：‘04B0000004’ + MAC + ’FF’

响应：‘0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000’

2、SM1 算法

条件 1：文件标识符：’EF11’(透明文件)
通过 Select File 方式访问

	明文方式读取 最大记录长度为’FF’
操作：	读取透明文件中长度为’FF’个字节的数据。
步骤 1：	选择’EF11’
命令：	‘00A4000002EF11’
响应：	‘9000’
步骤 1：	明文写入数据
命令：	‘00B00000 FF’
响应：	‘0000000000000000000000000000000000000000000000000000000000 0000000000000000000000000000000000000000000000000000000000 0000000000000000000000000000000000000000000000000000000000 0000000000000000000000000000000000000000000000000000000000 0000000000000000000000000000000000000000000000000000000000 0000000000000000000000000000000000000000000000000000000000 0000000000000000000000000000000000000000000000000000000000 0000000000000000000000000000000000000000000000000000000000 0000000000000000000000000000000000000000000000’

条件 2：	文件标识符：’EF12’ (透明文件) 通过 Select File 方式访问 MAC 方式读取(SM1 算法)
操作：	读取透明文件中长度为’6F’个字节的数据。
步骤 1：	选择’EF12’
命令：	‘00A4000002EF12’
响应：	‘9000’
步骤 2：	取 8 字节随机数 Random。
命令：	‘00840000 08’
响应：	Random = ’0DF898BB16B877DF ’
步骤 4：	计算 MAC 初始值 InitialData= Random 8 字节+’0000000000000000’ 命令头 HeadData= ‘04B0000004’ MAC 计算数据为：MACData = Random+InitialData + HeadData SM1 MAC（Key，MACData，MAC） MAC=’ AF2F6314’
步骤 5：	MAC 读取数据
命令：	‘04B0000004’ + MAC + ’6F’
响应：	‘0000000000000000000000000000000000000000000000000000000000 0000000000000000000000000000000000000000000000000000000000

000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000'

条件 3：文件标识符：'EF13' (透明文件)
通过 Select File 方式访问
密文方式读取(SM1 算法)

操作：读取透明文件中长度为'6F'个字节的数据。
步骤 1：选择'EF12'
命令：'00A4000002EF12'
响应：'9000'
步骤 2：读取数据。
命令：'00B00000 70'
响应：'6A413DB0EAC7761EA470708211FAB8A1CEE479B4B459804539FF5D4842015644CEE479B4B459804539FF5D4842015644CEE479B4B459804539FF5D4842015644CEE479B4B459804539FF5D4842015644CEE479B4B459804539FF5D4842015644CEE479B4B459804539FF5D4842015644'

条件 4：文件标识符：'EF14' (透明文件)
通过 Select File 方式访问
密文+MAC 方式读取(SM1 算法)

操作：读取透明文件中长度为'6F'个字节的数据。
步骤 1：选择'EF12'
命令：'00A4000002EF12'
响应：'9000'
步骤 2：取 8 字节随机数 Random。
命令：'00840000 08'
响应：Random = ' 0D3BC8AFC240CAFD '
步骤 3：计算 MAC
初始值 InitialData= Random 8 字节+'0000000000000000'
命令头 HeadData= '04B0000004'
MAC 计算数据为：MACData = Random+InitialData + HeadData
SM1 MAC（Key，MACData，MAC）
MAC=' E0B650B1'

9.2.1.11 READ RECORD 命令

9.2.1.11.1 命令描述

READ RECORD 命令读记录文件中指定的记录。即可通过指定记录号方式读取记录数据，也可通过记录标识符方式读取记录数据。当以 TLV 结构访问文件时，IC 卡视记录的第一个字节为‘T’，将整个记录返回。当文件的 ACr 中的 CER 为‘1’时，应先执行 GET CHALLENGE 命令。

9.2.1.11.2 使用条件和安全

在个人化状态下，READ RECORD 命令的执行不需要满足文件的读条件和读属性，以明文方式读出；在应用状态下，READ RECORD 命令的执行必须满足相应文件的读条件和读属性。

9.2.1.11.3 命令格式

代码	数　值								
CLA	‘00’或‘04’								
INS	‘B2’								
P1	记录号（‘00’表示当前记录） 记录标识符（‘00’表示按记录号指定第一条、最后一条、下一条、前一条）								
P2	b8	b7	b6	b5	b4	b3	b2	b1	说　明
	0	0	0	0	0	-	-	-	当前文件
	x	x	x	x	x	-	-	-	通过 SFI 方式访问
	1	1	1	1	1	-	-	-	保留
	-	-	-	-	-	1	x	x	P1 作为记录号
	-	-	-	-	-	1	0	0	P1 指定的记录号
	-	-	-	-	-	1	0	1	从 P1 到记录尾
	-	-	-	-	-	1	1	0	从记录尾到 P1
	-	-	-	-	-	0	x	x	P1 作为记录标识符
	-	-	-	-	-	0	0	0	P1 指向相同标识符的第一条
	-	-	-	-	-	0	0	1	P1 指向相同标识符的最后一条
	-	-	-	-	-	0	1	0	P1 指向相同标识符的下一条
	-	-	-	-	-	0	1	1	P1 指向相同标识符的前一条
	其他值								保留
Lc	1) 不存在——明文方式 2)‘04’—— 命令报文校验方式								
DATA	1）不存在——明文方式 2）MAC——校验方式								

Le	期望返回的记录数据

可能的命令/响应有：

CER	CIPH	命令	响应
0	0	00 B2 P1 P2 Le	明文数据 || SW1 SW2
0	1	04 B2 P1 P2 Le	密文数据 || SW1 SW2
1	0	04 B2 P1 P2 Lc MAC Le	明文数据 || SW1 SW2
1	1	04 B2 P1 P2 Lc MAC Le	密文数据 || SW1 SW2

9.2.1.11.4 响应信息

响应信息中的数据为明文或密文数据。

响应信息中可能返回的状态码有：

SW1	SW2	说　明
90	00	命令执行成功
61	xx	还有 xx 字节需要返回
62	81	回送的数据可能有错
64	00	标志状态位没变
65	81	写 EEPROM 失败
67	00	Lc 长度错误
69	81	当前文件不是记录文件
69	82	不满足安全状态
69	83	认证密钥锁定
69	84	引用数据无效(未申请随机数)
69	85	使用条件不满足
69	86	没有选择当前文件
69	88	安全信息（MAC 和加密）数据错误
6A	81	功能不支持
6A	82	未找到文件
6A	83	未找到记录
6A	85	Lc 与 TLV 结构不匹配
6A	86	P1、P2 参数错
6A	88	未找到密钥数据
6C	xx	Le 错误，‘xx’表示实际长度
61	xx	射频模式下，CASE4 的情况，Le 错误，‘xx’表示实际长度
6D	00	命令不存在
6E	00	CLA 错

93	03	应用永久锁定

9.2.1.11.5 命令详解

操作 1： 通过记录号读取记录文件’0016’当中的记录。

条件： 标识符为’0016’的记录文件中包含 6 条记录，采用 TLV 格式，
其中：
第 1 条记录为’1806111111111111’
第 2 条记录为’1806222222222222’
第 3 条记录为’1906333333333333’
第 4 条记录为’1806444444444444’
第 5 条记录为’2006555555555555’
第 6 条记录为’1806666666666666’
用铭文方式读取

方法 1： 通过 SFI 访问记录文件
通过记录号读取记录

命令： ‘00B201B4 08’

响应： ’1806111111111111’

方法 2： 通过 Select File 访问记录文件
通过记录号读取记录

步骤 1： 选择’0016’文件

命令： ‘00A40200020016’

响应： ‘9000’

步骤 2： 读取第 1 条记录

命令： ‘00B20104 08’

响应： ’1806111111111111’

操作 2： 通过记录标识符，读取记录文件’0016’当中的记录。。

条件： 标识符为’0016’的记录文件中包含 6 条记录，采用 TLV 格式，
其中：
第 1 条记录为’1806111111111111’
第 2 条记录为’1806222222222222’
第 3 条记录为’1906333333333333’
第 4 条记录为’1806444444444444’
第 5 条记录为’2006555555555555’
第 6 条记录为’1806666666666666’

方法 1： 通过 SFI 访问记录文件
通过记录标识符读取记录（P1 指向相同标识符的第一条）
即，读取标识符为’18’的第一条记录’ 1806111111111111’

命令： ‘00B218B0 08’

响应：　’1806111111111111’
方法 2：　通过 SFI 访问记录文件
通过记录标识符读取记录（P1 指向相同标识符的最后一条）
即，读取标识符为’18’的最后一条记录’ 1806666666666666’
命令：　‘00B218B1 08’
响应：　‘1806666666666666’
方法 3：　通过 SFI 访问记录文件
通过记录标识符读取记录（P1 指向相同标识符的前一条）
当前记录位置为标识符’18’的最后一条记录。
即，读取标识符为’18’最后一条记录的前一条相同标识符的记录（记录号为’4’）
‘1806444444444444’
命令：　‘00B218B3 08’
响应：　‘1806444444444444’
方法 3：　通过 SFI 访问记录文件
通过记录标识符读取记录（P1 指向相同标识符的下一条）
当前记录位置为第 4 条记录。
即，读取识符为’18’这条记录的下一条相同标识的记录（记录号为’6’）
‘1806FFFFFFFFFFFF’
命令：　‘00B218B2 08’
响应：　‘1806FFFFFFFFFFFF’

通过 Select File 访问记录文件后，读取相同标识符文件的操作略。

9.2.1.12 SELECT FILE 命令

9.2.1.12.1 命令描述

SELECT FILE 命令通过文件标识或应用名选择 IC 卡中的 MF、DDF、ADF 或 EF 文件。

9.2.1.12.2 使用条件和安全

SELECT FILE 命令无使用条件限制。该命令不能用于选择安全文件（SF）。

9.2.1.12.3 命令格式

代码	数　值
CLA	‘00’
INS	‘A4’
P1	‘00’通过 FID 选择 MF、DF、EF，当 Lc=‘00’时，选 MF ‘01’通过 FID 选择 DF ‘02’通过 FID 选择当前 DF 下的 EF ‘03’选择父目录（Lc=‘00’） ‘04’通过 DF 名选择应用
P2	‘00’ ‘02’选择下一个文件（P1=‘04’时）
Lc	P1=‘00’时，Lc=‘00’或‘02’ P1=‘01’~‘02’时，Lc=‘02’ P1=‘03’时，Lc=‘00’ P1=‘04’时，Lc=‘01’~‘10’
DATA	P1=‘00’ 时，不存在或 FID（2 字节） P1=‘01’ 时，FID（2 字节） P1=‘02’ 时，FID（2 字节） P1=‘03’ 时，不存在 P1=‘04’ 时，应用名（AID）
Le	FCI 文件的信息长度（选择 MF、DF、ADF 时）

9.2.1.12.4 响应信息

响应信息的结构：

PBOC2.0 规范：

MF/DDF：‘6F L {84 L {DF 名} A5 L {88 L {DIR-SFI} }9F0C L {FCI 文件内容}}’

ADF：　　6F L {84 L {DF 名} A5 L {9F080102}9F0C L {FCI 文件内容}}’

响应信息中可能返回的状态码有：

SW1	SW2	说　明

90	00	命令执行成功
62	83	选择文件无效
62	84	FCI 格式与 P2 指定的不符
64	00	标志状态位没变
67	00	Lc 长度错误
6A	81	功能不支持
6A	82	未找到文件
6A	86	P1、P2 参数错
6A	87	Lc 与 P1-P2 不匹配
6D	00	命令不存在
6E	00	CLA 错
93	03	应用永久锁定

9.2.1.12.5 命令详解

操作 1： 选择 MF

条件： MF 中的 DIR-SFI=03，MF 没有定义名称

命令： 00A4000000

响应： ‘6F1E8400A51A8801039F0C10001122334455667788
99AABBCCDDEEFF9F080102’

根据 FCI 信息格式进行数据分组：

T（Tag）	L（Len）	V（Value）
‘61’	‘1E’	‘8400A51A8801039F0C10001122334455667788 99AABBCCDDEEFF9F080102’
‘84’	‘00’	DF 名不存在
‘A5’	‘1A’	‘8801039F0C10001122334455667788 99AABBCCDDEEFF’
‘88’	‘01’	‘03’（目录基本文件（DIR 文件）的 SFI）
‘9F0C’	‘10’	‘00112233445566778899AABBCCDDEEFF’（FCI 文件内容）
‘9F08’	‘01’	‘02’ 版本信息

操作 2： 选择 MF 下的 ADF

条件： 当前环境在 MF 下，选择 MF 下文件标识符为’ADF2’的 ADF 文件。
该 ADF 中包含 FCI 文件，ADF 名为’ D15600000501’

命令： ‘00A4010002ADF2’

响应： ‘6F198406D15600000501A50F9F0C08112233445566778 89F080102’

T（Tag）	L（Len）	V（Value）
‘6F’	‘19’	‘8406D15600000501A50F9F0C08 1122334455667788 9F080102’
‘84’	‘06’	‘D15600000501’

‘A5’	‘0F’	9F0C0811223344556677889F080102
‘9F0C’	‘08’	‘1122334455667788’
‘9F08’	‘01’	‘02’

操作 2： 选择 EF

条件： 选择当前 ADF 下文件标识符为’EF21’的 EF 文件。

命令： ‘00A4020002EF21’

响应： ‘9000’

操作 3： 选择当前 ADF 的父目录 MF

条件： 当前的 ADF 的父目录为 MF。此 MF 中的 DIR-SFI=03，MF 没有定义名称。

命令： ‘00A4030000’

响应： ‘6F1E8400A51A8801039F0C10001122334455667788
99AABBCCDDEEFF9F080102’

操作 4： 在 MF 下用应用名称选择 ADF

条件： 当前环境在 MF 下，选择 MF 下文件名称为‘D15600000501’的 ADF 文件。

命令： ‘00A4040006’ + ‘D15600000501’

响应： ‘6F198406D15600000501A50F9F0C0811223344556677889F080102’

9.2.1.13 UPDATE BINARY 命令

9.2.1.13.1 命令描述

UPDATE BINARY 命令用于更新透明文件中的数据。当文件的 ACw 中的 CER 为‘1’时，应先执行 GET CHALLENGE 命令。

9.2.1.13.2 使用条件和安全

在个人化状态下，UPDATE BINARY 命令的执行不需要满足文件的改写条件，并且以明文方式写入；在应用状态下，UPDATE BINARY 命令的执行必须满足相应文件的改写属性。

说明：

若连续三次执行此命令失败，IC 卡将永久锁定此应用并返回状态码‘9303’。

9.2.1.13.3 命令格式

代码	数　值								
CLA	‘00’或‘04’								
INS	‘D6’								
P1	b8	b7	b6	b5	b4	b3	b2	b1	说　明
	0	x	x	x	x	x	x	x	当前文件高位地址
	1	0	0	x	x	x	x	x	通过 SFI 方式访问
P2	若 P1 的 b8=0，P2 为文件的低位地址 若 P1 的 b8=1，P2 为文件地址								
Lc	DATA 域数据长度 DES： **明文方式：** ‘00’< Lc ≤‘FF’ **加密方式：** ‘08’≤ Lc ≤‘F8’（模 8） **校验方式：** ‘04’< Lc ≤‘FE’ **校验加密方式：**‘0C’≤ Lc ≤‘FC’（模 8 + 4） SM1： **明文方式：** ‘00’< Lc ≤‘FF’ **加密方式：** ‘10’≤ Lc ≤‘F0’（模 16） **校验方式：** ‘04’< Lc ≤‘FF’ **校验加密方式：**‘14’≤ Lc ≤‘F4’（模 16+ 4）								
DATA	明文方式：　明文数据 加密方式：　密文数据 校验方式：　明文数据||校验码 校验加密方式：密文数据||校验码								

Le	不存在

9.2.1.13.4　响应信息

响应信息中可能返回的状态码有：

SW1	SW2	说　明
90	00	命令执行成功
65	81	写 EEPROM 失败
67	00	Lc 长度错误
69	81	当前文件不是透明文件
69	82	不满足安全状态
69	83	认证密钥锁定
69	84	引用数据无效（未申请随机数）
69	85	使用条件不满足
69	86	未选择文件
69	88	安全信息（MAC 和加密）数据错误
6A	81	功能不支持
6A	82	未找到文件
6A	86	P1、P2 参数错
6A	88	未找到密钥数据
6B	00	起始地址超出范围
6D	00	命令不存在
6E	00	CLA 错
93	03	应用永久锁定

9.2.1.13.5　命令详解

1、DES 算法

条件 1：　文件标识符：’EF11’(透明文件)
　　　　　通过 Select File 方式访问
　　　　　明文方式写入
　　　　　最大记录长度为’FF’

操作：　　向透明文件中添加长度为’FF’个字节的数据。
　　　　　‘0000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000

000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000’为记录内容 DataIn

步骤 1：　选择’EF11’
命令：　‘00A4000002EF11’
响应：　‘9000’
步骤 2：　明文写入数据
命令：　‘00D60000FF’ + DataIn
响应：　‘9000’

条件 2：　文件标识符：’EF12’ (透明文件)
通过 Select File 方式访问
明文+MAC 方式写入(DES 算法)
详解中使用的数据长度为’6F’

操作：　向透明文件中添加长度为’6F’个字节的数据。
‘00000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000’为记录内容 DataIn

步骤 1：　选择’EF12’
命令：　‘00A4000002EF12’
响应：　‘9000’
步骤 2：　取 8 字节随机数 Random，作为 MAC 计算初始值 InitialData。
命令：　‘00840000 08’
响应：　Random
步骤 3：　用应用维护密钥 DAMK（本例取 16 字节’00’）计算 MAC
初始值 InitialData= Random 8 字节+’0000000000000’
命令头 HeadData= ‘04D6000073’
MAC 计算数据为：MACData = InitialData + HeadData + DataIn
3DES MAC（DAMK，MACData，MAC）
MAC=’609B361C’
步骤 4：　明文 + MAC 写入数据
命令：　‘04D6000073’ + DataIn + MAC
响应：　‘9000’

条件 3：　文件标识符：’EF13’ (透明文件)
通过 Select File 方式访问
密文方式写入(DES 算法)
详解中使用的数据长度为’6F’

操作： 向透明文件中添加长度为’6F’个字节的数据。
‘00’为记录内容 DataIn

步骤 1： 选择’EF13’

命令： ‘00A4000002EF13’

响应： ‘9000’

步骤 2： 用应用维护密钥 DAMK（本例取 16 字节’00’）计算密文
3DES Encryption（DAMK，(Ld+DataIn)，EncData）
密文EncData =
‘E02BA426582B63318CA64DE9C1B123A78CA64DE9C1B123A78CA64DE9C1B123A78CA64DE9C1B123A78CA64DE9C1B123A78CA64DE9C1B123A78CA64DE9C1B123A78CA64DE9C1B123A78CA64DE9C1B123A78CA64DE9C1B123A78CA64DE9C1B123A78CA64DE9C1B123A78CA64DE9C1B123A7’

步骤 3： 密文写入数据

命令： ‘04D6000070’ + EncData

响应： ‘9000’

条件 4： 文件标识符：’EF14’ (透明)
通过 Select File 方式访问
密文+MAC 方式写入(DES 算法)
详解中使用的数据长度为’6F’

操作： 向透明文件中添加长度为’6F’个字节的记录。
‘00’为记录内容 DataIn

步骤 1： 选择’EF14’

命令： ‘00A4000002EF14’

响应： ‘9000’

步骤 2： 用应用维护密钥 DAMK（本例取 16 字节’00’）计算密文
3DES Encryption（DAMK，(Ld+DataIn)，EncData）
密文EncData =
‘E02BA426582B63318CA64DE9C1B123A78CA64DE9C1B123A78CA64DE9C1B123A78CA64DE9C1B123A78CA64DE9C1B123A78CA64DE9C1B123A78CA64DE9C1B123A78CA64DE9C1B123A78CA64DE9C1B123A78C

A64DE9C1B123A78CA64DE9C1B123A78CA64DE9C1B123A78CA64DE9C1B123A7’

步骤 3： 取 8 字节随机数 Random，作为 MAC 计算初始值 InitialData。
命令： ‘00840000 08’
响应： Random
步骤 4： 用应用维护密钥 DAMK（本例取 16 字节’00’）计算 MAC
初始值 InitialData= Random 8 字节+’0000000000000000’
命令头 HeadData= ‘04D6000074’
MAC 计算数据为：MACData = InitialData + HeadData + EncData
3DES MAC（DAMK，MACData，MAC）
MAC=’7E1F960D’
步骤 5： 密文+MAC 写入数据
命令： ‘04D6000074’ + EncData + MAC
响应： ‘9000’

2、SM1 算法

条件 1： 文件标识符：’EF11’(透明文件)
通过 Select File 方式访问
明文方式写入
最大记录长度为’FF’

操作： 向透明文件中添加长度为’FF’个字节的数据。
‘00’为记录内容 DataIn

步骤 1： 选择’EF11’
命令： ‘00A4000002EF11’
响应： ‘9000’
步骤 2： 明文写入数据
命令： ‘00D60000FF’ + DataIn
响应： ‘9000’

条件 2： 文件标识符：’EF12’ (透明文件)
通过 Select File 方式访问

明文+MAC 方式写入(SM1 算法)
详解中使用的数据长度为’6F’

操作：向透明文件中添加长度为’6F’个字节的数据。
‘00000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000’为记录内容 DataIn

步骤 1：选择’EF12’
命令：‘00A4000002EF12’
响应：‘9000’
步骤 2：取 8 字节随机数 Random，作为 MAC 计算初始值 InitialData。
命令：‘00840000 08’
响应：Random=’7E1E9F84D52CF83F’
步骤 3：用应用维护密钥 DAMK（本例取 16 字节’00’）计算 MAC
初始值 InitialData= Random 8 字节+’0000000000000’
命令头 HeadData= ‘04D6000073’
MAC 计算数据为：MACData = InitialData + HeadData + DataIn
SM1 MAC（DAMK，MACData，MAC）
MAC=’20BADF7E’
步骤 4：明文 + MAC 写入数据
命令：‘04D6000073’ + DataIn + MAC
响应：‘9000’

条件 3：文件标识符：’EF13’ (透明文件)
通过 Select File 方式访问
密文方式写入(SM1 算法)
详解中使用的数据长度为’6F’

操作：向透明文件中添加长度为’6F’个字节的数据。
‘00000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000’为记录内容 DataIn

步骤 1：选择’EF13’
命令：‘00A4000002EF13’
响应：‘9000’
步骤 2：用应用维护密钥 DAMK（本例取 16 字节’00’）计算密文
SM1 Encryption（DAMK，(Ld+DataIn)，EncData）
密文EncData =

‘6A413DB0EAC7761EA470708211FAB8A1CEE479B4B459804539FF5D4842015644CEE479B4B459804539FF5D4842015644CEE479B4B459804539FF5D4842015644CEE479B4B459804539FF5D4842015644CEE479B4B459804539FF5D4842015644CEE479B4B459804539FF5D4842015644’

步骤 3：密文写入数据
命令：‘04D6000070’ + EncData
响应：‘9000’

条件 4：文件标识符：’EF14’ (透明)
通过 Select File 方式访问
密文+MAC 方式写入(SM1 算法)
详解中使用的数据长度为’6F’

操作：向透明文件中添加长度为’6F’个字节的记录。
‘00’为记录内容 DataIn
步骤 1：选择’EF14’
命令：‘00A4000002EF14’
响应：‘9000’
步骤 2：用应用维护密钥 DAMK（本例取 16 字节’00’）计算密文
SM1Encryption（DAMK，(Ld+DataIn)，EncData）
密文 EncData =
‘6A413DB0EAC7761EA470708211FAB8A1CEE479B4B459804539FF5D4842015644CEE479B4B459804539FF5D4842015644CEE479B4B459804539FF5D4842015644CEE479B4B459804539FF5D4842015644CEE479B4B459804539FF5D4842015644CEE479B4B459804539FF5D4842015644’
步骤 3：取 8 字节随机数 Random，作为 MAC 计算初始值 InitialData。
命令：‘00840000 08’
响应：Random=’ AE7468FF4CF37F1D’
步骤 4：用应用维护密钥 DAMK（本例取 16 字节’00’）计算 MAC
初始值 InitialData= Random 8 字节+’0000000000000000’
命令头 HeadData= ‘04D6000074’
MAC 计算数据为：MACData = InitialData + HeadData + EncData
SM1 MAC（DAMK，MACData，MAC）
MAC=’ 085EE51D’
步骤 5：密文+MAC 写入数据
命令：‘04D6000074’ + EncData + MAC

响应：　‘9000’

9.2.1.14 UPDATE RECORD 命令

9.2.1.14.1 命令描述

UPDATE RECORD 命令用于更新记录文件中的数据。当文件的 ACw 中的 CER 为‘1’时，应先执行 GET CHALLENGE 命令。更新的记录长度必须等于原记录长度。循环定长记录文件的添加也可采用 P2=‘xxxxx011’方式实现。

9.2.1.14.2 使用条件和安全

在个人化状态下，UPDATE RECORD 命令的执行不需要满足文件的改写条件，并且以明文方式写入；在应用状态下，UPDATE RECORD 命令的执行必须满足相应文件的改写条件和改写属性。

9.2.1.14.3 命令格式

<table>
<tr><td>代码</td><td colspan="9">数　值</td></tr>
<tr><td>CLA</td><td colspan="9">‘00’或‘04’</td></tr>
<tr><td>INS</td><td colspan="9">‘DC’</td></tr>
<tr><td>P1</td><td colspan="9">记录号（‘00’表示当前记录）
记录标识符（‘00’表示按记录号指定第一条、最后一条、下一条、前一条）</td></tr>
<tr><td rowspan="12">P2</td><td>b8</td><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>说　明</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>-</td><td>-</td><td>-</td><td rowspan="3">当前文件
通过 SFI 方式访问
保留</td></tr>
<tr><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>1</td><td>x</td><td>x</td><td>P1 作为记录号</td></tr>
<tr><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>1</td><td>0</td><td>0</td><td>P1 指定的记录号</td></tr>
<tr><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>0</td><td>x</td><td>x</td><td>P1 作为记录标识符</td></tr>
<tr><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>0</td><td>0</td><td>0</td><td>P1 指向相同标识符的第一条</td></tr>
<tr><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>0</td><td>0</td><td>1</td><td>P1 指向相同标识符的最后一条</td></tr>
<tr><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>0</td><td>1</td><td>0</td><td>P1 指向相同标识符的下一条</td></tr>
<tr><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>0</td><td>1</td><td>1</td><td>P1 指向相同标识符的前一条</td></tr>
<tr><td colspan="8">任何其他值</td><td>保留</td></tr>
<tr><td>Lc</td><td colspan="9">DATA 域数据长度
DES：
明文方式：　‘00’< Lc ≤‘FF’
加密方式：　‘08’≤ Lc ≤‘F8’（模 8）
校验方式：　‘04’< Lc ≤‘FF’
校验加密方式：‘0C’≤ Lc ≤‘FC’（模 8 + 4）</td></tr>
</table>

	SM1： **明文方式：** ‘00’< Lc ≤‘FF’ **加密方式：** ‘10’≤ Lc ≤‘F0’（模 16） **校验方式：** ‘04’< Lc ≤‘FF’ **校验加密方式：**‘14’≤ Lc ≤‘F4’（模 16+ 4） 新数据的长度必须等于原记录长度
DATA	明文方式： 明文记录数据 加密方式： 密文记录数据 校验方式： 明文记录数据||校验码 校验加密方式：密文记录数据|校验码
Le	不存在

9.2.1.14.4 响应信息

响应信息中可能返回的状态码有：

SW1	SW2	说 明
90	00	命令执行成功
65	81	写 EEPROM 失败
67	00	Lc 长度错误
69	81	当前文件不是记录文件
69	82	不满足安全状态
69	83	认证密钥锁定
69	84	引用数据无效（未申请随机数）
69	85	使用条件不满足
69	86	未选择文件
69	88	安全信息（MAC 和加密）数据错误
6A	81	功能不支持
6A	82	未找到文件
6A	83	未找到记录
6A	84	存储空间不够
6A	85	Lc 与 TLV 结构不匹配
6A	86	P1、P2 参数错
6A	88	未找到密钥数据
6D	00	命令不存在
6E	00	CLA 错
93	03	应用永久锁定

9.2.1.14.5 命令详解

操作 1： 通过记录号更新记录文件’0016’当中的记录。

条件：　该记录采用 TLV 格式，更新后记录的内容为’1806111111111111’，
用明文方式写入。
方法 1：　通过 SFI 访问记录文件
通过记录号更新记录
命令：　‘00DC01B4081806111111111111’
响应：　‘9000’
方法 2：　通过 Select File 访问记录文件
通过记录号更新记录
步骤 1：　选择’0016’文件
命令：　‘00A40200020016’
响应：　‘9000’
步骤 2：　更新第 1 条记录
命令：　‘00DC0104081806111111111111’
响应：　‘9000’

操作 2：　通过记录标识符，更新记录文件’0016’当中的记录。。
条件：　标识符为’0016’的记录文件中包含 6 条记录，采用 TLV 格式，
其中：
第 1 条记录为’1806111111111111’
第 2 条记录为’1806222222222222’
第 3 条记录为’1906333333333333’
第 4 条记录为’1806444444444444’
第 5 条记录为’2006555555555555’
第 6 条记录为’1806666666666666’
方法 1：　通过 SFI 访问记录文件
通过记录标识符更新记录（P1 指向相同标识符的第一条）
即，将标识符为’18’的第一条记录’ 1806111111111111’更新为：
‘1806AAAAAAAAAAAA’
命令：　‘00DC18B0081806AAAAAAAAAAAA’
响应：　‘9000’
方法 2：　通过 SFI 访问记录文件
通过记录标识符更新记录（P1 指向相同标识符的最后一条）
即，将标识符为’18’的最后一条记录’ 1806666666666666’更新为：
‘1806FFFFFFFFFFFF’
命令：　‘00DC18B1081806FFFFFFFFFFFF’
响应：　‘9000’
方法 3：　通过 SFI 访问记录文件
通过记录标识符更新记录（P1 指向相同标识符的前一条）

当前记录位置为标识符’18’的最后一条记录。
即，将标识符为’18’最后一条记录的前一条相同标识符的记录（记录号为’4’）
‘1806444444444444’更新为：‘1806DDDDDDDDDDD’
命令： ‘00DC18B3081806DDDDDDDDDDD’
响应： ‘9000’
方法 3： 通过 SFI 访问记录文件
通过记录标识符更新记录（P1 指向相同标识符的下一条）
当前记录位置为第 4 条记录。
即，将标识符为’18’这条记录的下一条相同标识的记录（记录号为’6’）
‘1806FFFFFFFFFFF’更新为：‘1806EEEEEEEEEEE’
命令： ‘00DC18B2081806EEEEEEEEEEE’
响应： ‘9000’

通过 Select File 访问记录文件后，更新相同标识符文件的操作略。

9.2.1.15 VERIFY PIN 命令

9.2.1.15.1 命令描述

VERIFY PIN 命令的目的是 IC 卡验证终端提供的 PIN。

9.2.1.15.2 使用条件和安全

执行 VERIFY PIN 命令前，必须满足 PIN 的访问权限。PIN 验证失败，其计数器减 1，当计数器为‘0’值时，PIN 被锁定。

9.2.1.15.3 命令格式

代码	数 值								
CLA	‘00’								
INS	‘20’								
P1	‘00’								
P2	b8	b7	b6	b5	b4	b3	b2	b1	说 明
	0	-	-	-	-	-	-	-	MF 或 DDF 下的 PIN
	1	-	-	-	-	-	-	-	ADF 下的 PIN
	-	x	x	x	x	x	x	x	口令标识
	0	0	0	0	0	0	0	0	MPIN
Lc	PIN 长度： ‘02’—‘06’								
DATA	PIN 数据								
Le	不存在								

9.2.1.15.4 响应信息

响应信息中可能返回的状态码有：

SW1	SW2	说 明
90	00	命令执行成功
63	Cx	验证失败。‘x’表示可以重试的次数
64	00	标志状态位没变
65	81	写 EEPROM 失败
67	00	Lc 长度错误
69	82	不满足安全状态
69	83	认证 PIN 锁定
69	84	记录空间已满
69	85	使用条件不满足
6A	81	功能不支持
6A	86	P1、P2 参数错
6A	88	未找到 PIN 数据
6D	00	命令不存在
6E	00	CLA 错

93	03	应用永久锁定

9.2.1.15.5 命令详解

操作 1： 认证主 PIN

条件： 在当前 MF 下认证主 PIN=‘0000”。

命令： ‘00200000020000’

响应： ‘9000’

操作 2： 认证个人 PIN

条件： 在当前 MF 下的个人 PIN=’0000’，PIN 的标识符为’01’

命令： ‘00200001020000’

响应： ‘9000’

操作 3： 当前 ADF 下的个人 PIN

条件： 在当前 ADF 下的 PIN=‘0000’，PIN 的标识符为’01’。

命令： ‘00200081020000’

响应： ‘9000’

9.2.2 金融专用命令

9.2.2.1 CHANGE PIN 命令

9.2.2.1.1 命令描述

CHANGE PIN 命令允许持卡人将当前个人密码更新为新的密码。可将有效的 PIN 更新为缺省 PIN，或将缺省 PIN 更新为有效 PIN。

9.2.2.1.2 使用条件和安全

更新 PIN 命令中的数据域是以明文方式传送的。命令不受 PIN 的访问权限的限制。更新 PIN 时，要通过原 PIN 的认证。如果原 PIN 认证没有通过，不能完成 PIN 的更新，同时将 PIN 的限制计数器减一。

9.2.2.1.3 命令格式

代码	值								
CLA	‘80’								
INS	‘5E’								
P1	‘01’								
P2	b8	b7	b6	b5	b4	b3	b2	b1	说　明
	0	-	-	-	-	-	-	-	MF 或 DDF 下的 PIN
	1	-	-	-	-	-	-	-	ADF 下的 PIN
	-	x	x	x	x	x	x	x	PIN 标识符
	0	0	0	0	0	0	0	0	MPIN
Lc	‘05’—‘0D’								
Data	当前 PIN ||‘FF’|| 新的 PIN								
Le	不用								

9.2.2.1.4 响应信息

响应信息中可能返回的状态码有：

SW1	SW2	含义
90	00	命令执行成功
63	Cx	验证失败，’x’ 表示重试次数
65	81	EEPROM 损坏，导致卡锁定
67	00	Lc 长度错
69	83	验证 PIN 锁定
69	85	使用条件不满足
6A	80	数据域参数不正确
6A	81	功能不支持
6A	86	P1，P2 参数不正确

6A	88	未找到 PIN 数据
6D	00	命令不存在
6E	00	命令类型错
93	03	应用永久锁定

9.2.2.1.5 命令详解

操作 1： MPIN 修改。

条件： 旧 PIN：’0000’，新 PIN：’1234’

命令： ‘805E010005’ + ‘0000FF1234’

响应： ‘9000’

操作 2： 标识为’01’的个人 PIN 修改。

条件： 旧 PIN：’0000’，新 PIN：’1234’

命令： ‘805E010105’ + ‘0000FF1234’

响应： ‘9000’

9.2.2.2 CREDIT FOR LOAD 命令

9.2.2.2.1 命令描述

CREDIT FOR LOAD 命令用于金融圈存交易。

9.2.2.2.2 使用条件和安全

在执行 CREDIT FOR LOAD 命令之前，应先成功执行 INITIALIZE FOR LOAD 命令。

9.2.2.2.3 命令格式

代码	值
CLA	‘80’
INS	‘52’
P1	‘00’
P2	‘00’
Lc	‘0B’
Data	交易日期（4 字节） 交易时间（3 字节） MAC2（4 字节）
Le	‘04’（TAC）

计算 MAC2 的数据包括：

—交易金额　　4 字节
—交易类型　　1 字节
—终端机编号　　6 字节
—交易日期　　4 字节
—交易时间　　3 字节

9.2.2.2.4 响应信息

命令执行成功返回的数据包括以下内容：

—TAC　　4 字节

计算 TAC 的数据包括：
—新余额　　4 字节
—联机交易序号（加 1 前）2 字节
—交易金额　　4 字节
—交易类型　　1 字节
—终端机编号　　6 字节

—交易日期　　4 字节
—交易时间　　3 字节

命令执行不成功，则只返回状态码 SW1 和 SW2。

SW1	SW2	含义
90	00	命令执行成功
61	xx	正常处理。‘xx’表示可以通过后续 GET RESPONSE 命令得到的额外数据长度
65	81	EEPROM 损坏，导致卡锁定
67	00	Lc 长度错
69	01	命令不接受（无效状态）
69	85	使用条件不满足
6A	81	功能不支持
6A	86	参数 P1、P2 不正确
6D	00	命令不存在
6E	00	命令类型错
93	02	MAC 无效
93	03	应用被永久锁定

9.2.2.2.5 命令详解

操作：　完成圈存操作
条件：　已经成功执行过 INITIALIZE FOR LOAD 命令
　　终端当前的日期：2007 ，10， 22
　　终端当前的时间：18 ：18 ：18
　　通过圈存交易中的过程密钥计算得到的 MAC2 ：’E2B9F3D2’
　　（MAC 计算参照“安全管理”一章中金融环境下交易 MAC 的计算）
命令：　‘805200000B20071022181818E2B9F3D2’
响应：　返回 TAC ：’45DD1FB8’
（TAC 计算参照“安全管理”一章中金融环境下交易 MAC 的计算，使用 TAC 密钥）

9.2.2.3 DEBIT FOR PURCHASE/CASH WITHDRAW 命令

9.2.2.3.1 命令描述

DEBIT FOR PURCHASE/CASH WITHDRAW 命令用于金融消费/取现交易。

9.2.2.3.2 使用条件和安全

执行 DEBIT FOR PURCHASE/CASH WITHDRAW 命令之前，应先成功执行 INITIALIZE FOR PURCHASE 命令或 INITIALIZE FOR CASH WITHDRAW 命令。

9.2.2.3.3 命令格式

代码	值
CLA	‘80’
INS	‘54’
P1	‘01’
P2	‘00’
Lc	‘0F’
Data	终端交易序号（4 字节） 交易日期（4 字节） 交易时间（3 字节） MAC1（4 字节）
Le	‘08’（TAC＋MAC2）

计算 MAC1 的数据包括：

—交易金额　　4 字节
—交易类型　　1 字节
—终端机编号　　6 字节
—交易日期　　4 字节
—交易时间　　3 字节

9.2.2.3.4 响应信息

命令执行成功返回的数据包括以下内容：

—TAC　　4 字节
—MAC2　　4 字节

计算 TAC 的数据包括：

—交易金额　　4 字节
—交易类型　　1 字节
—终端机编号　　6 字节
—终端交易序号　　2 字节

—交易日期　　4 字节
—交易时间　　3 字节

计算 MAC2 的数据包括：
—交易金额　　4 字节

命令执行不成功，则只返回状态码 SW1 和 SW2。

SW1	SW2	含义
90	00	命令执行成功
61	xx	正常处理。‘xx’表示可以通过后续 GET RESPONSE 命令得到的额外数据长度
65	81	EEPROM 损坏，导致卡锁定
67	00	Lc 长度错
69	01	命令不接受（无效状态）
69	85	使用条件不满足
6A	81	功能不支持
6A	86	参数 P1、P2 不正确
6D	00	命令不存在
6E	00	命令类型错
93	02	MAC 无效
93	03	应用被永久锁定

9.2.2.3.5 命令详解

操作：　完成消费或取现操作
条件 1：　已经成功执行过 INITIALIZE FOR PURCHASE 命令
　　得到终端交易序号 ：’00000001’
　　终端当前的日期：2007 ，10， 22
　　终端当前的时间：18 ： 18 ： 18
　　通过消费交易中的过程密钥计算得到的 MAC1 ：’25B0E9A’
　　（MAC 计算参照“安全管理”一章中金融环境下交易 MAC 的计算）
命令：　‘805401000F0000000120071022181818 25B0E9A’
响应：　返回 TAC ： ‘F4A51EFF’
（TAC 计算参照“安全管理”一章中金融环境下交易 MAC 的计算，使用 TAC 密钥）
返回 MAC2：’66693766’（'MAC 计算参照“安全管理”一章中金融环境下交易 MAC 的计算，使用消费交易中的过程密钥）
条件 2：　已经成功执行过 INITIALIZE FOR CASH WITHDRAW 命令
　　得到终端交易序号 ：’00000001’
　　终端当前的日期：2007 ，10， 22
　　终端当前的时间：18 ： 18 ： 18

通过取现交易中的过程密钥计算得到的 MAC1 ：’B3C28CCD’
（MAC 计算参照“安全管理”一章中金融环境下交易 MAC 的计算）
命令： ‘805401000F0000000120071022181818B3C28CCD’
响应： 返回 TAC ：’CF084AA3’
（TAC 计算参照“安全管理”一章中金融环境下交易 MAC 的计算，使用 TAC 密钥）
返回 MAC2 ：0F3676BA
（MAC 计算参照“安全管理”一章中金融环境下交易 MAC 的计算，使用消费交易中的过程密钥）

9.2.2.4 DEBIT FOR UNLOAD 命令

9.2.2.4.1 命令描述

DEBIT FOR UNLOAD 命令用于金融圈提交易。

9.2.2.4.2 使用条件和安全

在执行 DEBIT FOR UNLOAD 命令之前， 应先成功执行 INITIALIZE FOR UNLOAD 命令。

9.2.2.4.3 命令报文

代码	值
CLA	‘80’
INS	‘54’
P1	‘03’
P2	‘00’
Lc	‘0B’
Data	交易日期（4 字节） 交易时间（3 字节） MAC2（4 字节）
Le	‘04’（MAC3）

计算 MAC2 的数据包括：

—交易金额 4 字节
—交易类型 1 字节
—终端机编号 6 字节
—交易日期 4 字节
—交易时间 3 字节

9.2.2.4.4 响应信息

命令执行成功返回的数据包括以下内容：

—MAC3 4 字节

计算 MAC3 的数据包括：

—新余额 4 字节
—联机交易序号（加 1 前） 2 字节
—交易金额 4 字节
—交易类型 1 字节
—终端机编号 6 字节
—交易日期 4 字节
—交易时间 3 字节

命令执行不成功，则只返回状态码 SW1 和 SW2。

SW1	SW2	含义
90	00	命令执行成功
61	xx	正常处理。‘xx’表示可以通过后续 GET RESPONSE 命令得到的额外数据长度
65	81	EEPROM 损坏，导致卡锁定
67	00	Lc 长度错
69	01	命令不接受（无效状态）
6A	81	功能不支持
6A	86	参数 P1、P2 不正确
6D	00	命令不存在
6E	00	命令类型错
93	02	MAC 无效
93	03	应用被永久锁定

9.2.2.4.5 命令详解

操作： 完成圈提操作

条件： 已经成功执行过 INITIALIZE FOR UNLOAD 命令
终端当前的日期：2007 ，10， 22
终端当前的时间：18 ： 18 ： 18
通过圈提交易中的过程密钥计算得到的 MAC2 ：’453BC1F3’
（MAC 计算参照“安全管理”一章中金融环境下交易 MAC 的计算）

命令： ‘805403000B20071022181818743BC564’

响应： 返回 MAC3：’FDF4CD94’

（MAC 计算参照“安全管理”一章中金融环境下交易 MAC 的计算，使用圈提交易中的过程密钥）

9.2.2.5 GET BALANCE 命令

9.2.2.5.1 命令描述

GET BALANCE 命令用于查询电子存折或电子钱包余额。在电子存折余额中包括透支额。

9.2.2.5.2 使用条件和安全

读取电子钱包，电子存折余额时需验证个人密码（PIN）。

9.2.2.5.3 命令格式

代码	值
CLA	‘80’
INS	‘5C’
P1	‘00’
P2	‘01’：用于 ED； ‘02’：用于 EP；
Lc	不存在
Data	不存在
Le	‘04’

9.2.2.5.4 响应信息

SW1	SW2	含义
90	00	命令执行成功
69	82	不满足安全状态
69	85	使用条件不满足
6A	81	功能不支持
6A	86	参数 P1，P2 不正确
6D	00	命令不存在
6E	00	命令类型错
93	03	应用被永久锁定

9.2.2.5.5 命令详解

操作：　得到电子存折或电子钱包的余额
条件 1：　查询电子存折余额，要求通过 PIN 验证
命令：　‘805C000104’
响应：　返回余额：’00002710’
条件 2：　查询电子钱包余额
命令：　‘805C000204’
响应：　返回余额：’00000000’

9.2.2.6 GET TRANSACTION PROOF 命令

9.2.2.6.1 命令描述

GET TRANSACTION PROVE 命令用于获取金融交易认证码（TAC 和 MAC）。

9.2.2.6.2 使用条件和安全

在金融交易被迫中断时（掉电或提前拔卡），IC 卡会保持金融交易瞬间所处的状态和重要数据。重新插卡时，如果被中断的金融交易已经完成，执行该命令时返回上次金融交易的 TAC 和 MAC；如果未完成，则执行该命令时返回状态码‘9406’（所需 MAC 不可用）。

卡片在应用开发状态时不提供金融交易保护功能。金融交易被中断后，重新插卡执行该命令 IC 卡返回状态码‘9406’。

9.2.2.6.3 命令格式

代码	值
CLA	‘80’
INS	‘5A’
P1	‘00’
P2	交易类型标识。
Lc	‘02’
Data	联机或脱机交易序号。
Le	‘08’

P2 交易类型：

01—ED 圈存
02—EP 圈存
03—圈提
04—ED 取款
05—ED 消费
06—EP 消费
07—ED 修改透支限额
08—信用消费

9.2.2.6.4 响应信息

命令执行成功返回的数据包括以下内容：

—MAC 4 字节
—TAC 4 字节

命令执行不成功，则只返回状态码 SW1 和 SW2。

SW1	SW2	含义

90	00	命令执行成功
61	xx	正常处理。‘xx’表示可以通过后续 GET RESPONSE 命令得到的额外数据长度
69	85	使用条件不满足
67	00	Le 长度错
6A	81	功能不支持
6A	86	P1 参数错
6D	00	命令不存在
6E	00	命令类型错
93	03	应用被永久锁定

9.2.2.6.5 命令详解

操作： 取得交易认证码

条件： ED 圈存交易类型标识： ‘01’
获得圈存联机交易序号：’0001’

命令： ‘805A0001020001’

响应： 返回 MAC ： ‘D1433650’
返回 TAC： ‘83270316’

9.2.2.7 INITIALIZE FOR CASH WITHDRAW 命令

9.2.2.7.1 命令描述

INITIALIZE FOR CASH WITHDRAW 命令用于金融初始化取现交易。命令执行后 IC 卡为金融取现交易状态。

9.2.2.7.2 使用条件和安全

INITIALIZE FOR CASH WITHDRAW 命令的执行仅对 DEBIT FOR PURCHASE/CASH WITHDRAW 命令有效。命令执行前，需要验证个人密码（PIN）。

9.2.2.7.3 命令格式

代码	值
CLA	‘80’
INS	‘50’
P1	‘02’
P2	‘01’：用于 ED 取现交易； 其它值保留。
Lc	‘0B’
Data	密钥索引号（1 字节） 交易金额（4 字节） 终端机编号（6 字节）
Le	‘0F’

9.2.2.7.4 响应信息

命令执行成功返回的数据包括以下内容：

—ED 余额 4 字节
—ED 脱机交易序号 2 字节
—透支限额 3 字节
—密钥版本号 1 字节
—算法标识 1 字节
—伪随机数 4 字节

命令执行不成功，则只返回状态码 SW1 和 SW2。

SW1	SW2	含义
90	00	命令执行成功
61	xx	正常处理。‘xx’表示可以通过后续 GET RESPONSE 命令得到的额外数据长度
65	81	EEPROM 损坏，导致卡锁定

67	00	Lc 长度错
69	82	不满足安全状态
69	85	使用条件不满足
6A	81	功能不支持
6A	86	参数 P1、P2 不正确
6D	00	命令不存在
6E	00	命令类型错
93	03	应用被永久锁定
94	01	金额不足
94	02	交易计数器达到最大值
94	03	密钥索引不支持

9.2.2.7.5 命令详解

操作： 用户卡获得取现初始化的状态

条件： 建立电子存折文件
通过 PIN 验证
取现密钥索引：’06’
获得交易金额：’00000008’
获得终端机编号 ：’130000000001’

命令： ‘805202010B01000003E8130000000001’

响应： 返回
ED 余额： ‘000007C8’
ED 脱机交易序号： ‘0001’
透支限额： ‘000000’
密钥版本号： ‘00’
算法标识： ‘00’
伪随机数： ‘68EB54FD’

9.2.2.8 INITIALIZE FOR LOAD 命令

9.2.2.8.1 命令描述

INITIALIZE FOR LOAD 命令用于金融初始化圈存交易。命令执行后 IC 卡为金融圈存交易状态。

9.2.2.8.2 使用条件和安全

INITIALIZE FOR LOAD 命令仅对 CREDIT FOR LOAD 命令有效。在 ED 操作前，需要验证个人密码（PIN）。

9.2.2.8.3 命令格式

代码	值
CLA	‘80’
INS	‘50’
P1	‘00’
P2	‘01’：ED 圈存 ‘02’：EP 圈存
Lc	‘0B’
Data	密钥索引号（1 字节） 交易金额（4 字节） 终端机编号（6 字节）
Le	‘10’

9.2.2.8.4 响应信息

命令执行成功返回的数据包括以下内容：

—ED 或 EP 余额 4 字节
—ED 或 EP 联机交易序号 2 字节
—密钥版本号 1 字节
—算法标识 1 字节
—伪随机数 4 字节
—MAC1 4 字节

计算 MAC1 的数据包括：

—旧余额 4 字节
—交易金额 4 字节
—交易类型 1 字节
—终端机编号 6 字节

命令执行不成功，则只返回状态码 SW1 和 SW2。

SW1	SW2	含义

90	00	命令执行成功
61	xx	正常处理。‘xx’表示可以通过后续 GET RESPONSE 命令得到的额外数据长度
65	81	EEPROM 损坏，导致卡锁定
67	00	Lc 长度错
69	82	不满足安全状态
69	85	使用条件不满足
6A	81	功能不支持
6A	86	参数 P1，P2 不正确
6D	00	命令不存在
6E	00	命令类型错
93	03	应用被永久锁定
94	02	交易计数器达到最大值
94	03	密钥索引不支持

9.2.2.8.5 命令详解

操作： 用户卡获得圈存初始化的状态

条件： 建立电子存折文件或电子钱包文件
对于电子存折文件，通过 PIN 验证
圈存密钥索引：’01’
获得交易金额：’000003E8’
获得终端机编号 ：’130000000001’

命令： ‘805000010B05000003E8130000000001’

响应： 返回
余额： ‘00000000’
联机交易序号： ‘0000’
密钥版本号： ‘00’
算法标识： ‘00’
伪随机数： ‘FE2C509C’
MAC1 ： ‘910B705F’

（MAC 计算参照“安全管理”一章中金融环境下交易 MAC 的计算，使用圈存过程密钥）

9.2.2.9 INITIALIZE FOR PURCHASE 命令

9.2.2.9.1 命令描述

INITIALIZE FOR PURCHASE 命令用于金融初始化消费交易。命令执行后 IC 卡为金融消费交易状态。

9.2.2.9.2 使用条件和安全

INITIALIZE FOR PURCHASE 命令仅 DEBIT FOR PURCHASE/CASH WITHDRAW 命令有效。在 ED 操作前，需要验证个人密码（PIN）。

9.2.2.9.3 命令格式

代码	值
CLA	‘80’
INS	‘50’
P1	‘01’
P2	‘01’：用于 ED； ‘02’：用于 EP；
Lc	‘0B’
Data	密钥索引号（1 字节） 交易金额（4 字节） 终端机编号（6 字节）
Le	‘0F’

9.2.2.9.4 响应信息

命令执行成功返回的数据包括以下内容：

—ED 余额 4 字节
—ED 脱机交易序号 2 字节
—透支限额 3 字节
—密钥版本号 1 字节
—算法标识 1 字节
—伪随机数 4 字节

命令执行不成功，则只返回状态码 SW1 和 SW2。

SW1	SW2	含义
90	00	执行命令成功
61	xx	正常处理。‘xx’表示可以通过后续 GET RESPONSE 命令得到的额外数据长度。
65	81	EEPROM 损坏，导致卡锁定

67	00	Lc 长度错
69	82	不满足安全状态
69	85	使用条件不满足
6A	81	功能不支持
6A	86	参数 P1、P2 不正确
6D	00	命令不存在
6E	00	命令类型错
93	03	应用被永久锁定
94	01	金额不足
94	02	交易计数器达到最大值
94	03	密钥索引不支持

9.2.2.9.5 命令详解

操作： 用户卡获得消费初始化的状态

条件 1： 建立有电子存折文件或电子钱包文件
对于电子存折文件，通过 PIN 验证
消费密钥索引：’01’
获得交易金额：’00000008’
获得终端机编号 ：’130000000001’

命令： ‘805001010B01000003E8130000000001’(电子存折)

响应： 返回
余额： ‘000007D0’
脱机交易序号： ‘0000’
透支限额： ‘000000’
密钥版本号： ‘00’
算法标识： ‘00’
伪随机数： ‘41DD741F’

9.2.2.10 INITIALIZE FOR UNLOAD 命令

9.2.2.10.1 命令描述

INITIALIZE FOR UNLOAD 命令用于金融初始化圈提交易。命令执行后 IC 卡为金融圈提交易状态。

9.2.2.10.2 使用条件和安全

INITIALIZE FOR UNLOAD 命令仅对 DEBIT FOR UNLOAD 命令有效。命令执行前，需要验证个人密码（PIN）。

9.2.2.10.3 命令报文

代码	值
CLA	‘80’
INS	‘50’
P1	‘05’
P2	‘01’：用于 ED 其它值保留。
Lc	‘0B’
Data	密钥索引号（1 字节） 交易金额（4 字节） 终端机编号（6 字节）
Le	‘10’

9.2.2.10.4 响应信息

命令执行成功返回的数据包括以下内容：

—ED 余额 4 字节
—ED 联机交易序号 2 字节
—密钥版本号 1 字节
—算法标识 1 字节
—伪随机数 4 字节
—MAC1 4 字节

计算 MAC1 的数据包括：

—旧余额 4 字节
—交易金额 4 字节
—交易类型 1 字节
—终端机编号 6 字节

命令执行不成功，则只返回状态码 SW1 和 SW2。

SW1	SW2	含义

90	00	命令执行成功
61	xx	正常处理。‘xx’表示可以通过后续 GET RESPONSE 命令得到的额外数据长度
65	81	EEPROM 损坏，导致卡锁定
67	00	Lc 长度错
69	82	不满足安全状态
69	85	使用条件不满足
6A	81	功能不支持
6A	86	参数 P1，P2 不正确
6D	00	命令不存在
6E	00	命令类型错
93	03	应用被永久锁定
94	01	金额不足
94	02	交易计数器达到最大值
94	03	密钥索引不支持

9.2.2.10.5 命令详解

操作： 用户卡获得圈提初始化的状态

条件： 建立有电子存折文件
通过 PIN 验证
圈提密钥索引：’08’
获得交易金额：’000003E8’
获得终端机编号 ：’111111111111’

命令： ‘805005010B08000003E8111111111111’

响应： 返回
个人帐户余额： ‘000007B8’
联机交易序号： ‘0002’
密钥版本号： ‘00’
算法标识： ‘00’
伪随机数： ‘50DAF8B5’
MAC1： ‘02D3DB82’

9.2.2.11 INITIALIZE FOR UPDATE 命令

9.2.2.11.1 命令描述

INITIALIZE FOR UPDATE 命令用于金融初始化修改透支限额交易。命令执行后 IC 卡为金融修改透支限额交易状态。

9.2.2.11.2 使用条件和安全

INITIALIZE FOR UPDATE 命令仅对 UPDATE OVERDRAW LIMIT 命令有效。命令执行前，需要验证个人密码（PIN）。

9.2.2.11.3 命令格式

代码	值
CLA	‘80’
INS	‘50’
P1	‘04’
P2	‘01’
Lc	‘07’
Data	密钥索引号（1 字节） 终端机编号（6 字节）
Le	‘13’

9.2.2.11.4 响应信息

命令执行成功返回的数据包括以下内容：

—ED 余额 4 字节
—ED 联机交易序号 2 字节
—旧透支限额 3 字节
—密钥版本号 1 字节
—算法标识 1 字节
—伪随机数 4 字节
—MAC1 4 字节

计算 MAC1 的数据包括：

—旧余额 4 字节
—旧透支限额 3 字节
—交易类型 1 字节
—终端机编号 6 字节

命令执行不成功，则只返回状态码 SW1 和 SW2。

SW1	SW2	含义

61	xx	正常处理。‘xx’表示可以通过后续 GET RESPONSE 命令得到的额外数据长度
65	81	EEPROM 损坏，导致卡锁定
67	00	Lc 长度错
69	82	不满足安全状态
69	85	使用条件不满足
6A	81	功能不支持
6A	86	参数 P1，P2 不正确
6D	00	命令不存在
6E	00	命令类型错
93	03	应用被永久锁定
94	02	交易计数器达到最大值
94	03	密钥索引不支持

9.2.2.11.5 命令详解

操作： 用户卡获得修改透支限额初始化的状态

条件： 建立有电子存折文件
通过 PIN 验证
修改透支限额密钥索引： ‘09’
获得终端机编号 ：’130000000001’

命令： 805004010709130000000001

响应： 返回
ED 余额： ‘000007C0’
ED 联机交易序号： ‘0004’
就透支限额： ‘000000’
密钥版本号： ‘00’
算法标识： ‘00’
伪随机数： ‘2341F557’
MAC1 ： ‘4293CC25’
（MAC 计算参照“安全管理”一章中金融环境下交易 MAC 的计算，使用修改透支限额的过程密钥）

9.2.2.12 RELOAD PIN 命令

9.2.2.12.1 命令描述

RELOAD PIN 命令用于重装 PIN。

9.2.2.12.2 使用条件和安全

此命令只能在金融应用环境下执行。

本命令只适用于主 PIN 重装。对于其他 PIN，请参见 PIN CHANGE/UNLOCK 命令。

RELOAD 命令执行必须满足 PIN 的访问权限和写控制属性。命令的数据域格式为明文校验。

9.2.2.12.3 命令报文

代码	值
CLA	‘80’
INS	‘5E’
P1	‘00’
P2	‘00’
Lc	‘06’~‘0A’
Data	重装的 PIN 值 || MAC
Le	不存在

9.2.2.12.4 响应信息

SW1	SW2	含义
90	00	命令执行成功
65	81	EEPROM 损坏，导致卡锁定
67	00	Lc 长度错
69	83	验证方法锁定
69	85	使用条件不满足
69	88	安全报文数据项不正确
6A	80	数据域参数不正确
6A	81	功能不支持
6A	86	参数 P1、P2 不正确
6D	00	命令不存在
6E	00	命令类型错
6A	88	引用数据找不到
93	03	应用永久锁定

9.2.2.12.5 命令详解

操作：	金融环境下重装 MF 的主 PIN
条件：	重装的新 PIN 值：8888
步骤 1：	用 MF 主 PIN 信息中 RLK-KID 指向密钥 RLK（本例取：RLK，16 字节'00'），左右异或后的结果 RLK_MAC 计算 MAC。 初始值 InitialData='0000000000000000' 8 字节 NEWPIN= '8888' MAC 计算数据为：MACData = InitialData + NEWPIN DES MAC（RLK_MAC，MACData，MAC） MAC='E88182F1'
步骤 2：	重装 MF 的主 PIN
命令：	'805E000006' + '8888E88182F1'
响应：	'9000'

9.2.2.13 UPDATE OVERDRAW LIMIT 命令

9.2.2.13.1 命令描述

UPDATE OVERDRAW LIMIT 命令用于修改透支限额。修改透支限额表示允许持卡人透支消费的额度，是一种信用的体现。修改透支限额是金融机构为持卡人设定的透支额度。

9.2.2.13.2 使用条件和安全

UPDATE OVERDRAW LIMIT 命令执行之前，必须成功执行 INITIALIZE FOR UPDATE 命令。

9.2.2.13.3 命令报文

代码	值
CLA	‘80’
INS	‘58’
P1	‘00’
P2	‘00’
Lc	‘0E’
Data	新透支限额（3 字节） 交易日期（4 字节） 交易时间（3 字节） MAC2（4 字节）
Le	‘04’（TAC）

计算 MAC2 的数据包括：

—新透支限额 3 字节
—交易类型 1 字节
—终端机编号 6 字节
—交易日期 4 字节
—交易时间 3 字节

9.2.2.13.4 响应信息

命令执行成功返回的数据包括以下内容：

—TAC 4 字节

计算 TAC 的数据包括：

—新余额 4 字节
—联机交易序号（加 1 前） 2 字节
—新透支限额 3 字节
—交易类型 1 字节
—终端机编号 6 字节
—交易日期 4 字节
—交易时间 3 字节

命令执行不成功，则只返回状态码 SW1 和 SW2。

SW1	SW2	含义
90	00	命令执行成功
61	xx	正常处理。‘xx’表示可以通过后续 GET RESPONSE 命令得到的额外数据长度
65	81	EEPROM 损坏，导致卡锁定
67	00	长度错误
69	01	命令不接受（无效状态）
69	83	认证方法锁定
69	85	使用条件不满足
6A	81	功能不支持
6A	86	参数 P1、P2 不正确
6D	00	命令不存在
6E	00	命令类型错
93	02	MAC 无效
94	01	金额不足

9.2.2.13.5 命令详解

操作： 完成修改透支限额操作

条件： 成功执行 INITIALIZE FOR UPDATE 命令
新透支限额 ： ‘0003E8’
终端当前的日期：2007.10.22
终端当前的时间：18:18:18
通过修改透支限额交易中的过程密钥计算得到的 MAC2 ：’B873D8C3’
（MAC 计算参照“安全管理”一章中金融环境下交易 MAC 的计算）

命令： ‘805800000E0003E820071022181818B873D8C3’

响应： 返回 TAC ：’322A7C5A’
（TAC 计算参照“安全管理”一章中金融环境下交易 MAC 的计算，使用 TAC 密钥）

9.2.3 金融扩展命令

9.2.3.1 DEBIT FOR CAPP PURCHASE 命令

9.2.3.1.1 命令描述

DEBIT FOR CAPP PURCHASE 命令用于复合应用消费交易。

9.2.3.1.2 使用条件和安全

执行 DEBIT FOR CAPP PURCHASE 命令之前，应先成功执行 UPDATE CAPP DATA CASHE。

9.2.3.1.3 命令格式

代码	值
CLA	‘80’
INS	‘54’
P1	‘01’
P2	‘00’
Lc	‘0F’
Data	见下表
Le	‘08’

此命令格式的数据域定义见下表：

说明	长度（字节）
终端交易序号	4
交易日期	4
交易时间	3
MAC1	4

计算 MAC1 的数据包括：

—交易金额　　4 字节
—交易类型　　1 字节
—终端机编号　　6 字节
—交易日期　　4 字节
—交易时间　　3 字节

9.2.3.1.4 响应信息

此命令执行成功的响应报文数据域见下表：

说明	长度（字节）
TAC	4
MAC2	4

计算 TAC 的数据包括：

—交易金额　　4 字节
—交易类型　　1 字节

—终端机编号　　6 字节
—终端交易序号　　2 字节
—交易日期　　4 字节
—交易时间　　3 字节

计算 MAC2 的数据包括：

—交易金额　　4 字节

此命令执行成功的状态码是'9000'。

IC 卡可能回送的错误状态见下表：

SW1	SW2	说明
'65'	'81'	内存错误
'67'	'00'	长度错误
'69'	'01'	命令不接受（无效状态）
'69'	'85'	使用条件不满足
'93'	'01'	金额不足
'93'	'02'	MAC 无效

9.2.3.1.5 命令详解

操作：　完成复合消费操作
条件 1：　已经成功执行过 UPDATE CAPP DATA CASHE 命令
得到终端交易序号 ：00000001
终端当前的日期：2003 ，10， 10
终端当前的时间：15 ：30 ：00
通过消费交易中的过程密钥计算得到的 MAC1 ：11D3245B
（MAC 计算参照"安全管理"一章中金融环境下交易 MAC 的计算）
命令：　'805401000F0000000120031010153000 11D3245B'
响应：　返回 TAC ： D1433650
（TAC 计算参照"安全管理"一章中金融环境下交易 MAC 的计算，使用 TAC 密钥）
返回 MAC2：83270316
（MAC 计算参照"安全管理"一章中金融环境下交易 MAC 的计算，使用消费交易中的过程密钥）

9.2.3.2 DEBIT FOR UNLOCK 命令

9.2.3.2.1 命令描述

DEBIT FOR UNLOCK 命令用于对电子钱包进行解扣操作。

9.2.3.2.2 使用条件和安全

应用处于灰锁空闲状态。

9.2.3.2.3 命令格式

DEBIT FOR UNLOCK 命令格式见下表：

代码	值
CLA	‘E0’
INS	‘7E’
P1	‘08’
P2	‘01’
Lc	‘1B’
Data	见下表
Le	‘04’

此命令格式的数据域定义见下表：

说明	长度（字节）
交易金额	4
交易序号	2
终端机编号	6
终端交易序号	4
交易日期（终端）	4
交易时间（终端）	3
GMAC	4

计算 GMAC 的数据包括：

—交易金额　　4 字节

9.2.3.2.4 响应信息

此命令执行成功的响应报文数据域见下表：

说明	长度（字节）
TAC	4

计算 TAC 的数据包括：

—交易金额　　4 字节
—交易类型标识=‘93’　　1 字节
—终端机编号　　6 字节
—终端交易序号　　4 字节

—交易日期　　3 字节
—交易时间　　4 字节

此命令执行成功的状态码是’9000’。

IC 卡可能回送的错误状态见下表：

SW1	SW2	说明
‘64’	‘00’	状态标志位未改变
‘65’	‘81’	内存错误
‘67’	‘00’	长度错误
‘69’	‘01’	命令不接受（无效状态）
‘69’	‘85’	使用条件不满足
‘93’	‘01’	金额不足
‘93’	‘02’	MAC 无效
‘94’	‘06’	所需 MAC 和 TAC 不可用

9.2.3.2.5 命令详解

操作：　完成复合消费操作

条件 1：　应用处于灰锁空闲状态。
交易金额：0000000A
得到终端交易序号 ：00000001
获得终端机编号 ：130000000001
终端当前的日期：2003 ，10， 10
终端当前的时间：15 ：30 ：00
通过消费交易中的过程密钥计算得到的 GMAC ：11D3245B
（MAC 计算参照“安全管理”一章中金融环境下交易 MAC 的计算）

命令：　‘E07E08011B0000000A0000000113000000000012003101015300011D3245B’

响应：　返回 TAC ： D1433650
（TAC 计算参照“安全管理”一章中金融环境下交易 MAC 的计算，使用 TAC 密钥）

9.2.3.3 GET LOCK PROOF 命令

9.2.3.3.1 命令描述

GET LOCK PROOF 命令用于读取电子钱包应用的灰锁状态以及相关的证明码。

9.2.3.3.2 使用条件和安全

如果应用没有执行任何交易，执行该命令返回‘6985’。

9.2.3.3.3 命令格式

GET LOCK PROOF 命令格式见下表：

代码	值
CLA	‘E0’
INS	‘CA’
P1	‘00’：普通读取 ‘01’：清除 TACUF（交易验证码待读标志）
P2	‘00’
Lc	不存在
Data	不存在
Le	‘1E’或不存在

此命令格式的数据域不存在。

9.2.3.3.4 响应信息

此命令执行成果，根据 P1 的参数、电子钱包应用的灰锁状态和 TACUF，形成不同的响应报文数据域，其关系见下表。

参数、状态与 GET LOCK PROOF 响应报文数据域的关系

P1 参数	TACUF	灰锁状态	响应报文数据域	
			数据域列表	报文中的状态字
‘00’	标志复位	无灰锁	表 10-1	‘00’
		有灰锁	表 10-2	‘01’
	标志置位	不影响	表 10-3	‘10’
‘01’	将 TACUF 标志复位	不影响	不存在	

如果命令执行不成功，则只在响应报文中回送 SW1 和 SW2。

10-1 正常状态 GET LOCK PROOF 命令执行成功的响应报文数据域

说明	长度（字节）
状态字（＝’00’表示当前应用无灰锁）	1
上次发生的解扣、联机解扣的交易类型	1

标识	
上次解扣、联机解扣的电子钱包（’01’）	1
上次解扣、联机解扣的电子钱包余额	4
上次解扣、联机解扣的电子钱包的交易序号	2
上次解扣、联机解扣的终端机编号	6
上次解扣、联机解扣的日期	4
上次解扣、联机解扣的时间	3
上次解扣、联机解扣的交易金额	4
上次解扣的 TAC 或联机解扣的 MAC3	4

10-2 灰锁状态 GET LOCK PROOF 命令执行成功的响应报文数据域

说明	长度（字节）
状态字（＝’01’表示当前应用已灰锁）	1
灰锁的交易类型标识	1
被灰锁的电子钱包（’01’）	1
被灰锁的电子钱包余额	4
被灰锁的电子钱包交易序号	2
执行 GREY LOCK 时的终端机编号	6
执行 GREY LOCK 时的日期	4
执行 GREY LOCK 时的时间	3
灰锁时的 MAC2	4
灰锁时的 GTAC	4

10-3 TAC 未读时 GET LOCK PROOF 命令执行成功的响应报文数据域

说明	长度（字节）
状态字(＝’10’表示当前应用 TAC 未读）	1
上次解扣的交易类型标识	1
上次解扣的电子钱包（’01’）	1
上次解扣的电子钱包余额	4
上次解扣的电子钱包交易序号	2
上次执行解扣的终端机编号	6
上次执行解扣的日期	4
上次执行解扣的时间	3
解扣交易金额	4
上次解扣的 TAC	4

此命令执行成功的状态码是’9000’。

IC 卡可能回送的错误状态见下表：

SW1	SW2	说明
‘64’	‘00’	状态标志位未改变
‘65’	‘81’	内存错误

‘69’	‘85’	使用条件不满足
‘6A’	‘86’	P1、P2 参数不正确

9.2.3.3.5 命令详解

操作： 获得电子钱包应用的灰锁状态以及相关的证明码

条件 1： 正常状态普通读取。

命令： ‘E0CA00001E’

响应： 状态字
上次发生的解扣、联机解扣的交易类型标识
上次解扣、联机解扣的电子钱包（’01’）
上次解扣、联机解扣的电子钱包余额
上次解扣、联机解扣的电子钱包的交易序号
上次解扣、联机解扣的终端机编号
上次解扣、联机解扣的日期
上次解扣、联机解扣的时间
上次解扣、联机解扣的交易金额
上次解扣的 TAC 或联机解扣的 MAC3

条件 2： 灰锁状态普通读取。

命令： ‘E0CA00001E’

响应： 状态字
灰锁的交易类型标识
被灰锁的电子钱包（’01’）
被灰锁的电子钱包余额
被灰锁的电子钱包的交易序号
执行 GREY LOCK 时的终端机编号
执行 GREY LOCK 时的日期
执行 GREY LOCK 时的时间
灰锁时的 MAC2
灰锁时的 GTAC

条件 3： TAC 未读时普通读取。

命令： ‘E0CA00001E’

响应： 状态字
上次发生的解扣的交易类型标识
上次解扣的电子钱包（’01’）
上次解扣的电子钱包余额
上次解扣的电子钱包的交易序号
上次执行解扣的终端机编号
上次执行解扣的日期
上次执行解扣的时间
解扣交易金额
上次解扣的 TAC

条件 4： 清除 TACUF（交易验证码待读标志）

命令：　‘E0CA0100’
响应：　‘9000’

9.2.3.4 GET TRANSACTION PROVE 命令

9.2.3.4.1 命令描述

GET TRANSACTION PROVE 命令提供了一种在交易过程中的恢复机制。用于获取交易认证码 TAC（或者 GTAC）和 MAC。

9.2.3.4.2 使用条件和安全

在电子存折或者扩展电子钱包交易被迫中断时（掉电或提前拔卡），IC 卡具有恢复机制。即重新插卡后，如果交易已经完成，执行该命令时返回上次交易的 TAC 和 MAC；如果未完成，则执行该命令时返回状态码‘9406’（所需 MAC 不可用）。卡片在应用开发状态时不提供交易保护功能。交易被中断后，重新插卡执行该命令 IC 卡返回状态码‘9406’。

9.2.3.4.3 命令格式

代码	值
CLA	‘80’
INS	‘5A’
P1	‘00’
P2	要取的 MAC 或者 TAC 所对应的交易类型标识。
Lc	‘02’
Data	联机或脱机交易序号。
Le	‘08’

P2 交易类型：

01—ED 圈存
02—扩展电子钱包圈存
03—ED 圈提
04—ED 取款
05—ED 消费
06—扩展电子钱包消费
07—ED 修改透支限额
09—扩展电子钱包复合消费

9.2.3.4.4 响应信息

命令执行成功返回的数据包括以下内容：

—MAC　　4 字节
—TAC （或者 GTAC）　　4 字节

命令执行不成功，则只返回状态码 SW1 和 SW2。

SW1	SW2	含义
90	00	命令执行成功
69	85	使用条件不满足
69	01	命令不接受

67	00	Le 长度错
6A	81	功能不支持
6A	86	P1 参数错
6D	00	命令不存在
6E	00	命令类型错
93	03	应用被永久锁定
94	06	所需 MAC 不可用

9.2.3.4.5 命令详解

操作： 获取交易认证码 TAC（或者 GTAC）和 MAC

条件 1： 在电子存折或者扩展电子钱包交易被迫中断时。
要取的 MAC 或者 TAC 所对应的交易类型标识为’09’。
脱机交易序号为 0001

命令： ‘805A0009020001’

响应： MAC：83270316
GTAC：D1433650

9.2.3.5 GREY LOCK 命令

9.2.3.5.1 命令描述

GREY LOCK 命令用于灰锁电子钱包。

9.2.3.5.2 使用条件和安全

在执行 GREY LOCK 命令前，必须成功执行 INITIALIZE FOR GREY LOCK，即应用已处于预灰锁状态。

9.2.3.5.3 命令格式

GREY LOCK 命令格式见下表：

代码	值
CLA	‘E0’
INS	‘7C’
P1	‘08’
P2	‘00’
Lc	‘13’
Data	见下表
Le	‘08’

此命令格式的数据域定义见下表：

说明	长度（字节）
终端交易序号	4
终端随机数	4
交易日期（终端）	4
交易时间（终端）	3
MAC1	4

计算 MAC1 的数据包括：

—交易类型标识=‘91’ 1 字节
—终端机编号 6 字节
—交易日期 4 字节
—交易时间 3 字节

9.2.3.5.4 响应信息

此命令执行成功的响应报文数据域见下表：

说明	长度（字节）
GTAC	4
MAC2	4

计算 MAC2 的数据包括：

—扩展电子钱包余额 4 字节

—扩展电子钱包脱机交易序号（加 1 前） 2 字节

计算 GTAC 的数据包括：

—交易类型标识=‘91’ 4 字节
—终端机编号 6 字节
—终端交易序号 4 字节
—交易日期 3 字节
—交易时间 4 字节

此命令执行成功的状态码是’9000’。

IC 卡可能回送的错误状态见下表：

SW1	SW2	含义
‘64’	‘00’	状态标志位未改变
‘65’	‘81’	内存错误
‘67’	‘00’	长度错误
‘69’	‘01’	命令不接受（无效状态）
‘69’	‘85’	使用条件不满足
‘93’	‘02’	MAC 无效

9.2.3.5.5 命令详解

操作： 完成灰锁操作

条件 1： 已经成功执行过 INITIALIZE FOR GREY LOCK 命令
得到终端交易序号 ：00000001
终端随机数：1234567812345678
终端当前的日期：2003 ，10， 10
终端当前的时间：15 ：30 ：00
通过消费交易中的过程密钥计算得到的 MAC1 ：11D3245B
（MAC 计算参照“安全管理”一章中金融环境下交易 MAC 的计算）

命令： ‘E07C0800130000000112345678123456782003101015300011D3245B’

响应： 返回 GTAC ： D1433650
（GTAC 计算参照“安全管理”一章中金融环境下交易 MAC 的计算，使用 TAC 密钥）
返回 MAC2：83270316
（MAC 计算参照“安全管理”一章中金融环境下交易 MAC 的计算，使用消费交易中的过程密钥）

9.2.3.6 GREY UNLOCK 命令

9.2.3.6.1 命令描述

GREY UNLOCK 命令用于联机解扣交易。

9.2.3.6.2 使用条件和安全

应用处于灰锁状态。执行 GREY UNLOCK 命令前，必须成功执行 INITIALIZE FOR GREY UNLOCK 命令。

9.2.3.6.3 命令格式

GREY UNLOCK 命令格式见下表：

代码	值
CLA	‘E0’
INS	‘7E’
P1	‘09’
P2	‘00’
Lc	‘0F’
Data	见下表
Le	‘04’

此命令格式的数据域定义见下表：

说明	长度（字节）
交易金额	4
交易日期（主机）	4
交易时间（主机）	3
MAC2	4

计算 MAC2 的数据包括：

—应补扣的交易金额　　4 字节
—交易类型标识=‘95’　　1 字节
—终端机编号　　6 字节
—交易日期　　3 字节
—交易时间　　4 字节

9.2.3.6.4 响应信息

此命令执行成功的响应报文数据域见下表：

说明	长度（字节）
MAC3	4

计算 MAC3 的数据包括：

—扩展电子钱包余额　　4 字节
—扩展电子钱包联机交易序号（加 1 前）　　2 字节
—交易金额　　4 字节

—交易类型标识=‘95’ 1 字节
—终端机编号 6 字节
—交易日期 3 字节
—交易时间 4 字节

此命令执行成功的状态码是’9000’。

IC 卡可能回送的错误状态见下表：

SW1	SW2	说明
‘64’	‘00’	状态标志位未改变
‘65’	‘81’	内存错误
‘67’	‘00’	长度错误
‘69’	‘01’	命令不接受（无效状态）
‘69’	‘85’	使用条件不满足
‘93’	‘02’	MAC 无效
‘94’	‘01’	金额不足

9.2.3.6.5 命令详解

操作： 完成联机解扣交易
条件 1： 已经成功执行过 INITIALIZE FOR GREY UNLOCK 命令
交易金额 ：0000000A
终端当前的日期：2003 ，10， 10
终端当前的时间：15 ：30 ：00
通过消费交易中的过程密钥计算得到的 MAC2 ：11D3245B
（MAC 计算参照“安全管理”一章中金融环境下交易 MAC 的计算）
命令： ‘E07E09000F0000000A2003101015300011D3245B’
响应： 返回 MAC3：83270316
（MAC 计算参照“安全管理”一章中金融环境下交易 MAC 的计算，使用消费交易中的过程密钥）

9.2.3.7 INITIALIZE FOR CAPP PURCHASE 命令

9.2.3.7.1 命令描述

INITIALIZE FOR CAPP PURCHASE 命令用于初始化复合应用消费交易。

9.2.3.7.2 使用条件和安全

该命令的执行需要在金融应用中。命令执行成功后，应用进入 CAPP1 状态。允许交易金额为 0。

9.2.3.7.3 命令格式

INITIALIZE FOR CAPP PURCHASE 命令格式见下表：

代码	值
CLA	‘80’
INS	‘50’
P1	‘03’
P2	‘02’
Lc	‘0B’
Data	见下表
Le	‘0F’

此命令格式的数据域定义见下表：

说明	长度（字节）
密钥索引号	1
交易金额	4
终端机编号	6

9.2.3.7.4 响应信息

此命令执行成功的响应报文数据域见下表：

说明	长度（字节）
电子钱包余额	4
电子钱包交易序号	2
透支限额	3
密钥版本号（DPK）	1
密钥标识（DPK）	1
伪随机数（IC 卡）	4

如果命令执行不成功，则只在响应报文中回送 SW1 和 SW2。

此命令执行成功的状态码是’9000’。

IC 卡可能回送的错误状态见下表：

SW1	SW2	说明

‘65’	‘81’	内存错误
‘69’	‘85’	使用条件不满足
‘94’	‘01’	金额不足
‘94’	‘03’	密钥索引不支持
‘94’	‘02’	交易计数器达到最大值
‘94’	‘08’	应用灰锁锁定

9.2.3.7.5 命令详解

操作： 用户卡获得复合消费初始化的状态

条件： 建立有扩展电子钱包文件
消费密钥索引：01
获得交易金额：0000000A
获得终端机编号 ：130000000001

命令： 805003020B010000000A130000000001

响应： 返回
余额： 00002710
电子钱包交易序号： 0001
透支限额： 000000
密钥版本号： 01
密钥标识： 01
伪随机数： 13D22145

9.2.3.8 INITIALIZE FOR LOAD 命令

9.2.3.8.1 命令描述

INITIALIZE FOR LOAD 命令用于金融初始化圈存交易。命令执行后 IC 卡为金融圈存交易状态。

9.2.3.8.2 使用条件和安全

INITIALIZE FOR LOAD 命令仅对 CREDIT FOR LOAD 命令有效。在 ED 操作前，需要验证个人密码（PIN）。

9.2.3.8.3 命令格式

代码	值
CLA	‘80’
INS	‘50’
P1	‘00’
P2	‘01’：ED 圈存 ‘02’：EP 圈存
Lc	‘0B’
Data	密钥索引号（1 字节） 交易金额（4 字节） 终端机编号（6 字节）
Le	‘10’

9.2.3.8.4 响应信息

命令执行成功返回的数据包括以下内容：

—ED 或 EP 余额 4 字节
—ED 或 EP 联机交易序号 2 字节
—密钥版本号 1 字节
—算法标识 1 字节
—伪随机数 4 字节
—MAC1 4 字节

计算 MAC1 的数据包括：

—旧余额 4 字节
—交易金额 4 字节
—交易类型 1 字节
—终端机编号 6 字节

命令执行不成功，则只返回状态码 SW1 和 SW2。

SW1	SW2	含义
90	00	命令执行成功
65	81	EEPROM 损坏，导致卡锁定
67	00	Lc 长度错
69	82	不满足安全状态
69	85	使用条件不满足
6A	81	功能不支持

6A	86	参数 P1，P2 不正确
6D	00	命令不存在
6E	00	命令类型错
93	03	应用被永久锁定
94	03	密钥索引不支持

9.2.3.8.5 命令详解

操作： 用户卡获得圈存初始化的状态

条件： 建立电子存折文件或电子钱包文件
对于电子存折文件，通过 PIN 验证
圈存密钥索引：’01’
获得交易金额：’000003E8’
获得终端机编号 ：’130000000001’

命令： ‘805000010B05000003E8130000000001’

响应： 返回
余额： ‘00000000’
联机交易序号： ‘0000’
密钥版本号： ‘00’
算法标识： ‘00’
伪随机数： ‘FE2C509C’
MAC1 ： ‘910B705F’

（MAC 计算参照“安全管理”一章中金融环境下交易 MAC 的计算，使用圈存过程密钥）

9.2.3.9 INITIALIZE FOR GREY LOCK 命令

9.2.3.9.1 命令描述

INITIALIZE FOR GREY LOCK 命令用于初始化灰锁操作。

9.2.3.9.2 使用条件和安全

初始化灰锁是对扩展电子钱包进行的操作。

9.2.3.9.3 命令格式

INITIALIZE FOR GREY LOCK 命令格式见下表：

代码	值
CLA	‘E0’
INS	‘7A’
P1	‘08’
P2	‘01’
Lc	‘07’
Data	见下表
Le	‘0F’

此命令格式的数据域定义见下表：

说明	长度（字节）
密钥索引号	1
终端机编号	6

9.2.3.9.4 响应信息

此命令执行成功的响应报文数据域见下表：

说明	长度（字节）
电子钱包余额	4
电子钱包交易序号	2
透支限额	3（全 0）
密钥版本号	1
算法标识	1
伪随机数	4

此命令执行成功的状态码是’9000’。

IC 卡可能回送的错误状态见下表：

SW1	SW2	说明
‘65’	‘81’	内存错误
‘69’	‘01’	命令不接受（无效状态）
‘69’	‘85’	使用条件不满足

‘94’	‘03’	密钥索引号不支持

9.2.3.9.5 命令详解

操作： 用户卡获得灰锁操作初始化的状态

条件： 建立有扩展电子钱包文件
消费密钥索引：01
获得终端机编号 ：130000000001

命令： E07A08010701130000000001

响应： 返回
余额： 00002710
电子钱包交易序号： 0001
透支限额： 000000
密钥版本号： 01
算法标识： 00
伪随机数： 13D22145

9.2.3.10 INITIALIZE FOR GREY UNLOCK 命令

9.2.3.10.1 命令描述

INITIALIZE FOR GREY UNLOCK 命令用于初始化联机解扣交易。

9.2.3.10.2 使用条件和安全

应用处于灰锁状态。该命令是对扩展电子钱包进行操作。

9.2.3.10.3 命令格式

INITIALIZE FOR GREY UNLOCK 命令格式见下表：

代码	值
CLA	‘E0’
INS	‘7A’
P1	‘09’
P2	‘01’
Lc	‘07’
Data	见下表
Le	‘12’

此命令格式的数据域定义见下表：

说明	长度（字节）
密钥索引号	1
终端机编号	6

9.2.3.10.4 响应信息

此命令执行成功的响应报文数据域见下表：

说明	长度（字节）
电子钱包余额	4
电子钱包脱机交易序号	2
电子钱包联机交易序号	2
密钥版本号	1
密钥算法	1
伪随机数	4
MAC1	4

计算 MAC1 的数据包括：

—扩展电子钱包余额 4 字节
—扩展电子钱包脱机交易序号 2 字节
—交易类型标识=‘95’ 1 字节
—终端机编号 6 字节

此命令执行成功的状态码是’9000’。

IC 卡可能回送的错误状态见下表：

SW1	SW2	说明
‘65’	‘81’	内存错误
‘67’	‘00’	长度错误
‘69’	‘01’	命令不接受（无效状态）
‘69’	‘85’	使用条件不满足
‘6A’	‘86’	P1、P2 参数不正确
‘94’	‘03’	密钥索引号不支持

9.2.3.10.5 命令详解

操作： 用户卡获得联机解扣初始化的状态

条件： 建立有扩展电子钱包文件
消费密钥索引：01
获得终端机编号 ：130000000001

命令： E07A09010701130000000001

响应： 返回
电子钱包余额： 00002710
电子钱包脱机交易序号： 0001
电子钱包联机交易序号：0001
透支限额： 000000
密钥版本号： 01
密钥算法： 00
伪随机数： 13D22145
MAC1： 83270316

9.2.3.11 INITIALIZE FOR PURCHASE 命令

9.2.3.11.1 命令描述

INITIALIZE FOR PURCHASE 命令用于金融初始化消费交易。命令执行后 IC 卡为金融消费交易状态。

9.2.3.11.2 使用条件和安全

INITIALIZE FOR PURCHASE 命令仅 DEBIT FOR PURCHASE/CASH WITHDRAW 命令有效。在 ED 操作前，需要验证个人密码（PIN）。

9.2.3.11.3 命令格式

代码	值
CLA	‘80’
INS	‘50’
P1	‘01’
P2	‘01’：用于 ED； ‘02’：用于 EP；
Lc	‘0B’
Data	密钥索引号（1 字节） 交易金额（4 字节） 终端机编号（6 字节）
Le	‘0F’

9.2.3.11.4 响应信息

命令执行成功返回的数据包括以下内容：

—ED 余额 4 字节
—ED 脱机交易序号 2 字节
—透支限额 3 字节
—密钥版本号 1 字节
—算法标识 1 字节
—伪随机数 4 字节

命令执行不成功，则只返回状态码 SW1 和 SW2。

SW1	SW2	含义
90	00	执行命令成功
61	xx	正常处理。‘xx’表示可以通过后续 GET RESPONSE 命令得到的额外数据长度。
65	81	EEPROM 损坏，导致卡锁定
67	00	Lc 长度错
69	82	不满足安全状态
69	85	使用条件不满足
6A	81	功能不支持
6A	86	参数 P1、P2 不正确
6D	00	命令不存在
6E	00	命令类型错

93	03	应用被永久锁定
94	01	金额不足
94	02	交易计数器达到最大值
94	03	密钥索引不支持
‘94’	‘08’	应用灰锁锁定

9.2.3.11.5 命令详解

操作： 用户卡获得消费初始化的状态

条件 1： 建立有电子存折文件或电子钱包文件
对于电子存折文件，通过 PIN 验证
消费密钥索引：’01’
获得交易金额：’00000008’
获得终端机编号 ：’130000000001’

命令： ‘805001010B01000003E8130000000001’(电子存折)

响应： 返回

余额：	‘000007D0’
脱机交易序号：	‘0000’
透支限额：	‘000000’
密钥版本号：	‘00’
算法标识：	‘00’
伪随机数：	‘41DD741F’

9.2.3.12 UPDATE CAPP DATA CACHE 命令

9.2.3.12.1 命令描述

UPDATE CAPP DATA CACHE 命令用于复合应用消费交易中更新复合应用数据缓存，缓存数据将被 DEBIT FOR CAPP PURCHASE 命令用于改写复合应用专用文件中相关记录。

9.2.3.12.2 使用条件和安全

UPDATE CAPP DATA CACHE 命令执行之前，必须成功执行 INITIALIZE FOR CAPP PURCHASE 命令。如果更新数据长度小于记录长度，则卡片应在数据后面自动填充‘0’至记录尾。

9.2.3.12.3 命令格式

此命令格式见下表：

代码	值								
CLA	‘80’								
INS	‘DC’								
P1	复合应用类型标识符								
P2	B8	B7	B6	B5	B4	B3	B2	B1	含义
	0	0	0	0	0	—	—	—	RFU
	X	X	X	X	X	—	—	—	复合应用专用文件 SFI
	1	1	1	1	1	—	—	—	RFU
	—	—	—	—	—	0	0	0	第一个标识符出现的记录
	—	—	—	—	—	X	X	X	RFU
	其它值								RFU
Lc	后续数据域的长度								
Data	更新记录								
Le	不存在								

9.2.3.12.4 响应信息

响应报文数据域不存在。

此命令执行成功的状态码是’9000’。

IC 卡可能回送的错误状态码见下表：

SW1	SW2	含　义
‘65’	‘81’	内存失败（修改失败）
‘67’	‘00’	长度错误（Lc 域为空）
‘69’	‘81’	命令与文件结构不相容
‘69’	‘82’	不满足安全状态
‘69’	‘86’	不满足命令执行的条件（不是当前的 EF）

‘6A’	‘81’	不支持此功能
‘6A’	‘82’	未找到文件
‘6A’	‘83’	未找到记录
‘6A’	‘84’	文件中存储空间不够
‘94’	‘07’	复合应用禁止

9.2.3.12.5 命令详解

操作： 复合应用数据缓存被更新
条件： 建立有扩展电子钱包文件
复合应用类型标识符：09
复合应用专用文件标识：0019
要更新的数据长度：10 个字节
要更新的数据内容：09088877665544332211
命令： 80DC09C80A09088877665544332211
响应： 9000

9.2.4 建设事业专用命令

建设事业专用命令与金融专用命令的不同点见下表：

章节	命令	建设事业命令的不同之处
9.4.4.3	DEBIT FOR CAPP PURCHASE	参与 MAC1 计算的源数据多了 9 个字节的安全认证识别码。
9.2.4.4	DEBIT FOR PURCHASE/CASH WITHDRAW	参与 MAC1 计算的源数据多了 9 个字节的安全认证识别码。
9.2.4.6	DEBIT FOR UNLOCK	参与 GMAC 计算的源数据多了 9 个字节的安全认证识别码。
9.2.4.8	GET MESSAGE	增加此命令用于读取安全认证识别码

9.2.4.1 CHANGE PIN 命令

见“金融专用命令”的9.2.2.1一节。

9.2.4.2 CREDIT FOR LOAD 命令

见“金融专用命令”的9.2.2.2一节。

9.2.4.3 DEBIT FOR CAPP PURCHASE 命令

见“金融专用命令”的9.2.3.1一节。

9.2.4.4 DEBIT FOR PURCHASE/CASH WITHDRAW 命令

9.2.4.4.1 命令描述

DEBIT FOR PURCHASE/CASH WITHDRAW 命令用于建设部钱包普通消费交易、电子存折消费交易，取现仅限于金融交易。

9.2.4.4.2 使用条件和安全

执行 DEBIT FOR PURCHASE/CASH WITHDRAW 命令之前，应先成功执行 INITIALIZE FOR PURCHASE 命令或 INITIALIZE FOR CASH WITHDRAW 命令。

9.2.4.4.3 命令格式

代码	值
CLA	‘80’
INS	‘54’
P1	‘01’
P2	‘00’
Lc	‘0F’
Data	终端交易序号（4 字节） 交易日期（4 字节） 交易时间（3 字节） MAC1（4 字节）
Le	‘08’（TAC＋MAC2）

计算 MAC1 的数据包括：

—交易金额 4 字节
—交易类型 1 字节
—终端机编号 6 字节
—交易日期 4 字节
—交易时间 3 字节
—安全认证识别码 9 字节

9.2.4.4.4 响应信息

命令执行成功返回的数据包括以下内容：

—TAC 4 字节
—MAC2 4 字节

计算 TAC 的数据包括：

—交易金额 4 字节
—交易类型 1 字节
—终端机编号 6 字节
—终端交易序号 2 字节
—交易日期 4 字节
—交易时间 3 字节

计算 MAC2 的数据包括：

—交易金额　　4 字节

命令执行不成功，则只返回状态码 SW1 和 SW2。

SW1	SW2	含义
90	00	命令执行成功
65	81	EEPROM 损坏，导致卡锁定
67	00	Lc 长度错
69	01	命令不接受（无效状态）
69	85	使用条件不满足
6A	81	功能不支持
6A	86	参数 P1、P2 不正确
6D	00	命令不存在
6E	00	命令类型错
93	02	MAC 无效
93	03	应用被永久锁定

9.2.4.4.5 命令详解

操作：　完成消费或取现操作
条件 1：　已经成功执行过 INITIALIZE FOR PURCHASE 命令
得到终端交易序号 ：’00000001’
终端当前的日期：2007 ，10， 22
终端当前的时间：18 ： 18 ： 18
通过消费交易中的过程密钥计算得到的 MAC1 ：’25B0E9A’
（MAC 计算参照“安全管理”一章中金融环境下交易 MAC 的计算）
命令：　‘805401000F0000000120071022181818 25B0E9A’
响应：　返回 TAC ： ‘F4A51EFF’
（TAC 计算参照“安全管理”一章中金融环境下交易 MAC 的计算，使用 TAC 密钥）
返回 MAC2：’66693766’（'MAC 计算参照“安全管理”一章中金融环境下交易 MAC 的计算，使用消费交易中的过程密钥）
条件 2：　已经成功执行过 INITIALIZE FOR CASH WITHDRAW 命令
得到终端交易序号 ：’00000001’
终端当前的日期：2007 ，10， 22
终端当前的时间：18 ： 18 ： 18
通过取现交易中的过程密钥计算得到的 MAC1 ：’B3C28CCD’
（MAC 计算参照“安全管理”一章中金融环境下交易 MAC 的计算）
命令：　‘805401000F0000000120071022181818 B3C28CCD’
响应：　返回 TAC ：’CF084AA3’
（TAC 计算参照“安全管理”一章中金融环境下交易 MAC 的计算，使用 TAC 密钥）
返回 MAC2 ：0F3676BA
（MAC 计算参照“安全管理”一章中金融环境下交易 MAC 的计算，使用消费交易中的过程密钥）

9.2.4.5 DEBIT FOR UNLOAD 命令

见“金融专用命令”的9.2.2.4一节。

9.2.4.6 DEBIT FOR UNLOCK 命令

9.2.4.6.1 命令描述

DEBIT FOR UNLOCK 命令用于对扩展电子钱包进行解扣操作。

9.2.4.6.2 使用条件和安全

应用处于灰锁空闲状态。

9.2.4.6.3 命令格式

代码	值
CLA	‘E0’
INS	‘7E’
P1	‘08’
P2	‘01’
Lc	‘1B’
Data	交易金额（4 字节） 扩展电子钱包脱机交易序号（2 字节） 终端机编号（6 字节） 终端交易序号（4 字节） 交易日期（4 字节） 交易时间（3 字节） GMAC（4 字节）
Le	‘04’（TAC）

计算 GMAC 的数据包括：

—交易金额　　4 字节
—安全认证识别码　　9 字节

9.2.4.6.4 响应信息

命令执行成功返回的数据包括以下内容：

—TAC　　4 字节

计算 TAC 的数据包括：

—交易金额　　4 字节
—交易类型标识=‘93’　　1 字节
—终端机编号　　6 字节
—终端交易序号　　4 字节
—交易日期　　3 字节
—交易时间　　4 字节

命令执行不成功，则只返回状态码 SW1 和 SW2。

SW1	SW2	含义
64	00	状态标志位未改变
90	00	命令执行成功

65	81	EEPROM 损坏，导致卡锁定
67	00	长度错误
69	01	命令不接受（无效状态）
69	82	不满足安全状态
69	83	认证方法锁定
69	85	使用条件不满足
6A	81	功能不支持
6A	86	参数 P1、P2 不正确
6D	00	命令不存在
6E	00	命令类型错
93	02	MAC 无效
94	01	金额不足
94	06	所需 MAC 和 TAC 不可用(命令报文中的建设事业脱机交易序号与卡内记录的不符)

9.2.4.6.5 命令详解

操作： 完成复合消费操作

条件 1： 应用处于灰锁空闲状态。
交易金额：0000000A
得到终端交易序号 ：00000001
获得终端机编号 ：130000000001
终端当前的日期：2003 ，10， 10
终端当前的时间：15 ：30 ：00
通过消费交易中的过程密钥计算得到的 GMAC ：11D3245B
（MAC 计算参照“安全管理”一章中金融环境下交易 MAC 的计算）

命令： ‘E07E08011B0000000A00000001130000000000120031010153000 11D3245B’

响应： 返回 TAC ： D1433650
（TAC 计算参照“安全管理”一章中金融环境下交易 MAC 的计算，使用 TAC 密钥）

9.2.4.7 GET BALANCE 命令

见“金融专用命令”的9.2.2.5一节。

9.2.4.8 GET MESSAGE 命令

9.2.4.8.1 命令描述

GET MESSAGE 命令用于消费交易，读取 CPU 卡中的安全认证识别码，即芯片安全区内的安全认证识别码或者芯片中的 MID||UID0UID1UID2UID3||四字节认证码，将安全认证识别码发送给 PSAM 卡进行认证。

9.2.4.8.2 使用条件和安全

该命令应在任意目录下都可以执行。

9.2.4.8.3 命令格式

代码	值
CLA	‘80’
INS	‘CA’
P1	‘00’
P2	‘00’
Lc	不存在
Data	不存在
Le	‘09’

9.2.4.8.4 响应信息

执行成功的响应报文数据域：

说明	长度（字节）
安全认证识别码	9

命令执行不成功，则只返回状态码 SW1 和 SW2。

SW1	SW2	含义
90	00	命令执行成功
67	00	Le 长度错
6A	81	功能不支持
6A	86	P1 参数错
6D	00	命令不存在
6E	00	命令类型错
93	03	应用被永久锁定

9.2.4.8.5 命令详解

操作：　得到 CPU 卡中的安全认证识别码
条件 1：　在建设事业环境的目录下。
命令：　‘80CA000009’
响应：　返回安全认证识别码：’551A4C790155555555’

9.2.4.9 GET TRANSACTION PROVE 命令

见“金融专用命令”的9.2.3.4一节。

9.2.4.10 GET LOCK PROOF 命令

见“金融专用命令”的9.2.3.3一节。

9.2.4.11 GREY LOCK 命令

见“金融专用命令”的9.2.3.5一节。

9.2.4.12 GREY UNLOCK 命令

见“金融专用命令”的9.2.3.6一节。

9.2.4.13 INITIALIZE FOR CAPP PURCHASE 命令

见“金融专用命令”的9.2.3.7一节。

9.2.4.14 INITIALIZE FOR CASH WITHDRAW 命令

见“金融专用命令”的9.2.2.7一节。

9.2.4.15 INITIALIZE FOR GREY LOCK 命令

见“金融专用命令”的9.2.3.9一节。

9.2.4.16 INITIALIZE FOR GREY UNLOCK 命令

见“金融专用命令”的9.2.3.10一节。

9.2.4.17 INITIALIZE FOR LOAD 命令

见“金融专用命令”的9.2.2.8一节。

9.2.4.18 INITIALIZE FOR PURCHASE 命令

见“金融专用命令”的9.2.2.9一节。

9.2.4.19 INITIALIZE FOR UNLOAD 命令

见“金融专用命令”的9.2.2.10一节。

9.2.4.20 INITIALIZE FOR UPDATE 命令

见“金融专用命令”的9.2.2.11一节。

9.2.4.21 RELOAD PIN 命令

见“金融专用命令”的9.2.2.12一节。

9.2.4.22 UPDATE CAPP DATA CACHE 命令

见“金融专用命令”的9.2.3.12一节。

9.2.4.23 UPDATE OVERDRAW LIMIT 命令

见“金融专用命令”的9.2.2.13一节。

9.2.5 个人化命令

9.2.5.1 CLEAR DF 命令

9.2.5.1.1 命令描述

CLEAR DF 删除 DF（包括 MF、DDF、ADF）文件体的内容。

9.2.5.1.2 使用条件和安全

CLEAR DF 命令的执行必须通过相应 DF 下主控 KEY 的认证。它不受 FREEZE MF 的影响，即使在 FREEZE MF 命令成功执行后，该命令仍然有效。

9.2.5.1.3 命令格式

代码	数　值
CLA	‘BF’
INS	‘CE’
P1	‘00’
P2	‘00’
Lc	不存在
DATA	不存在
Le	不存在

9.2.5.1.4 响应信息

响应信息中可能的状态码为：

SW1	SW2	说　明
90	00	命令执行成功
65	81	写 EEPROM 失败
69	82	不满足安全状态
69	85	使用条件不满足
6A	81	功能不支持
6A	86	P1、P2 参数错
6D	00	命令不存在
6E	00	CLA 错

9.2.5.1.5 命令详解

1、DES 算法

操作：　删除当前 DF 文件体的内容。
条件：　选择所要删除 的 DF ，通过当前 DF 的 MK 认证
步骤 1：选择 ADF1
命令：　‘00A4000002 ADF1’
响应：　‘9000’
步骤 2：取 8 字节随机数 Random，作为鉴别数据产生因子。
命令：　‘00840000 08’
响应：　Random=’ 8F8D5AEA85880901’

步骤3： 终端用与当前 ADF1 的主控密钥 ADF1_MK 相同的密钥（本例取：16字节'00'）对随机数加密，产生鉴别数据。
3DES Encryption（ADF1_MK，Random，VeriData）
鉴别数据计算结果为：VeriData = ' 82FE8A38C35A59DF'
步骤4： 外部认证 ADF1 主控密钥
命令： '0082000008' + VeriData
响应： '9000'
步骤5： 删除 ADF 文件体内容
命令： 'BFCE0000'
响应： '9000'

2、SM1 算法

操作： 删除当前 DF 文件体的内容。
条件： 选择所要删除 的 DF ，通过当前 DF 的 MK 认证
步骤1： 选择 ADF1
命令： '00A4000002 ADF1'
响应： '9000'
步骤2： 取 8 字节随机数 Random，作为鉴别数据产生因子。
命令： '00840000 08'
响应： Random
步骤3： 终端用与当前 ADF1 的主控密钥 ADF1_MK 相同的密钥（本例取：16字节'00'）对随机数加密，产生鉴别数据。
SM1 Encryption（ADF1_MK，Random，VeriData）
鉴别数据计算结果为：VeriData = ' F24531389C4769C7'
步骤4： 外部认证 ADF1 主控密钥
命令： '0082000008' + VeriData
响应： '9000'
步骤5： 删除 ADF 文件体内容
命令： 'BFCE0000'
响应： '9000'

9.2.5.2 CREATE FILE 命令

9.2.5.2.1 命令描述

CREATE FILE 命令用于在卡内建立各种类型的文件，并为所建文件分配应用空间。执行一次 CREATE FILE 命令只能建立一个文件。

文件占用空间的计算方法：

用户数据空间 = EEPROM 容量-336 字节；
MF 文件 = 用户数据空间；
DDF、ADF 文件 = 48 字节 + 空间大小
透明、变长记录、定长记录、循环记录文件 = 16 字节 + 文件大小
安全文件 = 12 字节 + 文件大小
电子存折文件 = 20 字节
电子钱包文件 = 16 字节

9.2.5.2.2 使用条件和安全

该命令的使用条件是：
1、卡经过了初始化；
2、建立 MF 前，应正确认证制造商密钥，初始的 MK 为‘0’值。
3、在 MF 下建立文件，必须通过 MF 的主控密钥 MK 的认证；
在 DDF 下建立文件，必须通过 DDF 的主控密钥 MK 的认证；
在 ADF 下建立文件，必须通过 ADF 的主控密钥 MK 的认证。

对于用户卡，在符合《中国金融集成电路（IC）卡规范》的应用环境中，ADF 下扩展电子钱包文件与电子钱包文件不能同时存在。

9.2.5.2.3 命令格式

代码	数　值
CLA	‘80’
INS	‘E0’
P1	‘00’：创建文件 ‘80’：个人化结束
P2	P1=‘00’： ‘00’：MF 文件 ‘01’：DDF 文件 ‘02’：ADF 文件 ‘03’：透明文件 ‘04’：线性定长文件 ‘05’：线性变长文件 ‘07’：循环定长文件 ‘09’：电子钱包文件 ‘0A’：电子存折文件 ‘0B’：安全文件 ‘0E’：扩展电子钱包文件 P1=‘80’：

	‘00’ 其它保留
Lc	DATA 域的数据长度
DATA	
MF	
11 到 27 字节	文件标识符(File-ID)——2 字节（必须是‘3F00H’） 环境类型（App-Type）——1 字节 保留字节(RFU)——1 字节（‘00H’） ATS 文件 SF(ATS-SFI)——1 字节 目录文件 SFI(DIR-SFI)——1 字节 FCI 文件 SFI(FCI-SFI)——1 字节 主控密钥控制属性（ACw）——1 字节 Reload 密钥标识符（RLD-KID）——1 字节 卡锁定密钥标识符(BLK-KID)——1 字节 主控密钥限制数(Limit)——1 字节 MF 应用 AID(MF-Name)—可选，最大 16 字节
DDF	
13 到 29 字节	文件标识符(File-ID)——2 字节 文件大小(LNG)——2 字节 环境类型(App-Type)——1 字节 保留字节(RFU)——2 字节（‘0000H’） 目录文件 SFI(DIR-SFI)——1 字节 FCI 文件 SFI(FCI-SFI)——1 字节 主控密钥控制属性(ACw)——1 字节 Reload 密钥标识符(RLD-KID)——1 字节 卡锁定密钥标识符(BLK-KID)——1 字节 主控密钥限制数(Limit)——1 字节 DDF 应用 AID(DDF-Name)—可选，最大 16 字节
ADF	
13 到 29 字节	文件标识符(File-ID)——2 字节 文件大小(LNG)——2 字节 应用类型——1 字节 保留字节(RFU)——3 字节（‘000000H’） FCI 文件 SFI（FCI-SFI）——1 字节 主控密钥控制属性(ACw)——1 字节 Reload 密钥标识符(RLD-KID)——1 字节 应用锁定/解锁密钥标识符(BLD-KID)——1 字节 主控密钥限制数(Limit)——1 字节 ADF 应用 AID(ADF-Name)—可选，最大 16 字节
安全文件	

8 字节	文件标识符(File-ID)——2 字节 文件大小(LNG)——2 字节 Reload 密钥标识符(WT-KID)——1 字节 文件控制属性(ACw)——1 字节 密钥改写权限(Write-Right)——2 字节
透明 EF 文件	
13 字节	文件标识符(File-ID)——2 字节 文件大小(LNG)——2 字节 保留字节(RFU)——1 字节（‘00H’） 文件读属性(ACr)——1 字节 文件写属性(ACW)——1 字节 文件读权限(Read-Right)——2 字节 文件写权限(Write-Right)——2 字节 文件读加密/校验密钥标识符(RT-KID)——1 字节 文件写加密/校验密钥标识符(WT-KID)——1 字节
线性定长记录 EF 文件	
15 字节	文件标识符(File-ID)——2 字节 记录的长度(RL)——1 字节 最大记录个数(RN)——1 字节 已存在记录的个数(RE)——1 字节 保留字节(RFU)——2 字节（‘0000H’） 文件读属性(ACr)——1 字节 文件写属性(AWr)——1 字节 文件读权限(Read-Right)——2 字节 文件写权限(Write-Right)——2 字节 文件读加密/校验密钥标识符(RT-KID)——1 字节 文件写加密/校验密钥标识符(WT-KID)——1 字节
线性变长记录 EF 文件	
14~(14+n) 个字节	文件标识符(File-ID)——2 字节 文件大小(LNG)——2 字节 保留字节(RFU)——2 字节（‘0000H’） 文件读属性(ACr)——1 字节 文件写属性(AWr)——1 字节 文件读权限(Read-Right)——2 字节 文件写权限(Write-Right)——2 字节 文件读加密/校验密钥标识符(RT-KID)——1 字节 文件写加密/校验密钥标识符(WT-KID)——1 字节 第 1 条记录的长度 L1——1 字节（可选） 第 2 条记录的长度 L2——1 字节（可选） …… 第 n 条记录的长度 Ln——1 字节（可选）
循环记录 EF 文件	

14 字节	文件标识符(File-ID)——2 字节 记录的长度(RL)——1 字节 最大记录个数(RN)——1 字节 已存在记录个数(RE)——1 字节 保留字节(RFU)——1 字节（‘00H’） 文件读属性(ACr)——1 字节 文件写属性(AWr)——1 字节 文件读权限(Read-Right)——2 字节 文件写权限(Write-Right)——2 字节 文件读加密/校验密钥标识符(RT-KID)——1 字节 文件写加密/校验密钥标识符(WT-KID)——1 字节
电子钱包 EF 文件	
6 字节	文件标识符(File-ID)——2 字节 余额上限(Bala-Limit)——4 字节
电子存折 EF 文件	
6 字节	文件标识符(File-ID)——2 字节 余额上限(Bala-Limit)——4 字节
扩展电子钱包 EF 文件	
11 字节	文件标识符(File-ID)——2 字节 余额上限(Bala-Limit)——4 字节 透支额度（over-amount）——4 字节 圈存交易明细 SFI（Load-List-SFI）——1 字节
Le	不存在

9.2.5.2.4 响应信息

响应信息中可能的状态码为：

SW1	SW2	说　明
90	00	命令执行成功
65	81	写 EEPROM 失败
67	00	Lc 长度错误
69	82	不满足安全状态
69	85	使用条件不满足
6A	80	数据域参数不正确（建立同名文件）
		1、建立 MF，FID 不是 3F00
		2、建立非 MF 文件，FID 是 3F00
		3、FID 是 0000 或 FFFF
		4、建立同名 FID 文件
		5、线性变长记录文件预开记录长度超过文件大小
		6、定长记录文件预开记录个数超过最大记录个数
		7、PBOC 环境下建立扩展电子钱包文件或者建设事业环境下建立电子钱包文件
6A	81	功能不支持
6A	84	空间已满

6A	86	P1、P2 参数错
6D	00	命令不存在
6E	00	CLA 错
93	03	应用永久锁定

9.2.5.2.5 命令详解

操作： 建立 MF（金融环境）

条件： File-ID： ’3F00’
App-Type： ’00’（金融环境，DES 算法）
RFU： ’00’
ATR-SFI： ’02’
DIR-SFI： ’03’
FCI-SFI： ’04’
ACw： ’39’（MK 长度为 16 字节，主控密钥用密文校验模式写入）
RLD-KID： ’00’（主控密钥重装密钥标识，使用主控密钥）
BLK-KID： ’00’（环境解锁密钥标识，使用主控密钥）
Limit： ’0F’（主控密钥限制次数为 15 次）
无 MF- NAME，
通过制造商密钥认证。

命令： ‘80E000000B’ + ‘3F00000002030438000 0F’

响应： ‘9000’

操作： 建立 MF（建设事业环境）

条件： File-ID： ’3F00’
App-Type： ’08’（建设事业环境，DES 算法）
RFU： ’00’
ATR-SFI： ’02’
DIR-SFI： ’03’
FCI-SFI： ’04’
ACw： ’39’（MK 长度为 16 字节，主控密钥用密文校验模式写入）
RLD-KID： ’00’（主控密钥重装密钥标识，使用主控密钥）
BLK-KID： ’00’（环境解锁密钥标识，使用主控密钥）
Limit： ’0F’（主控密钥限制次数为 15 次）
无 MF- NAME，
通过制造商密钥认证。

命令： ‘80E000000B’ + ‘3F00100020304380000 0F’

响应： ‘9000’

操作： 建立 DDF（金融环境）

条件： File-ID： ’DDF1’
LNG： ‘0100’
App-Type： ’00’（金融环境，DES 算法）
RFU： ’0000’
DIR-SFI： ’00’
FCI-SFI： ’00’

ACw： ’39’（MK 长度为 16 字节，主控密钥用密文校验模式写入）
RLD-KID： ’00’（主控密钥重装密钥标识，使用主控密钥）
BLK-KID： ’00’（环境解锁密钥标识，使用主控密钥）
Limit： ’0F’（主控密钥限制次数为 15 次）
无 DDF- NAME。
命令： ‘80E000010’ + ‘DDF10000010000000000100000F’
响应： ‘9000’

操作： 建立 DDF（建设事业环境）
条件： File-ID： ’DDF1’
LNG： ‘0100’
App-Type： ’08’（建设事业环境，DES 算法）
RFU： ’0000’
DIR-SFI： ’00’
FCI-SFI： ’00’
ACw： ’39’（MK 长度为 16 字节，主控密钥用密文校验模式写入）
RLD-KID： ’00’（主控密钥重装密钥标识，使用主控密钥）
BLK-KID： ’00’（环境解锁密钥标识，使用主控密钥）
Limit： ’0F’（主控密钥限制次数为 15 次）
无 DDF- NAME。
命令： ‘80E000010’ + ‘DDF10100010000000000100000F’
响应： ‘9000’

操作： 建立 ADF
条件： File-ID： ’ADF1’
LNG： ‘0600’
App-Type ’00’（金融环境，DES 算法）
RFU： ’000000’
FCI-SFI： ’00’
ACw： ’39’（MK 长度为 16 字节，主控密钥用密文校验模式写入）
RLD-KID： ’00’（主控密钥重装密钥标识，使用主控密钥）
BLK-KID： ’00’（环境解锁密钥标识，使用主控密钥）
Limit： ’0F’（主控密钥限制次数为 15 次）
ADF-NAME： ‘D15600000500’
命令： ‘80E0000213’ + ‘ADF1060000000000000900000FD15600000500’
响应： ‘9000’

操作： 建立 ADF
条件： File-ID： ’ADF1’
LNG： ‘0600’
App-Type ’08’（建设事业环境，DES 算法）
RFU： ’000000’
FCI-SFI： ’00’

ACw： ’39’（MK 长度为 16 字节，主控密钥用密文校验模式写入）
RLD-KID： ’00’（主控密钥重装密钥标识，使用主控密钥）
BLK-KID： ’00’（环境解锁密钥标识，使用主控密钥）
Limit： ’0F’（主控密钥限制次数为 15 次）
ADF-NAME： ‘D15600000500’

命令： ‘80E0000213’ + ‘ADF1060008000000000900000FD15600000500’
响应： ‘9000’

操作： 建立透明文件 EF14 0100 00 00 30 0000 0000 00 00

条件 File-ID： ’EF14’
LNG： ‘0100’
RFU： ’00’
ACR： ’00’（文件用明文方式读取，不需验证 PIN）
ACw： ’30’（文件用密文+MAC 方式写入，不需验证 PIN）
Read-Right： ’0000’（开放该文件的读权限）
Write-Right： ’0000’（开放该文件的写权限）
RT-KID： ’00’（读控制密钥使用当前环境或应用的主控密钥）
WT-KID： ’00’（写控制密钥使用当前环境或应用的主控密钥）

命令： ‘80E000030D’ + ‘EF1401000000300000000000000’
响应： ‘9000’

操作： 建立线性定长记录文件

条件： File-ID： ’EF24’
RL： ‘F7’（每条记录长度 247 字节）
RN： ‘02’（最大记录个数 2 条）
RE： ‘01’（预开 1 条记录）
RFU： ‘00’
RF： ‘00’（采用非 TLV 格式记录数据）
ACR： ’00’（文件用明文方式读取，不需验证 PIN）
ACW： ’30’（文件用密文+MAC 方式写入，不需验证 PIN）
Read-Right： ’0000’（开放该文件的读权限）
Write-Right： ’0000’（开放该文件的写权限）
RT-KID： ’00’（读控制密钥使用当前环境或应用的主控密钥）
WT-KID： ’00’（写控制密钥使用当前环境或应用的主控密钥）

命令： ‘80E000040F’ + ’ EF24F702010000003000000000000’
响应： ‘9000’

操作： 建立线性变长记录文件

条件： File-ID： ’EF34’

LNG： ‘0100’
RFU： ‘00’
RF： ‘00’（采用非 TLV 格式记录数据）
ACR： ’00’（文件用明文方式读取，不需验证 PIN）
ACW： ’30’（文件用密文+MAC 方式写入，不需验证 PIN）
Read-Right ’0000’（开放该文件的读权限）
：
Write-Right ’0000’（开放该文件的写权限）
：
RT-KID： ’00’（读控制密钥使用当前环境或应用的主控密钥）
WT-KID： ’00’（写控制密钥使用当前环境或应用的主控密钥）

命令： ‘80E000050E’ + ‘EF3401000000030000000000000’
响应： ‘9000’

操作： 建立循环定长记录文件

条件： File-ID： ’EF44’
RL： ‘F7’（每条记录长度 247 字节）
RN： ‘02’（最大记录个数 2 条）
RE： ‘01’（预开 1 条记录）
RF： ‘00’（采用非 TLV 格式记录数据）
ACR： ’00’（文件用明文方式读取，不需验证 PIN）
ACW： ’30’（文件用密文+MAC 方式写入，不需验证 PIN）
Read-Right ’0000’（开放该文件的读权限）
：
Write-Right ’0000’（开放该文件的写权限）
：
RT-KID： ’00’（读控制密钥使用当前环境或应用的主控密钥）
WT-KID： ’00’（写控制密钥使用当前环境或应用的主控密钥）

命令： ‘80E000070E’ + ‘EF44F702010000300000000000000’
响应： ‘9000’

操作： 建立安全文件

条件： File-ID： ’0001’
LNG： ‘01A8’
WT-KID： ‘00’（重装控制密钥为当前环境或应用的主控密钥）
ACw： ‘33’（写入密钥和修改密钥采用密文校验模式）
Write-Right ’0000’（开放密钥的修改权限）
：

命令： ‘80E0000B08’ + ‘000101A800330000’
响应： ‘9000’

操作： 建立电子钱包文件（金融）

条件： File-ID： ‘EB00’
Bala-Limit： ‘00002710’（ 余额上限为 10000 元）

命令： ‘80E0000906’ + ‘EB0000002710’
响应： ‘9000’

操作： 建立电子存折文件（金融）
条件： File-ID： ‘ED00’
Bala-Limit： ‘00002710’（ 余额上限为 10000 元）
命令： ‘80E0000A06’ + ‘ED0000002710’
响应： ‘9000’

操作： 建立扩展电子钱包文件
条件： File-ID： ‘0004’
Bala-Limit： ‘000186A0’（ 余额上限为 10000 元）
over-amount： ‘000186A0’
Load-List-SFI： ‘17’
命令： ‘80E0000E0B’ + ‘0004000186A0000186A017‘
响应： ‘9000’

9.2.5.3 FREEZE MF 命令

9.2.5.3.1 命令描述

FREEZE MF 锁定对 MF 文件的重构。在应用的建立阶段，为了调试，应用开发者可以反复对卡文件系统进行重构。而 FREEZE MF 命令提供了对此功能的锁定功能。即在成功的执行了 FREEZE MF 命令，MF 被冻结不可重构，但是建立 DF 和 EF、删除 DF 等功能仍然可以执行。

9.2.5.3.2 使用条件和安全

FREEZE MF 命令的执行必须在 MF 下，并且通过 MF 下主控 KEY 的认证。该命令成功执行后自动失效。

9.2.5.3.3 命令格式

代码	数　值
CLA	‘BF’
INS	‘FE’
P1	‘00’
P2	‘00’
Lc	不存在
DATA	不存在
Le	不存在

9.2.5.3.4 响应信息

响应信息中可能的状态码为：

SW1	SW2	说　明
90	00	命令执行成功
65	81	写 EEPROM 失败
69	82	不满足安全状态
69	85	使用条件不满足
6A	81	功能不支持
6A	86	P1、P2 参数错
6D	00	命令不存在
6E	00	CLA 错

9.2.5.3.5 命令详解

在成功认证 MF 主控密钥后，直接执行此命令即可。

9.2.5.4 GET INFO 命令

9.2.5.4.1 命令描述

GET INFO 命令读取卡的特征信息。卡的特征信息由卡供应商初始化卡时写入。

9.2.5.4.2 使用条件和安全

GET INFO 命令可以自由使用，不受安全条件限制，即使在卡锁定以后。

9.2.5.4.3 命令格式

代码	数　值
CLA	‘BF’
INS	‘C8’
P1	‘00’取芯片特征信息 ‘01’取工厂码
P2	‘00’
Lc	不存在
DATA	不存在
Le	P1=00 时 Le=‘0F’ P1=01 时 Le=‘20’

9.2.5.4.4 响应信息

响应信息中的数据：

芯片特征信息：

说　明	长度（字节）
芯片商注册标识号（‘0081’）	2
COS 标识符（‘4D’）	1
COS 版本号（‘36’）	1
COS 版本修订号（‘01’）	1
用户空间	2
卡片状态	1
应用环境	1
当前目录类型	1
当前应用状态	1
当前目录空间	2
当前目录剩余空间	2

卡片状态：

高 4 位表示卡片类型：
——‘0’，用户卡；

低 4 位表示卡片状态：
——‘A’，初始状态；

——‘B’，:开发状态；
——‘C’，工作状态；
——‘D’，锁定状态；
——‘E’，EEPROM 损坏
——‘9’，个人化状态

当前目录类型：
——‘39’，MF
——‘3A’，DDF
——‘38’，ADF

当前应用环境类型：
——‘00’，金融应用，使用 DES 算法
——‘20’，金融应用，使用 SM1 算法
——‘08’，建设事业应用，使用 DES 算法
——‘28’，建设事业应用，使用 SM1 算法

当前应用状态：
——‘38’，工作状态
——‘78’，临时锁定状态
——‘B8’，永久锁定状态

响应信息中可能的状态码为：

SW1	SW2	说　明
90	00	命令执行成功
67	00	Lc 长度错误
6A	86	P1、P2 参数错
6D	00	命令不存在
6E	00	CLA 错

9.2.5.4.5 命令详解

操作：　取卡内特征信息
命令：　‘BFC80000 0F’
响应：　‘00814D22103EB00B0139383EB01E50’
　　芯片商注册标识号：　‘0081’
　　COS标识符：　‘4D’
　　COS版本号：　‘36’
　　COS版本修订号：　‘01’
　　EEPROM空间：　‘259E’
　　卡片状态：　‘0B’（开发状态）
　　当前规范类型：　‘08’
　　当前目录类型：　‘39’（MF）
　　当前应用状态：　‘38’（工作状态）
　　当前目录空间：　‘259E’
　　当前目录剩余空间：　‘1868’

9.2.5.5 WRITE KEY 命令

9.2.5.5.1 命令描述

WRITE KEY 命令用于在指定安全文件中初始建立一个密钥记录。

9.2.5.5.2 使用条件和安全

在个人化状态下，WRITE KEY 命令的执行是明文方式，并且在执行该命令前，不需要满足文件的写条件。在应用状态下，WRITE KEY 命令执行必须满足相应 KEY 文件的写条件和写属性，并要符合 ACw 中规定的写模式。WRITE KEY 命令安全报文应符合金融规范。

9.2.5.5.3 命令格式

代码	数　值
CLA	‘80’或‘84’
INS	‘D4’
P1	‘00’建立新密钥　（P2≠‘00’） ‘01’重装已有密钥（P2=‘00’时表示主控密钥）
P2	密钥文件标识
Lc	DATA 域的数据长度
DATA	加密或不加密的密钥信息 MAC 数据元(4 字节，由 KEY 文件的 ACw 决定 MAC 是否存在)
Le	不存在

当 P1=‘00’时，“密钥信息”是指安全文件定义的四种密钥结构定义的格式。

当 P1=‘01’时，“密钥信息”是指：用途＋标识＋版本＋密钥值本身。

9.2.5.5.4 响应信息

响应信息中可能的状态码为：

SW1	SW2	说　明
90	00	命令执行成功
65	81	写 EEPROM 失败
67	00	Lc 长度错误
69	82	不满足安全状态
69	83	认证密钥锁定
69	84	引用数据无效（未申请随机数）
69	85	使用条件不满足
69	88	安全信息（MAC 和密文）数据错误
6A	80	数据域参数错误
6A	81	功能不支持
6A	82	未找到文件
6A	83	未找到密钥数据
6A	84	文件空间已满

6A	86	P1、P2 参数错
6A	88	未找到密钥数据
6D	00	命令不存在
6E	00	CLA 错
93	03	应用永久锁定

9.2.5.5.5 命令详解

1、DES 算法

操作 1： 向标识符为’0001’的安全文件中新建密钥。

条件： 用途 标识 版本 算法 Access-Right Limit ‘00’

‘01’ ‘02’ ‘03’ ‘00’ ‘0000’ ‘03’ ‘00’

密钥值：’ 00000000000000000000000000000000’

密文＋MAC 方式写入密钥。

步骤 1： 用当前环境主控密钥 MK（本例取 16 字节’00’）计算密文

DataIn=’0102030000000300000000000000000000000000000000’

3DES Encryption（MK，(Ld+DataIn)，EncData）

密文 EncData =’

6A0F5774A86E4E3E8CA64DE9C1B123A78CA64DE9C1B123A7D9031B0271BD5A0A’

步骤 2： 取 8 字节随机数 Random，作为 MAC 计算初始值。

‘00840000 08’

Random = ’ DA0CF4E19D8C8549’

步骤 3： 用 MK 计算 MAC

初始值 InitialData=’ DA0CF4E19D8C8549’ 8 字节

命令头 HeadData= ‘84D4000124’

MAC 计算数据为：MACData = InitialData + HeadData + EncData

3DES MAC（MK，MACData，MAC）

MAC=‘198C5CD3’

步骤 4： 密文＋MAC 方式密钥

命令： ‘84D4000124’ + EncData + MAC

响应： ‘9000’

操作 2： 更新主控密钥。

条件： 用途 标识 版本

‘00’ ‘00’ ‘00’

密钥值：’ 00000000000000000000000000000000’

密文＋MAC 方式写入密钥。

步骤 1： 用当前环境主控密钥 MK（本例取 16 字节’00’）计算密文

DataIn=’0000000000000000000000000000000000000000’

3DES Encryption（MK，(Ld+DataIn)，EncData）

密文 EncData =’

433CF82F6BDA75F68CA64DE9C1B123A7E943D7568AEC0C5C’

步骤 2： 取 8 字节随机数 Random，作为 MAC 计算初始值。
命令： ‘00840000 08’
响应： Random = ’ 7E6AF6769EF3A06E’
步骤 3： 用 MK 计算 MAC
初始值 InitialData=’ 7E6AF6769EF3A06E’ 8 字节
命令头 HeadData= ‘84D401001C’
MAC 计算数据为：MACData = InitialData + HeadData + EncData
3DES MAC（MK，MACData，MAC）
MAC=‘73AB2406’
步骤 4： 密文＋MAC 写入密钥
命令： ‘84D401001C’+ EncData + MAC
响应： ‘9000’

2、SM1 算法

操作 1： 向标识符为’0001’的安全文件中新建密钥。

条件：	用途	标识	版本	算法	Access-Right	Limit	‘00’
	‘01’	‘02’	‘03’	‘02’	‘0000’	‘03’	‘00’

密钥值：’ 00000000000000000000000000000000’
密文＋MAC 方式写入密钥。
步骤 1： 用当前环境主控密钥 MK（本例取 16 字节’00’）计算密文
DataIn=’0102030200000300000000000000000000000000000000’
SSF33 Encryption（MK，(Ld+DataIn)，EncData）
密文 EncData =’
E4E7666D054792D86F5975F3F4ED4D50418600C58DB062809FB8F66DFDC6AA7F’
步骤 2： 取 16 字节随机数 Random，作为 MAC 计算初始值。
‘00840000 10’
Random = ’AA019A223D42DBE9F2C9F58E09FFC2F0’
步骤 3： 用 MK 计算 MAC
初始值 InitialData=’ AA019A223D42DBE9F2C9F58E09FFC2F0’ 16 字节
命令头 HeadData= ‘84D4000124’
MAC 计算数据为：MACData = InitialData + HeadData + EncData
SSF33 MAC（MK，MACData，MAC）
MAC=‘AECC6B6C’
步骤 4： 密文＋MAC 方式密钥
命令： ‘84D4000124’ + EncData + MAC
响应： ‘9000’

操作 2： 更新主控密钥。

条件：	用途	标识	版本
	‘00’	‘00’	‘00’

密钥值：’ 00000000000000000000000000000000’
密文＋MAC 方式写入密钥。

步骤 1： 用当前环境主控密钥 MK（本例取 16 字节’00’）计算密文
DataIn=’00000000000000000000000000000000’
SSF33 Encryption（MK，(Ld+DataIn)，EncData）
密文 EncData =’
‘968CC1ADD0EB93372EDEFF72B70942AB313FEEFB5A08084CF2BAE3AC518EB49D’
步骤 2： 取 16 字节随机数 Random，作为 MAC 计算初始值。
命令： ‘00840000 10’
响应： Random = ’ 03FCCF7D45E603961A37B4449E437E27’
步骤 3： 用 MK 计算 MAC
初始值 InitialData=’ 03FCCF7D45E603961A37B4449E437E27’ 16 字节
命令头 HeadData= ‘84D401001C’
MAC 计算数据为：MACData = InitialData + HeadData + EncData
SSF33 MAC（MK，MACData，MAC）
MAC=‘62109E7A’
步骤 4： 密文＋MAC 写入密钥
命令： ‘84D4010024’+ EncData + MAC
响应： ‘9000’

10 卡片个人化

卡片个人化是对卡片根据应用的需要所进行的操作。

卡片个人化所要完成的任务：

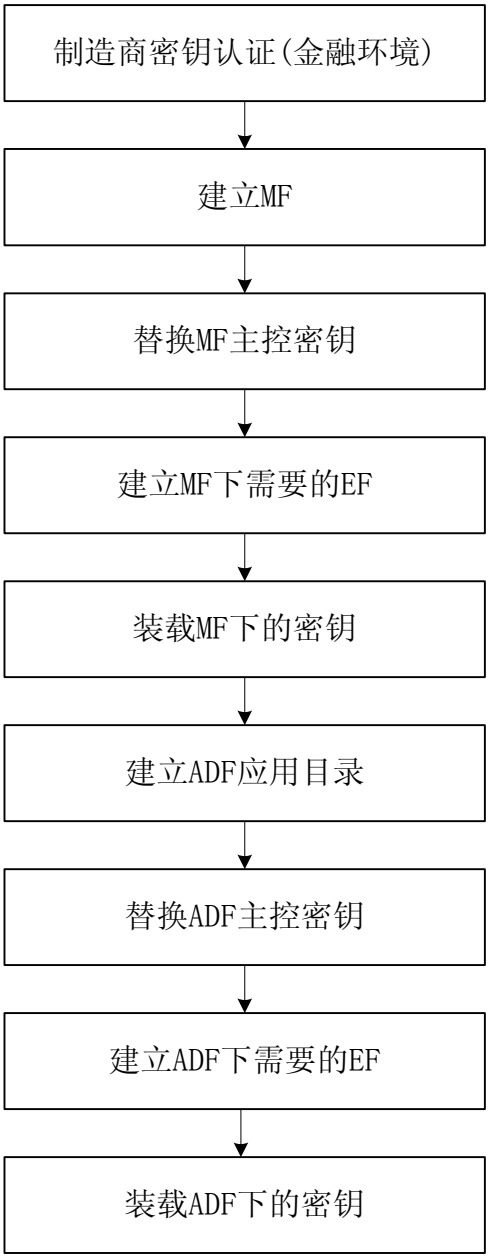
1、创建应用所需要的文件结构；

2、完成应用所需要的密钥的装载；

3、写入应用要求的数据。

结束个人化，卡片将进入到调试状态。个人化阶段不允许执行金融专用命令、金融扩展命令、建设事业专用命令。

卡片个人化简易流程见下图：



在卡片进入个人化状态后，需要为卡片建立文件系统，并装载密钥，写入应用数据，这个过程为个人化阶段。由于个人化是在特定环境中进行，为了加快个人化速度，个人化阶段对部分用户接口命令将不应用卡片的安全策略，即对密钥写入命令、应用数据的部分读写命令不应用文件的读写条件和读写属性控制机制、对部分改写 EEPROM 数据的用户命令不使用镜像保护机制。在个人化完成后，通过一个明显的个人化结束动作，结束个人化状态，启用文件的安全控制机制。

执行Write Key、 Append Record、Update Record、Read Record、Update

Binary、Read Binary命令，可以忽略密钥文件访问条件，数据可以以明文方式传送给卡片。

所有带有写操作的命令都不启动镜像保护机制。

11 交易流程

11.1 金融应用交易流程

11.1.1 交易预处理

图 11-1给出了对电子存折/电子钱包应用的所有交易类型共有的预处理流程。

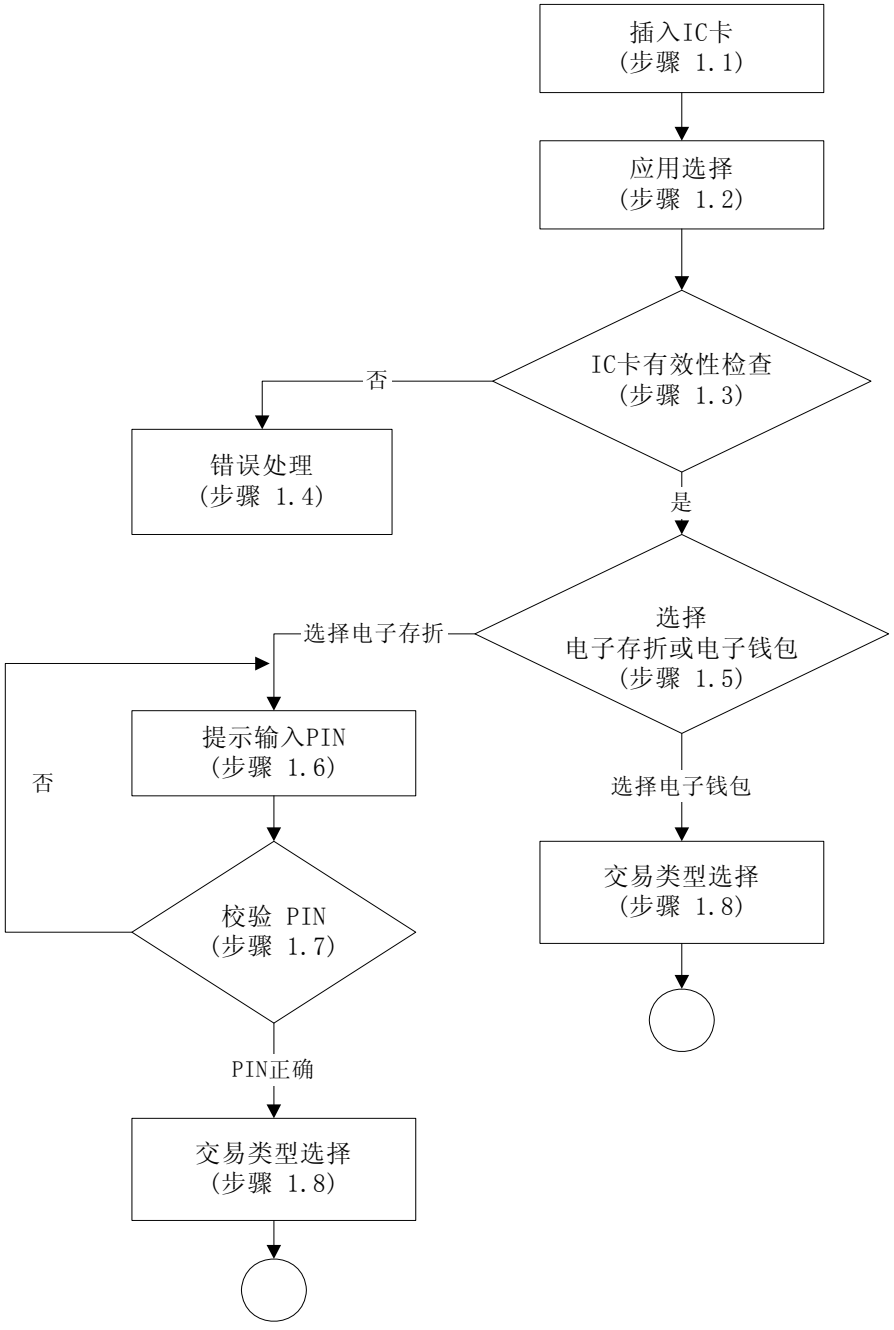


图 11-1 交易预处理流程

11.1.1.1 插入 IC 卡(步骤 1.1)

终端应具有检测 IC 卡是否已经插入读卡器的功能。如果 IC 卡已经插入，终端将继续执行 11.1.1.2 节的应用选择功能。

11.1.1.2 应用选择(步骤 1.2)

执行 SELECT FILE 命令进行应用选择。

应用选择的执行过程请参见《中国金融集成电路(IC)卡规范》第 1 部分:卡片规范的“应用选择”部分。电子存折/电子钱包应用的应用标识符（AID）将由全国金融标准化技术委员会负责分配和维护。

成功地选择了电子存折/电子钱包应用后，IC 卡回送包含发卡方专用数据在内的文件控制信息 FCI。下表定义了此应用必备的发卡方专用数据。

数据字段的描述	长度 (字节)
发卡方机构标识符	8
应用类型标识	1
应用版本号	1
应用序列号	10
应用启用日期	4
应用有效日期	4
发卡方自定义 FCI 数据	2

应用类型标识 (ATI) 在应用选择时由 IC 卡回送给终端。它标明电子存折/电子钱包应用在卡上的存在情况。

11.1.1.3 IC 卡有效性检查(步骤 1.3)

对于 SELECT 命令送回的数据，终端将对这些数据进行以下检查：

——该卡是否在终端存储的黑名单卡之列（使用发卡方标识和应用序列号）；

——终端是否支持该发卡方标识符；

——终端是否支持 IC 卡上的应用(使用应用类型标识(ATI)来检查)；

——终端是否支持从 IC 卡回送的应用版本号所代表的应用版本；

——应用是否在有效期内。

如果以上任一条件不满足，交易将按 11.1.1.4 节描述进行，否则按 11.1.1.5 中的描述进行。

11.1.1.4 错误处理(步骤 1.4)

以上任一条件不满足时终端所作的处理不在本手册范围内。

11.1.1.5 选择电子存折或电子钱包(步骤 1.5)

终端根据应用选择时获得的应用类型标识判别 IC 卡支持 ED、EP 的情况。

如果 IC 卡和终端只同时支持 ED 或 EP 之一，则终端将自动地选择到 ED 或 EP，继而进行 11.1.1.6 或 11.1.1.8 中所描述的步骤。

如果 IC 卡仅支持一种应用并且该应用不被终端支持，则该过程终止。

如果 IC 卡和终端彼此都支持 ED 和 EP 两种应用，终端应向持卡人提供选择 ED 或 EP 的过程，在这一过程中持卡人可以从中选择一种应用进行交易。

11.1.1.6 提示输入持卡人密码（PIN）(步骤 1.6)

选择了电子存折/电子钱包的应用后，终端将提示持卡人输入 PIN。

11.1.1.7 校验 PIN(步骤 1.7)

持卡人输入 PIN 后，终端将用 VERIFY 命令来校验持卡人输入的 PIN 是否正确。VERIFY 命令在本手册中命令部分定义。

当 IC 卡收到校验（VERIFY）命令后，它将进行以下操作：

——检查 PIN 尝试计数器。如果 PIN 尝试计数器为零时，PIN 被锁住且不能执行相应的命令。这种情况下，IC 卡返回状态‘6983’（认证方式锁定），终端结束交易过程。

——如果 PIN 没有被锁，将命令数据中的 PIN 和 IC 卡中存放的 PIN 进行比较。

——如果以上两 PIN 相同，IC 卡将 PIN 尝试计数器置为 PIN 重试的最大次数并返回状态‘9000’。IC 卡必须在断电之前或选择其他应用前记住 PIN 已经成功验证。交易处理按 11.1.1.8 节的描述继续进行。

——如果以上两 PIN 并不相同，IC 卡将 PIN 尝试计数器减 1 并返回状态‘63Cx’，这里’x’是 PIN 尝试计数器的新值。在这种情况下，终端将检查 x 的值。如果 x 是零，将终止交易并且卡片自动锁 PIN。否则，终端将提示重新输入 PIN 并且重复以上过程。

如果持卡人输入的 PIN 是正确的，交易流程执行 11.1.1.8 节。

11.1.1.8 交易类型选择(步骤 1.8)

终端应该具备让持卡人选择交易类型的功能。每次交易最多只能选择一种交易类型。

对电子存折应用来说，持卡人应能选择如下交易类型：圈存、 圈提、消费、取现、修改透支限额、查询余额、查询明细。

对电子钱包应用来说，持卡人应能选择如下交易类型：圈存、消费、查询余额。

11.1.2 圈存交易

通过圈存交易，持卡人可将其在银行帐户上资金划入电子存折或电子钱包中。这种交易必须在金融终端上联机进行并要求提交个人密码（PIN）（无论电子存折或电子钱包）。

11.1.2.1 发出 INITIALIZE FOR LOAD 命令(步骤 2.1)

终端述发出 INITIALIZE FOR LOAD 指令启动圈存交易。

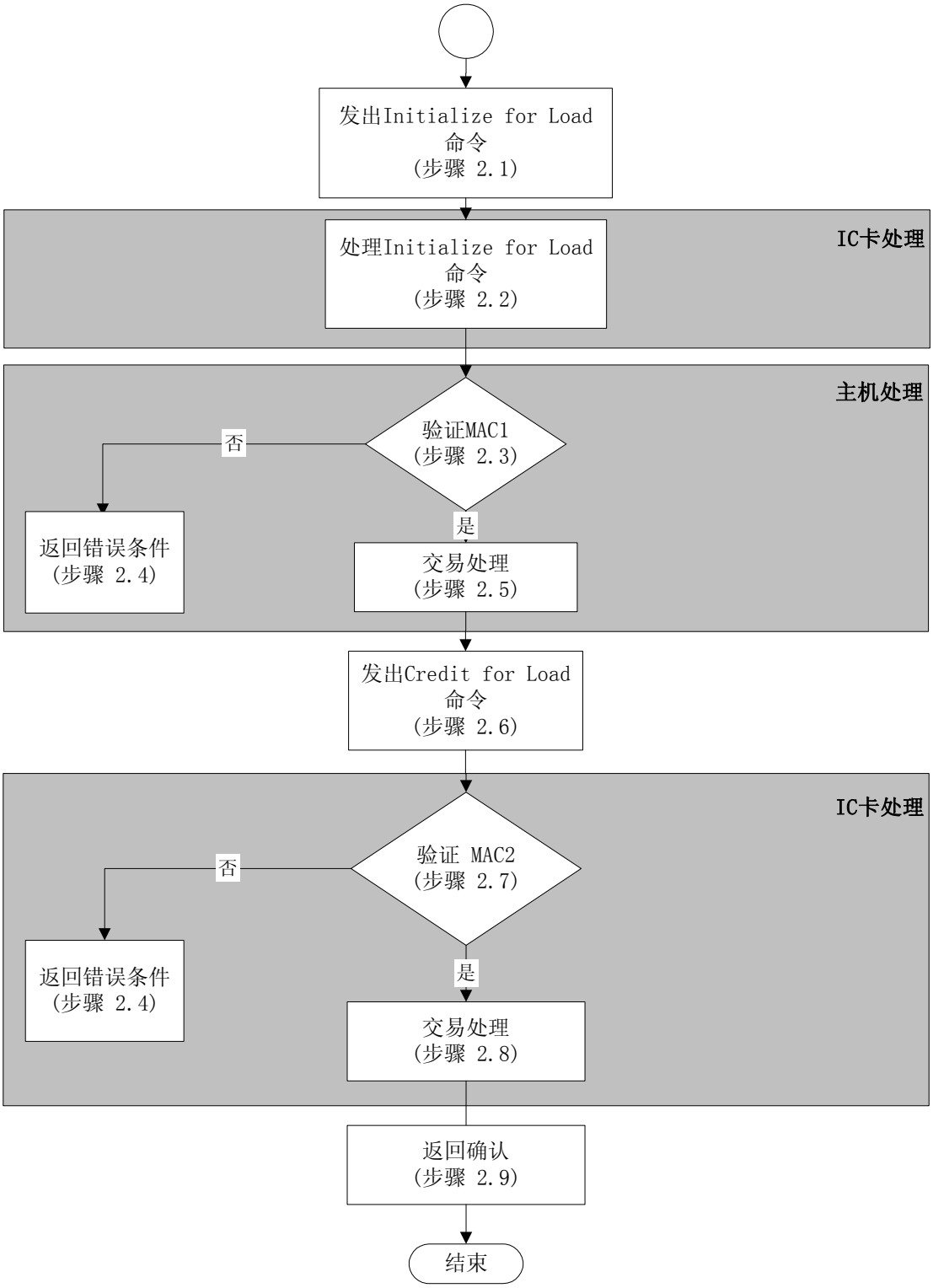


图 11-2 圈存交易处理流程

11.1.2.2 处理圈存初始化(步骤 2.2)

收到圈存初始化命令后, IC 卡将进行以下操作：

——检查命令中包含的密钥索引是否能被 IC 卡支持。如果不支持，返回状态码‘9403’（不支持的密钥索引）且不返回其他数据，命令的处理结束。

——IC 卡产生一个伪随机数（ICC），过程密钥 SESLK 和一个报文验证码（MAC1 表示），以以供主机验证圈存交易和 IC 卡的合法性。

过程密钥 SESLK 被用于电子存折或电子钱包的圈存交易。过程密钥是用 DLK 密钥和安全管理一章中的机制产生的。用来产生过程密钥的输入数据如下:

SESLK：伪随机数（ICC）||电子存折联机交易序号或电子钱包联机交易序号||‘8000’

MAC1 的计算机制见安全管理一章。SESLK 作用于以下数据进行 MAC1 计算(按所列顺序):

——电子存折或电子钱包余额

——交易金额

——交易类型标识

——终端机编号

IC 卡将把 INITIALIZE FOR LOAD 命令的响应报文送给终端处理。如果 IC 卡返回的状态不是‘9000’，终端将终止交易。

11.1.2.3 验证 MAC1(步骤 2.3)

收到圈存初始化命令的响应报文后，终端把该响应报文定义的数据传给发卡机构主机。主机将生成 SESLK 并且确认 MAC1 是否有效。如果 MAC1 有效，交易处理将按 11.1.2.5 节描述的继续执行。如果 MAC1 无效，交易处理将执行 11.1.2.4 中所描述的步骤节。

11.1.2.4 返回错误状态(步骤 2.4)

如果出现使圈存交易不能被接受的条件，则主机应通知终端。送给终端的报文格式和内容，以及终端采取的动作在本手册的讨论范围以外。

11.1.2.5 交易处理(步骤 2.5)

在确认能够进行圈存交易后，主机从持卡人在银行的相应帐户中减去持卡人输入的圈存金额。

主机也会产生一个报文验证码(MAC2 表示)，供 IC 卡对主机合法性进行检查。安全管理一章中描述了主机用来生成 MAC2 的机制。SESLK 作用于以下数据进行 MAC2 计算(按所列顺序):

——交易金额

——交易类型标识

——终端机编号

——交易日期 (主机)

——交易时间 (主机)

在成功地进行了圈存交易后，主机将电子存折联机交易序号或电子钱包联机交易序号加 1，并发送一个圈存交易接受报文给终端，其中包括 MAC2、交易日期（主机）和交易时间（主机）。

11.1.2.6 发出 CREDIT FOR LOAD 命令(步骤 2.6)

在收到主机的圈存交易接受报文后，终端会发出 CREDIT FOR LOAD 命令给 IC 卡以更新卡上电子存折或电子钱包余额。

11.1.2.7 验证 MAC2 (步骤 2.7)

收到 CREDIT FOR LOAD 命令后, IC 卡必须确认 MAC2 是有效的。如果 MAC2 有效, 交易处理将执行 11.1.2.8 节。 如果 MAC2 无效，状态‘9302’（MAC 无效）会被返回给终端。终端对错误所应采取相应的动作。

11.1.2.8 交易处理(步骤 2.8)

IC 卡将电子存折或电子钱包联机交易序号加 1，并且把交易金额加在电子存折或电子钱包的余额上。IC 卡必须全部成功地完成以上几个步骤或者一个也不完成。

在电子存折或电子钱包圈存交易中，IC 卡用以下数据组成的一个记录更新交易明细：

——电子存折或电子钱包联机交易序号

——交易金额

——交易类型标识

——终端机编号

——交易日期 (主机)

——交易时间 (主机)

TAC 的计算机制见安全管理一章。TAC 的计算不用过程密钥方式，它用 DTK 左右 8 位字节异或运算的结果对以下数据进行加密运算来产生(按所列顺序):

——电子存折或电子钱包余额

——电子存折或电子钱包联机交易序号（加 1 前）

——交易金额

——交易类型标识

——终端机编号

——交易日期（主机）

——交易时间（主机）

11.1.2.9 返回确认(步骤 2.9)

在成功完成步骤 11.1.2.8 后， IC 卡将 CREDIT FOR LOAD 命令的响应报文 TAC 返回给终端。主机可以不马上验证 TAC。

11.1.3 圈提交易

通过圈提交易，持卡人可以把电子存折中的部分或全部资金划回到其相应账户上。这种交易必须在金融终端上联机进行并要求验证个人密码（PIN）。只有电子存折应用支持圈提交易。

11.1.3.1 发出 INITIALIZE FOR UNLOAD 命令(步骤 3.1)

终端发出 INITIALIZE FOR UNLOAD 命令启动圈提交易。

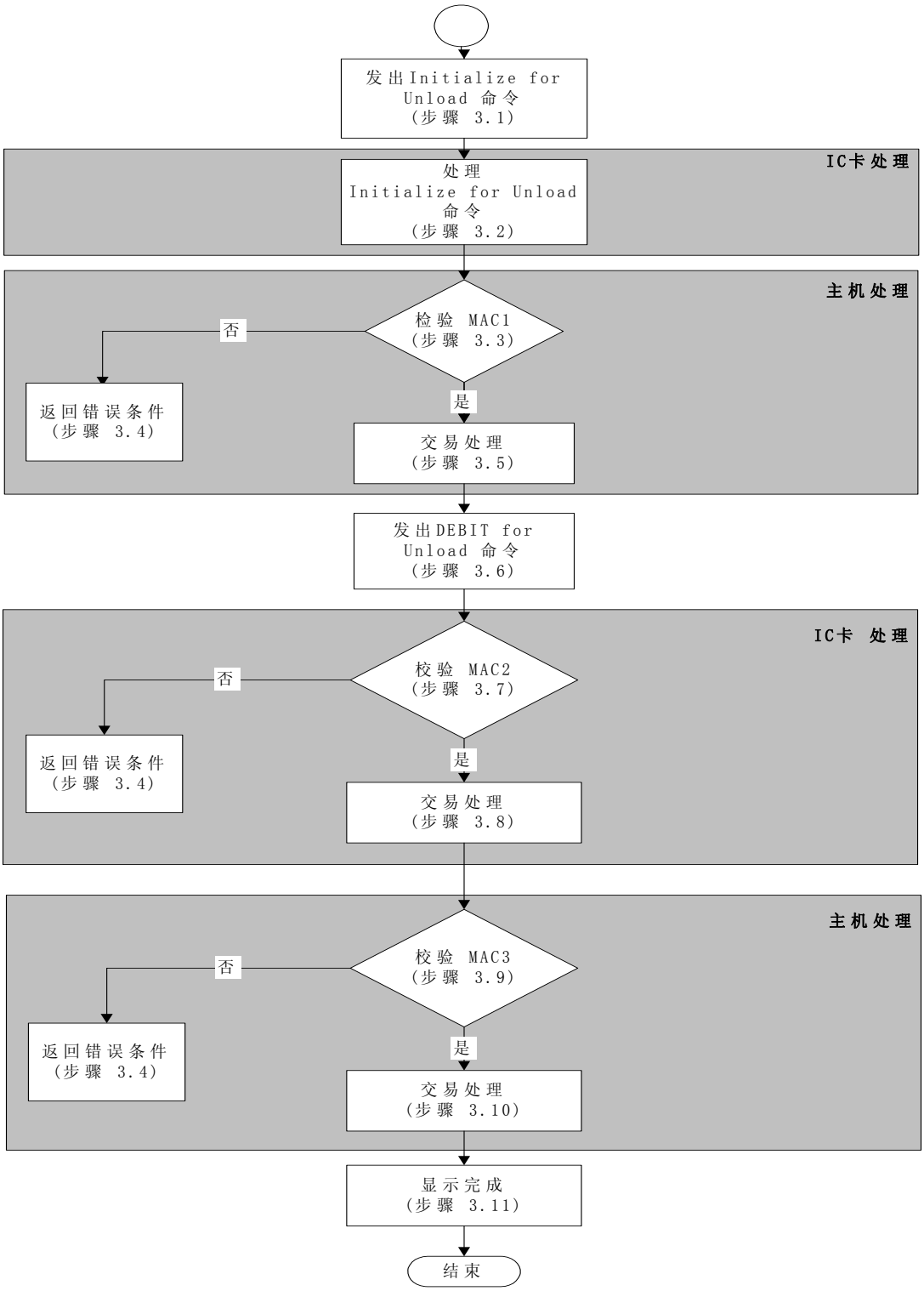


图 11-3 圈提交易处理流程

11.1.3.2 处理 INITIALIZE FOR UNLOAD 命令(步骤 3.2)

收到 INITIALIZE FOR UNLOAD 命令后，IC 卡将进行一下操作：

——检查是否支持命令中提到的密钥索引号。如果不支持，则回送状态码‘9403’（不支持的密钥索引）。但不回送任何其他数据，命令处理结束。

——检查命令中包括的交易金额是否超过电子存折余额。如果超过，则回送状态码

‘9401’（资金不足），但不回送任何其他数据。

在通过以上检查后，IC 卡将产生一个伪随机数（ICC）、过程密钥 SESULK 和一个报文鉴别码（MAC1），供主机验证圈提交易及 IC 卡的合法性。

SESULK 是用于电子存折圈提交易的过程密钥。该过程密钥是利用 DULK 并按照安全管理中过程密钥一节所描述的机制产生的。用来产生该过程密钥的输入数据如下：

SESULK：伪随机数（ICC）||电子存折联机交易序号||‘8000’

MAC1 的计算机制见安全管理一章。用 SESULK 对以下数据加密产生 MAC1（按所列顺序）：

—电子存折余额（交易前）

—交易金额

—交易类型标识

—终端机编号

IC 卡应向终端回送前面所定义的 INITIALIZE FOR UNLOAD 命令的响应报文和状态码‘9000’。在收到 INITIALIAZE FOR UNLOAD 的响应报文后，终端将一个包含 INITIALIAZE FOR UNLOAD 响应报文数据域中规定的数据的圈提许可请求报文 MAC1 送往发卡方主机。

11.1.3.3 验证 MAC1(步骤 3.3)

主机将产生 SESULK 并验证 MAC1 是否有效。如果 MAC1 有效，将执行 11.1.3.5 中的步骤。否则终端回送一个错误状态码，交易处理将转而执行 11.1.3.4 中所描述的步骤。

11.1.3.4 回送错误状态(步骤 3.4)

如果不接受圈提交易，主机应通知终端。终端的处理方式不在本手册范围之内。

11.1.3.5 主机处理(步骤 3.5)

主机确认能够进行圈提交易后，将产生一个报文鉴别码（MAC2），以供 IC 卡对主机合法性检查。下面列出包含在 DEBIT FOR UNLOAD 命令中从主机经由终端传到 IC 卡的数据。

MAC2 的计算机制见安全管理一章。用 SESULK 对以下数据进行加密（按所列顺序）产生 MAC2：

—交易金额

—交易类型标识

—终端机编号

—交易日期（主机）

—交易时间（主机）

主机向终端发送一个圈提交易接受报文，其中至少应包括交易日期（主机）、交易时间（主机）和 MAC2。

11.1.3.6 发出 DEBIT FOR UNLOAD 命令(步骤 3.6)

终端收到主机的圈提交易接受报文后，向 IC 卡发出 DEBIT FOR UNLOAD 命令以更新卡上电子存折余额。

11.1.3.7 验证 MAC2(步骤 3.7)

IC 卡必须确认 MAC2 是有效的。如果 MAC2 有效，交易处理将执行 11.1.3.8 中所描述的步骤。否则向终端回送状态码‘9302’（MAC2）无效。

11.1.3.8 交易处理(步骤 3.8)

IC 卡将电子存折联机交易序号加 1，并从卡上的电子存折余额中扣减交易金额。IC 卡必须成功地完成以上所有步骤或者一个也不完成。

IC 卡将产生报文鉴别码（MAC3）。并通过 DEBIT FOR UNLOAD 命令的响应报文将以下数据经终端送往主机。

MAC3 的计算机制见安全管理一章。用 SESULK 对以下数据加密产生 MAC3（按所列顺序）：

—电子存折余额（交易后）
—电子存折联机交易序号（加 1 前）
—交易金额
—交易类型标识
—终端机编号
—交易日期（主机）
—交易时间（主机）

IC 卡用以下数据组成的一个记录更新交易明细：

—电子存折联机交易序号（加 1 前）
—交易金额
—交易类型标识
—终端机编号
—交易日期（主机）
—交易时间（主机）

11.1.3.9 验证 MAC3(步骤 3.9)

主机收到（经由终端）IC 卡回送的 MAC3 后，应确认 MAC3 是否有误。如果 MAC3 有效，交易处理将执行 11.1.3.10 中描述的步骤。否则将向终端回送一个错误状态码。

11.1.3.10 交易处理(步骤 3.10)

发卡方主机将交易金额加在持卡人的相应银行账户上，并将主机的电子存折联机交易序号加 1。

主机向终端回送一个完成报文，表示持卡人的账户已更新。本手册不规定报文的内容和形式。

11.1.3.11 显示完成(步骤 3.11)

在收到主机的完成报文后，终端将向持卡人显示交易完成信息。

如果需要，终端应能向持卡人提供交易纸凭证。

11.1.4 消费交易

消费交易允许持卡人使用电子存折或电子钱包的余额进行消费。此交易可以在销售点终端（POS）上脱机进行。使用电子存折进行的消费交易必须提交个人密码（PIN），使用电子钱包则不需要。

11.1.4.1 发出 INITIALIZE FOR PURCHASE 命令(步骤 4.1)

终端发出 INITIALIZE FOR PURCHASE 命令启动消费交易

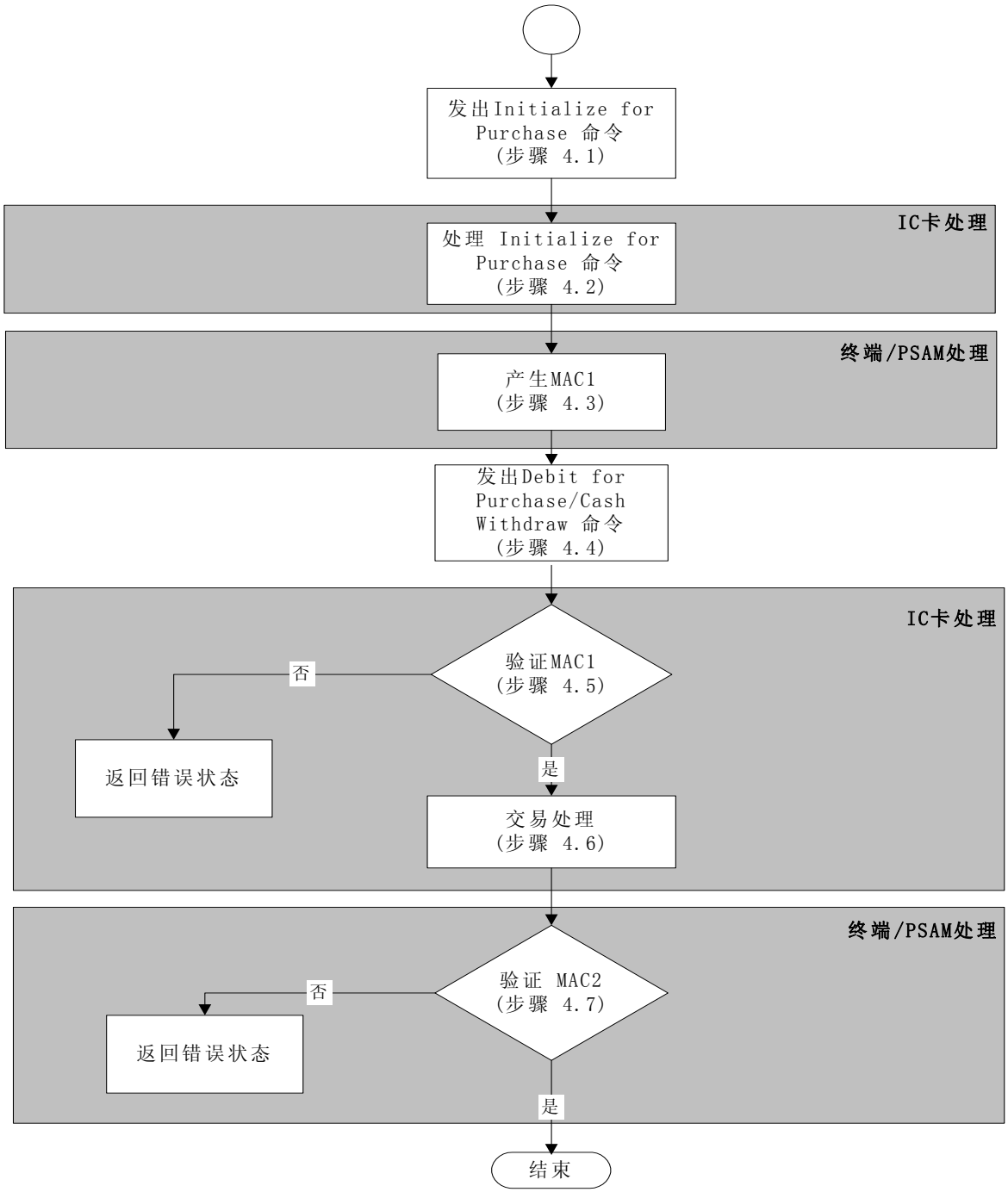


图 11-4 消费交易处理流程

11.1.4.2 处理 INITIALIZE FOR PURCHASE 命令(步骤 4.2)

IC 卡收到 INITIALIZE FOR PURCHASE 命令后，IC 卡将进行以下操作：

——检查命令中包含的密钥索引是否被 IC 卡支持。如果不支持，返回状态码‘9403’（不支持的密钥索引）且不返回其他数据。

——检查电子存折余额或电子钱包余额是否大于或等于交易金额。如果小于交易金额，状态码‘9401’（资金不足）返回给终端，不返回其他数据。终端采取的措施不在本手册的范围内。

如果以上检验均无错误，IC 卡产生一个伪随机数（ICC），过程密钥 SESPK 以用于验证 MAC1。过程密钥是用 DPK 并按照安全管理一章中描述的机制产生的。用于生成过程密钥的输入数据如下：

SESPK：伪随机数（ICC）||电子存折脱机交易序号或电子钱包脱机交易序号||终端交易序号的最右两个字节。

11.1.4.3 产生 MAC1(步骤 4.3)

使用伪随机数（ICC）和 IC 卡返回的电子存折脱机交易序号或电子钱包脱机交易序号，终端的安全存取模块（PSAM）将产生一样的过程密钥（SESPK）和一个报文认证码（MAC1），供 IC 卡来验证 PSAM 的合法性。

MAC1 的计算机制见安全管理一章。SESPK 作用于以下数据进行 MAC1 的计算(按所列顺序):

——交易金额

——交易类型标识

——终端机编号

——交易日期（终端）

——交易时间（终端）

11.1.4.4 发出 DEBIT FOR PURCHASE/CASH WITHDRAW 命令 (步骤 4.4)

终端发出 DEBIT FOR PURCHASE/CASH WITHDRAW 命令。

11.1.4.5 验证 MAC1(步骤 4.5)

在收到 DEBIT FOR PURCHASE 命令后，IC 卡要验证 MAC1 的有效性。如果 MAC1 是有效的, 交易处理将继续执行 11.1.4.6 节。如果 MAC1 是无效的，错误状态‘9302’（MAC 无效）被返回给终端。

11.1.4.6 交易处理(步骤 4.6)

IC 卡从电子存折或电子钱包余额中扣减消费的金额，将电子存折或电子钱包脱机交易序号加 1。IC 卡必须全部成功地完成以上几个步骤或者一个也不完成，如果余额或序号的更新均没有成功，交易明细也不应被更新。

IC 卡产生一个报文验证码 (MAC2 表示)供 PSAM 对 IC 卡合法性进行检查。并通过 DEBIT FOR PURCHASE 命令响应报文回送以下数据，作为 PSAM 产生 MAC2 的输入数据。MAC2 的产生机制参见安全管理一章。用 SESPK 于以下数据进行加密产生 MAC2：

——交易金额

IC 卡按照安全管理一章中描述的机制用密钥 DTK 左右 8 字节异或运算后的结果产生 TAC。TAC 将被写入终端交易明细，以便后来传给主机进行交易验证。下面是用来生成 TAC 的数据，它们以明文形式包含在 CREDIT FOR PURCHASE 命令的响应报文中从 IC 卡传传到终端：

——交易金额
——交易类型标识
——终端机编号
——终端交易序号
——交易日期 (终端)
——交易时间 (终端)

对于电子存折消费交易，IC 卡将用以下数据组成的一个记录更新交易明细。

——电子存折脱机交易序号
——交易金额
——交易类型标识
——终端机编号
——交易日期 (终端)
——交易时间 (终端)

11.1.4.7 验证 MAC2 (步骤 4.7)

收到从 IC 卡(经过终端)传来的 MAC2 后, PSAM 要验证 MAC2 的有效性。MAC2 验证的结果被传送到终端以便采取必要的措施。终端采取的措施不在本手册的范围之内。

11.1.5 取现交易

取现交易允许持卡人从电子存折中提取现金。此交易必须在金融终端上进行，但可以脱机处理。只有电子存折应用支持此交易,且必须提交个人密码 PIN。

11.1.5.1 发出 INITIALIZE FOR CASH WITHDRAW 命令(步骤 5.1)

终端发出 INITIALIZE FOR CASH WITHDRAW 命令启动取现交易

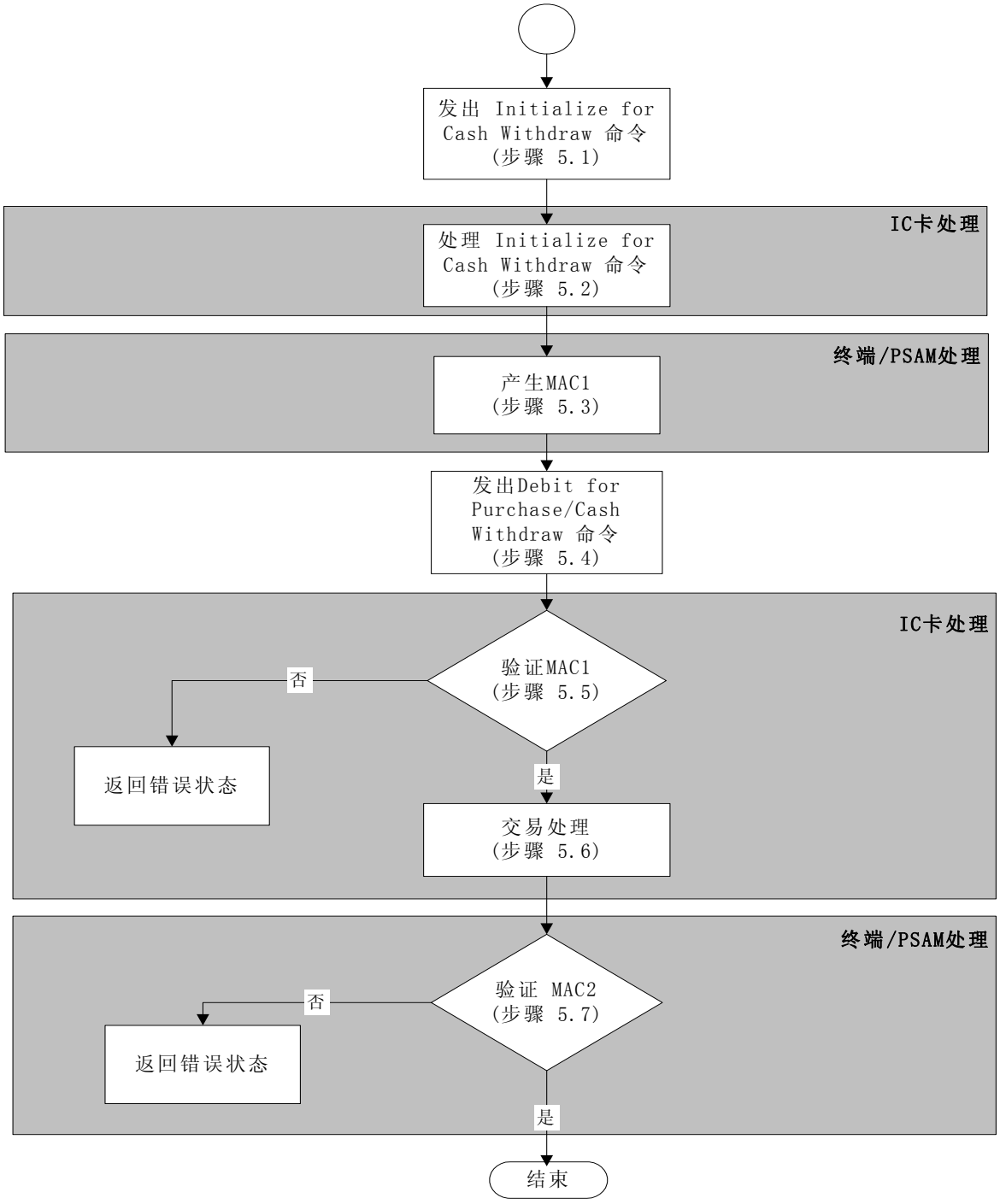


图 11-5 取现交易处理流程

11.1.5.2 处理 INITIALIZE FOR CASH WITHDRAW(步骤 5.2)

收到 INITIALIZE FOR CASH WITHDRAW 命令后，IC 卡将进行以下操作：

——检查是否支持命令中提供的密钥索引号。如果不支持，则回送状态码‘9403’

（不支持的密钥索引），但不回送其他数据。

——检查电子存折余额是否大于或等于交易金额。如果小于交易金额，状态码‘9401’（资金不足）返回给终端，不返回其他数据。终端采取的措施不在本手册的范围内。

对以上错误状态终端的处理不在本手册的范围内。

通过以上检查之后，IC 卡将产生一个伪随机数（ICC）和一个过程密钥 SESPK。该过程密钥是利用 DPK 并按安全管理一章描述的机制产生的。用于产生过程密钥的输入数据如下：

SESPK：伪随机数（ICC）||电子存折脱机交易序号||终端交易序号的最右两个字节。

11.1.5.3 验证 MAC1(步骤 5.3)

验证了交易金额有效之后，终端使用伪随机数（ICC）和 IC 卡回送的电子存折的脱机交易序号来产生相同的过程密钥（SESPK）和一个报文签别码（MAC1），供 IC 卡来验证 PSAM 的合法性。

MAC1 的计算机制见附录 B。用 SESPK 对以下数据加密产生 MAC1(按所列顺序):

——交易金额
——交易类型标识
——终端机编号
——交易日期（终端）
——交易时间（终端）

11.1.5.4 发出 DEBIT FOR PURCHASE/CASH WITHDRAW 命令(步骤 5.4)

终端发出 DEBIT FOR PURCHASE/CASH WITHDRAW 命令。

11.1.5.5 验证 MAC1(步骤 5.5)

在收到 DEBIT FOR PURCHASE/CASH WITHDRAW 命令后，IC 卡将验证 MAC1 的有效性。如果 MAC1 有效，交易处理会继续执行 11.1.5.6 中所描述的步骤。否则将向终端回送错误状态码‘9302’（MAC 无效）。终端对错误状态的处理不在本手册范围以内。

11.1.5.6 交易处理(步骤 5.6)

IC 卡从卡上的电子存折余额中扣减取现交易金额，将电子存折脱机交易序号加 1。IC 卡必须成功地完成以上所有步骤或者一个也不完成，如果余额或序号的更新没有成功，交易明细也不应被更新。

IC 卡产生一个报文签别码 (MAC2)供 PSAM 对 IC 卡合法性进行检查。IC 卡通过 DEBIT FOR PURCHASE/CASH WITHDRAW 命令响应报文将以下数据送给

PSAM (通过终端) ，作为产生 MAC2 的输入数据。用 SESPK 对以下数据加密产生 MAC2：

——交易金额

IC 卡执照安全管理一章中描述的机制直接用 DTK 产生 TAC。TAC 将被写入终端交易明细，以便于主机进行验证。下面是用来产生 TAC 的数据，它们以明文形式包含在 CREDTE FOR PURCHASE/CASH WITHDRAW 命令的响应报文中从 IC 卡传送到终端：

——交易金额
——交易类型标识
——终端机编号
——终端交易序号
——交易日期 (终端)
——交易时间 (终端)

IC 卡将用以下数据组成的一个记录更新 IC 卡交易明细。

——电子存折脱机交易序号
——交易金额
——交易类型标识
——终端机编号
——交易日期 (终端)
——交易时间 (终端)

11.1.5.7 验证 MAC2(步骤 5.7)

在收到从 IC 卡(经过终端)传来的 MAC2 后, PSAM 将验证 MAC2 的有效性。MAC2 验证的结果被传送到终端以便采取必要的措施。终端的处理不在本手册的范围之内。

11.1.6 修改透支限额交易

“透支功能”是从技术上支持的一种基于电子存折应用的有限信用功能。当电电子存折中的实际金额不足时，它为持卡人提供了一种在发卡方所允许的透支额度内继续进行交易的方便性。修改透支限额交易必须在金融终端上联机进行，且必须验证个人密码（PIN）。

是否使用“透支功能”以及允许透支的额度由发卡方决定。修改透支限额交易的具体业务做法和要求不在本手册的范围之内

如果透支限额存在，电子存折的余额是实际圈存余额与透支限额之和。

11.1.6.1 发出 INITIALIZE FOR UPDATE 命令(步骤 6.1)

终端发出 INITIALIZE FOR UPDATE 命令启动后修改透支限额交易。

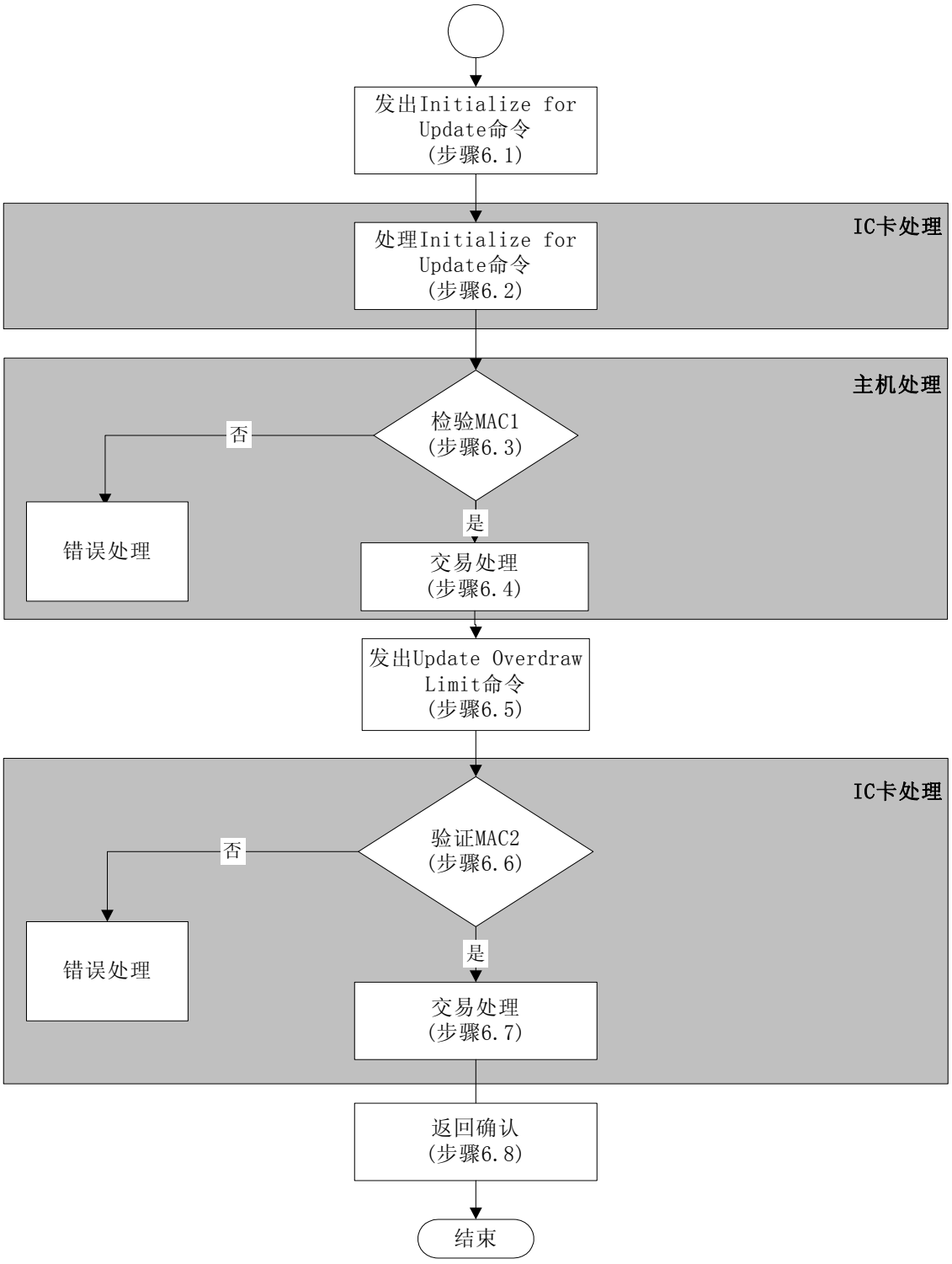


图 11-6 修改透支限额交易

11.1.6.2 处理 INITIALIZE FOR UPDATE 命令(步骤 6.2)

收到 INITIALZE FOR UPDATE 命令后，IC 卡将进行以下操作：

——检查是否支持命令中提供的密钥索引号。如果不支持，则回送状态码‘9403’（不支持的密钥索引），但不回送其它数据。

终端对以上错误所做的处理不在本手册的范围内

在通过了以上检查之后，IC 卡将产生一个伪随机数（ICC）、一个过程密钥（SESUK）和一个报文鉴别码（MAC1）。该过程密钥是利用 DUK 并按安全管理一章中描述的机制产生的。用于产生过程密钥的输入数据如下：

SESUK：伪随机数（ICC）||电子存折联机交易序号||‘8000’

MAC1 的计算机制见安全管理一章。用 SESULK 对以下数据加密产生 MAC1（按所列顺序）：

—电子存折余额（交易前）

—透支限额（交易前）

—交易类型标识

—终端机编号

11.1.6.3 验证 MAC1(步骤 6.3)

在收到 INITIALIZE FOR UPDATE 命令执行成功的响应报文后，终端应向主机传送 INITIALIZE FOR UPDATE 命令响应报文中定义的数据以及其他主机需要的数据以便于 MAC1 的验证。

利用终端传来的报文，主机将产生与 IC 卡相同的过程密钥（SESUK）来验证 MAC1。

如果 MAC1 有效，交易处理将执行 11.1.6.4 中所描述的步骤，否则，主机应向终端送错误状态码。终端针对错误状态所做的处理不在本手册的范围内。

11.1.6.4 主机处理(步骤 6.4)

假定主机已经知道 IC 卡的透支限额。

基于 MAC1（或者其他由主机决定的验证标准）验证的结果，主机将决定是否允许修改透支限额。

如果主机拒绝交易，则应向终端发送一个拒绝报文，结束交易处理。

如果主机允许交易，则应生成一个报文鉴别码（MAC2），以供 IC 卡对主机合法性进行检查。

MAC2 的计算机制见安全管理一章。用 SESULK 对以下数据进行加密（按所列顺序）产生 MAC2：

—透支限额（交易后）

—交易类型标识

—终端机编号

—交易日期（主机）

—交易时间（主机）

主机将电子存折联机交易序号加 1。

主机向终端发送一个至少包括新透支限额、交易日期（主机）、交易时间（主机）和 MAC2 的许可信息。

11.1.6.5 发出 UPDATE OVERDRAW LIMIT 命令(步骤 6.5)

如果主机同意交易，终端将发出 UPDATE OVERDRAW LIMIT 命令。

11.1.6.6 验证 MAC2(步骤 6.6)

IC 卡必须认证 MAC2 的有效性。如果 MAC2 有效，交易处理将执行 11.1.6.7 中所描

述的步骤。否则向终端回送状态码‘9302’（MAC2）无效。终端对此错误状态所做的处理不在本手册的范围内。

11.1.6.7 交易处理(步骤 6.7)

IC 卡将按照安全管理一章所描述的机制，直接用密钥 DTK 对以下数据加密产生一个 TAC：

——电子存折余额（交易后）
——电子存折联机交易序号（加 1 前）
——电子存折透支限额（交易后）
——交易类型标识
——终端机编号
——交易日期（主机）
——交易时间（主机）

将当前电子存折余额置为新的电子存折余额，更新透支限额并使电子存折联机交易序号加 1。这三个修改必须全部完成，或一个也不完成。

——电子存折联机交易序号
——透支限额
——交易类型标识
——终端机编号
——交易日期（主机）
——交易时间（主机）

IC 卡通过响应报文将 TAC 和状态码‘9000’传送给终端。

IC 卡将用以下数据组成的一个记录更新 IC 卡交易明细。

——电子存折脱机交易序号
——透支限额
——交易类型标识
——终端机编号
——交易日期 (终端)
——交易时间 (终端)

11.1.6.8 回送确认(步骤 6.8)

IC 卡在 UPDATE OVERDRAW LIMIT 命令中的响应报文中回送 TAC 和一个完成码，表明透支限额已经被成功更新。

11.1.7 查询余额交易

持卡人可以通过终端或其他读卡设备读取电子存折/钱包中的余额。此交易一般脱机进行。在电子存折应用中进行此交易必须提交个人密码（PIN）。电子钱包则不需要。

终端通过 GET BLANCE 命令来实现查询余额交易。

11.1.8 查询明细交易

持卡人可以通过终端或其他读卡设备读取电子存折中的交易明细记录。此交易一般采用脱机方式处理。交易时需提交个人密码（PIN）。

终端发出一个 READ RECORD 命令来获得交易明细。这个命令会回送某个交易明细记录中所含的所有数据。交易明细文件为循环记录文件，且至少包含 10 条记录。

交易明细中的记录用记录号寻址。记录号范围从 1 到 n，n 是文件中记录的最大个数。最近写入的记录号为 1，前一记录号为 2，如此类推直到 n。n 代表文件中最早写入的记录。

11.1.9 应用维护功能

以下交易必须在有相应密钥的设备上执行。

11.1.9.1 安全报文

电子存折/电子钱包应用涉及到的安全机制，请参考本手册安全管理一章，并作如下改动和增补：

——在传送一个包含安全报文的命令前，主机向终端发送一个报文，要求从 IC 卡获得一个随机数。终端向 IC 卡发出一个 GET CHALLENGE 命令。从 IC 卡回送的随机数被送往主机以用于安全报文处理。

——从 IC 卡回送的 4 字节随机数后缀以‘00 00 00 00’，所得到的结果作为初始值。

——不采用过程密钥。除去 UNBLOCK PIN 命令外，均用导出的应用维护密钥（DAMK）来计算 MAC。UNBLOCE PIN 命令采用导出的 PIN 解锁密钥来产生 MAC。

——全部采用双字节密钥的 3DEA 算法。

11.1.9.2 卡片锁定

终端发出 CARD BLOCK 命令来锁定卡片。

此命令参照手册“命令”部分。其安全机制在 11.1.9.1 中描述。命令的成功执行使得 IC 卡中的所有应用无效。在这种情况下，进行应用选择将会回送状态码“6A81”（功能不被支持）。

11.1.9.3 应用锁定

终端发出 APPLICATION BLOCK 命令来锁定应用。

此命令的用法由发卡方自行决定。

此命令参照手册“命令”部分。其安全机制在 11.1.9.1 中描述。在本手册所述的应用中，命令的成功执行导致 IC 卡中的电子存折/电子钱包应用无效。在这种状态下：

——选择此应用时，对 SELECT 命令 IC 卡回送文件控制信息（FCI）和状态码‘6A81’（功能不被支持）。

——在应用被选择后，除以下情况外，IC 卡对其它命令只回送状态码‘6985’（使用的条件不满足）：

a） 当用 SELECT 命令选择此应用或其他应用时；

b） 当用 GET CHALLANGE 命令为 UNBOLCK PIN 命令产生 MAC 时；

c） APPLICATION BLOCK 命令；

d） CARD BOLCK 命令；

如果在命令参数 P2 中指明永久性锁定此应用，IC 卡将设置一个内部标志以表明不允许执行 APPLICATION UNBLOCK 命令。

此命令的执行并不改变电子存折联机交易序号和电子钱包联机交易序号的值。

11.1.9.4 应用解锁

终端发出 APPLICATION UNBLOCK 命令来对应用解锁，此命令参照手册“命令”部分，其安全机制在 11.1.9.1 中描述。

如果对某应用连续三次解锁失败，则 IC 卡将永久锁定此应用并回送状态码‘9303’（应

用永久锁定）。

如果在 APPLICATION UNBLOCK 命令中使用了永久锁定的选项，IC 卡将回送状态码‘6983’（认证方式锁定）且不再对应用解锁。

APPLICATION UNBLOCK 命令的成功执行，使应用重新恢复成有效状态。在此之后，该应用对所有命令的响应就像应用锁定和应用解锁没有执行过一样。

此命令的执行并不改变电子存折联机交易序号和电子钱包联机交易序号的值。

11.1.9.5 PIN 解锁

终端发出 UNBLOCK PIN 命令对 PIN 解锁，此命令参照手册“命令”部分，其安全机制在 11.1.9.1 中描述。

在命令报文中，P2 取‘00’值。使用 DPUK 对 PIN 数据加密（参考手册“安全管理”部分）。

如果 PIN 连续三次解锁失败，则 IC 卡将永久锁定此应用并回送状态码‘9303’（应用永久锁定）。

11.1.9.6 二进制形式修改

终端文件的安全要求，发出 UPDATE BINARY 指令。

如果三次执行此命令均告失败，则 IC 卡将永久锁定此应用并回送状态码‘9303’（应用永久锁定）。

11.1.9.7 更改 PIN

这个功能不需要 MAC，它可以在任意支持该命令的终端上执行。

当 IC 卡接到此命令时，它将：

——检查 PIN 尝试计数器。如果为 0，PIN 已锁定，此命令不能执行。在这种情况下，IC 卡回送状态码‘6983’（认证方式锁定）。

——如果 PIN 没有锁定，则命令中的‘当前 PIN’会和 IC 卡上存放的 PIN 比较。如果二者相同，IC 卡将进行以下操作：

a）将 IC 卡上的 PIN 改为命令中的新 PIN；

b）将 PIN 尝试计数器置为 PIN 重试的最大次数。

——如果卡上的 PIN 和命令中的‘当前 PIN’并不相同，IC 卡将进行以下操作：

a）将 PIN 尝试计数器减 1；

b）回送状态码‘63Cx’，这里 x 是 PIN 尝试计数器的新值。如达到零， 卡片自动锁 PIN。

11.1.9.8 重装 PIN

终端发出 RELOAD PIN 命令来重装 PIN。

按照安全管理一章中的机制用密钥 DRPK 来产生一个 MAC。

当这个命令失败三次，应用被永久锁定。

11.1.10 外部认证

EXTERNAL AUTHENTICATE 命令的目的是 IC 卡验证外部接口设备的有效性，使接口设备对 IC 卡获得某种操作授权。

接口设备提供的认证数据应按以下规则产生：

1、Lc = ‘08’

2、用 GET CHALLENGE 命令向 IC 卡申请一组随机数，作为认证数据输入因子（8 字节）。

3、用指定密钥对随机数作加密计算，产生认证数据（8 字节），参见“安全计算”一节。

IC 卡外部认证过程：

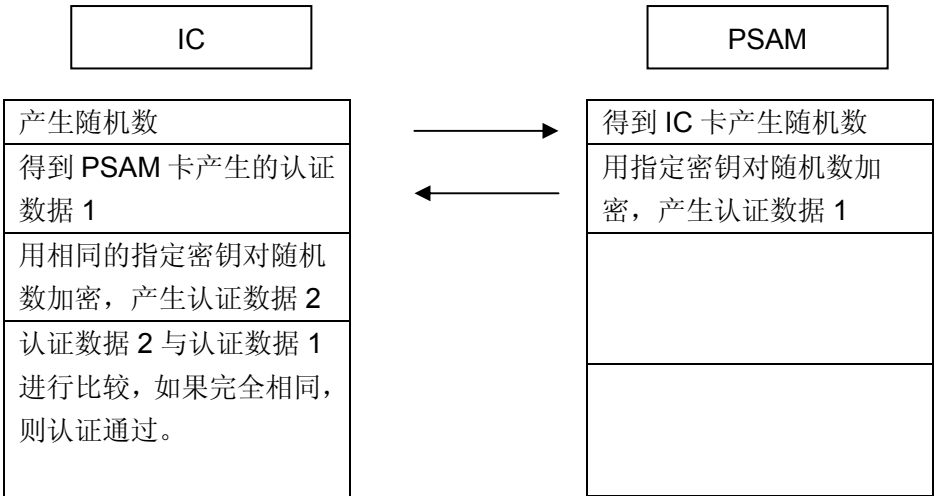


图 11-7 金融外部认证流程图

11.2 扩展金融应用交易流程

11.2.1 交易预处理

11.2.1.1 标准的交易预处理（步骤 1.1）

引用金融应用交易流程 11.1.1 节。

补充定义如下：

“选择 IC 卡”部分：对于非接触式 IC 卡，终端应具有检测 IC 卡是否进入射频有效场强范围内的能力。一旦终端检测到有效 IC 卡进入，终端应具备分辨多张有效 IC 卡进入的情况，并依次逐卡自动选择或人工选择一张特定 IC 卡。当卡片选定后，终端将继续专项应用选择功能。终端应根据支持的应用自动选择卡片上的应用。

“提示输入个人密码（PIN）”和“校验 PIN”部分：本规范定义的应用中的消费交易、复合应用消费交易、灰锁消费交易的交易预处理无需此部分。

“交易类型选择”部分：对电子钱包应用来说，持卡人应能选择如下交易类型：圈存、 消费、查询余额、复合应用消费交易和灰锁消费交易。

以下补充交易预处理步骤为可选项。

11.2.1.2 发出 GET LOCK PROOF（P1＝’00’）命令（步骤 1.2）

终端发出 GET LOCK PROOF（P1＝’00’）命令对电子钱包的状态进行查询。

11.2.1.3 判断 TACUF（交易验证码待读标志）（步骤 1.3）

IC 卡收到 GET LOCK PROOF（P1＝’00’）命令后，首先根据 TACUF 判断上次的解扣交易的 TAC 码是否未被终端正确读取。如果 TACUF 为 1，即上次的解扣交易的 TAC 码有待读取，则进入 11.2.1.4 节所描述的步骤：否则，进入 11.2.1.5 节所描述的步骤进行灰锁标志的判断。

如果应用未发生过灰锁、解扣、联机解扣交易，则 IC 卡返回’6985’出错信息给终端，但不回送其他数据，同时结束交易预处理流程。

11.2.1.4 返回 TAC 码（步骤 1.4）

IC 卡将上次的解扣交易的产生的未成功读取的 TAC 码返回给终端，以供终端形成补充交易数据包，进入 11.2.1.9 节所描述的步骤。详细的响应报文参见 9.2.4.10 节中的表 10-4。

11.2.1.5 判断灰锁标志（步骤 1.5）

IC 卡对所选择的电子钱包应用进行灰锁判断，如果当前应用中的电子钱包应用无灰锁，则进入 11.2.1.6 节所描述的步骤，返回正常信息给终端。否则进入 11.2.1.7 节所描述的步骤，返回灰锁信息给终端。

11.2.1.6 返回正常信息（步骤 1.6）

IC 卡将正常信息返回给终端，详细的响应报文参见 9.2.4.10 节中的表 10-2。交易预处理流程完成。

11.2.1.7　返回灰锁信息（步骤 1.7）

IC 卡将上次的灰锁交易的产生日期、时间、MAC2、GTAC 等返回给终端，详细的响应报文参见 9.2.4.10 节中的表 10-3。终端进入 11.2.1.8 节启动补扣交易流程。

11.2.1.8　进行补扣交易（步骤 1.8）

补扣交易的详细描述参见 11.2.9，完成后交易预处理流程结束。

11.2.1.9　进行补充交易（步骤 1.9）

补充交易的详细描述参见 11.2.10，完成后交易预处理流程结束。

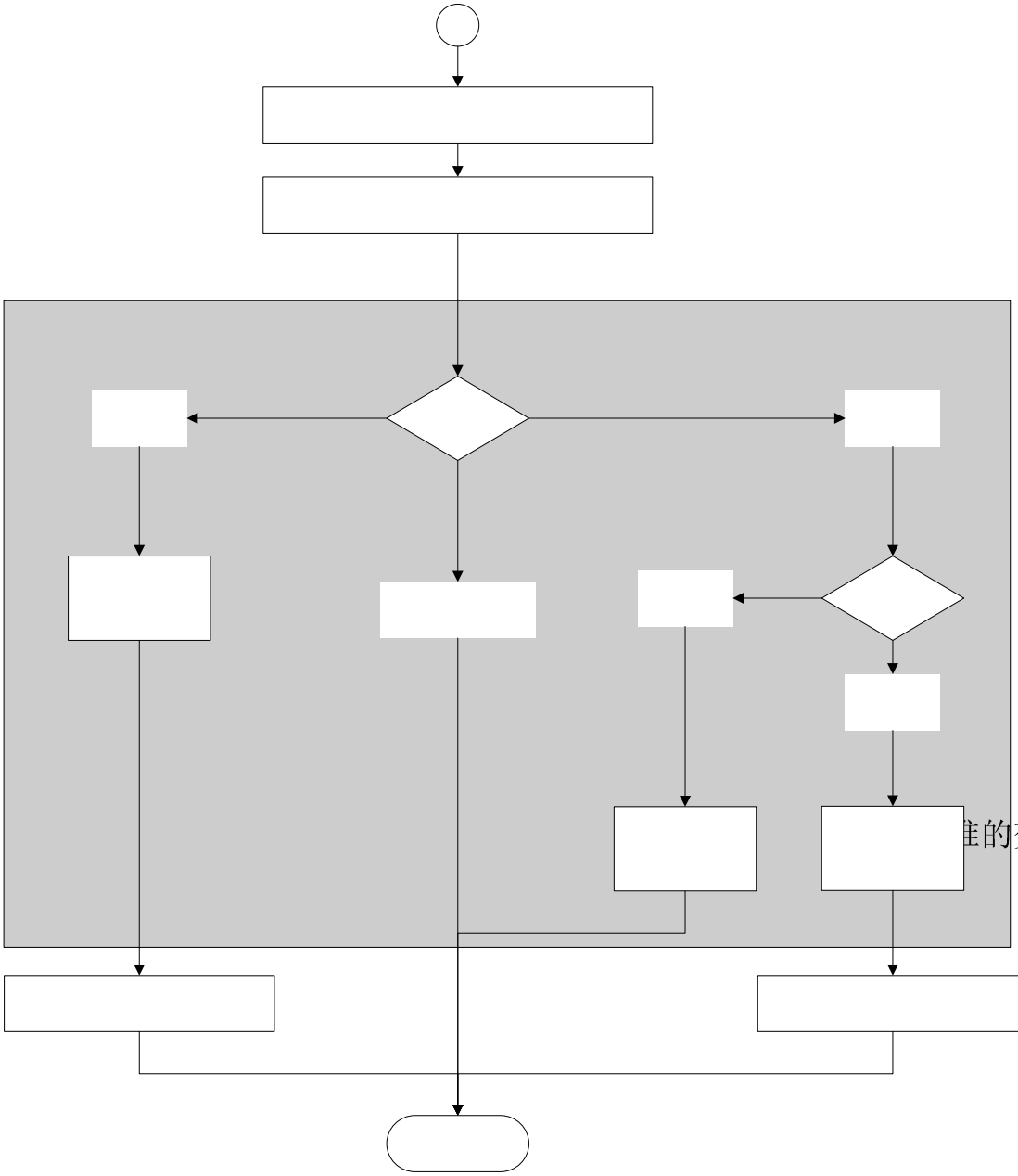


图 11-8 加入灰锁机制的交易预处理流程

11.2.2 圈存交易

引用金融应用交易流程 11.1.2 节。

补充定义如下：

“交易处理”部分：交易明细定义为：

———电子钱包交易序号

———交易金额

———交易类型标识

———终端机编号

———交易日期

———交易时间

“处理 INITIALIZE FOR LOAD“部分：增加一检查过程：

———检查钱包是否被灰锁。如果灰锁，则回送状态码’9408’（钱包灰锁锁定），但不回送其它信息，同时终止命令的处理过程。

11.2.3 消费交易

引用“金融应用交易流程”一节的 11.1.4 节。

补充定义如下：

“交易处理”：IC 卡从电子钱包余额中扣减消费的金额，电子钱包交易序号加 1，更新电子钱包消费交易记录。IC 卡必须成功地完成以上所有步骤或者一个也不完成。

对于电子钱包消费交易，IC 卡将用以下数据组成的一个记录更新交易明细。

———交易金额

———交易类型标识 ‘06’

———电子钱包脱机交易序号

———终端机编号

———交易日期（终端）

———交易时间（终端）

“处理 INITIALIZE FOR PURCHASE“部分：增加一检查过程：

———检查钱包是否被灰锁。如果灰锁，则回送状态码’9408’（钱包灰锁锁定），但不回送其它信息，同时终止命令的处理过程。

11.2.4 复合应用消费交易

复合应用消费交易允许持卡人使用电子钱包的余额进行购物或获取服务。此交易可以在终端设备或其它读卡设备上脱机进行。此交易无需提交个人密码（PIN）。

复合应用消费交易允许消费金额为 0。

11.2.4.1 发出 INITIALIZE FOR CAPP PURCHASE 命令（步骤 4.1）

终端发出 INITIALIZE FOR CAPP PURCHASE 命令启动复合应用消费交易。

11.2.4.2 处理 INITIALIZE FOR CAPP PURCHASE 命令（步骤 4.2）

IC 卡收到 INITIALIZE FOR CAPP PURCHASE 命令后，将进行以下操作：

检查是否支持命令中提供的密钥索引号。如果不支持，则回送状态码'9403'（不支持的密钥索引），但不回送其他数据。

检查钱包是否被灰锁，如果灰锁，则回送状态码'9408'（钱包灰锁锁定），但不回送其它信息，同时终止命令的处理过程。

检查电子钱包余额是否大于或等于交易金额。如果小于交易金额，则回送状态码'9401'，但不回送其它数据。终端应采取的措施不在本规范的范围内。

在通过以上检查之后，IC 卡将产生一个伪随机数（ICC）和过程密钥。用于产生该过程密钥的输入数据如下：

SESPK：伪随机数（ICC）||电子钱包交易序号||终端交易序号的最右两个字节

11.2.4.3 产生 MAC1（步骤 4.3）

使用伪随机数（ICC）和 IC 卡回送的电子钱包交易序号，终端的安全存取模块（PSAM）将产生一个过程密钥（SESPK）和一个报文认证码（MAC1），供 IC 卡来验证 PSAM 的合法性。

用 SESPK 对以下数据进行加密产生 MAC1（按所列顺序）：

———交易金额
———交易类型标识
———终端机编号
———交易日期（终端）
———交易时间（终端）

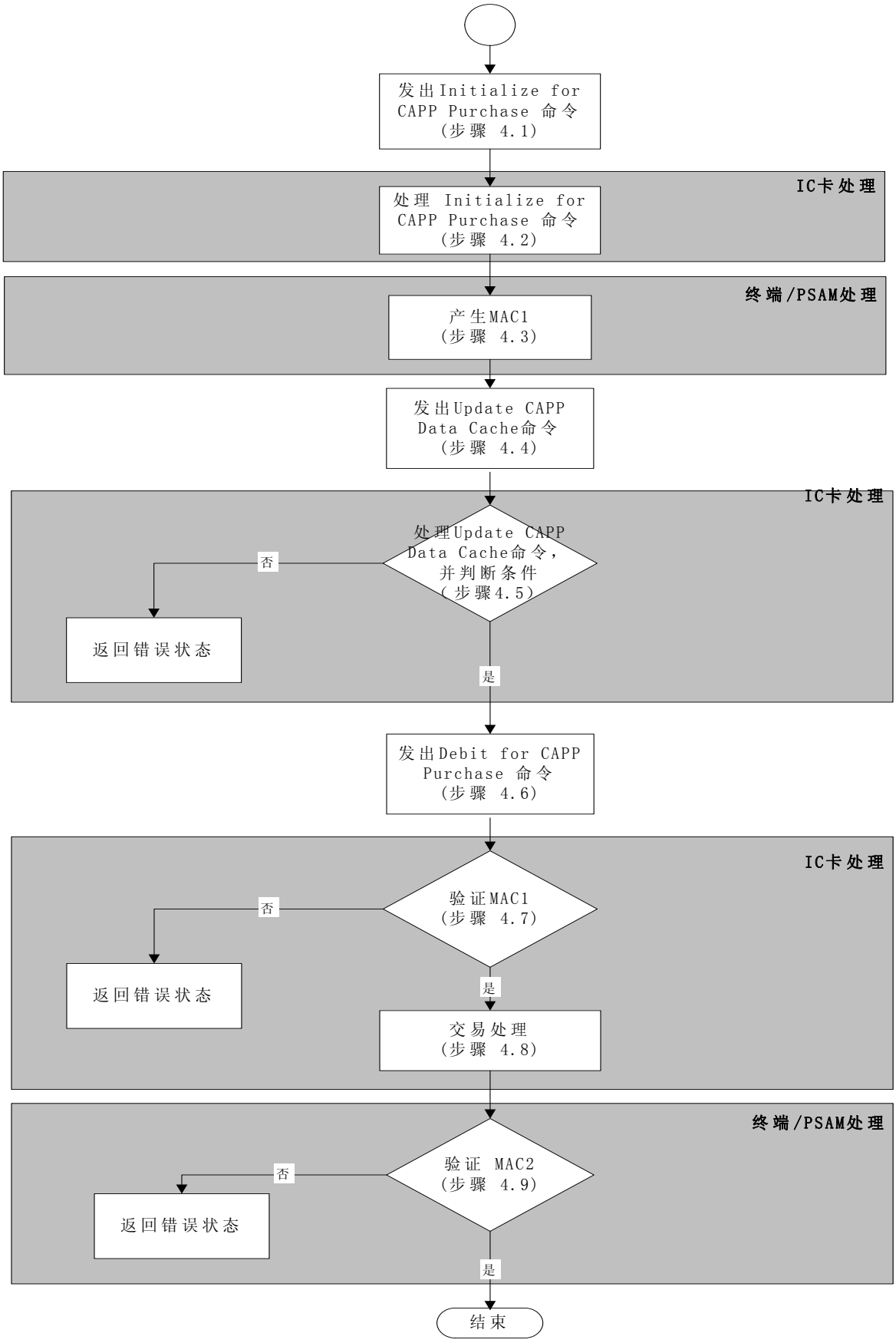


图 11-9 复合消费交易流程

11.2.4.4 发出 UPDATE CAPP DATA CACHE 命令（步骤 4.4）

终端发出 UPDATE CAPP DATA CACHE 命令。

11.2.4.5 处理 UPDATE CAPP DATA CACHE 命令（步骤 4.5）

IC 卡在收到 UPDATE CAPP DATA CACHE 命令后，将进行以下操作：

如果命令中存在 SFI 域，检查卡片当前应用下是否存在与命令中 SFI 值相同的文件。如果不存在，回送状态码'6A82'（未找到文件），但不回送其它数据。终端应终止此次复合应用消费交易。

根据命令中的复合应用类型标识符，查询复合应用专用文件中是否存在相同标识符的记录。如果不存在，则回送状态码'6A83'（未找到记录），但不回送其它数据。终端应终止此次复合应用消费交易。

检查复合应用专用文件中相应记录中的应用锁定标志字节。如果应用锁定标志为设置，则回送状态码'9407'（复合应用禁止），但不回送其它数据。终端应终止此次复合应用消费交易。

检查命令中的数据域长度是否大于复合应用专用文件中相应记录的长度。如果大于，则回送状态码'6A84'（文件中存储空间不够），但不回送其它数据。终端应终止此次复合应用消费交易。

在通过以上检查后，IC 卡应暂存命令中的 SFI、记录号、复合应用类型标识符和数据域。复合应用专用文件中相应记录中的数据不得通过此命令更新。

11.2.4.6 发出 DEBIT FOR CAPP PURCHASE 命令（步骤 4.6）

终端发出 DEBIT FOR CAPP PURCHASE 命令。

11.2.4.7 验证 MAC1（步骤 4.7）

在收到 DEBIT FOR CAPP PURCHASE 命令后，IC 卡将验证 MAC1 的有效性。如果 MAC1 有效，交易处理将继续执行 11.2.4.8 中所描述的步骤。否则将向终端回送错误状态码'9302'（MAC 无效）。

11.2.4.8 交易处理（步骤 4.8）

IC 卡从电子钱包余额中扣减消费的金额，电子钱包交易序号加 1，根据 11.2.4.8 中暂存的数据更新复合应用专用文件，更新电子钱包消费交易记录。IC 卡必须成功地完成以上所有步骤或者一个也不完成。

在根据 11.2.4.5 中暂存的数据更新复合应用专用文件时，如果更新数据长度小于记录长度，IC 卡应在数据后自动填充'00'至记录尾。

IC 卡产生一个报文签别码 （MAC2）供 PSAM 对其进行合法性检查，并通过 DEBIT FOR CAPP PURCHASE 命令响应报文回送以下数据，作为 PSAM 产生 MAC2 的输入数据。用 SESPK 对以下数据进行加密产生 MAC2：

———交易金额

TAC 将被写入终端交易明细，以便于主机进行交易验证。下面是用来生成 TAC 的数据，它们以明文形式通过 DEBIT FOR CAPP PURCHASE 命令的响应报文从 IC 卡传送到终端：

———交易金额
———交易类型标识
———终端机编号
———终端交易序号
———交易日期 （终端）
———交易时间 （终端）
对于电子钱包消费交易，IC 卡将用以下数据组成的一个记录更新交易明细。
———交易金额
———交易类型标识’09’
———电子钱包脱机交易序号
———终端机编号
———交易日期（终端）
———交易时间（终端）

11.2.4.9 验证 MAC2 （步骤 4.9）

在收到 IC 卡（经过终端）传来的 MAC2 后， PSAM 要验证 MAC2 的有效性。MAC2 验证的结果被传送到终端以便采取必要的措施。

11.2.5 查询余额交易

持卡人可以通过终端或其他读卡设备读取电子钱包中的余额。此交易一般脱机进行。此交易无需提交个人密码（PIN）。

终端利用 GET BALANCE 命令实现查询余额交易。

11.2.6 查询明细交易

持卡人可以通过终端或其他读卡设备读取卡片中的交易明细记录。此交易一般采用脱机方式处理。交易时无需提交个人密码（PIN）。

终端发出一个 READ RECORD 命令来获得交易明细。这个命令会回送某个交易明细记录中所含的所有数据。交易明细文件为循环记录文件。

11.2.7 灰锁消费交易

灰锁消费交易允许持卡人使用电子钱包进行灰锁消费。此交易可以脱机进行。

11.2.7.1 发出 INITIALIZE FOR GREY LOCK 命令（步骤 7.1）

终端发出 INITIALIZE FOR GREY LOCK 命令启动灰锁消费交易。

11.2.7.2 处理 INITIALIZE FOR GREY LOCK 命令（步骤 7.2）

IC 卡收到 INITIALIZE FOR GREY LOCK 命令后，将进行以下操作：

———检查命令中包含的密钥索引号是否被 IC 卡支持。如果不支持，返回状态码'9403'（不支持的密钥索引号）且不返回其他数据。

在通过以上检查之后，IC 卡将产生一个伪随机数，这个伪随机数将包含在本命令的响应报文中返回终端。

之后，IC 卡将内部的 TACUF 复位。

11.2.7.3 计算 MAC1（步骤 7.3）

使用 IC 卡返回的伪随机数和电子钱包脱机交易序号，终端的安全存取模块（PSAM）将产生一个终端随机数（TRAN），一个过程密钥（GSESPK）和一个报文认证码（MAC1），供 IC 卡来验证 PSAM 的合法性。

过程密钥 GSESPK 被用于电子钱包的灰锁消费交易。

过程密钥的产生分两步：即先是用 DPK 密钥产生的中间密钥，再用中间密钥采用下述的算法产生过程密钥。

用来产生中间密钥的输入数据如下：

TMPCK：伪随机数（ICC） || 电子钱包脱机交易序号 || 终端交易序号的最右两个字节。

用中间密钥对终端随机数（TRAN）加密，运算的结果产生过程密钥：

GSESPK = DES（TMPCK，TRAN || '80000000'）

用 GSESPK 对以下数据进行加密产生 MAC1（按所列顺序）：

———交易类型标识

———终端机编号

———交易日期

———交易时间

11.2.7.4 发出 GREY LOCK 命令（步骤 7.4）

终端发出 GREY LOCK 命令。

11.2.7.5 验证 MAC1（步骤 7.5）

IC 卡收到 GREY LOCK 命令后，将产生同样的过程密钥（GSESPK）并验证 MAC1 是否有效。如果 MAC1 是有效的，交易处理将继续执行 11.2.7.6 节。如果 MAC1 是无效的，IC 卡返回错误状态码'9302'（MAC 无效）给终端。

11.2.7.6　灰锁处理（步骤 7.6）

IC 卡将电子钱包脱机交易序号加 1，并将电子钱包应用灰锁。

IC 卡产生一个报文鉴别码(MAC2)供 PSAM 对 IC 卡合法性进行检查,并同时将 MAC2 写入内部文件。MAC2 将包含在从卡传送到 PSAM（通过终端）GREY LOCK 的命令响应报文和 GET LOCK PROOF 的命令响应报文中。

MAC2 的计算机制见 JR/T 0025.2《电子电子钱包/电子存折应用规范》附录 B。用 GSESPK 对以下这些数据进行加密产生 MAC2：

———电子钱包余额

———电子钱包脱机交易序号（加 1 前）

IC 卡也应该直接用密钥 DTK 产生一个 GTAC。GTAC 将包含在从卡传送到 PSAM（通过终端）的 GREY LOCK 的命令响应报文和 GET LOCK PROOF 的命令响应报文中。如果之后出现交易异常中断等，使 DEBIT FOR UNLOCK 指令无法当时执行成功，GTAC 可供终端纳入终端异常交易数据中，以便后来上传给主机进行灰锁验证。

下面是用来生成 GTAC 的要素：

———-交易类型标识

———终端机编号

———终端交易序号

———交易日期（终端）

———交易时间（终端）

IC 卡应把 GSESPK 存贮到安全的内部文件中,（或者，IC 卡也可以将终端随机数、伪随机数（ICC)、终端交易序号等，写入内部文件，通过计算重新获得)，以备交易中途 IC 卡掉电后，在后续交易流程中恢复过程密钥 GSESPK。

IC 卡将用以下数据组成的一个记录来更新内部专用明细。这个明细记录中的数据将包含在 GET LOCK PROOF 的命令响应报文中，由 IC 卡返回给终端。

———交易类型标识（’91’＝电子钱包灰锁）

———电子钱包代号（’01’＝电子钱包）

———电子钱包余额

———电子钱包脱机交易序号

———终端机编号

———交易日期

———交易时间

———MAC2

———GTAC

IC 卡必须全部成功地完成以上几个步骤或者一个也不完成,如果脱机交易序号的更新、电子钱包应用灰锁状态的设置没有成功，交易明细也不应更新。

11.2.7.7　验证 MAC2（步骤 7.7）

在收到 IC 卡（经终端）传来的 MAC2 后，PSAM 要验证 MAC2 的有效性。MAC2 如果有效，交易继续进行 11.2.7.8 节所描述的步骤；如果 MAC2 是无效的，终端应停止交易并采取相应的措施。

11.2.7.8　持卡人进行消费行为（步骤 7.8）

持卡人进行消费行为。在进行消费过程中，允许终端对 IC 卡下电。若下电以后，IC 卡重新上电，经过交易预处理（选择应用、验证个人密码等）后应可以继续执行 11.2..7.9 节所描述的步骤而不受影响。

11.2.7.9　产生 GMAC（步骤 7.9）

安全存取模块（PSAM）根据专用消费的金额，用过程密钥（GSESPK）产生一个报文认证码（GMAC），供 IC 卡来验证 PSAM 的合法性。

用 GSESPK 对以下数据进行加密产生 GMAC：

———交易金额

11.2.7.10　发出 DEBIT FOR UNLOCK 命令（步骤 7.10）

终端发出 DEBIT FOR UNLOCK 命令。

11.2.7.11　检查脱机交易序号和余额（步骤 7.11）

收到 DEBIT FOR UNLOCK 命令后，IC 卡将进行以下操作：

———检查脱机交易序号是否匹配，如果脱机交易序号不匹配，IC 卡将返回’9406’（脱机交易序号错），但不回送其他数据。

———检查电子钱包余额是否大于或等于交易金额。如果小于交易金额，则回送状态码’9401’（金额不足），但不回送其他数据，IC 卡不操作内部出错计数器，终端应采取相应的措施。

通过上面的检查后，IC 卡进入 11.2.7.12 节。

11.2.7.12　验证 GMAC（步骤 7.12）

IC 卡验证 GMAC 的有效性。如果 GMAC 是有效的，将 IC 卡内部的解扣出错计数器复位，交易处理将继续执行 11.2.7.13 节。如果 GMAC 是无效的，IC 卡返回错误状态码’9302’（MAC 无效）给终端，同时操作解扣出错计数器，3 次出错则临时锁住应用以防止恶意试探。该解扣出错计数器将在应用解锁命令执行成功后被复位。

11.2.7.13　交易处理（步骤 7.13）

IC 卡从卡上的电子钱包余额中减去灰锁消费的交易金额（如果交易金额为 0，则省略对余额的修改）、将电子钱包解锁、并将卡内的 TACUF（交易验证码待读标志）置位。

IC 卡应该直接用密钥 DTK 产生一个 TAC。TAC 将被写入终端交易数据包，以便后来传给主机进行交易验证。

下面是用来生成 TAC 的要素（按所列顺序）：

———交易金额
———交易类型标识（’93’=电子钱包解扣）
———终端机编号（发出 DEBIT FOR UNLOCK 命令的终端）
———终端交易序号（发出 DEBIT FOR UNLOCK 命令的终端）

———交易日期（发出 DEBIT FOR UNLOCK 命令的日期）
———交易时间（发出 DEBIT FOR UNLOCK 命令的时间）

对于电子钱包的灰锁消费交易，IC 卡将用以下数据组成的一个记录更新标准交易明细。

———电子钱包脱机交易序号
———交易金额
———交易类型标识（'93'=电子钱包解扣）
———终端机编号（发出 DEBIT FOR UNLOCK 命令的终端）
———交易日期（发出 DEBIT FOR UNLOCK 命令的日期）
———交易时间（发出 DEBIT FOR UNLOCK 命令的时间）

对于电子钱包的灰锁消费交易，IC 卡将用以下数据组成的一个记录更新内部专用明细文件，以便以后终端可以通过 GET LOCK PROOF 命令得到：

———交易类型标识
———电子钱包代号（'01'=ET）
———电子钱包余额
———电子钱包脱机交易序号
———终端机编号（发出 DEBIT FOR UNLOCK 命令的终端）
———交易日期（发出 DEBIT FOR UNLOCK 命令的日期）
———交易时间（发出 DEBIT FOR UNLOCK 命令的时间）
———交易金额
———TAC

IC 卡必须全部成功地完成以上几个步骤或者一个也不完成，如果余额的更新、TACUF 的置位、电子钱包应用的灰锁状态的恢复没有成功，标准交易明细和内部专用明细也不应被更新。

11.2.7.14　回送确认（步骤 7.14）

IC 卡在 DEBIT FOR UNLOCK 命令的响应报文中回送 TAC 码和 SW1SW2＝'9000'，表明余额已被更新而且电子钱包应用已解锁。

11.2.7.15　读取 TAC 码〈步骤 7.15）

终端读取由 IC 卡发来的 TAC 码，合成完整的交易成交数据包。

11.2.7.16　发出 GET LOCK PROOF（P1＝'01'〕命令（步骤 7.16）

终端发出 GET LOCK PROOF（P1＝'01'）命令。

11.2.7.17　处理 GET LOCK PROOF（P1='01'）命令（步骤 7.17）

IC 卡将内部的 TACUF（交易验证码待读标志）复位。

11.2.8 联机解扣交易

联机解扣交易允许将 IC 卡上的电子钱包应用解锁并扣除相应的交易金额。本交易必须在联机的终端上进行。

11.2.8.1 发出 INITIALIZE FOR GREY UNLOCK 命令（步骤 8.1）

终端发出 INITIALIZE FOR GREY UNLOCK 命令启动联机解扣交易。

11.2.8.2 处理 INITIALIZE FOR GREY UNLOCK 命令〈步骤 8.2）

IC 卡收到 INITIALIZE FOR GREY UNLOCK 命令后，将进行以下操作：

———检查 IC 卡是否支持命令中包含的密钥索引号。如果不支持，返回状态码’9403’（不支持的密钥索引号），且不返回其他数据。

在通过以上检查之后，IC 卡将产生一个伪随机数（ICC）、过程密钥 SESULK 和一个报文鉴别码（MAC1），供主机来验证联机解扣交易和 IC 卡的合法性。

过程密钥 SESULK 被用于电子钱包的联机解扣交易。过程密钥是用 DULK 密钥产生的。

用来产生过程密钥的输入数据如下：

SESULK：伪随机数（ICC）|| 电子钱包联机交易序号 || ‘8000’

用 SESUIK 对以下数据加密产生 MAC1（按所列顺序）：
———电子钱包余额
———电子钱包脱机交易序号
———交易类型标识
———终端机编号

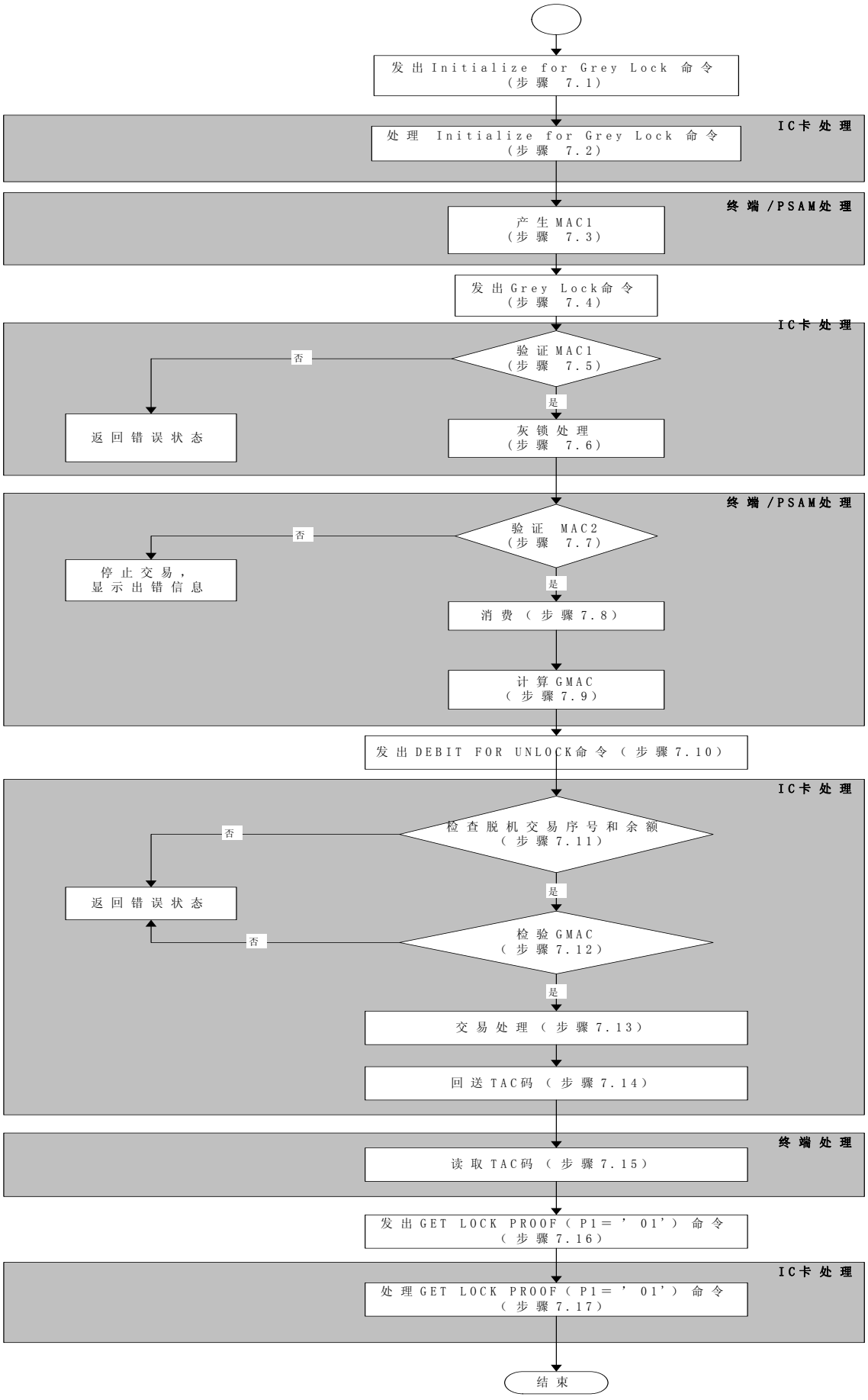

发出Initialize for Grey Lock 命令
（步骤 7.1）
IC卡处理
处理 Initialize for Grey Lock 命令
（步骤 7.2）
终端/PSAM处理
产生MAC1
（步骤 7.3）
发出Grey Lock命令
（步骤 7.4）
IC卡处理
验证MAC1
（步骤 7.5）
否
是
返回错误状态
灰锁处理
（步骤 7.6）
终端/PSAM处理
验证 MAC2
（步骤 7.7）
否
停止交易，
显示出错信息
是
消费（步骤7.8）
计算GMAC
（步骤7.9）
发出DEBIT FOR UNLOCK命令（步骤7.10）
IC卡处理
检查脱机交易序号和余额
（步骤7.11）
否
是
返回错误状态
检验GMAC
（步骤7.12）
否
是
交易处理（步骤7.13）
回送TAC码（步骤7.14）
终端处理
读取TAC码（步骤7.15）
发出GET LOCK PROOF（P1＝’01’）命令
（步骤7.16）
IC卡处理
处理GET LOCK PROOF（P1＝’01’）命令
（步骤7.17）
结束

图 11-10 灰锁消费交易流程

IC 卡将把 INITIALIZE FOR GREY UNLOCK 命令的响应报文送给终端处理。

如果 IC 卡返回的状态不是'9000'，终端将终止交易。

在收到 INITIALIZE FOR GREY UNLOCK 命令的响应报文后，终端将一个包含表 38 中数据的联机解扣许可请求数据包送往发卡方主机。

11.2.8.3 验证 MAC1（步骤 8.3）

主机将生成 SESULK 并且确认 MAC1 是否有效。如果 MAC1 有效，交易处理将按 11.2.8.5 节描述的步骤继续执行；否则主机返回一个错误状态码，交易处理将转至 11.2.8.4 节描述的步骤。

11.2.8.4 回送错误状态（步骤 8.4）

如果出现使联机解扣交易不能被接受的情况，则主机应通知终端。终端将采取相应的措施。

11.2.8.5 主机处理（步骤 8.5）

在确认能够进行联机解扣交易后，主机将产生一个报文鉴别码（MAC2），供 IC 卡对主机合法性进行检查。

用 SESULK 对以下数据进行加密产生 MAC2（按所列顺序）：

———应补扣的交易金额

———交易类型标识

———终端机编号

———交易日期（主机）

———交易时间（主机）

主机发送一个联机解扣交易接受报文给终端，其中至少包括 MAC2、交易日期（主机）和交易时间（主机）。

11.2.8.6 发出 GREY UNLOCK 命令（步骤 8.6）

终端收到主机的联机解扣交易响应报文后，向 IC 卡发出 GREY UNLOCK 命令，以更新卡上电子钱包余额、并将电子钱包应用解锁。

11.2.8.7 验证 MAC2（步骤 8.7）

收到 GREY UNLOCK 命令后，IC 卡先检查电子钱包余额是否大于或等于交易金额。如果小于交易金额，则回送状态码'9401'（金额不足），但不回送其他数据。IC 卡还要验证 MAC2 的有效性。如果 MAC2 是有效的，交易处理将继续执行 11.2.8.8 节所描述的步骤；否则 IC 卡返回错误状态码'9302'（MAC 无效）给终端。

11.2.8.8 交易处理（步骤 8.8）

IC 卡从卡上的电子钱包余额中减去交易金额（如果交易金额为 0，则省略对余额的修改），将电子钱包联机交易序号加 1，将内部的解扣出错计数器复位，并将电子钱包应用解锁。

IC 卡产生一个报文鉴别码（MAC3），包含在从卡传送到主机（通过终端）的 GREY UNLOCK 命令的响应报文中，以供主机对联机解扣交易的成功的合法性进行检查。

用 SESULK 对以下数据进行加密产生 MAC3（按所列的顺序）：

———电子钱包余额
———电子钱包联机交易序号（加 1 前）
———交易金额
———交易类型标识
———终端机编号
———交易日期（主机）
———交易时间（主机）

在对电子钱包的联机解扣交易中，IC 卡用以下数据组成的一个记录来更新标准交易明细：

———电子钱包联机交易序号
———交易金额
———交易类型标识（'95'=电子钱包联机解扣）
———终端机编号
———交易日期（主机）
———交易时间（主机）

对于电子钱包的联机解扣交易，IC 卡将用以下数据组成的一个记录更新内部专用明细文件，以便以后终端可以通过 GET LOCK PROOF 命令得到：

———交易类型标识
———电子钱包代号（'01'=电子钱包）
———电子钱包余额
———电子钱包联机交易序号
———终端机编号
———交易日期
———交易时间
———交易金额
———MAC3

IC 卡必须全部成功地完成以上几个步骤或者一个也不完成，如果上述的操作没有成功，标准交易明细和内部专用明细也不应更新。

11.2.8.9 验证 MAC3（步骤 8.9）

主机收到从 IC 卡（经过终端）传来的 MAC3 后，应验证 MAC3 的有效性。

如果 MAC3 正确，则执行 11.2.8.10 中描述的步骤，否则主机发给终端错误状态码。

11.2.8.10 显示完成（步骤 8.10）

在收到主机的完成报文后，终端做相应的处理，显示完成信息。

11.2.9 补扣交易

补扣交易允许持卡人在灰锁的电子钱包应用中，对电子钱包补扣上次消费交易未扣除的交易额，并将电子钱包应用解锁。

本交易必须在拥有该电子钱包的上次异常交易记录的终端上进行。异常交易记录至少包括灰锁的电子钱包应用的应用序列号、灰锁的电子钱包脱机交易序号、应扣的交易金额、GMAC。拥有异常交易记录的终端可以是产生灰卡的终端，也可以是通过网络通讯得到异常交易记录的其他终端。

在交易预处理流程中发现电子钱包应用已灰锁时，进入本交易流程。

交易预处理流程中，终端收到 IC 卡返回的 GET LOCK PROOF（P1＝’00’）的命令响应报文后，得到上次的灰锁操作的产生日期、时间、MAC2、GTAC 等数据。这些数据是终端进行补扣交易的依据。详细的响应报文参见 9.2.4.10.4 节中的表 10-4。

11.2.9.1 查找异常交易记录（步骤 9.1）

终端得到 IC 卡的响应报文后，在异常交易记录中进行查找，如果有符合条件的异常交易记录，就进入 11.2.9.2 节所描述的步骤；否则显示相应的提示信息。

11.2.9.2 发出 DEBIT FOR UNLOCK 命令（步骤 9.2）

终端发出 DEBIT FOR UNLOCK 命令。

11.2.9.3 检查脱机交易序号和余额（步骤 9.3）

IC 卡收到 DEBIT FOR UNLOCK 命令后，将进行以下操作：

———检查脱机交易序号是否匹配，如果脱机交易序号不匹配，IC 卡将返回’9406’（脱机交易序号错），但不回送其他数据。

———检查电子钱包余额是否大于或等于交易金额。如果小于交易金额，则回送状态码’9401’（金额不足），但不回送其他数据，IC 卡不操作内部出错计数器，终端应采取相应的措施。

通过上面的检查后，IC 卡进入 11.2.9.4 节所描述的步骤。

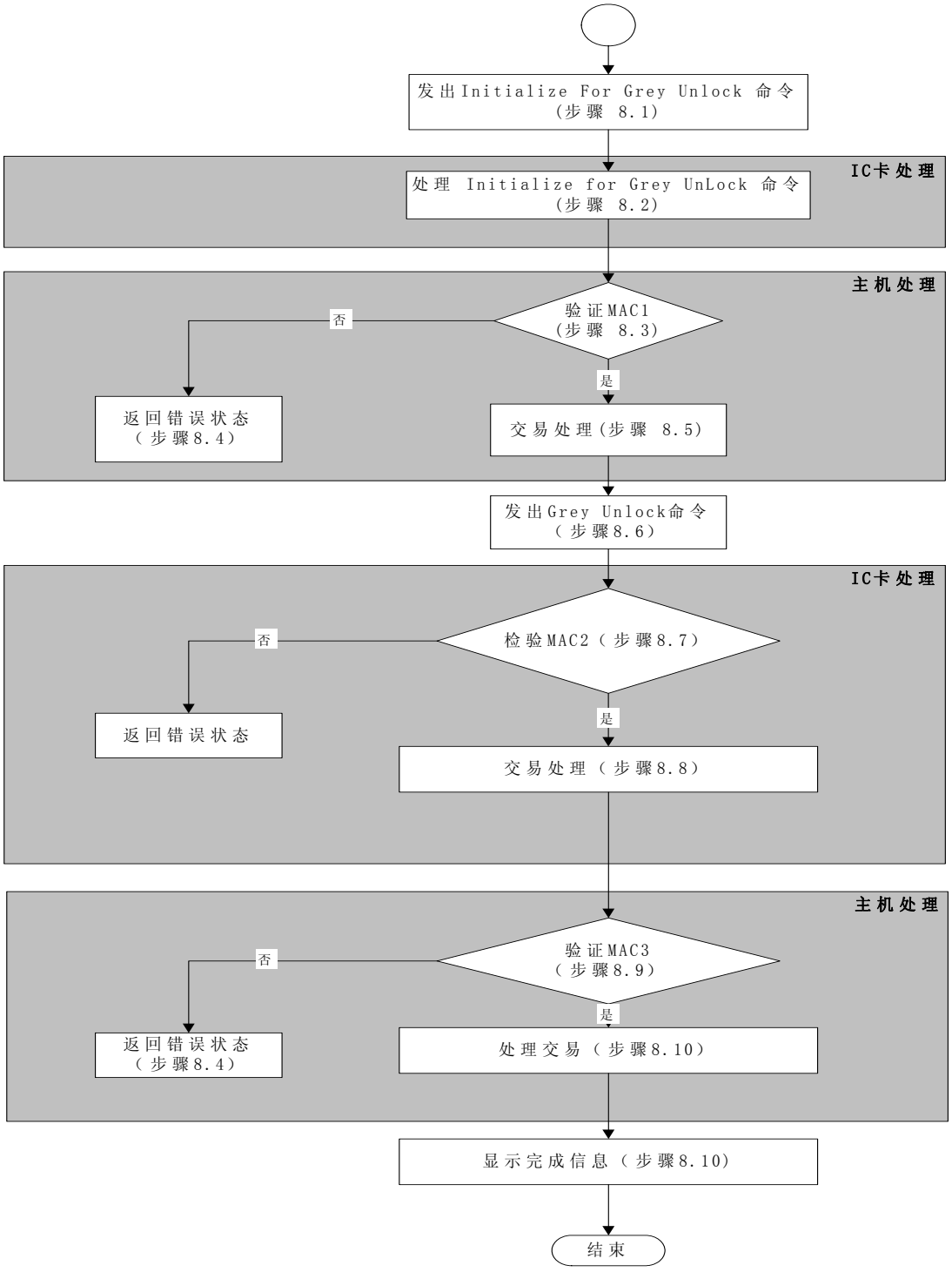


图 11-11 联机解扣交易流程

11.2.9.4 验证 GMAC（步骤 9.4）

IC 卡验证 GMAC 的有效性。如果 GMAC 是有效的，将 IC 卡内部的解扣出错计数器复位，交易处理将继续执行 11.2.9.5 节。如果 GMAC 是无效的，IC 卡返回错误状态码'9302'（MAC 无效）给终端，同时操作内部的解扣出错计数器，出错达到 3 次则临时锁住应用以防止恶意试探。

11.2.9.5 交易处理（步骤 9.5）

IC 卡从卡上的电子钱包余额中减去灰锁消费的交易金额（如果交易金额为 0，则省略对余额的修改）、将电子钱包应用解锁，并将卡内的 TACUF（交易验证码待读标志）置位。

IC 卡应该直接用密钥 DTK 产生一个 TAC。TAC 将被写入终端交易数据包，以便后来传给主机进行交易验证。

下面是用来生成 TAC 的要素：

———交易金额
———交易类型标识
———终端机编号（发出 DEBIT FOR UNLOCK 命令的终端）
———终端交易序号（发出 DEBIT FOR UNLOCK 命令的终端）
———交易日期（发出 DEBIT FOR UNLOCK 命令的日期）
———交易时间（发出 DEBIT FOR UNLOCK 命令的时间）

对于电子钱包的补扣交易，IC 卡将用以下数据组成的一个记录更新标准交易明细。

———电子钱包脱机交易序号
———交易金额
———交易类型标识（电子钱包解扣）
———终端机编号（发出 DEBIT FOR UNLOCK 命令的终端）
———交易日期（发出 DEBIT FOR UNLOCK 命令的日期）
———交易时间（发出 DEBIT FOR UNLOCK 命令的时间）

对于电子钱包的补扣交易，IC 卡将用以下数据组成的一个记录更新内部专用明细文件，以便以后终端可以通过 GET LOCK PROOF 命令得到：

———交易类型标识
———电子钱包代号（'01'=电子钱包）
———电子钱包余额
———电子钱包脱机交易序号码
———终端机编号（发出 DEBIT FOR UNLOCK 命令的终端）
———交易日期（发出 DEBIT FOR UNLOCK 命令的日期）
———交易时间（发出 DEBIT FOR UNLOCK 命令的时间）
———交易金额
———TAC

IC 卡必须全部成功地完成以上几个步骤或者一个也不完成，如果余额的更新、TACUF 的置位、电子钱包应用解锁未成功，标准交易明细和内部专用明细也不应被更新。

11.2.9.6 回送 TAC 码（步骤 9.6）

IC 卡在 DEBIT FOR UNLOCK 命令的响应报文中回送 TAC 码，表明余额已被更新而且电子钱包应用已解锁。

11.2.9.7　读取 TAC 码（步骤 9.7）

终端读取由 IC 卡发来的 TAC 码，合成完整的交易成交数据包。

11.2.9.8　发出 GET LOCK PROOF（P1＝’01’）命令（步骤 9.8）

终端发出 GET LOCK PROOF（P1＝’01’）命令。

11.2.9.9　处理 GET LOCK PROOF（P1＝’01’）命令（步骤 9.9）

IC 卡将内部的 TACUF（交易验证码待读标志）复位。

11.2.10 补充交易

补充交易允许终端读取上次灰锁消费交易或补扣交易中未获得的 TAC，上送主机以供验证。

在交易预处理流程中发现电子钱包解扣操作已完成而 TAC 未成功读取，则进入本交易流程。

交易预处理流程中，终端收到 IC 卡返回的 GET LOCK PROOF（P1＝’00’）的命令响应报文后，得到上次的灰锁操作的产生日期、时间、TAC 等数据。这些数据是终端进行补充交易的依据。

详细的响应报文参见 9.2.4.10.4 节中的表 10-4。

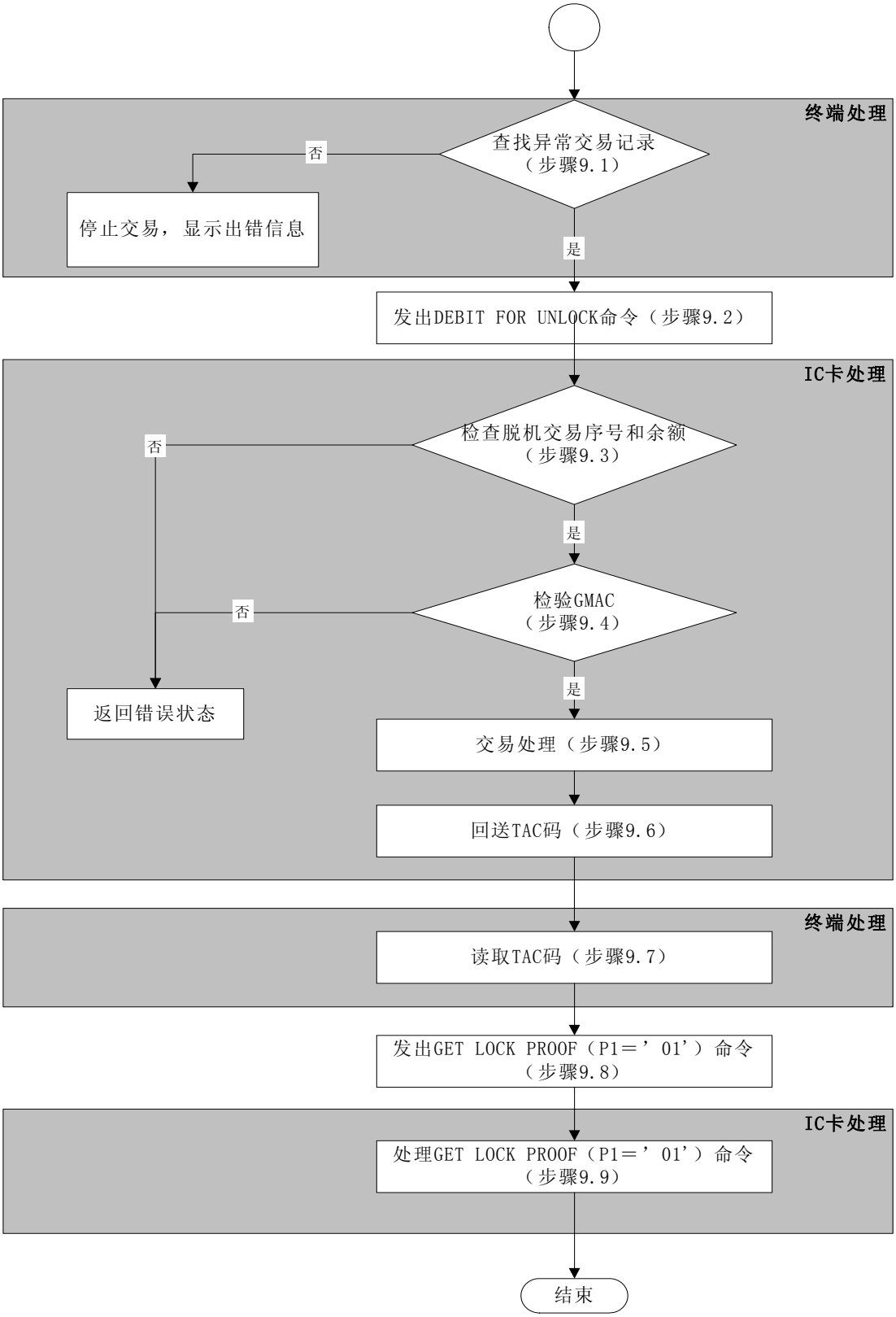


图 11-12 补扣交易流程

11.2.10.1 读取 TAC（步骤 10.1）

终端通过 GET LOCK PROOF 命令中的响应报文，获得上次灰锁消费或补扣交易的

TAC。

11.2.10.2　形成补充交易数据包（步骤 10.2）

终端将读到的 TAC，和其他的相关数据合成一条补充交易数据包，以便上送主机。

11.2.10.3　发出 GET LOCK PROOF（P1＝’01’）命令（步骤 10.3）

终端发出 GET LOCK PROOF（P1＝’01’）命令。

11.2.10.4　处理 GET LOCK PROOF（P1＝’01’）命令（步骤 10.4）

IC 卡将内部的 TACUF 复位。

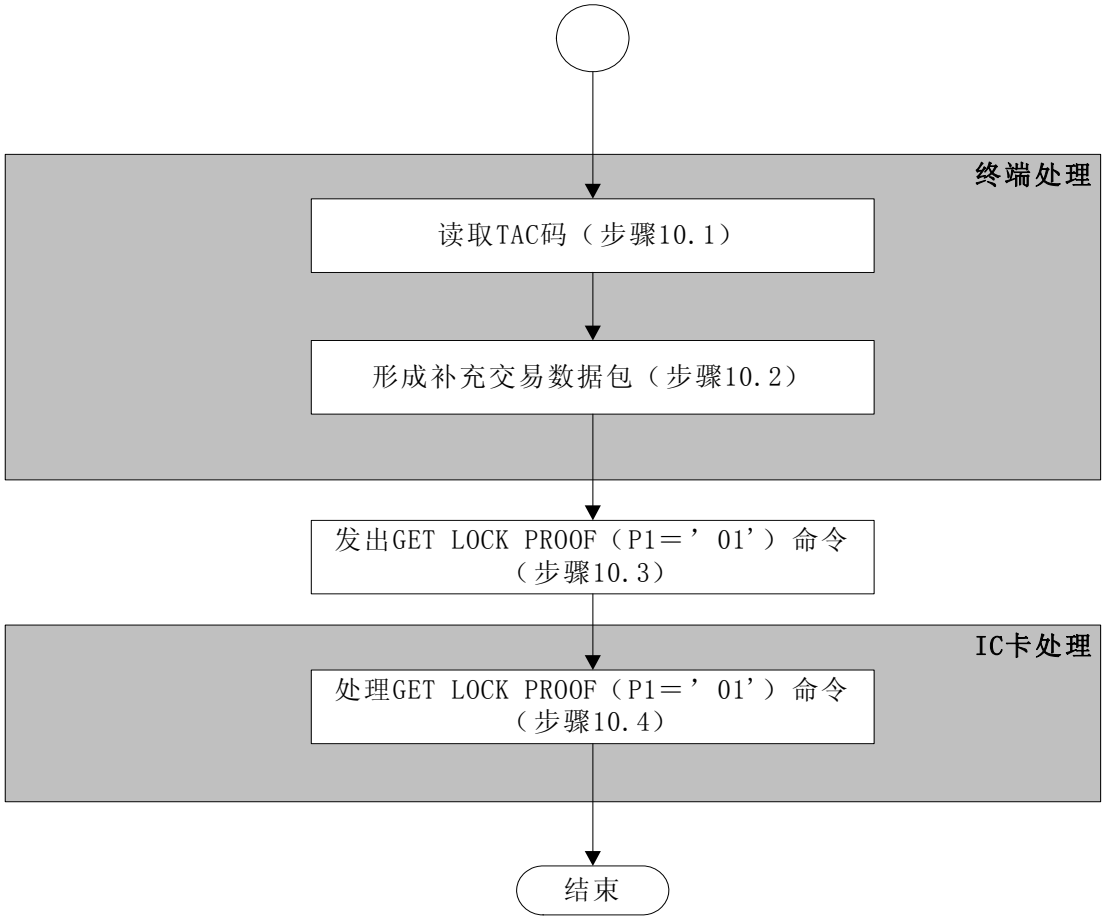


图 11-13 补充交易流程

11.3 建设事业应用交易流程

11.3.1 交易预处理

引用“扩展金融应用交易流程”一节的 11.2.1 节。

11.3.2 圈存交易

引用“扩展金融应用交易流程”一节的 11.2.2 节。

11.3.3 圈提交易

引用“金融应用交易流程”一节的 11.1.3 节。

11.3.4 消费交易

消费交易1允许持卡人使用电子存折或电子钱包的余额进行购物或获取服务。此交易可以在销售点终端（POS）上脱机进行。使用电子存折进行的消费交易应提交个人密码（PIN），使用电子钱包则不需要。

消费交易处理流程见图 11-14。

补充定义如下：

“交易处理”：IC卡从电子钱包余额中扣减消费的金额，电子钱包交易序号加1，更新电子钱包消费交易记录。IC卡应成功地完成以上所有步骤或者一个也不完成。

对于电子钱包消费交易，IC卡将用以下数据组成的一个记录更新交易明细。

—— 交易金额
—— 交易类型标识‘06’
—— 卡片交易序号
—— 终端机编号
—— 交易日期（终端）
—— 交易时间（终端）

“处理INITIALIZE FOR PURCHASE”部分，增加一检查过程：

—— 检查钱包是否被灰锁。如果灰锁，则返回状态码‘9408’（钱包灰锁锁定），但不返回其他信息，同时终止命令的处理过程。

“发出 GET MESSAGE 命令”部分，增加一认证过程：

—— 检查PSAM卡消费密钥权限状态。

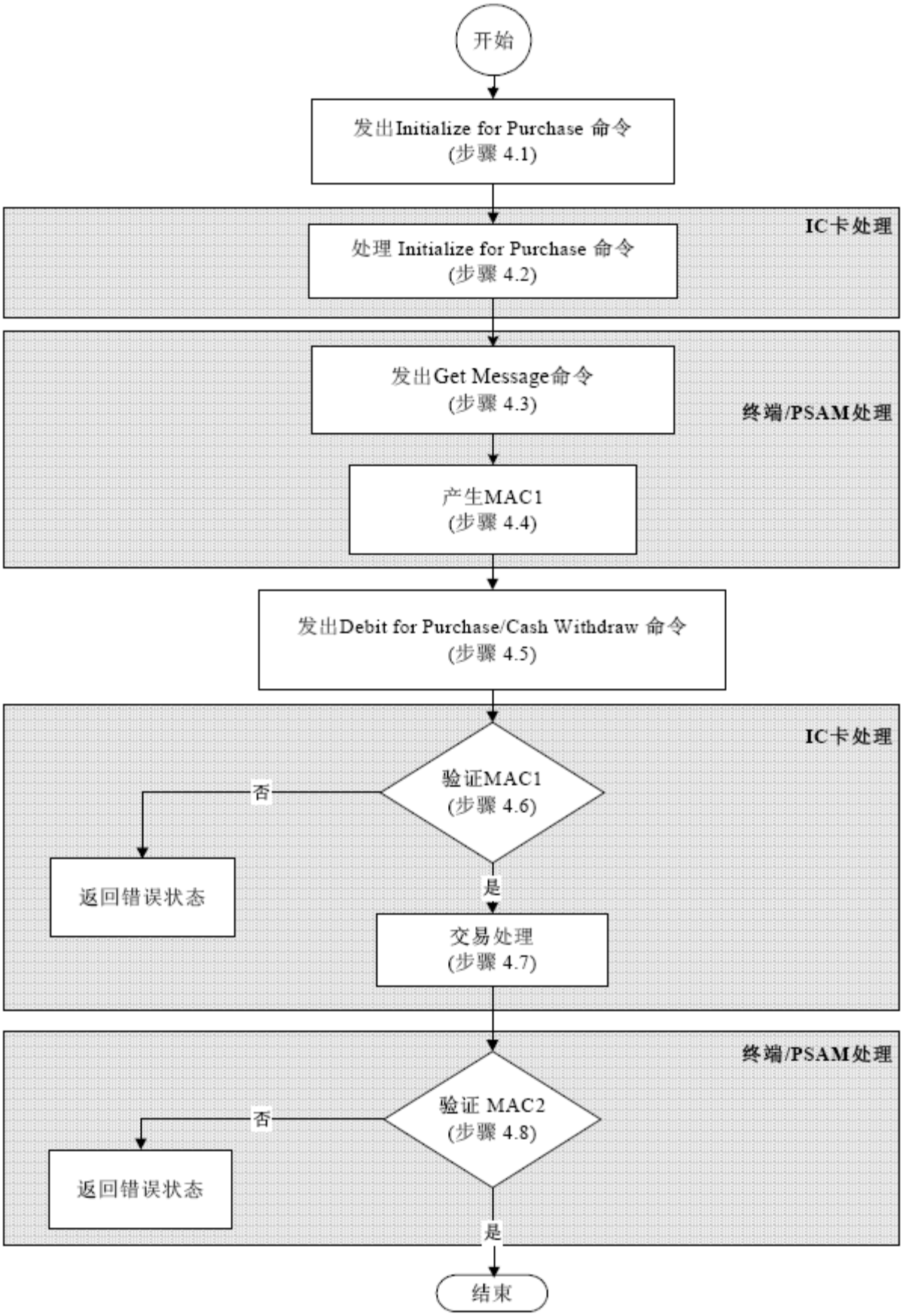

开始
发出Initialize for Purchase 命令
(步骤 4.1)
IC卡处理
处理 Initialize for Purchase 命令
(步骤 4.2)
发出Get Message命令
(步骤 4.3)
终端/PSAM处理
产生MAC1
(步骤 4.4)
发出Debit for Purchase/Cash Withdraw 命令
(步骤 4.5)
IC卡处理
验证MAC1
(步骤 4.6)
否
返回错误状态
是
交易处理
(步骤 4.7)
终端/PSAM处理
验证 MAC2
(步骤 4.8)
否
返回错误状态
是
结束

图 11-14 消费交易处理流程

11.3.4.1 发出 INITIALIZE FOR PURCHASE 命令（步骤 3.1）

终端发出INITIALIZE FOR PURCHASE命令启动消费交易。

11.3.4.2 处理 INITIALIZE FOR PURCHASE 命令（步骤 3.2）

IC卡收到INITIALIZE FOR PURCHASE命令后，将进行以下操作：

—— 检查是否支持命令中提供的密钥索引号。如果不支持，则返回状态码‘9403’（不支持的密钥索引），但不返回其他数据；

—— 检查电子存折余额或电子钱包余额是否大于或等于交易金额。如果小于交易金额，则返回状态码‘9401’（资金不足），但不返回其他数据。

在通过以上检查之后，IC卡将在发出DEBIT FOR PURCHASE/CASH WITHDRAW 命令（步骤4.5）中产生一个伪随机数（ICC）和过程密钥用于验证MAC1。该过程密钥是利用DPK并参见产生的机制参见附录C。

用于产生该过程密钥的输入数据如下：

SESPK：伪随机数（ICC）||电子存折脱机交易序号或电子钱包脱机交易序号||终端交易序号的最右两个字节。

11.3.4.3 发出 GET MESSAGE 命令（步骤 3.3）

检查PSAM卡消费密钥权限状态，得到安全认证识别码。

11.3.4.4 产生 MAC1（步骤 3.4）

使用伪随机数（ICC）和IC卡返回的电子存折脱机交易序号或电子钱包脱机交易序号，终端的安全
存取模块（PSAM）将产生一个过程密钥（SESPK）和一个报文认证码（MAC1），供IC卡来验证PSAM的合法性，MAC1的计算机制参见附录C。

用SESPK对以下数据进行加密产生MAC1（按所列顺序）：

—— 交易金额；
—— 交易类型标识；
—— 终端机编号；
—— 交易日期（终端）；
—— 交易时间（终端）；
—— 安全认证识别码。

11.3.4.5 发出 DEBIT FOR PURCHASE/CASH WITHDRAW 命令（步骤 3.5）

终端发出DEBIT FOR PURCHASE/CASH WITHDRAW命令。

11.3.4.6 验证 MAC1（步骤 3.6）

在收到DEBIT FOR PURCHASE/CASH WITHDRAW命令后，IC卡将验证MAC1的有

效性。如果MAC1有效， 交易处理将继续执行交易处理（步骤4.7）中所描述的步骤。否则将向终端返回错误状态码‘9302’（MAC无效）。

11.3.4.7 交易处理（步骤 3.7）

IC卡从电子存折余额或电子钱包余额中扣减消费的金额，并将电子存折或电子钱包脱机交易序号加1。IC卡应成功地完成以上所有步骤或者一个也不完成。只有余额和序号的更新均成功后，交易明细才可更新。

IC卡产生一个报文认证码（MAC2）供PSAM对其进行合法性检查，并通过DEBIT FOR PURCHASE/CASHWITHDRAW命令响应报文返回以下数据，作为PASM产生MAC2的输入数据。用SESPK对以下数据进行加密产生MAC2：

—— 交易金额。

用密钥DTK左右8位字节异或运算后的结果产生TAC。TAC将被写入终端交易明细，以便于主机进行交易验证。下面是用来生成TAC的数据，它们以明文形式通过CREDTE FOR PURCHASE/CASH WITHDRAW命令的响应报文从IC卡传送到终端：

—— 交易金额；
—— 交易类型标识；
—— 终端机编号；
—— 终端交易序号；
—— 交易日期（终端）；
—— 交易时间（终端）。

对于电子存折消费交易和电子钱包消费交易（可选），IC卡将用以下数据组成的一个记录更新交易
明细：

—— 电子存折脱机交易序号或电子钱包脱机交易序号；
—— 交易金额；
—— 交易类型标识；
—— 终端机编号；
—— 交易日期（终端）；
—— 交易时间（终端）。

11.3.4.8 验证 MAC2（步骤 3.8）

在收到IC卡（经过终端）传来的MAC2后，PSAM要验证MAC2的有效性。MAC2验证的结果被传送到终端以便采取必要的措施。

11.3.5 复合应用消费交易

复合应用消费交易允许持卡人使用电子钱包的余额进行购物或获取服务。此交易可以在终端设备或其他读卡设备上脱机进行。此交易无需提交个人密码（PIN）。复合应用说明参见附录F。

复合应用消费交易允许消费金额为0。

复合应用消费交易见图 11-15。

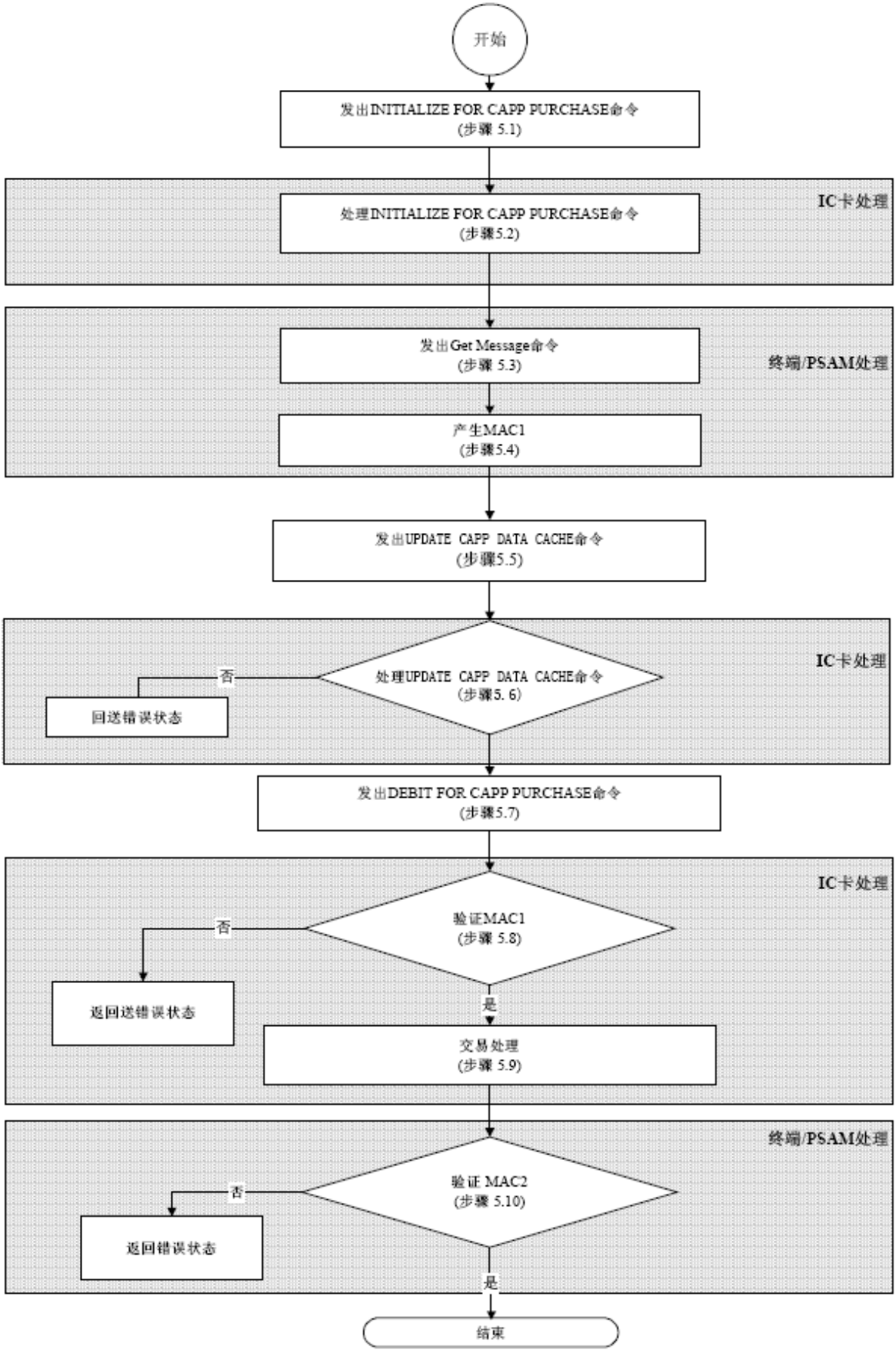


图 11-15 复合应用消费交易

11.3.5.1 发出 INITIALIZE FOR CAPP PURCHASE 命令（步骤 4.1）

终端发出INITIALIZE FOR CAPP PURCHASE命令启动复合应用消费交易。

11.3.5.2 处理 INITIALIZE FOR CAPP PURCHASE 命令（步骤 4.2）

IC卡收到INITIALIZE FOR CAPP PURCHASE 命令后，将进行以下操作：

检查是否支持命令中提供的密钥索引号。如果不支持，则返回状态码‘9403’（不支持的密钥索引），但不返回其他数据。

检查钱包是否被灰锁，如果灰锁，则返回状态码‘9408’（钱包灰锁锁定），但不返回其他信息，同时终止命令的处理过程。

检查电子钱包余额是否大于或等于交易金额。如果小于交易金额，则返回状态码’9401’，但不返回其他数据。

在通过以上检查之后，IC卡将产生一个伪随机数（ICC）和过程密钥。该过程密钥产生的机制参见“安全计算”一章，用于产生该过程密钥的输入数据如下：

SESPK：伪随机数（ICC）||电子钱包交易序号||终端交易序号的最右两个字节。

11.3.5.3 发出 GET MESSAGE 命令（步骤 4.3）

检查PSAM卡消费密钥权限状态。

11.3.5.4 产生 MAC1（步骤 4.4）

使用伪随机数（ICC）和IC卡返回的电子存折脱机交易序号或电子钱包脱机交易序号，终端的安全存取模块（PSAM）将产生一个过程密钥（SESPK）和一个报文认证码（MAC1），供IC卡来验证PSAM的合法性。

用SESPK对以下数据进行加密产生MAC1（按所列顺序）：

—— 交易金额
—— 交易类型标识
—— 终端机编号
—— 交易日期（终端）
—— 交易时间（终端）
—— 安全认证识别码

11.3.5.5 发出 UPDATE CAPP DATA CACHE 命令（步骤 4.5）

终端发出UPDATE CAPP DATA CACHE命令。

11.3.5.6 处理 UPDATE CAPP DATA CACHE 命令（步骤 4.6）

IC卡在收到UPDATE CAPP DATA CACHE 命令后，将进行以下操作：

如果命令中存在SFI域，检查卡片当前应用下是否存在与命令中SFI值相同的文件。如果不存在，返回状态码‘6A82’（未找到文件），但不返回其他数据。终端应终止此次复合应用消费交易。

根据命令中的复合应用类型标识符，查询复合应用专用文件中是否存在相同标识符的记录。如果不存在，则返回状态码‘6A83’（未找到记录），但不返回其他数据。终端应终止此次复合应用消费交易。

检查复合应用专用文件中相应记录中的应用锁定标志字节。如果应用锁定标志为设置，则返回状态码‘9407’（复合应用禁止），但不返回其他数据。终端应终止此次复合应用消费交易。

检查命令中的数据域长度是否大于复合应用专用文件中相应记录的长度。如果大于，则返回状态码‘6A84’（文件中存储空间不够），但不返回其他数据。终端应终止此次复合应用消费交易。

在通过以上检查后，IC卡应暂存命令中的SFI、记录号、复合应用类型标识符和数据域。复合应用

专用文件中相应记录中的数据不得通过此命令更新。

11.3.5.7 发出 DEBIT FOR CAPP PURCHASE 命令（步骤 4.7）

终端发出DEBIT FOR CAPP PURCHASE命令。

11.3.5.8 验证 MAC1（步骤 4.8）

在收到DEBIT FOR CAPP PURCHASE命令后，IC卡将验证MAC1的有效性。如果MAC1有效，交易处理将继续执行,否则将向终端返回错误状态码‘9302’（MAC无效）。

11.3.5.9 交易处理（步骤 4.9）

IC卡从电子钱包余额中扣减消费的金额，电子钱包交易序号加1，根据处理UPDATE CAPP DATA CACHE 命令（步骤4.6）中暂存的数据更新复合应用专用文件，更新电子钱包消费交易记录。IC 卡应成功地完成以上所有步骤或者一个也不完成。

在根据中处理UPDATE CAPP DATA CACHE 命令（步骤4.6）暂存的数据更新复合应用专用文件时，如果更新数据长度小于记录长度，IC卡应在数据后自动填充‘00’至记录尾。

IC 卡产生一个报文认证码 （MAC2）供PSAM 对其进行合法性检查，并通过DEBIT FOR CAPP PURCHASE 命令响应报文返回以下数据，作为PSAM产生MAC2的输入数据。用SESPK 对以下数据进行加密产生MAC2：

——— 交易金额

用密钥DTK左右8位字节异或运算后的结果产生TAC。TAC将被写入终端交易明细，以便于主机进行交易验证。下面是用来生成TAC的数据，它们以明文形式通过DEBIT FOR CAPP PURCHASE命令的响应报文从IC卡传送到终端：

——— 交易金额
——— 交易类型标识
——— 终端机编号
——— 终端交易序号
——— 交易日期（终端）

——— 交易时间（终端）

对于电子钱包消费交易，IC卡将用以下数据组成的一个记录更新交易明细。

——— 交易类型标识‘09’
——— 交易金额
——— 卡片交易序号
——— 终端机编号
——— 交易日期（终端）
——— 交易时间（终端）

11.3.5.10　验证 MAC2 （步骤 4.10）

在收到IC卡（经过终端）传来的MAC2后，PSAM要验证MAC2的有效性。MAC2验证的结果被传送到终端以便采取必要的措施。

11.3.6 修改透支限额交易

引用“金融应用交易流程”一节的 11.1.6 节。

11.3.7 查询余额交易

引用“扩展金融应用交易流程”一节的 11.2.5 节。

11.3.8 查询明细交易

引用“扩展金融应用交易流程”一节的 11.2.6 节。

11.3.9 灰锁消费交易

灰锁消费交易允许持卡人使用电子钱包进行灰锁消费。此交易可以脱机进行。

灰锁消费交易流程见图 11-16。

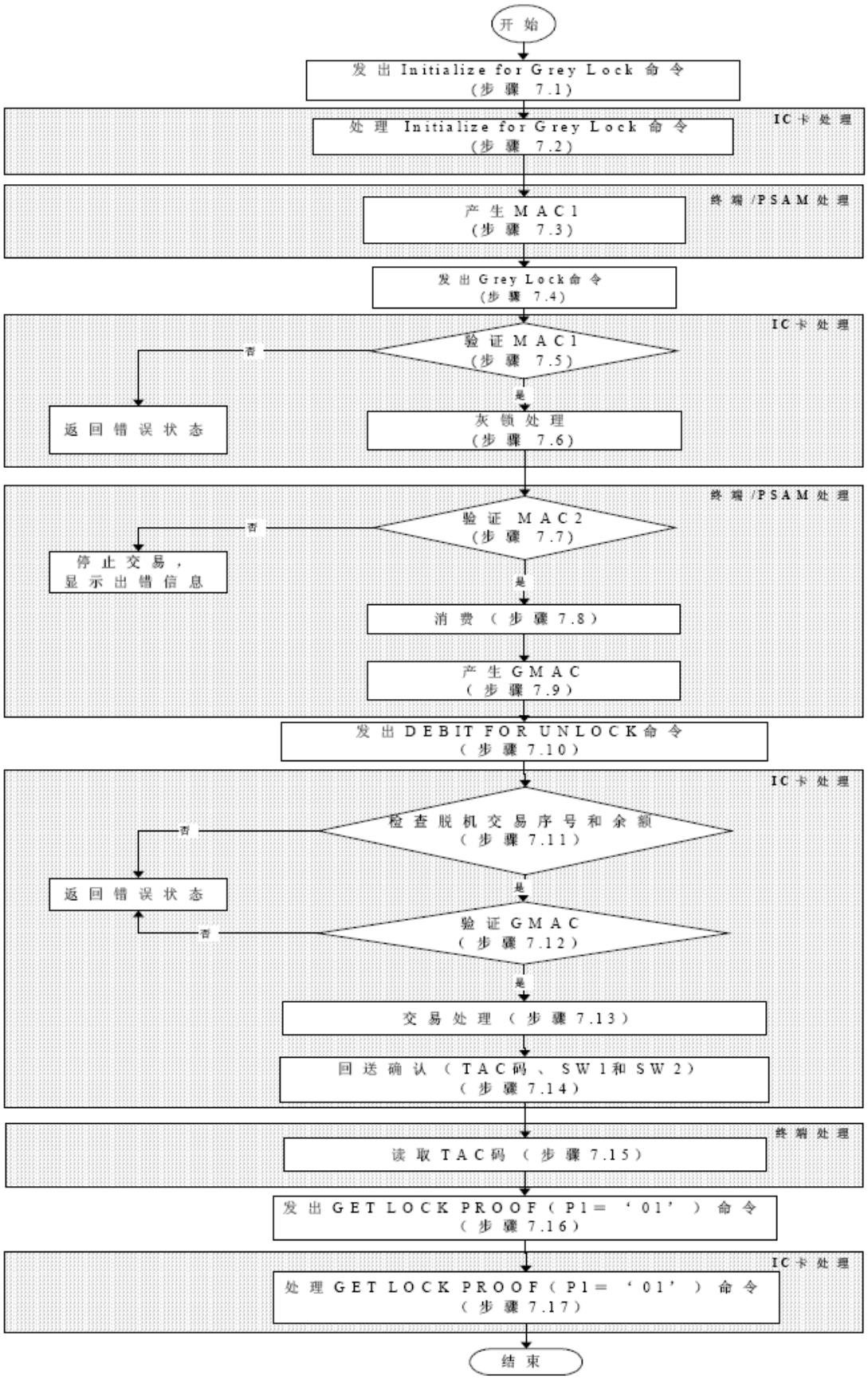

开始
发出Initialize for Grey Lock命令
(步骤 7.1)
IC卡处理
处理 Initialize for Grey Lock命令
(步骤 7.2)
终端/PSAM处理
产生MAC1
(步骤 7.3)
发出Grey Lock命令
(步骤 7.4)
IC卡处理
验证MAC1
(步骤 7.5)
否
是
返回错误状态
灰锁处理
(步骤 7.6)
终端/PSAM处理
验证 MAC2
(步骤 7.7)
否
是
停止交易，
显示出错信息
消费 (步骤7.8)
产生GMAC
(步骤7.9)
发出DEBIT FOR UNLOCK命令
(步骤7.10)
IC卡处理
检查脱机交易序号和余额
(步骤7.11)
否
是
返回错误状态
验证GMAC
(步骤7.12)
否
是
交易处理 (步骤7.13)
回送确认 (TAC码、SW1和SW2)
(步骤7.14)
终端处理
读取TAC码 (步骤7.15)
发出GET LOCK PROOF (P1='01') 命令
(步骤7.16)
IC卡处理
处理GET LOCK PROOF (P1='01') 命令
(步骤7.17)
结束

图 11-16 灰锁消费交易流程

11.3.9.1 发出 INITIALIZE FOR GREY LOCK 命令（步骤 7.1）

终端发出 INITIALIZE FOR GREY LOCK 命令启动灰锁消费交易。

11.3.9.2 处理 INITIALIZE FOR GREY LOCK 命令（步骤 7.2）

IC 卡收到 INITIALIZE FOR GREY LOCK 命令后，将进行以下操作：

—— 检查命令中包含的密钥索引号是否被 IC 卡支持。如果不支持，返回状态码‘9403’（不支持的密钥索引号）且不返回其他数据。

在通过以上检查之后，IC 卡将产生一个伪随机数，这个伪随机数将包含在本命令的响应报文中返回终端，之后，IC 卡将内部的 TACUF 复位。

11.3.9.3 计算 MAC1（步骤 7.3）

使用IC卡返回的伪随机数和电子钱包脱机交易序号，终端的安全存取模块（PSAM）将产生一个终端随机数（TRAN），一个过程密钥（GSESPK）和一个报文认证码（MAC1），供IC卡来验证PSAM的合法性。

过程密钥GSESPK被用于电子钱包的灰锁消费交易。

过程密钥的产生分两步：即先是用DPK密钥并参见附录C产生的中间密钥，再用中间密钥采用下述的算法产生过程密钥。

用来产生中间密钥的输入数据如下：

TMPCK：伪随机数（ICC）|| 电子钱包脱机交易序号 || 终端交易序号的最右两个字节。

用中间密钥对终端随机数（TRAN）加密，运算的结果产生过程密钥：

GSESPK =DES（TMPCK，TRAN || ‘80000000’）。

用 GSESPK 对以下数据进行加密产生 MAC1（按所列顺序）：

—— 交易类型标识
—— 终端机编号
—— 交易日期
—— 交易时间

11.3.9.4 发出 GREY LOCK 命令（步骤 7.4）

终端发出 GREY LOCK 命令。

11.3.9.5 验证 MAC1（步骤 7.5）

IC 卡收到 GREY LOCK 命令后，将产生同样的过程密钥（GSESPK）并验证 MAC1 是否有效。如果 MAC1 是有效的，交易处理将继续。如果 MAC1 是无效的，IC 卡返回错误状态码‘9302’（MAC 无效）给终端。

11.3.9.6　灰锁处理（步骤 7.6）

IC 卡将电子钱包脱机交易序号加 1，并将电子钱包应用灰锁。

IC 卡产生一个报文认证码(MAC2)供 PSAM 对 IC 卡合法性进行检查，并同时将 MAC2 写入内部文件。MAC2 将包含在从卡传送到 PSAM（通过终端）GREY LOCK 的命令响应报文和 GET LOCK PROOF 的命令响应报文中。

用 GSESPK 对以下这些数据进行加密产生 MAC2：

— 电子钱包余额
— 电子钱包脱机交易序号（加 1 前）

参见附录 C 用密钥 DTK 产生一个 GTAC。GTAC 将包含在从卡传送到 PSAM（通过终端）的 GREY LOCK 的命令响应报文和 GET LOCK PROOF 的命令响应报文中。如果之后出现交易异常中断等，使 DEBIT FOR UNLOCK 指令无法当时执行成功，GTAC 可供终端纳入终端异常交易数据中，以便后来上传给主机进行灰锁验证。

下面是用来生成 GTAC 的要素：

— 交易类型标识
— 终端机编号
— 终端交易序号
— 交易日期（终端）
— 交易时间（终端）

IC 卡应把 GSESPK 存贮到安全的内部文件中，（IC 卡也可以将终端随机数、伪随机数（ICC）、终端交易序号等，写入内部文件，通过计算重新获得），以备交易中途 IC 卡掉电后，在后续交易流程中恢复过程密钥 GSESPK。

IC 卡将用以下数据组成的一个记录来更新内部专用明细。这个明细记录中的数据将包含在 GET LOCKPROOF 的命令响应报文中，由 IC 卡返回给终端。

— 交易类型标识（‘91’＝电子钱包灰锁）
— 电子钱包代号（‘01’＝电子钱包）
— 电子钱包余额
— 电子钱包脱机交易序号
— 终端机编号
— 交易日期
— 交易时间
— MAC2
— GTAC

IC 卡应全部成功地完成以上几个步骤或者一个也不完成，如果脱机交易序号的更新、电子钱包应用灰锁状态的设置没有成功，交易明细也不应更新。

11.3.9.7　验证 MAC2（步骤 7.7）

在收到 IC 卡(经终端)传来的 MAC2 后，PSAM 要验证 MAC2 的有效性。MAC2 如果有效，交易继续进行持卡人进行消费行为（步骤 7.8）中所描述的步骤；如果 MAC2 是

无效的，终端应停止交易并采取相应的措施。

11.3.9.8 持卡人进行消费行为（步骤 7.8）

持卡人进行消费行为。在进行消费过程中，允许终端对 IC 卡下电。若下电以后，IC 卡重新上电，经过交易预处理（选择应用、验证个人密码等）后应可以继续执行产生 GMAC（步骤 7.9）中所描述的步骤而不受影响。

11.3.9.9 产生 GMAC（步骤 7.9）

安全存取模块（PSAM）根据专用消费的金额，用过程密钥（GSESPK）产生一个报文认证码（GMAC），供 IC 卡来验证 PSAM 的合法性。GMAC 的计算机制参见“安全计算”一章。

用 GSESPK 对以下数据进行加密产生 GMAC：

— 交易金额；
— 安全认证识别码。

11.3.9.10 发出 DEBIT FOR UNLOCK 命令（步骤 7.10）

终端发出 DEBIT FOR UNLOCK 命令。

11.3.9.11 检查脱机交易序号和余额（步骤 7.11）

收到 DEBIT FOR UNLOCK 命令后，IC 卡将进行以下操作：

— 检查脱机交易序号是否匹配，如果脱机交易序号不匹配，IC 卡将返回‘9406’（脱机交易序号错），但不返回其他数据；

— 检查电子钱包余额是否大于或等于交易金额。如果小于交易金额，则返回状态码‘9401’（金额不足），但不返回其他数据，IC 卡不操作内部出错计数器，终端应采取相应的措施。

通过上面的检查后，IC 卡进入验证 GMAC（步骤 7.12）。

11.3.9.12 验证 GMAC（步骤 7.12）

IC 卡验证 GMAC 的有效性。如果 GMAC 是有效的，将 IC 卡内部的解扣出错计数器复位，交易处理将继续执行交易处理（步骤 7.13）。如果 GMAC 是无效的，IC 卡返回错误状态码‘9302’（MAC 无效）给终端，

同时操作解扣出错计数器，3 次出错则临时锁住应用以防止恶意试探。该解扣出错计数器将在应用解锁命令执行成功后被复位。

11.3.9.13 交易处理（步骤 7.13）

IC 卡从卡上的电子钱包余额中减去灰锁消费的交易金额（如果交易金额为 0，则省略对余额的修改）、将电子钱包解锁、并将卡内的 TACUF（交易验证码待读标志）置位。

用密钥 DTK 产生一个 TAC。TAC 将被写入终端交易数据包，以便后来传给主机进行交易验证。

下面是用来生成 TAC 的要素（按所列顺序）：

— 交易金额
— 交易类型标识（‘93’=电子钱包解扣）
— 终端机编号（发出 DEBIT FOR UNLOCK 命令的终端）
— 终端交易序号（发出 DEBIT FOR UNLOCK 命令的终端）
— 交易日期（发出 DEBIT FOR UNLOCK 命令的日期）
— 交易时间（发出 DEBIT FOR UNLOCK 命令的时间）

对于电子钱包的灰锁消费交易，IC 卡将用以下数据组成的一个记录更新标准交易明细。

— 电子钱包脱机交易序号
— 交易金额
— 交易类型标识（‘93’=电子钱包解扣）
— 终端机编号（发出 DEBIT FOR UNLOCK 命令的终端）
— 交易日期（发出 DEBIT FOR UNLOCK 命令的日期）
— 交易时间（发出 DEBIT FOR UNLOCK 命令的时间）

对于电子钱包的灰锁消费交易，IC 卡将用以下数据组成的一个记录更新内部专用明细文件，以便以后终端可以通过 GET LOCK PROOF 命令得到：

— 交易类型标识
— 电子钱包代号（‘01’=ET）
— 电子钱包余额
— 电子钱包脱机交易序号
— 终端机编号（发出 DEBIT FOR UNLOCK 命令的终端）
— 交易日期（发出 DEBIT FOR UNLOCK 命令的日期）
— 交易时间（发出 DEBIT FOR UNLOCK 命令的时间）
— 交易金额
— TAC

IC 卡应全部成功地完成以上几个步骤或者一个也不完成，如果余额的更新、TACUF 的置位、电子钱包应用的灰锁状态的恢复没有成功，标准交易明细和内部专用明细也不应被更新。

11.3.9.14　返回确认（步骤 7.14）

IC 卡在 DEBIT FOR UNLOCK 命令的响应报文中返回 TAC 码和 SW1SW2＝‘9000’，表明余额已被更新而且电子钱包应用已解锁。

11.3.9.15　读取 TAC 码〈步骤 7.15〉

终端读取由 IC 卡发来的 TAC 码，合成完整的交易成交数据包。

11.3.9.16　发出 GET LOCK PROOF（P1＝‘01’）命令（步骤 7.16）

终端发出 GET LOCK PROOF（P1＝‘01’）命令。

11.3.9.17　处理 GET LOCK PROOF（P1='01'）命令（步骤 7.17）

IC 卡将内部的 TACUF（交易验证码待读标志）复位。

11.3.10 联机解扣交易

引用"扩展金融应用交易流程"一节的 11.2.8 节。

11.3.11 补扣交易

引用"扩展金融应用交易流程"一节的 11.2.9 节。

11.3.12 补充交易

引用"扩展金融应用交易流程"一节的 11.2.10 节。

附录 A

编号	命　令	类别	操作码	功能描述
基本命令集				
1	APPEND RECORD	00/04	E2	添加记录
2	APPLICATION BLOCK	84	1E	应用锁定
3	APPLICATION UNBLOCK	84	18	解锁当前锁定的应用
4	CARD BLOCK	84	16	环境锁定
5	EXTERNAL AUTHENTICATE	00	82	外部认证
6	GET CHALLENGE	00	84	取随机数
7	GET RESPONSE	00	C0	取响应
8	INTERNAL AUTHENTICATE	00	88	内部认证
9	READ BINARY	00/04	B0	读透明文件
10	READ RECORD	00/04	B2	读记录
11	SELECT FILE	00	A4	选择文件或应用
12	UPDATE BINARY	00/04	D6	修改透明文件内容
13	UPDATE RECORD	00/04	DC	修改记录
14	VERIFY PIN	00	20	验证个人密码
金融专有命令				
15	CHANGE PIN	80	5E	持卡人更新密码
16	CREDIT FOR LOAD	80	52	圈存
17	DEBIT FOR PURCHASE/CASH WITHDRAW	80	54	普通消费/存折取现
18	DEBIT FOR UNLOAD	80	54	圈提
19	GET BALANCE	80	5C	读余额
20	GET TRANSACTION PROOF	E0	7A	联机解扣初始化
21	INITALIZE FOR CASH WITHDRAW	80	50	初始化取现交易
22	INITALIZE FOR LOAD	80	50	圈存初始化
23	INITALIZE FOR PURCHASE	80	50	普通消费初始化
24	INITALIZE FOR UNLOAD	80	50	圈提初始化
25	INITALIZE FOR UPDATE	80	50	修改透支限额初始化
26	RELOAD PIN	80	5E	重装个人密码
27	UPDATE OVERDRAW LIMIT	80	58	修改透支限额
金融扩展命令				
28	DEBIT FOR CAPP PURCHASE	80	54	复合应用消费交易
29	DEBIT FOR UNLOCK	E0	7E	电子钱包解扣操作
30	GET LOCK PROOF	E0	CA	读取电子钱包应用的灰锁状态以及相关的证明码
31	GET TRANSACTION PROOF	80	5A	获取交易认证码 TAC（或者

编号	命　令	类别	操作码	功能描述
				GTAC）和 MAC
32	GREY LOCK	E0	7C	灰锁电子钱包
33	GREY UNLOCK	E0	7E	联机解扣交易
34	INITIALIZE FOR CAPP PURCHASE	80	50	初始化复合应用消费交易
35	INITIALIZE FOR LOAD	80	50	金融初始化圈存交易
36	INITIALIZE FOR GREY LOCK	E0	7A	初始化灰锁操作
37	INITIALIZE FOR GREY UNLOCK	E0	7A	初始化联机解扣
38	INITIALIZE FOR PURCHASE	80	50	初始化消费
39	UPDATE CAPP DATA CACHE	80	DC	更新复合应用数据缓存
建设事业专用命令				
40	CHANGE PIN	80	5E	持卡人更新密码
41	CREDIT FOR LOAD	80	52	圈存
42	DEBIT FOR CAPP PURCHASE	80	54	复合应用消费交易
43	DEBIT FOR PURCHASE/CASH WITHDRAW	80	54	普通消费/存折取现
44	DEBIT FOR UNLOAD	80	54	圈提
45	DEBIT FOR UNLOCK	E0	7E	电子钱包解扣操作
46	GET BALANCE	80	5C	读余额
47	GET MESSAGE			
48	GET TRANSACTION PROVE	80	5A	获取交易认证码 TAC（或者 GTAC）和 MAC
49	GET LOCK PROOF	E0	CA	读取电子钱包应用的灰锁状态以及相关的证明码
50	GREY LOCK	E0	7C	灰锁电子钱包
51	GREY UNLOCK	E0	7E	联机解扣交易
52	INITIALIZE FOR PURCHASE	80	50	初始化消费
53	INITIALIZE FOR CASH WITHDRAW	80	50	初始化取现交易
54	INITIALIZE FOR GREY LOCK	E0	7A	初始化灰锁操作
55	INITIALIZE FOR GREY UNLOCK	E0	7A	初始化联机解扣
56	INITIALIZE FOR LOAD	80	50	金融初始化圈存交易
57	INITIALIZE FOR UNLOAD	80	50	圈提初始化
58	INITIALIZE FOR UPDATE	80	50	修改透支限额初始化
59	RELOAD PIN	80	5E	重装个人密码
60	UPDATE CAPP DATA CACHE	80	DC	更新复合应用数据缓存

编号	命　令	类别	操作码	功能描述
61	UPDATE OVERDRAW LIMIT	80	58	修改透支限额
发卡管理命令				
62	CLEAR DF	BF	CE	删除 DF 文件体内容
63	CREATE FILE	80	E0	建立文件
64	FREEZE MF	BF	FE	冻结 MF
65	GET INFO	BF	C8	取卡的特征信息
66	WRITE KEY	80/84	D4	更新密钥
67	SET SYS PARAM	BF	0A	设置系统参数