ICS 35.240.15
L 64

中华人民共和国国家标准

GB/T 16649.4—2010/ISO/IEC 7816-4:2005

识别卡　集成电路卡
第4部分:用于交换的结构、安全和命令

Identification Cards—Integrated circuit cards—
Part 4:Organization,security and commands for interchange

(ISO/IEC 7816-4:2005,IDT)

2010-12-01 发布　　2011-04-01 实施

中华人民共和国国家质量监督检验检疫总局
中国国家标准化管理委员会　发布

目　次

前　言

GB/T 16649 在总标题《识别卡　集成电路卡》下目前由下述 14 个部分构成：

——第 1 部分：带触点的卡　物理特性；

——第 2 部分：带触点的卡　触点的尺寸和位置；

——第 3 部分：带触点的卡　电信号和传输协议；

——第 4 部分：用于交换的结构、安全和命令；

——第 5 部分：应用标识符的国家编号体系和注册规程；

——第 6 部分：行业间数据元；

——第 7 部分：用于结构化卡查询语言(SCQL)的行业间命令；

——第 8 部分：与安全相关的行业间命令；

——第 9 部分：用于卡管理的命令；

——第 10 部分：带触点的卡　同步卡的电信号和复位应答；

——第 11 部分：通过生物识别方法的个人验证(制定中)；

——第 12 部分：带触点的卡　USB 电气接口和操作规程；

——第 13 部分：在多应用环境中用于应用管理的命令(制定中)；

——第 15 部分：密码信息应用。

本部分为 GB/T 16649 的第 4 部分。

本部分等同采用国际标准 ISO/IEC 7816-4:2005《识别卡　集成电路卡　第 4 部分：用于交换的结构、安全和命令》(英文版)。

为便于使用，本部分作了下列编辑性修改：

a)　删除国际标准前言；

b)　将“本文件”改为“本部分”。

本部分的附录 A、附录 B、附录 C、附录 D 是资料性附录。

本部分由中华人民共和国工业和信息化部提出。

本部分由全国信息技术标准化技术委员会(SAC/TC 28)归口。

本部分起草单位：中国电子技术标准化研究所、北京握奇数据系统有限公司。

本部分主要起草人：金倩、冯敬、耿力、袁理、王文峰、乔申杰。

引　言

GB/T 16649 是规定集成电路卡参数和交换中集成电路卡使用的系列国际标准。集成电路卡是用于信息交换(该信息交换由外界和卡上集成电路之间商定)的识别卡。作为信息交换的结果,卡传送信息(计算结果、存储的数据),和/或更改其内容(数据存储、结果记忆)。

——有 4 个部分规定了带电触点的卡,其中有 3 部分还规定了电接口:

GB/T 16649.1 规定了带触点的卡的物理特性;

GB/T 16649.2 规定了触点的尺寸和位置;

GB/T 16649.3 规定了异步卡的电接口和传输协议;

GB/T 16649.10 规定了同步卡的电接口和复位应答。

——所有其他部分均独立于物理接口技术。它们用于通过触点和/或射频访问的卡:

GB/T 16649.4 规定了用于交换的组件、安全和命令;

GB/T 16649.5 规定了应用提供者的注册;

GB/T 16649.6 规定了用于交换的行业间数据元;

GB/T 16649.7 规定了用于结构化卡查询语言的命令;

GB/T 16649.8 规定了用于安全操作的命令;

GB/T 16649.9 规定了用于卡管理的命令。

识别卡　集成电路卡
第4部分:用于交换的结构、安全和命令

1　范围

GB/T 16649的本部分规定了:

——在接口处交换的命令-响应对的内容;

——获取卡内数据元和数据对象的方法;

——用于描述卡的操作特性的历史字节的结构和内容;

——当处理命令时在接口处所看到的卡内应用和数据的结构;

——访问卡内文件和数据的方法;

——定义访问卡内文件和数据的权限的安全体系结构;

——卡内识别和选择应用的方法和机制;

——安全报文传输的方法;

——访问卡采用的算法的方法。本部分不描述这些算法。

本部分不涵盖卡内和/或外界的内部实现。

本部分独立于物理接口技术。它适用于通过触点、近耦合和射频等方式访问的卡。

2　规范性引用文件

下列文件中的条款通过GB/T 16649的本部分的引用而成为本部分的条款。凡是注日期的引用文件,其随后所有的修改单(不包括勘误的内容)或修订版均不适用于本部分,然而,鼓励根据本部分达成协议的各方研究是否可使用这些文件的最新版本。凡是不注日期的引用文件,其最新版本适用于本部分。

GB/T 16649.3　识别卡　带触点的集成电路卡　第3部分:电信号和传输协议(GB/T 16649.3—2006,ISO/IEC 7816-3:1997,IDT)

GB/T 16649.6　识别卡　带触点的集成电路卡　第6部分:行业间数据元(GB/T 16649.6—2001,idt ISO/IEC 7816-6:1996)

GB/T 16263.1—2006　信息技术　ASN.1编码规则　第1部分:基本编码规则(BER)、正则编码规则(CER)和非典型编码规则(DER)规范(ISO/IEC 8825-1:2002,IDT)

3　术语和定义

下列术语和定义适用于本部分。

3.1

访问规则　access rule

包含针对一个操作的访问模式和操作前要满足的安全条件的数据元。

3.2

复位应答文件　Answer-to-Reset file

表示卡操作特性的可选基本文件。

3.3

应用 application

为满足特定功能所需的数据结构、数据元和程序模块。

3.4

应用 DF application DF

卡上承载应用的结构。

3.5

应用标识符 application identifier

标识应用的数据元(最多 16 字节)。

3.6

应用标签 application label

在人-机界面处使用的数据元。

3.7

应用提供者 application provider

提供卡上应用的组成部分的实体。

3.8

应用模板 application template

与应用相关的数据对象的集合,其中的数据对象包括一个应用标识符数据对象。

3.9

非对称加密技术 asymmetric cryptographic technique

采用两种相关变换进行加密的技术,一种是公钥运算(由公共密钥定义),另一种是私钥运算(由私有密钥定义)。这两种变换具有以下属性,即私钥运算不能通过给定的公钥运算导出。

3.10

证书 certificate

由发行证书的认证中心使用其私钥对实体的公钥、身份信息以及其他相关信息进行签名,形成的不可伪造的数据。

3.11

命令响应对 command-response pair

接口处两个报文的集合,一个命令 APDU 跟着一个相反方向上的响应 APDU。

3.12

数据元 data element

在接口处所看到的信息项,可以是名称、逻辑内容描述、格式和编码。

3.13

数据对象 data object

在接口处所看到的信息,由 tag 字段(必备)、长度字段(必备)和值字段(可选)串联组成。

3.14

数据单元 data unit

在支持数据元的 EF 内能被明确引用的最小的位集合。

3.15

专用文件 dedicated file

包含文件控制信息和可分配的存储空间(可选)的结构。

3.16

DF 名称 DF name

唯一地标识卡内专用文件的数据元(最多 16 字节)。

3.17

数字签名 digital signature

附加于数据串的数据,或对数据串的加密变换,它能够验证该数据串的原始性和完整性,保护数据串不被伪造。

3.18

目录文件 directory file

可选的 EF,包含卡支持的应用列表和可选相关数据元。

3.19

基本文件 elementary file

共用同一文件标识符和同一安全属性的数据单元或记录或数据对象的集合。

3.20

文件 file

卡上应用和(或)数据的结构,如处理命令时接口处所看到的。

3.21

文件标识符 file identifier

用于文件访问的数据元(2 字节)。

3.22

头列表 header list

无定界的 tag 字段和长度字段对的串联。

3.23

识别卡 identification card

一种可识别其持卡人和发卡方的卡,卡上载有其预期应用和有关交易所要求输入的数据。

3.24

内部 EF internal EF

用于存储由卡解释的数据的 EF。

3.25

密钥 key

控制加密操作的符号序列(例如,在动态鉴别、签名制作、签名验证中的加密、解密、私密操作或公共操作)。

3.26

主文件 master file

唯一的 DF,它代表卡上层次结构文件的根。

3.27

偏移 offset

在支持数据单元的 EF 中为顺序引用数据单元的编号,在记录中为顺序引用字节的编号。

3.28

父文件 parent file

在层次结构文件中,一个给定文件的上层 DF。

3.29

口令 password

应用可能需要的、用来鉴别卡的用户的数据。

3.30

路径 path

无定界的文件标识符的链接。

3.31

私钥 private key

一个实体使用的非对称密钥对中仅被该实体使用的密钥。

3.32

提供者 provider

具有或已经获得权力来创建卡中 DF 的权力机构。

3.33

公钥 public key

一个实体使用的非对称密钥对中可以公开的密钥。

3.34

记录 record

在支持 EF 的记录中，可以被卡引用和处理的字节串。

3.35

记录标识符 record identifier

在支持 EF 的记录中，用于引用一个或多个记录的编号。

3.36

记录号 record number

在支持 EF 的记录中，唯一标识每个记录的顺序号。

3.37

注册的应用提供者标识符 registered application provider identifier

唯一标识一个应用提供者的数据元(5 字节)。

3.38

保密密钥 secret key

对称加密技术中仅供指定实体所用的密钥。

3.39

安全报文传输 secure messaging

用于加密保护(部分)命令-响应对的一组方法。

3.40

安全属性 security attribute

卡内各种资源(包括存储的数据和数据处理功能)的使用条件，表现为包含一条或若干条访问规则的数据元。

3.41

安全环境 security environment

卡上应用所需的用于安全报文传输或安全操作的组件的集合。

3.42

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。在不掌握保密密钥的情况下，不可能推导出发送方或接收方的数据变换。

3.43

tag 列表 tag list

无定界的 tag 字段的串联。

3.44

模板 template

构成结构化 BER-TLV 数据对象值字段的 BER-TLV 数据对象的集合。

3.45

工作 EF working EF

用于存储不被卡解释的数据的 EF。

4 符号和缩略语

AID	应用标识符
APDU	应用层协议数据单元
ARR	访问规则引用
ASN.1	抽象语法记法
AT	用于鉴别的控制引用模板
ATR	复位应答
BER	ASN.1 的基本编码规则
CCT	用于密码校验和的控制引用模板
CLA	类别字节
CRT	控制引用模板
CT	用于保密性的控制引用模板
DF	专用文件
DIR	目录
DST	用于数字签名的控制引用模板
EF	基本文件
EF.ARR	访问规则引用文件
EF.ATR	复位应答文件
EF.DIR	目录文件
FCI	文件控制信息
FCP	文件控制参数
FMD	文件管理数据
HT	用于哈希编码的控制引用模板
INS	指令字节
KAT	密钥协议控制引用模板
Lc field	用于编码编号 Nc 的长度字段
LCS byte	生存期状态字节
Le field	用于编码编号 Ne 的长度字段
MF	主文件
Nc	命令数据字段中的字节数
Ne	响应数据字段中期望的字节最大数
Nr	响应数据字段中的字节数
PIX	专用应用标识符扩展

P1-P2	参数字节(为了区分而插入的短划线无意义)
RFU	保留供将来使用
RID	注册的应用提供者标识符
SC	安全条件
SCQL	结构化卡查询语言
SE	安全环境
SEID byte	安全环境标识符字节
SM	安全报文传输
SW1-SW2	状态字节(为了区分而插入的短划线无意义)
(SW1-SW2)	字节 SW1 和 SW2 的串联的值(首字节为最高有效字节)
TLV	tag、长度、值
{T-L-V}	数据对象(为了区分而插入的短划线和花括号无意义)
'XX'	使用十六进制数字'0'至'9'和'A'至'F'的记法,等效于 16 进制的 XX

5 用于交换的结构

为了实现交换,本章定义了下列基本特征结构:

1) 命令-响应对;
2) 数据对象;
3) 应用和数据的结构;
4) 安全体系结构。

5.1 命令-响应对

表 1 示出了一个命令-响应对,即一个命令 APDU 跟随着一个相反方向上的响应 APDU(见 GB/T 16649.3)。通过接口的命令-响应对不可以有交叉,也就是说响应 APDU 应在发起下一个命令-响应对之前接收到。

表 1 命令-响应对

字 段	描 述	字 节 数
命令头	类别字节 CLA	1
	指令字节 INS	1
	参数字节 P1-P2	2
Lc 字段	Nc 编码为 0 则不存在,Nc 编码大于 0 则存在	0、1 或 3
命令数据字段	Nc 编码为 0 则不存在,Nc 编码大于 0 则以 Nc 字符串的形式存在	Nc
Le 字段	Ne 编码为 0 则不存在,Ne 编码大于 0 则存在	0、1、2 或 3
响应数据字段	Nr 编码为 0 则不存在,Nr 编码大于 0 则以 Nc 字符串的形式存在	Nr(最多 Ne)
响应尾标	状态字节 SW1-SW2	2

在所有包含 Lc 和 Le 字段的命令-响应对中,短长度字段和扩展长度字段不应结合在一起:或者是均为短长度字段,或者是均为扩展长度字段。

如果卡明确表明具有处理历史字节(见 8.1.1)中的或 EF.ATR(见 8.2.1.1)中的"扩展 Lc 和 Le 字段"(见表 88,第 3 软件功能表)的能力,则卡处理短长度字段和扩展长度字段;否则卡仅处理短长度字段。

Nc 指示命令数据字段中的字节数。Lc 字段编码 Nc。

——如果 Lc 字段不存在,则 Nc 为零。

——由 1 个字节组成的短 Lc 字段不能置为'00'。

- 从'01'到'FF',字节编码 Nc 从 1 到 255。

——由 3 个字节组成的扩展 Lc 字段:1 个置为'00'的字节后随 2 个置为非'0000'的字节。

- 从'0001'到'FFFF',2 个字节编码 Nc 从 1 到 65535。

Ne 指示期望的响应数据字段中的最大字节数。Le 字段编码 Ne。

——如果 Le 字段不存在,则 Ne 为零。

——由 1 个字节组成的短 Le 字段可以为任何值。

- 从'01'到'FF',字节编码 Ne 从 1 到 255。
- 如果字节被置成'00',则 Ne 为 256。

——如果 Lc 字段不存在,则由 3 个字节(1 个置为'00'的字节后随 2 个置为任意值的字节)组成的扩展 Le 字段,如果扩展 Lc 字段存在,则由 2 个字节(可以是任何值)组成的扩展 Le 字段:

- 从'0001'到'FFFF',2 个字节编码 Ne 从 1 到 65535。
- 如果 2 个字节被置成'0000',则 Ne 为 65536。

Nr 指示响应数据字段中的字节数。Nr 应小于或等于 Ne。因此在所有的命令-响应对中,Le 字段不存在是未接收到响应数据字段的标准方式。如果 Le 字段仅包含置为'00'的字节,则 Ne 为最大值,即对短 Le 字段是在 256 以内,对扩展 Le 字段为 65536,应返回所有可用字节。

如果处理中断,则卡将或许不响应;如果出现响应 APDU,那么响应数据字段应不存在,并且 SW1-SW2 应指出一个差错。

P1-P2 指出处理命令的控制和选项。参数字节置为'00'通常不提供进一步的限定。对参数字节的编码不存在其他通用约定。

用于编码类别字节 CLA(见 5.1.1)、指令字节 INS(见 5.1.2)和状态字节 SW1-SW2(见 5.1.3)的通用约定在下面规定。在这些字节中,除非另有规定,RFU 位应置为 0。

5.1.1 类别字节

CLA 指示命令的类别。根据 GB/T 16649.3,值'FF'是无效的。CLA 的位 8 区分是行业间类别还是专用类别。

位 8 置为 0 表示是行业间类别。值 000xxxxx 和 01xxxxxx 在下面规定。值 001xxxxx 由 ISO/IEC JTC1/SC17 保留供将来使用。

——表 2 规定了 000xxxxx 为首要行业间值。

- 位 8、7、6 置为 000。
- 位 5 控制命令链(见 5.1.1.1)。
- 位 4 和 3 指明安全报文传输(见第 6 章)。
- 位 2 和 1 编码从 0 到 3 的逻辑通道号(见 5.1.1.2)。

表 2 CLA 首要行业间值

b8	b7	b6	b5	b4 b3	b2 b1	含 义
0	0	0	x	— —	— —	命令链控制(见 5.1.1.1)
0	0	0	0	— —	— —	——命令是命令链的最后一条或命令链仅此一条命令
0	0	0	1	— —	— —	——命令不是命令链的最后一条
0	0	0	—	x x	— —	安全报文传输指示
0	0	0	—	0 0	— —	——无 SM 或无指示
0	0	0	—	0 1	— —	——专用 SM 格式
0	0	0	—	1 0	— —	——SM 根据 6,根据 6.2.3.1 命令头不处理
0	0	0	—	1 1	— —	——SM 根据 6,根据 6.2.3.1 命令头鉴别
0	0	0	—	— —	x x	从 0 到 3 的逻辑通道号(见 5.1.1.2)

——表 3 规定了 01xxxxxx 为更进一步的行业间值。

- 位 8 和 7 置为 01。
- 位 6 指明安全报文传输(见第 6 章)。
- 位 5 控制命令链(见 5.1.1.1)。
- 位 4 到 1 编码从 0 到 15,该值加上 4 即为从 4 到 19 的逻辑通道号(见 5.1.1.2)。

表 3 CLA 更进一步行业间值

b8	b7	b6	b5	b4	b3	b2	b1	含　义
0	1	x	—	—	—	—	—	安全报文传输指示
0	1	0	—	—	—	—	—	——无 SM 或无指示
0	1	1	—	—	—	—	—	——SM 根据 6,根据 6.2.3.1 命令头不处理
0	1	—	x	—	—	—	—	命令链控制(见 5.1.1.1)
0	1	—	0	—	—	—	—	——命令是命令链的最后一条或命令链仅此一条命令
0	1	—	1	—	—	—	—	——命令不是命令链的最后一条
0	1	—	—	x	x	x	x	从 4 到 19 的逻辑通道号(见 5.1.1.2)

除值'FF'是无效的外,位 8 置为 1 表明是专用类别。应用定义其他位。

5.1.1.1 命令链

本条规定了在行业间类别中连续的命令-响应对可以被链接在一起的机制。该机制可以在执行多步处理时使用,例如,对单一命令传输的数据串过长时。

如果卡支持该机制,则应在历史字节(见 8.1.1)或 EF.ATR(见 8.2.1.1)中指明该机制(见表 88,第 3 软件功能表)。

本部分仅规定在发起一条不属于命令链一部分的命令-响应对前命令链即被终止的情况下的卡的行为。否则卡的行为没有规定。

对于行业间类别中的命令链,当其他 7 位为常数时,应使用 CLA 的位 5。

——如果位 5 被置为 0,则命令是命令链的最后一条或命令链仅此一条命令。

——如果位 5 被置为 1,则命令不是命令链的最后一条。

响应不是命令链最后一条的命令,SW1-SW2 被置为'9000'意味着处理已经完成,警告指示被禁止(见 5.1.3),而且下列特定错误情形将产生:

——如果 SW1-SW2 被置为'6883',则期望命令链的最后一条。

——如果 SW1-SW2 被置为'6884',则不支持命令链。

5.1.1.2 逻辑通道

本条规定了在行业间类别中命令-响应对能查阅逻辑通道的机制。

如果卡支持该机制,则应在历史字节(见 8.1.1)或 EF.ATR(见 8.2.1.1)中指明可用逻辑通道数的最大值(见表 88,第 3 软件功能表)。

——如果指明的通道数小于或等于 4,则仅用表 2。

——如果指明的通道数大于或等于 5,则另外还用到表 3。

为查阅行业间类别中的逻辑通道,规定了下列规则:

——CLA 编码命令-响应对的通道号;

——基本通道始终可用,即它不能被关闭。它的通道号为 0;

——不支持该机制(缺省值)的卡仅使用基本通道;

——在 CLA 编码尚未使用的通道号的情况下,可以通过 SELECT 命令来开放任何通道,或通过 MANAGE CHANNEL 命令的开放功能来开放任何其他通道;

——可以通过 MANAGE CHANNEL 命令的关闭功能来关闭任何通道,在关闭后,可以通过重用

使通道变得可用；

——在同一时刻，仅有1个通道可用。逻辑通道的使用并不消除对通过接口的交叉命令-响应对的禁令，即，响应APDU应该在发起另一个命令-响应对之前接收到(见5.1)；

——如果文件描述符字节(见表14)没有明确拒绝共享，则对相同的结构(见5.3)即，对同一DF，也可能是同一应用DF，也可能是同一EF，可以有多个通道被打开。

每一个逻辑通道都有它自己的安全状态(见5.4)。共享一个安全状态的方法不在本部分的范围之内。

5.1.2 指令字节

INS指明要操作的命令。根据GB/T 16649.3的规定，值'6X'和'9X'是无效的。

表4列出了目前已出版的GB/T 16649中规定的所有命令。

——表4.1以字母表顺序列出了命令名称。

——表4.2以数字顺序列出了命令名称。

表4.1 以字母表顺序排列的命令

命 令 名 称	INS	见
ACTIVATE FILE	'44'	第9部分
APPEND RECORD	'E2'	7.3.7
CHANGE REFERENCE DATA	'24'	7.5.7
CREATE FILE	'E0'	第9部分
DEACTIVATE FILE	'04'	第9部分
DELETE FILE	'E4'	第9部分
DISABLE VERIFICATION REQUIREMENT	'26'	7.5.9
ENABLE VERIFICATION REQUIREMENT	'28'	7.5.8
ENVELOPE	'C2', 'C3'	7.6.2
ERASE BINARY	'0E', '0F'	7.2.7
ERASE RECORD (S)	'0C'	7.3.8
EXTERNAL (/ MUTUAL) AUTHENTICATE	'82'	7.5.4
GENERAL AUTHENTICATE	'86', '87'	7.5.5
GENERATE ASYMMETRIC KEY PAIR	'46'	第8部分
GET CHALLENGE	'84'	7.5.3
GET DATA	'CA', 'CB'	7.4.2
GET RESPONSE	'C0'	7.6.1
INTERNAL AUTHENTICATE	'88'	7.5.2
MANAGE CHANNEL	'70'	7.1.2
MANAGE SECURITY ENVIRONMENT	'22'	7.5.11
PERFORM SCQL OPERATION	'10'	第7部分
PERFORM SECURITY OPERATION	'2A'	第8部分
PERFORM TRANSACTION OPERATION	'12'	第7部分
PERFORM USER OPERATION	'14'	第7部分
PUT DATA	'DA', 'DB'	7.4.3
READ BINARY	'B0', 'B1'	7.2.3
READ RECORD (S)	'B2', 'B3'	7.3.3
RESET RETRY COUNTER	'2C'	7.5.10
SEARCH BINARY	'A0', 'A1'	7.2.6
SEARCH RECORD	'A2'	7.3.7
SELECT	'A4'	7.1.1

表 4.1（续）

命 令 名 称	INS	见
TERMINATE CARD USAGE	'FE'	第 9 部分
TERMINATE DF	'E6'	第 9 部分
TERMINATE EF	'E8'	第 9 部分
UPDATE BINARY	'D6', 'D7'	7.2.5
UPDATE RECORD	'DC', 'DD'	7.3.5
VERIFY	'20', '21'	7.5.6
WRITE BINARY	'D0', 'D1'	7.2.4
WRITE RECORD	'D2'	7.3.4
——在行业间类别中，所有不在 GB/T 16649 中定义的有效 INS 代码由 ISO/IEC JTC1/SC17 保留供将来使用。		

表 4.2 以数字顺序排列的命令

INS	命 令 名 称	见
'04'	DEACTIVATE FILE	第 9 部分
'0C'	ERASE RECORD (S)	7.3.8
'0E', '0F'	ERASE BINARY	7.2.7
'10'	PERFORM SCQL OPERATION	第 7 部分
'12'	PERFORM TRANSACTION OPERATION	第 7 部分
'14'	PERFORM USER OPERATION	第 7 部分
'20', '21'	VERIFY	7.5.6
'22'	MANAGE SECURITY ENVIRONMENT	7.5.11
'24'	CHANGE REFERENCE DATA	7.5.7
'26'	DISABLE VERIFICATION REQUIREMENT	7.5.9
'28'	ENABLE VERIFICATION REQUIREMENT	7.5.8
'2A'	PERFORM SECURITY OPERATION	第 8 部分
'2C'	RESET RETRY COUNTER	7.5.10
'44'	ACTIVATE FILE	第 9 部分
'46'	GENERATE ASYMMETRIC KEY PAIR	第 8 部分
'70'	MANAGE CHANNEL	7.1.2
'82'	EXTERNAL (/ MUTUAL) AUTHENTICATE	7.5.4
'84'	GET CHALLENGE	7.5.3
'86', '87'	GENERAL AUTHENTICATE	7.5.5
'88'	INTERNAL AUTHENTICATE	7.5.2
'A0', 'A1'	SEARCH BINARY	7.2.6
'A2'	SEARCH RECORD	7.3.7
'A4'	SELECT	7.1.1
'B0', 'B1'	READ BINARY	7.2.3
'B2', 'B3'	READ RECORD (S)	7.3.3
'C0'	GET RESPONSE	7.6.1
'C2', 'C3'	ENVELOPE	7.6.2
'CA', 'CB'	GET DATA	7.4.2
'D0', 'D1'	WRITE BINARY	7.2.6
'D2'	WRITE RECORD	7.3.4
'D6', 'D7'	UPDATE BINARY	7.2.5

表 4.2（续）

INS	命 令 名 称	见
'DA', 'DB'	PUT DATA	7.4.3
'DC', 'DD'	UPDATE RECORD	7.3.5
'E0'	CREATE FILE	第 9 部分
'E2'	APPEND RECORD	7.3.6
'E4'	DELETE FILE	第 9 部分
'E6'	TERMINATE DF	第 9 部分
'E8'	TERMINATE EF	第 9 部分
'FE'	TERMINATE CARD USAGE	第 9 部分
——在行业间类别中，所有不在 GB/T 16649 中定义的有效 INS 代码由 ISO/IEC JTC1/SC17 保留供将来使用。		

GB/T 16649 规定了这些行业间类别的命令的使用。

——本部分（见第 7 章）规定了用于交换的命令。

——GB/T 16649.7 规定了用于结构化卡查询语言（SCQL）的命令。

——GB/T 16649.8 规定了用于安全操作的命令。

——GB/T 16649.9 规定了用于卡管理的命令。

在行业间类别中，INS 的位 1 指明数据字段格式，如下：

——如果位 1 置为 0（偶数 INS 代码），则不提供指示。

——如果位 1 置为 1（奇数 INS 代码），则应用如下的 BER-TLV 编码（见 5.2.2）：

- 在 SW1 不置为'61'的非链接命令中，如果有数据字段，则应该按 BER-TLV 编码；
- 命令链和（或）SW1 置为'61'的使用允许对单命令来说太长的数据串的传输。这样的处理会把数据字段中的数据对象分割开来并以一定的顺序在一个方向上连续地发送，即当发送时相反方向上没有数据字段。当链接命令和（或）使用 SW1 置为'61'时，所有相同方向和相同顺序的连续数据字段的串联应以 BER-TLV 编码。

5.1.3 状态字节

SW1-SW2 指示了处理状态。根据 GB/T 16649.3，所有不同于'6XXX'和'9XXX'的值都是无效的，此外，值'60XX'也是无效的。

值'61XX'、'62XX'、'63XX'、'64XX'、'65XX'、'66XX'、'68XX'、'69XX'、'6AXX'、'6CXX'是行业间的。根据 GB/T 16649.3，除了'6700'、'6B00'、'6D00'、'6E00'、'6F00'和'9000'是行业间的外，'67XX'、'6BXX'、'6DXX'、'6EXX' '6FXX'和'9XXX'都是私有的。

图 1 示出了用于 SW1-SW2 的值'9000'和'61XX'到'6FXX'的结构化图解。

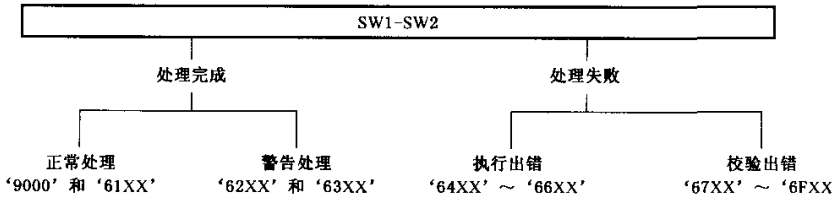

图 1 SW1-SW2 值的结构化图解

表 5 列出了所有 SW1-SW2 行业间值以及它们通常的含义。不在 GB/T 16649 中定义的 SW1-SW2 行业间值由 ISO/IEC JTC1/SC17 保留供将来使用。

表 5 SW1-SW2 行业间值的通常含义

	SW1-SW2	含 义
正常处理	'9000'	无进一步限定
	'61XX'	SW2 编码表示仍然可以获取的数据字节数(见下面文本)
警告处理	'62XX'	非易失存储器状态无变化(在 SW2 中进一步的限定)
	'63XX'	非易失存储器状态变化(在 SW2 中进一步的限定)
执行出错	'64XX'	非易失存储器状态无变化(在 SW2 中进一步的限定)
	'65XX'	非易失存储器状态变化(在 SW2 中进一步的限定)
	'66XX'	安全相关的发布
校验出错	'6700'	错误的长度;无进一步提示
	'68XX'	CLA 中的功能不被支持(在 SW2 中进一步的限定)
	'69XX'	不允许的命令(在 SW2 中进一步的限定)
	'6AXX'	错误的参数 P1-P2(在 SW2 中进一步的限定)
	'6B00'	错误的参数 P1-P2
	'6CXX'	错误的 Le 字段;SW2 编码准确的有效数据字节数(见下面的文本)
	'6D00'	指令代码不被支持或无效
	'6E00'	类别不被支持
	'6F00'	没有精确的诊断

如果处理失败返回 SW1 为'64'至'6F',则没有响应数据字段。

如果 SW1 置为'63'或'65',则表明非易失性存储器状态发生改变。如果 SW1 置为除了'63'和'65'之外的'6X'值,则表明非易失性存储器状态未发生改变。

在对非链(见 5.1.1.1)中结尾命令的响应中,禁止行业间警告提示(见 GB/T 16649.3),即 SW1 不能置为'62'或'63'。

两种 SW1 行业间值不依赖于传输协议:

——如果 SW1 置为'61',则处理完成,在发起其他命令之前,可以先调用与之有相同 CLA 且 SW2(数据字节数仍然有效)作为短 Le 字段的 GET RESPONSE 命令。

——如果 SW1 置为'6C',则处理失败,在发起其他命令之前,可以重新调用原有命令且以 SW2(确切的有效字节数)为短 Le 字段。

表 6 列出了 GB/T 16649 中使用的所有特定行业间警告和错误情形。

表 6 特定行业间警告和错误情形

SW1	SW2	含 义
'62'	'00'	没有信息被给出
(警告)	'02'到'08'	由卡触发(见 8.6.1)
	'81'	返回数据的一部分,数据可能被损坏
	'82'	读出 Ne 字节之前文件或记录已结束
	'83'	选择的文件无效
	'84'	FCI 未按照 5.3.3 格式化
	'85'	选择的文件为终止状态
	'86'	没有来自卡传感器的有效数据
'63'	'00'	没有信息被给出
(警告)	'81'	文件被上一次写入填满
	'CX'	通过'X'(值从 0 至 15)提供的计数器(正确的含义依赖于命令)

表 6（续）

SW1	SW2	含　　义
'64'	'00'	运行出错
（错误）	'01'	卡需要返回数据
	'02'至'80'	由卡控制触发（见 8.6.1）
'65'	'00'	没有信息被给出
（错误）	'81'	存储器故障
'68'	'00'	没有信息被给出
（错误）	'81'	逻辑通道不被支持
	'82'	安全报文不被支持
	'83'	无命令链结束
	'84'	命令链接不被支持
'69'	'00'	没有信息被给出
（错误）	'81'	命令与文件结构不兼容
	'82'	安全状态不被满足
	'83'	鉴别方法被阻塞
	'84'	引用的数据无效
	'85'	使用的条件不被满足
	'86'	命令不被允许（无当前 EF）
	'87'	期望的 SM 数据对象失踪
	'88'	SM 数据对象不正确
'6A'	'00'	没有信息被给出
（错误）	'80'	在数据字段中的不正确参数
	'81'	功能不被支持
	'82'	文件或应用未找到
	'83'	记录未找到
	'84'	无足够的文件存储空间
	'85'	Nc 与 TLV 结构不一致
	'86'	不正确的参数 P1-P2
	'87'	Nc 与 P1-P2 不一致
	'88'	引用的数据未找到（正确的含义依赖于命令）
	'89'	文件已存在
	'8A'	DF 名已存在
——其他所有 SW2 的值均被 ISO/IEC JTC1/SC17 定义为 RFU。		

5.2　数据对象

如果以 TLV 方式编码，那么任何数据字段或数据字段的串联就是一个数据对象的序列。本条定义了两类数据对象：SIMPLE-TLV 数据对象和 BER-TLV 数据对象。

5.2.1　SIMPLE-TLV 数据对象

每一个 SIMPLE-TLV 数据对象将由 2～3 个连续字段构成：一个必备的 tag 字段、一个必备的长度字段和一个条件可选值字段。一个记录（见 7.3.1）可能是一个 SIMPLE-TLV 数据对象。

——tag 字段由一个字节编码表示 tag 号从 1 到 254。'00'和'FF'为无效值。如果一个记录是一个 SIMPLE-TLV 数据对象，则其 tag 可用作记录标识符。

——长度字段由 1 到 3 个连续字节构成。

- 如果第 1 个字节不是置为'FF',则长度字段包含一个字节编码表示一个从 0 到 254 的数 N。
- 如果第 1 个字节置为'FF',则长度字段含 3 个字节,随后的 2 个字节编码表示从 0 到 65535 的数 N。

——如果 N 为 0,则没有值字段,即数据对象为空,否则(N>0),值字段包含 N 个连续字节。

5.2.2 BER-TLV 数据对象

每一个 BER-TLV 数据对象由 2～3 个连续字段构成(见 GB/T 16263.1 中的 ASN.1 基本编码原则):一个必备的 tag 字段、一个必备的长度字段和一个条件可选值字段。

——tag 字段由 1 个或者多个连续字节构成。它指明了类型、编码并编码 tag 数。tag 字段的第一个字节不能为'00'(见 GB/T 16263.1)。

——长度字段由 1 个或者多个连续字节构成。它是长度值的编码,即数 N。

——如果 N 为 0,则无值字段,即数据对象为空,否则(N>0),值字段包含 N 个连续字节。

5.2.2.1 BER-TLV tag 字段

GB/T 16649 支持的 tag 字段为 1、2,或 3 字节长,更长的 tag 字段为 RFU。

tag 字段头字节的 b8 和 b7 表明对象类型:

——b8b7=00:数据对象为通用类型;

——b8b7=01:数据对象为应用类型;

——b8b7=10:数据对象为特定上下文类型;

——b8b7=11:数据对象为私有类型。

tag 字段头字节的 b6 表明编码方式:

——b6=0:数据对象简单编码,即值字段不是以 BER-TLV 方式编码;

——b6=1:数据对象结构化编码,即值字段以 BER-TLV 方式编码。

如果 tag 字段头字节的 b5～b1 不是全置为 1,则其编码表示数从 0 至 30 且表明 tag 由单个字节构成。

否则(b5～b1 全置为 1),tag 字段是由一个或多个后续字节构成。

——每个后续字节的 b8 应置为 1,除非是最后一个字节;

——第 1 个后续字节的 b7～b1 位不能全部置为 0;

——第 1 个后续字节的 b7～b1 位,以及随后的每个后续字节(包括最后一个字节)的 b7～b1 位共同编码构成 tag 号。

表 7 描述了 tag 字段的第 1 个字节,其中'00'为无效值。

表 7 BER-TLV tag 字段的第一个字节

b8	b7	b6	b5	b4	b3	b2	b1	含义
0	0	—	—	—	—	—	—	通用类别,GB/T 16649 未定义
0	1	—	—	—	—	—	—	应用类别,由本规范定义
1	0	—	—	—	—	—	—	特定上下文类型,GB/T 16649 定义
1	1	—	—	—	—	—	—	私有类型,GB/T 16649 未定义
—	—	0	—	—	—	—	—	简单编码
—	—	1	—	—	—	—	—	结构化编码
—	—	—	不全置为 1					Tag 号由 0 至 30(短 tag 字段即 1 个字节)
—	—	—	1	1	1	1	1	Tag 号大于 30(长 tag 字段即 2 或 3 个字节)

以 BER-TLV 方式编码的数据字段中,置为'00'的字段可能出现在数据对象的前部、之间或者后部(例如,由于对 EF 支持数据单元进行数据对象的删除或修改)。该填充方式在 GB/T 16649 中称作模板且不允许出现于数据对象的值字段中。

无论是历史字节(见 8.1.1),或者 EF.ATR(见 8.2.1.1),或文件控制信息(见表 12 中 tag'82'),

数据编码字节(见表 87)都表明了值'FF'是:

——对私有类的长 tag 字段的第一个字节有效,结构化编码(外在特征),或者

——对 tag 字段(默认值)的第 1 个字节无效,即在同样目的(填充方式)和相同情形下作为值'00'。在 2 或多个字节的 tag 字段中,第 2 个字节不能为'00'至'1E'以及'80'。

——在 2 字节 tag 字段中,第 2 个字节的 b8b7 置为 01,编码表示一个大于 30 的数,第 2 个字节从'1F'至'7F',对应 tag 号从 31 到 127;

——在 3 字节 tag 字段中,第 2 个字节 b8b7 置为 11 且剩余位不全置为 0,第 3 个字节 b8b7 置为 01 且剩余位任意,则第 2 个字节从'81'至'FF',第 3 个字节从'00'至'7F',对应 tag 号从 128 到 16383。

5.2.2.2 BER-TLV 长度字段

在短格式中,长度字段由单个字节构成且该字节的 b8 置为 0,b7～b1 表示值字段的字节数。该字节因此可编码为从 0 到 127 的数。

注:在 BER-TLV 长度字段中任何从 1 到 127 的数的编码方式与 Lc 和 Le 字段相同。但 0,128 及大于 128 的数的编码方式有区别,具体见 7.4.2 中 GET DATA 命令数据对象编码。

在长格式中,长度字段由 2 个或更多字节构成。首字节的 b8 置为 1,b7～b1 不尽相同,其编码表示后续字节的长度字段。这些后续字节编码表示值字段的字节数。

GB/T 16649 不采用 ASN.1 基本编码规则中定义的"可变长度"。

GB/T 16649 支持长度字段由 1,2,直到 5 个字节构成(见表 8),在 GB/T 16649 中,值'80'及'85'至'FF'不允许出现在长度字段的第一个字节。

表 8 GB/T 16649 中的 BER-TLV 长度字段

	1st byte	2nd byte	3rd byte	4th byte	5th byte	N
1 byte	'00'～'7F'	—	—	—	—	0～127
2 bytes	'81'	'00'～'FF'	—	—	—	0～255
3 bytes	'82'	'0000'～'FFFF'		—	—	0～65 535
4 bytes	'83'	'000000'～'FFFFFF'			—	0～16 777 215
5 bytes	'84'	'00000000'～'FFFFFFFF'				0～4 294 967 295

5.2.3 数据字段、值字段、数据对象和数据元

所有的命令数据字段或响应数据字段均可按照 BER-TLV 方式编码,例如,在一个命令响应对中,CLA 指示为安全报文时(见 6)或 INS 的 b1 位设置为 1(INS 奇数编码,见 5.1.2)时,数据字段按照 BER-TLV 方式编码。

——任何 BER-TLV 数据对象表示为{T-L-V},由一个 tag 字段紧随一个长度字段构成。根据长度字段表示的数字是否为 0,决定值字段的有无。

——任何结构化 BER-TLV 数据对象表示为{T-L-{T1-L1-V1}...{Tn-Ln-Vn}},由一个 tag 字段紧随一个长度字段构成。如果长度字段表示的数字不为 0,则该结构化数据对象的值字段将由一个或多个 BER-TLV 数据对象构成(例如模板),每一个 BER-TLV 数据对象包含 tag 字段、长度字段和值字段(若长度字段编码不为 0)。

某些数据字段(例如,处理数据单元的命令,见 7.2),SIMPLE-TLV 数据对象的值字段和某些原始 BER-TLV 数据对象的值字段由符合命令规范或数据对象标记的数据元组成。

某些其他数据字段(例如,处理记录的命令,见 7.3),某些原始 BER-TLV 数据对象的值字段由 SIMPLE-TLV 数据对象组成。

还有某些数据字段(例如,处理数据对象的命令,见 7.4),结构化 BER-TLV 数据对象的值字段,即模板,由 BER-TLV 数据对象组成。

5.2.4 数据元识别

数据元的识别遵照以下原则：

1) 如果表示数据元的 bit 数不是 8 的倍数，那么应在数据元对应上下文中规定如何映射为字节或字节串。除另有规定，最后一个字节从 bit 1 开始的相应位都应置为 1；
2) 在卡与接口设备之间，数据元一般出现于 BER-TLV 数据对象的值字段中；
3) 为了达到命令交换中取数据和引用的目的，数据元应与 BER-TLV 数据对象 tag 关联且可被嵌入在该数据对象中；
4) 数据元可被相关联的 BER-TLV tag 直接引用，也可以被属于同一上下文中的其他数据元关联；
5) 一个或多个 command-to-perform 数据对象可间接引用数据元；
6) 通用类(第 1 个字节为'01'至'3F')数据对象具有其通用含义；
7) 所有应用类(第 1 个字节为'40'至'7F')数据对象除特殊规定外均为行业间的，GB/T 16649 本部分和其他部分分配应用类的 tag。不在 GB/T 16649 中定义的所有应用类 tag 由 ISO/IEC JTC 1/SC17 保留；
8) 本部分定义了部分行业间数据元，此外，随着应用的需要，还会有更多的行业间数据元被定义，GB/T 16649.6 列出了详尽的已经在 GB/T 16649 中被规定的行业间数据元列表；
9) 卡中可能重复出现相同的行业间数据对象；
10) 在命令和响应数据字段中，除了 FCI(见 5.3.3)和安全报文传输(见第 6 章)，所有特定上下文类数据对象(首字节从'80'到'BF')均应在行业间模板中嵌套；
11) 附录 A 中列出了后续章节规定的数据字段中行业间数据对象的 tag 分配方案。当需要时，这些 tag 分配方案使用表 9 列出的 tag 分配授权的行业间数据对象。

表 9 tag 分配授权的行业间数据对象

tag	值
'06'	对象标识符(编码说明见 GB/T 16263.1，例子见附录 A)
'41'	国家代码(ISO 3166-1[1]中规定的编码)和任选的国家数据
'42'	发行者标识号(编码和注册在 GB/T 15694.1 中规定)和任选的发行者数据
'4F'	应用标识符(AID，编码规定见 8.2.1.2)

5.2.4.1 一致性的 tag 分配方案

tag 分配方案使用行业间数据对象和其他的数据对象。

这些其他的数据对象在 tag 为'70'至'77'('73'除外，见 5.2.4.3)行业间模板内实现嵌套。此间除 tag'41'，'42'和'4F'被分配授权外，其他应用类别 tag 的意义在 GB/T 16649 中未作定义。

特殊上下文类(首字节从'80'到'BF')行业间模板 tag 不允许为'65'(持卡人相关数据)、'66'(卡数据)、'67'(授权数据)和'6E'(应用相关数据)。

为保证 tag 位分配方案一致和对应授权的合理，可以使用 tag 为'78'的行业间模板，该模板应包含一个表 9 中所示的某种数据对象用于确定 tag 的分配授权。

——如果初始数据串(见 8.1.2)或 EF.ATR(见 8.2.1.1)中包含 tag'78'，则该授权对整个卡有效；

——如果 tag'78'在 DF 管理数据(见 5.3.3)中，则该授权对 DF 有效。

5.2.4.2 共存的 tag 分配方案

此类 tag 分配方案可能用到 GB/T 16649 中未定义的带有解释的 tag。为保证共存的 tag 分配方案一致和对应授权的合理，可以使用 tag 为'79'的行业间模板，该模板将包含一个表 9 中所示的某种数据对象。

——如果一个授权对整个卡有效，则初始数据串（见 8.1.2）或 EF.ATR（见 8.2.1.1）中包含 tag‘79’；

——如果一个授权对 DF 有效，则‘79’在 DF 管理数据（见 5.3.3）中出现。

该方案中，所有的行业间数据对象在 tag 为‘7E’的行业间模板内实现嵌套。如同 tag‘62’、‘64’、‘6F’（FCP，FMD 和 FCI 模板，见 5.3.3）和‘7D’（SM 模板，见 6）一样，tag‘79’和‘7E’不再用作其他。

5.2.4.3 独立的 tag 分配方案

这些 tag 的分配办法，可能采用另一种不同于 GB/T 16649 的解释，但它并不符合 5.2.4.2，这种 tag 分配方法不能用于交换，也与本文档不符。

tag‘53’用以表示使用行业间数据对象的任意数据元，而‘73’实现私有数据对象在任意模板的嵌套使用，从而保证私有数据元和数据对象的使用仍符合本文档。

5.3 应用与数据的结构

本条规定当处理交换用的行业间命令时在接口处所看到的关于数据和应用结构的信息。超出本条概述之外的数据和结构信息的实际存储位置不在 GB/T 16649 范围内。

本部分支持下列两种文件：专用文件（DF）和基本文件（EF）。

——DF 用于支持应用、文件夹和数据对象存储。一个应用 DF 对应一种应用。DF 可以作为其他文件的父文件。这些文件被称为该 DF 的直属文件。

——EF 用于存放数据。EF 文件不能作为其他文件的父文件。EF 分为两类：

- 内部 EF，用于存储由卡所解释的数据，即，为了管理和控制目的由卡所分析和使用的数据。
- 工作的 EF，用于存储不由卡所解释的数据，即，仅仅由外界待使用的数据。

本部分提供两种逻辑组织方式：

——图 2 例示了包含对应安全架构的 DF 层次结构（见 5.4）。在这种卡的组织结构中，处于根部的 DF 称之为主文件（MF）。所有 DF 可以是应用 DF，也可以有其 DF 子结构。

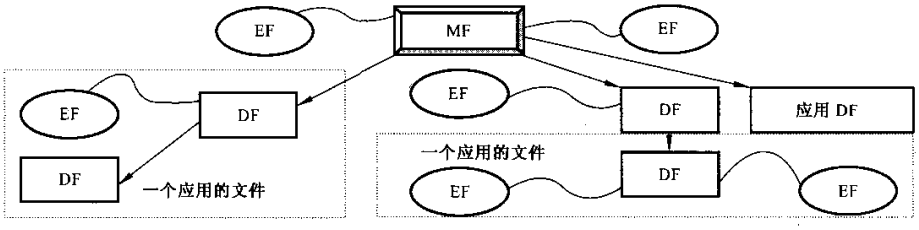

图 2 DF 层次结构例示图

——图 3 例示了平行结构的应用 DF，且接口处没有 MF，也没有任何可见 DF 子结构。该组织结构支持卡内的独立应用，在这些独立应用中应用 DF 可以包含其 DF 层次和对应安全结构。

图 3 独立应用 DF 例示图

5.3.1 结构选择

5.3.1.1 结构选择方法

选择了一个结构，则可以访问其数据，如果是 DF 结构，则可以访问其子结构。结构选择可以是隐式实现，即，复位和可能的协议和参数的选择后自动进行（见 GB/T 16649.3）。如果一个结构不能被隐式选择，则应进行显式选择，如利用以下 4 种方式之一。

通过 DF 名称选择——任何 DF 都可以通过按 1 至 16 个字节编码的 DF 名称来引用。任何应用标识符（AID，见 8.2.1.2）均可作为 DF 名称。为了通过 DF 名称进行无二义性的选择，例如通过应用标识符 AID 选择，每个 DF 名称应在给定的卡内是唯一的。

通过文件标识符选择——任何文件都可以通过按2字节编码的文件标识符来引用。如果MF通过文件标识符来引用,应使用'3F00'(保留值)。值'FFFF'被保留供将来使用。值'3FFF'被保留(见本条下文和7.4.1)。值'0000'被保留(见7.2.2和7.4.1)。为了通过文件标识符来无二义性地选择任何文件,在给定DF下的所有直接EF和DF都应具有不同的文件标识符。

通过路径选择——任何文件都可以通过路径来引用(文件标识符的串联)。该路径以MF或当前DF的标识符开始,并且以文件自身的标识符结束。在这两个标识符之间,路径由连续父DF(如果有)的标识符组成。文件标识符的次序总是在父级至子级的方向上。如果当前DF的标识符未知,值'3FFF'(保留值)可以用于路径的开始处。值'3F002F00'和'3F002F01'被保留(见8.2.1.1)。路径允许从MF或当前DF中无二义性地选择任何文件(见8.3)。

通过短EF标识符选择——任何EF都可以通过值在从1至30范围内的5位编码的短EF标识符来引用。用作短EF标识符的值0(即二进制的00000)引用了当前选择的EF,在MF级,值30(即二进制的11110)被保留(见8.2.1.1)。短EF标识符不能用在路径中或不能作为文件标识符(例如,在SELECT命令中)。

如果支持,可以按短EF标识符进行选择:

——如果第一个软件函数表(见表86)于历史字节(见8.1.1)或EF.ATR(见8.2.1.1)中出现,则可在卡级有效;

——如果短EF标识符(tag'88',见表12)于EF的控制参数(见5.3.3)中出现,则在EF级有效。

5.3.1.2 文件引用数据元

引用tag'51',表示行业间数据元为文件,该文件可以为任意长度。

——空数据对象引用MF;

——如果长度为1,该数据元b8至b4不全相等,且b3至b1置为000,b8至b4编码表示一个从1至30的数,则其为一个短EF标识符;

——如果长度为2,则该数据元为文件标识符;

——如果长度大于2,则该数据元为一条路径:

- 如果长度为偶数且头两个字节置为'3F00',则该路径为绝对路径。该数据元是由至少两个以MF标识符开始的文件标识符串联构成;
- 如果长度为偶数且头两个字节不是置为'3F00',则该路径为相对路径。该数据元是由至少两个以当前DF标识符开始的文件标识符串联构成;
- 如果长度为奇数,则该路径是受限制的。该数据元可以是无'3F00'的绝对路径,也可以是没有当前DF标识符的相对路径,跟随的一个字节被用于一条或多条SELECT命令(见7.1.1和8.3)中的P1。

表10示出了文件引用数据对象。

表10 文件引用数据对象

tag	长度	值
'51'	0	空数据对象引用MF
	1	短文件标识符(b8至b4编码表示一个从1至30的数;b3至b1置为000)
	2	文件标识符
	偶数,>2	绝对路径(头两个字节置为'3F00')
		相对路径(头两个字节不置为'3F00')
	奇数,>2	受限制的(最后一个字节将在一条或多条SELECT命令中用作P1)

5.3.2 数据引用方法

在DF中,数据可能引用为数据对象(见5.2)。在接口处DF可被视作一个能被数据处理命令处理

的数据对象集合(见 7.4)。

在 EF 中,数据可能引用为数据单元(见 7.2.1)、记录(见 7.3.1)或数据对象(见 5.2)。数据引用方式依赖于 EF。定义了以下 3 种 EF 结构。

——透明结构—在接口处 EF 可被看作一数据单元(见 7.2)序列,该序列通过数据单元操作命令访问。数据单元大小与 EF 相关。

——记录结构—在接口处 EF 可被看作一可独立标识的记录序列,该序列通过处理记录的命令来访问(见 7.3)。记录编号方法与 EF 相关。为按记录构成的 EF 定义了下列两种属性。

- 记录的长度:固定的或可变的。
- 记录的组织结构:按顺序(线性结构)或者按环形(循环结构)。

——TLV 结构—在接口处 EF 可看作一个数据对象集合,该集合通过用于处理数据对象的命令来访问(见 7.4)。这些在 EF 中的数据对象是 SIMPLE-TLV 或 BER-TLV 的则与 EF 相关。

为引用 EF 数据,卡必须至少支持图 4 中 5 种结构中的一种。

1) 透明结构;

2) 线性定长记录结构;

3) 线性变长记录结构;

4) 循环定长记录结构(箭头引用最近写入的记录);

5) TLV 结构。

图 4 EF 结构

5.3.3 文件控制信息 FCI

根据定义,文件控制信息(FCI)是可用于响应 SELECT 命令(见 7.1.1)的数据字节串。对于任何结构,例如任何 DF 或任何 EF,文件控制信息都可以呈现。

——如果第一个字节的值字段为'00'至'BF',则该字节串为 BER-TLV 编码。ISO/IEC JTC1/SC17 保留所有的未在本文档中定义的'00'至'BF'之间的值用于将来使用;

——如果第 1 个字节的值字段为'C0'至'FF',则该字节串未按本文档编码。

表 11 示出了 3 种行业间模板用来嵌套文件控制信息 BER-TLV 数据对象。

——FCP 模板是文件控制参数(FCP)的集合,即,在表 12 中列出的和下面定义的逻辑属性、结构属性和安全属性。在 FCP 模板中,特定上下文类(首字节为'80'至'BF')由文件控制参数保留,tag'85'和'A5'引用任意数据。

——FMD 模板是文件管理数据(FMD)的集合,即行业间数据对象(例如,如 8.2.1.2 中定义的应用标识符、8.2.1.4 中定义的应用标号和 GB/T 16649.6 中定义的应用有效日期,还可能是嵌套在如 8.2.1.3 中定义的应用模板中)。在 FMD 模板中,tag'53'和'73'引用任意数据。

——FCI 模板是文件控制参数和文件管理数据的集合。

表 11 与 FCI 相关的行业间模板

tag	值
'62'	文件控制参数(FCP 模板)
'64'	文件管理数据(FMD 模板)
'6F'	文件控制参数和文件管理数据(FCI 模板)

3 种模板可以根据选择“SELECT 命令”中的选项(见表 40)进行检索。

——如果 FCI 被置位,则 FCI tag 对引入相应数据字段的模板可选。

——如果 FCP 或 FMD 被置位,则对应 tag 必须出现。

表 12 列出了所有特定上下文类中的文件控制参数。如果一个文件存在控制参数时,表中还指出了它出现一次(明确表示)或重复出现(无表示)。

表 12 文件控制参数

tag	长度	值	适用于
'80'	变量	在文件中的数据字节数,不包括结构信息	任何 EF,1 次
'81'	2	在文件中的数据字节数,如果有,包括结构信息	任何文件,1 次
'82'	1	文件描述符字节(见 5.3.3.3 和表 14)	任何文件
	2	文件描述符字节后面紧跟着数据编码字节(见表 87)	
	3 或 4	文件描述符字节后面紧跟着数据编码字节和 1 个或 2 个字节的最大记录长度	任何带有记录结构的 EF
	5 或 6	文件描述符字节后面紧跟着数据编码字节和 2 个字节的最大记录长度以及 1 个或 2 个字节的记录个数	
'83'	2	文件标识符	任何文件
'84'	1～16	DF 名称	任何 DF
'85'	变量	非 BER-TLV 编码的专有信息	任何文件
'86'	变量	以专有格式的安全属性	任何文件
'87'	2	包含扩充 FCI 的 EF 标识符	任何 DF,1 次
'88'	0 或 1	短 EF 标识符(见 5.3.3.1)	任何 EF,1 次
'8A'	1	生命周期状态字节(LCS 字节,见 5.3.3.2 和表 13)	任何文件,1 次
'8B'	变量	扩展格式安全属性(见 5.4.3.3 和表 15)	任何文件,1 次
'8C'	变量	压缩格式安全属性(见 5.4.3.1)	任何文件,1 次
'8D'	2	包含安全环境模板的 EF 标识符(见 6.3.4)	任何 DF
'8E'	1	通道安全属性(见 5.4.3 和表 15)	任何文件,1 次
'A0'	变量	数据对象模板安全属性(见 5.4.3)	任何文件,1 次
'A1'	变量	私有格式模板安全属性	任何文件
'A2'	变量	模板包含一或多对数据对象: 短 EF 标识符(tag'88')-文件参考(tag'51',L>2,见 5.3.1.2)	任何 DF
'A5'	变量	BER-TLV 编码的私有信息	任何文件
'AB'	变量	扩展格式模板安全属性(见 5.4.3.2)	任何文件,1 次
'AC'	变量	加密机制标识模板(见 5.4.2)	任何 DF
——其他所有首字节为'80'至'BF'的特定上下文类型数据对象均保留。			

DF 的部分控制信息可以存储在某个应用控制下的 EF 文件中,在文件控制参数中通过 tag'87'引用,如果出现此类 EF,则 FCI 必须以 FCP tag 或 FCI tag 引入。

5.3.3.1 短 EF 标识符

以下规则定义了任意 EF 控制参数中 tag'88'的使用。

——如果卡支持短 EF 标识符选择(见 5.3.1.1)且无 tag'88',则标识符(tag 为'83')的第 2 个字节 b5～b1 编码表示短 EF 标识符。

——如果有 tag 位'88'且长度置为 0,则 EF 支持无短标识符。

——如果有 tag 位'88'且长度置为 1,如果数据元 b8～b4 不全相等且 b3～b1 置为 000,则 b8～b4 编码表示短 EF 标识符(1～30)。

5.3.3.2 生命周期状态字节

无论是卡,文件,还是其他对象,均有其生命周期。此生命周期状态可以帮助卡和接口设备区分在卡、文件和其他卡内数据对象使用时的不同逻辑安全状态。

为实现将生命周期作为一种属性(见 GB/T 16649.9)进行灵活管理,此处定义了 4 种原始的生命周期状态:

1) 创建状态;
2) 初始化状态;
3) 操作状态;
4) 终止状态。

该生命周期状态字节(LCS 字节)对应含义如表 13 所示:

——值'00'至'0F'为行业的;

——值'10'至'FF'为私有的。

表 13 生命周期状态字节

b8	b7	b6	b5	b4	b3	b2	b1	含义
0	0	0	0	0	0	0	0	没有信息给出
0	0	0	0	0	0	0	1	创建状态
0	0	0	0	0	0	1	1	初始化状态
0	0	0	0	0	1	—	1	操作状态(激活)
0	0	0	0	0	1	—	0	操作状态(停活)
0	0	0	0	1	1	—	—	终止状态
不全为零				×	×	×	×	专有的
——其他值由 ISO/IEC JTC1/SC17 保留供将来使用。								

以 tag'8A'引用,文件 LCS 字节可以出现在任意文件的控制参数中(见表 12)。

卡 LCS 字节可能在历史字节(见 8.1.1.3)中出现。以 tag'48'引用,卡 LCS 字节可能出现在 EF.ATR(见 8.2.1.1)中。如果有 MF,则卡至少处于创建状态。

注:除非特别规定,安全属性对操作状态有效。

5.3.3.3 文件描述符字节

以 tag'82'引用,任意文件的控制参数中可以包含数据元(见表 12)。

——数据元的第 1 个字节为文件描述字节(见表 14);

——如果数据元包含 2 个或更多字节,则第 2 个字节为数据编码字节(见表 87)。如果卡在若干位置设置了数据编码字节,则给定文件有效性在从 MF 至该文件的路径中最近点指示。

表 14 文件描述符字节

b8	b7	b6	b5	b4	b3	b2	b1	含义
0	x	—	—	—	—	—	—	文件可访问性
0	0	—	—	—	—	—	—	——不可共享的文件
0	1	—	—	—	—	—	—	——可共享的文件
0	—	1	1	1	0	0	0	DF

表 14（续）

b8	b7	b6 b5 b4	b3 b2 b1	含　义
0	—	不全置 1	— — —	EF 类型
0	—	0 0 0	— — —	——工作的 EF
0	—	0 0 1	— — —	——内部的 EF
0	—	其他所有值	— — —	——EF 专用类型使用
0	—			EF 结构
0	—	不全置 1	0 0 0	——没有信息被给出
0	—	不全置 1	0 0 1	——透明结构
0	—	不全置 1	0 1 0	——线性结构，固定长度，没有进一步的信息
0	—	不全置 1	0 1 1	——线性结构，固定长度，TLV 结构
0	—	不全置 1	1 0 0	——线性结构，可变长度，没有进一步的信息
0	—	不全置 1	1 0 1	——线性结构，可变长度，TLV 结构
0	—	不全置 1	1 1 0	——循环结构，固定长度，没有进一步的信息
0	—	不全置 1	1 1 1	——循环结构，固定长度，TLV 结构
0	—	1 1 1	0 0 1	——TLV 结构，用于 BER-TLV 数据对象
0	—	1 1 1	0 1 0	——TLV 结构，用于 SIMPLE-TLV 数据对象
——其他所有值均为 RFU。 ——“可共享”意味着至少支持在不同逻辑通道上的当前访问。				

5.4　安全体系结构

5.4.1　概述

本章描述安全状态、安全属性和安全机制。

安全状态

安全状态表示完成下列动作后可能获得的当前状态：

——复位应答（ATR）和可能的协议参数选择（PPS）和/或

——单个命令或一序列命令，可能执行的鉴别过程。

安全状态也可以从完成与所包含实体（如果有）的标识有关的安全规程中产生，例如，

——通过验证口令（例如，使用一个“VERIFY”命令）；或

——通过认证密钥（例如，使用“GET CHALLENGE”命令后面紧跟着“EXTERNAL AUTHENTICATE”命令，或使用 GENERAL AUTHENTICATE 命令序列）；或

——通过安全报文传输（例如，报文鉴别）。

考虑了 4 种安全状态：

——全局安全状态——在卡中使用 DF 层次，它可以通过完成与 MF 相关的鉴别规程进行修改（例如，附属于 MF 的口令或密钥的实体鉴别）；

——应用特定安全状态——它可以通过完成与应用相关的鉴别规程进行修改（例如，附属于特定应用的口令或密钥的实体鉴别）；它可以通过应用选择进行维护、恢复或被丢失；这种修改只与鉴别规程所属的应用相关。如果应用了逻辑通道，则应用特定安全状态依赖于逻辑通道；

——文件特定安全状态——它可以通过完成与 DF 相关的鉴别规程进行修改（例如，附属于特定 DF 的口令或密钥的实体鉴别）；它可以通过文件选择进行维护、恢复或被丢失；这种修改只与鉴别规程所属的应用相关。如果应用了逻辑通道，则文件特定安全状态依赖于逻辑通道；

——命令特定安全状态——仅在执行涉及使用安全报文传输的命令期间，它才存在；这种命令可以

保留未变化的其他安全状态。

安全属性

当安全属性存在时，它定义了允许的动作以及所处于的条件。文件的安全属性依赖于：它的分类(DF 或 EF)、在它的文件控制信息中的和/或在其父文件的文件控制信息中的任选参数。安全属性也可以与命令、数据对象及表格和视图联合。特别地，安全属性可以

——在数据访问之前强制性指定卡的安全状态；

——如果卡处于某种特定状态则限制某些函数对数据的访问(如 read only)；

——定义某些安全功能必须获取特定安全状态方能执行。

安全机制

本部分定义了下列安全机制：

——使用口令的实体鉴别：卡对从外界接收到的数据同保密的内部数据进行比较。该机制可以用来保护用户的权利。

——使用密钥的实体鉴别：待鉴别的实体必须按鉴别规程(例如，使用“GET CHALLENGE”命令后面紧跟着“EXTERNAL AUTHENTICATE”命令、GENERAL AUTHENTICATE 命令序列)来证明了解的相关密钥。

——数据鉴别：使用保密的或公开的内部数据，卡校验从外界接收到的冗余数据。另一种方法是使用保密的内部数据，卡计算数据元(密码的校验和或者数字签名)，并且将其插入发送给外界的数据中。该机制可以用来保护提供者的权利。

——数据加密：使用保密的内部数据，卡解密在数据字段中接收到的密文。另一种方法是，使用秘密的或公开的内部数据，卡计算密码，并将其插入数据字段中，尽可能与其他数据一起进行。该机制可以用来提供保密性服务，例如，用于密钥管理和有条件的访问。除了密码机制外，数据保密性可以通过数据伪装来获得。在此情况下，卡计算伪装字节串，并通过“异或”运算将其加到从外界接收到的数据字节中，或将其加到发送给外界的数据字节中。该机制可以用来保护秘密，并且减少报文过滤的可能性。

鉴别的结果可以按照应用的要求记录到内部 EF 中。

5.4.2 密码机制标识符模板

按 tag‘AC’引用，任意 DF(见表 12)的控制参数中可以包含一个或多个加密机制标识符模板。每种对应该 DF 及其子结构的一种密码机制应用。这样的模板必须包含 2 个或更多的数据对象。

——第 1 个数据对象应为密码机制引用，tag 为‘80’(见表 33)。

——第 2 个数据对象应为对象标识符，tag 为‘06’，如 GB/T 16263.1 中所定义。该标识对象应是由密码机制限定或标准注册，如 ISO 标准。

密码机制的例子有加密算法(如 ISO/IEC 18033[18]、消息认证码(如 ISO/IEC 9797[7])、认证协议(如 ISO/IEC 9798[8])、数字签名(如 ISO/IEC 9796[6] 或 14888[16])、已注册密码算法(如 ISO/IEC 9979[9])等等。

——如果还有后续数据对象，则以 tag‘06’使用前导的已有机制(即操作模式，如 ISO/IEC 10116[11]、哈希功能，如 ISO/IEC 10118[12])，或显示参数(依赖于前导的已有机制的 tag)。

具体例子(解释见附录 A)

{'AC'-'0B'-{'80'-'01'-'01'}-{'06'-'06'-'28818C710201'}}

该模板将局部引用‘01’和 ISO/IEC 18033-2[18] 中的第 1 个加密算法联合起来。

{'AC'-'11'-{'80'-'01'-'02'}-{'06'-'05'-'28CC460502'}-{'06'-'05'-'28CF060303'}}

第一个对象标识符指向 ISO/IEC 9798-5[8] 中第 2 种认证机制。第 2 个对象标识符指向 ISO/IEC 10118-3[12] 中第 3 个专用的散列函数。因此，该模板通过 SHA-1 将局部引用‘02’和 GQ2 联合起来。

5.4.3 安全属性

以 tag'86','8B','8C','8E','A0','A1'或'AB'引用,安全属性可以出现在任意文件(见表 12)的控制参数中。卡内任意对象(例如:命令、文件、数据对象、表/视图)可能与超过一个的安全属性和/或一个安全属性中包含的引用相关联。

以 tag'A0'引用,数据对象安全属性模板可以出现在任意文件的控制参数中。该模板是通过安全状态数据对象的串联(tag'86','8B','8C','8E','A0','A1','AB')和 tag 链数据对象(tag'5C',见 8.5.1)来显示某个文件内的相关数据对象。

以 tag'8E'引用,一种通道安全状态(至多一种)可以出现在任意文件(见表 12)和任意适合的安全环境(SE,见 6.3.3)的控制参数中。可以根据表 15 来解释。

——"不可共享":表示至多一个逻辑通道可用,通道物理技术受限;

——"安全":表示 SM 密钥(见 6)可用(例如,通过原有的鉴别来建立);

——"用户鉴别":表示用户应被鉴别(例如,成功的口令验证)。

表 15 通道安全属性

b8	b7	b6	b5	b4	b3	b2	b1	含义
0	0	0	0	0	—	—	1	不可共享
0	0	0	0	0	—	1	—	安全
0	0	0	0	0	1	—	—	用户鉴别
——其他值由 ISO/IEC JTC1/SC17 保留供将来使用。								

在 SCQL 环境下(见 GB/T 16649.7,结构化卡查询语言命令),安全属性可在 SCQL 操作中指定,例如 CREATE TABLE 和 CREATE VIEW 命令。如果是使用基于本章定义的安全属性,那么将在 SCQL 操作的安全属性参数中被'8B','8C'或'AB'取代。

格式——本章为对象绑定和安全属性定义了 2 种格式,基于位图的压缩格式和利用 TLV 列表管理扩展压缩格式的扩展格式。

5.4.3.1 压缩格式

在压缩格式中,访问规则由一个访问模式字节紧随一个或多个安全条件字节组成。对象访问控制受控于与相关对象绑定的访问规则。如果以'8C'(见表 12)标记的数据对象值字段中包含多个访问规则,则取"或(OR)"条件。

访问模式字节——从 b7 至 b1 的每一位,置 0 则表示无对应安全条件,置 1 则有。当 b8 置为 1,则 b7～b4 可能用于其他命令,如特定应用命令。

表 16～表 19 定义了 DF,EF,数据对象和表格/视图对应的访问模式字节。

表 16 DF 访问模式字节

b8	b7	b6	b5	b4	b3	b2	b1	含义
0	—	—	—	—	—	—	—	b7～b1 依照此表设置
1	—	—	—	—	—	—	—	b3～b1 依照此表设置(b7～b4 专有)
0	1	—	—	—	—	—	—	DELETE FILE (self)
0	—	1	—	—	—	—	—	TERMINATE CARD USAGE (MF), TERMINATE DF
0	—	—	1	—	—	—	—	ACTIVATE FILE
0	—	—	—	1	—	—	—	DEACTIVATE FILE
—	—	—	—	—	1	—	—	CREATE FILE (DF creation)
—	—	—	—	—	—	1	—	CREATE FILE (EF creation)
—	—	—	—	—	—	—	1	DELETE FILE (child)

表 17 EF 访问模式字节

b8	b7	b6	b5	b4	b3	b2	b1	含义
0	—	—	—	—	—	—	—	b7～b1 依照此表设置
1	—	—	—		—	—	—	b3～b1 依照此表设置(b7～b4 专有)
0	1	—	—	—	—	—		DELETE FILE
0	—	1	—	—	—	—	—	TERMINATE EF
0	—	—	1	—	—	—	—	ACTIVATE FILE
0	—	—		1	—	—	—	DEACTIVATE FILE
—	—	—		—	1	—	—	WRITE BINARY, WRITE RECORD, APPEND RECORD
—	—	—	—	—	—	1	—	UPDATE BINARY, UPDATE RECORD, ERASE BINARY, ERASE RECORD(S)
—	—	—	—	—	—	—	1	READ BINARY, READ RECORD(S), SEARCH BINARY, SEARCH RECORD

表 18 数据对象访问模式字节

b8	b7	b6	b5	b4	b3	b2	b1	含义
0	—	—	—	—	—	—	—	b7～b1 依照此表设置
1	—	—	—	—	—	—	—	b3～b1 依照此表设置(b7～b4 专有)
0	x	x	x	x	—	—	—	000(其他值保留供将来使用)
—	—	—	—	—	1	—	—	MANAGE SECURITY ENVIRONMENT
—	—	—	—	—	—	1	—	PUT DATA
—	—	—	—	—	—	—	1	GET DATA

表 19 表格/视图访问模式字节

b8	b7	b6	b5	b4	b3	b2	b1	含义
0	—	—	—	—	—	—	—	b7～b1 依照此表设置
1	—		—	—	—	—	—	b3～b1 依照此表设置(b7～b4 专有)
0	1	—	—	—	—	—	—	CREATE USER, DELETE USER
0	—	1	—	—	—	—	—	GRANT, REVOKE
0	—	—	1	—	—	—	—	CREATE TABLE, CREATE VIEW, CREATE DICTIONARY
0	—	—	—	1	—	—	—	DROP TABLE, DROP VIEW
—	—	—	—	—	1	—	—	INSERT
—	—	—	—	—	—	1	—	UPDATE, DELETE
—	—	—	—	—	—	—	1	FETCH

安全条件字节——每一个安全条件字节指定了访问规则对应的安全机制。表 20 示出了安全条件字节。

表 20 安全条件字节

b8	b7	b6	b5	b4	b3	b2	b1	含义
0	0	0	0	0	0	0	0	无条件
1	1	1	1	1	1	1	1	Never

表 20（续）

b8	b7	b6	b5	b4	b3	b2	b1	含义
—	—	—	—	0	0	0	0	不涉及安全环境
—	—	—	—	不全相等				安全环境标识符(1～14 字节,SEID 字节,见 6.3.4)
—	—	—	—	1	1	1	1	保留
0	—	—	—	—	—	—	—	至少一种条件
1	—	—	—	—	—	—	—	所有条件
—	1	—	—	—	—	—	—	安全报文
—	—	1	—	—	—	—	—	外部认证
—	—	—	1	—	—	—	—	用户认证(比如口令)

b8～b5 指明了所需的安全条件。如果不全相等,则 b4～b1 确定一种安全环境(见 6.3.4,SEID 字节从 1 到 14)和该环境中定义的安全机制(由 b7～b5 指定)以用于命令保护、外部证明和用户鉴定。

——如果 b8 置为 1,则必须满足 b7～b5 指定的所有条件;

——如果 b8 置为 0,则至少满足一种 b7～b5 指定的条件;

——如果 b7 置为 1,则安全环境的控制引用模板(见 6.3.1)由 b4～b1 指定,例如,一个 SEID 字节从 1 至 14,描述安全报文传输是否应用于对应命令的数据字段和应答数据字段(见表 35 使用限定字节)。

5.4.3.2 扩展格式

在扩展格式中,访问规则由一个访问模式数据对象紧随一个或多个安全条件数据对象组成。对象访问控制受控于与相关对象绑定的访问规则。以'AB'标记的模板可以出现在该规则下任意文件(见表 12)的控制参数中。

访问模式数据对象——一个访问模式数据对象包含一个访问模式字节(见表 16～表 19)或命令描述列表或一种私有状态机描述。后续的安全条件数据对象均与命令相关。表 21 示出了访问模式数据对象。

表 21 访问模式数据对象

tag	长度	值	含义
'80'	1	访问模式字节	见表 16～表 19
'81'至'8F'	变长	命令头描述	(部分)命令头列表(见表 22)
'9C'	变长		私有的状态机描述

如果 tag 从'81'至'8F',则访问模式数据元表示为命令头 CLA、INS、P1 和 P2 四字节值的可能组合的列表。根据 tag 的 b4～b1,该列表仅包含如表 22 所描述的值。可能会用集合来定义一个命令集,例如,tag'87'下的 INS P1 P2,INS P1 P2,...的值。

表 22 tag 为'81'至'8F'的访问模式数据对象

b8	b7	b6	b5	b4	b3	b2	b1	含义
1	0	0	0	x	x	x	x	命令描述包括:
1	0	0	0	1	—	—	—	——(CLA)即 CLA 的值
1	0	0	0	—	1	—	—	——(INS)即 INS 的值
1	0	0	0	—	—	1	—	——(P1)即 P1 的值
1	0	0	0	—	—	—	1	——(P2)即 P2 的值
——CLA 的值作为通道号编码为 0,表示描述与逻辑通道无关。 ——INS 的编码为偶数,表示描述与数据字段格式指示无关。								

安全条件数据对象——依据表23，安全条件数据对象定义了访问受特定访问模式数据对象保护的对象所需的安全操作。如果用作安全条件，由'A4'(AT)、'B4'(CCT)、'B6'(DST)或'B8'(CT)引用的控制引用模板(见6.3.1)包含使用限定数据对象(见表35)以指示安全动作。

表23 安全条件数据对象

tag	长度	值	含义
'90'	0	—	Always
'97'	0	—	Never
'9E'	1	安全条件字节	见表20
'A4'	变长	控制引用模板	外部或用户认证依赖于使用限定符
'B4''B6''B8'	变长	控制引用模板	命令SM和响应依赖于使用限定符
'A0'	变长	安全条件数据对象	至少满足一种安全条件(OR模板)
'A7'	变长	安全条件数据对象	安全条件倒置(NOT模板)
'AF'	变长	安全条件数据对象	满足所有安全条件(AND模板)

一些安全条件数据对象可能为同一操作控制：

——如果安全条件数据对象按OR模板嵌套(tag'A0')，则动作之前必须满足至少一个安全条件；

——如果安全条件数据对象不按OR模板嵌套(tag'A0')或按AND模板嵌套(tag'AF')，则动作之前必须满足所有安全条件；

——如果安全条件数据对象按NOT模板嵌套(tag'A7')，则在安全条件不满足之前，其值已为真。

5.4.3.3 访问规则引用

扩展格式的访问规则可以保存在EF中(支持线性变长记录结构)。该EF被命名为EF.ARR。一个或多个访问规则可以按记录号保存在每一个记录中，该记录号被命名为ARR字节。表24给出了EF.ARR的布局。

表24 ARR布局

记录号(ARR字节)	记录内容(1个或多个访问规则)
1	访问模式数据对象，一个或多个安全条件数据对象，访问模式数据对象，……
2	访问模式数据对象，一个或多个安全条件数据对象，……

按tag'8B'引用，扩展格式(见表25)下安全属性数据对象可以出现在任意文件(见表12)的控制参数中。

——如果长度为1，则值字段为一个ARR字节，对应一个隐含的EF.ARR记录；

——如果长度为3，则值字段为一个文件标识符紧随一个ARR字节，该文件标识符引用EF.ARR且ARR字节即EF.ARR中的记录号；

——如果长度为偶数且至少为4，则值字段为文件标识符紧随一个或多个字节对，每一对包含一个SEID字节紧随一个ARR字节，该SEID字节标识由ARR字节引用的访问规则应用的安全环境。

表25 安全属性数据对象应用扩展格式

tag	长度	值
'8B'	1	ARR字节(1个字节)
	3	文件标识符(2个字节)-ARR字节(1个字节)
	偶数，>3	文件标识符(2个字节)-SEID字节(1个字节)-ARR字节(1个字节)-[SEID字节-ARR字节]-……

当前 SE 的 ARR 字节指明访问规则对应用 DF 的当前访问有效。

注：如果未在早期的 MANAGE SECURITY ENVIRONMENT 命令中设置 SE，则默认 SE 即为当前 SE。

5.4.4 安全支持数据元

本条定义了安全支持数据元的集合，该数据元中包含管理数据操作方式的规则。安全支持数据元扩展和细化了控制引用数据对象。卡将把它们作为应用所需安全机制的基本支持。应用可能引用它们作为安全报文和安全操作(见 GB/T 16649.8)。本条既不描述安全支持数据元的特征，如长度，也不定义能修改它们的值的算法。

原则——卡必须按如下原则维护和使用安全支持数据元的值：

——更新的值将由卡计算得出或者由外部提供，且对于特定类型的安全支持数据元将满足特定规则；

——更新必须在导致更新的命令进行输出之前进行。更新将不依赖于命令的未完成状态。如果在可能导致更新的操作中，须先对数据元的值进行更新，然后用于应用；

——特定应用的功能执行对特定应用安全支持数据元的访问是受限的。

注：命令-响应对中实际的安全实现是依赖于应用特定的算法和协议的，而卡仅仅提供数据元和对应使用规则的支持。

数据元——卡用以支持命令-响应对安全的数据元称为级数值。级数值在卡的生命周期中随特定事件而增加，每次卡激活对应级数值便不同。定义以下两种级数值——卡会话计数器和会话标识符。

——卡会话计数器在每次卡激活时计数(增加)一次；

——会话标识符从卡会话计数器和外部提供数据计算得来。

定义以下两类级数值：

——内部级数值：对于某种应用，用来记录特定事件发生的次数，发生该事件数据元便递增一次，且卡对该数据元提供专门的重置归 0 功能。内部级数值不可被外部控制，类似于卡内实时安全标记。内部级数值可用于密码计算。

——外部级数值：对于某种应用，将只能被外部数据更新。且待更新的数值必须比当前卡内所存的值大。

引用——卡按以下方式提供对安全支持数据元的访问。

——EF 可以出现于 MF(如卡会话计数器)或应用 DF(如特定应用级数值)中；

——补充数据对象(tag 为'88'，'92'，'93'，见表 33)可以出现于控制引用模板。如果 SE 明确使用这些数据元，那么这些 tag 可用；

——在以 tag'7A'引用的行业间模板中，特定上下文类别(首字节从'80'到'BF')用于安全支持数据对象(见表 26)。

表 26 安全支持数据对象

tag	值
'7A'	带下面 tag 的支持安全的数据对象集
'80'	卡会话计数器
'81'	卡会话标识符
'82'至'8E'	文件选择计数器
'93'	数字签名计数器
'9F2X'	内部进度值('X'为指定索引，例如引用文件选择计数器的索引)
'9F3Y'	外部进度值('Y'为指定索引，例如引用外部时戳的索引)
——本文中，ISO/IEC JTC1/SC17 保留任何其他的特定上下文类的数据对象(首字节从'80'到'BF')。	

6 安全报文

安全报文(Secure messaging)通过确保数据秘密性和数据鉴别这两个基本安全功能，来保护全部或部

分命令-响应对或拼接起来的连续的数据字段(命令链,见5.1.1.1,SW1设置为'61')。安全报文通过应用一个或多个安全机制来实现。也可通过一种DF(见5.3.3)控制参数中的密码机制标识符模板(参考5.4.2)来识别。每种安全机制包括密码算法,操作模式,密钥,参数(输入数据),还经常包含初始化数据。

——数据字段的传送和接收可与安全机制的处理交替进行。本规范不排除通过系列的分析处理余下的数据字段部分所用的机制和安全选项的决定;

——两个或多个安全机制可能会使用相同的密码算法,尽管会有不同的操作模式。因此而产生的补充规则并不排除本特性。

6.1 SM字段和SM数据对象

根据定义,SM格式以及SM模板(tag'7D')中的任何命令或响应数据字段都是SM字段。每一个SM字段被编码为BER-TLV(见5.2.2),其中的特定的上下文类(首字节从'80'到'BF')保留作SM数据对象。在命令-响应对中,SM格式可被默认地选择,如在发出命令前获知,或者明确地选择,如在CLA中指示(见5.1.1)。

注:命令链和/或设置SW1为'61'的方式引出一命令序列,数据字段(以及相关的数据对象)可以被拆分为连续的更小的数据字段。这样的情况下,当使用SM格式时,在同一方向上的同一序列的拼接起来的所有的连续的数据字段就是一个SM数据字段。

表27示出了本文档中列出的在特定的上下文类的SM数据对象。一些SM数据对象(SM tag '82','83','B0','B1')为递归式的,如无格式值字段就是一个SM字段。

表27 SM数据对象

tag	值
'80','81'	未编码为BER-TLV的无格式值
'82','83'	密码(编码为BER-TLV的无格式值并包含SM数据对象,如SM字段)
'84','85'	密码(编码为BER-TLV的无格式值,但不包含SM数据对象)
'86','87'	填充内容指示字节,其后是密码(未编码为BER-TLV的无格式值)
'89'	命令头(CLA INS P1 P2,4字节)
'8E'	密码校验和(至少4字节)
'90','91'	哈希编码
'92','93'	证书(未编码为BER-TLV的数据)
'94','95'	安全环境标识符(SEID字节)
'96','97'	不安全的命令-响应对中编码Ne的1个或2个字节(可能为空)
'99'	处理状态(SW1-SW2,2字节,可能为空)
'9A','9B'	输入的用于数字签名计算的数据元(值字段时有符号的)
'9C','9D'	公钥
'9E'	数字签名
'A0','A1'	用于计算哈希编码的输入模板(模板本身也是哈希值)
'A2'	用于验证密码校验和的输入模板(包括模板本身)
'A4','A5'	用于认证控制引用模板(AT)
'A6','A7'	用于密钥协商的控制引用模板(KAT)
'A8'	用于验证数字签名的输入模板(模板是有符号的)
'AA','AB'	用于哈希编码的控制引用模板(HT)
'AC','AD'	用于数字签名计算的输入模板(连接的值字段是有符号的)
'AE','AF'	用于验证证书的输入模板(连接的值字段是有符号的)
'B0','B1'	编码为BER-TLV的无格式值并包含SM数据对象,如SM字段
'B2','B3'	编码为BER-TLV的无格式值,但不包含SM数据对象
'B4','B5'	用于密码校验和的控制引用模板(CCT)
'B6','B7'	用于数字签名的控制引用模板(DST)

表 27（续）

tag	值
'B8','B9'	用于保证秘密性的控制引用模板(CT)
'BA','BB'	响应描述符模板
'BC','BD'	用于数字签名计算的输入模板(模板是有符号的)
'BE'	用于验证证书的输入模板(模板被验证)
——在本上下文中,ISO/IEC JTC1/SC17 保留了特定上下文类的任何其他数据对象(第一字节从'80'到'BF')。	

在 SM 字段中，每个 SM 数据对象(特定上下文类)的 tag 字段(tag 奇偶)的最后一个字节的 bit 1 用于指示，该 SM 数据对象是否包括在用于认证(密码校验和，见 6.2.3.1，或者数字签名，见 6.2.3.2)的数据元的计算中(bit 1 为 1，奇 tag 数，表示包括；bit 1 为 0，偶 tag 数，表示不包括)。如果出现，那么其他类的数据对象(如行业间数据对象)也包括在计算中。如果该计算出现，该数据元为认证(SM tag '8E','9E')的某 SM 数据对象的值字段，出现在该 SM 字段的最后。

有两类 SM 数据对象：

——每个基本 SM 数据对象(见 6.2)携带一个无格式值，或者输入，或者一个安全机制的结果。

——每个附加 SM 数据对象(见 6.3)携带一个控制引用模板，或者一个安全环境标识符，或者一个响应描述符模板。

注：基本 SM 数据对象也用于控制安全操作(见 GB/T 16649.8)。附加 SM 数据对象也用于管理安全环境(见 7.5.11)。安全报文使用的全局安全方法与安全操作共享一些安全相关的方法，如安全中使用的原子方法。附录 B 为这 2 种方法之间的协同的示例。

6.2 基本 SM 数据对象

6.2.1 用于封装无格式值的 SM 数据对象

对于 SM 数据字段和非 BER-TLV 编码的数据，封装是必要的。对不包括 SM 的 BER-TLV 和数据对象，封装是可选择的。表 28 为封装了无格式值的 SM 数据对象。

表 28 封装无格式值的 SM 数据对象

tag	值
'B0','B1'	编码为 BER-TLV 的无格式值，并包含 SM 数据对象(如 SM 字段)
'B2','B3'	编码为 BER-TLV 的无格式值，不包含 SM 数据对象
'80','81'	没有编码为 BER-TLV 的无格式值
'89'	命令头(CLA INS P1 P2,4 字节)
'96','97'	不安全的命令-响应对中编码 Ne 的 1 个或 2 个字节(可能为空)
'99'	处理状态(SW1-SW2,两字节，可能为空)

6.2.2 用于保持秘密性的 SM 数据对象

表 29 显示了用于保持数据秘密性的 SM 数据对象。

表 29 用于保持数据秘密性的 SM 数据对象

tag	值
'82','83'	密文(编码为 BER-TLV 的无格式值，并包含 SM 数据对象，如 SM 字段)
'84','85'	密文(编码为 BER-TLV 的无格式值，不包含 SM 数据对象)
'86','87'	填充内容指示字节，其后是密码(没有编码为 BER-TLV 的无格式值)

用于保持秘密性的安全机制由某种操作模式的密码算法组成。当没有明确指出并且没有暗含选择某种机制时，应用默认的机制保持秘密性：

——对有填充指示的密码算法，默认机制是“电子密码本”模式的块密码。该分组密码可能包含填充字节。保持数据秘密性的填充可能影响到传输：密文（一个或多个分组）可能长于此无格式值。

——对无填充指示的密文，默认机制是流密码。在这种情况中，密码是数据字节串与同样长度的隐蔽串的异或。隐蔽的方式中数据没有填充，且通过同样的方式恢复得到数据字节串。

无格式值未编码为 BER-TLV 格式时需要指明有填充和/或内容。如果有填充但未指明，应用6.2.3.1中列出的规则。表 30 为填充—内容指示字节。

表 30 填充—内容指示字节

值	含 义
'00'	没有更多的意义
'01'	根据 6.2.3.1 指定的方式填充
'02'	无填充
'1X'	1～4 个加密信息而不是加密密钥的秘密密钥（'X'表示'0'到'F'的任意值） '11'表示第一密钥（如，付费电视系统中的“事件”控制字） '12'表示第二密钥（如，付费电视系统中的“奇数”控制字） '13'表示接连的第一、二密钥（如，付费电视系统中一对控制字）
'2X'	用于加密密钥而不是信息的私钥（'X'表示'0'到'F'的任意值） （如，在付费电视系统中，加密控制字的操作密钥和加密操作密钥的控制密钥）
'3X'	非对称密钥对中的私钥（'X'表示'0'到'F'的任意值）
'4X'	口令（'X'表示'0'到'F'的任意值）
'80'到'8E'	私有
——根据 ISO/IEC JTC1/SC17，其他值保留为将来使用。	

6.2.3 用于认证的 SM 数据对象

表 31 为用于认证的 SM 数据对象。

表 31 用于认证的数据对象

tag	值
'8E'	密码校验和（至少 4 字节）
'90','91'	哈希编码
'92','93'	证书（非 BER-TLV 格式的数据）
'9C','9D'	公钥
'9E'	数字签名
输入数据对象（另见 GB/T 16649.8）	
'9A','9B'	用于计算数字签名的输入模板值字段是有符号的）
'A0','A1'	用于计算哈希编码的输入模板（模板本身也被哈希）
'A2'	用于验证密码校验和的输入模板（包括模板本身）
'A8'	用于验证数字签名的输入模板（模板是有符号的）
'AC','AD'	用于数字签名计算的输入模板（连接的值字段是有符号的）
'AE','AF'	用于验证证书的输入模板（连接的值字段是有符号的）
'BC','BD'	用于计算数字签名的输入模板（模板是有符号的）
'BE'	用于验证证书的输入模板（模板是有符号的）

6.2.3.1 密码校验和数据元

计算密码校验和涉及到一个初始化检查块，密钥和加密算法（见 ISO/IEC 18033[18]）或哈希函数

(见 ISO/IEC 10118[12])。

计算的方法可以是系统规范中的一部分。或者使用密码机制标识符模板(见 5.4.2)标识一个确定的计算方法的标准。

除非特别申明,应该采用下列计算方法。在密钥的控制下,算法主要将 k 字节(通常为 8,16 或 20)的当前的输入块转换为同样大小的当前输出块。计算过程分以下几个阶段:

初始化阶段:初始化阶段设置初始化检查块为下列块之一:

——空块,如 k 字节设置为'00';

——链接块,如之前的计算所产生的结果,即对命令,为之前的命令中的最后的检查块,对响应,为之前响应的最后的检查块;

——提供的初始化值块,如外部提供的;

——在密钥的控制下将辅助数据转换为辅助块。如果辅助数据长度小于 k 字节,在之前加入 0 以达到块的长度。

持续阶段:为保护命令头(CLA INS P1 P2),可以对其进行封装(SM tag'89')。但如果 CLA 中的 bit 8～bit 6 设置为 000,且 bit 4、bit 3 设置为 11(见 5.1.1),第一个数据块由命令头(CLA INS P1 P2)组成,其后是一个设置为'80'的字节,并且 k-5 个字节设置为'00'。

密码校验和应包含任何有奇 tag 数的安全报文数据对象和任何第一字节不为'80'到'BF'的数据对象。这些数据对象应以相连数据块的形式包含在当前检查块中。依据以下原则划分数据块:

——相邻的数据对象之间的边界上的数据块应该是连续的。

——在每个包含的数据对象的后面增加填充,并且填充之后是一个未包含的数据对象,或者没有数据对象。填充至少由一个必须设置为'80'的字节组成,如果还有其他字节,将会是 0 到 k-1 个设置为'00'的字节,直到相应的数据块填至 k 字节。由于填充字节未被传送,填充将不会影响到认证。

这样的机制中,操作模式为"加密分组链"(见 ISO/IEC 10116[11])。第 1 个输入是检查块与第 1 个数据块的异或。第 1 个输出由第 1 个输入产生。当前输入是前一个输出与当前数据块的异或。当前输出由当前输入产生。

最终阶段:最后的检查块就是最后的输出。最终阶段从最后的检查块中提取密码校验和(前 m 字节,至少有 4 字节)。

6.2.3.2 数字签名的数据元

数字签名方案依赖于非对称密码技术(见 ISO/IEC 9796[6],14888[16])。计算过程中使用到了哈希函数(见 ISO/IEC 10118[12])。输入的数据由数字签名的输入数据对象的数据字段,或者某个数字签名的输入模板的多个数据字段的值字段拼接而成,可由 6.2.3.1 中指定的方法确定。

6.3 辅助的 SM 数据对象

表 32 为辅助的 SM 数据对象。

表 32 辅助的 SM 数据对象

tag	含义
'94','95'	安全环境标识符(SEID 字节)
'A4','A5'	认证中有效的控制引用模板(AT)
'A6','A7'	密钥协商中有效的控制引用模板(KAT)
'AA','AB'	哈希编码中有效的控制引用模板(HT)
'B4','B5'	密码校验和中有效的控制引用模板(CCT)
'B6','B7'	数字签名中有效的控制引用模板(DST)
'B8','B9'	保持秘密性中有效的控制引用模板(CT)
'BA','BB'	响应描述符模板

6.3.1 控制引用模板

已经定义6类控制引用模板，即认证中有效(AT)，密钥协商(KAT)，哈希编码(HT)，密码校验和(CCT)，数字签名(DST)和通过对称密钥(CT-sym)或非对称密钥技术(CT-asym)来保持秘密性(CT)。

每种安全机制包括一个处于某种操作模式的密码算法，使用一个密钥，并且可能会有初始化数据。可以隐式选择这些项，如发出命令前知道，或者显式的选择这些项，如通过嵌在控制引用模板中的控制引用数据对象。在控制引用模板中，保留特定上下文类(第1字节为'80'到'BF')作为控制引用数据对象。

SM字段中，控制引用模板的最后的可能位置就在应用引用机制的第1个数据对象之前。例如，密码校验和中有效的模板(CCT)的最后的可能位置就在计算中引入的第1个数据对象之前。

每一个控制引用对象在同类机制中的新的控制引用出现之前保留有效。例如，某条命令能为下一条命令提供控制引用。

6.3.2 控制引用模板中的控制引用数据对象

每个控制引用模板(CRT)是一个控制引用数据对象的集合：一个密码机制引用，一个文件和密钥引用，一个初始化数据引用，一个使用限定符和在保持秘密性的控制引用模板的内容加密引用。

——密码机制引用表示一个处于某种操作模式的加密算法。任何DF的控制参数(见表12中的tag'AC')可包含密码机制标识符模板(见5.4.2)。每个模板指出该密码机制引用的含义。

——文件引用(与5.3.1.2中的编码相同)表示在文件中的某处密钥引用有效。如果没有文件引用，密钥引用在当前DF(可能是一个应用的DF)中有效。密钥引用明确地识别使用的密钥。

——当应用于密码校验和时，初始化数据引用指出初始检查块。如果没有初始化数据引用且没有隐式地选择初始检查块则应用空块。并且在通过流密码传输第一个用于保持秘密性的数据对象前，保持秘密性的模板应该提供辅助数据以初始化秘密字节串的计算。

表33列出控制引用数据对象并指出相关的控制引用模板。所有的控制引用数据对象都在特定上下文的类中。

表33 控制引用模板中的控制引用数据对象

tag	值	AT	KAT	HT	CCT	DST	CT-asym	CT-sym
'80'	密码机制引用	×	×	×	×	×	×	×
文件和密钥引用								
'81'	——(与5.3.1.2中的编码相同)	×	×	×	×	×	×	×
'82'	——DF名(见5.3.1.1)	×	×	×	×	×	×	×
'83'	——秘密密钥引用(直接使用) ——公钥引用 ——引用数据限定符	× × ×	× ×	× ×	×	 ×	 ×	×
'84'	——计算会话密钥的引用 ——私钥引用	× ×	× ×		×	 ×	 ×	×
'A3'	——密钥使用模板(见下面的文本)	×	×	×	×	×	×	×
初始化数据引用：初始检查块								
'85'	——L=0，空块			×	×			×
'86'	——L=0，块链			×	×			×
'87'	——L=0，前一个初始值块加一 ——L=k，初始值块			 ×	× ×			×

表 33（续）

tag	值	AT	KAT	HT	CCT	DST	CT-asym	CT-sym
初始化数据引用：辅助数据元（另见 5.4.3）								
'88'	——L=0，前一个交换的挑战加一 ——L>0，无下一步指示				×	×	×	×
'89'到'8D'	——L=0，专有数据元索引 ——L>0，专有数据元值				× ×			×
'90'	——L=0，卡提供的哈希编码			×		×		
'91'	——L=0，卡提供的随机数 ——L>0，随机数		×		×	× ×	× ×	
'92'	——L=0，卡提供的时戳 ——L>0，时戳			×		× ×	× ×	
'93'	——L=0，前一个数字签名计数器加一 ——L>0，数字签名计数器			×		× ×	× ×	× ×
'94'	源于密钥的挑战或数据元	×			×			×
'95'	使用限定符字节（见下面文本）	×	×		×	×	×	×
'8E'	密码内容模板（见下面文本）						×	×
——本文中，ISO/IEC JTC1/SC17 保留任何其他特定上下文类的数据对象（首字节从'80'到'BF'）。 ——CRT 可能含行业间数据对象，如 AT 中的证书持有者的授权（tag'5F4C'，见 6.3.4），HT 或 DST 中的头列表或扩展的头列表（tag'5D'和'4D'，见 8.5.1）。								

在任何控制引用模板中，密钥使用模板（tag'A3'）可使一个文件和密钥引用与密钥使用计数器和/或密钥重试计数器关联（见表 34）。

表 34　密钥使用数据对象

tag	值
'A3'	带下面 tag 的密钥使用数据对象集
'80'至'84'	表 33 中列出的文件和密钥引用
'90'	密钥使用计数器
'91'	密钥重试计数器
——本文中，ISO/IEC JTC1/SC17 保留任何其他的特定上下文类的数据对象（第一字节从'80'到'BF'）。	

在任何用于认证的控制引用模板（AT）、用于密钥协商的控制引用模板（KAT）、用于密码校验和的控制引用模板（CCT）、用于保持秘密性的控制引用模板（CT）或用于数字签名的控制引用模板（DST）中，用法限定字节（tag'95'）可指定该模板作安全条件（见 5.4.3.2 和表 23）或，遵循 MANAGE SECURITY ENVIRONMENT 命令（见 7.5.11）。表 35 为用法限定字节。

表 35　用法限定字节

b8	b7	b6	b5	b4	b3	b2	b1	含　义
1	—	—	—	—	—	—	—	验证（DST，CCT），加密（CT），外部认证（AT），密钥协商（KAT）
—	1	—	—	—	—	—	—	计算（DST，CCT），解密（CT），内部认证（AT），密钥协商（KAT）
—	—	1	—	—	—	—	—	响应数据字段中的安全报文（CCT，CT，DST）

表 35（续）

b8	b7	b6	b5	b4	b3	b2	b1	含　义
—	—	—	1	—	—	—	—	命令数据字段中的安全报文(CCT,CT,DST)
—	—	—	—	1	—	—	—	基于密码的用户认证(AT)
—	—	—	—	—	1	—	—	基于生物特征的用户认证(AT)
—	—	—	—	—	—	x	x	xxxxxx00(任何其他值保留作未来使用)

在任何保持秘密性的控制引用模板中(CT),密码内容引用(tag'8E')可指定加密的内容。值字段的第一个字节是必备的,其名称是密码描述符字节。表 36 为密码描述符字节示意。

表 36　密码描述符字节

值	含　义
'00'	无更多的含义
'1X'	一到四个加密信息而不是加密密钥的秘密密钥('X'表示'0'到'F'的任意值) '11'表示第一密钥(如,付费电视系统中的"事件"控制字) '12'表示第二密钥(如,付费电视系统中的"奇数"控制字) '13'表示接连的第一、二密钥(如,付费电视系统中一对控制字)
'2X'	用于加密密钥而不是信息的秘密密钥('X'表示'0'到'F'的任意值) (如,在付费电视系统中,加密控制字的操作密钥和加密操作密钥的控制密钥)
'3X'	非对称钥对中的私钥('X'表示'0'到'F'的任意值)
'4X'	口令('X'表示'0'到'F'的任意值)
'80'至'FF'	专有
——根据 ISO/IEC JTC1/SC17,其他值保留为未来使用。	

6.3.3　安全环境

本条说明引用密码算法、操作模式、协议、程序、密钥与安全报文和安全操作(见 GB/T 16649.8)所需的任何其他的数据安全环境(SE)。SE 包含存储在卡中的数据元,或经指定的算法计算的结果数据元。SE 可包含初始化用于该环境的非持续的数据的机制,如,会话密钥。SE 可为处理计算结果提供指示,如卡中存储。行业间的 SE 模板(tag'7B')描述了一个 SE。

SE 标识符:SE 标识符(SEID 字节)可引用任何安全环境,如,安全报文、通过 MANAGE SECURITY ENVIRONMENT 命令(见 7.5.11)存储和转存。

——除非在应用中另外指定,'00'表示空环境,其中没有定义安全报文和认证。

——'FF'表示该环境中无操作可执行。

——除非在应用中另外指定,'01'为默认 SE 所保留,并一直可用。本条未指定默认 SE 的内容,可能为空。

——'EF'保留为将来使用。

组件:控制引用模板(CRT)可描述 SE 的各种组件。环境定义中的某个机制指定的任何相关的控制引用(文件,密钥或数据),应该根据相关的 DF 在该机制使用前确定下来。绝对控制引用(如,绝对路径)不需要确定。在 SE 中,组件可能有两种:一种在命令数据字段中的 SM 才有效,另一种在响应数据字段中的 SM 有效。

在任意的卡操作期间,应通过默认或作为卡执行命令的结果激活当前的 SE。当前 SE 包含下面的组件中的一个或多个:

——某些组件属于与当前 DF 关联的默认 SE。

——某些组件通过使用安全报文的命令传送。

——某些组件通过 MANAGE SECURITY ENVIRONMENT 命令传送。

——某些组件被 MANAGE SECURITY ENVIRONMENT 命令中的 SEID 字节调用。

当发生下列事件前，当前 SE 有效：直到有热复位或触点终止(见 GB/T 16649.3)，或上下文的改变(如，通过选择不公的应用 DF)，或设置 MANAGE SECURITY ENVIRONMENT 命令，或替换当前 SE。

在 SM 中，在 CRT 中传输的控制引用数据对象应优先于任何其他当前 SE 中出现的相应的控制引用数据对象。

证书持有者的认证：认证过程可使用卡可验证的证书，如，通过使用公开密钥(见 GB/T 16649.8)的 VERIFY CERTIFICATE 操作，卡能够解释和检验模板。在这样的证书中，证书持有者的认证(如，角色标识符)可转换为通过 tag'5F4C'引用的行业间数据元。如果这样的数据元在安全条件中使用，以实现访问数据或功能，这时该数据对象(tag'5F4C')应出现在用于认证的控制引用模板(AT)中，以描述认证的过程。

注：因此 GB/T 16649 不赞成使用 tag'5F4B'。

访问控制：卡可在包含行业间 SE 模板(tag'7B')的 EF 中(见表 12 中的 tag'8D')存储用于访问控制的安全环境。行业间 SE 模板中(tag'7B')，特定上下文类(第一字节'80'至'BF')为安全环境数据对象保留。如表 37 列出，每个包含的 SE，安全环境模板包含一个 SEID 字节数据对象(tag'80')，一个可选的 LCS 字节对象(tag'8A')，一个或多个可选加密机制标识符模板(tag'AC')和一个或多个 CRT (tag'A4','A6','AA','B4','B6','B8'，作为 SM tag)。

表 37 安全环境数据对象

tag	值
'7B'	带下面 tag 的安全环境数据对象的集合
'80'	SEID 字节，强制
'8A'	密码机制标识符模板(见 5.4.2)，可选
'A4','A6','AA','B4','B6','B8'	CRTs(见 6.3.1)
——本文中 ISO/IEC JTC1/SC17 保留特定上下文类的任何其他数据对象(第一字节'80'至'BF')。	

如果 LCS 字节数据对象在 SE 模板中出现，则它表明该 SE 在哪个生命周期阶段有效。如果 SE 用于访问控制，如对文件，则该文件的 LCS 字节和该 SE 的 LCS 字节必须匹配。如果无 LCS 字节，则该 SE 在激活的操作状态下有效。

在 SE 模板中，如果 CRT 携带一些有相同 tag 的数据对象(如，某个密钥引用指定的数据对象)，则至少要有一个数据对象存在(OR 条件)。

SE 检索：当前 SE 中的 CRT 可以通过 GET DATA 命令来检索，该命令的 P1-P2 设置为'004D'(扩展的头列表，见 8.5.1)、该命令的数据字段由 SE 模板(tag'7B')组成、该 SE 模板由一对或多对 CRT tag+'80'组成(见 8.5.1，扩展头列表中设置为'80'的长度的使用)。

6.3.4 响应描述符模板

每个命令数据字段可包含一个响应描述符模板。如果出现在命令数据字段中，响应描述符模板应指示响应数据字段中所需的 SM 数据对象。在响应描述符模板中，安全机制尚未应用；接收实体应使用安全机制来构造响应数据字段。用于处理命令数据字段的安全项(算法、操作模式、密钥和初始的数据)可与处理响应数据字段的安全项不同。依照下面的规则：

——卡填充每个空的原始基本 SM 数据对象。

——响应描述符模板中出现的每个 CRT 应该在响应中的相同位置与相同的用于安全机制、文件和密钥的控制引用数据对象一起出现。

- 如果响应描述符模板提供辅助的数据，则响应中的相应的数据对象应该为空。

- 如果辅助数据的空引用数据对象出现在响应描述符模板中，则应该在响应中填充。

——通过相关的安全机制和选择的安全项，卡应该处理所有请求的基本 SM 数据对象。

6.4 命令-响应对中 SM 的效果

图 5 为命令-响应对的示例。

命令头	命令体
[数据字段]	[Le字段][数据字段][Le字段]

响应体	响应尾
[数据字段]	SW1-SW2

图 5 命令-响应对

下面规则应用于保护行业间类(见 5.1.1)的命令-响应对，如 CLA 的 bit 8、bit 7 和 bit 6 设置为 000 时，bit 4 从 0 变为 1，或者 CLA 的 bit 8 和 bit 7 设置为 01 时，bit 6 从 0 变为 1。记号 CLA* 为 CLA 中指明了安全报文。

——安全的命令数据字段是一个 SM 字段，它有下面的形式：

- 如果有命令数据字段(Nc>0)，则无格式值数据对象(SM tag'80','81','B2','B3')或保持秘密性的数据对象(SM tag'84','85','86','87')应携带该 Nc 字节。
- 可以封装命令头(4 字节)用以保护(SM tag'89')命令头。
- 如果存在 Le 字段，应该出现一个新的 Le 字段(仅包含一个设置为'00'的字节)和一个新的 Le 数据对象(SM tag 为'96','97')。Le 数据对象为零和为空时表示最大值，如根据该新 Le 字段是短型或扩展的，最大值为 256 或 65 536。

——安全的响应数据字段也是一个 SM 字段，它应解释如下：

- 如果存在，无格式值数据对象(SM tag'80','81','B2','B3')或保持秘密性的数据对象(SM tag'84','85','86','87')携带该响应数据字节。
- 如果存在，处理状态数据对象(SM tag'99')携带为保护而封装的 SW1-SW2。空的处理状态数据对象意为 SW1-SW2 设置为'9000'。

图 6 展示了响应安全的命令-响应对。

命令头	命令体
CLA*INS P1 P2	[新Lc字段]-{[安全数据字段]=[T-Nc-数据字节]-[T-'01'或'02'-Le]}-[新Le字段]

响应体	响应尾
[安全数据字段]=[T-Nr-数据字节]-[T-'02'-SW1-SW2]	SW1-SW2

图 6 安全的命令-响应对

当 INS 的 bit 1 置 1(奇 INS 码，见 5.1.2)时，非安全的数据字段编码为 BER-TLV 格式，并且 SM tag'B2'、'B3'、'84'和'85'应该用于它们的封装。否则，要保护的数据字段的格式并不总是明确的，推荐使用 SM tag'80'、'81'、'86'和'87'。

——安全的命令数据字段是一个 SM 字段，它们可包含更多的或其他的 SM 数据对象，如密码校验和(SM tag'8E')或最后面的数字签名(SM tag'9E')。

——新的 Le 字段中对安全命令的数据字段的字节数编码。

——如果安全响应数据字段中无期望的数据字段，那么应该无新的 Le 字段；否则新的 Le 字段只包含设置为'00'的字节。

——响应尾指明处理该安全命令后接收实体的状态。下面特定的错误条件可能会出现：

- 如果 SW1-SW2 设置为'6987'，则预期的安全报文数据丢失。
- 如果 SW1-SW2 设置为'6988'，则安全报文数据对象出错。

附录 B 提供了安全报文的示例。

7 交换命令

本章规定交换命令，分为以下 6 组：

1) 选择；
2) 数据单元操作；
3) 记录操作；
4) 数据对象操作；
5) 基本安全操作；
6) 传输操作。

并不强制所有的卡都遵循本文档，以支持上述命令或命令的所有选项。当需要交互时，一组与应用无关的卡服务和相关命令由第 8 章规定。

7.1 选择

除非在历史字节(见 8.1.1)中或在初始数据串(见 8.1.2)中有不同的规定，在复位应答之后，MF 和应用 DF 可通过基本逻辑通道(见 5.1.1.2)隐式地进行选择。

7.1.1 SELECT 命令

该命令完成时，将打开由 CLA(见 5.1.1)中数字所指定的逻辑通道(见 5.1.1.2)，如果尚未打开，则在该逻辑通道中设置一个当前结构，后续命令可以通过该逻辑通道隐式地引用该当前结构。

——选择的 DF(MF 或应用 DF)将成为该逻辑通道中的当前 DF。如果存在以前选择的 DF，则它不再通过该逻辑通道引用，并变成前一个当前 DF。在这种选择之后，可以通过该逻辑通道来引用一个隐含的当前 EF。

——选择 EF 时设置了一对当前文件：EF 及其父 DF 文件。

除非另外规定，下面的规则将适用于一个 DF 层次结构中每个打开的逻辑通道：

——如果当前 EF 被改变，或在没有当前 EF 时，将失去针对前一个当前 EF 的安全状态。

——如果当前 DF 是前一个当前 DF 的后代，或相同时，针对前一个当前 DF 的安全状态将保持不变。

——如果当前 DF 既不是前一个当前 DF 的后代，也不相同时，针对前一个当前 DF 的安全状态将丢失。先前的和新的当前 DF 的所有共同祖先，所共用的安全状态将维持不变。

表 38 SELECT 命令-响应对

CLA INS P1 P2	由 5.1.1 定义 'A4' 见表 39 见表 40
Lc	当 Nc=0 时不存在，当 Nc>0 时存在
数据字段	不存在或文件标识或路径或 DF 名称(根据 P1)
Le	当 Ne=0 时不存在，当 Ne>0 时存在

数据字段	不存在或文件控制信息(根据 P2)
SW1-SW2	见表 5 和表 6 相关定义，比如'6283'，'6284'，'6A80'，'6A81'，'6A82'，'6A86'，'6A87'

当 P1='00'时，卡会知道选择的文件是否为 MF、DF 或 EF，因为有文件标识的特定编码，或者根据命令执行的上下文。

——此时，P2='00'时，如果提供了文件标识，则在下列情况下，文件标识应唯一：1)当前 DF 的直接子文件；2)父 DF；3)父 DF 的直接子文件。

——此时，P2='00'时，如果数据字段不存在或等于'3F00'，则选择 MF。

当 P1='04'时，数据字段为 DF 名称，该名称可能是应用标识符(见 8.2.1.2)，也可能是应用标识的一部分(将右边截短)。如果支持，则这种带有相同数据字段的连续的命令将选择那些名称与数据字段相匹配的 DF，比如数据字段是以命令数据字段开始。如果卡接受了不带有数据字段的 SELECT 命令，则全部 DF 或 DF 的子集能够被连续选择。

如果 Le 字段仅包含'00'，则将返回对应于选择选项的所有字节，其长度对于短 Le 字段来说不超过 256，而对于扩充 Le 字段，长度不超过 65 536。如果没有 Le 字段，比如不返回任何文件控制信息，则响应数据字段也将不存在。

表 39 P1 定义

b8	b7	b6	b5	b4	b3	b2	b1	含义	命令数据字段
0	0	0	0	0	0	x	x	通过文件标识选择	
0	0	0	0	0	0	0	0	选择 MF、DF 或 EF	不存在或文件标识
0	0	0	0	0	0	0	1	选择子 DF	DF 标识
0	0	0	0	0	0	1	0	在当前 DF 下选择 EF	EF 标识
0	0	0	0	0	0	1	1	选择当前 DF 的父 DF	不存在
0	0	0	0	0	1	x	x	根据 DF 名选择	
0	0	0	0	0	1	0	0	通过 DF 名选择	如截短的应用标识
0	0	0	0	1	1	x	x	根据路径选择	
0	0	0	0	1	0	0	0	从 MF 中选择	无 MF 标识的路径
0	0	0	0	1	0	0	1	从当前 DF 中选择	无当前 DF 标识的路径
——任何其他值由 ISO/IEC JTC1/SC17 保留供将来使用。 ——当出现在历史字节或 EF.ART(见 8.2.1.1)中，第 1 个软件功能表(见表 86)指示卡所支持的选择方法。									

表 40 P2 定义

b8	b7	b6	b5	b4	b3	b2	b1	含　义
0	0	0	0	—	—	x	x	文件出现
0	0	0	0	—	—	0	0	首次或唯一出现
0	0	0	0	—	—	0	1	最后出现
0	0	0	0	—	—	1	0	下一次出现
0	0	0	0	—	—	1	1	前次出现
0	0	0	0	x	x	—	—	文件控制信息(见 5.3.3 和表 11)
0	0	0	0	0	0	—	—	返回 FCI 模版，可选使用的 FCI 标签和长度
0	0	0	0	0	1	—	—	返回 FCP 模版，强制使用的 FCP 标签和长度
0	0	0	0	1	0	—	—	返回 FMD 模版，强制使用的 FMD 标签和长度
0	0	0	0	1	1	—	—	当 Le 不存在无应答数据，当 Le 存在返回专有
——任何其他值由 ISO/IEC JTC1/SC17 保留供将来使用。								

7.1.2 MANAGE CHANNEL 命令

该命令打开或关闭除基本通道外的逻辑通道(见 5.1.1.2)，即从 1 到 19 的通道号(大于 19 保留将来使用)。

打开功能打开一个除基本通道外的新的逻辑通道。卡可以分配或接受该逻辑通道。

——如果 P1 的 bit 8 设置为'0'(比如 P1 设置为'00'，因为其他 7 位被保留将来使用)，则 MANAGE CHANNEL 将以如下方式打开一个编号从 1 到 19 的通道：

- 如果 P2 设置为'00'，则 Le 字段应被设置为'01'，并且响应数据字段应包含一个由卡分配的从'01'到'13'的非 0 的一个字节的通道号。

- 如果 P2 设置为从'01'到'13',则它编码了外部分配的非 0 通道号,此时 Le 字段不存在。

——当打开功能从一个基本逻辑通道上执行后(CLA 编码为 0 通道号码时),MF 或缺省的应用 DF 应被隐式地选择为新通道上的当前 DF。

——当打开功能从一个非基本逻辑通道上执行后(CLA 编码为非 0 通道号码时),CLA 指定编号的通道上的当前 DF 应成为新通道上的当前 DF。

关闭功能显式地关闭一个除基本通道外的逻辑通道,Le 字段应不存在。关闭后,该逻辑通道能够重新使用。

——如果 P1 的 bit 8 设置为'1'(比如 P1 设置为'80',因为其他 7 位被保留将来使用),则 MANAGE CHANNEL 将以如下方式关闭一个编号从 1 到 19 的通道:

- 如果 P2 设置为'00',则 CLA 中编号(非 0 的通道号)指定的通道将被关闭。
- 如果 P2 设置为'01'到'13',则 P2 中指定的通道将被关闭。

警告:如果 CLA 指示的既不是基本通道也不是 P2 指示的通道号,则关闭命令将失效。

表 41 MANAGE CHANNEL 命令-响应对

CLA	由 5.1.1 定义
INS	'70'
P1-P2	'0000'打开应答数据字段中号码指定的逻辑通道 '0001'到'0013'打开 P2 中号码指定的逻辑通道 '8000'关闭 CLA 指定的逻辑通道(除基本通道外) '8001'到'8013'关闭 P2 中号码指定的逻辑通道 其他值保留将来使用
Lc	当 Nc=0 时不存在
数据字段	不存在
Le	当 Ne=0 时不存在,当 Ne=1 时存在

数据字段	不存在(P1-P2 设置非 0 时)或 '01'到'13'(P1-P2 设置为'0000'时)
SW1-SW2	见表 5 和表 6 相关定义,比如'6200','6881','6A81'

7.2 数据单元操作

7.2.1 数据单元

在每个支持数据单元的 EF 内均有一个偏移指向每个数据单元。从 0 对应 EF 的第 1 个数据单元开始,偏移每加 1 对应其下一个数据单元。偏移数据元是最小字节数的二进制编码。指向不包含在 EF 中的数据单元将导致错误。

卡能够在历史字节(见 8.1.1),EF.ATR(见 8.2.1.1)以及任何文件的文件控制信息(见表 12 中的 tag'82')中提供数据编码字节(见表 87)。数据编码字节固定了数据单元的大小。

——如果卡在几个地方均提供了数据编码字节,则对给定的 EF 来说,从 MF 到该 EF 路径上离它最近位置的数据编码字节是有效的。

——路径中如果缺少指示,则数据单元大小对该 EF 来说是 1 字节(默认值)。

7.2.2 通则

该组中的所有命令在被应用到不支持数据单元的 EF 上时,应被中止。仅当安全状态满足 read、write、update、erase 或 search 等功能中定义的安全属性时,这些命令才能在 EF 上执行。

该组中每个命令可以使用短 EF 标识符或文件标识符。当命令发出时,如果存在一个当前 EF,则所有对应位设置为 0 时,操作可以完成。如果操作完成,则被标识的 EF 成为当前 EF。

INS P1 P2——该组所有命令应按如下方式使用 INS 的 bit 1 和 P1 的 bit 8。

——如果 INS 的 bit 1 为 0，P1 bit 8 为 1，则 P1 的 bit 6、bit 7 设置为 00(RFU)，P1 的 bit 5 到 bit 1 为 EF 短标识符，并且 P2(所有 8 位)编码为 0 到 255 的偏移。

——如果 INS 的 bit 1 为 0，P1 bit 8 为 0，则 P1-P2(15 位)编码为 0 到 32767 的偏移。

——如果 INS 的 bit 1 为 1，则 P1-P2 应标识 EF。如果 P1-P2 的前 11 位为 0，并且 P2 的 bit 5 到 bit 1 不相等，并且卡和(或)EF 文件支持按短文件标识符选择，则 P2 的 bit 5 到 bit 1 编码为 EF 短文件标识符(从 1 到 30 的数)。否则，P1-P2 为文件标识符。P1-P2 设置为'0000'标识当前 EF。至少一个带有 tag'54'的数据对象偏移应在命令数据字段中。当出现在命令或应答数据字段中时，数据应被封装进带有 tag'53'或'73'的自主数据对象中。

该组命令中，SW1-SW2 设置为'63CX'表示成功改变内存状态，但有一个内部重试次数。'X>0'表示重试次数。'X=0'表示不提供重试。

7.2.3 READ BINARY 命令

响应数据字段给出了支持数据单元的 EF 的(部分)内容。

如果 Le 字段仅包含设置为 '00' 的字节，则直到文件结尾的所有字节将被读出，对于短 Le 字段读出字节数长度不超过 256，对于扩展 Le 字段长度不超过 65 536。

表 42 READ BINARY 命令-响应对

CLA INS P1-P2	由 5.1.1 定义 'B0'或'B1' 见 7.2.2
Lc	当 Nc=0 时不存在，当 Nc>0 时存在
数据字段	不存在(INS='B0')或数据对象偏移(INS='B1')
Le	当 Ne>0 时存在

数据字段	读出的数据(INS='B0')或 读出封装数据的自定义的数据对象(INS='B1')
SW1-SW2	见表 5 和表 6 相关定义，比如'6281'，'6282'，'6700'，'6981'，'6982'，'6986'，'6A81'，'6A82'，'6B00'，'6CXX'

7.2.4 WRITE BINARY 命令

该命令根据文件属性将对 EF 文件执行下列操作之一：

——一次写入命令数据字段中指定的数据位(如果数据单元的字串不是在逻辑擦除状态下，则命令将失效)。

——将命令数据字段中数据位和卡中已存在数据进行逻辑 OR 操作(文件的逻辑擦除状态位为 0)。

——将命令数据字段中数据位和卡中已存在数据进行逻辑 AND 操作(文件的逻辑擦除状态位为 1)。

缺省情况下，比如历史字节(见 8.1.1)中，或 EF.ATR 中(见 8.2.1.1)以及从 MF 到指定的 EF 路径上的每个文件的控制参数(见表 12 中 tag'82')中的数据编码字节(见表 87)不存在，则逻辑 OR 操作将被应用到 EF 上。

表 43 WRITE BINARY 命令-响应对

CLA INS P1-P2	由 5.1.1 定义 'D0'或'D1' 见 7.2.2
Lc	当 Nc>0 时存在
数据字段	要写入的数据单元字串(INS='D0')或要写入的封装的自定义数据对象偏移(INS='D1')
Le	当 Ne=0 时不存在

表 43（续）

数据字段	不存在
SW1-SW2	见表 5 和表 6 相关定义，比如‘63CX’（见 7.2.2），‘6581’，‘6700’，‘6981’，‘6982’，‘6B00’（偏移超过 EF 大小）

7.2.5 UPDATE BINARY 命令

该命令执行用命令数据字段中数据位更新 EF 文件中已存在数据位的操作。当操作完成后，每个指定的数据单元的每一位将被更新为命令数据字段中的指定值。

表 44 UPDATE BINARY 命令-响应对

CLA INS P1-P2	由 5.1.1 定义 ‘D6’或‘D7’ 见 7.2.2
Lc	当 Nc>0 时存在
数据字段	要更新的数据单元字串（INS=‘D6’）或要更新的封装的自定义数据对象偏移（INS=‘D7’）
Le	当 Ne=0 时不存在

数据字段	不存在
SW1-SW2	见表 5 和表 6 相关定义，比如‘63CX’（见 7.2.2），‘6581’，‘6700’，‘6981’，‘6982’，‘6B00’（偏移超过 EF 大小）

7.2.6 SEARCH BINARY 命令

该命令执行在 EF 中搜索数据单元的操作，响应数据中返回找到的数据单元的偏移；EF 中该偏移处的字节串应同命令数据字段中搜索字节串相同。当 Le 不存在或没找到匹配串时，应答数据字段不存在。如果搜索字串不存在，则应答数据字段返回逻辑擦除状态的第一个数据单元的偏移。

表 45 SEARCH BINARY 命令-响应对

CLA INS P1-P2	由 5.1.1 定义 ‘A0’或‘A1’ 见 7.2.2
Lc	当 Nc=0 时不存在，当 Nc>0 时存在
数据字段	不存在或搜索字节串（INS=‘A0’）或偏移数据对象和封装搜索串的自定义数据对象（INS=‘A1’）
Le	当 Ne=0 时不存在，当 Ne>0 时存在

数据字段	不存在或匹配的第一个数据单元偏移（INS=‘A0’），或指示匹配搜索串的第一个数据单元的偏移数据对象
SW1-SW2	见表 5 和表 6 相关定义，比如‘6281’，‘6982’

7.2.7 ERASE BINARY 命令

该命令从一个指定偏移开始顺序设置 EF（部分）内容为逻辑擦除状态。

——如果 INS=‘0E’，则如果存在命令数据字段，则它编码为不被擦除的第 1 个数据单元的偏移，该偏移值应该比 P1-P2 中的值要高，如果数据字段不存在，则命令擦除整个 EF 文件。

——如果 INS=‘0F’，则如果存在命令数据字段，则它应由 0、1 或 2 个偏移数据对象组成，如果没有偏移，则命令擦除整个文件中的所有数据单元；如果有 1 个偏移，则它指向第 1 个要擦除的数据单元，直到擦除到文件结尾；如果有 2 个偏移，则它表示一个数据单元序列，第 2 个偏移应比第一个偏移高，它将擦除两个偏移中间的数据单元。

表 46 ERASE BINARY 命令-响应对

CLA INS P1-P2	由 5.1.1 定义 '0E'或'0F' 见 7.2.2
Lc	当 Nc=0 时不存在,当 Nc>0 时存在
数据字段	不存在或第一个不被擦除的数据单元偏移(INS='0E') 不存在或 1 或 2 个偏移数据对象(INS='0F')
Le	当 Ne=0 时不存在

数据字段	不存在
SW1-SW2	见表 5 和表 6 相关定义,比如'63CX'(见 7.2.2),'6581','6700','6981','6982','6B00'(偏移超过 EF 大小)

7.3 记录操作

7.3.1 记录

每个支持记录的 EF 内部,由一个记录号和(或)记录标识符来引用一个记录,引用 EF 外部的记录被视为错误。

由记录号引用:每个记录号是唯一的,并是顺序的

——在每个支持线性结构的 EF 中,当添加和写入时,记录号应该按顺序分配,比如按照创建的顺序。第 1 个记录(记录号为 1)是首先被创建的记录。

——在每个支持循环结构的 EF 中,记录号应该依次按逆序分配,比如第 1 个记录(记录号为 1)是最近被创建的记录。

下列附加的规则是为线性结构和循环结构定义的:

——0 应指向当前记录,比如那个由记录指针指向的记录。

由记录标识符引用:每个记录标识由应用提供。多个记录可以有相同的记录标识,这种情况下,则由记录中的数据来区别不同的记录。如果记录的数据字段是一个 SIMPLE-TLV 数据对象,则记录标识是数据对象的第一个字节,比如 SIMPLE-TLV tag。

由记录标识引用可以促使对一个记录指针进行管理。一次卡复位,一个 SELECT 命令或任何使用合法的短 EF 标识符来访问 EF 文件的命令均能够影响记录指针。但由记录号引用记录将不影响记录指针。

每次通过记录标识符的引用,目标记录的逻辑位置应被指出:第一次或最后一次记录的出现或相对于记录指针的下一次或前一次记录的出现。

——在每个支持线性结构的 EF 中,当添加和写入时,逻辑位置应该按顺序分配,比如按照创建的顺序。首先被创建的记录在第一个逻辑位置。

——在每个支持循环结构的 EF 中,逻辑位置应该依次按逆序分配,比如最近创建的记录是第一个逻辑位置。

下列附加的规则是为线性结构和循环结构定义的:

——首次出现的是指定标识符在第一个逻辑位置的记录。最后出现的是指定标识符在最后一个逻辑位置的记录。

——如果有当前记录,则下一个出现的应是有指定标识符的离该记录最近,并且逻辑位置要大于当前记录的记录。前一个出现的应是有指定标识的离该记录最近,并且逻辑位置要小于当前记录的记录。

——如果没有当前记录,则下一个出现的应等于第一个出现的记录,前次出现的应等于最后一个出现的记录。

——0 应指向第一个，而最后一个，下一个，前一个记录按数字顺序分配，独立于记录标识符。

7.3.2 通则

该组中的所有命令在被应用到不支持记录的 EF 上时，应被中止。仅当安全状态满足 read、write、append、update、erase 或 search 等功能中定义的安全属性时，这些命令才能在 EF 上执行。

该组中两个命令(read，update)使用奇数 INS 编码(数据字段编码为 BER-TLV)开始对指定记录部分的操作(部分读，部分更新)。然后一个偏移将指向记录中的每个字节：从记录的第一个字节开始，偏移每移动一个字节增加 1。指向记录外的字节将导致错误。根据需要，偏移数据是 tag 为'54'的二进制编码。当出现在命令或响应数据字段中，数据应被封装进 tag 为'53'或'73'的自定义数据对象中。

该组中每个命令都可以使用短 EF 标识符。当命令处理完后，标识 EF 成为当前 EF，并且记录指针被复位。如果在发起命令时存在一个当前 EF，则命令处理时无需指明 EF(通过设置相应的第 5 位为 0)。

P1——记录号或标识，从 1 到 254 的数字，编码为'01'到'FE'，0 被保留为特殊用途。255(以'FF'编码)保留供将来使用。

P2——8 位到 4 位为短文件标识(见表 47)，bit 3 到 bit 1 依赖命令。

表 47 P2 定义的短文件标识

b8	b7	b6	b5	b4	b3	b2	b1	含　义
0	0	0	0	0	—	—	—	当前 EF
不全为 0					—	—	—	短 EF 标识(从 1 到 30 的数字)
1	1	1	1	1	—	—	—	保留将来使用

该组命令中，SW1-SW2 设置为'63CX'表明内存状态改变成功，但在内部重试之后，'X>0'表示重试次数。'X=0'表示不提供重试。

7.3.3 READ RECORDS 命令

响应数据字段给出 EF 文件中指定记录的(部分)内容(或一个记录的开始部分)。

如果 INS='B2'并且如果记录是 SIMPLE-TLV 数据对象(见 5.2.1)，则表 50 给出了响应数据字段。Nr 和 TLV 结构的比较，表明唯一的记录(读一个记录)或最后一个记录(读所有记录)是完整的、不完整或填充的。

注：如果记录不是数据对象，则读所有记录功能将不按边界接受记录。

如果 INS='B3'，则命令读取由 P1 指定部分记录，命令数据字段包含一个数据对象的偏移(tag '54')，指向记录中读取的第一个字节。响应数据字段包含一个自定义数据对象(tag'53')封装所读数据。

表 48 READ RECORD(S)命令-响应对

CLA INS P1 P2	由 5.1.1 定义 'B2'或'B3' 记录号或记录标识('00'表示当前记录) 见表 49
Lc	当 Nc=0 时不存在，当 Nc>0 时存在
数据字段	不存在(当 INS='B2')，或数据对象偏移(当 INS='0F')
Le	当 Ne>0 时存在

数据字段	读取的数据(INS='B2')，或封装所读数据的自主数据对象(INS='B3')
SW1-SW2	见表 5 和表 6 相关定义，比如'6281'，'6282'，'6700'，'6981'，'6982'，'6986'，'6A81'，'6A82'，'6A83'，'6CXX'

表 49 P2

b8	b7	b6	b5	b4	b3	b2	b1	含 义
x	x	x	x	x	—	—	—	由表 47 定义的短文件标识
—	—	—	—	—	0	x	x	P1 中记录标识
—	—	—	—	—	0	0	0	读第一个出现
—	—	—	—	—	0	0	1	读最后一个出现
—	—	—	—	—	0	1	0	读下一个出现
—	—	—	—	—	0	1	1	读前一个出现
—	—	—	—	—	1	x	x	P1 中记录号
—	—	—	—	—	1	0	0	读记录 P1
—	—	—	—	—	1	0	1	读从 P1 到最后的所有记录
—	—	—	—	—	1	1	0	读从最后到 P1 的所有记录
					1	1	1	保留将来使用

如果 Le 字段仅包含‘00’的字节，则命令完整读取请求的单个记录或记录序列，依赖于 P2 的bit 3、bit 2 和 bit 1 并且对短 Le 字段长度限制在 256 字节，扩展 Le 字段长度限制在 65 536 字节。

表 50 INS=‘B2’时响应数据字段

情形 a——单记录的部分读(Le 字段不仅包含‘00’字节)

Tn(1 字节)	Ln(1 或 3 字节)	Vn 的起始字节

←Nr 字节→

情形 b——单记录的完整读(Le 字段仅包含‘00’字节)

Tn(1 字节)	Ln(1 或 3 字节)	Vn 的所有字节

情形 c——多记录的部分读(Le 字段不仅包含‘00’字节)

Tn-Ln-Vn	…	Tn+m-Ln+m-Vn+m(记录的首字节)

←Nr 字节→

情形 d——读多记录直到文件结尾(Le 字段仅包含‘00’字节)

Tn-Ln-Vn	…	Tn+m-Ln+m-Vn+m(记录的首字节)

7.3.4 WRITE RECORD 命令

该命令对 EF 文件执行下列操作：

——按给定的命令数据字段中数据一次写入一个记录(如果记录不处于逻辑可擦除状态则命令中止)；

——将给定的命令数据字段中数据和卡中已存在的数据按逻辑 OR 操作；

——将给定的命令数据字段中数据和卡中已存在的数据按逻辑 AND 操作。

缺省情况下，比如历史字节(见 8.1.1)中，或 EF.ATR 中(见 8.2.1.1)以及从 MF 到指定的 EF 路径上的每个文件的控制参数(见表 12 中 tag‘82’)中的数据编码字节不存在，则逻辑 OR 操作将被应用到 EF 上。

当使用当前记录寻址，命令将设置记录指针到成功写入的记录上。

如果应用到支持循环结构的固定大小的记录上，“前一个”操作(P2 的 3,2,1 位设置为 011)将按 APPEND RECORD 操作。

如果记录为 SIMPLE-TLV 数据对象(见 5.2.1)，则由表 53 定义命令数据字段。

表 51 WRITE RECORD 命令-响应对

CLA INS P1 P2	由 5.1.1 定义 'D2' 记录号('00'表示当前记录) 见表 52
Lc	当 Nc>0 时存在
数据字段	要写入的记录
Le	当 Ne=0 时不存在

数据字段	不存在
SW1-SW2	见表 5 和表 6 相关定义,比如'63CX'(见 7.3.2),'6581','6700','6981','6982','6986','6A81','6A82','6A83','6A84','6A85'

表 52 P2

b8	b7	b6	b5	b4	b3	b2	b1	含 义
x	x	x	x	x	—	—	—	由表 47 定义的短文件标识
—	—	—	—	—	0	x	x	P1 设置为 00
—	—	—	—	—	0	0	0	第一个记录
—	—	—	—	—	0	0	1	最后的记录
—	—	—	—	—	0	1	0	下一个记录
—	—	—	—	—	0	1	1	前一个记录
—	—	—	—	—	1	0	0	P1 中记录号
——任何其他值由 ISO/IEC JTC1/SC17 保留供将来使用。								

表 53 命令数据字段(一个完整记录)

Tn(1 字节)	Ln(1 或 3 字节)	Vn 的所有字节

7.3.5 UPDATE RECORD 命令

该命令根据命令数据字段中给定的字节更新指定的记录。当使用当前记录寻址时,命令应将记录指针指向成功更新的记录

——如果应用到支持定长记录线性结构或定长记录循环结构的 EF 上,则如果记录大小不同于存在的记录大小时,命令将中止。

——如果应用到支持变长记录线性结构或变长记录循环结构的 EF 上,则如果记录大小不同于存在的记录大小时,命令仍有效。

——如果应用到支持定长记录循环结构的 EF 上,则"前一个"选项(P2 的 3,2,1 位设置位 011)将按 APPEND RECORD 操作。

表 54 UPDATE RECORD 命令-响应对

CLA INS P1 P2	由 5.1.1 定义 'DC'或'DD' 记录号('00'表示当前记录) 见表 52(INS='DC')或表 55(INS='DD')
Lc	当 Nc>0 时存在
数据字段	更新的记录(INS='DC'),或数据对象偏移和封装更新数据的自定义数据对象(INS='DD')

表 54（续）

Le	当 Ne=0 时不存在

数据字段	不存在
SW1-SW2	见表 5 和表 6 相关定义，比如‘63CX’(见 7.3.2)，‘6581’，‘6700’，‘6981’，‘6982’，‘6986’，‘6A81’，‘6A82’，‘6A83’，‘6A84’，‘6A85’

如果 INS=‘DC’，并且如果记录是 SIMPLE-TLV 数据对象(见 5.2.1)，则表 53 显示数据命令字段。

如果 INS=‘DD’，则部分更新 P1 指定的记录。命令数据字段应包括指明要更新记录的第一个字节的偏移数据对象(tag‘54’)，或用于封装更新数据的自定义数据对象(tag‘53’和‘73’)。

表 55 INS=‘DD’时的 P2

b8	b7	b6	b5	b4	b3	b2	b1	含 义
x	x	x	x	x	—	—	—	由表 47 定义的短文件标识
—	—	—	—	—	1	x	x	P1 中记录号
—	—	—	—	—	1	0	0	替换
—	—	—	—	—	1	0	1	逻辑 AND
—	—	—	—	—	1	1	0	逻辑 OR
—	—	—	—	—	1	1	1	逻辑 XOR
——任何其他值由 ISO/IEC JTC1/SC17 保留供将来使用。								

7.3.6 APPEND RECORD 命令

该命令在支持线性结构的 EF 文件结尾写入一个新的记录，或在支持循环结构的 EF 中写入记录号为 1 的记录。当使用当前记录寻址时，命令应将记录指针指向成功添加的记录。

如果命令应用到记录已满的支持线性结构的 EF 中时，则命令中止，因为文件中没有足够的空间。

如果命令应用到记录已满的支持循环结构的 EF 中时，则记录号最高的记录被替换，该记录号变为 1。

如果记录为 SIMPLE-TLV 数据对象(见 5.2.1)时，则表 53 显示命令数据字段。

表 56 APPEND RECORD 命令-响应对

CLA INS P1 P2	由 5.1.1 定义 ‘E2’ ‘00’(其他值非法) 见表 47，位 3,2,1 设置为 000(其他值保留将来使用)
Lc	当 Nc>0 时存在
数据字段	添加的记录
Le	当 Ne=0 时不存在

数据字段	不存在
SW1-SW2	见表 5 和表 6 相关定义，比如‘63CX’(见 7.3.2)，‘6581’，‘6700’，‘6981’，‘6982’，‘6986’，‘6A81’，‘6A82’，‘6A83’，‘6A84’，‘6A85’

7.3.7 SEARCH RECORD 命令

该命令对 EF 中的记录进行简单的，增强的或专有搜索。搜索可以限制在具有给定标识的记录或比给定号码大或小的记录上，可以按照记录号增序或降序执行。搜索从记录的第一个字节(简单搜索)，或记录中指定的偏移(增强搜索)或记录中给定字节的首次出现(增强搜索)开始。响应数据字段给出了

与支持记录的 EF 中搜索条件匹配的记录数目。命令将记录指针指向第一个匹配的记录。

在支持变长记录线性结构的 EF 中，命令不考虑比搜索串长度短的记录。在支持定长记录线性结构或循环结构的 EF 中，如果搜索串比记录长，则命令中止。

表 57 SEARCH RECORD 命令-响应对

CLA	由 5.1.1 定义
INS	'A2'
P1	记录号或记录标识(00 表示当前记录)
P2	见表 58
Lc	当 Nc>0 时存在
数据字段	搜索串(P2 的 3,2 位不设置为 11,简单搜索)或 搜索标志(2 字节)跟搜索串(P2 的 3,2,1 设置为 110,增强搜索)或(P2 的 3,2,1 设置为 111)专有搜索
Le	当 Ne=0 时不存在,当 Ne>0 时存在

数据字段	不存在或记录数目
SW1-SW2	见表 5 和表 6 相关定义,比如'6282','6982','6CXX'
——响应数据字段不存在因为 Le 不存在或没有找到匹配。 ——响应数据字段不提供记录标识因为它们可能不唯一。	

表 58 P2

b8	b7	b6	b5	b4	b3	b2	b1	含 义
x	x	x	x	x	—	—	—	由表 47 定义的短文件标识
—	—	—	—	—	0	x	x	按 P1 中记录标识简单搜索
—	—	—	—	—	0	0	0	从首次出现向前
—	—	—	—	—	0	0	1	从最后出现向后
—	—	—	—	—	0	1	0	从下次出现向前
—	—	—	—	—	0	1	1	从前一次出现向后
—	—	—	—	—	1	0	x	按 P1 中记录号简单搜索
—	—	—	—	—	1	0	0	从 P1 向前
—	—	—	—	—	1	0	1	从 P1 向后
—	—	—	—	—	1	1	0	增强搜索(见表 59)
—	—	—	—	—	1	1	1	专有搜索

在增强搜索中(P2 的 bit 3,bit 2,bit 1 设置为 110),命令数据字段包含 1 个 2 字节的搜索指示,其后紧随搜索串。表 59 规定了第一个搜索指示字节。根据第 1 个字节,第 2 个搜索指示字节或者为偏移或者一个值,比如搜索要么从记录中的偏移(绝对位置)开始,要么从该值的首次出现开始。

表 59 搜索指示的第一个字节

b8	b7	b6	b5	b4	b3	b2	b1	含 义
0	0	0	0	0	—	—	—	下一个字节为偏移(从该位置开始)
0	0	0	0	1	—	—	—	下一个字节为值(从该值首次出现开始)
—	—	—	—	—	0	x	x	按P1 中记录标识
—	—	—	—	—	0	0	0	从首次出现向前
—	—	—	—	—	0	0	1	从最后出现向后
—	—	—	—	—	0	1	0	从下次出现向前
—	—	—	—	—	0	1	1	从前一次出现向后
—	—	—	—	—	1	x	x	按P1 中记录号
—	—	—	—	—	1	0	0	从 P1 向前
—	—	—	—	—	1	0	1	从 P1 向后

表 59（续）

b8	b7	b6	b5	b4	b3	b2	b1	含　义
—	—	—	—	—	1	1	0	从下一个记录向前
—	—	—	—	—	1	1	1	从前一个记录向后
——任何其他值由 ISO/IEC JTC1/SC17 保留供将来使用。								

7.3.8 ERASE RECORD(S) 命令

该命令设置 EF 中一个或多个记录为逻辑擦除状态，要么是由 P1 指定的记录，或从 P1 开始直到文件结尾的连续记录序列。擦除记录并不删除记录，并且仍然可以由 WRITE RECORD 命令或 UPDATE RECORD 命令访问。

表 60　ERASE RECORD(S)命令-响应对

CLA	由 5.1.1 定义
INS	'0C'
P1	记录号
P2	见表 61
Lc	当 Nc=0 时不存在
数据字段	不存在
Le	当 Ne=0 时不存在

数据字段	不存在
SW1-SW2	见表 5 和表 6 相关定义，比如'63CX'(见 7.3.2)，'6581'，'6700'，'6981'，'6982'，'6986'，'6A81'，'6A82'，'6A83'，'6A84'，'6A85'

表 61　P2

b8	b7	b6	b5	b4	b3	b2	b1	含　义
x	x	x	x	x	—	—	—	短 EF 标识符(根据表 47)
—	—	—	—	—	1	x	x	P1 中记录号
—	—	—	—	—	1	0	0	擦除 P1 记录
—	—	—	—	—	1	0	1	擦除从 P1 记录直到文件结尾
——任何其他值由 ISO/IEC JTC1/SC17 保留供将来使用。								

7.4 数据对象操作

7.4.1 通则

该组命令如果应用到不支持数据对象的结构(DF 或 EF)时将中止。它只能在安全状态满足应用上下文中功能所定义的安全条件时才能执行。

INS P1 P2——该组所有命令可使用一个奇 INS 编码(见 5.1.2)。INS 的第 1 位根据表 62 定义，同 P1-P2 一同使用。

表 62　P1-P2

条件	P1-P2 的值	含　义
偶 INS 编码	'0000'	用于文件导出(见 8.4)或面向卡的字串(见 8.6)
	'0040'到'00FF'	P2 中 BER-TLV tag(单字节)
	'0100'到'01FF'	专有
	'0200'到'02FF'	P2 中 SIMPLE-TLV tag
	'4000'到'FFFF'	P1-P2 中 BER-TLV(双字节)
奇 INS 编码	任何值	文件标识或短文件标识(见下面文字描述)
——任何其他值由 ISO/IEC JTC1/SC17 保留供将来使用。		

——如果 INS 的 bit 1 设置为 0，且 P1 设置为'00'，则 P2 从'40'到'FE'应为一个单字节 BER-TLV tag。如果 BER-TLV tag 合法并且表明一个结构化编码，则命令设置相应模板为当前上下文。值'00FF'用于获取所有上下文中可读的通用 BER-TLV 数据对象，或者表明命令数据字段为 BER-TLV 编码。

——如果 INS 的 bit 1 设置为 0，且 P1 设置为'01'，则 P2 从'00'到'FF'应是一个卡内部测试和专有服务的标识符，它的意义在给定应用上下文中明晰。

——如果 INS 的 bit 1 设置为 0，且 P1 设置为'02'，则 P2 从'01'到'FE'应是一个 SIMPLE-TLV tag，值'0020'保留供将来使用。值'02FF' 用于获取所有上下文中可读的通用 SIMPLE-TLV 数据对象，或者表明命令数据字段为 SIMPLE-TLV 编码。

——如果 INS 的 bit 1 设置为 0，如果 P1-P2 从'4000'到'FFFF'，则它们应该是 2 字节的 BER-TLV tag。如果 BER-TLV tag 合法并且表明一个结构化编码，则命令设置相应模板为当前上下文。那些未生效的 2 字节 BER-TLV tag 的值保留供将来使用，比如'4000'和'FFFF'。

——如果 INS 的 bit 1 设置为 1，则 P1-P2 标识一个文件。如果 P1-P2 的前 11 位设置为 0，且 P2 的 bit 5 至 bit 1 都不相等，并且如果卡和/或文件支持短 EF 标识符选择，则 P2 的第 5 到 1 位编码为短 EF 标识符(从 1 到 30)，否则，P1-P2 是文件标识。P1-P2 设置'3FFF'标识当前 DF，P1-P2 设置'0000'标识当前 EF，除非命令数据字段提供了文件引用数据对象(tag'51'，见 5.3.1.2)标识一个文件。如果命令完成，则标识的文件成为当前文件。

数据字段——该组命令的数据字段定义如下：

——如果 INS 的 bit 1 设置为 0，如果在当前上下文(比如特定应用环境或当前 DF)中请求或提供数据对象，则数据字段或数据字段的串联应包含数据对象的值字段，比如，在 SIMPLE-TLV 数据对象或原始 BER-TLV 数据对象情况时引用的数据元，或者是在结构化 BER-TLV 数据对象情况下所引用的模板。

——对 2 种 INS 编码，如果提供了一个数据对象集合或请求 EF 内容时，则相应的数据字段应包含数据对象。

7.4.2 GET DATA 命令

该命令返回支持数据对象的 EF 文件内容，或在当前上下文中(比如特定应用环境或当前 DF)可能是结构化的一个数据对象。

注：如果对单个响应数据字段来说信息太长，则卡应返回信息的开头，后面跟着 SW1-SW2 设置为'61XX'。然后，接下来的 GET RESPONSE 命令将提供'XX'字节的信息。该过程可以重复直到卡返回 SW1-SW2 为'9000'。

如果 INS='CB'，则命令数据字段将包含一个 tag 列表数据对象，或一个头列表数据对象，或一个扩展的头列表数据对象(tag'5C'，'5D'，'4D'，见 8.5.1)。

——如果是 tag 列表情况，响应数据字段将包含由按照列表中的顺序连接在一起的由 tag 列表中引用的数据对象(可以缺少一个或多个数据对象)，一个空列表将要求所有可用的数据对象。

——如果是头列表情况，响应数据字段将包含按头列表中引用顺序连接在一起的表中所引用的截短的数据对象。

——如果是扩展头列表情况，响应数据字段包含由扩展头列表得到的连接在一起的数据对象(见 8.5.1)。

当一个 tag 多次出现时，这种情况不能定义哪个数据对象被返回，因为它依赖于数据对象的内容、特征或定义。

如果物理接口不允许卡复位应答，比如通用串行总线或通过无线射频，则 GET DATA 命令将根据 P1-P2 设置获得特定信息。下列特定信息的一种将从卡中获取。

——INS='CA'时

- tag'5F51'，复位应答是一个符合 GB/T 16649.3 的最大 32 字节的串。

- tag‘5F52’,符合 8.1.1 的最大 15 字节的历史字节串。

——INS=‘CB’且在命令数据字段中含空 tag 列表,即‘5C00’时,

- 文件标识‘2F00’,EF.DIR 的内容是符合 8.2.1.1 的 BER-TLV 数据对象集。
- 文件标识‘2F01’,EF.ATR 的内容是符合 8.2.1.1 的 BER-TLV 数据对象集。

注 1:(背景信息),根据 GB/T 16649.3 定义的物理接口,卡对任何通过接触的冷复位或热复位操作进行应答。复位应答是异步字符序列。初始字符 TS 表示编码约定并提供基本时间单元的选择。尽管它可以根据约定解码,TS 是一个同步模式,并不是一个字节。根据 GB/T 16649.3,复位应答是最大 32 字节的字符串。一个称为强制格式字节的 T0,后跟着可选的接口字节,可选历史字节(根据 8.1.1 编码最多 15 个历史字节),和一个条件检查字节 TCK。当 TCK 存在时,所有从 T0 到 TCK 字节的异或为零。

注 2:对于 ATR 信息,如果 Le 字段数目少于实际长度,则卡不会返回开始信息并跟着 SW1-SW2 设置为‘6CXX’,而是应该使该命令中止,并仅返回 SW1-SW2 设置为‘6CYY’,表示实际可返回的数据字节长度。然而,‘6C00’表示 256 字节或更多。而 SW1-SW2 设置为‘61XX’表示还有‘XX’字节可获取。

如果 Le 字段仅包含‘00’字节,则所有要求的信息都将被返回,对于短 Le 字段来说长度限制为 256,对于扩展 Le 字段来说长度限制为 65 536。

表 63 GET DATA 命令-响应对

CLA INS P1-P2	由 5.1.1 定义 ‘CA’‘CB’ 见表 62
Lc	当 Nc=0 时不存在,当 Nc>0 时存在
数据字段	不存在(INS=‘CA’),或 tag 列表数据对象或扩展头列表数据对象(INS=‘CB’)
Le	当 Ne>0 时存在

数据字段	根据 P1-P2 的数据字节(INS=‘CA’)或 BER-TLV 数据对象的连接(INS=‘CB’)
SW1-SW2	见表 5 和表 6 相关定义,比如‘61XX’,‘6202’,‘6700’,‘6981’,‘6982’,‘6985’,‘6A81’,‘6A88’(数据对象没找到,比如引用的数据没找到),‘6CXX’

7.4.3 PUT DATA 命令

该命令管理支持数据对象的 EF 文件内容,或在当前上下文中(比如特定应用环境或当前 DF)可能是结构化的一个数据对象。比如,它允许发送一个“待执行命令”(tag‘52’)或一个持卡人证书(tag‘7F21’),可能对单个命令太长。如果数据对象对单个命令来说太长,则可以应用命令链(见 5.1.1.1)。数据对象的值是命令数据字段的连接。

数据对象的内容或自然定义应导致一个管理功能,比如一次写入、更新或添加。

SW1-SW2 设置为‘63CX’表示成功改变内存状态,但在内部重试之后,‘X>0’表示重试次数。‘X=0’表示不提供重试。

表 64 PUT DATA 命令-响应对

CLA INS P1-P2	由 5.1.1 定义 ‘DA’或‘DB’ 见表 62
Lc	当 Nc>0 时存在
数据字段	根据 P1-P2 的数据字节(INS=‘DA’)或 BER-TLV 数据对象的连接(INS=‘DB’)
Le	当 Ne=0 时不存在

数据字段	不存在
SW1-SW2	见表 5 和表 6 相关定义,比如‘63XX’,‘6581’,‘6700’,‘6981’,‘6982’,‘6985’,‘6A80’,‘6A81’,‘6A84’,‘6A85’

7.5 基本安全操作

7.5.1 通则

该组命令保留 P1-P2 用于算法引用和一些相关数据引用(比如 key)等。如果有当前密钥和当前算法,则命令可以隐式的使用它们。

P1——除非特别指定,P1 引用一个使用的算法:密码算法或生物识别算法(见 ISO/IEC 7816-11[4])。P1 设置为'00'表示不提供任何信息,比如引用在发出命令前已经事先确定,或由命令数据字段提供。

P2——除非特别指定,P2 根据表 65 来规定引用数据。P2 设置为'00'表示不提供任何引用信息。比如在命令发出前已限定引用,或由命令数据字段提供。限定字节可以是口令编号或密钥编号,或一个短文件标识。

表 65 P2

b8	b7	b6	b5	b4	b3	b2	b1	含 义
0	0	0	0	0	0	0	0	不提供任何信息
0	—	—	—	—	—	—	—	全局引用数据(比如 MF 特定口令或密钥)
1	—	—	—	—	—	—	—	特定引用数据(比如 DF 特定口令或密钥)
—	x	x	—	—	—	—	—	00(其他值保留将来使用)
—	—	—	x	x	x	x	x	限定词,比如引用数据号或秘密号

注:一个安全管理环境命令可以设置引用算法和/或一个引用数据限定词。

该组命令中,SW1-SW2 设置为'6300'或'63CX'表示验证失败。'X'>'0'表示重试次数。SW1-SW2 设置为'6A88'表示引用数据没有找到。

7.5.2 INTERNAL AUTHENTICATE 命令

该命令利用接口设备发来的询问数据和存储在卡中的秘密(比如密钥)计算卡认证数据。

——如果相关秘密属于 MF,则命令将卡作为整体认证。

——如果相关秘密属于 DF,则命令将认证该 DF。

任何认证可能在先前的命令(比如 VERIFY,SELECT)或选择(比如相关秘密)执行完毕后才能成功完成。

为了限制将来的相关秘密和算法的使用,卡能够记录命令发起的次数。

注:响应数据字段可以包含有进一步的安全功能使用的数据(比如随机数)。

表 66 INTERNAL AUTHENTICATE 命令-响应对

CLA INS P1-P2	由 5.1.1 定义 '88' 见 7.5.1 和表 65
Lc	当 Nc>0 时存在
数据字段	认证相关数据(比如询问)
Le	当 Ne>0 时存在
数据字段	认证相关数据(比如对询问的应答)
SW1-SW2	见表 5 和表 6 相关定义,比如'6300'(见 7.5.1),'63CX'(见 7.5.1),'6581','6700','6982','6983','6984','6A81','6A82','6A86','6A88'(见 7.5.1)

7.5.3 GET CHALLENGE 命令

该命令获取用于安全相关过程(比如 EXTERNAL AUTHENTICATE 命令)的质询信息(比如,一个密码验证的随机数,或用于声波特征生物鉴别的一段提示语句)。

质询信息至少在下一个命令是合法的,没有其他特定条件。

表 67 GET CHALLENGE 命令-响应对

CLA INS P1 P2	由 5.1.1 定义 '84' 见 7.5.1 '00'(其他值保留将来使用)
Lc	当 Nc=0 时存在
数据字段	不存在
Le	当 Ne>0 时存在

数据字段	挑战数据
SW1-SW2	见表 5 和表 6 相关定义,比如'6300'(见 7.5.1),'63CX'(见 7.5.1),'6581','6700','6982','6983','6984','6A81','6A82','6A86','6A88'(见 7.5.1)

7.5.4 EXTERNAL AUTHENTICATE 命令

该命令根据卡的计算结果(是或否)有条件的更新安全状态,该结果基于先前由卡发出的质询数据(比如 GET CHALLENGE 命令),一个存储在卡中的密钥或秘密,和由接口设备传输的认证数据共同计算得出。

任何成功的认证要求最近从卡中获取的质询信息。卡将记录不成功的认证(比如限制引用数据的使用次数等)。

命令数据字段不存在可以用于得到重试次数'X'(SW1-SW2 设置为'63CX')。或检查是否要求验证(SW1-SW2 设置为'9000')。

表 68 EXTERNAL AUTHENTICATE 命令-响应对

CLA INS P1-P2	由 5.1.1 定义 '82' 见 7.5.1 和表 65
Lc	当 Nc=0 时存在,当 Nc>0 时存在
数据字段	不存在或认证相关数据(质询的响应)
Le	当 Ne=0 时不存在

数据字段	不存在
SW1-SW2	见表 5 和表 6 相关定义,比如'6300'(见 7.5.1),'63CX'(见 7.5.1),'6581','6700','6982','6983','6984','6A81','6A82','6A86','6A88'(见 7.5.1)

互认证功能——互认证功能等效于 INTERNAL 和 EXTERNAL AUTHENTICATE 两条命令。它基于先前的 GET CHALLENGE 命令和保存在卡中的密钥或秘密。卡和接口设备共享认证相关数据,包括 2 个质询数据:一个由卡发出,另一个由接口设备发出。

注:命令可以用于实现 ISO/IEC 9798[8]中 2、3 部分的规定的认证。

该操作只有在安全状态满足其安全属性时才能执行。

表 69 MUTUAL AUTHENTICATE 功能的命令-响应对

CLA INS P1-P2	由 5.1.1 定义 '82' 见 7.5.1 和表 65
Lc	当 Nc>0 时存在
数据字段	认证相关数据
Le	当 Ne=0 时不存在

数据字段	认证相关数据
SW1-SW2	见表 5 和表 6 相关定义，比如'6300'(见 7.5.1)，'63CX'(见 7.5.1)，'6581'，'6700'，'6982'，'6983'，'6984'，'6A81'，'6A82'，'6A86'，'6A88'(见 7.5.1)

7.5.5 GENERAL AUTHENTICATE 命令

该命令细化 EXTERNAL、INTERNAL 和 MUTUAL AUTHENTICATE 命令功能，也即一个外部世界的实体认证卡中的实体(INTERNAL AUTHENTICATE 功能)，或者卡中实体认证外部世界实体(EXTERNAL AUTHENTICATE 功能)，或两者都认证(MUTUAL AUTHENTICATE 功能)。当相应的认证机制涉及到质询-响应时，EXTERNAL 和 INTERNAL AUTHENTICATE 命令将排除涉及证据-质询-响应 3 元组(见 ISO/IEC 9798-5[8])的认证机制。3 元组交互要求两个或更多的 GENERAL AUTHENTICATE 命令-响应对：该命令-响应对可以链接(见 5.1.1.1)。

该功能(INTERNAL、EXTERNAL 或 MUTUAL AUTHENTICATE)只能在安全状态满足操作的安全属性才能够实现。任何认证可能在先前的命令(比如 VERIFY，SELECT)或选择(比如相关秘密)执行完毕后才能成功完成。卡执行控制的结果(是或否)可以有条件的更新安全状态。卡可以记录功能发起的次数，用于限制相关秘密和算法的使用次数。卡将记录不成功的认证(比如限制引用数据的使用次数)。

表 70 GENERAL AUTHENTICATE 命令-响应对

CLA INS P1-P2	由 5.1.1 定义 '86'或'87' 见 7.5.1 和表 65
Lc	当 Nc>0 时存在
数据字段	认证相关数据
Le	当 Ne=0 时不存在，当 Ne>0 时存在

数据字段	不存在(根据 Le 不存在，比如最近的 EXTERNAL AUTHENTICATE 命令或操作被取消)或认证相关数据
SW1-SW2	见表 5 和表 6 相关定义，比如'6300'(见 7.5.1)，'63CX'(见 7.5.1)，'6581'，'6700'，'6982'，'6983'，'6984'，'6A81'，'6A82'，'6A86'，'6A88'(见 7.5.1)

当数据字段存在时，它应包含由 tag'7C'引用的行业间模板。在动态认证模板中上下文相关类(首字节为'80'到'BF')保留用于表 71 列出的动态认证数据对象。

表 71 动态认证数据对象

Tag	值
'7C'	按照下列 tag 设置动态认证数据对象
'80'	证据(比如小于所用的公共模块一个或多个正数)
'81'	质询(一个或多个数字,可能为 0,小于使用的公共指数)
'82'	响应(小于所用公共模块的一个或多个正数)
'83'	提交的响应(包含一个或多个质询的大随机数的哈希)
'84'	认证编码(一个证据数据对象和一个或多个数据字段哈希)
'85'	指数(由密钥协商技术用于建立会话密钥一个正数)
'A0'	身份数据模板
——在该上下文中,ISO/IEC JTC1/SC17 保留任何其他数据对象上下文相关类(首字节为'80'到'BF')。	

下列规则应用于用于动态验证的行业间模板:

——如果模板中数据对象为空,则它将在下一个数据字段的模板中完成。

——在第一个命令数据字段中,模板表示动态验证功能如下:

- 一个证据请求,比如一个空证据,表示一个 INTERNAL AUTHENTICATE 功能。
- 一个质询请求,比如一个空的质询,表示一个 EXTERNAL AUTHENTICATE 功能。
- 如果不存在空数据对象,则表示 MUTUAL AUTHENTICATE 功能。除非卡中止该操作,否则响应数据字段中模板应包含同命令数据字段中模板相同的数据字段。MUTUAL AUTHENTICATE 功能允许 2 个实体使用一对由 tag'85'引用的指数数据元协商一个会话密钥(见 ISO/IEC 11770-3[14]中的密钥协商技术)。

动态认证可以在会话期间保护交互的数据字段。2 个实体都维护一个当前哈希值,一次由一个命令或响应数据字段更新。通过包含带有 tag'80'的证据数据对象的方式,带有 tag'84'的数据对象传送一个由更新当前编码所产生的认证编码。验证者继续重构一个证据和一个认证编码:如果重构的证据非零并且如果 2 个编码相同,则认证成功。

附录 C 表明对应执行 INTERNAL、EXTERNAL 和 MUTULA AUTHENTICATE 功能以及带有扩展数据字段认证和密钥协商的相应 GENERAL AUTHENTICATE 命令-响应对。

7.5.6 VERIFY 命令

该命令对卡中存储的引用数据和接口设备发送的验证数据(比如口令)或来自卡上的传感器(比如指纹)进行比较。比较结果将更新安全状态。卡将记录不成功的比较(将限制引用数据的使用次数)。

——如果 INS='20',命令数据字段通常为验证数据。命令数据字段不存在用于检查要求验证(SW1-SW2='63CX'其中'X'表示重试次数),或不要求验证(SW1-SW2='9000')。

——如果 INS='21',命令数据字段应为验证数据对象(比如 tag'5F2E'见 ISO/IEC 7816-11[4]),通常不为空。一个带有扩展头列表的空的验证数据对象(tag'4D'见 8.5.1)表示验证数据将来自卡上的传感器。

表 72 VERIFY 命令-响应对

CLA	由 5.1.1 定义
INS	'20'或'21'
P1	'00'(其他值保留将来使用)
P2	见表 65
Lc	当 Nc>0 时存在,当 Nc=0 时不存在

表 72（续）

数据字段	验证数据或不存在(INS='20') 验证数据对象以及有条件的扩展头列表(INS='21')
Le	当 Ne=0 时不存在

数据字段	不存在
SW1-SW2	见表 5 和表 6 相关定义，比如'6286''6300'(见 7.5.1)，'63CX'(见 7.5.1)，'6581'，'6700'，'6982'，'6983'，'6984'，'6A81'，'6A82'，'6A86'，'6A88'(见 7.5.1)

7.5.7 CHANGE REFERENCE DATA 命令

该命令利用接口设备发送来的新的引用数据替换保存在卡中的引用数据，或将其同接口设备发送的验证数据进行比较并利用接口设备发送的新的引用数据有条件的替换原有数据。只有在安全状态满足该命令的安全属性时才能执行。

表 73 CHANGE REFERENCE DATA 命令-响应对

CLA INS P1 P2	由 5.1.1 定义 '24' '00'或'01'(其他值保留将来使用) 见表 65
Lc	当 Nc>0 时存在
数据字段	验证数据后跟无界符的新引用数据(P1 设置为'00') 或新的引用数据(P1 设置为'01')
Le	当 Ne=0 时不存在

数据字段	不存在
SW1-SW2	见表 5 和表 6 相关定义，比如'6300'(见 7.5.1)，'63CX'(见 7.5.1)，'6581'，'6700'，'6982'，'6983'，'6984'，'6A81'，'6A82'，'6A86'，'6A88'(见 7.5.1)

7.5.8 ENABLE VERIFICATION REQUIREMENT 命令

该命令打开要求比较引用数据和验证数据的开关。只有在安全状态满足该命令的安全属性时才能执行。

表 74 ENABLE VERIFICATION REQUIREMENT 命令-响应对

CLA INS P1 P2	由 5.1.1 定义 '28' '00'或'01'(其他值保留将来使用) 见表 65
Lc	当 Nc>0 时存在，当 Nc=0 时不存在
数据字段	不存在(P1 设置为'01') 验证数据(P1 设置为'00')
Le	当 Ne=0 时不存在

数据字段	不存在
SW1-SW2	见表 5 和表 6 相关定义，比如'6300'(见 7.5.1)，'63CX'(见 7.5.1)，'6581'，'6700'，'6982'，'6983'，'6984'，'6A81'，'6A82'，'6A86'，'6A88'(见 7.5.1)

7.5.9 DISABLE VERIFICATION REQUIREMENT 命令

该命令关闭要求比较引用数据和验证数据的开关，并且可能打开要求比较其他引用数据和验证数据的开关。只有在安全状态满足该命令的安全属性时才能执行。

表 75 DISABLE VERIFICATION REQUIREMENT 命令-响应对

CLA INS P1 P2	由 5.1.1 定义 '26' '00','01'或 100XXXXX 其中 XXXXX 是引用数据号(其他值保留将来使用) 见表 65
Lc	当 Nc>0 时存在，当 Nc=0 时不存在
数据字段	不存在(P1 设置为'01') 验证数据(P1 设置为'00'或 10XXXXX)
Le	当 Ne=0 时不存在

数据字段	不存在
SW1-SW2	见表 5 和表 6 相关定义，比如'6300'(见 7.5.1)，'63CX'(见 7.5.1)，'6581'，'6700'，'6982'，'6983'，'6984'，'6A81'，'6A82'，'6A86'，'6A88'(见 7.5.1)

7.5.10 RESET RETRY COUNTER 命令

该命令复位引用数据重试次数为其初始值，或在完成一次复位引用数据重试次数为其初始值中改变引用数据。只有在安全状态满足该命令的安全属性时才能执行。

表 76 RESET RETRY COUNTER 命令-响应对

CLA INS P1 P2	由 5.1.1 定义 '2C' '00','01','02','03'(其他值保留将来使用) 见表 65
Lc	当 Nc>0 时存在，当 Nc=0 时不存在
数据字段	不存在(P1 设置为'03') 复位代码后跟无界符的新引用数据(P1 设置为'00') 复位代码(P1 设置为'01') 新的引用数据(P1 设置为'02')
Le	当 Ne=0 时不存在

数据字段	不存在
SW1-SW2	见表 5 和表 6 相关定义，比如'6300'(见 7.5.1)，'63CX'(见 7.5.1)，'6581'，'6700'，'6982'，'6983'，'6984'，'6A81'，'6A82'，'6A86'，'6A88'(见 7.5.1)

7.5.11 MANAGE SECURITY ENVIROMENT 命令

该命令准备安全报文(见第 6 章)和安全命令(即 EXTERNAL、INTERNAL、GENERAL AUTHENTICATE，也可见 GB/T 16649.8 中的 PERFORM SECURITY OPERATION)。命令支持下列功能：

——SET，比如设置或替换当前 SE 的一个组件

——STORE，比如用 P2 中的 SEID 字节保存当前 SE

——RESTORE，比如用保存在卡中的 SE 替换当前 SE，并由 P2 中的 SEID 字节标识

——ERASE，比如擦除保存在卡中的 SE，并由 P2 中的 SEID 字节标识

表 77 MANAGE SECURITY ENVIROMENT 命令-响应对

CLA INS P1 P2	由 5.1.1 定义 '22' 见表 78 见表 79
Lc	当 Nc>0 时存在，当 Nc=0 时不存在
数据字段	不存在(STORE,RESTORE 和 ERASE)或控制引用数据对象的连接(SET)
Le	当 Ne=0 时不存在

数据字段	不存在
SW1-SW2	见表 5 和表 6 相关定义，比如'6600','6987','6988','6A88'(见 7.5.1)

表 78 P1

b8	b7	b6	b5	b4	b3	b2	b1	含 义
—	—	—	1	—	—	—	—	命令数据字段中的安全消息
—	—	1	—	—	—	—	—	响应数据字段中的安全消息
—	1	—	—	—	—	—	—	计算、解密、内部认真和密钥协商
1	—	—	—	—	—	—	—	验证、加密、外部认证和密钥协商
—	—	—	—	0	0	0	1	SET
1	1	1	1	0	0	1	0	STORE
1	1	1	1	0	0	1	1	RESTORE
1	1	1	1	0	1	0	0	ERASE
——任何其他值由 ISO/IEC JTC1/SC17 保留供将来使用。								

表 79 P2

值	含 义
'XX'	STORE、RESTORE 和 ERASE 等情况下的 SEID 字节(GET SE 情况下设置为'00')
	SET 或 GET CRT 情况下命令数据字段中控制引用模板 tag
'A4'	——认证的控制引用模板(AT)
'A6'	——密钥协商的控制引用模板(KAT)
'AA'	——哈希编码的控制引用模板(HT)
'B4'	——密码校验和的控制引用模板(CCT)
'B6'	——数字签名的控制引用模板(DST)
'B8'	——保密性的控制引用模板(CT)
——任何其他值由 ISO/IEC JTC1/SC17 保留供将来使用。	

KEY DERIVATION 功能(密钥派生功能)——主密钥概念要求由主密钥派生卡中的一个密钥。表 80 显示了 MANAGE SECURITY ENVIROMENT 命令的派生密钥的用法。假定主密钥和算法均已选定(否则，MANAGE SECURITY ENVIROMENT 命令能够额外选择一个密钥和算法)。

注：依赖引用的算法，从主密钥派生密钥的数据可以是后继命令的部分输入数据(比如 EXTERNAL AUTHENTICATE)。单 MANAGE SECURITY ENVEROMENT 命令中的派生密钥用法不是必需的。

表 80 KEY DERIVATION 功能的命令-响应对

CLA INS P1 P2	由 5.1.1 定义 '22' 'X1'(SET,见表 78) CRT tag(比如后跟 EXTERNAL AUTHENTICATE 则为'A4',或一个 VERIFY CRYPTOGRAPHIC CHECKSUM 则为'B4')
Lc	当 Nc>0 时存在
数据字段	{'94'-L-派生密钥的数据(强制)};须有 SM 数据对象
Le	当 Ne=0 时不存在

数据字段	不存在
SW1-SW2	见表 5 和表 6 相关定义,比如'6600','6987','6988','6A88'(引用数据没找到)

7.6 传输处理

7.6.1 GET RESPONSE 命令

该命令发送(部分)响应 APDU,这些响应 APDU 不能被可用的传输协议发送。见 GB/T 16649.3 中例子。

如果 Le 字段仅包含'00'字节,则所有可用字节应被返回,对于短 Le 字段长度限制为 256,对于长 Le 字段长度限制为 65 536。

表 81 GET RESPONSE 功能的命令-响应对

CLA INS P1-P2	由 5.1.1 定义 'C0' '0000'(其他值保留将来使用)
Lc	当 Nc=0 时存在
数据字段	不存在
Le	当 Ne>0 时存在

数据字段	任何错误情况下不存在,或根据 Ne 的部分响应 APDU
SW1-SW2	见表 5 和表 6 相关定义,比如'61XX'('XX'表示多少额外的可用字节由后续的 GET RESPONSE 命令取回),'6281','6700','6A81','6A82','6A86','6CXX'

7.6.2 ENVELOPE 命令

该命令发送(部分)命令 APDU 或 BER-TLV 数据对象,这些命令 APDU 或 BER-TLV 数据对象不能被可用的传输协议发送。见 GB/T 16649.3 中例子。

注:附录 B 显示 ENVELOPE 命令在安全消息上的用法。

表 82 ENVELOPE 命令-响应对

CLA INS P1-P2	由 5.1.1 定义 'C2','C3' '0000'(其他值保留将来使用)
Lc	当 Nc>0 时存在
数据字段	部分命令 APDU(INS='C2')或(部分)BER-TLV 数据对象(INS='C3')
Le	当 Ne>0 时存在,当 Ne=0 时不存在

数据字段	(部分)响应 APDU(INS='C2')或不存在
SW1-SW2	见表 5 和表 6 相关定义,比如'6700'

8 与应用无关的卡服务

本章描述了与应用无关的卡服务，其在下面的文本中被称作“卡服务”。本章包含下列内容：

1) 卡标识；
2) 应用标识和选择；
3) 通过路径选择；
4) 数据检索；
5) 数据元检索；
6) 卡发起的字节串。

卡服务的目的是提供卡和接口设备之间的交换机制，它们(卡和接口设备)两者除了都遵循本部分标准外，彼此互不了解。卡服务来源于下列内容的任何组合：

——历史字节(见 8.1.1)；

——EF.DIR 和 EF.ATR 的内容(见 8.2.1.1)；

——命令序列。

除非另外指定，命令 APDU 均使用 CLA＝‘00’，即，没有命令链接，没有安全报文和基本逻辑通道。

只要一个应用在卡内已经被标识和选择，就没有必要遵循本章。应用可以使用与本部分兼容的其他机制来获得类似的功能。因此，这种解决方法可能不保证交换。

8.1 卡标识

该服务允许接口设备标识并处理卡。历史字节(见 8.1.1)为卡标识提供一般的支持。卡通过其逻辑内容直接(例如通过卡服务数据字节，见 8.1.1.2.3)和/或间接(例如通过初始的访问数据，见 8.1.1.2.4，该数据用来指明对复位应答后直接隐式选择的文件的访问以及可能的协议和参数选择)向外界提供信息。因此，此时获得的数据(即初始数据串，见 8.1.2)，后来有可能不可检索。

8.1.1 历史字节

8.1.1.1 目的和一般结构

历史字节描述卡的操作特性。

——当卡应答复位时，复位应答信息可能包含历史字节(见 GB/T 16649.3)。

——当物理接口不允许卡复位应答时，例如通用串行总线或通过射频 RF 访问，GET DATA 命令(见 7.4.2)可能获取历史字节(tag 为‘5F52’)。

第 1 个历史字节称为“分类指示符字节”。表 83 汇总了分类指示符字节为‘00’、‘10’或‘8X’时历史字节的格式。分类指示符字节为此外的其他值时说明历史字节为专有格式。

表 83 分类指示符字节

值	含义
‘00’	历史字节的最后 3 个字节为状态指示符(见 8.1.1.3)
‘10’	引用 8.1.1.4
‘80’	状态指示符可能存在于压缩 TLV 数据对象中(1 个、2 个或 3 个字节，见 8.1.1.3)
‘81’到‘8F’	保留供将来使用
——其他值表明为专有的格式。	

——如果第 1 个历史字节为‘00’，则其余字节由可选的连续的压缩 TLV 数据对象和后面紧跟的必备的状态指示符(最后 3 个字节，不在 TLV 中)组成。

——如果第 1 个历史字节为‘80’，则其余字节由可选的连续的压缩 TLV 数据对象组成；最后一个数据对象可能带有一个长度为 1 个、2 个或 3 个字节的状态指示符。

任何行业间 BER-TLV 数据对象由标志字段'4X',长度字段'0Y'和 Y 个字节的值字段组成,BER-TLV 数据对象可以被转化为压缩 TLV 数据对象,压缩 TLV 数据对象则由一个字节的"压缩首标"(设置为'XY')和 Y 个字节的值字段组成。

此后定义的任何行业间数据元都可能出现在 EF. ATR 中。如果数据元出现在 EF. ATR 中,则为 BER-TLV 数据对象的格式,即标志字段'4X',长度字段'0Y'和 Y 个字节的数据字段。

8.1.1.2 可选的数据元

8.1.1.2.1 国家或发行者指示符

由压缩首标的'1Y'、'2Y'引用,此行业间数据元为国家或发行者指示符(见表 9,tag 为'41'和'42')。表 84 对国家或发行者指示符进行说明。

表 84 国家或发行者指示符

压缩首标	值
'1Y'	国家编码(见 ISO 3166-1[1])和任选的国家数据
'2Y'	发行者标识号(见 GB/T 15694.1)和可选的发行者数据

——国家指示符由国家编码(3 个由'0'到'9'组成的 4 元组,见 ISO 3166-1[1])和紧跟其后的数据(至少一个 4 元组)组成,后者由相关的国家标准化组织选定(奇数个 4 位组)。

——发行者指示符由发行者标识号(见 GB/T 15694.1)和可能的紧跟其后的数据组成,如果后者存在,将由卡发行者选定(例如编码采用主账号 Primary Account Number)。

注:GB/T 15694.1 规定,发行者标识号由奇数个'0'到'9'4 元组组成,通过将最后字节的 b4～b1 设置为 1111 映射成字节串。

8.1.1.2.2 应用标识符

由压缩首标'FY'引用,此行业间数据元为应用标识符(AID,见 8.2.1.2,在表 9 中 tag 为'4F')。如果出现在历史字节或初始数据串中(见 8.1.2),AID 代表一个被隐式选择的应用(见 8.2.2.1)。

8.1.1.2.3 卡服务数据

由压缩首标'31'引用,此行业间数据元指示卡内支持本章描述的服务的方法。表 85 对卡服务数据字节进行说明。

表 85 卡服务数据字节

b8	b7	b6	b5	b4	b3	b2	b1	含　义
x	x	—	—	—	—	—	—	应用选择
1	—	—	—	—	—	—	—	——使用全部 DF 名称
—	1	—	—	—	—	—	—	——使用部分 DF 名称
—	—	x	x	—	—	—	—	BER-TLV 数据对象存在
—	—	1	—	—	—	—	—	——于 EF. DIR(见 8.2.1.1)内
—	—	—	1	—	—	—	—	——于 EF. ATR(见 8.2.1.1)内
—	—	—	—	x	x	x	—	访问EF. DIR 和EF. ATR
—	—	—	—	1	0	0	—	——通过 READ BINARY 命令访问(透明结构)
—	—	—	—	0	0	0	—	——通过 READ RECORD 命令访问(记录结构)
—	—	—	—	0	1	0	—	——通过 GET DATA 命令访问(TLV 结构)
—	—	—	—	其他值			—	保留供将来使用
—	—	—	—	—	—	—	0	有MF 的卡
—	—	—	—	—	—	—	1	没有MF 的卡

如果卡服务数据字节出现在历史字节或初始数据串(见 8.1.2)中,则是用来指示是否存在 EF. DIR 和 EF. ATR(见 8.2.1.1)文件及如何访问。如果历史字节或初始数据串中不存在卡服务数据字节,则说明卡只支持隐式应用选择(默认值)。

8.1.1.2.4 初始访问数据

由压缩首标'4Y'引用，此行业间数据元用来指示在复位应答及可能的协议和参数选择之后的第1条命令APDU。该命令APDU将在8.1.2进行详细说明。

8.1.1.2.5 卡发行者数据

由压缩首标'5Y'引用，GB/T 16649不对此行业间数据元进行定义。卡发行者定义长度、结构和编码。

8.1.1.2.6 预先发行的数据

由压缩首标'6Y'引用，GB/T 16649不对此行业间数据元进行定义。卡制造商定义卡制造商、集成电路名称、集成电路制造商、ROM掩膜版本、操作系统版本等的长度、结构和编码。此行业间数据元可能包含一个集成电路制造商标识符(见GB/T 16649.6)。

8.1.1.2.7 卡能力

由压缩首标'71'、'72'或'73'引用，此行业间数据元最多由3个软件功能表组成：即第一个表、前2个表或全部3个表。

——第1个软件功能表指明卡所支持的选择方法。

——第2个软件功能表称为"数据编码字节"，该字节还可能出现在文件控制参数(tag为'82'，见表12)的第2个字节。

——第3个软件功能表指示链接命令，处理扩展的Lc和Le、管理逻辑通道的能力。

表86 第1个软件功能表(选择方法)

b8	b7	b6	b5	b4	b3	b2	b1	含　义
x	x	x	x	x	—	—	—	DF选择(见5.3.1)
1	—	—	—	—	—	—	—	——通过全部DF名称
—	1	—	—	—	—	—	—	——通过部分DF名称
—	—	1	—	—	—	—	—	——通过路径
—	—	—	1	—	—	—	—	——通过文件标识符
—	—	—	—	1	—	—	—	隐式DF选择
—	—	—	—	—	1	—	—	所支持的短EF标识符
—	—	—	—	—	—	1	—	所支持的记录号
—	—	—	—	—	—	—	1	所支持的记录标识符

表87 第2个软件功能表(数据编码字节)

b8	b7	b6	b5	b4	b3	b2	b1	含　义
1	—	—	—	—	—	—	—	支持TLV结构的基本文件
—	x	x	—	—	—	—	—	写功能的行为
—	0	0	—	—	—	—	—	——一次性写
—	0	1	—	—	—	—	—	——专有的
—	0	0	—	—	—	—	—	——写'或'
—	1	1	—	—	—	—	—	——写'和'
—	—	—	—	x	x	x	x	数据单元的大小，以4元组为单位(从1到32 768个4元组，即16 384字节)(2的幂，例如0001＝2个4元组＝1字节，默认值)
—	—	—	x	—	—	—	—	BER-TLV tag字段的第一个字节值为'FF'(见5.2.2.1)
—	—	—	0	—	—	—	—	——无效(用于填充，默认值)
—	—	—	1	—	—	—	—	——有效(长私有tag，结构化的编码)

表 88 第 3 个软件功能表(命令链接、长度字段及逻辑通道)

b8	b7	b6	b5	b4	b3	b2	b1	含 义
1	—	—	—	—	—	—	—	命令链接(见 5.1.1.1)
—	1	—	—	—	—	—	—	扩展的 Lc 和 Le 字段(见 5.1)
—	—	—	x	x	—	—	—	逻辑通道号分配(见 7.1.2)
—	—	—	1		—	—	—	——由卡分配
—	—	—		1	—	—	—	——由接口设备分配
—	—	—	0	0	—	—	—	没有逻辑通道
—	—	—	—	—	y	z	t	逻辑通道的最大数目(见 5.1.1 和 5.1.2) ——当 y、z 和 t 不全为 1 时,值为 4y+2z+t+1,即从 1 到 7 ——当 y=z=t=1 时表示值为 8 或更多
—	—	x	—	—	—	—	—	保留供将来使用

8.1.1.3 状态指示符

如果分类指示符字节值为'00',则历史字节的最后 3 个字节为状态指示符,即卡生命周期状态字节(记为 LCS)后面紧跟 2 个处理状态字节(记为 SW1-SW2)。

如果分类指示符字节值为'80',则压缩首标为'81'、'82'或'83'的行业间数据元可能作为长度为 1 个、2 个或 3 个字节(其他长度被保留供 ISO/IEC JTC1/SC17 使用)的状态指示符出现在历史字节尾部。

——如果长度为 1,则数据元为卡生命周期状态字节 LCS。

——如果长度为 2,则数据元为 2 个处理状态字节 SW1-SW2。

——如果长度为 3,则数据元为 LCS 后面紧跟着 SW1-SW2。

LCS 的说明见 5.3.3.2 和表 13,其值'00'表示状态不被报告。SW1-SW2 的说明见 5.1.3 和表 5、表 6,其值'0000'表示状态不被报告。

8.1.1.4 DIR 数据引用

如果分类指示符为'10',则后随字节为 DIR 数据引用。该字节的编码及含义超出了本部分准的范围。

8.1.2 初始数据串恢复

由历史字节(见 8.1.1.2.4)中压缩首标'4Y'或 EF.ATR(见 8.2.1.1)中 tag'44'引用,行业间数据元"初始访问数据"表示一条命令 APDU。

——如果长度为 1,则命令 APDU 为如下 READ BINARY 命令(见 7.2.3):CLA INS P1 P2 设置为'00 B0 00 00',Le 字段设置为初始访问数据的第 1 个也是唯一的字节。

——如果长度为 2,则初始访问数据的第 1 个字节用来表示所读的 EF 的结构(b8)和短标识符(b5-1),见表 89。

- 如果第 1 个字节的 b8 为 1,则命令 APDU 为如下 READ BINARY 命令(见 7.2.3):CLA INS 设置为'00 B0',P1 设置为初始访问数据的第 1 个字节,P2 为'00',Le 字段设置为初始访问数据的第 2 个字节。
- 如果第 1 个字节的 b8 为 0,则命令 APDU 为如下 READ RECORD 命令(见 7.2.3):CLA INS P1 设置为'00 B2 01',P2 的 b8-4 设置为初始访问数据(短 EF 标识符)的第 1 个字节的 b5-1,P2 的 b3-1 设置为'110',Le 字段设置为初始访问数据的第 2 个字节。

表 89 当长度为 2 时初始访问数据的第 1 个字节

b8	b7	b6	b5	b4	b3	b2	b1	含 义
x	—	—	—	—	—	—	—	EF 结构
0	—	—	—	—	—	—	—	——记录结构
1	—	—	—	—	—	—	—	——透明结构
—	—	—	x	x	x	x	x	短 EF 标识符
—	x	x	—	—	—	—	—	x00x xxxx（其他值保留供将来使用）

——如果长度为 5 或更大，则命令 APDU 由 Y 个字节的初始访问数据组成。

命令 APDU 应该提交给卡。如果过程完成，响应数据字段为每个应用都会用到的一串行业间数据对象“初始数据串”。

8.2 应用标识和选择

该服务允许接口设备知道卡内激活的应用（如果有）以及如何标识和选择卡内的应用。

8.2.1 应用标识

8.2.1.1 EF. DIR 和 EF. ATR

基本文件 EF. DIR 和 EF. ATR 为应用标识和选择提供一般的支持。它们由一组 BER-TLV 数据对象组成。在这些文件内，对 BER-TLV 数据对象的删除和修改可能会引起对数据对象的填充（之前，之间或之后）。

EF. DIR ——该文件指明卡支持的一系列应用，由一组应用模板和/或应用标识符数据对象以任意顺序组成。该文件指明在选择指定的应用时需要执行哪些命令。

MF 是 EF. DIR 下的父文件：路径“3F002F00”指向 EF. DIR。在 MF 级，如果存在，短文件标识符 30（二进制为 11110）也指向该文件。

EF. ATR ——该文件指明卡的操作特性。它包含一组行业间数据对象，由于与应用选择无关，或 EF. DIR 文件不存在，这些数据不能放到 EF. DIR 文件中。

MF 是 EF. ATR 的父文件，路径“3F002F01”指向 EF. ATR。

8.2.1.2 应用标识符

由历史字节（见 8.1.1）的压缩首标‘FY’或初始数据串（见 8.1.2）、EF. DIR、EF. ATR 和任何 DF 管理数据（见 5.3.3）的 tag‘4F’引用，或此行业间数据元指示一个应用。

应用标识符最多由 16 个字节组成，第 1 个字节的 b8～b5 用来指明分类，如表 90 所示。

表 90 应用标识符的分类

值	分 类	含 义
‘0’到‘9’	—	保留以便与 GB/T 15694.1（见附录 D）向后兼容
‘A’	国际的	应用提供者根据 ISO/IEC 7816-5[4]进行国际注册
‘B’，‘C’	—	按照 ISO/IEC JTC1/SC17 要求保留供将来使用
‘D’	国家的	应用提供者根据 GB/T 16649.5 进行国家注册（ISO 3166-1[1]）
‘E’	标准的	对象标识符根据 GB/T 16263.1 对标准进行标识
‘F’	专有的	不注册应用提供者

图 7 为国际 AID 的说明。它包含 5 个字节的注册应用提供者标识符（国际 RID）和专有的应用标识符扩展（PIX），后者是任选的，最多 11 个字节。

——国际 RID 唯一标识应用提供者（见 ISO/IEC 7816-5[4]）。

- 第 1 个字节的 b8～b5 设置为 1010，即第 1 个 4 位组为‘A’。
- 之后的 9 个 4 位组的取值为‘0’到‘9’。

——扩展为自由编码，允许应用提供者标识不同的应用。

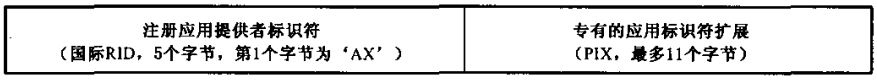

图 7 国际 AID

图 8 为国家 AID 的说明。它包含 5 个字节的注册应用提供者标识符(国家 RID)和专有的应用标识符扩展(PIX)，后者是任选的，最多 11 个字节。

——国家 RID 唯一标识应用提供者(见 GB/T 16649.5)。

- 第 1 个字节的 b8～b5 设置为 1101，即第 1 个 4 位组为‘D’。
- 之后的 3 个 4 位组的(取值为‘0’到‘9’)组成国家代码(见 ISO 3166-1[1])。
- 其余的 6 个 4 位组的取值建议为‘0’到‘9’。

——扩展为自由编码，允许应用提供者标识不同的应用。

图 8 国家 AID

图 9 为标准 AID 的说明。它最多包含 16 个字节。第 1 个字节设置为 1110 1000，即‘E8’。ISO/IEC JTC1/SC17 保留‘E0’到‘E7’和‘E9’到‘EF’供将来使用。其后跟随一个对象标识符(GB/T 16263.1)用来说明指定应用的标准(见附录 A 中的例子，比如 ISO/IEC 7816-12[4] 中通过生物方法进行个人识别，或 ISO/IEC 7816-15[4] 中加密信息应用)。最后可能跟随应用标识符扩展(根据识别标准指定)来标识不同的实现。

图 9 标准 AID

图 10 为专有 AID 的说明。它最多包含 16 个字节。第 1 个字节的 b8～b5 设置为 1111，即‘F’。在该类别中，由于应用提供者未注册，不同的应用提供者可能使用相同的 AID。

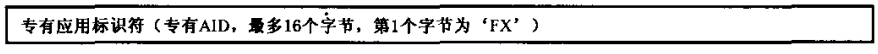

图 10 专有 AID

8.2.1.3 应用模板

由 tag‘61’引用，此行业间模板可能包含在 EF.DIR(见 8.2.1.1)、EF.ATR(见 8.2.1.1)和任何 DF(见 5.3.3)管理数据中。

——这样的模板仅包含一个应用标识符，如果一个目录有多个有效的应用标识符，则每一个应用标识符将处于不同的应用模板。

——这样的模板还可能包含其他的和应用相关的行业间数据对象，表 91 列出这些数据对象，并在后面的内容里进行定义。

表 91 应用识别与选择使用的行业间数据对象

tag	值
‘4F’	应用标识符
‘50’	应用标签
‘51’	文件引用
‘52’	命令 APDU

表 91（续）

tag	值
'53''73'	任意数据、任意模板
'5F50'	统一资源定位符（见 IETF RFC 1738[19]和 IETF RFC 2396[20]）
'61'	应用相关的数据对象集合

8.2.1.4 其他行业间数据元

下列行业间数据元为应用标识和选择提供一般的支持。

应用标签——由 tag'50'引用，GB/T 16649 不定义此行业间数据元。它由应用提供者定义并用于人机界面，例如展示的商标。

文件引用——由 tag'51'引用，定义见 5.3.1.2。

自主数据（或模板）——由 tag'53'（或'73'）引用，包含应用提供者定义的相关数据元（或嵌入数据对象）。

统一资源定位符——由 tag'5F50'引入，定义见 IETF RFC 1738[19]和 IETF RFC 2396[20]。它指向接口设备中用来和卡内应用通信的部分软件。

8.2.2 应用选择

卡至少应能支持下列应用选择方法中的一种。

1) 隐式应用选择；

2) 使用应用标识符作为 DF 名称进行应用选择；

3) 使用 EF.DIR 和 EF.ATR 进行应用选择。

8.2.2.1 隐式应用选择

如果某应用被隐式选择，则其应用标识符应出现在历史字节（见 8.1.1）或初始数据串（见 8.1.2）内，应用标识符这样出现即表示为隐式选择的应用。如果某隐式选择应用的应用标识符没有出现在历史字节和初始数据串内，则其应包含在 EF.ATR 中（见 8.2.1.1）。

注：隐式应用选择不推荐用于多应用卡。

8.2.2.2 使用 AID 作为 DF 名称进行应用选择

在多应用环境下，卡应该明确响应以应用标识符（AID，见 8.2.1.2）作为 DF 名称的 SELECT 命令。因此，接口设备可以明确地选择应用而不必事先检查该应用是否存在卡内。

卡应能支持 CLA INS P1 P2 设置为'00 A4 04 00'的、数据字段是给定的完整的应用标识符的 SELECT 命令（见表 39）。通过判断应用是否存在，卡完成或终止命令。如果使用截短的 DF 名称选择，完整的 DF 名称作为文件控制参数将出现在响应的数据字段，由 tag'84'引用（见表 12）。如果卡支持使用截短的 DF 名称选择，则第 1 次选择是与实现相关的，例如，第 1 次出现在静态列表内或前一会话中最后被激活的应用。如果后面还有选择操作，卡应能支持 CLA INS P1 P2 为'00 A4 04 02'、具有相同命令数据字段的 SELECT 命令。

8.2.2.3 使用 EF.DIR 和 EF.ATR 进行应用选择

对多应用的接口设备，使用 EF.DIR 或 EF.ATR 可能比前面的方法更有效。

——如果应用标识符数据对象不是应用模板的一部分，并且不和文件引用或待执行命令数据对象同时出现，那么使用 AID 作为 DF 名称进行选择。

——如果应用标识符数据对象是应用模板的一部分，同时还有文件引用数据对象（见 5.3.1.2）出现，且其数据字段包含 2 个或更多字节，那么将依据 8.3 要求按路径进行选择。

——如果应用标识符数据对象是应用模板的一部分，同时还有一个或多个待执行命令数据对象出现，则应用选择使用指定的命令。如果有多个这样的命令，则按照它们在模板里出现的顺序执行。

8.3 通过路径选择

该服务允许通过路径选择 EF 或未命名的 DF，即文件引用数据元(见 5.3.1.2)由 3 个或更多字节组成。

——当长度为偶数时，路径为绝对路径还是相对路径取决于前两个字节的设置是否为'3F00'。最后两个字节用来区别 DF 和 EF。

- 对于指向 DF 的路径，选择通过一条或多条 SELECT 命令执行，其中 CLA INS P1 P2 Lc 设置为'00 A4 01 00 02'。
- 对于指向 EF 的路径，如果长度为 4 个字节或更多，选择通过一条或多条 SELECT 命令执行，其中 CLA INS P1 P2 Lc 设置为'00 A4 01 00 02'。最后一条(可能是唯一的)选择命令使用路径的最后 2 个字节(EF 标识符)，其中 CLA INS P1 P2 Lc 设置为'00 A4 02 00 02'。

——当长度为奇数时，路径是限定的。它或者由不含'3F00'的绝对路径组成，或者由不含当前 DF 标识符的相对路径组成，后面的一个字节用作一条或多条 SELECT 命令的 P1。P1 的值决定选择的方式。

- 如果 P1 的值为'08'或'09'，则卡应支持一条选择命令：限定路径指定 P1、Lc 和数据字段的值，P2 设置为'00'。
- 其他情况下，卡应支持一条或多条 SELECT 命令，其 P1 设置为限定路径的最后一个字节，P2 与 Lc 设置为'0002'。路径上的文件按顺序选择。

8.4 数据检索

该服务允许接口设备读存储在 DF 和 EF 中的数据。

一旦某个 DF 被选择，和交换相关的内容将成为 GET DATA 命令(见 7.4.2)的响应数据字段，GET DATA 命令的 CLA INS 设置为'00 CA'，其后的 P1-P2 设置为'00 FF'(对于 BER-TLV 数据对象)或'02 FF'(对于 SIMPLE-TLV 数据对象)，接下来的 Le 仅包含一个字节'00'。

一旦某个 EF 被选择，如果文件描述符字节(见表 14)出现在 EF 的控制参数里，和交换相关的内容将成为对 READ 命令(见 7.4.2)响应的数据字段。

——READ BINARY(见 7.2.3)命令的 CLA INS P1 P2 设置为'00 B0 00 00'，其后的 Le 仅包含一个字节'00'。

——READ RECORD(见 7.3.3)命令的 CLA INS P1 P2 设置为'00 B2 00 05'，其后的 Le 仅包含一个字节'00'。

——GET DATA(见 7.4.2)命令的 CLA INS P1 P2 设置为'00 CA 00 00'，其后的 Le 仅包含一个字节'00'。

如果文件描述符字节没有出现在 EF 的文件控制参数里，则命令 APDU 遵守如下规则。

——如果第一个软件功能表(见表 86)出现在历史字节或 EF.ATR 中，并且表明支持记录，则命令 APDU 为 READ RECORD，如上述说明。

——否则，即第一个软件功能表没有出现在历史字节或 EF.ATR 中或该表未表明支持记录，则命令 APDU 为 READ BINARY，如上述说明。

8.5 数据元检索

该服务允许接口设备检索用于交换的行业间数据元。

——在选择应用之前，如果行业间数据对象存在，应依次直接或间接从历史字节(见 8.1.1)，初始数据串(见 8.1.2)，EF.ATR 和 EF.DIR(见 8.2.1.1)中检索。这些行业间数据对象按照 5.2.4中指定的 tag 分配方案进行说明。

——一旦应用被选择，应直接或间接从应用 DF 的管理数据(见 5.3.3)或当前 DF 中特定的 EF 中检索行业间数据对象。

- 行业间数据对象可能出现在任何文件的管理数据中(见 5.3.3)。
- 行业间数据元可以在封装模板(wrapper)内引用的文件中检索(见 8.5.1)。对未命名的 DF 或通过路径识别的 EF 的选择在 8.3 中定义。从选定的 EF 或 DF 中读取数据在 8.4 中定义。
- 行业间数据对象可以使用 GET DATA 命令(见 7.4.2)检索。

8.5.1 数据元的间接引用

元列表、tag 列表、头列表、扩展头列表和封装模板(wrapper)为间接引用字节串中数据元的行业间数据元,例如支持数据单元的 EF 的内容,完整的命令 APDU 的数据字段(见 8.4),签名的字节串(见 GB/T 16649.8)。这样的数据元指导卡如何解释命令的数据字段或如何构造响应的数据字段。

元列表——由 tag'5F41'引用,此行业间数据元表示需要检索的信息不是以数据对象的方式出现,而是在应用的控制之下。它仅在封装模板内使用。其结构和返回信息不在 GB/T 16649 范围内。

tag 列表——由 tag'5C'引用,此行业间数据元是 tag 字段的无定界串联。字节串由数据对象组成,数据对象的顺序与 tag 列表的顺序相同。

头列表——由 tag'5D'引用,此行业间数据元是成对的 tag 字段和长度字段的无定界串联。字节串由值字段组成,值字段的顺序与头列表的顺序相同。

扩展头列表——由 tag'4D'引用,此行业间数据元是成对的 tag 字段和长度字段的无定界串联。字节串构造方式如下:

——如果 tag 表明为简单编码,则 tag 字段和长度字段对由 tag 引用的数据代替。长度为零意味着字节串内包含完整的数据对象/数据元。非零长度表明要检索的字节的最大数,以及可能需要进行删节。

——如果 tag 表明为结构化编码,且长度不为零('80'除外),则后面的值字段为扩展头列表。如果 tag 表明为结构化编码,且长度为零,则可忽略。如果 tag 表明为结构化编码,且长度为'80',则字节串内包含完整的结构化的数据对象/完整的模板。

——卡应忽略扩展头列表中和目标结构不匹配的数据元。

字节串组成可能为下列情形之一:

——原始数据对象的值字段,可能需要根据指定的长度进行删节(情形 1);或

——原始数据对象,可能需要根据指定的长度进行删节,嵌套在相应的模板内,其长度遵守 BER-TLV 的规则(情形 2);

——如果长度设置为'80',则其将被实际长度代替。字节串内包含完整的结构化的数据对象/完整的模板。

字节串的编码(也就是数据对象或数据元)由适当的 INS 代码或其他命令参数指示,例如,它或者是适当的数据字段的编码(结构化的用来存放数据对象,简单型用来存放数据元),或者是用于 PERFORM SECURITY OPERATION 命令(见 GB/T 16649.8)的 tag'AC'、'BC'(见表 31)。

例如,下列扩展头列表引用 3 种简单数据对象。

简单型 T1	'00'	结构化的 tag T	L=4	简单型 T2	'00'	简单型 T3	L=5

简单型 T1	L1	值 1

简单型 T2	L2	值 2

简单型 T3	L3(≥5)	值 3

情形 1:字节串为数据元的串连。

值 1	值 2	值 3 的前 5 个字节

情形 2:字节串为数据对象的串连。

T1	L1	值 1	T	L=L2+9	T2	L2	值 2	T3	L=5	值 3 的前 5 个字节

封装——由 tag'63'引用,此行业间模板由两个数据对象组成。

——第一个数据对象为数据元列表(tag 为'5F41')、tag 列表(tag 为'5C')、头列表(tag 为'5D')或扩展头列表(tag 为'4D')。

——第二个数据对象是对 EF 的引用(tag 为'51',见 5.3.1.2)和/或一条或多条将要执行命令(tag 为'52')。如果是多条,则命令 APDU 将按照出现的顺序处理。

由 tag 列表等引用的数据对象或头列表等引用的数据元应包含在被引用的文件内,或者为对最后一条命令 APDU 的响应的(部分)数据字段。在封装内,仅给出一条间接引用。可能存在不只一个封装。

例如,下列封装模板由 tag 列表和一条可执行命令组成。

{'63'-L-{'5C'-L-(Tag1-Tag2-Tag3)}-{'52'-L-命令 APDU}}

8.6 卡发起的字节串

该服务允许卡发起字节串。

为了清晰,本条所讲的查询视为卡发送的字节串(或其中一部分),应答视为外界实体发送的响应(或其中一部分)。例如,一组完整的查询可以构成一条命令 APDU,一组完整的应答可以构成一条响应 APDU,从而允许从卡到接口设备,或从卡到卡之间的通讯服务,该通讯服务可能通过网络实施。

本条说明了下列 3 个特征:

——如何使用 SW1-SW2 表示卡想要发送字节串,并希望得到响应。

——接口设备如何使用 GET DATA 命令检索来自卡的查询,如何使用 PUT DATA 命令向卡传送应答(如果有)。

——字节串的格式。

8.6.1 卡触发

SW1-SW2 设置为'62XX',其中'XX'的取值'02'到'80',表示卡发出一条长度为'XX'字节的查询,接口设备将检索该查询,同时卡可能在期待该查询的响应。

SW1-SW2 设置为'64XX',其中'XX'的取值'02'到'80',表示卡终止了命令的执行。命令的执行是否能完成取决于是否能恢复长度为'XX'字节的查询,卡可能在期待该查询的响应。

如果 SW1-SW2 出现在历史字节内并取上述数值,则其含义如上所述。

如果用来传送应答的 PUT DATA 命令(见 7.4.3)执行失败,SW1-SW2 设置为'64XX',则

——'64XX'取值为'6402'到'6480'时,表示卡想要另外发送至少一条长度为'XX'的查询。

——'64XX'取值为'6401'时,表示卡正在等待立即响应。

8.6.2 查询和应答

为了从卡内检索长度为'XX'字节的查询,接口设备将发送 P1-P2 为'0000'、Le 为'XX'的 GET DATA 命令。

——SW1-SW2 设置为'62XX',其中'XX'的取值从'02'到'80',表示在外界处理卡发起的字节串之前,接口设备应再次检索一条长度为'XX'的查询并将其和已经检索到的查询串联。

——SW1-SW2 设置为'9000'意味着卡发起的字节串是完整的,它可能在外界进行处理。

为了向卡传送应答,接口设备将发送 P1-P2 为'0000'的 PUT DATA 命令。如果响应太长而不能由一条命令完成,则多条 PUT DATA 命令将被链接在一起(见 5.1.1.1)。每条 PUT DATA 命令传送一条应答,这些应答的串联即构成响应。

8.6.3 格式

卡发起的字节串的首字节的值用来指明以下格式。

——如果首字节的值为'FF',则接下来的字节将按照 ISO/IEC TR 9577[5]的规定编码为初始的协议标识符,字节串应遵守指定的协议。

——否则(即首字节的值不等于'FF'),卡发起的字节串及其响应将构成命令响应对。

除了正确使用 GET DATA、PUT DATA 命令及状态字节 SW1-SW2 之外,其他任何情形都和卡指示的传输协议相关。本条不对响应需求和可能存在的响应内容的实体做出假设。

附 录 A
（资料性附录）
对象标识符和标记分配方案示例

A.1 对象标识符

ISO 标准中，对象标识符的第 1 字节是‘28’，即十进制的 40（参见 ISO/IEC 8825-1）。其后跟随一个或多个字节序列；如果有多于一个字节，那么字节序列中最后一个字节的 bit8 置 0 并且之前的所有字节的 bit8 置 1。字节序列中所有字节的 bit7 至 bit1 的拼接是一个号码的编码。每个号码应该通过尽量少的字节编码，也就是说，值‘80’在字节序列中的第 1 个字节是非法的。第 1 个数字是标准的编号，如果存在第 2 个数字，它是一个多部分标准中的部分编号。

例子 1，{iso(1) standard(0) ic-cards(7816)}作为 ISO/IEC 7816 的参考。

——7816 等于‘1E88’，如 0001 1110 1000 1000，如 2 个 7 bit 的块：0111101 0001000。

——在每个字节的 bit8 插入适当的值后，第 1 个序列的编码就是 1011 1101 0000 1000，即‘BD08’。

数据元‘28 BD08’可用于标准分类的 AID（见 8.2.1.2）

AID=‘E8 28 BD08 0B XX ... XX’（ISO/IEC 7816-11 指定应用标识符扩展‘XX ... XX’）。

AID=‘E8 28 BD08 0F XX ... XX’（ISO/IEC 7816-15 指定应用标识符扩展‘XX ... XX’）。

例子 2，{iso(1) standard(0) e-auth(9798) part(5)}作为 ISO 9798-5[8] 的参考。第 1 个字节序列如下：

——9798 等于‘2646’，如 0010 0110 0100 0110，如 2 个 7 bit 的块：1001100 1000110。

——在每个字节的 bit8 插入适当的值后，第 1 个序列的编码就是 11001100 01000110，即‘CC46’。

数据元‘28 CC46 05 02’引用了 ISO/IEC 9798-5[8] 中的第 2 个机制，即 GQ2。这样的标识符可以携带在一个数据对象中（标记‘06’，通用类，参见 ISO/IEC 8825-1）。

DO={‘06 05 28 CC 46 05 02’}

例子 3，{iso(1) standard(0) mess(9992) part(2)} 作为 ISO 9992-2[10] 的引用。第 1 个字节序列如下：

——9992 等于‘2708’，如 0010 0111 0000 1000，如 2 个 7 bit 的块：1001110 0001000。

——在每个字节的 bit8 插入适当的值后，第 1 个序列的编码就是 1100 1110 0000 1000，即‘CE08’。

数据元是‘28 CE08 02’（第 2 个系列是‘02’）。其可在数据对象中携带。

DO={‘06 04 28 CE 08 02’}

A.2 标记分配方案

默认的标记分配方案示例

DO1={‘59 02 95 02’}

DO2={‘5F 24 03 97 03 31’}

DO1（标记‘59’，卡终止日期），编码 1995 年 2 月作为卡终止日期（参见 ISO/IEC 7816-6[4]）。

DO2（标记‘5F24’，应用终止日期），编码 1997 年 3 月 31 日作为应用终止日期。

兼容的标记分配方案示例

DO1={‘78 06’{‘06 04 28 CE 08 02’}}

DO2={‘5F 24 03 97 03 31’}

DO3={‘70 04’ {‘80 02 XX XX’}}

DO4={‘67 0A’ {‘5F 29 03 XX XX XX’} {‘81 02 XX XX’}}

DO1(标记'78',兼容标记分配授权)指出一个在 ISO 9992-2[10]定义的兼容的标记分配方案通过其对象标识符引用,如果 DO1 在初始化数据串(见 8.1.2)或 EF.ATR(见 8.2.1.1)中出现,该标记分配方案在整个卡中有效。如果 DO1 出现在一个 DF 的管理数据(见 5.3.3),该标记分配方案授权在该 DF 中出现。

DO2(标记'5F24',应用终止日期)编码 1997 年 3 月 31 日为应用终止日期。

DO3(标记'70',符合其包含的标记分配授权的行业间模板)包含数据对象,标记'80',在 ISO 9992-2[10]中定义;标记'70'的含义在 ISO 9992-2[10]也有定义。

DO4(标记'67',认证数据模板)包含交换的配置数据对象,标记'5F29',和一个定义在 ISO 9992-2[10]中的数据对象,标记'81';标记'67'的含义在 ISO/IEC 7816-4[4]定义。

另外一个兼容的标记分配方案示例

DO2={'5F 24 03 97 03 31'}

DO3={'70 0C'{'06 04 28 CE 08 02'} {'80 04 XX XX XX XX'}}

DO4={'67 06'{'5F 29 03 XX XX XX'}}

DO2(标记'5F24',应用终止日期)编码 1997 年 3 月 31 日为应用终止日期。

DO3(标记'70',符合其包含的对象标识符定义的行业间模板)包含一个数据对象,标记'06',该数据对象指定之后的数据对象,标记'80',在 ISO 9992-2[10]定义。标记'70'的含义也在 ISO 9992-2[10]定义。

DO4(标记'67',行业间认证数据模板)包含交换的配置数据对象,标记'5F29'。注意,由于这个选择不能传输行业间带标记'78'的数据对象,它不能包含 ISO 9992-2[10]中定义的数据对象。

共存的标记分配方案示例

DO1={'79 05' {'06 03 28 XX XX'}}

DO2={'7E 06'{'5F 24 03 97 03 31'}}

DO3={'70 06' {'XX XX XX XX XX XX'}}

DO1(标记'79',共存的标记分配授权)指出一个定义在通过'28'开头的对象标识符引用的标准(ISO 标准)中的共存的标记分配方案。这个方案中,DO1 必须:

——如果该标记分配授权在整个卡中有效,则 DO1 必须出现在初始化数据串(见 8.1.2)中或 EF.ATR(见 8.2.1.1)中,否则

——如果该标记分配授权在该 DF 中有效,则 DO1 必须出现在 DF 中的管理数据中(见 5.3.3)。

DO2(标记'7E')是一个嵌套行业间数据对象的行业间模板。注意行业间数据对象"应用终止日期"(标记'5F24')出现,编码 1997 年 3 月 31 日为应用终止日期。

DO3(标记'70',根据在模板'79'中指出的标记分配授权进行解释的行业间模板)只可根据对象标识符指定的标准解释。

附 录 B
（资料性附录）
安全报文传输示例

B.1 密码校验和

本章说明了 GB/T 16649.3 中定义的四种情形的命令-响应对中的安全报文传输（见第 6 章）与密码校验和（见 6.2.3.1）的使用方法。

在下面的例子中，记法 CLA * 表示在数据字段中使用安全报文传输：在 CLA 中（见 5.1.1），bit8、bit7、bit6 置为 000 且 bit4 置 1，或者 bit8、bit7、bit6 置为 011。

在下面的例子中，记法 CLA * * 表示 CLA 的 bit8、bit7、bit6 为 000，且 bit4、bit3 为 11，即用于认证的数据元的计算中应该包含命令头。

另外，命令头可封装在标记为'89'的数据对象中，即一个包含在用于认证的数据元计算中的 SM 数据对象。

在下面的例子中，记法 T * 表示标记字段中最后一个字节的 bit1 置为 1（奇标记数），即该 SM 数据对象应该包含在用于认证的数据元计算中。

——情形 1—无命令数据，无响应数据

非安全的命令—响应对如下所示：

命令头	命令体
CLA INS P1 P2	Absent

响应体	响应尾
Absent	SW1-SW2

——情形 1.a—未保护的状态字节

安全的命令 APDU 如下所示：

命令头	命令体
CLA * INS P1 P2	{New Lc field}-{New data field(=T -L-Cryptographic checksum)}

如果密码校验和的长度为 4 字节，则新的 Lc 字段置为'06'。

新数据字段=1 个数据对象={T-L-密码校验和}

产生密码校验和的数据（CLA * 的 bit3 置 1）= 1 个分组={CLA * * INS P1 P2 填充}

安全的响应 APDU 如下所示：

响应体	响应尾
Absent	SW1-SW2

——情形 1.b—受保护的状态字节

安全的命令 APDU 如下所示：

命令头	命令体
CLA * INS P1 P2	{New Lc field}-{New data field(=T -L-Cryptographic checksum)}-{New Le field(='00')}

新数据字段=1 个数据对象={T-L-密码校验和}

产生密码校验和的数据(CLA * 的 bit3 置 1)=1 个分组={CLA * * INS P1 P2 填充}

安全的响应 APDU 如下所示:

响应体	响应尾
New data field(={T* -L-SW1-SW2}-{T-L-Cryptographic checksum})	SW1-SW2

新数据字段=2 个数据对象= {T * -L-SW1-SW2}-{T-L -密码校验和}

产生密码校验和的数据=1 个分组={T * -L-SW1-SW2-填充}

——情形 2—无命令数据,有响应数据

非安全的命令—响应对如下所示:

命令头	命令体
CLA INS P1 P2	Le field

响应体	响应尾
Data field	SW1-SW2

安全的命令 APDU 如下所示:

命令头	命令体
CLA * INS P1 P2	New Lc field-New data field-{New Lc field(one or two bytes set to '00')}

新数据字段=2 个数据对象={T * -L-Le}-{T-L -密码校验和}

产生密码校验和的数据=

1 个分组={T * -L-Le-填充},如果 CLA * 的 bit3 置 0

2 个分组={CLA * * INS P1 P2 填充}-{T * -L-Le-填充},如果 CLA * 的 bit3 置 1

安全的响应 APDU 如下所示:

响应体	响应尾
New data field(={T * -L-Plain value}-{T * -L-SW1-SW2}-{T-L-Cryptographic checksum})	SW1-SW2

新数据字段=3 个数据对象=

{T * -L-无格式值}-{T * -L-SW1-SW2}-{T-L-密码校验和}

产生密码校验和的数据=1 个或多个分组={T * -L -无格式值-T * -L-SW1-SW2-填充}

——情形 3—有命令数据,没有响应数据

非安全的命令—响应对如下所示:

命令头	命令体
CLA INS P1 P2	Lc field-Data field

响应体	响应尾
Absent	SW1-SW2

——情形 3.a—未保护的状态字节

安全的命令 APDU 如下所示:

命令头	命令体
CLA * INS P1 P2	New Lc field-New data field

新数据字段=2 个数据对象={T * -L -无格式值}-{T-L -密码校验和}

产生密码校验和的数据=

1 个或多个分组＝{T＊-L -无格式值-填充}，如果 CLA＊的 bit3 置 0

2 个或多个分组＝{CLA ＊＊ INS P1 P2 填充}-{T＊-L -无格式值-填充}，如果 CLA＊的 bit3 置 1

安全的响应 APDU 如下所示：

响应体	响应尾
Absent	SW1-SW2

——情形 3.b—受保护的状态字节

安全的命令 APDU 如下所示：

命令头	命令体
CLA ＊ INS P1 P2	New Lc field-New data field-New Le field(＝'00')

新数据字段＝2 个数据对象＝{T＊-L -无格式值}-{T-L -密码校验和}

产生密码校验和的数据＝

1 个或多个分组＝{T＊-L -无格式值-填充}，如果 CLA＊的 bit3 置 0

2 个或多个分组＝{CLA ＊ ＊ INS P1 P2 填充}-{T＊-L -无格式值-填充}，如果 CLA＊的 bit3 置 1

安全的响应 APDU 如下所示：

响应体	响应尾
New data field(＝{T＊-L-SW1-SW2}-{T-L-Cryptographic checksum})	SW1-SW2

新数据字段＝2 个数据对象＝{T＊-L-SW1-SW2}-{T-L -密码校验和}

产生密码校验和的数据＝1 个分组＝{T＊-L-SW1-SW2 -填充}

——情形 4—有命令数据，有响应数据

非安全的命令—响应对如下所示：

命令头	命令体
CLA INS P1 P2	Lc field-Data field-Le field

响应体	响应尾
Data field	SW1-SW2

安全的命令 APDU 如下所示：

命令头	命令体
CLA ＊ INS P1 P2	New Lc field-New data field-New Le field(one or two bytes set to '00')

新数据字段＝3 个数据对象＝{T＊-L -无格式值}-{T＊-L-Le}-{T-L -密码校验和}

产生密码校验和的数据＝

1 个或多个分组＝{T＊-L -无格式值-T＊-L-Le -填充}，如果 CLA＊的 bit3 置 0

2 个或多个分组＝{CLA＊ ＊ INS P1 P2 填充}-{T＊-L -无格式值-T＊-L-Le -填充}，如果 CLA＊的 bit3 置 1

安全的响应 APDU 如下所示：

响应体	响应尾
New data field(＝{T＊-L-Plain value}-{T＊-L-SW1-SW2}-{T-L-Cryptographic checksum})	SW1-SW2

B.2 密文

带填充或者不带填充的密文的使用(见 6.2.2),出现在命令数据字段和响应数据字段中。与之前示例的无格式值数据对象不同,保持秘密性的数据对象应按如下形式使用:

——情形 a—未编码为 BER-TLV 的无格式值

数据字段={T-L -填充-内容指示符字节-密文}

产生密文的无格式值=1 个或多个分组=未编码为 BER-TLV 的无格式值,可能根据指示符字节填充

——情形 b—编码为 BER-TLV 的无格式值

数据字段={T-L-密文}

产生密文的无格式值=秘密字节串=BER-TLV 数据对象(依据算法及其操作模式的填充)

B.3 控制引用

控制引用(见 6.3.1 和 6.3.2)的使用如下:

命令数据字段={T-L-控制引用模板},

其中,控制引用模板={T-L-文件引用}-{T-L-密钥引用}

B.4 响应描述符

响应描述符(见 6.3.3)的使用如下:

命令数据字段={T-L-响应描述符},

其中,响应描述符={T(无格式值)-'00'-T(密码校验和)-'00'}

响应数据字段={T-L-无格式值}-{T-L-(密码校验和}

B.5 命令 ENVELOPE

ENVELOPE 命令(见 7.6.2)的使用如下:

命令数据字段={T-L-填充-内容指示字节-密文}

产生密文的无格式值=命令 APDU(以 CLA * INS P1 P2 开始),根据指示符字节填充

响应数据字段={T-L-填充-内容指示符字节-密文}

产生密文的无格式值=响应 APDU,根据指示符字节填充

B.6 安全报文和安全操作之间的协同

本章中应用下列符号和缩写术语。

CC 密码校验和

CG 密文

CLA ** 带 SM 指示的 CLA(bit8,7 和 6 置 000 且 bit4 和 3 置 11)

DS 数字签名

MSE 管理安全环境

PCI 填充内容指示字节

PSO 执行安全操作

SMC 安全模块卡

USC 用户智能卡

下面的示例说明了如何使用安全模块卡(SMC),产生发送给用户卡(USC)的安全的命令 APDU 和处理从用户卡接受的相对应的安全的响应 APDU,即产生和处理 SM 格式的数据字段。该示例阐述了

2 种方法之间的协同——通过安全操作实现的原子方法(参见 GB/T 16649.8)和通过安全报文实现的全局方法(见第 6 章)。

示例中假设 USC 和 SMC 已经完成相互之间的认证流程,如基于卡上可以验证的证书。该认证流程包含一个密钥传输或者密钥协商机制,因此在该流程后,USC 和 SMC 中存在 2 个可用的对称密钥:

——1 个用于密码校验和计算的对称会话密钥;

——2 个用于计算密文的对称会话密钥。

鉴别的流程在 USC 和 SMC 中初始化了一个或多个计数器。该示例中未说明 USC 和 SMC 如何维护和使用这些计数器。

SMC 中所有的命令—响应对都是 PSO 命令,未使用安全报文,但使用 SM 数据对象(和 MSC 命令设置的 SM 密钥)。

USC 中所有的命令—相应对使用安全报文,并且命令头包含在密码校验和的计算中,即 CLA 转换为 CLA **。

图 B.1 为产生一条安全的命令 APDU 的通常原则。

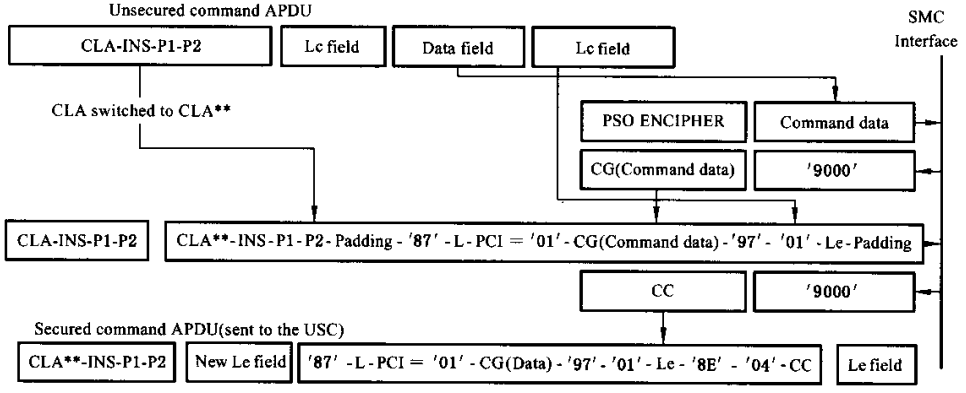

图 B.1 安全的命令 APDU 的产生

图 B.2 为处理一条安全的响应 APDU 的通常原则。

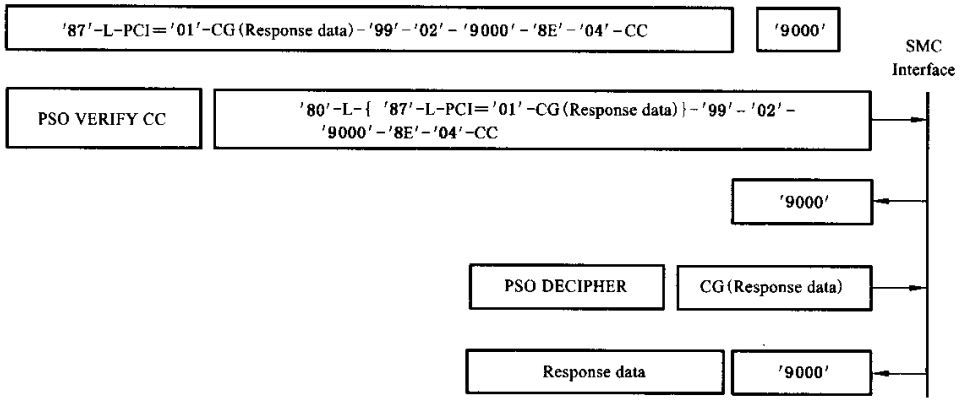

图 B.2 安全的响应 APDU 的处理

下面假设的情景解释了数字签名(DS)的计算中为什么秘密签名密钥的使用需要正确的口令。分三步执行该情景:

步骤 1——口令验证

1.1 发送到 SMC 的命令：MSE SET〈CT，{'83'-'01'-'81'}〉

——该例子中，用于密码校验和计算的会话密钥的引用为'81'。

SMC 响应：OK

1.2 发送到 SMC 的命令：MSE SET〈CCT，{'83'-'01'-'82'}〉

——该例子中，用于密码校验和计算的会话密钥的引用为'81'。

SMC 响应：OK

1.3 发送到 SMC 的命令：PSO ENCIPHER〈Password〉

SMC 响应：〈CG(Password)〉

1.4 发送到 SMC 的命令：PSO COMPUTE CC〈CLA **-INS-P1-P2-Padding-{'87'-L-PCI-CG(Password)}-{'97'-'01'-Le}-Padding〉

SMC 响应：〈CC〉

——现在接口设备构造安全的 VERIFY 命令 APDU。

1.5 发送到 USC 的命令：VERIFY〈{'87'-L-PCI='01'-CG(Password)}-{'97'-'01'-Le}-{'8E'-'04'-CC}〉

USC 响应：〈{'99'-'02'-SW1-SW2}-{'8E'-'04'-CC}〉

1.6 发送到 SMC 的命令：PSO VERIFY CC〈{'80'-'04'-('99'-'02'-SW1-SW2)}-{'8E'-'04'-CC}〉

SMC 响应：OK

步骤 2——哈希码的计算

2.1 发送到 SMC 的命令：PSO COMPUTE CC〈CLA **-INS-P1-P2-Padding-{'81'-L-({'90'-L-Intermediate Hash}-{'80'-L-Last block})}-{'97'-'01'-Le}-Padding〉

SMC 响应：〈CC〉

2.2 发送到 USC 的命令：PSO HASH〈{'81'-L1（=4+L2+L3)-({'90'-L2-Intermediate Hash}-{'80'-L3-Last block})}-{'8E'-'04'-CC}〉

——USC 将该哈希码作为内部结果保存用于后续的数字签名的计算。

USC 响应：〈{'99'-'02'-SW1-SW2}-{'8E'-'04'-CC}〉

2.3 发送到 SMC 的命令：PSO VERIFY CC〈{'80'-'04'-({'99'-'02'-SW1-SW2})}-{'8E'-'04'-CC}〉

SMC 响应：OK

步骤 3——数字签名计算

3.1 发送到 SMC 的命令：PSO COMPUTE CC〈CLA **-INS-P1-P2-Padding-{'97'-'01'-'00'}〉

SMC 响应：〈CC〉

3.2 发送到 USC 的命令：PSO COMPUTE DS〈{'97'-'01'-'00'}-{'8E'-'04'-CC}〉

USC 响应：〈'81'-L-DS-'8E'-'04'-CC〉

3.3 发送到 SMC 的命令：PSO VERIFY CC〈{'80'-L1（=2+L2)-('81'-L2-DS)}-{'8E'-'04'-CC}〉

SMC 响应：OK

附 录 C
（资料性附录）
GENERAL AUTHENTICATE 命令产生的 AUTHENTICATE 功能的示例

C.1 引言

两个或多个 GENERAL AUTHENTICATE 命令—响应对实现一个 AUTHENTICATE 功能。

——如果使用了命令链，那么该链中从第 1 个命令到倒数第 2 个命令的 CLA 置 0xx1 xxxx，最后一个命令的 CLA 置 0xx0 xxxx；其他 6 bit 应在链中保持常量（见 5.1.1.1）。

——INS P1 P2 置为‘86 00 00’或‘87 00 00’。

——Lc 字段的值由命令数据字段中的数据对象决定。根据是否期望响应数据字段，Le 字段设置为‘00’或者空缺。

本附录示例了 GENERAL AUTHENTICATE 命令的数据字段，它实现了诸如 ISO/IEC 9798-5[8] 所列出的机制，即使用零知识技术的机制。

——验证者知道某个公开问题，证明者知道该公开问题的秘密答案。

——作为零知识协议的结果，验证者确信证明者知道该公开问题的一个答案。但是，该答案仍然是一个秘密。

注：ISO/IEC 9798-5[8] 列出两种 GQ 技术。

——通过提供指数 v 为素数（如，$257=2^8+1$，$65\,537=2^{16}+1$ 或 $2^{36}+2^{13}+1$）的 RSA 密钥，GQ1 技术允许在不知道 RSA 签名值的情况下验证该 RSA 签名。如正在使用的 RSA 签名标准（如，参见 ISO/IEC 14888-2[16]）所述，格式机制将证明者的标识数据（一个模板）转换为公开数 G。相应的秘密数 Q 就是该标识数的 RSA 签名。证明者和验证者知道该公开的 RSA 密钥。GQ1 协议证明了证明者知道他的标识数据的该 RSA 的签名。

——通过提供模数 n，两个素数的乘积，GQ2 技术允许在不知他们或提供提供素数但不展现他们的情况下，验证这些素数。该方法使用了一个安全参数 $k>0$，并且前 m 个素数，即 m 个基础数据，使得 $k\times m$ 在 8 至 36 取值。每个公开的数是某个基础数据的平方：$G=g^2$。相应的秘密数 Q 是 G 的 2^{k+1} 次方根。如果至少存在一个基本数 g，使得 g 和 n 的雅可比符号为-1 且如果 n 与 1 模 4 同余，那么 GQ2 协议证明了 n 是合数且证明者知道其因子。

该协议代表性的交换了 3 个数据，即一个证据、一个质询和一个响应。

——证明者分两步进行：第 1 步中，证明者秘密地选择一个新的随机数并根据“证据规则”将它转换为一个证据；第 2 步中，收到质询后，证明者根据“响应规则”，结合新的随机数和秘密数产生该质询的响应，并删除该新的随机数。

——验证者根据“验证规则”从该质询和响应重构证据。

根据定义，包含 3 个数的 3 元组，即一个证据，一个质询和一个响应检验该“验证规则”。任何实体可以在“公开模式”下由任意质询和响应随机产生 3 元组。鉴定人或观察者不能区分“公开模式”下（不知道该秘密的实体）产生的随机的 3 元组，和“秘密模式”下（知道该秘密的实体）产生的随机 3 元组。

本附录示例了 3 种 AUTHENTICATE 功能。

——INTERNAL AUTHENTICATE 功能—外部的验证者认证卡中的证明者。

——EXTERNAL AUTHENTICATE 功能—卡中的验证者认证外部的验证者。

——MUTUAL AUTHENTICATE 功能—双方认证。

C.2 INTERNAL AUTHENTICATE 功能

如果第一个数据字段携带证据请求，即一个空的证据（‘80 00’）或一个空的认证码（‘84 00’），则该

功能是 INTERNAL AUTHENTICATE。

——基本协议(2 个命令-响应对)

来自卡的证据

命令数据字段	{'7C'-'02'-{'80'-'00'}}
响应数据字段	{'7C'-L1(=2+L2)-{'80'-L2-Witness}}

来自外界的质询和来自卡的响应

命令数据字段	{'7C'-L1(=4+L2)-{'81'-L2-Challenge}-{'82'-'00'}}
响应数据字段	{'7C'-L1(=2+L2)-{'82'-L2-Response}}

——提交的质询(2 个命令-响应对)

来自卡的证据

命令数据字段	{'7C'-L1(=4+L2)-{'83'-L2-Committed Challenge}-{'80'-'00'}}
响应数据字段	{'7C'-L1(=2+L2)-{'80'-L2-Witness}}

注:该提交的质询确保质询和证据的在选择时都是独立的。

来自外界的质询和来自卡的响应

命令数据字段	{'7C'-L1(=4+L2)-{'81'-L2-Challenge}-{'82'-'00'}}
响应数据字段	{'7C'-L1(=2+L2)-{'82'-L2-Response}if the challenge is correct Absent if the challenge is incorrect}

——数据字段认证的扩展(2 个命令-响应对)

卡片已经计算了之前交换的数据字段的哈希值:其结果是当前的哈希码。卡包含其用于获取认证码的证据数据对象并通过标记'84'传送。

来自卡的证据

命令数据字段	{'7C'-'04'-{'84'-'00'}}
响应数据字段	{'7C'-L1(=2+L2)-{'84'-L2-Authentication code}}

来自外界的质询和来自卡的响应

命令数据字段	{'7C'-L1(=4+L2)-{'81'-L2-Challenge}-{'82'-'00'}}
响应数据字段	{'7C'-L1(=2+L2)-{'82'-L2-Response}}

C.3 EXTERNAL AUTHENTICATE 功能

如果第 1 个数据字段携带质询请求,即一个空的质询('81 00')或一个空的提交的质询('83 00'),则该功能是 EXTERNAL AUTHENTICATE。

——基本协议(2 个命令-响应对)

来自外界的证据和来自卡的质询

命令数据字段	{'7C'-L1(=4+L2)-{'80'-L2-Witness}-{'81'-'00'}}
响应数据字段	{'7C'-L1(=2+L2)-{'81'-L2-Challenge}}

来自外界的响应和来自卡的证明

命令数据字段	{'7C'-L1(=2+L2)-{'82'-L2-Response}}
响应数据字段	Absent

——提交的质询(3 个命令-响应对)

来自卡的提交的质询

命令数据字段	{'7C'-'02'-{'83'-'00'}}
响应数据字段	{'7C'-L1(=4+L2)-{'83'-L2-Committed challenge}-{'80'-'00'}}

来自外界的证据和来自卡的质询

命令数据字段	{'7C'-L1(=4+L2)-{'80'-L2-Witness}-{'81'-'00'}}
响应数据字段	{'7C'-L1(=2+L2)-{'81'-L2-Challenge}}

来自外界的响应和来自卡的证明

命令数据字段	{'7C'-L1(=2+L2)-{'82'-L2-Response}}
响应数据字段	Absent

——数据字段认证的扩展(2 个命令-响应对)

证明者已经计算了之前交换的数据字段的哈希值:其结果是当前的哈希码。它包含其用于获取认证码的证据数据对象并通过标记'84'传送。

来自外界的证据和来自卡的质询

命令数据字段	{'7C'-L1(=4+L2)-{'84'-L2-Authentication code}-{'81'-'00'}}
响应数据字段	{'7C'-L1(=2+L2)-{'81'-L2-Challenge}}

来自外界的响应和来自卡的证明

命令数据字段	{'7C'-L1(=2+L2)-{'82'-L2-Response}}
响应数据字段	Absent

C.4 MUTUAL AUTHENTICATE 功能

如果第 1 个数据字段携带非空的数据对象,则该功能是 MUTUAL AUTHENTICATE;外部在响应中请求与命令数据字段中相同的数据对象。

——基本协议(3 个命令-响应对)

证据

命令数据字段	{'7C'-L1(=2+L2)-{'81'-L2-Witness}}
响应数据字段	{'7C'-L1(=2+L2)-{'81'-L2-Witness}}

质询

命令数据字段	{'7C'-L1(=2+L2)-{'81'-L2-Challenge}}
响应数据字段	{'7C'-L1(=2+L2)-{'81'-L2-Challenge}}

响应

命令数据字段	{'7C'-L1(=2+L2)-{'82'-L2-Response}}
响应数据字段	{'7C'-L1(=2+L2)-{'82'-L2-Response}} if the response is correct Absent if the response is incorrect

——提交的质询(4 个命令-响应对)

提交的质询

命令数据字段	{'7C'-L1(=2+L2)-{'83'-L2-Committed challenge}}
响应数据字段	{'7C'-L1(=2+L2)-{'83'-L2-Committed challenge}}

证据

命令数据字段	{'7C'-L1(=2+L2)-{'80'-L2-Witness}}
响应数据字段	{'7C'-L1(=2+L2)-{'80'-L2-Witness}}

质询

命令数据字段	{'7C'-L1(=2+L2)-{'81'-L2-Challenge}}
响应数据字段	{'7C'-L1(=2+L2)-{'81'-L2-Challenge}} if the challenge is correct Absent if the challenge is incorrect

响应

命令数据字段	{'7C'-L1(=2+L2)-{'82'-L2-Response}}
响应数据字段	{'7C'-L1(=2+L2)-{'82'-L2-Response}} if the response is correct Absent if the response is incorrect

——密钥协定的扩展(4 个命令-响应对)

一对指数数据元允许会话密钥的协定(参见 ISO/IEC 11770-3[14])。

第 1 个命令-响应对交换嵌入一个"指数"数据元的动态认证模板。在示例中,因为会话中之前没有

消息交换，初始的哈希码为空块。为获取当前的哈希码而包含了命令的数据字段，即第一个动态认证模板；为更新当前的哈希码而包含了响应数据字段，即第 2 个动态认证模板；当前的哈希码在双方实体中应该保持相同。最后为获取一个认证码(各个实体之间不同)，包含了一个证据数据对象(非零，未发送，各个实体间不同)。

第 2 个命令-响应对交换了嵌套认证码的带标记'84'的动态的认证模板。

指数

命令数据字段	{'7C'-L1(=2+L2)-{'85'-L2-Exponential}}
响应数据字段	{'7C'-L1(=2+L2)-{'85'-L2-Exponential}}

证据

命令数据字段	{'7C'-L1(=2+L2)-{'84'-L2-Authentication code}}
响应数据字段	{'7C'-L1(=2+L2)-{'84'-L2-Authentication code}}

质询

命令数据字段	{'7C'-L1(=2+L2)-{'81'-L2-Challenge}}
响应数据字段	{'7C'-L1(=2+L2)-{'81'-L2-Challenge}}

响应

命令数据字段	{'7C'-L1(=2+L2)-{'82'-L2-Response}}
响应数据字段	{'7C'-L1(=2+L2)-{'82'-L2-Response}} if the response is correct Absent if the response is incorrect

附 录 D
（资料性附录）
使用发行者标识号的应用标识符

D.1 背景信息

GB/T 16649.5 中，应用标识符中能够使用发行者标识号。本附录指出该 AID 的格式。

D.2 格式

在第 1 个字节的 bit8 到 bit5 置为‘0’到‘9’之间的 AID 中，第 1 个也可能是唯一的字段应该是某个根据 GB/T 15694.1 的发行者标识号。

注：GB/T 15694.1 中，一个发行者标识号可由奇数个从‘0’到‘9’的 4bit 组组成。然后，通过设置最后一个字节的 bit4 到 1 为 1111，它被映射到一个字节串中。

如果存在专用应用标识符扩展，那么一个设置为‘FF’的字节用于分离这 2 个字段。

图 D.1 示例了一个使用发行者标识号的 AID：它长达 6 字节。

发行者标识号（GB/T 15694.1） （2个或更多字节）	‘FF’	专有的应用标识符扩展 （PIX）

图 D.1 使用发行者标识号的 AID

参 考 文 献

[1] ISO 3166-1: 1997 Codes for the representation of names of countries and their subdivisions—Part 1:Country codes

[2] ISO/IEC 7810:2003 Identification cards—Physical characteristics

[3] ISO/IEC 7812-1:2000 Identification cards—Identification of issuers—Part 1:Numbering system

[4] ISO/IEC 7816 (all parts) Identification cards—Integrated circuit cards

[5] ISO/IEC TR 9577:1999 Information technology—Protocol identification in the network layer

[6] ISO/IEC 9796 (all parts) Information technology—Security techniques—Digital signature schemes giving message recovery

[7] ISO/IEC 9797 (all parts) Information technology—Security techniques—Message Authentication Codes (MACs)

[8] ISO/IEC 9798 (all parts) Information technology—Security techniques—Entity authentication

[9] ISO/IEC 9979:1999 Information technology—Security techniques—Procedures for the registration of cryptographic algorithms

[10] ISO 9992-2:1998 Financial transaction cards—Messages between the integrated circuit card and the card accepting device—Part 2: Functions messages (commands and responses) data elements and structures

[11] ISO/IEC 10116:1997 Information technology—Security techniques—Modes of operation for an n-bit block cipher

[12] ISO/IEC 10118 (all parts) Information technology—Security techniques—Hash-functions

[13] ISO/IEC 10536 (all parts) Identification cards—Contactless integrated circuit(s) cards—Closecoupled cards

[14] ISO/IEC 11770 (all parts) Information technology—Security techniques—Key management

[15] ISO/IEC 14443 (all parts) Identification cards—Contactless integrated circuit(s) cards—Proximity cards

[16] ISO/IEC 14888 (all parts) Information technology—Security techniques—Digital signatures with appendix

[17] ISO/IEC 15693 (all parts) Identification cards—Contactless integrated circuit(s) cards—Vicinity cards

[18] ISO/IEC 18033 (all parts) Information technology—Security techniques—Encryption algorithms

[19] IETF RFC 1738:1994 Uniform resource locators (URL)

[20] IETF RFC 2396:1998 Uniform resource locators (URL):General syntax