第二部分　行业间交换命令

目　　录

1　范围 .....
2　参考文件 .....
3　定义 .....
4　缩略语和记录 .....
5　基本组织结构 .....
5.1　数据结构 .....
5.2　卡的安全体系结构 .....
5.3　APDU 报文结构 .....
5.4　命令首标、数据字段和响应尾标用的编码约定 .....
5.5　逻辑信道 .....
5.6　安全报文交换 .....
6　基本的行业间命令 .....
6.1　READ BINARY 命令 .....
6.2　WRITE BINARY 命令 .....
6.3　UPDATE BINARY 命令 .....
6.4　ERASE BINARY 命令 .....
6.5　READ RECORD 命令 .....
6.6　WRITE RECORD 命令 .....
6.7　APPEND RECORD 命令 .....
6.8　UPDATE RECORD 命令 .....
6.9　GET DATA 命令 .....
6.10　PUT DATA 命令 .....
6.11　SELECT FILE 命令 .....
6.12　VERIFY 命令 .....
6.13　INTERNAL AUTHENTICATE 命令 .....
6.14　EXTERNAL AUTHENTICATE 命令 .....
6.15　GET CHALLENGE 命令 .....
6.16　MANAGE CHANNEL 命令 .....
7　面向传输的行业间命令 .....
7.1　GET RESPONSE 命令 .....
7.2　ENVELOPE 命令 .....
8　历史字节 .....
9　与应用无关的卡服务 .....

附录

A　通过 T＝0 传输 APDU 报文 .....
B　通过 T＝1 传输 APDU 报文 .....
C　记录指针管理 .....
D　使用 ANS.1 基本编码规则 .....
E　卡轮廓的举例 .....
F　使用的安全报文交换 .....

1 范围

本规范规定了:

——由接口设备至卡以及相反方向所发送的报文、命令和响应的内容;

——在复位应答期间卡所发送的历史字节的结构及内容;

——当处理交换用的行业间命令时，在接口处所看到的文件和数据的结构;

——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利的安全体系结构;

——安全报文交换的方法;

——访问卡所处理算法的方法。本标准不描述这些算法。

2 参考文件

ISO3166:1993	国家名称表示的代码
ISO/IEC7812—1:1993	识别卡-发行者的标识-第 1 部分:编号系统
ISO/IEC7816—3:1997	识别卡-带触点的集成电路卡-第 3 部分:信号和传输协议
ISO/IEC7816—5:1994	识别卡-带触点的集成电路卡-第 5 部分:应用标识符的编号系统和登记规程
ISO/IEC7816—6	识别卡-带触点的集成电路卡-第 6 部分:行业间数据元
ISO/IEC8825:1990	信息技术-开放系统互连-抽象语法记法 1(ASN.1)的基本编码规则
ISO/IEC9796:1991	信息技术-安全技术-给出报文恢复的数字签名方案
ISO/IEC9797:1994	信息技术-安全技术-使用利用块密码算法的“密码检验”函数的数据完整性机制
ISO/IEC9979:1991	数据密码技术-密码算法登记规程
ISO/IEC10116—1:1994	信息技术-安全技术- n 比特块密码算法的操作方式
ISO/IEC10118—1:1994	信息技术-安全技术-散列函数-第 1 部分:概述
ISO/IEC10118—2	信息技术-安全技术-散列函数-第 2 部分:使用 n 比特块密码算法的散列函数

3 定义

下列定义适用于本规范。

3.1 复位应答文件 Answer-to Reset file

表示卡操作特性的基本文件。

3.2 命令响应对 Command-response pair

两种报文的集合:命令后面紧跟着响应。

3.3 数据单元 data unit

可以无二义性地被引用的最少位集合。

3.4 数据元 data element

在接口处所看到的信息，为它定义了名称、逻辑内容描述、格式和编码。

3.5 数据对象 data object

在接口处所看到的信息，它由标签、长度和值(即，数据元)组成。在本部分规范中，数据对象称之为 BER—TLV、压缩 TLV 和简单 TLV 数据单元。

3.6 专用文件 dedicated file

包含文件控制信息和任选地供分配用的存储器的文件。它可以是 EFs 和/或 DFs 的父辈。

3.7 DF 名称 DF name

唯一地标识了卡内专用文件的字节串。

3.8 目录文件 directory file

ISO/IEC7816 第 5 部分定义的基本文件。

3.9 基本文件 elementary file

共享同一文件标识符的数据单元或记录的集合。它不可能是另一文件的父辈。

3.10 文件控制参数 file control parameters

文件的逻辑、结构和安全的属性。

3.11 文件标识符 file identifier

用来寻址文件的 2 字节二进制值。

3.12 文件管理数据 file management data

除文件控制参数(例如，有效日期，应用标号)外，关于文件的任何信息。

3.13 内部基本文件 internal elementary file

用来存储由卡所解释数据的基本文件。

3.14 主文件 master file

表示文件结构根的强制性唯一专用文件。

3.15 报文 message

由接口设备向卡所发送的字节串，反之亦然，但不包括在 ISO/IEC7816 第 3 部分定义的面向传输的字符。

3.16 父辈文件 parent file

在分级结构范围内，直接在某一给定文件之前的专用文件。

3.17 口令 password

应用可以要求的数据，通过其用户将它呈现给卡。

3.18 路径 path

文件标识符的并置，而无需定界。如果路径以主文件的标识符开始，则它是一条绝对的路径。

3.19 提供者 provider

具有或曾获得在卡内建立专用文件权利的管理机构。

3.20 记录 record

可以由卡处理为一整体的并且可由记录号或记录标识符所引用的字节串。

3.21 记录标识符 record identifier

与记录相关的值，可用来引用那个记录。在一个基本文件内几个记录可以具有的相同标识符。

3.22 记录号 record number

分配给每个记录的顺序号，它唯一地标识其基本文件内的记录。

3.23 工作的基本文件 working elementary file

用来存储不由卡所解释数据的基本文件。

4 缩略语和记号

下列缩略语适用于本部分规范。

APDU	应用协议数据单元
ATR	复位应答
BER	ASN.1 的基本编码规则(见附录 D)
CLA	类别字节
DIR	目录
DF	专用文件
EF	基本文件
FCI	文件控制信息
FCP	文件控制参数
FMD	文件管理数据
INS	指令字节
MF	主文件
P1—P2	参数字节
PTS	协议类型选择
RFU	保留供将来使用
SM	安全报文交换
SW1—SW2	状态字节
TLV	标记、长度、值
TPDU	传输协议数据单元

下列记法适用于本部分规范。

“0”至“9”和“A”至“F”	16 个十六进制数字
(B_1)	字节(B_1)的值
$B_1 \| B_2$	字节 B_1(最高有效字节)和 B_2(最低有效字节)的并置
($B_1 \| B_2$)	字节 B_1 和 B_2 并置的值
#	编号

5 基本组织结构

5.1 数据结构

本条包含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逻辑结构的信息。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本部分规范范围内。

5.1.1 文件组织结构

本部分规范支持下列两种文件。

——专用文件(DF)。

——基本文件(EF)。

卡内数据的逻辑组织结构由下列专用文件的结构化分级组成。

——在根处的 DF 称作主文件(MF)。该 MF 是必备的。

——其他 DF 是任选的。

定义了下列两种类型的 EF。

——内部 EF——那些 EF 预期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——那些 EF 预期用于不由卡所解释的数据，即，仅仅由外界待使用的数据。

图 1 示出了卡内逻辑文件组织结构的举例。

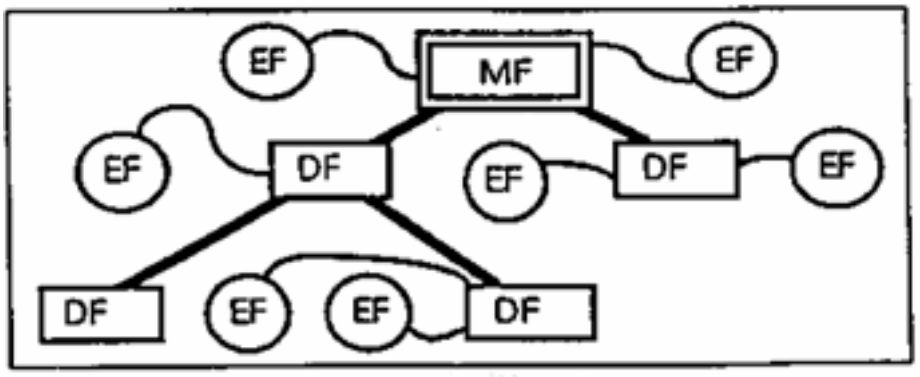

图 1　逻辑文件组织结构(举例)

5.1.2　文件引用方法

当文件不能被默认地选择时，应有可能至少通过下列方法之一来选择它。

——通过文件标识符引用——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(见通过路径的引用)。为了通过文件标识符来选择无二义性的任何文件，直接在给定 DF 下的所有 EF 和 DF 都应具有不同的文件标识符。

——通过路径引用——任何文件都可以通过路径来引用(文件标识符的并置)。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，路径由连续父辈 DFs(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。路径允许从 MF 或当前 DF 中无二义性地选择任何文件。

——通过短 EF 标识符引用——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 引用了当前选择的 EF。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 FILE 命令中)。

——通过 DF 名称引用——任何 DF 都可以通过按 1 至 16 个字节编码的 DF 名称来引用。为了通过 DF 名称进行无二义性的选择(例如，当借助 ISO/IEC7816 第 5 部分定义的应用标识符选择时)，每个 DF 名称应在给定的卡内是唯一的。

5.1.3　基本文件结构

定义了下列 EF 的结构。

——透明结构——在接口处 EF 可看作为一序列数据单元。

——记录结构——在接口处 EF 可看作为一序列各自可标识的记录。

为按记录构成的 EF 定义了下列属性。

——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)。

为了构造 EF，卡至少应支持下列四种方法之一。

——透明 EF。

——带有固定长度记录的线性 EF。

——带有可变长度记录的线性文件。

——带有固定长度记录的循环 EF。

图 2 示出了这四种 EF 结构

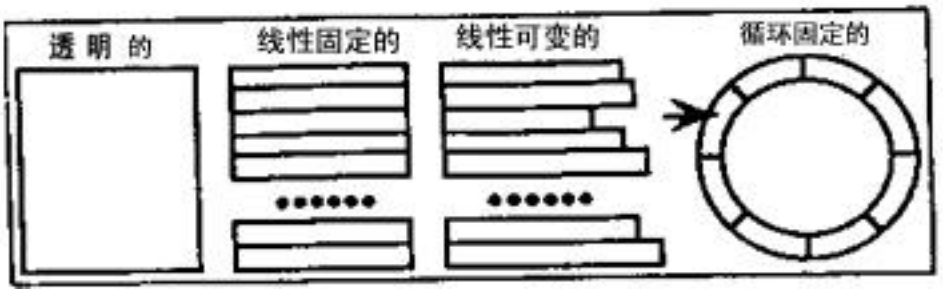

图 2　EF 结构

注:图上的箭头引用了最当前写的记录。

5.1.4　数据引用方法

数据可以作为记录、数据单元或数据对象加以引用。数据可被认为是存储在单个连续序列记录(在记录结构的 EF 内)或者是存储在单个连续序列数据单元(在透明结构的 EF 内)。引用超出 EF 的记录或数据单元是一次差错。

数据引用方法、记录编号方法和数据单元长度都是与 EF 有关的特征。卡能在 ATR、ATR 文件和任何文件控制信息中提供指示。当卡在几个地方提供了指示时，对给定 EF 有效的指示就是在从 MF 至那个 EF 的路径范围内最接近那个 EF 的一个指示。

5.1.4.1　记录引用

在每个记录结构的 EF 内，每个记录可以通过记录标识符和/或记录号来引用。记录标识符和记录号都是带有值的在从‘01’至‘FE’范围内无符号 8 比特整数。值‘00’被保留用于特定目的。值‘FF’为 RFU。

通过记录标识符引用应引起对记录指针的管理。卡的复位、选择文件和运载有效短 EF 标识符的任何命令都能影响记录指针。通过记录号引用应不影响记录指针。

——通过记录标识符引用——每个记录标识符由应用来提供。如果记录是在报文的数据字段中的简单 TLV 数据对象(见本部分规范 5.4.4)，则记录标识符是数据对象的第 1 个字节。在记录结构的 EF 内，记录可以具有相同记录标识符，在此情况下，在记录中所包含的数据可以用来辨别这些记录。

每次使用记录标识符进行引用，一个指针应指定目标记录的逻辑位置：第 1 个或最后一个出现，下一个或先前一个出现都与记录指针有关。

——在每个线性结构的 EF 内，当写入或添加时，逻辑位置应有序地被分配，即按建立的次序。因此，第 1 个建立的记录是在第 1 个逻辑位置中。

——在每个循环结构的 EF 内，逻辑位置应按相反的次序来分配，即，最当前建立的记录是在第 1 个逻辑位置中。

为了线性结构和循环结构，定义下列附加规则。

——第 1 个出现应是带有规定标识符的记录，并是在第 1 个逻辑位置中；最后一个出现应是带有规定标识符的记录，并且是在最后一个逻辑位置中。

——当不存在当前记录时，下一个出现应等价于第 1 个出现;先前一个出现应等价于最后一个出现。

——当存在当前记录时，下一个出现应是带有规定标识符的最近记录，但是在比当前记录更大的逻辑位置中；先前一个出现应是带有规定标识符的最近记录，但是在比当前记录更小的逻辑位置中。

——值‘00’应按编号顺序表示第 1 个、下一个或先前一个记录，但与记录标识符无关。

——通过记录号引用——在每个记录结构的 EF 内，记录号是唯一的和顺序的。

——在每个线性结构的 EF 内，当写入或添加时，记录号应有序地被分配，即按建立的次序。因此，第 1 个记录(记录号 1，#1)是第一个创建的的记录。

——在每个循环结构的 EF 内，记录号应按相反的次序来分配，即，第 1 个记录(记录号 1，#1)是最近建立的记录。

为了线性结构和循环结构，定义了下列附加规则。

——值‘00’应表示当前记录，即，通过记录指针所固定的那个记录。

5.1.4.2 数据单元引用

在每个透明结构的 EF 内，每个数据单元可以通过偏移量(例如，在 READ BINARY 命令中，见本部分规范 6.1)来引用。它是一个无符号整数，按照相应命令中的选项，它被限制在 8 位或 15 位。对于 EF 的第 1 个数据单元值为 0，对于每个后续数据单元，偏移增加 1。

通过默认，即，如果卡没有给出指示，则数据单元的长度为 1 个字节。

注：

1)记录结构的 EF 可以支持数据单元引用，在它支持的情况下，数据单元可以包含有结构化信息以及数据，例如，线性结构中的记录号。

2)在记录结构的 EF 内，数据单元引用可以不提供预期结果，因为在 EF 中的记录存储次序未知，例如在循环结构中的存储次序。

5.1.4.3 数据对象引用

每个数据对象(本部分规范 5.4.4 定义的)是以引用它的标记起头的。标记在 ISO/IEC7816 的本部分和其他部分进行规定。

5.1.5 文件控制信息

文件控制信息(FCI)是可用于响应 SELECT FILE 命令的数据字节串。对于任何文件，文件控制信息都可以呈现。

当文件控制信息编码为 BER—TLV 数据对象时，表 1 引入了预期用来运送文件控制信息的三种样板。

——FCP 样板预期用来运送文件控制参数(FCP)，即，在表 2 中定义的任何 BER—TLV 数据对象。

——FMD 样板预期用来运送文件管理数据(FMD)，即在本部分规范或本规范其他部分中规定的 BER—TLV 数据对象(例如，第 5 部分定义的应用标号以及第 6 部分定义的应用有效日期)。

——FCI 样板预期用来运送文件控制参数和文件管理数据。

表 1 与 FCI 相关的样板

标记	值
‘62’	文件控制参数(FCP 样板)
‘64’	文件管理数据(FMD 样板)
‘6F’	文件控制信息(FVI 样板)

三种样板可以根据选择“SELECT FILE 命令”中的选项(见表 59)进行检索。如果 FCP 或 FMD 选项被置位，则使用相应的样板是强制性的。如果 FCI 选项被设置，则使用 FCI 样板是任选的。

在应用的控制下，文件控制信息的一部分可以附加地存在于工作的 EF 中，并且可按照标签‘87’加以引用。对于编码的这种 EF 的文件控制信息，使用 FCP 或 FCI 样板是必备的。

不按照本部分规范编码的文件控制信息可以引入如下。

——‘00’或大于‘9F’的任何值——编码的后续字节串是专有的。

——标记=‘53’——数据对象的值字段由未按 TLV 编码的自由选定的数据组成。

——标记=‘73’——数据对象的值字段由自由选定的 BER—TLV 数据对象组成。

表 2　文件控制参数

标记	L	值	适用于
‘80’	2	在文件中的数据字节数，不包括结构信息	透明 EFs
‘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
‘82’	1 2 3 或 4	文件描述符字节(见表 3) 文件描述符字节后面紧跟着数据编码字节(见表 86) 文件描述符字节后面紧跟着数据编码字节和最大记录长度	任何文件 任何文件 带有记录结构的 EFs
‘83’	2	文件标识符	任何文件
‘84’	1～16	DF 名称	DFs
‘85’	变量	专有信息	任何文件
‘86’	变量	安全属性编码超出本规范本部分的范围	任何文件
‘87’	2	包含扩充 FCI 的 EF 标识符	任何文件
‘88’～‘9E’		RFU	
‘9FXY’		RFU	

表 3　文件描述符字节

b8	b7	b6	b5	b4	b3	b2	b1	含　义
0	×	-	-	-	-	-	-	文件可访问性
0	0	-	-	-	-	-	-	—不可共享的文件
0	1	-	-	-	-	-	-	—可共享的文件
0	-	×	×	×	-	-	-	文件类型
0	-	0	0	0	-	-	-	—工作的 EF
0	-	0	0	1	-	-	-	—内部的 EF
0	-	0	1	0	-	-	-	—保留供 EFs 的专有类型用
0	-	0	1	1	-	-	-	—保留供 EFs 的专有类型用
0	-	1	0	0	-	-	-	—保留供 EFs 的专有类型用
0	-	1	0	1	-	-	-	—保留供 EFs 的专有类型用
0	-	1	1	0	-	-	-	—保留供 EFs 的专有类型用
0	-	1	1	1	-	-	-	—DF
0	-	-	-	-	×	×	×	EF 结构
0	-	-	-	-	0	0	0	—没有信息被给出
0	-	-	-	-	0	0	1	—透明
0	-	-	-	-	0	1	0	—线性固定，没有进一步的信息
0	-	-	-	-	0	1	1	—线性固定，简单 TLV
0	-	-	-	-	1	0	0	—线性可变，没有进一步的信息
0	-	-	-	-	1	0	1	—线性可变，简单 TLV
0	-	-	-	-	1	1	0	—循环，没有进一步的信息
0	-	-	-	-	1	1	1	—循环，简单 TLV
1	×	×	×	×	×	×	×	RFU
“可共享”意味着至少支持在不同逻辑信道上的当前访问。								

5．2　卡的安全体系结构

本条描述下列特征：

——安全状态；

——安全属性;

——安全机制。

将安全属性与安全状态相比较，以执行命令和/或访问文件。

5.2.1 安全状态

安全状态表示完成下列动作后所获得的可能的当前状态:

——复位应答(ATR)和可能的协议类型选择(PTS)和/或;

——单个命令或一序列命令，可能执行的认证规程。

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，

——通过证明了解口令(例如，使用一个 VERIFY 命令);

——通过证明了解密钥(例如，使用“GET CHALLENGE”命令后面紧跟着“EXTERNAL AUTHENTICATE”命令);

——通过安全报文交换(例如，报文鉴别)。

考虑了三种安全状态:

——全局安全状态——它可以通过完成与 MF 相关的鉴别规程进行修改(例如，通过连接到 MF 的口令或密钥的实体鉴别);

——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，通过连接到特定 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(见本部分规范 6.10.2);这种修改只与鉴别规程所属的应用相关;

——命令特定安全状态——仅在执行涉及使用安全报文交换(见本部分规范 5.6)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。

如果逻辑信道的概念适用，则特定安全状态可以依赖于逻辑信道(见本部分规范 5.5.1)。

5.2.2 安全属性

当安全属性存在时，它定义了允许的动作以及完成这种动作要执行的规程。

安全属性可以与每个文件相关，并且安排为了允许对文件进行操作而应该满足的安全条件。文件的安全属性依赖于:

——它的种类(DF 或 EF);

——在它的文件控制信息中的和/或在其父辈文件的文件控制信息中的任选参数。

注:安全状态也可以与其他对象(例如，密钥)相关。

5.2.3 安全机制

本规范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)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被登录到内部 EF 中。

5.3 APDU 报文结构

应用协议中的一个步骤由发送命令、接收实体处理它以及发回的响应组成。因此，特定的响应对应于特定的命令，称作为命令响应对。

应用协议数据单元(APDU)可包含有命令报文或响应报文，它从接口设备发送到卡，或者相反地由卡发送到接口设备。

在命令响应对中，命令报文和响应报文都可以包含有数据，于是引起了由表 4 概括的四种情况。

表 4　命令响应对内的数据

情况	命令数据	期望的响应数据
1	无数据	无数据
2	无数据	有数据
3	有数据	无数据
4	有数据	有数据

5.3.1 命令 APDU

如图 3 所示(也见表 6)，本规范本部分所定义的命令 APDU 由下列内容组成：

——必备的 4 字节首标(CLA　INS P1　P2)；

——有条件的可变长度主体。

首标	主体
CLA INS P1 P2	[L_c 字段][数据字段][L_e 字段]

图 3　命令 APDU 结构

在命令 APDU 的数据字段中呈现的字节数用 L_c 来表示。

在响应 APDU 的数据字段中期望的字节最大数用 L_e(期望数据的长度)来表示。当 L_e 字段只包含 0 时，则要求有效数据字节的最大数。

图 4 按照表 4 定义的 4 种情况示出了命令 APDU 的 4 种结构。

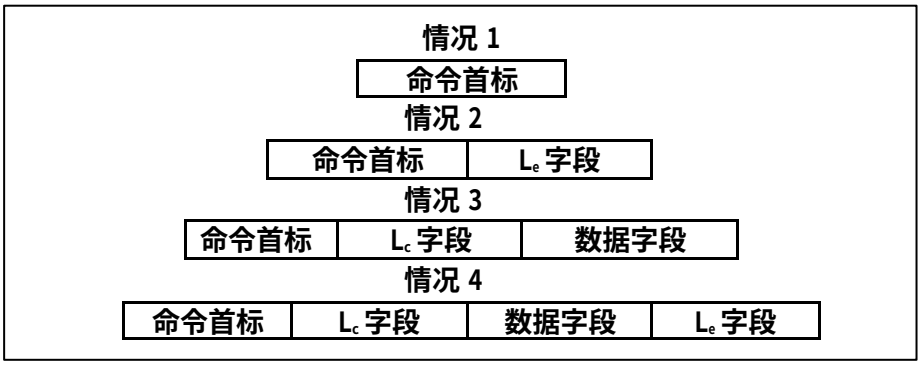

图 4　命令 APDU 的 4 种结构

在情况 1 时，长度为空，因此 L_c 字段和数据字段都为空。长度 L_e 也为空;因此，L_e 字段为空。从而，主体为空。

在情况 2 时，长度 L_c 为空;因此，L_c 字段和数据字段都为空。长度 L_e 不为空;因此，L_e 字段存在。从而，主体由 L_e 字段组成。

在情况 3 时，长度 L_c 不为空;因此，L_c 字段存在，并且数据字段由 L_c 后续字节组成。长度 L_e 为空;因此，L_e 字段为空。从而，主体由 L_c 字段后紧跟着数据字段组成。

在情况 4 时，长度 L_c 不为空;因此，L_c 字段存在，并且数据字段由 L_c 后续字节组成。长度 L_c 也不为空;因此，L_c 字段也存在。从而主体由 L_c 字段后紧跟着数据字段和 L_c 字段组成。

5.3.2 命令主体用的解码约定

在情况 1 时，命令 APDU 的主体为空。这种命令 APDU 未运载长度字段。

在情况 2、3 和 4 时，命令 APDU 的主体由 B_1 至 B_L 所表示的 L 字节组成，如图 5 所示。这种主体运载了 1 或 2 长度字段;B_1 是第 1 个长度字段的一部分。

命令主体

$B_1 \cdots\cdots B_L$(L 字节)

图 5　不空的主体

在卡能力(见本部分规范 8.3.6)中，在命令 APDU 内，卡说明了 L_c 字段和 L_e 字段既可为短的(一个字节、默认值)，也可为扩充的(显式语句)。

因此，情况 2、3 和 4 既可为短的(一个字节用于每个长度字段)也可为扩充的(B_1 的值为‘00’，并且每个长度值都按 2 个其他字节进行编码)。

表 5 示出了按照表 4 和图 4 中定义的四种情况及可能的 L_c L_e 扩展的命令 APDU 的解码。

表 5　命令 APDU 的解码

条　件		情　况	
L=0	—	1	
L=1	—	短的 2	(2S)
L=1＋(B_1);(B_1)≠0;	—	短的 3	(3S)
L=2＋(B_1);(B_1)≠0;	—	短的 4	(4S)
L=3; (B_1)=0;	—	扩充的 2	(2E)
L=3＋(B_2 ‖ B_3);(B_1=0;(B_2 ‖ B_3)≠0		扩充的 3	(3E)
L=5＋(B_2 ‖ B_3);(B_1)=0;(B_2 ‖ B_3)≠0		扩充的 4	(4E)

任何其他命令 APDU 为无效的。

L_e 用的解码约定:

如果 L_e 的值不为全空而按 1 个或 2 个字节进行编码，则 L_E 的值等于该字节的值，它位于从 1 至 255(或 65 535)的范围内;所有这些位的空值意味着 L_e 的最大值为 256(或 65 536)。

前 4 种情况适用于所有卡。

情况 1——L=0;主体为空。

- 没有字节用于值为 0 的 L_c 。
- 没有数据字节存在。
- 没有字节用于值为 0 的 L_e。

情况 2S——L=1。

- 没有字节用于值为 0 的 L_c。
- 没有数据字节存在。
- B_1 编码值从 1 至 255 的 L_e。

情况 3S——L=1+(B_1)，并且(B_1)≠0。

- B_1 编码了值从 1 至 255 的 L_c(≠0)。
- B_2～B_L 都是数据字段中的 L_c 字节。
- 没有字节用于值为 0 的 L_e。

情况 4S——L=2＋(B_1)，并且(B_1)≠0。

- B_1 编码了值从 1 至 255 的 L_c(≠0)。
- B_2～B_{L-1} 都是数据字段中的 L_c 字节。
- B_L 编码了从 1 至 256 的 L_e。

后 3 种情况也适用于指示扩充 L_c 和 L_e(见本部分规范 8.3.6，卡能力)的卡。

情况 2E——L=3，并且(B_1)=0

- 没有字节用于值为 0 的 L_c。

- 没有数据字节存在。
- L_e 字段由 3 个字节组成，其中 B_2 和 B_3 编码了值从 1 至 65 536 的 L_e。

情况 3E——3＋(B_2 ‖ B_3)，(B_1)=0，并且(B_2 ‖ B_3)≠0。

- L_c 字段由前 3 个字节组成，其中，B_2 和 B_3 编码了值从 1 至 65 535 的 L_c(≠0)。
- B_4 至 B_L 都是数据字段中的 L_c 字节。
- 没有字节用于值为 0 的 L_e。

情况 4E——L=5＋(B_2 ‖ B_3)，(B_1)=0，并且(B_2 ‖ B_3)≠0。

- L_c 字段由前 3 个字节组成，其中，B_2 和 B_3 编码了值从 1 至 65 535 的 L_c(≠0)。
- B_4 至 B_{L-2} 都是数据字段中的 L_c 字节。
- L_e 字段由最后的 2 个字节 B_{L-1} 和 B_L 组成;它们编码了值从 1 至 65 536 的 L_e。

对于本规范本部分定义的每个传输协议，附属到本部分的附录(每个协议一个)规定了先前 7 种情况中的每一种用的运输 APDU 的命令响应对。

5.3.3 响应 APDU

如图 6 所示(也见表 7)，本规范本部分定义的响应 APDU 由下列内容组成：

——有条件的可变长度主体;

——必备的 2 字节尾标(SW1 SW2)。

主体	尾标
［数据字段］	SW1 SW2

图 6 响应 APDU 结构

在响应 APDU 的数据字段中呈现的字节数用 L_r 来表示。

尾标编码了处理“命令响应对”之后的接收实体的状态。

注:如果该命令被放弃，则响应 APDU 是一个尾标，它按 2 个状态字节来编码差错条件。

5.4 命令首标、数据字段和响应尾标用的编码约定

表 6 示出了命令 APDU 的内容

表 6 命令 APDU 内容

代码	名称	长度	描 述
CLA	类别	1	指令的类别
INS	指令	1	指令代码
P1	参数 1	1	指令参数 1
P2	参数 2	1	指令参数 2
L_c 字段	长度	变量 1 或 3	在命令的数据字段中呈现的字节数
数据字段	数据	变量=L_c	在命令的数据字段中发送的字节串
L_e 字段	长度	变量≤3	在向命令响应的数据字段中期望的字节最大数

表 7 示出了响应 APDU 的内容

表 7 响应 APDU 内容

代码	名称	长度	描 述
数据字段	数据	变量=L_r	在响应的数据字段中收到的字节串
SW1	状态字节 1	1	命令处理状态
SW2	状态字节 2	1	命令处理受限字符

后续条规定了类别字节、指令字节、参数字节、数据字段字节和状态字节用的编码约定。

除非另有规定，在这些字节中，RFU 的比特都编码为 0，并且 RFU 字节也都编码为‘00’。

5.4.1 类别字节

按照与表 9 一起使用的表 8，命令中的类别字节 CLA 用来指出：

——命令和响应在什么程度上应遵循本规范本部分;

——当适用(见表 9)时，安全报文交换的格式及逻辑信道号。

表 8　CLA 的编码及含义

值	含　　义
‘0X’	按照本规范定义　命令和响应的结构和编码(对于编码‘X’　见表 9)。
‘10’～‘7F’	RFU
‘8X’‘9X’	按本规范定义命令和响应的结构。‘X’除外(对于‘X’编码　见表 9) 命令和响应的编码及含义是专有的。
‘AX’	除非通过应用上下文另有规定 按照本规范定义命令和响应的编码及含义(对于编码‘X’　见表 9)。
‘B0’～‘CF’	按照本规范本部分 命令和响应的结构
‘D0’～‘FE’	命令和响应的专有结构
‘FF’	保留供 PTS 用

表 9　当 CLA=‘0X’、‘8X’、‘9X’或‘AX’时，半字节‘X’的编码及含义

b4	b3	b2	b1	含　　义
×	×	—	—	安全报文(SM)格式
0	×	—	—	▪没有 SM 或 SM 不按照 5.6
0	0	—	—	——没有 SM 或没有 SM 指示
0	1	—	—	—专有的 SM 格式
1	X	—	—	▪安全报文交换按照 5.6
1	0	—	—	—不被鉴别的命令首标
1	1	—	—	—被鉴别的命令首标 (关于命令首标的用法见 5.6.3.1)
—	—	×	×	逻辑信道号(按照 5.5) (当不使用逻辑信道号时或当逻辑信道#0 被选择时，b2 b1=00)

5.4.2　指令字节

命令中的指令字节 INS 应予以编码，以便允许使用本规范第 3 部分定义的任何协议进行传输。表 10 示出了必然无效的 ISN 代码

表 10　无效的 INS 代码

b8	b7	b6	b5	b4	b3	b2	b1	含　　义
×	×	×	×	×	×	×	1	—奇数值
0	1	1	0	×	×	×	×	—‘6X’
1	0	0	1	×	×	×	×	—‘9X’

表 11 示出了本规范定义的 INS 代码。当 CLA 的值位于从‘00’至‘7F’的范围内时，INS 代码的其他值有待 ISO/IEC JTC1 SC17 进行分配。

表 11　本规范定义的 INS 代码

值	命令名称	条款
‘0E’	ERASE BINARY	6.4
‘20’	VERIFY	6.12
‘70’	MANAGE CHANNEL	6.16
‘82’	EXTERNAL AUTHENTICATE	6.14
‘84’	GET CHALLENGE	6.15
‘88’	INTERNAL AUTHENTICATE	6.13
‘A4’	SELECT FILE	6.11
‘B0’	READ BINARY	6.1
‘C0’	GET RESPONSE	7.1

‘C2’	ENVELOPE	7.2
‘CA’	GET DATA	6.9
‘D0’	WRITE BINARY	6.2
‘D2’	WRITE RECORD	6.6
‘D6’	UPDATE BINARY	6.3
‘DA’	PUT DATA	6.10
‘DC’	UPDATE RECORD	6.8
‘E2’	APPEND RECORD	6.7

5.4.3 参数字节

命令中的参数字节 P1—P2 可以具有任何值。如果参数字节不提供进一步的限定，则它应置为‘00’。

5.4.4 数据字段字节

每个数据字段应具有下列三种结构之一：

——每个 TLV 编码的数据字段应由一个或多个 TLV 编码的数据对象组成；

——每个非 TLV 编码的数据字段应按照相应命令的规范由一个或多个数据元组成；

——专用编码的数据字段结构在本规范中不予规定。

本规范支持在数据字段中的下列两种类型的 TLV 编码的数据对象：

——BER—TLV 数据对象；

——简单 TLV 数据对象。

本规范不使用‘00’或‘FF’作为标记值。

每个 BER—TLV 数据对象应由 2 个或 3 个连续的字段(见 ISO8825 和附录 D)组成。

——标记字段 T 由一个字节或多个连续字节组成。它编码了类别、类型和编号。

——长度字段由一个字节或多个连续字节组成。它编码了整数 L。

——如果 L 不为空，则值字段 V 由 L 个连续字段组成。如果 L 为空，则数据对象为空：不存在值字段。

每个简单 TLV 数据对象应由 2 个或 3 个连续字段组成。

——标记字段 T 由单个字节组成，从 1 至 254 中的一个编号(例如，一个记录标识符)。它对类别和结构类型不进行编码。

——长度字段由 1 个字节或 3 个连续字节组成。如果长度字段的首字节处于从‘00’至‘FE’的范围内，则长度字段由单个字节组成，该字节编码从 0 至 254 中的一个整数 L。如果首字节等于‘FF’，则长度字段后续 2 个字节使用从 0 至 65535 中的值编码了一个整数 L。

——如果 L 不为空，则值字段 V 由 L 个连续字节组成。如果 L 为空，则数据对象为空：不存在有效字段。

某些命令(例如，SELECT FILE)的数据字段，简单 TLV 数据对象的值字段和某些原始 BER—TLV 数据对象的值字段都预期用于编码一个或多个数据元。

某些其他命令(例如，面向记录的命令)的数据字段、其他原始 BER—TLV 数据对象的值字段都预期用于编码一个或多个简单 TLV 数据对象。

某些其他命令(例如，面向对象的命令)的数据字段和结构化 BER—TLV 数据对象的值字段都预期用于编码一个或多个 BER—TLV 数据对象。

注：在 TLV 编码的数据对象之前、之间或之后，无任何含义‘00”或‘FF’字节可以出现(例如，由于擦除的或修改的 TLV 编码的数据对象引起的)。

5.4.5 状态字节

响应的状态字节 SW1—SW2 表示了卡内的处理状态。图 7 示出了本规范本部分定义的值的结构方案。

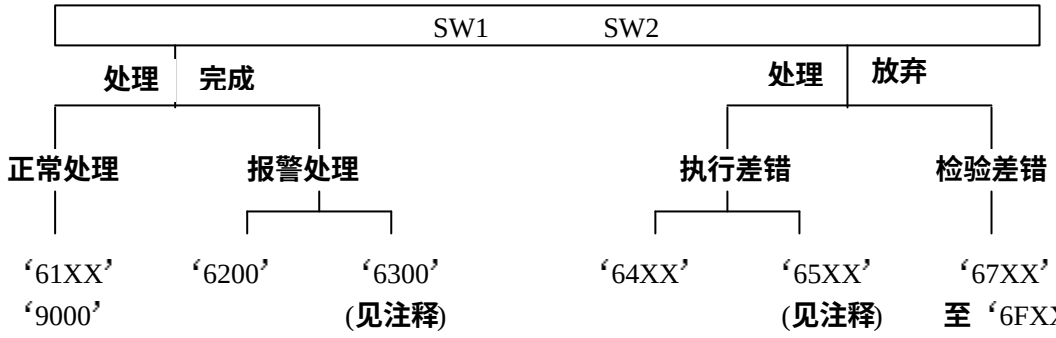

图 7　状态字节的结构方案

注:当 SW1=‘63’或‘65’时，非易失存储器的状态变化。当 SW1=除‘63’和‘65’外的‘6X’时，非易失存储器的状态不变化。

由于本规范本部分的规定的原因，本部分不定义 SW1—SW2 的下列值:

——‘60XX’;

——如果‘XX’≠‘00’，在每种情况下，‘67XX’，‘6BXX’，‘6DXX’。6EXX’，‘6FXX’;

——如果‘XXX’≠‘000’，‘9XXX’。

无论哪个协议被使用，SW1—SW2 的下列值要予以定义(见附录 A 的举例)。

——如果使用响应(其中 SW1=‘6C’)来中途停止命令，当在发出任何其他命令之前重新发出同一命令，则 SW2 指示将该值给予短的 L_e 字段(被请求数据的准确长度)。

——如果使用响应(其中 SW1=‘61’)来处理命令(它可以是情况 2 或 4，见表 4 和图 4)，则在发出任何其他命令之前发出的 GET RESPONSE 命令中，SW2 指示将最大值给予短的 L_e 字段(额外数据长度仍然有效)。

注:类似于由‘61XX’所提供的功能可以在应用级上通过‘9FXX’来提供。

通过表 13～18 所完成的表 12 示出了本规范本部分定义的 SW1—SW2 值的一般含义。对于每个命令，相应的条款提供了更详细的含义。

当 SW1 的值为‘62’，‘63’，‘65’，‘68’，‘69’和‘6A’时，表 13～18 规定了 SW2 的值。除了本规范本部分不定义的从‘F0’至‘FF’值之外，表 13 至表 18 不定义的 SW2 值都是 RFU。

表 12　SW1—SW2 的编码

SW1—SW2	含　　义
	正常的处理
‘9000’	—无进一步限定
‘61XX’	—SW2 指示仍然有效的响应字节数 (见下面文本)
	报警处理
‘62XX’	—非易失存储器状态不变化 (在 SW2 中进一步的限定，见表 13)
‘63XX’	—非易失存储器状态变化 (在 SW2 中进一步的限定，见表 14)

续表 12　SW1—SW2 的编码

	执行差错
‘64XX’	—非易失存储器状态不变化 (SW2=‘00’，其他值都是 RFU)
‘65XX’	—非易失存储器状态变化 (在 SW2 中进一步的限定，见表 15)

‘66XX’	—保留供安全相关的发布使用 (本规范本部分不定义)
	校验差错
‘6700’	—错误的长度
‘68XX’	—CLA 的功能不被支持 (在 SW2 中进一步的限定，见表 16)
‘69XX’	—不允许的命令 (在 SW2 中进一步的限定，见表 17)
‘6AXX’	—错误的参数 P1～P2 (在 SW2 中进一步的限定，见表 18)
‘6B00’	—错误的参数 P1～P2
‘6CXX’	—错误的长度 L_e:SW2 指示准确的长度 (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

表 13　当 SW1=‘62’时，SW2 的编码

SW2	含　　义
‘00’	没有信息被给出
‘81’	返回数据的一部分可能被损坏
‘82’	读出 L_e 字节之前，文件/记录已结束
‘83’	选择的文件无效
‘84’	FCI 未按照 5.1.5 格式化

表 14　当 SW1=‘63’时，SW2 的编码

SW2	含　　义
‘00’	没有信息被给出
‘81’	通过最后写入来填满文件
‘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 15　当 SW1=‘65’时，SW2 的编码

SW2	含　　义
‘00’	没有信息被给出
‘81’	存储器故障

表 16　当 SW1=‘68’时，SW2 的编码

SW2	含　　义
‘00’	没有信息被给出
‘81’	逻辑信道不被支持
‘82’	安全报文不被支持

表 17　当 SW1=‘69’时，SW2 的含义

SW2	含　　义
‘00’	没有信息被给出
‘81’	命令与文件结构不兼容
‘82’	安全状态不被满足
‘83’	认证方法被阻塞

‘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

表 18　当 SW1=‘6A’ 时，SW2 的编码

SW2	含　　义
‘00’	没有信息被给出
‘80’	在数据字段中的不正确参数
‘81’	功能不被支持
‘82’	文件未找到
‘83’	记录未找到
‘84’	无足够的文件存储空间
‘85’	L_c 与 TLV 结构不一致
‘86’	不正确的参数 P1—P2
‘87’	L_c 与 P1—P2 不一致
‘88’	引用的数据未找到

5.5　逻辑信道

5.5.1　一般概念

在接口处看到的逻辑信道作为与 DF 的逻辑链路进行工作。

在一个逻辑信道上应存在独立的活动，而与另一个信道上的活动无关。也就是说，在一个逻辑信道上的命令相互关系应独立于另一个逻辑信道上的命令相互关系。然而，逻辑信道可以共享与应用相关的安全状态，因此，可以具有与安全有关的跨越逻辑信道的命令相互关系(例如，命令 VERIFY)。

提供给某一逻辑信道的命令运载了在 CLA 字节中的相应逻辑信道号(见表 8 和 9)。逻辑信道的编号从 0 至 3。如果卡支持逻辑信道机制，则有效逻辑信道的最大编号可以在卡能力中指出(见本部分规范 8.3.6 节)命令响应对按当前描述的那样进行工作。

本规范本部分仅支持在启动后续命令响应对之前应完成的命令响应对。跨越逻辑信道应该没有命令及其响应的交错；在收到命令与发送响应给该命令之间，只有一个逻辑信道是活动的。当逻辑信道被开放时，它保持开放，直到由 MANAGE CHANNEL命令显式地关闭为止。

注：

1)如果不排除，对同一 DF，可以开放一个以上的逻辑信道(见 5.1.5 文件可访问性)。

2)如果不排除，一个以上的逻辑信道可以选择同一 EF(见 5.1.5 文件可访问性)。

3)在逻辑信道上的 SELECT FILE 命令将开放当前 DF 及可能的当前 EF。因此，每个逻辑信道有一个当前 DF 及可能的一个当前 EF 作为 SELECT FILE 命令行为的结果以及使用短 EF 标识符访问命令的文件。

5.5.2　基本逻辑信道

基本逻辑信道永久有效。当被编号时，它的编号为 0。当类别字节按照表 8 和 9 进行编码时，位 1 和 2 编码了逻辑信道号。

5.5.3　打开逻辑信道

逻辑信道可通过成功地完成下列内容来打开：

——通过分配类别字节中的大于 0 的逻辑信道号来完成引用的 DF 的 SELECT FILE 命令；

——或者，完成 MANAGE CHANNEL 命令的开放功能，该命令分配在命令 APDU 中的 0 以

外的逻辑信道号或请求卡分配的和响应中返回的逻辑信道号。

5.5.4 关闭逻辑信道

MANAGE CHANNEL 命令的关闭功能可以用来显式地使用逻辑信道号关闭逻辑信道。关闭之后，逻辑信道号可供重新使用。基本逻辑信道应不予关闭。

5.6 安全报文交换

安全报文交换的目的是通过确保两种基本安全功能来保护往返于卡的报文部分，这两种功能是:数据鉴别和数据安全性。

安全报文交换可通过应用一种或多种安全机制来获得。每种安全机制都涉及算法、密钥、自变量、经常还有初始数据。

- 对于安全机制的执行，数据字段的发送和接收可以交错进行。本规范不妨碍通过顺序地分析哪些机制和哪些安全项目应该用于处理数据字段的其余部分所作的决定。
- 两种或两种以上的安全机制可以使用带有不同操作方式的相同算法(见 ISO10116)。填充规则的现有规范不排除这种特征。

本条定义了 SM 相关数据对象的三种类型:

——普通值数据对象，预期用来运载普通数据;

——安全机制数据对象，预期用来运载安全机制的计算结果;

——辅助安全数据对象，预期用来运载控制引用和响应描述符。

5.6.1 SM 格式概念

在涉及基于密码安全机制的每个报文中，数据字段应符合 ASN.1 的基本编码规则(见 ISO8825 和附录 D)，除非通过类别字节另有指示(见本部分规范 5.4.1)。

在数据字段中，可以选择现存的 SM 格式:

——隐式地选择，即，在发出命令之前已知;

——显式地选择，即，通过类别字节来固定(见表 9)。

本规范本部分定义的 SM 格式是 BER—TLV 编码的。

- 上下文特定的标签类别(范围从‘80’至‘BF’)被保留供 SM 用。
- 其他类别的数据对象可以呈现(例如，应用特定类别的数据对象)。
- 某些与 SM 相关的数据对象是递归的:它们的普通值字段仍然是 BER—TLV 编码的，因此上下文特定的类别自然是被保留供 SM 用。

在上下文特定类别中，标签的位 1 决定了 SM 相关的数据对象会(b1=1)或不会(b1=0)集成到认证用的数据对象的计算中。如果呈现，其它类别数据对象会集成到该计算中。

5.6.2 普通值数据对象

对于不按 BER—TLV 编码的数据以及对于包括与 SM 相关的数据对象的 BER—TLV，ENVELOPE 都是强制性的。对于不包括与 SM 相关的数据对象的 BER—TLV，ENVELOPE 是任选的。表 19 示出了 ENVELOPE 用的普通值数据对象。

表 19 普通值数据对象

标　记	值
	简明值由下列内容组成
‘B0’，‘B1’	—BER—TLV 包括与 SM 相关的数据对象
‘B2’，‘B3’	—BER—TLV 但不包括与 SM 相关的数据对象
‘80’，‘81’	—不是 BER—TLV 编码的数据
‘99’	—SM 状态信息(例如 SW1—SW2)

5.6.3 认证用的数据对象

5.6.3.1 密码“校验和”数据对象

密码校验和的计算(见 ISO9797)包含有初始校验块、密钥以及不可逆密码算法块。

在相关密钥的控制下，该算法在本质上将现行的 K 字节(典型地是 8 或 16)输入块变换成现行的相同长度输出块。

密码校验和的计算按下列连续步骤执行：

——初始步骤——初始步骤设置下列块之一的初始校验块：

- 空块，即，K 字节值为‘00’
- 链接块，即，由先前计算命令(先前命令的最后一校验块)和响应(先前响应的最后一校验块)的结果，例如，由外界提供的初始值块；
- 按照相关密钥从变换辅助数据产生的辅助块。如果辅助数据小于 K 字节，则它通过置为 0 的位为起头，直至块长度。

——相继步骤——当表 9 可用(CLA=‘0X’，‘8X’，‘9X’或‘AX’)时，如类别字节的位 b4 和 b3 置为 1，则第 1 个数据、块由命令 APDU(CLA INS P1 P2)的首标后随值为‘80’的一个字节和值为 ‘00’的 5 个字节的 K 组成。

密码“校验和”应集成具有标记(b1=1)的任何 SM 相关数据对象和带有标记超出范围‘80’至‘BF’的任何数据对象。这些数据对象应通过数据块集成到当前的校验块中。分解数据块应按下列方法进行：

——分块应在被集成的相邻数据对象之间的边界处继续进行。

——填充应在被集成的每个数据对象(既可后随不被集成的数据对象，也可不后随进一步的数据对象)的结束处使用。

填充由一个值为‘80’的必备字节组成，如果需要可后随置为‘00’的 0 至(K—1)个字节，直到相应的数据块被填充直至 K 个字节为止。当填充字节不被发送时，鉴别的填充不影响传输。

操作方式为“密码块链接”(见 ISO10116)。第 1 个输入是初始校验块与第 1 个数据块的异或运算结果。第 1 个输出由第 1 个输入产生。当前输入是先前输出与当前数据块的异运算结果。最终的校验块就是最后的输出。

——最终步骤——最终步骤从最终校验块中抽取出密码“校验和”（开始的 m 个字节，至少 4 个）。

表 20 示出了密码“校验和”数据对象。

表 20 密码“校验和”数据对象

标记	值
‘8E’	密码校验和(至少 4 个字节)

5.6.3.2 数字签名数据对象

数字签名计算典型地基于非对称密码技术，数字签名有两种类型：

——带有附件的数字签名；

——给出报文恢复的数字签名。

带有附件数字签名的计算隐含着使用散列函数(见 ISO10118)。数据输入或者由数字签名输入数据对象的值(见表 21)组成，或者通过本部分规范 5.6.3.1 定义的机制来确定。

给出报文恢复的数字签名的计算(见 ISO9796)不隐含使用散列函数。然后，根据应用的需要，散列代码可以呈现作为恢复报文的一部分，而该报文本身可以是 BER—TLV 编码的。表 21 示出了数字签名相关的数据对象。

表 21 数字签字有关的数据对象

标　记	值
‘9A’，　‘BA’	数字签名输入数据
‘9E’	数字签名

5.6.4　保密性的数据对象

保密性数据对象预期用于运载密码，其普通值由下列三种情况之一组成：

——BER—TLV，包含 SM 相关的数据对象；

——BER—TLV，不包含 SM 相关的数据对象；

——不是 BER—TLV 编码的数据。

当普通值不是由 BER—TLV 编码数据组成时，则必须指示填充。当填充被应用但未被指示时，则本部分规范 5.6.3.1 定义的规则可应用。

表 22 示出了保密性的数据对象。

表 22　保密性的数据对象

标　　记	值
‘82’，‘83’ ‘84’，‘85’	密码，简明值由下列内容组成： —BER—TLV，包含 SM 相关数据对象 —BER—TLV，但不是 SM 相关的数据对象
‘86’，‘87’	填充指示符字节(见表 23)后随密码(普通值不在 BER—TLA 中编码)

保密性的每一个数据对象可以使用任何密码算法以及任何操作方式，归用于适合的算法引用(见本部分规范 5.6.5.1)。在不存在算法引用的情况下以及当没有机制可隐式地被选择用于保密性时，一种默认机制可应用。

对于密码的计算，该密码之前是填充指示符时，默认机制就是使用“电子密码本”的块密码法(见 ISO10116)。使用块密码可以包含填充。保密性的填充对传输有影响，因为密码(一个或多个块)大于明文。

表 23 示出了填充指示符字节

表 23　填充指示符字节

值	含　　义
‘00’	—没有进一步的指示
‘01’	—按本部分规范 5.6.3.1 定义进行填充
‘02’	—没有填充
‘80”～‘8E”	—专有的其他值为 RFU

对于密码的计算，该密码之前不是填充指示符字节时，默认机制就是使用“异或”运算的流密码。在这种情况下，密码是被伪装的数据字节串与伪装的相同长度串进行“异或”运算的结果。因此，伪装要求在由相同操作所恢复的值字段中没有填充及被伪装的数据对象。

5.6.5　辅助安全数据对象

算法、密钥和可能的初始数据可以被选择用于每种安全机制。

——显示地，即，发布命令之前已知；

——显示地，通过控制引用嵌套在控制引用样板中。

每个命令报文可以运载响应描述符样板，该样板安排了在响应中所要求的数据对象。在响应描述符内，安全机制仍不被应用；接收实体应将该机制应用到构造响应中。

5.6.5.1　控制引用

表 24 示出了控制引用样板。

表 24　控制引用样板

标　记	含　　义
‘B4’，‘B5’	—对密码“校验和”有效的样板
‘B6’，‘B7’	—对数字签名有效的样板
‘B8’，‘B9’	—对保密性有效的样板

控制引用样板的最后可能位置正好在被引用机制使用的第 1 个数据对象之前。例如，密码“校验和”用的样板的最后可能位置正好在被集成到计算中的第 1 个数据对象之前。每个控制引用保持有效，直到新的控制被提供用于相同机制。例如，一个命令可以为下一个命令安排控制引用。

每个控制引用样板预期用于运载控制引用数据对象(见表 25):算法引用，文件引用，密钥引用，初始数据引用，并且仅在保密性的控制引用样板中，密码内容引用。

算法引用安排了算法及其操作方式(见 ISO9979 和 10116)。算法引用的结构和编码不在本规范本部分中定义。

文件引用表示了密钥引用有效文件。如果没有文件引用呈现，则密钥引用在当前 DF 中有效。

密钥引用标识了被使用的密钥。

当初始数据引用被应用到密码“校验和”时，该初始数据引用安排了初始校验块。如果没有初始数据引用呈现，并且没有初始校验块显式地被选择，则空块应被使用。此外，在发送保密性的第 1 个数据对象之前，使用一种流密码时，保密性用的样板应为伪装字节串的计算提供辅助数据。

密码内部引用规范了密码的内容(例如，秘密密钥、初始口令、控制字)。值字段的第 1 个字节是已命名的密码描述符字节，并且是强制性的。范围‘00’至‘7F’为 RFU。范围‘80’至‘FF’为专有的。

表 25　控制引用数据对象

标　记	值
‘80’	算法引用
‘81’ ‘82’	文件引用 —文件标识符或路径 —DF 名称
‘83’ ‘84’	密钥引用 —对于直接使用 —对于计算会话密钥
‘85’ ‘86’ ‘87’ ‘88’ ‘89’到 ‘8D’	初始数据引用 *初始校验块 —L=0，空块 —L=0，链接块 —L=0，先前的初始值块加 1 L=k，初始值块 *辅助数据 —L=0，先前交换的询问加 1 L≠0，没有进一步的指示 —L=0，专用数据元的索引 L≠0，专用数据元的值
‘8E’	密码内部引用

5．6．5．2　响应描述符

如果在命令 APDU 的数据字段中呈现响应描述符样板，则它应安排相应响应的结构。空数据对象应列出产生响应所需要的全部数据。

处理命令报文的数据字段所使用的安全项目(算法、密钥及初始数据)可以不同于产生后续响应报文的数据字段所使用的那些安全项目。

下列规则应该使用。

——卡应填充每个空的原始数据对象。

——呈现在响应描术符中的每个控制引用样板对于算法、文件和密钥而言应该在带有相同控制引用的相同位置上呈现在响应中。如果响应描述符提供了辅助数据，则在响应中数据对象应为空。如果辅助数据的空引用数据对象呈现在响应描述符中，则在响应中它应为空。

——通过相关的安全机制，使用选择的安全项目时，卡应产生所有请求的安全机制数据对象。表 26 示出了响应描述符样板。

表 26　响应描述符样板

标记	值
‘BA’　‘BB’	响应描述符

5. 6. 6　SM 状态条件

在使用安全报文交换的任何命令中，下列特定差错条件可能发生。

——SW1=‘69’，同时 SW2=

- ‘87’:期望的 SM 数据对象失踪。
- ‘88’:SM 数据对象不正确。

6　基本的行业间命令

对于遵循本规范本部分的所有卡而言，应该不强制要求支持本部分描述的所有命令或支持命令的所有选项。

当进行国际交换时，卡系统服务及相关命令和选项的集合应按照第 9 章的定义加以使用。表 11 提供了本规范本部分定义的命令概要。

安全报文交换(见本部分规范 5.6)对报文结构的影响不在本章中描述。

在 6.X.5 的每一条中所给出的差错和报警条件的列表不是穷举的(见本规范本部分 5.4.5)。

6. 1　READ BINARY 命令

6.1.1　定义和范围

READ BINARY 响应报文给出了带有透明结构的 EF 内容的一部分。

6.1.2　使用与安全的条件

当命令包含了有效的短 EF 标识符时，它将文件置位为当前 EF。

根据当前选择的 EF 来处理该命令。仅当安全状态满足了用于该功能的为该 EF 而定义的安全属性时，才能执行该命令。

如果命令被应用到不带有透明结构的 EF，则应放弃该命令。

6.1.3　命令报文

表 27　READ BINARY 命令 APDU

CLA	按 5.4.1 定义的
INS	‘B0’
P1—P2	见以下文本
L_c 字段	空
数据字段	空
L_e 字段	待读的字节数

如果在 P1 中 b8=1，则 P1 的 b7 和 b6 置为 0(RFU 若干位)，P1 的 b5 至 b1 是短 EF 标识符，并且 P2 是在从文件开始的数据单元中被读的第 1 个字节的偏移。

如果在 P1 中 b8=0，则 P1‖P2 是在从文件开始的数据单元中被读的第 1 个字节的偏移。

6.1.4 响应报文(标称情况)

如 L_e 字段仅包含若干“0”，则对于短的长度在不超过 256 的范围内或者对扩充长度在不超过 65536 的范围内，所有字节(直到文件结束为止)应被读出。

表 28 READ BINARY 响应 APDU

数据字段	读的数据(L_e 字节)
SW1—SW2	状态字节

6.1.5 状态条件

下列特定报警条件可能发生。

——SW1=‘62’，同时 SW2=

·‘81’:被返回数据的一部分可以被损坏。

·‘82’:读 L_e 字节之前达到的文件结束。

下列特定差错条件可能发生。

——SW1=‘67’，同时 SW2=

·‘00’:错误的长度(错误的 L_e 字段)。

——SW1=‘69’，同时 SW2=

·‘81”:命令与文件结构不兼容。

·‘82’:安全状态不被满足。

·‘86”:命令不被允许(没有当前 EF)。

SW1=‘64’，同时 SW2=

·‘81’:功能不被支持。

·‘82’:文件未被找到。

——SW1=‘6B’，同时 SW2=

·‘00’:错误的参数(偏移超出 EF)。

——SW1=‘6C’，同时 SW2=

‘XX’:错误的长度(错误的 L_e 字段;‘XX’表示正确长度)。

6.2 WRITE BINARY 命令

6.2.1 定义和范围

WRITE BINARY 命令报文启动将二进制值写入 EF。

根据文件属性，命令应执行下列操作之一:

——早已存在卡内的位与在命令 APDU 中给出的位进行逻辑“或”运算(该文件位的逻辑擦除状态为“0”)。

——对早已存在卡内的位与在命令 APDU 中给出的位进行逻辑“与”运算(该文件位的逻辑擦除状态为“1”)。

——将命令 APDU 中给出的位一次写入卡的操作。

当在数据编码字节中未给出指示(见表 86)时，则逻辑“或”行为应该适用。

6.2.2 使用与安全的条件

当命令包含了有效的短 EF 标识符时，它将文件置位为当前 EF。

根据当前选择的 EF 来处理该命令。仅当安全状态满足了用于写功能的安全属性时，才能执行该命令。

一旦 WRITE BINARY 已经被应用到一次写 EF 的数据单元，如果数据单元的内容或被连接到该数据单元的逻辑擦除状态指示符(如果有)不同于逻辑擦除状态，则涉及该数据单元的任何进一步的写操作将被放弃。

如果命令被施加到不带有透明结构的 EF，则应放弃该命令。

6.2.3 命令报文

表 29　WRITE BINARY 命令 APDU

CLA	按 5.4.1 定义的
INS	‘D0’
P1—P2	见以下文本
L_c 字段	后续数据字段的长度
数据字段	待写的数据单元串
L_e 字段	空

如果在 P1 中 b8=1，则 P1 的 b7 和 b6 显域 0(RFU 若干位)，P1 的 b5 至 b1 是短 EF 标识符，并且 P2 是在从文件开始的数据单元中被写的第 1 个字节的偏移。

如果在 P1 中 b8=0，则 P1‖P2 是在从文件开始的数据单元中被写的第 1 个字节的偏移。

6.2.4　响应报文(标称情况)

表 30　WRITE BINARY 响应 APDU

数据字段	空
SW1—SW2	状态字节

6.2.5　状态条件

下列特定报警条件可能发生。

——SW1=‘63’，同时 SW2=

·‘CX’:计数器(成功的写，但是在使用内部重试例行程序之后，‘X’≠0 表示重试数；‘X’=0 意味着没有计数器被提供)。

下列特定差错条件可能发生。

——SW1=‘65’，同时 SW2=

·‘81’:存储器故障(不成功的写)

——SW1=‘67’，同时 SW2=

·‘00’:错误的长度(错误的 L_c 字段)

——SW1=‘69’，同进 SW2=

·‘81’:命令与文件结构不兼容

·‘82’:安全状态不被满足

·‘86’:命令不被允许(没有当前 EF)

——SW1=‘64’，同时 SW2=

·‘81’:功能不被支持

·‘82’:文件未被找到

——SW1=‘6B’，同时 SW2=

·‘00’:错误的参数(偏移超出 EF)。

6.3　UPDATE BINARY 命令

6.3.1　定义和范围

UPDATE BINARY 命令报文启动使用在命令 APDU 中给出的位来更新早已呈现在 EF 中的位。

6.3.2　使用与安全的条件

当命令包含了有效的短 EF 标识符时，它将文件置位为当前 EF。

根据当前选择的 EF 来处理该命令。仅当安全状态满足了用于更新功能的安全属性时，才能执行该命令。

如果命令被施加到不带有透时结构的 EF，则应放弃该命令。

6.3.3　命令报文

表 31　UPDATE BINARY 命令 APDU

CLA 按	按 5.4.1 定义的
INS	见以下文本
P1—P2	后续数据字段的长度
数据字段	待更新的数据单元串
L_e 字段	空

如果在 P1 中 b8=1，则 P1 的 b7 和 b6 置为 0(RFU 若干位)，P1 的 b5 至 b1 是短 EF 标识符，并且 P2 是在从文件开始的数据单元中被更新的第 1 个字节的偏移。

如果在 P1 中 b8=0，则 P1‖P2 是在从文件开始的数据单元中被更新的第 1 个字节的偏移。

6.3.4 响应报文(标称情况)

表 32 UPDATE BINARY 响应 APDU

数据字段	空
SW1—SW2	状态字节

6.3.5 状态条件

下列特定报警条件可能发生。

——SW1=‘63’，同时 SW2=

·‘CX’:计数器(成功的更新，但是在使用内部重试例行程序之后，‘X’≠0 表示重试数’‘X’=0 意味着没有计数器被提供)。

下列特定差错条件可能发生。

——SW1=‘65’，同时 SW2=

·‘81’:存储器故障(不成功的更新)。

——SW1=‘67’，同时 SW2=

·‘00’:错误的长度(错误的 L_c 字段)。

——SW1=‘69’，同时 SW2=

·‘81’:命令与文件结构不兼容。

·‘82’:安全状态不被满足。

·‘86’:命令不被允许(没有当前 EF)。

——SW1=‘6A’，同时 SW2=

·‘81’:功能不被支持。

·‘82’:文件未被找到。

——SW1=‘69’，同时 SW2=

·‘00’:错误的参数(偏移超出 EF)。

6.4 ERASE BINARY 命令

6.4.1 定义和范围

ERASE BINARY 命令报文顺序地从给出的偏移开始将 EF 的内容的一部分置为其逻辑擦除的状态。

6.4.2 使用与安全的条件

当命令包含了有效的短 EF 标识符时，它将文件置为当前 EF。

根据当前选择的 EF 来处理该命令。仅当安全状态满足了用于擦除功能的安全属性时，才能执行该命令。

如果命令被施加到不带有透明结构的 EF，则应放弃该命令。

6.4.3 命令报文

表 33 ERASE BINARY 命令 APDU

CLA	按 5.4.1 定义的
INS	‘0E’
P1—P2	见以下文本
L_c 字段	空或‘02’
数据字段	见以下文本
L_e 字段	空

如果在 P1 中 b8=1，则 P1 的 b7 和 b6 置为‘0’(RFU 若干位)，P1 的 b5 至 b1 是短 EF 标识符，P1 是在从文件开始的数据单元中被擦除的第 1 个字节的偏移。

如果在 P1 中 b8=0，则 P1‖P2 是在从文件开始的数据单元中被擦除的第 1 个字节的偏移。

如果数据字段呈现，它编码不被擦除的第 1 个数据单元的偏移。该偏移应大于在 P1—P2 中编码的一个偏移。当数据字段为空时，该命令擦除到该文件的结束端。

6.4.4 响应报文(标称情况)

表 34 ERASE BINARY 响应 APDU

数据字段	空
SW1—SW2	状态字节

6.4.5 状态条件

下列特定报警条件可能发生。

——SW1=‘63’，同时 SW2=

·‘CX’:计数器(成功的擦除，但是在使用内部重试例行程序之后，‘X’≠0 表示重试数，‘X’=0 意味着没有计数器被提供)。

下列特定差错条件可能发生:

——SW1=‘65’，同时 SW2=

·‘81’:存储器故障（不成功的擦除)。

——SW1=‘67’:同时 SW2=

·‘00’:错误的长度(错误的 L_e 字段)。

——SW1=‘69’，同时 SW2=

·‘81’:命令与文件结构不兼容。

·‘82’:安全状态不被满足。

·‘86’:命令不被允许(没有当前 EF)。

——SW1=‘6A’，同时 SW2=

·‘81’:功能不被支持

·‘82’:文件未找到

——SW1=‘6B’，同时 SW2=

·‘00’:错误的参数(偏移超出 EF)。

6.5 READ RECORD 命令

6.5.1 定义和范围

READ RECODE 响应报文给出了 EF 的规定记录的内容或 EF 的一个记录开始部分的内容。

6.5.2 使用与安全的条件

仅当安全状态满足了用于读功能的该 EF 的安全属性时，才能执行该命令。

如果在发出命令的时刻，当前选择了 EF，则该命令可以被处理，而无需该文件的标识。

当命令包含了有效的短 EF 标识符时，它将文件置为当前 EF，并且复位当前记录指针。

如果命令被施加到不带有记录结构的 EF，则应放弃该命令。

6.5.3 命令报文

表 35　READ RECORD（S）命令 APDU

CAL	按 5.4.1 定义的
INS	‘B2’
P1	记录号或被读的第 1 个记录的标识符(‘00’表示当前记录)
P2	引用控制 按照表 36
L_c 字段	空
数据字段	空
L_e 字段	被读字节数

表 36　引用控制 P2 的编码

b8 b7 b6 b5 b4 b3 b2 b1	含　义
0 0 0 0 0 - - -	—当前选择的 EF
× × × × × - - - (不全相等)	—短 EF 标识符
1 1 1 1 1 - - -	RFU
- - - - - 1 × ×	利用 P1 中的记录号
- - - - - 1 0 0	—READ RECODE#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	RFU
- - - - - 0 × ×	利用 P1 中的记录标识符
- - - - - 0 0 0	—读第 1 个出现(标识符)
- - - - - 0 0 1	—读最后一个出现(标识符)
- - - - - 0 1 0	—读下一个出现(标识符)
- - - - - 0 1 1	—读先前一个出现(标识符)

6.5.4　响应报文(标称情况)

如果 L_e 字段仅包含了若干‘0’，则根据 P2 的 b3 b2 b1，并且对于短的长度在不超过 256 的范围内，或者对扩充的长度在不超过 65536 的范围内，该命令应完整地读出：

——单个请求的记录；

——或请求的记录序列

表 37　READ RECODE（S）响应 APDU

数据字段	L_r(可以等于 L_e)字节 见表 38
SW1—SW2	状态字节

当记录是简单 TLV 数据对象(见 5.4.5)时，表 38 示出了响应报文数据字段的格式。

表 38—1　当读一个记录时，响应的数据字段

情况 a——部分读的一个记录

T_n	L_n	记录中的前面若干数据字节
1 个字节	1 个或 3 个字节	

————————————----------L_e 字节————————————————----------

当 L_e 字段不仅包含了‘0’时，该情况适用。

情况 b——完整读的一个记录

T_n	L_n	记录的整个数据字节
1 个字节	1 个或 3 个字节	L_n 字节

当 L_e 字段仅包含若干‘0’时，该情况适用。

表 38—2　当读几个记录时，响应的数据字段

情况 c——部分读的记录序列

记录#n $T_n \| L_n \| V_n$	…… ……	记录#n＋m 中的前面若干字节 $T_{n+m} \| L_{n+m} \| V_{n+m}$

——————————----------L_e字节——————————----------

当 L_e 字段不仅包含了若干‘0’时，该情况适用。

情况 d——读多个记录直到文件结束

记录#n $T_n \| L_n \| V_n$	…… ……	记录#n＋m $T_{n+m} \| L_{n+m} \| V_{n+m}$

当 L_e 字段仅包含了若干‘0’时，该情况适用。

数据字段的长度与其 TLV 结构相比较给出了数据的性质:唯一记录(读一个记录)或最后一个记录(读所有记录)是不完整的，完整的或添加的。

注:如 TLV 编码不被使用,则读所有记录的功能导致接收的几个记录没有标准的记录定界。

6.5.5 状态条件

下列特定报警条件可能发生。

——SW1=‘62’，同时 SW2=

·‘81’:被返回数据的一部分可以被损坏。

·‘82’:在 L_e 字节之前已到达记录结束端。

下列特定差错条件可能发生。

——SW1=‘67’，同时 SW2=

·‘00’:错误的长度(空的 L_e 字段)

——SW1=‘69’，同时 SW2=

·‘81’:命令与文件结构不兼容。

·‘82’:安全状态不被满足。

——SW1=‘6A’:同时 SW2=

·‘81’:功能不被支持。

·‘82’:文件未被找到。

·‘83’:记录未被找到。

——SW1=‘6C’，同时 SW2=

·‘XX’:错误的长度(错误的 L_e 字段;‘XX’表示正确的长度)。

6.6 WRITE RECORD 命令

6.6.1 定义和范围

WRITE RECORD 命令报文启动下列操作之一:

——写一次记录;

——对早已呈现在卡内的记录数据字节与在命令 APDU 中给出的记录数据字节进行逻辑“或”运算;

——对早已呈现在卡内的记录数据字节与在命令 APDU 中给出的记录数据字节进行逻辑“和”运算。

当在数据编码字节中未给出指示(见表 86)时，逻辑“或”运算应该适用。

当使用当前记录寻址时，该命令应将记录指针设置在成功的 WRITE RECORD 上。

6.6.2 使用与安全的条件

仅当安全状态满足了用于写功能的该 EF 的安全属性时，才能执行该命令。

如果在发出命令的时刻，当前选择了 EF，则该命令可以被处理，而无需该文件的标识。

当命令包含了有效的短 EF 标识符时，它将文件置为当前 EF，并且复位当前记录指针。

如果命令被施加到不带有记录结构的 EF，则应放弃该命令。

被施加到循环文件的“先前”的命令选项(P2＝××××011)具有和 APPEND RECORD 相同的行为。

6.6.3 命令报文

表 39 WRITE RECORD 命令 APDU

CLA	按 5.4.1 定义的
INS	‘D2’
P1	P1＝‘00’指明当前记录 P1≠‘00’是所规定记录的号
P2	按照表 40
L_c 字段	后续数据字段的长度
数据字段	待写的记录
L_e 字段	空

表 40 引用控制 P2 的编码

b8	b7	b6	b5	b4	b3	b2	b1	含义
0	0	0	0	0	-	-	-	——当前选择的 EF
×	×	×	×	×	-	-	-	——短 EF 标识符
(不全相等)								
-	-	-	-	-	0	0	0	——第 1 个记录
-	-	-	-	-	0	0	1	——最后一个记录
-	-	-	-	-	0	1	0	——下一个记录
-	-	-	-	-	0	1	1	——先前一个记录
-	-	-	-	-	1	0	0	——在 P1 中给出的记录号
任何其他值								RFU

当记录为简单 TLV 数据对象(见本部分规范 5.4.4)时，表 41 示出了命令报文数据字段的格式。

表 41 命令的数据字段

完整写的一个记录

T_n 1 个字节	L_n 1 个或 3 个字节	记录的整个数据字节 L_n 字节

6.6.4 响应报文(标称情况)

表 42 WRITE RECORD 响应 APDU

数据字段	空
SWL-SW2	状态字节

6.6.5 状态条件

下列特定报警条件可能发生。

——SW1＝‘63’， 同时 SW2＝

- ‘CX’:计数器(成功的写，但是使用内部重试例行程序之后，‘X’≠‘0’表示重试数；‘X’＝‘0’意味着没有计数器被提供)。

下列特定差错条件可能发生。

——SW1＝‘65’， 同时 SW2＝

- ‘81’:存储器故障(不成功的写)。

——SW1＝‘67’， 同时 SW2＝

- ‘00’:错误的长度(空的 L_c 字段)。

——SW1＝‘69’， 同时 SW2＝

- ‘81’:命令与文件结构不兼容。
- ‘82’:安全状态不被满足。
- ‘86’:命令不被允许(没有当前 EF)。

——SW1＝‘6A’， 同时 SW2＝

- ‘81’:功能不被支持。
- ‘82’:文件未被找到。
- ‘83’:记录未被找到。
- ‘85’:L_c 与 TLV 结构不一致。

6.7 APPEND RECORD 命令

6.7.1 定义和范围

APPEND RECORD 命令报文启动在线性结构 EF 的结束端添加记录，或者在循环结构(见本部分规范 5.1.4)的 EF 内写记录号 1。

命令应将记录指针设置在成功添加的记录上。

6.7.2 使用与安全的条件

仅当安全状态满足了用于添加功能的该 EF 的安全属性时，才可执行该命令。

如果在发布命令的时刻，当前选择了 EF，则该命令可以被处理，而无需该文件的标识。当命令包含了有效的短 EF 标识符时，它将文件置位为当前 EF，并且复位当前记录指针。

如果命令被应用到不带有记录结构的 EF，则应放弃该命令。

注:如果该命令被应用到有很多记录的循环结构 EF，则带有最高记录号的记录可被代替。该记录变成为录记号 1。

6.7.3 命令报文

表 43 APPEND RECORD 命令 APDU

CLA	按 RPV 5.4.1 定义
INS	‘E2’
P1	只有 P1＝‘00’是有效的
P2	按照表 44
L_c 字段	后续数据字段的长度
数据字段	待添加的记录
L_e 字段	空

表 44 引用控制 P2 的编码

b8	b7	b6	b5	b4	b3	b2	b1	含义
0	0	0	0	0	-	-	-	——当前选择的 EF
×	×	×	×	×	-	-	-	——短 EF 标识符
(不全相等)								
任何其他值								RFU

当记录是简单 TLV 数据对象(见 5.4.4)时，表 45 示出了命令报文数据字段的格式。

表 45 命令的数据字段

完整添加的一个记录		
T_n 1 个字节	L_n 1 个或 3 个字节	记录的完整数据字节 L_n 字节

6.7.4 状态条件

下列特定报警条件可能发生。

——SW1＝‘63’， 同时 SW2＝

▪‘CX’:计数器(成功的添加，但是在使用内部重试例行程序之后，‘X’≠‘0’表示重试数;‘X’＝‘0’意味着没有计数器被提供)。

下列特定差错条件可能发生。

——SW1＝‘65’， 同时 SW2＝

▪‘81’:存储器故障(不成功的添加)。

——SW1＝‘67’， 同时 SW2＝

▪‘00’:错误的长度(空的 L_c 字段)。

——SW1＝‘69’， 同时 SW2＝

▪‘81’:命令与文件结构不兼容。

▪‘82’:安全状态不被满足。

▪‘86’:命令不被允许(没有当前 EF)。

——SW1＝‘6A’， 同时 SW2＝

▪‘81’:功能不被支持。

▪‘82’:文件未被找到。

▪‘84’:无足够的文件存储空间。

▪‘85’:L_c 与 TLV 结构不一致。

6.8 UPDATE RECORD 命令

6.8.1 定义和范围

UPDATE RECORD 命令报文启动使用命令 APDU 给出的位来更新特定记录。

当使用当前记录寻址时，该命令应将记录指针设置在成功的更新记录上。

6.8.2 使用与安全的条件

仅当安全状态满足了用于更新功能的该 EF 的安全属性时，才能执行该命令。

如果在发布命令的时刻，当前选择了 EF，则该命令可以被处理，而无需该文件的标识。

当命令包含了有效的短 EF 标识符时，它将文件置位为当前 EF，并且复位当前记录指针。

如果命令被施加到不带有记录结构的 EF，则应放弃该命令。

当命令适用于带有线性固定结构或循环结构的 EF 时，如果该记录长度不同于现有记录的长度，则应放弃该命令。

当命令适用于带有线性可变结构的 EF 时，并且当该记录长度不同于现有记录的长度时，则可以完成该命令。

被施加到循环文件的“先前”的命令选项(P2＝XXXXX011)具有和 APPEND RECORD 相同的行为。

6.8.3 命令报文

表 47 UPDATE RECORD 命令 APDU

CLA	按 5.4.1 定义的
INS	‘DC’
P1	P1＝‘00’指明当前记录 P≠‘00’是所规定记录的号
P2	按照表 48
L_c 字段	后续数据字段的长度
数据字段	待更新的记录
L_e 字段	空

表 48 引用控制 P2 的编码

b8 b7 b6 b5 b4 b3 b2 b1	含义

0 0 0 0 0 - - - × × × × × - - - （不全相等）	——当前选择的 EF ——短 EF 标识符
- - - - - 0 0 0 - - - - - 0 0 1 - - - - - 0 1 1 - - - - - 1 0 0	——第 1 个记录 ——最后一个记录 ——先前一个记录 ——在 P1 中给出的记录号
任何其他值	RFU

当记录是简单 TLV 数据对象(见 5.4.4)时，表 49 示出了命令报文数据字段的格式。

表 49　命令的数据字段

完整更新的一个记录		
T_n 1 个字节	L_n 1 个或 3 个字节	记录的完整数据字节 L_n 字节

6.8.4　响应报文(标称情况)

表 50　UPDATE RECORD 响应 APDU

数据字段 SWL-SW2	空 状态字节

6.8.5　状态条件

下列特定报警条件可能发生。

——SW1＝‘63’，　同时 SW2＝

- ‘CX’:计数器(成功的更新，但是在使用内部重试例行程序之后，‘X’≠‘0’表示重试数;‘X’＝‘0’意味着没有计数器被提供)。

下列特定差错条件可能发生。

——SW1＝‘65’，　同时 SW2＝

- ‘81’:存储器故障(不成功的更新)。

——SW1＝‘67’，　同时 SW2＝

- ‘00’:错误的长度(空的 L_c 字段)。

——SW1＝‘69’，　同时 SW2＝

- ‘81’:命令与文件结构不兼容。
- ‘82’:安全状态不被满足。
- ‘86’:命令不被允许(没有当前 EF)。

——SW1＝‘6A’，　同时 SW2＝

- ‘81’:功能不被支持。
- ‘82’:文件未被找到。
- ‘83’:记录未被找到。
- ‘84’:无足够的文件存储空间。
- ‘85’:L_c 与 TLV 结构不一致。

6.9　GET DATA 命令

6.9.1　定义和范围

GET DATA 命令可在当前上下文(例如，应用特定环境或当前 DF)范围内用于检索一个原始数据对象或者包含在结构化数据对象中所包含的一个或多个数据对象。

6.9.2　使用与安全的条件

仅当安全状态满足了通过功能用的上下文范围内的应用所定义的安全条件时，才能执

行该命令。

6.9.3 命令报文

表 51 GET DATA 命令 APDU

CLA	按 5.4.1 定义的
INS	‘CA’
P1-P2	见表 52
L_c 字段	空
数据字段	空
L_e 字段	在响应时期望的字节数

表 52 参数 P1-P2 的编码

值	含 义
‘0000’至‘003F’	RFU
‘0040’至‘00FF’	P2 中的 BER-TLV 标签(1 个字节)
‘0100’至‘01FF’	应用数据(专有编码)
‘0200’至‘02FF’	P2 中的简单 TLV 标签
‘0300’至‘3FFF’	RFU
‘0400’至‘FFFF’	P1-P2 中的 BER-TLV 标签(2 个字节)

得到应用数据

- 当 P1-P2 的值位于从‘0100’至‘01FF’的范围时， P1-P2 的值应是被保留的一个标识符，它可在给定的应用上下文范围内用于卡内部测试和用于有意义的专有服务。

GET DATA 对象

- 当 P1-P2 的值位于从‘0040’至‘00FF’的范围时， P2 的值应是单个字节的 BER-TLV 标签。值‘00FF’被保留，为了获得上下文内可读的所有公共的 BER-TLV 数据对象。
- 当 P1-P2 的值位于从‘0200’至‘02FF’的范围时，P2 的值应是简单 TLV 标签。值‘0200’是 RFU。值‘02FF’被保留，为了获得在上下文内可读的所有公共的简单 TLV 数据对象。
- 当 P1-P2 的值位于从‘0400’至‘FFFF’的范围时，P1-P2 的值应是 2 个字节的 BER-TLV 标记。值‘4000’和‘FFFF’是 RFU。

当请求原始数据对象时，响应报文的数据字段应包含结构化数据对象的值。

当请求原始数据对象时，响应报文的数据字段应包含结构化数据对象的值，即包含其标签，长度和值的数据对象。

6.9.4 响应报文(标称情况)

如果 L_e 字段仅包含若干“0”，则对于短的长度在不超过 256 的范围内或者对于扩充的长度在不超过 65536 的范围内，所有要求的信息应被返回。

表 53 GET DATA 响应 APDU

数据字段	L_r(可以等于 L_e)字节
SWL-SW2	状态字节

6.9.5 状态条件

下列特定报警条件可能发生。

——SW1＝‘62’， 同时 SW2＝

- ‘81’:被返回数据的一部分可以被损坏。

下列特定差错条件可能发生。

——SW1＝‘67’， 同时 SW2＝

▪‘00’:错误的长度(空的 L_e 字段)

——SW1=‘69’, 同时 SW2=

▪‘82’:安全状态不被满足。

▪‘85’:使用的条件不被满足。

——SW1=‘6A’, 同时 SW2=

▪‘81’:功能不被支持。

▪‘82’:文件未被找到。

▪‘88’:引用的数据(数据对象)未被找到。

——SW1=‘6C’, 同时 SW2=

▪‘XX’:错误的长度(错误的 L_e 字段;‘XX’表示正确的长度)。

6.10 PUT DATA 命令

6.10.1 定义和范围

PUT DATA 命令可在当前上下文(例如，应用特定环境或当前 DF)范围内用于存储一个原始数据对象或者包含在结构化数据对象中的一个或多个数据对象。正确的存储功能(写一次和/或更新和/或添加)通过数据对象的定义和性质来引出。

注:例如，该命令可用来更新数据对象。

6.10.2 使用与安全的条件

仅当安全状态满足了通过功能用的上下文内的应用所定义的安全条件时，才能执行该命令。

6.10.3 命令报文

表 54 PUT DATA 命令 APDU

CLA	按 5.4.1 定义的
INS	‘DA’
P1-P2	见表 55
L_c 字段	后续数据字段的长度
数据字段	待写的参数和数据
L_e 字段	空

表 55 参数 P1-P2 的编码

值	含 义
‘0000’至‘003F’	RFU
‘0040’至‘00FF’	P2 中的 BER-TLV 标记(1 个字节)
‘0100’至‘01FF’	应用数据(专有编码)
‘0200’至‘02FF’	P2 中的简单 TLV 标记
‘0300’至‘3FFF’	RFU
‘4000’至‘FFFF’	P1-P2 中的 BER-TLV 标记(2 个字节)

存储应用数据

▪当 P1-P2 的值位于从‘0100’至‘01FF’的范围内时， P1-P2 的值应是被保留的一个标识符，它可在给定的应用上下文范围内用于卡内部测试和用于有意义的专有服务。

存储数据对象

▪当 P1-P2 的值位于从‘0040’至‘00FF’的范围内时， P2 的值应是单个字节的 BER-TLV 标记。值‘00FF’被保留，为了表示数据字段运载了 BER-TLV 数据对象。

▪ 当 P1-P2 的值位于从‘0200’至‘02FF’的范围时，P2 的值应是简单 TLV 标记。值‘0200’为 RFU。值‘02FF’被保留，为了表示数据字段运载了简单 TLV 数据对象。

▪ 当 P1-P2 的值位于从‘4000’至‘FFFF’的范围内时，P1-P2 的值应是 2 个字节的 BER-TLV 标记。值‘4000’和‘FFFF’为 RFU。

当提供了原始数据对象时，命令报文的数据字段应包含对应于原始数据对象的值。

当提供了结构化数据对象时，命令报文的数据字段应包含结构化数据对象的值，即包括其标记、长度和值的数据对象。

6.10.4 响应报文(标称情况)

表 56 PUT DATA 响应 APDU

数据字段	空
SW1-SW2	状态字节

6.10.5 状态条件

下列特定报警条件可能发生。

——SW1=‘63’， 同时 SW2=

▪‘CX’:计数器(成功的存储，但是在使用内部例行程序之后，‘X’≠‘0’表示重试数；‘X’=‘0’意味着没有计数器被提供)。

下列特定差错条件可能发生。

——SW1=‘65’， 同时 SW2=

▪‘81’:存储器故障(不成功的存储)

——SW1=‘67’， 同时 SW2=

▪‘00’:错误的长度(错误的 L_c 字段)

——SW1=‘69’， 同时 SW2=

▪‘82’:安全状态不被满足。

▪‘85’:使用的条件不被满足。

——SW1=‘6A’， 同时 SW2=

▪‘80’:数据字段中的不正确参数。

▪‘81’:功能不被支持。

▪‘84’:无足够的文件存储空间。

▪‘85’:L_c 与 TLV 结构不一致。

6.11 SELECT FILE 命令

6.11.1 定义和范围

成功的 SELECT FILE 在逻辑信道内(见本部分规范 5.5)设置当前文件。后续命令可以通过那个逻辑信道隐式地引用该当前文件。

选择 DF(它可以是 MF)时可将其设置为当前 DF。在这种选择之后，隐式当前 EF 可以通过那个逻辑信道来引用。

选择 EF 时设置了一对当前文件:EF 及其父辈文件。

在应答复位之后，MF 可通过基本逻辑信道(见 5.5.2)隐式地进行选择，除非在历史字节(见本部分规范第 8 章)中或在初始数据串(见本部分规范第 9 章)中有不同的规定。

注:利用 DF 名称的直接选择可以用来选择按照本规范第 5 部分所登记的应用。

6.11.2 使用与安全的条件

下列条件应该适用于每个开放逻辑信道。

除非另有规定，否则按照下列规则，正确执行命令可修改安全状态(见 5.2.1)。

——在当前 EF 被改变时，或在没有当前 EF 时，专门针对以前的当前 DF 的安全状态(如果有)被丢失。

——在当前 DF 是以前的当前 DF 的后代，或同代时，专门针对以前的当前 EF 的安全状

态被保持。

——在当前 DF 既不是以前的当前 DF 的后代，也不是同代时，专门针对以前的当前 EF 的安全状态被丢失。先前和新当前 DF 的所有共同祖先，所共用的安全状态被保持。

6.11.3 命令报文

表 57 SELECT FILE 命令 APDU

CLA	按 5.4.1 定义的
INS	'A4'
P1	选择控制，见表 58
P2	选择选项，见表 59
L_c 字段	空或后续数据字段的长度
数据字段	如果存在下列内容，则按照 P_1-P_2 ——文件标识符 ——MF 的路径 ——当前 DF 的路径 ——DF 名称
L_e 字段	空或在响应时期望的数据最大长度

表 58 选择控制 P1 的编码

b8	b7	b6	b5	b4	b3	b2	b1	含义
0	0	0	0	0	0	×	×	通过文件标识符来选择
0	0	0	0	0	0	0	0	——选择 MF、DF 或 EF (数据字段＝标识符或空)
0	0	0	0	0	0	0	1	——选择子女 DF (数据字段＝DF 标识符)
0	0	0	0	0	0	1	0	——根据当前 DF 选择 EF (数据字段＝EF 标识符)
0	0	0	0	0	0	1	1	——选择当前 DF 的父辈 DF (空的数据字段)

续表 58 选择控制 P_1 的编码

b8	b7	b6	b5	b4	b3	b2	b1	含义
0	0	0	0	0	1	×	×	通过 DF 名称来选择
0	0	0	0	0	1	0	0	——通过 DF 名称直接选择 (数据字段＝DF 名称)
0	0	0	0	0	1	0	1	RFU
0	0	0	0	0	1	1	0	RFU
0	0	0	0	0	1	1	1	RFU
0	0	0	0	0	1	×	×	通过路径来选择(见 5.1.2)
0	0	0	0	1	0	0	0	一由 MF 选择(数据字段＝路径 而无需 MF 的标识符)
0	0	0	0	1	0	0	1	一由当前 DF 选择(数据字段＝路径 而无需当前 DF 的标识符)
0	0	0	0	1	0	1	0	RFU
0	0	0	0	1	0	1	1	RFU
任何其他值								RFU

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

当 P1-P2=‘0000’时，如果文件标识符被提供，则在下列环境下，该文件标识符应是唯一的：

——当前 DF 的直接子女，

——父辈 DF，

——父辈 DF 的直接子女，

如果 P1-P2=‘0000’，如果数据字段为空或等于‘3F00’，则选择 MF。

当 P1=‘04’时，数据字段为 DF 名称，但可能权利被截断。当被支持时，带有相同数据字段的这种连续命令应选择名称与数据字段相匹配的 DF，即，以命令数据字段开始。如果卡接受了带有空数据字段的 SELECT FILE 命令，则全部 DF 或 DF 的子集可以连续地被选择。

注：关于卡所支持的选择方法见本部分规范 8.3.6。

表 59　选择选项 P2 的编码

b8	b7	b6	b5	b4	b3	b2	b1	含义
0	0	0	0	–	–	0	0	——第 1 个或唯一出现(选项)
0	0	0	0	–	–	0	1	——最后一个出现(选项)
0	0	0	0	–	–	1	0	——下一个出现(选项)
0	0	0	0	–	–	1	1	——先前一个出现(选项)
0	0	0	0	×	×	–	–	文件控制信息选项(见 5.1.5)
0	0	0	0	0	0	–	–	——返回 FC1，任选的样板
0	0	0	0	0	1	–	–	——返回 FCP 样板
0	0	0	0	1	0	–	–	——返回 FMD 样板
任何其他值								RFU

6.11.4　响应报文(标称情况)

如果 L_e 字段仅包含“0”，则对于短的长度在不超过 256 的范围内或对于扩充的长度在不超过 65536 的范围内，对应于选择选项的全部字节应被返回。

表 60　SELECT FILE 响应 APDU

数据字段 SW1-SW2	信息按照 P2(至多 L_e 个字节) 状态字节

6.11.5　状态条件

下列特定报警条件可能发生。

——SW1=‘62’，同时 SW2=

- ‘83’：选择的文件无效。
- ‘84’：FCI 格式化未按照 5.1.5。

下列特定差错条件可能发生。

——SW1=‘6A’，同时 SW2=

- ‘81’：功能不被支持。
- ‘82’：文件未找到。
- ‘86’：不正确的参数 P1-P2。
- ‘87’：L_c 与 P1-P2 不一致。

6.12　VERIFY 命令

6.12.1　定义和范围

VERIFY 命令启动从接口设备送入卡内的验证数据与卡内存储的引用数据(例如，口令)进行比较。

6.12.2　使用与安全的条件

安全状态可以被修改为比较的结果。不成功的比较可以记录在卡内(例如，为了限制使用引用数据的进一步企图数)。

6.12.3 命令报文

表 61 VERIFY 命令 APDU

CLA	按 5.4.1 定义的
INS	‘20’
P1	‘00’(其他值为 RFU)
P2	引用数据的限定符，见表 62
L_c 字段	空或后续数据字段的长度
数据字段	空或验证数据
L_e 字段	空

表 62 引用控制 P2 的编码

b8	b7	b6	b5	b4	b3	b2	b1	含义
0	0	0	0	0	0	0	0	——没有信息被给出
0	-	-	-	-	-	-	-	——全局引用数据 (例如 卡的口令)
1	-	-	-	-	-	-	-	——特定引用数据 (例如 DF 特定口令)
-	×	×	-	-	-	-	-	‘00’(其他值为 RFU)
-	-	-	×	×	×	×	×	——引用数据号

注:

1)在 VERIFY 命令无二义性地引用了保密数据的那些卡中，P2=‘00’被保留，用来指示没有特定的限定符被使用。

2)例如，引用数据号可以是一个口令号或一个短 EF 标识符。

3)当主体为空时，命令既可用来检索进一步允许的重试数(SW1-SW2=‘63CX’)或用来校验是否不要求验证(SW1-SW2=‘9000’)。

6.12.4 响应报文(标称情况)

表 63 VERIFY 响应 APDU

数据字段	空
SW1-SW2	状态字节

6.12.5 状态条件

下列特定报警条件可能发生。

——SW1=‘63’， 同时 SW2=

- ‘00’:没有信息被给出(验证失败)。
- ‘CX’:计数器(验证失败;‘X’表示进一步允许的重试数)。

下列特定差错条件可能发生。

——SW1=‘69’， 同时 SW2=

- ‘83’:鉴别方法被阻塞。
- ‘84’:引用的数据无效。

——SW1=‘6A’，同时 SW2=

- ‘86’:不正确的参数 P1-P2。
- ‘88’:引用的数据未被找到。

6.13 INTERNAL AUTHENTICATE 命令

6.13.1 定义和范围

INTERNAL AUTHENTICATE 命令启动卡使用从接口设备发送来的询问数据和在卡内存储的相关秘密(例如，密钥)来计算鉴别数据。

当该相关秘密被连接到 MF 时，命令可以用来鉴别整个卡。

当该相关秘密被连接到另一个 DF 时，命令可以用来鉴别那个 DF。

6.13.2 使用与安全的条件

命令的成功执行可能受先前命令(例如，VERIFY，SELECT FILE)或选择(例如，相关的秘密)的成功完成的支配。

当发布命令时，如果当前选择了密钥和算法，则该命令可以隐式地使用该密钥和算法。

已发出命令的次数可以记录在卡内，以便限制使用相关秘密或算法的进一步企图数。

6.13.3 命令报文

表 64 INTERNAL AUTHENTICATE 命令 APDU

CLA	按 5.4.1 定义的
INS	'88'
P1	在卡内引用的算法
P2	引用的秘密，见表 65
L_c 字段	后续数据字段的长度
数据字段	鉴别相关的数据(例如，询问)
L_e 字段	在响应中期望的字节最大数

P1='00'表示没有信息被给出。在发出命令之前引用的算法为已知，或在数据字段中提供。

P2='00'表示没有信息被给出。在发出命令之前引用的秘密为已知，或在数据字段中提供。

表 65 引用控制 P2 的编码

b8	b7	b6	b5	b4	b3	b2	b1	含义
0	0	0	0	0	0	0	0	——没有信息被给出
0	-	-	-	-	-	-	-	——全局引用数据 (例如，MF 特定密钥)
1	-	-	-	-	-	-	-	——特定引用数据 (例如，DF 特定密钥)
-	×	×	-	-	-	-	-	'00'(其他值为 RFU)
-	-	×	-	×	×	×	×	——秘密的号

注:例如，秘密的号可以是一个密钥号或一个短 EF 标识符。

6.13.4 响应报文(标称情况)

表 66 INTERNAL AUTHENTICATE 响应 APDU

数据字段	鉴别相关的数据 (例如，对询问的响应)
SW1-SW2	状态字节

注:响应报文可以包括对一步的应用安全功能有用的数据(例如:随机数)。

6.13.5 状态条件

下列特定差错条件可能发生。

——SW1='69'， 同时 SW2=

- '84':引用的数据无效。
- '85':使用的条件不被满足。

——SW1='6A'， 同时 SW2=

- '86':不正确的参数 P1-P2。
- '88':引用的数据未被找到。

6.14 EXTERNAL AUTHENTICATE 命令

6.14.1 定义和范围

EXTERNAL AUTHENTICATE 命令使用卡计算的结果(是或否)有条件地来更新安全状态，

而该卡的计算是以该卡先前发出(例如，通过 GET CHALLENGE 命令)的询问、在卡内存储的可能的秘密密钥以及接口设备发送的鉴别数据为基础的。

6.14.2 使用与安全的条件

命令的成功执行要求从卡获得的最后询问是有效的。

不成功的比较可以被记录在卡内(例如，为了限制使用引用数据的进一步企图数)。

6.14.3 命令报文

表 67 EXTERNAL AUTHENTICATE 命令 APDU

CLA	按 5.4.1 定义的
INS	'82'
P1	在卡内引用的算法
P2	秘密的引用，见表 68
L_e 字段	空或后续数据字段的长度
数据字段	空或鉴别相关的数据(例如，对询问的响应)
L_e 字段	空

P1='00'表示没有信息被给出。在发出命令之前引用的算法为已知，或在数据字段中提供。

P2='00'表示没有信息被给出。在发出命令之前引用的秘密为已知，或在数据字段中提供。

表 68 引用控制 P2 的编码

b8	b7	b6	b5	b4	b3	b2	b1	含　　义
0	0	0	0	0	0	0	0	——没有信息被给出
0	-	-	-	-	-	-	-	——全局引用数据(例如，MF 特定密钥)
1	-	-	-	-	-	-	-	——特定引用数据(例如，DF 特定密钥)
-	×	×	-	-	-	-	-	'00'(其他值为 RFU)
-	-	-	×	×	×	×	×	——秘密的号

注:

1)例如，秘密的号可以是一个密钥号或一个短 EF 标识符。

2)当主体为空时，命令既可用来检索进一步允许的重试数(SW1-SW2='63CX')或用来校验是否不要求验证(SW1—SW2='9000')。

6.14.4 响应报文(标称情况)

表 69 EXTERNAL AUTHENTICATE 响应 APDU

数据字段	空
SW1-SW2	状态字节

6.14.5 状态条件

下列特定报警条件可能发生。

——SW1='63'， 同时 SW2=

- '00':没有信息被给出(鉴别失效)。
- 'CX':计数器(鉴别失效;'X'表示进一步允许的重试数)。

下列特定差错条件可能发生。

——SW1='67'， 同时 SW2=

- '00':错误的长度(L_c 字段不正确)。

——SW1='69'， 同时 SW2=

- '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(在上下文中命令不被允许)。

——SW1=‘6A’, 同时 SW2=

- ‘86’:不正确的参数 P1-P2。
- ‘88’:引用的数据未被找到。

6.15 GET CHALLENGE 命令

6.15.1 定义和范围

GET CHALLENGE命令要求发出一个询问(例如,随机数)以便用于安全相关的规程(例如,EXTERNAL AUTHENTICATE 命令)。

6.15.2 使用与安全的条件

询问至少对下一个命令是有效的。在本规范本部分未规定进一步的条件。

6.15.3 命令报文

表 70 GET CHALLENGE 命令 APDU

CLA	按 5.4.1 定义的
INS	‘84’
P1-P2	‘0000’(其他值为 RFU)
L_c 字段	空
数据字段	空
L_e 字段	在响应中期望的最大长度

6.15.4 响应报文(标称情况)

表 71 GET CHALLENGE 响应 APDU

数据字段	询问
SWL-SW2	状态字节

6.15.5 状态条件

下列特定差错条件可能发生。

——SW1=‘6A’, 同时 SW2=

- ‘81’:功能不被支持。
- ‘86’:不正确的参数 P1-P2。

6.16 MANAGE CHANNEL 命令

6.16.1 定义和范围

MANAGE CHANNEL 命令打开和关闭逻辑信道。

开放功能打开了新逻辑信道,而不是基本逻辑信道。提供选项为了卡分配逻辑信道号,或为了将逻辑信道号供应给卡。

关闭功能显式地关闭逻辑信道,而不是基本逻辑信道。在成功关闭之后,该逻辑信道可加以重新使用。

6.16.2 使用与安全的条件

当由基本逻辑信道执行开放功能时,则在成功开放之后,MF 应隐式地被选择作为当前 DF,并且新逻辑信道的安全状态应和 ATR 之后的基本逻辑信道的安全状态相同。新逻辑信道的安全状态应和任何其他逻辑信道的安全状态分开。

当由不是基本逻辑信道的某一逻辑信道执行开放功能时,则在成功开放之后,曾发出命令的逻辑信道的当前 DF 应被选择作为当前 DF,并且新逻辑信道的安全状态应和曾执行开放功能的逻辑信道的安全功能相同。

在成功的关闭功能之后,与该逻辑信道相关的安全状态被丢失。

6.16.3 命令报文

表 72 MANAGE CHANNEL 命令 APDU

CLA	按 5.4.1 定义的
INS	‘70’
P1	P1＝‘00’打开逻辑信道 P1＝‘80’关闭逻辑信道(其他值为 RFU)
P2	‘00’，‘01’，‘02’，‘03’(其他值为 RFU)
L_c 字段	空
数据字段	空
L_e 字段	‘01’，如果 P1-P2＝‘0000’ 空，如果 P1-P2≠‘0000’

P1 的位 b8 用来表示开放功能或关闭功能;如果 b8 为“0”，则 MANAGE CHANNEL 应打开逻辑信道，如果 b8 为“1”，则 MANAGE CHANNEL 应关闭逻辑信道。

对于开放功能(P1＝“00”)，P2 的位 b1 和 b2 用来按照与类别字节(见本部分规范 5.4.1)相同的方式来编制逻辑信道号;P2 的其他位为 RFU。

——当 P2 的 b1 和 b2 为空时，则卡将分配在数据字段的位 b1 和 b2 中返回的逻辑信道号。

——当 P2 的 b1 和/或 b2 不为空时，它编码某一逻辑信道号，而不是基本逻辑信道，则卡将打开外部分配的逻辑信道号。

6.16.4 响应报文(标称情况)

表 73 MANAGE CHANNEL 响应 APDU

数据字段	逻辑信道号，如果 P1-P2＝‘0000’ 空，如果 P1-P2≠‘0000’
SW1-SW2	状态字节

6.16.5 状态条件

下列特定报警条件可能发生。

——SW1＝‘62’， 同时 SW2＝

- ‘00’:没有信息被给出。

7 面向传输的行业间命令

对于遵循本规范本部分的所有卡而言，应该不强制要求支持本部分描述的所有命令或所支持命令的所有选项。

当要求进行国际交换时，卡的系统服务及相关命令的集合和选项应按照本规范本部分第 9 章中的定义使用。

表 11 提供了本规范本部分定义的命令概要。

安全报文交换(见 5.6)对报文结构的影响不在本章中描述。

在 7.X.5 的每一条中所给出的差错和报警条件的列表不是穷举的(见 5.4.5)。

7.1 GET RESPONSE 命令

7.1.1 定义和范围

GET RESPONSE 命令用于从卡发送至接口设备、用可用的协议不能传送的那一些的 APDU(或 APDU 的一部分)。

7.1.2 使用与安全的条件

无条件。

7.1.3 命令报文

表 74 GET RESPONSE 命令 APDU

CLA	按 5.4.1 定义的
INS	‘C0’
P1-P2	‘0000’(其他值为 RFU)
L_c 字段	空
数据字段	空
L_e 字段	在响应中期望的数据最大长度

7.1.4 响应报文(标称情况)

如果 L_e 字段仅包含“0”，则对于短的长度在不超过 256，或者对于扩展的长度不超过 65536 的范围内，所有有效字节应被返回。

表 75 “GET RESPONSE”响应 APDU

数据字段	按照 L_e 的 APDU(的一部分)
SWL-SW2	状态字节

7.1.5 状态条件

下列特定正常处理可能发生。

——SW1＝‘61’， 同时 SW2＝

- ‘XX’:正常处理:更多的数据字节是有效的(‘XX’表示在后续 GET RESPONSE 仍然有效的额外字节数)。

下列特定报警条件能发生。

——SW1＝‘62’， 同时 SW2＝

- ‘81’:返回数据的一部分可能已损坏。

下列特定差错条件可能发生。

——SW1＝‘67’， 同时 SW2＝

- ‘00’: 长度错误(L_e 字段不正确)。

——SW1＝‘6A’， 同时 SW2＝

- ‘86’:参数 P1-P2 不正确。

——SW1＝‘6C’， 同时 SW2＝

- ‘XX’:长度错误(L_e 字段错误;‘XX’表示正确的长度)。

7.2 ENVELOPE 命令

7.2.1 定义和范围

ENVELOPE 命令用来发送那些不能由有效协议来发送的 APDU，或 APDU 的一部分，或任何数据串。

注:对于 SM 的 ENVELOPE 命令的用发在附录 F 中示出。

7.2.2 使用与安全的条件

无条件。

7.2.3 命令报文

表 76 ENVELOPE 命令 APDU

CLA	按 5.4.1 定义的
INS	‘C2’
P1-P2	‘0000’(其他值为 RFU)
L_c 字段	后续数据字段的长度
数据字段	APDU(的一部分)
L_e 字段	空或期望数据的长度

当对于发送数据串而言根据 T＝0 来使用 ENVELOPE 命令时，在 ENVELOPE 命令 ADPU 中的空数据字段意味着“数据串的结束”。

7.2.4 响应命令(标称情况)

表 77　ENVELOPE 响应 APDU

数据字段	空或按照 L_e 的 APDU(的一部分)
SWL-SW2	状态字节

注:状态字节属于 ENVELOPE 命令所有。在 ENVELOPE 命令的数据字段中所发送的命令的状态字节可能在 ENVELOPE 命令响应的数据字段中找到。

7.2.5　状态条件

下列特定差错条件可能发生。

——SW1＝‘67’，　同时 SW2＝

- ‘00’:长度错误的(不正确的 L_c 字段。)

8　历史字节

8.1　目的和一般结构

当按照本规范 7816-3 确定传输协议时，历史字节告诉外界如何使用该卡。

历史字节数(至多 15 个字节)按本规范 7816-3 进行规定并编码。

历史字节所运载的信息也可以在 ATR 文件(默认 EF 标识符＝‘2F01’)中找到。

如果存在，历史字节可由 3 个数据字段组成:

——一个必备的种类指示符(1 个字节)，

——任选的压缩 TLV 数据对象，

——一个有条件的状态指示符(13 个字节)。

8.2　种类指示符(必备的)

种类指示符是第 1 个历史字节。如果种类指示符等于‘00’，‘10’或‘8X’，则历史字节的格式应符合本规范本部分。

表 78　种类指示符的编码

值	含义
‘00’	状态信息应呈现在历史字节的结束处(不在 TLV 中)。
‘10’	在本部分规范 8.5 中规定
‘80’	状态信息(如果存在)包含在任选的压缩 TLV 数据对象中。
‘81’至‘8F’	RFU
其他值	专有的

8.3　任选的压缩 TLV 数据对象

压缩 TLV 数据对象的编码可从适合于带有标记＝‘4X’及长度＝‘0Y’的 RER-TLV 数据对象的 ASN.1 基本编码规则(见 ISO8825 和附录 D)推导出。这种数据对象的编码用‘XY’来代替，后面紧跟数据的 Y 字节。在本章中，‘X’系指标记号，‘Y’系指长度。

除本章中定义的数据对象外，历史字节还可以包含有本规范第 5 部分定义的数据对象。在这种情况下，在本规范第 5 部分中定义的标记和长度字段的编码应按上述要求进行修改。

当在本章中定义的压缩 TLV 数据对象出现在 ATR 文件中时，它们应按照 ASN.1 的基本编码规则进行编码(即标记＝‘4X’，长度＝‘0Y’)。

在本规范中未定义的所有应用类别标记被保留供 ISO 用。

8.3.1 国家/发行者指示符

当存在时，该数据对象表示一个国家或一个发行者。

该数据对象可通过‘1Y’或‘2Y’来引入。

表 79 国家/发行者指示符的编码

标记	长度	值
‘1’	可变	国家代码和国家数据
‘2’	可变	发行者标识号

标记‘1’后面紧跟着适合的长度(1 个 4 位字节)以及紧跟着 ISO3166 定义的表示国家的 3 个数字。后面紧跟着(奇数个 4 位字节)的数据由相关的国家标准团体进行选择。

标记‘2’后面紧跟着适合的长度(1 个 4 位字节)以紧跟着 ISO7812 第 1 部分定义的发行者标识号。如果发行者标识号包含有奇数个数字，则它应使用值为‘F’的 4 位字节正确地进行填充。

8.3.2 卡服务数据

该数据对象表示为了支持第 9 章描述的服务在卡内有效的方法。

该数据对象通过‘31’来引入。

当该数据对象不存在时，卡仅支持显式应用选择。

表 80 与应用无关的卡服务用的卡轮廓

b8	b7	b6	b5	b4	b3	b2	b1	含义
1	-	-	-	-	-	-	-	——通过全 DF 名称的直接应用选择
-	1	-	-	-	-	-	-	——通过部分 DF 名称的选择(见 9.3.2)

续表 80 与应用无关的卡服务用的卡轮廓

b8	b7	b6	b5	b4	b3	b2	b1	含义
								数据对象有效
-	-	1	-	-	-	-	-	——在 DIR 文件中
-	-	-	1	-	-	-	-	——在 ATR 文件中
								文件 I/O 服务，通过:
-	-	-	-	1	-	-	-	READ RECORD 命令
-	-	-	-	0	-	-	-	READ BINARY 命令
-	-	-	-	-	×	×	×	‘000’(其他值为 RFU)

注:DIR 文件和 ATR 文件的内容可以给出关于选择方法的信息。

8.3.3 初始访问数据

该任选的数据对象允许检索在本规范中定义的数据对象串。该数据对象所检索的串称作“初始数据串”。

该数据对象通过‘41’，‘42’或‘45’来引入。

在本章中所描述的任何命令 APDU 被假定为是在复位应答之后所发送的第 1 个命令。因此，在该点的有效数据不是可以以后可检索的。

8.3.3.1 长度＝‘1’

当仅提供 1 个字节的信息时，它表示为了检索初始数据串而执行的命令长度。执行的命令是按如下结构的 READ BINARY 命令。

表 81　当长度＝‘1’时，命令的编码

CLA	‘00’(5.4.1)
INS	‘B0’
P1-P2	‘0000’
L_c 字段	空
数据字段	空
L_e 字段	初始访问数据中的值字段的第1个字节并且是唯一的字节(表示被读的字节数)

8.3.3.2　长度＝‘2’

当提供 2 个字节的信息时，第 1 个字节表示文件结构(透明或记录)和被读的 EF 的短标识符。第 2 个字节表示为了检索初始数据串而执行的读命令长度。

表 82　第 1 个字节的结构

b8	＝0 面向记录的文件　＝1 透明文件
b7-b6	‘00’(其他值为 RFU)
b5-b1	EF 短标识符

当 b8＝0 时，执行的命令是按如下结构的 READ RECORD 命令。

表 83　当 b8＝0 时，命令的编码

CLA	‘00’(见 5.4.1)
INS	‘B2’
P1	‘01’
P2	短 EF 标识符(来自初始访问数据的第 1 个字节)后面紧跟着 b3-b2-b1＝110
L_c 字段	空
数据字段	空
L_e 字段	初始访问数据中的值字段的第 2 个字节和最后一个字节(表示被读的字节数)

当 b8＝1 时，执行的命令是按如下结构的 READ BINARY 命令。

表 84　当 b8＝1 时，命令的编码

CLA	‘00’(见 5.4.1)
INS	‘B0’
P1	初始访问数据中的第 1 个字节的值
P2	‘00’
L_c	字段空
数据字段	空
L_e 字段	初始访问数据中的值字段的第 2 个字节和最后一个字节(表示被读的字节数)

8.3.3.3　长度＝‘5’

在初始访问数据中找到的值由执行的命令 APDU 组成。当执行时，该命令在其响应数据字段中提供初始数据串。

8.3.4　卡发行者数据

该数据对象是任选的并且为可变长度。结构和编码由卡发行者进行定义。

该数据对象通过‘5Y’来引入。

8.3.5 预先发行的数据

该数据对象是任选的并且为可变长度。结构和编码在本规范本部分中不予定义。它可以用来表示:

——卡制造商

——集成电路类型

——集成电路制造商

——ROM 掩模版本

——操作系统版本

该数据对象通过‘6Y’来引入。

8.3.6 卡能力

该数据对象是任选的并且为可变长度。其值字段由第 1 个软件功能表，或者由前面的 2 个软件功能表，或者由 3 个软件功能表组成。

该数据对象通过‘71’，‘72’或‘73’来引入。

表 85 第 1 个软件功能表

b8	b7	b6	b5	b4	b3	b2	b1	含义
								DF 选择
1	-	-	-	-	-	-	-	——通过全 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	-	-	-	——隐式地
								EF 管理
-	-	-	-	-	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 86 示出了第 2 个软件功能表，它是数据编码类型。该数据编码类型也可以作为带有标记‘82’的文件控制参数中的第 2 个数据元而存在(见 5.1.5 中的表 2)

表 86 第 2 个软件功能表

(数据编码类型)

b8	b7	b6	b5	b4	b3	b2	b1	含义
-	×	×	-	-	-	-	-	写功能的行为
-	0	0	-	-	-	-	-	——一次写
-	0	1	-	-	-	-	-	——专有
-	1	0	-	-	-	-	-	——写‘或’
-	1	1	-	-	-	-	-	——写‘和’
-	-	-	-	-	×	×	×	数据单元长度（以 4 位字节 Nibble 位单位） (幂为 2，例如，001＝2 Nibble) (默认值＝1 个字节 byte)
-	-	-	×	×	-	-	-	0...00...(其他值为 RFU)

表 87 示出了第三个软件功能表

表 87 第三个软件功能表

b8	b7	b6	b5	b4	b3	b2	b1	含义

X - - - - - - -	0(1 为 RFU)
- 1 - - - - - -	——扩充的 L_c 和 L_e 字段
- - X - - - - -	0(1 为 RFU)
- - - X X - - -	逻辑信道管理
1	——通过卡
1	——通过接口设备
- - - 0 0 - - -	无逻辑信道
- - - - - X - -	0(1 为 RFU)
- - - - - - X Y	逻辑信道的最大数(＝2X＋Y＋1)

8.4 状态信息

状态信息由 3 个字节组成:卡生存状态(1 个字节)和 2 个状态字节 SW1-SW2。

卡生存状态的值‘00’表示没有卡生存状态被提供。值‘80’至‘FE’为专有的。所有其他值为 RFU。

SW1-SW2 的值‘9000’表示按 5.4.5 定义的进行正常处理。

SW1-SW2 的值‘0000’表示该状态未予表示。

如果种类指示符的值为‘80’，则状态信息可以呈现在压缩 TLV 数据对象中。在这种情况下，标记号为‘8’。当长度为‘1’时，则值为卡生存状态。当长度为‘2’时，则值为 SW1-SW2。当长度为‘3’时，则值为卡生存状态后紧跟着 SW1-SW2。长度的其他值被保留供 ISO 用。

8.5 DIR 数据引用

如果种类指示符为‘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规范本部分的范围。

9 与应用无关的卡服务

9.1 定义和范围

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”。其目的是提供在卡和接口设备之间的交换机制，它们(卡和接口设备)两者除了都遵循本规范外，它们彼此互不了解。

卡服务可通过下列内容的任何组合来支持。

——历史字节

——一个或多个保留 EF 的内容

——行业间命令的序列。

命令使用 CLA＝‘00’(见本部分规范 5.4.1)，即，没有安全报文交换和基本逻辑信道。

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。应用该使用与本规范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。

已定义了下列卡服务。

——卡标识服务——该服务允许接口设备标识卡以及如何处理。

——应用选择服务——该服务允许接口设备了解什么应用在卡(如果有)内活动以及如何选择和起动在内卡的应用。

——数据对象检索服务——该服务允许检索在本规范本部分或其他部分中定义的数据对象。本章描述了仅用于行业间数据对象的标准机制。

——文件选择服务——该服务允许选择无名的 DFs 和 EF。

——文件 I/O 服务——该服务允许访问存储在 EF 中的数据。

9.2　卡标识服务

该功能由卡根据其逻辑内容以及所有应用可能感兴趣的某些一般数据对象(例如，行业间数据对象)提供给外界的信息组成。称作“卡标识数据”的信息可由卡按历史字节以及可能直接在复位应答之后隐式选择的文件来给出。

对该文件的访问在初始访问数据信息中进行表示(见本部分规范 8.3.3)。

如果历史字节的初始访问数据不指示读命令，则对执行命令的响应包含有卡标识数据。

9.3　应用选择服务

一个应用可在卡内被隐式地选择或通过其名称被显式地选择。

9.3.1　隐式应用选择

当应用在卡内被隐式地选择时，按本规范第 5 部分定义的应用标识符应在卡标识数据中进行表示。如果该标识符在卡标识数据中不存在，则它应存在于 ATR 文件中。

9.3.2　直接应用选择

多应用环境的卡应能实际地响应由 SELECT FILE 命令所执行的直接应用选择，而该 SELECT FILE 命令规定了应用标识符作为 DF 名称。

应用标识符应在命令 APDU 中完整地予以提供。在通过部分 DF 名称的应用选择的情况下，与所建议的名称相匹配的下一个应用可以被选择，并且全 DF 名称象带有标记‘84’的文件控制参数那样可用于文件命令的响应报文(见 5.1.5 的表 2)。

执行的命令 APDU 如下。

表 88　直接应用选择用的命令编码

CLA	‘00’ (见 5.4.1)
INS	‘A4’
P1-P2	‘0400’
L_c 字段	数据字段的字节长度
数据字段	全或部分 DF 名称
L_e 字段	存在，仅包含了“0”

9.4　数据对象检索服务

与应用无关的国际交换所使用的数据对象在本规范本部分和其他部分中进行定义。

对那些数据对象的检索依赖于下列方法之一或两者：

——在卡标识数据中存在数据对象

——在 DIR 文件(路径＝‘3F002F00’)中存在数据对象或在 ATR 文件(路径＝‘3F002F01’)中存在数据对象。

通过间接的方法检索数据对象所必需的信息在本规范第 6 部分中进行定义。

9.5　文件选择服务

当 EF 的路径为已知时，被发出的 SELECT FILE 命令数等于路径长度除以 2，减 1(路径总是以当前 DF 开始)。

如果路径长度大于 4 个字节，则直到路径的所有有效 DF 标识符都已被使用为止，一个或多个 SELECT FILE 命令应使用下列命令 APDU 来执行。

表 89　使用文件标识符选择 DF 的命令的编码

CLA	‘00’ (见 5.4.1)
INS	‘A4’
P1-P2	‘0100’
L_c 字段	‘02’
数据字段	DF 标识符(来自路径的字节 3 和 4)
L_e 字段	空

最后一个选择并且可能是唯一的选择是带有下列命令 APDU 的 EF 选择

表 90　选择 EF 的命令的编码

CLA	‘00’ (见 5.4.1)
INS	‘A4’
P1-P2	‘0200’
L_c 字段	‘02’
数据字段	EF 标识符(路径的最后 2 个字节)
L_e 字段	空

9.6　文件 I/O 服务

一旦用于行业间交换的文件已经被选择，与交换相关的内容应通过下列命令 APDU 之一加以返回。

- 如果第 1 个软件功能表不存在，或者不指示支持面向记录的命令，则下列命令应予执行。

表 91　读透明文件的命令的编码

CLA	‘00’ (见 5.4.1)
INS	‘B0’
P1-P2	‘0000’
L_c	字段空
数据字段	空
L_e 字段	存在，仅包含了“0”

- 如果第 1 个软件功能表指示了支持面向记录的命令，则下列命令应予执行。

表 92　读面向记录文件的命令的编码

CLA	‘00’ (见 5.4.1)
INS	‘B2’
P1-P2	‘0005’
L_c 字段	空
数据字段	空
L_e 字段	存在，仅包含了“0”

附　录　A
(标准的附录)
通过 T＝0 传输 APDU 报文

A.1　情况 1

通过将值‘00’分配给 P3，把命令 APDU 映射到 T＝0 命令 TPDU。

命令 APDU	CLA	INS	P1	P2	
命令 TPDU	CLA	INS	P1	P2	P3＝‘00’

响应 TPDU 被映射到响应 APDU，而没有任何变化。

响应 TPDU	SW1	SW2
响应 APDU	SW1	SW2

A.2　情况 2 短的

在该情况下，L_e 的值从 1 至 256，并且按字节 B_1 进行编码(B_1＝‘00’意味着最大值，即 L_e＝256)。

命令 APDU 被映射到 T＝0 命令 TPDU，而没有任何变化。

C-APDU	CLA	INS	P1	P2	L_e＝B_1

C-TPDU	CLA INS P1 P2	P3＝B_1

按照接受的 L_e 和按照处理的命令，响应 TPDU 被映射到响应 APDU。

情况 2S.1——L_e 被接受

响应 TPDU 被映射到响应 APDU，而没有任何变化。

R-TPDU	L_e 字节	SW1 SW2

R-APDU	L_e 字节	SW1 SW2

情况 2S.2——L_e 明确地不被接受

如果长度是错误的， L_e 不能被不支持提供数据服务的卡所接受。

来自卡的响应 TPDU 表示卡放弃该命令被放弃是由于错误的长度:(SW1)＝‘67’。响应 TPDU 被映射到响应 APDU，而没有任何变化。

R-TPDU	SW1＝‘67’ SW2

R-APDU	SW1＝‘67’ SW2

情况 2S.3—L_e 不被接受， L_a 被指出

L_e 不能被卡所接受，并且卡指出了有效长度 L_a。

来自卡的响应 TPDU 指示该命令是由于错误的长度引起的，并且指示正确的长度为 L_a:(SW1)＝‘6C’，以及 SW2 编码了 L_a。

如果传输系统不支持重新发出同一命令的服务，它应将响应 TPDU 映射到响应 APDU，而没有任何变化。

R-TPDU	SW1＝‘6C’ SW2= L_a

R-APDU	SW1＝‘6C’ SW2= L_a

如果传输系统支持重新发出同一命令的服务，它应重新发出将值 L_a 分配给参数 P3 的同一命令 TPDU。

TPDU	CLA INS P1 P2	P3＝SW2

响应 TPDU 由 L_a 字节后面紧跟着 2 个状态字节组成。

如果 L_a 小于 L_e 或等于 L_e ，则响应 TPDU 被映射到响应 APDU，而没有任何变化。

R-TPDU	L_a 字节	SW1 SW2

R-APDU	L_a 字节	SW1 SW2

如果 L_a 大于 L_e ，则响应 TPDU 被映射到主体的前面的 L_e 字节并且被映射到状态字节 SW1-SW2。

R-TPDU	L_a 字节	SW1 SW2

R-APDU	L_e($<L_a$)字节	SW1 SW2

情况 2S.4-SW1-SW2＝‘9XYZ’，‘9000’除外

响应 TPDU 被映射到响应 APDU，而没有任何变化。

A.3 情况 3 短的

在该情况下，L_c 的值从 1 至 255，并且按字节 B_1 进行编码(B≠‘00’)。
命令 APDU 被映射到 T=0 命令 TPDU，而没有任何变化。

C-APDU	CLA INS P1 P2	$L_c=B_1$	L_c 字节
C-TPDU	CLA INS P1 P2	P3=B_1	L_c 字节

响应 TPDU 被映射到响应 APDU，而没有任何变化。

R-TPDU	SW1 SW2
R-APDU	SW1 SW2

A.4 情况 4 短的

在该情况下，L_c 的值从 1 至 255，并且按字节 B_1 进行编码，L_e 的值从 1 至 256，并且按字节 B_L 进行编码(B_L=‘00’意味着最大值，即，L_e=256)。
命令 APDU 被映射到截断主体的最后一个字节的 T=0 命令 TPDU。

C-APDU	CLA INS P1 P2	$B_1=L_c$	L_c 字节	B_L
C-TPDU	CLA INS P1 P2	P3=B_1	L_c 字节	

情况 4S.1—命令不被接受

来自卡的第 1 个响应 TPDU 表示卡放弃的命令:SW1=‘6X’，‘61’除外。
响应 TPDU 被映射到响应 APDU，而没有任何变化。

R-TPDU	SW1=‘6X’ SW2
R-APDU	SW1=‘6X’ SW2

情况 4S.2—命令被接受

来自卡的第 1 个响应 TPDU 表示卡执行的命令:SW1-SW2=‘9000’。
传输系统应通过分配值 L_e 给参数 P3 将 GET RESPONSE 命令 TPDU 发送给卡。

C-TPDU	CLA INS=GET RESPONSE P1 P2	P3=B_L

依赖于来自卡的第 2 个响应，传输系统应象上述情况 2S.1，2S.2，2S.3 和 2S.4 那样作出反应。

情况 4S.3—命令被接受，同时信息被增加

来自卡的第 1 个响应 TPDU 表示卡执行的命令，并且给出关于有效数据字节的长度:SW1=‘61’，并且 SW2 编码 L_x。

传输系统应通过分配最小 L_x 和 L_e 给参数 P3 将 GET RESPONSE 命令 TPDU 发送给卡。

C-TPDU	CLA INS=GET RESPONSE P1 P2	P3=min(L_e, L_x)

第 2 个响应 TDPU 被映射到响应 APDU，而没有任何变化。

R-TPDU	P3 字节	SW1 SW2
R-APDU	P3 字节	SW1 SW2

情况 4S.4——SW1-SW2＝‘9XYZ’，‘9000’除外

响应 TPDU 被映射到响应 APDU，而没有任何变化。

A.5 情况 2 扩充的

在该情况下，L_e的值从 1 至 65536，并且按 3 个字节进行编码:(B_1)＝‘00’，(B_2‖B_3)＝任何值(B_2和 B_3的值为‘0000’意味着最大值，即 L_e＝65536)。

C-APDU	CLA INS P1 P2	B_1＝‘00’ B_2B_3＝L_e

情况 2E.1——$L_e \leq 256$，B_1＝‘00’，B_2 B_3从‘0001’至‘0100’。

命令 APDU 应通过分配 B_3的值给参数 P3 而被映射到命令 TPDU。传输系统的处理应按照情况 2S 进行。

C-TPDU	CLA INS P1 P2	P3=B_3

情况 2E.2——L_e>256，B.1＝‘00’，B_2 B_3＝‘0000’或从‘0101’至‘FFFF’。

命令 APDU 应通过分配值‘00’给参数 P3 而被映射到命令 TPDU。

C-TPDU	CLA INS P1 P2	P3=‘00’

a)如果来自卡的第 1 个响应 TPDU 表示卡放弃的命令是由于错误的长度(SW1＝‘67’)引起的，则响应 TPDU 应被映射到响应 APDU，而没有任何变化。

R-TPDU	SW1＝‘67’ SW2
R-APDU	SW1＝‘67’ SW2

b)如果来自卡的第 1 个响应 TPDU 表示被放弃的命令是由于错误的长度引起的，并且表示正确的长度为 L_a(SW1＝‘6C’和 SW2＝L_a)，则传输系统应象情况 2S.3 描述的那样完成处理。

c)第 1 个响应 TPDU 是 256 个数据字节后面紧跟着 SW1-SW2＝‘9000’，这就意味着卡具有不大于 256 个数据字节，和/或不支持 GET RESPONSE 命令。则传输系统应将响应 TPDU 映射到响应 APDU，而没有任何变化。

R-TPDU	256 字节	SW1＝‘90’ SW2＝‘00’
R-APDU	256 字节	SW1＝‘90’ SW2＝‘00’

d)如果来自卡的第 1 个响应 TPDU 或后续的响应 TPDU 为 SW1＝‘61’，则 SW2 编码 L_x，而该 L_x是从卡得到的额外字节量(SW2 的值为‘00’表示 256 个额外字节或更多)，传输系统应计算 L_m＝L_e-(先前收到的响应 TPDU(s)的主体长度之和)，以获得被卡检索的其余字节量。

如果 L_m＝0，则传输系统应将所有收到的响应 TPDU 的主体与最后收到的响应 TPDU 的尾标一起并置到响应 APDU。

如果 L_m >0，则传输系统应通过分配最小 L_x和 L_m给参数 P3 来发出 GET RESPONSE 命令。来自卡的相应响应 TPDU 应进行处理:

——如果 SW1＝‘61’，根据情况 d)。

——如果 SW1＝‘90’，当 L_m＝0 时，如上所述。

A.6 情况 3 扩充的

在该情况下，L_c 的值从 1 至 65536，并且按 3 个字节进行编码:(B_1)＝‘00’，(B_2 ‖ B_3) ≠ ‘0000’。

C-APDU	CLA INS P1 P2	B_1＝‘00’ B_2B_3＝L_c	L_c 字节

情况 3E.1——0 < L_c <256，B_1 ＝ ‘00’，B_2 ＝ ‘00’，B_3 ≠ ‘00’。

命令 APDU 可通过分配 B_3 的值给参数 P_3 而被映射到命令 TPDU。

C-APDU	CLA INS P1 P2	P3＝B_3	L_c 字节

在该情况下，L_c 的值从 1 至 255，并且按 1 个字节进行编码。

响应 TPDU 被映射到响应 APDU，而没有任何变化。

R-TPDU	SW1 SW2

R-APDU	SW1 SW2

情况 3E.2—— L_c >255，B_1 ＝ ‘00’，B_2 ≠ ‘00’，B_3 ＝任何值。

如果传输系统不支持 ENVELOPE 命令，则它应返回差错响应 APDU，意味着长度是错误的:SW1＝‘67’。

R-TPDU	SW1＝‘67’ SW2

R-APDU	SW1＝‘67’ SW2

如果传输系统支持 ENVELOPE 命令，它应将 APDU 分解成小于 256 的长度段，并且将这些连续的段发送到连续 ENVELOPE 命令 TPDUs 的主体。

C-TPDU	CLA INS＝ENVELOPE P1 P2	P3	P3 字节

来自卡的第 1 个响应 TPDU 表示卡不支持 ENVELOPE 命令(SW1＝‘6D’)，该 TPDU 应被映射到响应 APDU，而没有任何变化。

R-TPDU	SW1＝‘6D’ SW2

R-APDU	SW1＝‘6D’ SW2

如果来自卡的第 1 个响应 TPDU 表示卡支持 ENVELOPE 命令(SW1-SW2＝‘9000’)，传输系统应按需要发送进一步的 ENVELOPE 命令。

R-TPDU	SW1-SW2=‘9000’

C-TPDU	CLA INS＝ENVELOPE P1 P2	P3	P3 字节

对应于最后一个 ENVELOPE 命令的响应 TPDU 被映射到响应 APDU，而没有任何变化。

R-TPDU	SW1 SW2

R-APDU	SW1 SW2

A.7 情况 4 扩充的

在该情况下，L_c 的值从 1 至 65536，并且按 3 个字节进行编码：(B_1)＝‘00’，(B_2 ‖ B_3) ≠‘0000’，以及 L_e 的值从 1 至 65536，并且按 2 个字节进行编码：(B_{L-1} ‖ B_L)＝任何值(B_{L-1} 和 B_L 的值为‘0000’意味着是最大值，即，L_e＝65536)。

C-APDU	CLA INS P1 P2	B_1＝‘00’ B_2B_3＝L_c	L_c 字节	$B_{L-1}B_L$＝L_e

情况 4E.1——L_c＜256，B_1＝‘00’，B_2＝ ‘00’，B_3 ≠‘00’。

命令 APDU 可通过截断最后 2 个字节 B_{L-1} 和 B_L 以及通过分配 B_3 的值给参数 P3 而被映射到命令 TPDU。

C-TPDU	CLA INS P1 P2	P3＝B_3	L_c 字节

在该情况下， L_c 的值从 1 至 255 字节，并且按 1 个字节进行编码。

a)如果在来自卡的第 1 个响应 TPDU 中 SW1＝‘6X’，则响应 TPDU 被映射到响应 APDU，而没有任何变化。

R-TPDU	SW1＝‘6X’ SW2

R-APDU	SW1＝‘6X’ SW2

b)如果在来自卡的第 1 个响应 TPDU 中 SW1＝‘90’，则，如果 L_e＜257($B_{L-1}B_L$ 的值为‘0001’至‘0100’)则传输系统应通过分配 B_L 的值给参数 P3 来发送 GET RESPONSE 命令 TPDU。传输系统的后续处理应按照上述情况 2S.1、2S.2、2S.3 和 2S.4 进行。

如果 L_e＞256($B_{L-1}B_L$ 的值从‘0000’或大于‘0100’)，则传输应通过分配值‘00’给数 P3 来发送 GET RESPONSE 命令 TPDU。传输系统的后续处理应按照上述情况 2E.2 进行。

c)如果在来自卡的第 1 个响应 TPDU 中 SW1＝‘61’，则传输系统应如上述情况 2E.2 d)规定的那样继续进行。

情况 4E.2——L_c＞255，B_1＝‘00’，B_2 ≠ ‘00’，B_3 ＝任何值。

传输系统应按照上述情况 3E.2 继续进行，直到已经完整地将命令 APDU 发送到卡为止。然后，它应如上述情况 4E.1 a)和 c)规定的那样继续进行。

附　录　B
(标准的附录)
通过 T＝1 传输 APDU 报文

B.1　情况 1

命令 APDU 被映射到Ⅰ块的信息字段，而没有任何变化。

命令 APDU	CLA　INS　P1　P2
信息字段	CLA　INS　P1　P2

在响应中收到的Ⅰ块的信息字段被映射到响应 APDU，而没有任何变化。

信息字段	SW1　SW2
命令 APDU	SW1　SW2

B.2　情况 2(短的和扩充的)

命令 APDU 被映射到Ⅰ块的信息字段，而没有任何变化。

C-APDU	CLA　INS　P1　P2	L_e 字段
信息字段	CLA　INS　P1　P2	L_e 字段

响应 APDU 由：

——在响应中收到的Ⅰ块的信息字段组成，

——或者在响应中收到的连续Ⅰ块的顺序连接的信息字段组成。这些块应予以链接。

信息字段	数据字段	SW1-SW2

或者顺序连接的信息字段

字段数据 ……	……
…… ……	……
…… 字段	SW1-SW2

R-APDU	数据字段	SW1-SW2

B．3 情况 3(短的和扩充的)

命令 APDU 没有任何变化地被映射到：

——某一Ⅰ块的信息字段，

——或应链接的连续Ⅰ块的顺序连接的信息字段。

C-APDU	CLA INS P1 P2	L_c 字段	数据字段

信息字段	CLA INS P1 P2	L_c 字段	数据字段

或者顺序连接的信息字段

CLA INS P1 P2	L_c 字段	数据…
…… …… ……		
…… …… 字段		

在响应中收到的Ⅰ块的信息字段被映射到响应 APDU，而没有任何变化。

信息字段	SW1 SW2

R-APDU	SW1 SW2

B.4 情况 4(短的和扩充的)

命令 APDU 没有任何变化地被映射到：

——某一Ⅰ块的信息字段，

——或者应链接的连续Ⅰ块的顺序连接的信息字段。

C-APDU	CLA INS P1 P2	L_c 字段	数据字段	L_e 字段

信息字段	CLA INS P1 P2	L_c 字段	数据字段	L_e 字段

或者顺序连接的并置的信息字段

CLA INS P1 P2	L_c 字段	数据
……	……	……
……	字段	L_e 字段

响应 APDU 由

——在响应中收到的Ⅰ块的信息字段组成，

——或者在响应中收到的连续Ⅰ块的顺序连接的信息字段组成。这些块应予以链接。

信息字段	数据字段	SW1-SW2

或者并置的信息字段

数据	……	……
……	……	……
……	字段	SW1-SW2

R-APDU	数据字段	SW1-SW2

附 录 C
(提示的附录)
记录指针管理

C.1 情况 1

情况1涉及在选择功能(显式的或隐式的)之后所发生的第1个命令。当前记录指针(CP)是未定义的。

命令 (READ RECODE)	记录 (在响应中)	CP 的位置 (在命令之后)
下一个(id＝aa)	第 1 个带有 id＝aa 如果未找到，则差错	记录读出 未定义
先前一个(id＝bb)	最后一个带有 id＝bb 如果未找到，则差错	记录读出 未定义
第 1 个(id＝cc)	第 1 个带有 id＝cc 如果未找到，则差错	记录读出 未定义
最后一个(id＝dd)	最后一个带有 id＝dd 如果未找到，则差错	记录读出 未定义
下一个(id＝00)	第 1 个	记录读出
先前一个(id＝00)	最后 1 个	记录读出
第 1 个(id＝00)	第 1 个	记录读出
最后一个(id＝00)	最后一个	记录读出
记录＃00	差错	未定义
记录＃ee	＃ee 如果未找到，则差错	未定义 未定义
P1＝‘00’，P2＝XXXX X101	差错	未定义

v＝‘00’，P2＝XXXX X110	差错	未定义
＃jj，P2＝XXXX X101	＃jj 至最后一个 如果＃jj 未找到，则差错	未定义 未定义
＃kk，P2＝XXXX X110	最后一个至＃kk 如果＃kk 未找到，则差错	未定义 未定义

C2 情况 2

情况 2 涉及后续命令。当前记录指针(CP)是被定义的。

命令 (READ RECODE)	记录 (在响应中)	CP 的位置 (在命令之后)
下一个(id＝aa)	下一个带有 id＝aa 如果没有下一个，则差错	记录读出 未变化
先前一个(id＝bb)	先前一个带有 id＝bb 如果没有下一个，则差错	记录读出 未变化
第 1 个(id＝cc)	第 1 个带有 id＝cc 如果未找到，则差错	记录读出 未变化
最后一个(id＝dd)	最后 1 个带有 id＝dd 如果未找到，则差错	记录读出 未变化
下一个(id＝00)	CP＋1 如果 CP＝最后一个，则差错	先前 CP＋1 未变化
先前一个(id＝00)	CP－1 如果 CP＝第 1 个，则差错	先前 CP－1 未变化
第 1 个(id＝00)	第 1 个第 1 个记录	读出
最后 1 个(id＝00)	最后 1 个最后一个	记录
记录＃00	CP	未变化
记录＃ee	＃ee 如果未找到，则差错	未变化 未变化
P1＝‘00’，P2＝XXXX X101	CP 至最后 1 个	未变化
P1＝‘00’，P2＝XXXX X110	最后 1 个至 CP	未变化
＃jj， P2＝XXXX X101	＃jj 至最后 1 个 如果＃jj 未找到，则差错	未变化 未变化
＃kk， P2＝XXXX X110	最后 1 个至＃kk 如果＃kk 未找到，则差错	未变化 未变化

附　录　D
(提示的附录)
使用 ANS.1 基本编码规则

D.1　BER-TLV 数据对象

每个 BER-TLV 数据对象(见 ISO8825)应由 2 或 3 个连续字段组成。

——标记字段 T 由一个或多个连续字节组成。它编码了类别、类型和编号。

——长度字段由一个或多个连续字节组成。它编码了整数 L。

——如果 L 不为空，则值字段 V 由 L 个连续字节组成。如果 L 为空，则数据对象为空：没有值字段。

本规范既不使用‘00’作为标记值，也不使用‘FF’作为标记值。

注:在 BER-TLV 数据对象之前、之间或之后，没有任何含义的‘00’或‘FF’字节可以出现(例如，由于擦除的或修改的 TLV 编码数据对象所引起)。

D.2　标记字段

标记字段中的引导字节的位 b8 和 b7 应编码标记类别，即，数据对象的类别。

——b8-b7＝00 引入全局类别的标记。

——b8-b7＝01 引入应用类别的标记。

——b8-b7＝10 引入上下文特定类别的标记。

——b8-b7＝11 引入专用类别的标记。

标记字段中的引导字节的位 b6 应编码标记类型，即，数据对象的类型。

——b6＝0 引入原始数据对象。

——b6＝1 引入结构化数据对象。

如果引导字节的位 b5 至 b1 不是都置为“1”，则它们应编码等于标记编号的一个整数，而该标记编号位于从 0 至 30 的范围内。然后标记字段由单个字节组成。

另一种方法(在引导字段中 b5 到 b1 置为“1”)标记字段应继续 1 个或多个后续字节。

——每个后续字节的 b8 应置为“1”，除非它是最后一个后续字节。

——第 1 个后续字节的位 b7 至 b1 应不是都置为“0”。

——第 1 个后字节的位 b7 至 b1，后面紧跟着每个进一步的后续字节的位 b7 至 b1，一直到并包括最后一个后续字节的位 b7 至 b1 都应编码等于标记编号的一个整数(因此是精确的正数)。

D.3 长度字段

在短形式中，长度字段由单个字节组成，其中位 b8 应置为“0”，并且位 b7 至 b1 应编码等于值字段中的字节数的一个整数，因此从 0 至 127 的任何长度可以利用 1 个字节来编码。

在长形式中，长度字段由一个引导字节组成，其中位 b8 应置为“1”，并且位 b7 至 b1 应不是全相同，因此，编码的一个正整数等于在长度字段中的后续字节数。那些后续字节应编码等于在值字段中的字节数的一个整数。因此在 APDU 限制(高达 65536)范围内的任何长度可以利用 3 个字节来编码。

注:本规范不使用 ASN.1 基本编码规则所规定的不定长度(见 ISO8825)。

D.4 值字段

在本规范本部分中，某些原始 BER-TLV 数对象的值字段由 0 个、1 个或多个简单 TLV 数据对象组成。

任何其他原始 BER-TLV 数据对象的值字段由通过数据对象规范所确定的 0 个、1 个或多个数据元组成。

每个结构化 BER-TLV 数据对象的值字段由 0 个、1 个或多个 BER-TLV 数据对象组成。

附　录　E
(提示的附录)
卡轮廓的举例

E.1　引言

本附录定义了许多卡轮廓，以指导应用设计者选择命令用于其应用中。这些轮廓也可用来帮助规定卡内要求的特征。卡轮廓可以进行组合。

E.2　轮廓 M

该轮廓的卡至少具有如下特征和命令。

——文件结构

- 透明结构
- 带有固定长度记录的线性结构

——命令

- READ BINARY 和 UPDATE BINARY，同时
  P1，b8＝0，
  长度高达 256 个字节。
- READ RECODE 和 UPDATE RECORD，同时
  P2，b8 至 b4＝0，
  P2，b3＝1，
  P2，b3 b2 b1＝000，001，010 或 011，并且 P1＝0。
- SELECT FILE，同时

P1-P2＝‘0000’。

- VERIFY，同时
  P1-P2＝‘0001’或‘0002’。
- INTERNAL AUTHENTICATE，同时
  P1-P2＝‘0000’。

E.3 轮廓 N

该轮廓和 M 相同，加上在 SELECT FILE 命令中的附加选项 P＝‘04’。

E.4 轮廓 O

该轮廓的卡至少具有如下特征和命令。

——文件结构

- 透明结构
- 带有固定长度记录的线性结构。
- 带有可变长度记录的线性结构。
- 带有固定长度记录的循环结构。

——命令

- READ BINARY、WRITE BINARY 和 UPDATE BINARY，同时
  P1，b8＝0，
  长度高达 256 个字节。
- READ RECODE、WRITE RECORD 和 UPDATE RECORD，同时
  P2，b8 至 b4＝0，
  P2，b3＝1，
  P2，b3 b2 b1＝000，001，010 或 011，并且 P1＝0。
- APPEND RECORD，同时
  P1-P2＝‘0000’。
- SELECT FILE，同时
  P1＝‘00’、‘01’、‘02’、‘03’、‘04’或‘09’，
  P2＝‘00’。
- VERIFY，同时
  P1-P2＝‘0001’或‘0002’。
- INTERNAL AUTHENTICATE，同时
  P1-P2＝‘0000’。
- EXTERNAL AUTHENTICATE，同时
  P1-P2＝‘0000’。
- GET CHALLENGE，同时
  P1-P2＝‘0000’。

E.5 轮廓 P

该轮廓的卡至少有下列特征和命令。

——文件结构

- 透明结论

——历史字节

- 卡服务数据(＝‘3188’)。

- 初始访问数据(＝‘4164’)。

——命令

- READ BINARY 和 UPDATE BINARY，同时

P1，b8＝0，

长度高达 64 个字节。

- SELECT FILE，同时

  P1-P2＝‘0400’。

- VERIFY，同时

  P1-P2＝‘0001’或‘0002’。

- INTERNAL AUTHENTICATE，同时

  P1-P2＝‘0000’。

E.6 轮廓 Q

该轮廓的卡至少具有如下特征和命令。

——历史字节

- 初始访问数据(＝‘45’-得到)。
- 卡能力(＝‘7180’)。

——安全报文交换

——命令

- GET DATA 和 PUT DATA，同时

  标记在 P1-P2 中

- SELECT FILE，同时

  P1-P2＝‘0401’、‘0402’或‘0403’。

- VERIFY，同时

  P1＝‘00’

- INTERNAL AUTHENTICATE
- EXTERNAL AUTHENTICATE
- GET CHALLENGE。

附　录　F
(提示的附录)
用的安全报文交换

F.1　缩略语

下列编略语适用于本附录。

CC　　密码检验和
CG　　密码
CH　　命令循环(CLA　INS　P1　P2)
CR　　控制引用
FR　　文件引用
KR　　密钥引用
L　　长度
PB　　填充字节(‘80’后面紧跟着 0 至 K-1 次‘00’，其中 K 为块长度)
P1　　填充指示符字节
PV　　简明值
RD　　响应描述符
T　　标记
‖　　并置

对于所有举例，CLA 表示通过合适的值(‘0X’、‘8X’、‘9X’或‘AX’)使用的安全报文交换，其中 CLA 的 b4 置为“1”(见本部分规范 5.4.1 和表 9)。

F.2 使用密码校验和

为了表 4 和图 4 中定义的 4 种情况而示出了使用的密码校验和(见本部分规范 5.6.3.1)。

——情况 1——没有数据，没有数据

命令数据字段＝Tcc ‖ Lcc ‖ CC

CC(CLA 中 b3＝1)所覆盖的数据＝第 1 个并且是唯一的一个数据块＝CH ‖ PB

情况 1 的命令被交换成情况 3 的命令。

——情况 2——没有数据，有数据

命令数据字段＝Tcc ‖ Lcc ‖ CC

CC(CLA 中 b3＝1)所覆盖的数据＝第 1 个并且是唯一的一个数据块＝CH ‖ PB

响应数据字段＝T_{PV}(b1＝1) ‖ L_{PV} ‖ PV ‖ Tcc ‖ Lcc ‖ CC

CC 所覆盖的数据＝数据块＝T_{PV} (b1＝1) ‖ L_{PV} ‖ PV ‖ PB

——情况 3.a——有数据，没有数据

命令数据字段＝T_{PV}(b1=1) ‖ L_{PV} ‖ PV ‖ Tcc ‖ Lcc ‖ CC

CC(CLA 中 b3＝0)所覆盖的数据＝数据块＝T_{PV}(b1＝1) ‖ L_{PV} ‖ PV ‖ PB

——情况 3.b——有数据，没有数据

命令数据字段＝T_{PV}(b1＝0) ‖ L_{PV1} ‖ PV1 ‖ T_{PV2}(b1＝1) ‖ L_{PV2} ‖ PV2 ‖ Tcc ‖ Lcc ‖ CC

CC(CLA 中 b3＝1)所覆盖的数据＝数据块＝CH ‖ PB ‖ T_{PV2}(b1＝1) ‖ L_{PV2} ‖ PV2 ‖ PB

——情况 4——有数据，有数据

命令数据字段＝T_{PV}(b1＝1) ‖ L_{PV} ‖ PV ‖ Tcc ‖ Lcc ‖ CC

CC(CLA 中 b3＝0)所覆盖的数据＝数据块＝T_{PV}(b1＝1) ‖ L_{PV} ‖ PV ‖ PB

响应数据字段＝T_{PV}(b1＝1) ‖ L_{PV} ‖ PV ‖ Tcc ‖ Lcc ‖ CC

CC 所覆盖的数据＝数据块＝T_{PV}(b1＝1) ‖ L_{PV} ‖ PV ‖ PB

F.3 使用密码

密码的使用（见本部分规范 5.6.4）如下所示，有填充与不填充两种情况。

——情况 a——没有在 BER-TV 中编码的简单数据

命令数据字段=T_{CG} ‖ L_{CG} ‖ PI ‖ CG

CG 传送的数据=数据块=没有 BER-TV 编码的数据和填充字节，如果在 PI 中被指出

——情况 b——在 BER-TV 中编码的简单数据

命令数据字段=T_{CG} ‖ L_{CG} ‖ CG

CG 传送的数据=隐藏字节串=BER-TV 数据对象（按运算法则填充和它的操作模式）

F.4 使用控制引用

控制引用的用法（见本部分规范 5.6.5.1）如下。

命令数据字段=T_{CR} ‖ L_{CR} ‖ CR

此处 CR=T_{FR} ‖ L_{FR} ‖ FR ‖ T_{KR} ‖ L_{KR} ‖ KR

F.5 使用应答描述符

应答描述符的用法（见本部分规范 5.6.5.2）如下。

命令数据字段=T_{RD} ‖ L_{RD} ‖ RD

此处 RD=T_{PV} ‖ ’00’ ‖ T_{CC} ‖ ’00’

数据字段响应=T_{PV} ‖ L_{PV} ‖ PV ‖ T_{CC} ‖ L_{CC} ‖ CC

F.6 使用 ENVELOPE 命令

ENVELOPE 命令的用法（见本部分规范 7.2）如下。

命令数据字段=T_{CG} ‖ L_{CG} ‖ PI ‖ CG

CG 传送的数据=由 CH 开始的命令 APDU 和根据 PI 的填充字节

数据字段响应=T_{CG} ‖ L_{CG} ‖ PI ‖ CG

CG 传送的数据=响应 APDU 和根据 PI 的填充字节