ICS 03.220.20;35.240.15
R 07
备案号:

JT

中华人民共和国交通运输行业标准

JT/T 978.6—2015

城市公共交通IC卡技术规范 第6部分:安全

Technical specification on IC card for urban public transport—Part6:Security

2015-05-21 发布　　2015-07-15 实施

中华人民共和国交通运输部　发布

目　　次

前　言

JT/T 978《城市公共交通IC卡技术规范》由7部分组成：

——第1部分：总则；

——第2部分：卡片；

——第3部分：读写终端；

——第4部分：信息接口；

——第5部分：非接触接口通信；

——第6部分：安全；

——第7部分：检测项目。

本部分为JT/T 978的第6部分。

本部分按照GB/T 1.1—2009给出的规则起草。

本部分由中华人民共和国交通运输部运输服务司提出。

本部分由全国城市客运标准化技术委员会(SAC/TC 529)归口。

本部分起草单位：中国交通通信信息中心、交通运输部公路科学研究院、交通运输部科学研究院、北京市政交通一卡通有限公司、南京市市民卡有限公司、武汉城市一卡通有限公司、银川市公共交通有限公司、泰州市凤城一卡通有限公司、中钞信用卡产业发展有限公司、北京中电华大电子设计有限责任公司、武汉天喻信息产业股份有限责任公司、北京同方微电子有限公司、上海华虹集成电路有限责任公司、大唐微电子技术有限公司、恩智浦(中国)管理有限公司。

本部分主要起草人：汪宏宇、李岚、唐猛、郭炯光、梅新明、王刚、刘好德、张永军、邢钊、张必成、莫若、李勇、嵇云峰、丁吉、王睿、丹明波、孟庆云、雷雨、程跃、畅江。

城市公共交通 IC 卡技术规范

第 6 部分:安全

1 范围

JT/T 978 的本部分规定了与城市公共交通 IC 卡相关的电子现金安全体系、电子钱包安全体系、卡片安全、终端安全以及加密机制。

本部分适用于城市公共交通 IC 卡卡片、终端及密钥管理系统的设计与研发。

2 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件,仅注日期的版本适用于本文件。凡是不注日期的引用文件,其最新版本(包括所有的修改单)适用于本文件。

GB/T 16649.4 识别卡集成电路卡 第 4 部分:用于交换的结构、安全和命令

GB/T 18238.3 信息技术 安全技术 散列函数 第 3 部分:专用散列函数

GB/T 20269—2006 信息安全技术 信息系统安全管理要求

GB/T 20271—2006 信息安全技术 信息系统通用安全技术要求

GB/T 22239—2008 信息安全技术 信息系统安全等级保护基本要求

GB/T 25058—2010 信息安全技术 信息系统安全等级保护实施指南

GB/T 25070—2010 信息安全技术 信息系统等级保护安全设计技术要求

GB/T 27929 银行业务 采用对称加密技术进行报文鉴别的要求

GM/T 0002 SM4 分组密码算法

GM/T 0003 SM2 椭圆曲线公钥密码算法

GM/T 0004 SM3 密码杂凑算法

JT/T 904—2014 交通运输行业信息系统安全等级保护定级指南

JT/T 978.1 城市公共交通 IC 卡技术规范 第 1 部分:总则

JT/T 978.2 城市公共交通 IC 卡技术规范 第 2 部分:卡片

JT/T 978.3 城市公共交通 IC 卡技术规范 第 3 部分:读写终端

3 术语与定义

JT/T 978.1 及 GB/T 18238.3 界定的以及下列术语和定义适用本文件。

3.1

密钥管理系统 key management system

管理密钥/证书生成、发行、更新等业务的系统。

3.2

非对称加密技术 asymmetric cryptographic technique

采用公开变换(由公钥定义)和私有变换(由私钥定义)这两种相关变换的加密技术。

3.3

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。

3.4

数字签名　key expiry date

对数据的一种非对称加密变换,简称签名。

3.5

公钥　public key

非对称密钥对中可公开的密钥。在数字签名时,公钥用于验证。

3.6

私钥　private key

非对称密钥对中不可公开的密钥,在数字签名中,私钥用于签名。

3.7

认证　authentication

确认一个实体所宣称的身份的措施。

3.8

证书　certificate

由密钥管理系统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,形成的不可伪造的电子数据。

3.9

公钥证书　public key certificate

使用私钥对公钥信息签名后产生的证书,包括根公钥证书、发卡机构公钥证书和卡片公钥证书。

3.10

串联　concatenation

通过把第二个元素的字节添加到第一个元素的结尾,将两个元素连接起来。每个元素中的字节在结果串中的顺序和原来从卡片发到读写终端时的顺序相同,即高位字节在前。在每个字节中位按由高到低的顺序排列。

3.11

哈希结果　hash result

哈希函数的输出位串。

3.12

密钥生命周期　key life cycle

密钥管理的所有阶段,包括计划、生成、回收、销毁和存档。

3.13

报文　message

由终端向卡或卡向终端发出的,不含传输控制字符的字节串。

3.14

报文鉴别码　message authentication code

对交易数据及其相关参数进行运算后产生的代码,主要用于验证报文的完整性。

3.15

填充　padding

向数据串某一端添加附加位。

3.16

保密密钥　secret key

对称加密技术中仅供指定实体所用的密钥。

4 符号、代号和缩略语

下列符号、代号和缩略语适用于本文件。

AAC——应用认证密文(Application Authentication Cryptogram)

AC——应用密文(Application Cryptogram)

ADF——应用定义文件(Application Definition File)

AEF——应用基本文件(Application Elementary File)

AFL——应用文件定位器(Application File Locator)

AID——应用标识符(Application Identifier)

AIP——应用交互特征(Application Interchange Profile)

APDU——应用协议数据单元(Application Protocol Data Unit)

ARC——授权响应码(Authorization Response Code)

ARPC——授权响应密文(Authorization Response Cryptogram)

ARQC——授权请求密文(Authorization Request Cryptogram)

ATC——应用交易计数器(Application Transaction Counter)

b——二进制

$C := (A||B)$ ——将 m 位数字 B 和 n 位数字 A 进行链接,定义为:$C = 2^m A + B$

CA——认证中心(Certificate Authority)

CBC——密码块链接(Cipher Block Chaining)

CDA——复合动态数据认证/应用密文生成(Combined DDA/AC Generation)

CDOL——卡片风险管理数据对象列表(Card Risk Management Data Object List)

CLA——命令报文的类别字节(Class Byte of the Command Message)

C - MAC——命令—报文鉴别码(Command - Message Authentication Code)

cn——压缩数字型

DDA——动态数据认证(Dynamic Data Authentication)

DDOL——动态数据认证数据对象列表(Dynamic Data Authentication Data Object List)

DES——数据加密标准(Data Encryption Standard)

ECB——电子密码本(Electronic Code Book)

EF——基本文件(Elementary File)

FCI——文件控制信息 (File Control Information)

FCP——文件控制参数(File Control Parameter)

fDDA——快速 DDA(Fast DDA)

FMD——文件管理数据(File Management Data)

H: = Hash[MSG]——用 160 位的 HASH 函数对任意长度的报文 MSG 进行 HASH 运算。

HSM——硬件安全模块(Hardware Secure Module)

IC——集成电路(Integrated Circuit)

ICC——集成电路卡(Integrated Circuit Card)

IMK——发卡机构主密钥(Issuer Master Key)

INS——命令报文的指令字节(Instruction Byte of Command Message)

KEK/TK——密钥交换密钥/传输密钥(Key Exchange Key/Transport Key)

Key A——密钥的左半部分

Key B——密钥的右半部分

K_{ENC}——卡片独有的密钥,用于产生加密会话密钥

K_{DEK}——卡片独有的密钥,用于产生对称密钥的保密密钥

K_{MAC}——卡片独有的密钥,用于产生 C - MAC 会话密钥

KMC——对称主密钥,用于在个人化过程中分散密钥来产生 K_{ENC},K_{DEK},K_{MAC}

KMCID——对称主密钥标识符

K_S——过程密钥(Session Key)

L_{DD}——卡片动态数据长度(Length of the ICC Dynamic Data)

MAC——报文鉴别码(Message Authentication Code)

MDK——主密钥(Master Key)

MMYY——月、年(Month,Year)

n——数字型

N_{CA}——根公钥模长(Length of the Certification Authority Public Key Modulus)

N_I——发卡机构公钥模长(Length of the Issuer Public Key Modulus)

N_{IC}——卡片公钥模长(Length of the ICC Public Key Modulus)

P1——参数 1

P2——参数 2

PAN——主账号(Primary Account Number)

P_{CA}——根公钥(Certification Authority Public Key)

P_I——发卡机构公钥(Issuer Public Key)

P_{IC}——卡片公钥(ICC Public Key)

PEK/TK——PIN 加密密钥(PIN Exchange Key /Transport Key)

PIN——个人识别码(Personal Identification Number)

RID——注册的应用提供商标识(Registered Application Provider Identifier)

RSA——Rivest、Sharmir 和 Adleman 提出的一种非对称密钥算法的简称

SAM——安全认证模块(Secure Authentication Module)

S_{CA}——根私钥(Certification Authority Private Key)

SFI——短文件标识符(Short File Identifier)

SHA——安全哈希算法(Secure Hash Algorithm)

S_I——发卡机构私钥(Issuer Private Key)

S_{IC}——卡片公钥(ICC Private Key)

SM2——SM2 椭圆曲线公钥密码算法(Public Key Cryptographic Algorithm SM2 Based on Elliptic Curves)

SM3——SM3 密码杂凑算法(SM3 Cryptographic Hash Algorithm)

SM4——SM4 分组密码加密算法(SM4 Cryptographic Algorithm)

SW——命令返回状态字(Status Word)

Verify(P_K)[X,S]——用公钥 P_K,通过非对称算法,对数据块 X 的签名结果 S 进行验证

TC——交易证书(Transaction Certificate)

TLV——表示标签、长度以及值的组合(Tag Length Value)

TVR——终端验证结果(Terminal Verification Results)

UDK——子密钥(Unique Key)

$X := Recover(P_K)[Y]$——用公钥 P_K,通过非对称可逆算法,对数据块 Y 进行恢复

$X := ALG^{-1}(K)[Y]$——用密钥 K,通过 64 位或 128 位分组加密方法,对 64 位或 128 位数据块 Y 进行解密

$Y:=ALG(K)[X]$——用密钥 K,通过 64 位或 128 位分组加密方法,对 64 位或 128 位数据块 X 进行加密

$Y:=Sign(S_K)[X]$——用私钥 S_K,通过非对称可逆算法,对数据块 X 进行签名

$A=B$——数值 A 等于数值 B

$A\equiv B \bmod n$——整数 A 与 B 对于模 n 同余,即存在一个整数 d,使得 $(A-B)=dn$

$A \bmod n$——A 整除 n 的余数,即:唯一的整数 r,$0\leqslant r<n$,存在一个整数 d,使得 $A=dn+r$

$A:=B$——A 被赋予数值 B

$A\oplus B$——A 与 B 进行异或计算

5 电子现金安全体系

5.1 证书和密钥

5.1.1 根公钥文件命名

5.1.1.1 概述

密钥管理系统生成根公钥及根私钥,根据算法不同(国际通用密码算法简称国际算法,国产密码算法简称国密算法),分为两种命名规则。

5.1.1.2 使用国际算法的根公钥文件

文件名格式为:11010000.JTAA。其中:

a) 11010000:4 字节服务标识,标识一个交通行业服务,将相应应用的私有应用标识扩展(PIX),右补十六进制'0'构成;

b) JT:标识国际算法;

c) AA:根公钥索引,以 0xAA 表示。

文件名示例如:11010000.JT01。

5.1.1.3 使用国密算法的根公钥文件

文件名格式为:11010000.JGAA。其中:

a) 11010000:4 字节服务标识,标识一个交通行业服务,将相应应用的私有应用标识扩展(PIX),右补十六进制'0'构成;

d) JG:标识国密算法;

c) AA:根公钥索引,以 0xAA 表示。

文件名示例如:11010000.JG01。

5.1.2 发卡机构证书请求文件命名

5.1.2.1 概述

由发卡机构密钥管理系统生成的公钥,按照以下格式生成发卡机构证书请求文件,根据算法不同,分为两种命名规则。

5.1.2.2 使用国际算法的发卡机构证书请求文件

文件名格式为:JTAAAAAA.ITNP。其中:

a) JT:固定前缀;

b) AAAAAA:记录号,发卡机构应使用该记录号申请发卡机构公钥证书;

c) IT:标识国际算法;

d) NP:固定后缀。

文件名示例如:JT123456.ITNP。

5.1.2.3 使用国密算法的发卡机构证书请求文件

文件名格式为:JTAAAAAA.IGNP。其中:

a) JT:固定前缀;

b) AAAAAA:记录号,发卡机构应使用该记录号申请发卡机构公钥证书;

c) IG:标识国密算法;

d) NP:固定后缀。

文件名示例如:JT123456.IGNP。

5.1.3 发卡机构证书文件命名

5.1.3.1 概述

发卡机构证书文件由认证中心对发卡机构证书请求文件签名后生成,根据算法不同,分为两种命名规则。

5.1.3.2 使用国际算法的发卡机构证书文件

名称格式为:AAAAAA.ITNN。其中:

a) AAAAAA:申请记录号,与签名发卡机构公钥数据中的记录号相同;

b) IT:标识国际算法;

c) NN:用来签发发卡机构公钥证书的根公钥的索引。

文件名示例如:110101.IT01。

5.1.3.3 使用国密算法的发卡机构证书文件

名称格式为:AAAAAA.IGNN。其中:

a) AAAAAA:申请记录号,与签名发卡机构公钥数据中的记录号相同;

b) IG:标识国密算法;

c) NN:用来签发发卡机构公钥证书的根公钥的索引。

文件名示例如:110101.IG01。

5.1.4 电子现金对称密钥

电子现金相关的对称密钥类型见表1。

表1 电子现金相关的对称密钥类型

密钥类型	用途	长度(字节)
应用密文主密钥	产生卡片应用密文子密钥,用于应用密文的产生和验证	16
安全报文认证(MAC)密钥	产生卡片MAC子密钥,用于安全报文鉴别码的产生和验证	16
安全报文加密密钥	产生卡片加密子密钥,用于加密解密安全报文	16
应用开通密钥	产生卡片扩展应用开通子密钥,用于与扩展应用相关的安全报文鉴别码的产生和验证,这个密钥用于增加指定的扩展应用扩展文件的记录	16
扩展应用管理密钥	产生卡片扩展应用管理子密钥,用于与扩展应用相关的安全报文鉴别码的产生和验证,这个密钥用于保护指定的扩展应用扩展文件中记录的信息	16

发卡机构主密钥可以分散出卡片子密钥,在交易过程中从子密钥派生出相应的过程密钥。

5.1.5 子密钥推导方法

5.1.5.1 国际算法子密钥推导流程

5.1.5.1.1 利用16字节的发卡机构主密钥(IMK)分散得出用于密文生成、发卡机构认证和安全报文的卡片子密钥,以主账号(PAN)和主账号序列号(若主账号序列号不存在,则用一个字节'00'代替)的最右16个数字作为输入数据X,生成16字节的卡片子密钥作为输出,见图1。

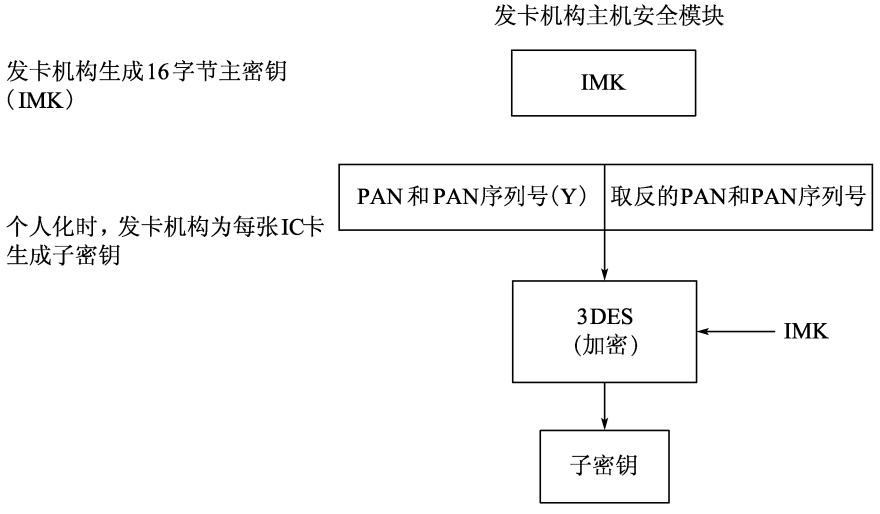


图1 国际算法的子密钥推导流程

5.1.5.1.2 国际算法的子密钥推导流程如下:

a) 若X的长度小于16个数字,X右对齐,在最左端填充十六进制的'00'以获得8字节的Y。若X的长度至少有16个数字,则Y由X的最右边的16个数字组成;

b) 计算2个8字节的数字:

ZL: = ALG(IMK)[Y]。

以及

ZR: = ALG(IMK)[Y ⊕('FF'||'FF'||'FF'||'FF'||'FF'||'FF'||'FF'||'FF')]。

并定义

Z: = (ZL||ZR)。

16字节的卡片子密钥就等于Z。对于DES算法,Z的每个字节的最低位应保证密钥的每一字节都有奇数个非0位(为了符合对DES密钥奇校验的要求)。

5.1.5.2 国密算法子密钥推导流程

5.1.5.2.1 利用16字节的发卡机构主密钥(IMK)分散得出用于密文生成、发卡机构认证和安全报文的卡片子密钥,以主账号(PAN)和主账号序列号(若主账号序列号不存在,则用一个字节'00'代替)的最右16个数字作为输入数据X,生成16字节的卡片子密钥作为输出,见图2。

5.1.5.2.2 国密算法的子密钥推导流程如下:

a) 若X的长度小于16个数字,X右对齐,在最左端填充十六进制的'00'以获得8字节的Y。若X的长度至少有16个数字,则Y由X的最右边的16个数字组成;

b) 计算:

Z: = ALG(IMK)[Y||(Y ⊕('FF'||'FF'||'FF'||'FF'||'FF'||'FF'||'FF'||'FF'))]。

16字节的卡片子密钥就等于Z。

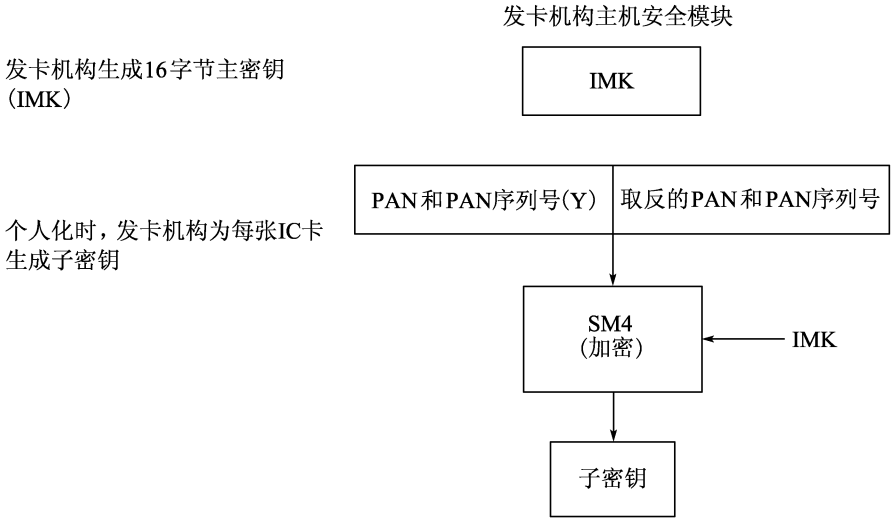


图 2　国密算法子密钥推导流程

5.1.6　过程密钥的产生

5.1.6.1　国际算法过程密钥的产生

国际算法的 MAC 和数据加密过程密钥的产生如下所述(在本节中将过程密钥的左半部分和右半部分称为“过程密钥 A”和“过程密钥 B”),见图 3。

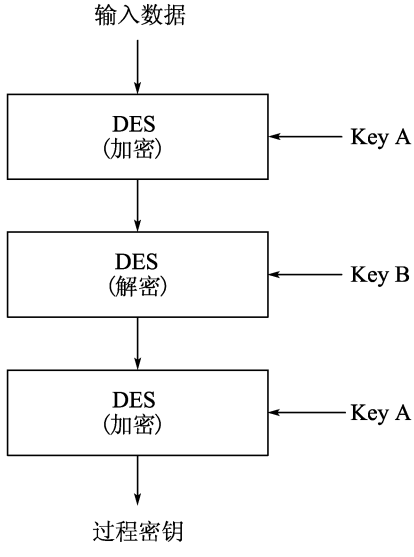


图 3　国际算法的单长度过程密钥的产生流程

5.1.6.1.1　单长度 DES 过程密钥

国际算法的单长度 DES 过程密钥产生流程如下：

a)　卡片/发卡机构使用 MAC 密钥或数据加密密钥进行算法处理(在本条中将密钥的左半部分和右半部分称为“Key A”和“Key B”)；

b)　将当前的 ATC 在其左边用十六进制数字‘00’填充到 8 个字节,用 Key A 和 Key B 对该数据作图 3 所示的 3DES 运算产生过程密钥 A。

Z：=3DES(Key)[‘00’||‘00’||‘00’||‘00’||‘00’||‘00’||‘00’||ATC]。

5.1.6.1.2 双长度 DES 过程密钥的产生

国际算法的双长度 DES 过程密钥产生流程如下：

a) 卡片/发卡机构使用 MAC 密钥或数据加密密钥进行算法处理(在本条中将密钥的左半部分和右半部分称为“Key A”和“Key B”)；

b) 将当前的 ATC 在其左边用十六进制数字‘00’填充到 8 个字节,用 Key A 和 Key B 对该数据作图 3 所示的 3DES 运算产生过程密钥 A。

将当前的 ATC 异或十六进制值 FFFF 后在其左边用十六进制数字‘00’填充到 8 个字节,使用相同方法对该数据作图 3 所示的 3DES 运算得到过程密钥 B。

ZL：=3DES(Key)[‘00’||‘00’||‘00’||‘00’||‘00’||‘00’||ATC]。

ZR：=3DES(Key)[‘00’||‘00’||‘00’||‘00’||‘00’||‘00’||(ATC ⊕‘FFFF’)]。

为符合对 DES 密钥奇校验的要求,DES 密钥每个字节的最低位应被设成能够保证密钥的 8 个或 16 个字节的每一个都有奇数个非 0 位。

5.1.6.2 国密算法过程密钥的产生

国密算法的 MAC 和数据加密过程密钥产生流程如下：

a) 卡片/发卡机构使用 MAC 密钥或数据加密密钥进行算法处理(密钥为“Key”)；

b) 将当前的 ATC 在其左边用十六进制数字‘00’填充到 8 个字节记为数据源 A,将当前的 ATC 异或十六进制值 FFFF 后在其左边用十六进制数字‘00’填充到 8 个字节记为数据源 B,将数据源 A 和数据源 B 串联,用选定的密钥对该数据作图 4 所示的运算产生过程密钥。

Z：=ALG(Key)[‘00’||‘00’||‘00’||‘00’||‘00’||‘00’||ATC||‘00’||‘00’||‘00’||‘00’||‘00’||‘00’||(ATC ⊕‘FFFF’)]。

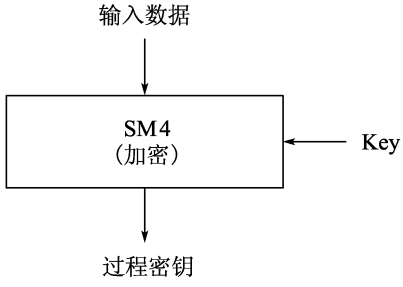


图 4 国密算法的过程密钥产生流程

5.2 动态数据认证

5.2.1 概述

5.2.1.1 终端采用公钥技术来验证卡片数据的一种方法。在动态数据认证过程中,终端验证卡片上的静态数据以及卡片产生的交易相关信息的签名,DDA 用以确认卡片上的发卡机构应用数据自卡片个人化后没有被非法篡改,并确认卡片的真实性,防止卡片的非法复制。

5.2.1.2 用于动态数据认证的记录应是 TLV 编码格式,并且 Tag =“70”。记录中用于动态数据认证的数据取决于记录所属文件的 SFI：

a) 对于 SFI 从 1 到 10 的文件,记录的 Tag("70")和记录长度不用于动态数据认证处理,读记录(READ RECORD)命令响应数据域中所有其他数据(SW1,SW2 除外)都参与动态数据认证;
b) 对于 SFI 从 11 到 30 的文件,记录的 Tag("70")和记录长度用于动态数据认证处理,因而读记录(READ RECORD)命令响应数据域中所有数据(SW1,SW2 除外)都参与动态数据认证;
c) 若用于动态数据认证的文件中的记录的 Tag 不是"70",则认为动态数据认证已经执行并失败,终端应设置 TSI 的"脱机数据认证已执行"位,以及 TVR 相应的"脱机静态数据认证失败"位"脱机动态数据认证失败"位或"CDA 失败"位。

5.2.2 动态数据认证方式

动态数据认证有以下两种方式:
a) 标准的动态数据认证,在卡片行为分析前执行。卡片根据卡内标识数据以及由动态数据认证数据对象列表所标识的终端数据生成一个数字签名;
b) 复合动态数据认证/应用密文生成,在生成应用密文(GENERATE AC)命令发出后执行。在交易证书或授权请求密文的情况下,卡片根据卡内标识数据得到一个数字签名,这些数据包括交易证书或授权请求密文,以及由卡片风险管理数据对象列表[对第一条生成应用密文(GENERATE AC)命令是 CDOL1,对第二条生成应用密文(GENERATE AC)命令是 CDOL2]标识的由终端生成的不可预知数 AIP 指明卡片支持的选项。

5.2.3 卡片数据元

支持动态数据认证的卡片应包含以下数据元:
a) 根公钥索引:该单字节数据元包含一个二进制数字,指明终端使用的根公钥;
b) 发卡机构公钥证书:该变长数据元由密钥管理系统提供给发卡机构。终端验证这个数据元时,按 5.2.6.3 描述的过程认证发卡机构公钥和其他的数据;
c) 卡片公钥证书:该变长数据元由发卡机构提供给卡片。终端验证这个数据元时,按 5.2.6.4 描述的过程认证卡片公钥和其他的数据;
d) 发卡机构公钥的余项:一个变长数据元。5.2.6.1 有进一步的解释;
e) 发卡机构公钥指数:一个由发卡机构提供的变长数据元,见 5.2.6.1;
f) 卡片公钥的余项:一个变长数据元。5.2.6.1 有进一步的解释;
g) 卡片公钥指数:一个由发卡机构提供的变长数据元。5.2.6.1 有进一步的解释;
h) 卡片私钥:一个存放在卡片内部的变长数据元,用来按 5.2.6.5 描述的过程生成签名的动态应用数据;
i) 签名的动态应用数据:一个由卡片使用同卡片公钥证书所认证的卡片公钥相对应的卡片私钥生成的变长数据元。它是一个数字签名,包含了 5.2.6.5 描述的存放在卡片中的或由卡片生成的以及终端中的关键数据元。

5.2.4 公钥体系结构

支持动态数据认证的终端应为每个注册的应用提供商标识存储 6 个根公钥,且应使同密钥相关的密钥信息和每一个密钥相关联(以使终端能在将来支持多种算法,允许从一个算法过渡到另一个,见 8.3)。在给定 RID 和卡片根公钥索引的情况下,终端应定位该公钥及相关信息。

动态数据认证使用的公钥体系结构见图 5。

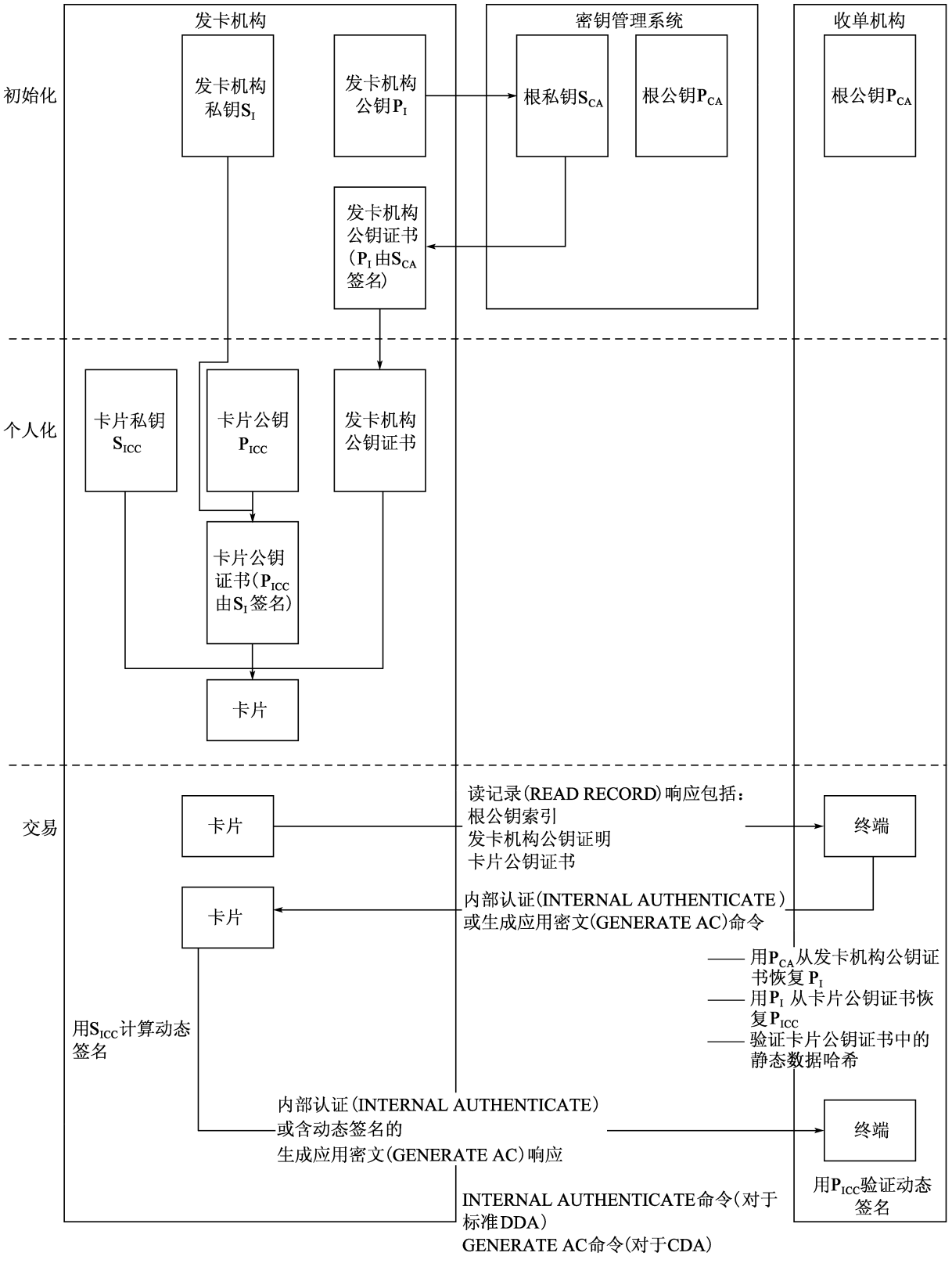


图 5　DDA 公钥体系结构

5.2.5 国际算法密钥和证书

5.2.5.1 概述

终端采用公钥算法验证卡片上的签名和证书来实现动态数据认证。公钥技术使用私钥产生加密数据(证书或签名),该加密数据可以被公钥解密而用于验证和数据恢复。RSA 公钥模的位长度应是 8 的倍数,最左边(高)字节的最左(高)一位为 1。所有的长度以字节为单位。

若卡片上的静态应用数据不是唯一的(比如卡片针对国际和国内交易使用不同的 CVM),卡片应支持多个卡片公钥证书(或静态数据签名),若被签名的静态应用数据在卡片发出后会被修改,卡片应支持卡片公钥证书(或静态数据签名)的更新。

5.2.5.2 公私钥对

5.2.5.2.1 概述

密钥管理系统和发卡机构使 RSA 算法产生根公私钥对、发卡机构公私钥对以及卡片公私钥对。

5.2.5.2.2 根公私钥对

公私钥对应分配一个唯一的根公钥索引。根公钥及其索引由收单机构加载到终端,根私钥由密钥管理系统保管并保证其私密性和安全性。要求如下:

a) 终端应存放根公钥、RID 和根公钥索引;

b) 终端应通过 RID 和根公钥索引定位根公钥,根公钥模长应在附录 A.2.1 中所定义的范围内,根公钥指数应等于 3 或 $2^{16}+1$。

5.2.5.2.3 发卡机构公私钥对

发卡机构产生发卡机构公私钥对,并从密钥管理系统获取发卡机构公钥证书。发卡机构将其公钥发送给密钥管理系统,密钥管理系统使用模长大于等于发卡机构公钥模长并且公钥有效期晚于发卡机构公钥有效期的根私钥对其进行签名。要求如下:

a) 卡片应包含发卡机构公钥证书及其用来验证发卡机构证书的根公钥索引;

b) 发卡机构公钥模长应小于等于根公钥最大模长,发卡机构公钥模长应在附录 A.2.1 中所定义的范围内。发卡机构公钥指数应等于 3 或 $2^{16}+1$;

c) 终端应通过 RID 和根公钥索引定位根公钥,并用根公钥从发卡机构证书恢复发卡机构公钥,再用发卡机构公钥恢复并验证卡片上的发卡机构应用数据。

5.2.5.2.4 卡片公私钥对

支持 DDA 还要求发卡机构为每张卡片产生卡片公私钥对,卡片私钥存放在卡片中的安全存贮区域,卡片公钥由发卡机构私钥签名,产生卡片公钥证书并存放在卡片中。要求如下:

a) 卡片公钥模长应小于等于发卡机构公钥模长,卡片公钥模长应在附录 A.2.1 中所定义的范围内。卡片公钥指数应等于 3 或 $2^{16}+1$;

b) 终端应通过 RID 和根公钥索引定位根公钥,并用认证中公钥从发卡机构公钥证书恢复发卡机构公钥,然后用发卡机构公钥从卡片公钥证书恢复卡片公钥,并用卡片公钥验证卡片的动态签名数据。

5.2.6 国际算法动态数据认证

5.2.6.1 安全要求

密钥和证书安全应满足如下要求:

a) 一张卡片应拥有自身的唯一的公私钥对,公私钥对由一个私有的签名密钥和相对应的公开的验证密钥组成。卡片公钥应存放在卡片上的公钥证书中;

b) 动态数据认证采用三层公钥证书方案。每一个卡片公钥由它的发卡机构认证,而密钥管理系统认证发卡机构公钥。为验证卡片的签名,终端应先通过验证两个证书来恢复和验证卡片公

钥,再用该公钥来验证卡片的动态签名;

c) 应按9.3.1中指明的签名方案将根私钥 S_{CA} 应用到表2中指定的数据,将发卡机构私钥 S_I 应用到表3中指定的数据,以分别获得发卡机构公钥证书和卡片公钥证书;

d) 根公钥模长为 N_{CA} 个字节。根公钥指数应等于3或 $2^{16}+1$;

e) 发卡机构的公钥模长为 N_I 个字节($N_I \leqslant N_{CA}$)。若 $N_I > (N_{CA}-36)$,则发卡机构公钥模被分成两部分,即一部分包含模中最高的 $N_{CA}-36$ 个字节(发卡机构公钥中最左边的数字);另一部分包含剩下的模中最低的 $N_I-(N_{CA}-36)$ 个字节(发卡机构公钥余项)。发卡机构公钥指数应等于3或 $2^{16}+1$;

f) 卡片的公钥模长为 N_{IC} 个字节($N_{IC} \leqslant N_I \leqslant N_{CA}$)。若 $N_{IC} > (N_I-42)$,则卡片公钥模被分成两部分,即一部分包含模中最高的 N_I-42 个字节(卡片公钥中最左边的数字);另一部分包含剩下的模中最低的 $N_{IC}-(N_I-42)$ 个字节(卡片公钥余项)。卡片公钥指数应等于3或 $2^{16}+1$;

g) 若卡片上的静态应用数据不唯一,则卡片应支持多卡片公钥证书,若被签名的静态应用数据在卡片发出后会被修改,则卡片应支持卡片公钥证书的更新;

h) 为完成动态数据认证,终端应先恢复和验证卡片公钥。卡片公钥认证需要的所有信息见表4,并存放在卡片中。除了RID可以从AID中获得外,其他信息可通过读记录(READ RECORD)命令得到。若缺少这些数据中的任意一项,则动态数据认证失败。

表2 由密钥管理系统签名的发卡机构公钥数据

字 段 名	长度(字节)	说 明	格 式
证书格式	1	十六进制,值为'02'	b
发卡机构识别号	4	主账号最左面的3个~8个数字(在右边补上十六进制数'F')	cn8
证书失效日期	2	MMYY,在此日期后,这张证书无效	n4
证书序列号	3	由密钥管理系统分配给这张证书的唯一的二进制数	b
哈希算法标识	1	标识用于在数字签名方案中产生哈希结果的哈希算法	b
发卡机构公钥算法标识	1	标识使用发卡机构公钥的数字签名算法	b
发卡机构公钥长度	1	标识发卡机构公钥模的字节长度	b
发卡机构公钥指数长度	1	标识发卡机构公钥指数的字节长度	b
发卡机构公钥或发卡机构公钥的最左边字节	$N_{CA}-36$	若 $N_I \leqslant N_{CA}-36$,这个字段包含了在右边补上了 $N_{CA}-36-N_I$ 个值为'BB'的字节的整个发卡机构公钥。若 $N_I > N_{CA}-36$,这个字段包含了发卡机构公钥最高位的 $N_{CA}-36$ 个字节	b
发卡机构公钥的余项	0或 $N_I-N_{CA}+36$	这个字段只有在 $N_I > N_{CA}-36$ 时才出现。它包含了发卡机构公钥最低位的 $N_I-N_{CA}+36$ 个字节	b
发卡机构公钥指数	1或3	发卡机构公钥指数等于3或 $2^{16}+1$	b
注:此为哈希算法输入。			

表 3　由发卡机构签名的卡片公钥数据

字　段　名	长度(字节)	说　　明	格　　式
证书格式	1	十六进制,值为'04'	b
应用主账号	10	主账号(在右边补上十六进制数'F')	cn20
证书失效日期	2	MMYY,在此日期后,这张证书无效	n4
证书序列号	3	由发卡机构分配给这张证书的唯一的二进制数	b
哈希算法标识	1	标识用于在数字签名方案中产生哈希结果的哈希算法	b
卡片公钥算法标识	1	标识使用在卡片公钥上的数字签名算法	b
卡片公钥长度	1	标识卡片公钥的模的字节长度	b
卡片公钥指数长度	1	标识卡片公钥指数的字节长度	b
卡片公钥或卡片公钥的最左边字节	$N_I - 42$	若 $N_{IC} \leqslant N_I - 42$,这个字段包含了在右边补上了 $N_I - 42 - N_{IC}$ 个值为'BB'的字节的整个卡片公钥。若 $N_{IC} > N_I - 42$,这个字段包含了卡片公钥最高位的 $N_I - 42$ 个字节	b
卡片公钥的余项	0 或 $N_{IC} - N_I + 42$	这个字段只有在 $N_{IC} > N_I - 42$ 时才出现,它包含了卡片公钥最低位的 $N_{IC} - N_I + 42$ 个字节	b
卡片公钥指数	1 或 3	卡片公钥指数等于 3 或 $2^{16}+1$	b
待认证的静态数据	变长	待认证的静态数据:一个用来验证卡片静态数据的签名。在卡片个人化阶段,使用发卡机构私钥签名的数据,保存在卡片中	b
注 1:认证过程的输入由被 AFL 标识的记录组成,其后跟有 AIP[若 AIP 被可选的静态数据认证标签列表(标签"9F4A")标识。若静态数据认证标签列表存在,它应仅包含标识 AIP 用的标签"82"]。 注 2:此为哈希算法输入。			

表 4　动态认证中的公钥认证所需的数据对象

标　　签	长度(字节)	说　　明	格式
—	5	注册的应用提供商标识	b
8F	1	根公钥索引	b
90	N_{CA}	发卡机构公钥证书	b
92	$N_I - N_{CA} + 36$	发卡机构公钥的余项(若存在)	b
9F32	1 或 3	发卡机构公钥指数	b
9F46	N_I	卡片公钥证书	b
9F48	$N_{IC} - N_I + 42$	卡片公钥的余项(若存在)	b
9F47	1 或 3	卡片公钥指数	b
—	变长	待认证的静态数据:一个用来验证卡片静态数据的签名。在卡片个人化阶段,使用发卡机构私钥签名的数据,保存在卡片中	—

5.2.6.2　根公钥的获取

终端读取根公钥索引,使用公钥索引和 RID,确认并取得存放在终端的根公钥的模、指数以及相应

算法。若终端没有存储与这个索引及 RID 相关联的密钥,则动态数据认证失败。

5.2.6.3 发卡机构公钥的获取

发卡机构公钥的获取流程如下:

a) 若发卡机构公钥证书的长度与 5.2.6.2 中根公钥模长度不同,则动态数据认证失败;

b) 使用根公钥和相应的算法按照 9.3.1 中指明的恢复函数恢复发卡机构公钥证书,获得在表 5 中指明的恢复数据。若恢复数据的结尾不等于“BC”,则动态数据认证失败;

c) 若恢复数据头不是“6A”,则动态数据认证失败;

d) 若证书格式不是“02”,则动态数据认证失败;

e) 将表 5 中的第 2 个到第 10 个数据元(即从证书格式直到发卡机构公钥或发卡机构公钥的最左边字节)从左到右连接,再把发卡机构公钥的余项加在后面(若有),最后是发卡机构公钥指数;

f) 使用指定的哈希算法(从哈希算法标识得到)对上一步的连接结果计算得到哈希结果;

g) 若上一步计算得到的哈希结果和恢复出的哈希结果不同,则动态数据认证失败;

h) 检验发卡机构识别号与主账号最左面的 3 个 ~8 个数字(允许发卡机构识别号在其后填充的“F”)。若不匹配,则动态数据认证失败;

i) 若证书失效日期在当天的日期之前,则动态数据认证失败;

j) 检验连接起来的 RID、根公钥索引、证书序列号是否有效。若失效,则动态数据认证失败;

k) 若发卡机构公钥算法标识无法识别,则动态数据认证失败;

l) 以上所有的检验都通过,连接发卡机构公钥的最左边字节和发卡机构公钥的余项(若存在),得到发卡机构公钥模,继续下一步取得卡片公钥。

表 5 从发卡机构公钥证书恢复数据的格式

字 段 名	长度(字节)	说 明	格 式
恢复数据头	1	十六进制,值为‘6A’	b
证书格式	1	十六进制,值为‘02’	b
发卡机构标识	4	主账号最左面的 3 个 ~8 个数字(在右边补上十六进制数‘F’)	cn8
证书失效日期	2	MMYY,在此日期后,这张证书无效。	n4
证书序列号	3	由密钥管理系统分配给这张证书的唯一的二进制数	b
哈希算法标识	1	标识用于在数字签名方案中产生哈希结果的哈希算法	b
发卡机构公钥算法标识	1	标识使用在发卡机构公钥上的数字签名算法	b
发卡机构公钥长度	1	标识发卡机构公钥的模的字节长度	b
发卡机构公钥指数长度	1	标识发卡机构公钥指数的字节长度	b
发卡机构公钥或发卡机构公钥的最左边字节	$N_{CA}-36$	若 $N_I \leqslant N_{CA}-36$,这个字段包含了在右边补上了 $N_{CA}-36-N_I$ 个值为‘BB’的字节的整个发卡机构公钥。若 $N_I > N_{CA}-36$,这个字段包含了发卡机构公钥最高位的 $N_{CA}-36$ 个字节	b
哈希结果	20	发卡机构公钥以及相关信息的哈希值	b
恢复数据结尾	1	十六进制,值为‘BC’	b

5.2.6.4 卡片公钥的获取

卡片公钥的获取流程如下：

a) 若卡片公钥证书的长度与在5.2.6.3获得的发卡机构公钥模长度不同，则动态数据认证失败；

b) 使用发卡机构公钥和相应的算法将9.3.1中指明的恢复函数应用到卡片公钥证书上，获得在表6中指明的恢复数据。若恢复数据的结尾不等于“BC”，则动态数据认证失败；

c) 若恢复数据头不是“6A”，则动态数据认证失败；

d) 若证书格式不是“04”，则动态数据认证失败；

e) 将表6中的第2个到第10个数据元(即从证书格式直到卡片公钥或卡片公钥的最左边字节)从左到右连接，再把卡片公钥的余项(若有)和卡片公钥指数加在后面，最后是AFL标示的待认证的静态数据，若静态数据认证标签列表存在，它应仅包含标识AIP用的标签“82”。若静态数据认证标签列表存在，并且其包含非“82”的标签，则动态数据认证失败；

f) 把指定的哈希算法(从哈希算法标识得到)应用到上一步的连接结果从而得到哈希结果；

g) 若上一步计算得到的哈希结果和恢复出的哈希结果不同，则动态数据认证失败；

h) 若恢复得到的主账号和从卡片读出的应用主账号不同，则动态数据认证失败；

i) 若证书失效日期在当天的日期之前，则动态数据认证失败；

j) 若卡片公钥算法标识无法识别，则动态数据认证失败；

k) 以上所有的检验都通过，连接卡片公钥的最左边字节和卡片公钥的余项(若存在)，得到发卡机构公钥模，继续下一步执行动态数据认证。

表6 从卡片公钥证书恢复数据的格式

字 段 名	长度(字节)	说 明	格 式
恢复数据头	1	十六进制，值为‘6A’	b
证书格式	1	十六进制，值为‘04’	b
应用主账号	10	主账号(在右边补上十六进制数‘F’)	cn20
证书失效日期	2	MMYY，在此日期后，这张证书无效	n4
证书序列号	3	由发卡机构分配给这张证书的唯一的二进制数	b
哈希算法标识	1	标识用于在数字签名方案中产生哈希结果的哈希算法	b
卡片公钥算法标识	1	标识使用在卡片公钥上的数字签名算法	b
卡片公钥长度	1	标识卡片公钥的模的字节长度	b
卡片公钥指数长度	1	标识卡片公钥指数的字节长度	b
卡片公钥或卡片公钥的最左边字节	N_I-42	若$N_{IC} \leq N_I-42$，这个字段包含了在右边补上了N_I-42-N_{IC}个值为‘BB’的字节的整个卡片公钥。若$N_{IC}>N_I-42$，这个字段包含了卡片公钥最高位的N_I-42个字节	b
哈希结果	20	卡片公钥以及相关信息的哈希值	b
恢复数据结尾	1	十六进制，值为‘BC’	b

5.2.6.5 标准动态数据认证

5.2.6.5.1 动态签名的生成

终端按上述的过程取得了卡片公钥，除了表4中指明的数据，动态数据认证所需的数据对象见表8。动态签名的生成流程如下：

a) 终端发出内部认证(INTERNAL AUTHENTICATE)命令,命令中包含由 DDOL 指定的数据元;
b) 卡片可包含 DDOL,但终端应有一个缺省的,由密钥管理系统指定的 DDOL,以防在卡片没有提供 DDOL 的情况下使用;
c) DDOL 应包含由终端生成的不可预知数(标签“9F37”,4 个字节的二进制数);
d) 若下面的任一情况发生,动态数据认证失败:
   1) 卡片和终端都不含有 DDOL;
   2) 卡片上的 DDOL 不包含不可预知数;
   3) 卡片上没有 DDOL 并且终端上缺省的 DDOL 不包含不可预知数。
e) 卡片使用卡片私钥和相应的算法并按 9.3.1 对表 7 中指明的数据生成数字签名,称为签名的动态应用数据。

表 7 需签名的动态应用数据

字 段 名	长度(字节)	说 明	格 式
签名的数据格式	1	十六进制,值为‘05’	b
哈希算法标识	1	标识用于产生哈希结果的哈希算法	b
卡片动态数据长度	1	标识卡片动态数据的字节长度 L_{DD}	b
卡片动态数据	L_{DD}	由卡片生成和/或存储在卡片上的动态数据	—
填充字节	$N_{IC}-L_{DD}-25$	$(N_{IC}-L_{DD}-25)$个值为‘BB’的填充字节	b
终端动态数据	变长	由 DDOL 指定的数据元连接而成	—

注 1:此为哈希算法的输入。
注 2:卡片动态数据的字节长度 L_{DD} 满足 $0 \leqslant L_{DD} \leqslant N_{IC}-25$。卡片动态数据的最左边的 3 个~9 个字节应由一个字节长的卡片动态数字长度后面跟随的 2 个~8 个卡片动态数字的值(标签“9F4C”,2 个~8 个二进制字节)组成。卡片动态数字是由一个由卡片生成的,随时间而变的参数,宜使用 ATC 作为卡片动态数字。

表 8 生成和检验动态签名所需要的其他数据对象

标 签	长度(字节)	值	格 式
9F4B	N_{IC}	签名的动态应用数据	b
9F49	变长	DDOL	b

5.2.6.5.2 动态签名的验证

动态签名验证流程如下:

a) 若签名的动态应用数据的长度不同于卡片公钥模的长度,则动态数据认证失败;
b) 为了获得在表 9 中指明的恢复数据,使用卡片公钥和相应的算法将 9.3.1 中指明的恢复函数应用到签名的动态应用数据上。若恢复数据的结尾不等于“BC”,则动态数据认证失败;
c) 若恢复数据头不是“6A”,则动态数据认证失败;
d) 若签名数据格式不是“05”,则动态数据认证失败;
e) 将表 9 中的第 2 个到第 6 个数据元(即从签名数据格式直到填充字节)从左到右连接,再把 DDOL 中指定的数据元加在后面;
f) 把指定的哈希算法(从哈希算法标识得到)应用到上一步的连接结果从而得到哈希结果;
g) 若上一步计算得到的哈希结果和恢复出的哈希结果不同,则动态数据认证失败;
h) 若以上所有的检验都成功,则动态数据认证成功。在表 9 中恢复得到的卡片动态数据中所包含的卡片动态数字应被存放在标签“9F4C”中。

表 9　从签名的动态应用数据恢复的数据格式

字　段　名	长度(字节)	说　　明	格　　式
恢复数据头	1	十六进制,值为'6A'	b
签名数据格式	1	十六进制,值为'05'	b
哈希算法标识	1	标识用于在数字签名方案中产生哈希结果的哈希算法	b
卡片动态数据长度	1	标识卡片动态数据的字节长度	b
卡片动态数据	L_{DD}	由卡片生成和/或存储在卡片上的动态数据	-
填充字节	$N_{IC}-L_{DD}-25$	$(N_{IC}-L_{DD}-25)$个值为'BB'的填充字节	b
哈希结果	20	动态应用数据以及相关信息的哈希值	b
恢复数据结尾	1	十六进制,值为'BC'	b

5.2.7　国密算法密钥和证书

5.2.7.1　概述

终端使用 SM2 公钥密码算法验证卡片上的签名和证书以实现动态数据认证。SM2 算法使用私钥产生证书或签名,该证书或签名可被公钥验证。SM2 公钥密码算法的数字签名不具备消息恢复功能,其公钥及相关信息以明文形式包含在证书数据中,后面附一个数字签名。

5.2.7.2　公私钥对

5.2.7.2.1　概述

密钥管理系统和发卡机构使用 SM2 算法产生根公私钥对、发卡机构公私钥对以及卡片公私钥对。

5.2.7.2.2　根公私钥对

密钥管理系统产生 SM2 算法公私钥对,每个公私钥对都应分配一个唯一的根公钥索引。根公钥及其索引由收单机构加载到终端,根私钥由密钥管理系统保管并保证其私密性和安全性。要求如下:

a)　终端应有足够空间存放根公钥及其对应的注册应用提供商标识(RID)和根公钥索引;

b)　终端应通过 RID 和根公钥索引定位根公钥。

5.2.7.2.3　发卡机构公私钥对

发卡机构应产生发卡机构公私钥对,并从密钥管理系统获取发卡机构公钥证书。发卡机构将其公钥发送给密钥管理系统,密钥管理系统使用公钥有效期晚于发卡机构公钥有效期的根私钥对其进行签名。要求如下:

a)　卡片应包含发卡机构公钥证书及其用来验证发卡机构证书的根公钥索引,发卡机构私钥由发卡机构保管并保证其私密性和安全性;

b)　终端应通过 RID 和根公钥索引定位根公钥,并用根公钥验证发卡机构证书,然后用发卡机构公钥验证卡片上的发卡机构应用数据。验证签名数据时,根据发卡机构证书的"发卡机构公钥签名算法标识"字段再次检查算法类型。

5.2.7.2.4　卡片公私钥对

发卡机构应为每张卡片产生卡片公私钥对,也可由卡片自己产生卡片公私钥对,卡片私钥存放在卡片中的安全存储区域,卡片公钥由发卡机构私钥签名,产生卡片公钥证书并存放在卡片中。

终端应通过 RID 和根公钥索引定位根公钥,并用认证根公钥验证发卡机构公钥证书,然后用发卡机构公钥验证卡片公钥证书,并用卡片公钥验证卡片的动态签名数据。验证签名数据时,根据卡片公钥证书的"卡片公钥签名算法标识"字段再次检查算法类型。

5.2.8 国密算法动态数据认证

5.2.8.1 安全要求

符合国密算法的密钥和证书安全应满足如下要求：

a) 一张卡片应拥有它自己的公私钥对，公私钥对应由一个私有的签名密钥和相对应的公开的验证密钥组成。卡片公钥应存放在卡片上的公钥证书中；

b) 动态数据认证采用三层公钥证书方案。卡片公钥由它的发卡机构认证，认证发卡机构公钥。为验证卡片的签名，终端应先通过验证两个证书来验证卡片公钥，再用该公钥来验证卡片的动态签名；

c) 应用根私钥 S_{CA} 对表 10 中指定的数据计算 SM2 签名获得发卡机构公钥证书（表 11），用发卡机构私钥 S_I 对表 12 中指定的数据计算 SM2 签名，以获得卡片公钥证书；

d) 若卡片上的静态应用数据是唯一的，则卡片应支持多卡片公钥证书，若被签名的静态应用数据在卡片发出后会被修改，则卡片应支持卡片公钥证书的更新；

e) 为完成动态数据认证，终端应先验证卡片公钥。卡片公钥认证需要的所有信息见表 13，并存放在卡片中。除了 RID 可以从 AID 中获得外，其他信息可通过读记录（READ RECORD）命令得到。若缺少这些数据中的任意一项，则动态数据认证失败。

表 10 由密钥管理系统签名的发卡机构公钥数据

字 段 名	长度(字节)	说 明	格 式
证书格式(记录头)	1	十六进制，值为‘12’	b
发卡机构标识	4	主账号最左面的 3 个～8 个数字（在右边补上十六进制数‘F’）	cn 8
证书失效日期	2	MMYY，在此日期后，这张证书无效	n 4
证书序列号	3	由密钥管理系统分配给这张证书的唯一的二进制数	b
发卡机构公钥签名算法标识	1	标识发卡机构公钥对应的数字签名算法。SM2 算法为‘04’	b
发卡机构公钥加密算法标识	1	标识发卡机构公钥对应的加密算法，保留项	b
发卡机构公钥参数标识	1	用于标识椭圆曲线参数，同时确定 N_I	b
发卡机构公钥长度	1	标识发卡机构公钥字节长度	b
发卡机构公钥	N_I	SM2 公钥是椭圆曲线上的一个点	b
注：此为待签名数据。			

表 11 发卡机构公钥证书的格式

字 段 名	长度(字节)	说 明	格 式
证书格式	1	十六进制，值为‘12’	b
发卡机构标识	4	主账号最左面的 3 个～8 个数字（在右边补上十六进制数‘F’）	cn 8
证书失效日期	2	MMYY，在此日期后，这张证书无效	n4
证书序列号	3	由密钥管理系统分配给这张证书的，唯一的二进制数	b

表 11（续）

字 段 名	长度(字节)	说 明	格 式
发卡机构签名公钥算法标识	1	标识发卡机构公钥对应的数字签名算法。SM2 算法为‘04’	b
发卡机构公钥加密算法标识	1	标识发卡机构公钥对应的加密算法,保留项	b
发卡机构公钥参数标识	1	用于标识椭圆曲线参数,同时确定 N_I	b
发卡机构公钥长度(字节)	1	标识发卡机构公钥字节长度	b
发卡机构公钥	N_I	如是 SM2 算法,该字段是椭圆曲线上的一个点	b
数字签名	N_{CA}	密钥管理系统对表 10 的数据计算的 SM2 签名 r||s	b

表 12 由发卡机构签名的卡片公钥数据

字 段 名	长度(字节)	说 明	格 式
证书格式	1	值为‘14’	b
应用主账号	10	主账号(在右边补上十六进制数‘F’)	cn 20
证书失效日期	2	MMYY,在此日期后,这张证书无效	n4
证书序列号	3	由发卡机构分配给这张证书的唯一的二进制数	b
卡片公钥签名算法标识	1	标识卡片公钥对应的数字签名算法	b
卡片公钥加密算法标识	1	标识卡片公钥对应的加密算法,保留项	b
卡片公钥参数标识	1	用于标识椭圆曲线参数,同时确定 N_{IC}	b
卡片公钥长度	1	标识卡片公钥的字节长度	b
卡片公钥	N_{IC}	若卡片公钥算法标识对应于 SM2,该字段为椭圆曲线上的一个点	b
待认证的静态数据	变长	待认证的静态数据:一个用来验证卡片静态数据的签名。在卡片个人化阶段,使用发卡机构私钥签名的数据,保存在卡片中	b

注 1:此为待签名数据。

注 2:对表 12 中数据进行 SM2 签名的结果是两个大整数 r 和 s,将字节串 r||s 附着在表 12 除“待认证的静态数据”外的数据之后就形成了用 SM2 签名的卡片公钥证书,证书的格式见表 14。

注 3:认证过程的输入由被 AFL 标识的记录组成,其后跟有 AIP[若 AIP 被可选的静态数据认证标签列表(标签“9F4A”)标识。若静态数据认证标签列表存在,它应仅包含标识 AIP 用的标签“82”]。

表 13 动态认证中的公钥认证所需的数据对象

标 签	长度(字节)	描 述	格 式
—	5	注册的应用提供商标识	b
8F	1	根公钥索引	b

表 13（续）

标　签	长度(字节)	描　述	格　式
90	$N_{CA}+N_I+14$	SM2 签名的发卡机构公钥证书数据,格式见表 11	b
9F46	$N_I+N_{IC}+20$	SM2 签名的卡片公钥证书数据,格式见表 14	b
—	变长	待认证的静态数据:一个用来验证卡片静态数据的签名。在卡片个人化阶段,使用发卡机构私钥签名的数据,保存在卡片中	—

5.2.8.2　根公钥的获取

终端读取根公钥索引。使用这个索引和 RID,终端应确认并取得存放在终端的根公钥的相关信息。若终端没有存储与这个索引及 RID 相关联的密钥,则动态数据认证失败。

5.2.8.3　发卡机构公钥的获取

终端获取的发卡机构证书数据见表 11。

发卡机构公钥的获取流程如下:

a）　获取并解析表 11 所示的发卡机构公钥证书数据。若失败,则静态数据认证失败;

b）　若证书格式的值不是“12”,则动态数据认证失败;

c）　检验发卡机构标识是否匹配主账号最左面的 3 个～8 个数字(允许发卡机构标识在其后补“F”)。若不一致,则动态数据认证失败;

d）　若证书失效日期在今天的日期之前,则动态数据认证失败;

e）　检验连接起来的 RID、根公钥索引、证书序列号是否有效。若无效,则动态数据认证失败;

f）　若发卡机构公钥算法标识不为“04”(SM2 算法),则动态数据认证失败;

g）　准备表 11 中前 9 个数据元(即表 10 数据);

h）　使用根公钥和相应的签名算法按照 9.4.3 中指明的验证函数对表 11 所定义的数字签名进行验证。若验证签名失败,则动态数据认证失败;

i）　以上所有的检验都通过,继续下一步取得卡片公钥。

5.2.8.4　卡片公钥的获取

终端获取的卡片公钥证书数据见表 14,卡片公钥的获取流程如下:

a）　获取并解析表 14 所示的经过卡片公钥证书数据。若失败,则静态数据认证失败;

b）　若证书格式不是“14”,则动态数据认证失败;

c）　若证书中的主账号和从卡片读出的应用主账号不同,则动态数据认证失败;

d）　若证书失效日期在今天的日期之前,则动态数据认证失败;

e）　准备表 14 中的前 9 个数据元以及静态数据(用于验证签名)。静态数据认证标签列表存在,并且其包含非“82”的标签,则动态数据认证失败。若发卡机构公钥算法标识不为“04”(SM2 算法),则动态数据认证失败;

f）　若卡片公钥算法标识不为“04”(SM2 算法),则动态数据认证失败;

g）　使用发卡机构公钥和相应的发卡机构签名算法将 9.4.3 中指明的验证函数对表 14 的数字签名进行验证。若验证签名失败,则动态数据认证失败;

h）　以上所有的检验都通过,继续下一步执行动态数据认证。

表 14　发卡机构使用 SM2 签名的卡片公钥证书的格式

字　段　名	长度(字节)	说　明	格　式
证书格式	1	十六进制,值为‘14’	b

表 14（续）

字 段 名	长度(字节)	说 明	格 式
应用主账号	10	主账号(在右边补上十六进制数‘F’)	cn 20
证书失效日期	2	MMYY,在此日期后,这张证书无效	n4
证书序列号	3	由发卡机构分配给这张证书的唯一的二进制数	b
卡片公钥签名算法标识	1	标识卡片公钥对应的数字签名算法	b
卡片公钥加密算法标识	1	标识卡片公钥对应的加密算法,保留项	b
卡片公钥参数标识	1	用于标识椭圆曲线参数,同时确定 N_{IC}	b
卡片公钥长度	1	标识卡片公钥的字节长度	b
卡片公钥	N_{IC}	若卡片公钥算法标识对应于 SM2,该字段是椭圆曲线上的一个点	b
数字签名	N_I	发卡机构对表 12 数据计算的 SM2 签名 r||s	b

5.2.8.5 标准动态数据认证

5.2.8.5.1 动态签名的生成

卡片使用 SM2 算法生成动态签名,动态签名的生成流程如下:

a) 终端发出内部认证(INTERNAL AUTHENTICATE)命令,命令中包含由 DDOL 指定的数据元素,这些数据元按照 JT/T 978.2 要求的规则连接在一起;

b) 卡片使用卡片私钥按照表 15 中指明的数据计算 SM2 签名,得到 SM2 签名动态应用数据,其格式见表 17。

表 15 需签名的动态应用数据

字 段 名	长度(字节)	说 明	格 式
签名的数据格式	1	值为‘15’表示用 SM2 签名	b
卡片动态数据长度	1	标识卡片动态数据的字节长度 L_{DD}	b
卡片动态数据	L_{DD}	由卡片生成和/或存储在卡片上的动态数据	–
终端动态数据	变长	由 DDOL 指定的数据元连接而成	–
注 1:此为待签名数据。 注 2:卡片动态数据的最左边的 3 个 ~9 个字节由一个字节长的卡片动态数字长度后面跟随的 2 个 ~8 个卡片动态数字的值(标签“9F4C”,2 个 ~8 个二进制字节)组成。 注 3:卡片动态数字是由一个由卡片生成的,随时间而变的参数,宜使用 ATC 作为卡片动态数字。			

除在表 13 中指明的数据外,动态数据认证所需的数据对象见表 16。

表 16 生成和检验动态签名所需要的其他数据对象

标 签	长度(字节)	说 明	格 式
9F4B	$N_{IC}+L_{DD}+2$	SM2 签名动态应用数据,格式见表 17	b
9F49	变长	DDOL	b

5.2.8.5.2 动态签名的验证

终端获取的签名动态应用数据的格式见表17,包括被签名的明文数据及数字签名。终端使用卡片的公钥验证动态应用数据的签名,若动态数据认证成功,表17中的卡片动态数据中所包含的卡片动态数字应被存放在标签“9F4C”中。

动态签名的验证流程如下:

a) 获取并解析表17所示的经过发卡机构签名的动态数据。若失败,则静态数据认证失败;

b) 若签名数据格式不是“15”,则动态数据认证失败;

c) 准备表17中的前3个数据元(即从签名数据格式直到卡片动态数据)及DDOL中指定的数据元(即表15数据)用于验证签名;

d) 使用卡片公钥和相应的卡片签名算法将9.4.3中指明的验证函数对表17的数字签名进行验证。若验证签名失败,则动态数据认证失败;

e) 以上所有的检验都成功,动态数据认证成功。终端获取卡片动态数据中所包含的卡片动态数字并存放在标签“9F4C”中。

表17 卡片使用SM2签名的动态应用数据的格式

字 段 名	长度(字节)	说 明	格 式
签名的数据格式	1	十六进制,值为‘15’	b
卡片动态数据长度	1	标识卡片动态数据的字节长度 L_{DD}	b
卡片动态数据	L_{DD}	由卡片生成和/或存储在卡片上的动态数据	–
数字签名	N_{IC}	卡片对表15中数据计算的SM2签名 r||s	b

5.3 应用密文和发卡机构认证

5.3.1 概述

卡片生成应用密文(TC、ARQC或AAC),以及发卡机构生成授权响应密文(ARPC)并由卡片校验的方法。

5.3.2 应用密文产生

5.3.2.1 数据源选择

一个应用密文由基于以下数据生成的报文鉴别码组成:

a) 引用卡片的DOL并通过生成应用密文(GENERATE AC)命令或其他命令从终端传输到卡片的数据;

b) 卡片内部访问的数据。

数据源的选择需包含在应用密文生成中见JT/T 978.2,最小数据元见表18;可选的应用密文生成数据源见表19。

表18 建议的应用密文生成中使用的最小数据集

值	来 源
授权金额(数字)	终端
其他金额(数字)	终端
终端国家代码	终端
终端验证结果	终端

表 18（续）

值	来　源
交易货币代码	终端
交易日期	终端
交易类型	终端
不可预知数	终端
应用交互特征	卡片
应用交易计数器	卡片

表 19　可选的应用密文生成数据源

值	来　源
卡片验证结果	卡片

5.3.2.2　国际算法应用密文生成

应用密文生成的方法是以一个唯一的 16 字节的卡片应用密文(AC)子密钥 MK_{AC} 以及按 5.3.2.1 的描述选择的数据作为输入,然后按以下流程计算得到的 8 字节的应用密文:

a) 从卡片应用密文(AC)子密钥 MK_{AC} 和两字节的卡片应用交易计数器作为输入,分散得到 16 字节的应用密文过程密钥 SK_{AC},使用 5.1.6.1 中指明的过程密钥分散函数;

b) 使用上一步分散得到的 16 字节的应用密文过程密钥并将 9.1.2 中指明的 MAC 算法应用到经选择的数据来生成 8 字节的应用密文。

5.3.2.3　国密算法应用密文生成

使用一个唯一的 16 字节卡片应用密文(AC)子密钥 MK_{AC},和 5.3.2.1 描述的数据源作为输入,按以下流程计算得到 8 字节的应用密文:

a) 以卡片应用密文(AC)子密钥 MK_{AC} 和两字节的卡片应用交易计数器作为输入,使用 5.1.6 描述的算法,生成 16 字节的应用密文过程密钥 SK_{AC};

b) 终端将 CDOL 中指定的终端数据通过生成应用密文命令传送给卡片。若 CDOL 中有要交易证书(TC)哈希结果,终端要将此数据放到命令数据域中;

c) 根据卡片风险管理的结果,卡片决定返回的密文类型为 TC、AAC 或 ARQC。生成密文的数据块:

   1) 交易证书(TC)哈希结果(若存在);

   2) 生成应用密文命令中送进卡片的数据。不包括 TC 哈希结果;

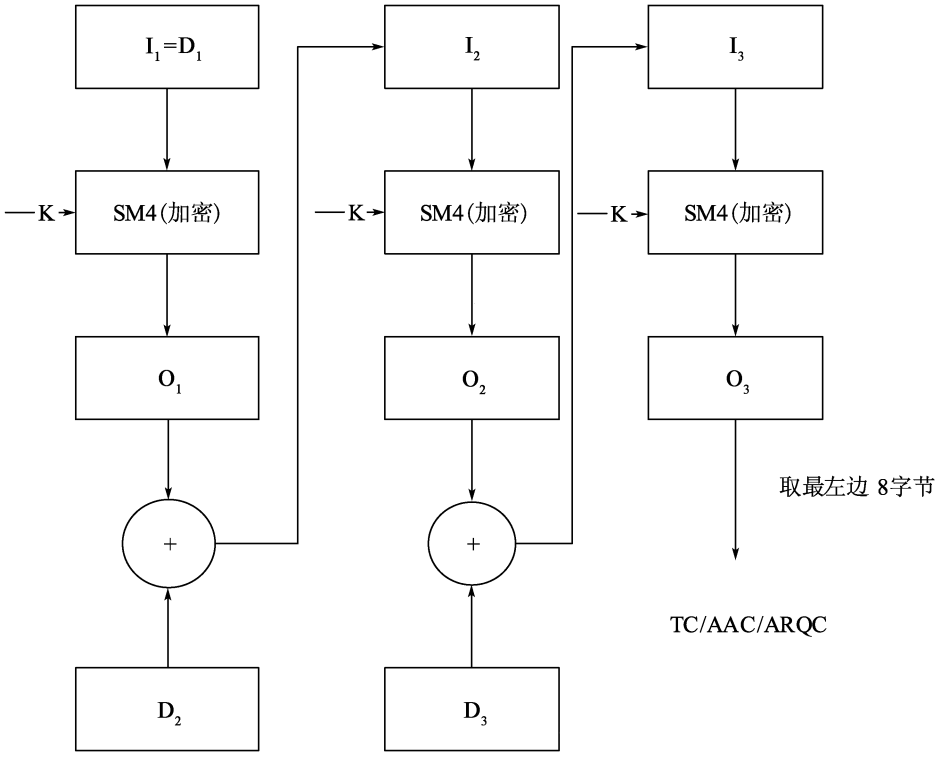
   3) 卡片内部数据。

d) 将上述数据块分成 16 字节一组:D1、D2、D3……;

e) 若最后一块数据块的长度为 16 字节,后面补 16 字节数据块:“80 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00”,若最后一块;数据块的长度小于 16 字节,后面补一个字节 80,若仍然不够 16 字节,补 00 直到 16 字节;

f) 见图 6,按照 9.2.2 中指明的 MAC 算法,使用过程密钥 SK_{AC} 计算生成应用密文(TC、ARQC 或 AAC);

g) 取上一步计算结果的左边 8 字节,得到 8 字节的密文。



说明：

I——输入；　　D——数据块；

O——输出；　　K——密钥；

+——异或。

图 6　国密算法的 TC/AAC/ARQC 的生成算法

5.3.3　国际算法发卡机构认证

生成 8 字节的授权响应密文 ARPC 的方法是将 16 字节的应用密文过程密钥 SK_{AC} 按照附录 A.1.1 指明的对称加密算法对 8 字节长的由卡片按 5.3.2.2 描述的方法生成的 ARQC 和 2 字节的授权响应码 ARC 进行加密：

a）　在 2 字节的 ARC 的后面补上 6 个‘00’字节来获得一个 8 字节的数：X：=（ARC||‘00’||‘00’||‘00’||‘00’||‘00’||‘00’）；

b）　计算 Y：= ARQC ⊕ X；

c）　计算 ARPC：= ALG(SK_{AC})[Y]。

5.3.4　国密算法发卡机构认证

生成 8 字节的授权响应密文 ARPC 的方法是将 16 字节的应用密文过程密钥 SK_{AC} 按照附录 A.1.2 指明的对称加密算法对 5.3.2.3 生成的 8 字节长的 ARQC 和 2 字节的授权响应码 ARC 进行加密，：

a）　在 2 字节的 ARC 的后面补上 6 个‘00’字节来获得一个 8 字节的数：X ：=（ARC||‘00’||‘00’||‘00’||‘00’||‘00’||‘00’）；

b）　计算 Y：= ARQC ⊕ X；

c）　计算 ARPC0：

将 Y 左对齐后面补 8 个字节 00 形成 D；

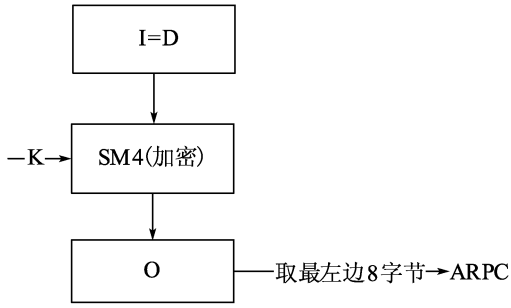
D：= Y||‘00’||‘00’||‘00’||‘00’||‘00’||‘00’||‘00’||‘00’。

基于16字节分组加密算法获得16字节ARPC0：

ARPC0：=SM4(SKAC)[D]；

d） 取ARPC0的左边8字节得到ARPC。

ARPC的生成方法见图7。



说明：

I —— 输入；　　D——数据块；

O——输出；　　K——密钥；

图7 国密算法的生成ARPC的算法

5.4 交通行业信息的保护

5.4.1 密钥说明

5.4.1.1 概述

涉及交通行业信息保护的密钥分为两大类，一类是应用开通密钥，一类是扩展应用管理密钥。

5.4.1.2 应用开通密钥

应用开通密钥用于发卡机构在指定的扩展应用的扩展应用文件中新增应用记录，该记录新增成功后，即意味着该应用的开通。应用开通密钥由发卡机构在个人化时创建，每个扩展应用文件对应一个扩展应用开通密钥，每个扩展应用的开通都由此密钥以安全报文的方式保护，卡片的应用开通密钥由同一个应用开通主密钥分散得到。

5.4.1.3 扩展应用管理密钥

扩展应用管理密钥用于对指定的扩展应用扩展文件中的每一条记录进行保护。

扩展应用管理密钥分为互联互通密钥和地区扩展应用管理密钥两类，互联互通密钥用于保护全国范围内相关的交通行业信息，地区扩展应用管理密钥用于保护本地区的相关交通行业信息。

扩展应用管理密钥在地区使用时应存放于SAM卡中，相关安全机制见本部分第9章。

5.4.1.4 密钥的生成和管理

应用开通密钥可由发卡机构自行生成和管理，互联互通密钥应由密钥管理系统统一生成和管理，地区扩展应用密钥可由发卡机构或相关交通行业方生成和管理。

密钥管理系统生成指定的互联互通主密钥，该密钥同时存放在交通行业相关终端的互联互通SAM卡中。各发卡机构的互联互通密钥由互联互通密钥主密钥分散生成，各发卡机构再利用本发卡机构的互联互通密钥分散得到所发行卡片的互联互通密钥并灌装至用户卡中。

5.4.2 安全机制

终端应使用新增记录(APPEND RECORD)指令在指定扩展应用的扩展应用文件中新增应用记录，即开通新的扩展应用。使用APPEND RECORD命令在扩展应用文件中新增应用记录，使用更新应用数

据缓存(UPDATE CAPP DATA CACHE)命令更新应用文件数据,这两条指令都应强制带有安全报文,以便卡片确认指令来自于合法的终端。

安全报文应以'00'||'00'||'00'||'00'||'00'||'00'||ATC 作为初始向量参与 MAC 运算。MAC 的计算方法见 5.5 中关于报文鉴别码的描述。终端在发送 APPEND RECORD 和 UPDATE CAPP DATA CACHE 指令之前,可以通过发送取数据(GET DATA)指令,或者通过发送获取处理选项(GPO)指令获取 ATC。

在 APPEND RECORD 指令中,应附带有扩展应用管理密钥设置。扩展应用开通后,此扩展应用的应用数据的修改权限,由对应的扩展应用管理密钥以安全报文的方式控制。终端通过 UPDATE CAPP DATA CACHE 指令修改扩展应用数据。扩展应用支持应用失效功能,即交通行业终端在更新应用数据时将应用有效标识置零。

5.5 安全报文

5.5.1 概述

安全报文通过报文鉴别码(MAC)保障数据的完整性和对发卡机构的认证,通过对数据域的加密保障数据的机密性。

5.5.2 报文格式

使用的报文格式见 JT/T 978.3 的定义。报文所涉及的命令的数据域没有将 BER - TLV 编码用于安全报文,使用安全报文的命令的发送者及当前被选择的应用应知道数据域中包含的数据对象以及这些数据对象的长度。根据 GB/T 16649.4,符合此格式的安全报文是通过将命令的类型字节的低半字节设置为'4'明确指定的。当应用为电子钱包模式时,卡中的 FCI 表明某个命令的数据域的数据是否需要加密传输,是否应以加密的方式处理。

5.5.3 使用的密钥

在安全报文计算中用到的过程密钥应按照 5.1.6 中描述的流程来生成。

5.5.4 国际算法报文 MAC 的计算

MAC 计算应满足如下内容:

a) MAC 计算使用 9.1.2 中描述的机制;

b) 要保护的报文应按照支付系统的专有规范来构建,至少包含 C-APDU(CLA INS P1 P2)的头部以及命令数据(若存在);

c) 在本条中 MAC 长度为 4,在按上面描述的方法计算得到 8 个字节的结果后,取其中最左面的(最高)4 字节来得到 MAC。

5.5.5 国密算法报文 MAC 的计算

MAC 计算应满足如下内容:

a) MAC 计算使用 9.2.2 中描述的机制;

b) 要保护的报文应按照支付系统的专有规范来构建,至少包含 C-APDU(CLA INS P1 P2)的头部以及命令数据(若存在);

c) 在本条中 MAC 长度为 4,在按上面描述的方法计算得到 16 个字节的结果后,取其中最左面的(最高)4 字节来得到 MAC。

5.5.6 国际算法报文私密性

对明文/加密命令数据域的加/解密使用9.1.1中描述的机制。

5.5.7 国密算法报文私密性

对明文/加密命令数据域的加/解密使用9.1.1中描述的机制。

6 电子钱包安全体系

6.1 密钥

6.1.1 密钥关系表

卡片中存储的电子钱包的密钥见表20。发卡机构应管理唯一的工作主密钥,卡片子密钥的推导流程见6.1.2。

表20 卡片中存储的电子钱包的密钥

密　钥	发卡机构	卡　片	终端(PSAM)
用于消费交易的密钥	消费主密钥(MPK)	消费子密钥(DPK),由MPK用应用主账号推导获得	消费主密钥(MPK)
用于圈存交易的密钥	圈存主密钥(MLK)	圈存子密钥(DLK),由MLK用应用主账号推导获得	—
用于圈提交易的密钥	圈提主密钥(MULK)	圈提子密钥(DULK),由MULK用应用主账号推导获得	—
消费交易中用于产生TAC的密钥	TAC主密钥(MTK)	TAC子密钥(DTK),由MTK用应用主账号推导获得	—
用于应用维护功能的密钥	应用主控密钥(MAMK)	应用主控子密钥(DAMK),由MAMK用应用主账号推导获得	—
应用解锁密钥	应用解锁主密钥(MUBK)	应用主控解锁子密钥(DUBK)由MUBK用应用主账号推导获得	—
应用锁定密钥	应用锁定主密钥(MBK)	应用主控锁定子密钥(DBK)由MBK用应用主账号推导获得	应用锁定主密钥(MBK)

6.1.2 子密钥推导流程

6.1.2.1 国际算法子密钥推导流程

6.1.2.1.1 左半部分的推导流程

国际算法的双倍长DPK左半部分的推导流程如下:

a) 将应用主账号的最右16个数字作为输入数据;

b) 将MPK作为加密密钥;

c) 国际算法的DPK左半部分推导流程见图8,用MPK对输入数据进行3DES运算。

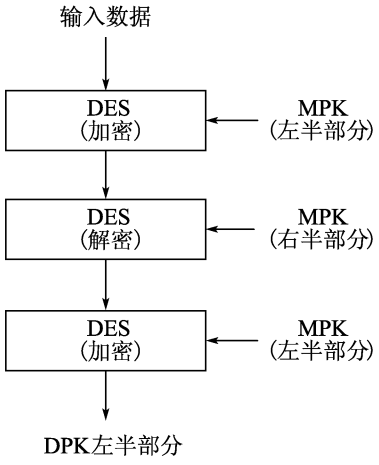


图 8　国际算法的 DPK 左半部分推导流程

6.1.2.1.2　右半部分的推导流程

国际算法的双倍长 DPK 右半部分的推导流程如下：

a）　将应用主账号的最右 16 个数字的求反作为输入数据；

b）　将 MPK 作为加密密钥；

c）　国际算法的 DPK 右半部分推导流程见图 9，用 MPK 对输入数据进行 3DES 运算。

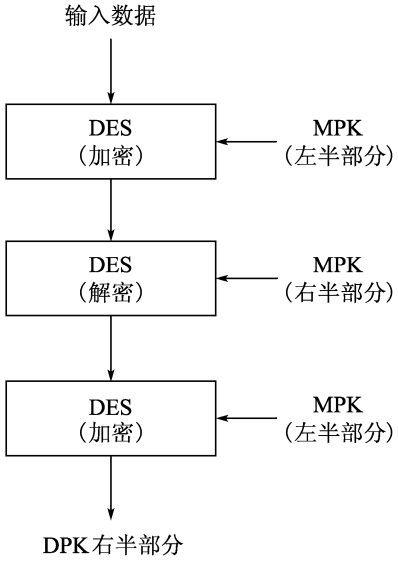


图 9　国际算法的 DPK 右半部分推导流程

6.1.2.2　国密算法子密钥推导流程

国密算法的子密钥推导流程如下：

a）　取 8 字节作为分散因子；

b）　输入数据由 8 字节分散因子加上分散因子取反组成；

c）　国密算法的 DPK 推导流程见图 10，用主密钥对输入数据进行 SM4 加密运算得到 16 字节结果作为子密钥。

6.1.3　过程密钥的产生

6.1.3.1　国际算法过程密钥的产生流程

国际算法的过程密钥是在交易过程中用可变数据产生的单倍长密钥，过程密钥产生后只能在某过

程/交易中使用一次,国密算法的过程密钥产生流程见图 11。

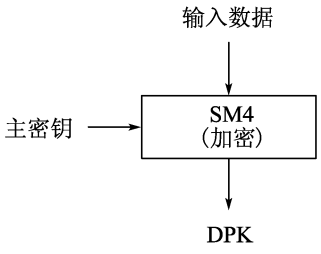


图 10　国密算法的 DPK 推导流程

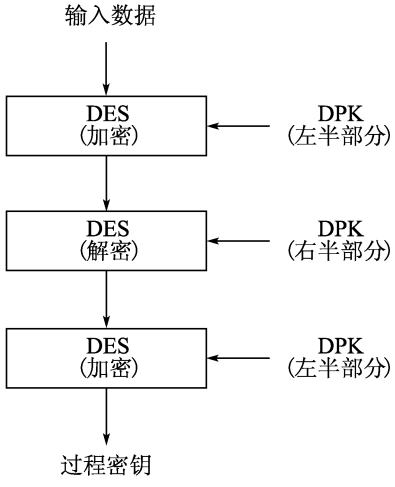


图 11　国际算法的过程密钥产生流程

6.1.3.2　国密算法过程密钥的产生流程

国密算法的过程密钥是在交易过程中用可变数据产生的十六字节长密钥。

过程密钥产生后只能在某过程/交易中使用一次,过程密钥的产生流程如下:

a)　国际算法中定义的 8 字节输入数据内容保持不变,将其在右侧以十六进制“0x 00 00 00 00 00 00 00 00”的方式补齐,形成 16 字节数据作为 SM4 的输入(对原输入数据已经为 16 字节的交易,不需要补齐);

b)　国密算法的过程密钥产生流程见图 12,用 16 字节子密钥,经过一次 SM4 加密产生 16 字节过程密钥。

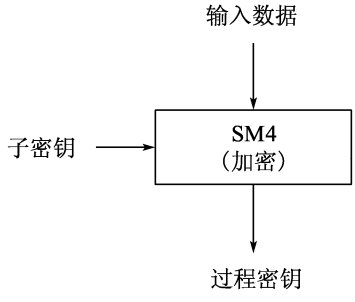


图 12　国密算法的过程密钥产生流程

6.2　交易 MAC/TAC 计算流程

6.2.1　国际算法 MAC/TAC 的计算流程

国际算法产生 MAC/TAC 的单倍长 DES 计算流程如下:

a） 将一个 8 个字节长的初始值设定为十六进制的“0x 00 00 00 00 00 00 00 00”；

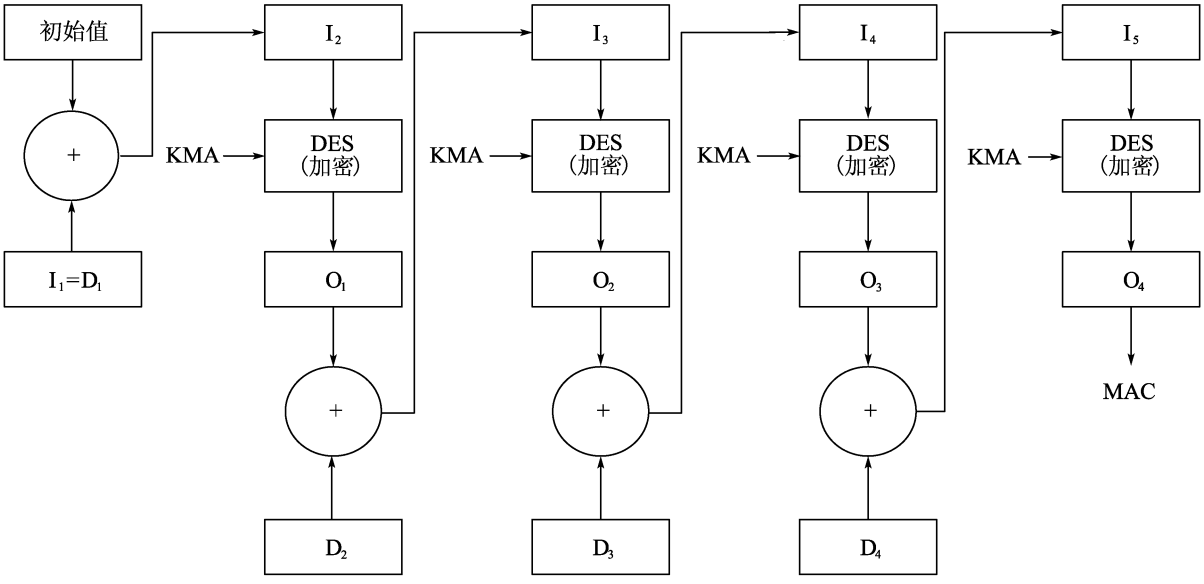
b） 将所有的输入数据按指定顺序串联成一个数据块；

c） 将串联成的数据块分割为 8 字节长的数据块组，标识为 D_1、D_2、D_3 与 D_4 等。分割到最后，余下的字节组成一个长度小于等于 8 字节的最后一块数据块；

d） 若最后一个数据块长度为 8 字节，则在此数据块后附加一个 8 字节长的数据块，附加的数据块为十六进制的“0x 80 00 00 00 00 00 00 00”。若最后一个数据块长度小于 8 字节，则该数据块的最后填补一个十六进制‘80’。若填补之后的数据块长度等于 8 字节，则跳至第五步。若填补之后的数据块长度仍小于 8 字节，则在数据块后填补十六进制‘00’至数据块长度为 8 字节；

e） MAC 的产生是通过上述步骤产生的数据块组，由过程密钥进行加密运算，过程密钥的产生流程见图 11。TAC 的产生是通过上述步骤产生的数据块组，由 DTK 密钥左右 8 位字节进行异或运算的结果进行加密运算。MAC/TAC 的计算流程见图 13；

f） 最终值的左 4 字节为 MAC/TAC。



说明：

I——输入；　　D——数据块；

O——输出；　　KMA——MAC 过程密钥 A；

+——异或运算。

图 13　国际算法的 MAC/TAC 计算流程

6.2.2　国密算法 MAC/TAC 的计算流程

国密算法产生 MAC/TAC 的 SM4 计算流程如下：

a） 将一个 16 个字节长的初始值设定为十六进制的“0x 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00”。；

b） 将所有的输入数据按指定顺序串联成一个数据块；

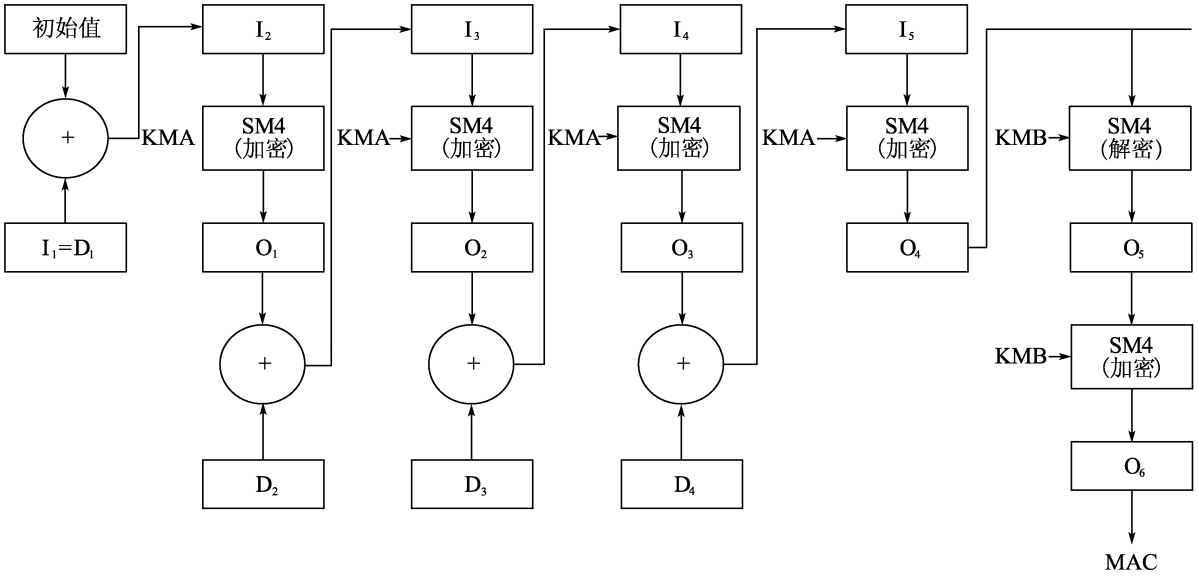
c） 将串联成的数据块分割为 16 字节长的数据块组，标识为 D_1、D_2、D_3 与 D_4 等。分割到最后，余下的字节组成一个长度小于等于 8 字节的最后一块数据块；

d） 若最后一个数据块长度为 16 字节，则在此数据块后附加一个 16 字节长的数据块，附加的数据块为十六进制的“0x 80 00 00 00 00 00 00 00 00 00 00 00 00 00 00”。若最后一个数据

块长度小于16字节,则该数据块的最后填补一个十六进制‘80’。若填补之后的数据块长度等于16字节,则跳至第五步。若填补之后的数据块长度仍小于16字节,则在数据块后填补十六进制‘00’至数据块长度为16字节;

e) MAC的产生是通过上述步骤产生的数据块组,由过程密钥进行加密运算,过程密钥的产生流程见图12。TAC的产生是通过上述方法产生的数据块组,由DTK直接进行加密运算。MAC/TAC的计算流程见图14;

f) 将16字节运算结果按8字节分块做异或运算,得到8字节数据,取高4字节得到认证码MAC/TAC。



说明:

I——输入;　　D——数据块;

KMA——MAC过程密钥A;　　KMB——MAC过程密钥B;

O——输出;　　+——异或运算。

图14　国密算法的MAC/TAC计算流程

6.3　安全报文计算流程

6.3.1　概述

安全报文通过报文鉴别码(MAC)来保障数据的完整性,通过对数据域的加密来保障数据的机密性。

6.3.2　报文格式

使用的报文格式见JT/T 978.3的定义。报文所涉及的命令的数据域没有将BER－TLV编码用于安全报文,使用安全报文的命令的发送者及当前被选择的应用应知道数据域中包含的数据对象以及这些数据对象的长度。根据GB/T 16649.4,符合此格式的安全报文是通过将命令的类型字节的低半字节设置为‘4’明确指定的。当应用基于电子钱包模式时,卡中的FCI表明某个命令的数据域的加密传输方式。

6.3.3　使用的密钥

在安全报文计算中,根据发卡机构需求来决定是否使用过程密钥。如使用过程密钥,应按照6.1.4

中描述的流程来生成。

6.3.4 国际算法报文 MAC 的计算流程

产生 MAC 的单重或三重 DES 计算流程如下：

a） 取 8 个字节的十六进制数字‘0’作为初始变量；

b） 按照顺序将以下数据串联在一起形成数据块：

1） CLA、INS、P1、P2 和 Lc；

2） 在 JT/T 978.2 中定义的数据；

3） 在命令的数据域中(若存在)包含明文或加密的数据。(例：若要更改个人识别码，加密后的个人识别码数据块放在命令数据域中传输)。

c） 将该数据块分成 8 字节为单位的数据块，标号为 D_1、D_2、D_3 和 D_4 等。最后的数据块有可能是 1 个～8 个字节；

d） 根据数据块长度进行补位：

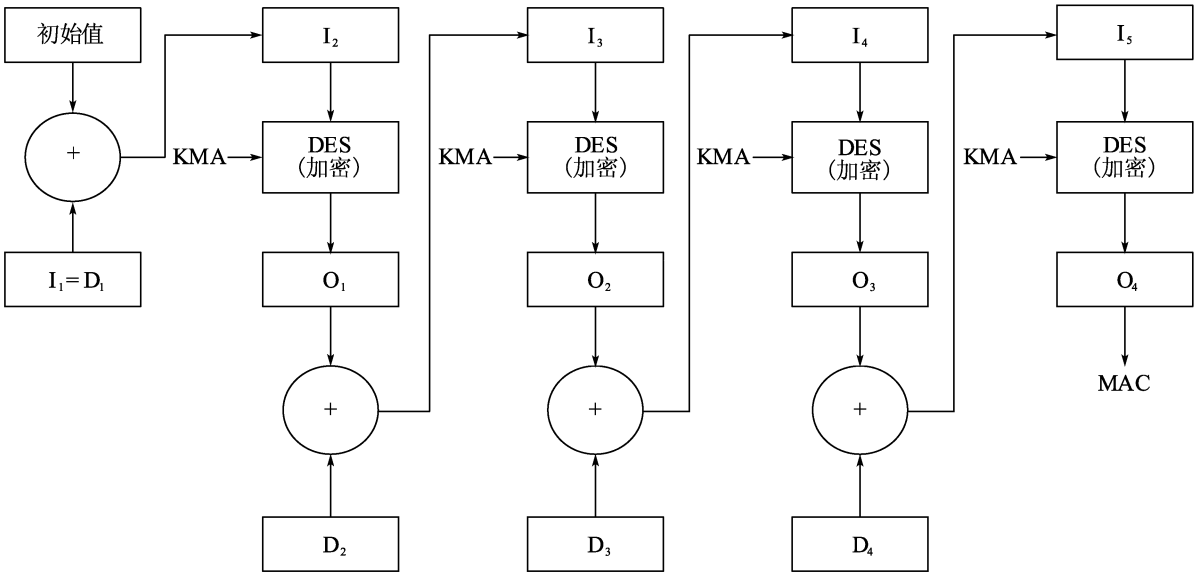
1） 最后的数据块长度是 8 字节的话，则在其后加上十六进制数字“0x 80 00 00 00 00 00 00 00”，转到步骤 e)；

2） 最后的数据块长度不足 8 字节，则在其后加上十六进制数字‘80’，若达到 8 字节长度，则转入步骤 e)；否则在其后加入十六进制数字‘00’直到长度达到 8 字节。

e） 对这些数据块使用 MAC 密钥进行加密，若安全报文传送支持单长度密钥，则按照图 15 的方式来产生 MAC[根据在步骤 b)中产生的数据块长度的不同，确定计算的步数]；

f） 最终得到从计算结果左侧取得的 4 字节长度的 MAC；

安全报文传送的处理支持双长度密钥，按照图 16 的方式来产生 MAC，[根据步骤 b)产生的数据块的长度，确定计算的步数]。



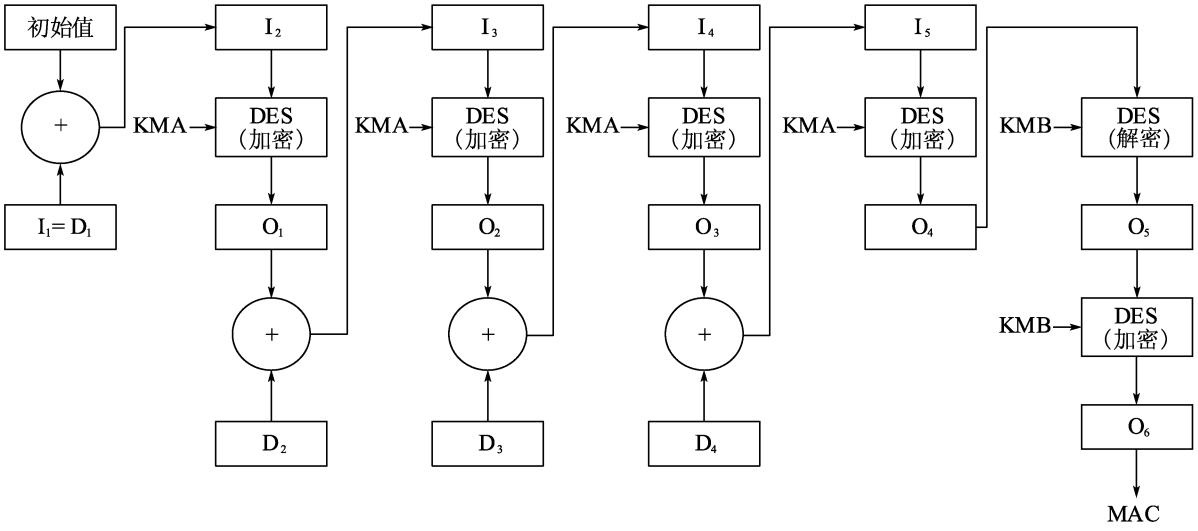
说明：

I——输入；　　D——数据块；

O——输出；　　KMA——MAC 密钥 A；

+——异或运算。

图 15　国际算法的单长度密钥 MAC 计算流程



说明：

I——输入；　　D——数据块；

KMA——MAC 密钥 A；　　KMB——MAC 密钥 B；

O——输出；　　+——异或运算。

图 16　国际算法的双长度密钥 MAC 计算流程

6.3.5　国密算法报文 MAC 的计算流程

产生 MAC 的使用 SM4 加密计算流程如下：

a）　取 16 个字节的十六进制数字‘0’作为初始变量；

b）　按照顺序将以下数据串联在一起形成数据块：

1）　CLA、INS、P1、P2 和 Lc；

2）　在 JT/T 978.2 中定义的数据；

3）　在命令的数据域中(若存在)包含明文或加密的数据(例：若要更改个人识别码，加密后的个人识别码数据块放在命令数据域中传输)。

c）　将该数据块分成 16 字节为单位的数据块，标号为 D_1、D_2、D_3 和 D_4 等。最后的数据块有可能是 1 个～16 个字节；

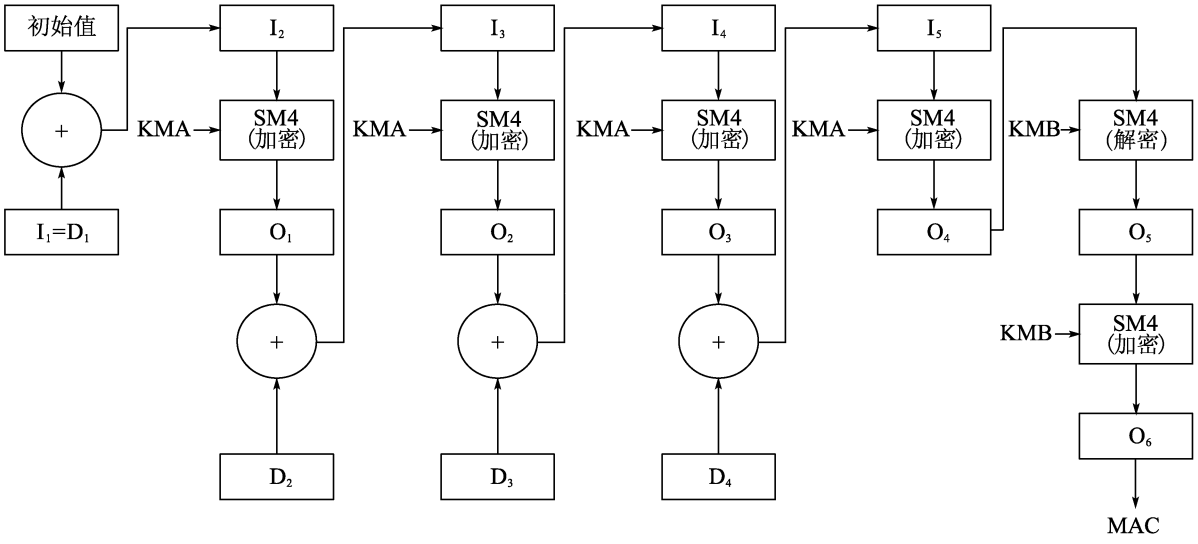
d）　根据数据块进行补位：

1）　最后的数据块长度是 16 字节的话，则在其后加上十六进制数字“0x 80 00 00 00 00 00 00 00 00 00 00 00 00 00 00”，转到步骤 e)；

2）　最后的数据块长度不足 16 字节，则在其后加上十六进制数字‘80’，若达到 16 字节长度，则转入步骤 e)；否则在其后加入十六进制数字‘0’直到长度达到 16 字节。

e）　对这些数据块使用 MAC 密钥进行加密，MAC 的产生流程见图 17；

f）　最终将 16 字节计算结果左右 8 字节异或，得到 8 字节数据，再取高 4 字节作为 MAC。



说明：

I——输入；　　D——数据块；

KMA——MAC 密钥 A；　KMB——MAC 密钥 B；

O——输出；　　+——异或运算。

图 17　国密算法的 MAC 计算流程

6.3.6　国际算法报文私密性

6.3.6.1　数据加密计算流程

数据加密计算流程如下：

a）　用 LD 表示明文数据的长度，在明文数据前加上 LD 产生新的数据块；

b）　将流程 a）中生成的数据块分解成 8 字节数据块，标号为 D_1、D_2、D_3 和 D_4 等。最后一个数据块长度有可能不足 8 字节；

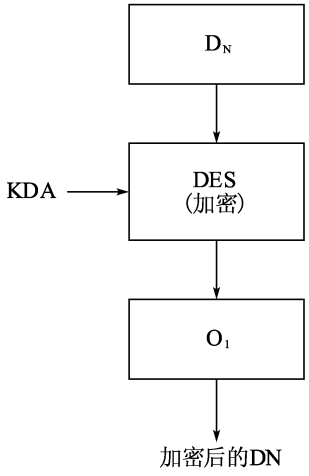
c）　最后（或唯一）的数据块长度等于 8 字节，转入流程 d）；若不足 8 字节，在右边添加十六进制数字‘80’。若长度已达 8 字节，转入步骤 d）；否则，在其右边添加 1 字节十六进制数字’0’直到长度达到 8 字节；

d）　每一个数据块使用 9.1.1 中描述的数据加密方式加密：

1）　采用单长度数据加密密钥，数据块的加密流程见图 18（使用数据加密密钥 A 进行加密）；

2）　采用双长度数据加密密钥，数据块的加密流程见图 19（使用数据加密密钥 A 和 B 来进行加密）；

e）　计算结束后，所有加密后的数据块依照原顺序连接在一起（加密后的 D_1、加密后的 D_2 等）。并将结果数据块插入到命令数据域中。

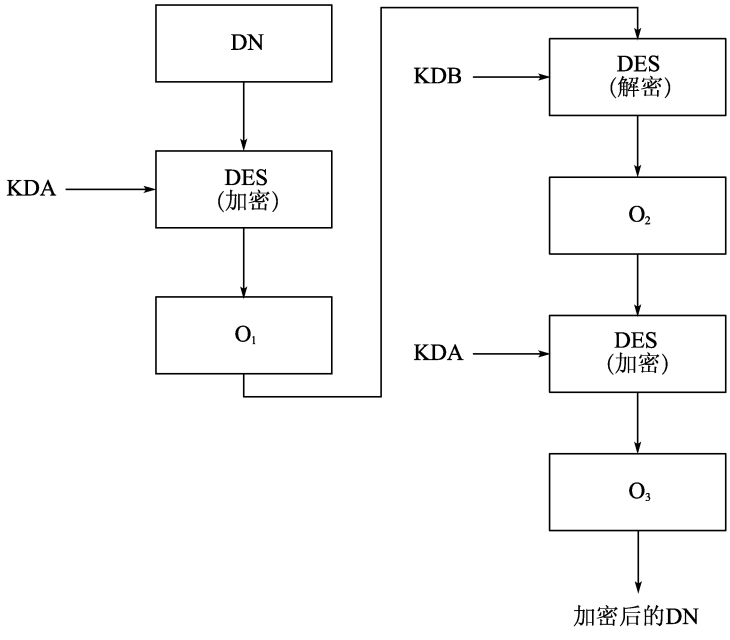


说明：

D——数据块；　　KDA——数据加密过程；

O——输出。

图 18　国际算法的单长度密钥的数据加密流程



说明：

I——输入；　　D——数据块；

O——输出；　　KDA——数据加密密钥 A；

KDB——数据加密密钥 B。

图 19　国际算法的双长度密钥的数据加密流程

6.3.6.2　数据解密计算流程

卡片接收到命令之后，应将包含在命令中的加密数据进行解密。数据解密计算流程如下：

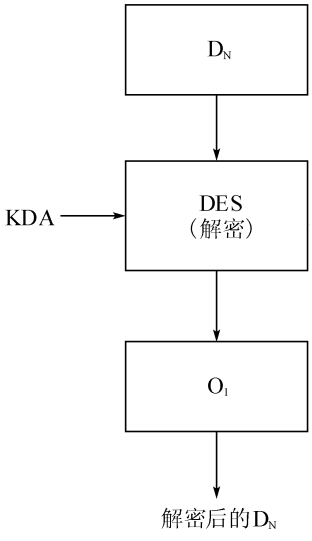
a）　将命令数据域中的数据块分解成 8 字节长的数据块，标号为 D_1、D_2、D_3 和 D_4 等；

1）　采用单长度数据加密密钥，数据块解密流程见图 20（使用数据加密密钥 A 进行解密）；

2）　采用双长度数据加密密钥，数据块的解密流程见图 21（使用数据加密密钥 A 和 B 来进行解密）。

b） 计算结束后，所有解密后的数据块依照顺序（解密后的 D_1、解密后的 D_2 等）链接在一起。数据块由 LD、明文数据、填充字符（若在 6.3.6.1 描述的加密过程中增加的话）组成；

c） LD 表示明文数据的长度，用来恢复明文数据。

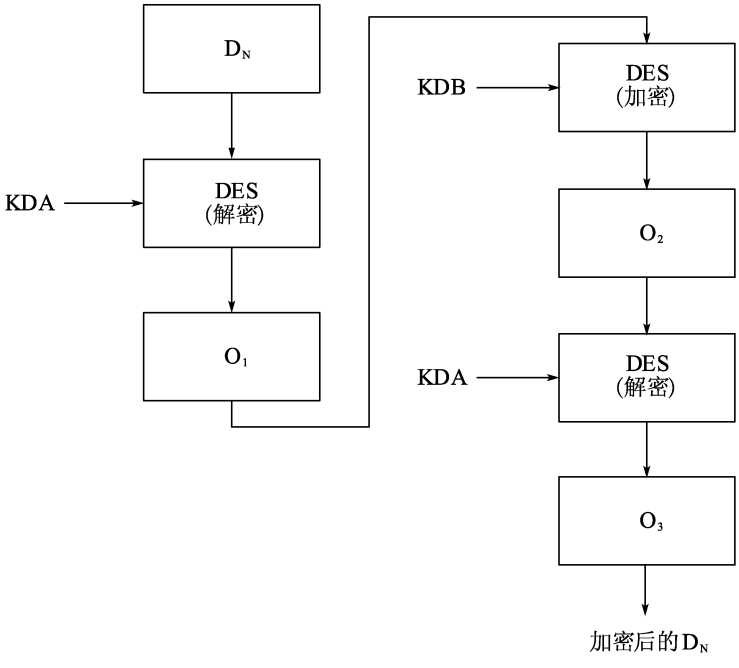


说明：

D——数据块；　　KDA——数据加密密钥 A；

O——输出。

图 20　国际算法的单长度密钥的数据解密流程



说明：

I——输入；　　D——数据块；

O——输出；　　KDA——数据加密密钥 A；

KDB——数据加密密钥 B。

图 21　国际算法的双长度密钥的数据解密流程

6.3.7 国密算法报文私密性

6.3.7.1 数据加密计算流程

数据加密计算流程如下：

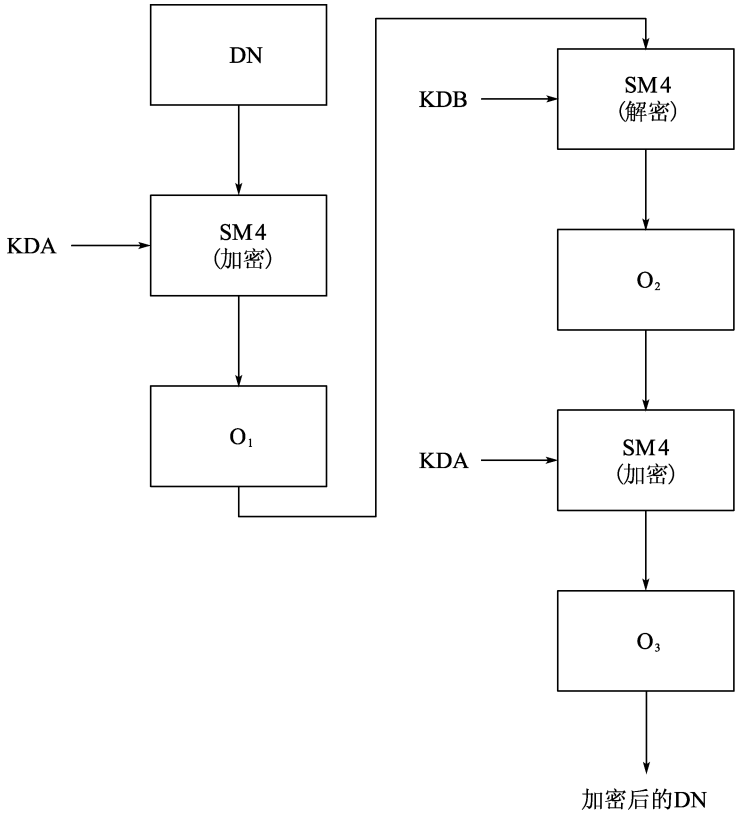
a) 用 LD 表示明文数据的长度，在明文数据前加上 LD 产生新的数据块；

b) 将步骤 a) 中生成的数据块分解成 16 字节数据块，标号为 D_1、D_2、D_3 和 D_4 等。最后一个数据块长度有可能不足 16 字节；

c) 最后（或唯一）的数据块长度等于 16 字节，转入步骤 d)；若不足 16 字节，在右边添加十六进制数字‘80’。若长度已达 16 字节，转入步骤 d)；否则，在其右边添加 1 字节十六进制数字‘0’直到长度达到 16 字节；

d) 每一个数据块使用 9.2.1 中描述的数据加密方式加密，数据块的加密流程见图 22；

e) 计算结束后，依照原顺序连接所有加密后的数据块（加密后的 D_1、加密后的 D_2 等）。并将结果数据块插入到命令数据域中。



说明：

I——输入；　　D——数据块；

O——输出；　　KDA——数据加密密钥 A；

KDB——数据加密密钥 B。

图 22　国密算法的数据加密流程

6.3.7.2 数据解密计算

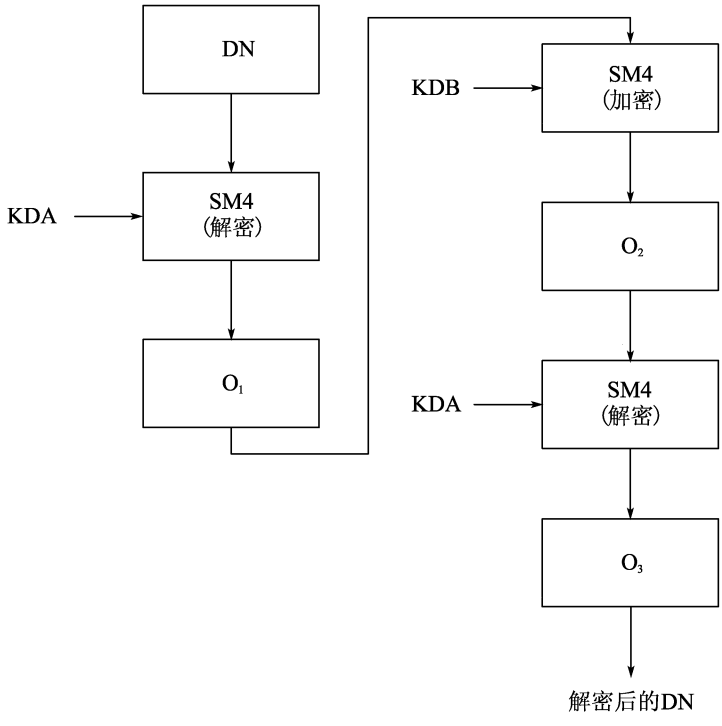
卡片接收到命令之后，应将包含在命令中的加密数据进行解密。数据解密计算流程如下：

a) 将命令数据域中的数据块分解成 16 字节长的数据块，标号为 D_1、D_2、D_3 和 D_4 等。数据块的解密流程见图 23；

b) 计算结束后，依照原顺序连接所有解密后的数据块（解密后的 D_1、解密后的 D_2 等）。数据块由

LD、明文数据、填充字符(若在6.3.7.1描述的加密过程中增加的话)组成;

c) LD表示明文数据的长度,用来恢复明文数据。



说明:

I——输入;　　D——数据块;

O——输出;　　KDA——数据加密密钥A;

KDB——数据加密密钥B。

图23　国密算法的数据解密流程

7　卡片安全

7.1　共存应用

不同应用间应设计一道“防火墙”以防止跨过应用进行非法访问。包括所有不在本规范定义中的应用和其他的恶意应用。

不同应用不应与卡中共存的个人化要求和应用规则发生冲突。

7.2　密钥的独立性

特定功能(如:AC密钥)的加密/解密密钥不能被任何其他功能所使用,包括保存在卡片中的密钥和用来产生、派生、传输这些密钥的密钥。

7.3　卡片内部安全体系

7.3.1　卡片内部安全目标

为了保证卡片操作系统使用合适的安全机制,并在卡片内部为所有数据及处理过程提供安全性和

完整性保障;为访问数据文件和使用的命令与加密算法而设计的。

7.3.2 卡片内部安全概述

7.3.2.1 一般要求

这一安全体系的基础结构包括两个基本特性:

a) “安全域”的建立;

b) 对每个 EF 的存取采用指定的访问条件。

7.3.2.2 安全域

卡片内部安全域应满足如下要求:

a) 操作系统控制对所有数据和可执行资源(即数据文件、记录、命令和加密密钥与算法)的访问,应通过执行选择(SELECT)和获取处理选项(GPO)命令实现安全域。这些命令建立描述安全域的相关信息,并定义数据和可执行资源的范围;

b) 由于卡片操作系统是在文件层次上使用这些信息和实现对数据的访问控制,发卡机构不应将同一层次下访问条件不同的数据并到同一个文件中;

c) 应使用选择(SELECT)命令访问应用管理数据(AMD),AMD 指定能够被后续指令访问的所有数据文件,记录以及可执行资源;

d) 应用管理数据决定可访问的文件和可执行资源,文件和记录编号应在获取处理选项(GPO)命令响应内的应用文件定位器(AFL)中提供;

e) 发卡机构应限制在交易期间被存取的资源,应用管理数据的初始化状态(在个人化阶段被定义)应仅包含交易过程中可被访问的数据文件;

f) 初始的应用管理数据应在选择应用时建立,并且在个人化时被定义。

7.3.2.3 基本文件(EF)访问条件

对于基本文件的访问,前提是至少执行一次选择(SELECT)命令并且安全域已经建立。一旦安全域建立,并且后续读取(如读记录命令)或者更新数据(如修改记录命令)命令被发送到一个基本文件的时候,基本文件的访问控制(由文件控制信息的文件控制参数定义)被强制使用,见 7.3.3。使用安全通信或 VERIFY 命令(或者包含二者)作为访问条件的文件只有在这些条件都满足以后被请求的访问才能继续执行。基本文件的访问条件应用于所有命令,以提供对卡片数据的外部访问,如读记录(READ RECORD)、取数据(GET DATA)、设置数据(PUT DATA)、修改记录(UPDATE RECORD)等命令。

7.3.3 文件控制信息

7.3.3.1 概述

文件控制信息(FCI)附属于每个 ADF 或 AEF,描述了文件的特性。文件控制信息在个人化期间为每个文件建立,定义了初始的安全域。

7.3.3.2 应用管理数据

7.3.3.2.1 安全域的定义

应用管理数据描述的安全域定义以下内容:

a) 在应用范围内可以被存取的资源,应用基本文件和内部基本文件(如个人识别码 PIN、密钥、参数);

b) 可在应用的上下文范围内被执行的命令;

c) 命令与资源之间的关系。

7.3.3.2.2 安全域定义的资源类型

安全域由应用管理数据说明的相关资源定义。没有被包含在应用管理数据内的资源不能被应用所使用。对应用来说安全域是相互独立的。共有以下两类资源被定义:

a) 数据资源(见7.3.3.3.);
b) 可执行代码资源。

7.3.3.3 数据资源

7.3.3.3.1 一般要求

数据资源可以是以下列出的任意一个:

a) 数据文件及其记录;
b) 密钥;
c) PIN。

7.3.3.3.2 数据标识

数据资源是指文件内的数据元,由卡片内部的唯一标识符所识别。文件由卡片内部唯一的文件标识符所标识。不包含在文件内的数据元则由一个唯一数据标识所标识。运行应用所需的任何数据资源应在应用管理数据内标识,要求如下:

a) 对包含了数据元(可由应用管理数据定义的命令访问)的文件而言,SFI(在应用内被唯一标识,并且可从外部被引用)与文件标识(在卡片内被唯一标识,并且可从内部被引用)之间的关系被维护在应用管理数据内;
b) 对未被包含在文件内的数据对象[可由应用管理数据定义的命令如取数据(GET DATA)命令访问]而言,数据对象标签(可从外部被引用)与唯一数据标识(在卡片内部,并且可从内部被引用)之间的关系被维护在应用管理数据内。

7.3.3.3.3 密钥标识

密钥不应从外部被引用。对保存在文件内的密钥,应用管理数据维护了在执行应用管理数据定义的命令和加密算法时定位密钥所必需的文件标识和指向密钥的引用;对不保存在文件内的密钥,应用管理数据维护了在执行应用管理数据定义的命令和加密算法时定位密钥所必需的卡片内部的唯一密钥标识。

7.3.3.3.4 PIN/口令标识

PIN/口令只能从外部通过应用管理数据和安全通信共同定义的命令被引用。对保存在文件内的PIN或者口令而言,应用管理数据维护了在执行应用管理数据定义的命令和加密算法时定位PIN/口令所必需的文件标识和指向PIN/口令的引用;对不保存在文件内的PIN/口令而言,应用管理数据维护了在执行应用管理数据定义的命令和加密算法时定位PIN/口令所必需的卡片内部的唯一PIN/口令标识。

7.3.3.3.5 可执行代码资源

可执行代码资源包括:

a) 命令;
b) 加密算法。

7.3.3.3.6 命令标识

命令资源包括CLA和INS字节,操作系统用他们来查找命令的位置。命令资源项包括了命令访问的数据的属性,有时还有与密钥和算法相关的参数属性。

7.3.3.3.7 算法标识

算法资源建立了为应用而定义的算法标识,与操作系统用来定位可执行代码的实际算法引用之间的联系。

7.3.4 文件控制参数

每个基本文件在其文件控制信息中包含一个文件控制参数(FCP),它保存了同文件的访问条件相关的附加信息。该信息在个人化期间被放在卡片内,并且同保存在ADF的文件控制信息内的应用管理数据一起,由卡片操作系统用于建立应用的安全域。基本文件的访问条件见表21。

表 21　基本文件的访问条件

读　取	更　新	访 问 条 件
是/否	是/否	
是/否	是/否	安全通信
是/否	是/否	校验
(不可用)	是/否	数据加密

注 1:读取一栏表示使用读取命令,如读记录(READ RECORD)或取数据(GET DATA)命令,存取基本文件内部的数据。“更新”一栏表示使用更新命令,如修改记录(UPDATE RECORD)或设置数据(PUT DATA)命令,存取基本文件内部的数据。

注 2:文件控制参数指出是否在发卡机构脚本修改记录(UPDATE RECORD)命令中以加密或者明文格式传送数据。

注 3:文件控制参数也作为一个组件用于实现应用管理数据的逻辑结构。

7.3.5　卡片本地数据的访问条件

数据访问条件适用于可被读记录(READ RECORD)、修改记录(UPDATE RECORD)、取数据(GET DATA)命令或其他合适的类似命令访问的数据,具体如下:

a) 可被设置数据(PUT DATA)命令与安全通信改变的数据,以及可被取数据(GET DATA)命令读取的数据:
   1) 连续脱机交易下限(“9F58”);
   2) 连续脱机交易上限(“9F59”);
   3) 连续脱机交易限制数(国际—国家);
   4) 连续脱机交易限制数(国际);
   5) 累计交易总额限制;
   6) 累计交易总额限制(两种货币);
   7) 累计交易总额上限;
   8) 货币转换因子。
b) 可被应用私有的 PIN 修改/解锁(PIN CHANGE/UNBLOCK)命令与安全通信所更新的数据,以及不可被读取的数据:PIN;
c) 可被取数据(GET DATA)命令读取的数据,以及可被 PIN 修改/解锁(PIN CHANGE/UNBLOCK)命令与安全通信重新设置为预定限制的数据:PIN 尝试计数器。

8　终端安全

8.1　终端数据安全性要求

8.1.1　一般要求

终端存在两种类型的数据:

a) 通用数据:包括时间、终端识别号、终端交易记录等。外界可对这些数据进行访问,但不允许进行无授权修改;
b) 敏感数据:包括根公钥、用于 PIN 加密的对称密钥及终端内部的参数。在未授权的情况下,外界不允许对这类数据进行访问和修改。

8.1.1.1　通用数据的安全要求

通用数据可存放在存储器中。在更新参数以及下载新的应用程序时,终端应做到:

a) 验证更新方的身份,对于应用程序重新下载,只允许终端制造厂商、终端所有者或者经终端所有者或代理方批准的第三方执行;

b) 校验下载参数及应用程序的完整性。

对存储器要求应做到:无论在什么情况下,终端的应用数据都不应随意改变或丢失,并保证数据有效。所有与交易相关的数据均应以记录形式存储于终端存储器中,终端应保证这些数据的完整性。

8.1.1.2 敏感数据的安全要求

敏感数据应存放在终端安全模块中。

8.1.2 安全模块的物理安全要求

安全模块的硬件设计应能保证在物理上限制对其内部存储的敏感数据的存取与窃取,以及对安全模块的非授权使用和修改。一旦安全模块受到非法的攻击,其自身应立即完成对内部敏感数据的删除。同时,安全模块也应具有足够的安全特性,防止数据被非法篡改。安全模块的任何部分的损坏或失效都不能导致敏感数据的泄露。若安全模块是由多个分离部件组合而成,并且处理的数据在这些部件之间传递,则各部件须保持相同的安全级别。

8.1.3 安全模块的逻辑安全要求

安全模块的逻辑设计应保证,调用任何单一功能或组合功能,都不应导致敏感数据的泄露。对于某些敏感操作应具有一定的权限限制。

安全模块中可存放多组根公钥及其相关信息。若在终端使用过程中,需要更新或撤回根公钥,应使用安全报文方式;当需要以安全报文方式传递信息时,安全模块应能够实现安全报文传递。

8.2 终端设备安全性要求

8.2.1 防入侵设备

8.2.1.1 概述

防入侵的设备应保证在正常运行环境中,不应泄露或改变任何在设备中存储、处理或传输的敏感数据。

8.2.1.2 物理安全性

防入侵的设备应限制对内部敏感数据的物理访问,阻止窃取数据或未经授权的使用。设备应被设计和构造为:

a) 不允许轻易入侵设备并对设备的软硬件进行增加、替换或修改;

b) 设备的包装不能采用普通的易复制的材料;

c) 当设备的任何部件发生任何故障时,不应导致敏感的数据的泄漏;

d) 若设备的设计需要部分部件在物理上分离,并且处理的数据或持卡人的指令在这些分离的部件之间传递,则对设备的所有部件的保护等级应是相同的;

e) 对交换敏感数据如明文 PIN 来说,将不同的部件整合在单一的防入侵的外壳中是必要的条件。

8.2.1.3 逻辑安全性

终端逻辑安全性的要求如下:

a) 防入侵的设备应防止非法函数导致敏感数据的泄漏,即使在使用合法的函数的情况下,也应有逻辑保护敏感数据的安全;

b) 若终端被置于一种“敏感状态”,即允许通常情况下不被允许的函数的状态(如人工安装密

钥),该转换应在两个或两个以上可信赖的人员的协助下进行。若用密码或其他明文数据来控制转换过程,则这些密码的输入也要用和其他敏感数据一样的方式来保护;

c) 为了将由未经授权的对敏感函数的使用所导致的风险降到最小,对敏感状态应有调用函数次数(适当的)的限制和时间限制。一旦达到了这些限制,设备应返回正常状态;

d) 在交易结束或超时后,防入侵的设备应自动清除内部的缓存。

8.3 终端密钥管理要求

8.3.1 终端密钥种类

在终端中存在的密钥种类见表22。

表22 终端内部保存的密钥种类

密钥名称	用途	密钥形式	条件
根公钥	用于动态数据认证	非对称密钥	M
根公钥维护密钥	用于导入,更新和撤回根公钥	对称密钥	M
PIN加密密钥	用于保护PINPAD到终端的用户PIN	对称密钥	O
电子现金互联互通密钥	用于保护交通行业信息	对称密钥	O-储存在扩展应用SAM卡中
电子钱包消费密钥	用于电子钱包的交易密文信息计算	对称密钥	M-储存在钱包应用SAM卡中
注:M-必备,O-可选。			

8.3.2 电子现金根公钥管理

8.3.2.1 概述

对收单机构管理终端中的根公钥的要求包括以下阶段:

a) 将根公钥导入终端;

b) 根公钥在终端中的存储;

c) 根公钥在终端中的使用;

d) 从终端中撤回根公钥。

8.3.2.2 根公钥导入

当密钥管理系统决定导入一个新的根公钥时,应保证将新的公钥从密钥管理系统分发给每一个收单机构。

8.3.2.3 根公钥储存和更新

根公钥储存和更新应满足如下要求:

a) 具备动态数据认证的终端应对每个卡片应用的RID提供6个根公钥的支持;

b) 每一个根公钥由5个字节的RID和1个字节的根公钥索引唯一标识。根公钥索引对于每个RID是唯一的,由密钥管理系统进行分配。对于每一个根公钥,表23描述在终端中有用的数据元的最小集;

c) 根公钥算法标识指明了与相应的根公钥一起使用的数字签名算法,哈希算法标识指定了在数字签名方案中用来生成哈希结果的哈希算法。算法选择与切换的流程见附录B。根公钥储存于终端的安全模块中,可以任意读取,但更新应使用安全报文。根公钥校验和用来保证根公钥及其相关数据准确无误接收到。

表 23　存储在终端中的根公钥相关数据元的最小集

名　称	长　度	说　明	格式
注册的应用提供商标识(RID)	5	指定根公钥和哪个密钥管理系统的关联	b
根公钥索引	1	和 RID 一起指定根公钥	b
密钥管理系统哈希算法标识	1	标识用于在数字签名方案中产生哈希结果的哈希算法	b
根公钥算法标识	1	标识使用在根公钥上的数字签名算法	b
根公钥模	变长最大为 248	根公钥模部分的值	b
根公钥指数	1 或 3	根公钥指数部分的值,等于 3 或 $2^{16}+1$	b
根公钥校验值	20	使用附录 A.3 指定的哈希算法对根公钥所有部分(RID、根公钥索引、根公钥模、根公钥指数)的连接计算得到的校验值	b

8.3.2.4　根公钥回收

当决定撤回某一个根公钥时,收单机构应保证在一个确定的时间后它的终端在交易中不再将这个根公钥用于静态和动态数据认证。

以下的原则适用于收单机构将根公钥从它的终端撤回:

a)　终端应能够验证它从收单机构收到的撤回通告没有错误;

b)　终端应能够验证收到的撤回通告确实是来自于它的合法收单机构;

c)　收单机构应能够确认一个特定的根公钥已经正确地从它的终端撤回。

8.3.3　SAM 卡

SAM 卡应由密钥管理系统统一发行和管理,不参与互联互通的地区扩展应用 SAM 卡应由各地区发卡机构发行和管理。

SAM 卡中存放电子现金互联互通主密钥以及电子钱包消费主密钥。

9　加密机制

9.1　国际算法对称加密机制

9.1.1　加密解密

9.1.1.1　填充并分块

填充及分块的流程如下:

a)　填充数据:

1)　若报文 MSG 的长度不是分组长度的整数倍,在 MSG 的右端加上 1 个‘80’字节。在右端加上最少的‘00’字节,使得结果报文的长度 MSG: = (MSG||‘80’||‘00’||‘00’||…||‘00’)是分组长度的整数倍;

2)　若报文 MSG 的长度是分组长度的整数倍,不对数据作填充。

b)　被加密数据首先应被格式化为以下形式的数据块:

1)　明文数据的长度,不包括填充字符;

2)　明文数据;

3)　填充字符(按上述填充方式)。

c) 将 MSG 拆分为 8 字节或 16 字节的块 $X_1, X_2, \cdots, X_K$。

9.1.1.2 密文计算

用加密过程密钥 K_S 以 ECB 模式的分组加密算法将块 $X_1, X_2, \cdots, X_K$ 加密为分组长度的块 $Y_1, Y_2, \cdots, Y_K$。

当 $i=1,2,\cdots,K$ 时,分别计算:

$Y_i := ALG(K_S)[X_i]$。

9.1.1.3 密文解密

解密流程如下:

当 $i=1,2,\cdots,K$ 时,分别计算:

$X_i := ALG^{-1}(K_S)[Y_i]$。

处理填充字节:

为了得到原来的报文 MSG,将块 $X_1, X_2, \ldots, X_K$ 连接起来,若使用了填充(见上文),从最后一块 X_K 中删除('80'||'00'||'00'||...||'00')字节串的结尾。

记为:

$MSG = DEC(K_S)[Y]$。

9.1.2 报文鉴别码

9.1.2.1 基于 64 位分组加密算法的 MAC 计算流程

计算一个 s 字节的 MAC($4 \leqslant s \leqslant 8$)是依照 GB/T 27929 规范,采用 CBC 模式的 64 位分组加密算法。更准确地说,用 MAC 过程密钥 K_S 对任意长度的报文 MSG 计算 MAC 值 S 的流程如下:

a) 依据 GB/T 16649.4 对报文 MSG 进行填充并分块:

1) 在 MSG 的右端强制加上 1 个'80'字节;

2) 在右端加上最少的'00'字节,使得结果报文的长度 MSG: = (MSG||'80'||'00'||'00'||…||'00')是 8 字节的整数倍;

3) 将 MSG 拆分为 8 字节的块 $X_1, X_2, \cdots, X_K$。

b) MAC 过程密钥:K_S 可使用最左端密钥块 $K_S = K_{SL}$,也可以由最左端密钥块和最右端密钥块连接而成 KS = (KSL||KSR);

c) 密文计算:

用 MAC 过程密钥的最左端块 K_{SL},以 CBC 模式的分组加密处理 8 字节块 $X_1, X_2, \ldots, X_K$:

$H_i := ALG(K_{SL})[X_i \oplus H_{i-1}]$,这里 $i=1,2,\ldots,K$。

H_0 的初始值 H_0: = ('00'||'00'||'00'||'00'||'00'||'00'||'00'||'00')。

依照 GB/T 27929 算法 3 计算 8 字节的块 H_{K+1}:

$H_{K+1} := ALG(K_{SL})[ALG^{-1}(K_{SR})[H_K]]$。

MAC 值 S 等于 H_{K+1}的 s 个最高位字节。

9.1.2.2 基于 128 位分组加密算法的 MAC 计算方法

采用 CBC 模式的 128 位分组加密算法以及 MAC 过程密钥 K_S 对任意长度的报文 MSG 计算一个 s 字节的 MAC($4 \leqslant s \leqslant 8$)值 S 的步骤如下:

9.1.2.2.1 填充并分块

依据 GB/T 16649.4 对报文 MSG 进行填充:

a) 在 MSG 的右端强制加上 1 个'80'字节;

b) 在右端加上最少的'00'字节,使得结果报文的长度 MSG: = (MSG||'80'||'00'||'00'||…||'00')是 16 字节的整数倍;

c) 将 MSG 拆分为 16 字节的块 $X_1, X_2, \cdots, X_K$。

9.1.2.2.2 MAC 过程密钥

MAC 过程密钥 K_S 长度为 16 字节。

9.1.2.2.3 密文计算

用 MAC 过程密钥以 CBC 模式的分组加密处理 16 字节块 $X_1, X_2, \ldots, X_K$：

$H_i := ALG(K)[X_i \oplus H_{i-1}]$，这里 $i = 1, 2, \ldots, K$。

H_0 的初始值 $H_0 :=$ ('00'||'00'||'00'||'00'||'00'||'00'||'00'||'00'||'00'||'00'||'00'||'00'||'00'||'00'||'00'||'00')。

用以下方法计算 8 字节的块 H_{K+1}。

$H_{K+1} := H_{KL} \oplus H_{KR}$。

MAC 值 S 等于 H_{K+1} 的 s 个最高位字节。

9.2 国密算法对称加密机制

9.2.1 加密解密

9.2.1.1 填充并分块

填充并分块的流程如下：

a) 填充数据：

1) 若报文 MSG 的长度不是分组长度的整数倍，在 MSG 的右端加上 1 个'80'字节，再在右端加上最少的'00'字节，使得结果报文的长度 MSG := (MSG||'80'||'00'||'00'||…||'00')是分组长度的整数倍；

2) 若报文 MSG 的长度是分组长度的整数倍，不对数据作填充。

b) 被加密数据首先要被格式化为以下形式的数据块：

1) 明文数据的长度，不包括填充字符；

2) 明文数据；

3) 填充字符(按上述填充方式)。

c) 将 MSG 拆分为 16 字节的块 $X_1, X_2, \cdots, X_K$。

9.2.1.2 密文计算

用加密过程密钥 K_S，以 ECB 模式的分组加密算法将块 $X_1, X_2, \cdots, X_K$ 加密为 16 字节的块 $Y_1, Y_2, \cdots, Y_K$。

当 $i = 1, 2, \cdots, K$ 时，分别计算：

$Y_i := ALG(K_S)[X_i]$。

9.2.1.3 解密过程

解密过程如下：

当 $i = 1, 2, \cdots, K$ 时，分别计算：

$X_i := ALG-1(K_S)[Y_i]$。

处理填充字节：

为得到原来的报文 MSG，将块 $X_1, X_2, \cdots, X_K$ 连接起来，若使用了填充(见上文)，从最后一块 X_K 中删除尾部的('80'||'00'||'00'||...||'00')。

记为：

$MSG = DEC(K_S)[Y]$。

9.2.2 报文鉴别码

9.2.2.1 填充并分块

依据 GB/T 16649.4 对报文 MSG 进行填充：

a） 在 MSG 的右端加上 1 个‘80’字节；

b） 在右端加上最少的‘00’字节，使得结果报文的长度 MSG：=（MSG||‘80’||‘00’||‘00’||…||‘00’）是 16 字节的整数倍；

c） 将 MSG 拆分为 16 字节的块 $X_1, X_2, \cdots, X_K$。

9.2.2.2 MAC 过程密钥

MAC 过程密钥 K_S 长度为 16 字节。

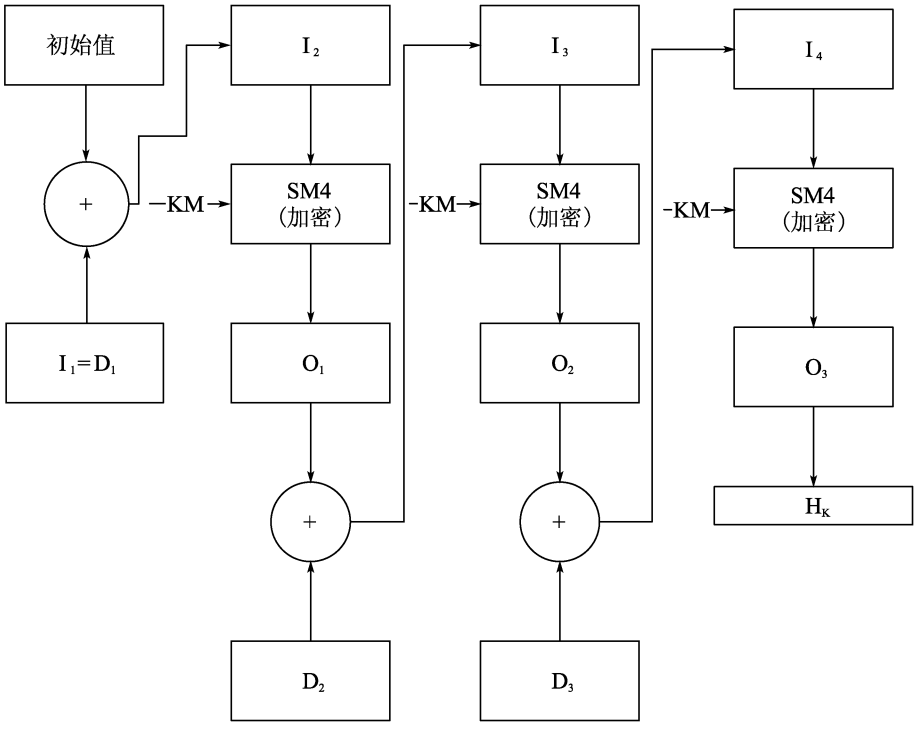
9.2.2.3 密文计算

用 MAC 过程密钥以 CBC 模式的分组加密处理 16 字节块 $X_1, X_2, \cdots, X_K$：

H_i：= ALG(K)[$X_i \oplus H_{i-1}$]，这里 $i = 1, 2, \cdots, K$。

H_0 的初始值 H_0：=（‘00’||‘00’||‘00’||‘00’||‘00’||‘00’||‘00’||‘00’||‘00’||‘00’||‘00’||‘00’||‘00’||‘00’||‘00’||‘00’）。

计算过程见图 24。



说明：

I——输入；　　D——数据块；

O——输出；　　KM——MAC 过程密钥；

+——异或。

图 24　使用 SM4 算法计算 HK 的流程

9.3 国际算法非对称加密机制

9.3.1 用于报文恢复的数字签名方案

9.3.1.1 概述

使用 GB/T 27929 规范的 HASH 函数，对指定报文数据恢复数字签名。

9.3.1.2 算法

数字签名方案使用下面两种算法：

a) 一种可逆的非对称算法，由一个依赖于私钥 S_K 的签名函数 $Sign(S_K)[\ ]$ 和一个依赖于公钥 P_K 的恢复函数 $Recover(P_K)[\ ]$ 组成。两个函数都将 N 字节的数字映射为 N 字节的数字，并且任何 N 字节的数字 X 有以下特性：

$Recover(P_K)[Sign(S_K)[X]] = X$；

b) 一种哈希算法 Hash[ ]，将任意长度的报文映射为一个 20 字节的哈希值。

9.3.1.3 数字签名产生

对由至少 $N-21$ 字节长的由任意长数据 L 组成的报文 MSG 计算签名 S 的步骤如下：

a) 计算报文 M 的 20 字节的 HASH 值 H：= Hash[MSG]；

b) 将 MSG 拆分成两部分 MSG = (MSG1 || MSG2)，其中，MSG1 由 MSG 最左端（最高位）的 $N-22$ 个字节组成，MSG2 由 MSG 剩余的（最低位）的 $L-N+22$ 个字节组成；

c) 定义一个字节 B：= '6A'；

d) 定义一个字节 E：= 'BC'；

e) 将 N 字节的块 X 定义为块 B，MSG1，H 和 E 的连接，因此，X：= (B || MSG1 || H || E)；

f) 数字签名 S 被定义为 N 个字节的数字：S：= $Sign(S_K)[X]$。

9.3.1.4 数字签名验证

数字签名验证步骤如下：

a) 检查数字签名 S 是否由 N 个字节组成；

b) 由数字签名 S 恢复得到 N 个字节的数字 X，X = $Recover(P_K)[S]$；

c) 将块 X 分割成 X = (B || MSG1 || H || E)：

1) B 为 1 字节长；

2) H 为 20 字节长；

3) E 为 1 字节长；

4) MSG1 由剩余的 $N-22$ 个字节组成。

d) 检查字节 B 是否等于'6A'；

e) 检查字节 E 是否等于'BC'；

f) 计算 MSG = (MSG1 || MSG2)，并检查是否满足 H = Hash[MSG]。

当且仅当这些检查都正确时，这条接收的报文被认为是真实的。

9.4 国密算法非对称加密机制

9.4.1 算法

SM2 签名方案使用下面三种函数：

a) 一个依赖于私钥 S_K 的签名函数 $Sign(S_K)[M]$，该函数输出两个相同长度的数字 r 和 s；

b) 一个依赖于公钥 P_K 的验证函数 $Verify(P_K)[M, Sign(S_K)[M]]$，该函数输出 True 或 False，表示验证正确或失败；

c) 一个哈希算法 H[ ]，将任意长度的报文映射为一个 32 字节的哈希值。

9.4.2 数字签名产生

对任意长度的数据组成的报文 MSG 计算签名 S 的步骤如下：

a) 计算 $Z_A = H_{256}(ENTL_A || ID_A || a || b || x_G || y_G || x_A || y_A)$。其中 ID_A 固定设置为 16 字节定长的十六进制数据 0x31，0x32，0x33，0x34，0x35，0x36，0x37，0x38，0x31，0x32，0x33，0x34，0x35，0x36，0x37，0x38；$ENTL_A$ 值为两个字节数据 0x00，0x80；

b) 计算报文 MSG 的 32 字节的 Hash 值 $h := H_{256}[Z_A||MSG]$;

c) 计算 $Sign(S_K)[h]$,得到两个数字 r 和 s;

d) 数字签名 S 被定义为 $S := r||s$,即数字签名 S 由数字 r 和 s 串联而成。

9.4.3 数字签名验证

对任意长数据组成的报文 MSG 验证签名 S 的过程如下:

a) 计算 $Z_A = H_{256}(ENTL_A || ID_A || a || b || x_G || y_G || x_A || y_A)$。其中 ID_A 固定设置为 16 字节定长的十六进制数据 0x31, 0x32, 0x33, 0x34, 0x35, 0x36, 0x37, 0x38, 0x31, 0x32, 0x33, 0x34, 0x35, 0x36, 0x37, 0x38;$ENTL_A$ 值为两个字节数据 0x00, 0x80;

b) 计算报文 MSG 的 32 字节的 Hash 值 $h := H_{256}[Z_A||MSG]$; $Verify(P_K)[h,S]$,若函数输出 True 表示验证正确,若输出 False,表示验证失败。

10 系统安全要求

10.1 概述

各级系统间的关系见 JT/T 978.1,各级系统的安全保障应按照本节要求进行。

10.2 安全要求

清分结算系统和入网机构系统的规划、设计、建设、验收、运维等全生命周期的安全保障应按国家信息安全和信息系统安全等级保护相关要求进行。

10.3 系统等级保护要求

部、省(区域)、省会城市的清分结算系统和入网机构系统应按国家信息安全等级保护要求和 JT/T 904—2014 确定为三级,开展规划、设计、建设、测评和运维等安全保障工作;其他城市根据发卡量和资金规模等参照国家相关标准和 JT/T 904—2014,自行评估对社会秩序、公众利益的作用和影响确定信息安全等级保护保护等级,按相关标准执行。

附 录 A
（规范性附录）
算法

A.1 对称加密算法

A.1.1 DES

DES 算法是以 64 位分组为单位进行运算，密钥长度为 8 字节。该算法被允许用于安全报文传送 MAC 机制密文运算，算法的详细过程见 GB/T 27929，并参考 GB/T 17964。

3DES 加密是指使用双长度（16 字节）密钥 K = (KL||KR) 将 8 字节明文数据分组加密成密文数据分组，如下所示：

Y = DES(KL)[DES-1(KR)[DES(KL)[X]]]。

解密的方式如下：

X = DES-1(KL)[DES(KR)[DES-1(KL)[Y]]]。

A.1.2 SM4

算法定义见 GM/T 0002。

A.2 非对称加密算法

A.2.1 RSA

A.2.1.1 该可逆算法是经批准用于加密和生成数字签名的算法。公钥指数的值只允许是 3 和$2^{16}+1$。该算法产生的密文及数字签名的长度与模长，见表 A.1。

表 A.1 对模长字节数的强制上限

名 称	最大长度
根公钥模	248
发卡机构公钥模	248
卡片公钥模	248

A.2.1.2 根公钥模的长度 N_{CA}，发卡机构公钥模的长度 N_I，卡片公钥模的长度 N_{IC}，应满足 $N_{IC} \leqslant N_I \leqslant N_{CA}$ 和 $N_{PE} \leqslant N_I \leqslant N_{CA}$。在选择公钥模长时，应考虑到比较密钥生命周期同预期的因数分解进程。

A.2.1.3 发卡机构公钥指数和卡片公钥指数的值由发卡机构决定。密钥管理系统，发卡机构和卡片公钥指数应等于 3 和 $2^{16}+1$。

A.2.1.4 标识本数字签名算法的公钥算法标识应编码为十六进制‘01’。

注 1：卡片中一个记录的长度最长不超过 254 字节（包括 Tag 和 Length），因而实际卡片公钥和发卡机构公钥长度应小于最大长度 248 字节。命令数据长度最长为 255 字节，响应数据最长为 256 字节，动态签名数据作为卡片响应数据，也限制了卡片公钥的最大长度。

注 2：如卡片具备 DDA 和 CDA，包含卡片证书的记录模板的长度，即卡片公钥证书长度加上证书（“9F46”）和记录模板（“70”）的 Tag 和 Length 不超过 254 字节，则卡片公钥长度不超过 247 字节，因而发卡机构公钥长度最大长度也不超过 247 字节。

注 3：根据发卡机构应用数据长度不同，卡片公钥最大长度在 205 到 240 之间。若生成应用密文（GENERATE AC）命

令响应包含其他可选数据,卡片公钥最大长度还应减去这些数据的长度(包括 Tag 和 Length)。若卡片应用具备内部认证(INTERNAL AUTHENTICATION)命令的格式二,卡片公钥最大长度还应减去 7 字节。

A.2.1.5 密钥

使用奇数公钥指数 e 的 RSA 数字签名方案的私钥 S_K 有两个素数 p 和 q,满足:$p-1$、$q-1$ 与 e 互质,以及私钥 d,满足:$ed \equiv 1 \bmod (p-1)(q-1)$。组成相对应的公钥 P_K 由公钥模 $n=pq$ 和公钥指数 e 组成。

A.2.1.6 签名函数

使用奇数公钥指数的 RSA 签名函数被定义为:$S = Sign(S_K)[X] := X^d \bmod n, 0 < X < n$。

这里 X 是用于签名的数据,S 为对应的数字签名。

A.2.1.7 恢复函数

使用奇数公钥指数的 RSA 恢复函数被定义为:$X = Recover(P_K)[S] := S^e \bmod n$。

A.2.1.8 密钥的生成

密钥管理系统与发卡机构应对其各自的 RSA 公/私钥生成过程的安全性负责。

A.2.2 SM2

算法定义见 GM/T 0003。

A.3 哈希算法

A.3.1 SHA-1

SHA-1 对任意长度的报文的输入,产生一个 20 字节的哈希值。SHA-1 算法见 GB/T 18238.3。

本哈希算法的标志编码为十六进制数'01'。

A.3.2 SM3

算法定义见 GM/T 0004。

附 录 B
（规范性附录）
算法选择与交易流程

B.1 新增数据元

双算法卡片同时支持 RSA/SHA1/DES 和 SM2/SM3/SM4 算法（以下简称“SM 算法”），在卡片进行交易时，需根据新增的数据元进行算法选择。新增的数据元见表 B.1。

表 B.1 新增数据元

数据元名称	标 签	长 度	格 式
SM 算法支持指示器	DF69	1	b

B.2 算法应用方案

B.2.1 本条包括单 SM 算法卡和双算法卡两种方案的技术要求。单 SM 算法卡指一张卡片支持 SM2/SM3/SM4 算法；双算法卡指一张卡片同时支持 SM2/SM3/SM4 与 RSA/SHA-1/3DES 两套算法。如采用单 SM 算法卡方案，在卡片个人化阶段，需要向卡内写入支持 SM2 算法所需的相关数据元；如采用双算法卡方案，在卡片个人化阶段，需要向卡内写入支持 RSA 算法和 SM2 算法两种算法所需的相关数据元。卡片个人化完成之后，卡片交易时应通过和终端之间的交互确定使用的算法。

B.2.2 技术要求主要包括如下：本条所述支持 SM 算法的终端应同时支持 RSA/SHA-1/3DES 及 SM2/SM3/SM4 两套算法环境，并使用 SM 算法支持指示器进行算法选择。

a) 卡片个人化阶段：
   1) 单 SM 算法卡：在卡片个人化数据包括了 SM2 算法所需的所有数据元；
   2) 双算法卡：在卡片个人化数据包括了 RSA 算法所需的所有数据元，同时包括了 SM2 算法所需的所有数据元。
b) 卡片应用流程：见图 B.1；
c) 应用执行情况：不同类型终端和卡片在标准扣款交易应用的执行情况见表 B.2。

表 B.2 标准扣款交易应用的执行情况

类 型	仅支持 RSA/SHA-1/3DES 算法的卡片	双 算 法 卡	单 SM 算法卡
仅支持 RSA/SHA-1/3DES 算法的终端	RSA/SHA-1/3DES 算法流程	RSA/SHA-1/3DES 算法流程	脱机验证失败，尝试进行联机交易
支持双算法的终端	RSA/SHA-1/3DES 算法流程	SM 算法流程	SM 算法流程

B.3 基于 SM 算法的标准扣费交易流程

使用 SM 算法的标准扣费交易流程与使用 RSA/SHA-1/3DES 算法的流程基本一致，见图 B.1。主要区别在基于 SM 算法的支付流程中需要终端和卡片增加算法选择的步骤，并且卡片在收到获取处理选项（GPO）指令以后，需要根据终端发送的 DF69 进行判定，若发现卡片不具备终端要求的算法，则卡片需要返回获取处理选项（GPO）指令的状态码为‘6985’，从而实现脱机拒绝。若终端和卡片均支持并选择了 SM 算法进行交易处理，则卡片需要返回采用 SM 算法计算 TC、动态认证数据、SM 算法的发卡机

构自定义数据(“9F10”)、SM 算法对应的 AFL 等数据给终端,终端再进行数据读取以及完成 fDDA 认证操作。

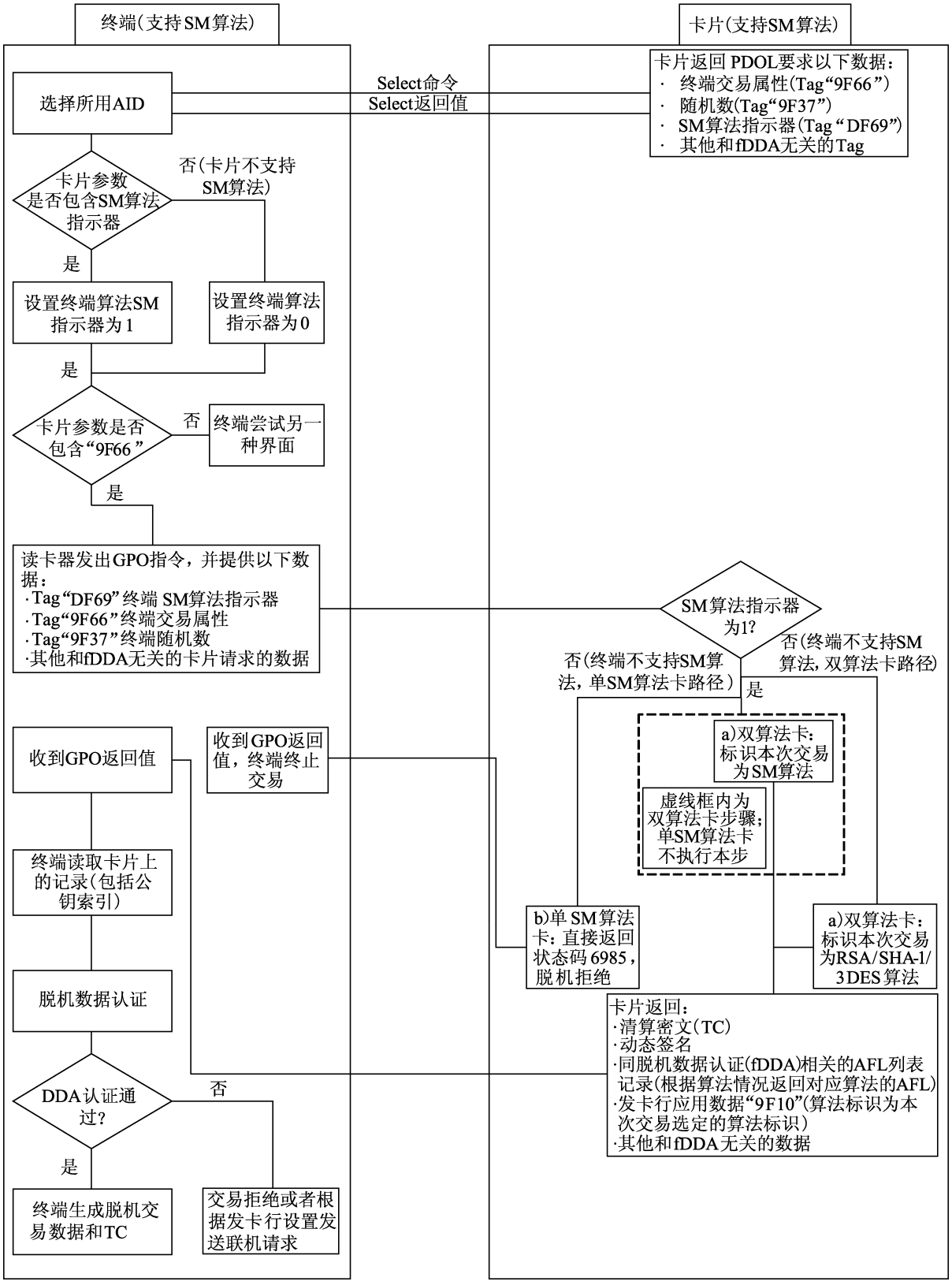


图 B.1　标准扣费交易流程

B.4 个人化相关密钥的初始化

KMC 密钥的长度应为 16 字节。个人化密钥(KENC、KMAC 和 KDEK)的产生采用 SM4 算法进行：

a) KENC：=SM4(KMC)[KEYDATA 的最右 6 个字节||'F0'||'01'|| KEYDATA 的最右 6 个字节||'0F'||'01']；

b) KMAC：=SM4(KMC)[KEYDATA 的最右 6 个字节||'F0'||'02'|| KEYDATA 的最右 6 个字节||'0F'||'02']；

c) KDEK：=SM4(KMC)[KEYDATA 的最右 6 个字节||'F0'||'03'|| KEYDATA 的最右 6 个字节||'0F'||'03']。

参 考 文 献

[1] GB/T 17964 信息安全技术 分组密码算法的工作模式。