

中华人民共和国交通运输部

城市公共交通 IC 卡安全技术规范 (试行)

中华人民共和国交通运输部 发布

目次

前言	III
1 范围	1
2 规范性引用文件	1
3 术语与定义	2
4 符号和缩略语	6
5 电子现金密钥管理体系	8
5.1 认证中心公钥管理	8
5.2 发卡机构公钥管理	13
5.3 全国运营机构及发卡机构对称密钥管理	13
6 电子钱包密钥管理体系	15
6.1 密钥关系表	15
6.2 子密钥推导方法	16
7 电子现金动态数据认证	19
7.1 国际算法密钥和证书	20
7.2 国际算法动态数据认证 (DDA)	21
7.3 国密算法密钥和证书	28
7.4 国密算法动态数据认证 (DDA)	29
8 电子现金应用密文和发卡机构认证	33
8.1 应用密文产生	33
8.2 国际算法发卡机构认证	35
8.3 国密算法发卡机构认证	36
8.4 密钥管理	36
9 电子现金行业信息的保护	37
9.1 密钥说明	37
9.2 安全机制	37
10 安全报文	37
10.1 报文格式	38
10.2 电子现金国际算法报文完整性及其验证	38
10.3 电子现金国密算法报文完整性及其验证	38
10.4 电子钱包报文完整性及其验证	39
10.5 电子现金国际算法报文私密性	41
10.6 电子现金国密算法报文私密性	41
10.7 电子钱包报文私密性	41
10.8 电子现金密钥管理	45

目次

11	卡片安全	45
11.1	共存应用	45
11.2	密钥的独立性	45
11.3	卡片内部安全体系	45
11.4	卡片中密钥的种类	48
12	终端安全	49
12.1	终端数据安全性要求	49
12.2	终端设备安全性要求	50
12.3	终端密钥管理要求	51
13	个人化安全	53
13.1	安全综述	53
13.2	初始化安全	53
13.3	密钥定义	54
13.4	管理要求	56
13.5	安全模块	61
13.6	风险审计	62
14	安全机制	62
14.1	国际算法对称加密机制	62
14.2	国密算法对称加密机制	66
14.3	国际算法非对称加密机制	70
14.4	国密算法非对称加密机制	70
15	认可的算法	71
15.1	对称加密算法	71
15.2	非对称加密算法	72
15.3	哈希算法	73

前 言

城市公共交通IC卡系列技术规范由5个规范组成：

- 《城市公共交通IC卡卡片技术规范》；
- 《城市公共交通IC卡读写终端技术规范》；
- 《城市公共交通IC卡信息接口技术规范》；
- 《城市公共交通IC卡非接触接口通讯技术规范》；
- 《城市公共交通IC卡安全技术规范》。

本规范由中华人民共和国交通运输部提出。

本规范主要起草单位：

本规范主要起草人：

本规范为首次发布。

城市公共交通 IC 卡安全技术规范

1 范围

本规范描述了城市公共交通IC卡安全功能方面的要求以及为实现这些安全功能所涉及的安全机制和获准使用的加密算法，包括：IC卡动态数据认证方法、IC卡和发卡机构之间的通讯安全以及相关的对称及非对称密钥的管理。具体内容如下：

- 动态数据认证；
- 应用密文和发卡机构认证；
- 行业信息的保护；
- 安全报文；
- 卡片安全；
- 终端安全；
- 对称和非对称密钥管理体系；
- 个人化安全。

此外，还包括为实现这些安全功能所涉及的安全机制和获准使用的加密算法的规范。

本规范适用于开展基于城市公共交通IC卡应用的地区、发卡机构以及收单机构。其使用对象主要是与城市公共交通IC卡应用相关的卡片、终端及加密设备等的设计、制造、管理、发行以及应用系统的研制、开发、集成和维护等相关部门（单位）。

2 规范性引用文件

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

GB/T 16649.5 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注册程序（GB/T 16649.5—2002）

GB/T 20547.2 银行业务 安全加密设备（零售） 第2部分：金融交易中设备安全符合性检测清单（GB/T 20547.2—2006）

GM/T 0002-2012 SM4分组密码算法

GM/T 0003.1-2012 SM2椭圆曲线公钥密码算法 第1部分：总则

GM/T 0003.2-2012 SM2椭圆曲线公钥密码算法 第2部分：数字签名算法

GM/T 0003.3-2012 SM2椭圆曲线公钥密码算法 第3部分：密钥交换协议

GM/T 0003.4-2012 SM2椭圆曲线公钥密码算法 第4部分：公钥加密算法

GM/T 0003.5-2012 SM2椭圆曲线公钥密码算法 第5部分：参数定义

GM/T 0004-2012 SM3密码杂凑算法

GB/T 16649.4 识别卡 带触点的集成电路卡 第4部分：用于交换的结构、安全和命令（GB/T 16649.4—2010）

GB/T 21078.1 银行业务 个人识别码的管理与安全 第1部分：ATM和POS系统中联机PIN处理的基本原则和要求（GB/T 21078.1-2007）

GB/T 15852.1 信息技术 安全技术 消息鉴别码 第1部分：采用分组密码的机制（GB/T 15852.1-2008）

GB/T 27929 银行业务 采用对称加密技术进行报文鉴别的要求 (GB/T 27929-2001)

GB/T 17964 信息安全技术 分组密码算法的工作模式 (GB/T 17964-2008)

3 术语与定义

下列术语和定义适用本规范。

3.1

提前回收 accelerated revocation

在已公布的密钥失效日期到期前回收密钥。

3.2

应用 application

卡片和终端之间的应用协议和相关的数据集。

3.3

非对称加密技术 asymmetric cryptographic technique

采用两种相关变换的加密技术：公开变换（由公钥定义）和私有变换（由私钥定义）。这两种变换存在在获得公开变换的情况下是不能够通过计算得出私有变换的特性。

3.4

认证 authentication

确认一个实体所宣称的身份的措施。

3.5

字节 byte

由指明的8位数据b1到b8组成，从最高有效位（MSB，b8）到最低有效位（LSB，b1）。

3.6

卡片 card

城市公共交通IC卡系统定义的支付卡片。

3.7

证书 certificate

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

3.8

认证中心 certification authority

证明公钥和其它相关信息同其拥有者相关联的可信的第三方机构。

3.9

命令 command

终端向IC卡发出的一条报文，该报文启动一个操作或请求一个响应。

3.10

泄露 `compromise`

机密或安全被破坏。

3.11

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IC卡发到终端时的顺序相同，即高位字节在前。在每个字节中位按由高到低的顺序排列。

3.12

密文 `cryptogram`

加密运算的结果。

3.13

加密算法 `cryptographic algorithm`

为了隐藏或显现数据信息内容的变换算法。

3.14

密钥有效期 `cryptoperiod`

某个特定的密钥被授权可以使用的时间段，或者某个密钥在给定的系统中有效的时间段。

3.15

解密 `decipherment`

对应加密过程的逆操作。

3.16

数字签名 `digital signature`

对数据的一种非对称加密变换。该变换可以使数据接收方确认数据的来源和完整性，保护数据发送方发出和接收方收到的数据不被第三方篡改，也保护数据发送方发出的数据不被接收方篡改。

3.17

电子现金 (EC) `electronic cash (EC)`

基于城市公共交通IC卡应用上实现的小额支付功能。

3.18

电子钱包 (EP) `electronic purse (EP)`

一种为方便持卡人小额消费而设计的金融IC卡应用。它支持圈存、消费等交易。

3.19

加密 `encipherment`

基于某种加密算法对数据做可逆的变换从而生成密文的过程。

3.20

交易 transaction

由于持卡者和商户之间的商品或服务交换行为而在持卡者、发卡机构、商户和收单机构之间产生的信息交换、资金清算和结算行为。

3.21

哈希函数 hash function

将位串映射为定长位串的函数，它满足以下两个条件：

——对于一个给定的输出，不可能推导出与之相对应的输入数据；

——对于一个给定的输入，不可能通过计算得到具有相同的输出的另一个输入。

另外，如果要求哈希函数具备防冲突功能，则还应满足以下条件：

——不可能通过计算找到两个不同的输入具有相同的输出。

3.22

哈希结果 hash result

哈希函数的输出位串。

3.23

集成电路 integrated circuit(s)

具有处理和/或存储功能的电子器件。

3.24

集成电路卡 integrated circuit(s) card

内部封装一个或多个集成电路用于执行处理和存储功能的卡片。

3.25

接口设备 interface device

终端上与城市公共交通IC卡进行通讯处理的部分，包括其中的机械和电气部分。

3.26

密钥 key

控制加密转换操作的符号序列。

3.27

密钥失效日期 key expiry date

用特定密钥产生的签名不再有效的最后期限。用此密钥签名的发卡机构证书必须在此日期或此日期之前失效。在此日期后，此密钥可以从终端删除。

3.28

密钥引入 key introduction

产生、分发和开始使用密钥对的过程。

3.29

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。

3.30

密钥更换 key replacement

回收一个密钥，同时引入另外一个密钥来代替它。

3.31

密钥回收 key revocation

回收使用中的密钥以及处理其使用后的遗留问题的密钥管理过程。密钥回收可以按计划回收或提前回收。

3.32

密钥回收日期 key revocation date

在此日期后，任何仍在使用的合法卡不会包含用此密钥签名的证书。因此，密钥可以从终端上被删除。对按计划密钥回收，密钥回收日期应等同于密钥失效日期。

3.33

逻辑泄露 logical compromise

由于密码分析技术和/或计算能力的提高，对密钥造成的泄露。

3.34

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。

3.35

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。

3.36

填充 padding

向数据串某一端添加附加位。

3.37

明文 plaintext

未被加密的信息。

3.38

物理泄露 physical compromise

由于没有安全的保护，或者硬件安全模块的被盗或被未经授权的人存取等事实对密钥造成的泄露。

3.39

计划回收 `planned revocation`

按照公布的密钥失效日期进行的密钥回收。

3.40

潜在泄露 `potential compromise`

密码分析技术和/或计算能力的提高达到了可能造成某个特定长度的密钥的泄露的情况。

3.41

私钥 `private key`

一个实体的非对称密钥对中含有的供实体自身使用的密钥，在数字签名方案中，私钥用于签名。

3.42

公钥 `public key`

在一个实体使用的非对称密钥对中可以公开的密钥。在数字签名方案中，公钥用于验证。

3.43

公钥证书 `public key certificate`

由认证中心签名的不可伪造的某个实体的公钥信息。

3.44

保密密钥 `secret key`

对称加密技术中仅供指定实体所用的密钥。

3.45

响应 `response`

IC卡处理完成收到的命令报文后，返回给终端的报文。

3.46

SM 算法 `SM algorithm`

使用SM2/SM3/SM4分别作为非对称算法/哈希/对称算法的算法环境，相对于RSA/SHA-1/DES环境。

3.47

对称加密技术 `symmetric cryptographic technique`

发送方和接收方使用相同保密密钥进行数据变换的加密技术。在不掌握保密密钥的情况下，不可能推导出发送方或接收方的数据变换。

3.48

终端 `terminal`

在交易点安装、用于与IC卡配合共同完成交易的设备。它应包括接口设备，也可包括其它的部件和接口（如与主机的通讯）。

4 符号和缩略语

下列缩略语和符号适用于本部分。

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)
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 (Binary)
$C:=(A B)$	将m位数字B和n位数字A进行链接, 定义为: $C=2^m A+B$
CBC	密码块链接 (Cipher Block Chaining)
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)
cn	压缩数字型 (Compressed Numeric)
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)
ES	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)
FIPS	联邦信息处理标准 (Federal Information Processing Standard)
$H:=Hash[MSG]$	用160位的HASH函数对任意长度的报文MSG进行HASH运算。
IC	集成电路 (Integrated Circuit)
ICC	集成电路卡 (Integrated Circuit Card)
IFD	接口设备 (Interface Device)
INS	命令报文的指令字节 (Instruction Byte of Command Message)
K_S	过程密钥 (Session Key)
L_{DD}	IC卡动态数据长度 (Length of the ICC Dynamic Data)
MAC	报文鉴别码 (Message Authentication Code)
MMYY	月、年 (Month, Year)
n	数字型 (Numeric)
N_{CA}	认证中心公钥模长 (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}	IC卡公钥模长 (Length of the ICC Public Key Modulus)
P1	参数1 (Parameter 1)
P2	参数2 (Parameter 2)
PAN	主账号 (Primary Account Number)
P_{CA}	认证中心公钥 (Certification Authority Public Key)

P_I	发卡机构公钥 (Issuer Public Key)
P_{IC}	IC卡公钥 (ICC Public 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
S_{CA}	认证中心私钥 (Certification Authority Private Key)
SDA	静态数据认证 (Static Data Authentication)
SFI	短文件标识符 (Short File Identifier)
SHA	安全哈希算法 (Secure Hash Algorithm)
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
TC	交易证书 (Transaction Certificate)
TLV	标签、长度、值 (Tag Length Value)
$X := \text{Recover}(PK) [Y]$	用公钥PK, 通过非对称可逆算法, 对数据块Y进行恢复
$X := \text{ALG}^{-1}(K) [Y]$	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
$Y := \text{ALG}(K) [X]$	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
$Y := \text{Sign}(SK) [X]$	用私钥SK, 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
$A \equiv B \pmod n$	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, $0 \leq r < n$, 存在一个整数d, 使得 $A=dn+r$
$A:=B$	A被赋予数值B

5 电子现金密钥管理体系

城市公共交通IC卡系统和发卡机构都需要建立一套完整的密钥管理体系,城市公共交通IC卡系统需要建立认证中心,负责管理和使用用于动态数据认证的认证中心公私钥对。发卡机构需要建立一套完整的密钥管理体系,用于动态数据认证和交易流程。

5.1 认证中心公钥管理

本条定义了在城市公共交通IC卡系统中管理认证中心公钥的原则与策略的总体框架,认证中心公钥用于第7章指明的动态数据认证。

5.1.1 认证中心公钥生命周期

在普通环境下的认证中心公钥的生命周期可以被分为以下的连续的阶段:

- 计划;
- 生成;
- 分发;
- 使用;
- 回收 (按计划)。

5.1.1.1 计划

在计划阶段，城市公共交通IC卡系统调查和研究在不远的将来导入新的认证中心公钥的需求。这些需求包括需要密钥的数量以及这些密钥的参数。

计划阶段一个重要的部分是评估非对称加密算法的安全性，来决定已存在的或将要采用的新的密钥的预期生命期。这样的评估引导了对新密钥的长度和失效日期的设置，潜在的对已存在密钥的失效日期的修改，以及更换密钥的全面计划。

5.1.1.2 生成

如果计划阶段的结果需要导入新的认证中心公钥，这些密钥必须由认证中心产生。更准确的说，认证中心将以一种安全的方式来产生需要的认证中心公私钥对，以供将来使用。

在生成之后必须保证认证中心私钥的私密性，认证中心公钥与私钥的完整性也必须保证。

5.1.1.3 分发

在密钥分发阶段，认证中心将新产生的认证中心公钥发布给它的成员发卡机构和收单机构作以下的用途：

- 对于发卡机构，用于在使用阶段校验由认证中心提供的发卡机构公钥证书；
- 对于收单机构，用于将认证中心公钥安全地导入商户终端。

为了防止导入假的认证中心公钥，认证中心，发卡机构和收单机构之间的接口必须保证认证中心公钥分发的完整性。

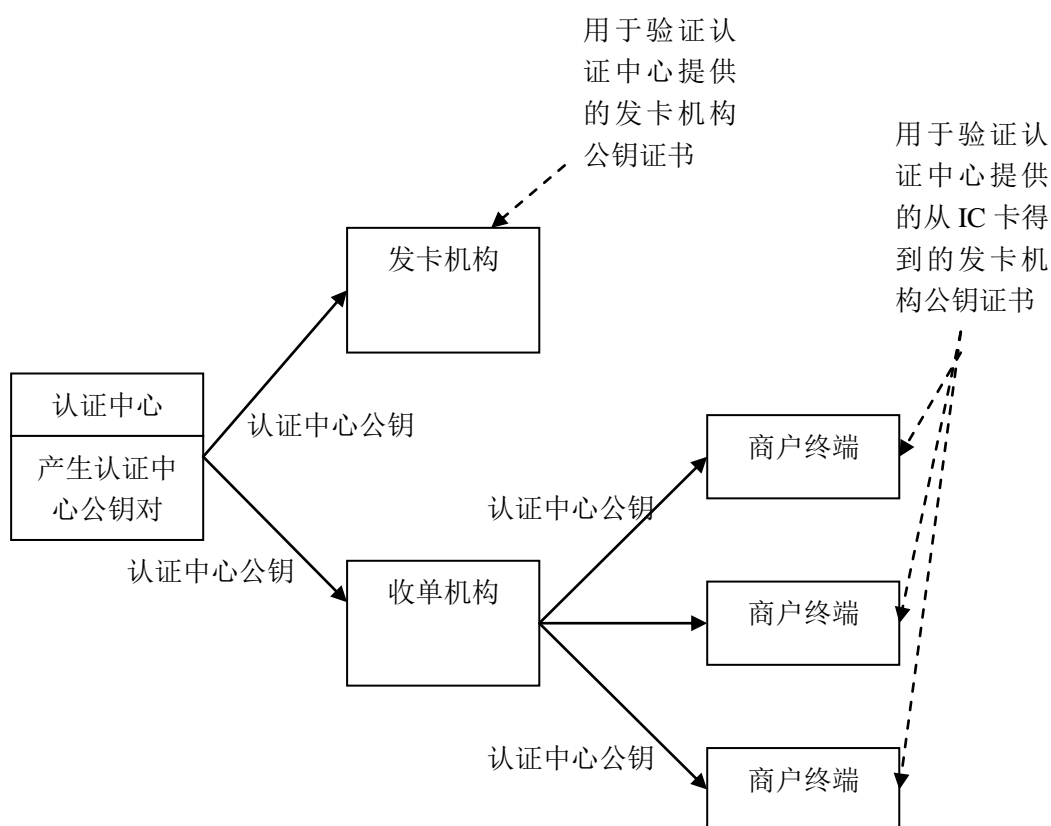


图1 认证中心公钥的分发

5.1.1.4 使用

认证中心公钥被商户终端用于完成静态或动态数据认证。认证中心私钥被认证中心用于生成发卡机构公钥证书。更准确地说，发生了下面的交互操作：

- 发卡机构生成自己的发卡机构公钥并发送给认证中心；
- 认证中心用认证中心私钥对发卡机构公钥签名以生成发卡机构公钥证书并将其发还给发卡机构；
- 发卡机构用认证中心公钥校验收到的发卡机构公钥证书是否正确。如果正确，发卡机构就可以将其作为 IC 卡的个人化数据的一部分。

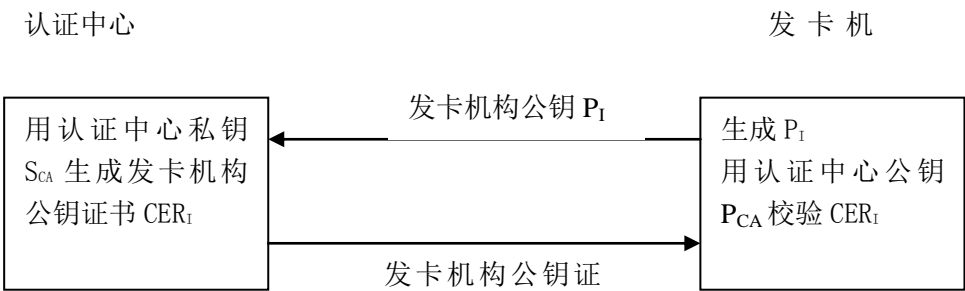


图2 发卡机构公钥的分发

为了防止导入假的发卡机构公钥，认证中心和发卡机构之间的接口必须保证提交的发卡机构公钥的完整性。

5.1.1.5 回收（按计划）

一旦某一对认证中心公钥到了计划阶段已设置好的失效日期，它必须从服务中删除。实际上，这意味着：

- 在失效日期之后，由认证中心私钥生成的发卡机构公钥证书就不再有效了。因此发卡机构必须保证用这样的发卡机构公钥证书个人化的 IC 卡在认证中心公钥的失效日期前中止使用；
- 在失效日期前的适当时候，认证中心应该停止用对应的认证中心私钥对发卡机构公钥签名。
- 在失效日期之后，收单机构应该在规定的期限内将认证中心公钥从终端中删除。

5.1.2 认证中心公私钥对泄漏

当认证中心公私钥对泄露时，必须实施紧急状态，这最终可能会导致在计划的失效日期之前将认证中心公钥提前回收。在这种情况下，密钥生命周期中会有一些附加的阶段：

- 监测；
 - 评估；
 - 决策；
 - 回收（提前的）。
- 这些阶段在下面进行描述。

5.1.2.1 监测

认证中心公私钥对的泄露可以是真正的泄露，例如，已经过确认的，认证中心的安全性被破坏；或者已经过确认的，密钥被用密码分析学的方法破解。另外，泄露也可能是：

- 有所怀疑的：系统监控，会员或持卡人的投诉显示有欺诈交易发生而且可能是由于密钥泄露引起的，但未经确认；
- 潜在的：密码分析学技术，例如因数分解已经发展到了在已有资源下可以将现有长度的任意密钥破解出来的水平，但没有证据表明这已经发生；
- 对密钥泄露的检测包括对城市公共交通 IC 卡系统被物理上非法闯入的察觉、由城市公共交通 IC 卡系统和它的会员安装的欺诈和 risk 管理系统对脱机的欺诈交易的报告以及从密钥组织收集到的因数分解技术发展的情报。

5.1.2.2 评估

对一个认证中心公钥泄露（或潜在的）的评估包括技术，风险，欺诈性以及最重要的对城市公共交通 IC 卡系统及其成员的商业影响。评估结果包括对泄露的确认，综合成本和泄露带来的风险后决定可能的系列行动，以及提供用来支持决策的评估结果。

5.1.2.3 决策

根据评估所产生的结果，城市公共交通 IC 卡系统将决定针对一个密钥泄露应采取的一系列行动。最坏的情况下，这个决定会包括在计划的失效日期之前对认证中心公钥的提前回收。

5.1.2.4 回收（提前的）

决定回收认证中心公钥后，需要通知城市公共交通 IC 卡系统成员相应密钥的新的失效日期。之后的处理和 5.1.1.5 条中所描述的按计划的回收一样。

5.1.3 认证中心密钥管理策略

5.1.3.1 计划阶段

在这个阶段主要任务包括：

- 1) 对现有公钥的抗攻击能力分析以及对新公钥对的需求分析；
- 2) 决定所需新公钥对的数量和参数，这些参数包括：
 - 公钥长度的选择；
 - 公钥失效期。

对于公钥失效期的问题，应遵循以下策略：

- 所有的认证中心公钥都将 12 月 31 日作为按计划的失效日期；
- 收单机构在按计划的失效日期以后有六个月的过渡期（直到下一个日历年的 6 月 30 日）从所有的终端上回收过期的密钥。强制密钥回收的情况不应在过渡期结束前出现，并且可以根据城市公共交通 IC 卡系统的判断而延迟回收；
- 所有新的认证中心公钥都在 12 月 31 日以前发布；
- 收单机构有六个月的过渡期（直到下一个日历年的 6 月 30 日）将新的密钥安装到所有的终端；
- 收单机构有六个月的时间在所有终端上安装新的密钥（截止到下一年的 6 月 30 号），无论何时，新的公钥将在 12 月 31 号前发布，从而有较长的时间进行公钥安装；
- 城市公共交通 IC 卡系统不会在下一年的 1 月 1 号前使新的认证中心公钥生效用于合法交易；
- 在提前回收的情况下，从所有终端上回收该密钥的六个月过渡期不变，但是固定的 12 月 31

日不适用。将密钥回收通知给会员以及时间安排是每个城市公共交通 IC 卡系统的责任。

5.1.3.2 生成阶段

在这个阶段主要任务包括：

- 认证中心以一种安全的方式来产生认证中心公私钥对；
- 对于每一个注册的应用提供商标识（RID），指向一个特定的认证中心公钥的认证中心公钥索引具有唯一的值。一个特定的密钥的认证中心公钥索引的值不能改变。

5.1.3.3 分发阶段

在这个阶段主要任务包括：

- 认证中心将新产生的认证中心公钥发布给它的成员发卡机构和收单机构；
- 收单机构将这些公钥导入到商户终端中去。

5.1.3.4 使用阶段

在这个阶段主要任务包括：

- 为发卡机构签发发卡机构证书；
- 发卡机构用认证中心公钥校验收到的发卡机构公钥证书是否正确，并通过卡片个人化装载到 IC 卡。

对于使用认证中心私钥进行签名，应遵循以下策略：

- 在将密钥发布给收单机构至少 6 个月后，认证中心才开始使用认证中心公私钥对中的私钥进行签名；
- 发卡机构所发行的用户 IC 卡的失效日期必须不晚于这张卡上的发卡机构公钥证书的失效日期，也必须不晚于用来生成这个发卡机构公钥证书的认证中心公钥已公布（在卡片发行时）的回收日期；
- 一张发卡机构公钥证书的失效日期必须不晚于用来生成这个发卡机构公钥证书的认证中心公钥已公布（在证书发放时）的回收日期；
- 一张 IC 卡公钥证书的失效日期必须不晚于用来生成这个 IC 卡公钥证书的发卡机构公钥的失效日期。

5.1.3.5 监测阶段

在这个阶段主要任务包括对认证中心私钥安全性进行监控。私钥泄露形式包括物理泄露、逻辑泄露、可疑泄露、潜在泄露以及已确认泄露。

5.1.3.6 评估阶段

本阶段只适用于提前回收。

在这个阶段主要任务是对由公钥泄漏带来的对商业运作的影响进行评估，包括：

- 确认泄露；
- 决定可能采取的系列行动；
- 比较该行动的成本和由泄露带来的成本及风险；
- 提交评估结果以支持决策。

5.1.3.7 决策阶段

本阶段只适用于提前回收。作为评估阶段的结果，该阶段城市公共交通IC卡系统决定对认证中心公私钥对的泄露采取一系列的行动。

5.1.3.8 回收阶段

在这个阶段主要任务包括：

- 1) 从服务中收回一个密钥及处理它使用后的遗留事项的密钥管理过程。密钥回收可以是按计划的，也可以是提前的。针对认证中心公钥的情况，回收意味着私钥不再用来生成发卡机构公钥证书；
- 2) 公钥的拷贝从商户终端中删除。

对于公钥的回收，应遵循以下策略：

- 所有的认证中心公钥都以 12 月 31 日作为计划的失效日期。收单机构有 6 个月的过渡期（直到下一个日历年的 6 月 31 日）来回收废除的密钥。强制密钥回收的情况不应在过渡期结束前出现，并且可以根据城市公共交通 IC 卡系统的判断而延迟回收。
- 遵循城市公共交通 IC 卡系统的规则，认证中心公钥的回收需要在 6 个月的时间内从所有终端的服务中撤回公钥部分。
- 在提前回收的情况下，导入和回收的预留时间和按计划的回收一样。但是，回收的日期根据城市公共交通 IC 卡系统的判断决定。

5.2 发卡机构公钥管理

发卡机构公钥管理可以参照认证中心公钥管理策略来制定其公钥管理策略。

在向认证中心申请公钥证书之前，发卡机构需要对密钥管理做一系列决策，它们包括：

- 需要产生的发卡机构公钥数量；
- 产生的密钥的长度。

发卡机构的公钥长度不能大于认证中心的最长密钥长度。

- 每个密钥的失效期

密钥的失效期必须不迟于认证中心用来签发证书的公钥的失效期。

- 密钥的指数

发卡机构制定好他们的密钥管理策略，并已经产生一对公私钥对后，将发卡机构公钥提交到认证中心。当从认证中心收到多个发卡机构公钥证书时，选择合适的证书加载到卡片中。

5.3 全国运营机构及发卡机构对称密钥管理

5.3.1 安全性要求

密钥管理系统必须具有用户安全管理、设备安全管理、密钥安全管理以及审计管理功能：

- 用户安全管理实现对系统操作员的权限进行控制和管理，防止系统被非法使用和越权使用；
- 设备安全管理实现系统中加密机等密码设备进行安全管理，密码设备必须具备相应的防止硬件攻击能力，并保证存储在密码设备上的密钥不能被非法读取或获得；
- 密钥安全管理，采用合理的安全性设计，确保密钥在存储、传输、使用等环节的安全；
- 审计管理，用来进行系统操作日志及其它相关信息的安全审计与管理。

5.3.2 功能性要求

5.3.2.1 密钥类型

系统管理的对称密钥种类见表1。

表1 管理的对称密钥类型

密钥类型	用途	长度
应用密文主密钥	产生IC卡应用密文子密钥，用于应用密文的产生和验证	16字节
安全报文认证（MAC）密钥	产生IC卡MAC子密钥，用于安全报文鉴别码的产生和验证	16字节
安全报文加密密钥	产生IC卡加密子密钥，用于加密解密安全报文	16字节
应用开通密钥	产生IC卡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字节
扩展应用管理密钥	产生IC卡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字节

发卡机构主密钥可以分散出IC卡子密钥，在交易过程中从子密钥派生出相应的过程密钥。其中MAC密钥用来产生报文的鉴别码（MAC），用于安全报文命令，如数据安全更新、发卡机构脚本等。加密密钥用来加密安全报文，AC密钥用来对TC、ARQC、AAC、ARPC进行加密计算。应用开通密钥用来产生与扩展应用相关的安全报文鉴别码（MAC），该鉴别码（MAC）主要用于增加扩展应用扩展文件的记录。扩展应用管理密钥用来产生与扩展应用相关的安全报文鉴别码（MAC），该鉴别码（MAC）主要用于保护扩展应用扩展文件中记录的信息。

5.3.2.2 密钥管理

系统必须实现如下密钥管理功能：

- 密钥产生功能，根据用户输入采用特定的密钥输入算法产生系统所需要的密钥，密钥产生可以采用种子码单方式，也可以采用随机生成的方式；
- 密钥传输功能，将系统密钥安全传输到交易认证设备或发卡加密设备中；
- 密钥备份、恢复功能，提供系统密钥的备份和恢复功能，以便于在系统崩溃时对系统密钥进行灾难性恢复；
- 密钥更新和回收。

5.3.2.3 加密设备功能

系统设备必须能够实现以下功能：

- 密钥分散功能，按照 14 章定义的分散方法，从保存在加密设备中的发卡机构主密钥分散出唯一的 IC 卡子密钥；
- 过程密钥生成功能，按照 14 章定义的分散方法，根据子密钥和输入数据，分散出过程密钥；
- 数据加密功能，根据子密钥或过程密钥进行数据加、解密；
- MAC 产生功能，根据 MAC 过程密钥和欲进行计算的数据，产生数据的校验码；
- ARQC 校验功能，根据交易数据校验 ARQC 的正确性；
- ARPC 生成功能，根据交易数据产生 ARPC。

5.3.3 密钥管理系统的部署

5.3.3.1 应用密钥的部署

与应用相关的密钥按维护主体分为如下三类：

- 由发卡机构自行维护：包括应用密文主密钥、安全报文认证（MAC）密钥、安全报文加密密钥、应用开通密钥。
- 由发卡机构或行业方自行维护：地区扩展应用管理密钥，这类密钥既可以由发卡机构自行维护，

也可以由与发卡机构合作的行业方维护，主要用于对区域性扩展应用信息的保护。

- 由全国运营机构统一管理：互联互通主密钥，为保证全国的联网通用，该密钥由全国运营机构统一管理，并按省级交通厅、发卡机构、用户卡的层级采用三级密钥管理体系，主要用于对全国性扩展应用信息的保护。

5.3.3.2 互联互通密钥管理系统

主要负责：

- 互联互通主密钥的生成和维护；
- 省级互联互通一级密钥的生成（通过分散互联互通主密钥）；
- 互联互通 SAM 卡中互联互通主密钥的灌装；
- 下属发卡机构互联互通二级密钥的生成（通过分散省级互联互通一级密钥）；
- 下属发卡机构互联互通二级密钥的分发。

5.3.3.3 发卡机构

主要负责：

- 发卡机构所有应用主密钥的生成（应用密钥的维护主体不同可能导致相关密钥的生成方式不同）；
- 用户卡中所有应用密钥的生成（通过分散发卡机构应用主密钥）；
- 用户卡中所有应用密钥的灌装。

6 电子钱包密钥管理体系

本部分描述了与电子钱包应用相关的设备实体之间的密钥关系，此处还描述了IC卡密钥的推导方法和过程密钥的产生方法。

6.1 密钥关系表

表2 IC 卡中存储的共用于电子存折和电子钱包应用的密钥

密钥	发卡机构	IC卡	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用应用序列号推导获得。	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用应用序列号推导获得。	N/A
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用应用序列号推导获得。	N/A
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用应用序列号推导获得。	N/A
应用解锁密钥	应用解锁主密钥（MUBK）	应用主控解锁子密钥（MUBK），由MUBK用应用序列号推导获得。	N/A

6.2 子密钥推导方法

本条描述了IC卡中密钥的推导方法。图3和图4描述了DPK推导的过程。

6.2.1 DPK 左半部分的推导方法

推导双倍长DPK左半部分的方法：

- 将应用序列号的最右 16 个数字作为输入数据；
- 将 MPK 作为加密密钥；
- 用 MPK 对输入数据进行 3DEA 运算。

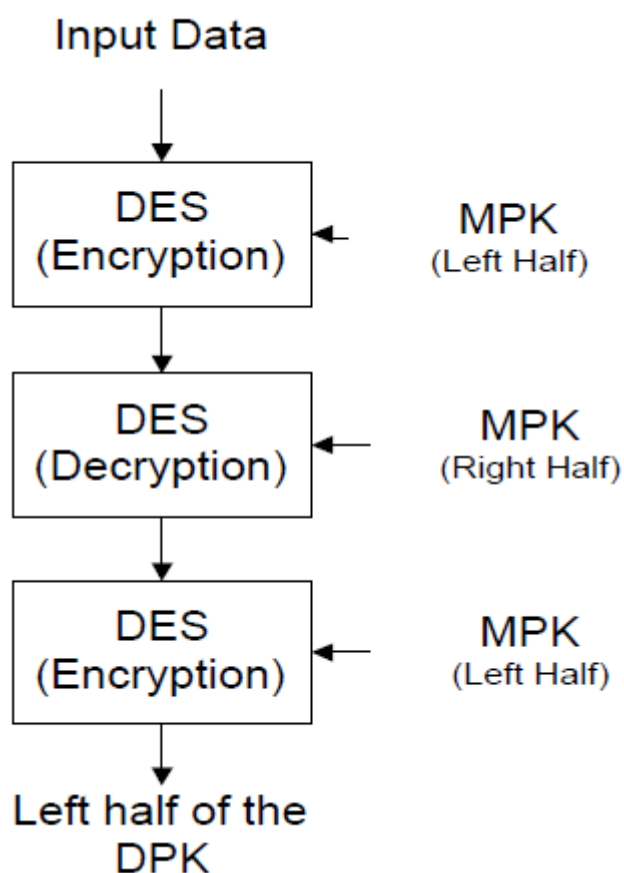


图3 推导 DPK 左边部分

6.2.2 DPK 右半部分的推导方法

推导双倍长DPK右半部分的方法：

- 将应用序列号的最右 16 个数字的求反作为输入数据；
- 将 MPK 作为加密密钥；
- 用 MPK 对输入数据进行 3DEA 运算。

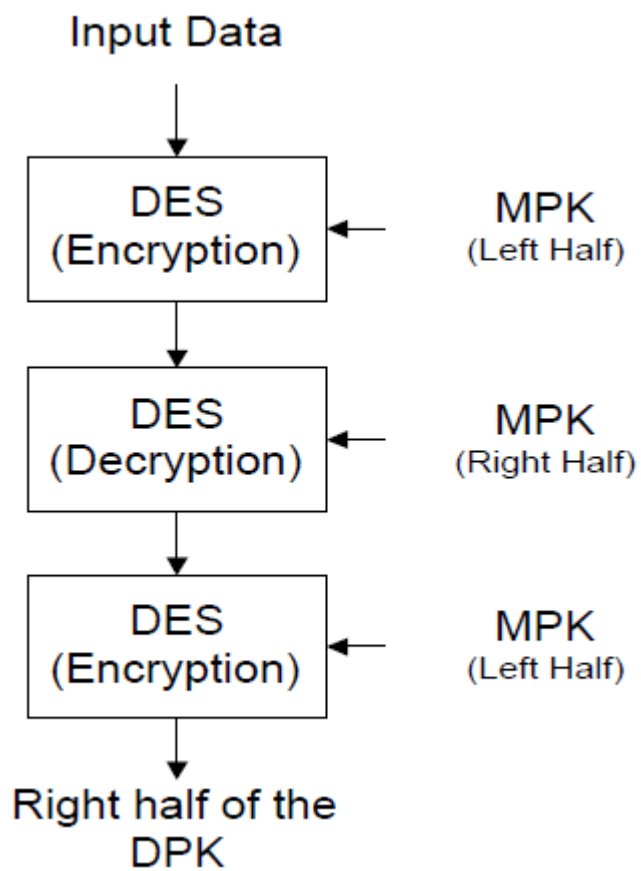


图4 推导 DPK 右边部分

6.2.3 过程密钥的产生

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。

图5描述了电子钱包进行消费交易时产生过程密钥的机制。这方法也用于不同交易类型的过程密钥的产生，但输入的数据取决于不同的交易类型。

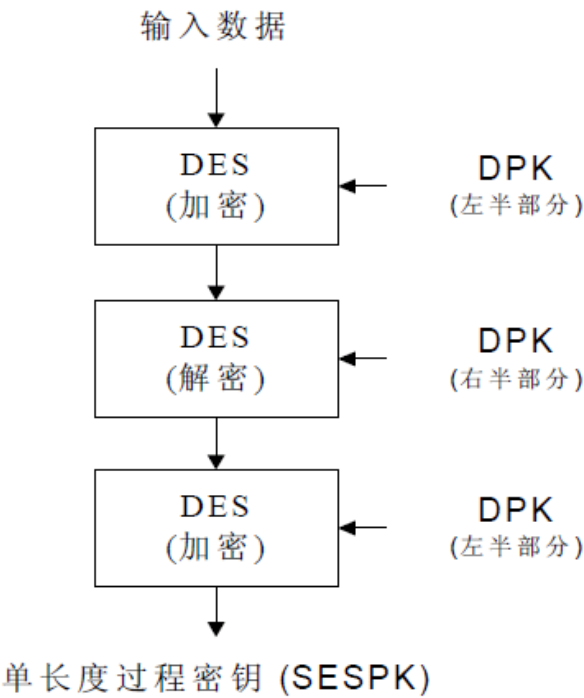


图5 过程密钥的产生

6.2.4 基于单长度 DEA 密钥的过程密钥

第一步：卡片/发卡机构决定是使用MAC DEA密钥A还是数据加密DEA密钥A来进行所选择的算法处理。（以后统称为“KeyA”）

第二步：用KeyA与预先决定的变量（如当前的交易序号）作异或运算产生过程密钥A。在作异或运算前，数据（如交易序号）如果少于8个字节，则在其右边用十六进制数字’ 0’ 填满。

6.2.5 基于双长度 DEA 密钥的过程密钥

第一步：卡片/发卡机构决定是使用MAC DEA密钥A和B还是数据加密DEA密钥A和B来进行所选择的算法处理（以后统称为“KeyA” 和 “KeyB”）。

第二步：用KeyA与预先决定的变量（如当前的交易序号）作异或运算产生过程密钥A。在作异或运算前，数据（例如：交易序号）如果少于8个字节，则在其右边用十六进制数字’ 0’ 填满。

用KeyB与第二步中产生的过程密钥A所用数据的非作异或运算得到过程密钥B。非运算是以位为单位的，把值为’ 1’ 的位转换为’ 0’ ，将值为’ 0’ 的位转换为’ 1’ 。在作异或运算前，数据如果少于8个字节，则在其右边用十六进制数字’ 0’ 填满。

6.2.6 MAC/TAC 的计算

MAC/TAC的产生使用以下单倍长DEA算法：

第一步：将一个8个字节长的初始值（Initial Vector）设定为16进制的“0x 00 00 00 00 00 00 00 00”。

第二步：将所有的输入数据按指定顺序串联成一个数据块。

第三步：将串联成的数据块分割为8字节长的数据块组，标识为D1、D2、D3与D4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。

第四步：如果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16进制的“0x 80 00 00 00 00 00 00”。如果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值为16进制“0x80”的字节。如果填补之后的数据块长度等于8字节，则跳至第五步。如果填补之后的数据块长度仍小于8字节，则在数据块后填补16进制“0x0”的字节至数据块长度为8字节。

第五步：MAC的产生是通过上述方法产生的数据块组，由过程密钥进行加密运算，过程密钥的产生方法见图5。TAC的产生是通过上述方法产生的数据块组，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或TAC的算法见图6描述。

第六步：最终值的左4字节为MAC或TAC。

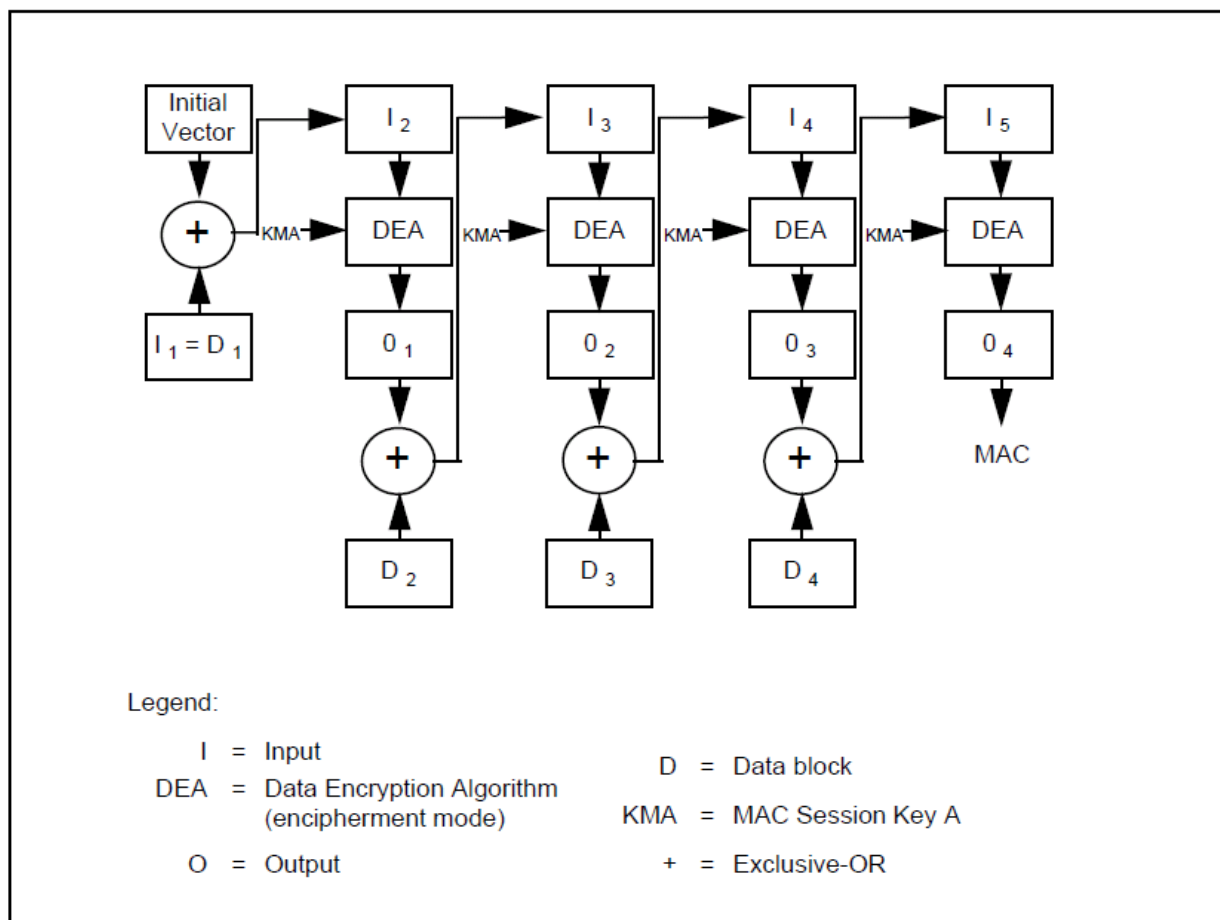


图6 MAC 和 TAC 的单倍长 DEA 密钥算法

7 电子现金动态数据认证

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能确认卡片上的发卡机构应用数据自卡片个性化后没有被非法篡改。DDA还能确认卡片的真实性，防止卡片的非法复制。DDA可以是标准动态数据认证或复合动态数据认证/应用密文生成（CDA）。AIP指明了IC卡支持的动态数据认证方法。

用于动态数据认证的记录必须是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：

——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，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；

- 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 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；
- 如果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，终端必须设置 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。

7.1 国际算法密钥和证书

终端通过采用公钥算法验证 IC 卡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。

如果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），卡片必须支持多 IC 卡公钥证书（或静态数据签名），如果被签名的静态应用数据在卡片发出后可能会被修改，卡片必须支持 IC 卡公钥证书（或静态数据签名）的更新。

7.1.1 认证中心

动态数据认证需要一个认证中心（CA），认证中心拥有高级别安全性的加密设备并用来签发发卡机构公钥证书。每一台符合本规范的终端都应为一个它能识别的应用保存相应的认证中心公钥。

7.1.2 公私钥对

认证中心和发卡机构必须使用 15.2 条中指定的非对称算法产生认证中心公私钥对，发卡机构公私钥对以及 IC 卡公私钥对。在本章中对动态数据认证过程及相关数据元的描述以 RSA 算法为例。

7.1.2.1 认证中心公私钥对

认证中心最多会产生 6 个公私钥对，每个公私钥对都将分配一个唯一的认证中心公钥索引。认证中心公钥及其索引由收单机构加载到终端，认证中心私钥由认证中心保管并保证其私密性和安全性。

终端必须有足够空间存放认证中心公钥及其对应的注册的应用提供商标识（RID）和认证中心公钥索引。终端通过 RID 和认证中心公钥索引定位认证中心公钥。

认证中心公钥模长必须在 15.2.1 条中所定义的范围内，认证中心公钥指数必须等于 3 或 $2^{16}+1$ 。

7.1.2.2 发卡机构公私钥对

发卡机构产生发卡机构公私钥对，并从认证中心获取发卡机构公钥证书。发卡机构将其公钥发送给认证中心，认证中心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认证中心私钥对其进行签名。

IC 卡必须包含发卡机构公钥证书及其用来验证发卡机构证书的认证中心公钥索引，发卡机构私钥由发卡机构保管并保证其私密性和安全性。

发卡机构公钥模长必须小于等于认证中心公钥最大模长，发卡机构公钥模长必须在 15.2.1 条中所定义的范围内。发卡机构公钥指数必须等于 3 或 $2^{16}+1$ 。

终端通过注册的应用提供商标识（RID）和认证中心公钥索引定位认证中心公钥，并用认证中心公钥从发卡机构证书恢复发卡机构公钥，然后用发卡机构公钥恢复并验证卡片上的发卡机构应用数据。

7.1.2.3 IC 卡公私钥对

支持 DDA 还要求发卡机构为每张 IC 卡产生 IC 卡公私钥对，IC 卡私钥存放在 IC 卡中的安全存储区域，IC 卡公钥由发卡机构私钥签名，产生 IC 卡公钥证书并存放在卡片中。

IC卡公钥模长必须小于等于发卡机构公钥模长，IC卡公钥模长必须在15.2.1条中所定义的范围。IC卡公钥指数必须等于3或 $2^{16}+1$ 。

终端通过注册的应用提供商标识（RID）和认证中心公钥索引定位认证中心公钥，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IC卡公钥证书恢复IC卡公钥，并用IC卡公钥验证卡片的动态签名数据。

IC卡公钥对还可被用于脱机密文PIN验证，本规范对脱机密文PIN不作要求。

7.2 国际算法动态数据认证（DDA）

DDA的目的是确认存放在IC卡中和由IC卡生成的关键数据以及从终端收到的数据的合法性。DDA除了执行同SDA类似的静态数据认证过程，确保IC卡中的发卡机构数据在个人化以后没有被非法篡改，还能防止任何对这样的卡片进行伪造的可能性。

动态数据认证有以下可选的两种方式：

- 标准的动态数据认证，这种方式在卡片行为分析前执行。在这种方式下，IC卡根据由IC卡动态数据所标识的存放在IC卡中的或由IC卡生成的数据以及由动态数据认证数据对象列表所标识的从终端收到的数据生成一个数字签名；
- 复合动态数据认证/应用密文生成，这种方式在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，IC卡根据由IC卡动态数据所标识的存放在IC卡中的或由IC卡生成的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GENERATE AC命令是CDOL1，对第二条GENERATE AC命令是CDOL2）标识的由终端生成的不可预知数AIP指明IC卡支持的选项。

支持动态数据认证的IC卡必须包含下列数据元：

- 认证中心公钥索引：该单字节数据元包含一个二进制数字，指明终端应使用其保存的相应的认证中心公钥中的哪一个来验证IC卡；
- 发卡机构公钥证书：该变长数据元由相应的认证中心提供给发卡机构。终端验证这个数据元时，按7.2.3条描述的过程认证发卡机构公钥和其它的数据；
- IC卡公钥证书：该变长数据元由发卡机构提供给IC卡。终端验证这个数据元时，按7.2.4条描述的过程认证IC卡公钥和其它的数据；
- 发卡机构公钥的余项：一个变长数据元。7.2.1条有进一步的解释；
- 发卡机构公钥指数：一个由发卡机构提供的变长数据元。7.2.1条有进一步的解释；
- IC卡公钥的余项：一个变长数据元。7.2.1条有进一步的解释；
- IC卡公钥指数：一个由发卡机构提供的变长数据元。7.2.1条有进一步的解释；
- IC卡私钥：一个存放在IC卡内部的变长数据元，用来按7.2.5条描述的过程生成签名的动态应用数据。

支持动态数据认证的IC卡必须生成下列数据元：

- 签名的动态应用数据：一个由IC卡使用同IC卡公钥证书所认证的IC卡公钥相对应的IC卡私钥生成的变长数据元。它是一个数字签名，包含了7.2.5条描述的存放在IC卡中的或由IC卡生成的以及终端中的关键数据元。

为了支持动态数据认证，每一台终端必须能够为每个注册的应用提供商标识存储6个认证中心公钥，而且必须使同密钥相关的密钥信息能够同每一个密钥相关联（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12.3条）。在给定了RID和IC卡提供的认证中心公钥索引的情况下，终端必须能够定位这样的公钥以及和公钥相关的信息。

动态数据认证必须使用一种在14.3.1条和15.2.1条中指明的可逆的算法。7.2.1条包含了对动态数据认证过程中涉及到的密钥和证书的概述，7.2.2条到7.2.4条详细说明了认证过程中的起始步骤，即：

- 由终端恢复认证中心公钥;
- 由终端恢复发卡机构公钥;
- 由终端恢复 IC 卡公钥。

最后，7.2.5条详细说明了两种情况下动态签名的生成和验证过程。

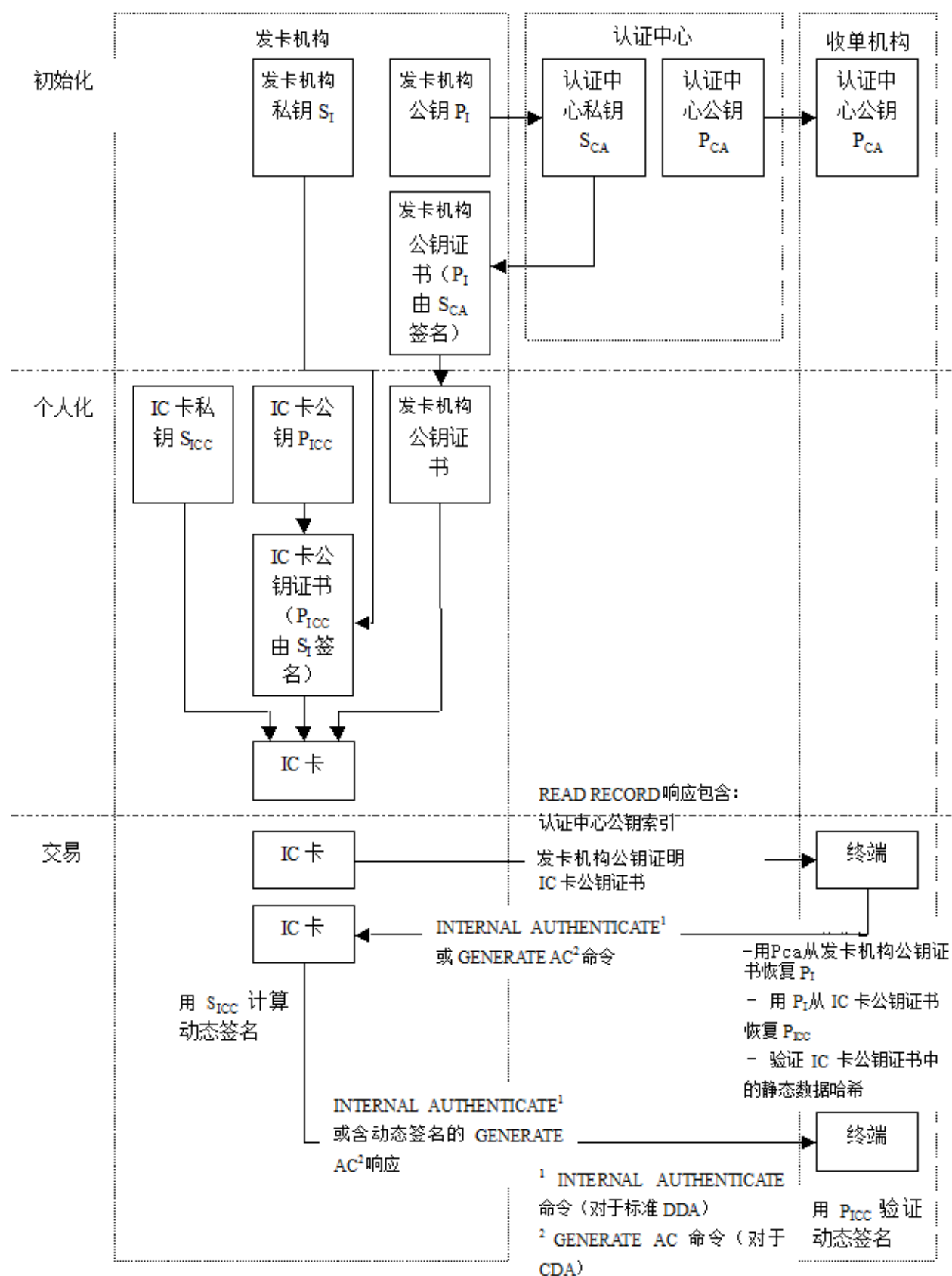


图7 DDA 证书和公钥体系结构

7.2.1 密钥和证书

为了支持动态数据认证，一张IC卡必须拥有它自己的唯一的公私钥对，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。IC卡公钥必须存放在IC卡上的公钥证书中。

动态数据认证采用了一个三层的公钥证书方案。每一个IC卡公钥由它的发卡机构认证，而认证中心认证发卡机构公钥。这表明为了验证IC卡的签名，终端需要先通过验证两个证书来恢复和验证IC卡公钥，然后用这个公钥来验证IC卡的动态签名。

按14.3.1条中指明的签名方案分别将认证中心私钥 S_{CA} 应用到表3中指定的数据以及将发卡机构私钥 S_I 应用到表4中指定的数据，以分别获得发卡机构公钥证书和IC卡公钥证书。

认证中心的公钥有一个 N_{CA} 个字节的公钥模。认证中心公钥指数必须等于3或 $2^{16}+1$ 。

发卡机构的公钥有一个为 N_I 个字节 ($N_I \leq N_{CA}$) 的发卡机构公钥模。如果 $N_I > (N_{CA}-36)$ ，那么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）。发卡机构公钥指数必须等于3或 $2^{16}+1$ 。

IC卡的公钥有一个为 N_{IC} 个字节 ($N_{IC} \leq N_I \leq N_{CA}$) 的IC卡公钥模。如果 $N_{IC} > (N_I-42)$ ，那么IC卡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（IC卡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（IC卡公钥余项）。IC卡公钥指数必须等于3或 $2^{16}+1$ 。

如果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），卡片必须支持多IC卡公钥证书，如果被签名的静态应用数据在卡片发出后可能会被修改，卡片必须支持IC卡公钥证书的更新。

为了完成动态数据认证，终端必须首先恢复和验证IC卡公钥（这一步叫做IC卡公钥认证）。IC卡公钥认证需要的所有信息在表5中详细说明，并存放在IC卡中。除了RID可以从AID中获得外，其它信息可以通过读取记录（READ RECORD）命令得到。如果缺少这些数据中的任意一项，那么动态数据认证失败。

表3 由认证中心签名的发卡机构公钥数据（即哈希算法的输入）

字段名	长度	描述	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3-8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	由认证中心分配给这张证书的二进制数	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	$N_{CA} - 36$	如果 $N_I \leq N_{CA} - 36$ ，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。 如果 $N_I > N_{CA} - 36$ ，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

表4 由发卡机构签名的 IC 卡公钥数据（即哈希算法的输入）

字段名	长度	描述	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
IC卡公钥算法标识	1	标识使用在IC卡公钥上的数字签名算法	b
IC卡公钥长度	1	标识IC卡公钥的模的字节长度	b
IC卡公钥指数长度	1	标识IC卡公钥指数的字节长度	b
IC卡公钥或IC卡公钥的最左边字节	$N_I - 42$	如果 $N_{IC} \leq N_I - 42$ ，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 IC 卡公钥。 如果 $N_{IC} > N_I - 42$ ，这个字段包含了IC卡公钥最高位的 $N_I - 42$ 个字节	b
IC卡公钥的余项	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IC卡公钥最低位的 $N_{IC} - N_I + 42$ 个字节	b
IC卡公钥指数	1或3	IC卡公钥指数等于3或 $2^{16} + 1$	b
需认证的静态数据	变长	需认证的静态数据：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

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

表5 动态认证中的公钥认证所需的数据对象

标签	长度	值	格式
—	5	注册的应用提供商标识	b
‘8F’	1	认证中心公钥索引	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（如果存在）	b
‘9F32’	1或3	发卡机构公钥指数	b
‘9F46’	N_I	IC卡公钥证书	b
‘9F48’	$N_{IC} - N_I + 42$	IC卡公钥的余项（如果存在）	b
‘9F47’	1或3	IC卡公钥指数	b
—	变长	需认证的静态数据：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。	—

7.2.2 认证中心公钥的获取

终端读取认证中心公钥索引。使用这个索引和RID，终端能够确认并取得存放在终端的认证中心公钥的模，指数和与密钥相关的信息，以及将使用的相应算法。如果终端没有存储与这个索引及RID相关联的密钥，那么动态数据认证失败。

7.2.3 发卡机构公钥的获取

- 1) 如果发卡机构公钥证书的长度不同于在前面的章条中获得的认证中心公钥模长度，那么动态数据认证失败。
- 2) 为了获得在表 6 中指定的恢复数据，使用认证中心公钥和相应的算法按照 14.3.1 条中指定的恢复函数恢复发卡机构公钥证书。如果恢复数据的结尾不等于“BC”，那么动态数据认证失败。

表6 从发卡机构公钥证书恢复数据的格式

字段名	长度	描述	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3-8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	由认证中心分配给这张证书的唯一二进制数	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	$N_{CA}-36$	如果 $N_I \leq N_{CA} - 36$ ，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。 如果 $N_I > N_{CA} - 36$ ，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

- 3) 检查恢复数据头。如果它不是“6A”，那么动态数据认证失败。
- 4) 检查证书格式。如果它不是“02”，那么动态数据认证失败。
- 5) 将表 6 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（如果有），最后是发卡机构公钥指数。
- 6) 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。
- 7) 把上一步计算得到的哈希结果和恢复出的哈希结果相比较。如果它们不一样，那么动态数据认证失败。
- 8) 检验发卡机构识别号是否匹配主账号最左面的 3-8 个数字（允许发卡机构识别号可能在其后填充的“F”）。如果不匹配，那么动态数据认证失败。
- 9) 确认证书失效日期中指定月的最后日期等于或迟于今天的日期。如果证书失效日期在今天的日期之前，那么证书已过期，动态数据认证失败。
- 10) 检验连接起来的 RID、认证中心公钥索引、证书序列号是否有效。如果无效，那么动态数据认证失败。
- 11) 如果发卡机构公钥算法标识无法识别，那么动态数据认证失败。

- 12) 如果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（如果有）以得到发卡机构公钥模，从而继续下一步取得 IC 卡公钥。

7.2.4 IC 卡公钥的获取

- 1) 如果 IC 卡公钥证书的长度不同于在前面的章条中获得的发卡机构公钥模长度，那么动态数据认证失败。
- 2) 为了获得在表 7 中指定的恢复数据，使用发卡机构公钥和相应的算法将 14.3.1 条中指定的恢复函数应用到 IC 卡公钥证书上。如果恢复数据的结尾不等于“BC”，那么动态数据认证失败。

表7 从 IC 卡公钥证书恢复数据的格式

字段名	长度	描述	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
IC卡公钥算法标识	1	标识使用在IC卡公钥上的数字签名算法	b
IC卡公钥长度	1	标识IC卡公钥的模的字节长度	b
IC卡公钥指数长度	1	标识IC卡公钥指数的字节长度	b
IC卡公钥或IC卡公钥的最左边字节	$N_I - 42$	如果 $N_{IC} \leq N_I - 42$ ，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 IC 卡公钥。 如果 $N_{IC} > N_I - 42$ ，这个字段包含了IC卡公钥最高位的 $N_I - 42$ 个字节	b
哈希结果	20	IC卡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

- 3) 检查恢复数据头。如果它不是“6A”，那么动态数据认证失败。
- 4) 检查证书格式。如果它不是“04”，那么动态数据认证失败。
- 5) 将表 7 中的第 2 个到第 10 个数据元（即从证书格式直到 IC 卡公钥或 IC 卡公钥的最左边字节）从左到右连接，再把 IC 卡公钥的余项（如果有）和 IC 卡公钥指数加在后面，最后是表 4 中指定的需认证的静态数据。如果静态数据认证标签列表存在，并且其包含非“82”的标签，那么动态数据认证失败。
- 6) 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。
- 7) 把上一步计算得到的哈希结果和恢复出的哈希结果相比较。如果它们不一样，那么动态数据认证失败。
- 8) 比较恢复得到的主账号和从 IC 卡读出的应用主账号是否相同。如果不同，那么动态数据认证失败。
- 9) 检验证书失效日期中指定月的最后日期是否等于或迟于今天的日期。如果不是，那么动态数据认证失败。
- 10) 如果 IC 卡公钥算法标识无法识别，那么动态数据认证失败。

- 11) 如果以上所有的检验都通过, 连接 IC 卡公钥的最左边字节和 IC 卡公钥的余项 (如果有) 以得到发卡机构公钥模, 继续按下面章条的描述执行实际的动态数据认证。

7.2.5 标准动态数据认证

7.2.5.1 动态签名的生成

假定终端已成功地按上面讲述的过程取得了 IC 卡公钥。动态签名的生成按以下的步骤进行:

- 1) 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, 这些数据元按《城市公共交通 IC 卡卡片技术规范》的附录 D 中指明的规则连接在一起。
- 2) IC 卡可能包含 DDOL, 但终端应有一个缺省的, 由城市公共交通 IC 卡系统指定的 DDOL, 以防在 IC 卡没有提供 DDOL 的情况下使用。
- 3) DDOL 必须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)。
- 4) 如果下面的任一情况发生, 动态数据认证失败:
 - IC 卡和终端都不含有 DDOL;
 - IC 卡上的 DDOL 不包含不可预知数;
 - IC 卡上没有 DDOL 并且终端上缺省的 DDOL 不包含不可预知数。
- 5) IC 卡使用 IC 卡私钥和相应的算法并按 14.3.1 条对表 8 中指明的数据生成数字签名。这个结果叫做签名的动态应用数据。

表8 需签名的动态应用数据 (即哈希算法的输入)

字段名	长度	描述	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
IC卡动态数据长度	1	标识IC卡动态数据的字节长度 L_{DD}	b
IC卡动态数据	L_{DD}	由IC卡生成和/或存储在IC卡上的动态数据	-
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	-

IC卡动态数据的字节长度 L_{DD} 满足 $0 \leq L_{DD} \leq N_{IC} - 25$ 。IC卡动态数据的最左边的3-9个字节应该由一个字节长的IC卡动态数字长度后面跟随的2-8个IC卡动态数字的值 (标签 “9F4C”, 2-8个二进制字节) 组成。IC卡动态数字是由一个由IC卡生成的, 随时间而变的参数, (例如它可以是不可预知数或者IC卡每收到一个内部认证 (INTERNAL AUTHENTICATE) 命令就加一的计数器)。本规范建议使用ATC作为IC卡动态数字。

除了表5中指明的数据, 动态数据认证所需的数据对象在表9中详细说明。

表9 生成和检验动态签名所需要的其它数据对象

标签	长度	值	格式
‘9F4B’	N_{IC}	签名的动态应用数据	b
‘9F49’	变长	DDOL	b

7.2.5.2 动态签名的验证

- 1) 如果签名的动态应用数据的长度不同于 IC 卡公钥模的长度, 那么动态数据认证失败。

- 2) 为了获得在表 10 中指明的恢复数据, 使用 IC 卡公钥和相应的算法将 14.3.1 条中指明的恢复函数应用到签名的动态应用数据上。如果恢复数据的结尾不等于“BC”, 那么动态数据认证失败。

表10 从签名的动态应用数据恢复的数据格式

字段名	长度	描述	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 ¹	b
IC卡动态数据长度	1	标识IC卡动态数据的字节长度	b
IC卡动态数据	L_{DD}	由IC卡生成和/或存储在IC卡上的动态数据	-
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

- 3) 检查恢复数据头。如果它不是“6A”, 那么动态数据认证失败。
- 4) 检查签名数据格式。如果它不是“05”, 那么动态数据认证失败。
- 5) 将表 10 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。
- 6) 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。
- 7) 把上一步计算得到的哈希结果和恢复出的哈希结果相比较。如果它们不一样, 那么动态数据认证失败。

如果以上所有的步骤都成功, 那么动态数据认证成功。在表 10 中恢复得到的 IC 卡动态数据中所包含的 IC 卡动态数字应被存放在标签“9F4C”中。

7.3 国密算法密钥和证书

终端通过 SM2 公钥密码算法验证 IC 卡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, 该证书或签名可以被公钥验证。本部分认可的 SM2 为中国国家标准的椭圆曲线公钥密码算法, 其数字签名不具备消息恢复功能。

对于用 RSA 签名的证书, 其中的公钥及其相关信息在验证证书的过程中被恢复出来。而对于用 SM2 签名的证书数据、公钥及其相关信息以明文形式包含在其中, 后面附着一个数字签名。

7.3.1 认证中心

动态数据认证需要一个认证中心 (CA), 认证中心拥有高级别安全性的加密设备并用来签发发卡机构公钥证书。每一台符合本规范的终端都应为一个它能识别的应用保存相应的认证中心公钥。认证中心需支持 SM2 算法。

7.3.2 公私钥对

认证中心、发卡机构和 IC 卡使用 SM2 算法产生认证中心公私钥对和发卡机构公私钥对以及 IC 卡公私钥对。

7.3.2.1 认证中心公私钥对

认证中心产生SM2算法公私钥对，每个公私钥对都将分配一个唯一的认证中心公钥索引。认证中心公钥及其索引由收单行加载到终端，认证中心私钥由认证中心保管并保证其私密性和安全性。

终端必须有足够空间存放认证中心公钥及其对应的注册应用提供商标识（RID）和认证中心公钥索引。终端通过RID和认证中心公钥索引定位认证中心公钥。

认证中心SM2公钥算法采用中国国家密码管理局推荐曲线参数。

7.3.2.2 发卡机构公私钥对

支持SDA或DDA都需要发卡机构产生发卡机构公私钥对，并从认证中心获取发卡机构公钥证书。发卡机构将其公钥发送给认证中心，认证中心使用公钥有效期晚于发卡机构公钥有效期的认证中心私钥对其进行签名。签名算法将标识在发卡机构证书的“发卡机构公钥签名算法标识”字段。

IC卡必须包含发卡机构公钥证书及其用来验证发卡机构证书的认证中心公钥索引，发卡机构私钥由发卡机构保管并保证其私密性和安全性。

终端通过注册应用提供商标识（RID）和认证中心公钥索引定位认证中心公钥，并用认证中心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，需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。

7.3.2.3 IC卡公私钥对

支持DDA还要求发卡机构为每张IC卡产生IC卡公私钥对，或者由IC卡自己产生IC卡公私钥对，IC卡私钥存放在IC卡中的安全存储区域，IC卡公钥由发卡机构私钥签名，产生IC卡公钥证书并存放在卡片中。

终端通过注册应用标识（RID）和认证中心公钥索引定位认证中心公钥，并用认证中公钥验证发卡机构公钥证书，然后用发卡机构公钥验证IC卡公钥证书，并用IC卡公钥验证卡片的动态签名数据。验证签名数据时，需根据IC卡公钥证书的“发卡机构公钥签名算法标识”字段再次检查算法类型。

7.4 国密算法动态数据认证（DDA）

7.4.1 密钥和证书

为了支持动态数据认证，一张IC卡必须拥有它自己的公私钥对，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。IC卡公钥必须存放在IC卡上的公钥证书中。

动态数据认证采用了一个三层的公钥证书方案。每一个IC卡公钥由它的发卡机构认证，而认证中心认证发卡机构公钥。这表明为了验证IC卡的签名，终端需要先通过验证两个证书来验证IC卡公钥，然后用这个公钥来验证IC卡的动态签名。

用认证中心私钥SCA对表11中指定的数据计算SM2签名获得发卡机构公钥证书，用发卡机构私钥SI对表13中指定的数据计算SM2签名获得IC卡公钥证书。

如果卡片上的静态应用数据不是唯一的，卡片必须支持多IC卡公钥证书，如果被签名的静态应用数据在卡片发出后可能会被修改，卡片必须支持IC卡公钥证书的更新。

为了完成动态数据认证，终端必须首先验证IC卡公钥（这一步叫做IC卡公钥认证）。使用SM2算法的IC卡公钥认证需要的所有信息在表14中详细说明，并存放在IC卡中。除了RID可以从AID中获得外，其它信息可以通过读取记录（READ RECORD）命令得到。如果缺少这些数据中的任意一项，那么动态数据认证失败。

表11 由认证中心签名的发卡机构公钥数据（待签名数据）

字段名	长度	描述	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 3-8 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	由认证中心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

表12 发卡机构公钥证书的格式

字段名	长度	描述	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 3-8 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	由认证中心分配给这张证书的, 唯一的二进制数	b
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	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}	认证中心对表 11 的数据计算的 SM2 签名 $r s$	b

表13 由发卡机构签名的 IC 卡公钥数据 (待签名数据)

字段名	长度	描述	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
IC 卡公钥签名算法标识	1	标识 IC 卡公钥对应的数字签名算法	b
IC 卡公钥加密算法标识	1	标识 IC 卡公钥对应的加密算法, 保留项。	b
IC 卡公钥参数标识	1	用于标识椭圆曲线参数, 同时确定 N_{IC} 。	b
IC 卡公钥长度	1	标识 IC 卡公钥的字节长度	b
IC 卡公钥	N_{IC}	如果 IC 卡公钥算法标识对应于 SM2, 该字段为椭圆曲线上的一个点	b
需认证的静态数据	变长	需认证的静态数据: 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	b

		片中。	
--	--	-----	--

对表13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13除“需认证的静态数据”外的数据之后就形成了用SM2签名的IC卡公钥证书，证书的格式请参见表15。

认证过程的输入由被AFL标识的记录组成，其后跟有AIP[如果AIP被可选的静态数据认证标签列表（标签‘9F4A’）标识]。如果静态数据认证标签列表存在，它必须仅包含标识AIP用的标签‘82’。

表14 动态认证中的公钥认证所需的数据对象

标签	长度	值	格式
—	5	注册的应用提供商标识	b
‘8F’	1	认证中心公钥索引	b
‘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IC卡公钥证书数据，格式见表15	b
—	变长	需认证的静态数据：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。	—

7.4.2 认证中心公钥的获取

终端读取认证中心公钥索引。使用这个索引和RID，终端必须确认并取得存放在终端的认证中心公钥的相关信息。如果终端没有存储与这个索引及RID相关联的密钥，那么动态数据认证失败。

7.4.3 发卡机构公钥的获取

认证中心如采用SM2签名发卡机构公钥证书，终端获取的发卡机构证书数据如表11所示，包括被签名的明文数据及数字签名：发卡机构公钥以明文形式包含在发卡机构公钥证书中，用认证中心的公钥验证发卡机构公钥证书中的签名字段若是正确的，则直接从发卡机构公钥证书中提取公钥信息。

验证步骤如下：

- 1) 获取并解析如表 12 所示的发卡机构公钥证书数据。如果失败，那么静态数据认证失败。
- 2) 检查证书格式。如果它不是“12”，那么动态数据认证失败。
- 3) 检验发卡机构标识是否匹配主账号最左面的 3-8 个数字(允许发卡机构标识可能在其后补“F”)。如果不一致，那么动态数据认证失败。
- 4) 检验证书失效日期中指定月的最后日期是否等于或迟于今天的日期。如果证书失效日期在今天的日期之前，那么证书已过期，动态数据认证失败。
- 5) 检验连接起来的 RID、认证中心公钥索引、证书序列号是否有效。如果无效，那么动态数据认证失败。
- 6) 检查发卡机构公钥算法标识是否为“04”（SM2 算法），如果不是，那么动态数据认证失败。
- 7) 准备表 12 中前 9 个数据元（即表 1 数据）。
- 8) 使用认证中心公钥和相应的认证中心签名算法按照 14.4.5 条中指定的验证函数对表 12 的数字签名进行验证。如果验证签名失败，那么动态数据认证失败。
- 9) 如果以上所有的检验都通过，继续下一步取得 IC 卡公钥。

7.4.4 IC 卡公钥的获取

发卡机构采用SM2签名IC卡公钥证书，终端获取的IC卡公钥证书数据如表15所示，包括被签名的明文数据及数字签名。IC卡公钥以明文形式包含在IC卡公钥证书中，用发卡机构的公钥验证IC卡公钥证书中的签名字段。如验证通过，则直接从IC卡公钥证书中提取公钥信息。

表15 发卡机构使用 SM2 签名的 IC 卡公钥证书的格式

字段名	长度	描述	格式
证书格式	1	十六进制，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
IC卡公钥签名算法标识	1	标识IC卡公钥对应的数字签名算法	b
IC卡公钥加密算法标识	1	标识IC卡公钥对应的加密算法，保留项。	b
IC卡公钥参数标识	1	用于标识椭圆曲线参数，同时确定 N_{IC} 。	b
IC卡公钥长度	1	标识IC卡公钥的字节长度	b
IC卡公钥	N_{IC}	如果IC卡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6数据计算的SM2签名 $r s$	b

验证步骤如下：

- 1) 获取并解析如表15所示的经过IC卡公钥证书数据。如果失败，则静态数据认证失败。
- 2) 检查证书格式。如果它不是“14”，那么动态数据认证失败。
- 3) 比较证书中的主账号和从IC卡读出的应用主账号是否相同。如果不同，那么动态数据认证失败。
- 4) 检验证书失效日期中指定月的最后日期是否等于或迟于今天的日期。如果证书失效日期在今天的日期之前，那么证书已过期，动态数据认证失败。
- 5) 准备表15中的前9个数据元以及静态数据（用于验证签名）。如果静态数据认证标签列表存在，并且其包含非“82”的标签，那么动态数据认证失败。检查发卡机构公钥算法标识是否为“04”（SM2算法），如果不是，那么动态数据认证失败。
- 6) 使用发卡机构公钥和相应的发卡机构签名算法将14.4.5条中指明的验证函数对表8的数字签名进行验证。如果验证签名失败，那么动态数据认证失败。

如果以上所有的检验都通过，继续按下面章节的描述执行实际的动态数据认证。

7.4.5 标准动态数据认证

7.4.5.1 动态签名的生成

IC卡使用SM2算法生成动态签名，动态签名的生成按以下的步骤进行：

- 1) 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《城市公共交通IC卡卡片技术规范》的附录D中指明的规则连接在一起。
- 2) IC卡使用IC卡私钥对表16中指明的数据计算SM2签名，得到如表18的格式的SM2签名动态应用数据。

表16 需签名的动态应用数据（待签名数据）

字段名	长度	描述	格式
签名的数据格式	1	值为‘15’表示用SM2签名	b

IC卡动态数据长度	1	标识IC卡动态数据的字节长度 L_{DD}	b
IC卡动态数据	L_{DD}	由IC卡生成和/或存储在IC卡上的动态数据	-
终端动态数据	变长	由DDOL指定的数据元连接而成	-

IC卡动态数据的最左边的3-9个字节应该由一个字节长的IC卡动态数字长度后面跟随的2-8个IC卡动态数字的值（标签‘9F4C’，2-8个二进制字节）组成。IC卡动态数字是由一个由IC卡生成的，随时时间而变的参数，（例如它可以是不可预知数或者IC卡每收到一个内部认证（INTERNAL AUTHENTICATE）命令就加一的计数器）。在本规范中建议使用ATC作为IC卡动态数字。

除了表16中指明的数据，动态数据认证所需的数据对象在表17中详细说明。

表17 生成和检验动态签名所需要的其它数据对象

标签	长度	值	格式
‘9F4B’	$N_{IC}+L_{DD}+2$	SM2签名动态应用数据，格式见表11	b
‘9F49’	变长	DDOL	b

7.4.5.2 动态签名的验证

IC卡采用SM2签名动态应用数据，终端获取的签名动态应用数据的格式如表18所示，包括被签名的明文数据及数字签名。终端使用IC卡的公钥验证动态应用数据的签名，如果动态数据认证成功，表18中的IC卡动态数据中所包含的IC卡动态数字应被存放在标签‘9F4C’中。

表18 IC卡使用SM2签名的动态应用数据的格式

字段名	长度	描述	格式
签名的数据格式	1	十六进制，值为‘15’	b
IC卡动态数据长度	1	标识IC卡动态数据的字节长度 L_{DD}	b
IC卡动态数据	L_{DD}	由IC卡生成和/或存储在IC卡上的动态数据	-
数字签名	N_{IC}	IC卡对表16中数据计算的SM2签名 $r s$	b

验证步骤如下：

- 1) 获取并解析如表 18 所示的经过发卡机构签名的动态数据。如果失败，则静态数据认证失败。
- 2) 检查签名数据格式。如果它不是“15”，那么动态数据认证失败。
- 3) 准备表 18 中的前 3 个数据元（即从签名数据格式直到 IC 卡动态数据）及 DDOL 中指定的数据元。（即表 10 数据用于验证签名）
- 4) 使用 IC 卡公钥和相应的 IC 卡签名算法将 14.4.5 条中指定的验证函数对表 18 的数字签名进行验证。如果验证签名失败，那么动态数据认证失败。
- 5) 如果以上所有的步骤都成功，那么动态数据认证成功。在终端获取数据（表 18）中的 IC 卡动态数据中所包含的 IC 卡动态数字应被存放在标签“9F4C”中。

8 电子现金应用密文和发卡机构认证

本章描述了IC卡生成应用密文（TC、ARQC或AAC），以及发卡机构生成授权响应密文（ARPC）并由IC卡校验的方法。对于这些密文在一个交易中的任务的更详细信息，《城市公共交通IC卡卡片技术规范》。

8.1 应用密文产生

8.1.1 数据源选择

一个应用密文是由基于以下数据生成的报文鉴别码组成的：

——引用 IC 卡的 DOL 并通过生成应用密文（GENERATE AC）命令或其它命令从终端传输到 IC 卡的数据；

——IC 卡内部访问的数据。

具体需包含在应用密文生成中的数据源的选择，见《城市公共交通IC卡卡片技术规范》附录F，建议的最小数据集由表19详细说明。

表19 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
其它金额（数字）	终端
终端国家代码	终端
终端验证结果	终端
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	IC卡
应用交易计数器	IC卡

可选的应用密文生成数据源见表20。

表20 可选的应用密文生成数据源

值	来源
卡片验证结果	IC卡

8.1.2 国际算法应用密文算法

应用密文生成的方法是以一个唯一的16字节的IC卡应用密文（AC）子密钥MK_{AC}以及按7.1.1条的描述选择的数据作为输入，然后按以下的两步计算8字节的应用密文：

- 1) 第一步从 IC 卡应用密文（AC）子密钥 MK_{AC} 和两字节的 IC 卡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 14.1.3 条中指明的过程密钥分散函数。
- 2) 第二步使用上一步分散得到的 16 字节的应用密文过程密钥并将 14.1.2 条中指明的 MAC 算法应用到经选择的数据来生成 8 字节的应用密文。

8.1.3 国密算法应用密文算法

以一个唯一的16字节IC卡应用密文（AC）子密钥MK_{AC}，和7.1.1节描述的数据源作为输入，按以下两步计算8字节的应用密文：

- 1) 以 IC 卡应用密文（AC）子密钥 MK_{AC} 和两字节的 IC 卡应用交易计数器作为输入，使用 14.2.3 节描述的算法，生成 16 字节的应用密文过程密钥 SK_{AC}。
- 2) 使用上一步生成的 16 字节应用密文过程密钥 SK_{AC} 和“经选择的数据”作为输入，按照 14.2.2 节中指明的 MAC 算法计算得到应用密文（TC、ARQC 或 AAC）。

详细密文生成的步骤如下：

步骤1：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。如果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。

步骤2：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：

- 交易证书（TC）哈希结果（如果存在）；
- 生成应用密文命令中送进卡片的数据。不包括 TC 哈希结果；
- 卡片内部数据。

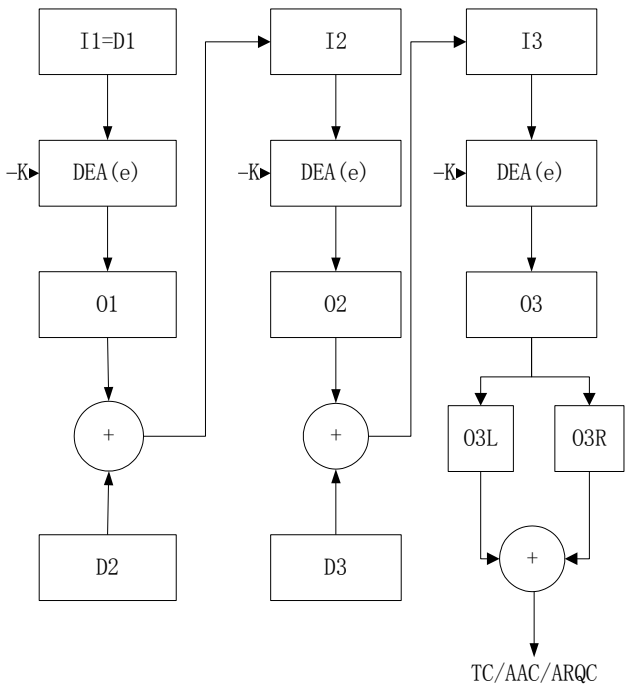
步骤3：将上述数据块分成16字节一组：D1、D2、D3……

步骤4：如果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；

如果最后一块数据块的长度小于16字节，后面补一个字节80，如果仍然不够16字节，补00直到16字节。

步骤5：如图8，使用过程密钥用对称密钥算法生成应用密文（过程密钥是由IC卡应用密文（AC）子密钥MKAC分散生成，具体生成方法在8.1.3中）。

步骤6：将上一步计算结果的左边8字节与右边8字节进行异或，得到8字节的密文。



说明：

- | | |
|----------------------|---------|
| I = 输入 | D = 数据块 |
| DEA(e)= 数据加密算法（加密模式） | K = 密钥 |
| O = 输出 | + = 异或 |

图8 TC/AAC/ARQC 的生成算法

8.2 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} （见8.1条）按照14.1条中指明的对称加密算法对8字节长的由IC卡按7.1条描述的方法生成的ARQC和2字节的授权响应码ARC进行加密：

- 1) 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：

$X := (ARC || '00' || '00' || '00' || '00' || '00' || '00')$ 。

- 2) 计算 $Y := ARQC \oplus X$ 。

- 3) 计算 ARPC

基于64位分组加密算法获得8字节的ARPC：

$ARPC := ALG(SK_{AC})[Y]$

基于128位分组加密算法获得16字节ARPC：

$ARPC := ALG(SK_{AC})[Y || '00' || '00' || '00' || '00' || '00' || '00' || '00' || '00']$

8.3 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} （见14.2.3节）按照15.1.2条中指明的对称加密算法对8.1.2条生成的8字节长的ARQC和2字节的授权响应码ARC进行加密：

- 1) 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：

$X := (ARC || '00' || '00' || '00' || '00' || '00' || '00' || '00')$ 。

- 2) 计算 $Y := ARQC \oplus X$ 。

- 3) 计算ARPC0：

将Y左对齐后面补8个字节00形成D；

$D := Y || '00' || '00' || '00' || '00' || '00' || '00' || '00' || '00' || '00'$

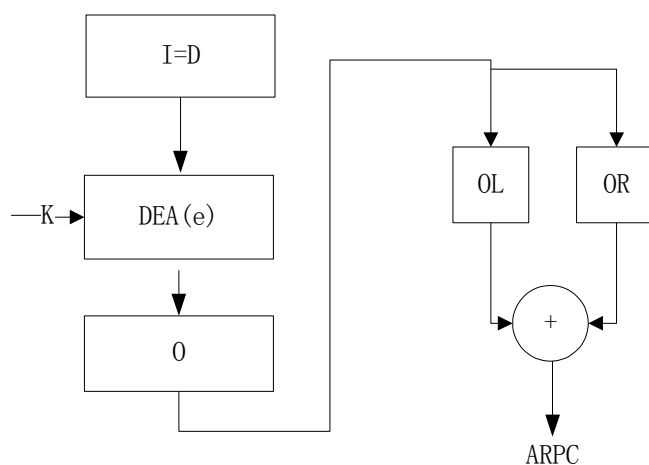
基于16字节分组加密算法获得16字节ARPC0：

$ARPC0 := ALG(SK_{AC})[D]$ ；

将ARPC0的左边8字节和右边8字节进行异或得到ARPC，即为

$ARPC := ARPC0_L \oplus ARPC0_R$

图9是ARPC的生成方法。



说明：

I = 输入

D = 数据块

DEA(e) = 数据加密算法（加密模式）

K = 密钥

0 = 输出

图9 生成 ARPC 的算法

8.4 密钥管理

应用密文和发卡机构认证的机制要求发卡机构管理唯一的发卡机构应用密文（AC）主密钥。IC卡应用密文（AC）子密钥的分散方法见14章。

9 电子现金行业信息的保护

9.1 密钥说明

涉及到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。

9.1.1 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行业开通密钥，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。

9.1.2 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行业信息，地区扩展应用管理密钥用于保护本地区的相关行业信息。

扩展应用管理密钥在地区使用时应存放于 SAM 卡中，相关安全机制参考本部分 14 章，指令参考《城市公共交通 IC 卡卡片技术规范》附录 A。

9.1.3 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，互联互通密钥应由全国运营机构统一生成和管理，地区扩展应用密钥可由发卡机构或相关行业方生成和管理。

互联互通密钥采用三级密钥管理体系，全国运营机构生成指定的互联互通主密钥，该密钥同时存放在行业相关终端的互联互通 SAM 卡中。各省交通厅的互联互通密钥由互联互通主密钥分散生成，各发卡机构的互联互通密钥由相应各省交通厅的省级互联互通密钥分散生成，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。

9.2 安全机制

终端使用 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 UPDATE CAPP DATA CACHE 命令更新应用文件数据，这两条指令都强制带有安全报文，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 的计算方法见第 10 章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 GET DATA 指令，或者通过发送 GPO 指令获取 ATC。

在 APPEND RECORD 指令中，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，即行业终端在更新应用数据时将应用有效标识置零。

10 安全报文

安全报文通过报文鉴别码（MAC）来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来保障数据的机密性。

10.1 报文格式

本规范使用的报文格式见《城市公共交通IC卡读写终端技术规范》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必须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数据是否需要加密传输，是否应该以加密的方式处理。

10.2 电子现金国际算法报文完整性及其验证

10.2.1 命令数据域

使用安全报文的命令的发送者以及当前被选择的应用必须知道包含在数据域中的数据元（包括MAC）及其相应的数据长度。MAC不是BER-TLV编码并且总是数据域中的最后一个数据元，并且它的长度总是4字节。

表21 以完整性和认证为目的的安全报文的命令数据域格式

值1	值2
命令数据（如果有）	MAC（4字节）

10.2.2 MAC 过程密钥分散

以完整性和认证为目的的安全报文的MAC生成的第一步包括从IC卡的唯一的16字节安全报文认证（MAC）子密钥和2字节ATC分散得到一个唯一的16字节安全报文鉴别（MAC）过程密钥。在14.1.3条中详细说明了一种分散的方法。

10.2.3 MAC 的计算

MAC是通过使用按照14.1.3条中描述的方法分散得到的MAC过程密钥并将14.1.2条中描述的机制应用在所要保护的报文上计算得到的。

要保护的报文必须按照城市公共交通IC卡系统的专有规范来构建。但总是包含了C-APDU（CLA INS P1 P2）的头部以及命令数据（如果存在）。

在本部分中MAC长度为4，在按上面描述的方法计算得到8个字节的結果后，取其中最左面的（最高）4字节来得到MAC。

10.3 电子现金国密算法报文完整性及其验证

10.3.1 MAC 过程密钥分散

以完整性和认证为目的的安全报文的MAC生成的第一步包括从IC卡的唯一的16字节安全报文鉴别（MAC）子密钥和2字节ATC分散得到一个唯一的16字节安全报文鉴别码（MAC）过程密钥。过程密钥分散方法参见14.2.3节。

10.3.2 MAC 的计算

MAC是通过使用按照10.3.1节中描述的方法分散得到的MAC过程密钥并将14.2.2节中描述的机制应用在所要保护的报文上计算得到的。

要保护的报文必须按照支付系统的专有规范来构建。但总是包含了C-APDU（CLA INS P1 P2）的头部以及命令数据（如果存在）。

在本部分中MAC长度为8，在按上面描述的方法计算得到16个字节的后果后，取其中最左面的（最高）8字节来得到MAC。

10.4 电子钱包报文完整性及其验证

MAC是使用命令的所有元素（包括命令头）产生的。一条命令的完整性，包括命令数据域（如果存在的话）中的数据元，通过安全报文传送得以保证。

10.4.1 MAC 的位置

MAC是命令数据域中最后一个数据元。

10.4.2 MAC 的长度

本部分中MAC的长度规定为4个字节。

10.4.3 MAC 密钥的产生

在安全信息处理过程中用到的MAC过程密钥是按照6.2.3中描述的过程密钥的产生过程产生的。MAC DEA密钥的原始密钥用于产生MAC过程密钥。

10.4.4 MAC 的计算

按照如下的方式使用单重或三重DEA加密方式产生MAC：

第一步：取8个字节的16进制数字‘0’作为初始变量。

第二步：按照顺序将以下数据串联在一起形成数据块：

——CLA、INS、P1、P2 和Lc；

——所有在《城市公共交通IC卡卡片技术规范》中定义的数据；

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

第三步：将该数据块分成8字节为单位的数据块，标号为D1、D2、D3和D4等。最后的数据块有可能是1-8个字节。

第四步：如果最后的数据块长度是8字节的话，则在其后加上16进制数字‘80 00 00 00 00 00 00 00’，转到第五步。

如果最后的数据块长度不足8字节，则在其后加上16进制数字‘80’，如果达到8字节长度，则转入第五步；否则在其后加入16进制数字‘0’直到长度达到8字节。

第五步：对这些数据块使用MAC过程密钥进行加密，过程密钥按照6.2.3描述的方式产生。如果安全报文传送支持单长度的MAC DEA密钥，则依照图10的方式使用MAC过程密钥来产生MAC（根据在第二步中产生的数据块长度的不同，有可能在计算中会多于或少于四步）。

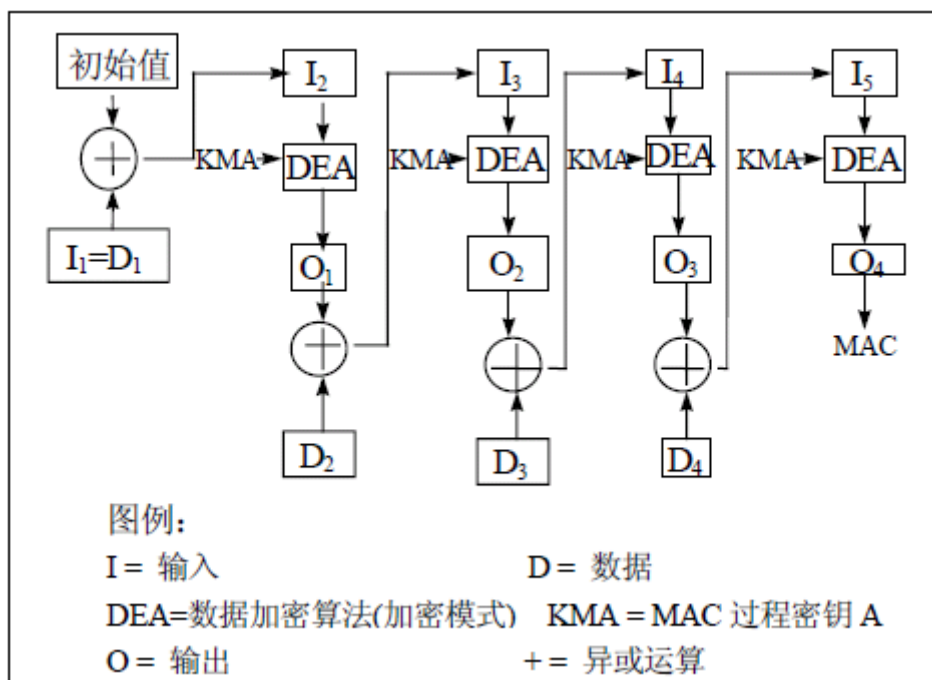


图10 单长度 DEA 密钥的 MAC 算法

如果安全报文传送的处理支持双长度MAC DEA密钥，则使用MAC过程密钥A和B（MAC的产生见图11），（根据第二步产生的数据块的长度，计算过程有可能多于或少于四步）。

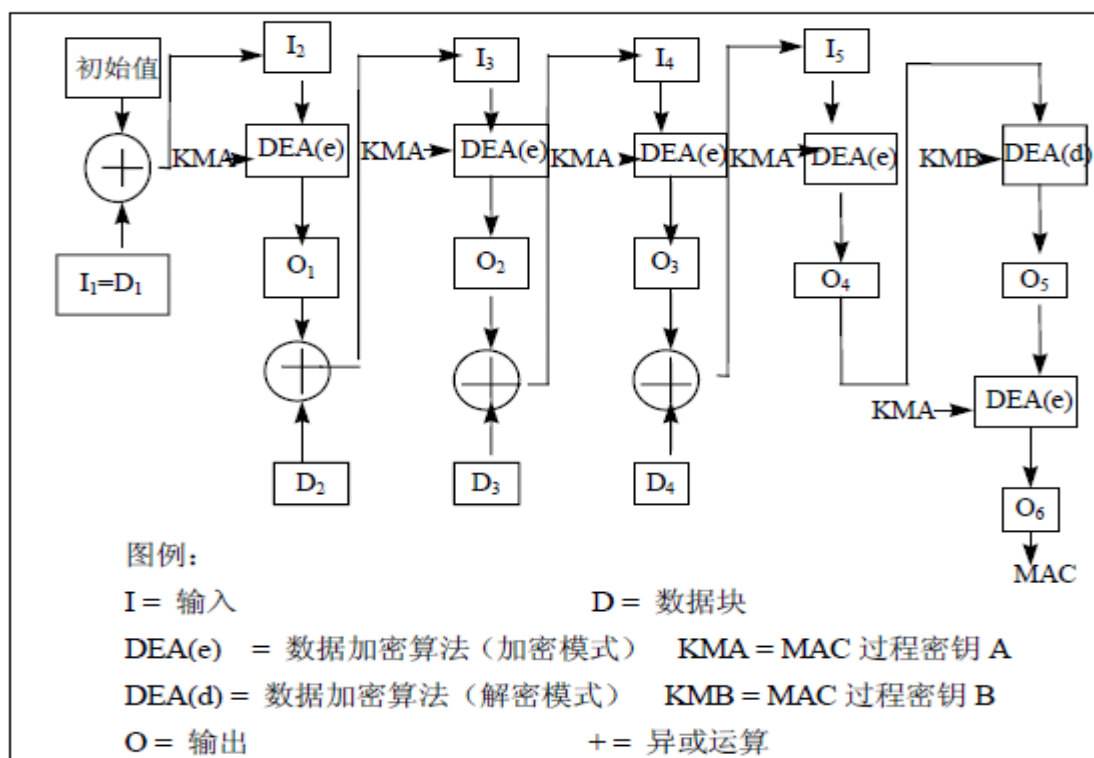


图11 双长度 DEA Key 的 MAC 算法

第六步：最终得到是从计算结果左侧取得的4字节长度的MAC。

10.5 电子现金国际算法报文私密性

10.5.1 命令数据域

在命令数据域中除了MAC以外，其它明文数据域都被加密。

表22 以私密性为目的的安全报文的命令数据域格式

值1	值2
密文（加密的数据）	MAC（如果存在）

10.5.2 加密过程密钥分散

以私密性为目的的安全报文的加/解密的第一步包括从IC卡的唯一的16字节安全报文加密子密钥和2字节ATC分散得到一个唯一的16字节加密过程密钥。在14.1.3条中详细说明了一种这样的方法。

10.5.3 加密解密

对明文/加密命令数据域的加/解密是通过使用按照14.1.3条中描述的方法分散得到的加密过程密钥并应用14.1.1条中描述的机制进行的。

10.6 电子现金国密算法报文私密性

10.6.1 加密过程密钥分散

以私密性为目的的安全报文的加/解密的第一步包括从IC卡的唯一的16字节安全报文加密子密钥和2字节ATC分散得到一个唯一的16字节加密过程密钥。在14.2.3节中详细说明了一种这样的方法。

10.6.2 加密解密

对明文/加密命令数据域的加/解密是通过使用按照10.6.1条中描述的方法分散得到的加密过程密钥并应用14.2.1条中描述的机制进行的。

10.7 电子钱包报文私密性

为保证命令中明文数据的保密性，可以将数据加密。所使用的数据加密技术，应被命令发送方和当前卡中被选择的应用所了解。

10.7.1 数据加密密钥的计算

在安全报文处理过程中用到的数据，加密过程密钥按照6.2.3中描述的方式产生。数据加密过程密钥的产生过程是从卡中的数据加密DEA密钥开始的。

10.7.2 被加密数据的结构

当命令中要求的明文数据需要加密时，它先要被格式化为以下形式的数据块：

- 明文数据的长度，不包括填充字符（LD）；
- 明文数据；
- 填充字符（根据10.7.3 的要求）。

然后整个数据块使用10.7.3中描述的数据加密技术进行加密。

10.7.3 数据加密计算

数据加密技术如下所述：

第一步：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。

第二步：将第一步中生成的数据块分解成8字节数据块，标号为D1、D2、D3和D4等等。最后一个数据块长度有可能不足8位。

第三步：如果最后（或唯一）的数据块长度等于8字节，转入第四步；如果不足8字节，在右边添加16进制数字'80'。如果长度已达8字节，转入第四步；否则，在其右边添加1字节16进制数字'0'直到长度达到8字节。

第四步：每一个数据块使用10.5.1中描述的数据加密方式加密。

如果采用单长度数据加密DEA密钥，数据块的加密见图12（使用数据加密过程密钥A进行加密）。

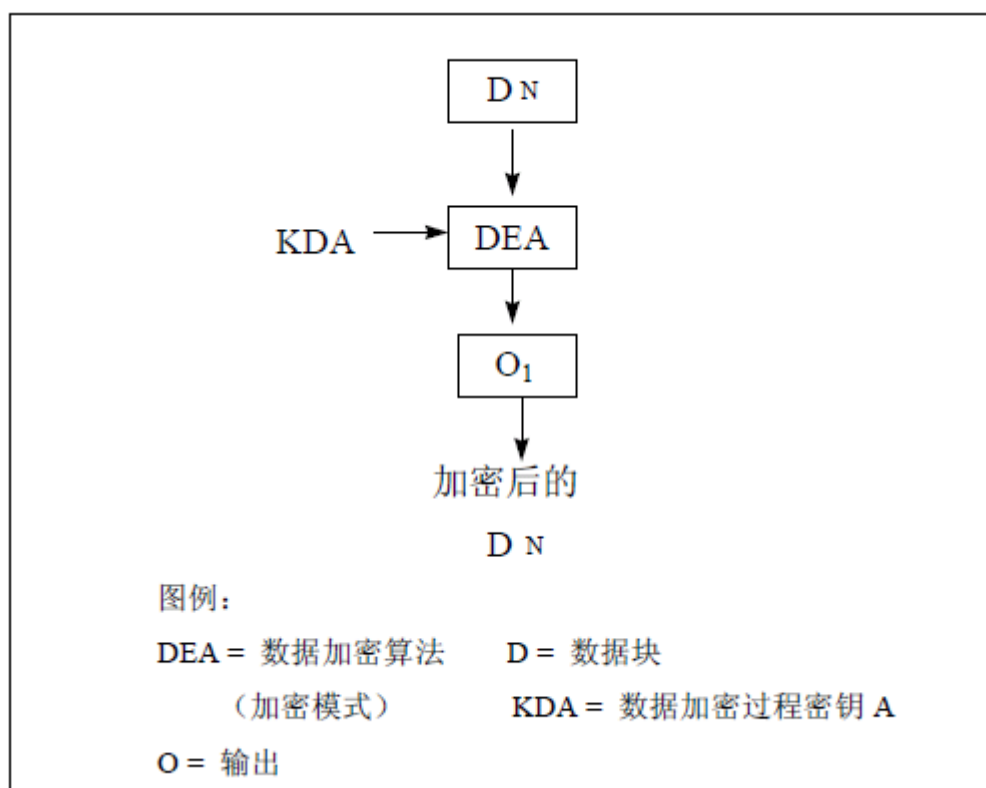


图12 单长度 DEA 密钥的数据加密

如果采用双长度数据加密DEA密钥，则数据块的加密见图13（使用数据加密过程密钥A和B来进行加密）。

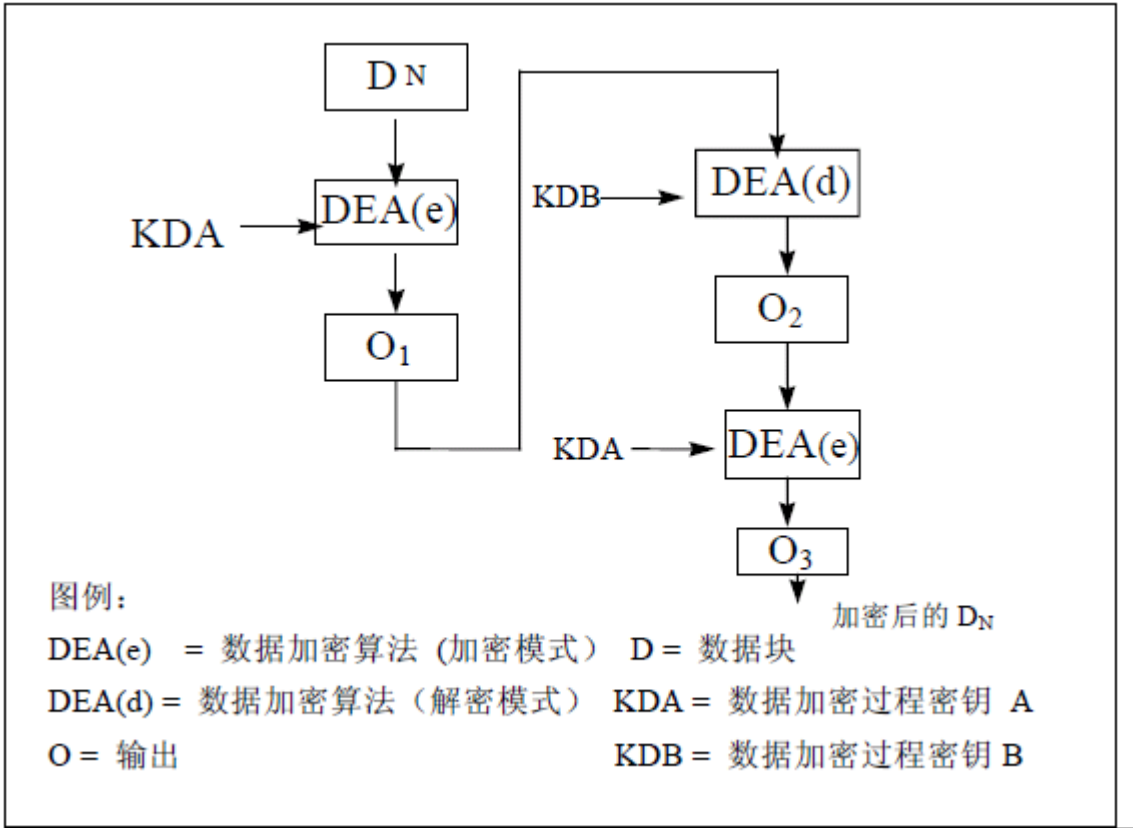


图13 使用双长度 DEA 密钥的数据加密

第五步：计算结束后，所有加密后的数据块依照原顺序连接在一起（加密后的D1、加密后的D2等）。并将结果数据块插入到命令数据域中。

10.7.4 数据解密计算

卡片接收到命令之后，需要将包含在命令中的加密数据进行解密。数据解密的技术如下：

第一步：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D1、D2、D3和D4等等。每个数据块使用如10.7.1所描述的方法产生的数据加密过程密钥进行解密。

如果采用单长度数据加密的DEA密钥，数据块解密见图14（使用数据加密过程密钥A进行解密）。

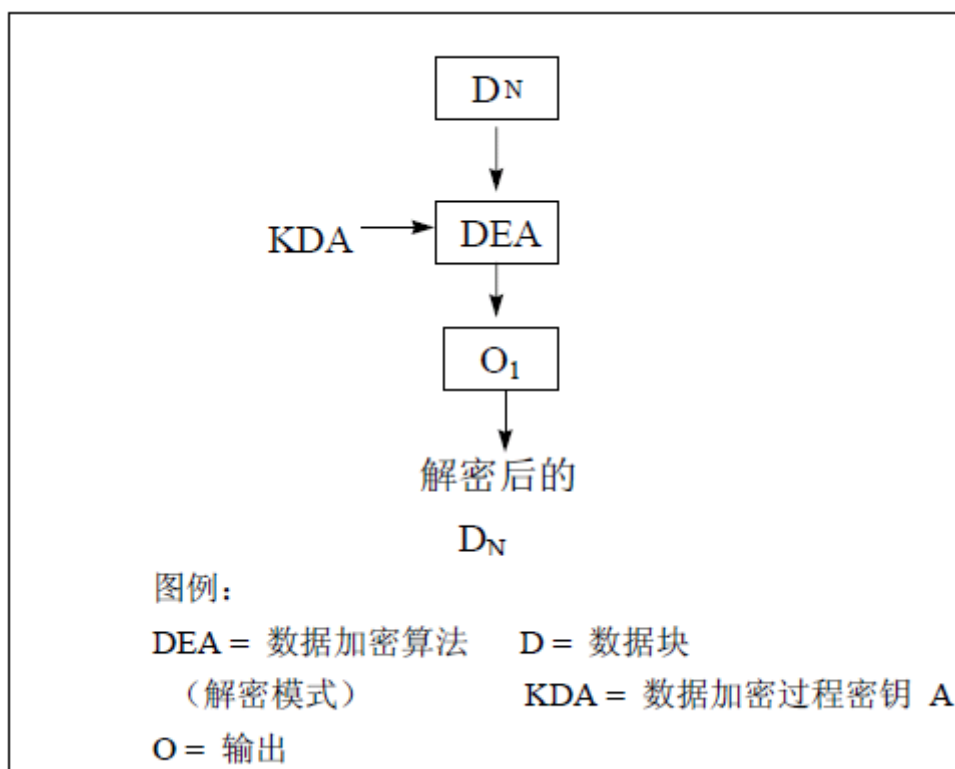


图14 使用单长度 DEA 密钥的数据解密

如果采用双长度数据加密的DEA密钥，则数据块的解密见图15（使用数据加密过程密钥A和B来进行解密）。

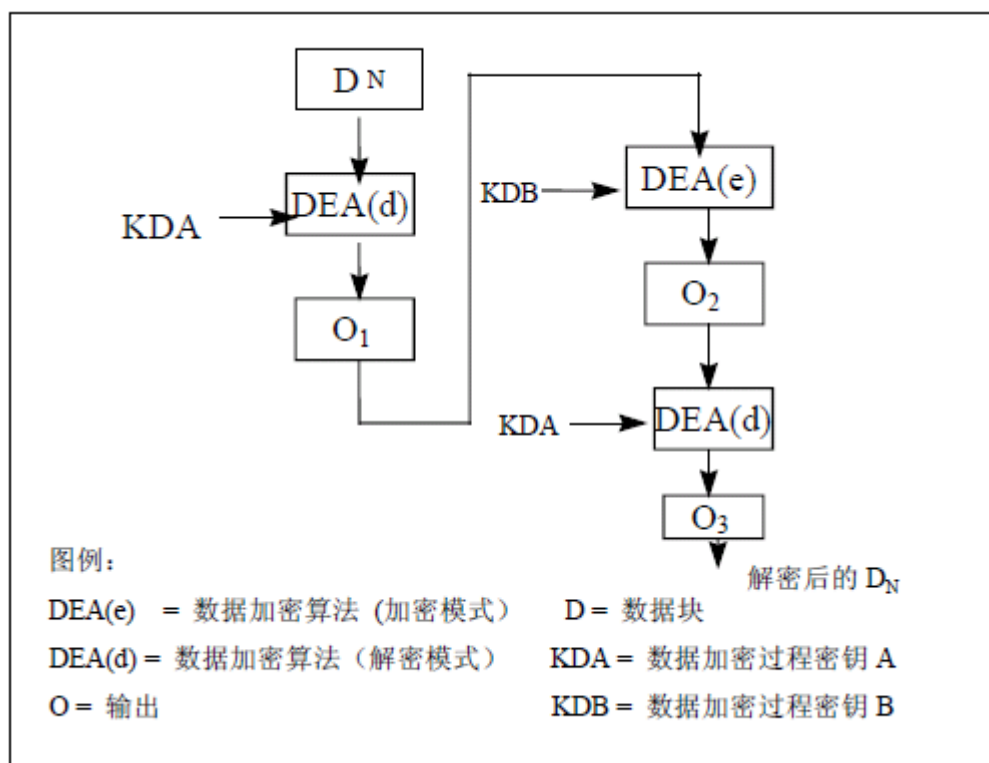


图15 使用双长度 DEA 密钥的数据解密

第二步：计算结束后，所有解密后的数据块依照顺序（解密后的D1、解密后的D2等）链接在一起。数据块由LD、明文数据、填充字符（如果在10.7.3描述的加密过程中增加的话）组成。

第三步：因为LD表示明文数据的长度，因此，它被用来恢复明文数据。

10.8 电子现金密钥管理

安全报文机制要求发卡机构管理唯一的IC卡安全报文认证（MAC）和安全报文加密主密钥。IC卡安全报文认证（MAC）和加密子密钥的分散方法见14章。

11 卡片安全

11.1 共存应用

为了解决独立地管理一张卡上的不同应用的安全问题，每一个应用应该放在一个单独的ADF中。亦即在应用之间应该设计一道“防火墙”以防止跨过应用进行非法访问。另外，每一个应用也不应该与卡中共存的个人化要求和应用规则发生冲突。

11.2 密钥的独立性

用于一种特定功能（如：AC密钥）的加密/解密密钥不能被任何其它功能所使用，包括保存在IC卡中的密钥和用来产生、派生、传输这些密钥的密钥。

11.3 卡片内部安全体系

本条介绍了卡片内部安全的体系结构。对于那些由卡片操作系统控制、并影响任何卡片数据或执行代码的处理过程而言，这一体系的使用将受到限制。

11.3.1 卡片内部安全目标

这一安全体系的目标是保证卡片操作系统使用合适的安全机制，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。这一体系是为访问数据文件和使用的命令与加密算法而设计的。

11.3.2 卡片内部安全概述

这一安全体系的基础结构包括两个基本特性：

- “安全域”的建立；
- 对每个 EF 的存取采用指定的访问条件。

11.3.2.1 安全域

由于操作系统控制了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，这就使得建立安全域成为可能。这一点是通过执行SELECT和GP0命令实现的。这些命令用于建立描述安全域的相关信息，并且（在任何时间）定义了指定数据和可执行资源可以被访问的范围。

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，因此发卡机构就必须认真考虑怎样将数据对象与数据元合并到文件当中。换句话说，在同一层次可访问的数据可以与相似的数据合并到一个文件中去，相反地，访问条件不同的数据不应被并到同一个文件中。

处理SELECT命令使得卡片操作系统信息，即应用管理数据（AMD）可以被访问，AMD指定了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。

应用管理数据（在选择应用之后提交给操作系统）决定了应用可以访问的文件和可执行资源。GPO 命令将使操作系统改变安全域的状态，这就使得对其他文件与记录的引用成为了可能。文件和记录编号则由该命令在应用文件定位器（Application File Locator）中提供。

由于SELECT和GPO命令的执行建立了安全域，发卡机构可以限制在交易期间被存取的资源，包括决定该资源是被包含在应用管理数据和应用文件定位之内，或是被排除在外。不被应用管理数据和应用文件定位引用的数据文件不能够被访问。不被应用管理数据和应用文件定位引用的命令或者加密算法，则不能够在当前安全域范围内被使用。应用管理数据的初始化状态（在个人化阶段被定义）仅包含了处理该应用交易的过程中可以被访问的那些数据文件。

初始的应用管理数据在选择应用时建立，并且在个人化时被定义。其细节则在以下章条描述。

11.3.2.2 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READ RECORD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。文件控制参数的细节在9.3.4提到。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IC卡数据的外部访问，如READ RECORD、GET DATA、PUT DATA、UPDATE RECORD等命令。

11.3.3 文件控制信息

文件控制信息（File Control Information, FCI）附属于每个应用定义文件（Application Definition File, ADF）或者应用基本文件（Application Elementary File, 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。应用定义文件的文件控制信息包含了文件管理数据（File Management Data, FMD），后者可能包含应用管理数据。而应用管理数据定义了应用的安全域。

11.3.3.1 应用管理数据

应用管理数据在个人化期间建立以定义初始的安全域，可以保存在应用定义文件的文件管理数据中。

应用管理数据描述的安全域定义以下内容：

- 在应用范围内可以被存取的资源，应用基本文件（Application Elementary File, AEF）和内部基本文件（如个人识别码 PIN、密钥、参数）；
- 可在应用的上下文范围内被执行的命令；
- 命令与资源之间的关系。

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；换句话说，不同应用的安全域定义可能完全不同。

共有以下两类资源被定义：

- 数据资源（见 11.3.3.2）；
- 可执行代码资源。

此外，资源还可被定义为“尚未分配给应用的”，使得卡片在个人化后可以使用相应的命令将资源分配给应用。资源及其相互间的关系由应用管理数据（AMD）描述。

11.3.3.2 数据资源

数据资源可以是以下列出的任意一个：

- 数据文件及其记录；
- 密钥；
- PIN。

11.3.3.2.1 数据标识

数据资源是指可能被包括在文件内的数据元。数据资源由IC卡内部的唯一标识符所识别。文件由IC卡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必须在应用管理数据内标识。

对包含了数据元（可以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，并且可以从外部被引用）与文件标识（在IC卡内被唯一标识，并且可以从内部被引用）之间的关系被维护在应用管理数据内。

对于未被包含在文件内的数据对象（可以由应用管理数据定义的命令如GET DATA命令访问）而言，数据对象标签（可以从外部被引用）与唯一数据标识（在IC卡内部，并且可以从内部被引用）之间的关系被维护在应用管理数据内。

11.3.3.2.2 密钥标识

密钥可以保存在文件内，也可以是一个独立数据元。密钥不能从外部被引用。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须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须的IC卡内部的唯一密钥标识。

11.3.3.2.3 PIN/口令标识

PIN或者口令可以保存在文件内，也可以是一个独立数据元。PIN和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/口令所必须的文件标识和指向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/口令所必须的IC卡内部的唯一PIN/口令标识。

11.3.3.2.4 可执行代码资源

可执行代码资源包括：

- 命令；
- 加密算法。

11.3.3.2.5 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可能访问的数据的属性，有时还有与密钥和算法相关的参数属性。

11.3.3.2.6 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。

11.3.4 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IC卡内，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IC卡操作系统用于建立应用的安全域。基本文件的访问见表23。

表23 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

读取一栏表示使用读取命令，如READ RECORD或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UPDATE RECORD或PUT DATA命令，存取基本文件内部的数据。

文件控制参数指出是否在发卡机构脚本UPDATE RECORD命令中以加密或者明文格式传送数据。

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。此外，文件控制参数还为卡片上各应用的基本文件描述了强制安全访问条件。

11.3.5 IC卡本地数据建议访问条件

以下建议的数据访问条件适用于可被READ RECORD、UPDATE RECORD、GET DATA命令或其他合适的类似命令访问的数据。

——本建议针对那些只可使用 READ RECORD 命令读取的数据：所有已标签的可以由外部引用的 IC 卡本地数据在没有安全通讯的访问条件下应该设置只读状态。

——此建议列举了可能被 PUT DATA 命令与安全通信改变的数据，以及可能被 GET DATA 命令读取的数据：

- 连续脱机交易下限（“9F58”）；
- 连续脱机交易上限（“9F59”）；
- 连续脱机交易限制数（国际-国家）；
- 连续脱机交易限制数（国际）；
- 累计交易总额限制；
- 累计交易总额限制（两种货币）；
- 累计交易总额上限；
- 货币转换因子。

——此建议列举了可能被应用私有的 PIN CHANGE/UNBLOCK 命令与安全通信所更新的数据，以及不能被读取的数据：

- 参考 PIN。

——此建议列举了可能被 GET DATA 命令读取的数据，以及可能被 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。

11.4 卡片中密钥的种类

在卡片中可能存在的密钥的种类有：

表24 卡片上保存的密钥种类

密钥名称	用途	密钥形式	存在条件	说明
应用密文密钥	用于交易中产生应用密文和发卡机构认证	对称密钥	必须存在	由发卡机构应用密文主密钥，按13.1.4条定义的分散方法获得，在交易过程中，按13.1.3条定义的方法派生过程密钥，用于应用密文产生和发卡机构认证
安全报文认证（MAC）密钥	用于安全报文中计算MAC	对称密钥	必须存在	由发卡机构安全报文认证主密钥，按13.1.4条定义的分散方法获得，在交易过程中，按13.1.3条定义的方法派生过程密钥，用于MAC计算和验证
安全报文加密密钥	用于脚本中数据的加密	对称密钥	必须存在	由发卡机构安全报文加密主密钥，按13.1.4条定义的分散方法获得，在交易过程中，按13.1.3条定义的方法派生过程密钥，用于报文加密
卡片公私钥对	用于动态数据认证	非对称密钥	卡片支持DDA或CDA	由发卡机构私钥对卡片公钥及相关信息签名产生IC卡公钥证书
应用开通密钥	用于开通扩展应用时指定文件记录的增加	对称密钥	卡片支持相关的扩展应用信息的记录	由发卡机构应用开通主密钥，按13.1.4条定义的分散方法获得，在开通应用的过程中，按13.1.3条定义的方法派生过程密钥，用于报文加密
电子现金互联互通密钥	用于在读取或更新扩展应用中指定记录时进行保护	对称密钥	卡片支持相关的扩展应用信息的记录且指定的应用已经开通	由发卡机构扩展应用管理主密钥，按13.1.4条定义的分散方法获得，在交易的过程中，按13.1.3条定义的方法派生过程密钥，用于报文加密

12 终端安全

12.1 终端数据安全性要求

12.1.1 一般要求

终端一般存在两种类型的数据：

- 通用数据：包括时间、终端识别号、终端交易记录等。外界可以对这些数据进行访问，但不允许进行无授权修改；
- 敏感数据：包括认证中心公钥、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。

12.1.1.1 通用数据的安全要求

通用数据一般存放在存储器中。在更新参数以及下载新的应用程序时，终端必须做到：

- 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；
- 校验下载参数及应用程序的完整性。

对存储器要求必须做到：无论在什么情况下，终端的应用数据都不会随意改变或丢失，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。终端须保证这些数据的完整性。

12.1.1.2 敏感数据的安全要求

敏感数据一般应存放在终端安全模块中。

安全模块是一种能够提供必要的安全机制以防止外界对终端所储存或处理的数据进行非法攻击的硬件加密模块。

此模块主要负责保存和处理所有的敏感数据，这些数据包括各种密钥和内部参数。此外该模块还应提供必须的加密功能。对于安全模块的硬件形式在此规范中将不做具体要求。

在正常的操作环境下，对于对称密钥的安全模块必须要求：出入模块的、以及其内部存放的和正在处理的数据不会由于模块自身或其接口造成任何泄露和改变。

12.1.2 安全模块的物理安全要求

安全模块的硬件设计必须能保证在物理上限制对其内部存贮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，其自身必须能够立即完成对内部敏感数据的删除。同时，安全模块也必须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。如果安全模块是由多个分离部件组合而成，而处理的数据又必须在这些部件之间传递，那么各部件须保持相同的安全级别。

12.1.3 安全模块的逻辑安全要求

一个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，都不会导致敏感数据的泄露。对于某些敏感操作，必须有一定的权限限制。

安全模块中可存放多组认证中心公钥及其相关信息。认证中心公钥通常在终端投入使用之前，被导入到安全模块中。如果在终端使用过程中，需要更新或撤回认证中心公钥，必须使用安全报文。实现这一操作通常必须在特殊的授权情况下完成。

当需要以安全报文方式传递信息时，安全模块必须能够实现安全报文传递。

安全模块必须可以实现第15章中所定义的对称算法和非对称算法。

12.2 终端设备安全性要求

12.2.1 防入侵设备

一个防入侵的设备必须保证在它的正常的运行环境中，设备或它的接口不会泄露或改变任何输入或输出设备的、存储在设备中的或者在设备中处理的敏感数据见第6章。

当一个防入侵的设备在一个安全的受控环境中运行时，对该设备特性的要求可以降低，因为受控环境和对设备的管理提供了对设备的保护。

12.2.1.1 物理安全性

一个防入侵的设备必须被设计为限制对内部存储的敏感数据的物理访问，并且阻止窃取数据，未经授权的使用或者未经授权的对设备的修改。这些目标总体上要求将对入侵的抵御、对入侵的检测、对入侵的指示或反应机制结合起来，如可视或有声的报警。

一台不处于运行状态的防入侵的设备，必须不包含在任何以前的交易中用过的加密密钥或者其它的敏感数据（例如PIN），但可以包含只是出于提高防入侵能力的目的的认证信息。如果是在该设备和存储在其中的密钥重新投入使用前能够监测到闯入即使它被非法闯入也不会影响安全。如果设备被设计为允许内部访问，那么在进入时敏感数据必须立即被擦除。一个防入侵的设备依赖于用户对针对物理安全的攻击的监测。因此，这种设备必须被设计为具有足够的防入侵特性，使得任何入侵对于持卡人都应该是明显的或者能被商户或收单机构监测到。

设备必须被设计和构造为：

- 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；如果在没有特别的技巧和专门的装备，并且不对设备造成严重的、显而易见的破坏的前提下，不允许测定或修改任何敏感数据后重新安装设备；
- 只有真正进入设备，才能做到对输入的，存储的或处理的敏感数据的未经授权的访问或修改；
- 包装材料不能采用普通的，以防止使用一般都具备的材料生产‘看上去一样’的假冒复制品；
- 当设备的任何部件发生任何故障时，不会导致秘密或敏感的数据的泄露；
- 如果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，那么对设备的所有部件的保护等级应该是相同的；
- 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。

12.2.1.2 逻辑安全性

防入侵的设备必须被设计为没有单一的函数或函数的组合能够导致敏感数据的泄露，不被一些多指令或任何指令的混合体轻易攻破，除了在终端中实现的安全机制明确允许的以外。即使在使用合法的函数的情况下，也必须有足够的逻辑保护使其不会危及敏感数据的安全。这个要求可以通过内部的统计监控或控制对敏感函数调用的最小时间间隔来实现。

如果终端可以被置于一种“敏感状态”，即允许通常情况下不被允许的函数的状态（例如，人工安装密钥），这样的转换必须在两个或两个以上可信赖的人员的协助下进行。如果用密码或其它明文数据来控制转换到敏感状态，那么这些密码的输入也要用和其它敏感数据一样的方式来保护。

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，对敏感状态必须有调用函数次数（适当的）的限制和时间限制。一旦达到了这些限制，设备必须返回正常状态。

在交易结束或超时后，防入侵的设备必须自动清除内部的缓存。

12.3 终端密钥管理要求

12.3.1 终端密钥种类

在终端中可能存在的密钥的种类有：

表25 终端内部保存的密钥种类

密钥名称	用途	密钥形式	存在条件
认证中心公钥	用于动态数据认证	非对称密钥	必须存在
认证中心公钥维护密钥	用于导入，更新和撤回认证中心公钥	对称密钥	必须存在
PIN加密密钥	用于保护PINPAD到终端的用户PIN	对称密钥	可选
电子现金互联互通密钥	用于保护行业信息	对称密钥	可选（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	必须存在 (储存在钱包应用 SAM 卡中)

12.3.2 电子现金认证中心公钥管理

这一条规定了对收单机构管理终端中的认证中心公钥的要求。这些要求包括以下阶段：

- 将认证中心公钥导入终端；
- 认证中心公钥在终端中的存储；
- 认证中心公钥在终端中的使用；
- 从终端中撤回认证中心公钥。

12.3.2.1 认证中心公钥导入

当一个城市公共交通IC卡系统决定导入一个新的认证中心公钥时，必须保证将新的公钥从认证中心分发给每一个收单机构。保证新的认证中心公钥和相关数据传送给它的终端是收单机构的责任。在将认证中心公钥及其校验和导入到安全模块的过程中，必须通过带报文鉴别码的安全报文机制进行传输，安全模块校验通过后必须返回供确认的验证码，具体采用的安全机制，本部分不作具体规定。

以下的原则适用于一个收单机构将新的认证中心公钥导入它的终端：

- 终端必须能够验证从收单机构收到的认证中心公钥和相关数据没有错误；
- 终端必须能够验证收到的认证中心公钥和相关数据确实是来自它的合法收单机构；
- 收单机构必须能够确认新的认证中心公钥已真正地，正确地导入它的终端。

12.3.2.2 认证中心公钥储存和更新

支持静态和/或动态数据认证的终端必须对每个城市公共交通IC卡应用的RID提供6个认证中心公钥的支持，这些应用都基于本规范。

每一个认证中心公钥由5个字节的RID和1个字节的对于每个RID唯一的、由该城市公共交通IC卡系统分配给某个特定的认证中心公钥的认证中心公钥索引唯一标识。

对于每一个认证中心公钥，表26详细说明了在终端中有用的数据元的最小集。

RID和认证中心公钥索引一起唯一标识了一个认证中心公钥，并将它和正确的城市公共交通IC卡系统联系起来。

认证中心公钥算法标识指明了与相应的认证中心公钥一起使用的数字签名算法，即在14章和15章中指定的数字签名方案中应使用的非对称算法。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。

认证中心公钥储存于终端的安全模块中，可以任意读取，但更新必须使用安全报文，具体信息见第10章。认证中心公钥校验和用来保证认证中心公钥及其相关数据准确无误接收到。随后终端可以用这个数据元重新验证认证中心公钥及其相关数据的完整性。

对存储的认证中心公钥的完整性的验证应该定期进行。

表26 存储在终端中的认证中心公钥相关数据元的最小集

名称	长度	描述	格式
注册的应用提供商标识 (RID)	5	指定认证中心公钥和哪个城市公共交通IC卡系统相关	b
认证中心公钥索引	1	和RID一起指定认证中心公钥	b
认证中心哈希算法标识	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

认证中心公钥算法标识	1	标识使用在认证中心公钥上的数字签名算法	b
认证中心公钥模	变长最大 为 248	认证中心公钥模部分的值	b
认证中心公钥指数	1或3	认证中心公钥指数部分的值，等于3或 $2^{16}+1$	b
认证中心公钥校验值	20	使用15.3条指定的哈希算法对认证中心公钥所有部分（RID、认证中心公钥索引、认证中心公钥模、认证中心公钥指数）的连接计算得到的校验值	b

12.3.2.3 认证中心公钥使用

交易中对认证中心公钥的使用必须遵照本部分的规定。

12.3.2.4 认证中心公钥回收

当城市公共交通IC卡系统已经决定撤回它的某一个认证中心公钥时，收单机构必须保证在一个确定的时间后它的终端在交易中不再将这个认证中心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认证中心公钥从它的终端撤回：

- 终端必须能够验证它从收单机构收到的撤回通告没有错误；
- 终端必须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；
- 收单机构必须能够确认一个特定的认证中心公钥已经真正地、正确地从它的终端撤回。

认证中心公钥的撤回指令也应通过安全报文传输，有关认证中心公钥回收和相应涉及的时间表的更多细节，见第7章。

12.3.3 SAM 卡

SAM卡分为互联互通SAM卡和地区扩展应用SAM卡两种，互联互通SAM卡由全国运营机构统一发行和管理，地区扩展应用SAM卡由各地区相关发卡机构或行业方发行和管理。

互联互通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。

13 个人化安全

13.1 安全综述

在对城市公共交通IC卡个人化的过程中，每一个步骤都有其特定的安全要求。现就各方面的要求，制定出以下的规范。

关于加密方式，根据本规范所提出的方式来对数据进行加密。

13.2 初始化安全

密钥数据（KEYDATA）必须按表27设置，该数据由KMCID和芯片序号（CSN）组成。KMCID是个人化主密钥标识符，应由发卡机构或个人化厂商提供。KMCID的长度为6个字节。CSN是IC卡片物理标识符最右边的4个字节。

表27 KEYDATA 的初始存储内容

字段	长度	格式
KMC（例如 IIN/BIN，左对齐，用 1111b/半字节填充）标识	6	BCD

芯片序号 (CSN)	4	二进制数
------------	---	------

KEYDATA (密钥数据) 是每个IC卡应用分区都可以访问的一个数据单元, KMCID是INITIALIZE UPDATE 命令响应数据的一部分, 并给定位IC卡发行商的KMC提供了方便。

在IC卡上必须存在‘个人化主密钥 (KMC)’的版本号, 这个主密钥用来为每个应用生成初始的个人化密钥 (K_{ENC} 、 K_{MAC} 和 K_{DEK})。

必须为每张IC卡生成一个加密分散密钥 (K_{ENC}), 并把它写入相应的应用中。这个密钥用来生成IC卡密文和验证主机密文。如果密文的安全等级要求STORE DATA命令的数据字段是加密的, 这个分散密钥还用来在CBC模式下对该命令的数据字段进行解密。

K_{ENC} 是一个16字节 (112比特加奇偶校验位) 的DES密钥。

K_{ENC} 密钥用以下方法推算:

$K_{ENC} := \text{DES3}(\text{KMC})[\text{KEYDATA的6个最低有效字节} || 'F0' || '01'] || \text{DES3}(\text{KMC})[\text{KEYDATA的6个最低有效字节} || '0F' || '01']$ 。

必须为每张IC卡生成一个校验码分散密钥 (K_{MAC}) 并写入相应的IC卡。这个密钥用来校验EXTERNAL AUTHENTICATE命令使用的C-MAC。同时当STORE DATA命令的密文安全级要求命令中的数据采用MAC时, 这个密钥也用来校验STORE DATA命令使用的C-MAC。

K_{MAC} 是一个16字节 (112比特加奇偶校验位) 的DES密钥。

K_{MAC} 应采用以下方法导出:

$K_{MAC} := \text{DES3}(\text{KMC})[\text{KEYDATA的6个最低有效字节} || 'F0' || '02'] || \text{DES3}(\text{KMC})[\text{KEYDATA的6个最低有效字节} || '0F' || '02']$ 。

必须为每张IC卡生成一个密钥加密分散密钥 (K_{DEK}) 并将它写入相应的IC卡。这个密钥用来在ECB模式下对STORE DATA命令收到的机密数据进行解密。

K_{DEK} 是一个16字节 (112比特加奇偶校验位) 的DES密钥。

K_{DEK} 应采用以下方法导出:

$K_{DEK} := \text{DES3}(\text{KMC})[\text{KEYDATA的6个最低有效字节} || 'F0' || '03'] || \text{DES3}(\text{KMC})[\text{KEYDATA的6个最低有效字节} || '0F' || '03']$ 。

必须把IC卡响应INITIALIZE UPDATE命令时返回来的序列计数器初始化为‘0001’。

13.3 密钥定义

13.3.1 个人化密钥描述

表28 个人化密钥描述

密钥名称	密钥共享	用途	主密钥	卡密钥	对话密钥
发卡机构主密钥	发卡机构、IC卡厂商和个人化设备	IC卡厂商使用这个KMC生成卡片级密钥 (K_{ENC} 、 K_{MAC} 、 K_{DEK}), 并将它们写到卡上。	KMC		
		用来创建一个对话密钥, 利用该对话密钥可创建密文和以CBC模式加密机密数据。		K_{ENC}	SKU_{ENC}
		用来创建一个对话密钥, 利用该对话密钥可创建命令处理过程中所使用的C-MAC。		K_{MAC}	SKU_{MAC}
		用来创建一个对话密钥, 利用该对话密钥可在ECB模式下加密DES密钥或灵活的加密其它机密数据。		K_{DEK} 数据加密密钥	SKU_{DEK}
发卡机构密钥	发卡机构和数	对发卡机构与数据准备设备之间的脱机PIN及其它	KEK_{ISS}		

城市公共交通 IC 卡安全技术规范

交换密钥	据准备设备	机密数据进行保护。			
数据加密密钥/传输密钥	数据准备设备和个人化设备	对数据准备设备与个人化设备之间的脱机 PIN 及其它机密数据进行保护。 下列特殊类型的数据传输密钥可能会被使用： ——PEK/TK: PIN 加密密钥，用于保护 PIN 数据； ——KEK/TK: 密钥交换密钥，用于保护 DES 密钥。	DEK/TK		
MAC 密钥（校验码密钥）	在个人化数据文件中，由数据准备设备向个人化设备提供	用于保证在个人化数据文件中，提供给个人化设备的应用数据的完整性。	MAC 密钥	不适用	不适用

13.3.2 应用密钥描述

表29 应用密钥描述

密钥名称	密钥共享	用途	主密钥	卡密钥	对话密钥
联机验证密钥	发卡机构和卡	主密钥用来生成唯一的卡片密钥，用于卡片和发卡机构进行联机验证。	MDK	UDK	SUDK（用于通用密文）
信息认证密钥	发卡机构和卡	主密钥用来生成唯一的卡片密钥，这个卡片密钥用于生成进行发卡后的数据更新所需要的消息认证对话密钥。	MAC MDK	MAC UDK	SUDK MAC
应用数据加密密钥	发卡机构和卡	主密钥用来生成唯一的卡片密钥，这个卡片密钥用于生成对发卡后更新机密数据（脱机 PIN）进行加密的对话密钥。	ENC MDK	ENC UDK	SUDK ENC
ICC 私钥	发卡机构和卡	由发卡机构生成并安全地存储在卡上。在动态数据验证（DDA）处理过程中，用这个私钥对动态数据进行数据签名。个人化完成以后，发卡机构通常不持有该密钥。			
应用开通密钥	发卡机构和卡	主密钥用来生成唯一的卡片密钥，这个卡片密钥用于在指定的扩展应用记录文件中新增记录。			
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）	全国运营机构/发卡机构或行业方和卡	主密钥用来生成唯一的卡片密钥，这个卡片密钥用于在指定的扩展应用记录文件中更新记录。			
电子现金消费主密钥	全国运营机构	主密钥用来生成唯一的卡片密钥，这个卡片密钥用于电子现金的消费。	MPK	PK UDK	SUPK

13.4 管理要求

13.4.1 环境

13.4.1.1 创建安全数据的环境要求

参见各相关组织的相关个人化安全和质量管理要求。

13.4.1.2 安全数据的产生

开始进行个人化之前，必须创建相应的加密密钥，这些密钥可以由发卡机构创建，也可以由个人化厂商创建。如果由个人化厂商创建，必须按本部分的规定进行。

至少应生成以下的密钥：

1) 发卡机构主密钥（KMC）：用来派生 K_{MAC} 、 K_{ENC} 和 K_{DEK} 三个密钥。

- K_{MAC} ——用来锁闭中国交通运输集成电路（IC）卡的应用区，并对个人化过程中装载到卡片的个人化数据进行检验，证实它们完整无损，且没有被修改；
- K_{ENC} ——用来生成 IC 卡密文和验证主机密文；
- K_{DEK} ——用来加密在个人化过程中写入卡片的保密数据。

KMC 对每个发卡机构是独有的，而 K_{MAC} 、 K_{ENC} 和 K_{DEK} 对每张卡是独有的。

2) 主密钥（MDK）——用来导出：UDK——用于联机的卡认证和发卡机构认证。

3) 就每个 BIN（发卡机构标识码）而言，MDK 通常是唯一的，而 UDK 对每张卡都必须是唯一的。

4) 发卡机构公私钥对——通常由发卡机构生成，公钥应传输给电子支付卡认证机构，供其创建发卡机构公钥证书。私钥被保存在发卡机构的 HSM（主机加密模块）内。

5) 密钥交换密钥（KEK）——用来对发卡机构个人化输入文件中的机密数据进行加密，每个发卡机构的 KEK 必须是唯一的。

6) 传输密钥（TK）——用来对数据准备系统向个人化系统传送的发卡机构个人化输入文件中的机密数据进行加密。

7) 作为选择，也可以用发卡机构公私钥对生成这些密钥。

8) ICC 公私钥对——IC 卡利用这一对密钥执行 DDA 和 CDA/AC 密文生成算法。其中，公钥须经过发卡机构私钥的签名，才能获得发卡机构公钥证书。

9) 每张卡的 ICC 公私钥对必须是独一无二的。

10) MDK ENC——用来导出：UDK ENC——用来加密发卡机构的脚本机密信息。

11) MDK MAC——用来校验发卡机构的脚本信息。

MDK ENC 和 MDK MAC 至少对于每个 BIN 是独一无二的，而 UDK ENC 和 UDK MAC 必须对每张卡都是唯一的。

如果发卡机构生成自己的密钥，就必须创建 ZMK，以便联机传输这些密钥。

在 IC 卡之外执行的一切加密和解密操作必须在 HSM 上进行。

密钥区：

一般说来，个人化过程有三个密钥区：在发卡机构和数据准备系统之间有一个密钥区，在数据准备系统和个人化设备之间有一个密钥区，在个人化设备和卡片之间还有一个密钥区，如图16所示。

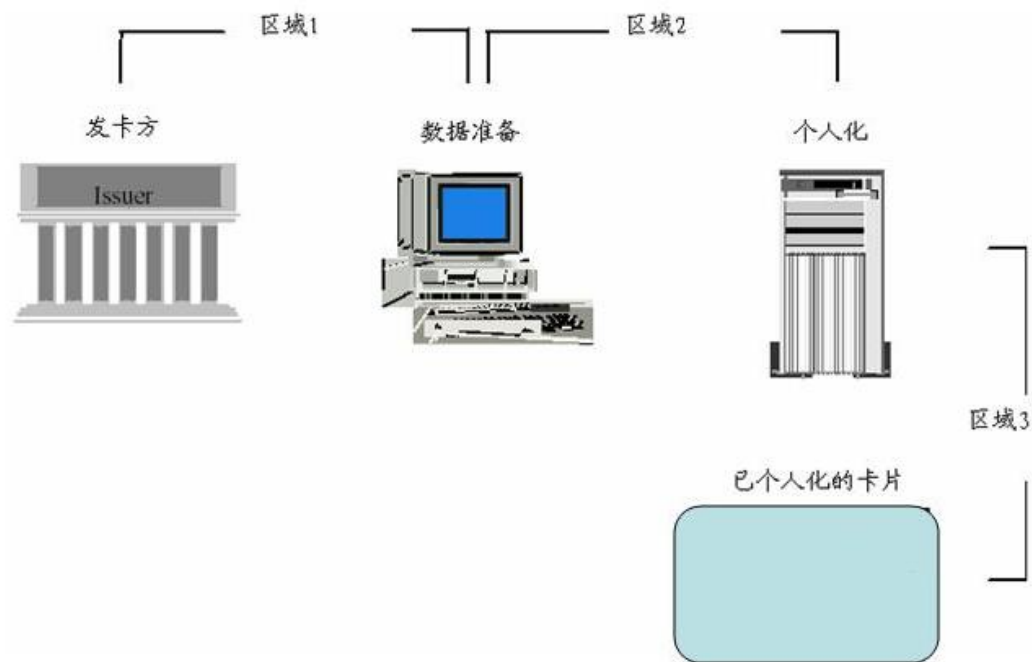


图16 密钥区

在发卡机构和数据准备系统之间的密钥区里，建有一个通称为KEK（密钥交换密钥）的加密密钥，采用该密钥对电子支付卡的安全信息进行加密。

该密钥可以由发卡机构生成，也可以由个人化设备生成，但必须遵循本部分的要求。除了这些要求以外：

- KEK 对每个发卡机构必须是独一无二的；
- KEK 至少必须逐年进行更改。

接收来自发卡机构的机密数据时，必须把KEK密钥替换成TK（传输密钥），以便在数据准备过程和个人化设备之间对机密数据进行加密。

该密钥必须独特于其它所有密钥，并且只承担加密机密数据的任务。

个人化设备接收这些机密数据时，机密数据必须从TK变换成IC卡的 K_{DEK} ，然后通过个人化过程被传送给IC卡，再由IC卡对它们进行解密和存储。

13.4.1.3 发卡机构客户资料到个人化数据的安全流程

接收来自发卡机构的个人化文件时，文件信息必须：

- 始终得到安全的存储，访问这些信息的权利必须严格地局限于业务需求者；
- 在成功完成个人化之后，将生产系统内的数据清除干净。

另外，机密信息必须：在HSM上从KEK解译成TK，以便将机密信息向前传输给个人化设备。

此外，数据准备系统至少必须：位于一个能够控制数据存取的中间安全区，并将数据访问权局限于业务需求者。

13.4.1.4 数据组的安全要求

加密过程的安全要求应适合于给定的数据组及IC卡用途，而且无论是在数据准备过程中，还是在个人化设备相关的本机处理过程中，都必须与相应的加密过程协调一致。

13.4.1.5 个人化过程中的安全要求

在个人化阶段，个人化设备必须：

- 在 HSM 上执行本部分定义的 K_{DEK} 推算过程；
- 将个人化文件中的机密信息从传输密钥 TK 解译成 K_{DEK} ，以便将其传送给卡片。这一解译过程必须在 HSM 上执行。

另外，个人化设备必须安装在工厂的高安全区并符合各支付组织的相关生产安全标准规定的一切安全要求和程序要求。

13.4.2 操作

13.4.2.1 密钥的形成与分发

当加密算法没有得到正确实施时，加密算法的预定作用将受到负面的影响。一种安全的实施将取决于规范所需的不同密钥被签发者管理的好坏程度。以下材料的目的是提供不同算法类型所扮演的加密角色的一个概述，以及提出安全地管理密钥所必需的基本要求。

非对称（RSA）密钥管理：

IC卡的安全性取决于私钥（签名）的保护。不能保证用来对静态或动态数据元签名的私钥的安全性将使IC卡面临被伪造的风险。私钥面临的主要风险包括：

- 成功地分解 RSA 模数；
- 私钥自身的泄漏。

为了限制这些风险所代表的潜在的泄露问题，我们推荐使用以下发卡机构要求。

私钥（签名）的安全性取决于许多因素，包括：

- RSA 密钥模数位的长度，例如：1024 和 1152；
- 组成公钥/私钥模数的主要数字的质量；
- 用来从物理上保障（保护）私钥（签名）不受未经授权的访问和暴露/危害的影响的方法，特别是当 IC 卡或其它安全加密设备（SCD）使用它们时为密钥提供的保护。

RSA密钥生成：

当生成RSA公私钥对时，推荐在一台物理安全的设备的受保护内存中完成这个过程。这种设备必须包含一个随机或伪随机数字生成器，执行原始校验例程，并支持篡改响应机制。

- RSA 私钥（签名）可能对物理安全设备而言是暂时性的；
- 密钥生成将利用一个随机或伪随机过程，以使得不可能预测出任何密钥或者确定密钥空间中的某些密钥比其它任意密钥可能性更大；
- 个人计算机或其它类似的不安全设备，即不被信任的设备，将永远不能用来生成 RAS 公私钥对。

密钥传输和存储：

为了保护公私钥对的完整性，对发卡机构而言，确保这种密钥数据使用以下步骤非常重要：

- 公钥应能被确保安全以及用一种能够保证它们完整性的方式来传输。推荐公钥始终在诸如一个证书之类的数据结构中传输，或者可以跟一个报文鉴别码（MAC）来保证完整性，这个报文鉴别码是由一个仅用于该用途的密钥按照 GB/T 15852.1 定义的算法应用于公钥和相关数据而得。也推荐使用双重控制技巧来确保公钥的接收方有办法验证它的发送方和完整性，即通过公钥上的一个校验值的单独和独立的传输来实现这一点；
- 私钥必须用一种能够保证它们的完整性和私密的方式来保障安全和传输。传输机制可能包括：
 - 一台安全加密设备；
 - 利用至少与加密相等力量的对称算法来对被保护密钥的私钥进行解密；
 - 作为几个部分（在 IC 卡上保障安全），并使用一个对称算法来进行解密。

对称（DES）密钥管理：

本规范中的DES密钥用于特殊的事务功能。DES密钥是在个人化期间从一个主导出密钥（Master Derivation Key）中导出的。最终的卡片级密钥是唯一的。

DES发卡机构主密钥包括：

- 发卡机构主导出密钥（IDKAC）：用来导出卡片密钥，该密钥用于生成称为应用密文（AC）的MAC；
- 发卡机构安全报文主密钥（IMKSMC IMKSMI）：用来导出卡片密钥，这些卡片密钥用在卡片和验证系统之间的安全报文中，即卡片锁定、应用锁定/解锁、更新卡片特定数据和修改PIN。

密钥生成：

发卡机构将使用以下原则来使密钥数据在创建期间泄漏的机会最小化：

- 在生成DES密钥时，它们必须要么在一台由篡改响应机制保护的物理安全的设备中生成，要么必须由授权的工作人员以一部分一部分的形式生成（见下文）。设备必须包含一个随机或伪随机的数字生成器；
- 任何时候一个未被保护的密钥都不能存在于一台物理安全的设备的被保护内存之外。任何时候物理安全的设备都不能输出纯文本的密钥，除非作为密码或者以两个或更多部分的形式输出；
- 当密钥由授权工作人员通过一个将各部分组合的过程来生成时，必须要求每一方生成一个和要生成的密钥一样长的部分。密钥组合过程在一个物理安全的设备内部进行。此外，组合各部分的方法应当是，知道了各部分的任何一个子集也无法知道密钥值。分开的密钥由一个管理机构掌握，至少应有一个部分持有人是发卡机构的一名员工；
- 应当为实际密钥的全部计算校验位；
- 个人电脑或类似的不安全设备永远不能用来生成密钥资料；
- 如果发现任何密钥存在于一个物理安全的设备之外，或者密钥的各个部分被人所知，或者有被单个人掌握的嫌疑，那么该密钥将被认为已被泄漏，并且必须用一个新的密钥来替换它。

密钥传输和存储：

DES密钥可能需要被传输和存储。例如包括将DES密钥从发卡机构的站点传输给一个第三方的处理商或卡片个性化供应商。当DES密钥正被传输或存储时，以下措施将限制数据泄漏的潜在危险：

- DES密钥可以被安全地转移到一块安全令牌或智能卡的保护之下，以进行传输和存储；
- DES密钥只能以以下方式在安全令牌或智能卡的受保护内存之内进行传输或存储。

利用双重控制和分持机密的原则，以两个或更多部分的形式作为密码，密码是用一个由各方安全地建立的传输或存储密钥来创建的。

13.4.2.2 根密钥明文数据的保存

- 1) 一旦接收到密钥资料，负责的密钥管理人员必须立即检查邮包是否篡改，并且必须验证内容；
- 2) 如果接收的管理人员对密钥数据的完整性有任何不确定的地方，必须立即通知发送方。发送方与接收方商议决定密钥数据将来的状况。关于继续使用密钥资料的任何决定的基础必须记录在案并由双方保留；
- 3) 如果硬拷贝数据要保留任意一段时间，那么各个硬拷贝组成部分、安全令牌或智能卡必须保存在一个序列化的保密信封中；
- 4) 这个序列化的保密信封必须持续保存在一个物理安全的容器中，这个容器仅能由指定的密钥管理人员或预备人员访问。每次对密钥数据的访问都必须记入日志，包括时间、日期、信封序列号、目的和签名。这些日志将可以向任何相应的请求机构提供；
- 5) 密钥资料永远不能在超过任务所需的访问必需的时间之后保留在保密信封和它们的物理安全的环境之外。

13.4.2.3 其他密钥数据的保存

下面给出了关于密钥存储问题的一些一般的指导，它适用于非对称和对称密钥存储：

1) PC 板的使用

一块向主机提供加密服务的 PC 板可以看作是 HSM 的一种形式和类似的期望保护级别。

注：使用加密安全设备的主要原因是保护密钥。如果使用HSM主机的系统自身是不安全的，那么攻击者将更容易危害系统的软件功能，而忽略HSM。

2) 访问控制

所有在卡外和HSM外保存的密钥都应当保持在至少双重的控制之下。

3) HSM 和 IC 安全内存

一般而言，HSM将包含单独的存储和处理设备，而密钥资料将跨内部硬件总线传送。由于这个原因，当检测到了危害时，HSM清除（或归零）它的内存是很重要的。此外，HSM的硬件设计解决电磁辐射的问题也很重要。HSM一般设计位于一个安全的环境之中。

13.4.2.4 操作流程

发卡机构在发卡之前必须执行以下几个步骤，这些步骤有时还需要在使用城市公共交通IC卡系统的过程中得以执行：

1) 生成发卡机构密钥对

发卡机构必须安全地生成并保存一对或几对公钥和私钥。私钥将被用来签署IC卡静态数据或IC卡公钥证书（这取决于IC卡各自执行的数据认证是静态的还是动态的）。在支付模式允许的情况下，建议发卡机构为每个发卡机构标识码（BIN）或首标分配不同的密钥对，这样，一旦发卡机构的私钥被泄密，就可以把相应的BIN锁闭起来。

2) 生成发卡机构密钥

发卡机构必须根据IC卡密钥的推算需要而安全地生成并保存一个或几个密钥。

接收“CA 公钥（Public Key）”

发卡机构必须接收并安全地保存一个或几个CA公钥。这些公钥必须以一定的方式进行传输，使发卡机构能够对它们的完整性和数据源进行核实。CA公钥将用来验证发卡机构公钥证书。

3) 请求并接收发卡机构公钥证书

就发卡机构公钥而言，发卡机构必须获得相应的发卡机构公钥证书。为此，须将每个发卡机构公钥传输给CA认证机构（CA），继而发卡机构会收到发卡机构的公钥证书。发卡机构公钥必须以一定的方式传输给CA认证机构，使之能够对公钥的完整性和数据源进行核实。接收CA认证机构发来的公钥证书时，发卡机构可以采用CA公钥对证书进行验证。

4) 传输“发卡机构加密密钥”

如果发卡机构希望授权给第三方生成和验证IC卡密码，发卡机构必须将推算IC卡密钥所使用的发卡机构加密密钥安全地传输给第三方。

13.4.3 管理规范

13.4.3.1 人员管理

负责管理加密密钥和密钥要素及其它密钥数据设备的人员必须由不同的参与方（即发卡机构、第三方处理商和/或IC卡个人化厂商）指派。

指派专人负责监控密钥数据时，必须落实足够的保密控制措施，以保证任何个人或未经认可的个人没有任何机会读取密钥的数据成分。

密钥保管人必须是正式受托的职员，决不可以是临时用工或顾问。

另外，为了确保服务的连续性，可以把候补人员当作主要密钥管理人的“备份”。选择“备份”管理人的标准应该和选择主要密钥管理人的标准相同。

密钥管理人的责任重大，而且是发卡机构安全协议的一个基本组成部分，他们所要管理的密钥数据是发卡机构发卡程序中最重要加密操作码。每个发卡机构应对内部密钥管理程序和下列业务的有关人员的作用进行核查：

- 1) 密钥管理人员的职责包括密码资料的控制、验证和安全存储；
- 2) 密钥管理人或其“备份”的责任是：
 - 接收和安全存储密钥元；
 - 对读取和使用密钥数据的记录或日志进行管理，包括读取次数、日期、目的和重新安全存储情况；
 - 对传输给发卡机构控制权限以外的其它所有指定人员的密钥数据进行验证；
 - 对过期密钥元的销毁进行签名作证；
 - 时常根据需要将密钥数据输入安全加密模块；
 - 依据数据所有人的通知，指导和监视过期密码资料的销毁。
- 3) 密钥数据最初生成时的密钥管理人，应负责保护该数据，并将其转发给接收单位的指定密钥管理人，这个责任还包括对数据收讫进行验证。

13.4.3.2 操作管理

参见各支付组织的相关个人化安全和管理要求。

13.4.3.3 文档管理

1) 数据传输安全管理

参见各支付组织的相关个人化安全和管理要求。

2) 数据存储介质的管理

参见各支付组织的相关个人化安全和管理要求。

3) 数据信息使用的控制

参见各支付组织的相关个人化安全和管理要求。

13.5 安全模块

防止篡改的要求：篡改的防止可以分为物理和逻辑两个安全领域。

13.5.1 物理安全属性

物理安全包括以下属性：

- 对侵入的保护，包括擦除敏感数据；
- 对将导致敏感信息暴露的未授权修改的保护；
- 防止对设备运转带来的电磁辐射的监控的保护。

13.5.2 逻辑安全属性

逻辑安全特性包括以下属性：

- 真实性的验证；
- 设备功能集的设计确保没有单个或设备功能的组合将导致敏感信息的泄露；
- 存在的、用来确保密钥分割的机制；
- 敏感状态操作需要双重控制；

——包含的用来验证软件下载的技巧。

13.5.3 功能需求

一个HSM的最小的需求应围绕着对以下内容的支持：

- 密钥值生成；
- 密钥值交换；
- 密钥配置文件分离（逻辑分割密钥属性）；
- 密钥值输出和输入；
- 密钥值的安全存储。

13.5.4 安全模块等级

HSM须符合国家制订的法规。

13.6 风险审计

在每个IC卡应用的个人化过程的最后，必须创建这个应用的个人化过程的记录。在整个个人化过程的最后，必须创建包含所有的IC卡应用的个人化过程纪录的审计文档。

这些记录可保证对个人化过程的可审计性和可跟踪性。

14 安全机制

14.1 国际算法对称加密机制

14.1.1 加密解密

对数据的加密采用分组长度为64位（8字节）或128位（16字节）分组加密算法，可以是电子密码本（ECB）模式或密码块链接（CBC）模式。本规范选用ECB模式作为加密解密模式。

用加密过程密钥 K_s 对任意长度的报文MSG加密的步骤如下：

1) 填充并分块

——如果报文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，然后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；

——如果报文MSG的长度是分组长度的整数倍，不对数据作填充。

被加密数据首先要被格式化为以下形式的数据块：

- 明文数据的长度，不包括填充字符；
- 明文数据；
- 填充字符（按上述填充方式）。

然后MSG被拆分为8字节或16字节的块 X_1, X_2, \dots, X_K 。

2) 密文计算

ECB模式：

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, \dots, X_K 加密为分组长度的块 Y_1, Y_2, \dots, Y_K 。

因此当 $i=1, 2, \dots, K$ 时分别计算：

$$Y_i: = \text{ALG}(K_s)[X_i]。$$

CBC模式：

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

因此当 $i=1, 2, \dots, K$ 时分别计算：

$$Y_i = \text{ALG}(K_S)[X_i \oplus Y_{i-1}],$$

Y_0 的初始值为：

——对应 64 位分组加密算法

$$Y_0 = ('00' || '00' || '00' || '00' || '00' || '00' || '00' || '00');$$

——对应 128 位分组加密算法

$$Y_0 = ('00' || '00' || '00' || '00' || '00' || '00' || '00' || '00' || '00' || '00' || '00' || '00' || '00' || '00' || '00' || '00');$$

记为：

$$Y = (Y_1 || Y_2 || \dots || Y_K) = \text{ENC}(K_S)[\text{MSG}].$$

3) 密文解密

解密过程如下：

ECB模式：

当 $i=1, 2, \dots, K$ 时分别计算：

$$X_i = \text{ALG}^{-1}(K_S)[Y_i].$$

CBC模式：

当 $i=1, 2, \dots, K$ 时分别计算：

$$X_i = \text{ALG}^{-1}(K_S)[Y_i] \oplus Y_{i-1},$$

Y_0 的初始值为：

——对应 64 位分组加密算法

$$Y_0 = ('00' || '00' || '00' || '00' || '00' || '00' || '00' || '00');$$

——对应 128 位分组加密算法

$$Y_0 = ('00' || '00' || '00' || '00' || '00' || '00' || '00' || '00' || '00' || '00' || '00' || '00' || '00' || '00' || '00' || '00');$$

4) 为了得到原来的报文 MSG，将块 X_1, X_2, \dots, X_K 连接起来，如果使用了填充（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：

$$\text{MSG} = \text{DEC}(K_S)[Y].$$

14.1.2 报文鉴别码

14.1.2.1 基于 64 位分组加密算法的 MAC 计算方法

MAC 的长度为 4 字节。

计算一个 s 字节的 MAC（ $4 \leq s \leq 8$ ）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 的步骤如下：

1) 填充并分块

依据 GB/T 16649.4 对报文 MSG 进行填充，因此在 MSG 的右端强制加上 1 个 '80' 字节，然后再在右端加上最少的 '00' 字节，使得结果报文的长度 $\text{MSG} = (\text{MSG} || '80' || '00' || '00' || \dots || '00')$ 是 8 字节的整数倍。

然后 MSG 被拆分为 8 字节的块 X_1, X_2, \dots, X_K 。

2) MAC 过程密钥

MAC 过程密钥 K_S 既可以只包括最左端密钥块 $K_S = K_{SL}$ ，也可以由最左端密钥块和最右端密钥块连接而成 $K_S = (K_{SL} || K_{SR})$ 。

3) 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, \dots, X_K ：

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

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

用以下的两种方法的一种计算8字节的块 H_{K+1} 。

- 依照 GB/T 27929 算法 1: $H_{K+1} = H_K$;
- 依照 GB/T 27929 算法 3: $H_{K+1} = \text{ALG}(K_{SL}) [\text{ALG}^{-1}(K_{SR}) [H_K]]$ 。

本规范使用第二种计算方法。

MAC值S等于 H_{K+1} 的s个最高位字节。

14.1.2.2 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 ($4 \leq s \leq 8$) 值S的步骤如下：

1) 填充并分块

依据GB/T 16649.4对报文MSG进行填充，因此在MSG的右端强制加上1个‘80’字节，然后再在右端加上最少的‘00’字节，使得结果报文的长度 $\text{MSG} = (\text{MSG} || '80' || '00' || '00' || \dots || '00')$ 是16字节的整数倍。

然后MSG被拆分为16字节的块 X_1, X_2, \dots, X_K 。

2) MAC 过程密钥

MAC过程密钥 K_s 长度为16字节。

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, \dots, X_K ：

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

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。

14.1.3 过程密钥分散

14.1.3.1 基于 64 位分组加密算法的过程密钥分散方法

MAC和数据加密过程密钥的产生如下所述（在本条中统称为“过程密钥A”和“过程密钥B”）：

1) 单长度 DES 过程密钥

第一步：卡片/发卡机构决定是使用MAC密钥A和B还是数据加密密钥A和B来进行所选择的算法处理。（以后统称为“KeyA”和“KeyB”）

第二步：将当前的ATC在其左边用十六进制数字‘0’填充到8个字节，用KeyA和KeyB对该数据作如图11所示的3-DES运算产生过程密钥A。

$Z = 3\text{-DES}(\text{Key})['00' || '00' || '00' || '00' || '00' || '00' || \text{ATC}]$ 。

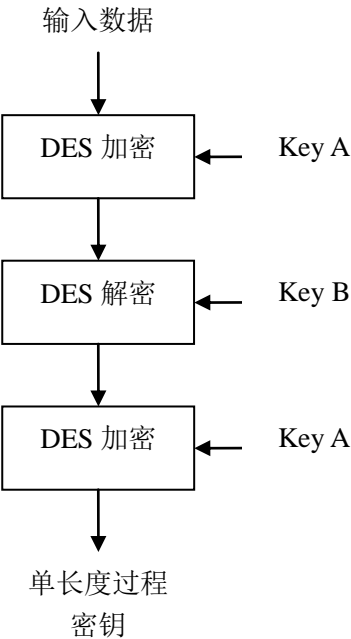


图17 单长度过程密钥的产生

2) 双长度 DES 过程密钥

第一步：卡片/发卡机构决定是使用MAC密钥A和B还是数据加密密钥A和B来进行所选择的算法处理。（以后统称为“KeyA”和“KeyB”）

第二步：将当前的ATC在其左边用十六进制数字“0”填充到8个字节，用KeyA和KeyB对该数据作如图11所示的3-DES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“0”填充到8个字节，使用相同方法对该数据作如图11所示的3-DES运算得到过程密钥B。

$$Z_L: = 3\text{-DES}(\text{Key})[‘00’\|‘00’\|‘00’\|‘00’\|‘00’\|‘00’\|ATC] \text{ 。}$$
$$Z_R: = 3\text{-DES}(\text{Key})[‘00’\|‘00’\|‘00’\|‘00’\|‘00’\|‘00’\|(ATC \oplus ‘FFFF’)] \text{ 。}$$

为了符合对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。

14.1.3.2 基于 128 位分组加密算法的过程密钥分散方法

MAC和数据加密过程密钥的产生如下所述：

第一步：卡片/发卡机构决定是使用MAC密钥还是数据加密密钥来进行所选择的算法处理。

第二步：将当前的ATC在其左边用十六进制数字’0’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“0”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如图12所示的运算产生过程密钥。

$$Z: = \text{ALG}(\text{Key})[[‘00’\|‘00’\|‘00’\|‘00’\|‘00’\|‘00’\|ATC\|‘00’\|‘00’\|‘00’\|‘00’\|‘00’\|(ATC \oplus ‘FFFF’)] \text{ 。}$$

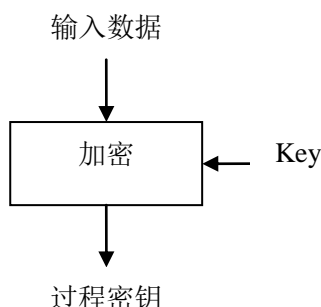


图18 128 位分组加密算法过程密钥的产生

14.1.4 子密钥分散

本条指定了一种利用一个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IC卡子密钥的方法。

这一方式以主账号（PAN）和主账号序列号（如果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，以及16字节的发卡机构主密钥IMK作为输入，生成16字节的IC卡子密钥MK作为输出：

- 1) 如果主账号和主账号序列号 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“0”以获得 8 字节的 Y。如果 X 的长度至少有 16 个数字，那么 Y 由 X 的最右边的 16 个数字组成。
- 2) 计算 2 个 8 字节的数字
 - a) 基于64位分组加密算法的计算方法：

$$Z_L: = \text{ALG}(\text{IMK})[Y]。$$

以及

$$Z_R: = \text{ALG}(\text{IMK})[Y \oplus ('FF' || 'FF' || 'FF' || 'FF' || 'FF' || 'FF' || 'FF' || 'FF')]。$$

并定义

$$Z: = (Z_L || Z_R)。$$

- b) 基于128位分组加密算法的计算方法：

$$Z: = \text{ALG}(\text{IMK})[Y || (Y \oplus ('FF' || 'FF' || 'FF' || 'FF' || 'FF' || 'FF' || 'FF' || 'FF'))]。$$

16字节的IC卡子密钥MK就等于Z，此外对于DES算法，Z的每个字节的最低位应被设成能够保证MK的16个字节的每一个都有奇数个非0位（为了符合对DES密钥奇校验的要求）。

14.2 国密算法对称加密机制

14.2.1 加密解密

对数据的加密采用 16 字节分组加密算法，可以是电子密码本（ECB）模式或密码块链接（CBC）模式。本文选用 ECB 模式。

用加密过程密钥 K_s 对任意长度的报文 MSG 加密的步骤如下：

- 1) 填充并分块：

——如果报文 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，然后在右端加上最少的‘00’字节，使得结果报文的长度 $\text{MSG}: = (\text{MSG} || '80' || '00' || '00')$

||...|| ‘00’)是分组长度的整数倍;

——如果报文 MSG 的长度是分组长度的整数倍, 不对数据作填充。

被加密数据首先要被格式化为以下形式的数据块:

- 明文数据的长度, 不包括填充字符
- 明文数据
- 填充字符(按上述填充方式)。

然后 MSG 被拆分为 16 字节的块 X_1, X_2, \dots, X_k 。

2) 密文计算

- ECB 模式

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, \dots, X_k 加密为 16 字节的块 Y_1, Y_2, \dots, Y_k 。

因此当 $i = 1, 2, \dots, k$ 时分别计算: $Y_i := \text{ALG}(K_s)[X_i]$ 。

- CBC 模式

用加密过程密钥 K_s 以 CBC 模式的分组加密算法将块 X_1, X_2, \dots, X_k 加密为 16 字节的块 Y_1, Y_2, \dots, Y_k ;

因此当 $i = 1, 2, \dots, k$ 时分别计算: $Y_i := \text{ALG}(K_s)[X_i \oplus Y_{i-1}]$;

Y_0 的初始值为:

$Y_0 := (\text{'00'} \parallel \text{'00'} \parallel \text{'00'} \parallel \text{'00'} \parallel \text{'00'} \parallel \text{'00'} \parallel \text{'00'} \parallel \text{'00'} \parallel \text{'00'} \parallel \text{'00'} \parallel \text{'00'} \parallel \text{'00'} \parallel \text{'00'} \parallel \text{'00'} \parallel \text{'00'} \parallel \text{'00'})$

记为: $Y := (Y_1 \parallel Y_2 \parallel \dots \parallel Y_k) = \text{ENC}(K_s)[\text{MSG}]$ 。

3) 解密过程

- ECB 模式:

当 $i = 1, 2, \dots, k$ 时分别计算: $X_i := \text{ALG}_{-1}(K_s)[Y_i]$ 。

- CBC 模式:

当 $i = 1, 2, \dots, k$ 时分别计算: $X_i := \text{ALG}_{-1}(K_s)[Y_i] \oplus Y_{i-1}$ 。

Y_0 的初始值为:

$Y_0 := (\text{'00'} \parallel \text{'00'} \parallel \text{'00'} \parallel \text{'00'} \parallel \text{'00'} \parallel \text{'00'} \parallel \text{'00'} \parallel \text{'00'} \parallel \text{'00'} \parallel \text{'00'} \parallel \text{'00'} \parallel \text{'00'} \parallel \text{'00'} \parallel \text{'00'} \parallel \text{'00'} \parallel \text{'00'})$ 。

为了得到原来的报文 MSG, 将块 X_1, X_2, \dots, X_k 连接起来, 如果使用了填充(见上文), 从最后一块 X_k 中删除尾部的(‘80’ || ‘00’ || ‘00’ || ... || ‘00’)。

记为: $\text{MSG} = \text{DEC}(K_s)[Y]$ 。

14.2.2 报文鉴别码

采用CBC模式的16字节分组加密算法以及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 ($4 \leq S \leq 8$) 值S的步骤如下:

1) 填充并分块

依据GB/T 16649.4对报文MSG进行填充, 因此在MSG的右端强制加上1个‘80’字节, 然后再在右端加上最少的‘00’字节, 使得结果报文的长度 $\text{MSG} := (\text{MSG} \parallel \text{'80'} \parallel \text{'00'} \parallel \text{'00'} \parallel \dots \parallel \text{'00'})$ 是16字节的整数倍。

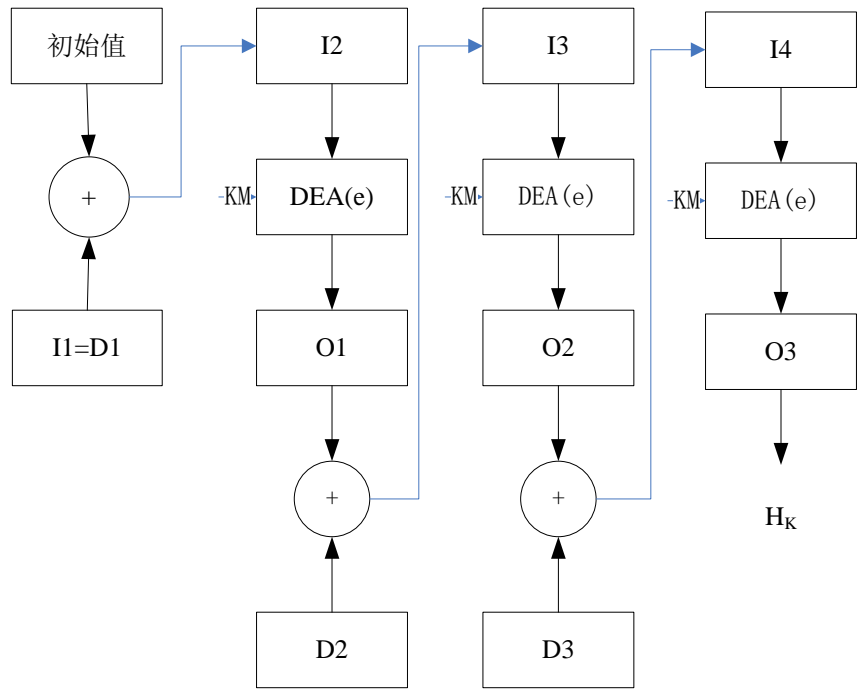
然后MSG被拆分为16字节的块 X_1, X_2, \dots, X_k 。

2) MAC 过程密钥

MAC过程密钥 K_s 长度为16字节。

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, \dots, X_K ：
 $H_i := \text{ALG}(K) [X_i \oplus H_{i-1}]$ ，这里 $i = 1, 2, \dots, K$ 。
 H_0 的初始值 $H_0 := (\text{'00'} \parallel \text{'00'} \parallel \text{'00'} \parallel \text{'00'} \parallel \text{'00'} \parallel \text{'00'} \parallel \text{'00'} \parallel \text{'00'} \parallel \text{'00'} \parallel \text{'00'} \parallel \text{'00'} \parallel \text{'00'})$ 。



说明：

I = 输入	D = X = 数据块
DEA(e) = 数据加密算法（加密模式）	KM = MAC过程密钥
O = 输出	+ = 异或

图19 使用 SM4 算法计算 HK 的算法

最终密文生成分以下两种情况：

- 在报文完整性及验证时，取 H_k 的前 8 字节作为 MAC 值。
- 计算应用密文（TC、ARQC 或 AAC）时，将 H_k 的左边 8 字节与右边 8 字节进行异或，得到 8 字节的密文： $H_{k+1} := H_{KL} \oplus H_{KR}$ 。

14.2.3 过程密钥产生

MAC 和数据加密过程密钥的产生如下所述：

第一步：卡片/发卡机构决定是使用 MAC 密钥还是数据加密密钥来进行所选择的算法处理。

第二步：将当前的 ATC 在其左边用十六进制数字‘0’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‘0’填充到 8 个字节记为数据源 B，将数据源 A 和数据源 B 串连，用选定的密钥对该数据作如图 10 所示的运算产生过程密钥。

$Z := \text{ALG}(\text{Key}) [\text{'00'} \parallel \text{'00'} \parallel \text{'00'} \parallel \text{'00'} \parallel \text{'00'} \parallel \text{'00'} \parallel \text{'00'} \parallel \text{ATC} \parallel \text{'00'} \parallel \text{'00'} \parallel \text{'00'} \parallel \text{'00'} \parallel \text{'00'} \parallel (\text{ATC} \oplus \text{'FFFF'})]$ 。

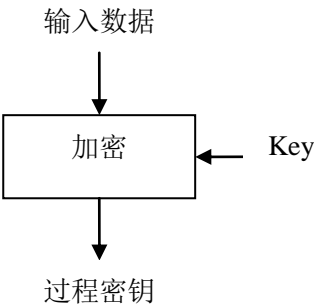


图20 过程密钥产生流程

14.2.4 子密钥分散

本节指定了一种利用一个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IC卡子密钥的方法。这一方式以主账号（PAN）和主账号序列号（如果主账号序列号不存在，则用一个字节‘00’代替）的最右16个数字及其衍生数据作为输入数据，以及16字节的发卡机构主密钥IMK作为密钥，生成16字节的IC卡子密钥MK作为输出：

- 1) 将主账号和主账号序列号连接生成数据块 X，如果 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 ‘0’ 以获得 8 字节的 Y。如果 X 的长度至少有 16 个数字，那么 Y 由 X 的最右边的 16 个数字组成。
 - 2) 计算：
 $Z:=\text{ALG}(\text{IMK})[Y|| (Y\oplus (‘\text{FF}’ || ‘\text{FF}’ || ‘\text{FF}’ || ‘\text{FF}’ || ‘\text{FF}’ || ‘\text{FF}’ || ‘\text{FF}’ || ‘\text{FF}’))]$ 。
- 16字节的IC卡子密钥 MK就等于Z。

图11是IC卡子密钥分散流程。

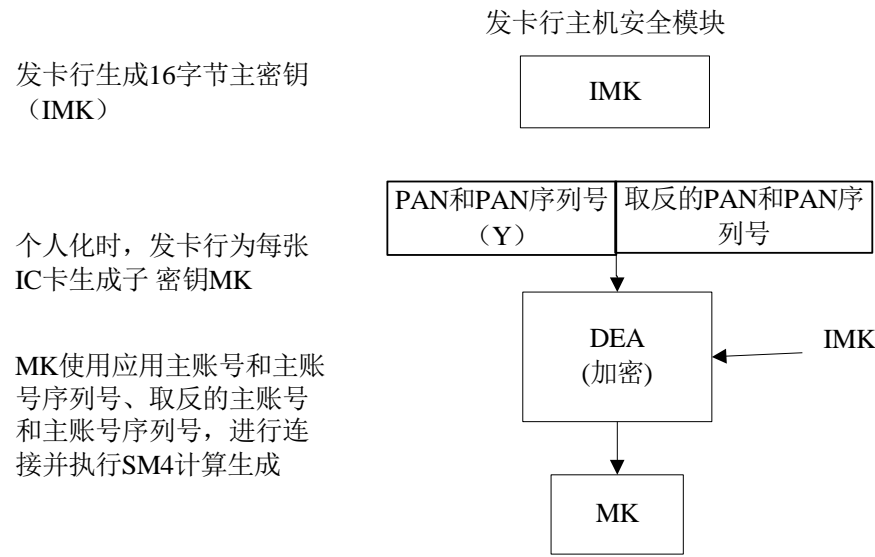


图21 子密钥分散

14.3 国际算法非对称加密机制

14.3.1 用于报文恢复的数字签名方案

本条描述了使用依照GB/T 27929规范的HASH函数的给定报文恢复数字签名方案，本规范的静态和动态数据认证都使用这一方案。

14.3.1.1 算法

数字签名方案使用下面两种算法：

——一个可逆的非对称算法，由一个依赖于私钥 S_K 的签名函数 $\text{Sign}(S_K)[\]$ 和一个依赖于公钥 P_K 的恢复函数 $\text{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：

$$\text{Recover}(P_K)[\text{Sign}(S_K)[X]] = X$$

——一个哈希算法 $\text{Hash}[\]$ ，将任意长度的报文映射为一个 20 字节的哈希值。

14.3.1.2 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 的过程如下：

- 1) 计算报文 M 的 20 字节的 HASH 值 $H: = \text{Hash}[\text{MSG}]$;
- 2) 将 MSG 拆分成两部分 $\text{MSG} = (\text{MSG1} \parallel \text{MSG2})$ ，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；
- 3) 定义一个字节 $B: = '6A'$;
- 4) 定义一个字节 $E: = 'BC'$;
- 5) 将 N 字节的块 X 定义为块 B ， MSG1 ， H 和 E 的连接，因此：

$$X: = (B \parallel \text{MSG1} \parallel H \parallel E);$$
- 6) 数字签名 S 被定义为 N 字节的数字：

$$S: = \text{Sign}(S_K)[X]。$$

14.3.1.3 数字签名验证

相应的签名验证过程如下：

- 1) 检查数字签名 S 是否由 N 个字节组成；
 - 2) 由数字签名 S 恢复得到 N 字节的数字 X 。

$$X = \text{Recover}(P_K)[S]。$$
 - 3) 将块 X 分割成 $X = (B \parallel \text{MSG1} \parallel H \parallel E)$ ，
 - B 为 1 字节长；
 - H 为 20 字节长；
 - E 为 1 字节长；
 - MSG1 由剩余的 $N-22$ 个字节组成。
 - 4) 检查字节 B 是否等于 $'6A'$ 。
 - 5) 检查字节 E 是否等于 $'BC'$ 。
 - 6) 计算 $\text{MSG} = (\text{MSG1} \parallel \text{MSG2})$ ，并检查是否满足 $H = \text{Hash}[\text{MSG}]$ 。
- 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。

14.4 国密算法非对称加密机制

本节描述SM2数字签名方案。

14.4.1 参数

SM2和RSA的一个重要不同是应用系统内需要统一的椭圆曲线参数,本规范使用中国国家密码管理局推荐的曲线参数进行数字签名签名,涉及到的参数包括:

- 一个大素数 p ;
- 大整数 a 和 b , 定义曲线方程 $y^2 = x^3 + ax + b \mod p$;
- 椭圆曲线的阶 n , 表示满足方程 $y^2 = x^3 + ax + b \mod p$ 的点的数量, 要求 n 为素数;
- 一个椭圆曲线上的点 $G = (G_x, G_y)$, 满足方程 $G_y^2 = G_x^3 + aG_x + b \mod p$, G 被称为基点, 通过基点可以生成椭圆曲线上的所有点。

14.4.2 密钥对

SM2密钥对包括私钥 S_k 和公钥 P_k 。

- S_k 是一个小于 n 的正整数, 使用随机数产生;
- $P_k = (x, y)$ 是椭圆曲线上的点, 即满足方程 $y^2 = x^3 + ax + b \mod p$, P_k 的长度为 p 的 2 倍。

14.4.3 算法

SM2签名方案使用下面三种函数:

- 一个依赖于私钥 S_k 的签名函数 $\text{Sign}(S_k)[M]$, 该函数输出两个相同长度的数字 r 和 s ;
- 一个依赖于公钥 P_k 的验证函数 $\text{Verify}(P_k)[M, \text{Sign}(S_k)[M]]$, 该函数输出 True 或 False, 表示验证正确或失败;
- 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。

14.4.4 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的过程如下:

- 1) 计算 $Z_A = H_{256}(\text{ENTL}_A \parallel \text{ID}_A \parallel a \parallel b \parallel x_G \parallel y_G \parallel x_A \parallel y_A)$ 。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;
- 2) 计算报文MSG的32字节的HASH值 $h := H_{256}[Z_A \parallel \text{MSG}]$;
- 3) 计算 $\text{Sign}(S_k)[h]$, 得到两个数字 r 和 s ;
- 4) 数字签名S被定义为 $S := r \parallel s$, 即数字签名S由数字 r 和 s 串联而成。

14.4.5 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:

- 1) 计算 $Z_A = H_{256}(\text{ENTL}_A \parallel \text{ID}_A \parallel a \parallel b \parallel x_G \parallel y_G \parallel x_A \parallel y_A)$ 。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;
 - 2) 计算报文MSG的32字节的HASH值 $h := H_{256}[Z_A \parallel \text{MSG}]$;
- $\text{Verify}(P_k)[h, S]$, 若函数输出True表示验证正确, 若输出False, 表示验证失败。

15 认可的算法

15.1 对称加密算法

15.1.1 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，算法的详细过程在GB/T 27929、GB/T 17964中定义。

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

$$Y=DES(K_L)[DES^{-1}(K_R)[DES(K_L)[X]]]$$

解密的方式如下：

$$X=DES^{-1}(K_L)[DES(K_R)[DES^{-1}(K_L)[Y]]]$$

单倍DES仅允许用于14.1.2.1条中指明的使用GB/T 27929中的算法3（3DES用于最后一个分组）的MAC机制。

15.1.2 SM4

SM4算法定义见GM/T 0002。

15.2 非对称加密算法

15.2.1 RSA

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相等。

表30 对模长字节数的强制上限

描述	最大长度
认证中心公钥模	248字节
发卡机构公钥模	248字节
IC卡公钥模	248字节

认证中心公钥模的长度 N_{CA} ，发卡机构公钥模的长度 N_I ，IC卡公钥模的长度 N_{IC} ，必须满足 $N_{IC} \leq N_I \leq N_{CA}$ 和 $N_{PE} \leq N_I \leq N_{CA}$ 。

注1：IC卡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IC卡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IC卡响应数据，也限制了IC卡公钥的最大长度。

注2：如卡片支持DDA和CDA，包含IC卡证书的记录模板的长度，即IC卡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IC卡公钥长度不超过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，IC卡公钥最大长度在205到240之间。如果GENERATE AC命令响应包含其他可选数据，IC卡公钥最大长度还应减去这些数据的长度（包括Tag和Length）。如果卡片应用支持INTERNAL AUTHENTICATION格式二，IC卡公钥最大长度还应减去7字节。

在选择公钥模长时，应该考虑到比较密钥生命周期同预期的因数分解进程。

发卡机构公钥指数和IC卡公钥指数的值由发卡机构决定。认证中心，发卡机构和IC卡公钥指数必须等于3或 $2^{16}+1$ 。

标识本数字签名算法的公钥算法标识必须编码为十六进制'01'。

使用奇数公钥指数的RSA算法的密钥及签名和恢复函数由下文详细说明。

15.2.1.1 密钥

使用奇数公钥指数e的RSA数字签名方案的私钥 S_k 由两个素数p和q，满足：

$p-1$, $q-1$ 与 e 互质,

以及私钥 d , 满足:

$$ed \equiv 1 \pmod{(p-1)(q-1)}。$$

组成相对应的公钥 P_k 由公钥模 $n=pq$ 和公钥指数 e 组成。

15.2.1.2 签名函数

使用奇数公钥指数的RSA签名函数被定义为:

$$S = \text{Sign}(S_k)[X]: = X^d \pmod{n}, 0 < X < n。$$

这里 X 是用于签名的数据, S 为对应的数字签名。

15.2.1.3 恢复函数

使用奇数公钥指数的RSA恢复函数被定义为:

$$X = \text{Recover}(P_k)[S]: = S^e \pmod{n}。$$

15.2.1.4 密钥的生成

城市公共交通IC卡系统与发卡机构必须对其各自的RSA公/私钥生成过程的安全性负责。

15.2.2 SM2

SM2算法是一种椭圆曲线公钥密码算法, 其密钥长度为256比特, 可参考GM/T 0003。

15.3 哈希算法

15.3.1 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16进制数'01'。

15.3.2 SM3

SM3算法是一种密码杂凑算法, 其输出为256比特, 可参考GM/T 0004。
