

中国金融集成电路（IC）卡

借记/贷记规范

第四部分：安全规范

中国金融集成电路（IC）卡标准修订工作组

二零零四年五月

目 次

1. 引言	1
2. 范围	1
3. 参考资料	1
4. 定义	1
5. 缩略语和符号	4
6. 脱机数据认证	6
6.1 密钥和证书	7
6.1.1 认证中心	7
6.1.2 公开密钥对	7
6.2 静态数据认证 (SDA)	8
6.2.1 密钥和证书	11
6.2.2 认证中心公钥获取	13
6.2.3 发卡行公钥获取	13
6.2.4 签名的静态应用数据验证	14
6.3 动态数据认证 (DDA)	15
6.3.1 密钥和证书	18
6.3.2 认证中心公钥的获取	20
6.3.3 发卡行公钥的获取	20
6.3.4 IC卡公钥的获取	22
6.3.5 标准动态数据认证	23
6.3.6 复合动态数据认证/应用密文生成(CDA)	25
7. 应用密文和发卡行认证	30
7.1 应用密文产生	30
7.1.1 数据源选择	30
7.1.2 应用密文算法	31
7.2 发卡行认证	31
7.3 密钥管理	31
8. 安全报文	31
8.1 报文格式	32
8.2 报文完整性及其验证	32
8.2.1 命令数据域	32
8.2.2 MAC过程密钥分散	32
8.2.3 MAC的计算	32
8.3 报文私密性	32
8.3.1 命令数据域	32
8.3.2 加密过程密钥分散	33
8.3.3 加密解密	33
8.4 密钥管理	33
9. 卡片安全	33
9.1 共存应用	33
9.2 密钥的独立性	33
9.3 卡片内部安全体系	33

9.3.1	卡片内部安全目标.....	33
9.3.2	卡片内部安全概述.....	33
9.3.3	文件控制信息.....	34
9.3.4	文件控制参数.....	36
9.3.5	IC卡本地数据建议访问条件.....	37
9.4	卡片中密钥的种类.....	37
10.	终端安全.....	38
10.1	终端数据安全性要求.....	38
10.1.1	一般要求.....	38
10.1.2	安全模块的物理安全要求.....	39
10.1.3	安全模块的逻辑安全要求.....	39
10.2	终端设备安全性要求.....	39
10.2.1	防入侵设备.....	39
10.2.2	PINPAD安全性.....	40
10.3	终端密钥管理要求.....	41
10.3.1	终端密钥种类.....	41
10.3.2	认证中心公钥管理.....	42
11.	密钥管理体系.....	43
11.1	认证中心公钥管理.....	43
11.1.1	认证中心公钥生命周期.....	44
11.1.2	认证中心公钥对泄漏.....	46
11.1.3	认证中心密钥管理策略.....	47
11.2	发卡行公钥管理.....	49
11.3	发卡行对称密钥管理.....	50
11.3.1	安全性要求.....	50
11.3.2	功能性要求.....	50
12.	安全机制.....	51
12.1	对称加密机制.....	51
12.1.1	加密解密.....	51
12.1.2	报文认证码.....	53
12.1.3	过程密钥分散.....	54
12.1.4	子密钥分散.....	56
12.2	非对称加密机制.....	57
12.2.1	用于报文恢复的数字签名方案.....	57
13.	认可的算法.....	58
13.1	对称加密算法.....	58
13.1.1	DES.....	58
13.1.2	SSF33.....	58
13.2	非对称加密算法.....	58
13.2.1	RSA.....	58
13.3	哈希算法.....	60
13.3.1	SHA-1.....	60

图 表

图表 6-1: SDA证书和公钥体系结构.....	10
图表 6-2: DDA证书和公钥体系结构.....	18
图表 11-1: 认证中心公钥的分发.....	45
图表 11-2: 发卡行公钥的分发.....	46
图表 12-1单长度过程密钥的产生.....	55
图表 12-2128位分组加密算法过程密钥的产生.....	56

表 格

表格 6-1: SDA和DDA的比较.....	6
表格 6-2: SDA, DDA和CDA处理优先级.....	7
表格 6-3: 由认证中心签名的发卡行公钥数据（即哈希算法的输入）.....	11
表格 6-4: 由发卡行签名的静态应用数据（即哈希算法的输入）.....	12
表格 6-5: 静态数据认证用到的数据对象.....	12
表格 6-6 从发卡行公钥证书恢复数据的格式.....	13
表格 6-7: 从签名的静态应用数据恢复数据的格式.....	14
表格 6-8: 由认证中心签名的发卡行公钥数据（即哈希算法的输入）.....	18
表格 6-9: 由发卡行签名的IC卡公钥数据（即哈希算法的输入）.....	19
表格 6-10: 动态认证中的公钥认证所需的数据对象.....	20
表格 6-11: 从发卡行公钥证书恢复数据的格式.....	20
表格 6-12: 从IC卡公钥证书恢复数据的格式.....	22
表格 6-13: 需签名的动态应用数据（即哈希算法的输入）.....	23
表格 6-14: 生成和检验动态签名所需要的其它数据对象.....	24
表格 6-15: 从签名的动态应用数据恢复的数据格式.....	24
表格 6-16: 需签名的动态应用数据（即哈希算法的输入）.....	26
表格 6-17: IC卡动态数据的内容.....	26
表格 6-18: 在CDA中GENERATE AC命令返回的数据对象.....	27
表格 6-19: 生成AAC时GENERATE AC命令返回的数据对象.....	27
表格 6-20: 发卡行应用数据.....	27
表格 6-21: 从签名动态应用数据恢复的数据的格式.....	28
表格 7-1: 建议的应用密文生成中使用的最小数据集.....	30
表格 7-2: 可选的应用密文生成数据源.....	31
表格 8-1: 以完整性和认证为目的的安全报文的命令数据域格式.....	32
表格 8-2: 以私密性为目的的安全报文的命令数据域格式.....	32
表格 9-1: 基本文件的访问条件.....	36
表格 9-2: 卡片上保存的密钥种类.....	37
表格 10-1: 终端内部保存的密钥种类.....	41
表格 10-2: 存储在终端中的认证中心公钥相关数据元的最小集.....	42
表格 11-1: 管理的对称密钥类型.....	50
表格 13-1: SSF33同DES的比较.....	58
表格 13-2: 对模长字节数的强制上限.....	59

1. 引言

本规范包含了中国金融集成电路（IC）卡借记贷记应用安全功能方面的要求，包括：IC卡脱机数据认证方法，IC卡和发卡行之间的通讯安全，以及相关的对称及非对称密钥的管理。

本文包括以下具体内容：

- 脱机数据认证
- 应用密文和发卡行认证
- 安全报文
- 卡片安全
- 终端安全
- 对称和非对称密钥管理体系

此外，本文还包括了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。

2. 范围

《中国金融集成电路（IC）卡借记/贷记应用安全规范》适用于由银行发行或受理的金融借记/贷记IC卡应用与安全有关的设备、卡片、终端机具及管理。其使用对象主要是与金融借记贷记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、集成和维护等部门（单位）。

3. 参考资料

EMV规范文档

- EMV 2000 Integrated Circuit Card Specification for Payment Systems, Version 4.0, Book 2, Security and Key Management

VIS规范文档

- VISA Integrated Circuit Card Application Overview , Version 1.4.0
- VISA Integrated Circuit Card Card Specification , Version 1.4.0
- VISA Integrated Circuit Card Terminal Specification , Version 1.4.0

中国集成电路（IC）卡文件

- 《中国金融集成电路(IC)卡规范》第1部分：卡片规范 （V1.0）
- 《中国金融集成电路(IC)卡规范》第2部分：应用规范 （V1.0）
- 《中国金融集成电路（IC）卡规范》第3部分：终端规范 （V1.0）

4. 定义

以下的术语用于本规范：

提前回收 - 在已公布的密钥失效日期到期前回收密钥。

应用 - 包括终端和卡片之间的应用协议及其相关的数据集合。

非对称加密技术 - 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）。这两种变换存在这样一种特性：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。

认证 - 确认一个实体所宣称的身份的措施。

字节 - 8个二进制位。

卡片 - 支付系统中定义的支付卡片。

证书 - 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。

证书回收 - 由发行证书的实体废除一个有效证书的过程。

认证中心 - 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。

密文 - 加密的信息。

命令 - 终端向IC卡发出的一个操作并要求返回应答的报文。

泄露 - 机密或安全被破坏。

串联 - 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。在每个字节中比特按由高到低的顺序排列。一组元素或对象可以按下面的方式连接：将第一对元素连接成新的元素，把它作为第一个元素再连接下一个元素，以此类推。

加密算法 - 为了隐藏或显现数据信息内容的变换算法。

密钥有效期 - 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。

数据完整性 - 数据没有被以未经授权的方式改变或者破坏的特性。

解密 - 对应加密过程的逆操作。

数字签名 - 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。

加密 - 基于某种加密算法对数据作可逆的变换从而生成密文的过程。

金融交易 - 在持卡人和商户或收单行之间发生的通过支付来换取货物或服务的行为。

哈希函数 - 将一个位串映射成固定长度的位串的函数，它满足以下两个特性：

- 给定一个输出，不可能通过计算得到与该输出对应的输入；
- 给定一个输入，不可能通过计算得到具有相同的输出的另一个输入。

另外，如果哈希函数要求防冲突，它还必须满足以下特性：

- 不可能通过计算找到两个不同的输入具有相同的输出。

哈希结果 - 哈希函数输出的位串。

集成电路 - 被设计用来完成处理和/或存储功能的电子器件。

集成电路卡 - 内部封装一个或多个集成电路，来完成处理和存储功能的卡片。

接口设备 – 终端上插入IC卡的部分，包括诸如机械和电气等相关设备。

密钥 – 加密转换中控制操作的一组符号。

密钥失效日期 – 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行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。

密钥导入 – 产生、分发和开始使用密钥对的过程。

密钥生命周期 – 密钥管理的所有阶段，包括计划、生成、回收、销毁和存档。

密钥更换 – 回收一个密钥，同时导入一个密钥来代替它。

密钥回收 – 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。

密钥回收日期 – 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。

密钥撤回 – 作为密钥回收的一部分，将密钥从服务中删除的过程。

逻辑泄露 – 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。

报文 – 由终端发送给卡片（或反之）的一串字节，不包括传输控制字符。

报文认证码 – 对数据的一种对称加密变换，为保护数据发送方发出和接收方收到的数据不被第三方伪造。

填充 – 向数据串某一边添加附加位。

密码键盘 – 用于输入个人密码的一组数字和命令按键。

明文 – 未加密的信息。

物理泄露 – 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。

计划回收 – 按照公布的密钥失效日期进行的密钥回收。

潜在泄露 – 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。

私钥 – 在一个实体使用的非对称密钥对中仅被该实体使用的密钥。在数字签名方案中，私钥定义了签名函数。

公钥 – 在一个实体使用的非对称密钥对中可以被公众使用的密钥。在数字签名方案中，公钥定义了验证函数。

公钥证书 – 由认证中心签名的不可伪造的某个实体的公钥信息。

响应 – IC卡接收到命令报文经过处理后返回给终端的报文。

对称加密技术 – 产生方和接受方使用同一个保密密钥进行转换的加密技术。如果不知道保密密钥，是无法通过计算得到产生方和接受方的转换信息的。

终端 – 在交易点安装的设备，和IC卡一起完成金融交易。它包括接口设备，也可以包括其它的部件和接口，例如和主机的通讯。

5. 缩略语和符号

下面为本规范用到的缩略语和符号：

AAC	应用认证密文
AC	应用密文
AFL	应用文件定位器
AID	应用标识符
AIP	应用交互特征
ADF	应用定义文件
APDU	应用协议数据单元
ARC	授权响应码
ARPC	授权响应密文
ARQC	授权请求密文
ATC	应用交易序号
ATM	自动柜员机
b	二进制
CBC	密码块链接
CDOL	卡片风险管理数据对象列表
CLA	命令报文的类别字节
cn	压缩数字
DDA	动态数据认证
DDOL	动态数据认证数据对象列表
DES	数据加密标准
ECB	电子密码本
EF	基本文件
FIPS	联邦信息处理标准
hex.	十六进制数
IC	集成电路
ICC	集成电路卡
IEC	国际电工委员会
IFD	接口设备
INS	命令报文的指令字节
K _M	主密钥

K_S	过程密钥
L_{DD}	IC卡动态数据长度
MAC	报文认证码
MMYY	月，年
N	数字
N_{CA}	认证中心公钥模长
N_I	发卡行公钥模长
N_{IC}	IC卡公钥模长
P1	参数1
P2	参数2
PAN	主帐号
P_{CA}	认证中心公钥
P_I	发卡行公钥
P_{IC}	IC卡公钥
PIN	个人鉴别码
SFI	短文件标识符
RID	注册的应用提供商标识
RSA	Rivest, Shamir, Adleman算法
S_{CA}	认证中心私钥
SDA	静态数据认证
S_I	发卡行私钥
S_{IC}	IC卡私钥
SHA	安全哈希算法
TC	交易证书
TLV	标签，长度，值
var.	变长

以下符号适用于本规范：

‘0’-‘9’和‘A’-‘F’	16 进制数字
$A := B$	A 被赋予数值 B
$A = B$	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 模 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 / n	A 整除 n，即：唯一的整数d，存在一个整数r， $0 \leq r < n$ ，使得
	$A = dn + r$
$Y := \text{ALG}(K)[X]$	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(在13.1节中有详细说明)
$X = \text{ALG}^{-1}(K)[Y]$	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(在13.1节有详细说明)
$Y := \text{Sign}(S_K)[X]$	用私钥 S_K ，通过非对称可逆算法，对数据块X进行签名(在13.2节中有详细说明)
$X = \text{Recover}(P_K)[Y]$	用公钥 P_K ，通过非对称可逆算法，对数据块Y进行恢复(在13.2节中有详细说明)
$C := (A \parallel B)$	将m位数字B和n位数字A进行链接，定义为： $C = 2^m A + B$
$H := \text{Hash}[\text{MSG}]$	用160位的HASH函数对任意长度的报文MSG进行HASH运算。

以下术语适用于本规范：

专用的	本规范未定义和/或在本规范范围之外的。
必须	表示一种强制性的要求。
应该	表示一种建议。
可选	表示可自行决定支持或不支持

6. 脱机数据认证

脱机数据认证是终端采用公钥技术来验证卡片数据的方法，脱机数据认证有2种形式：

- 静态数据认证（SDA）
- 动态数据认证（DDA）

在静态数据认证过程中，终端验证卡片上静态数据的合法性，SDA能确认卡片上的发卡行应用数据自卡片个人化后没有被非法篡改。

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行应用数据自卡片个人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卡支持的脱机数据认证方法。

脱机数据认证的结果影响到卡片和终端是执行脱机交易，联机授权还是拒绝交易。表格 6-1列出了SDA和DDA的比较。

表格 6-1： SDA和DDA的比较

	SDA	DDA
--	-----	-----

确认卡片数据未被篡改	是	是
防止卡片复制	否	是
要求卡片支持非对称加密算法	否	是
要求终端支持非对称加密算法	是	是
包含发卡行公钥证书	是	是
包含卡片公钥证书	否	是
公钥解密次数	2	3

脱机数据认证仅执行一种验证方式，三种脱机验证方式的优先级从高到低依次为：CDA，标准DDA，SDA。表格 6-2列出了卡片和终端支持不同脱机验证方式的情况下脱机验证的执行情况。

表格 6-2：SDA，DDA和CDA处理优先级

AIP指示卡片支持	终端支持SDA	终端支持SDA和标准DDA	终端支持SDA，标准DDA和CDA
SDA	SDA	SDA	SDA
SDA和标准DDA	SDA	标准DDA	标准DDA
SDA，标准DDA和CDA	SDA	标准DDA	CDA

6.1 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脱机数据认证。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。

6.1.1 认证中心

脱机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行公钥证书。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。

6.1.2 公开密钥对

认证中心和发卡行必须使用13.2节中指定的非对称算法产生认证中心公私钥对，发卡行公私钥对以及IC卡公私钥对。在本章中对脱机数据认证过程及相关数据元素的描述以RSA算法为例。

6.1.2.1 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。

认证中心公钥模长必须在13.2.1节中所定义的范围，认证中心公钥指数必须等于3或 2^{16+1} 。

6.1.2.2 发卡行公私钥对

支持SDA或DDA都需要发卡行产生发卡行公私钥对，并从认证中心获取发卡行公钥证书。发卡行将其公钥发送给认证中心，认证中心使用所有模长大于等于发卡行公钥模长并且公钥有效期晚于发卡行公钥有效期的认证中心私钥对其进行签名。

IC卡必须包含发卡行公钥证书及其用来验证发卡行证书的认证中心公钥索引，发卡行私钥由发卡行保管并保证其私密性和安全性。

发卡行公钥模长必须小于等于认证中心公钥最大模长，发卡行公钥模长必须在13.2.1节中所定义的范围。发卡行公钥指数必须等于3或 2^{16+1} ，

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，并用认证中心公钥从发卡行证书恢复发卡行公钥，然后用发卡行公钥恢复并验证卡片上的发卡行应用数据。

6.1.2.3 IC卡公私钥对

支持DDA还要求发卡行为每张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卡公钥由发卡行私钥签名，产生IC卡公钥证书并存储在卡片中。

IC卡公钥模长必须小于等于发卡行公钥模长，IC卡公钥模长必须在13.2.1节中所定义的范围。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，并用认证中公钥从发卡行公钥证书恢复发卡行公钥，然后用发卡行公钥从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，本规范中对脱机密文PIN不作要求，具体内容请参见EMV 2000 第二册 第7章。

6.2 静态数据认证（SDA）

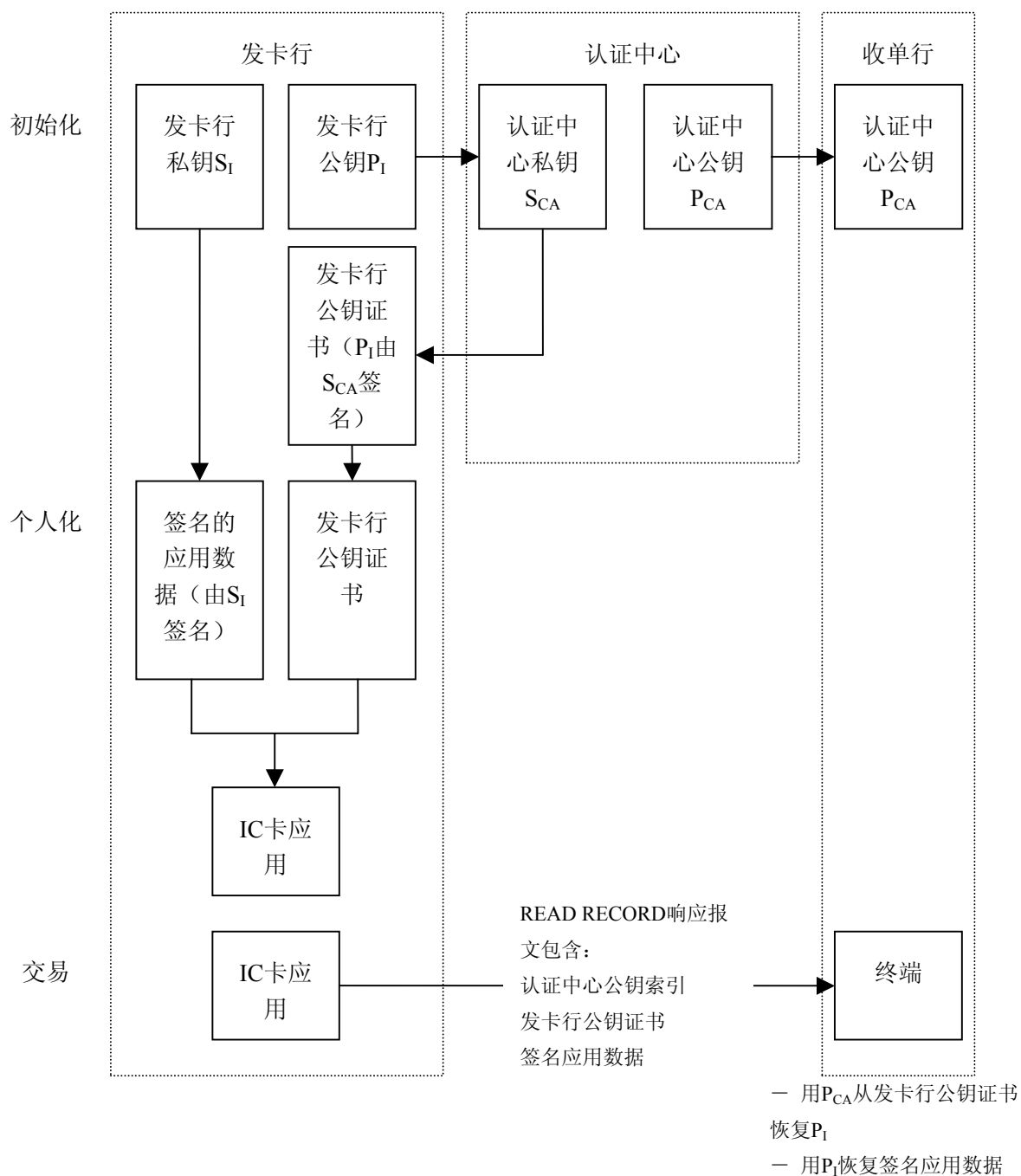
SDA的目的是确认存放在IC卡中的由应用文件定位器（AFL）和可选的静态数据认证标签列表所标识的，关键的静态数据的合法性，从而保证IC卡中的发卡行数据在个人化以后没有被非法篡改。

支持静态数据认证的IC卡个人化后应包含下列数据元素：

- 认证中心公钥索引：该单字节数据元素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对中的哪一个来验证IC卡。
- 发卡行公钥证书：该变长数据元素由认证中心提供给发卡行。当终端验证这个数据元素时，按6.2.3节描述的过程认证发卡行公钥和其它的数据。

- **签名的静态应用数据:**由发卡行使用同发卡行公钥证书所认证的发卡行公钥相对应的发卡行私钥产生的变长数据元素。它是一个对存放在IC卡中的关键静态数据元素的数字签名，在6.2.4节中有详细描述。
- **发卡行公钥的余项:**一个变长数据元素。它在IC卡中的存在是可选的。6.2.1节有进一步的解释。
- **发卡行公钥指数:**一个由发卡行提供的变长数据元素。6.2.1节有进一步的解释。

为了支持静态数据认证，每一台终端应该能为每个注册的应用提供商标识（RID）存储六个认证中心公钥，而且必须使得和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，并允许从一个算法过渡到另一个，参见10.3节）。在给定RID和IC卡提供的认证中心公钥索引的情况下，终端应能定位这样的公钥以及和公钥相关的信息。



图表 6-1：SDA证书和公钥体系结构

静态数据认证必须使用一种在12.2.1节和13.2节中指明的可逆算法。6.2.1节包含了对静态数据认证过程中涉及到的密钥和证书的概述，6.2.2节到6.2.4节详细说明了认证过程中主要的三个步骤，即：

- 由终端恢复认证中心公钥。
- 由终端恢复发卡行公钥。
- 由终端验证签名的静态应用数据。

6.2.1 密钥和证书

为了支持静态数据认证，一张IC卡必须包含签名的静态应用数据，它是用发卡行私钥签名的。发卡行公钥必须以公钥证书形式存放在IC卡中。

为了获得发卡行公钥证书，使用认证中心私钥 S_{CA} ，对表格 6-3中指定的数据应用12.2.1节中指定的签名方案。

认证中心的公钥对有一个公钥模，该公钥模为 N_{CA} 个字节。认证中心公钥指数必须等于3或 $2^{16}+1$ 。

为了获得签名的静态应用数据，使用发卡行私钥 S_I ，对表格 6-4中指定的数据应用12.2.1节中指定的签名方案。

发卡行的公钥对有一个发卡行公钥模，该公钥模为 N_I 个字节（ $N_I \leq N_{CA}$ ）。如果 $N_I > (N_{CA}-36)$ ，那么发卡行公钥模被分成两部分，即一部分包含公钥模中高位的 $N_{CA}-36$ 个字节（发卡行公钥中最左边的数字）；另一部分包含剩下的低位 $N_I - (N_{CA}-36)$ 个字节（发卡行公钥的余项）。发卡行公钥指数必须等于3或 $2^{16}+1$ 。

所有静态数据认证需要的信息在表格 6-5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，静态数据认证即告失败。

表格 6-3：由认证中心签名的发卡行公钥数据（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
发卡行标识	4	主帐号最左面的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
发卡行公钥算法标识	1	标识使用在发卡行公钥上的数字签名算法	b
发卡行公钥长度	1	标识发卡行公钥模的字节长度	b
发卡行公钥指数长度	1	标识发卡行公钥指数的字节长度	b
发卡行公钥数位或发卡行公钥的最左边部分	$N_{CA} - 36$	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’的字节的整个发卡行公钥。 如果 $N_I > N_{CA} - 36$ ，这个字段包含了发卡行公钥最高位的 $N_{CA} - 36$ 个字节	b

发卡行公钥余项	0 或 $N_I - N_{CA} + 36$	这个字段只有在 $N_I > N_{CA} - 36$ 时才出现。它包含了发卡行公钥最低位的 $N_I - N_{CA} + 36$ 个字节	b
发卡行公钥指数	1或3	发卡行公钥指数等于3或 $2^{16} + 1$	b

表格 6-4：由发卡行签名的静态应用数据（即哈希算法的输入）

字段名	长度	描述	格式
签名数据格式	1	十六进制，值为‘03’	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
数据验证代码	2	由发卡行分配的代码	b
填充字节	$N_I - 26$	填充字节由 $N_I - 26$ 个值为‘BB’的字节组成 ³	b
需认证的静态数据	变长	在中国金融集成电路（IC）卡借记贷记应用卡片规范10.3.1节指明的需认证的静态数据（参见下文）	—

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，则它必须仅包含标识AIP用的标签‘82’。

表格 6-5：静态数据认证用到的数据对象

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}	发卡行公钥证书	b
‘92’	$N_I - N_{CA} + 36$	发卡行公钥的余项（如果有）	b
‘9F32’	1或3	发卡行公钥指数	b
‘93’	N_I	签名的静态应用数据	b
—	变长	在中国金融集成电路（IC）卡借记贷记应用卡片规范10.3.1节指明的需认证的静态数据（参见上文）	—

6.2.2 认证中心公钥获取

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模、指数和与密钥相关的信息，以及相应的将使用的算法。如果终端没有存储与这个索引及RID相关联的密钥，那么静态数据认证失败。

6.2.3 发卡行公钥获取

1. 如果发卡行公钥证书的长度不同于在前面的过程中获得的认证中心公钥模长度，那么静态数据认证失败。
2. 为了获得在表格 6-6中指定的恢复数据，使用认证中心公钥和相应的算法按照12.2.1节中指定的恢复函数恢复发卡行公钥证书。如果恢复数据的结尾不等于‘BC’，那么静态数据认证失败。

表格 6-6 从发卡行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
发卡行标识	4	主帐号最左面的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
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
发卡行公钥算法标识	1	标识使用在发卡行公钥上的数字签名算法	B
发卡行公钥长度	1	标识发卡行公钥的模的字节长度	B
发卡行公钥指数长度	1	标识发卡行公钥指数的字节长度	B
发卡行公钥或发卡行公钥的最左边字节	$N_{CA}-36$	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’的字节的整个发卡行公钥。 如果 $N_i > N_{CA} - 36$ ，这个字段包含了发卡行公钥最高位的 $N_{CA} - 36$ 个字节	B
哈希结果	20	发卡行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
--------	---	-------------	---

- 检查恢复数据头。如果它不是‘6A’，那么静态数据认证失败。
- 检查证书格式。如果它不是‘02’，那么静态数据认证失败。
- 将表格 6-6中的第二个到第十个数据元素（即从证书格式直到发卡行公钥或发卡行公钥的最左边字节）从左到右连接，再把发卡行公钥的余项加在后面（如果有），最后是发卡行公钥指数。
- 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。
- 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，那么静态数据认证失败。
- 检验发卡行标识是否匹配主帐号最左面的3-8个数字（允许发卡行标识可能在其后补‘F’）。如果不一致，那么静态数据认证失败。
- 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，静态数据认证失败。
- 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，那么静态数据认证失败。
- 如果发卡行公钥算法标识无法识别，那么静态数据认证失败。
- 如果以上所有的检验都通过，连接发卡行公钥的最左边字节和发卡行公钥的余项（如果有）以得到发卡行公钥模，以继续下一步签名的静态应用数据的检验。

6.2.4 签名的静态应用数据验证

- 如果签名静态应用数据的长度不等于发卡行公钥模的长度，那么静态数据认证失败。
- 为了获得在表格 6-7中指明的恢复数据，使用发卡行公钥和相应的算法并将12.2.1节中指明的恢复函数应用到签名的静态应用数据上。如果恢复数据的结尾不等于‘BC’，那么静态数据认证失败。

表格 6-7：从签名的静态应用数据恢复数据的格式

字段名	长度	描述	格式
恢复数据头	1	十六进制，值为‘6A’	b
签名数据格式	1	十六进制，值为‘03’	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
数据验证代码	2	由发卡行分配的代码	b
填充字节	$N_t - 26$	填充字节由 $N_t - 26$ 个值为‘BB’的字节组成	b
哈希结果	20	需认证的静态应用数据的哈希值	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’，那么静态数据认证失败。
4. 检查签名数据格式。如果它不是‘03’，那么静态数据认证失败。
5. 将表格 6-7中的第二个到第五个数据元素（即从签名数据格式直到填充字节）从左到右连接，再把中国金融集成电路（IC）卡借记贷记应用卡片规范10.3.1节中指定的需认证的静态数据加在后面。如果静态数据认证标签列表存在，并且其包含非‘82’的标签，那么静态数据认证失败。
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，那么静态数据认证失败。
8. 如果以上所有的步骤都成功，那么静态数据认证成功。在表格 6-7中的恢复得到的数据验证代码应被存放在标签‘9F45’中。

6.3 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卡中的发卡行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素：

- 认证中心公钥索引：该单字节数据元素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对中的哪一个来验证IC卡。
- 发卡行公钥证书：该变长数据元素由相应的认证中心提供给发卡行。终端验证这个数据元素时，按6.3.3节描述的过程认证发卡行公钥和其它的数据。
- IC卡公钥证书：该变长数据元素由发卡行提供给IC卡。终端验证这个数据元素时，按6.3.4节描述的过程认证IC卡公钥和其它的数据。
- 发卡行公钥的余项：一个变长数据元素。6.3.1节有进一步的解释。
- 发卡行公钥指数：一个由发卡行提供的变长数据元素。6.3.1节有进一步的解释。

- IC卡公钥的余项：一个变长数据元素。6.3.1节有进一步的解释。
- IC卡公钥指数：一个由发卡行提供的变长数据元素。6.3.1节有进一步的解释。
- IC卡私钥：一个存放在IC卡内部的变长数据元素，用来按6.3.5节和6.3.6节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素：

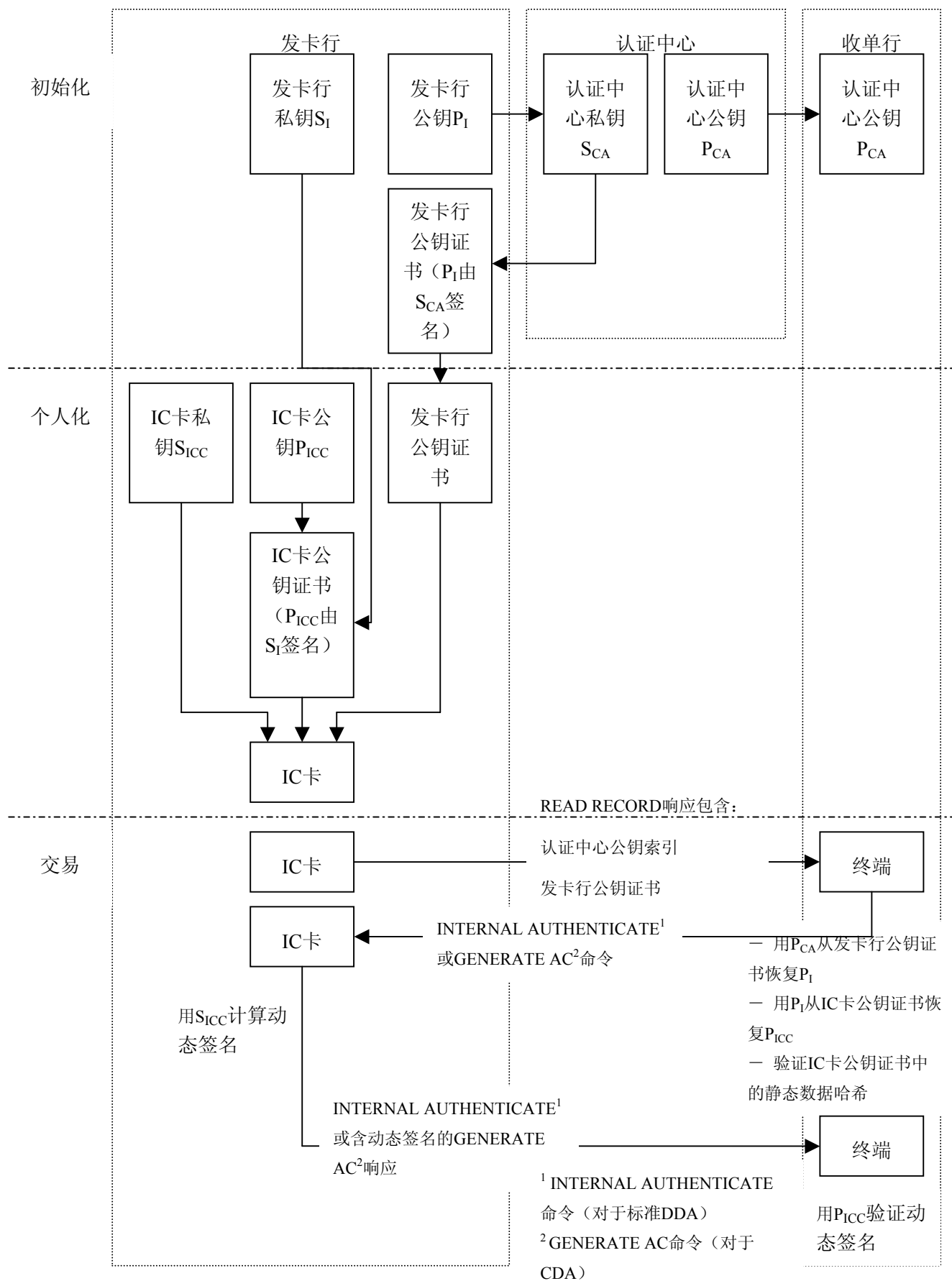
- 签名的动态应用数据：一个由IC卡使用同IC卡公钥证书所认证的IC卡公钥相对应的IC卡私钥生成的变长数据元素。它是一个数字签名，包含了6.3.5节和6.3.6节描述的存放在IC卡中的或由IC卡生成的以及终端中的关键数据元素。

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六个认证中心公钥，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，参见10.3节）。在给定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12.2.1节和13.2节中指明的可逆的算法。6.3.1节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，6.3.2节到6.3.4节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。
- ◆ 由终端恢复发卡行公钥。
- ◆ 由终端恢复IC卡公钥。

最后，6.3.5节和6.3.6节详细说明了两种情况下动态签名的生成和验证过程。



图表 6-2：DDA证书和公钥体系结构

6.3.1 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钥对，公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卡公钥由它的发卡行认证，而认证中心认证发卡行公钥。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12.2.1节中指明的签名方案分别将认证中心私钥 S_{CA} 应用到表格 6-8中指定的数据以及将发卡行私钥 S_I 应用到表格 6-9中指定的数据，以分别获得发卡行公钥证书和IC卡公钥证书。

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

发卡行的公钥对有一个为 N_I 个字节 ($N_I \leq N_{CA}$) 的发卡行公钥模。如果 $N_I > (N_{CA}-36)$,那么发卡行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（发卡行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（发卡行公钥余项）。发卡行公钥指数必须等于3或 $2^{16}+1$ 。

IC卡的公钥对有一个为 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格 6-10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。

表格 6-8：由认证中心签名的发卡行公钥数据（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
发卡行识别号	4	主帐号最左面的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
发卡行公钥算法标识	1	标识使用发卡行公钥的数字签名算法	b

发卡行公钥长度	1	标识发卡行公钥模的字节长度	b
发卡行公钥指数长度	1	标识发卡行公钥指数的字节长度	b
发卡行公钥或发卡行公钥的最左边字节	$N_{CA} - 36$	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’的字节的整个发卡行公钥。 如果 $N_I > N_{CA} - 36$ ，这个字段包含了发卡行公钥最高位的 $N_{CA} - 36$ 个字节	b
发卡行公钥的余项	0 或 $N_I - N_{CA} + 36$	这个字段只有在 $N_I > N_{CA} - 36$ 时才出现。它包含了发卡行公钥最低位的 $N_I - N_{CA} + 36$ 个字节	b
发卡行公钥指数	1或3	发卡行公钥指数等于3或 $2^{16} + 1$	b

表格 6-9：由发卡行签名的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
应用主帐号	10	主帐号（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	由发卡行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	在中国金融集成电路（IC）卡借记贷记应用卡片规范10.3.1节详细说明了需认证的静态数据（参见下文）	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’。

表格 6-10：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}	发卡行公钥证书	b
‘92’	$N_I - N_{CA} + 36$	发卡行公钥的余项（如果存在）	b
‘9F32’	1或3	发卡行公钥指数	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	在中国金融集成电路（IC）卡借记贷记应用卡片规范10.3.1节详细说明了需认证的静态数据（参见上文）	—

6.3.2 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。

6.3.3 发卡行公钥的获取

1. 如果发卡行公钥证书的长度不同于在前面的章节中获得的认证中心公钥模长度，那么动态数据认证失败。
2. 为了获得在表格 6-11中指定的恢复数据，使用认证中心公钥和相应的算法按照12.2.1节中指定的恢复函数恢复发卡行公钥证书。如果恢复数据的结尾不等于‘BC’，那么动态数据认证失败。

表格 6-11：从发卡行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
发卡行标识	4	主帐号最左面的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
发卡行公钥算法标识	1	标识使用在发卡行公钥上的数字签名算法	b
发卡行公钥长度	1	标识发卡行公钥的模的字节长度	b
发卡行公钥指数长度	1	标识发卡行公钥指数的字节长度	b
发卡行公钥或发卡行公钥的最左边字节	$N_{CA}-36$	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’的字节的整个发卡行公钥。 如果 $N_I > N_{CA} - 36$ ，这个字段包含了发卡行公钥最高位的 $N_{CA} - 36$ 个字节	b
哈希结果	20	发卡行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

- 检查恢复数据头。如果它不是‘6A’，那么动态数据认证失败。
- 检查证书格式。如果它不是‘02’，那么动态数据认证失败。
- 将表格 6-11中的第二个到第十个数据元素（即从证书格式直到发卡行公钥或发卡行公钥的最左边字节）从左到右连接，再把发卡行公钥的余项加在后面（如果有），最后是发卡行公钥指数。
- 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。
- 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。
- 检验发卡行识别号是否匹配主帐号最左面的3-8个数字（允许发卡行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。
- 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。
- 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. 如果发卡行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，连接发卡行公钥的最左边字节和发卡行公钥的余项（如果有）以得到发卡行公钥模，从而继续下一步取得IC卡公钥。

6.3.4 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节中获得的发卡行公钥模长度，那么动态数据认证失败。
2. 为了获得在表格 6-12中指定的恢复数据，使用发卡行公钥和相应的算法将12.2.1节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。

表格 6-12：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
应用主帐号	10	主帐号（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	由发卡行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
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. 将表格 6-12: 中的第二个到第十个数据元素（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，最后是中国金融集成电路（IC）卡借记贷记应用卡片规范10.3.1节指明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. 比较恢复得到的主帐号和从IC卡读出的应用主帐号是否相同。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）以得到发卡行公钥模，继续按下面章节的描述执行实际的动态数据认证。

6.3.5 标准动态数据认证

6.3.5.1 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指定的数据元素，这些数据元按中国金融集成电路（IC）卡借记贷记应用规范6.3节中指明的规则连接在一起。

IC卡可能包含DDOL，但终端应有一个缺省的，由支付系统指定的DDOL，以防在IC卡没有提供DDOL的情况下使用。

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）。

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。
2. IC卡使用IC卡私钥和相应的算法并按12.2.1节对表格 6-13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。

表格 6-13: 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ICC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）。在本规范中建议使用ATC作为IC卡动态数字。

除了表格 6-10中指明的数据，动态数据认证所需的数据对象在表格 6-14中详细说明。

表格 6-14：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

6.3.5.2 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. 为了获得在表格 6-15中指明的恢复数据，使用IC卡公钥和相应的算法将12.2.1节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。

表格 6-15：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. 将表格 6-15中的第二个到第六个数据元素（即从签名数据格式直到填充字节）从左到右连接，再把DDOL中指定的数据元素加在后面。
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。在表格 6-15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。

6.3.6 复合动态数据认证/应用密文生成(CDA)

6.3.6.1 动态签名的生成

假定：

- 终端已成功按上面讲述的过程取回了IC卡公钥。
- IC卡和终端都支持复合动态数据认证 / 应用密文生成，且IC卡在收到生成应用密文（GENERATE AC）命令时按照中国金融集成电路（IC）卡借记贷记应用卡片规范附录B.6给出对应的响应。

复合动态签名和应用密文生成按以下的步骤进行：

1. 终端根据中国金融集成电路（IC）卡借记贷记应用卡片规范附录B.6中的定义发出生成应用密文（GENERATE AC）命令。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）必须包含由终端生成的不可预知数（标签‘9F37’，4字节二进制数）。如果不是这样，那么复合动态数据认证 / 应用密文生成失败。
2. 如果IC卡将以TC或ARQC作为响应，则IC卡执行如下步骤：
 - 1) IC卡生成TC或ARQC；
 - 2) IC卡应用由哈希算法标识指示的哈希算法对从左到右连接的如下数据元进行运算：
 - ◆ 在第一个GENERATE AC命令情形下：
 - 由PDOL中指明，并按在其中出现的顺序，由终端在GET PROCESSING OPTIONS命令中发送给IC卡的数据元的值。
 - 由CDOL1中指明，并按在其中出现的顺序，由终端在第一个GENERATE AC命令中发送给IC卡的数据元的值。
 - IC卡在响应该GENERATE AC命令返回的数据元的标签、长度和值，根据它们返回的顺序且不包括签名动态应用数据。
 - ◆ 在第二个GENERATE AC命令情形下：

- 由PDOL中指明，并按在其中出现的顺序，由终端在GET PROCESSING OPTIONS命令中发送给IC卡的数据元的值。
- 由CDOL1中指明，并按在其中出现的顺序，由终端在第一个GENERATE AC命令中发送给IC卡的数据元的值。
- 由CDOL2中指明，并按在其中出现的顺序，由终端在第二个GENERATE AC命令中发送给IC卡的数据元的值。
- IC卡在响应该GENERATE AC命令返回的数据元的标签、长度和值，根据它们返回的顺序且不包括签名动态应用数据。

20字节的运算结果称作交易数据哈希值。

3) IC卡利用卡片中保存的IC卡私有密钥 S_{IC} 对表格 6-16中的数据运用12.2.1节定义的数字签名方案和相应算法，将结果称作签名的动态应用数据。

表格 6-16：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	标识用于产生交易数据哈希值和数字签名方案中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
不可预知数	4	由终端生成的不可预知数	b

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32-38个字节由表格 6-17中指明的数据连接而成。

表格 6-17：IC卡动态数据的内容

长度	值	格式
1	IC卡动态数字长度	b
2 – 8	IC卡动态数字	b
1	密文信息数据	b
8	TC或ARQC	b

20	交易数据哈希值	b
----	---------	---

IC卡动态数字是一个由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卡在交易时每收到一个生成应用密文（GENERATE AC）命令就加1的计数器）。在本规范中建议使用ATC作为IC卡动态数字。

IC卡对生成应用密文（GENERATE AC）命令的响应必须按照中国金融集成电路（IC）卡借记贷记应用卡片规范附录B.6 GENERATE AC命令响应报文数据域格式一节中指明的格式2（带有标签‘77’的结构数据对象）编码，且必须包含表格 6-18中指明的三个必须数据对象（在响应中按TLV编码），或可选包含发卡行应用数据。

表格 6-18：在CDA中GENERATE AC命令返回的数据对象

标签	长度	值	存在
‘9F27’	1	密文信息数据	必须
‘9F36’	2	应用交易序号	必须
‘9F4B’	N _{IC}	签名的动态应用数据	必须
‘9F10’	变长，最长32	发卡行应用数据	可选

- 如果IC卡以AAC作为响应，那么IC卡的响应必须按照中国金融集成电路（IC）卡借记贷记应用卡片规范附录B.6 GENERATE AC命令响应报文数据域格式一节中指明的格式1或格式2编码，且必须包含表格 6-19中指明的三个必须数据对象，可选包含发卡行应用数据。

表格 6-19：生成AAC时GENERATE AC命令返回的数据对象

标签	长度	值	存在
‘9F27’	1	密文信息数据	必须
‘9F36’	2	应用交易序号	必须
‘9F26’	8	应用认证密文	必须
‘9F10’	变长，最长32	发卡行应用数据	可选

如果存在发卡行应用数据（标签‘9F10’），应按照表格 6-20所示的格式编码

表格 6-20：发卡行应用数据

标签	长度	值	存在
	1	长度指示符	必须

	1	分散密钥索引	必须
	1	密文版本号	必须
	4	卡片验证结果（CVR）	必须
	1	算法标识	必须
	变长	发卡行自定义数据	可选

分散密钥索引指示IC卡产生应用密文所使用的是哪个密钥，密文版本号指示了应用密文的计算方式，7.1节描述了一种生成应用密文的方法，密文版本号和算法标识的定义请参考中国金融集成电路（IC）卡借记贷记应用卡片规范附录E。

6.3.6.2 动态签名的验证

如果IC卡以AAC响应，那么复合动态数据认证 / 应用密文生成（CDA）失败。

如果IC卡以TC或ARQC响应，那么终端从响应中取回表格 6-19中前面的四个数据对象并且执行以下步骤。

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，那么复合动态数据认证 / 应用密文生成失败。
2. 为了获得在表格 6-21中指明的恢复数据，使用IC卡公钥和相应的算法并将12.2.1节中指定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’，那么复合动态数据认证 / 应用密文生成失败。

表格 6-21：从签名动态应用数据恢复的数据的格式

字段名	长度	描述	格式
恢复数据头	1	十六进制，值为‘6A’	b
签名数据格式	1	十六进制，值为‘05’	b
哈希算法标识	1	标识用于产生交易数据哈希值和数字签名方案中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	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’，那么复合动态数据认证 / 应用密文生成失败。
4. 检查签名数据格式。如果它不是‘05’，那么复合动态数据认证 / 应用密文生成失败。
5. 从IC卡动态数据中取得表格 6-17中指明的数据。
6. 检查从IC卡动态数据中取得的密文信息数据是否等于从产生应用密文（GENERATE AC）命令的响应中获得的密文信息数据。如果不等，那么复合动态数据认证 / 应用密文生成失败。
7. 将表格 6-21中的第二个到第六个数据元素（即从签名数据格式直到填充字节）从左到右连接，再把不可预知数加在后面。
8. 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。
9. 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，那么复合动态数据认证 / 应用密文生成失败。
10. 将下列数据元从左到右连接：
 - ◆ 在第一个GENERATE AC命令情形下：
 - 由PDOL中指明，并按在其中出现的顺序，由终端在GET PROCESSING OPTIONS命令中发送给IC卡的数据元的值。
 - 由CDOL1中指明，并按在其中出现的顺序，由终端在第一个GENERATE AC命令中发送给IC卡的数据元的值。
 - IC卡在响应该GENERATE AC命令返回的数据元的标签、长度和值，根据它们返回的顺序且不包括签名动态应用数据。
 - ◆ 在第二个GENERATE AC命令情形下：
 - 由PDOL中指明，并按在其中出现的顺序，由终端在GET PROCESSING OPTIONS命令中发送给IC卡的数据元的值。
 - 由CDOL1中指明，并按在其中出现的顺序，由终端在第一个GENERATE AC命令中发送给IC卡的数据元的值。
 - 由CDOL2中指明，并按在其中出现的顺序，由终端在第二个GENERATE AC命令中发送给IC卡的数据元的值。
 - IC卡在响应该GENERATE AC命令返回的数据元的标签、长度和值，根据它们返回的顺序且不包括签名动态应用数据。
11. 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交易数据哈希值。
12. 把上一步计算得到的交易数据哈希值和步骤5中从IC卡动态数据中恢复出的交易数据哈希值相比较。如果它们不一样，那么复合动态数据认证 / 应用密文生成（CDA）失败。

如果以上所有的步骤都成功，那么复合动态数据认证 / 应用密文生成（CDA）成功。在表格 6-17 中恢复得到的IC卡动态数据中所包含的IC卡动态数字和ARQC或TC应被相应地存放在标签‘9F4C’和‘9F26’中。

7. 应用密文和发卡行认证

本章描述了IC卡生成应用密文（TC，ARQC或AAC），以及发卡行生成授权响应密文（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，请参见中国金融集成电路（IC）卡借记贷记应用卡片规范。

7.1 应用密文产生

7.1.1 数据源选择

一个应用密文是由基于以下数据生成的报文认证码组成的：

- 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请参见中国金融集成电路（IC）卡借记贷记应用卡片规范附录D.1，建议的最小数据集由表格 7-1详细说明。

表格 7-1：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
应用交易序号	IC卡

可选的应用密文生成数据源见表格 7-2。

表格 7-2：可选的应用密文生成数据源

值	来源
卡片验证结果	IC卡

7.1.2 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节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卡应用交易序号作为输入，分散得到十六字节的应用密文过程密钥SK_{AC}，使用12.1.3节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12.1.2节中指明的MAC算法应用到经选择的数据来生成8字节的应用密文。

7.2 发卡行认证

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7.1节）按照13.1节中指明的对称加密算法对8字节长的由IC卡按7.1节描述的方法生成的ARQC和2字节的授权响应代码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
 $X := (ARC \parallel '00' \parallel '00' \parallel '00' \parallel '00' \parallel '00' \parallel '00')$ 。
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

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

7.3 密钥管理

应用密文和发卡行认证的机制要求发卡行管理唯一的发卡行应用密文（AC）主密钥。IC卡应用密文（AC）子密钥的分散方法见12.1.4节。

8. 安全报文

安全报文通过报文认证码（MAC）来保障数据的完整性和对发卡行的认证，通过对数据域的加密来保障数据的机密性。

8.1 报文格式

本规范使用的报文格式参照《中国金融集成电路（IC）卡卡片规范》8.3.1节中的定义，报文格式同EMV 2000 第二册 9.2.1节定义的以完整性和验证为目的的安全报文格式二和9.3.1节定义的以私密性为目的的安全报文格式二一致。

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ISO/IEC 7816-4，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。

8.2 报文完整性及其验证

8.2.1 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素（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素，并且它的长度总是4字节。

表格 8-1：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）

8.2.2 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字节安全报文认证（MAC）过程密钥。在12.1.3节中详细说明了一种分散的方法。

8.2.3 MAC的计算

MAC是通过使用按照8.2.2节中描述的方法分散得到的MAC过程密钥并将12.1.2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。但总是包含了命令APDU（CLA INS P1 P2）的头部以及命令数据（如果存在）。

在本规范中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。

8.3 报文私密性

8.3.1 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。

表格 8-2：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)
-----------	------------

8.3.2 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12.1.3节中详细说明了一种这样的方法。

8.3.3 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8.3.2节中描述的方法分散得到的加密过程密钥并应用12.1.1中描述的机制进行的。

8.4 密钥管理

安全报文机制要求发卡行管理唯一的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12.1.4。

9. 卡片安全

9.1 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。

9.2 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。

9.3 卡片内部安全体系

本节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。

9.3.1 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。

9.3.2 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。

9.3.2.1 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GET PROCESSING OPTIONS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，因此发卡行就必须认真考虑怎样将数据对象与数据元素合并到文件当中。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，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。GET PROCESSING OPTIONS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GET PROCESSING OPTIONS指令的执行建立了安全域，发卡行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。其细节则在以下章节描述。

9.3.2.2 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（如更新记录命令）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文件控制参数一节提到。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。

9.3.3 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。

9.3.3.1 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。

9.3.3.1.1 安全域

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（如个人密码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（在数据资源小节内描述）
- 可执行代码资源（在可执行代码资源小节内描述）

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。

9.3.3.2 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

9.3.3.2.1 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。

9.3.3.2.2 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。

9.3.3.2.3 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。

9.3.3.3 可执行代码资源

可执行代码资源包括：

- 命令
- 加密算法

9.3.3.3.1 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。

9.3.3.3.2 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。

9.3.4 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。基本文件的访问如表格 9-1所示：

表格 9-1：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行脚本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。

9.3.5 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”）
 - ✓ 连续交易限制（国际-国家）
 - ✓ 连续交易限制（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

9.4 卡片中密钥的种类

在卡片中可能存在的密钥的种类有：

表格 9-2：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行认证	对称密钥	必须存在	由发卡行应用密文主密钥，按12.1.4节定义的分散方法获得，在交易过程中，按12.1.3节定义的方法派生过程密钥，用于应用密文产生和发卡行认证
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	由发卡行安全报文认证主密钥，按12.1.4节定义的分散方法获得，在交易过程

				中,按12.1.3节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	由发卡行安全报文加密主密钥,按12.1.4节定义的分散方法获得,在交易过程中,按12.1.3节定义的方法派生过程密钥,用于报文加密
卡片公私钥对	用于脱机数据认证	非对称密钥	卡片支持DDA或CDA	由发卡行私钥对卡片公钥及相关信息签名产生IC卡公钥证书

10. 终端安全

10.1 终端数据安全性要求

10.1.1 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。

10.1.1.1 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。

10.1.1.2 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。

10.1.2 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

10.1.3 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13章中所定义的对称算法和非对称算法，用于PINPAD到终端的用户PIN加密以及脱机数据认证。

10.2 终端设备安全性要求

10.2.1 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（关于对防入侵的设备的进一步要求请参阅ISO 13491）。

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。

10.2.1.1 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行监测到。

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。

10.2.1.2 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。

10.2.2 PINPAD安全性

PINPAD必须是一个防入侵的设备。它必须支持输入4-12个数字的PIN。如果PINPAD有显示屏，必须显示每一个输入的数字。但是依照ISO 9564-1，输入的PIN的值不应该显示或者不会被视觉或听觉的反馈方式泄露。

如果终端支持脱机PIN校验，则IC卡接口设备（IFD）和PINPAD要么被集成为单一的防入侵的设备，要么是两个分离的防入侵的设备。

- 如果IFD和PINPAD是集成的并且脱机PIN被以明文格式传递给卡片，那么在明文PIN被直接从PINPAD传到IFD的情况下，PINPAD不对脱机PIN进行加密。
- 如果IFD和PINPAD是集成的并且脱机PIN被以明文格式传递给卡片，但脱机明文PIN不是被直接从集成的PINPAD传到IFD，那么PINPAD必须依照ISO 9564-1（或相当的被支付系统批准的其它方式）对脱机PIN进行加密，再将其传递给IFD。IFD随后对脱机PIN解密，再以明文传递给卡。
- 如果IFD和PINPAD不是集成的并且脱机PIN以明文格式传递给卡片，那么PINPAD必须依照ISO 9564-1（或相当的被支付系统批准的其它方式）对脱机PIN进行加密，再将其传递给IFD。IFD随后对脱机PIN解密，再以明文传递给卡。

PIN的加密过程必须发生在下面两种其一的情况

如果终端支持联机PIN校验，当PIN被输入后，必须依照ISO 9564-1对PIN进行加密来保护PIN，并且对PIN的传输必须符合支付系统的规则。

显示在PINPAD上提示输入PIN的信息必须由PINPAD生成。这并不意味着只有和PIN相关的信息才能在PINPAD上显示，但其它的信息在显示前必须被PINPAD批准。PINPAD必须拒绝任何未经批准的的信息的显示。

对于有人值守的终端，金额输入过程必须和PIN输入过程分开，以避免意外地将PIN显示在终端的显示屏上。特别是如果在同一个键盘上输入金额和PIN，在允许输入PIN之前，金额必须先被持卡人确认。

PINPAD必须被设计为能提供隐私性和机密性，使得在正常的使用中，只有持卡人能够看到输入或显示的信息。PINPAD的安装和替换必须保证它的周边环境为持卡人输入PIN提供了足够的隐密性，从而将PIN暴露给他人的风险降到最低。

PINPAD必须在以下两种条件发生后自动清除内部缓存：

- 在交易结束后。
- 在超时的情况下，包括在一个PIN字符输入后过去了很长的时间的情况。

10.3 终端密钥管理要求

10.3.1 终端密钥种类

在终端中可能存在的密钥的种类有：

表格 10-1：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	用于脱机数据认证	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

10.3.1.1 认证中心公钥

用于进行脱机数据认证。

10.3.1.2 认证中心公钥维护密钥

对认证中心公钥的导入，更新和撤回必须使用基于12.1.2节描述的报文认证码，每台POS终端保存唯一认证中心公钥维护密钥，该密钥由收单行的认证中心公钥维护主密钥对终端序列号应用12.1.4节描述的子密钥分散方法获得的。

认证中心公钥维护密钥必须保存在安全模块中，安全性要求参见10.1节。

10.3.2 认证中心公钥管理

这一节规定了对收单行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。

10.3.2.1 认证中心公钥的导入

当一个支付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支付系统分发给每一个收单行。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行的责任。在将认证中心公钥及其校验和导入到安全模块的过程中，必须通过带报文认证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，本规范不作具体规定。

以下的原则适用于一个收单行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行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行。
- 收单行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。

10.3.2.2 认证中心公钥的储存和更新

支持静态和/或动态数据认证的终端必须对每个借记/贷记应用的RID提供6个认证中心公钥的支持，这些应用都基于EMV 2000支付系统规范(4.0版本)。

每一个认证中心公钥由5个字节的标识支付系统的RID和1个字节的对于每个RID唯一的、由该支付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，表格 10-2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，并将它和正确的支付系统联系起来。

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，即在本规范12.2.1节和13.2.1节中指明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，具体信息参见10.3.2.1节。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。

表格 10-2：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支付系统相关	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13.3节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

10.3.2.3 认证中心公钥的使用

交易中对认证中心公钥的使用必须遵循本规范。

10.3.2.4 认证中心公钥的回收

当支付系统已经决定撤回它的某一个认证中心公钥时，收单行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行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行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行。
- 收单行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，请参见EMV 2000第二册第十章。

11. 密钥管理体系

支付系统和发卡行都需要建立一套完整的密钥管理体系，支付系统需要建立认证中心，负责管理和使用用于脱机数据认证的认证中心公私钥对。发卡行需要建立一套完整的密钥管理体系，用于脱机数据认证和交易流程。

11.1 认证中心公钥管理

本节定义了支付系统中管理认证中心公钥的原则与策略的总体框架，认证中心公钥用于本规范所指定的静态和动态数据认证。

11.1.1 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）

11.1.1.1 计划

在计划阶段，支付系统调查和研究在不远的将来导入新的认证中心公钥对的需求。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。

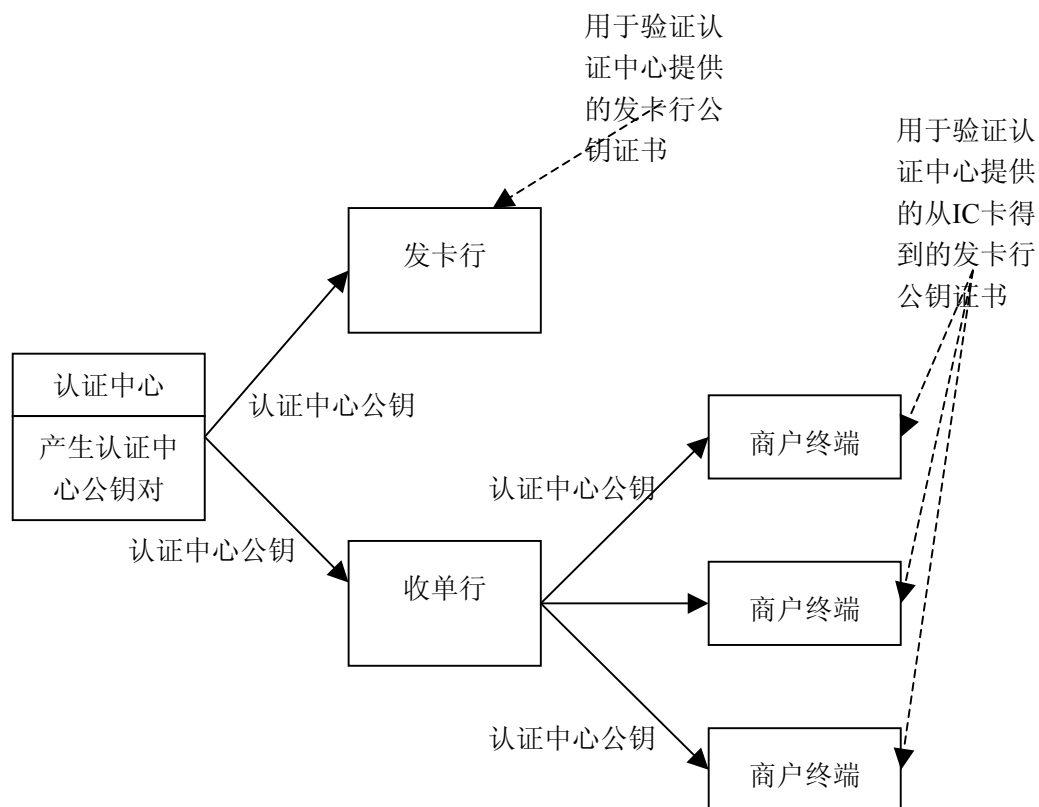
11.1.1.2 生成

如果计划阶段的结果需要导入新的认证中心公钥对，这些密钥必须由支付系统产生。更准确的说，支付系统认证中心（由支付系统运作的在物理和逻辑上都高度安全的组织）将以一种安全的方式来产生需要的认证中心私钥/公钥对，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。

11.1.1.3 分发

在密钥分发阶段，支付系统认证中心将新产生的认证中心公钥发布给它的成员发卡行和收单行作以下的用途：



图表 11-1：认证中心公钥的分发

- ◆ 对于发卡行，用于在使用阶段校验由认证中心提供的发卡行公钥证书。
- ◆ 对于收单行，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，支付系统认证中心，发卡行和收单行之间的接口必须保证认证中心公钥分发的完整性。

11.1.1.4 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。认证中心私钥被支付系统认证中心用于生成发卡行公钥证书。更准确地说，发生了下面的交互操作：



图表 11-2：发卡行公钥的分发

- 发卡行生成自己的发卡行公钥并发送给支付系统认证中心。
- 支付系统认证中心用认证中心私钥对发卡行公钥签名以生成发卡行公钥证书并将其发还给发卡行。
- 发卡行用认证中心公钥校验收到的发卡行公钥证书是否正确。如果正确，发卡行就可以将其作为IC卡的个人化数据的一部分。

为了防止导入假的发卡行公钥，支付系统认证中心和发卡行之间的接口必须保证提交的发卡行公钥的完整性。

11.1.1.5 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行公钥证书就不再有效了。因此发卡行必须保证用这样的发卡行公钥证书个人化的IC卡在认证中心公钥对的失效日期前中止使用。
- 在失效日期前的适当时候，支付系统认证中心应该停止用对应的认证中心私钥对发卡行公钥签名。

在失效日期之后，收单行应该在规定的期限内将认证中心公钥从终端中删除。

11.1.2 认证中心公钥对泄漏

当认证中心公钥对泄露时，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，密钥的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。

11.1.2.1 监测

认证中心公钥对的泄露可以是真正的泄露，例如已经过确认的，支付系统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支付系统被物理上非法闯入的察觉、由支付系统和它的会员安装的欺诈和风险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。

11.1.2.2 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支付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。

11.1.2.3 决策

根据评估所产生的结果，支付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。

11.1.2.4 回收（提前的）

决定回收认证中心公钥后，需要通知支付系统成员相应密钥的新的失效日期。之后的处理和11.1.1.5节中所描述的按计划的回收一样。

11.1.3 认证中心密钥管理策略

各阶段的定义及其密钥管理的原则和策略可以参考EMV 2000 第二册10.2节。下面针对各个阶段认证中心密钥管理策略实施工作进行说明。

11.1.3.1 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。

- 收单行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。
- 所有新的认证中心公钥都在12月31日以前发布
- 收单行有六个月的过渡期（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。
- 收单行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）。
- 新的认证中心公钥对从当年的7月1日开始生效。
- 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支付系统的责任

11.1.3.2 生成阶段

在这个阶段主要任务包括：

1. 认证中心以一种安全的方式来产生认证中心私钥/公钥对
2. 对于每一个注册的应用提供者标识符（RID），指向一个特定的认证中心公钥对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

11.1.3.3 分发阶段

在这个阶段主要任务包括：

1. 认证中心将新产生的认证中心公钥发布给它的成员发卡行和收单行
2. 收单行将这些公钥导入到商户终端中去

11.1.3.4 使用阶段

在这个阶段主要任务包括：

1. 为发卡行签发发卡行证书
 2. 发卡行用认证中心公钥校验收到的发卡行公钥证书是否正确，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行至少6个月后，认证中心才开始使用认证中心公钥对中的私钥进行签名。
- 发卡行所发行的用户IC卡的失效日期必须不晚于这张卡上的发卡行公钥证书的失效日期，也必须不晚于用来生成这个发卡行公钥证书的认证中心公钥对已公布（在卡片发行时）的回收日期。
- 一张发卡行公钥证书的失效日期必须不晚于用来生成这个发卡行公钥证书的认证中心公钥对已公布（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卡公钥证书的发卡行公钥的失效日期。

11.1.3.5 监测阶段

在这个阶段主要任务包括：

1. 对认证中心私钥安全性进行监控。私钥泄露形式包括物理的、逻辑的、可疑的、潜在的以及已确认的。

11.1.3.6 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。

11.1.3.7 决策阶段

本阶段只适用于提前回收。作为评估阶段的结果，该阶段支付系统决定对认证中心公钥对的泄露采取一系列的行动。

11.1.3.8 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。针对认证中心公钥对的情况，回收意味着私钥不再用来生成发卡行公钥证书
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。收单行有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。
- 遵循支付系统的规则，认证中心公钥对的回收需要在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，回收的日期根据支付系统的判断决定。

11.2 发卡行公钥管理

发卡行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，发卡行需要对密钥管理做一系列决策，它们包括：

- 需要产生的发卡行公钥数量。
- 产生的密钥的长度

发卡行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

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

11.3 发卡行对称密钥管理

11.3.1 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

11.3.2 功能性要求

11.3.2.1 密钥类型

系统管理的对称密钥种类如下表所示：

表格 11-1：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，用于安全报文认证码的产生和验证	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

发卡行主密钥可以分散出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安全校验码（MAC），用于安全报文命令，如数据安全更新、发卡行脚本等，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。

11.3.2.2 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

11.3.2.3 加密设备功能

系统设备必须能够实现以下功能：

1. 密钥分散功能，按照12.1.4节定义的分散方法，从保存在加密设备中的发卡行主密钥分散出唯一的IC卡子密钥
2. 过程密钥生成功能，按照12.1.3节定义的分散方法，根据子密钥和输入数据，分散出过程密钥
3. 数据加密功能，根据子密钥或过程密钥进行数据加、解密
4. 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
5. ARQC校验功能，根据交易数据校验ARQC的正确性
6. ARPC生成功能，根据交易数据产生ARPC

12. 安全机制

12.1 对称加密机制

12.1.1 加密解密

本规范中定义的加密解密机制参照《中国金融集成电路（IC）卡卡片规范》8.3.3节。

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模式作为本规范所使用的加密解密模式。

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 $\underline{MSG} := (MSG \parallel '80' \parallel '00' \parallel '00' \parallel \dots \parallel '00')$ 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：

- 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）

然后 \underline{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

$$\text{— 对应64位分组加密算法} \quad Y_0 := ('00' \parallel '00' \parallel '00' \parallel '00' \parallel '00' \parallel '00' \parallel '00' \parallel '00')$$

$$\text{— 对应128位分组加密算法} \quad Y_0 := ('00' \parallel '00' \parallel '00' \parallel '00' \parallel '00' \parallel '00' \parallel '00' \parallel '00' \parallel '00' \parallel '00' \parallel '00' \parallel '00')$$

记为:

$$Y := (Y_1 \parallel Y_2 \parallel \dots \parallel Y_K) = \text{ENC}(K_S)[\text{MSG}]。$$

解密过程如下:

1. 密文解密

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

$$\text{— 对应64位分组加密算法} \quad Y_0 := ('00' \parallel '00' \parallel '00' \parallel '00' \parallel '00' \parallel '00' \parallel '00' \parallel '00')$$

$$\text{— 对应128位分组加密算法} \quad Y_0 := ('00' \parallel '00' \parallel '00' \parallel '00' \parallel '00' \parallel '00' \parallel '00' \parallel '00' \parallel '00' \parallel '00' \parallel '00' \parallel '00')$$

2. 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：

$$\text{MSG} = \text{DEC}(\text{K}_S)[\text{Y}]。$$

12.1.2 报文认证码

12.1.2.1 基于64位分组加密算法的MAC计算方法

在本规范中，MAC计算方法参照《中国金融集成电路（IC）卡卡片规范》8.3.2节，MAC的长度s为4字节。

计算一个s字节的MAC（ $4 \leq s \leq 8$ ）是依照ISO/IEC 9797-1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KS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ISO/IEC 7816-4（等价于ISO/IEC 9797-1中的模式2）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 $K_S = (K_{SL} \parallel K_{SR})$ 。

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 ：

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

H_0 的初始值 $H_0 := (\text{'00'} \parallel \text{'00'} \parallel \text{'00'} \parallel \text{'00'} \parallel \text{'00'} \parallel \text{'00'} \parallel \text{'00'} \parallel \text{'00'})$ 。

用以下的两种方法的一种计算8字节的块 H_{K+1} 。

- 依照ISO/IEC 9797-1算法1： $H_{K+1} := H_K$ 。
- 依照ISO/IEC 9797-1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。

12.1.2.2 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ISO/IEC 7816-4（等价于ISO/IEC 9797-1中的模式2）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 :

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

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

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。

12.1.3 过程密钥分散

12.1.3.1 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: (在本节中统称为“过程密钥A”和“过程密钥B”)

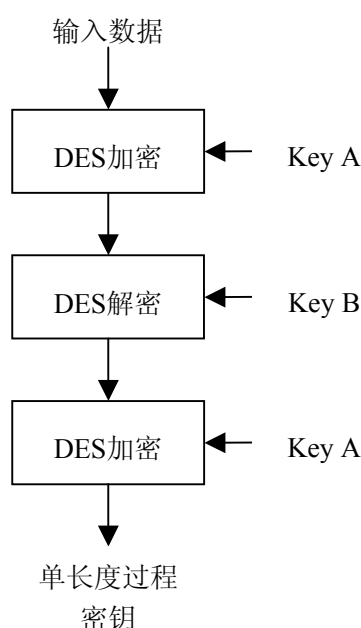
1. 单长度DES过程密钥

单长度DES过程密钥的分散方法参照《中国金融集成电路(IC)卡应用规范》附录B3定义的方法。

第一步: 卡片/发卡行决定是使用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“Key A”和“Key B”)

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Key A和Key B对该数据作如图表 12-1所示的3-DES运算产生过程密钥 A。

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



图表 12-1单长度过程密钥的产生

2. 双长度DES过程密钥

第一步：卡片/发卡行决定是使用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“Key A”和“Key B”)

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Key A和Key B对该数据作如图表 12-1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表 12-1所示的3-DES运算得到过程密钥 B。

$$Z_L := 3\text{-DES}(\text{Key})['00' \parallel '00' \parallel '00' \parallel '00' \parallel '00' \parallel '00' \parallel \text{ATC}]$$

$$Z_R := 3\text{-DES}(\text{Key})['00' \parallel '00' \parallel '00' \parallel '00' \parallel '00' \parallel '00' \parallel (\text{ATC} \oplus \text{'FFFF'})]$$

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。

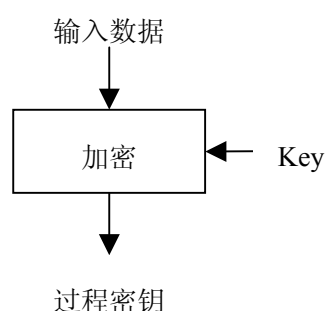
12.1.3.2 基于128位分组加密算法的过程密钥分散方法

MAC和数据加密过程密钥的产生如下所述：

第一步：卡片/发卡行决定是使用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表 12-2所示的运算产生过程密钥。

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



图表 12-2128位分组加密算法过程密钥的产生

12.1.4 子密钥分散

本节指定了一种利用一个16字节的发卡行主密钥IMK分散得出用于密文生成、发卡行认证和安全报文的IC卡子密钥的方法。

在本规范中，子密钥的分散算法参照《中国金融集成电路（IC）卡应用规范》附录B2定义的方法。

这一方式以主帐号（PAN）的最右16个数字作为输入数据，以及16字节的发卡行主密钥IMK作为输入，生成16字节的IC卡子密钥MK作为输出：

1. 如果主帐号X的长度小于16个数字，在右端填充十六进制的F以获得8字节数字格式的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

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

并定义

$$Z := (Z_L \parallel Z_R)$$

b) 基于128位分组加密算法的计算方法

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

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）。

12.2 非对称加密机制

12.2.1 用于报文恢复的数字签名方案

本节里描述了使用依照ISO/IEC 9796-2规范的HASH函数的给定报文恢复数字签名方案，本规范中的静态和动态数据认证都使用这一方案。

12.2.1.1 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，并且对于任何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。

12.2.1.2 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

$$S := \text{Sign}(S_K)[X]$$

12.2.1.3 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 $X = (B \parallel \text{MSG1} \parallel H \parallel E)$ ，这里
 - 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。

13. 认可的算法

13.1 对称加密算法

13.1.1 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ISO 8731-1、ISO 8732、ISO/IEC 10116中定义。

3-DES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：

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

解密的方式如下：

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

单倍DES仅允许用于12.1.2.1节中指明的使用ISO 9797-1中的算法3 (3DES用于最后一个分组)的MAC机制。

13.1.2 SSF33

SSF33算法是以128位分组为单位进行运算，密钥长度为16字节，该算法也可以被用于安全报文传送和MAC机制密文运算。

表格 13-1：SSF33同DES的比较

	DES	3-DES	SSF33
密钥长度	8字节	16字节	16字节
分组长度	8字节	8字节	16字节

使用SSF33算法和基于3-DES的对称加密机制使用相同长度的密钥，能够同《中国金融集成电路(IC)卡规范》原有的基于3-DES的密钥管理兼容，其区别在于分组长度不同，在加密，计算MAC和密钥分散时填充和计算方式不同，但报文认证码和密钥分散输出结果的长度同3-DES算法保持一致。

13.2 非对称加密算法

13.2.1 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。

表格 13-2：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
发卡行公钥模	248字节
IC卡公钥模	248字节

认证中心公钥模的长度 N_{CA} ，发卡行公钥模的长度 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 $N_{PE} \leq N_I \leq N_{CA}$ 。

在选择公钥模长时，应该考虑到比较密钥的生命周期同预期的因数分解进程。

发卡行公钥指数和IC卡公钥指数的值由发卡行决定。认证中心，发卡行和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。

13.2.1.1 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，

以及私钥指数 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。

13.2.1.2 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。

13.2.1.3 恢复函数

使用奇数公钥指数的RSA恢复函数被定义为

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

13.2.1.4 密钥的生成

支付系统与发卡行必须对其各自的RSA公/私钥生成过程的安全性负责。

13.3 哈希算法

13.3.1 SHA-1

该算法在FIPS 1801 中被标准化。SHA-1对任意长度的报文的输入，产生一个20字节的哈希值。
本哈希算法的标志编码为16进制数'01'。