Q/CUP

中 国 银 联 股 份 有 限 公 司 企 业 标 准

Q/CUP 042.4—2011
代替Q/CUP 031-2008

中国金融集成电路（IC）卡借记/贷记应用发卡行及应用安全指南

China financial integrated circuit card debit/credit application-
Issuer and Application Security Guide

（报批稿）

中国银联股份有限公司 发布

目　次

前　言

本标准在编写过程中主要依据《中国金融集成电路（IC）卡规范》（JR/T0025－2005）借记贷记应用。

本标准给出了符合《中国金融集成电路（IC）卡规范》借记贷记应用的发卡行及应用安全指南，供成员发卡行实施PBOC迁移时参考使用。

本标准由中国银联股份有限公司提出。

本标准由中国银联股份有限公司技术管理部组织制定。

本标准的主要起草单位：中国银联IC卡应用部。

本标准的主要起草人：刘先、徐晋耀、李春欢。

中国集成电路（IC）卡借记/贷记应用发卡行及应用安全指南

1 范围

本指南的编写目的是为发卡行实施 PBOC 迁移计划提供一个安全指南。它引述其它规范性文档的专业信息，或提供这些文档的索引信息。

2 规范性引用文件

下列文件中的条款通过本标准的引用而成为本标准的条款。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标准，然而，鼓励根据本标准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标准。

JR/T 0025.4-2005 中国金融集成电路（IC）卡规范第4部分：借记/贷记应用规范

JR/T 0025.5-2005 中国金融集成电路（IC）卡规范第5部分：借记/贷记卡片规范

JR/T 0025.6-2005 中国金融集成电路（IC）卡规范第6部分：借记/贷记终端规范

JR/T 0025.7-2005 中国金融集成电路（IC）卡规范第7部分：借记/贷记安全规范

JR/T 0025.10-2005 中国金融集成电路（IC）卡规范第10部分：借记/贷记应用个人化指南规范

Q/CUP 018.1—2006 金融IC 卡借记/贷记应用根CA公钥认证规范

EMV 4.1 Integrated Circuit Card Specifications for Payment Systems, Books 1 to 4

3 术语和定义

本标准采用下列术语和定义：

3.1 中国金融集成电路（IC）卡规范 （2005 版）

系中华人民共和国金融行业标准之一，由中国人民银行起草并于 2005 年 3 月 10 日发布/实施，用来规范金融行业集成电路（IC）卡应用的规范。包含 10 个部分，涵盖电子钱包/电子存折应用和借记/贷记应用，本文只涉及借记/贷记应用，以下简称为“PBOC2.0”。

3.2 金融 IC 卡借记/贷记应用根 CA(Financial IC Card Debit/Credit Applications Root CA)

由中国人民银行授权建立的、由中国银联统一管理的服务于金融行业 IC 卡安全应用的根认证中心，以下简称“IC 卡根 CA”。实现该根认证中心功能的应用系统是“金融 IC 卡借记/贷记应用根 CA 系统”，以下简称“IC 卡根 CA 系统”。

3.3 认证 Authentication

用于确认数据真实性和完整性的密码过程。

3.4 公钥证书 Public Key Certificate

由认证中心利用其私钥签名的包含一个公钥、实体身份和其它信息的公钥证书，该证书具有不可篡改性。

3.5 证书认证机构（CA）Certification Authority

由其它多个实体信任的机构，用于为其它实体生成和签发公钥证书。

3.6 密码算法 Cryptographic Algorithm

用于认证数据或保护数据的规则及过程，比如对数据的加解密。该算法具有下述特征：除了穷举搜索，不可能确定任何相关秘密控制参数，即密钥或私钥。

3.7 哈希函数 Hash Function

将数据从大的空间映射到小的空间的函数。该函数满足下列特性：

1. 从函数的已知输出找到能够产生该输出的函数输入在计算上是不可行的。
2. 从函数的已知输出找到能够产生该输出的函数的第二个输入在计算上是不可行的。

3.8 密钥周期 Cryptoperiod

一个时间周期，在该周期内一个特定密钥被授权使用，或在该周期内用于特定系统的密钥是有效的。

3.9 数字签名 Digital Signature

一种对于数据的密码变换，该变换若正确实现，能够提供下列功能：

- 信息源认证
- 数据完整性保护
- 签名者无法抵赖

3.10 双重控制 Dual Control

使用两个或多个实体来保护敏感信息或功能，使得任何单个实体无法访问或使用该信息或功能。

3.11 哈希值 Hash Value

对一则信息应用哈希函数后的结果。

3.12 集成电路卡(IC 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

3.13 密钥组件 Key Component

具有随机性特性的至少两个参数中的一个，能由一或多种参数通过组合构成密钥。

3.14 密钥对 Key Pair

公钥密码技术中，公钥和其对应的私钥。

3.15 密钥材料 Key Material

用于建立和维持密钥的数据（即：密钥、证书、初始向量等）。

3.16 支付系统 Payment System

一个支付系统包括若干参与方，这里发卡行和收单行根据支付系统规则和相关的风险评估将部分职责委托给不同的实体。

3.17 物理安全器件 Physically Secure Device

具有可忽略的不被探测到的对器件进行非授权访问和更改器件内数据内容概率的模块。

3.18 私钥 Private Key

在非对称密码算法（公钥）系统中，实体密钥对中仅由该实体知道的密钥。这里的私钥与对称密码算法中的秘密密钥不同。

3.19 伪随机 Pseudo-random

指一个过程，其输出在统计上是随机和不可预测的，尽管其输出由一个确定的算法过程产生。

3.20 公钥 Public Key

在非对称密钥系统中，可以公开的实体密钥。

3.21 秘密密钥 Secret Key

用于对称密码算法的密钥，该密钥的泄漏会导致整个对称密码系统的失密。秘密密钥不同于非对称密钥对中的私钥。

3.22 安全密码器件 Secure Cryptographic Device

对诸如密钥等秘密信息提供安全存储的器件，并基于这些秘密信息提供安全服务。

3.23 签名密钥 Signature Key

见私钥。

3.24 知识分割 Split Knowledge

一种两个或多个实体存在的情形，实体如密钥管理者，分别秘密地掌握一个密钥的构成的部分信息，这些密钥管理者通过合作可以构造或恢复整个密钥，但是任何个人无法获取该密钥。

3.25 验证密钥 Verification Key

见公钥。

4 符号和缩略语

以下缩略语和符号表示适用于本规范：

AAC	应用认证密文(Application Authentication Cryptogram)
AC	应用密文(Application Cryptogram)
AIP	应用交互特征(Application Interchange Profile)
ARPC	授权响应密文(Authorization Response Cryptogram)
ARQC	授权请求密文(Authorization Request Cryptogram)
ATC	应用交易序号(Application Transaction Counter)
ATM	自动柜员机(Automated Teller Machine)
CA	证书认证机构(Certificate Authority)
CDA	复合动态数据认证/应用密文生成(Combined DDA/AC Generation)
CDOL	卡片风险管理数据对象列表（Card Rish Management Data Object List）
CVM	持卡人验证方法（Cardholder Verification Method）
DDA	动态数据认证(Dynamic Data Authentication)
DES	数据加密标准算法（Data Encryption Standard）
EMV	Europay MasterCard VISA
HSM	硬件安全模块(Hardware Security Module)
IAD	发卡行应用数据(Issuer Application Data)
IC	集成电路(Integrated Circuit)
IC 卡	集成电路卡(Integrated Circuit Card)
IMK_{AC}	应用密文发卡行主密钥(Issuer Master Key for Application Cryptogram)
IMK_{SMC}	安全报文加密发卡行主密钥(Issuer Master Key for Secure Messaging for Confidentiality)
IMK_{SMI}	安全报文完整性发卡行主密钥(Issuer Master Key for Secure Messaging for Integrity)
MAC	报文认证码(Message Authentication Code)
MK	主密钥(Master Key)
MK_{AC}	应用密文卡片主密钥(Card Master Key for Application Cryptogram compution)
MK_{SMC}	安全报文加密卡片主密钥(Card Master Key for Secure Messaging for Confidentiality)
MK_{SMI}	安全报文完整性卡片主密钥(Card Master Key for Secure Messaging for Integrity)
PAN	应用主账号(Application Primary Account Number)
PDOL	处理选项数据对象列表(Processing Options Data Object List)
POS	销售点终端(Point of Sale)
PBOC	中国人民银行(People’s Bank of China)
RSA	由 Rivest,Shamir 和 Adleman 发明的非对称密码算法（Rivest,Shamir，Adleman Algorithm）
SCD	安全密码器件(Secure Cryptographic Device)
SDA	静态数据认证(Session Key for Application Cryptogram compution)
SK_{AC}	应用密文过程密钥(Session Key for Application Cryptogram compution)
SK_{SMC}	安全报文加密过程密钥（Session Key for Secure Messaging for Confidentiality）
SK_{SMI}	安全报文完整性过程密钥（Session Key for Secure Messaging for Integrity）

TC	交易证书（Transaction Certificate）
3DES	双倍密钥长度的三重 DES 算法（Two Key Triple DES）
TVR	终端验证结果(Terminal Verification Results)

5 PBOC2.0 规范借贷记应用与密码技术－概述

5.1 介绍

本概述的目的为发卡行安全指南提供一个框架介绍。同时介绍PBOC2.0借贷记应用支付系统模型以及在该框架内不同实体的角色。此外也介绍了与这个支付框架相关的密码技术基础。

5.2 支付系统模型

PBOC2.0借贷记应用支付系统包含下列类型的实体：

- 持卡人，
- 商户，
- 发卡行，
- 收单行，
- 支付系统组织（如银联、Visa等）。

在支付系统模型中每一个实体的角色如下。

5.2.1 持卡人

持卡人的角色包括下列内容：

- 通过与发卡行签署协议从发卡行获取一张包含PBOC2.0借贷记支付产品应用的银联IC卡。
- 选择、记住或（可能的）更新持卡人的PIN。
- 将其银联IC卡插入或通过收银员插入接收银联IC卡的PBOC2.0借贷记应用支付的终端(ATM、POS等）。

5.2.2 商户

商户角色包括：

- 与收单行签署协议使得其终端为收单行IC卡支付产品和支付方案所接受，以对IC卡进行收单服务。
- 接受包含支付产品的IC卡进行交易。
- 通过其终端收集购物交易信息并传递给发卡行获得购物款。

5.2.3 发卡行

发卡行角色包括：

- 与持卡人签署协议，完成卡片个人化并向持卡人颁发一张包含PBOC2.0借贷记支付应用的银联IC卡。这也包括生成和向卡片内安装所需的密钥以支持支付应用。
- 处理卡片联机交易。包括验证卡片发来的数据和密码报文，并生成密码报文以便卡片对发卡行进行认证。同时也包括在交易授权过程中的持卡人联机PIN的验证。
- 在需要时生成卡片应用更新脚本。
- 向收单行交付卡片交易金额。
- 向其它实体安全传递所需的密钥以支持支付系统的正常运行。

5.2.4 收单行

收单行的角色如下：

- 与商户签署协议并部署该商户的支付终端。这包括在这些终端内安装并管理IC卡根CA的公钥，同时确保终端中的IC卡公钥的数据完整性。
- 处理卡片支付交易并向商户支付交易金额。
- 向发卡行传递收集的交易数据以便得到这些交易的结算。

- 设定脱机、联机交易接收条件并管理相应风险。

5.2.5 支付系统

支付系统的角色包括：

- 对产品及服务制定系统规则并确认这些规则得到遵守。
- 生成并发布IC卡根CA公钥。
- 为发卡行签发用于发行PBOC2.0借贷记应用IC卡的发卡行公钥证书。
- 运行用于PBOC2.0卡片交易所需的发卡行与收单行间的支付通讯网络。
- 在上述支付网内为交易处理清算和结算。

5.3 密码技术基础

密码技术是IC卡应用中广泛使用的技术。历史上，密码技术用于数据加密。在今天密码技术应用除了数据加密外还包括数据完整性、认证和抗抵赖等其它密码服务。在这些不同的密码服务增加的同时，对实现不同厂商及不同密码产品和服务的互操作的技术标准显得日益重要。本文叙述了这些技术标准的主要方面。

本文下面设立的密钥管理原则可应用于使用密钥的卡片个人化过程、认证协议的管理以及交易的清算等各种环境。这些准则基于相关密码技术标准。

当代密码技术基于两个组成部分：密码算法和密钥。密码算法定义了如何从明文得到密文或者反之（若算法是加密算法），以及定义了如何从数据得到数字签名，并且数字签名如何被验证（若算法是数字签名算法）。通常表示为一个比特序列的密钥用于密码算法的输入，并确保对算法的知识无法使未授权方得以解密敏感数据或假冒他人的数字签名。

当代密码技术的安全取决于在算法公开的情况下对密钥的私密性保护及安全管理。密码算法通常公布于众，并常常由广泛的密码学专业人士深入分析研究。例如DES算法已经以各种标准及其它文件形式发表，RSA算法基于广泛熟知的数学原理。这些不同密码算法的操作安全完全取决于密钥和私钥在其整个生命周期内是如何管理的。这样，每个用户秘密掌握的唯一密钥确保了非授权方无法解密敏感数据或无法伪造他人的数字签名。

通常有两类密码算法：对称（或秘密）密钥算法、非对称（或公/私钥）密码算法。对称算法和非对称算法都应用于PBOC2.0规范借贷记应用中。

5.3.1 对称算法

对称（或秘密）密钥算法要求相同的秘密密钥既用于加密过程也用于解密过程。这样，算法的安全性完全取决于对该秘密密钥的保护。

PBOC2.0借贷记应用支持DES算法的使用，该算法在今天被广泛用于金融和工业界。DES在密码算法中属于分组密码的一类，因为该类算法以分组的方式处理数据。DES算法的输入是64比特的数据组，DES以16轮迭代方式使用一个56比特密钥将其输入数据组转换成64比特的输出数据组。

本文所述的3DES算法实际是使用两个DES的56比特密钥进行三轮DES运算。首先用第一个DES密钥对数据进行DES加密运算，然后对上述结果利用第二个DES密钥进行解密运算，最后再利用第一个DES密钥对上述结果进行DES加密运算。因此本文所说的DES密钥为一对DES密钥，整个算法为2密钥3DES、或双倍密钥长度的3DES。所有PBOC2.0规范借贷记应用的DES密钥长度为16字节128比特，包括16比特的密钥校验位。

潜在的DES密钥的风险包括：

- 在物理上泄漏DES秘密密钥。
- 对存储在IC卡内的DES密钥采用旁路攻击。
- 对DES密钥采用穷搜索攻击，目前这种攻击在计算上对于2密钥3DES是不可行的。
- 实际破解DES算法，目前对3DES算法的实际攻击在计算上是不可行的。

PBOC2.0规范使用3DES对称密码算法进行联机交易报文的加密和数据完整性保护、进行联机实体（卡片和发卡行）认证。在IC卡的个人化系统中使用3DES算法进行传输数据的加密及完整性保护。

分组密码算法，或在PBOC2.0规范借贷记应用中指对称密码算法通常是基于简单的逻辑组合运算，并将简单的逻辑组合运算进行多次迭代，如DES对基本逻辑运算进行16次迭代，成为16次轮运算。这样算法运行中需要的资源很小，运行速度非常高，适用于对大的数据做加解密运算和完整性保护。目前还没有发现比穷搜索更有效的攻击3DES的密码分析算法，但存在其它攻击方式，比如旁路攻击等。

5.3.2 非对称算法

非对称算法或公钥算法要求在通讯的两端使用两个不同但相关联的密钥：一个是公钥，另一个是私钥，其中公钥以可信的方式公开而私钥仅由私钥持有者知道并安全管理。一些公钥密码算法如RSA可以用于加密运算和数字签名运算。在使用公钥算法做加密运算时，加密者使用通讯对方的公钥对数据加密，并将数据加密结果传递给对方。对方收到加密数据后使用自己的私钥对加密数据解密。RSA也可用于数字签名运算，签名者利用自己的私钥对数据的散列值（哈希值）进行RSA运算，即数字签名运算，其后任何人在得到签名者的可信公钥时可以使用该公钥对该签名数据进行RSA运算得到签名数据的哈希值，若计算得到的哈希值与收到的哈希值相同则签名验证成功。在PBOC2.0规范借贷记应用中使用RSA非对称算法生成数字签名用于卡片脱机数据认证。

公钥密码算法通常基于一个数学难题。公钥密码算法的设计是没有比实际解决这个数学难题更好的方法来攻击该公钥算法。RSA是基于和数分解这一数学难题，也就是已知一个和数，该和数由两个素数相乘得到，在仅知该和数时试图分解得到这两个素数在计算上是不可行的。在RSA中这个和数就是公钥模数，分解了该模数就得到了RSA的私钥。RSA中公钥模数越大（即公钥模长、公钥长度）越大，分解该模数或破解对应的私钥更困难。

5.3.2.1 非对称（RSA）密钥

RSA算法中私钥的安全性取决于下列因素：

- 以比特位表示的RSA公钥模的长度，即公钥模长，如1024比特、1152比特、1408比特和1984比特。
- 在私钥存储、传递或使用时防止私钥被非授权访问或泄漏的物理安全性。
- 构成公钥模数的两个素数的质量。

私钥的潜在风险包括：

- 物理上将私钥泄漏。
- 实际分解公钥模数。
- 对存储于IC卡内的私钥进行旁路攻击。
- 对RSA算法的破解－目前看在公钥模足够大时破解RSA在计算上是不可行的。

5.3.2.2 证书与认证机构

在传统密码系统内，密钥管理主要专注于对共享的秘密密钥的建立及维护。随着公钥或非对称密码技术使用的不断扩展，密钥管理工作的内涵有所变化。在公钥密码系统中，密钥成对出现因此称为密钥对，密钥对中的公钥可以广泛发布，而密钥对中的私钥需要严格保护其私密性，并仅由密钥持有者知道该私钥。

公钥需要以可信的方式发布给任何需要使用该公钥的实体。虽然公钥数据本身不需要保持数据的私密性或进行加密保护，但公钥数据的接收者需要确信得到的公钥数据未经过更改，并且来自合法的公钥持有者或可信机构，否则一个篡改过的公钥与真实的私钥持有者对应的公钥毫不相干。PBOC2.0规范借贷记应用采用公钥数字证书这样一个在公钥密码系统中广泛使用的工具来解决这个问题。

为理解证书，我们需要一个特定的公钥密码系统，称为数字签名方案。利用数字签名方案，一个实体的私钥被用来签署一则报文，即生成一个称为数字签名的比特串，该比特串或数字签名实际上是被签名报文和实体私钥的函数的输出。接收者收到一则报文和附带的数字签名后，可以使用签发者的公钥来验证该数字签名。这样数字签名可以用来检查报文源和报文数据的完整性，同时可以抵御一个签名者对已经签署了一则报文的抵赖。

一张证书是数字签名的一种形式，能够用来验证数据源的真实性和被签名公钥数据的完整性。一张证书包含一个公钥并链接其它相关数据（包括用户名字或个人标识码以及证书失效日期），并且由一个被称为证书认证机构（CA）的可信实体使用其私钥签名。这个CA将其自身的公钥以某种物理上可信的方式发布给实体用户群，该用户群中的任一用户一旦得到该CA公钥的可信拷贝，就可以验证由该CA签发的所有证书，这样就可以获取其它用户的公钥的可信拷贝。

在PBOC2.0借贷记应用环境中的IC卡根CA，或金融IC卡借记/贷记应用根CA正是起到了认证机构的作用。该IC卡根CA通过对每个发卡行公钥进行签名，即为每个发卡行签发公钥证书。发卡行也起到认证机构的作用并通过为每一张IC卡的公钥及相关数据签名来生成IC卡公钥证书。IC卡根CA将其公钥以可信方式通过收单行传递到终端以便验证发卡行证书，这样终端就可得到发卡行公钥的可信拷贝，并在其后验证IC卡的公钥。

5.4 PBOC2.0 借贷记应用认证方式

5.4.1 卡片和数据认证方法

PBOC2.0借贷记应用支持下列方法以验证卡片是真实的并且卡片内的相关数据自发卡行对卡片个人化后没有被更改.

5.4.1.1 脱机数据认证

在PBOC2.0借贷记应用中使用的卡片脱机数据认证有静态数据认证（SDA）、动态数据认证（DDA）以及复合动态数据认证，即动态数据认证/应用密码报文生成(CDA)。脱机认证是指卡片数据认证是由终端而不是由发卡行来完成的。

- 在SDA验证过程中，终端验证一个卡片发来的卡片数据的静态签名（也就是该签名对每笔交易都是一样的），以便确认这些卡片数据在卡片个人化后没有经过更改。
- 在DDA验证过程中，终端验证一个从卡片发来的动态签名（该签名对每笔交易均不同），这个动态签名是卡片利用其卡私钥对当前动态交易数据和卡片数据做的。DDA可以验证卡片不是非法假冒的伪卡，同时卡片关键数据在个人化后没有更改。
- 在CDA验证过程中，卡片对包括联机密码报文在内的交易利用其私钥生成一动态签名，以提供对DDA数据的保护，同时确保任何中间器件当交易信息在卡片和终端间传递时没有更改关键交易数据。

5.4.1.2 联机数据认证

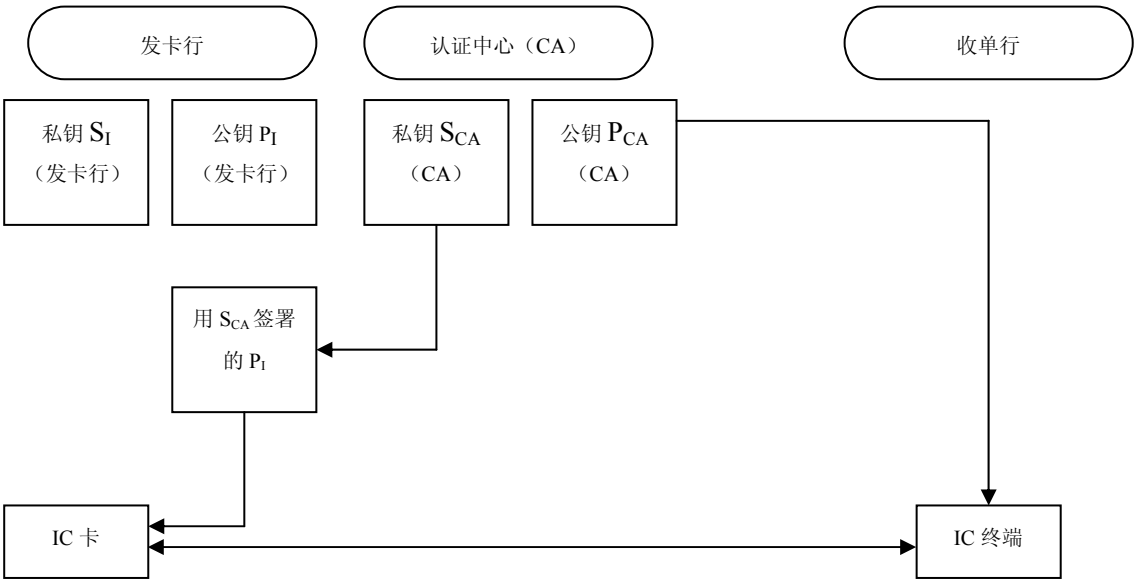
PBOC2.0的联机数据认证用于在联机状态下由发卡行认证卡片以及由卡片认证发卡行，同时用于保护收到的数据的真实性。

- 在联机卡片认证处理过程中，发卡行联机系统确认卡片联机发来的ARQC密码报文是用正确的卡片秘密密钥（3DES）对重要的交易数据进行密码运算得到，以证明卡片不是伪卡同时交易数据没有经过更改。
- 在联机发卡行认证处理过程中，卡片确认由发卡行利用其秘密密钥（3DES）生成的ARPC密码报文是用正确的秘密密钥生成的，并确认该授权回复是由合法的发卡行生成并传来，而且数据未经更改，这样，根据报文内容卡片执行内部卡片管理操作，如重置脱机PIN计数器等。
- 在安全报文处理过程中，发卡行利用其秘密密钥生成并向卡片传递一个由MAC密码报文保护的更新脚本。卡片仅在验证了该MAC密码报文后对卡片应用该更新脚本，卡片成功验证了该MAC证明该密码报文是由合法的发卡行生成并且在数据传递过程中数据没有经过更改。安全报文也用来加密敏感数据，比如在发卡行和卡片间传递更新的PIN。
- 对于已批准的交易，终端发送一个由卡片生成的密码报文TC，包含清算信息，供发卡行验证，以便作为完成的交易的确认凭证。

5.4.2 静态数据认证（SDA）

SDA是一个基于公钥密码技术的静态签名验证机制，终端通过验证卡片内由发卡行利用其私钥签名的卡片关键个人化数据来证实该卡片的关键数据在个人化后没有经过更改。

静态数据认证中数据和密钥的关系如图1所示。



IC卡提供给终端:

- 由认证机构签署的发卡行公钥（P_I）
- 带有数字签名的卡数据

终端:

- 使用认证中心公钥来验证发卡行公钥是由 CA 签署的
- 使用发卡行公钥来验证卡数据的数字签名

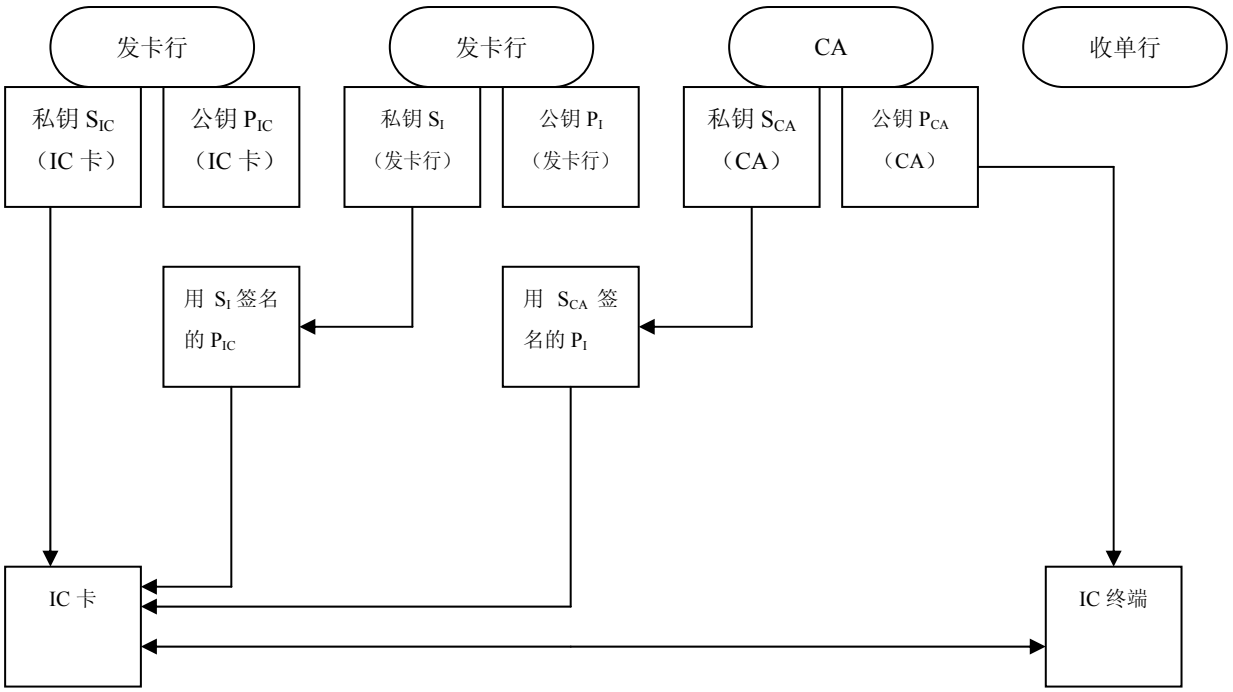
图 1：静态数据认证

由于SDA是对卡片静态数据的签名进行验证，这样对于脱机数据认证而言可能出现将一张合法的卡片内的合法静态数据签名拷贝到另一张卡片内生成一张非法复制的卡片，这样的非法复制的卡片在脱机环境中可以通过静态数据认证，也就是可以进行脱机交易，但由于无法将合法的卡片内正确的秘密密钥（3DES密钥）拷贝到另一张卡片内，因此这样的非法复制卡无法成功进行联机交易。

5.4.3 动态数据认证（DDA/CDA）

DDA和CDA是基于公钥密码技术的动态签名验证机制，终端通过验证由卡片利用其私钥对当前动态交易数据及关键卡片数据或当前动态交易数据及关键卡片数据和交易密码报文的签名来认证卡片并确认卡片内关键数据的真实性，由于动态签名需要卡片用其私钥签名，而卡片私钥一般无法从卡片内读取，因而动态数据认证排除了复制的卡片进行脱机交易的可能。

动态数据认证相关的数据及密钥的关系见图2。



IC卡提供给终端：

- 由发卡行签名的 IC 卡公钥（P_{IC}）
- 由认证中心签名的发卡行公钥（P_I）
- 带有数字签名的卡片和终端数据

终端：

- 使用认证中心公钥来验证发卡行公钥是否由 CA 签名的
- 使用发卡行公钥来验证 IC 卡公钥是否由发卡行签名的
- 使用 IC 卡公钥来验证卡数据的数字签名

图 1：动态数据认证

除了应用密码报文和其它交易数据（如交易批准状态）也包括进动态应用数据的签名外，CDA与DDA的功能非常类似。CDA的特征可以使得终端验证上述数据是由正确的卡片生成的，并且传输的交易数据在卡片与终端间传递时没有经过更改。

5.4.4 认证方式的比较

下面的表1说明了这些认证方式对卡片交易的安全性所提供的保护及其影响，以供选择认证方式时考虑。

表 1：卡片认证比较 1

	SDA	DDA	CDA	联机卡片认证
检测非法卡片静态数据	√	√	√	√
检测卡片静态数据的更改	√	√	√	√
检测非法复制的卡		√	√	√

片				
检测卡片与终端间通讯中外界的攻击			∨	∨
是否可应用于脱机交易	∨	∨	∨	

应该指出，SDA、DDA和CDA这三种脱机认证方式提供了安全性上呈递增等级的交易安全防护。下面的表2说明这些认证方式的实现对卡片、终端、主机处理和交易时间上的影响。

表 2：卡片认证比较 2

	SDA	DDA	CDA	联机卡片认证
卡片		必须支持 RSA	必须支持 RSA	必须支持 3DES
终端	必须支持 RSA	必须支持 RSA	必须支持 RSA	必须具有联机能力
发卡行发卡及主机系统		必须个人化 IC 卡的公私钥对	必须个人化 IC 卡的公私钥对	必须个人化 IC 卡秘密密钥，必须支持 IC 卡交易授权
交易时间	两次终端 RSA 运算	三次终端 RSA 运算和一次 IC 卡 RSA 运算	三次终端 RSA 运算和一次 IC 卡 RSA 运算	要求联机授权

5.4.5 持卡人验证方式

PBOC2.0规范借贷记应用支持下列持卡人验证方式：

- 手写签名
- 联机加密PIN，持卡人将联机PIN键入终端，终端PIN的PAD将该联机PIN加密与卡片的联机交易请求密码报文一道发送给发卡行以便发卡行对持卡人和交易请求进行验证。
- 脱机明文PIN，持卡人将脱机PIN键入终端，终端将该脱机PIN传递给卡片，卡片比较收到的脱机PIN与安全存储于卡片内的脱机PIN，然后卡片告知终端脱机PIN验证是否成功。
- 脱机明文PIN验证加签名
- 签名
- 不需要CVM（认为CVM通过）
- CVM处理失败（认为CVM失败）
- 出示证件

5.5 交易授权系统

交易授权是这样一个过程,发卡行或发卡行的代表批准或拒绝一笔交易以作为由商户终端通过收单行传递给发卡行的联机交易和授权请求的回复。

联机交易授权请求包括卡片生成一个ARQC授权请求密码报文,发卡行验证该密码报文以确认该卡片是真实的并且交易数据没有经过更改。该联机交易请求也包括卡片和终端对脱机处理结果的标识。

在向卡片回复ARQC时，发卡行可选地生成一个ARPC授权响应密码报文并传递给卡片。卡片验证该ARPC以确认该密码报文来自真实的发卡行并且报文数据在传递过程中没有更改过。

除了上述ARPC外，发卡行可以利用发卡行脚本指令执行卡片的发卡后更新。比如，发卡行可以更改脱机PIN或更新卡片风险控制参数。发卡行通过生成一个MAC密码报文来保护脚本指令的数据完整性，卡片在执行发卡行传递的脚本指令前对该MAC密码报文进行验证，这里采用加密来保护机密数据的安全。

6 PBOC2.0 卡片发卡的安全性

本章涉及发卡行在发卡过程中需要完成的与安全相关的功能：

- 生成、管理并安全存储发卡行公私钥对。

- 将发卡行公钥发送给IC卡根CA以便获取根CA为其签发的发卡行公钥证书。
- 接收并存储发卡行公钥证书及相关的IC卡根CA公钥以便对收到的发卡行证书签名进行验证。
- 为DDA或CDA应用生成IC卡公私钥对。
- 使用发卡行私钥签发IC卡公钥证书或者对应于SDA应用签署卡片静态数据。
- 生成并安全存储发卡行对称主密钥（秘密密钥）。
- 使用发卡行主密钥导出用于联机授权和安全报文应用的IC卡主密钥（秘密密钥）。
- 将卡片个人化过程需要的相关密钥以安全方式传递给卡片个人化系统，这包括发卡行主密钥的输入/输出（仅当发卡行将卡片联机验证服务委托给第三方）。

6.1 发卡行工作

发卡行在卡片发行及整个卡片生命周期内需要完成下列任务：

- 发卡前准备，在发行IC卡前需完成的工作，
- 卡片生产（SDA卡），发行支持静态数据认证的IC卡的步骤，
- 卡片生产（DDA卡和CDA卡），发行支持动态数据认证的IC卡的步骤，
- 卡片发行，向持卡人提供新生成的PBOC2.0卡的步骤，
- 联机交易处理，支持卡片使用的程序，包括对发自PBOC2.0IC卡的联机密码报文的验证、持卡人联机PIN的验证、以及使用发卡行脚本指令对卡片应用的更新。
- 交易清算，支持PBOC2.0借贷记应用交易清算和结算的程序。
- 作废，在卡片发行程序工作中和发行程序执行后，各种密钥将作废并被销毁。

6.1.1 发卡前准备

在发卡前，发卡行需要完成下列工作。这些工作在密钥变更和证书过期时也需要完成。

6.1.1.1 非对称（RSA）密钥

密钥对生成。发卡行需要安全生成及安全存储公私钥对。需要使用位于物理上安全的设备内受到保护的存储器，利用随机或伪随机数生成器和标准素性检测程序在安全密码硬件内生成这些公私钥对。

发卡行需要生成的公私钥对包括：

- 发卡行公私钥对。其私钥用于签发卡片静态数据以及签发IC卡公钥证书。其公钥递交给IC卡根CA用于获取IC卡根CA签发的发卡行公钥证书。
- IC卡公私钥对。发卡行将IC卡私钥在卡片个人化过程中写入卡片内以用于卡片DDA和CDA认证，发卡行利用发卡行私钥为IC卡公钥签发IC卡公钥证书并将该证书在卡片个人化过程中写入卡内。

以比特位表示的RSA公钥模长度（即1024、1152、1408和1984比特）应适应于卡片计划的整个使用期的安全要求。本文强烈建议发卡行选择足够大的公钥模长度以抵御对RSA的计算攻击，也就是分解公钥模数。出于对系统支付效率的考虑，建议IC卡根CA使用与发卡行签发的卡片生命周期内安全需求相适应的公钥模长相对应的私钥来签发发卡行证书，而不是只使用最大IC卡根CA公钥模长对应的私钥签发发卡行证书。发卡行的公钥模长应与IC卡根CA签发该发卡行证书的公钥模长相同以提高整个系统效率并抵御风险。

发卡行应周期性检查评估其正在使用的RSA公私钥对的公钥模长，当发现某个公钥模长代表的安全性不再符合安全要求时，应终止该公私钥对的使用并引进更大公钥长度的发卡行公私钥对。同样也应评估发卡行公钥指数相应于发行的卡片的安全性。

为RSA密钥生成而生成的素数应从至少2^{100}个素数中随机选择一个。构成公钥模数的两个素数p和q的大小应至少相差2^{128}。

公私钥对生成中的素数生成应采用标准的素数生成算法，关于素数生成应参考ISO/IEC 18032。

在选择公钥指数时主要需要考虑的因素是运行效率。通常终端计算RSA的时间在选择公钥指数为$2^{16}+1$时比选择公钥指数为3明显长。而密码学认为IC卡的RSA实现取公钥指数为3或公钥指数为$2^{16}+1$在公钥模长足够大时均具有有效安全性，所以公钥指数应选择3。

密钥生成使用的随机或伪随机过程应满足，生成的密钥是不可预测的，或不可能预测特定密钥在密钥空间中比其它密钥更可能出现。随机或为随机数生成器应符合ISO/IEC 18031及国家商用密码管理局的随机性测试。

用于生成RSA密钥对及保护私钥的物理安全器件（硬件安全密码模块）应具有抗篡改功能，并符合本文第8.1节安全密码器件的安全要求。

本文建议发卡行对不同的BIN号使用不同的公私钥对生成IC卡数据，以限制发卡行使用相同公私钥对生成的卡片数目，以降低因发卡行私钥泄漏影响的卡片数量。

发卡行在将发卡行公钥传递给根CA以签发发卡行证书过程中，应确保传递的发卡行公钥数据未经篡改。

对支付系统IC卡根CA，不同的发卡行的公私钥对应不同。

密钥生成的过程指南见本文第7.1节。

接收IC卡根CA公钥。发卡行需要接收和安全存储一个或多个根CA公钥。

发卡行应根据PBOC2.0规范和金融IC卡借记/贷记应用根CA公钥认证规范在收到根CA的公钥时验证公钥数据的完整性及公钥数据信息源。

申请和接收发卡行公钥证书。发卡行需要向根CA传递每一个发卡行公钥并接收根CA返回的发卡行公钥证书。

发卡行向根CA传递发卡行公钥需确保根CA能够验证传递的发卡行公钥数据的完整性及发卡行公钥数据信息源。

在收到根CA签发的发卡行证书后，发卡行应使用相应的根CA公钥验证该发卡行证书。

具体技术和业务细节见金融IC卡借记/贷记应用根CA公钥认证规范。

发卡行私钥及发卡行证书的备份。发卡行必须对其发卡行私钥进行备份。备份的私钥应以加密的方式存在，或存储于安全密码器件中，见本文7.2.1节；发卡行应备份其发卡行证书。

6.1.1.2 对称（3DES）密钥

发卡行主密钥的生成。发卡行需要安全生成并安全存储一个或多个发卡行主密钥以用于导出对每一个卡片应用都唯一的IC卡主密钥。这需要使用位于安全器件内的具有安全保护存储器，并使用随机或伪随机数生成器。

PBOC2.0借贷记应用中的3DES密钥应用于特定交易功能。卡片主密钥由发卡行主密钥分散导出并在卡片个人化时写入卡片。每张卡片的主密钥都是唯一的。

发卡行主密钥包括：

- 发卡行应用密文主密钥（IMK_{AC}），用来导出卡片密钥，该卡片密钥用于在交易中生成MAC，即应用密码报文AC。
- 发卡行主密钥用于安全报文中的完整性保护（IMK_{SMI}），用来导出卡片密钥，该卡片密钥用来在卡片和授权系统间的发卡后处理过程中安全报文的完整性保护，即用于卡片锁定、应用锁定/解锁、更新卡片特定数据和PIN变更。
- 发卡行主密钥用于安全报文中的机密性（IMK_{SMC}），用来导出卡片密钥，该卡片密钥该卡片密钥用来在卡片和授权系统间的发卡后处理过程中安全报文的机密性保护，即用于卡片锁定、应用锁定/解锁、更新卡片特定数据和PIN变更。

3DES密钥应可以在具有抗篡改防护机制的物理上安全的器件中生成，或由授权的个人各自掌握部分密钥分段并通过密钥分段的组合过程产生整个密钥，在密钥生成时每一授权个人生成其密钥分段。密钥组合过程在物理上安全的器件内完成。而且，密钥组合方法应确保知道任何密钥分段的子集仍无法获知整个密钥。在密钥以分段的形式在卡片个人化机构生成时，至少一个密钥分段应由发卡行的雇员生成。

密钥生成使用的随机或伪随机过程应使得预测任何密钥是不可能的，或在密钥空间中确认特定密钥比其它密钥更可能出现是不可能的。关于随机或伪随机数生成器的进一步信息可参看ISO/IEC 18031，并且随机数或伪随机数生成器应通过国家商用密码管理局关于随机数生成器的伪随机性测试。

本文建议发卡行针对不同的BIN号采用不同的发卡行密钥以限制使用一个发卡行密钥生成的卡片的数目。

密钥只应用于特定的密码目的并只用于设定的应用，而不是任何其它目的，也就是说，对于应用密码报文、安全报文完整性以及安全报文加密等不同的密码应用应使用不同的主密钥导出。

发卡行对称主密钥应做周期性（每年）更新。这并不意味发卡行必须周期性变更已颁发的卡片内的密钥，而是发卡行必须能够一次管理多个密钥形式，一旦由该密钥形式分散导出的卡片密钥对应的卡片有效期到期并且随后没有任何争议时，发卡行应将该发卡行密钥形式销毁。

关于密钥生成的进一步指南见本文第7.1节。

- **发卡行主密钥的传递**。如果发卡行将其卡片个人化或产生及验证联机密码报文的职责委托给第三方或委托给发卡行自己的处理中心的不同系统，则发卡行需要向该第三方或发卡行机构内另一个系统安全传递用于分散导出IC卡密钥的发卡行主密钥（见本文第7.2节）。
- **发卡行主密钥的备份**。发卡行必须对其发卡行主密钥进行备份。

在备份关键的发卡行主密钥时，本文建议被备份的发卡行主密钥或者用另一个与该发卡行主密钥相同长度的密钥加密形式存在，或者采用将该发卡行主密钥分成两个以上的分段，使用双人或多人控制安全机制，每个实体各自只掌握各自的密钥分段值。这个过程应能够被审计。

6.1.2 安全计数器

如果使用安全计数器来限制密钥的使用，则该安全计数器的阀值需要事先确定。

对一个典型的脱机环境安全计数器阀值，本指南的推荐值为：

- 脱机PIN解密限制＝1024
- ATC限制＝20000
- AC和SMI安全计数器阀值限制＝1024。

注意具体的支付系统可以提供不同的阀值。

6.1.3 卡片生产（SDA）

对于发行每一张SDA卡，发卡行应执行下列与安全相关的步骤。

- **静态数据准备**。发卡行生成为卡片个人化生成相关卡片数据。
- **签署静态数据**。发卡行通过使用发卡行私钥来签署所选的卡片静态数据来产生卡片的签名静态应用数据。

发卡行应遵循PBOC2.0规范和要求生成卡片静态数据：

- **IC卡主密钥的分散导出**。IC卡的用于应用密码报文和脚本报文安全的主密钥应由适当的发卡行主密钥，使用PAN和PAN序列号等卡片数据分散导出。
- **PIN生成**。如果支持脱机PIN，应为IC卡生成脱机PIN数值。脱机PIN应与联机PIN相同。
- **向卡片个人化过程提供卡片数据**。向卡片个人化系统安全传递发卡行公钥证书、根CA公钥索引、签署的卡片静态应用数据、分散导出的卡片主密钥、导出密钥索引（如果使用），以及PIN。

发卡行应将所有个人化所需的数据安全地传递到卡片个人化系统并安全写入卡片。

在上述个人化数据由发卡行到卡片中的整个传递过程中，应采用MAC或签名的形式安全保护传输的数据。

对秘密密钥和PIN数据的传递应保持数据机密性。

6.1.4 卡片生产（DDA 和 CDA）

在发行每一张支持动态数据认证的卡片时发卡行需要执行下列安全相关的步骤：

- **卡片静态数据准备**。发卡行为卡片个人化生成静态数据。
- **签署静态数据**。如果也要求SDA能力，发卡行使用其发卡行私钥对选择的卡片静态数据签名以生成签名的静态应用数据。
- **IC卡公私钥对的生成**。对每一张卡片，发卡行应安全生成卡片唯一的公私钥对。

- **签署IC卡公钥以生成IC卡公钥证书**。发卡行应使用其私钥之一签发IC卡公钥证书及IC卡静态应用数据。用于签发IC卡公钥证书的发卡行私钥对应的发卡行公钥证书应个人化到卡片中。
- **IC卡主密钥的分散导出**。发卡行应使用适当的发卡行主密钥分散导出IC卡的用于应用密码报文和脚本报文安全的主密钥，在分散导出过程中应使用PAN和PAN序列号的卡片数据。
- **PIN生成**。如果支持脱机PIN，发卡行应生成IC卡的脱机PIN。脱机PIN应与联机PIN一样。
- **向卡片个人化过程提供卡片数据**。发卡行应安排并将IC卡根CA公钥索引、签名的卡片静态数据、导出密钥索引（若使用）、导出的秘密密钥以及PIN传递给卡片个人化过程。

所有个人化所需的卡片数据应安全地传递给卡片个人化机构并安全写入卡片。

所有个人化所需的数据在离开生成该数据的系统到最终写入进卡片的整个过程中应以MAC或数字签名的形式保护。

卡片秘密密钥和卡片私钥，以及卡片PIN应以保证数据机密性的方式加以保护。

完成卡片个人化后，IC卡发卡行及个人化机构应销毁IC卡私钥的任何记录。

6.1.5 卡片发行

发卡行在完成每一张卡片的发行后应执行下列安全性相关的步骤。

- **安排并将IC卡和PIN传递给持卡人**。个人化后的IC卡和PIN应分别安全地传递给持卡人，该传递应遵循ISO 9564－1的规定。

6.2 发卡行发卡后行为

发卡行应为每笔联机交易执行下列步骤。

- **密码报文交换**。作为在商户端（具有联机能力）卡片交易安全程序的一部分，IC卡或商户终端可要求包括卡片认证的联机授权。

联机授权包括从卡片向发卡行传递一个应用密码报文（ARQC）。卡片利用IC卡主密钥（在个人化过程中由发卡行主密钥分散导出）或由IC卡主密钥分散导出的工作密钥加密卡片、终端和交易数据生成该密码报文。发卡行使用相同的密钥验证该ARQC。

发卡行可使用与验证ARQC时使用的相同密钥生成一个回复密码报文(ARPC)。ARPC是发卡行加密ARQC和诸如授权回复码或卡片状态更新数据的结果。卡片利用相同的密钥验证ARPC以验证该回复来自合法的发卡行并且数据在传输中未经更改。

- **安全报文**。作为整个卡片管理过程的一部分，安全报文（使用分散导出的IC卡密钥对报文做密码保护）可由发卡行传递到IC卡以更新卡片内的数据。这些更新可更改PIN、重置PIN计数器、锁死或解锁应用、锁死卡片、以及变更卡片风险管理数据。
- **持卡人验证**。在一些联机环境中（如ATM），将要求发卡行验证随授权报文传来的加密PIN。

6.2.1 交易清算

发卡行需要为每笔批准的交易执行下列安全相关的步骤。

- **TC处理**。作为每一笔IC卡交易的结果，IC卡利用卡片秘密密钥生成一个交易密码报文（TC）并传递给商户终端。该TC再由商户传递给收单行，然后收单行再将该TC传递给发卡行以作为交易清算过程的部分步骤。作为交易清算和结算过程的一部分，可要求发卡行验证该TC的正确性。

6.2.2 密钥作废

本节为销毁作废的密钥和相关密钥材料提供指南。

6.2.2.1 密钥保持

基本规则是一个密钥在已知不再需要的时间点前应被保留或保持，在决定该密钥不再需要时应销毁该密钥。

应以下列方式保持发卡行非对称密钥对：

- **发卡行私钥**，一直保持到该私钥不再用于任何卡片的个人化为止，即该私钥对应的发卡行公钥证书有效期到期。
- **发卡行公钥**，一直保持到由该公钥参与的卡片认证不会出现任何争议为止。

应以下列方式保持发卡行对称密钥：

- **发卡行主密钥**，一直保持到所有使用由该发卡行主密钥导出的卡片密钥的卡片的使用不会再出现任何争议为止。

6.2.2.2 密钥销毁

作废的密钥材料应使用适应于存储该密钥材料的介质的方法予以销毁。

- 若作废的密钥存储于IC卡中，则应在使用任何芯片数据擦除方法后将卡片上芯片位置钻一个孔使得在物理上彻底销毁该芯片。然而，过期的卡片并非在发卡行控制范围内，所以这些卡片可能没有被有效废弃，而是仍具有部分原有功能，尽管商户终端不会再接受由这些卡片进行的交易。
- 对存储于磁介质内的密钥材料，该磁介质应按照相关剩余磁性标准进行消磁。对可移动介质应予以物理销毁（即切割成碎片）。注意，现代硬盘具有重新映像功能，这可能降低销毁过程的有效性，所以应采用物理销毁硬盘及清除数据技术。
- 存储于SCD内的废弃密钥应采用SCD安全机制予以擦除。

在销毁废弃的密钥材料过程中应安排一个独立第三方，如内部审计员见证密钥材料销毁的整个过程，并记录该过程。该过程的记录应按照发卡行文件保留策略保留一个时期。

7 密钥管理实践

PBOC2.0借贷记应用交易系统的安全实现取决于发卡行是否能够按照下面的通用要求来安全生成和管理PBOC2.0借贷记应用所需的密钥。

关于零售银行业密钥管理的进一步指南见ISO11568。

7.1 密钥生成－通用指南

下列通用指南应在生成对称密钥和非对称密钥对的过程中采用。许多过程要求使用随机比特生成或素数生成方法。

PBOC2.0借贷记应用发卡行系统密钥管理首先需要安全生成各种密钥。在PBOC2.0借贷记应用交易系统密钥存在于下述位置：

- 位于IC卡根CA系统
- 位于发卡行
- 位于发卡系统
- 位于卡片中
- 位于终端内

密钥或用于卡片发卡过程或用于卡片交易过程。这些密钥的生成方式分为下述两类：

- 单个生成的主密钥或保护卡片生产过程的密钥
- 大批量生成的卡片密钥

这些密钥需要备份以便在意外发生时用于恢复系统，同时这类密钥通常需要分段保存以便在操作密钥时由双人或多人控制。密钥生成过程需要记录审计日志。大批量的卡片密钥生成发生于拥有安全防护措施的卡片生产系统，同样整个安全过程需要记录审计日志。

所有的密钥生成需要按照一个定义的过程进行，该过程需要事先制定并且确保密钥生成时，所有相关角色都到位。具体的过程取决于涉及的应用系统。

PBOC2.0借贷记应用主密钥应按照定义好的密钥生成过程生成，卡片密钥应在安全卡片生产环境中生成。密钥生成使用的密码模块需要符合人民银行及国家商用密码管理局的相关要求。密钥生成使用的伪随机数生成器应符合人民银行及国家商用密码管理局相关规定。

用于生成密钥的环境必须是安全的，该环境应适用于密钥生成方式，使用的设备和设备配置必须对各种攻击包括电磁辐射分析等具有有效防护。

进一步的关于随机比特生成的指南见ISO/IEC18031，关于素数的生成指南见ISO/IEC18032，相关随机性测试和素数检测需符合国家商用密钥管理局的测试规范。

所有密钥生成应根据已制定的过程进行。该过程应明确定义，并确保所有需要的角色均在合适的时间到位。该过程的细节取决于进行密钥生成的具体系统。应确保密钥生成环境的安全。应确保密钥生成使用的设备、方法以及设备的配置的安全，使得整个密钥生成过程能够抵御诸如电磁辐射攻击等的特定旁路攻击。

密钥生成过程应清楚定义所有密钥材料的使用目的和其生命周期，以确保：

- 生成的密钥不用于其它目的，
- 测试和生产密钥材料不混淆，
- 在需要时进行密钥替换

对每一类型的密钥，应制定一个程序以明确该密钥何时生成、生成及存储该密钥的位置及方法以及该密钥生成需要的角色。这个程序应是明确的并对于所有角色都易于遵循。掌握密钥组件（分段）的个人（称为密钥管理人）并非总是具有该程序的详细知识也并不经常需要使用该程序。

这个程序应说明什么人有权授权该程序的运行。应定义用于生成密钥的系统是如何建立的，并且定义系统生成密钥后应采取什么动作来清空与该密钥相关的数据。应明确构成密钥分段的密钥组件应如何存储和由谁进行存储。

该程序应定义针对一个密钥需要生成多少个密钥数据拷贝以及这些密钥数据是如何保护的。注意，对于多数密钥需要至少两份密钥数据拷贝，一个作为生产系统使用，另一个作为离场备份用途。备份的密钥以及其生产系统中的密钥都需要相同安全程度的保护。

该程序的执行过程应生成记录，使得审计者在以后可根据程序记录、程序执行的授权和由设定的参与者签名的审计记录来判定该密钥已经安全地生成、存储以及使用。

应确保密钥数据的完整性。位于硬件安全密码模块内的校验数值和该密码模块外生成的MAC值适用于该目的。如果使用硬件密码模块内的校验数值，该校验数值的计算应针对整个密钥分段组件和整个实际的密钥数值进行。为密钥分段组件计算的校验值应与该密钥分段组件一道传输。

如果发现任何秘密密钥或私钥位于物理上安全的设备之外，或者这些密钥的所有分段组件都由或怀疑由单个个人控制，则应认为这些密钥已被泄漏。

7.1.1 RSA 密钥生成

本指南建议PBOC2.0借贷记应用中的RSA公钥算法密钥对应在硬件安全模块中生成，应从2^{100}个素数中随机选取。此外，对于任何公私钥对，构成公钥模数M的素数p和素数q（p*q=M）应至少相差2^{128}。应使用人民银行及国家商用密钥管理局认可的算法生成素数，比如利用Fermat素性检测使用Gordon算法选择强素数，并对选择的素数使用Miller－Rabin检测进行检测。检测中需要的迭代取决于密钥长度。比如若要求素数的错误率小于2^{-100}，则Miller－Rabin检测对于1024位的RSA密钥应迭代7次。对于发卡行公钥对应选择更多次数的迭代，因为发卡行公钥对的生成较少发生。同时也可以使用确定性方式生成素数，当该确定性方式生成的数值的素性是可证明的。

7.1.2 证书生成

公钥证书的生成应符合PBOC2.0规范和金融IC卡借记/贷记应用根CA公钥认证规范。此外若由密码硬件模块生成公钥证书，则应在证书输出前检查该证书以避免故障攻击。签发证书使用的安全密码硬件模块应能够抵御故障攻击。

公钥证书完整性的关键之一是由IC卡根CA签发发卡行公钥证书。发卡行公钥证书由发卡行将自己的公钥递交给IC卡根CA，然后IC卡根CA对该公钥签发证书。除了签发发卡行公钥证书，IC卡根CA公钥认证服务也负责发布根CA公钥。发卡行必须确认收到的根CA公钥是真实的，而根CA必须确认收到的用来对其签发证书的发卡行公钥是真实的。中国银联金融IC卡借记/贷记应用根CA公钥认证服务提供了工具和手段以实现上述目标。

7.1.3 对称密钥生成和分散

PBOC2.0借贷记应用中所有对称算法密钥应至少是双倍长度的，即128比特（对应于DES单倍密钥长度56比特而言）。所有卡片交易相关的对称算法密钥长度都需满足这一要求。所有其它用于卡片生产过程中的对称算法密钥（如用于密钥传输或安全通讯密钥）应至少是128比特。所有用于保护卡片生产过程的密钥都应安全地生成。

应避免使用DES算法中的弱密钥和半弱密钥。DES中的弱密钥及半弱密钥总共有16个，这样随机生成的DES密钥为弱或半弱密钥的概率为2^{-52}。一旦发现随机数生成器产生一个弱密钥或半弱密钥，则该随机数生成器失效的可能性远大于其它可能，此时应检查该随机数生成器的状态。对于由分散算法导出的密钥，由于不改变如主帐号PAN或PAN序列号等的分散参数是无法通过重新生成密钥来避免弱密钥的，因此可以忽略由分散导出的密钥的弱密钥性和半弱密钥性的检查。

生成的PBOC2.0借贷记应用对称密钥应以分段方式从密钥生成设备中输出和存储，只有所有密钥分段值持有者或一个定义的分段值的子集的所有分段值持有者在一起才能重新恢复该密钥。应确保密钥分段的持有者无法访问其它密钥分段持有者的密钥分段值。本指南建议对称算法密钥至少应分为两段。

7.1.4 用于生成密钥的随机数生成器

本指南要求用于生成PBOC2.0借贷记应用密钥的随机数生成器应符合人民银行及国家商用密码管理局的相关检测要求，并符合FIPS140－2标准的等级3要求。密码硬件模块应符合人民银行及国家商用密码管理局的相关检测要求，并符合FIPS140－2标准的等级3要求。

7.2 密钥的存储和传递

7.2.1 密钥存储

在PBOC2.0借贷记应用体系中密钥存储于下列位置：

- 位于发卡行生成卡片个人化数据的系统
- 位于发卡行的发卡系统
- 位于卡片内
- 位于IC卡根CA系统及其服务系统
- 位于收单行终端
- 位于发卡行主机系统的加密机内
- 位于密钥传输系统

通常对IC卡内的密钥有备份需求。所有密钥在存储过程中需维持私密性、完整性和可应用性。

7.2.1.1 硬件与软件

一般密钥或存储于硬件内（如加密机）或软件内（如主机计算机系统）。实际上密钥通常存储于磁存储介质（如磁盘）、可变硅存储器、或长期稳定硅存储器中。密钥在存储过程中受到各种物理保护手段的保护，比如：

- 加密机或IC卡的抗篡改装置
- 加密机、IC卡和主机系统内操作系统的逻辑保护

存储于磁介质内的密钥仅做短暂存储，因为磁盘的寿命只有几年。存储于硅存储器的密钥可作长期存储，硅存储器存储的数据可以维持至少10年。

无论密钥如何存储，都需要保护以避免泄漏。应采用物理方法保护存储密钥的器件（比如将密钥存储器件使用防篡改封装放置于保险柜内）。存储密钥的器件在将密钥导出前应检查密钥的完整性。存储于数据库的加密过的密钥尤其需要检测其数据完整性，因为加密过的密钥虽然难以读取但可以被更改。存储于安全器件（如IC卡）中的密钥的数据完整性一般只需要以非密钥方式的校验和检测以判断意外的发生，但是当密钥可以被变更时（比如存储于数据库中），应采用密码技术的报文认证码MAC来验证数据完整性。

7.2.1.2 访问控制

应对所有存储在硬件安全密码模块外的密钥的访问采用至少双人控制。

7.2.1.3 硬件安全密码模块和集成电路安全存储器

硬件安全密码模块与集成电路在一些方面具有相同特征但在其它方面差别很大。一般，硬件安全密码模块包含若干个独立的存储及处理部件，密钥由内部硬件数据线在这些存储及处理部件间传递。这样，当探测到数据泄漏时必须清空硬件安全密码模块的存储器。同样，硬件安全密码模块的设计应考虑电磁辐射泄漏的防护。硬件安全密码模块一般是设计用来在安全环境内运行的。相反，IC卡的设计使得IC卡可以在未受保护的环境中使用，其存储器具有防篡改保护。

7.2.2 RSA 密钥的传递和存储

公钥应以能够验证该公钥数据的完整性和信息源的可认证性的方式传递。具体细节见金融IC卡借记/贷记应用根CA公钥认证规范。

公钥应位于一张签名的公钥证书内传递、或由专门用于生成MAC的密钥生成的MAC保护、或者在双重控制下（即公钥的校验值与公钥数据分别独立传递）。MAC算法应遵循ISO16609。

公钥的接收者应具有适当方法来验证所收到的公钥数据、所信任的根CA公钥、共享的MAC密钥或校验值的数据源和信息完整性。

私钥应保持机密性，若私钥需要传输则应确保传输的私钥的完整性和机密性。私钥传输机制宜包括：

- 私钥位于安全密码硬件内一起传输，
- 使用与受到保护的私钥具有至少相同安全强度的对称密码算法对传输的私钥加密，
- 如果使用私钥分段组件传输，应采用双重控制和私钥数据分段组件分别独立掌握的原则。

业务恢复策略应要求对公私钥对进行备份，应确保这种备份的安全性。在发生系统灾难性故障时将需要使用这些备份的公私钥对以尽快恢复系统的正常运行。备份的系统密钥应受到相同的安全保护。

注意，若发卡行私钥被无意删除，则对已颁发的卡片使用没有任何影响，此时发卡行应生成新的发卡行公私钥对并获取新的发卡行证书，新的发卡行公私钥对的使用不影响已有卡片的使用。

7.2.3 3DES 密钥的传输和存储

可能需要传输和存储3DES密钥。例如从发卡行场所将3DES密钥传输给第三方处理机构或给卡片个人化厂商。当对3DES密钥传输和存储时，应采用下列能够限制密钥数据泄漏风险的措施。

明文的3DES密钥数据可以位于在安全硬件或IC卡内并在安全硬件的保护下传递和存储。

当3DES密钥不位于安全硬件内时应仅以下列方式传输或存储：

- 3DES密钥数据以分成两端或多端组件的方式采用双重控制和密钥数据分段组件分别独立掌握的原则，或
- 位于一个密码报文内（加密的3DES密钥），该报文由已建立的系统传输密钥或存储密钥生成。

7.3 密钥管理的实践与职责

7.3.1 密钥管理人员安排

负责管理加密密钥、密钥分段组件、安全硬件以及其它密钥器件的人员应由参与的不同方设定，这些参与方包括发卡行、第三方处理机构和卡片个人化厂商。

在设定负责管理控制密钥材料过安全硬件的人员时，应采取充分地控制以确保未授权的个人无法访问包含密钥的数据或安全硬件。

密钥管理者应是可信赖的雇员而不是零时合同员工或提供咨询的外部人员。

为了且保业务连续性，应安排相对已安排的主要密钥管理人员而言的备用密钥管理人员。选择备用密钥管理人员的准则应与选择主要密钥管理人员一致。

7.3.2 密钥管理功能

密钥管理职责十分重要并构成了发卡行安全协议的基础。由密钥管理者管理的密钥通常是发卡行发行及管理卡片使用和交易过程涉及的密钥应用中最重要的密钥。

每个发卡行应按照下列规则审核其内部密钥管理程序以及密钥管理相关责任人。

- 密钥管理者的职责包括控制密钥材料、验证密钥材料以及确保这些密钥材料的安全存储。
- 密钥管理者负责：

- 接收和安全存储密钥分段组件和安全硬件，包括进行密钥材料的认证和确认收到密钥材料。
- 维持记录和日志以跟踪对密钥材料的访问和使用，包括访问时间、日期、用途以及将密钥材料返还并安全存储。
- 验证所有的向那些不在发卡行控制范围内而由其它参与方设定的人员传递密钥材料。
- 见证过期或作废密钥组件的销毁。
- 在需要时将密钥材料输入到安全密钥模块中。
- 指导监控作废密钥材料的销毁。

7.3.3 将密钥材料传输到第三方机构的程序

应在向第三方机构进行的密钥传输过程中使用一式两份表格，这两分表格应与密钥材料一道由密钥发送这传递到密钥接收者。该表格应确认密钥发送者身份以及传递的密钥材料。密钥发送者应在发送时审核该表格并在两分表格上签名。在密钥接收者收到传输的密钥材料及表格后，接收者应立即对比收到的表格验证收到的密钥材料，并在两分表格上签名后将一份表格传回发送者。

在向第三方传递密钥分段组件和其它密码相关数据时，每一密钥分段组件应发送给不同的个人并使用不同的传递服务，即挂号信件、特快专递、信使等。在传递过程中各个密钥分段组件应位于安全硬件或IC卡内，这样挂号信件就可满足要求。

应事先通知密钥材料的接受者将向该接受者传递密钥组件。

7.3.4 第三方机构安全保护密钥材料的物理程序

一旦收到传递的密钥材料，负责密钥管理的人员应立即检查包含密钥材料的包裹的封装以确认在中途未经启封，并验证包裹内的内容。

密钥材料接收者一旦对所传递的密钥材料的完整性有任何疑问，应立即通知密钥材料发送者。发送者在接受者的配合下应确定该密钥材料的未来状态。关于继续使用该密钥材料的任何决定及理由应做文件记录并由发送着和接受者双方保存。

如果密钥材料的硬拷贝需要在任何特定时期内保存，该密钥各组件的硬拷贝、安全硬件或IC卡应位于在封装上排序的抗篡改封装信封内并存储于物理上安全的地点。包含密钥材料单个组件的信封应各自存储于不同的物理安全地点，这些地点应禁止共享访问。

序列化的抗篡改封装信封应受到物理上安全的容器的持续性保护，对这些信封的访问仅限于设定的密钥管理者或备用的密钥管理者。对这些密钥材料的每次访问都应生成访问日志，日志应包括访问时间、日期、信封序列号、用途以及访问者签名。这些日志应提供给任何适当的请求机构进行审核。

密钥材料应在访问者完成设定的访问使用目的后立即交还并重新放回抗篡改信封，存储于设定的物理安全环境。

8 硬件和软件安全考虑

8.1 安全密码器件（SCD）

在ISO标准中（如ISO13491），安全密码器件（SCD）是指一类广泛的设备用来为密码运算提供特定程度的保护。高端的安全密码器件是一系列连接到主机处理器、个人化设备和和授权系统的硬件安全模块，低端的安全密码器件是单个安全硬件令牌和IC卡。

本指南这部分的内容是用来使得相关读者对作为发卡行风险管理实践的一部分的交易系统需要的通用安全属性有更全面的理解和认识。

8.1.1 抗篡改要求

抗篡改可以分为两个安全相关的领域：物理上的和逻辑上的。

8.1.1.1 物理安全属性

物理安全包括下列属性：

- 对穿透性攻击包括擦除敏感数据在内的防护。

- 对可以导致敏感信息泄漏的非授权更改的防护。
- 对针对运行密码算法的器件的电磁辐射、功耗等旁路攻击的防护。

安全密码器件应针对穿透性攻击具有有效防护，或者采用足够的物理防护措施使得穿透性攻击是不可行的，或者采用探测机制探测任何穿透性攻击并启用反措施。当安全密码器件以设定的方式运行于设定的环境时，成功的穿透性攻击可能导致密钥和其它敏感数据的擦除。

为了探测器件内的密钥和其它敏感数据而对安全密码器件的变更需要将该器件置于专业分析试验设备中。安全密码器件应对用于插入导线或连接窃听设备的篡改具有有效防范，一旦探测到这类事件的发生，安全密码器件应损毁这个器件的功能使得其不再工作。通过对器件的变更来确认或获取敏感信息需要高度的技能和专业设备及代价。

安全密码器件应使用被动的物理屏蔽装置以：

- 屏蔽X射线或其它射线以及电磁辐射用来探测器件运行密码运算所释放的射线或外部发射的射线。
- 提供私密性屏蔽使得在人工装载密钥数据时密钥信息无法由参与工作的人员观察到。

安全密码器件的安全性部分依赖于运行环境的安全，应确保非授权的安全密码器件的移出是不可行的或者这样的行为应导致防篡改反应机制的激活。

8.1.1.2 逻辑安全属性

逻辑安全包括下列属性：

- 软件的完整性验证。
- 功能集的设计应确保不存在单个或组合的器件功能能够导致敏感信息的泄漏。
- 确保密钥只被用于已定义的用途的机制。
- 对运行密码算法的器件的时间、扰动、执行序列等方面的旁路攻击的防护。
- 需要双重控制的敏感状态操作。
- 对于数据的认证及软件更新的认证技术。

8.1.2 密钥存储－通用指南

本节提供应用于对称和非对称密钥存储的密钥存储通用指南。

8.1.2.1 硬件存储与软件存储

通常将密钥存储于诸如硬件安全模块内等的硬件位置或存储于诸如主机计算机系统内的软件位置。实际上，密钥具体存储于下列介质中。

密钥一般存储于诸如磁盘等磁介质中、短暂使用的硅存储器内或者诸如IC卡内等长效硅存储器中。通常在密钥存储过程中，使用一系列物理方法来保护密钥，如硬件安全模块或IC卡的抗篡改功能，或对位于硬件安全模块、IC卡及主机计算机系统的操作系统的逻辑保护。

存储于磁介质的密钥应视为短暂存储。一般认为可移动磁介质的使用寿命不超过几年。

存储于非易失性硅存储器内的密钥可以有至少十年的存储寿命。

只要存储密钥，就应确保存储的密钥不被泄漏。用于存储密钥的器件应得到有效的物理安全保护，如使用防篡改封装并放置于安全的物理环境。使用密钥的器件应在使用该密钥前验证该密钥数据的完整性。验证密钥数据的完整性对于以加密形式存储于数据库内的密钥尤其重要。加密的密钥使得非授权人员无法读取该密钥的具体数值，但是对加密密钥数据的替换可能导致篡改攻击。对于存储在诸如IC卡等硬件安全器件中的密钥使用非密钥技术的校验和检查应该是足够的。对于存储在诸如数据库等的对数据更改没有基本防护的存储器的密钥应使用密码技术的MAC认证码进行数据完整性保护。

8.1.2.2 硬件安全模块

一般，一个硬件安全模块（HSM）包含分离的存储和处理组件，密钥通过内部硬件数据线在组件间传递。因此，在检测到密钥泄漏时应清空其存储器。同时硬件安全模块的硬件设计应对电磁辐射等物理信号探测等的旁路攻击具有防范功能。

用于密钥过程处理的在特定PC机上的插入式电路板应被视为硬件安全模块的一种形式，并应受到一般硬件安全模块所需的同等程度的保护。使用密码安全器件的主要目的是保护位于其中的密钥。如果使用硬件安全模块的主机系统本身不安全，则攻击者有可能绕过硬件安全模块而攻击该系统的软件功能并最终使得密钥泄漏。

需要重视的是应确保用于卡片交易中应用密码报文处理的密钥不能用于其它目的或误用该密钥，即使是授权的个人也不能执行大量的非授权密码运算（即，对单个合法卡片传来的MAC的验证失败、或对单个合法卡片不恰当地生成大量的MAC保护的安全报文或ARPC）。

发卡行应通过API在硬件安全模块内导出生成卡片密钥和生成发卡行脚本。因为数据分组的格式可能不同，在一些情况下系统内连接硬件安全模块的API支持各种灵活性以便处理输入数据的偏移量，这类实现有可能被内部攻击者利用，调用穿过API的软件以便收集和比较API返回的结果。如果加密密钥的过程既被用于密钥传递也被用于密钥管理，则可能出现上述攻击。因此需确保一种特定的密钥处理方式或特定的密钥只用于一个特定的目的而不应用于其它目的。

应禁止在一般用途的个人计算机或其它类似的非安全器件内生成密钥材料。

应执行有效的对于硬件安全模块使用的访问控制，使用有效的软件件管理控制，日志记录和持续监控。系统内硬件安全模块的API应只支持实际使用的数据格式并屏蔽其它数据格式。

8.2 IC卡应用安全

发卡行负责为持卡人选择恰当的芯片卡。为了维护人民银行、银联及广大发卡行的利益，PBOC为合适的安全产品提供了审核程序。所有相关应用产品都需通过PBOC的审核批准。

与设计用来在位于系统安全环境内使用的硬件安全模块不同，IC卡设计用于在未受保护的环境中使用。一般，IC卡的使用安全取决于其自身的抗篡改机制以保护卡片敏感信息。

卡片应用安全性的管理包括IC卡平台、应用开发安全以及发卡后应用数据安全。虽然相关具体细节超出了本指南的范围，但是遵守下面所列的内容规则对于PBOC应用的开发、实现是十分重要的，相关内容应在IC卡安全审核评估过程中予以确认。

- 卡片应用开发应在严格管理和审计的环境中进行以消除诸如恶意代码和多余的测试功能等卡片应用的不希望具有的特征。
- 密码功能的设计应尽量减少通过时间分析和功耗分析可能泄漏的密钥信息。
- 对单个密钥的使用应被过程密钥处理和安全计数器等机制有效限制。
- 发卡行卡片应用功能应被限制以消除非授权卡片数据操纵，并且应常规性地检查卡片应用数据的完整性。
- 发卡行应只使用被人民银行批准的IC卡平台。
- 卡片应用软件应在严格控制的环境中开发。该环境应当不仅在物理上是安全的，并且在管理上使用有效程序确保应用数据和应用源代码的数据完整性。
- 用于卡片个人化用途的数据应根据发卡行的数据及IT安全策略进行管理。秘密的应用数据包括密钥、PIN和其它设定的秘密数据应始终以加密形式出现并具有数据完整性保护，使得任何参与者均无法访问到这些秘密数据的明文。
- 秘密的应用数据只应在PBOC2.0借贷记应用流程定义的范围内使用而不应出现在该流程范围之外。同样应禁止任何PBOC2.0借贷记应用未定义的数据更新、重置、及增量数据的处理。

8.3 欺诈的探测

发卡行应基于其自身的风险评估使用位于卡片联机交易请求报文中的卡片数据并处理授权请求。卡片脱机处理的结果也出现在联机交易授权请求报文的终端验证结果（TVR）和发卡行应用数据（IAD）中。发卡行在决策授权时应考虑任何诸如脱机PIN验证失败等的负面结果。授权请求的其它域，如ATC等，在发卡行授权决策时也应予以评估。发卡行也应核对卡片传来的卡内个人化数据如AIP等以确认这些个人化数据是真实的。

发卡行应主动监控PBOC借贷记应用交易以探测任何IC卡及终端的欺诈性使用以及可能的伪卡。发卡行应监控的范围包括：

- 联机发送的脱机指示符－当决策是否批准或拒绝一笔交易时，应使用联机发送的位于IAD或其它联机数据内的指示符。这些指示符可能显示脱机认证失败、脱机PIN验证失败、绕开脱机PIN验证、以及其它脱机处理结果，这些结果可能预示着欺诈的发生。
- 联机密码报文－验证授权请求及清算报文中的密码报文以确认卡片是合法的。
- ATC－检查复制的ATC，检查收到的ATC值小于上次交易使用的ATC并且当前收到的ATC值与上次交易的ATC值之间有巨大的间隔。注意在重复的授权请求和当前一次PIN验证失败时的后续联机PIN验证的授权请求中使用复制的ATC是合法的。
- 在芯片卡终端读取芯片卡的磁条数据－出现这种情况可能预示着欺诈但也可能是芯片卡的数据读取问题造成。
- 芯片卡数据不一致－检查在收到的授权请求和清算报文中的芯片卡数据是否与该卡片的个人化数据及更新数据一致。
- 脚本指令－发卡行应确保发卡行发送的芯片交易报文不被已经被发卡行脚本指令锁死的卡片及应用读取。

应执行下列操作以避免欺诈及信用损失：

- 在卡片被偷走时应锁定该卡片应用或该卡片。
- 调整最低限额以反映允许的持卡人当前的脱机交易能力。

9 PBOC2.0 借贷记应用使用的密钥

本章给出了PBOC2.0借贷记应用使用的密钥。

9.1 PBOC2.0 规范借贷记应用使用的密钥

9.1.1 3DES 主密钥

在PBOC2.0借贷记应用中对联机交易报文的保护采用双倍密钥长度的3DES分组密码算法和密钥。发卡行应具有安全的方式生成和传递发卡行3DES主密钥。发卡行需要生成和使用若干发卡行3DES主密钥并利用卡片PAN和PAN序列号导出卡片的若干唯一3DES主密钥。卡片主密钥在卡片个人化时安全写入卡片。卡片个人化系统和交易及授权系统需要安装发卡行主密钥。

在卡片交易过程中使用的3DES密钥是由IC卡主密钥按照PBOC2.0规范定义的密钥分散算法导出的过程密钥。而IC卡主密钥是在卡片个人化时由发卡行主密钥分散导出的。过程密钥仅当需要使用时由主密钥分散导出，发卡行不需要存储IC卡主密钥，因为IC卡主密钥可由发卡行主密钥分散导出。

PBOC2.0借贷记应用使用的发卡行主密钥包括：

- 发卡行应用密报主密钥IMK_{AC}

  一个应用密报AC，如ARQC、ARPC、AAC或TC，是一个采用过程密钥计算的报文鉴别码MAC。该过程密钥由相应的IC卡主密钥MK_{AC}分散导出，而这个IC卡主密钥MK_{AC}由发卡行IMK_{AC}导出并在个人化时写入卡片。AC用于在卡片和发卡行交易授权系统间对交易报文进行鉴别。
- 发卡行安全报文主密钥IMK_{SMC}和IMK_{SMI}

  发卡行安全报文用于发卡后的特定交易过程中卡片和交易授权系统间的报文传输安全（如卡片锁定、应用锁定或解锁、卡片数据更新、PIN变更或PIN解锁）。在上述过程中应对报文加密和报文完整性保护使用不同的密钥，这些密钥由IC卡主密钥分散导出。发卡行安全报文主密钥IMK_{SMC}用于导出卡片安全报文加密主密钥MK_{SMC}，而发卡行安全报文主密钥IMK_{SMI}用于导出卡片安全报文完整性主密钥MK_{SMI}。安全报文的加密使用安全报文加密过程密钥SK_{SMC}，该过程密钥是由MK_{SMC}导出的。安全报文的MAC计算使用安全报文MAC过程密钥SK_{SMI}，该过程密钥是由MK_{SMI}导出的。安全报文用于卡片和发卡行交易授权系统间的特定发卡行过程处理，包括：
  - 卡片锁定

- 应用锁定或解锁
- 卡片数据更新
- PIN变更或解锁

9.1.2 卡片静态数据认证（SDA）使用的密钥

SDA需要发卡行在卡片个人化时使用发卡行RSA公私钥对的私钥对特定静态数据进行签名并将签名数据写入卡片。首先，发卡行生成发卡行RSA公私钥对，然后发卡行的公钥由IC卡根CA签发发卡行公钥证书，发卡行的私钥用于签署静态数据。在卡片个人化时，发卡行将发卡行公钥证书及签名的卡片静态数据写入卡片。发卡行证书和签名的静态数据的格式见PBOC2.0安全规范和金融IC卡借记/贷记应用根CA公钥认证规范。

终端在验证卡片SDA过程中执行下列步骤：

1. 从卡片读取发卡行公钥证书。
2. 使用存储在终端的IC卡根CA公钥验证该发卡行证书。
3. 从该发卡行证书解析出发卡行公钥。
4. 读取卡片静态签名数据并用发卡行公钥验证该静态数据的签名。

见金融IC卡借记/贷记应用根CA公钥认证规范。

9.1.3 卡片动态数据认证（DDA）使用的密钥

DDA需要卡片拥有自己的RSA公私钥对，其公钥由发卡行按照PBOC2.0的安全规范签发卡片公钥证书。卡片私钥存储在卡片中，并在卡片执行DDA过程中签署一动态签名数据由终端验证。卡片的公钥由终端在验证卡片DDA时使用。IC卡的RSA公私钥对对每张IC卡都是唯一的。该卡片公私钥对由发卡行在卡片个人化过程中生成，然后发卡行将该卡片私钥写入卡片，并对该卡片私钥进行签名生成卡片公钥证书然后将卡片公钥证书写入卡片。

在终端验证卡片DDA过程中，首先利用终端内存储的IC卡根CA公钥验证从卡片读取的发卡行公钥证书，然后从发卡行证书中解析出发卡行公钥并用该发卡行公钥验证从卡片中读取的IC卡公钥证书，然后从IC卡公钥证书中解析出卡片公钥，用该卡片公钥验证卡片发来的DDA签名数据的签名。

复合动态数据认证CDA是DDA的扩展。CDA不仅使用卡片私钥对动态交易数据签名，也对应用密码报文签名。这样可以抵御针对交易数据的攻击。

9.1.4 其它密钥

除了上述用于卡片交易过程的密钥外，发卡行还需要拥有在卡片个人化过程中用于个人化数据安全传输的密钥，见本标准第10.9节。对于多应用IC卡，除了上述密钥外可能还需要针对其它应用的密钥。

9.2 过程密钥的导出

所有PBOC2.0借贷记应用都拥有若干IC卡主密钥，这些IC卡主密钥分别由不同的发卡行主密钥导出。IC卡主密钥有MK_{AC}、MK_{SMC}、MK_{SMI}。过程密钥SK_{AC}、SK_{SMC}、SK_{SMI}由相应的IC卡主密钥分散导出，见本文10.1节。

9.3 密钥长度和使用周期

密码系统的安全性随着该系统使用的密钥的长度的增加而增强。随着硬件设备的更新以及密码算法攻击者拥有的计算资源的扩大，通常系统内特定密钥长度的密钥的安全性随着时间不断下降因而需要以长度更长的密钥取代。

本指南建议使用的对称3DES密钥为双倍长度，即112比特，加上16比特的校验位构成128比特。对于非对称公私钥对，目前PBOC2.0借贷记应用使用的最大公钥模长为1984比特。

公钥长度在很大程度上由IC卡根CA确定。公钥长度的选择需要平衡交易时间，支持设备、以及安全性诸多因素。

符合安全要求的最小公钥长度随着攻击的改进和攻击资源的变化而不断变化。密码界在2005年成功破解的RSA公钥长度为640比特，即成功分解了长度为640比特的RSA公钥模。学术界的下一个攻击目标是长度为704比特的RSA公钥模。一般认为分解长度为512比特的RSA公钥模在计算上是可行的。

在DDA的三级公钥体系中根CA公私钥对的安全性无疑最为重要，其泄漏将导致整个公钥体系的崩溃，同时使用根CA公钥的数据验证只发生在终端，终端的计算能力通常远大于卡片计算能力。因此对根CA公私钥的安排应考虑最大安全性，公钥模长及使用周期应在可能的情况下严格控制。

发卡行公私钥对比卡片公私钥对的安全更为重要，因为单个卡片私钥的泄漏只影响该卡片的脱机交易，而卡片交易的风险控制，可以将伪卡的风险限定在一个较小的范围，此外卡片的计算能力通常比终端低，所以选择小于根CA公钥长度的卡片公钥长度在确保适当的安全性需求的基础上可以改进卡片执行签名运算的效率。

发卡行私钥的泄密对发卡行运行有极大负面影响，将导致大批能够通过脱机认证的伪卡的出现，或导致所有用该私钥签名的SDA卡和DDA卡的失效。因此发卡行应使用更大的公钥模长。发卡行公钥模长对卡片进行动态数据的签名时间没有影响。使用根CA公钥验证发卡行公钥证书，使用发卡行公钥验证静态数据或卡片证书是在终端完成的，终端的运算功能通常远大于卡片，所以应选择发卡行公钥模长与签发该发卡行证书的根CA公钥模长相同以最大体现安全性并兼顾运算效率。

综上所述本指南根据金融IC卡借记/贷记应用根CA公钥认证规范规定对于IC卡根CA公钥长度及使用周期见表3（经评估的2008年度银联根CA公钥有效期方案）：

表 3：PBOC 公钥方案及有效期

公钥模长	公钥指数	公钥有效期
1024 比特	3 或 $2^{16}+1$	2010 年 12 月 31 日
1152 比特	3 或 $2^{16}+1$	2016 年 12 月 31 日
1408 比特	3 或 $2^{16}+1$	2018 年 12 月 31 日
1984 比特	3 或 $2^{16}+1$	2018 年 12 月 31 日

注意，银联将对IC卡根CA公钥长度及有效期做周期性审核，有可能在未来对表3的根CA公钥方案作修订。

根据PBOC2.0规范和金融IC卡借记/贷记应用根CA公钥认证规范，卡片公钥模长N_{IC}、发卡行公钥模长N_I和根CA公钥模长N_{CA}应满足：$N_{IC} \leq N_I \leq N_{CA}$。公钥指数可选3或$2^{16}+1$。本指南建议卡片公钥模长$N_{IC}$应不小于768比特。

本指南建议发卡行公钥模长与（签署该发卡行公钥证书的）根CA公钥模长相同，公钥指数只选3。这是因为发卡行公钥模长选择为与签发该发卡行公钥证书的根CA公私钥对中公钥模长相同可以最大体现安全性与运行效率最佳化。公钥指数为3或$2^{16}+1$在安全性上没有差别但公钥指数选3可大大提高运行效率。

发卡行对称主密钥应做周期性（每年）更新。这并不意味发卡行必须周期性变更已颁发的卡片内的密钥，而是发卡行必须能够一次管理多个密钥形式，一旦由该密钥形式分散导出的卡片密钥对应的卡片有效期到期并且随后没有任何争议时，发卡行应将该发卡行密钥形式销毁。

9.4 PBOC2.0 借贷记应用密钥汇总

本节汇总PBOC2.0借贷记应用交易过程中所用的密钥。汇总的密钥见表4。

表 4：PBOC2.0 借贷记应用密钥汇总

名称	标识	类型	共享	用途	生成	注释
IC 卡根 CA 公钥	P_{CA}	RSA 公钥	根 CA、发卡行、终端	验证发卡行公钥证书	根 CA	公钥(以自签名证书形式传输)
IC 卡根 CA 私钥	S_{CA}	RSA 私钥	不共享	签发发卡行证书	根 CA	存储，使用于根 CA 加密机
发卡行 RSA 公私钥对	P_I S_I	RSA	公钥：根 CA、发卡行、卡片； 私钥：不共享	公钥：验证 SDA、验证卡片证书 私钥：生成 SDA、	发卡行	发卡行公钥由根 CA 签发证书，该证书在卡片个人

				签发卡片证书。		化时写入卡片
卡片 RSA 公私钥对	P_{IC} S_{IC}	RSA	私钥：发卡行、卡片； 公钥：卡片（其证书在交易中传给终端）	私钥：生成 DDA，公钥：验证 DDA	由发卡行生成并在个人化时写入卡片	公私钥对每张卡片唯一，卡片公钥由发卡行私钥签发证书
发卡行应用密文主密钥	IMK_{AC}	3DES	发卡行、个人化和授权系统	导出卡片应用密文主密钥 MK_{AC}	发卡行	由个人化和授权系统共享
发卡行安全报文加密主密钥	IMK_{SMC}	3DES	发卡行、个人化和授权系统	导出卡片安全报文加密主密钥 MK_{SMC}	发卡行	由个人化和授权系统共享
发卡行安全报文 MAC 主密钥	IMK_{SMI}	3DES	发卡行、个人化和授权系统	导出卡片安全报文 MAC 主密钥 MK_{SMI}	发卡行	由个人化和授权系统共享
卡片应用密文主密钥	MK_{AC}	3DES	发卡行、卡片	导出卡片应用密文过程密钥 SK_{AC}	发卡行用 IMK_{AC} 导出并在个人化时写入卡片	对每张卡唯一
卡片安全报文加密主密钥	MK_{SMC}	3DES	发卡行、卡片	导出卡片安全报文加密过程密钥	发卡行用 IMK_{SMC} 导出并在个人化时写入卡片	对每张卡唯一
卡片安全报文 MAC 主密钥	MK_{SMI}	3DES	发卡行、卡片	导出卡片安全报文 MAC 过程密钥	发卡行用 IMK_{SMI} 导出并在个人化时写入卡片	对每张卡唯一
卡片应用密文过程密钥	SK_{AC}	3DES	仅在交易过程中短暂存在，由卡片及授权系统共享	生成应用密文	卡片（授权系统）用 MK_{AC} 和 ATC 导出	仅用于一个交易过程，然后丢弃
卡片安全报文加密过程密钥	SK_{SMC}	3DES	仅在交易过程中短暂存在，由卡片及授权系统共享	对安全报文加解密	卡片（授权系统）用 MK_{SMC} 和 ATC 导出	仅用于一次脚本处理，然后丢弃
卡片安全报文 MAC 过程密钥	SK_{SMI}	3DES	仅在交易过程中短暂存在，由卡片及授权系统共享	对安全报文计算、验证 MAC	卡片（授权系统）用 MK_{SMI} 和 ATC 导出	仅用于一次脚本处理，然后丢弃

10 PBOC2.0借贷记应用所用的密码算法

本章提供PBOC2.0借贷记应用所用的密码算法及涉及的密钥。

10.1 卡片主密钥及过程密钥生成使用的密码算法

10.1.1 卡片主密钥生成密码算法

发卡行拥有三个主密钥：应用密文发卡行主密钥IMK_{AC}、安全报文加密发卡行主密钥IMK_{SMC}、安全报文MAC发卡行主密钥IMK_{SMI}。

卡片也具有三个主密钥：应用密文卡片主密钥MK_{AC}、安全报文加密卡片主密钥MK_{SMC}、安全报文MAC卡片主密钥MK_{SMI}。

卡片主密钥由相应的发卡行主密钥利用3DES算法对卡片PAN及PAN序列号的链接值加密生成的密文得出。

MK_{AC}＝3DES(IMK_{AC})[输入]||3DES(IMK_{AC})[输入⊕ ‘FF FF FF FF FF FF FF FF’]

MK_{SMC}＝3DES(IMK_{SMC})[输入]||3DES(IMK_{SMC})[输入⊕ ‘FF FF FF FF FF FF FF FF’]

MK_{SMI}＝3DES(IMK_{SMI})[输入]||3DES(IMK_{SMI})[输入⊕ ‘FF FF FF FF FF FF FF FF’]

其中输入由卡片PAN和PAN序列号链接值经过处理构成，当PAN和PAN序列号的链接为8字节（16个数字）长，则以数字格式的该数据链接Y作为输入；若PAN和PAN序列号的链接小于16个数字，则右对齐，前面补16进制的0以获得8字节数字格式的Y；如果PAN和PAN序列号的链接的长度至少右16个数字，则以Y＝该链接的最右边16个数字。

3DES为双倍密钥长度的三轮标准DES运算：密钥为(密钥A||密钥B)，密钥A和密钥B都是56比特单DES密钥，3DES的三轮DES运算如下：

1. 使用密钥A对输入数据进行单DES加密运算，
2. 使用密钥B对上一步的结果做单DES解密运算，
3. 使用密钥A对上一步结果做单DES加密运算。

10.1.2 卡片过程密钥生成密码算法

卡片联机交易均采用卡片主密钥导出的过程密钥来计算联机交易应用密文和联机交易安全报文。其中应用密文由卡片应用过程密钥SK_{AC}计算、安全报文加密采用卡片安全报文加密过程密钥SK_{SMC}计算、安全报文MAC采用卡片安全报文MAC过程密钥SK_{SMI}计算。

卡片过程密钥SK_{AC}、SK_{SMC}、SK_{SMI}分别由卡片主密钥MK_{AC}、MK_{SMC}、MK_{SMI}导出。过程密钥的导出为使用卡片相应主密钥对卡片ATC值的变换进行加密得出的密文就是相应的过程密钥。对卡片ATC的变换为：

X:= ‘00’||‘00’||‘00’||‘00’||‘00’||‘00’||（ATC）

Y:=‘00’||‘00’||‘00’||‘00’||‘00’||‘00’||(ATC⊕‘FFFF’)

过程密钥A:=3DES(MK)[X]

过程密钥B:=3DES(MK)[Y]

最终过程密钥为：过程密钥A||过程密钥B。这里“||”表示数据链接。因此：

SK_{AC}=3DES(MK_{AC})[‘00’||‘00’||‘00’||‘00’||‘00’||‘00’||ATC]||3DES(MK_{AC})[‘00’||‘00’||‘00’||‘00’||‘00’||‘00’||(ATC⊕‘FFFF’)]

SK_{SMC}=3DES(MK_{SMC})[‘00’||‘00’||‘00’||‘00’||‘00’||‘00’||ATC]||3DES(MK_{SMC})[‘00’||‘00’||‘00’||‘00’||‘00’||‘00’||(ATC⊕‘FFFF’)]

SK_{SMI}=3DES(MK_{SMI})[‘00’||‘00’||‘00’||‘00’||‘00’||‘00’||ATC]||3DES(MK_{SMI})[‘00’||‘00’||‘00’||‘00’||‘00’||‘00’||(ATC⊕‘FFFF’)]

10.2 卡片与发卡行联机应用密文处理

卡片联机交易中卡片需要使用密码算法生成TC、AAC、ARQC，发卡行需要生成ARPC。

10.2.1 联机应用密文生成的密码算法

联机应用密文ARQC、AAC、TC的生成使用的密码算法为遵照ISO/IEC 9797-1规范算法3，采用CBC模式的64位分组加密算法构成的MAC算法，基本分组加密算法为DES算法，即标准DES算法。

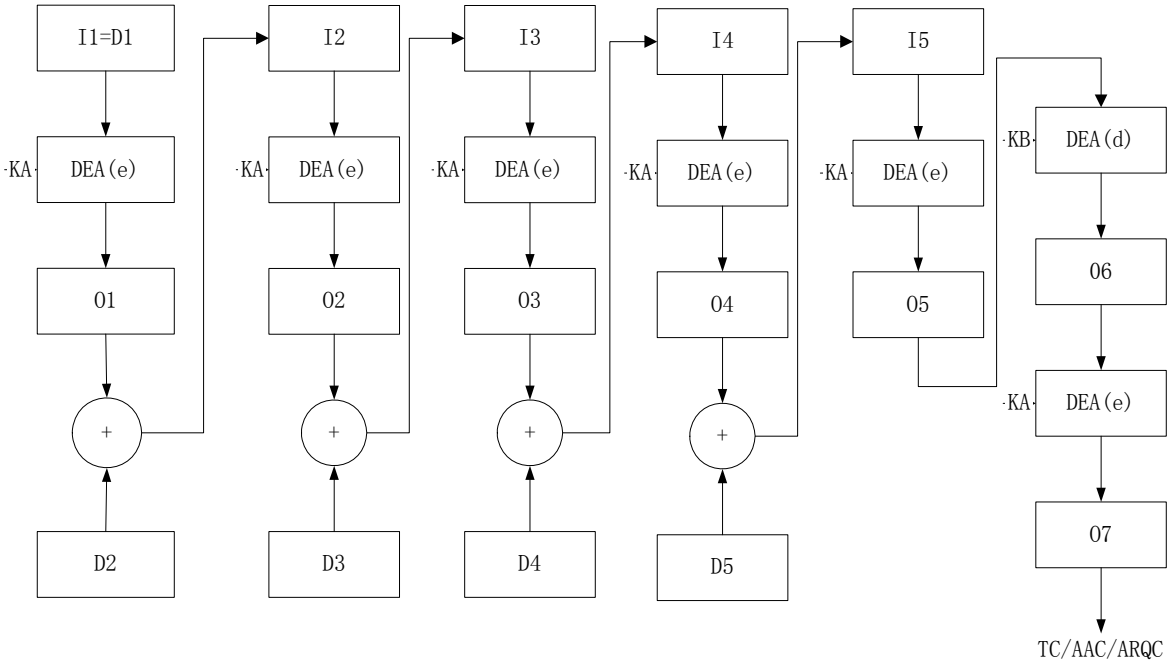
ARQC:=MAC(SK_{AC})[输入]

TC:＝MAC(SK_{AC})[输入]

AAC:＝MAC(SK_{AC})[输入]

MAC算法构成如下：

将算法的输入分成8字节的数据分组：D_1，D_2，…D_n，K_A为过密钥A，K_B为过程密钥B，按照如下算法计算：



说明：

I = 输入	D = 数据块
DEA(e)= 数据加密算法（加密模式）	KA = 密钥A
DEA(d)= 数据加密算法（解密模式）	KB = 密钥B
O = 输出	+ = 异或

图 3：计算联机应用密文的 MAC 算法

ARPC的计算采用3DES算法如下：

ARPC:=3DES(SK_{AC})[输入]

10.2.2 联机应用密文的生成

10.2.2.1 ARQC、TC 和 AAC 的生成

联机应用密文ARQC、TC和AAC计算采用下列算法：

ARQC:=MAC(SK_{AC})[输入]

TC:＝MAC(SK_{AC})[输入]

AAC:＝MAC(SK_{AC})[输入]

算法的输入构成如下：

在计算ARQC、AAC和TC时的输入均为按照表4顺序排列的表5中的数据（见PBOC2.0卡片规范）：

表 5：TC/AAC/ARQC 数据元顺序

数据元	来自终端的数据	在交易证书（TC）哈希中的顺序	卡片内数据
授权金额	●	●	
其它金额	●	●	
终端国家代码	●	●	
终端验证结果	●	●	
交易货币代码	●	●	
交易日期	●	●	
交易类型	●	●	
不可预知数	●	●	
应用交互特征（AIP）			●
应用交易计数器（ATC）			●
卡片验证结果（CVR）			●

表4数据的按原有顺序的链接值作为MAC算法的输入（若不是8字节的整数倍则右补‘80’和最少个数的‘00’构成8字节的整数倍）。

MAC算法遵照ISO/IEC 9797-1规范算法3，采用CBC模式的64位分组加密算法计算如下：

将算法输入分成8字节的数据块：$X_1, X_2, \ldots, X_K$。依照ISO/IEC 9797-1算法3：$H_{K+1} := ALG(K_{SL})[ALG^{-1}(K_{SR})[H_K]]$。这里$H_0$的初始值 H_0 :=（‘00’ || ‘00’ || ‘00’ || ‘00’ || ‘00’ || ‘00’ || ‘00’ || ‘00’）。（见PBOC2.0安全规范第11.1.2.1）。

10.2.2.2 ARPC 的生成

ARPC =3DES(SK_{AC})[输入]，其中输入的构成如下：

将应用密文AC（一般是ARQC，有时是AAC）与外部响应码ARC异或（外部响应码先左对齐并右补6个“00”字节），将这个异或结果值作为输入（见PBOC2.0卡片规范第D.1）。即：

ARPC=3DES(SK_{AC})[输入]＝3DES(SK_{AC})[AC ⊕ (ARC||(‘00’ || ‘00’ || ‘00’ || ‘00’ || ‘00’ || ‘00’)]

10.3 卡片与发卡行安全报文处理

安全报文处理用于发卡行脚本指令处理，分为加密处理和完整性处理。PBOC2.0借贷记应用涉及的发卡行脚本命令有：应用锁定、应用解锁、卡片锁定、PIN修改/解锁、设置数据和修改记录。其中的PIN修改/解锁脚本处理需要对PIN进行加密，其余的发卡行脚本仅需计算MAC值。

10.3.1 安全报文过程密钥

安全报文过程密钥的计算完全按照本文第9.2节如下：

安全报文加密过程密钥SK_{SMC}的计算使用卡片安全报文加密主密钥MK_{SMC}、安全报文MAC过程密钥SK_{SMI}的计算使用卡片安全报文MAC主密钥MK_{SMI}。

10.3.2 安全报文密码算法

安全报文加密的算法为：ECB模式的3DES（SK_{SMC}）[输入]，见PBOC2.0安全规范第11.1.1节和本文下一节。

10.3.2.1 安全报文加密算法

对数据的加密采用分组长度为64位（8字节）分组加密算法，是电子密码本（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：

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MSG被拆分为8字节或16字节的块X_1，X_2，…，X_K。

2. 密文计算

ECB模式

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

因此当i ＝1，2，…，K时分别计算：

$Y_i := ALG(K_S)[X_i]$。

10.3.2.2 安全报文加密算法输入

安全报文加密仅在发卡行生成PIN修改脚本命令时使用，此时需要对新的PIN进行加密。

不使用当前PIN修改PIN的PIN加密数据的产生过程按照下列步骤进行：

1. 生成8字节PIN数据块D3：

a) 生成一个8字节数据块D1：

字节1		字节2		字节3		字节4		字节5	字节6	字节7	字节8
0	0	0	0	0	0	0	0	安全报文加密过程密钥A的最右边4个字节			

b) 生成第二个8字节数据块D2：

字节1		字节2		字节3		字节4		字节5		字节6		字节7		字节8	
0	N	P	P	P	P	P/F	P/F	P/F	P/F	P/F	P/F	P/F	P/F	F	F

N：新PIN的数字个数（16进制）

P：新PIN值，长度4-12个数字（2-6字节）

c) D1和D2执行异或得到D3

2. 使用当前PIN生成8字节数据块D4：

字节1		字节2		字节3		字节4		字节5		字节6		字节7		字节8	
P	P	P	P	P/0	P/0	P/0	P/0	P/0	P/0	P/0	P/0	0	0	0	0

3. 将数据块D3和D4执行异或得到D。

上述D即为安全报文加密的输入。

使用当前PIN修改PIN的PIN加密数据的产生过程按照下列步骤进行：

1. 生成8字节PIN数据块D3：

a) 生成一个8字节数据块D1：

字节1		字节2		字节3		字节4		字节5	字节6	字节7	字节8
0	0	0	0	0	0	0	0	安全报文加密过程密钥A的最右边4个字节			

b) 生成第二个8字节数据块D2：

N：新PIN的数字个数（16进制）

P：新PIN值，长度4-12个数字（2-6字节）

c) D1和D2执行异或得到D3

2. 使用当前PIN生成8字节数据块D4：

字节1		字节2		字节3		字节4		字节5		字节6		字节7		字节8	
P	P	P	P	P/0	P/0	P/0	P/0	P/0	P/0	P/0	P/0	0	0	0	0

3. 将数据块D3和D4执行异或得到D。

上述D即为安全报文加密的输入。

10.3.2.3 安全报文 MAC 算法

安全报文MAC的算法为：MAC＝MAC4(SK_{SMI})[输入]，这里MAC4表示MAC的前4个字节，MAC的计算可参见PBOC2.0第11.1.2.1节,输入的构成见PBOC2.0附录C。

MAC的长度s为4字节。

计算一个4字节的MAC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SK_{SMI}对任意长度的报文MSG计算MAC值S的步骤如下。

填充并分块：依据GB/T 16649.4（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MSG:=(MSG ||‘80’||‘00’||‘00||…||‘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，… ,X_K。

MAC过程密钥为SK_{SMI}＝(K_{SL} ‖ K_{SR})，其中K_{SL}为SK_{SMI}的左半部分或密钥A，K_{SR}为SK_{SMI}的右半部分或密钥B。

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

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

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

依照ISO/IEC 9797-1算法3：$H_{K+1} := ALG(K_{SL})[ALG^{-1}(K_{SR})[H_K]]$。

MAC值S等于H_{K+1}的4个最高位字节。

这里的ALG指标准DES算法。

10.3.2.4 安全报文 MAC 算法输入

用于发卡行标本处理的安全报文类型有应用锁定、应用解锁、卡片锁定、PIN修改/解锁、设置数据和修改记录。这些安全报文均需要计算MAC。

命令中需要加密的数据（PIN修改）加密以后再计算MAC。MAC使用对称密钥算法计算的，步骤如下：

1. 初始值为8字节全零。（此步骤可省略）
2. 下列数据按顺序排列得到一个数据块D：
   - CLA，INS，P1，P2，Lc（Lc的长度包括MAC的长度）
   - ATC（对于发卡行脚本处理，此ATC在请求中报文中上送）
   - 应用密文（对于发卡行脚本处理，此应用密文通常是ARQC，或AAC，在请求报文中上送）
   - 命令数据域中的明文或密文数据（如果存在）
3. 将上述数据块D分成8字节长的数据块D_1，D_2，D_3…最后一块数据块的字节长度为1到8。
4. 如果最后一块数据块的长度为8字节，后面补8字节数据块：‘80 00 00 00 00 00 00 00’，执行步骤5。如果最后一块数据块的长度小于8字节，后面补一个字节80，如果长度到8字节，执行步骤5。如果仍然不够8字节，补00直到8字节。
5. 用MAC过程密钥SK_{SMI}对数据块进行加密。
6. MAC的计算结果为8字节，从最左边的字节开始取4字节。

10.4 脱机数据认证处理

卡片脱机数据认证分静态数据认证、动态数据认证和复合动态数据认证/应用密文生成。

10.4.1 金融 IC 卡借记/贷记应用公钥认证体系概述

金融IC卡借记/贷记应用公钥认证体系符合《中国金融集成电路（IC）卡规范 》，为所有遵循该标准的金融IC卡借记/贷记提供公钥认证服务，也同时为其它金融IC卡数据认证提供途径（如成员机构终端提供的对外卡数据认证）。

按照《中国金融集成电路（IC）卡规范 》和金融IC卡借记/贷记应用根CA公钥认证规范，金融IC卡借记/贷记应用公钥认证体系包括根CA、各发卡行CA及其发行的银联标准金融IC卡、收单机构及其ATM和/或POS。金融IC卡借记/贷记应用公钥认证系统使用公钥密码技术进行金融IC卡静态数据、动态数据或复合数据的生成及认证，以提供高度安全的金融IC卡交易认证服务。

—根CA负责生成根CA公私钥对并管理根CA的公私钥信息、签发发卡行CA公钥证书，将根CA公钥信息安全传递给收单机构，是金融IC卡借记/贷记应用公钥认证体系的信任顶点。

—各发卡行CA是根CA的子CA，负责生成发卡行CA的公私钥对，向根CA申请并管理自己的公钥证书。此外，发卡行CA与发卡系统交互，为IC卡签署静态应用数据以便进行静态数据认证（SDA），或生成IC卡公私钥对、签发IC卡公钥证书以便进行动态数据认证（DDA），同时将自己的公钥证书、IC卡公钥证书、IC卡私钥、RID、根CA公钥索引也写入IC卡。

—收单机构负责建立终端管理系统，将根CA公钥分发到受理终端（POS/ATM），并对远程终端进行设备管理、状态监控及信息管理（包括程序、参数下载）。

在《中国金融集成电路（IC）卡规范 》标准卡支付系统中，IC卡存放IC卡公钥证书及IC卡私钥（或静态数据）和发卡行公钥证书、RID以及根CA公钥索引；受理终端存放根CA公钥、公钥索引和相关的RID。

在支付过程中，受理终端通过验证IC卡的应用数据的签名进行IC卡数据认证，其过程是：首先读取IC卡内的RID、IC卡公钥证书、发卡行公钥证书和根CA公钥索引，通过RID和根CA公钥索引定位特定的根CA公钥，通过根CA公钥验证发卡行公钥证书，通过发卡行公钥验证IC卡公钥证书，通过IC卡公钥验证IC卡内的动态签名或直接利用发卡行公钥验证IC卡内的静态签名。整个IC卡数据认证完全离线进行。见图4：

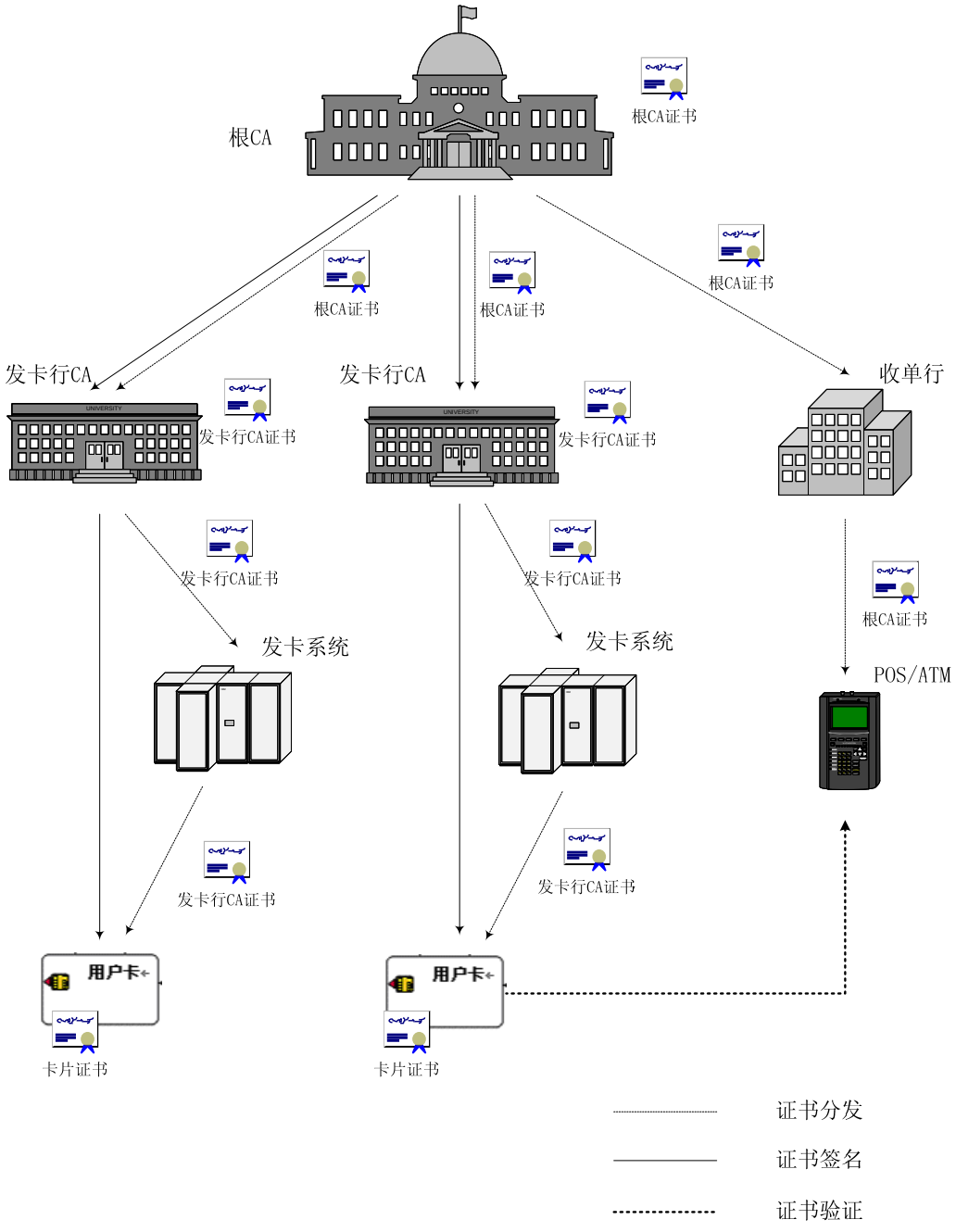


图 4：IC 卡根 CA 公钥认证体系

10.4.2 脱机认证过程使用的密钥

1. 静态签名数据的生成及验证使用的密钥有：
   - 用于签发发卡行公钥证书的IC卡根CA公私钥对
   - 用于签发卡片静态数据的发卡行公私钥对
2. 动态数据认证的生成及验证使用的密钥有：
   - 用于签发发卡行公钥证书的IC卡根CA公私钥对
   - 用于签发卡片公钥证书的发卡行公私钥对
   - 用于签发卡片动态数据的卡片公私钥对
3. 复合动态数据认证/应用密文生成及验证过程使用的密钥除了上述动态数据认证使用的密钥外还包括卡片应用密文过程密钥，见本文第10.1节。

10.4.3 脱机认证过程使用的密码算法

脱机认证使用的签名算法为RSA算法、使用的哈希函数为SHA-1，见PBOC2.0安全规范第12章。

10.4.4 脱机认证生成和验证过程

10.4.4.1 静态数据认证

静态数据认证由终端验证卡片中的静态数据的数字签名来完成。其目的是确认存放在银联标准IC卡中关键的静态数据的合法性，可以发现在卡片个人化以后对卡内的发卡行数据未经授权的改动，能有效地检测银联标准IC卡内关键静态数据的真实性，如图5所示。

整个银联标准IC卡静态数据认证的过程说明如下：

1） 发卡行的密钥管理系统产生发卡行公/私钥对P_I和S_I，并将公钥P_I传送至根CA；

2） 根CA用自己的私钥S_{CA}对发卡行公钥P_I进行数字签名，产生发卡行公钥证书，连同根CA公钥证书（包括RID及根CA公钥索引）返回给发卡行密钥管理系统；

3） 发卡行密钥管理系统用发卡行私钥S_I对卡片静态数据进行数字签名，将签名结果、发卡行公钥证书、RID及根CA公钥索引传送至发卡系统；

4） 发卡系统在个人化时将静态数字签名、发卡行公钥证书、RID以及根CA公钥索引写入每一张卡片中；

5） 根CA将其公钥P_{CA} 、RID、根CA公钥索引及其它相关信息经收单机构传送至终端管理系统；

6） 收单机构终端管理系统把根CA公钥P_{CA}、RID、根CA公钥索引及其它相关信息下载至终端；

7） 银联标准IC卡进行交易时，脱机静态数据认证过程如下：

- 终端从卡片中读取发卡行公钥证书及签名数据，使用根CA公钥索引和RID找到根CA公钥P_{CA}，由P_{CA}恢复出发卡行公钥P_I并验证其有效性；
- 终端使用恢复的发卡行公钥P_I验证卡片签名数据的有效性。

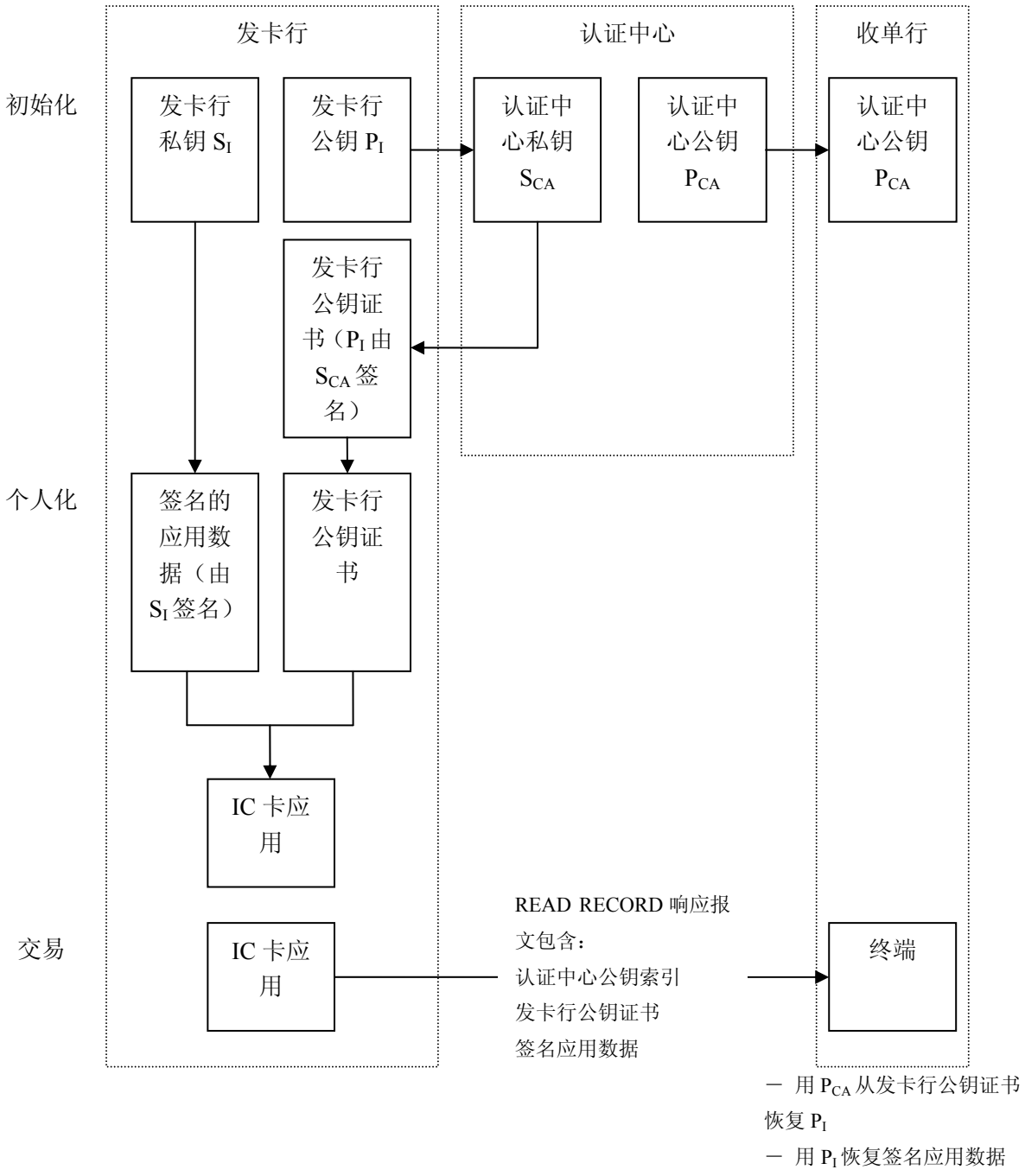


图 5：SDA 证书和公钥体系结构

10.4.4.2 动态数据认证

在动态数据认证（DDA）过程中，终端验证卡片上的静态数据以及卡片产生的当前动态交易数据的签名。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，更重要的是DDA还能确认卡片的真实性，防止卡片的非法复制和伪造。

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）。银联标准IC卡动态数据认证如图6所示。

在这种方式下，银联标准IC卡将来自卡片的动态交易数据以及由动态数据认证数据对象列表(DDOL)所标识的终端数据生成一个数字签名（见《中国金融集成电路（IC）卡规范 》）。

银联标准IC卡标准动态数据认证整体过程说明如下：

1） 发卡行密钥管理系统产生发卡行公私钥对S_I和P_I，并将发卡行公钥P_I传送至根CA；

2） 根CA用自己的私钥S_{CA}对发卡行公钥进行数字签名，产生发卡行公钥证书，连同根CA公钥证书（包括RID和根CA公钥索引）返回给发卡行密钥管理系统；

3） 发卡行密钥管理系统为每一张银联标准IC卡产生一对公私钥对S_{ICC}和P_{ICC}，并用发卡行私钥S_I对IC卡公钥P_{ICC}进行数字签名，产生IC卡公钥证书；

4） 发卡行密钥管理系统将发卡行公钥证书、IC卡公钥证书、IC卡私钥、RID以及根CA公钥索引传送至发卡系统；

5） 发卡系统在个人化时将发卡行公钥证书、IC卡公钥证书、IC卡私钥、RID及根CA公钥索引写入卡片中；

6） 根CA将其公钥P_{CA} 、RID、根CA公钥索引及其它相关信息经收单机构传送至终端管理系统；

7） 收单机构终端管理系统把根CA公钥P_{CA}、RID、根CA公钥索引及其它相关信息下载至终端；

8） 银联标准IC卡进行交易时的脱机标准动态数据认证过程如下：

- 终端从卡片读取发卡行公钥证书、IC卡公钥证书、RID以及根CA公钥索引，利用RID和根CA公钥索引定位根CA公钥P_{CA}，使用根CA公钥P_{CA}恢复出发卡行公钥P_I并验证其有效性，使用恢复的发卡行公钥P_I恢复出IC卡公钥P_{ICC}并验证其有效性；
- 终端向IC卡发送内部认证命令（INTERNAL AUTHENTICATE）（见《中国金融集成电路（IC）卡规范 》）请求一个动态签名；卡片对内部认证命令中的终端数据和IC卡交易动态数据进行连接，由IC卡私钥S_{ICC}对该连接数据进行数字签名并返回给终端；
- 终端使用IC卡公钥P_{ICC}对上一步骤的数字签名进行验证。

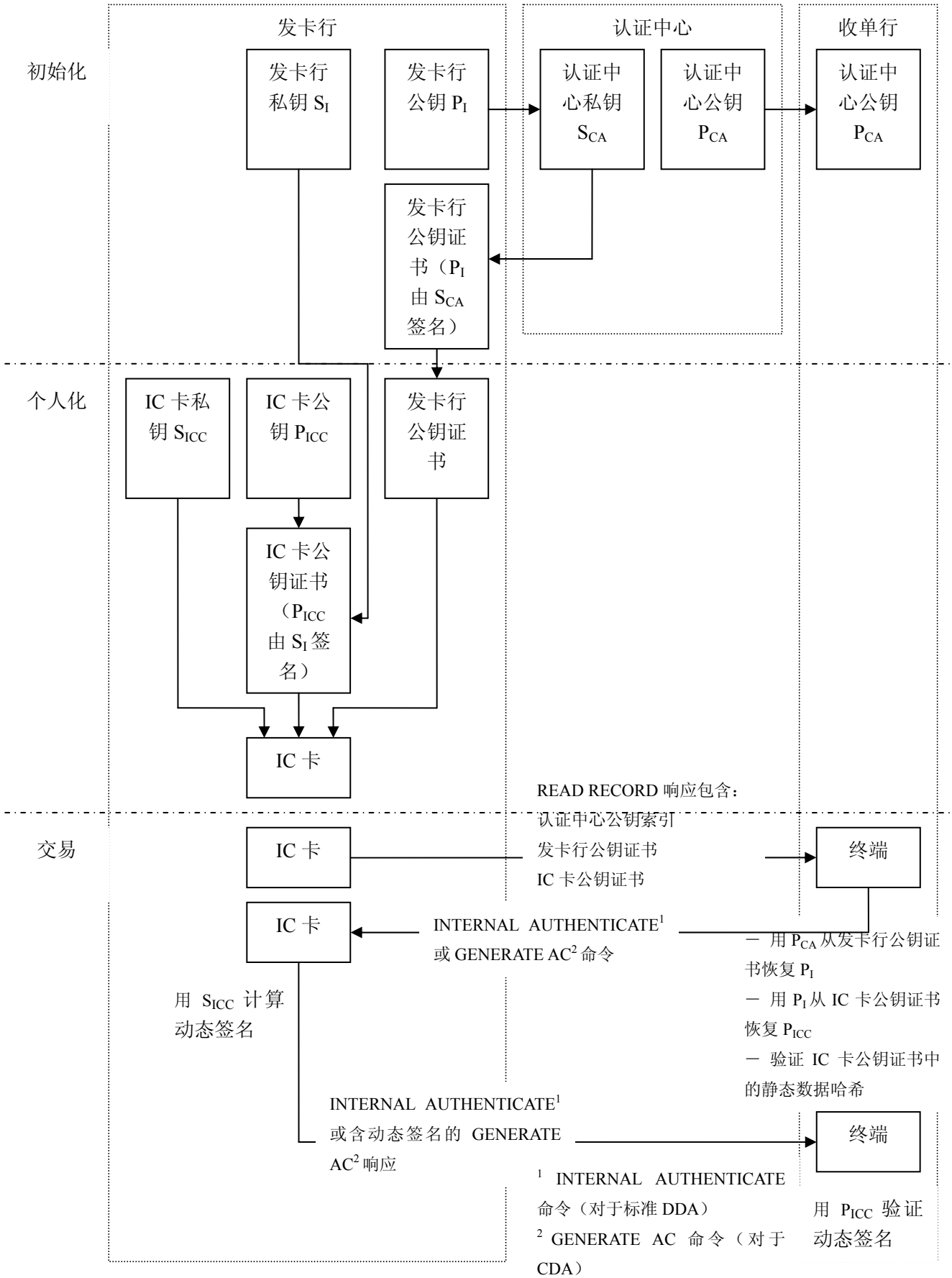

发卡行
认证中心
收单行
初始化
发卡行私钥 S_I
发卡行公钥 P_I
认证中心私钥 S_{CA}
认证中心公钥 P_{CA}
认证中心公钥 P_{CA}
发卡行公钥证书（P_I 由 S_{CA} 签名）
个人化
IC 卡私钥 S_{ICC}
IC 卡公钥 P_{ICC}
发卡行公钥证书
IC 卡公钥证书（P_{ICC} 由 S_I 签名）
IC 卡
READ RECORD 响应包含：
认证中心公钥索引
发卡行公钥证书
IC 卡公钥证书
交易
IC 卡
终端
IC 卡
INTERNAL AUTHENTICATE[1] 或 GENERATE AC[2] 命令
— 用 P_{CA} 从发卡行公钥证书恢复 P_I
— 用 P_I 从 IC 卡公钥证书恢复 P_{ICC}
— 验证 IC 卡公钥证书中的静态数据哈希
用 S_{ICC} 计算动态签名
INTERNAL AUTHENTICATE[1] 或含动态签名的 GENERATE AC[2] 响应
终端
用 P_{ICC} 验证动态签名
[1] INTERNAL AUTHENTICATE 命令（对于标准 DDA）
[2] GENERATE AC 命令（对于 CDA）

图 6：DDA 证书和公钥体系结构

10.4.4.3 复合动态数据认证/应用密文生成

该方式在第一个请求应用密文命令发出后执行（见《中国金融集成电路（IC）卡规范 》）。银联标准IC卡将来自卡片的数据包括应用密文以及来自终端的数据生成一个数字签名。

银联标准IC卡复合动态数据认证/应用密文生成的整体过程如下：

1） 发卡行密钥管理系统产生发卡行公私钥对S_I和P_I，并将发卡行公钥P_I传送至根CA；
2） 根CA用自己的私钥S_{CA}对发卡行公钥进行数字签名，产生发卡行公钥证书，连同根CA公钥证书（包括RID和根CA公钥索引）返回给发卡行密钥管理系统；
3） 发卡行密钥管理系统为每一张银联标准IC卡产生一对公私钥对S_{ICC}和P_{ICC}，并用发卡行私钥S_I对IC卡公钥P_{ICC}进行数字签名，产生IC卡公钥证书；
4） 发卡行密钥管理系统将发卡行公钥证书、IC卡公钥证书、IC卡私钥、RID以及根CA公钥索引传送至发卡系统；
5） 发卡系统在个人化时将发卡行公钥证书、IC卡公钥证书、IC卡私钥、RID及根CA公钥索引写入卡片中；
6） 根CA将其公钥P_{CA} 、RID、根CA公钥索引及其它相关信息经收单机构传送至终端管理系统；
7） 收单机构终端管理系统把根CA公钥P_{CA}、RID、根CA公钥索引及其它相关信息下载至终端；
8） 银联标准IC卡进行交易时的脱机复合动态数据认证过程如下：
   - 终端从IC卡中读取发卡行公钥证书、IC卡公钥证书、RID及根CA公钥索引。
   - 终端使用RID和根CA公钥索引定位根CA公钥P_{CA}，使用根CA公钥P_{CA}验证发卡行公钥证书的签名并恢复出发卡行公钥P_I。
   - 终端使用发卡行公钥P_I验证IC卡公钥证书的签名并恢复出IC卡公钥P_{ICC}。
   - 终端生成一不可预知数并与其它相关数据一并传给IC卡。
   - IC卡使用其自身的私钥S_{ICC}对收到的终端数据（包括不可预知数、交易数据）和其它IC卡数据（包括TC/ARQC）做数字签名并发送给终端。
   - 终端使用IC卡公钥P_{ICC}验证IC卡传递的签名数据。

卡片脱机认证处理的细节见PBOC2.0安全规范的第五章和金融IC卡借记/贷记应用根CA公钥认证规范。

10.5 IC 卡根 CA 公钥文件

IC卡根CA公钥证书以IC卡根CA公钥文件形式进行传递，见金融IC卡借记/贷记应用根CA公钥认证规范。

根CA公钥文件名格式为：01010000.CAA。其中：

——01010000 标识银联借记/贷记服务；

——C 标识中国银联；

——AA 为根 CA 的公钥索引，以 0xAA 表示。

10.5.1 根 CA 公钥文件内容

根CA公钥文件是二进制数据，其格式和内容如表6所示。

表 6：根 CA 公钥文件内容

字段名	长度（字节数）
未签名根 CA 公钥输出扩展	$35+N_{CA}+e_{CA}$ 这里 N_{CA} 和 e_{CA} 分别是根 CA 公钥模长度和公钥指数长度，以字节数表示
自签名根 CA 公钥	N_{CA}

10.5.2 未签名根 CA 公钥输出扩展

未签名根CA公钥输出扩展是根CA公钥文件的第一部分，其内容如表7所示。

表 7：未签名根 CA 公钥输出扩展

字段名	长度（字节数）	描述	格式
记录头	1	十六进制 ‘20’	b
服务标识	4	标识一个中国银联借记贷记服务，将相应应用的私有应用标识扩展(PIX)，右补十六进制‘0’构成。 ‘01010000’ = 借、贷记 ‘01010100’ = 借记 ‘01010200’ = 贷记 ‘01010300’ = 准贷记 ‘01010600’= 储值型电子现金	b
根 CA 公钥模长	2	根 CA 公钥模的长度 N_{CA}，以十六进制表示。N_{CA} 是一个偶数	b
根 CA 公钥算法标识	1	表示生成根 CA 公钥的密码算法	b
根 CA 公钥指数长度	1	根 CA 公钥指数长度 e_{CA}，以十六进制表示	b
注册的应用提供商标识 (RID)	5	标识银联 RID，为十六进制‘A000000333’	b
根 CA 公钥索引	1	唯一标识根 CA 公钥	b
根 CA 公钥模	N_{CA}	根 CA 公钥模	b
根 CA 公钥指数	e_{CA}	根 CA 公钥指数	b
哈希值	20	本表第 6 到第 9 项的（从 RID 到根 CA 公钥指数）连接数据的哈希值	b

注：储值型电子现金，或称为纯电子现金，即后台无对应的借记/贷记账户

10.5.3 自签名根 CA 公钥

自签名根CA公钥是根CA公钥文件的第二部分。自签名根CA公钥是根CA按照《中国金融集成电路（IC）卡规范 》第七部分第12章规定的签名算法利用根CA签名私钥对根CA公钥证书数据（表8进行签名所得）。

表 8：根 CA 公钥证书数据

字段名	长度（字节数）	描述	格式
记录头	1	十六进制 ‘21’	b
服务标识	4	标识一个中国银联借记贷记服务，将相应应用的私有应用标识扩展(PIX)，右补十六进制‘0’构成。 ‘01010000’ = 借、贷记 ‘01010100’ = 借记 ‘01010200’ = 贷记 ‘01010300’ = 准贷记 ‘01010600’= 储值型电子现金	b

字段名	长度（字节数）	描述	格式
注册的应用提供商标识 (RID)	5	标识银联 RID：为十六进‘A000000333’	b
根 CA 公钥索引	1	唯一标识根 CA 公钥	b
证书失效日期	2	月和年(MMYY)，在该月最后一日之后证书失效	n 4
根 CA 公钥算法标识	1	十六进制 ‘01’ 标识生成根 CA 公钥的公钥算法，见《中国金融集成电路（IC）卡规范 》第七部分第 12 章。	b
根 CA 公钥模的左边部分	$N_{CA}-36-e_{CA}$	根 CA 公钥模的左边 $N_{CA}-36-e_{CA}$ 字节，这里 N_{CA} 和 e_{CA} 分别表示根 CA 公钥模长和公钥指数长度。	b
哈希算法标识	1	十六进制 ‘01’ ，标识生成哈希值的哈希算法，见《中国金融集成电路（IC）卡规范 》第七部分第 12 章。	b
根 CA 公钥指数长度	1	为十六进制字节数，表示根 CA 公钥指数长度 e_{CA} 。	b
根 CA 公钥指数	e_{CA}	根 CA 公钥指数。	b
哈希值	20	下列数据连接后计算的哈希值：RID、根 CA 公钥索引、根 CA 公钥模、根 CA 公钥指数。	b

表8根CA公钥证书数据中哈希值的计算见《中国金融集成电路（IC）卡规范 》第七部分第12章。

10.6 发卡行公钥输入文件

发卡行为获得发卡行生产型公钥证书或测试型公钥证书，需向IC卡根CA提交自签名的发卡行公钥证书申请文件，主要是发卡行公钥输入文件，见金融IC卡借记/贷记应用根CA公钥认证规范。

发卡行公钥输入文件的文件名格式为：“YLTTTTTT.INP”。其中“YL”为前缀；“TTTTTT”是记录号，唯一标识一个发卡行的一次公钥证书申请，由银联统一管理和分发，发卡行必须使用该记录号。

10.6.1 发卡行公钥输入文件结构

发卡行公钥输入文件是一个二进制文件，由两部分组成，如表9所示。

表 9：公钥输入文件

字段名	长度（字节数）
未签名发卡行公钥输入扩展	$7+N_I+e_I$
自签名发卡行公钥数据	N_I

10.6.2 未签名发卡行公钥输入扩展

未签名发卡行公钥输入扩展是发卡行公钥输入文件的第一部分。该输入扩展提供发卡行公钥信息，具体格式见表10。

表 10：未签名发卡行公钥输入扩展

字段名	长度（字节数）	描述	编码格式
记录头	1	十六进制值 ‘22’	b
发卡行公钥模长	1	发卡行公钥模长(N_I)的十六进制值(字节数)	b
发卡行公钥模	Var	未经签名的发卡行公钥模(N_I)	b
发卡行公钥指数长度	1	发卡行公钥指数长度（e_I）(字节数)	b
发卡行公钥指数	Var	发卡行公钥指数	b

发卡行公钥算法标识	1	标识用于发卡行公钥的数字签名算法，见《中国金融集成电路（IC）卡规范 》第七部分第 12 章	b
记录号	3	发卡行公钥证书申请记录号	n 6

10.6.3 自签名发卡行公钥数据

自签名发卡行公钥数据是发卡行公钥输入文件的第二部分，发卡行利用所申请发卡行公钥证书中的公钥对应的私钥对该发卡行公钥数据进行签名。根CA使用该发卡行公钥来验证签名的公钥数据。

发卡行使用发卡行私钥对发卡行公钥数据（表10中的所有数据）进行签名，形成自签名发卡行公钥数据。其中，发卡行私钥应与提交给中国银联的发卡行公钥相对应，签名算法参见《中国金融集成电路（IC）卡规范 》第七部分第12章的数字签名算法。自签名发卡行公钥数据的长度必须等于用来签名的发卡行公私钥对中公钥的长度。

表11中的哈希值是将表4中除了哈希值外所有数据依原有顺序连接后进行哈希计算得到的。该哈希值仅用于根CA对发卡行提交的公钥数据进行认证，与组成发卡行公钥证书的哈希值不是同一个值。

表 11：发卡行公钥数据

字段名	长度（字节）	描述	格式
记录头	1	十六进制‘23’	b
服务标识	4	标识一个中国银联借记贷记服务，将相应应用的私有应用标识扩展(PIX)，右补十六进制‘0’构成。 ‘01010000’ = 借、贷记 ‘01010100’ = 借记 ‘01010200’ = 贷记 ‘01010300’ = 准贷记 ‘01010600’= 储值型电子现金	b
证书格式	1	十六进制‘02’	b
发卡行标识	4	主帐号（PAN）最左面的 3-8 个数字。（不足部分右补十六进制数‘F’）	cn 8
证书失效日期	2	月和年(MMYY)，在该月最后一天之后证书失效	n 4
记录号	3	发卡行公钥证书申请记录号	n 6
哈希算法标识	1	标识用来产生哈希值的哈希算法，见《中国金融集成电路（IC）卡规范 》第七部分第 12 章	b
发卡行公钥算法标识	1	标识发卡行使用的数字签名算法，见《中国金融集成电路（IC）卡规范 》第七部分第 12 章	b
发卡行公钥模长	1	以十六进制表示的发卡行公钥模长(N_I)（字节数）	b
发卡行公钥指数长度	1	以十六进制表示的发卡行公钥指数长度（字节数）	b
发卡行公钥模的左边部分	$N_I - (39 + e_I)$	公钥模(N_I)的最左 $N_I - (39 + e_I)$部分，这里 N_I 表示发卡行公钥模长的字节数，e_I 表示发卡行公钥指数所占的字节数	b
发卡行公钥指数	e_I	发卡行公钥指数，为 3 或 65537，以十六进制存储为‘03’或‘01 00 01’	b
哈希值	20	发卡行公钥及其相关信息的哈希结果，为本表中除哈希值外从记录头依顺序到发卡行公钥指数的哈希值	b

成员发卡行生成自签名发卡行公钥数据的流程如图6所示。

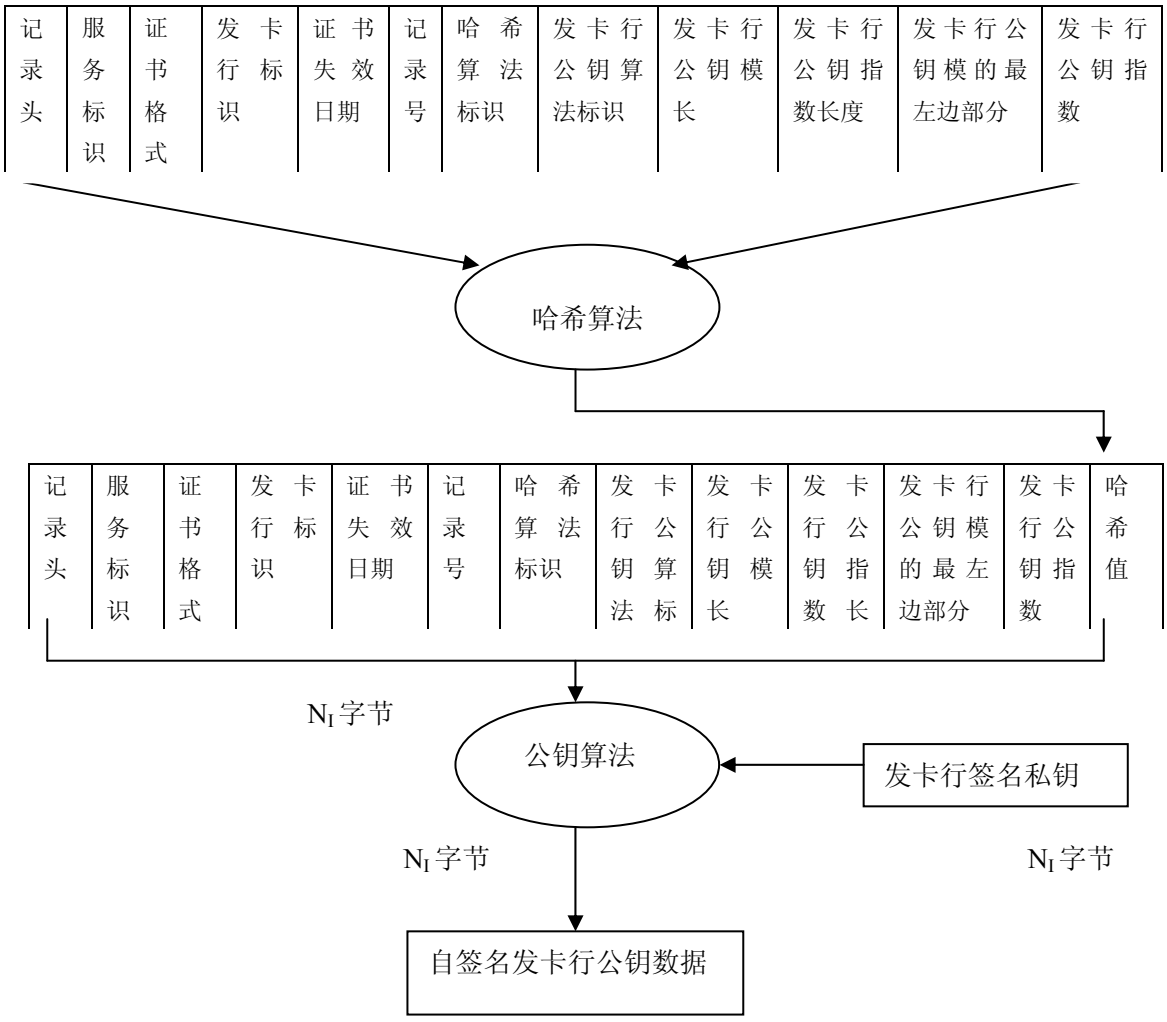


图 6：生成自签名发卡行公钥数据的流程

10.7 发卡行公钥输出文件

IC卡根CA为发卡行签发的发卡行证书的形式为发卡行公钥输出文件，见金融IC卡借记/贷记应用根CA公钥认证规范。发卡行公钥输出文件名格式为：AAAAAA.INN。其中：“AAAAAA”是申请记录号，与本标准第10.6节自签名发卡行公钥数据中的记录号相同；I是固定值(‘I’)表示发卡行；NN是用来签发发卡行公钥证书的根CA公钥的索引。

10.7.1 发卡行公钥证书输出文件结构

发卡行公钥证书输出文件的格式见表 12。

表 12：发卡行公钥证书输出文件

字段名	长度（字节数）
未签名发卡行公钥输出扩展	$17+e_I$ 若 $N_I \leq N_{CA}+36$

	$53+N_I-N_{CA}+e_I$ 若 $N_I > N_{CA}+36$ 这里，e_I、N_I、和 N_{CA} 分别表示发卡行公钥指数字节数、发卡行公钥模长字节数、和根 CA 用来生成该发卡行公钥证书的公钥模长字节数
签名的发卡行公钥证书	N_{CA}
根 CA 单独签名	N_{CA}

10.7.2 未签名发卡行公钥输出扩展

未签名发卡行公钥输出扩展是发卡行公钥输出文件的第一部分，其格式见表13。该公钥输出扩展提供了该发卡行公钥证书信息。

表 13：未签名发卡行公钥输出扩展

字段名	长度（字节数）	描述	格式
记录头	1	十六进制值‘24’	b
服务标识	4	标识一个中国银联借记贷记服务，将相应应用的私有应用标识扩展(PIX)，右补十六进制‘0’构成。 ‘01010000’ = 借、贷记 ‘01010100’ = 借记 ‘01010200’ = 贷记 ‘01010300’ = 准贷记 ‘01010600’= 储值型电子现金	b
发卡行标识	4	主帐号（PAN）最左面的 3-8 个数字。（不足部分右补十六进制数’F’）	cn 8
证书序列号	3	由根 CA 系统分配的唯一标识证书的二进制数	b
证书失效日期	2	月和年(MMYY)，在该月最后一天之后证书失效	n 4
发卡行公钥余项长度	1	发卡行公钥模余项的长度	b
发卡行公钥余项	0 和$(N_I - N_{CA}+36)$的最大值	该字段仅在 $N_I > N_{CA} - 36$ 时存在, 由发卡行公钥(N_I)的最低 $N_I- N_{CA} + 36$ 字节组成。	b
发卡行公钥指数长度	1	以十六进制表示的发卡行公钥指数的长度（字节数）	b
发卡行公钥指数	e_I	发卡行公钥指数，为 3 或 65537，存储为十六进制‘03’或 十六进制 ‘01 00 01’	b
根 CA 公钥索引	1	根 CA 系统用来签发发卡行公钥证书的公钥索引	b

10.7.3 签名的发卡行公钥证书

签名的发卡行公钥证书是发卡行公钥证书输出文件的第二部分，由根CA利用相应根CA私钥对表11中的发卡行公钥数据进行签名产生。

表 14：发卡行公钥证书数据

字段名	长度（字节数）	描述	格式
恢复数据头	1	十六进制值‘6A’	b
证书格式	1	十六进制值‘02’	b
发卡行标识	4	主帐号最左面的 3 到 8 位数字（从第 1 位开始）（不足部分右补十六进制‘F’）	cn 8
证书失效日期	2	月和年份(MMYY) ，在本月最后一天之后证书失效	n 4

证书序列号	3	由根 CA 系统分配的唯一标识证书的二进制数	b
哈希算法标识	1	标识用来产生数字签名哈希值的哈希算法	b
发卡行公钥算法标识	1	标识用于发卡行公钥的数字签名算法	b
发卡行公钥长度	1	以十六进制表示的发卡行公钥的模长(N_I)（字节数）	b
发卡行公钥指数长度	1	以十六进制表示的发卡行公钥指数的长度（字节数）	b
发卡行公钥模的左边部分	$N_{CA}-36$	如果 $N_I \leqslant N_{CA}-36$，该字段包含了完整的发卡行公钥模(N_I)，并在右面填充 $N_{CA}-36-N_I$ 个字节的‘BB’。 如果 $N_I > N_{CA}-36$，该字段包含了发卡行公钥模(N_I)的最高 $N_{CA}-36$ 字节	b
哈希值	20	发卡行公钥及其相关信息的哈希结果	n
恢复数据尾	1	十六进制值‘BC’	b

根CA通过未签名发卡行公钥输出扩展（表13）中的根CA公钥索引来定位根CA公钥及私钥，并通过该根CA私钥对发卡行公钥证书数据（表14）进行签名得到签名的发卡行公钥证书。

10.7.4 根 CA 单独签名

由于未签名发卡行公钥输出扩展没有经过根CA签名，为保证其文件的数据完整性、信息源可认证性以及与签名的发卡行公钥证书的有效绑定，在发卡行公钥证书输出文件中加入了根CA单独签名。根CA单独签名是发卡行公钥证书输出文件的第三部分，由根CA通过根CA私钥对根CA单独签名数据（表12）进行签名，其中，根CA私钥通过未签名发卡行公钥输出扩展（表13）中的根CA公钥索引来定位，使用的签名算法见《中国金融集成电路（IC）卡规范 》第七部分第12章。

发卡行对根CA单独签名的验证本标准不做强制性要求。

表 15：根 CA 单独签名数据

文件	长度（字节数）	描述	格式
记录头	1	十六进制 ‘00’	b
分组格式编码	1	十六进制 ‘01’	b
右边添加字符	N_{CA}-24	N_{CA}-24 字节十六进制 ‘FF’	b
分隔符	1	十六进制 ‘00’	b
算法标识	1	根 CA 使用的哈希算法标识，为十六进制‘01’	b
哈希值	20	对未签名发卡行公钥输出扩展和发卡行公钥证书数据进行连接后的数据计算哈希值	b

10.8 卡片证书及签名静态数据生成

10.8.1 发卡行签发的卡片证书

发卡行在发行支持DDA/CDA的卡片过程中需要签发卡片公钥证书。发卡行CA签发的卡片公钥证书以卡片公钥文件形式组成。

卡片公钥文件名格式为：AAAAAAAAAA.ICNNNNNN。其中：“AAAAAAAAAA”是卡片主帐号（PAN）；IC是固定值(‘IC’)表示卡片；NNNNNN是用来签发卡片公钥证书的发卡行CA公钥证书序列号。

10.8.1.1 卡片公钥文件内容

卡片公钥文件由未签名的卡片公钥输出扩展和签名的卡片证书组成。

表 16：卡片公钥文件

字段名	长度（字节数）
未签名卡片公钥输出扩展	$18+e_{IC}$ 若 $N_{IC} \leqslant N_I+42$ $60+N_{IC}-N_I+e_{IC}$ 若 $N_{IC} > N_I+42$

	这里，e_{IC}、N_{IC}、和 N_I 分别表示卡片公钥指数字节数、卡片公钥模长字节数、和发卡行 CA 用来生成该卡片公钥证书的公钥模长字节数
签名的卡片公钥证书	N_I

10.8.1.2 未签名卡片公钥输出扩展

未签名卡片公钥输出扩展是卡片公钥文件的第一部分，其格式见表17。该公钥输出扩展提供了该卡片公钥证书信息。

表 17：未签名卡片公钥输出扩展

字段名	长度（字节数）	描述	格式
记录头	1	十六进制值‘26’	b
卡片主帐号（PAN）	10	主帐号（在右边补上十六进制数‘F’）	cn 20
证书序列号	3	由发卡行 CA 系统分配的唯一标识卡片证书的二进制数	b
证书失效日期	2	月和年(MMYY)，在该月最后一天之后证书失效	n 4
卡片公钥余项长度	1	卡片公钥模余项的长度	b
卡片公钥余项	0 和($N_{IC} - N_I$+ 42)的最大值	该字段仅在 $N_{IC} > N_I - 42$ 时存在, 由卡片公钥(N_{IC})的最低 N_{IC}-N_I + 42 字节组成。	b
卡片公钥指数长度	1	以十六进制表示的卡片公钥指数的长度（字节数 e_{IC}）	b
卡片公钥指数	e_{IC}	卡片公钥指数，为 3 或 65537，存储为十六进制‘03’ 或 十六进制 ‘01 00 01’	b

10.8.1.3 签名的卡片公钥证书

签名的卡片公钥证书是卡片公钥文件的第二部分，由发卡行CA利用相应发卡行CA私钥对表18中的卡片公钥数据进行签名产生。

表 18：卡片公钥证书数据

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 $N_{IC} \le N_I-42$，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

注：哈希结果是将表18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）卡规范借记贷记卡片规范》指明的需认证的静态数据，得到的结果计算哈希值。

10.8.1.4 卡片公钥以及相关信息的哈希值计算

在表18的哈希结果是将表18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）卡规范借记贷记卡片规范》指明的需认证的静态数据，将这些数据连接得到的结果计算哈希值。其中《中国金融集成电路（IC）卡规范借记贷记卡片规范》指明的需认证的静态数据见表19：

表 19：待签名的静态应用数据

字段名	长度(字节数)	描述	格式
应用交互特征（AIP）	2	说明此应用中卡片支持的功能。	b
应用生效日期	3	卡片中应用启用日期。	n
应用失效日期	3	卡片应用失效日期。	n
应用主账号	10	PAN，不足位右补“F”。	cn
应用主账号序列号	1	用来表示卡片中使用同一个账号的不同应用。	n
应用用途控制 AUC	2	标明发卡行指定的卡片应用上的一些限制，包括地域使用和服务类型等。用于提供更灵活的卡片服务控制（类似服务代码）。	b
持卡人验证方法（CVM）列表	12	按照优先顺序列出卡片应用支持的所有持卡人验证方法。注意：一个应用中可以有多个 CVM 列表，例如一个用于国内交易，一个用于国际交易。	b
发卡行行为代码（IAC）——缺省	5	指定当交易请求联机但是终端不能完成联机上送的交易拒绝的条件。	b
发卡行行为代码（IAC）——拒绝	5	指定交易不进行联机直接拒绝的条件。	b
发卡行行为代码（IAC）——联机	5	指定交易联机上送的条件。	b
发卡行国家代码	2	指明卡片发行者的国家。	b

注1：如果应用中签名的数据不是唯一，卡片必须支持多个SAD。举例来说，卡片给国内和国际交易分别设置CVM列表，而CVM列表是签名数据。

注2：如果发行后的卡片有修改签名数据的能力，则卡片必须支持修改SAD的能力。

注3：如果AIP也要签名，SDA标识列表包括AIP的标签，如果支持DDA则建议将AIP做签名。除了AIP不能有其它数据标签。

10.8.2 发卡行 CA 签发卡片静态数据签名

根据《中国金融集成电路（IC）卡规范借记贷记规范》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。
- RID
- 发卡行公钥证书。
- 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。
- 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。
- 发卡行公钥指数：一个由发卡行提供的变长数据元素。

发卡行CA需要生成上述数据并将上述数据传给数据准备系统。

《中国金融集成电路(IC)卡规范借记贷记卡片规范》指明的静态数据认证用到的数据对象见表20，静态数据认证中使用的卡片数据见表21。

表 20：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）卡规范借记/贷记卡片规范》9.3.1 节指明的需认证的静态数据（见表 21）	–

表 21： 静态数据认证中使用的卡片数据

数据元	描述
CA 公钥索引（PKI）	和发卡行公钥证书一起由 CA 提供。定义了终端里用于认证发卡行公钥证书的 CA 公钥
发卡行公钥证书	证书中包括了使用 CA 私钥签名的发卡行公钥
发卡行公钥指数	用于 RSA 算法中恢复发卡行公钥证书。值为 3 或 65537
发卡行公钥余项	发卡行公钥没有包含在发卡行公钥证书中的部分（如果有）
AID 中的注册应用标识部分（RID）	和 CA 公钥索引一起用来标识终端中的公钥
签名的静态应用数据（SAD）	一个用来验证卡片静态数据的签名。在卡片个人化阶段，使用发卡行私钥签名的 SAD 保存在卡片中。推荐下列数据用来生成签名： 应用交互特征 AIP（如果支持 DDA） 应用生效日期 应用失效日期 应用主账号 应用主账号序列号 应用用途控制 AUC 持卡人验证方法（CVM）列表 发卡行行为代码——缺省 发卡行行为代码——拒绝 发卡行行为代码——联机 发卡行国家代码（“5F28”） 如果应用中签名的数据不是唯一，卡片必须支持多个 SAD。举例来说，卡片给国内和国际交易分别设置 CVM 列表，而 CVM 列表是签名数据。 如果发行后的卡片有修改签名数据的能力，则卡片必须支持修改 SAD 的能力。
SDA 标签列表	如果 AIP 也要签名，SDA 标识列表包括 AIP 的标签，如果支持 DDA 则建议将 AIP 做签名。除了 AIP 不能有其它数据标签。

发卡行CA签发的卡片静态数据是按照下面表22数据进行签名操作。

表 22： 待签名的静态应用数据格式

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‘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	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

10.8.2.1 静态签名数据组成

发卡行CA签发静态数据是对表22数据进行签名，而表22中的哈希结果是将表19（由表21数据产生）数据依次连接得到的数据进行哈希值计算，对表18数据的签名结果即为签名的卡片静态数据。

10.9 发卡行卡片个人化过程使用的密钥及密码算法

10.9.1 卡片个人化过程的安全规则

卡片个人化过程包括下列数据准备和数据个人化过程：

- 用于个人化PBOC2.0借贷记应用卡片的IC卡数据生成
- 将上述数据传递到卡片个人化系统
- 将上述数据加载到卡片中

在整个过程中应确保安全、敏感数据不被非授权泄漏和更改。

发卡行在卡片个人化过程中应确保下列数据的完整性：

- 磁道二数据
- 持卡人姓名
- 主帐号PAN
- 主帐号序列号
- 卡片失效期
- 相关公钥证书
- 卡片风险管理数据、计数器限制、和其它卡片数据

发卡行应在卡片个人化过程中确保下列数据的机密性和完整性：

- 卡片的三个3DES主密钥
- 卡片的基于RSA的密钥和签名数据
- 持卡人PIN

个人化数据应由报文鉴别码MAC或数字签名保护，该保护从这些数据离开数据生成系统直到存储于IC卡存储器内的整个过程中有效。个人化指令或加载指令在传递到卡片时应能够处理MAC或签名，以及可能的加密数据。

持卡人PIN需要在整个个人化过程中加密保护。

敏感个人化数据、密钥在传递过程中应加密保护，其明文应仅在硬件安全模块或IC卡内出现。这些数据应在硬件安全模块内产生，并且仅以密文方式从硬件安全模块内输出。同样，用于计算MAC、计算签名、加密敏感数据的传输密钥应以密文出现，仅仅在硬件安全模块和IC卡内可以以明文出现。应在安全条件下进行硬件安全模块和IC卡的密钥加载。

应在同一个硬件安全模块内生成发卡行公私钥对、生成IC卡公私钥对、相应的公钥证书和最终存储于IC卡的3DES密钥。在这些密钥及敏感数据仅在加密后才能由硬件安全模块输出。

10.9.2 卡片个人化过程使用的密钥

发卡行在卡片个人化过程中应生成所有卡片数据，包括表22所列的所有需要写入卡片的密钥数据，并将上述数据安全写入卡片。发卡行的卡片个人化过程通常涉及数据准备系统和个人化系统，这些系统

可能位于一个发卡行内也可能位于相关厂商，需要确保在个人化过程中始终确保所有卡片数据的数据完整性，对机密卡片数据如PIN和密钥需要始终维持数据机密性。

发卡行在数据准备过程中应安全生成表22所列的所有需要写入卡片的密钥数据和其它PBOC2.0规范规定的卡片数据，并将这些数据安全地传递给个人化系统，由个人化系统将这些卡片数据安全写入卡片。卡片个人化过程涉及多个密钥，因此在个人化之前应生成相关的个人化密钥。用于发卡行和卡片间数据保护的个人化密钥包括：

- 发卡行主密钥（KMC）：发卡行主密钥由发卡行生成并与数据准备系统和个人化系统共享，发卡行主密钥用于导出卡片或卡片应用的密钥K_{ENC}、K_{MAC}和K_{DEK}。
- 卡片或卡片应用密文密钥K_{ENC}：用来生成卡片密文或验证主机密文。
- 卡片或卡片应用MAC密钥K_{MAC}：用来锁闭中国金融集成电路（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。
- 卡片或卡片应用加密密钥K_{DEK}：用来加密在个人化过程中写入卡片的保密数据。

上述密钥仅用于卡片个人化过程中的数据保护。

卡片加密分散密钥K_{ENC}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3DES密钥。

K_{ENC}密钥用以下方法推算：

K_{ENC} ：= 3DES（KMC）[KEYDATA的6个最低有效字节|| ‘F0’ || ‘01’]||3DES（KMC）[KEYDATA的6个最低有效字节 || ‘0F’ || ‘01’]。

卡片校验码分散密钥K_{MAC}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3DES密钥。

K_{MAC}应采用以下方法导出：

K_{MAC} ：= 3DES（KMC）[KEYDATA的6个最低有效字节|| ‘F0’ || ‘02’]||3DES（KMC）[KEYDATA的6个最低有效字节 || ‘0F’ || ‘02’]。

卡片密钥加密分散密钥K_{DEK}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3DES密钥。

K_{DEK}应采用以下方法导出：

K_{DEK} ：=3DES（KMC）[KEYDATA 的6个最低有效字节 || ‘F0’ || ‘03’]|| 3DES（KMC）[KEYDATA的6个最低有效字节|| ‘0F’ || ‘03’]。

此外在发卡行、数据准备系统和个人化系统间在整个卡片个人化过程中还需要保护系统间传输的个人化数据，整个发卡行个人化系统还包括下列个人化密钥：

- 密钥交换密钥（KEK）——为3DES密钥，用来对发卡行个人化输入文件中的机密数据进行加密，并在发卡行和数据准备系统间对这些数据进行安全保护，由发卡行与数据准备系统共享。每个发卡行的KEK必须是唯一的。
- 传输密钥（DEK/TK）——为3DES密钥，用来对数据准备系统向个人化系统传送的发卡行个人化输入文件中的机密数据进行加密，由数据准备系统和个人化系统共享。
- MAC密钥－－为3DES密钥，由数据准备系统在向个人化系统传递个人化文件时用于保护个人化数据的完整性。

这些个人化密钥均为16字节的3DES密钥，具体的使用及相关密码运算模式和算法可参见PBOC2.0规范的个人化指南及EMV通用个人化规范。

发卡行的个人化系统属于机构内部系统，其设置和部署可以有所不同，可能需要额外的密钥对位于不同区域的数据进行保护。

表 23：个人化密钥汇总

名称	标识	类型	共享	用途	生成
发卡行主密钥	KMC	3DES	发卡行、卡商、个人化设备	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。	发卡行
卡片数据加密密钥	K_{ENC}	3DES	发卡行、卡片应用、卡商、个人化设备	用来生成一个过程密钥，利用该过程密钥可生成或解密密文和以 CBC 模式加密机密数据。	发卡行
卡片 MAC 密钥	K_{MAC}	3DES	发卡行、卡片应用、卡商、个人化设备	用来生成一个过程密钥，利用该过程密钥可生成或验证命令处理过程中所使用的 C-MAC。	发卡行
卡片数据加密密钥	K_{DEC}	3DES	发卡行、卡片应用、卡商、个人化设备	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。	发卡行
发卡行密钥交换密钥	KEK_{ISS}	3DES	发卡行、数据准备系统	对发卡行与数据准备设备之间的脱机 PIN 及其它机密数据进行保护。	发卡行
数据加密密钥	DEK	3DES	数据准备系统、个人化系统	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。	发卡行
传输密钥	TK	3DES	数据准备系统、个人化系统		发卡行
MAC 密钥	MACkey	MAC	每个个人化设备	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。	发卡行

11 PBOC2.0 借贷记应用交易密码运算

本章提供在PBOC2.0借贷记应用实现的各种密码操作的实例。包括：

1. 联机授权的卡片交易：
   - 卡片密钥导出
   - PBOC2.0联机交易过程密钥的导出
   - ARQC计算
   - ARPC计算
   - TC计算
2. PIN变更：
   - 安全报文加密卡片密钥的导出
   - 安全报文加密过程密钥的导出
   - PIN锁定加密
   - 用于脚本MAC的卡片密钥导出
   - 用于脚本MAC的过程密钥的导出
   - 脚本MAC的计算
3. 静态数据认证：
   - 签名静态应用数据的验证

4. 动态数据认证:
   - IC卡公钥证书的验证
   - 为DDA生成签名的动态数据
   - IC卡公钥证书的验证
   - DDA的签名动态数据的验证
5. 复合DDA/AC（CDA）生成:
   - 脱机计数器值的加密
   - CDA的签名动态数据生成
6. CDA验证

本章中3DES(K)[D]表示用3DES算法以密钥K对数据块D加密。CBC_3DES(K)[D]表示用3DEA以CBC模式以密钥K对数据D加密。MAC(K)[M]对报文M用密钥K采用ISO/IEC 9797-1算法3并以DES作为分组密钥的MAC算法计算的8字节MAC值。

11.1 联机授权卡片交易

本节叙述采用联机授权的卡片交易的密码运算，包括：

- 卡片主密钥导出
- 过程密钥导出
- ARQC计算
- ARPC计算
- TC计算

11.1.1 卡片主密钥导出

卡片AC主密钥MK_{AC}由发卡行AC主密钥IMK_{AC}对卡片PAN与PAN序列号的链接加密得出，见本标准第9.1.1节。

这里发卡行AC主密钥值IMK_{AC}为：‘F4 31 62 3E 49 97 89 C8 C2 F1 F7 25 AD 7F 89 49’

输入为PAN和PAN序列号的链接：

（PAN的最右边7字节(以n编码格式，若不足7字节则左补0以得到7个字节)。）

当PAN和PAN序列号的链接为8字节（16个数字）长，则以数字格式的该数据链接Y作为输入；若PAN和PAN序列号的链接小于16个数字，则右对齐，前面补16进制的0以获得8字节数字格式的Y；如果PAN和PAN序列号的链接的长度至少有16个数字，则以Y＝该链接的最右边16个数字。

对于PAN的值为：‘62 25 00 00 00 00 00 89’

PAN序列号（1字节n编码）。

对于PAN序列号（1字节n编码）1：‘01’

输入为（取PAN和PAN序列号的链接值的最右边16个数字作为输入）：

‘25 00 00 00 00 00 89 01’

输出按照MK_{AC}＝3DES(IMK_{AC})[输入]||3DES(IMK_{AC})[输入 ⊕ ‘FF FF FF FF FF FF FF FF’]＝3DES(IMK_{AC})[‘25 00 00 00 00 00 89 01’]||3DES(IMK_{AC})[‘DA FF FF FF FF FF 76 FE’]＝

‘89 E8 1E CA 27 6C 7D 50 A5 37 AE 51 2E 11 52 24’

完成奇偶校验后的值为：‘89 E9 1F CB 26 6D 7C 51 A4 37 AE 51 2F 10 52 25’

所以，MK_{AC}＝ ‘89 E9 1F CB 26 6D 7C 51 A4 37 AE 51 2F 10 52 25’。

11.1.2 卡片 AC 过程密钥导出

卡片AC过程密钥的导出见本标准第9.2节。

这里采用由上节计算出的卡片主密钥MK_{AC}值作为密钥：

‘89 E9 1F CB 26 6D 7C 51 A4 37 AE 51 2F 10 52 25’

输入为ATC和6个0字节的链接：

1. ATC值：‘00 02’

2. 6个0字节：‘00 00 00 00 00 00’
3. 链接：‘00 02 00 00 00 00 00 00’

输出为：

1. ZL := 3DES(Key)[‘00’ || ‘00’ || ‘00’ || ‘00’ || ‘00’ || ‘00’ || ATC]
2. ZR := 3DES(Key)[‘00’ || ‘00’ || ‘00’ || ‘00’ || ‘00’ || ‘00’ || (ATC ⊕ ‘FFFF’)]
3. 计算结果：ZL＝‘4F A1 BB 65 DC 84 58 BF’，ZR＝‘ED 43 C2 10 F4 FD 47 11’
4. ZL|| ZR＝ ‘4F A1 BB 65 DC 84 58 BF ED 43 C2 10 F4 FD 47 11’

完成奇偶校验后的值为：

‘4F A1 BA 64 DC 85 58 BF EC 43 C2 10 F4 FD 46 10’

所以卡片AC过程密钥SK_{AC}为：‘4F A1 BA 64 DC 85 58 BF EC 43 C2 10 F4 FD 46 10’。

11.1.3 ARQC 生成

ARQC的生成见本标准第10.2.2节。

这里用于ARQC生成的过程密钥SK_{AC}如上节为：

‘4F A1 BA 64 DC 85 58 BF EC 43 C2 10 F4 FD 46 10’

输入为下列数据的链接：

1. 授权金额：‘00 00 00 01 68 00’
2. 其它金额： ‘00 00 00 00 00 00’
3. 终端国家代码： ‘01 56’
4. 终端验证结果：‘00 00 00 80 00’
5. 交易货币代码：‘01 56’
6. 交易日期：‘07 11 09’
7. 交易类型：‘01’
8. 不可预知数：‘DC CD 7C 16’
9. 应用交互特征（AIP）：‘7D 00’
10. 应用交易序号（ATC）：‘00 02’
11. 卡片验证结果（CVR）：‘03 A0 00 02’

链接值作为输入（若不是8字节的整数倍则右补‘80’和最少个数的‘00’构成8字节的整数倍）：

‘00 00 00 01 68 00 00 00 00 00 00 00 01 56 00 00 00 80 00 01 56 07 11 09 01 DC CD 7C 16 7D 00 00 02 03 A0 00 02 80 00 00’

输出：

ARQC＝MAC(SK_{AC})[输入]：

‘8B 6F 6D 8F 49 64 A6 09’

11.1.4 ARPC 生成

ARQC的生成见本标准第10.2.2节。

这里过程密钥SK_{AC}与上节计算ARQC的过程密钥相同，并且过程密钥导出方式相同，其值为：

‘4F A1 BA 64 DC 85 58 BF EC 43 C2 10 F4 FD 46 10’

输入为下列数据的处理：

1. AC（ARQC）： ‘8B 6F 6D 8F 49 64 A6 09’
2. 授权响应码（ARC）：‘30 30’
3. 补足的授权响应码：授权响应码右补6个‘00’构成8个字节：‘30 30 00 00 00 00 00 00’
4. 将ARQC和补足的授权响应码异或构成输入

输出为ARPC=3DES(SK_{AC})[输入]＝3DES(SK_{AC})[AC ⊕ (ARC || ‘00’ || ‘00’ || ‘00’ || ‘00’ || ‘00’ || ‘00’)]＝：

‘09 12 44 22 2A 16 3D FA’

11.1.5 TC 生成

TC的生成见本标准第10.2.2节。

这里过程密钥SK_{AC}与上节相同为：

‘4F A1 BA 64 DC 85 58 BF EC 43 C2 10 F4 FD 46 10’

输入为下列数据的链接：

1. 授权金额： ‘00 00 00 01 68 00’
2. 其它金额： ‘00 00 00 00 00 00’
3. 终端国家代码： ‘01 56’
4. 终端验证结果： ‘00 00 00 80 00’
5. 交易货币代码： ‘01 56’
6. 交易日期： ‘07 11 09’
7. 交易类型： ‘01’
8. 不可预知数： ‘DC CD 7C 16’
9. 应用交互特征（AIP）： ‘7D 00’
10. 应用交易序号（ATC）： ‘00 02’
11. 卡片验证结果（CVR）： ‘03 60 00 02’

链接值作为输入（若不是8字节的整数倍则右补‘80’和最少个数的‘00’构成8字节的整数倍）：

‘00 00 00 01 68 00 00 00 00 00 00 00 01 56 00 00 00 80 00 01 56 07 11 09 01 DC CD 7C 16 7D 00 00 02 03 60 00 02’

输出：ARPC＝MAC(SK_{AC})[输入]＝

‘E4 65 D8 7C DD 8E 5A 48’

11.2 PIN 变更

本节叙述在安全报文中发卡行完成PIN变更所执行的密码运算，包括：

- 卡片安全报文加密主密钥导出
- 安全报文加密过程密钥导出
- PIN锁定加密
- 用于脚本MAC计算的卡片主密钥导出
- 脚本MAC过程密钥导出
- 脚本MAC计算

11.2.1 脚本加密卡片主密钥的导出

脚本加密卡片密钥的导出见本标准第10.1节。

密钥：发卡行安全报文加密主密钥IMK_{SMC}值为：

‘76 5E EC AB B0 A8 61 08 F1 CB 15 91 73 B9 7F 26’

输入为下列数据（PAN和PAN序列号）的链接：

当PAN和PAN序列号的链接为8字节（16个数字）长，则以数字格式的该数据链接Y作为输入；若PAN和PAN序列号的链接小于16个数字，则右对齐，前面补16进制的0以获得8字节数字格式的Y；如果PAN和PAN序列号的链接的长度至少右16个数字，则以Y＝该链接的最右边16个数字。

对于PAN 6225000000000089： ‘62 25 00 00 00 00 00 89’

PAN序列号（n编码，1字节）

对于PAN序列号（n编码，1字节）1： ‘01’

输入为： ‘25 00 00 00 00 00 89 01’

输出按照MK_{SMC}＝3DES(IMK_{SMC})[输入]||3DES(IMK_{SMC})[输入⊕ ‘FF FF FF FF FF FF FF FF’]＝3DES(IMK_{SMC})[‘25 00 00 00 00 00 89 01’] ||3DES(IMK_{SMC})[‘DA FF FF FF FF FF 76 FE’]＝：

‘26 CC 64 66 C9 E2 0D 5A 04 5A 18 51 68 04 F9 1F’

完成奇偶校验后的值为：

‘26 CD 64 67 C8 E3 0D 5B 04 5B 19 51 68 04 F8 1F’

脚本加密卡片密钥MK_{SMC}为：

‘26 CD 64 67 C8 E3 0D 5B 04 5B 19 51 68 04 F8 1F’

11.2.2 脚本加密过程密钥的导出

脚本加密过程密钥的导出见本标准第10.3.1节。

这里用于安全报文加密计算的卡片主密钥MK_{SMC}的导出见上节，为

‘26 CD 64 67 C8 E3 0D 5B 04 5B 19 51 68 04 F8 1F’

输入：

1. ATC：‘00 02’
2. 过程密钥A生成的输入：‘00 00 00 00 00 00’ || ATC
3. 过程密钥B生成的输入：‘00 00 00 00 00 00’ ||(ATC ⊕ ‘FFFF’)

输出：

过程密钥A:＝ Z_L＝3DES(Key)[‘00’ || ‘00’ || ‘00’ || ‘00’ || ‘00 || ‘00’ ||ATC]＝

‘3D BC 12 FE 5F D5 09 0B’

过程密钥B :＝Z_R＝3DES(Key)[‘00’ || ‘00’ || ‘00’ || ‘00’ || ‘00’ || ‘00’ || (ATC ⊕ ‘FFFF’)]＝

‘BF 05 36 C2 83 76 A4 DF’

过程密钥A与过程密钥B的链接为：

‘3D BC 12 FE 5F D5 09 0B BF 05 36 C2 83 76 A4 DF’

完成奇偶校验后的值为：

‘3D BC 13 FE 5E D5 08 0B BF 04 37 C2 83 76 A4 DF’

因此脚本加密过程密钥SK_{SMC}为：

‘3D BC 13 FE 5E D5 08 0B BF 04 37 C2 83 76 A4 DF’

11.2.3 脚本的加密

安全报文加密见本标准第10.3.2节。

安全报文加密的过程密钥SK_{SMC}如上节：

‘3D BC 13 FE 5E D5 08 0B BF 04 37 C2 83 76 A4 DF’

输入：

1. 需加密的数据为：‘04 12 34 FF 37 1C F2 A4’
2. 按照加密格式准备好的数据原文为输入数据：‘08 04 12 34 FF 37 1C F2 A4’

输出：

密文计算用过程加密密钥SK_{SMC}以ECB模式的3DES对输入数据进行加密运算。结果为：

‘64 52 95 FC F6 6D 07 5B 21 48 F1 21 79 0B E0 CC’

11.2.4 脚本 MAC 卡片主密钥导出

安全报文MAC计算见本标准10.3.2节。

这里安全报文完整性发卡行主密钥为IMK_{SMI}。假设IMK_{SMI}的值为：

‘AB AB F7 C8 FB 7C 40 3D 8F F7 3E A8 D0 F1 CE 38’

输入为下列数据（PAN和PAN序列号）的链接：

当PAN和PAN序列号的链接为8字节（16个数字）长，则以数字格式的该数据链接Y作为输入；若PAN和PAN序列号的链接小于16个数字，则右对齐，前面补16进制的0以获得8字节数字格式的Y；如果PAN和PAN序列号的链接的长度至少右16个数字，则以Y＝该链接的最右边16个数字。

对于PAN 6225000000000089：‘62 25 00 00 00 00 00 89’

PAN序列号（n编码，1字节）

对于PAN序列号（n编码，1字节）1：　‘01’

输入为：　‘25 00 00 00 00 00 89 01’

输出按照MK_{SMI}:＝3DES(IMK_{SMI})[输入]||3DES(IMK_{SMI})[输入⊕ ‘FF FF FF FF FF FF FF FF’]＝3DES(IMK_{SMI})[‘25 00 00 00 00 00 89 01’] ||3DES(IMK_{SMI})[‘DA FF FF FF FF FF 76 FE’]＝:

‘C5 B4 C9 C2 46 A0 A8 C4 49 77 57 0C 13 02 CB 89’

完成奇偶校验后的值为：

‘C4 B5 C8 C2 46 A1 A8 C4 49 76 57 0D 13 02 CB 89’

因此，卡片脚本MAC主密钥MK_{SMI}为：

‘C4 B5 C8 C2 46 A1 A8 C4 49 76 57 0D 13 02 CB 89’

11.2.5　脚本 MAC 过程密钥

安全报文（脚本）MAC过程密钥的导出见本标准第10.3.1节。

这里安全报文完整性（脚本MAC）卡片主密钥MK_{SMI}如上节为：

‘C4 B5 C8 C2 46 A1 A8 C4 49 76 57 0D 13 02 CB 89’

输入：

1. ATC：　‘00 02’
2. 过程密钥A生成的输入：‘00 00 00 00 00 00’|| ATC
3. 过程密钥B生成的输入：‘ 00 00 00 00 00 00’||(ATC⊕ ‘FFFF’)

输出：

过程密钥A:= Z_L=3DES(MK_{SMI})[‘00’ || ‘00’ || ‘00’ || ‘00’ || ‘00’ || ‘00’ || ATC]=

‘3D BC 12 FE 5F D5 09 0B’

过程密钥B :=Z_R= 3DES(MK_{SMI})[‘00’ || ‘00’ || ‘00’ || ‘00’ || ‘00’ || ‘00’ || (ATC⊕ ‘FFFF’)]=

‘BF 05 36 C2 83 76 A4 DF’

过程密钥A与过程密钥B的链接为：

‘3D BC 12 FE 5F D5 09 0B BF 05 36 C2 83 76 A4 DF’

完成奇偶校验后的值为：

‘3D BC 13 FE 5E D5 08 0B BF 04 37 C2 83 76 A4 DF’

因此脚本MAC过程密钥SK_{SMI}为：

‘3D BC 13 FE 5E D5 08 0B BF 04 37 C2 83 76 A4 DF’

11.2.6　PIN 变更脚本 MAC 的计算

安全报文(脚本)MAC计算见本标准第10.3.2节。

这里安全报文完整性过程密钥SK_{SMI}见上节，为：

‘3D BC 13 FE 5E D5 08 0B BF 04 37 C2 83 76 A4 DF’

输入：

计算MAC的输入数据= 命令头 || ATC || AC || 数据域(可选) || 80...

MAC输入数据如下：

‘84 24 00 02 14 00 02 8B 6F 6D 8F 49 64 A6 09 64 52 95 FC F6 6D 07 5B 21 48 F1 21 79 0B E0 CC 80’

使用MAC过程密钥按MAC算法计算出的结果是：

‘E2 4D 5F F1 45 CD 65 B5’

取前4字节做为MAC值：　‘E2 4D 5F F1’

最终的脚本命令为：

‘84 24 00 02 14 64 52 95 FC F6 6D 07 5B 21 48 F1 21 79 0B E0 CC E2 4D 5F F1’

11.3 静态数据认证

本节叙述进行卡片静态数据认证过程中终端执行的密码运算。

终端验证签名的静态应用数据的步骤包括：

1. 使用IC卡根CA公钥验证发卡行公钥证书
2. 使用从上一步获取的发卡行公钥验证签名的静态应用数据
3. 获取数据认证码

本节叙述验证签名静态应用数据的步骤二的运算。

11.3.1 签名静态应用数据的验证

11.3.1.1 发卡行公钥

发卡行公钥可以直接从发卡行公钥证书及发卡行公钥余项中提取。发卡行公钥证书格式和内容见金融IC卡借记/贷记应用根CA公钥认证规范的技术规范和本标准第10.7节。

发卡行公钥模：

C2 AB E7 63 CD 75 D5 7D DC D3 4C F6 32 AA 27 F5 E9 5A 52 04 56 2C 2D 39 E9 46 07 74
C7 61 B8 65 73 E9 D4 C1 B5 AC 4D AD A9 F4 2F 92 17 71 2B 73 D5 A6 6E 29 EA 8E 02 74
08 5F F6 33 CB 8E BB FA FB 13 F8 BC 82 63 84 E1 52 2F AB 4F C4 54 58 18 CB 6F 41 65
85 84 5E 7E 64 B7 21 A3 4B E4 8F AE F0 B0 78 DC BA DE BE 5F FA 22 A7 47 FF AB C8 EC
F6 2F E4 B0 96 94 9F AE 88 A3 31 79 28 73 16 3B EC D9 0D 75 D8 F1 57 0F 47 ED 40 F7
86 90 B7 FB

发卡行公钥指数：‘03’

签名的静态数据(Tag ’93’)如下：

A9 F0 A9 59 A1 FC FC 7D 49 E0 40 8D BF D5 5B 8D A8 E0 E1 94 AB C8 79 E6 21 52 8F 70
29 43 4D 70 FB B8 06 B6 65 9E 77 EE C2 0C 52 71 EE 6C EB 96 9D 05 47 F2 AA 70 80 10
AD DF 6A 0E 59 AD 60 1A 3A 2F 26 3D 0F F5 92 54 80 52 CF 37 2D 94 58 CC 9C A8 D7 54
1F 93 A0 A6 E7 95 BB EB 25 9B 1C F3 20 94 2B 10 59 93 65 50 1F EC 5E 3A 2E D9 99 C8
1D FE E4 3E 97 85 34 FC 82 85 EC 5C A4 32 FD E4 F5 23 49 BC C5 54 FE CC F6 5E AC 66
C7 B4 7D 72

执行以下验证过程：

1. 检查签名的静态应用数据长度与发卡行公钥模的长度是否相同(位/字节)： 比较结果：相同（1152/144）
2. 使用RSA算法恢复出签名数据的原文
   - 发卡行公钥指数=03,则：X:=(签名的静态应用数据)^{03} mod (发卡行公钥模值)，其中：X为恢复出的数据。
   - 从签名的静态应用数据恢复的原数据如下：

6A 03 01 DA C1 BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB
BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB
BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB
BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB
BB BB BB BB BB BB BB BB BB BB BB 4B 47 C5 8D 86 67 C3 10 44 40 AD DC F4 7C D9 8D 89
37 B2 CB BC

3. 检查恢复数据的格式是否正确：
   - 数据的头(第一个字节)：‘6A’，检查结果：通过。
   - 格式(第二个字节)：‘03’，检查结果：通过。
   - 数据结尾(最后一个字节)：‘BC’，检查结果：通过。

4. 链接用于计算哈希的数据，恢复数据中需要参与哈希计算的部分(第2至第5个数据源，即除头、尾和哈希结果外的所有数据)：

03 01 DA C1 BB

AFL中指明需要被认证的静态应用数据需要参与哈希计算，数据如下(顺序不可变)：

应用生效日期：‘5F 25 03 07 01 01’

应用失效日期：‘5F 24 03 12 12 31’

主账号(PAN)：‘5A 08 62 25 00 00 00 00 00 89’

主账号序列号：‘5F 34 01 01’

IAC一默认：‘9F 0D 05 FC F0 E4 08 00’

IAC一拒绝：‘9F 0E 05 00 10 00 00 00’

IAC一联机：‘9F 0F 05 FC F8 E4 F8 80’

应用使用控制：‘9F 07 02 FF C0’

持卡人验证方法列表：‘8E 12 00 00 00 00 00 00 00 00 5E 03 41 03 42 03 60 03 1F 00’

SDA标签清单：‘9F 4A 01 82’

AIP：‘7D 00’

需要认证的静态数据链接如下：

5F 25 03 07 01 01 5F 24 03 12 12 31 5A 08 62 25 00 00 00 00 00 89 5F 34 01 01 9F 0D 05 FC F0 E4 08 00 9F 0E 05 00 10 00 00 00 9F 0F 05 FC F8 E4 F8 80 9F 07 02 FF C0 8E 12 00 00 00 00 00 00 00 00 5E 03 41 03 42 03 60 03 1F 00 9F 4A 01 82 7D 00

用于计算的哈希的所有数据如下：

03 01 DA C1 BB 5F 25 03 07 01 01 5F 24 03 12 12 31 5A 08 62 25 00 00 00 00 00 89 5F 34 01 01 9F 0D 05 FC F0 E4 08 00 9F 0E 05 00 10 00 00 00 9F 0F 05 FC F8 E4 F8 80 9F 07 02 FF C0 8E 12 00 00 00 00 00 00 00 00 5E 03 41 03 42 03 60 03 1F 00 9F 4A 01 82 7D 00

5. 哈希算法标识(上述哈希输入数据的第二个字节，'01')指明使用SHA-1作为哈希函数

使用SHA-1函数计算出的上述数据的哈希结果如下：

‘4B 47 C5 8D 86 67 C3 10 44 40 AD DC F4 7C D9 8D 89 37 B2 CB’

6. 从恢复数据中获得的哈希值如下：

‘4B 47 C5 8D 86 67 C3 10 44 40 AD DC F4 7C D9 8D 89 37 B2 CB’

7. 比较哈希值：哈希值相同，SDA验证成功。

11.4 动态数据认证

生成和验证签名的动态应用数据包括下列步骤：

1. 由IC卡生成签名的动态应用数据
2. 验证发卡行公钥证书，包括提取发卡行公钥
3. 验证IC卡公钥证书，包括提取IC卡公钥
4. 验证签名的动态应用数据

本节叙述上述所有步骤，但不包括验证发卡行公钥证书。验证发卡行公钥证书类似于验证IC卡公钥证书。

11.4.1 动态应用数据签名的生成

IC卡私钥（标准形式）：

模：

9E B6 F2 D3 47 2F 73 35 B9 9F 03 D8 CD 69 E7 06 B7 E6 5B 8F 5A DD CE 0E 9A 00 9D C8
16 42 C3 F9 99 D3 C5 92 82 D0 4D B8 42 FC 68 58 09 B5 A7 99 76 88 AD 75 E2 C6 9D 91
18 97 27 E9 EF F0 A5 8D FE F8 31 37 D6 A9 15 9F 81 9E 31 61 0A C9 78 9A F1 33 BA 0A
4B 92 BD 1A 92 B9 E2 9E 6C 8B 62 01 67 03 E7 83 AA F8 57 D3 C4 BA 29 9B D3 CA C8 57
48 78 51 54 07 54 08 37 BB FC 6D F2 AE B5 6B 19 1F FF DF 36 6E CA D6 5A EB 15 10 77
1E 5E E5 8D

私钥指数（d）：

69 CF 4C 8C DA 1F A2 23 D1 14 AD 3B 33 9B EF 59 CF EE E7 B4 E7 3E 89 5F 11 55 BE 85
64 2C 82 A6 66 8D 2E 61 AC 8A DE 7A D7 52 F0 3A B1 23 C5 10 F9 B0 73 A3 EC 84 69 0B
65 BA 1A 9B F5 4B 19 09 54 A5 76 25 39 C6 0E 69 49 01 1D 81 C5 C2 B3 7A 43 56 18 92
4C FE 89 DE 8F 24 B5 56 B9 65 60 8D 49 64 A5 AF 9E 8B 3E AD ED BF 59 9F 18 9A F5 DA
A6 00 03 43 91 79 CB 24 77 6A D4 90 D4 59 10 9E 1D 28 DC DB EB 0D 6B 20 B8 A4 03 5A
FB 84 AA 93

RSA私钥（CRT模式）：

素数p：

D8 0A 38 E8 40 1A E7 3A 58 64 78 D9 A5 AA CB 3E 26 7D 73 6A 5E 07 E8 19 A2 9C 92 88
B3 C2 36 21 BF 73 96 E8 89 3B 6E A6 69 4F A6 71 A1 5B 2E 2C 3A 40 52 27 8A 48 D0 BD
6B 52 12 E0 E6 D8 E6 C2 F3 73 F9 AC 5C 5F 0A 63

素数q：

BC 12 4C 36 22 0A 84 29 33 CE 1C 55 32 6A 23 0E 95 85 5F 31 F8 6B 69 13 D6 50 5C 73
89 65 43 AD 20 A7 8C 44 A5 A6 E8 E8 E6 28 A5 FD 0B C2 29 54 CE 9B DC F1 E5 E7 01 6E
88 F0 81 0B A7 5D CE E6 E2 AB 11 C2 48 B8 DB 4F

dp：

90 06 D0 9A D5 67 44 D1 90 42 FB 3B C3 C7 32 29 6E FE 4C F1 94 05 45 66 6C 68 61 B0
77 D6 CE C1 2A 4D 0F 45 B0 D2 49 C4 46 35 19 A1 16 3C C9 72 D1 80 36 C5 06 DB 35 D3
9C E1 61 EB 44 90 99 D7 4C F7 FB C8 3D 94 B1 97

dq：

7D 61 88 24 16 B1 AD 70 CD 34 12 E3 76 F1 6C B4 63 AE 3F 76 A5 9C F0 B7 E4 35 92 F7
B0 EE 2D 1E 15 C5 08 2D C3 C4 9B 45 EE C5 C3 FE 07 D6 7D 61 88 24 16 B1 AD 70 CD 34
12 E3 76 F1 6C B4 63 AE 3F 76 A5 9C F0 B7 E4 35 92 F7 B0 EE 2D 1E 15 C5 08 2D C3 C4
9B 45 EE C5 C3 FE 07 D6 C6 38 89 BD 3D F6 99 44 AB 9F 05 F5 AB 5D 1A 3E 89 EF 41 C7
61 2C 30 7B 3C DF

Qinv：

46 29 3F FE 04 43 C7 17 F6 88 EE 6D 03 78 0F B8 42 9B 2A AF 5D EB 2B FE 7E 56 DA 0A
5C 34 02 AA 09 19 6A 01 8B 30 F1 0C 2C 7C AF B7 30 1C 64 6B FE 20 16 54 CA DE 09 DF
E4 AC 98 B8 B0 B7 D5 9D 71 AE 33 34 D1 D2 98 83

RSA公钥：

模(Module)：

9E B6 F2 D3 47 2F 73 35 B9 9F 03 D8 CD 69 E7 06 B7 E6 5B 8F 5A DD CE 0E 9A 00 9D C8
16 42 C3 F9 99 D3 C5 92 82 D0 4D B8 42 FC 68 58 09 B5 A7 99 76 88 AD 75 E2 C6 9D 91
18 97 27 E9 EF F0 A5 8D FE F8 31 37 D6 A9 15 9F 81 9E 31 61 0A C9 78 9A F1 33 BA 0A
4B 92 BD 1A 92 B9 E2 9E 6C 8B 62 01 67 03 E7 83 AA F8 57 D3 C4 BA 29 9B D3 CA C8 57
48 78 51 54 07 54 08 37 BB FC 6D F2 AE B5 6B 19 1F FF DF 36 6E CA D6 5A EB 15 10 77
1E 5E E5 8D

公钥指数：‘03’

按下列顺序链接计算哈希值的输入数据：

签名的数据格式：‘05’

哈希算法表示：‘01’ （SHA-1）

卡片动态数据长度：‘03’

卡片动态应用数据：‘02 00 02’

补位（144－28＝116字节，IC卡公钥模长为144字节）后的结果如下：

05 01 03 02 00 02 BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB
BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB
BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB
BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB
BB BB BB BB BB BB BB BB BB BB

需链接在上述数据后的终端不可预知数：

‘57 15 45 79’

所有用于计算动态应用数据哈希值的数据如下：

05 01 03 02 00 02 BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB
BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB
BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB
BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB
BB BB BB BB BB BB BB BB BB BB 57 15 45 79

使用SHA-1算法对上述数据计算的哈希值：

‘20 68 CE 12 C5 37 75 E4 44 54 B6 68 EF 76 9D 91 0E D1 00 CD’

生成签名：

首先按下列顺序链接数据：

数据头：‘6A’

需要签名的动态应用数据（哈希算法输入值的最左边122字节）：

05 01 03 02 00 02 BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB
BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB
BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB
BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB
BB BB BB BB BB BB BB BB BB BB

加上数据结尾：‘BC’

签名的输入数据为：

6A 05 01 03 02 00 02 BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB
BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB
BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB
BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB

BB BB BB BB BB BB BB BB BB BB 20 68 CE 12 C5 37 75 E4 44 54 B6 68 EF 76 9D 91 0E D1 00 CD BC

RSA运算模式：标准模式

签名的动态应用数据为：

47 5A F0 B1 72 48 D6 CD 3A D5 35 3F C5 8C 46 42 F4 AB FD 02 FA F5 95 14 E8 77 DF CF F6 FF 5B 01 93 6B A5 36 BA 14 0B A8 69 3B FD 8A 6C 76 EE 4A 22 F7 9D 39 3F DC B9 AE 81 D6 E8 E0 EE 12 98 2F 36 38 8B 04 B7 03 CF 18 B3 91 E0 93 20 58 D8 BD 91 02 95 04 E8 8C BF F2 7C 80 28 44 DC E7 96 DC FF 06 26 7A 32 97 D5 73 81 C5 BC F1 B9 B9 44 9D AB 25 F0 87 62 69 E3 EA 5C 4F 17 48 62 A3 B3 9F C1 31 F5 FD 7B 7F 03 B9 2A 8A 6E AC 92 F7 B9 1D

11.4.2 IC卡公钥证书的验证

从发卡行公钥证书及发卡行公钥余项中恢复的发卡行公钥如下：

发卡行公钥模：

C2 AB E7 63 CD 75 D5 7D DC D3 4C F6 32 AA 27 F5 E9 5A 52 04 56 2C 2D 39 E9 46 07 74 C7 61 B8 65 73 E9 D4 C1 B5 AC 4D AD A9 F4 2F 92 17 71 2B 73 D5 A6 6E 29 EA 8E 02 74 08 5F F6 33 CB 8E BB FA FB 13 F8 BC 82 63 84 E1 52 2F AB 4F C4 54 58 18 CB 6F 41 65 85 84 5E 7E 64 B7 21 A3 4B E4 8F AE F0 B0 78 DC BA DE BE 5F FA 22 A7 47 FF AB C8 EC F6 2F E4 B0 96 94 9F AE 88 A3 31 79 28 73 16 3B EC D9 0D 75 D8 F1 57 0F 47 ED 40 F7 86 90 B7 FB

发卡行公钥指数：‘03’

IC卡公钥证书(Tag ‘9F46’)如下：

30 C4 90 DB A9 3C ED D0 04 97 34 B8 97 39 CD C7 61 39 11 F9 44 B2 4B D0 72 FC 87 5E B5 A1 FF 52 E0 F1 90 FE 3C F7 74 3F D1 2F CA F7 57 ED DC E7 A3 A1 93 60 AB 19 BF 9E 0F 1D B7 6C E9 91 BA 7E 57 9E 49 D6 8C F2 A6 E2 4C A4 9A 36 75 D5 F4 7D 4A F9 FA 30 8A A4 21 0C A6 DF 9A 42 0C 0F E9 D2 CC E8 33 39 EC C9 53 7E B2 D9 35 48 5D BB DE 2D 82 28 8E DE 62 F0 A9 2B E0 17 5C 89 99 52 E2 82 FC 2D 6B F1 DF 48 CB B9 26 99 F7 3E 01 EF CB 31

执行以下验证过程验证IC卡公钥证书：

1. 检查IC卡公钥证书长度是否与发卡行公钥模长相同：
   比较结果：相同，为1152位或144字节。
2. 使用RSA算法恢复出签名数据的原文：
   发卡行公钥指数＝03，则：X＝(IC卡公钥证书)^3 mod (发卡行公钥模数)，其中X为恢复出的数据。
   从IC卡公钥证书中恢复的原数据如下：

6A 04 62 25 00 00 00 00 00 89 FF FF 12 20 00 00 02 01 01 90 01 9E B6 F2 D3 47 2F 73 35 B9 9F 03 D8 CD 69 E7 06 B7 E6 5B 8F 5A DD CE 0E 9A 00 9D C8 16 42 C3 F9 99 D3 C5 92 82 D0 4D B8 42 FC 68 58 09 B5 A7 99 76 88 AD 75 E2 C6 9D 91 18 97 27 E9 EF F0 A5 8D FE F8 31 37 D6 A9 15 9F 81 9E 31 61 0A C9 78 9A F1 33 BA 0A 4B 92 BD 1A 92 B9 E2 9E 6C 8B 62 01 67 03 E7 83 AA F8 82 44 00 08 41 1F 61 55 12 63 DF D8 60 40 57 65 D4 1E 8A 6F BC

3. 检查恢复数据的格式是否正确：
   恢复数据头（第一个字节）：‘6A’。
   检查结果：正确。
   证书格式（第二个字节）：‘04’。

检查结果：正确。

恢复数据结尾（最后一个字节）：‘BC’。

检查结果：正确。

4. 链接用于计算哈希值的数据：

恢复数据中需要参与哈希计算的部分(第2至第5个数据源，即除头、尾和哈希值外的所有数据)：

04 62 25 00 00 00 00 00 89 FF FF 12 20 00 00 02 01 01 90 01 9E B6 F2 D3 47 2F 73 35 B9 9F 03 D8 CD 69 E7 06 B7 E6 5B 8F 5A DD CE 0E 9A 00 9D C8 16 42 C3 F9 99 D3 C5 92 82 D0 4D B8 42 FC 68 58 09 B5 A7 99 76 88 AD 75 E2 C6 9D 91 18 97 27 E9 EF F0 A5 8D FE F8 31 37 D6 A9 15 9F 81 9E 31 61 0A C9 78 9A F1 33 BA 0A 4B 92 BD 1A 92 B9 E2 9E 6C 8B 62 01 67 03 E7 83 AA F8

参与哈希值计算的IC卡公钥指数与公钥余项：

IC卡公钥指数：‘03’

IC卡公钥余项：

57 D3 C4 BA 29 9B D3 CA C8 57 48 78 51 54 07 54 08 37 BB FC 6D F2 AE B5 6B 19 1F FF DF 36 6E CA D6 5A EB 15 10 77 1E 5E E5 8D

AFL中指明需要被认证的静态应用数据需要参与哈希计算，数据如下(顺序不可变)：

应用生效日期：‘5F 25 03 07 01 01’

应用失效日期：‘5F 24 03 12 12 31’

主账号(PAN)：‘5A 08 62 25 00 00 00 00 00 89’

主账号序列号：‘5F 34 01 01’

IAC－默认：‘9F 0D 05 FC F0 E4 08 00’

IAC－拒绝：‘9F 0E 05 00 10 00 00 00’

IAC－联机：‘9F 0F 05 FC F8 E4 F8 80’

应用使用控制：‘9F 07 02 FF C0’

持卡人验证方法列表：

‘8E 12 00 00 00 00 00 00 00 00 5E 03 41 03 42 03 60 03 1F 00’

SDA标签清单：‘9F 4A 01 82’

AIP：‘7D 00’

需要认证的静态数据连接如下：

5F 25 03 07 01 01 5F 24 03 12 12 31 5A 08 62 25 00 00 00 00 00 89 5F 34 01 01 9F 0D 05 FC F0 E4 08 00 9F 0E 05 00 10 00 00 00 9F 0F 05 FC F8 E4 F8 80 9F 07 02 FF C0 8E 12 00 00 00 00 00 00 00 00 5E 03 41 03 42 03 60 03 1F 00 9F 4A 01 82 7D 00

用于计算哈希值的所有数据如下：

04 62 25 00 00 00 00 00 89 FF FF 12 20 00 00 02 01 01 90 01 9E B6 F2 D3 47 2F 73 35 B9 9F 03 D8 CD 69 E7 06 B7 E6 5B 8F 5A DD CE 0E 9A 00 9D C8 16 42 C3 F9 99 D3 C5 92 82 D0 4D B8 42 FC 68 58 09 B5 A7 99 76 88 AD 75 E2 C6 9D 91 18 97 27 E9 EF F0 A5 8D FE F8 31 37 D6 A9 15 9F 81 9E 31 61 0A C9 78 9A F1 33 BA 0A 4B 92 BD 1A 92 B9 E2 9E 6C 8B 62 01 67 03 E7 83 AA F8 57 D3 C4 BA 29 9B D3 CA C8 57 48 78 51 54 07 54 08 37 BB FC 6D F2 AE B5 6B 19 1F FF DF 36 6E CA D6 5A EB 15 10 77 1E 5E E5 8D 03 5F 25 03 07 01 01 5F 24 03 12 12 31 5A 08 62 25 00 00 00 00 00 89 5F 34 01 01 9F 0D 05 FC F0 E4 08 00 9F 0E 05 00 10 00 00 00 9F 0F 05 FC F8 E4 F8 80 9F 07 02 FF C0 8E 12 00 00 00 00 00 00 00 00 5E 03 41 03 42 03 60 03 1F 00 9F 4A 01 82 7D 00

5. 哈希算法标识(上述哈希输入数据的第二个字节，‘01’)，指明使用SHA-1作为哈希函数

使用SHA-1函数计算出的上述数据的哈希结果如下：

‘82 44 00 08 41 1F 61 55 12 63 DF D8 60 40 57 65 D4 1E 8A 6F’

6. 从恢复数据中获得的哈希值如下：

‘82 44 00 08 41 1F 61 55 12 63 DF D8 60 40 57 65 D4 1E 8A 6F ‘

7. 比较哈希值：相同。

8. 检查IC卡证书中恢复的PAN与卡片的PAN是否一致

IC卡中的PAN：‘62 25 00 00 00 00 00 89’

证书中恢复的PAN：‘62 25 00 00 00 00 00 89 FF FF’

PAN比较结果：相同。

9. 检查IC卡证书的是否在有效期内

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IC卡公钥证书验证失败。

IC卡证书的有效期(MM YY)是：‘12 20’，即（YY YYMM）：‘20 20 12’

当前日期(YY YY MM)是：‘20 07 11’

证书失效期检查结果：通过。

10. 检查恢复数据中的IC卡公钥算法标识是否能识别

IC卡公钥算法标识是：‘01’，算法标识可以识别为RSA。

ICC卡公钥证书验证成功。链接从上面X获得的IC卡公钥模左边部分与IC卡公钥模余项得到IC卡的公钥模如下：

9E B6 F2 D3 47 2F 73 35 B9 9F 03 D8 CD 69 E7 06 B7 E6 5B 8F 5A DD CE 0E 9A 00 9D C8
16 42 C3 F9 99 D3 C5 92 82 D0 4D B8 42 FC 68 58 09 B5 A7 99 76 88 AD 75 E2 C6 9D 91
18 97 27 E9 EF F0 A5 8D FE F8 31 37 D6 A9 15 9F 81 9E 31 61 0A C9 78 9A F1 33 BA 0A
4B 92 BD 1A 92 B9 E2 9E 6C 8B 62 01 67 03 E7 83 AA F8 57 D3 C4 BA 29 9B D3 CA C8 57
48 78 51 54 07 54 08 37 BB FC 6D F2 AE B5 6B 19 1F FF DF 36 6E CA D6 5A EB 15 10 77
1E 5E E5 8D

11.4.3 验证签名的动态应用数据

IC卡公钥：

从IC卡公钥证书获取的IC卡公钥模：

9E B6 F2 D3 47 2F 73 35 B9 9F 03 D8 CD 69 E7 06 B7 E6 5B 8F 5A DD CE 0E 9A 00 9D C8
16 42 C3 F9 99 D3 C5 92 82 D0 4D B8 42 FC 68 58 09 B5 A7 99 76 88 AD 75 E2 C6 9D 91
18 97 27 E9 EF F0 A5 8D FE F8 31 37 D6 A9 15 9F 81 9E 31 61 0A C9 78 9A F1 33 BA 0A
4B 92 BD 1A 92 B9 E2 9E 6C 8B 62 01 67 03 E7 83 AA F8 57 D3 C4 BA 29 9B D3 CA C8 57
48 78 51 54 07 54 08 37 BB FC 6D F2 AE B5 6B 19 1F FF DF 36 6E CA D6 5A EB 15 10 77
1E 5E E5 8D

IC卡公钥指数：‘03’

签名的动态应用数据：

47 5A F0 B1 72 48 D6 CD 3A D5 35 3F C5 8C 46 42 F4 AB FD 02 FA F5 95 14 E8 77 DF CF
F6 FF 5B 01 93 6B A5 36 BA 14 0B A8 69 3B FD 8A 6C 76 EE 4A 22 F7 9D 39 3F DC B9 AE
81 D6 E8 E0 EE 12 98 2F 36 38 8B 04 B7 03 CF 18 B3 91 E0 93 20 58 D8 BD 91 02 95 04
E8 8C BF F2 7C 80 28 44 DC E7 96 DC FF 06 26 7A 32 97 D5 73 81 C5 BC F1 B9 B9 44 9D
AB 25 F0 87 62 69 E3 EA 5C 4F 17 48 62 A3 B3 9F C1 31 F5 FD 7B 7F 03 B9 2A 8A 6E AC
92 F7 B9 1D

签名验证步骤如下：

1. 检查签名的动态应用数据的长度与IC卡公钥模长是否相等

检查结果：相等，为1152位或144字节。

2. 使用RSA算法和IC卡公钥恢复签名的动态应用数据的结果如下：

6A 05 01 03 02 00 02 BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB
BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB
BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB
BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB
BB BB BB BB BB BB BB BB BB BB BB 20 68 CE 12 C5 37 75 E4 44 54 B6 68 EF 76 9D 91 0E
D1 00 CD BC

3. 检查恢复数据的格式是否正确：

恢复数据的头(第一个字节)：‘6A’。

检查结果：通过。

恢复数据结尾(最后一个字节)：‘BC’。

检查结果：通过。

4. 链接用于计算哈希值的数据：

恢复数据中需要参与哈希计算的部分(第2至第10个数据源，即除头、尾和哈希结果外的所有数据)：

05 01 03 02 00 02 BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB
BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB
BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB
BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB
BB BB BB BB BB BB BB BB BB BB

用于链接到上面数据后参与哈希计算的不可预知数：‘57 15 45 79’。

用于计算哈希值的所有数据如下：

05 01 03 02 00 02 BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB
BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB
BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB
BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB
BB BB BB BB BB BB BB BB BB BB 57 15 45 79

5. 哈希算法标示(上述哈希输入数据的第二个字节，'01')，指明使用SHA-1作为哈希函数

使用SHA-1函数计算出的上述数据的哈希结果如下：

‘20 68 CE 12 C5 37 75 E4 44 54 B6 68 EF 76 9D 91 0E D1 00 CD’

6. 从恢复数据中获得的哈希值如下：

‘20 68 CE 12 C5 37 75 E4 44 54 B6 68 EF 76 9D 91 0E D1 00 CD’

7. 比较哈希值：相同。签证签名的动态应用数据成功。

11.5 CDA 生成与验证

11.5.1 复合 DDA/应用密文生成（CDA）

本节提供复合DDA/AC生成的实例，这里应用密文是TC。

生成CDA中的签名动态应用数据：

第一步：分解GenerateAC命令输入数据、GPO命令输入数据

GenerateAC命令输入数据（CDOL1相关数据）：

授权金额：‘00 00 00 00 01 00’

其它金额：‘00 00 00 00 00 00’

终端国家代码：‘01 56’

交易货币代码：‘01 56’

交易日期：‘07 11 09’

交易类型：‘01’

不可预知数：‘BB 31 E4 A4’

GPO命令输入数据：‘01 56’

第二步： 位计算签名应用数据生成交易数据哈希值

计算交易哈希值得输入数据包括：GPO命令输入数据，GenAC命令输入数据，GenerateAC响应数据中除SDAD外的其他数据。

GPO命令输入数据如下：‘01 56’

GenAC命令输入数据如下：

‘00 00 00 00 01 00 00 00 00 00 00 01 56 00 00 00 00 00 01 56 07 11 09 01 BB 31 E4 A4’

GenerateAC响应数据中除SDAD外的其他数据：

‘9F 27 01 40 9F 36 02 00 01 9F 10 08 07 01 0A 03 90 00 02 01’

所有用于计算交易哈希的数据如下：

01 56 00 00 00 00 01 00 00 00 00 00 00 01 56 00 00 00 00 00 01 56 07 11 09 01 BB 31 E4 A4 9F 27 01 40 9F 36 02 00 01 9F 10 08 07 01 0A 03 90 00 02 01

对上述数据使用SHA-1算法计算的哈希值如下：

‘3E 93 51 D2 21 BC DF 2F 69 44 0C 4E 3A C6 8B E8 31 93 4B 49’

第三步： 计算待签名的动态应用数据的哈希值

按如下顺序连接待签名的动态应用数据：

签名数据格式： ‘05’

哈希算法标识：‘01’

IC卡动态数据长度：‘20’

IC卡动态数字：‘02 00 01’

应用密文类型（CID）：‘40’

应用密文（AC）：‘B4 AD F3 53 F2 21 E0 54’

交易数据哈希值：

‘3E 93 51 D2 21 BC DF 2F 69 44 0C 4E 3A C6 8B E8 31 93 4B 49’

补位(144-57=87字节,这里IC卡的公钥模长是144字节)：

不可预知数(终端随机数)：‘BB 31 E4 A4’

所有用于计算动态应用数据哈希值的数据如下：

05 01 20 02 00 01 40 B4 AD F3 53 F2 21 E0 54 3E 93 51 D2 21 BC DF 2F 69 44 0C 4E 3A C6 8B E8 31 93 4B 49 BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB 31 E4 A4

对上述数据使用SHA-1算法计算的哈希值如下：

‘7D 32 6A 62 A0 1E 47 42 C8 E8 42 E9 EA 0A 17 40 BC 74 59 04’

第四步： 使用IC卡私钥签署动态应用数据

按如下顺序链接所有数据：

1． 数据头：‘6A’

2． 计算动态应用数据哈希值的数据的最左边144-22=122个字节(即待签名的动态应用数据)：

05 01 20 02 00 01 40 B4 AD F3 53 F2 21 E0 54 3E 93 51 D2 21 BC DF 2F 69 44 0C 4E 3A C6 8B E8 31 93 4B 49 BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB

BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB
BB BB BB BB BB BB BB BB BB BB

3． 动态应用数据的哈希值：

‘7D 32 6A 62 A0 1E 47 42 C8 E8 42 E9 EA 0A 17 40 BC 74 59 04’

4． 数据尾：‘BC’

按以上顺序链接所有数据得到如下结果：

6A 05 01 20 02 00 01 40 B4 AD F3 53 F2 21 E0 54 3E 93 51 D2 21 BC DF 2F 69 44 0C 4E
3A C6 8B E8 31 93 4B 49 BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB
BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB
BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB
BB BB BB BB BB BB BB BB BB BB BB 7D 32 6A 62 A0 1E 47 42 C8 E8 42 E9 EA 0A 17 40 BC
74 59 04 BC

对上述数据运用RSA算法和IC卡私钥计算出签名的动态应用数据，签名动态应用数据＝（输入）^{d} mod（IC卡公钥模）如下：

92 E3 1D 96 B3 72 CE EA 7E CC C7 A5 96 AF 02 A0 2E E3 7C 6B 2B C2 F8 B8 07 2A 9A FB
39 03 1D D6 2F 4F 39 45 1A 07 0E C7 51 FC F9 36 02 04 07 7E A6 E9 61 72 1B E6 20 36
26 9F B2 FB 57 E2 4E A4 3B 09 05 E2 43 17 1D 7C 8C 6E F8 32 85 39 A0 E7 EB 0F B2 1B
1D DD E8 7E FB 6E F6 BE 03 D5 3F 0E 38 A3 79 1B EB 18 61 7B F6 C0 71 F8 02 BB 3E FB
49 E6 A8 06 27 50 D4 AE F9 8A F2 73 4C 66 D0 73 1E 9F EF 1B 69 4F D3 70 5A 61 7C 66
19 12 7D 79

至此完成CDA签名数据生成。

GenAC的响应数据如下：

77 81 A8 9F 27 01 40 9F 36 02 00 01 9F 4B 81 90 92 E3 1D 96 B3 72 CE EA 7E CC C7 A5
96 AF 02 A0 2E E3 7C 6B 2B C2 F8 B8 07 2A 9A FB 39 03 1D D6 2F 4F 39 45 1A 07 0E C7
51 FC F9 36 02 04 07 7E A6 E9 61 72 1B E6 20 36 26 9F B2 FB 57 E2 4E A4 3B 09 05 E2
43 17 1D 7C 8C 6E F8 32 85 39 A0 E7 EB 0F B2 1B 1D DD E8 7E FB 6E F6 BE 03 D5 3F 0E
38 A3 79 1B EB 18 61 7B F6 C0 71 F8 02 BB 3E FB 49 E6 A8 06 27 50 D4 AE F9 8A F2 73
4C 66 D0 73 1E 9F EF 1B 69 4F D3 70 5A 61 7C 66 19 12 7D 79 9F 10 08 07 01 0A 03 90
00 02 01

11.5.2 CDA 验证

第一步： 分解GenAC命令输入数据和响应数据，GPO命令输入数据

1． GenAC命令输入数据：

授权金额：‘00 00 00 00 01 00’

其他金额：‘00 00 00 00 00 00’

终端国家代码：‘01 56’

终端验证结果：‘00 00 00 00 00’

交易货币代码：‘01 56’

交易类型：‘01’

不可预知数：‘BB 31 E4 A4’

2． GenAC命令响应数据：

[9F27] CID：‘40’

[9F36] ATC：‘00 01’

3． [9F4B] 签名动态应用数据：

92 E3 1D 96 B3 72 CE EA 7E CC C7 A5 96 AF 02 A0 2E E3 7C 6B 2B C2 F8 B8 07 2A 9A FB
39 03 1D D6 2F 4F 39 45 1A 07 0E C7 51 FC F9 36 02 04 07 7E A6 E9 61 72 1B E6 20 36
26 9F B2 FB 57 E2 4E A4 3B 09 05 E2 43 17 1D 7C 8C 6E F8 32 85 39 A0 E7 EB 0F B2 1B
1D DD E8 7E FB 6E F6 BE 03 D5 3F 0E 38 A3 79 1B EB 18 61 7B F6 C0 71 F8 02 BB 3E FB
49 E6 A8 06 27 50 D4 AE F9 8A F2 73 4C 66 D0 73 1E 9F EF 1B 69 4F D3 70 5A 61 7C 66
19 12 7D 79

4. [9F10] IAD：‘07 01 0A 03 90 00 02 01’

5. GPO命令输入数据：‘01 56’

第二步： 验证签名的动态应用数据是否正确

IC卡公钥模：

9E B6 F2 D3 47 2F 73 35 B9 9F 03 D8 CD 69 E7 06 B7 E6 5B 8F 5A DD CE 0E 9A 00 9D C8
16 42 C3 F9 99 D3 C5 92 82 D0 4D B8 42 FC 68 58 09 B5 A7 99 76 88 AD 75 E2 C6 9D 91
18 97 27 E9 EF F0 A5 8D FE F8 31 37 D6 A9 15 9F 81 9E 31 61 0A C9 78 9A F1 33 BA 0A
4B 92 BD 1A 92 B9 E2 9E 6C 8B 62 01 67 03 E7 83 AA F8 57 D3 C4 BA 29 9B D3 CA C8 57
48 78 51 54 07 54 08 37 BB FC 6D F2 AE B5 6B 19 1F FF DF 36 6E CA D6 5A EB 15 10 77
1E 5E E5 8D

IC卡公钥指数：‘03’

签名的动态应用数据（Tag9F4B）：

92 E3 1D 96 B3 72 CE EA 7E CC C7 A5 96 AF 02 A0 2E E3 7C 6B 2B C2 F8 B8 07 2A 9A FB
39 03 1D D6 2F 4F 39 45 1A 07 0E C7 51 FC F9 36 02 04 07 7E A6 E9 61 72 1B E6 20 36
26 9F B2 FB 57 E2 4E A4 3B 09 05 E2 43 17 1D 7C 8C 6E F8 32 85 39 A0 E7 EB 0F B2 1B
1D DD E8 7E FB 6E F6 BE 03 D5 3F 0E 38 A3 79 1B EB 18 61 7B F6 C0 71 F8 02 BB 3E FB
49 E6 A8 06 27 50 D4 AE F9 8A F2 73 4C 66 D0 73 1E 9F EF 1B 69 4F D3 70 5A 61 7C 66
19 12 7D 79

按照下列方式验证签名：

1. 检查签名动态应用数据的长度与IC卡公钥模长是否相等

   检查结果：相等。长度为1152位、144字节。

2. 使用RSA算法和IC卡公钥恢复签名动态应用数据的结果如下：

6A 05 01 20 02 00 01 40 B4 AD F3 53 F2 21 E0 54 3E 93 51 D2 21 BC DF 2F 69 44 0C 4E
3A C6 8B E8 31 93 4B 49 BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB
BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB
BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB
BB BB BB BB BB BB BB BB BB BB BB 7D 32 6A 62 A0 1E 47 42 C8 E8 42 E9 EA 0A 17 40 BC
74 59 04 BC

   从恢复数据中提取的交易数据包括：

   [9F4C] IC卡动态数字：‘00 01’

   [9F27] 应用密文类型：‘40’

   [9F26] 应用密文：‘B4 AD F3 53 F2 21 E0 54’

3. 检查恢复数据的格式是否正确：

   恢复数据的头(第一个字节)：‘6A’。

   检查结果：通过。

   证书格式(第二个字节)：‘BC’。

   检查结果：通过。

4. 链接用于计算哈希值的数据：

恢复数据中需要参与哈希计算的部分(第2至第10个数据源，即除头、尾和哈希结果外的所有数据)：

05 01 20 02 00 01 40 B4 AD F3 53 F2 21 E0 54 3E 93 51 D2 21 BC DF 2F 69 44 0C 4E 3A C6 8B E8 31 93 4B 49 BB

需链接在上述数据的后面参与哈希计算的终端不可预知数：‘BB 31 E4 A4’

用于计算的哈希值的所有数据如下：

05 01 20 02 00 01 40 B4 AD F3 53 F2 21 E0 54 3E 93 51 D2 21 BC DF 2F 69 44 0C 4E 3A C6 8B E8 31 93 4B 49 BB 31 E4 A4

5． 哈希算法标识(上述哈希输入数据的第二个字节，’01’)，指明使用SHA-1作为哈希函数

使用SHA-1函数计算出的上述数据的哈希值如下：

‘7D 32 6A 62 A0 1E 47 42 C8 E8 42 E9 EA 0A 17 40 BC 74 59 04’

6． 从恢复数据中获得的哈希值如下：

‘7D 32 6A 62 A0 1E 47 42 C8 E8 42 E9 EA 0A 17 40 BC 74 59 04’

7． 比较哈希值：

哈希值相同。签名的动态应用数据验证成功。

第三步： 验证交易数据哈希值是否正确

计算交易哈希值的输入数据包括：GPO命令输入数据，GenAC命令输入数据，GenerateAC响应数据中除签名动态应用数据外的其他数据。

GPO命令输入数据如下：‘01 56’

GenAC命令输入数据如下：

‘00 00 00 00 01 00 00 00 00 00 00 01 56 00 00 00 00 00 01 56 07 11 09 01 BB 31 E4 A4’

GenerateAC响应数据中除签名动态应用数据外的其他数据：

‘9F 27 01 40 9F 36 02 00 01 9F 10 08 07 01 0A 03 90 00 02 01’

上述数据的链接值（哈希函数输入）：

‘01 56 00 00 00 00 01 00 00 00 00 00 00 01 56 00 00 00 00 00 01 56 07 11 09 01 BB 31 E4 A4 9F 27 01 40 9F 36 02 00 01 9F 10 08 07 01 0A 03 90 00 02 01’

签名动态应用数据中恢复出的交易哈希值如下：

‘3E 93 51 D2 21 BC DF 2F 69 44 0C 4E 3A C6 8B E8 31 93 4B 49’

重新计算的交易哈希值如下：

‘3E 93 51 D2 21 BC DF 2F 69 44 0C 4E 3A C6 8B E8 31 93 4B 49’

验证结果：哈希值相同，CDA验证成功。