ICS 35.240.40
A 11
备案号：

JR

中 华 人 民 共 和 国 金 融 行 业 标 准

JR/T 0025.17—2013

中国金融集成电路（IC）卡规范
第17部分：借记/贷记应用安全增强规范

China financial integrated circuit card specifications—
Part17: Enhanced debit/credit application security specification

2013-02-05 发布　　2013-02-05 实施

中国人民银行 发布

目　　次

前　言

JR/T 0025《中国金融集成电路（IC）卡规范》分为以下部分：

——第1部分：电子钱包/电子存折应用卡片规范（废止）；

——第2部分：电子钱包/电子存折应用规范（废止）；

——第3部分：与应用无关的IC卡与终端接口规范；

——第4部分：借记/贷记应用规范；

——第5部分：借记/贷记应用卡片规范；

——第6部分：借记/贷记应用终端规范；

——第7部分：借记/贷记应用安全规范；

——第8部分：与应用无关的非接触式规范；

——第9部分：电子钱包扩展应用指南（废止）；

——第10部分：借记/贷记应用个人化指南；

——第11部分：非接触式IC卡通讯规范；

——第12部分：非接触式IC卡支付规范；

——第13部分：基于借记/贷记应用的小额支付规范；

——第14部分：非接触式IC卡小额支付扩展应用规范；

——第15部分：电子现金双币支付应用规范；

——第16部分：IC卡互联网终端规范；

——第17部分：借记/贷记应用安全增强规范。

本部分为JR/T 0025的第17部分。

本部分依据GB/T 1.1-2009给出的规则起草。

本部分由中国人民银行提出。

本部分由全国金融标准化技术委员会（SAC/TC180）归口。

本部分主要起草单位：中国人民银行、国家密码管理局商用密码管理办公室、总参三部、中国工商银行、中国建设银行、中国邮政储蓄银行、中国银联股份有限公司、中国金融电子化公司、中国金融认证中心、银行卡检测中心、北京中电华大电子设计有限责任公司、北京诺君安信息技术有限公司、北京江南天安科技有限公司、北京华大信安科技有限公司、北京华大智宝电子系统有限公司、上海格尔软件股份有限公司、航天信息股份有限公司。

本部分主要起草人：王永红、李晓枫、陆书春、潘润红、杜宁、陈则栋、吴晓光、安晓龙、谢永泉、刘平、徐志忠、陈芳、汤洋、严伟峰、李东风、张永峰、赵宇、李春欢、张栋、汤沁莹、仲祺、施海平、李一凡、史大鹏、李建峰、李新、陈震宇、郑元龙、董浩然、韩小西、李国、汪朝晖、陈跃、谭武征、罗世新。

本部分为首次发布。

引　言

本部分是对JR/T 0025.7的扩展，以支持SM2、SM3和SM4等密码算法在借记/贷记应用中的使用。本部分介绍了认证中心、发卡行和IC卡使用SM2算法进行数字签名，使用SM3算法计算哈希值，使用SM4算法进行数据加密及安全报文计算的实现。

中国金融集成电路（IC）卡规范
第17部分：借记/贷记应用安全增强规范

1 范围

本部分作为JR/T 0025.7的增强，主要规定描述了基于SM2、SM3、SM4算法的借记/贷记应用安全功能方面的要求以及为实现这些安全功能所涉及的安全机制和获准使用的加密算法，包括：基于SM2、SM3的IC卡脱机数据认证方法，基于SM4的IC卡和发卡行之间的通讯安全以及为实现这些安全功能所涉及的安全机制和加密算法的规范。

本部分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、终端机具及管理等。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、集成和维护等相关部门（单位）。

2 规范性引用文件

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

JR/T 0025.4 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 中国金融集成电路（IC）卡规范 第5部分：借记/贷记应用卡片规范

JR/T 0025.7 中国金融集成电路（IC）卡规范 第7部分：借记/贷记应用安全规范

GM/T 0002 SM4分组密码算法

GM/T 0003 SM2椭圆曲线公钥密码算法

GM/T 0004 SM3密码杂凑算法

GM/T AAAA SM2密码算法使用规范

3 术语和定义

下列术语和定义适用于本文件。

3.1

应用 application

卡片和终端之间的应用协议和相关的数据集。

3.2

命令 command

终端向IC卡发出的一条报文，该报文启动一个操作或请求一个响应。

3.3

密文 cryptogram

加密运算的结果。

3.4

金融交易 financial transaction

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

3.5

集成电路 integrated circuit (IC)

具有处理和/或存储功能的电子器件。

3.6

集成电路卡（IC卡） integrated circuit(s) card (ICC)

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。

3.7

接口设备 interface device

终端上插入IC卡的部分，包括其中的机械和电气部分。

3.8

发卡行行为代码 issuer action code

发卡行根据TVR的内容选择的动作。

3.9

磁条 magstripe

包括磁编码信息的条状物。

3.10

支付系统环境 payment system environment

当符合JR/T 0025的支付系统应用被选择，IC卡中所确立的逻辑条件集合。

3.11

响应 response

IC卡处理完成收到的命令报文后，返回给终端的报文。

3.12

脚本 script

发卡行向终端发送的命令或命令序列，目的是向IC卡连续输入命令。

3.13

SM2算法 SM2 algorithm

一种椭圆曲线公钥密码算法，其密钥长度为256比特。

3.14

SM3算法 SM3 algorithm

一种密码杂凑算法，其输出为256比特。

3.15

SM4算法 SM4 algorithm

一种分组密码算法，分组长度为128比特，密钥长度为128比特。

3.16

终端 terminal

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

3.17

终端行为代码 terminal action code

收单行根据TVR的内容选择的动作。

4 符号和缩略语

下列符号和缩略语适用于本文件。

AAC 应用认证密文（Application Authentication Cryptogram）

AC 应用密文（Application Cryptogram）

AFL	应用文件定位器（Application File Locator）
AIP	应用交互特征（Application Interchange Profile）
ARC	授权响应码（Authorization Response Code）
ARPC	授权响应密文（Authorization Response Cryptogram）
ARQC	授权请求密文（Authorization Request Cryptogram）
ATC	应用交易计数器（Application Transaction Counter）
ATM	自动柜员机（Automated Teller Machine）
AUC	应用用途控制（Application Usage Control）
CDA	复合动态数据认证/应用密文生成（Combined DDA/AC Generation）
CDOL	卡片风险管理数据对象列表（Card Risk Management Data Object List）
CID	密文信息数据（Cryptogram Information Data）
Cn	压缩数字型（Compressed Numeric）
CVM	持卡人验证方法（Cardholder Verification Method）
CVR	卡片验证结果（Card Verification Results）
DDA	动态数据认证（Dynamic Data Authentication）
DDOL	动态数据认证数据对象列表（Dynamic Data Authentication Data Object List）
DES	DES 数据加密标准（Data Encryption Standard）
DOL	数据对象列表（Data Object List）
ECB	电子密码本（Electronic Code Book）
EF	基本文件（Elementary File）
EMV	Europay、MasterCard 和 VISA
FCI	文件控制信息（File Control Information）
GPO	获取处理选项（Get Processing Options）
IAC	发卡行行为代码（Issuer Action Code）
IC	集成电路（Integrated Circuit）
M	必备（Mandatory）
MAC	报文鉴别码（Message Authentication Code）
MDK	主密钥（Master Key）
N	数字型（Numeric）
O	可选（Optional）
PAN	主账号（Primary Account Number）
PIN	个人识别码（Personal Identification Number）
RSA	RSA 公钥密码算法（Rivest, Shamir, Adleman Algorithm）
S_{CA}	认证中心私钥（Certification Authority Private Key）
SAD	签名的静态应用数据（Signed Static Application Data）
SDA	静态数据认证（Static Data Authentication）
SFI	短文件标识符（Short File Identifier）
SHA	SHA 安全哈希（杂凑）算法（Secure Hash Algorithm 1）
S_I	发卡行私钥（Issuer Private Key）
S_{IC}	IC 卡私钥（ICC Private Key）
$Sign(S_K)[X]$	用私钥 S_K，通过 SM2 算法，对数据块 X 进行签名
SW1	状态字 1（Status Word One）
SW2	状态字 2（Status Word Two）
TC	交易证书（Transaction Certificate）

TLV	标签、长度、值（Tag Length Value）
TVR	终端验证结果（Terminal Verification Results）
UDK	子密钥（Unique Key）
Verify(P_K)[X,S]	用公钥 P_K，通过 SM2 算法，对数据块 X 的签名结果 S 进行验证
X:=ALG^{-1}(K)[Y]	用密钥 K，通过 64 位或 128 位分组加密方法，对 64 位或 128 位数据块 Y 进行解密
Y:=ALG(K)[X]	用密钥 K，通过 64 位或 128 位分组加密方法，对 64 位或 128 位数据块 X 进行加密

5 脱机数据认证

5.1 静态数据认证（SDA）

5.1.1 密钥和证书

认证中心使用认证中心私钥S_{CA}，对表1中指明的数据使用SM2算法进行签名，得到格式如表4所示的发卡行公钥证书。

发卡行使用发卡行私钥S_I，对表2中指明的数据使用SM2算法进行签名，得到签名的静态应用数据，其格式见表5。

执行静态数据认证需要的必要数据元在表3中定义，如果缺少这些数据中的任意一项，静态数据认证失败。

表1 由认证中心签名的发卡行公钥数据（待签名数据）

字段名	长度	描述	格式
证书格式（记录头）	1	十六进制，值为‘12’	B
发卡行标识	4	主账号最左面的 3-8 个数字。(在右边补上十六进制数‘F’)	Cn 8
证书失效日期	2	MMYY，在此日期后，这张证书无效	N 4
证书序列号	3	由认证中心分配给这张证书的唯一的二进制数	b
发卡行公钥签名算法标识	1	标识发卡行公钥对应的数字签名算法。SM2 算法为‘04’。	b
发卡行公钥加密算法标识	1	标识发卡行公钥对应的加密算法，暂不使用，取值‘00’。	b
发卡行公钥参数标识	1	用于标识所用的椭圆曲线。见附录 A.4	b
发卡行公钥长度	1	标识发卡行公钥字节长度	b
发卡行公钥	N_I	SM2 公钥是椭圆曲线上的一个点	b

对表 1 中的数据计算进行 SM2 签名的结果是两个大整数 r 和 s，签名格式见 GM/T AAAA，将字节串 r||s 附着在表 1 之后就形成了用 SM2 签名的发卡行公钥证书，其格式见表 4。密码算法标识的定义见附录 A。

表2 由发卡行签名的静态应用数据（待签名数据）

字段名	长度	描述	格式
签名数据格式	1	十六进制，值为‘13’	b
数据验证代码	2	由发卡行分配的代码	b
需认证的静态数据	变长	在 JR/T 0025.5 的 9.3.1 节指明的需认证的静态数据	-

表3　SM2 签名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’	$N_{CA}+N_{I}+14$	SM2 签名的发卡行公钥证书数据，格式见表 4	B
‘93’	$N_{I}+3$	SM2 签名静态应用数据，格式见表 5	B
–	变长	在 JR/T 0025.5 的 9.3.1 节指明的需认证的静态数据	–

5.1.2　发卡行公钥获取

认证中心采用SM2算法签名发卡行公钥证书，终端获取的发卡行证书数据如表4所示，包括被签名的明文数据及数字签名。发卡行公钥以明文形式包含在发卡行公钥证书中，用认证中心的公钥验证发卡行公钥证书中的签名字段。如验证通过，则从发卡行公钥证书中提取公钥信息。

表4　发卡行公钥证书的格式

字段名	长度	描述	格式
证书格式	1	十六进制，值为‘12’	B
发卡行标识	4	主账号最左面的 3-8 个数字（在右边补上十六进制数‘F’）	cn 8
证书失效日期	2	MMYY，在此日期后，这张证书无效	n4
证书序列号	3	由认证中心分配给这张证书的，唯一的二进制数	b
发卡行公钥签名算法标识	1	标识使用在发卡行公钥上的数字签名算法。SM2 算法为‘04’。	b
发卡行公钥加密算法标识	1	标识使用在发卡行公钥上的加密算法，暂不使用，取值‘00’。	b
发卡行公钥参数标识	1	用于标识椭圆曲线，同时确定 N_{I}。见附录 A.4	b
发卡行公钥长度	1	标识发卡行公钥字节长度	b
发卡行公钥	N_{I}	对于 SM2 算法是椭圆曲线上的一个点	b
数字签名	N_{CA}	认证中心对表 1 的数据计算的 SM2 签名 r||s	b

使用SM2算法签名的发卡行公钥证书验证步骤如下：

获取并解析如表4所示的发卡行公钥证书数据。如果失败，那么静态数据认证失败。

a) 检查证书格式的值。如果不是“12”，那么静态数据认证失败。

b) 比较发卡行标识的值是否与应用主账号最左面的3-8个数字一致（允许发卡行标识在其后补“F”）。如果不一致，那么静态数据认证失败。

c) 比较证书失效日期中指定年月的最后日期与当天的日期。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。

d) 检查连接起来的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。

e) 检查发卡行公钥签名算法标识，如果不是“04”，那么静态数据认证失败。

f) 准备表4中前9个数据元（即表1数据）。

g) 使用认证中心公钥和相应的认证中心签名算法按照8.2.3条中指明的验证方法对表4的数字签名进行验证。如果验证签名失败，那么静态数据认证失败。

h) 如果以上所有的检验都通过，继续下面的流程。

5.1.3 签名的静态应用数据验证

终端获取的SM2签名静态应用数据的格式如表5所示，包括被签名的明文数据及数字签名。

表5 签名的静态应用数据格式

字段名	长度	描述	格式
签名数据格式	1	十六进制，值为‘13’	B
数据验证代码	2	由发卡行分配的代码	B
数字签名	N_I	发卡行对表 2 中数据计算的 SM2 签名 r||s	B

验证步骤如下：

a) 获取并解析如表5所示的经过发卡行签名的签名静态数据。如果失败，则静态数据认证失败。

b) 检查签名数据格式的值。如果不是“13”，那么静态数据认证失败。

c) 准备表5的前2个数据元及JR/T 0025.5的9.3.1条中指明的需认证的静态数据(用于验证签名)。如果静态数据认证标签列表存在，并且其包含非“82”的标签，那么静态数据认证失败。

d) 使用发卡行公钥和相应的发卡行签名算法并将8.2.3条中指明的验证方法对表5的数字签名进行验证。如果验证签名失败，那么静态数据认证失败。

如果以上所有的步骤都成功，那么静态数据认证成功。终端应将表5中的数据验证代码存放在标签“9F45”中。

5.2 动态数据认证（DDA）

5.2.1 密钥和证书

认证中心使用认证中心私钥S_{CA}，对表1中指明的数据使用SM2算法进行签名，得到格式如表4所示的发卡行公钥证书。

发卡行使用发卡行私钥S_I，对表6中指定的数据使用SM2算法签名，得到IC卡公钥证书，其格式见表8。

执行动态数据认证需要的必要数据元在表7中定义，如果缺少这些数据中的任意一项，动态数据认证失败。

表6 由发卡行签名的 IC 卡公钥数据（待签名数据）

字段名	长度	描述	格式
证书格式	1	十六进制，值为‘14’	b
应用主账号	10	主账号（在右边补上十六进制数‘F’）	Cn 20
证书失效日期	2	MMYY，在此日期后，这张证书无效	n4
证书序列号	3	由发卡行分配给这张证书的唯一的二进制数	b
IC 卡公钥签名算法标识	1	标识 IC 卡公钥对应的数字签名算法	b
IC 卡公钥加密算法标识	1	标识 IC 卡公钥对应的加密算法，暂不使用，取值‘00’。	b
IC 卡公钥参数标识	1	用于标识椭圆曲线，同时确定 N_{IC}。见附录 A.4	b
IC 卡公钥长度	1	标识 IC 卡公钥的字节长度	b
IC 卡公钥	N_{IC}	如果 IC 卡公钥算法标识对应于 SM2,该字段为椭圆曲线上的一个点	b
参与脱机数据认证的静态数据	变长	在 JR/T 0025.5 中定义	b

对表6中数据进行SM2签名的结果是两个大整数r和s,将字节串r||s附着在表6的前9项之后就形成了用SM2签名的IC卡公钥证书，其格式见表8。

表7 动态认证中的公钥认证所需的数据对象

标签	长度	值	格式
–	5	注册的应用提供商标识	B
‘8F’	1	认证中心公钥索引	B

‘90’	$N_{CA}+N_I+14$	SM2 签名的发卡行公钥证书数据，格式见表 4	B
‘9F46’	$N_I+N_{IC}+20$	SM2 签名的 IC 卡公钥证书数据，格式见表 8	B
—	变长	JR/T 0025.5 第 9.3.1 节详细说明了需认证的静态数据	-

5.2.2 发卡行公钥的获取

见 5.1.2。

5.2.3 IC 卡公钥的获取

终端获取的IC卡公钥证书数据如表8所示。IC卡公钥以明文形式包含在IC卡公钥证书中，终端用发卡行的公钥验证IC卡公钥证书中的签名字段。如验证通过，则从IC卡公钥证书中提取公钥信息。

表8 发卡行使用 SM2 签名的 IC 卡公钥证书的格式

字段名	长度	描述	格式
证书格式	1	十六进制，值为‘14’	B
应用主账号	10	主账号（在右边补上十六进制数‘F’）	cn 20
证书失效日期	2	MMYY，在此日期后，这张证书无效	n4
证书序列号	3	由发卡行分配给这张证书的唯一的二进制数	B
IC 卡公钥签名算法标识	1	标识使用在 IC 卡公钥上的数字签名算法。SM2 算法为‘04’。	B
IC 卡公钥加密算法标识	1	标识使用在 IC 卡公钥上的加密算法，暂不使用，取值‘00’。	B
IC 卡公钥参数标识	1	用于标识所用的椭圆曲线。见附录 A.4	B
IC 卡公钥长度	1	标识 IC 卡公钥的字节长度	B
IC 卡公钥	N_{IC}	对于 SM2 算法是椭圆曲线上的一个点	B
数字签名	N_I	发卡行对表 6 数据计算的 SM2 签名 r||s	B

验证步骤如下：

a) 获取并解析如表8所示的经过IC卡公钥证书数据。如果失败，则动态数据认证失败。

b) 检查证书格式的值。如果不是“14”，那么动态数据认证失败。

c) 比较证书中的主账号和从IC卡读出的应用主账号是否相同。如果不同，那么动态数据认证失败。

d) 比较证书失效日期中指定年月的最后日期与当天的日期。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。

e) 准备表8中的前9个数据元以及JR/T 0025.5的9.3.1条指明的需认证的静态数据（用于验证签名）。如果静态数据认证标签列表存在，并且其包含非“82”的标签，那么动态数据认证失败。

f) 检查IC卡公钥签名算法标识，如果不是“04”，那么动态数据认证失败。

g) 使用发卡行公钥和相应的发卡行签名算法将8.2.3条中指明的验证方法对表8的数字签名进行验证。如果验证签名失败，那么动态数据认证失败。

h) 如果以上所有的检验都通过，继续下面的流程。

5.2.4 标准动态数据认证

5.2.4.1 动态签名的生成

使用 SM2 算法生成动态签名按以下的步骤进行：

a) 终端发出内部认证（INTERNAL AUTHENTICATE）命令，命令中包含由DDOL指定的数据元，这些数据元按JR/T 0025.4中指明的规则连接在一起。

b) IC卡使用IC卡私钥对表9中指明的数据计算SM2签名，得到如表11的格式的SM2签名动态应用数据。

表9 需签名的动态应用数据（待签名数据）

字段名	长度	描述	格式

签名的数据格式	1	十六进制，值为‘15’	B
IC 卡动态数据长度	1	标识 IC 卡动态数据的字节长度 L_{DD}	B
IC 卡动态数据	L_{DD}	由 IC 卡生成和/或存储在 IC 卡上的动态数据	–
终端动态数据	变长	由 DDOL 指定的数据元连接而成	–

IC卡动态数据的定义见JR/T 0025.7。

除了表9中指明的数据，动态数据认证所需的数据对象在表10中定义。

表10 生成和检验动态签名所需要的其它数据对象

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

5.2.4.2 动态签名的验证

使用SM2签名动态应用数据，终端获取的签名动态应用数据的格式如表11所示，包括被签名的明文数据及数字签名。终端使用IC卡的公钥验证动态应用数据的签名。

表11 IC 卡使用 SM2 签名的动态应用数据的格式

字段名	长度	描述	格式
签名的数据格式	1	十六进制，值为‘15’	B
IC 卡动态数据长度	1	标识 IC 卡动态数据的字节长度 L_{DD}	B
IC 卡动态数据	L_{DD}	由 IC 卡生成和/或存储在 IC 卡上的动态数据	–
数字签名	N_{IC}	IC 卡对表 9 中数据计算的 SM2 签名 r||s	B

验证步骤如下：

a) 获取并解析如表11所示的经过发卡行签名的动态数据。如果失败，则动态数据认证失败。

b) 检查签名的数据格式的值。如果不是“15”，那么动态数据认证失败。

c) 准备表11中的前3个数据元（即从签名数据格式直到IC卡动态数据）及DDOL中指定的数据元，即表9数据。

d) 使用IC卡公钥和相应的IC卡签名算法将8.2.3条中指明的验证方法对表11的数字签名进行验证。如果验证签名失败，那么动态数据认证失败。

如果以上所有的步骤都成功，那么动态数据认证成功。终端应将表11中的IC卡动态数据中所包含的IC卡动态数字存放在标签“9F4C”中。

5.2.5 复合动态数据认证/应用密文生成(CDA)

5.2.5.1 动态签名的生成

IC 卡使用 SM2 算法生成动态签名，复合动态签名和应用密文生成按以下的步骤进行：

a) 终端根据JR/T 0025.5中的定义发出生成应用密文（GENERATE AC）命令，并且命令中CDA请求位为1。

b) 如果IC卡将以TC或ARQC作为响应，则IC卡执行如下步骤：

——IC 卡生成 TC 或 ARQC；

——IC 卡应用由 SM3 对从左到右连接的如下数据元进行哈希运算：

- 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。

32 字节的哈希运算结果称作交易数据哈希值。

——IC 卡使用 IC 卡私钥对表 12 中的数据计算 SM2 签名，形成签名动态应用数据，格式如表 17 所示。

表12　需签名的动态应用数据（待签名数据）

字段名	长度	描述	格式
签名的数据格式	1	十六进制，值为‘15’	b
IC 卡动态数据长度	1	标识 IC 卡动态数据的字节长度 L_{DD}	b
IC 卡动态数据	L_{DD}	由 IC 卡生成和/或存储在 IC 卡上的动态数据	-
不可预知数	4	由终端生成的不可预知数	b

IC卡动态数据的最左边的32-38个字节由表13中指明的数据连接而成。

表13　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
32	交易数据哈希值	b

IC卡动态数字的定义见JR/T 0025.7。

IC卡对生成应用密文（GENERATE AC）命令的响应应按照JR/T 0025.5中定义的格式2（带有标签‘77’的结构数据对象）编码，且应包含表14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14　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	应用交易计数器	必须
‘9F4B’	$N_{IC}+L_{DD}+6$	SM2 签名的动态应用数据，在表 12 后附上 SM2 签名即得	必须
‘9F10’	变长，最长 32	发卡行应用数据	可选

c) 如果IC卡以AAC作为响应，那么IC卡的响应应按照JR/T 0025.5中定义的格式1或格式2编码，且应包含表15中指明的三个必须数据对象，可选包含发卡行应用数据。

表15　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	应用交易计数器	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16所示的格式编码。

表16　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密钥，密文版本号指示了应用密文的计算方式，6.1条描述了生成应用密文的方法。密文版本号和算法标识的定义，见附录A。

5.2.5.2　动态签名的验证

假定终端已成功地按上面讲述的过程取回了IC卡公钥。

IC卡采用SM2签名动态应用数据，终端获取的签名动态应用数据的格式如表17所示，包括被签名的明文数据及数字签名。终端使用IC卡的公钥验证动态应用数据的签名。

表17　SM2 签名的动态应用数据

字段名	长度	描述	格式
签名的数据格式	1	十六进制，值为‘15’	b
IC 卡动态数据长度	1	标识 IC 卡动态数据的字节长度 L_{DD}	b
IC 卡动态数据	L_{DD}	由 IC 卡生成和/或存储在 IC 卡上的动态数据	-
数字签名	N_{IC}	IC 卡对表 12 中数据计算的 SM2 签名 r||s	b

SM2验证步骤如下：

a) 获取并解析如表17所示的经过IC卡签名的动态数据。如果失败，那么复合动态数据认证/应用密文生成失败。

b) 检查签名的数据格式的值。如果不是“15”，那么复合动态数据认证/应用密文生成失败。

c) 从IC卡动态数据中取得表13中指明的数据。

d) 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。

e) 准备表17中的前3个数据元（即从签名数据格式直到IC卡动态数据）及不可预知数，即表12数据。

f) 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 GPO 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 GPO 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。

i） 使用SM3算法应用到上一步的连接结果从而得到交易数据哈希值。

j） 把上一步计算得到的交易数据哈希值和步骤3中从IC卡动态数据中取得的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。

k） 使用IC卡公钥和相应的算法并将8.2.3条中指明的验证方法对表17的数字签名进行验证。如果验证签名失败，那么复合动态数据认证/应用密文生成失败。

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。终端应将表13中的IC卡动态数据中所包含的IC卡动态数字存放在标签“9F4C”中，将表13中的ARQC值或TC值存放在标签“9F26”中。

6 应用密文和发卡行认证

6.1 应用密文产生

6.1.1 数据源选择

见JR/T 0025.7。

6.1.2 应用密文算法

以一个唯一的 16 字节 IC 卡应用密文（AC）子密钥 MK_{AC}，和 6.1.1 条描述的数据源作为输入，按以下两步计算 8 字节的应用密文：

a） 以IC卡应用密文（AC）子密钥MK_{AC}和两字节的IC卡应用交易计数器作为输入，使用8.1.3条描述的算法，生成16字节的应用密文过程密钥SK_{AC}。

b） 使用上一步生成的16字节应用密文过程密钥SK_{AC}和“经选择的数据”作为输入，按照8.1.2条中指明的MAC算法计算得到应用密文（TC、ARQC或AAC）。

详细密文生成的步骤如下：

步骤 1：终端将 CDOL 中指定的终端数据通过生成应用密文命令传送给卡片。如果 CDOL 中有要交易证书（TC）哈希结果，终端要将此数据放到命令数据域中。

步骤 2：根据卡片风险管理的结果，卡片决定返回的密文类型为 TC、AAC 或 ARQC。生成密文的数据块：

——交易证书（TC）哈希结果（如果存在）；

——生成应用密文命令中送进卡片的数据。不包括 TC 哈希结果；

——卡片内部数据。

步骤 3：按照 8.1.2 条的“填充并分块”中描述的方法对数据进行填充与分块。

步骤 4：如图 1，使用过程密钥用对称密钥算法生成应用密文（过程密钥是由 IC 卡应用密文（AC）子密钥 MK_{AC} 分散生成，具体生成方法见 8.1.3 条）。

步骤 5：取上一步计算结果的左边 8 字节，得到 8 字节的密文。

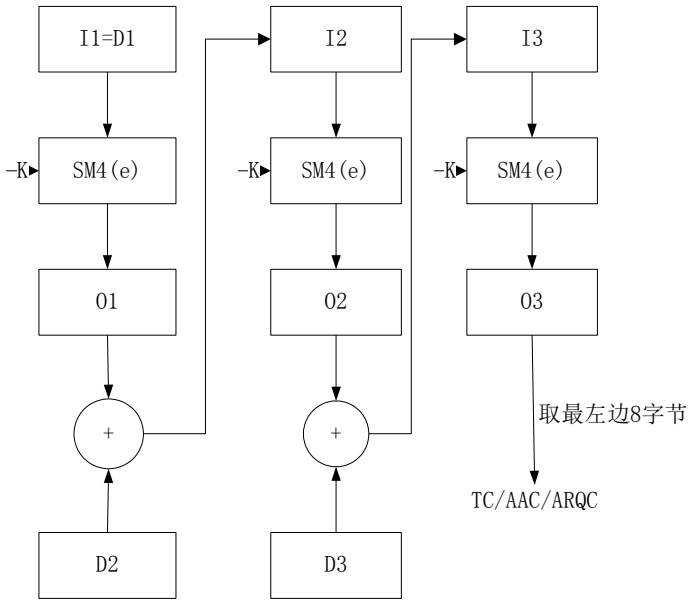


图1　TC/AAC/ARQC 的生成算法

6.2　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 8.1.2 条）按照 11.1.2 条中指明的对称加密算法对 8.1.2 条生成的 8 字节长的 ARQC 和 2 字节的授权响应码 ARC 进行加密，加密步骤如下：

a)　在2字节的ARC的后面补上6个‘00’字节来获得一个8字节的数

X :=（ARC||‘00’||‘00’||‘00’||‘00’||‘00’||‘00’）。

b)　计算 Y:= ARQC⊕X。

c)　计算ARPC：

将Y左对齐后面补8个字节00形成D；

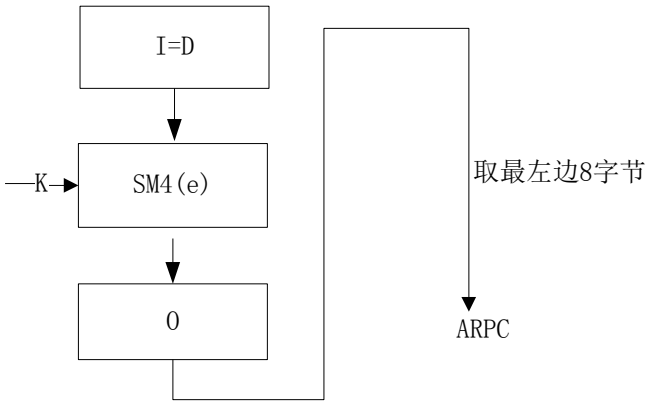
D:= Y ||‘00’||‘00’||‘00’||‘00’ || ‘00’ ||‘00’|| ‘00’ || ‘00’

基于16字节分组加密算法获得16字节ARPC0；

ARPC0:= SM4（SK_{AC}）[D]；

取ARPC0的左边8字节得到ARPC。

图2是ARPC的生成方法。



说明：

I = 输入　　D = 数据块

SM4(e)= SM4算法（加密模式）　　K = 密钥

0 = 输出

图2　生成 ARPC 的算法

7　安全报文

7.1　报文完整性及其验证

7.1.1　MAC 过程密钥产生

安全报文MAC生成的第一步包括从IC卡的唯一的16字节安全报文鉴别（MAC）子密钥和2字节ATC分散得到一个唯一的16字节安全报文鉴别码（MAC）过程密钥。过程密钥产生方法见8.1.3条。

7.1.2　MAC 的计算

MAC是通过使用按照7.1.1条中描述的方法产生的MAC过程密钥并将8.1.2条中描述的机制应用在所要保护的报文上计算得到的。

在本部分中MAC长度为4，在按上面描述的方法计算得到16个字节的结果后，取其中最左面的4字节作为MAC。

7.2　报文私密性

7.2.1　加密过程密钥产生

安全报文加/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。过程密钥产生方法见8.1.3条。

7.2.2　加密解密

对明文/加密命令数据域的加/解密是通过使用按照7.2.1条中描述的方法产生的加密过程密钥并应用8.1.1条中描述的机制进行的。

8　安全机制

8.1　对称加密机制

8.1.1　加密解密

对数据的加密采用 16 字节分组加密算法，可以是电子密码本（ECB）模式或密码块链接（CBC）模式。JR/T 0025 选用 ECB 模式作为加密解密模式。

用加密过程密钥 K_S 对任意长度的报文 MSG 加密的步骤如下：

a)　填充并分块

- 如果报文 MSG 的长度不是分组长度的整数倍，在 MSG 的右端加上 1 个‘80’字节，然后再在右端加上最少的‘00’字节，使得结果报文的长度 MSG：=(MSG||‘80’||‘00’||‘00’||…||‘00’)是分组长度的整数倍。
- 如果报文 MSG 的长度是分组长度的整数倍，不对数据作填充。

被加密数据首先要被格式化为以下形式的数据块：

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

然后 MSG 被拆分为 16 字节的块 X_1，X_2，…，X_k。

b) 密文计算

1) 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，. . . ，XK 加密为 16 字节的块 Y_1，Y_2，. . . ，Y_K，因此当 i ＝1，2，… ，K 时分别计算：$Y_i := ALG(K_S)[X_i]$。

2) 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，. . . ，XK 加密为 16 字节的块 Y_1，Y_2，. . . ，Y_K，因此当 i ＝1，2，… ，K 时分别计算：$Y_i := ALG(K_S)[X_i \oplus Y_{i-1}]$

Y_0的初始值为：

Y_0 :=（‘00’ || ‘00’ || ‘00’ || ‘00’ || ‘00’ || ‘00’ || ‘00’ || ‘00’ || ‘00’ || ‘00’ || ‘00’ || ‘00’ || ‘00’ || ‘00’ || ‘00’ || ‘00’）

记为：Y := (Y_1 || Y_2 || … || Y_K) = ENC(K_S)[MSG]。

解密过程如下：

a) ECB 模式

当 i = 1，2，... ，K 时，分别计算：$X_i := ALG^{-1}(K_S)[Y_i]$。

b) CBC 模式

当 i = 1，2，... ，K 时，分别计算：$X_i := ALG^{-1}(K_S)[Y_i] \oplus Y_{i-1}$。

Y_0的初始值为：

Y_0 :=（‘00’ || ‘00’ || ‘00’ || ‘00’ || ‘00’ || ‘00’ || ‘00’ || ‘00’ || ‘00’ || ‘00’ || ‘00’ || ‘00’ || ‘00’ || ‘00’ || ‘00’ || ‘00’）

为了得到原来的报文 MSG，将块 X_1，X_2，... ，X_K连接起来，如果使用了填充，从最后一块 X_k中删除尾部的(‘80’ || ‘00’ || ‘00’ || ... || ‘00’)，记为：MSG = DEC(K_S)[Y]。

8.1.2 报文鉴别码

采用 CBC 模式的 16 字节分组加密算法以及 MAC 过程密钥 K_S对任意长度的报文 MSG 计算一个 S 字节的 MAC（4≤S≤8）值 S 的步骤如下。

a) 填充并分块

依据GB/T 16649.4对报文MSG进行填充，因此在MSG的右端强制加上1个‘80’字节，然后再在右端加上最少的‘00’字节,使得结果报文的长度MSG:=(MSG || ‘80’ || ‘00’ || ‘00’ ||…|| ‘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，… ，X_K。

b) MAC过程密钥

MAC过程密钥K_S长度为16字节。

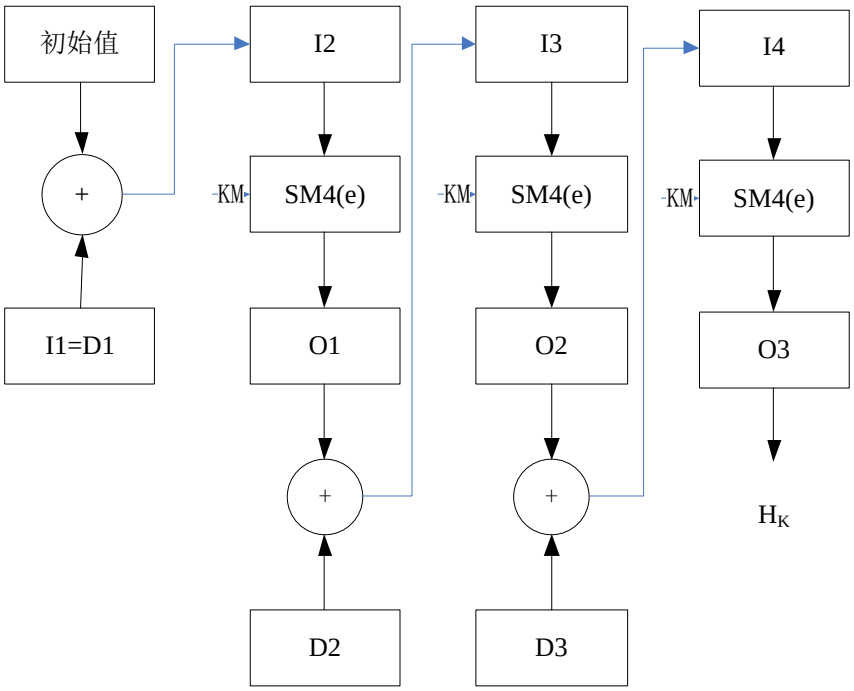
c) 密文计算

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，... ，X_K：

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

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

使用SM4算法计算H_K的算法如图3所示。



说明：

I = 输入　　D = X = 数据块
SM4(e)= SM4算法（加密模式）　　KM = MAC过程密钥
O = 输出　　+ = 异或

图3　使用 SM4 算法计算 H_K 的算法

最终密文生成分以下两种情况：

a) 在报文完整性及验证时，取 H_K 的前 4 字节作为 MAC 值。

b) 计算应用密文（TC、ARQC 或 AAC）时，取 H_K 的左边 8 字节作为应用密文。

8.1.3　过程密钥产生

MAC 和数据加密过程密钥的产生如下所述：

——第一步：卡片/发卡行决定是使用 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 4 所示的运算产生过程密钥。

Z :=SM4(Key)[‘00’ ||‘00’||‘00’ ||‘00’ || ‘00’ || ‘00’ ||ATC || ‘00’ || ‘00’ || ‘00’ || ‘00’ || ‘00’ || ‘00’ || (ATC⊕‘FFFF’)]

过程密钥产生流程如图 4 所示。

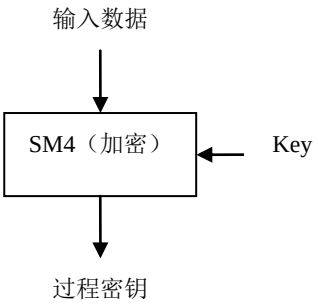


图4　过程密钥产生流程

8.1.4　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。

这一方式以主账号（PAN）和主账号序列号（如果主账号序列号不存在，则用一个字节‘00’代替）的最右16个数字及其衍生数据作为输入数据，以及16字节的发卡行主密钥IMK作为密钥，生成16字节的IC卡子密钥MK作为输出：

a) 将主账号和主账号序列号连接生成数据块X，如果X的长度小于16个数字，X右对齐，在最左端填充十六进制的‘0’以获得8字节的Y。如果X的长度至少有16个数字，那么Y由X的最右边的16个数字组成。

b) 计算：

Z:=SM4(IMK)[Y||(Y(‘FF’||‘FF’||‘FF’||‘FF’||‘FF’||‘FF’||‘FF’||‘FF’))]，16字节的IC卡子密钥 MK就等于Z。

IC卡子密钥分散流程图见5。

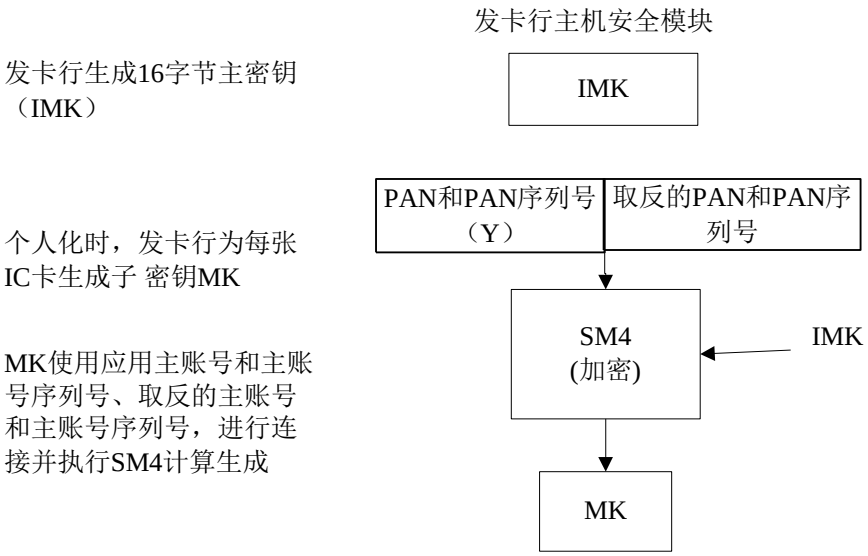


图5　子密钥分散

8.2　非对称密码机制

8.2.1　本部分使用 GM/T 0003 的椭圆曲线算法进行数字签名算法

SM2签名方案使用下面三种函数：

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

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

——一个哈希算法 SM3[ ]，将任意长度的报文映射为一个 32 字节的哈希值。

8.2.2 数字签名产生

对任意长度的数据组成的报文MSG计算签名S的过程如下：

a) 计算Z_A=SM3[$ENTL_A$ || ID_A || a || b || x_G ||y_G || x_A || y_A]。其中ID_A固定设置为16字节定长的十六进制数据0x31, 0x32, 0x33, 0x34, 0x35, 0x36, 0x37, 0x38, 0x31, 0x32, 0x33, 0x34, 0x35, 0x36, 0x37, 0x38；$ENTL_A$值为两个字节数据0x00, 0x80；

b) 计算报文MSG的32字节的HASH值 h:= SM3 [Z_A||MSG]；

c) 计算Sign(S_K)[h]，得到两个数字r和s；

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

8.2.3 数字签名验证

对任意长数据组成的报文MSG验证签名S的过程如下：

a) 计算Z_A=SM3[$ENTL_A$ || ID_A || a || b || x_G ||y_G || x_A || y_A]。其中ID_A固定设置为16字节定长的十六进制数据0x31, 0x32, 0x33, 0x34, 0x35, 0x36, 0x37, 0x38, 0x31, 0x32, 0x33, 0x34, 0x35, 0x36, 0x37, 0x38；$ENTL_A$值为两个字节数据0x00, 0x80；

b) 计算报文MSG的32字节的HASH值 h:= SM3 [Z_A||MSG]；

c) Verify(P_K)[h, S]，若函数输出True表示验证正确，若输出False，表示验证失败。

9 认可的算法

9.1 对称加密算法

本部分使用的对称加密算法为 SM4 算法,算法定义见 GM/T 0002。

9.2 非对称算法

本部分使用的非对称算法为SM2算法，算法定义见GM/T 0003。

9.3 哈希算法

本部分使用的哈希算法为SM3算法, 算法定义见GM/T 0004。

10 算法选择与交易流程

本章规定了支持SM2/SM3/SM4算法（以下简称SM算法）的IC卡在进行交易时的算法选择方案。

10.1 新增数据元

新增数据元的定义见表18。

表18 新增数据元

数据元名称	标签	长度	格式
SM算法支持指示器	DF69	1	b

10.2 SM 算法应用方案

本条包括了单 SM 算法卡和双算法卡两种方案的技术要求。单 SM 算法卡指一张金融 IC 卡支持 SM2/SM3/SM4 算法；双算法卡指一张金融 IC 卡同时支持 SM2/SM3/SM4 与 RSA/SHA-1/3DES 两套算法。

如采用单 SM 算法卡方案，在卡片个人化阶段，在卡内写入支持 SM2 算法所需的相关数据元；如采用双算法卡方案,在卡片个人化阶段,在卡内写入支持 RSA 算法和 SM2 算法两种算法所需的相关数据元。卡片个人化完成之后，卡片交易时应通过和终端之间的交互确定使用的算法。技术要求主要包括如下：

本条所述支持 SM 算法的终端应同时支持 RSA/SHA-1/3DES 及 SM2/SM3/SM4 两套算法环境，并使用 SM 算法支持指示器进行算法选择。

卡片个人化阶段：

a) 单 SM 算法卡：在卡片个人化数据包括了 SM2 算法所需的所有数据元；

b）双算法卡：在卡片个人化数据包括了 RSA 算法所需的所有数据元，同时包括了 SM2 算法所需的所有数据元。

两种算法数据元及在卡内的存储及芯片系统的访问相关要求，本部分不做具体要求。

卡片应用流程：

见 10.3、10.4 和 10.5。

应用执行情况：

不同类型终端和卡片在标准借记/贷记应用的执行情况如表 19 所示。

表19　标准借记/贷记应用和基于借记/贷记应用的小额支付的执行情况

	仅支持 RSA/SHA-1/3DES 算法的卡片	双算法卡	单 SM 算法卡
仅支持 RSA/SHA-1/3DES 算法的终端	RSA/SHA-1/3DES 算法流程	RSA/SHA-1/3DES 算法流程	脱机认证失败，可能尝试进行联机交易
支持双算法的终端	RSA/SHA-1/3DES 算法流程	SM 算法流程	SM 算法流程

不同类型终端和卡片在 qPBOC 应用的执行情况如表 20 所示。

表20　qPBOC 应用的执行情况

	仅支持 RSA/SHA-1/3DES 算法的卡片	双算法卡	单 SM 算法卡
仅支持 RSA/SHA-1/3DES 算法的终端	RSA/SHA-1/3DES 算法流程	RSA/SHA-1/3DES 算法流程	拒绝交易
支持双算法的终端	RSA/SHA-1/3DES 算法流程	SM 算法流程	SM 算法流程

10.3　借记贷记应用流程

10.3.1　流程概述

借记贷记应用流程如图 6 所示，具体步骤如下：

a）首先卡片在接收终端应用选择命令后，返回的PDOL数据列表中应包括SM算法支持指示器（见10.1）；具体流程详见10.3.2节；

b）终端根据系统算法支持情况，设置SM算法支持指示器标签，发送GPO指令至卡片；

c）双算法卡片根据GPO的指令参数，确定卡片的算法环境和数据（单SM算法卡不判断算法环境，直接返回相关数据），详见10.3.3节。终端根据卡片返回的AIP和AFL进行应用数据读取、脱机数据认证等其他处理流程；

d）脱机数据认证过程中，终端根据公钥索引检查算法类型；

e）当终端请求Generate AC命令时，卡片返回的相应报文中，包含ARQC、发卡行自定义数据，以及其他数据。在发卡行自定数据中包含对称加密、解密使用的算法；

f）终端收到数据之后，放入联机报文的55域，发送到发卡行进行联机认证；

g）发卡行收到数据之后，解析获得发卡行自定义数据，并根据算法标识字段指定的算法，进行数据加密、解密运算，验证ARQC和产生ARPC。

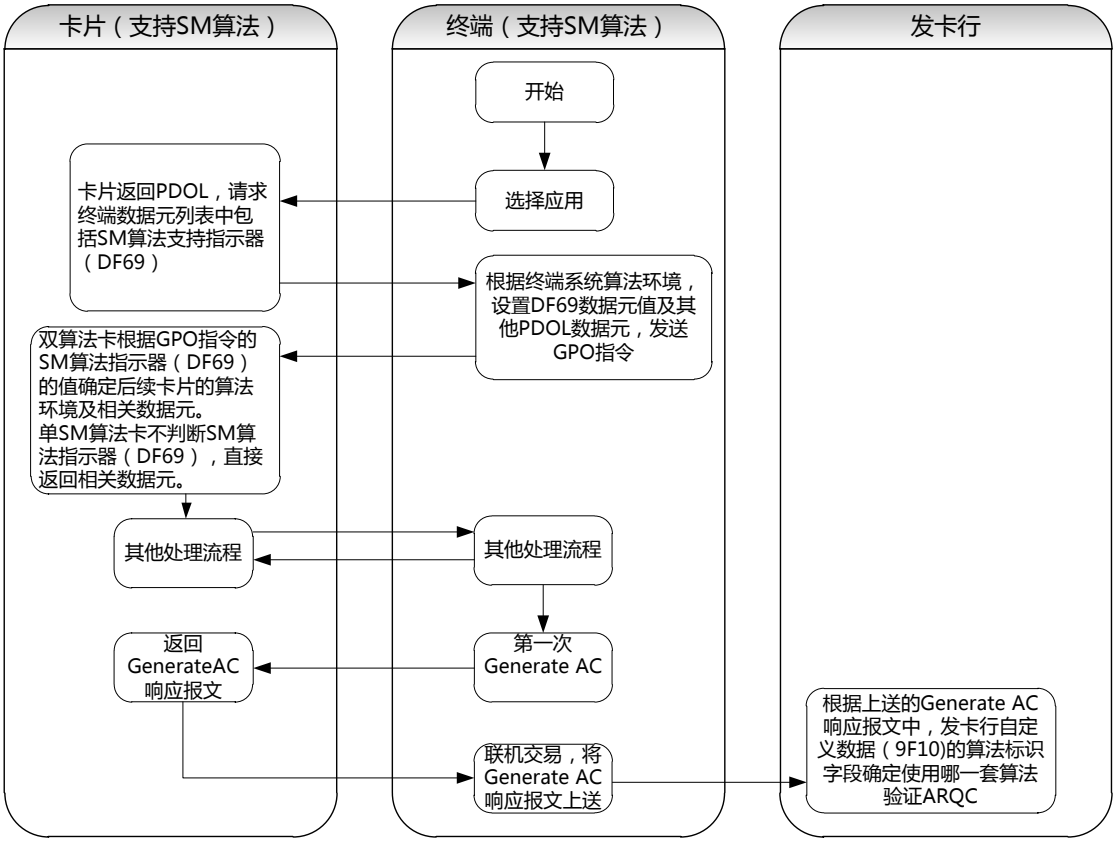


图6　SM 算法应用于借贷记应用的流程

10.3.2　应用选择

终端发送SELECT命令选择应用，卡片返回文件控制信息（FCI），如果卡片支持SM算法，则其中应包括PDOL，PDOL中应含有SM算法支持指示器（DF69）。

10.3.3　应用初始化

如果终端支持SM算法，终端将在GET PROCESSING OPTIONS命令中提供SM算法支持指示器（DF69设置为“1”）。对于单SM算法卡片和双算法卡片，处理流程分别如下：

a）　单SM算法卡片：

——直接返回GET PROCESSING OPTIONS响应数据为SM算法的AFL。AFL指明了SM算法相关的特定数据，包括SM2算法的公钥参数及证书的位置。

b）　双算法卡片：

——根据GPO指令的SM算法指示器（DF69）的值标识本次交易走SM算法流程,确定后续卡片的算法环境及相关数据元；

——返回GET PROCESSING OPTIONS响应数据为SM算法的AFL。AFL指明了SM算法相关的特定数据，包括SM2算法的公钥参数及证书的位置。

如果终端不支持SM算法，即SM算法指示标签（DF69）无法被终端识别，终端将在GET PROCESSING OPTIONS命令中提供一个指定长度的数据元，并将其数值部分设置为16进制的0。对于单SM算法卡片和双算法卡片，处理流程分别如下：

a）　单SM算法卡片：

—— 直接返回GET PROCESSING OPTIONS响应数据为SM算法AFL。AFL指明了SM算法相关的特定数据，包括SM2算法的公钥参数及证书的位置。这将导致脱机数据认证过程因无法找到对应的公钥索引而失败，交易可以尝试联机交易或拒绝交易。

b）　双算法卡片：

——根据GPO指令的SM算法指示器（DF69）的值标识本次交易默认走RSA/SHA-1/3DES算法流程，确定后续卡片的算法环境及相关数据元；

——返回GET PROCESSING OPTIONS响应数据为RSA/SHA-1/3DES算法的AFL。AFL指明了RSA/SHA-1/3DES算法相关的特定数据，包括RSA算法的公钥参数及证书的位置。

10.3.4 后续流程

见JR/T 0025.4的定义。

10.4 基于借记贷记应用的小额支付流程

SM算法应用于基于借记/贷记应用的小额支付流程如图7所示：

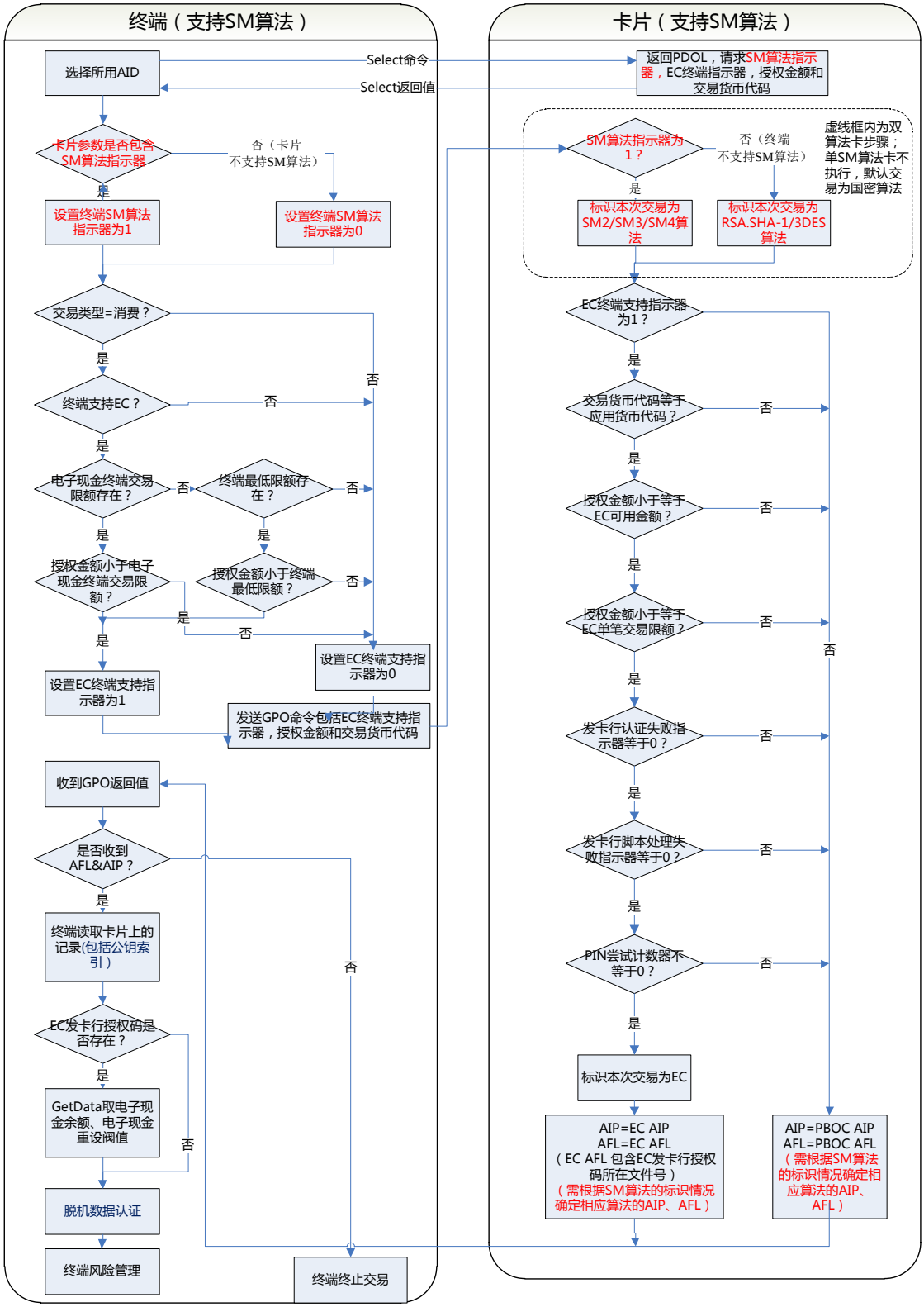


图7　SM 算法应用于基于借记/贷记应用的小额支付流程

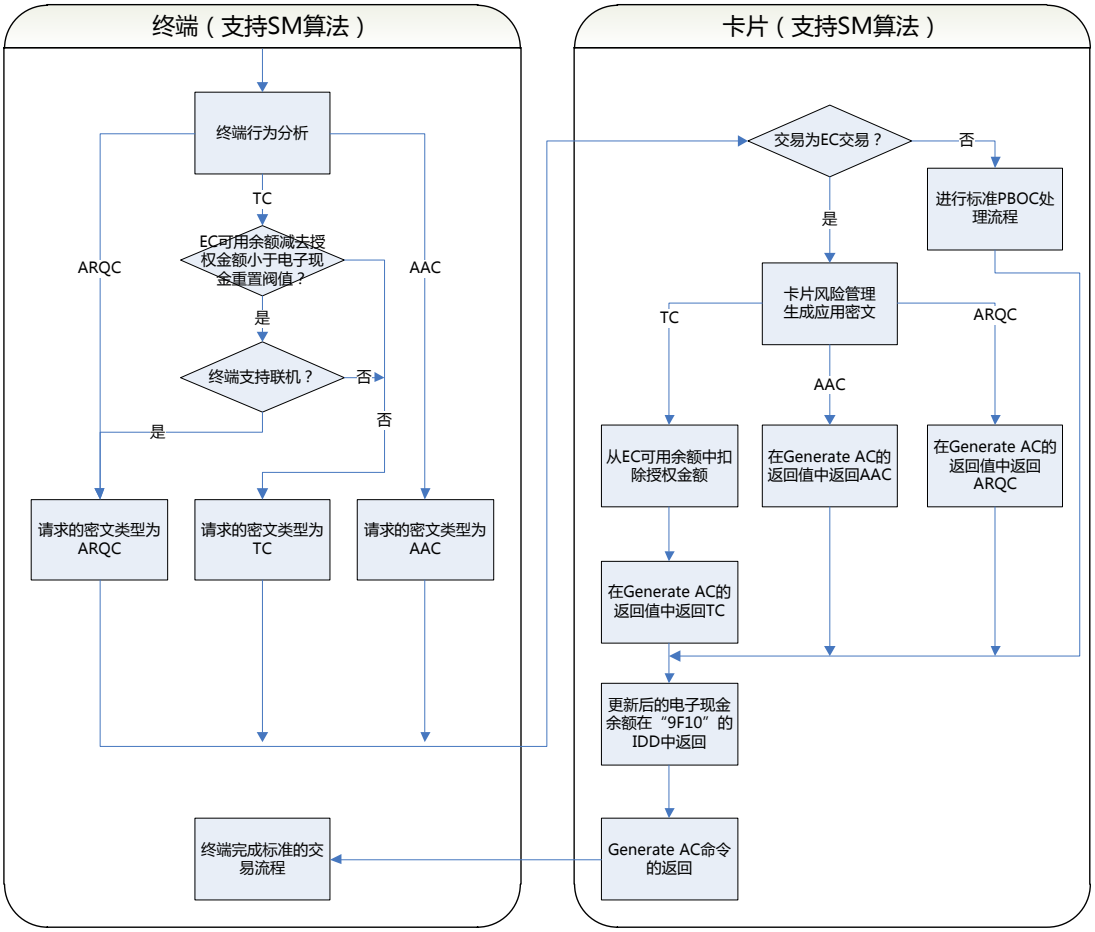


图 7(续) SM 算法应用基于借记/贷记应用的小额支付流程（续）

使用SM算法的小额支付流程与使用RSA/SHA-1/3DES算法的小额支付流程基本一致。主要区别在基于借记/贷记应用的小额支付流程中需要终端和卡片增加算法选择的步骤，这与SM算法应用标准借记/贷记相同，详见10.3.2和10.3.3条。脱机数据认证过程中，终端根据公钥索引检查算法类型；根据公钥证书中的“发卡行公钥算法标识”字段，再次检查算法类型。

10.5 qPBOC 应用流程

使用SM算法的qPBOC流程与使用RSA/SHA-1/3DES算法的qPBOC流程基本一致。主要区别在基于qPBOC支付流程中需要终端和卡片增加算法选择的步骤，并且卡片在收到GPO指令以后，需要根据终端发送的DF69进行判定，如果发现卡片不能支持终端要求的算法，那么卡片需要返回GPO指令的状态码为6985，从而实现脱机拒绝。

如果终端和卡片均支持并选择了SM算法进行交易处理，那么卡片需要返回采用SM算法计算TC、动态认证数据、SM算法的发卡行自定义数据（9F10）、SM算法对应的AFL等数据给终端，终端再进行数据读取以及完成fDDA认证操作。

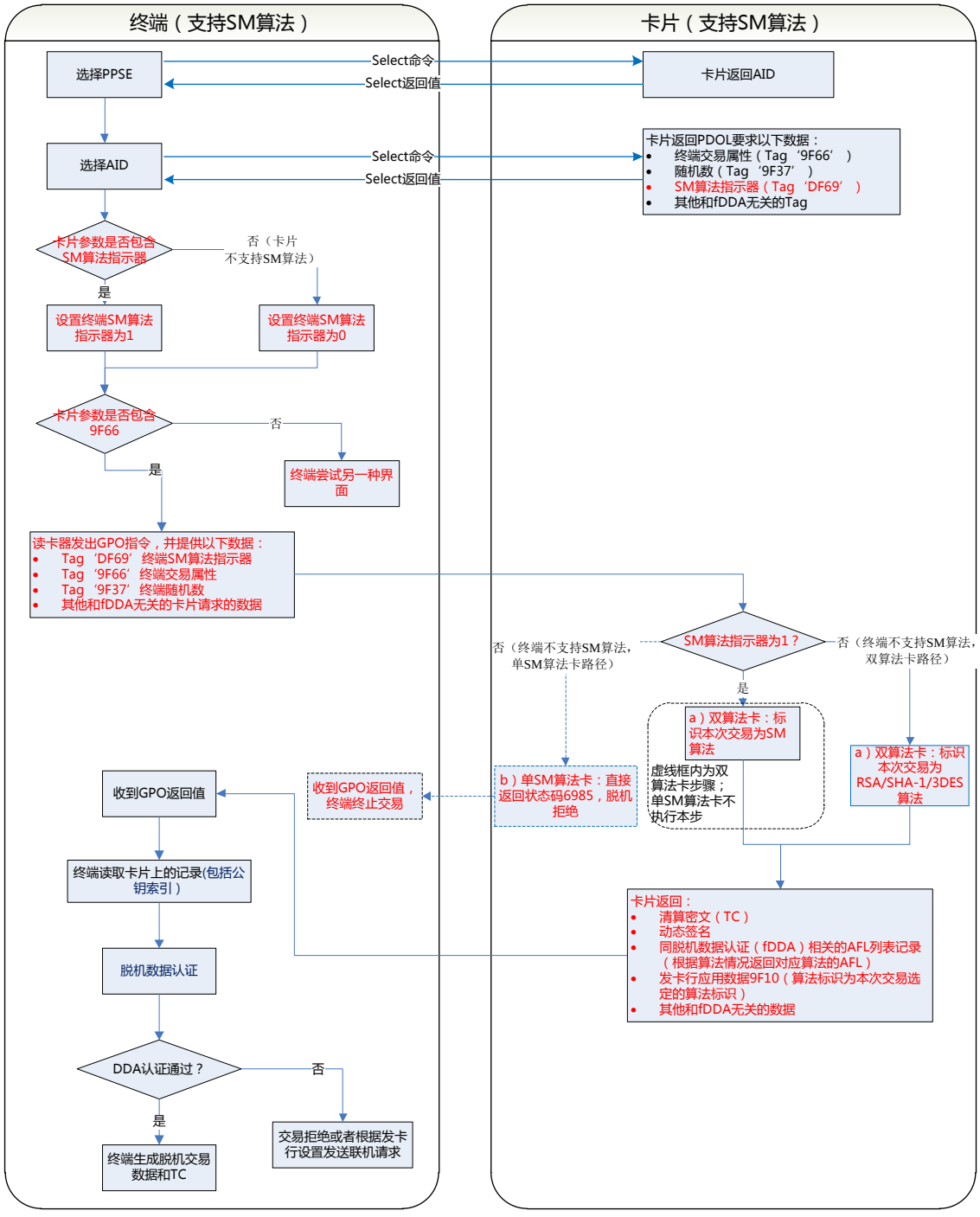


图8　SM 算法应用于 qPBOC 流程

10.6　个人化相关密钥的初始化

KMC 密钥的长度应为 16 字节。

个人化密钥（K_{ENC}、K_{MAC} 和 K_{DEK}）的产生采用 SM4 算法进行。

K_{ENC}：= SM4（KMC）[KEYDATA 的最右 6 个字节||‘F0’||‘01’|| KEYDATA 的最右 6 个字节 ||‘0F’||‘01’]。

K_{MAC}：= SM4（KMC）[KEYDATA 的最右 6 个字节||‘F0’||‘02’|| KEYDATA 的最右 6 个字节||‘0F’||‘02’]。

K_{DEK}：= SM4（KMC）[KEYDATA 的最右6个字节 ||‘F0’||‘03’|| KEYDATA 的最右6个字节||‘0F’||‘03’]。

11 PIN 修改/解锁命令数据计算方式

11.1 使用当前 PIN 修改 PIN 值

如果命令中的 P2 参数等于“01”，命令数据域包括 PIN 加密数据和 MAC，PIN 加密数据的产生过程按照下列步骤进行：

a) 发卡行确定用来给数据进行加密的安全报文加密主密钥,并分散生成卡片的安全报文加密子密钥：ENC UDK；

b) 生成过程密钥 Ks；

c) 生成 8 字节 PIN 数据块 D3：

1) 生成第一个 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	ENC UDK 的 5~8 字节			

2) 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）

3) D1 和 D2 执行异或得到 D3

d) 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

e) 将数据块 D3 和 D4 执行异或得到 D。

f) 用 Ks 对 D 进行加密（加密方法见本部分第 8.1.1 条），得到 PIN 加密数据。

11.2 不使用当前 PIN 修改 PIN 值

如果命令中的 P2 参数等于“02”，命令数据域包括 PIN 加密数据和 MAC，PIN 加密数据的产生过程按照下列步骤进行：

a) 发卡行确定用来给数据进行加密的安全报文加密主密钥,并分散生成卡片的安全报文加密子密钥：ENC UDK。

b) 生成过程密钥 Ks

c) 生成 8 字节 PIN 数据块 D：

1) 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	ENC UDK 的 5~8 字节			

2) 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）

3) D1 和 D2 执行异或得到 D。

d) 用 Ks 对 D 进行加密（加密方法见本部分第 8.1.1 条），得到 PIN 加密数据。

附 录 A
（规范性附录）
算法标识

A.1 公钥算法标识

表 A.1 列出了本部分使用的公钥签名算法标识。

表 A.1 公钥签名算法标识

公钥签名算法标识	签名算法	对应哈希算法
‘00’	无	无
‘01’	RSA	SHA-1
‘02’（EMV 规划使用）	ECC(ECDSA)	SHA-256
‘03’（EMV 规划使用）	ECC(ECDSA)	SHA-512
‘04’	SM2（数字签名算法）	SM3

表 A.2 列出了本部分使用的公钥加密算法标识。

表 A.2 公钥加密算法标识

公钥加密算法标识	加密算法	对应哈希算法
‘00’	无	无
‘01’	RSA	SHA-1
‘02’（EMV 规划使用）	ECC(ECIES)	SHA-256
‘03’（EMV 规划使用）	ECC(ECIES)	SHA-512
‘04’	SM2（公钥加密算法）	SM3

A.2 哈希算法标识

表 A.3 列出了本部分使用的哈希标识。

表 A.3 哈希算法标识

哈希算法标识	哈希算法
‘01’	SHA-1
‘03’（EMV 规划使用）	SHA-256
‘05’（EMV 规划使用）	SHA-512
‘07’	SM3

A.3 发卡行自定义数据（对称密钥算法标识）

发卡行自定义数据元中有一个 PBOC 自定义数据“算法标识”。此数据定义了卡片计算应用密文和安全报文采用的算法。长度为 1 个字节。其定义见见表 A.4。

表 A.4 对称算法标识

算法	值（16 进制）
3-DES	01

SM4	04

A.4 椭圆曲线标识

表 A.5 列出了椭圆曲线标识。

表 A.5 椭圆曲线标识

算法类型	强度	曲线	公钥长度	曲线标识
ECC	128 位	NIST P-256	64 字节	‘01’
ECC	256 位	NIST P-521	132 字节	‘02’
SM2	128 位	SM2 曲线	64 字节	‘11’

参考文献

[1] EMV支付系统集成电路卡规范：4.3，第1册～第4册
[2] VISA集成电路卡应用概述，1.4.0版
[3] VISA集成电路卡卡片规范，1.4.0版
[4] VISA集成电路卡终端规范，1.4.0版