# 实验报告

**班级：安全2101 姓名：张佳淇 学号：21281086**

## 实验题目

带有安全认证机制的钱包交易

实验内容：1、 要求钱包交易具有内、外部认证功能

2、 要求钱包需对加减钱的数据进行MAC校验。

3、 编写JCOP脚本对编写的Applet进行测试。

## 应用APDU命令

**APDU命令格式：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 内部认证 | 外部认证 | MAC校验 | **DEBIT命令：** | **GET\_BALANCE命令：** | **CREDIT命令：** |
| CLA: 0x80 | CLA: 0x80 | CLA: 0x80 | CLA: 0x80 | CLA: 0x80 | CLA: 0x80 |
| INS: 0x84 | INS: 0x82 | INS: 0x30 | INS: 0x40 | INS: 0x50 | INS: 0x50 |
| P1: 0x00 | P1: 0x00 | P1: 0x00 | P1: 0x00 | P1: 0x00 | P1: 0x00 |
| P2: 0x00 | P2: 0x00 | P2: 0x00 | P2: 0x00 | P2: 0x00 | P2: 0x00 |
| Lc: 08，预期返回的数据长度 | Lc: 08，预期返回的数据长度 | Lc: 0x0F，数据域长度，此处为15字节，包括：1字节充值金额，6字节AppletID, 8字节MAC | Lc: 0x0F，数据域长度，此处为15字节，包括：1字节消费金额，6字节AppletID, 8字节MAC | Le: 此处为0，不需要发送数据 | Lc: 0x0F，数据域长度，此处为15字节，包括：1字节消费金额，6字节AppletID, 8字节MAC |
| Data: 8字节已加密数据 | Data: 8字节已加密数据 | Data: 充值金额，appletID，MAC | Data: 消费金额，appletID，MAC |  | Data: 充值金额，appletID，MAC |

**返回数据说明：**

内部认证返回：0x08字节加密结果 卡片使用内部密钥加密输入数据后返回的结果，终端可以使用相同的密钥和数据进行验证。

外部认证：外部认证成功不返回数据，失败则会返回错误码。

MAC校验：MAC校验成功不返回数据，失败则会返回错误码。

DEBIT命令：消费成功不返回数据，失败则会返回错误码。

GET\_BALANCE命令返回：0x02字节余额 返回卡片余额，格式为大端序，例如：返回 \x00\x10 表示余额为 16 (0x10)

CREDIT命令：充值成功不返回数据，失败则会返回错误码。

## 程序设计说明

在该电子钱包应用中， Purse 负责处理用户的存款、消费请求，并调用积分应用 JavaLoyalty 进行积分管理。CyclicFile 类用于存储交易记录，JavaLoyaltyInterface 接口定义了电子钱包和积分应用之间的通信方式。

**1.主类：**

Purse.java 定义了电子钱包应用的主类 Purse，它继承自 Java Card 的 Applet 类，是整个应用的核心。

**2.功能方法：**

Purse 类

|  |  |
| --- | --- |
| 方法名称 | 功能描述 |
| install | 静态方法，用于安装电子钱包应用 |
| select | 当电子钱包应用被选中时调用 |
| deselect | 当电子钱包应用被取消选中时调用 |
| process | 处理传入的 APDU 命令 |
| credit | 存款功能 |
| debit | 消费功能 |
| debitPoint | 消费积分功能 |
| getBalance | 获取余额 |
| getRecord | 获取交易记录 |
| verify | 验证 PIN 码 |
| indoAuthentication | 内部认证 |
| outdoAuthentication | 外部认证 |
| getRandom | 获取随机数 |
| GenerateSignature | 生成MAC签名 |
| VerifySignature | 验证MAC签名 |
| grantPoints | 调用积分应用进行积分增减 |

CyclicFile 类：

|  |  |
| --- | --- |
| 方法名称 | 功能描述 |
| ReadRecord | 读取指定位置的记录 |
| AppendRecord | 添加新记录，循环存储 |

JavaLoyalty 类

|  |  |
| --- | --- |
| 方法名称 | 功能描述 |
| install | 静态方法，用于安装积分应用 |
| getShareableInterfaceObject | 获取共享接口对象 |
| grantPoints | 根据交易类型增加或减少积分 |
| process | 处理传入的 APDU 命令 |
| processReadBalance | 处理读取积分余额的命令 |
| processResetBalance | 处理重置积分余额的命令 |

JavaLoyaltyInterface 接口

|  |  |
| --- | --- |
| 方法名称 | 功能描述 |
| grantPoints | 定义积分增减方法 |

**3.数据结构：**

Purse 类:

pin: PIN 码

balance: 余额

record: 交易记录 (使用 CyclicFile 类实现)

indeskey, inCipherObj: 内部认证相关

outdeskey, outCipherObj: 外部认证相关

Random: 随机数数组

mackey, sig: MAC 签名相关

CyclicFile 类:

record: 存放所有记录的字节数组

maxrecord: 最大记录数量

recordsize: 每条记录的长度

currentrecord: 当前记录位置指针

buffer: 临时存储读取记录的缓冲区

JavaLoyalty 类:

balance : 积分余额

**4.处理流程：**

4.1 电子钱包应用 (Purse)

·接收外部 APDU 命令。根据 CLA 和 INS 字节判断命令类型。进行 PIN 码验证 (如果需要)。

·执行相应的功能：

存款/消费: 更新余额、记录交易信息。

消费积分: 调用积分应用 JavaLoyalty 进行积分增减。

获取余额/记录: 读取并返回相应数据。

安全相关: 执行加密、签名、认证等操作。

·返回响应 APDU。

4.2 积分应用 (JavaLoyalty)

·接收电子钱包应用的调用请求。

·根据请求类型执行积分增减或查询操作。

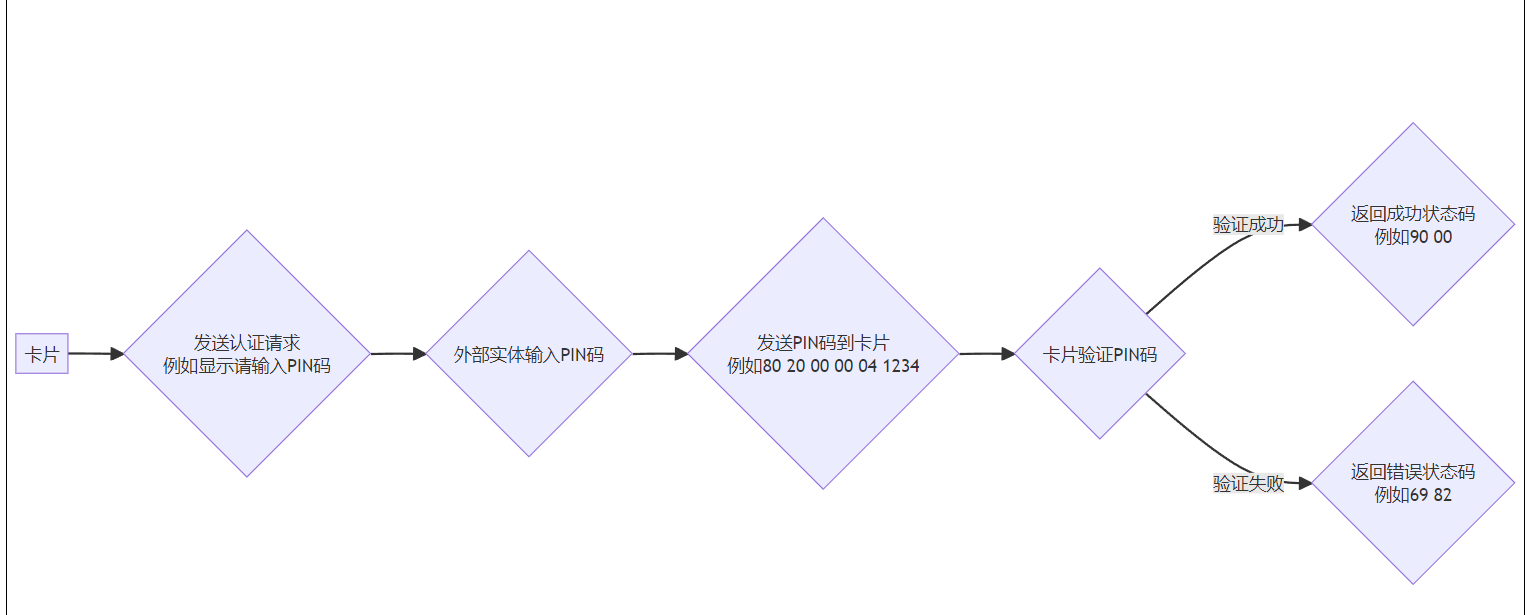
·返回操作结果给电子钱包应用。

4.3 循环记录 (CyclicFile)

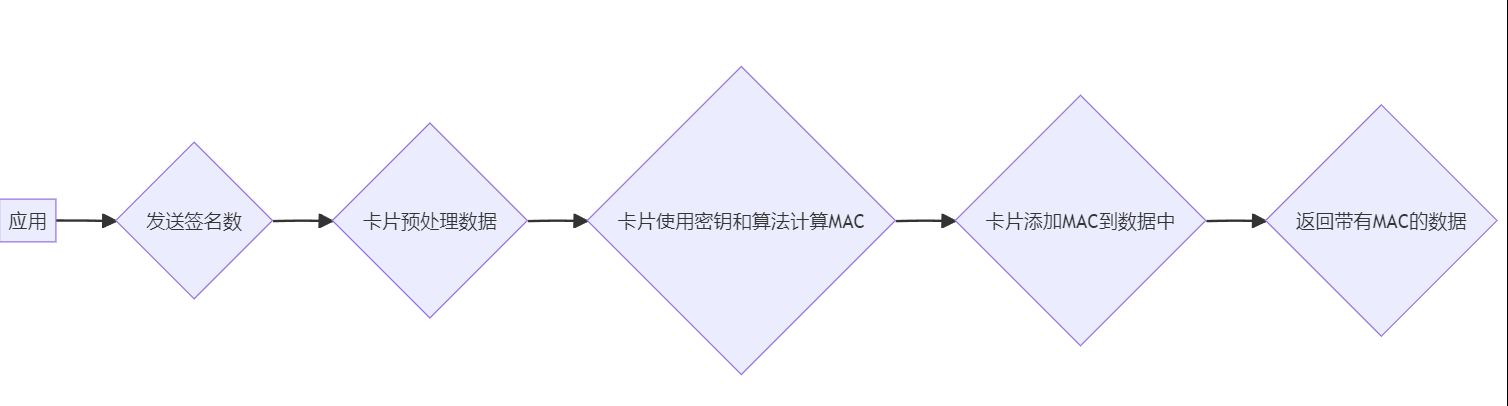
·AppendRecord: 在数组末尾追加新记录，如果数组已满，则从头开始覆盖。

·ReadRecord: 根据指定位置读取记录

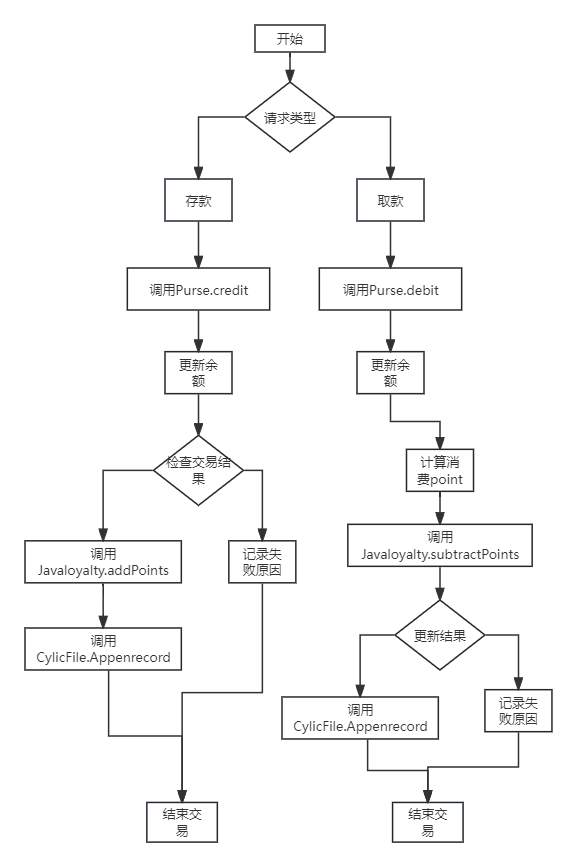
外部认证：



MAC签名：



具体钱包操作：



## 测试APDU命令集

1.选择钱包应用程序 /select 112233445503：确保应用程序能够正确被选择。

2.验证PIN：/send 8020000006010203040506 ：

3.获取随机数/send 8080000000，返回DATA：A8 DC 24 D9 3A BF 97 95 90 00，其中的A8 DC 24 D9 3A BF 97 95为生成的随机数，将随机数通过des中的ecb方法加密，秘钥为0102030405060708，可以获得对应的加密结果。

4.外部认证：/send 8082000008C091E93BBAE9891A，添加上面的加密结果后进行认证

5.内部认证，返回加密值/send 8084000008192450D0AE2D5F7F

6.加钱0x64的MAC认证：/send 80700000086430112233445503，获得返回随机数37 A2 B2 00 5B 3C DF F6

7. 减钱0x32的MAC认证：/send 80700000083240112233445503，获得随机数0B CC F5 C7 58 5A 2D EA

8.加钱0x64：/send 80300000096437A2B2005B3CDFF6

9.查余额：/send 8050000002，返回值为0094

10.减钱0x32：/send 8040000009320BCCF5C7585A2DEA

11.查余额：/send 8050000002，返回值为0032

## 测试运行结果



