新锁—手持功能说明

变更历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **日期** | **版本** | **修改内容** | **修改人** | **备注** |
| 2016/4/17 | V1.0 | 创建 | 周亚平 |  |
|  |  |  |  |  |
| 2016/6/15 | V2.0 | 增加存储标识，增加F5标志位，增加出入库，查询袋状态协议。  通讯协议和功能说明分开记录 | 周亚平 |  |
| 2016/9/27 | V3.0 | 修改新的封袋流程，增加内存分配。 | 周亚平 |  |
| 20170714 | 4.0 | 封袋添加写入流水号、调整流水号索引保到0x10.1中 | 黄柄癸 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

### 芯片定义

1. 定义13.56芯片:

13.56简称NFC卡，其唯一的卡号简称UID。

2. 定义900M芯片:

900M简称UHF超高频卡，其唯一的卡号简称TID

### 袋结构

空袋:每一个空袋，包含一个NFC卡和一个UHF芯片。

封签过的袋:每一个袋，包含一个NFC卡和两个UHF,其中袋中的UHF已经不能再被读写，只有标签上的UHF才可被读写，当批量扫描或者出入库时，扫描的就是外面的UHF。

### 袋ID内容组成

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 版本 | 地区码 | 残损标志 | 袋型号 | UID | 校验位 |
| 2位 | 3位 | 1位 | 2位 | 14位 | 2位 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 版本 | 地区码 | 残损标志 | 袋型号 | UID | 校验位 |
| 1byte | 1.5byte | 0.5byte | 1byte | 7byte | 1byte |

将ID组成24位的字符串数字，按照相邻两位转换为16进制的形式进行压缩，初始化的时候写入到NFC卡和UHF（指的是袋中那个UHF，在还没有被封签时，他是可以正常读取的）中。

注:校验和是前11个字节的异或和。

### 袋ID初始化过程

1.需要将版本信息，地区码，残损标志，袋型号，UID,校验位，共24位数据分别写入到NFC卡的用户地址（0x04-0x06），和UHF的EPC区中。其中NFC卡的用户地址是0x04-0xE1，每一个地址存储4个字节，共可读写888个字节，UHF的EPC区的存储大小刚好12个字节，且起始地址是0x02。

2.将标志位F1写入NFC卡的特定位置(0x10)。

3.初写目录索引等信息到NFC卡0x20中，详细地址见下文。

4.在写入基本信息成功后，会将袋ID数据发送到前置服务器。前置服务器判定是否初始化成功，主要是指有没有重复初始化。发送给前置服务器的主要目的是记录这些ID，便于后续袋管理。

上行:$ib 0553200680441E32CC5904F8 000#

注：05 版本号

532 地区码，532青岛

1. 完整 1 残损

01 袋型号2位 用阿拉伯数字0-9表示

分为100-1元 以代号表示

01 = 100元

02 = 1元

03 = 50元

04 = 20元

05 = 10元

06 = 5元

07 = 2元

08 = 0.5元

09 = 0.2元

10 = 0.1元

80441E32CC5904 NFC卡的UID号。

F8 : 校验和

000 包号，任务编号。

下行: \*ib Ok

Ok 插入成功

1. 插入失败，除去重复的情况外，其他因素引起的失败。
2. 重复初始化。

### 存储数据加密

在往NFC卡初始化时，数据采用NFC卡的UID号与数据进行异或后再错位的方式进行加密。因数据有12个字节，但UID只有7个字节，所以需要将UID（7个字节）+截取UID(后5个字节)拼成12字节后再进行异或处理，最后再将异或后数据前后三位调位处理。

在往UHF中初始化时，数据则采用UHF的TID号与数据进行异或再前后三位调位处理的方式进行加密。同时，也需要把TID进行拼接处理后，才能异或，和错位处理。

如果：进行异或加密后的数据长度如果小于6，需特殊处理。

Len = 0 或者NULL 时，直接返回NULL

Len =1 不变

Len =2 ,3 中间不变，交换一个

Len =4,5 交换两位,长度为5的时候，中间不变

其余：交换前后三位。

例如：

未加密的数据:0502800680441E32CC5904F8 密钥:0459CC321E4480（UID）

加密后的数据:015B95349E009E36954C36E6

解密后的数据: 0502800680441E32CC5904F8

# 标志位加密

1. 读取NFC卡的厂家UID卡号（7个字节）。
2. 生成随机值（0-9）。
3. 通过随机值生成一个密钥，目前共有十套密钥。
4. 通过判断UID[1] %10的值，选择哪一套数值代表F1🡪F4，目前也有十套数值。
5. 将第四步得到的值（包含着F1🡪F4）与 密钥进行异或
6. 将当前的密钥序号写在0x14位置。如：A8 其中高字节A代表版本1
7. 将第五步得到的值写在0x10位置。

# 标志位解密

1. 读取NFC卡的厂家UID卡号（7个字节）。
2. 计算UID[1] %10的值选择当前代表F1🡪F4是哪一套。
3. 读取0x14地址当前存储的密钥编号，选择对应密钥算法生成密钥值。
4. 将第二步得到的值（包含着F1🡪F4）与 密钥值进行异或，还原。

附十种密钥方式：

A0：UID[1] + UID[3] + UID[4]

A1：0x0 ^ UID[1] ^ UID[3] ^ UID[5]

A2：0x0 ^ UID[4] ^ UID[5] ^ UID[6]

A3：UID[1] + UID[2] + UID[3]

A4：UID[6] + 1

A5：UID[1] + UID[4] + UID[6]

A6：(UID[1] ^ UID[3] ) + UID[5]

A7：( UID[4] + UID[6] ) ^ UID[2]

A8：UID[2] + UID[4]

A9：( UID[3] + UID[5] ) ^ UID[4]

附十种代表F1🡪F4的方式：

1. 23 5f 8e 41
2. 8c 3d 6a 23
3. 95 3c 4b 11
4. 1c 3e 22 49
5. 36 47 88 26
6. 28 3d 6f 78
7. 99 2d 6c 24
8. 8c 7f 9d 36
9. 25 39 48 76
10. 2c 36 47 79

### 存储区域划分

1. NFC卡初始化时，将ID内容(12个byte)从起始0x04地址开始写入，共占用3个地址（12个byte），即NFC卡初始化的内容数据地址为（0x04—0x07）。
2. UHF初始化时，将ID内容(12个byte)从EPC区的起始0x02地址写入，共占用6个地址（12个byte），即UHF卡初始化的内容数据地址为EPC区。
3. 标志位F1的地址为0x10。
4. 启用码存放地址0x11—0x11(FFDDFFEE已启用，EEEEEEEE注销 00000000未启用)。
5. 目录存放在0x20—0x20。
6. 封签信息存放在0x30—0x3f。
7. 交接信息存放在0x40—0x8f。
8. UHF中存放封签事件码从User区的0x00开始存放。

注：3-7点都是针对NFC卡。

对于增加交接信息时，需要写读出当前目录中交接信息的开始地址，长度等，再通过计算后得到新的开始地址。

NFC卡内存规划具体如下表所示:

（每个地址长4个byte，用0x9.n-表示地址9的第n个字节，n=0，1，2，3）

|  |  |
| --- | --- |
| 内存地址 | 存放的内容 |
| 0x04—0x06 | 版本信息，地区码，残损标志，袋型号，TID，校验位，共24位 |
| 0x07—0x08 | 封签时，保存锁片的TID， 6个byte：0x07.0-0x08.1 |
| 0x09—0x0f | 暂时保留 |
| 0x10—0x10 | 10.0标志位 10.1交接信息索引 |
| 0x11—0x11 | 启用码 FFDDFFEE已启用，EEEEEEEE注销 000000尚未启用 |
| 0x12—0x12 | 测试和正常模式切换，FF测试，非FF 正常 |
| 0x13—0x13 | 上电次数（单片机写入） |
| 0x14—0x14 | 密钥选择（单片机写入）默认A0 |
| 0x15—0x15 | 上电时间（单片机写入） |
| 0x16—0x16 | 非法时上电时间记录备份（单片机写入） |
| 0x17—0x17 | 报警记录（单片机写入） 非法，电压低，上电超时，报警电压 |
| 0x18—0x18 | 记录锁，空袋的时候，超高频距离 |
| 0x19—0x1F | 暂时保留 |
| 0x20--0x20 | 暂时保留(原交接信息索引) |
| 0x23—0x2f | 暂时保留 |
| 0x30—0x3f | 封签信息(条码，袋码) 30byte |
| 0x40—0x8f | 交接信息 12byte |
| 0x90--0x92 | 封袋流水号 11byte |
| 0x93—0xe0 | 暂时保留 |
| 0xe1—0xe1 | 正在报警标记 0没有在报警 f1 f2 f3对应非法，电压低，上电超时 |

UHF卡内存规划如下表所示:

|  |  |  |
| --- | --- | --- |
| EPC | TID | USER |
| 0x02 地址开始12个字节的袋id内容（可读写） | 0x03 地址开始 6个字节的tid号（只读） | 0x0地址开始28字节的封签事件码（可读写） |

### 验空袋，启用空袋，注销空袋

首先，检测袋子是否已经初始化，通过读取NFC卡的袋信息，判别袋子是否已经初始化。

然后，检测袋子是否已经启用（袋子默认不启用），如果没有启用，发送袋id到前置服务器，判定是否给与启用，如果返回启用，修改0x11位置的启用码，标识已经启用，否则标识不启用。

其中，本地已启用的标识为FFDDFFEE，已注销EEEEEEEE 00000000初始化后尚未启用。

### 过程标志位变化

在袋ID初始时，会将地址为0x10的标志位初始为F1。再当插上锁片后，单片机会将标志位从F1🡪F2。

当手持进行封签时，会检测标志位是否为F2，然而确定锁片有没有插上。

手持正常封签后，会将标志位从F2🡪F3。

当手持进行开袋操作时，会检测标志位是否为F3，以确定有没有正确封袋，进而接下来开袋操作，

如果此时检测到电压为低于2.9V,会将标志位从F3🡪F5.继而手持会有相关返修提醒，

此时检测到电压为高于2.9V,会将标志位从F3🡪F4.

以下为单片机操作：

开袋完成后，拔出锁标签瞬间，单片机会重新上电，检测当前标志位是否为F4，

如果是F3，表示非法开袋，从而报警，闪烁红灯，以及蜂鸣器响。

如果此时是F4,则表示正常开袋，然后单片机将标志位F4🡪F1，循环使用。

如果此时为F5，表示该袋不可再使用。

### 封袋

封袋是指将带有UHF的标签插入到袋锁上。

在正常封袋之前需要检测如下东西：

0. 手持会根据任务列表与批信息，确定当前封袋的是否为上一个未封签完批，如果非上一次结束的批，则会弹出完整，残损，清分，未清分，复点，已复点等袋标识信息（1个字节）。

1. 袋子是否初始化，如果没有，显示没有初始化，提示先初始化。

2. 空袋是否已经启用（先本地验证，然后前置服务器验证），如果没有，显示该袋尚未启用，提示请先启用（目前服务器的操作时全部给与启用）。

3. 检测锁片是否已经插上，如果没有，显示尚未插上锁片，请插上锁片再操作。

4. 向服务器发送袋ID,标签的tid,前置确定是否可以正常封签，如何可以，返回封签事件码，时间和操作人员的编号，以及网点号。如果，失败，返回错误信息，或者重复封签操作。

5. 向袋写入封签事件码信息，以及交接信息和目录索引信息。然后，复制袋ID信息到标签中，以便进行门天线操作。如果这两点成功，则提示封签成功。否则，向服务器发送，写数据失败的命令，告知服务器写入数据失败，以便删除刚显示的封签信息。

注：是先插锁片，然后再进行封签操作。在插上锁片的瞬间，标志位会被更改。

袋标识 11 完整已清分

12 完整未清分

23残损已复点

24 残损未复点

封签的通讯协议如下：

第一步，验证空袋是否启用，见上文的启用中查询协议。

第二步，查询袋目前正处于的状态，以便确定是否可以进行封签。

第三步，发送袋ID和标签的TID到前置。

第四步，向NFC写入封签事件码，目录交接信息，以及向标签中写入袋ID

第五步，（如果需要）当第四步操作失败后，需向前置发送删除命令。

### 入库

首先从服务器查询当前袋的状态，判断是否处于已封签状态。

手持需要根据查询到的状态去判断当前的操作是否可做。

当该袋处于已封签的状态时，说明接下来是可以进行入库操作的,入库协议如下:

### 出库

首先从服务器查询当前袋的状态，判断是否处于已入库状态。

手持需要根据查询到的状态去判断当前的操作是否可做。

当该袋处于已入库的状态时，说明接下来是可以进行出库操作的,出库协议如下:

### 开袋

在正常开袋之前需要检测如下东西：

1.检测是否已经完成封签。（无需检测初始化，启用等信息，当封签完好时，表明其余就完好。）

如果以上封签检测通过后即可进行开袋操作:

1.手持改写标志位，以及更新目录索引信息，以及交接信息，

2.手持向服务器发送封签事件码，封签码，以及袋Id到服务器，服务器验证是否可以正常开袋。

附：操作流程

一、袋id内容初始化应用

1. 打开桌面红色的袋id初始化程序，进行袋id初始化。
2. 进入程序后，设置接收的服务器ip,此时需要打开电脑上的袋ID初始化程序（不是之前的前置，且两者不能同时打开）。
3. 点击确定后，进入初始化界面，请选择地区，完整，残损，袋型号。
4. 点击开始，扫描标签（指的是超高频），袋锁片（NFC卡）
5. 写入标签，就是往UHF中写入袋ID
6. 更新袋锁，指的是更新初始化信息到NFC卡中。

注意：目前NFC，UHF读卡距离都比较近，请注意距离。

二、手持基金应用。

封袋：

1.新初始化的基金锁是默认没有启用的，所以首先得在主界面的空袋管理点击启用空袋中启用空袋（目前仅仅是在本地，前置尚未实现接收）。启用空袋，扫描的是NFC卡。

2.考虑到在青岛，已跳过扫捆。

3.进入封袋页面，锁定批按钮初始化已经禁用。

4.如果不是上一个批，详细见上文描述，弹出袋标识选择，默认是11完整已清分。或者可以点击键盘删除按键，手动弹出选择框。

5.袋码，读取nfc卡的uid,以及存在的袋ID内容，标志位，是否启用等信息。

6.标签码，读取标签的UHF（在插上锁片的情况下，袋中的UHF是不能被得到的，这一点因为硬件的原因，我还没测试过）中的TID号。

7.信息校验，主要是发送封签请求给前置，判断是否可以正常封签。这个过程很快，得到需写入的封签事件码信息。

8.更新信息，主要是更改标志位，目录，索引等信息，以及封签事件码信息到NFC卡。以及将封签事件码写入到UHF的USER区。该过程耗时较久，得注意锁片与读卡器的距离。

10，写入成功后，显示扫描数量。如果失败，会再次向服务器发送失败命令。

开袋：

基本和封签一致，但是没有封签请求，总过程只会和服务器交互一次，即最后显示的信息校验。

新锁5通讯协议

## *版本*

5.0

编辑：周亚平

日期：2016-06-22

## *协议基础*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Head | Type | Contents | PacketNO | CS | End |
| 头标识 | 数据类型 | 数据内容 | 包号 | 异或校验 | 结束符 |

主要格式包头（head），数据类型（type），数据区（data），包号（packetNO），校验（cs），结束符（end）。采用字符串的形式发送。

手持端：

发送格式：开始$ 结束# $xxxx#

接收格式：开始\* 结束# \*xxxx#

服务器端和手持端发送和接收格式刚好相反。

## *登录*

S: $01 92EDE26A 000000 000000 00000000000000000 004#

01 : 功能标识，登录功能

92EDE26A : 卡号，用户IC卡卡号。

000000 : 密码

000000 00000000000000000 : 固定格式，未知。

R : \*01 01 1111114A 160623143711 087101001 006 004#

01 : 登录成功。

1111114A : 权限标识、

160622143711 : 时间2016-06-22 14:37:11

087101001 : 当前机构号

006 : 该卡在这个机构中序号，即用户编号

## *IC卡复核*

S: $00 02 92EDE26A 000000 000000 00000000000000000 003#

00 : 功能标识，复核功能

02 : 代表IC卡复核，01 是指纹复核，见文末指纹协议

92EDE26A : 卡号

000000 : 密码

000000 00000000000000000 : 固定格式，未知。

R: \*00 01 003#

01 : 复核通过 00 复核不通过

## *袋ID初始化*

S:$ib 0502800180441E32CC5904F8 000#

Ib 开始初始化

05 版本号

1. 地区码，028青岛

0 完整 1 残损

01 袋型号2位 用阿拉伯数字0-9表示

分为100-1元 以代号表示

01 = 100元

02 = 50元

03 = 20元

04 = 10元

05 = 5元

06 = 1元

80441E32CC5904 NFC卡的UID号。

000 包号，任务编号。

R: \*ib Ok

Ok 插入成功

1. 插入失败，除去重复的情况外，其他因素引起的失败。
2. 重复初始化。

## *袋启用*

S: $22 80441E32CC5904 30 0502800680441E32CC5904F8 001#

$22 包头

80441E32CC5904 用户ID，登录用户的卡号

30 启用的命令

0502800680441E32CC5904F8 袋id

R: \*22 30 Ok

Ok 表示给予启用

00 启用失败，不给予启用

01 已启用

## *袋注销*

S: $22 80441E32CC5904 31 0502800680441E32CC5904F8 001#

$22 包头

80441E32CC5904 用户ID，登录用户的卡号

31 注销的命令

0502800680441E32CC5904F8 袋id

R: \*22 31 Ok

Ok 表示注销成功

00 注销失败，可能是尚未开袋，或者权限不够之类的原因。

01 已注销

## *查询袋是否启用*

S：$22 80441E32CC5904 32 0502800680441E32CC5904F8 001#

$22 包头

80441E32CC5904 用户ID，登录用户的卡号（可以不需要）

32 查询的命令

0502800680441E32CC5904F8 袋id

R: \*22 32 00

00 已启用

01 已注销

02 未启用

## *封签（出入库，开袋）任务查询*

S: $07 92EDE26A 01 00000000000000000 002#

07 : 功能号，代表查询任务功能。

92EDE26A : 用户IC卡号。

01： 区别是查询封签，出入库，还是开袋。

01：查询当前批，封袋批查询

02：旧版入库、新版为82

03：旧版出库、新版为83

04：验封批查询

06：开袋批查询

00000000000000000 : 批编号，在这个协议，该值是固定的。

R1 : \*18 201608710100100035 809402390900900075 002#

18 : 表示没有封签任务。

201608710100100035 ： 没有任务时，该值是批编号。

809402390900900075

R2 : \*08 01 087101001BBBB000511 008#

08 : 有任务。

01: 封签任务。

087101001BBBB000511 : 银行列表任务，具体请参考银行列表解析。如果有多个任务时，会有多个值。

## *选择任务*

S : $17 92EDE26A 01 1 000000000000000000 009#

17 : 选择任务，功能号。

92EDE26A : 用户id卡号。

01 ：类别，封签任务。02 入库 03出库 04验封 05换袋 06 开袋

1 : 选择的是第一个任务。

R : \*18 201608710100100037 809402390900900073 010#

201608710100100037 : 批编号。

## *袋状态查询*

S: $22 2A4CAE68 20 055311060463C7321E448009 4354FEA7001F 201608710100100024 005#

20:查询标志

055311060463C7321E448009： 袋ID

R: \*22 30 01 087101001 160615162751 001 005#

01 : 已封签

如果是 00 可以封签

01已封签

02已入库

03已出库

06已开袋。

## *封签请求*

S： $22 80441E32 20 0502800680441E32CC5904F8 21498EB10009 11 201608710100100129 004#

80441E32：用户ID，登录用户的卡号

20： 请求封签命令符。

0502800680441E32CC5904F8： 袋id内容

21498EB10009： 标签TID号

11： 完整已清分，袋标识

201608710100100129： 批号

R：\*22 3001087101001160419134631001 004#

30：表示封签

01：请求成功 00失败

087101001 ：机构号

160419134631：时间，16年04月19日13点46分31秒

001 登录用户，机构编号的后三位

## *封签*

S：$22 80441E32 21 0502800680441E32CC5904F8 21498EB10009 201608710100100129 003#

21 ： 发送封签事件码的命令。

R：\*22 22000100019999 050271010476C9321E448024215B2DD3000B087101001002160927114754 087101001002160927114754 2016092708710200100011 005#

0001：个人扫描

0001：完成扫描

9999： 计划扫描 9999表示尚未确定计划数量

红色+087101001002+160927114754：封签事件码

087101001002 ： 网点号+工号

160927114754：时间

087101001002160927114754 ：服务器存储该数据的流水号

## *取消封签*

S： $22 80441E32 40 2016092708710200100013 201608710100100129 004#

80441E32：用户ID，登录用户的卡号

40： 请求封签命令符。

2016092708710200100013：服务器存储该数据的流水号

201608710100100129： 批号

## *锁定批*

S：$27 92EDE26A 00 0 201608710100100037 011#

27 : 功能标识，锁定批功能。

92EDE26A ：用户id卡，卡号。

00 : 固定值。

0 : 固定值。

201608710100100037 ： 批编号。

R: \*27 03 011#

27 : 批信息回复标识。

03: 未达到指定数量。详情查询错误文件。01 锁定批成功，02 该批已经锁定。

### 入库

S： $22 2A4CAE68 22 055311060463C7321E448009 4354FEA7001F 055311060463C7321E4480094354FEA7001F2A4CAE68160615155547 201608710100100024 006#

22：表示入库指令

055311060463C7321E448009： 袋ID

4354FEA7001F: 标签的TID

055311060463C7321E4480094354FEA7001F2A4CAE68160615155547

封签事件码

### 出库

S： $22 2A4CAE68 23 055311060463C7321E448009 4354FEA7001F 055311060463C7321E4480094354FEA7001F2A4CAE68160615155547 201608710100100024 006#

23：表示出库指令

055311060463C7321E448009： 袋ID

4354FEA7001F: 标签的TID

055311060463C7321E4480094354FEA7001F2A4CAE68160615155547 封签事件码

### 预扫描开始

S : $103 309EC92B 02 0 201608710100100034 004#

$103 ：预扫描开始，功能标识

309EC92B ：用户IC卡号

02 ：入库预扫描 03 出库预扫描

0 ：预扫描数量

201608710100100034 ：批编号

R : \*103 00 004#

00 : 成功

### 预扫描停止

S : $104 309EC92B 02 0 201608710100100034 004#

$ 104 ：预扫描停止，功能标识

309EC92B ：用户IC卡号

02 ：入库预扫描 03 出库预扫描

0 ：预扫描数量

201608710100100034 ：批编号

R : \* 104 00 004#

00 : 成功

### 预扫描结束

S : $105 309EC92B 02 0 201608710100100034 004#

$ 105 ：预扫描结束，功能标识

309EC92B ：用户IC卡号

02 ：入库预扫描 03 出库预扫描

0 ：预扫描数量

201608710100100034 ：批编号

R : \* 105 00 004#

00 : 成功

### 交接开始

S : $106 309EC92B 02 0 201608710100100034 004#

$106 ：交接开始，功能标识

309EC92B ：用户IC卡号

02 ：入库预扫描 03 出库预扫描

0 ：交接扫描数量

201608710100100034 ：批编号

R : \*106 00 004#

00 : 成功

### 交接停止

S : $107 309EC92B 02 0 201608710100100034 004#

$ 107 ：交接停止，功能标识

309EC92B ：用户IC卡号

02 ：入库预扫描 03 出库预扫描

0 ：交接扫描数量

201608710100100034 ：批编号

R : \* 107 00 004#

00 : 成功

### 查询天线柜是否准备

S : $108 309EC92B 01 004#

$ 108 ： 查询天线柜是否准备，功能标识

309EC92B ：用户IC卡号

01 ：标识 待后续使用

R : \* 47 01 004#

01 : 成功 00 未准备

### 开袋

S： $22 80441E32 26 0502800680441E32CC5904F8 21498EB10009 0502800680441E32CC5904F821498EB10009160419141003 201608710100100130 002#

80441E32 用户ID，登录用户的卡号

26 开袋命令号

0502800680441E32CC5904F8 袋ID内容

21498EB10009 标签的TID号

R：\*22 26000100020005 003#

0001：个人扫描

0001：完成扫描

0005： 计划扫描 9999表示尚未确定计划数量

# 查询银行列表

S: $08 92EDE26A 00 003#

08: 代表功能号，即查询银行列表功能。

00：不知，可能是获取所有吧？ //20160622

R: \*{"002701001":"武汉人行","002704001":"武汉工行","002706001":"武汉工行","002801000":"分行科技处","002801001":"人行营管部清分科","002801002":"人行营管部库一科","002801003":"人行营管部库务二科","002802001":"建行中心金库","002803001":"农行光华支行","002804001":"工行成都分行","071101001":"鄂州人行","071104001":"鄂州工行","087101001":"kmrh","087102001":"kmjh"} 003#

绿色即非标准的json格式的字符串。

E: \*02 005# 电脑处理数据异常

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*指纹\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# 查询是否需要更新

S: $14 00 1 000#

00 ： 代表功能号，即查询更新功能。

1:当前客户端存放的版本 当两个绿颜色值相同，则不需更新，接收如E所示。否则如R所示。

R: \*14 00 [{"UserID":"087101001002","Version":"1"}] 1 000#

[{"UserID":"087101001002","Version":"1"}]：服务器查询到的已录指纹所有用户的最高版本。

1：服务器当前的最高版本。

E: \*14 00 00 000# //服务器和客户端版本一致。

# 指纹注册

S: $14 01 002 1 0 \*\*\*（512字节指纹数据） 006#

02 ：代表功能号，即注册功能。

002 ：用户userid 的后三位

1 ： 一个用户同时需注册三个指纹，1表示序号，当前是第一个。

0 ： 版本号。

\*\*\*（512字节指纹数据） ： 512个字节的数据，1024个字符的字符串。

R: \*14 01 01 006#

01 ： 即注册成功。

E : \*02 005 006# 电脑处理数据出错。

# 查询用户版本

S: $14 02 002 005#

02 ：代表功能号，即查询版本功能。

002 ：用户userid 的后三位

R: \*14 02 1 005#

1 : 最高版本号。

# 指纹登录验证

S: $14 03 087101001002 001#

03： 代表功能号，即下载指纹登录功能。

087101001002 ：用户userid号。

R: \*01 01 111111 160621100940 087101001 002 709668456 001#

111111 ： 权限，1 ，0代表是否有相关权限。

160621100940 ： 时间，当前时间，可用来同步手持时间。

087101001 ：机构号。

002 ：服务器userid 的后三位，一般是一个机构的人员编号。

709668456 ：十进制的ic卡的卡号，因为指纹登录时，需要获得ic卡号。

E: \*14 00 00 000# 当前服务器没有该用户指纹信息。

# 指纹复核

S: $00 01 087101001002 001#

03： 代表功能号，即下载指纹登录功能。

087101001002 ：复核用户userid号。

R: \*01 01 111111 160621100940 087101001 002 709668456 001#

111111 ： 权限，1 ，0代表是否有相关权限。

160621100940 ： 时间，当前时间，可用来同步手持时间。

087101001 ：机构号。

002 ：服务器userid 的后三位，一般是一个机构的人员编号。

709668456 ：十进制的ic卡的卡号，因为指纹登录时，需要获得ic卡号。

E: \*14 00 00 000# 当前服务器没有该用户指纹信息。

# 下载指纹

S: $14 04 087101001002 1 1 001#

04： 代表功能号，即下载指纹功能。

087101001002 ：用户userid号。

1 ：版本号。

1 ：指纹序列号（一个人通常需要注册三个指纹，所以这里的取值为1,2,3）。

R: \*14 04 3 1 087101001002 1 \*\*\*\*\*\*（512字节） 001#

3 ： 用户指纹总数。

1 ： 此时为第1个指纹，序号。

\*\*\*\*\*\*（512字节） ：指纹数据，共512字节。

E: \*14 04 FF 001# //没有该用户指纹。

# 删除指纹

S: $14 05 087101001002 1 1 001#

04： 代表功能号，即下载指纹功能。

087101001002 ：用户userid号。

1 ：版本号。

1 ：指纹序列号（一个人通常需要注册三个指纹，所以这里的取值为1,2,3）。

R: \*14 05 00 001#

00 : 成功删除

E : \*02 005 001# 电脑处理数据出错。