

目录

修改历史.....	2
1. 币神卡简述.....	4
2. SDK 组成.....	6
图 1.....	7
图 2.....	7
5.1 Java API 流程介绍.....	8
5.2 伪代码示例 - 创建钱包.....	8
5.3 cn.btczen.sdk.BitcoinCallbackInterface 说明.....	12
5.4 cn.btczen.sdk.BitcoinCallbackInterface 说明.....	14
5.5 cn.btczen.sdk.BitcoinCallbackInterface 说明.....	15
5.6 参数说明.....	16

修改历史

版本号	日期 YYYYMMDD	作者	说明
1.0	20180809	WG	创建
1.1	20180821	WG	5.6.1 创建钱包: 修正输出参数中密语的 key 为 “SENTENCE”，不是“SENTENCES”
1.2	20180905	WG	5.2 伪代码示例 修正比特币、比特币 TESTNET 密钥路径 列表中的不正确注释。修正后与 demo 代码 保持一致。 5.6 参数说明 各章节均增加一个“异常状态码”表格， 列举可能的各种异常返回，并说明应如何 处理。
1.3	20181029	WG	增加亦来云支持说明

NFC-SDK 开发文档 V1.3

18 年 10 月 29 日

1. 币神卡简述



币神卡（BtcZen），采用高安全级别的金融卡芯片，搭载同等安全级别的操作系统及内置应用，为合作伙伴实现比特币、以太坊的冷钱包，提供了简洁、高效、低成本的安全解决方案。

目前版本的币神卡（BtcZen），支持下列操作：

1. 创建钱包

- (1) 支持 BIP32、39、43、44
- (2) 创建完毕后，卡内共生成 18 个 BIP32 密钥，详细见 1.1 章节列表
- (3) 目前不支持自定义密钥路径

2. 比特币交易签名

- (1) 用指定路径的密钥，对 raw transaction 的哈希值进行签名

3. 以太坊交易签名

- (1) 用指定路径的密钥，对 raw transaction 的哈希值进行签名

4. 删除钱包

- (1) 校验成功币神卡密码后，用户可删除钱包。之后即可重新创建

5. 删除钱包内助记词

- (1) 校验成功币神卡密码后，用户可删除钱包的助记词

6. 修改币神卡密码

- (1) 校验成功币神卡密码后，用户可修改密码为新密码。默认密码为 12345678。

7. 亦来云主链交易签名

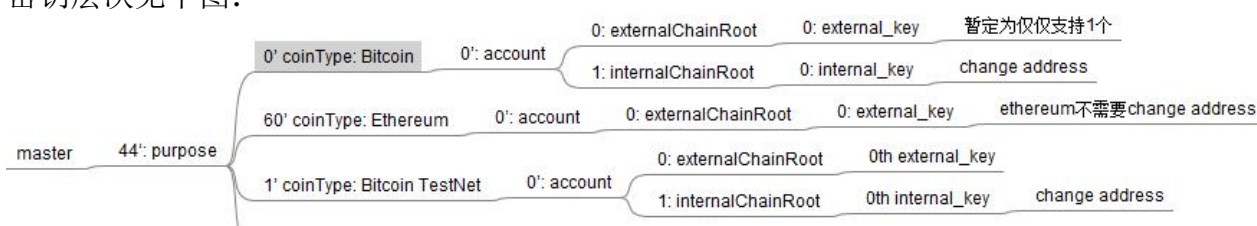
- (1) 用指定路径的密钥，对 raw transaction 的哈希值进行签名

1.1 卡内密钥路径列表

序号	路径	说明
1	m	master
2	m/44'	purpose
3	m/44'/0'	coin type: Bitcoin
4	m/44'/0'/0'	account
5	m/44'/0'/0'/0	external chain root
6	m/44'/0'/0'/1	internal chain root
7	m/44'/0'/0'/0/0	external chain root, external key 0
8	m/44'/0'/0'/1/0	internal chain root, internal key 0, change address
9	m/44'/1'	coin type: Bitcoin Testnet
10	m/44'/1'/0'	account
11	m/44'/1'/0'/0	external chain root
12	m/44'/1'/0'/1	internal chain root
13	m/44'/1'/0'/0/0	external chain root, external key 0
14	m/44'/1'/0'/1/0	internal chain root, internal key 0, change address
15	m/44'/60'	coin type: Ethereum
16	m/44'/60'/0'	account
17	m/44'/60'/0'/0	external chain root
18	m/44'/60'/0'/0/0	external chain root, external key 0
1	m	master (亦来云, ECC 曲线为 Secp256r1)
2	m/44'	Purpose (亦来云, ECC 曲线为 Secp256r1)

3	m/44'/0'	coin type: Bitcoin (亦来云, ECC 曲线为 Secp256r1)
4	m/44'/0'/0'	Account (亦来云, ECC 曲线为 Secp256r1)
5	m/44'/0'/0'/0	external chain root (亦来云, ECC 曲线为 Secp256r1)
6	m/44'/0'/0'/0'/0	external chain root, external key 0 (亦来云, ECC 曲线为 Secp256r1)

密钥层次见下图:



2. SDK 组成

为便于区块链合作伙伴将币神卡（BtcZen）集成到钱包类 android app 中，国链信安的 NFC-SDK 封装并提供了访问币神卡的操作接口。通过该 SDK，合作伙伴可快速实现冷钱包的创建、删除、导出卡内公钥、导出/删除助记词、修改卡密码、交易签名等操作。通过将冷钱包私钥安全存储于具有金融级安全机制的币神卡, App 自身无需再为私钥等敏感数据自行设计复杂、低效的安全存储机制。

NFC-SDK 由下列 2 部分组成:

1. nfc-sdk-xxxx.aar: 为 app 开发提供 android java 接口
2. libbtzcennative.so: 为上述 nfc-sdk-xxxx.aar 提供操作访问币神卡的应用协议

集成方法:

1. nfc-sdk-xxxx.aar: 见《BtcZen-nfc-sdk-AAR 加密集成手册.docx》
2. libbtzcennative.so: 应置于 app 的 src/main/jniLibs 目录, 与普通 .so 无异。

3. Demo 工程说明

btczen_sdk_demo_project.zip 演示了以比特币、以太坊的第三方开源模块作为基础，如何整合币神 NFC-SDK，实现对币神卡的各种操作访问。

该 demo 目前实现了下列操作：

1. 创建卡内冷钱包
2. 导出卡内冷钱包各密钥的公开信息
3. 比特币交易签名
4. 以太坊交易签名
5. 修改币神卡密码
6. 亦来云主链交易签名
7. 删除币神卡钱包

相关源代码见下图 1、图 2：

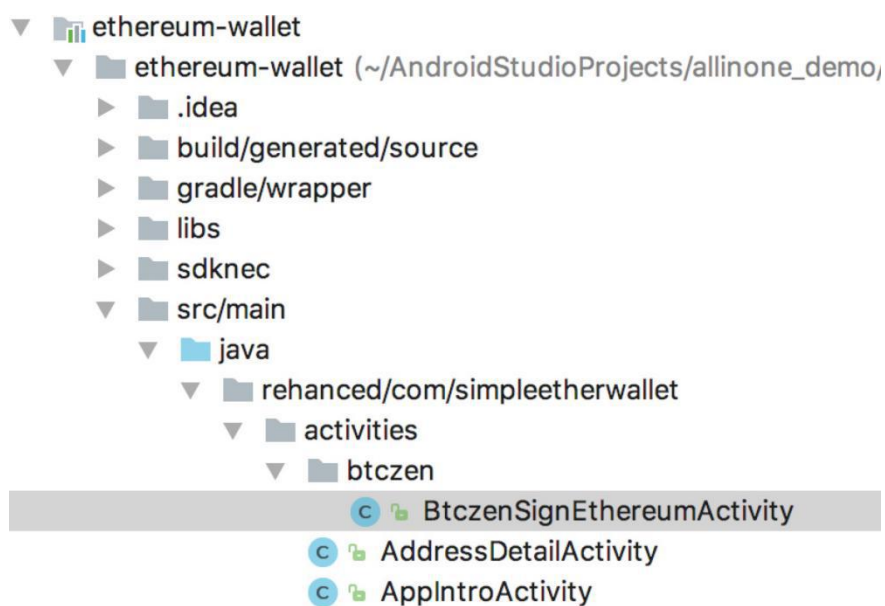


图 1

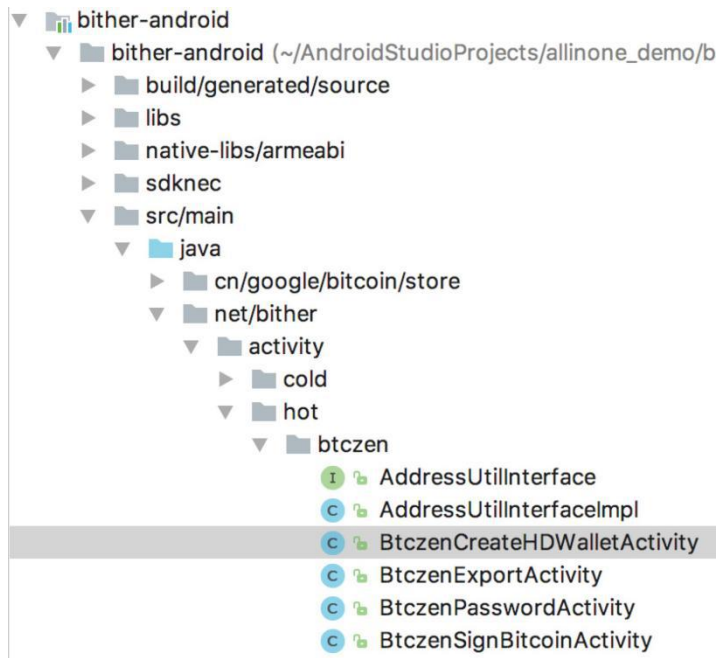


图 2

4. 应用权限

App 需做如下配置，以请求 NFC 权限：

```
<uses-permission android:name="android.permission.NFC" />  
<uses-feature android:name="android.hardware.nfc" android:required="true" />  
<uses-sdk android:minSdkVersion="10"/>
```

详细参考下列链接：

<https://developer.android.google.cn/guide/topics/connectivity/nfc/nfc>

其余配置，例如 Intent Filters、Tag Technologies、Pending Intents、Foreground Dispatch 无需 App 开发者设置，NFC-SDK 的 Java 代码会在运行时进行设置。

5. Java API 说明

本章节说明在 Android App 开发中如何调用 SDK 提供的 Java 类实现币神卡的各种操作。

5.1 Java API 流程介绍

1. 币神卡属于不联网的冷钱包，每次使用时（例如创建删除钱包、交易签名等操作时），用户均需在支持 NFC 的 Android 手机上做刷卡动作。
2. 在刷卡前，App 提示用户“输入币神卡密码”，以及操作所必需的其他参数（例如交易金额、新密码等）。
3. 各项必需的参数齐全后，App 通过在 Android activity 的 onCreate() onResume() onPause() 等钩子函数中，调用 `cn.btczen.sdk.BitcoinInterface` 类对象的相应钩子函数，请求 Android OS 捕获用户刷卡事件，并向 `cn.btczen.sdk.BitcoinInterface` 类对象指明要执行哪一种操作（如创建、删除、签名）。
4. 用户刷卡，Android OS 触发事件，`cn.btczen.sdk.BitcoinInterface` 类对象处理该事件，执行 App 指定的操作。操作过程中，向 `cn.btczen.sdk.BitcoinCallbackInterface` 类对象返回进度信息、异常信息、操作结果。
5. APP 通过 `cn.btczen.sdk.BitcoinCallbackInterface` 类对象获得进度信息、异常信息、操作结果，处理后展示给用户。

5.2 伪代码示例 - 创建钱包

5.2.1 导入下列 Java 类

```
import cn.btczen.sdk.BitcoinCallbackInterface;  
import cn.btczen.sdk.BitcoinInterface;  
import cn.btczen.sdk.Constants;
```

5.2.2 Activity 实现 BitcoinCallbackInterface 接口的各方法

```
void onGetDefaultNfcAdapterFailed();  
// 提示用户：手机 NFC 功能没有打开或不支持
```

```
void onGetDefaultNfcAdapterSucceed();
```

// 提示用户：手机 *NFC* 功能正常

void onIOException(String var1);

// 提示用户：刷卡失败（如刷卡位置不正确、刷卡时间太短导致操作中断等）

void onNfcAdapterDisabled();

// 提示用户：手机 *NFC* 功能没有打开或不支持

void onPcscException(String var1);

// 提示用户：操作失败（如卡密码不正确）

void onPleaseTapAgain();

// 提示用户：操作失败，请再次尝试

void onTagOperationStarted();

// 提示用户：操作已经开始，请耐心等待（目前创建钱包约需要 40 秒，其他操作小于 1 秒）

void onTagOperationSucceed(Bundle var1);

// 提示用户：操作成功，App 正在处理卡片返回的结果信息

5.2.3 Activity 在 `onCreate()` 中创建 `BitcoinInterface` 对象，创建时通过参数指明要进行 `BitcoinInterface.BtczenOperations.CREATE_HDWALLET` 操作，并调用该对象的 `onCreate()` 方法。具体操作亦可推迟至 `onTagOperationStarted()` 方法进行设置。

```
this.bitcoinInterface = new BitcoinInterface(this, this,  
BitcoinInterface.BtczenOperations.CREATE_HDWALLET);  
this.bitcoinInterface.onCreate(getIntent().getExtras());
```

5.2.4 Activity 在 `onNewIntent()` `onPause()` `onResume()` 方法中，调用 `BitcoinInterface` 对象的对应方法：

```
@Override  
protected void onNewIntent(Intent intent) {  
    super.onNewIntent(intent);  
    this.bitcoinInterface.onNewIntent(intent);  
}
```

```
@Override
```

```
protected void onPause() {  
    super.onPause();  
    this.bitcoinInterface.onPause();  
}
```

```
@Override  
protected void onResume() {  
    super.onResume();  
    this.bitcoinInterface.onResume();  
    this.progressDialog.dismiss();  
}
```

5.2.5 Activity 通过其各种 UI 元素，提示用户输入操作所需参数，例如创建钱包需要用户输入 2 项参数（卡密码、可选的助记词），在 onTagOperationStarted() 方法中，通过 BitcoinInterface.setBundle(parameters_bundle) 传递给 BitcoinInterface 对象。也可在此设定或修改要执行的操作。

```
@Override  
public void onTagOperationStarted() {  
  
    bitcoinInterface.setBundle(getBundle());  
    this.runOnUiThread(new Runnable() {  
        @Override  
        public void run() {  
            progressDialog.show();  
        }  
    });  
}
```

5.2.6 Activity 提示用户刷卡，随后在

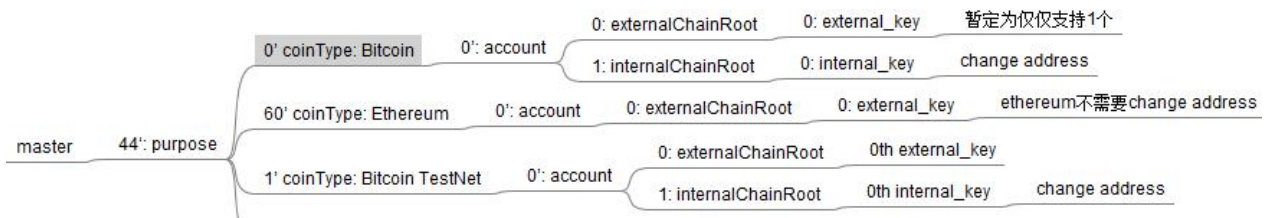
BitcoinCallbackInterface.onTagOperationSucceed() 方法中，处理 SDK API 返回的结果信息，提示用户创建是否成功。

```
@Override  
public void onTagOperationSucceed(final Bundle btczen_bundle) {  
    // btczen_bundle 中包含了钱包的各密钥的未压缩公钥、BIP39 序列化结果  
    //
```

```
}

```

5.2.7 以创建钱包为例，Bundle btczen_bundle 中包含了下列 BIP44 路径的扩展密钥的公钥、BIP32 序列化信息：



具体路径如下：

```

m
m/44'
m/44'/0'
m/44'/0'/0'
m/44'/0'/0'/0
m/44'/0'/0'/1
m/44'/0'/0'/0/0 // 以上为比特币密钥路径
m/44'/0'/0'/1/0
m/44'/1'
m/44'/1'/0'
m/44'/1'/0'/0
m/44'/1'/0'/1
m/44'/1'/0'/0/0 // 以上为比特币（Testnet）密钥路径
m/44'/1'/0'/1/0
m/44'/60'
m/44'/60'/0'
m/44'/60'/0'/0
m/44'/60'/0'/0/0 // 以上为以太坊密钥路径
m
m/44'
m/44'/0'
m/44'/0'/0'
m/44'/0'/0'/0
m/44'/0'/0'/0/0 // 以上为亦来云密钥路径（路径与比特币相同，但采用不同的 ECC 曲线）

```

其中，Bundle btczen_bundle 的 key 为密钥的路径，String 类型，值为 65 字节的未压缩公钥以及 BIP32 serialization，byte[] 类型。

net/bither/activity/hot/btczen/BtczenCreateHDWalletActivity.onTagOperationSucceed() 方法中给出了处理方法的示例。

5.3 cn.btczen.sdk.BitcoinCallbackInterface 说明

```
package cn.btczen.sdk;
```

```
/*
```

```
    Activity shall implements this interface, to receive operations notice from  
    BtcZen library,  
    and so as to info the current status what operations are executing.
```

```
    @author BtcZen.cn
```

```
*/
```

```
import android.os.Bundle;
```

```
public interface BitcoinCallbackInterface {
```

```
    /**
```

```
        * Invoked when BtcZen library tries to get default NFC adapter but failed.  
        *  
        * Android app should notice user to confirm if the mobile handset  
        supports NFC functions,  
        * or NFC functions already turned on, or NFC permission has been granted  
        to this app.
```

```
        *
```

```
        * This callback typically will be invoked while app is executing  
        onCreate(), but not limits to that.
```

```
        */
```

```
        void onGetDefaultNfcAdapterFailed();
```

```
    /**
```

```
        * Invoked when BtcZen library tries to get default NFC adapter and  
        succeed.
```

```
        *
```

```
        * Android app can simply omit this event and does nothing.  
        * This callback typically will be invoked while app is executing  
        onCreate(), but not limits to that.
```

```
        */
```

```
        void onGetDefaultNfcAdapterSucceed();
```

```
    /**
```

```
        * Invoked when BtcZen library is stopped because NFC functions of user  
        mobile handset not yet turned on.
```

```

    *
    * Android app should notice user to confirm if the mobile handset
    supports NFC functions,
    * or NFC functions already turned on, or NFC permission has been granted
    to this app.
    *
    * Note that this callback may be invoked for not only one time, while app
    executing onResume(), onPause() and so on.
    */
    void onNfcAdapterDisabled();

    /**
    * Invoked when BtcZen library finish all operations on BtcZen cards, and
    succeed.
    */
    void onTagOperationStarted();

    /**
    * Invoked when BtcZen library finish all operations on BtcZen cards, and
    succeed.
    */
    void onTagOperationSucceed(Bundle btczen_bundle);

    /**
    * Invoked when BtcZen library can't execute operations due to any NFC
    communication problem.
    *
    * @param info exception information
    */
    void onPcscException(String info);

    /**
    * Invoked when BtcZen library can't execute operations due to any Android
    OS problem.
    *
    * @param info exception information
    */
    void onIOException(String info);

    /**
    * Invoked when BtcZen library can't execute operations due to BtcZen card
    problem, for example,
    * wrong card used, wrong Btczen card PIN entered. App should promote user
    to confirm if they

```

```
* use the right card, if corrent PIN entered.
*/
void onPleaseTapAgain();

}
```

5.4 cn.btczen.sdk.BitcoinCallbackInterface 说明

```
public class BitcoinInterface {

    public enum BtczenOperations {
        IMPORT_PUBLIC(1), // 导出币神卡卡内密钥的公钥和 BIP32 序列化
        SIGN_BITCOIN(2), // 比特币交易签名
        SIGN_ETHEREUM(3), // 以太坊交易签名
        CHANGE_PASSWORD(4), // 修改币神卡密码
        CREATE_HDWALLET(5), // 创建钱包, 返回助记词和卡内密钥的公钥信息
        DELETE_HDWALLET(6), // 删除钱包
        DELETE_SENTENCES(7), // 删除密语
        SIGN_ELASTOS(8), // 亦来云交易签名
        SIGN_HASH_CURVE_ID(9), // 对交易哈希进行签名, 指定密钥的 ECC 曲线 ID
        NO_OPERATION(-1);
    }

    public BitcoinInterface(Activity activity, BitcoinCallbackInterface
callback, BtczenOperations operation) {
        // 略
    }

    public void onCreate(Bundle bundle) {
        // 略
    }

    public void onNewIntent(Intent intent) {
        // 略
    }

    public void onPause() {
        // 略
    }

    public void onResume() {
        // 略
    }
}
```

```
}
```

5.5 cn.btczen.sdk.BitcoinCallbackInterface 说明

```
public interface Constants {
```

```
    String INTENT_EXTRA_SENTENCE = "INTENT_EXTRA_SENTENCE"; // Used for create
operation
    String INTENT_EXTRA_PASSWORD = "INTENT_EXTRA_PASSWORD"; // Used for sign
operation
    String INTENT_EXTRA_FROM_PATH = "INTENT_EXTRA_FROM_PATH"; // for Bither &
Ether
    String INTENT_EXTRA_RAW_TX_SHA256ED = "INTENT_EXTRA_RAW_TX_SHA256ED"; //
for Bitcoin only
    String INTENT_EXTRA_RAW_TX_SIGNED = "INTENT_EXTRA_RAW_TX_SIGNED"; // for
return values of both Bitcoin & Ether
    String INTENT_EXTRA_OLD_PWD = "INTENT_EXTRA_OLD_PWD"; // Used only for
change PIN operation
    String INTENT_EXTRA_NEW_PWD = "INTENT_EXTRA_NEW_PWD"; // Used only for
change PIN operation
    String INTENT_EXTRA_ELASTOS_PUBLIC = "INTENT_EXTRA_ELASTOS_PUBLIC"; //
indicate all public keys of Elastos
    String INTENT_EXTRA_CURVE_ID = "INTENT_EXTRA_CURVE_ID"; // indicate what
ECC curve ID needed
    String INTENT_EXTRA_OPERATION = "INTENT_EXTRA_OPERATION"; // indicate what
operation needed

    // for Ether only
    String INTENT_EXTRA_AMOUNT = "INTENT_EXTRA_AMOUNT";
    String INTENT_EXTRA_FROM_ADDRESS = "INTENT_EXTRA_FROM_ADDRESS";
    String INTENT_EXTRA_RAW_TX_SHA3ED = "INTENT_EXTRA_RAW_TX_SHA3ED";
    String INTENT_EXTRA_TO_ADDRESS = "INTENT_EXTRA_TO_ADDRESS";
    String INTENT_EXTRA_GAS_PRICE = "INTENT_EXTRA_GAS_PRICE";
    String INTENT_EXTRA_GAS_LIMIT = "INTENT_EXTRA_GAS_LIMIT";
    String INTENT_EXTRA_DATA = "INTENT_EXTRA_GAS_DATA"; // Ethereum Data
    String INTENT_EXTRA_PUBLIC_ADDRESS = "INTENT_EXTRA_PUBLIC_ADDRESS"; //
for Ether only
    String INTENT_EXTRA_NONCE = "INTENT_EXTRA_NONCE"; // for Ether only

    int CURVE_ID_SECP256K1 = 0x00;
```



```

    int CURVE_ID_SECP256R1 = 0x10;
    int CURVE_ID_INVALID = 0xFF;

    int CURVE_ID_ELASTOS = CURVE_ID_SECP256R1;
}

public void setBundle(Bundle bundle) {
    bitcoinAdapter.setBundle(bundle);
}
}

```

5.6 参数说明

5.6.1 创建钱包

Bundle 内输入参数见下表：

key	value	类型	备注
INTENT_EXTRA_PASSWORD	币神卡密码	byte[]	UTF-8 编码
INTENT_EXTRA_SENTENCE	用户指定的助记词	byte[]	UTF-8 编码，可选

Bundle 内输出参数见下表：

key	value	类型	备注
INTENT_EXTRA_PASSWORD	币神卡密码	byte[]	UTF-8 编码，与输入相同
“SENTENCE” 注意：不是 INTENT_EXTRA_SENTENCE	助记词	byte[]	UTF-8 编码。若用户指定了助记词，则与输入相同。否则为新生成的助记词，需提示用户妥善保管
m	公钥+BIP32 序列化	byte[]	前 65 字节为未压缩的公钥；其余为 BIP32 序列化的公钥
m/44'	公钥+BIP32 序列化	byte[]	前 65 字节为未压缩的公钥；其余为 BIP32 序列化的公钥
... ..	公钥+BIP32 序列化	byte[]	对应 1.1 章节所列的其余 16 个密钥

异常状态码：

编码	说明	处理方法
6A84	钱包已创建，不能再次创建	提示用户：不能重复创建；或提示用户可以删除钱包后重新创建。
6A81	卡密码已锁死	提示用户：由于错误次数太多，卡密码已经锁定
63Cx	卡密码校验错误	提示用户：卡密码校验错误，还剩 x 次尝试机会（x 为 0-7）

5.6.2 比特币交易签名

Bundle 内输入参数见下表：

key	value	类型	备注
INTENT_EXTRA_PASSWORD	币神卡密码	byte[]	UTF-8 编码
INTENT_EXTRA_FROM_PATH	用户指定的密钥路径	byte[]	UTF-8 编码
INTENT_EXTRA_RAW_TX_SHA256ED	Raw transaction 的哈希值	byte[]	UTF-8 编码

Bundle 内输出参数见下表：

key	value	类型	备注
INTENT_EXTRA_PASSWORD	币神卡密码	byte[]	UTF-8 编码
INTENT_EXTRA_FROM_PATH	用户指定的密钥路径	byte[]	UTF-8 编码
INTENT_EXTRA_RAW_TX_SHA256ED	Raw transaction 的哈希值	ArrayList<byte[]>	UTF-8 编码，数组列表，以支持多 UTXO
INTENT_EXTRA_RAW_TX_SIGNED	Raw transaction 的哈希值的签名值	ArrayList<byte[]>	UTF-8 编码，数组列表，对应输入

编码	说明	处理方法
6A81	卡密码已锁死	提示用户：由于错误次数太多，卡密码已经锁定
63Cx	卡密码校验错误	提示用户：卡密码校验错误，还剩 x 次尝试机会（x 为 0-7）

6A82	找不到指定的密钥路径	APP 应确认输入参数中的密钥路径是否正确
6F00	待签名数据的长度不正确，应为 32 字节	APP 应确认输入参数中的哈希值长度是否正确

异常状态码：

5.6.3 以太坊交易签名

Bundle 内输入参数见下表：

key	value	类型	备注
INTENT_EXTRA_PASSWORD	币神卡密码	byte[]	UTF-8 编码
INTENT_EXTRA_FROM_PATH	用户指定的密钥路径	byte[]	UTF-8 编码
INTENT_EXTRA_RAW_TX_SHA3ED	Raw transaction 的哈希值	byte[]	UTF-8 编码

Bundle 内输出参数见下表：

key	value	类型	备注
INTENT_EXTRA_PASSWORD	币神卡密码	byte[]	UTF-8 编码
INTENT_EXTRA_FROM_PATH	用户指定的密钥路径	byte[]	UTF-8 编码
INTENT_EXTRA_RAW_TX_SHA3ED	Raw transaction 的哈希值	byte[]	UTF-8 编码
INTENT_EXTRA_RAW_TX_SIGNED	Raw transaction 的哈希值的签名值	Bundle	UTF-8 编码，数组列表，Bundle.get("0") 为签名值
INTENT_EXTRA_PUBLIC_ADDRESS	用户指定的密钥的公钥	byte[]	UTF-8 编码，便于应用对签名值进行校验

异常状态码：

编码	说明	处理方法
6A81	卡密码已锁死	提示用户：由于错误次数太多，卡密码已经锁定
63Cx	卡密码校验错误	提示用户：卡密码校验错误，还剩 x 次尝试机会（x 为 0-7）
6A82	找不到指定的密钥路径	APP 应确认输入参数中的密钥路径是否正确

		确
6F00	待签名数据的长度不正确，应为 32 字节	APP 应确认输入参数中的哈希值长度是否正确

5.6.4 修改卡密码

Bundle 内输入参数见下表：

key	value	类型	备注
INTENT_EXTRA_OLD_PWD	币神卡当前密码	byte[]	UTF-8 编码
INTENT_EXTRA_NEW_PWD	用户指定的新密码	byte[]	UTF-8 编码

Bundle 内输出参数见下表：

空 bundle，无返回。

异常状态码：

编码	说明	处理方法
6A81	卡密码已锁死	提示用户：由于错误次数太多，卡密码已经锁定
63Cx	卡密码校验错误	提示用户：卡密码校验错误，还剩 x 次尝试机会（x 为 0-7）
6A80	新密码或者旧密码长度错误	APP 应当强制新旧密码均为 8 字节，请检查用户输入长度

5.6.5 删除钱包

Bundle 内输入参数见下表：

key	value	类型	备注
INTENT_EXTRA_PASSWORD	币神卡当前密码	byte[]	UTF-8 编码

Bundle 内输出参数见下表：

key	value	类型	备注
INTENT_EXTRA_PASSWORD	币神卡当前密码	byte[]	UTF-8 编码，同输入

异常状态码：

编码	说明	处理方法
6A81	卡密码已锁死	提示用户：由于错误次数太多，卡密码已经锁定
63Cx	卡密码校验错误	提示用户：卡密码校验错误，还剩 x 次尝试机会（x 为 0-7）

5.6.6 删除钱包密码

Bundle 内输入参数见下表：

key	value	类型	备注
INTENT_EXTRA_PASSWORD	币神卡当前密码	byte[]	UTF-8 编码

Bundle 内输出参数见下表：

key	value	类型	备注
INTENT_EXTRA_PASSWORD	币神卡当前密码	byte[]	UTF-8 编码，同输入

异常状态码：

编码	说明	处理方法
6A81	卡密码已锁死	提示用户：由于错误次数太多，卡密码已经锁定
63Cx	卡密码校验错误	提示用户：卡密码校验错误，还剩 x 次尝试机会（x 为 0-7）

5.6.7 导出卡内密钥

Bundle 内输入参数见下表：

空 bundle，无需参数

Bundle 内输出参数见下表：

key	value	类型	备注
-----	-------	----	----

m	公钥+BIP32 序列化	byte[]	前 65 字节为未压缩的公钥；其余为 BIP32 序列化的公钥
m/44'	公钥+BIP32 序列化	byte[]	前 65 字节为未压缩的公钥；其余为 BIP32 序列化的公钥
... ..	公钥+BIP32 序列化	byte[]	对应 1.1 章节所列的其余 16 个密钥

异常状态码：

编码	说明	处理方法
6A81	卡密码已锁死	提示用户：由于错误次数太多，卡密码已经锁定
6983	卡尚未创建钱包	提示用户：请先创建钱包
6A82	APP 指定的密钥路径参数不正确，卡内没有匹配改路径的密钥	开发错误：请确认提供的密钥路径参数是否正确

5.6.3 亦来云交易签名

Bundle 内输入参数见下表：

key	value	类型	备注
INTENT_EXTRA_PASSWORD	币神卡密码	byte[]	UTF-8 编码
INTENT_EXTRA_FROM_PATH	用户指定的密钥路径	byte[]	UTF-8 编码
INTENT_EXTRA_RAW_TX_SHA3ED	Raw transaction 的哈希值	byte[]	UTF-8 编码
INTENT_EXTRA_CURVE_ID	CURVE_ID_ELASTOS	Int	0x10
INTENT_EXTRA_OPERATION	SIGN_ELASTOS	Int	

Bundle 内输出参数见下表：

key	value	类型	备注
INTENT_EXTRA_PASSWORD	币神卡密码	byte[]	UTF-8 编码
INTENT_EXTRA_FROM_PATH	用户指定的密钥路径	byte[]	UTF-8 编码

INTENT_EXTRA_RAW_TX_SHA3ED	Raw transaction 的哈希值	byte[]	UTF-8 编码
INTENT_EXTRA_RAW_TX_SIGNED	Raw transaction 的哈希值的签名值	Bundle	UTF-8 编码，数组列表，Bundle.get("0") 为签名值
INTENT_EXTRA_PUBLIC_ADDRESS	用户指定的密钥的公钥	byte[]	UTF-8 编码，便于应用对签名值进行校验
INTENT_EXTRA_CURVE_ID	CURVE_ID_ELASTOS	Int	0x10
INTENT_EXTRA_OPERATION	SIGN_ELASTOS	Int	

异常状态码：

编码	说明	处理方法
6A81	卡密码已锁死	提示用户：由于错误次数太多，卡密码已经锁定
63Cx	卡密码校验错误	提示用户：卡密码校验错误，还剩 x 次尝试机会（x 为 0-7）
6A82	找不到指定的密钥路径	APP 应确认输入参数中的密钥路径是否正确
6F00	待签名数据的长度不正确，应为 32 字节	APP 应确认输入参数中的哈希值长度是否正确

6. 参考资料

Android App 需要通过 NFC 实现对币神卡的物理层通讯访问，开发者可通过下列链接熟悉 Android NFC 相关 API：

1. <https://developer.android.google.cn/reference/android/nfc/package-summary>
2. <https://developer.android.google.cn/guide/topics/connectivity/nfc/nfc>
3. <https://developer.android.google.cn/guide/topics/connectivity/nfc/advanced-nfc>
4. <https://software.intel.com/zh-cn/android/articles/nfc-application-development-on-android-with-case-studies?language=es>
5. <https://www.nxp.com/cn/products/identification-and-security/nfc/nfc-developer-resources:NFC-APP-DEVELOPER-RESOURCE-HUB>

6. <https://www.jianshu.com/p/d044ebea9f12>