## 区块链基础及应用实验报告

### Ex1

# 网络空间安全学院 信息安全专业 2112492 刘修铭 0939

## 一、分币

1. 打开config.py文件,将之前领取bitcoin时生成的私钥替换到代码中。

```
1  my_private_key =
    CBitcoinSecret('cTQeRgRKPA3HaW415CDc9cygNvSXHGdEsFD6oPnhsjewHffoqp4F')
```

2. 打开split\_test\_coins .py文件,将要拆分的总金额、UTXO交易ID、UTXO索引、拆分数量等填入其中。

```
if __name__ == '__main__':
    amount_to_send = 0.015 # amount of BTC in the output you're splitting
    minus fee

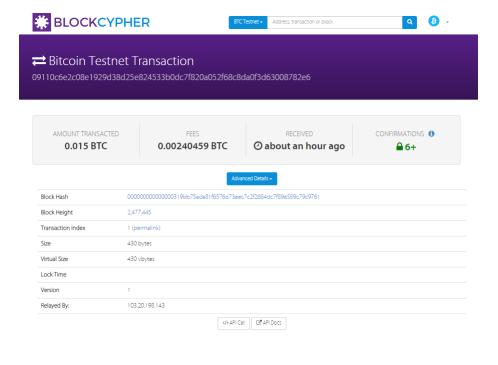
txid_to_spend = (
        '322c73499ab966deaf51fd2be62a1859f8a714102d5355671a40d8a2f74848fb')

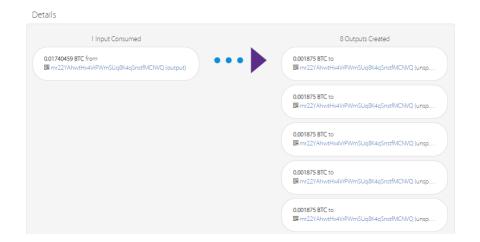
utxo_index = 0

n = 8 # number of outputs to split the input into

split_coins(amount_to_send, txid_to_spend, utxo_index, n)
```

- 。 此处选用拆分0.015bitcoin;
- 。 将领取bitcoin时的交易ID写入;
- 。 由于是该事务第一个输出值, 故index值为0;
- 。 为了保证后面实验的正常进行, 此处选择拆分为8个部分。
- 3. faucet截图





4. 输出信息

由于篇幅所限,输出信息将于附件output\_split.txt中展示。

## 二、发币

本部分主要使用ex1.py文件。

打开文件可以看到三个需要补全的部分。

- 1. P2PKH\_scriptPubKey函数
  - 经查询可知,该函数主要用于生成一个标准的PayToPublicKeyHash交易的输出脚本 scriptPubKey。课上学习可知,该部分主要由以下几部分构成:
    - OP\_DUP:将栈顶元素复制一份,压入栈中,便于后续的验证操作;
    - OP\_HASH160: 计算栈顶元素的RIPEMD160哈希值,并将结果压入栈中;
    - pubkey\_hash:接收方的比特币地址的公钥哈希;
    - OP\_EQUALVERIFY:比较栈中的两个元素是否相等,如果相等,继续执行,否则交易失败;

■ OP\_CHECKSIG:验证发送方的数字签名是否与公钥匹配,如果匹配,交易有效,否则交易失败。

```
def P2PKH_scriptPubKey(address):
2
      # 获取地址的脚本
3
      script = address.to_scriptPubKey() # 根据给定的比特币地址生成了一个脚本
4
      # 获取公钥哈希
5
      pubkey_hash = script[3:-2] # 从生成的脚本中提取了公钥哈希,用于识别接
   收方
6
      # 构建scriptPubKey
7
      script_pubkey = [
8
          OP_DUP,
                              # 复制栈顶元素
9
          OP_HASH160,
                              # 计算栈顶元素的哈希
10
                              # 公钥哈希
          pubkey_hash,
          OP_EQUALVERIFY,
                              # 检查栈顶两个元素是否相等
11
         OP_CHECKSIG
12
                              # 检查栈顶元素是否是有效签名
13
      1
14
      return script_pubkey
```

#### 2. P2PKH\_scriptSig函数

- 经查询, P2PKH\_scriptSig(txin, txout, txin\_scriptPubKey)函数主要用于生成一个有效脚本用来解 锁输出并发送回faucet, 其参数含义如下:
  - txin:表示输入的交易数据;
  - txout:表示输出的交易数据;
  - txin\_scriptPubKey:表示输入交易的脚本公钥。
- P2PKH交易是比特币中最常见的交易类型之一,它使用公钥哈希作为地址,并且需要提供与之对应的私钥进行签名验证。故而该函数需要完成以下几个任务:
  - 验证txin和txout的有效性,确保输入和输出的交易数据是有效的;
  - 解析txin\_scriptPubKey, 提取出公钥哈希;
  - 使用私钥对txin进行签名,生成一个脚本签名;
  - 将脚本签名和公钥作为输入的脚本签名 (scriptSig) 返回。

```
def P2PKH_scriptSig(txin, txout, txin_scriptPubKey):
2
       # 创建签名
 3
       signature = create_OP_CHECKSIG_signature(txin, txout,
    txin_scriptPubKey, my_private_key)
4
      # 获取公钥
 5
       public_key = my_public_key
 6
       # 构建脚本签名scriptSig
7
       script_sig = [
8
           signature, # 签名
9
           public_key # 公钥
10
11
       return script_sig
```

#### 。 该函数主要对一些参数进行配置

```
1 amount_to_send = 0.01 # 交易费
2 txid_to_spend = (
3 'a089277bdfefd68eff3c21c8e01247225263cb401dbe2f6f8da043c30ca8a212') # 之 前交易ID的hash
4 utxo_index = 0 # UTXO索引
```

#### 4. faucet截图

