区块链实验四实验报告

2011428 王天行，2012679 王娇妹

一、作业要求

在这项作业中，你需要实现 Alice 和 Bob 两方之间跨链原子交换代码的关键部分。Alice 在 BTC Testnet3 上有比特币，这是 project1 使用的标准比特币测试网。Bob 在 BCY Testnet 上拥有比特币，BCY Testnet 是 Blockcypher 的比特币测试网，由 Blockcypher 独家挖矿和维护。他们希望安全地交换各自 coin 的所有权，这是一个简单交易无法完成的事情，因为它们位于不同的区块链上。

这里的想法是围绕一个只有一方（Alice）知道的秘密 x 建立交易。在这些事务中，只有 H（x）将被发布，而 x 为秘密。交易将以这样的方式建立，一旦 x 被揭露，双方都可以赎回对方发送的硬币。如果 x 永远不会被揭露，双方将能够安全地取回他们的原始硬币，而不需要另一方的帮助。

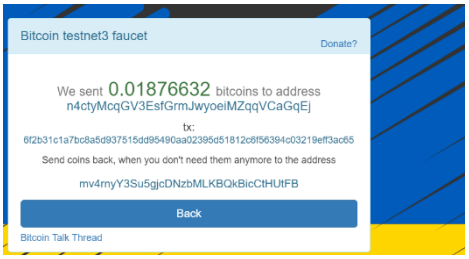
二、实验内容

（一）准备工作

1、为 Alice 和 Bob 创建 BTC testnet 密钥并填入 keys.py

```
alice_secret_key_BTC = CBitcoinSecret(
    'cNiGSvy9TyL9jwPRVQo3KgHCSmePBFKpZNgUCbP17jxQQmGCGJii')
bob_secret_key_BTC = CBitcoinSecret(
    'cSxyqn87vksytNWP8SEmj52iUBq2y22q4j9ZEDLKQUBPKfcLy5H6')
```

2、为 Alice 的 BTC 地址领取测试币



3、为 Alice 和 Bob 创建 BCY testnet 密钥并填入 keys.py

```
alice_secret_key_BCY = CBitcoinSecret.from_secret_bytes(
    x('fbf92855d8f637338074ad91f9277742eb8f1c0968b6ca3a37096396095ec59e'
))
bob_secret_key_BCY = CBitcoinSecret.from_secret_bytes(
```

```
    x('37fe2996a196014d40e353130550633b85cddd53bb1306e23ab307162182c8ed'
))
```

4、在 Blockcypher 测试网（BCY）上为 Bob 的 BCY 地址领取测试币

```
{
  "tx_ref":
"5909301a2fe5a22ac90e6135903cf8b8728ae14363c919abfb92c1e8b6bce7f8"
}
```

5、使用 split_test_coins.py（填写文件中的相关字段）划分领取的币

```
my_private_key =
CBitcoinSecret.from_secret_bytes(x('37fe2996a196014d40e353130550633b85c
ddd53bb1306e23ab307162182c8ed'))
my_public_key = my_private_key.pub
my_address = P2PKHBitcoinAddress.from_pubkey(my_public_key)
amount_to_send = 0.0009 # amount of BTC in the output you're splitting
minus fee
txid_to_spend = (
     '5909301a2fe5a22ac90e6135903cf8b8728ae14363c919abfb92c1e8b6bce7f8')
utxo_index = 0
n = 10 # number of outputs to split the input into
network = 'bcy-test' # either 'btc-test3' or 'bcy-test'
```

6、填写 swap.py

```
# 0.001 每份0.0009
alice_txid_to_spend     =
"19afad7e9a6d2f83b760b827666762a18e9b25c66e8d37098d37f03a478be85f"
alice_utxo_index        = 2
alice_amount_to_send    = 0.0009
bob_txid_to_spend       =
"0e978a39565bd40299bce70a67a5fac921dbccfc17fc6527794c438948f20f67"
bob_utxo_index          = 2
bob_amount_to_send      = 0.00008
btc_test3_chain_height  = 2408899
bcy_test_chain_height   = 565982
alice_locktime = 5
bob_locktime = 3
tx_fee = 0.00001
broadcast_transactions = True
alice_redeems = False #True
```

（二）完善 swap_scripts.py 中的脚本

1、coinExchangeScript 脚本

对于接受者赎回交易/交易超时而将比特币退还给发送者的两种情况，coinExchangeScript 采取 if-else 操作分别处理，所以整个脚本可以看作三个部分：
①检验是哪个解锁脚本。由于两个脚本长度不同，所以可以利用 OP_DEPTH 得到栈大小来检验是哪个解锁脚本。
②执行 OP_IF 后的分支（赎回交易）。
验证秘密 secret 与接收者签名 OP_HASH160, hash_of_secret, OP_EQUALVERIFY 将栈中的 secret 进行哈希运算，与公开的 Hash(x) 比较，检验秘密 secret 是否正确；public_key_recipient, OP_CHECKSIG 比对栈中的接受者签名是否正确。
③执行 OP_ELSE 后的分支（交易退还），验证发送者和接受者的两个签名。

```
def coinExchangeScript (public_key_sender, public_key_recipient,
  hash_of_secret):
    return [
        OP_DEPTH, 2, OP_EQUAL,
        OP_IF,
        OP_HASH160, hash_of_secret, OP_EQUALVERIFY,
        public_key_recipient, OP_CHECKSIG, OP_ELSE,
        2, public_key_sender, public_key_recipient,
        2, OP_CHECKMULTISIG, OP_ENDIF ]
```

2、赎回脚本

①对于赎回交易，只需将签名与秘密依次压栈。
②对于交易退还，把两个签名压栈即可，不过需要注意先 OP_0。

```
def coinExchangeScriptSig1(sig_recipient, secret):
    return [
        sig_recipient,
        secret    ]
def coinExchangeScriptSig2(sig_sender, sig_recipient):
    return [
        OP_0,
        sig_sender ,
        sig_recipient  ]
```

三、实验结果

```
alice_redeems = False
```

```
alice_redeems = True
```

四、问题解释

（一）以 Alice 用 coinExchangeScript 向 Bob 发送硬币为例：

如果 Bob 不把钱赎回来，Alice 为什么总能拿回她的钱?

当 Alice 用向 Bob 发送交易请求时，如果 Bob 不履行承诺，Alice 将不会在交易上签名，秘密 x 的值也不会披露，所以 Bob 不能拿到 Alice 的币。直到 time-locked transaction 解冻时，Alice 赎回自己的币。

为什么不能用简单的 1/2 multisig 来解决这个问题?

如果使用简单的 1/2 multsig，任一方都可以单独打破约定，从而使另一方受损。而使用以上机制可以保证双方的利益一定都不会受损，因为没有任何一方可以单独毁约，不通过对方的配合而赎回硬币。

（二）解释 Alice (Bob) 创建的一些交易内容和先后次序，以及背后的设计原理。

交易内容和先后次序：

1. Alice 创建第一笔交易，指定解锁方式为：双方共同签名，或 Bob 的签名加上 Alice 创建的 secret。
2. Alice 创建第二笔交易，指定解锁方式为：在到达指定的解锁时间 locktime 之后，可以赎回自己的币。
3. Bob 创建第一笔交易，指定解锁方式为：双方共同签名，或 Alice 的签名加上 Alice 创建的秘密 x（使用同样的 Hash(x)）。
4. Bob 创建第二笔交易，指定解锁方式为：在到达指定的解锁时间之后，可以赎回自己的币。

原理：

1. 在 Alice 创建交易后，Bob 需要知道 Alice 的 secret 才可以领取到 Alice 的币。
2. 在 Bob 创建交易后，Alice 可以索要 Bob 的币，使用其签名和 secret 来解锁 Bob 的第一笔交易。此时 secret 将会被传到区块链上。这时 Bob 在得知区块链上的 secrete 之后也就可以解锁第一笔交易，完成交易了。
3. 如果交易没有达成，双方可以在解锁时间后取回各自的硬币。

（三）本次作业中，一次成功的跨链原子交换中，资金是如何流转的？

1. 首先 Alice 随机生成一个密钥 secret，发起交易 1，向 Bob 比特币账户转帐。该交易只有在得到 Bob 的签名和密钥 K 时才能完成。
2. 在交易 1 发起前，Alice 会广播一笔回撤交易（交易 2），该交易表明若一定时间内交易 1 未完成，则将资产回退给 Alice；回撤交易得到 Alice 与 Bob 的共同签名后生效。同时，Alice 只有在交易 2 成功生效的情况下，才会发起交易 1。
3. Bob 接收到 Alice 的交易 2，若同意跨链转账则在交易 2 中添加签名。此时交易 2 生效，Alice 将交易 1 向全网广播。
4. Bob 发起交易 3，向 Alice 支付币以得到密钥 secret，Alice 在交易中输入密钥 secret 和签名后才可以得到 Bob 的币。为保障交易 3 中 Bob 资产安全，在交易 3 之前同样会发送一笔回撤交易（交易 4）。
5. Alice 收到交易 4 后，会附加自己的签名，回撤交易 4 生效后，Bob 将交易 3 在全网广播。
6. Alice 为获得 Bob 的币，在交易 3 中附上自己的签名和密钥 secret；此时交易 3 成功，Alice 获得 Bob 的币，Bob 获得密钥 secret。