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

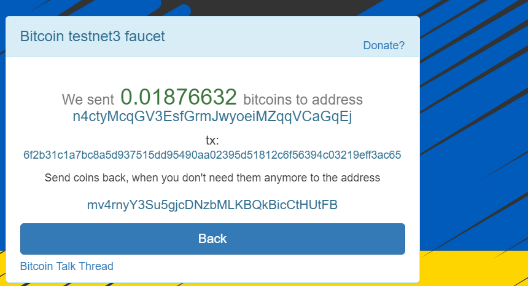
alice\_secret\_key\_BTC = CBitcoinSecret(

    '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('37fe2996a196014d40e353130550633b85cddd53bb1306e23ab307162182c8ed'))

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 [

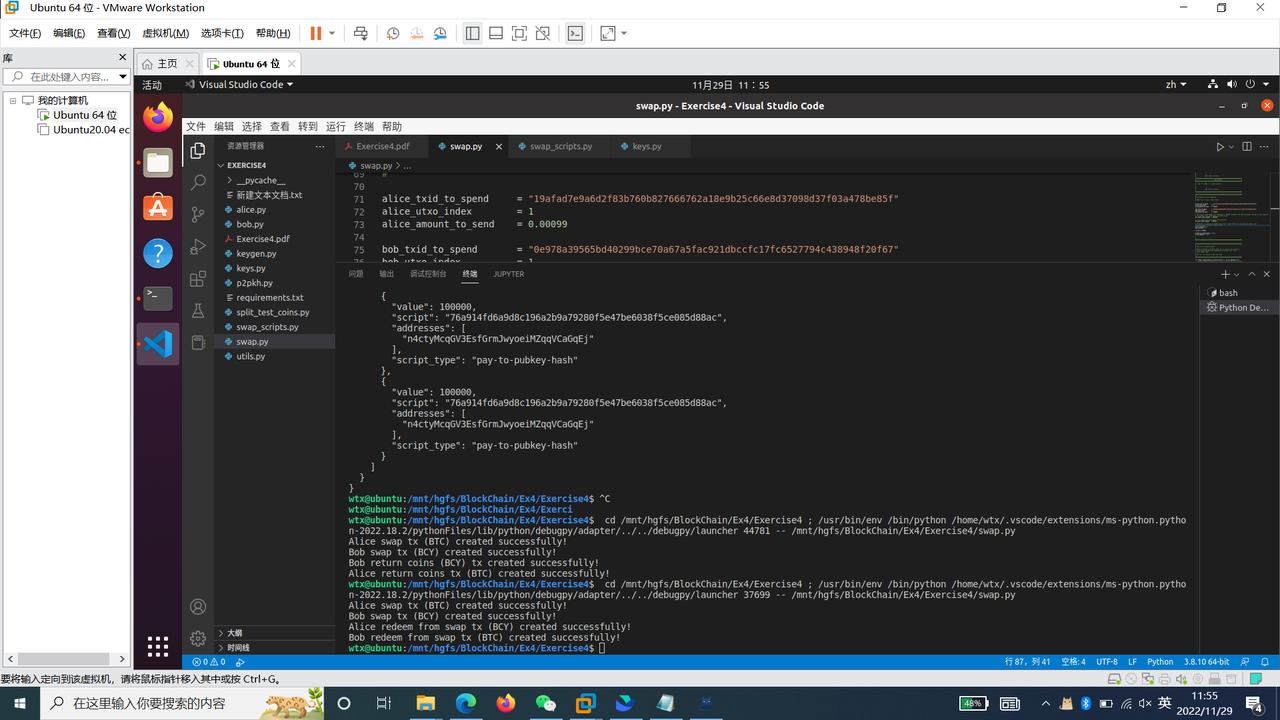
        OP\_0,

        sig\_sender ,

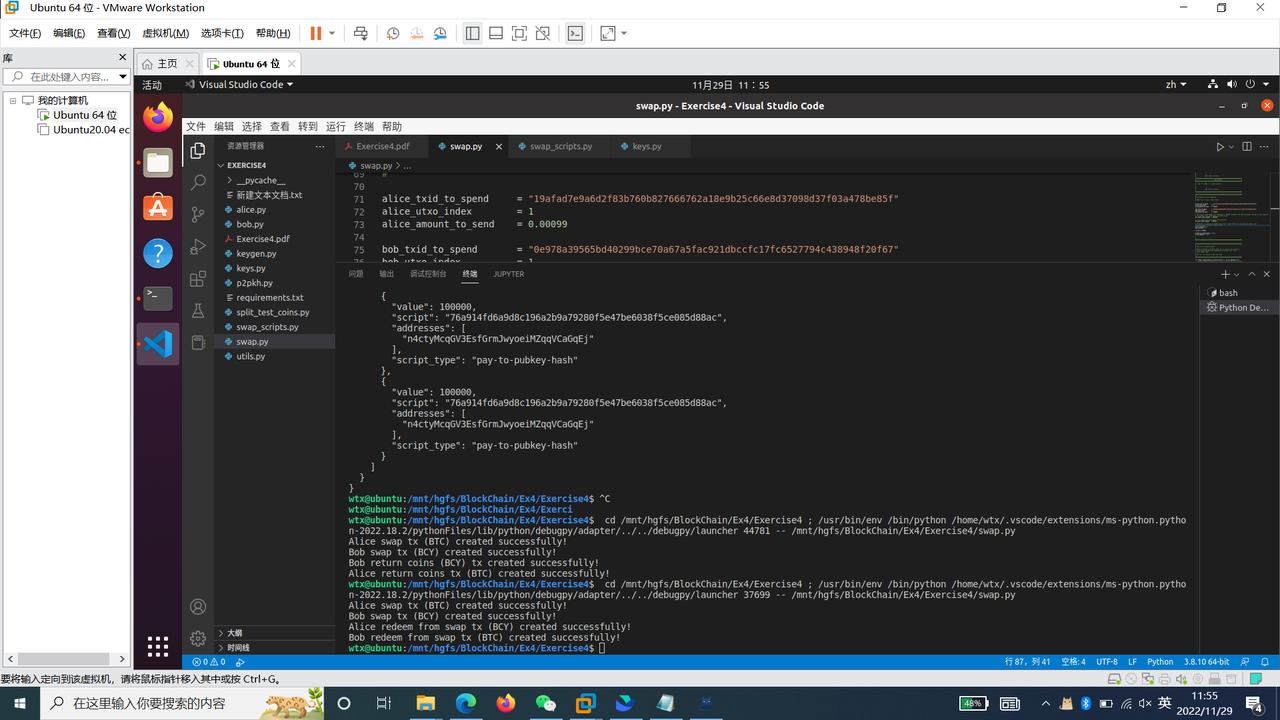
        sig\_recipient  ]

#### 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。