南开大学 Nankai University

南　　开　　大　　学

计　算　机　学　院

区块链基础及应用实验报告

跨链原子交换

于皓弛　　李祺萦

学号：2010137　　2013742

年级：2020 级

专业：信息安全法学、计算机科学与技术

指导教师：苏明

2022 年 12 月 4 日

目录

一、 代码内容讲解

在申请好 Alice 与 Bob 的两个测试网络的密钥并为他们领取测试币后，着手进行交换脚本的编写。考虑创建跨链原子交换所需事务所需的 ScriptPubKey。此交易必须可由接收者赎回（如果他们有一个与 Hash（x）对应的秘密 x），或者可以用发送者和接收者的两个签名赎回。对于 coinExchangeScript 的编写，需要先分辨采用的是哪个解锁脚本从而进行下一步操作。

对于用两个签名的赎回的脚本 coinExchangeScriptSig2，直接利用 P2MS，先将 CHECKMULTISIG 需要的 OP_0 入栈，之后是两个人的签名。

```
# This is the ScriptSig for sending coins back to the sender if unredeemed
LitchiHotpot
def coinExchangeScriptSig2(sig_sender, sig_recipient):
    return [
        OP_0, sig_sender, sig_recipient
    ]
```

图 1: 修改后的 coinExchangeScriptSig2

而对于知道秘密 X 的解锁脚本，则将签名和秘密依次压入栈即可。

```
# This is the ScriptSig that the receiver will use to redeem coins
LitchiHotpot
def coinExchangeScriptSig1(sig_recipient, secret):
    return [
        sig_recipient, secret
    ]
```

图 2: 修改后的 coinExchangeScriptSig1

之后就可以着手编写 coinExchangeScript。对于两个解锁脚本的判断可以采用 OP_DEPTH 判断脚本栈大小。sig1 大小为 2，sig2 大小为 3。所以采用 OP_EQUAL 判断。如果是 sig1，则先判断秘密是否正确。将 secret 哈希后与 hash_of_secret 验证，之后验证签名即可。如果是 sig2，则进行 CHECKMULTISIG。下面是编写好的脚本代码：

```
# This is the ScriptPubKey for the swap transaction
LitchiHotpot
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
    ]
```

图 3: coinExchangeScript

最后可以进行验证，首先在本地，不进行赎回。可以看到测试币原路返回。

```
Alice swap tx (BTC) created successfully!
Bob swap tx (BCY) created successfully!
Bob return coins (BCY) tx created successfully!
Alice return coins tx (BTC) created successfully!
```

图 4: 本地验证，不赎回

本地验证，赎回。赎回也成功。

```
Alice swap tx (BTC) created successfully!
Bob swap tx (BCY) created successfully!
Alice redeem from swap tx (BCY) created successfully!
Bob redeem from swap tx (BTC) created successfully!
```

图 5: 本地验证，赎回

进行广播之前，需要修改一个小 bug，将 alice 与 bob 文件里所有的 b2x 替换为 b2lx，才能返回正确的 json。

之后进行广播，等待二十多分钟后，验证成功。

首先是 Alice 赎回了 BCY。

```
Alice redeem from swap tx (BCY) created successfully!
201 Created
{
  "tx": {
    "block_height": -1,
    "block_index": -1,
    "hash": "c5181b7711b4ee590b5826b80e8c133f00945f5a7cc9713cb24853191b38a092",
    "addresses": [
      "CCWJ9XDb1apZrfLVGuQeFceG8ckcJXNkHy"
    ],
```

图 6: 广播，Alice 赎回

验证也成功。

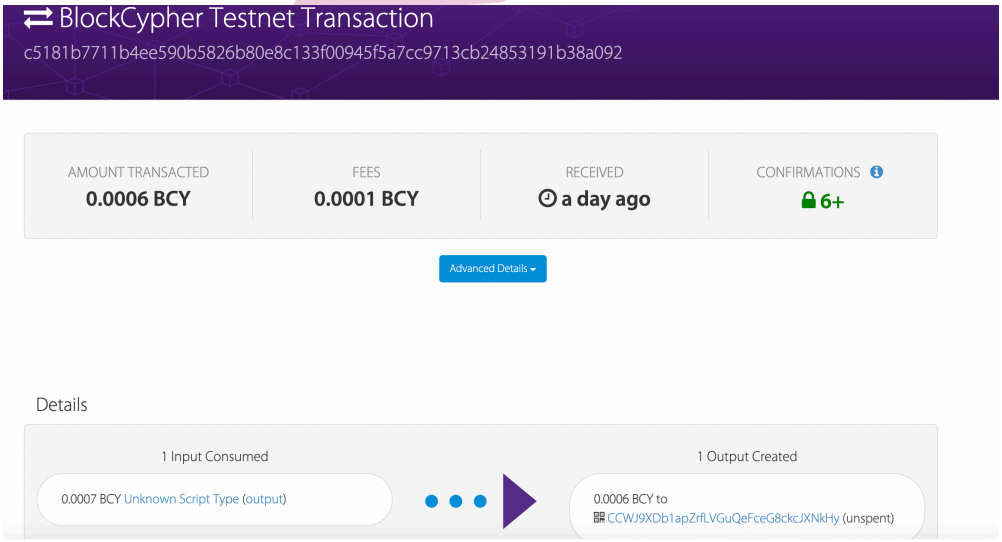

图 7: Alice 赎回

其次是 Bob 赎回了 BTC。

```
Bob redeem from swap tx (BTC) created successfully!
201 Created
{
  "tx": {
    "block_height": -1,
    "block_index": -1,
    "hash": "4cc312761ac96ed86d1018d719130587f98f99d224bb5226cb6431ee37d3f676",
    "addresses": [
      "mzF2hiV3PJoiokXqgBeanyhLeAFDh1biXH"
    ],
```


图 8: 广播，Bob 赎回

验证也成功。

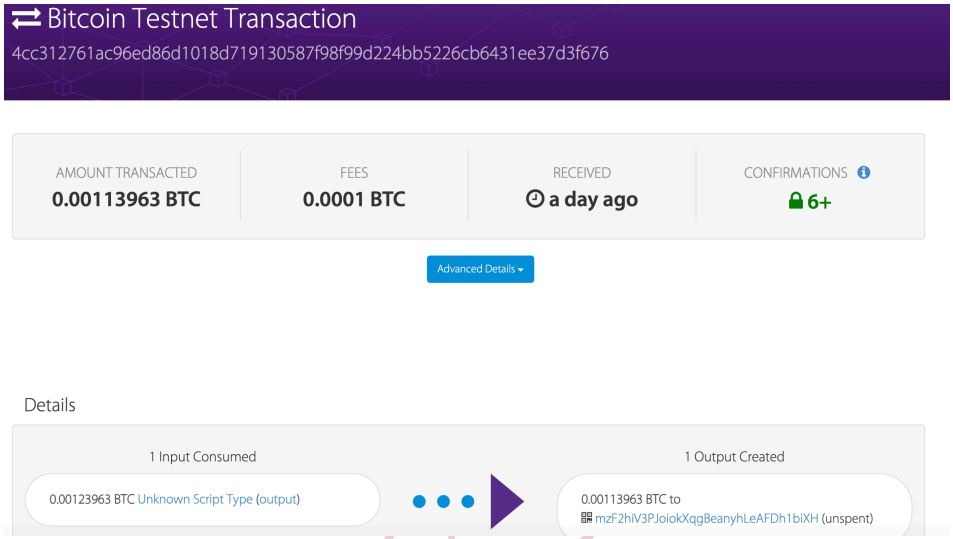

图 9: Bob 赎回

广播不赎回经过一段较长的时间后，测试币原路返回。相关交易哈希保存在“广播不赎回”文件里。

二、 问题解答

当 Alice 用 coinExchangeScript 向 Bob 发送硬币时，由于 Alice 设定了锁定时间的交易，若 Bob 一直不赎回，时间超过设定值后硬币将原路返回给 Alice，具体代码片段如下：

```
# Alice creates a time-locked transaction to return coins to herself
alice_return_coins_tx = alice.return_coins_tx(
    alice_amount_to_send - (2 * tx_fee),
    alice_swap_tx,
    btc_test3_chain_height + alice_locktime,
    alice_swap_scriptPubKey,
)
```

图 10: Alice 设定的 Time-locked 交易

如果只是简单使用 1/2Multisig，则 Bob 可能在 Alice 不赎回的情况下赎回硬币，这对于交易的发起者 Alice 来说是不安全的。

三、 设计原理

首先，Alice 生成一个秘密 X，并将 X 的哈希公布。

之后，Alice 创建一个可以被 Bob 用秘密 X 赎回或者被 Alice 与 Bob 两人签名赎回的交易。同时，Alice 也创建一个一旦超时就会将币原路返回的交易。之后 Alice 请 Bob 为两个交易签名。在 Bob 签完名之后，Alice 广播她创建的第一个交易。

Bob 也创建两个同样的交易并请 Alice 签名。签名后 Bob 将他创建的第一个交易广播。

如果 Alice 想要赎回交易，就需要将秘密 X 公布。此时 Bob 也知道了秘密 X，也可以将交易赎回。两人完成了交换。

设计原理就是需要满足原子交易，交易必须是全有或者全无。如果 Alice 不赎回交易，则 Bob 就无法赎回交易。在等待一段时间后，虚拟货币就会原路返回。这保证了交易的原子性。

四、 资金流转

在一次成功的跨链原子交易中，首先由发起者建立有两种赎回方式的交易，发送资金，然后要求接收者对交易进行签名，发送者将交易进行广播，接收者也上传自己的待交换资金。之后若秘密 x 未被揭露，等待一定时间后，资金会各自返回给双方，而当 x 被揭露，双方的资金就会返回给对方，硬币交换成功完成。