

跨链原子交换实验

姓名：李帅东 学号：2111231

姓名：齐明杰 学号：2113997

1 实验目的

本次实验的目的是实现**跨链原子交换**（Atomic Cross-Chain Swap）的关键部分。在这个过程中，参与者Alice和Bob将在不同的区块链上安全地交换加密货币的所有权。Alice拥有BTC Testnet3上的比特币，而Bob在BCY Testnet上拥有比特币。这种交换不能通过简单的交易实现，因为它们位于不同的区块链上。实验的核心是围绕一个**只有一方知道的秘密**进行交易设计，确保只有在秘密被揭露时，双方才能够赎回对方发送的硬币。如果秘密永远不被揭露，双方将能够安全地取回他们的原始硬币。

2 实验原理

跨链原子交换是一种加密技术，允许在不同的区块链之间交换资产，而无需信任第三方。这种方法的核心是使用加密哈希函数（例如SHA-256）来创建一个条件，只有在某个秘密值（ x ）被揭露时，条件才能被满足。

在这个实验中，交易是围绕一个**秘密值** x 构建的。只有哈希值 $H(x)$ 会被公开，而 x 保持秘密。当 x 被揭露时，参与方可以使用它来赎回对方链上的资产。这个过程中，关键的技术组件包括：

- Hash Time-Locked Contracts (HTLCs):** 这是实现原子交换的关键机制，它结合了哈希锁（确保只有知道秘密 x 的人才能解锁资金）和时间锁（确保如果交易在一定时间内未完成，资金将返回给发送方）。
- 秘密 x 的创建与管理:** x 是交易的关键，它的安全管理至关重要。只有当双方都遵守交易协议时，这个秘密才能正确地启用交换。
- 交易签名与验证:** 在跨链交换中，每个参与者必须在各自的区块链上创建并签名交易。这些交易包含了指向HTLCs的引用和特定的条件，如哈希值和时间锁。验证过程确保交易满足所有区块链协议的要求。
- 交易广播与确认:** 创建和签名后的交易需要在相应的区块链上广播。交易一旦被区块链网络确认，它们就变得不可逆转，从而确保了资产交换的不可撤销性。
- 秘密 x 的揭露与资产赎回:** 在交易被确认后，秘密 x 需要在规定的时间内被揭露。揭露秘密是赎回对方资产的关键。一旦 x 被公开，任何一方都可以使用它来解锁HTLC，从而完成资产的转移。
- 时间锁的作用:** 如果在规定时间内秘密 x 没有被揭露，资产将通过时间锁返回给原始所有者。这确保了即使一方未能遵守交易协议，另一方也不会损失资产。

3 实验过程

3.1 BTC创建账户 & 领取币

为Alice 和Bob创建密钥：执行kengen.py得到私钥和公钥：

```
1 Alice
2   Private key: cN29hw7R3vswDjSmjaPMNXdQtuD6n1DTAPP59XxHMcN8rTCUFcgz
3   Address: n2FUQDpz2LsRsfsFW2uF4XuNHRxDsLVAn
4
5 Bob
6   Private key: cUzSFAkmxCmNSaooaWUQoWv1Mu6Ji4Q8VAKG8vdiDSmyQ5sp1vXy
7   Address: n1BLLzAEjXKEbtkit6kurhJ5Bss6zRVqRR
```

为Alice的BTC领取测试币：

```
1 Txid: [80a78ff0d5915e6543c6605cb0812697b10a99e5c818f3463df0b3b2e607ee01]
```

We sent **0.01892766** bitcoins to address
n2FUQDpz2LsRsfsFW2uF4XuNHRxDsLVAn
tx: 80a78ff0d5915e6543c6605cb0812697b10a99e5c818f3463df0b3b2e607ee01
Send coins back, when you don't need them anymore to the address
mv4rnyY3Su5gjcDNzbMLKBQkBicCtHUtFB



3.2 BCY创建账户 & 领取币

3.2.1 Blockcypher 注册帐户

注册账户获取API token

- ID:2111231@mail.nankai.edu.cn
- pw:Blockchain2023
- Token:3cc99a91b23843d88d79e37f3ec3076a

3.2.2 创建BCY密钥

为Alice和Bob创建BCY密钥并且填入keys.py

- 使用指令 `curl -X POST https://api.blockcypher.com/v1/bcy/test/addr?token=YOURTOKEN`
- Alice

```

1  "private":
    "f983b81de6cab7a84636ae2cb50b622654c7712ca1dde6c030f3aa399e46d3e3",
2  "public":
    "02af90ced1a0e21f7abf6dfa72c6acd2cc9365419b9b2c91bd885bc4fae7c5c5b2",
3  "address": "C17UZqVPbajwgedoaY4a9MYiEzhnijmdFy",
4  "wif": "Bwh47JHKmobAXan2D4vBCSBqEum8LesqbDX21iBDJKzd9W94kpTU"

```

- Bob

```

1  "private":
    "217099cc98922bdfe8e05f4a05f9a8cbb10e9a72e1655ed96c3958d4395b9ca3",
2  "public":
    "0234e56b7c63bee9fbef88186a2ac9a4647ae50098efd8e7e7745da6f6e2a0a5dc",
3  "address": "BwbYhNJjd2HQhLfEWrxpiZ6nADA9PNMf2k",
4  "wif": "BpT2tqdYjkfF5TkNHSVD6vq2Sj1E2hVZ2KQCTfJ2hf8k9EkxbzQR"

```

3.2.3 在BCY网站上为Bob领取测试币

使用指令

```

1  curl -d '{"address": "BOB_BCY_ADDRESS", "amount": 1000000}'
    https://api.blockcypher.com/v1/bcy/test/faucet?
    token=3cc99a91b23843d88d79e37f3ec3076a

```

得到hash值

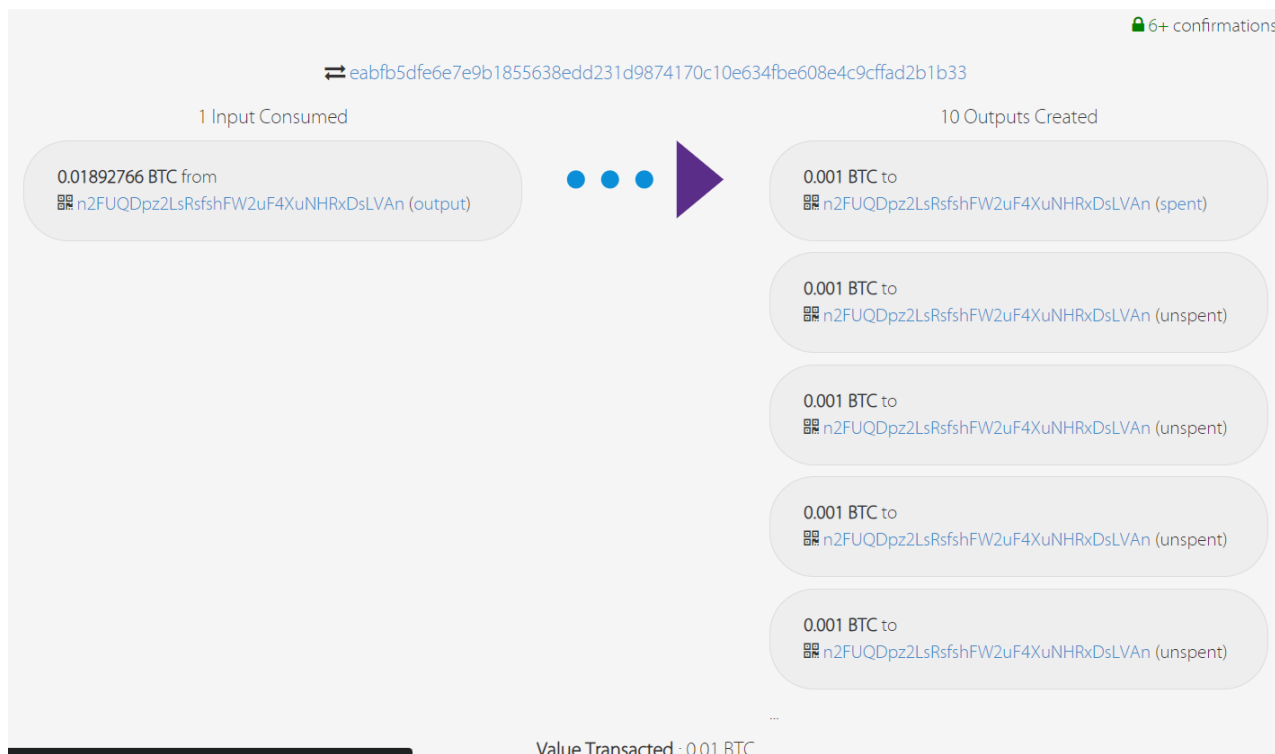
```

1  "tx_ref":
    "21f436a219c6640ec4c1c4186546f27ce6542ac8253922d4ac39b187adee5fb5"

```

3.3 划分币

我们使用split_test_coins.py将上述领取的Alice BTC币划分为10份。



对应response:

```

1 201 Created
2 {
3   "tx": {
4     "block_height": -1,
5     "block_index": -1,
6     "hash":
7     "eabfb5dfe6e7e9b1855638edd231d9874170c10e634fbe608e4c9cffad2b1b33",
8     "addresses": [
9       "n2FUQDpz2LsRsfsFW2uF4XuNHRxDsLVAn"
10    ],
11    "total": 1000000,
12    "fees": 892766,
13    "size": 497,
14    "vsize": 497,
15    "preference": "high",
16    "relayed_by": "221.238.245.25",
17    "received": "2023-11-13T02:41:58.220897667Z",
18    "ver": 1,
19    "double_spend": false,
20    "vin_sz": 1,
21    "vout_sz": 10,
22    "confirmations": 0,
23    "inputs": [
24      {
25        "prev_hash":
26        "80a78ff0d5915e6543c6605cb0812697b10a99e5c818f3463df0b3b2e607ee01",

```

```

25     "output_index": 0,
26     "script":
    "47304402202a2aada9f0ce0081979ce0c679a76fb8c135bb862101d61f789d4256e6b89c
    ee02205e35c7777b753ad10169b518a5c58993ad3a6c90ad9d2f6f11591a62e975bee4012
    10253e55d2f17f6c06039fbe9393c9e427329dbd4218090f3d2565cf8ffd22fca70",
27     "output_value": 1892766,
28     "sequence": 4294967295,
29     "addresses": [
30         "n2FUQDpz2LsRsfsFW2uF4XuNHRxDsLVAn"
31     ],
32     "script_type": "pay-to-pubkey-hash",
33     "age": 2538200
34 }
35 ],
36 "outputs": [
37     {
38         "value": 100000,
39         "script": "76a914e36d1e19cbc3f7a2da36e345d2a5940223a0f14088ac",
40         "addresses": [
41             "n2FUQDpz2LsRsfsFW2uF4XuNHRxDsLVAn"
42         ],
43         "script_type": "pay-to-pubkey-hash"
44     },
45     {
46         "value": 100000,
47         "script": "76a914e36d1e19cbc3f7a2da36e345d2a5940223a0f14088ac",
48         "addresses": [
49             "n2FUQDpz2LsRsfsFW2uF4XuNHRxDsLVAn"
50         ],
51         "script_type": "pay-to-pubkey-hash"
52     },
53     {
54         "value": 100000,
55         "script": "76a914e36d1e19cbc3f7a2da36e345d2a5940223a0f14088ac",
56         "addresses": [
57             "n2FUQDpz2LsRsfsFW2uF4XuNHRxDsLVAn"
58         ],
59         "script_type": "pay-to-pubkey-hash"
60     },
61     {
62         "value": 100000,
63         "script": "76a914e36d1e19cbc3f7a2da36e345d2a5940223a0f14088ac",
64         "addresses": [
65             "n2FUQDpz2LsRsfsFW2uF4XuNHRxDsLVAn"
66         ],
67         "script_type": "pay-to-pubkey-hash"

```

```

68     },
69     {
70         "value": 100000,
71         "script": "76a914e36d1e19cbc3f7a2da36e345d2a5940223a0f14088ac",
72         "addresses": [
73             "n2FUQDpz2LsRsfshFW2uF4XuNHRxDsLVAn"
74         ],
75         "script_type": "pay-to-pubkey-hash"
76     },
77     {
78         "value": 100000,
79         "script": "76a914e36d1e19cbc3f7a2da36e345d2a5940223a0f14088ac",
80         "addresses": [
81             "n2FUQDpz2LsRsfshFW2uF4XuNHRxDsLVAn"
82         ],
83         "script_type": "pay-to-pubkey-hash"
84     },
85     {
86         "value": 100000,
87         "script": "76a914e36d1e19cbc3f7a2da36e345d2a5940223a0f14088ac",
88         "addresses": [
89             "n2FUQDpz2LsRsfshFW2uF4XuNHRxDsLVAn"
90         ],
91         "script_type": "pay-to-pubkey-hash"
92     },
93     {
94         "value": 100000,
95         "script": "76a914e36d1e19cbc3f7a2da36e345d2a5940223a0f14088ac",
96         "addresses": [
97             "n2FUQDpz2LsRsfshFW2uF4XuNHRxDsLVAn"
98         ],
99         "script_type": "pay-to-pubkey-hash"
100     },
101     {
102         "value": 100000,
103         "script": "76a914e36d1e19cbc3f7a2da36e345d2a5940223a0f14088ac",
104         "addresses": [
105             "n2FUQDpz2LsRsfshFW2uF4XuNHRxDsLVAn"
106         ],
107         "script_type": "pay-to-pubkey-hash"
108     },
109     {
110         "value": 100000,
111         "script": "76a914e36d1e19cbc3f7a2da36e345d2a5940223a0f14088ac",
112         "addresses": [
113             "n2FUQDpz2LsRsfshFW2uF4XuNHRxDsLVAn"

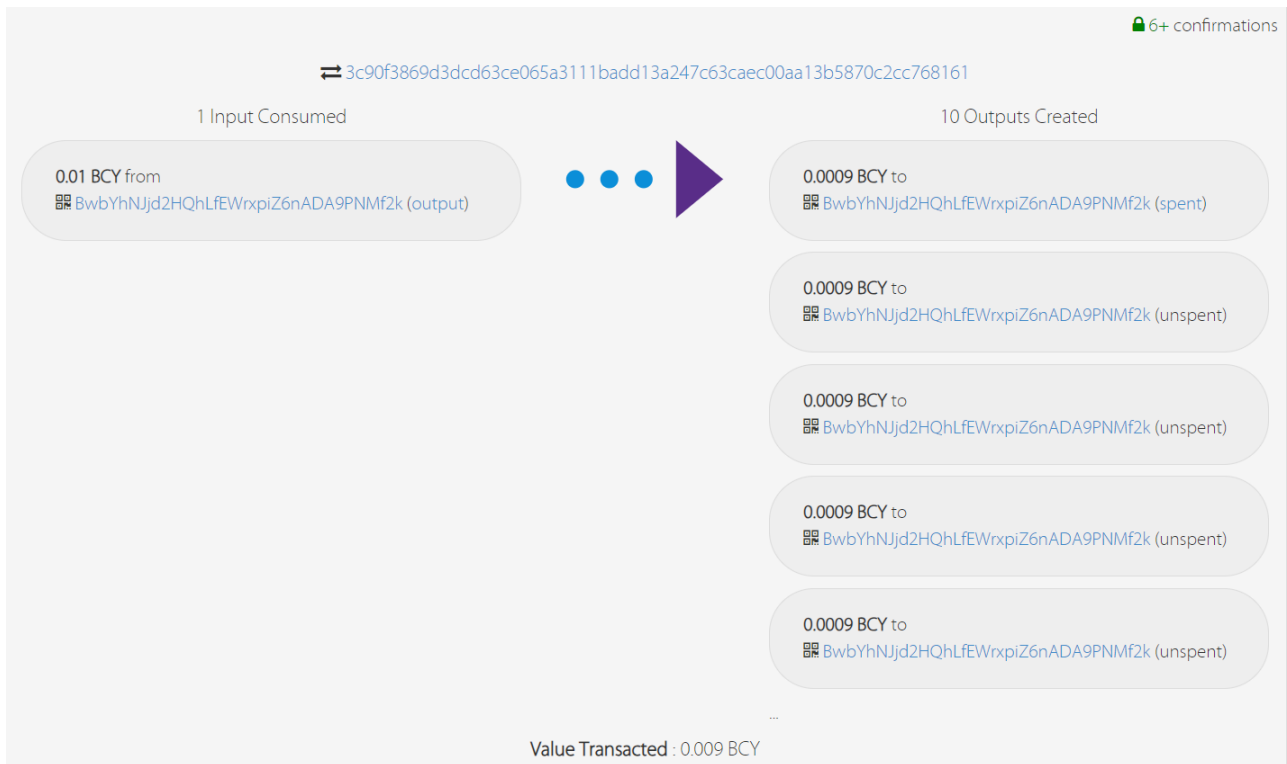
```

```

114     ],
115     "script_type": "pay-to-pubkey-hash"
116 }
117 ]
118 }
119 }
120

```

同样地，划分Bob的BCY币为10份：



```

1 201 Created
2 {
3   "tx": {
4     "block_height": -1,
5     "block_index": -1,
6     "hash":
7     "3c90f3869d3dcd63ce065a3111badd13a247c63caec00aa13b5870c2cc768161",
8     "addresses": [
9       "BwbYhNJd2HQhLfEWrxpiZ6nADA9PNMf2k"
10    ],
11    "total": 900000,
12    "fees": 100000,
13    "size": 497,
14    "vsize": 497,
15    "preference": "high",
16    "relayed_by": "221.238.245.25",
17    "received": "2023-11-13T02:43:59.016301481Z",
18    "ver": 1,

```

```

18     "double_spend": false,
19     "vin_sz": 1,
20     "vout_sz": 10,
21     "confirmations": 0,
22     "inputs": [
23         {
24             "prev_hash":
25             "21f436a219c6640ec4c1c4186546f27ce6542ac8253922d4ac39b187adee5fb5",
26             "output_index": 0,
27             "script":
28             "47304402203be5be84b0892dc1aea4095d7f49b5735f7530cd66ae00562fdc72ef17a64c
29             050220039a9a43bea7382a59b56be9cc719f28bece55de8d64cfd41af177225ee9045012
30             10234e56b7c63bee9fbef88186a2ac9a4647ae50098efd8e7e7745da6f6e2a0a5dc",
31             "output_value": 1000000,
32             "sequence": 4294967295,
33             "addresses": [
34                 "BwbYhNJjd2HQhLfEWrxpiZ6nADA9PNMf2k"
35             ],
36             "script_type": "pay-to-pubkey-hash",
37             "age": 1065779
38         }
39     ],
40     "outputs": [
41         {
42             "value": 90000,
43             "script": "76a9143108591cbc4ce4680ceeddedce861b29c901f8ce88ac",
44             "addresses": [
45                 "BwbYhNJjd2HQhLfEWrxpiZ6nADA9PNMf2k"
46             ],
47             "script_type": "pay-to-pubkey-hash"
48         },
49         {
50             "value": 90000,
51             "script": "76a9143108591cbc4ce4680ceeddedce861b29c901f8ce88ac",
52             "addresses": [
53                 "BwbYhNJjd2HQhLfEWrxpiZ6nADA9PNMf2k"
54             ],
55             "script_type": "pay-to-pubkey-hash"
56         },
57         {
58             "value": 90000,
59             "script": "76a9143108591cbc4ce4680ceeddedce861b29c901f8ce88ac",
60             "addresses": [
61                 "BwbYhNJjd2HQhLfEWrxpiZ6nADA9PNMf2k"
62             ],
63             "script_type": "pay-to-pubkey-hash"
64         }
65     ]
66 }

```



```

60 },
61 {
62     "value": 90000,
63     "script": "76a9143108591cbc4ce4680ceeddedce861b29c901f8ce88ac",
64     "addresses": [
65         "BwbYhNJjd2HQhLfEWrxpiZ6nADA9PNMf2k"
66     ],
67     "script_type": "pay-to-pubkey-hash"
68 },
69 {
70     "value": 90000,
71     "script": "76a9143108591cbc4ce4680ceeddedce861b29c901f8ce88ac",
72     "addresses": [
73         "BwbYhNJjd2HQhLfEWrxpiZ6nADA9PNMf2k"
74     ],
75     "script_type": "pay-to-pubkey-hash"
76 },
77 {
78     "value": 90000,
79     "script": "76a9143108591cbc4ce4680ceeddedce861b29c901f8ce88ac",
80     "addresses": [
81         "BwbYhNJjd2HQhLfEWrxpiZ6nADA9PNMf2k"
82     ],
83     "script_type": "pay-to-pubkey-hash"
84 },
85 {
86     "value": 90000,
87     "script": "76a9143108591cbc4ce4680ceeddedce861b29c901f8ce88ac",
88     "addresses": [
89         "BwbYhNJjd2HQhLfEWrxpiZ6nADA9PNMf2k"
90     ],
91     "script_type": "pay-to-pubkey-hash"
92 },
93 {
94     "value": 90000,
95     "script": "76a9143108591cbc4ce4680ceeddedce861b29c901f8ce88ac",
96     "addresses": [
97         "BwbYhNJjd2HQhLfEWrxpiZ6nADA9PNMf2k"
98     ],
99     "script_type": "pay-to-pubkey-hash"
100 },
101 {
102     "value": 90000,
103     "script": "76a9143108591cbc4ce4680ceeddedce861b29c901f8ce88ac",
104     "addresses": [
105         "BwbYhNJjd2HQhLfEWrxpiZ6nADA9PNMf2k"

```

```

106         ],
107         "script_type": "pay-to-pubkey-hash"
108     },
109     {
110         "value": 90000,
111         "script": "76a9143108591cbc4ce4680ceeddedce861b29c901f8ce88ac",
112         "addresses": [
113             "BwbYhNJjd2HQhLfEWrxpiZ6nADA9PNMf2k"
114         ],
115         "script_type": "pay-to-pubkey-hash"
116     }
117 ]
118 }
119 }

```

3.4 完善脚本

3.4.1 coinExchangeScript

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。

完善 swap_scripts.py 中的脚本 coinExchangeScript。

```

1 def coinExchangeScript(public_key_sender, public_key_recipient,
2   hash_of_secret):
3     return [
4         # fill this in!
5         #匹配是否包含接收的签名
6         public_key_recipient, #接受者的公钥放入脚本当中
7         OP_CHECKSIGVERIFY, #验证公钥是否有效
8         #复制栈顶的元素，因为要进行两种判断
9         OP_DUP,
10        #检查是不是发送者的签名
11        public_key_sender,
12        OP_CHECKSIG,
13
14        OP_IF, #如果发送者的签名有效，就执行OP_DROP，栈中移除一个元素
15        OP_DROP,
16        OP_1, #压栈1，代表True
17
18        OP_ELSE, #如果不成立
19        OP_HASH160, #计算提供秘密的哈希值
20        hash_of_secret,
21        OP_EQUAL, #比较秘密的哈希值

```

```

21 |         OP_ENDIF
22 |     ]

```

3.4.2 coinExchangeScriptSig1

在接收者知道秘密 x 的情况下，编写赎回交易所需的 ScriptSig。在 swap_scripts.py 中完善 coinExchangeScriptSig1。

```

1 | def coinExchangeScriptSig1(sig_recipient, secret):
2 |     return [
3 |         secret,
4 |         sig_recipient
5 |         # fill this in!包含了交易中的秘密和接受者的签名
6 |     ]

```

3.4.3 coinExchangeScriptSig2

在发送方和接收方都签署事务的情况下，编写赎回事务所需的 ScriptSig。在 swap_scripts.py 中完善 coinExchangeScriptSig2

```

1 | def coinExchangeScriptSig2(sig_sender, sig_recipient):
2 |     return [
3 |         sig_sender,
4 |         sig_recipient
5 |         # fill this in!
6 |     ]

```

4 跨链原子交换设计文档

4.1 解释你写的代码内容，以及 coinExchangeScript 是如何工作的

- coinExchangeScript

```

1 | def coinExchangeScript(public_key_sender, public_key_recipient,
2 |     hash_of_secret):
3 |     return [
4 |         # fill this in!
5 |         #匹配是否包含接收的签名
6 |         public_key_recipient, #接受者的公钥放入脚本当中
7 |         OP_CHECKSIGVERIFY, #验证公钥是否有效
8 |         #复制栈顶的元素，因为要进行两种判断
9 |         OP_DUP,
10 |        #检查是不是发送者的签名
11 |        public_key_sender,
12 |        OP_CHECKSIG,
13 |
14 |        OP_IF, #如果发送者的签名有效，就执行OP_DROP，栈中移除一个元素

```

```

14         OP_DROP,
15         OP_1, #压栈1, 代表True
16
17         OP_ELSE, #如果不成立
18         OP_HASH160, #计算提供秘密的哈希值
19         hash_of_secret,
20         OP_EQUAL, #比较秘密的哈希值
21         OP_ENDIF
22     ]

```

`coinExchangeScript` 函数生成的是跨链原子交换中使用的 ScriptPubKey，这是一个定义资产如何被赎回的条件脚本。下面是该脚本的组件及其工作原理：

1. `public_key_recipient`：接收者的公钥。这是用于验证接收者签名的关键部分。
2. `OP_CHECKSIGVERIFY`：这是一个操作码，用于验证栈顶部的签名是否与所提供的公钥匹配。如果签名有效，则继续执行脚本；如果无效，则脚本执行失败。
3. `OP_DUP`：这个操作码复制栈顶的元素。在这个脚本中，它用于保留一个公钥的副本，以便后续步骤中进行比较。
4. `public_key_sender`：发送者的公钥。这是用于验证发送者签名的另一个关键部分。
5. `OP_CHECKSIG`：又一个操作码，用于验证栈顶的签名是否与所提供的公钥（这次是发送者的公钥）匹配。
6. `OP_IF - OP_DROP - OP_1 - OP_ELSE - OP_HASH160 - hash_of_secret - OP_EQUAL - OP_ENDIF`：这一系列操作码实现了脚本的**逻辑分支**。它们定义了两种**赎回条件**：
 - 如果发送者的签名有效（由 `OP_CHECKSIG` 验证），则执行 `OP_DROP` 操作移除栈顶元素，然后 `OP_1`（代表真值）被推送到栈顶，表示交易条件满足。
 - 如果不是发送者的签名，脚本则检查提供的秘密是否有效。这是通过计算提供的秘密的哈希值（`OP_HASH160`），并将其与预先定义的 `hash_of_secret` 进行比较（`OP_EQUAL`）。如果它们相等，表示秘密正确，交易条件也满足。

这个脚本允许资产在两种情况下被赎回：一是**接收者提供了正确的秘密**（即知道了秘密x的值），二是**发送者和接收者共同签名**（即双方共同决定取消交易或修改其条款）。这种设计使得资产的交换既安全又灵活，保证了只有在满足特定条件时，资产才能被转移。

- `coinExchangeScriptSig1`

```

1 def coinExchangeScriptSig1(sig_recipient, secret):
2     return [
3         secret,
4         sig_recipient
5         # fill this in!包含了交易中的秘密和接受者的签名
6     ]

```

这个函数生成的是接收者用来赎回资产的脚本签名（ScriptSig）。其工作原理如下：

- `secret`：这是之前提到的秘密x的值。在整个跨链交换过程中，这个秘密是交换的核心。当接收者知道了这个秘密并准备好公开它时，他们将使用这个值来解锁交易。
- `sig_recipient`：这是接收者的数字签名。这个签名证明了交易是由资产的合法接收者发起的。

当这两个元素（秘密和接收者的签名）被提供时，它们满足了 `coinExchangeScript` 中定义的条件，允许接收者赎回资产。具体来说，它们匹配了脚本中定义的哈希值和签名验证条件。

- `coinExchangeScriptSig2`

```

1 def coinExchangeScriptSig2(sig_sender, sig_recipient):
2     return [
3         sig_sender,
4         sig_recipient
5         # fill this in!
6     ]

```

这个函数生成的是用于将资产返回给发送者的情况下的脚本签名。其工作原理如下：

- `sig_sender`：这是发送者的数字签名。这个签名用于证明交易是由资产的原始所有者发起的。
- `sig_recipient`：这也是接收者的数字签名。在这种情况下，接收者的签名是为了证明他们同意将资产返回给发送者。

当这两个签名都被提供时，它们满足了 `coinExchangeScript` 中的另一个条件，即允许双方共同签名来解锁资产。这通常用于交换未能按计划进行时，将资产安全退回给发送者的情况。

总结来说，`coinExchangeScriptSig1` 和 `coinExchangeScriptSig2` 分别对应了跨链原子交换中的两个主要场景：一是接收者赎回资产，二是交易取消时资产返回给发送者。这两个脚本签名的实现确保了交易的安全性和条件的满足。

4.2 以 Alice 用 coinExchangeScript 向 Bob 发送硬币为例

4.2.1 如果 Bob 不把钱赎回来，Alice 为什么总能拿回她的钱？

在这个过程中，Alice 首先创建了一笔支付BTC币的交易I，并将这笔交易记录在 `alice_swap_tx` 中。然后，她根据交易I的哈希值创建了超时赎回交易II，在一定时间后如果 Bob 不取走这笔钱，交易II会使资金返回给 Alice。最后，Alice 邀请 Bob 对交易II进行签名。如果 Bob 不签名，交易I将永远不会被广播，资金将始终保留在 Alice 手中。如果 Bob 签名，那么 Alice

就拥有了自己和Bob双方的签名，交易I将被广播，资金就不再有Alice或Bob的控制之下。

即使Bob没有提供与Alice交换的BCY币的交易，或者Alice没有赎回Bob存储在BCY链上的存款，导致交易未达成，Bob也无法取走交易I中的币。因此，在一定时间后，交易II生效，Alice将其广播，资金就能回到Alice手中。

这个过程的关键在于，在整个交换过程中，Alice能够确保只有在交易达成并且Bob履行了他的承诺时，才会释放资金。这种原子交换的机制确保了交易的安全性和可靠性，同时避免了信任第三方的需求。

4.2.2 为什么不能用简单的 1/2 multisig 来解决这个问题？

如果使用1/2multisig，只需要一个人的签名就能把钱赎回来。如此一来，Bob 就可以在Alice创建交易后，单独签名，“偷走”Alice的币。

4.3 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

[1]Alice创建一个秘密 x，并对该秘密进行hash计算

[2]Alice创建交易I，通过交易I，Alice可以将自己的BTC的币给Bob，但是Alice并不会广播出去。为了防止Alice单方面地赎回交易I，该交易需要Bob的签名才能赎回。并且，为了确保Alice在交易条件未达成时能够赎回自己的BTC，Alice必须得到Bob签名的延迟赎回脚本II才能将交易I广播。创建交易II，就必须使用交易I的哈希值，所以就得先让Alice创建交易I。

[3]Alice创建交易II之后的一段时间，如果Bob没有取走这笔钱，那么之前的交易I发出的BTC就会返回到Alice自己的账户上。

[4]Alice让Bob在交易II上面签名

[5]签名之后，Alice将交易I广播。这一时刻，Alice的BTC不在二人手中。

[6]Bob创建交易III将自己的BCY转发给Alice，其中包括交易I中存在的秘密 x 的哈希值，同样的先不广播

[7]Bob创建交易IV。一定时间后（小于交易II设置的时间），如果Alice没有取走这笔钱，之前发出的BCY就会返回至自己的账户中。

[8]Bob让Alice在交易IV上面签名。

[9]签名之后，Bob才会将交易III广播。同理，这一时刻，BCY不在双方手中。

[10]Alice使用自己签名的存有秘密 x 的交易在规定时间内取走交易III中的钱(新的交易V)。这一时刻，秘密 x 机会被揭露。【如果，Alice不去走交易III中的钱的话，一旦超过规定时间，Bob和Alice将会依次将交易II、IV广播，这个时候，双方就可以使双方的签名取回自己的币。】

[11]Bob使用自己的签名的包含秘密 `x` 的交易I中的钱取走。这个时候，依次跨链原子交易就完成了。

4.4 以该作业为例，一次成功的跨链原子交换中，数字货币是如何流转的？如果失败，数字货币又是如何流转的？

在(C)题当中，描述了一次完整的跨链原子交换。

在前四步骤，双方还都持有各自的币。

如果第四步成功，Alice就不持有自己的币了，当然的，Bob持有自己的币，但不持有Alice的币

第六到第八步时，Alice就不持有自己的币了，当然的，Bob持有自己的币，但不持有Alice的币

如果第九步成功，Alice和Bob既不持有自己的币，也不持有对方的币。

如果第十步成功，Alice就得到了Bob的币了。

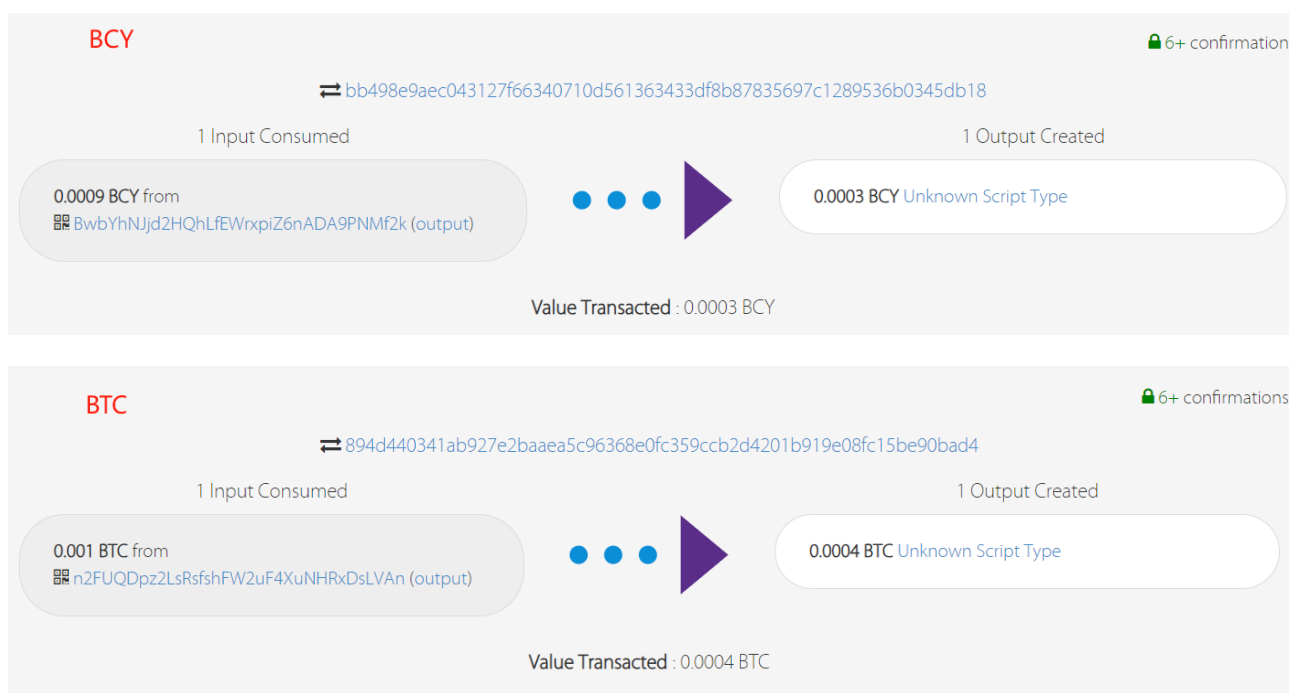
如果第十一步成功，Bob就得到了Alice的币了。

如果在任何关键步骤中，比如秘密x未按时公开或某一方未执行必要的交易，交换将失败。在这种情况下，由于设置了时间锁，资产将保留在原始所有者的控制下。对于 Alice 和 Bob 来说，如果交换失败，他们的资产将通过时间锁自动返回到他们各自的账户中。这种机制确保了即使交易未能完成，双方的资产也不会丢失。

5 实验结果

跨链原子交换的结果：

- `broadcast_transactions = True`




```

1 python3 '/home/linux/Documents/blockchain/lab4/swap.py'
2 Alice swap tx (BTC) created successfully!
3 201 Created
4 {
5     "tx": {
6         "block_height": -1,
7         "block_index": -1,
8         "hash":
9         "0a3987a0dd61cceb55c98aa36eaff5dc4e286bae3e1fc23e0e56ffcf52068ac9",
10        "addresses": [
11            "n2FUQDpz2LsRsfshFW2uF4XuNHRxDsLVAn"
12        ],
13        "total": 40000,
14        "fees": 60000,
15        "size": 265,
16        "vsize": 265,
17        "preference": "high",
18        "relayed_by": "221.238.245.25",
19        "received": "2023-11-13T04:45:42.778956031Z",
20        "ver": 1,
21        "double_spend": false,
22        "vin_sz": 1,
23        "vout_sz": 1,
24        "confirmations": 0,
25        "inputs": [
26            {
27                "prev_hash":
28                "eabfb5dfe6e7e9b1855638edd231d9874170c10e634fbe608e4c9cffad2b1b33",
29                "output_index": 1,
30                "script":
31                "47304402203c53f3f2e8af59c5eb193a653961335b881c568814f4cadd03e19f5dbd5e521
32                c0220285f5688ead8ff1eccbf392445e669ba93708d47456cedd4c44769396a42603501210
33                253e55d2f17f6c06039fbe9393c9e427329dbd4218090f3d2565cf8ffd22fca70",
34                "output_value": 100000,
35                "sequence": 4294967295,
36                "addresses": [
37                    "n2FUQDpz2LsRsfshFW2uF4XuNHRxDsLVAn"
38                ],
39                "script_type": "pay-to-pubkey-hash",
40                "age": 2538201
41            }
42        ],
43        "outputs": [
44            {
45                "value": 40000,

```



```

41         "script":
"2103ff55c4e48807009d94d0b3f639e26da4ec9d1cb46ed00a269c0dbe2f9b214bbdad762
10253e55d2f17f6c06039fbe9393c9e427329dbd4218090f3d2565cf8ffd22fca70ac63755
167a914853b775079232503df966e626618e1d388a957208768",
42         "addresses": null,
43         "script_type": "unknown"
44     }
45 ]
46 }
47 }
48 Bob swap tx (BCY) created successfully!
49 201 Created
50 {
51     "tx": {
52         "block_height": -1,
53         "block_index": -1,
54         "hash":
"e0900f87c2ce44c2e1b18b9f5e7fdf781c3bc08ac148c162f02482adc7b6778a",
55         "addresses": [
56             "BwbYhNJjd2HQhLfEWrxpiZ6nADA9PNMf2k"
57         ],
58         "total": 30000,
59         "fees": 60000,
60         "size": 266,
61         "vsize": 266,
62         "preference": "high",
63         "relayed_by": "221.238.245.25",
64         "received": "2023-11-13T04:45:45.134647543Z",
65         "ver": 1,
66         "double_spend": false,
67         "vin_sz": 1,
68         "vout_sz": 1,
69         "confirmations": 0,
70         "inputs": [
71             {
72                 "prev_hash":
"3c90f3869d3dcd63ce065a3111badd13a247c63caec00aa13b5870c2cc768161",
73                 "output_index": 1,
74                 "script":
"48304502210088afaa9c78618cbbf4e6f5077665e884754964c1d10afe0b0af16f7018a9e
3c202201a0dbc2b7a509d29755529e09f7b9dd937797ae8d7b5e0e6b26768aaadc5fed012
10234e56b7c63bee9fbef88186a2ac9a4647ae50098efd8e7e7745da6f6e2a0a5dc",
75                 "output_value": 90000,
76                 "sequence": 4294967295,
77                 "addresses": [
78                     "BwbYhNJjd2HQhLfEWrxpiZ6nADA9PNMf2k"

```

```

79         ],
80         "script_type": "pay-to-pubkey-hash",
81         "age": 1065785
82     }
83 ],
84 "outputs": [
85     {
86         "value": 30000,
87         "script":
            "2102af90ced1a0e21f7abf6dfa72c6acd2cc9365419b9b2c91bd885bc4fae7c5c5b2ad762
            10234e56b7c63bee9fbef88186a2ac9a4647ae50098efd8e7e7745da6f6e2a0a5dcac63755
            167a914853b775079232503df966e626618e1d388a957208768",
88         "addresses": null,
89         "script_type": "unknown"
90     }
91 ]
92 }
93 }
94
95 Sleeping for 20 minutes to let transactions confirm...
96 Bob return coins (BCY) tx created successfully!
97 Alice return coins tx (BTC) created successfully!
98 Sleeping for blocks to pass locktime...

```

- `broadcast_transactions = False`

```

(base) PS D:\study\大三\区块链\实验\Ex4> & D:/ProgramData/Anaconda3/python.exe d:/study/大三/区块链/实验/Ex4/swap.py
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!
○ (base) PS D:\study\大三\区块链\实验\Ex4>

```