南開大學

区块链实验四

目录

1 说明 **4**

1.1 小组成员 4

1.2 仓库 4

2 实验目的 **4**

3 实验原理 **5**

3.1 核心技术基础：哈希时间锁定合约（HTLC） 5

3.2 跨链原子交换工作流程（Alice/BTC、Bob/BCY） 5

3.2.1 coinExchangeScriptSig1（接收者用 x 赎回） 6

3.2.2 coinExchangeScriptSig2（双方签名赎回） 7

3.2.3 本地验证方法 7

4 设计文档 **7**

4.1 代码内容及 coinExchangeScript 工作原理 7

4.2 以 Alice 用 coinExchangeScript 向 Bob 发送硬币为例 9

4.2.1 Alice 如何确保拿回她的钱 9

4.2.2 为什么不能用简单的 1/2 multisig 解决 9

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 10

4.3.1 交易顺序概览 10

4.3.2 设计原理 11

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？ 11

4.4.1 成功情况下的数字货币流转 11

4.4.2 失败情况下的数字货币流转 12

4.4.3 总结 12

5 实验过程与结果分析 **12**

5.1 分币操作 12

5.2 交易操作 15

5.2.1 情况一 15

5.2.2 情况二 15

5.2.3 情况三 16
5.2.4 情况四 20

6 心得体会 **24**

1 说明

1.1 小组成员

程伟卿	2311865
田园	2313660

表 1: 成员

1.2 仓库

点击跳转

2 实验目的

核心功能实现：创建跨链原子交换交易代码，支持 Alice 与 Bob 在不同区块链（BTC Testnet3 和 BCY Testnet）间安全交换加密货币所有权，解决不同链上资产无法通过简单交易直接交换的问题。

关键脚本开发：完善 swap_scripts.py 中的核心脚本，包括：

- **coinExchangeScript：**构建跨链原子交换所需的 ScriptPubKey，确保交易可通过“秘密 x 验证”或“双方签名”两种方式赎回。
- **coinExchangeScriptSig1：**编写接收者（如 Bob）已知秘密 x 时，赎回交易所需的 ScriptSig。
- **coinExchangeScriptSig2：**编写发送者（如 Alice）与接收者（如 Bob）共同签名时，赎回交易所需的 ScriptSig。

代码验证与测试：

- **本地验证交易合法性：**运行 swap.py 并设置 broadcast_transactions=False，验证 ScriptSig + ScriptPK 是否返回 true，且分别测试 alice_redeems=True 和 alice_redeems=False 两种场景。
- **可选链上验证：**设置 broadcast_transactions=True，将交易发布到区块链，等待 20-60 分钟及以上，验证跨链交换全流程可行性。

3 实验原理

3.1 核心技术基础：哈希时间锁定合约（HTLC）

实验基于“哈希时间锁定合约（Hash Time-Locked Contracts, HTLC）”设计，核心围绕“秘密 x”与“哈希值 H (x)”展开，确保交易要么“双方成功交换资产”，要么“各自取回原始资产”，无中间风险状态。

3.2 跨链原子交换工作流程（Alice/BTC、Bob/BCY）

1. **准备阶段：密钥与测试币配置**

   - **生成密钥：**为 Alice/Bob 分别创建 BTC Testnet3 和 BCY Testnet 的密钥，填入 keys.py。
   - **获取测试币：**

     Alice 通过 https://coinfaucet.eu/en/btc-testnet/获取 BTC Testnet3 测试币。

     Bob 通过 Blockcypher API(需注册获取 token),调用 https://api.blockcypher.com/v1/bcy/test/ 获取 BCY Testnet 测试币。
   - **划分测试币：**运行 split_test_coins.py（填写相关字段），拆分领取的测试币，用于后续交易。

2. **核心交易逻辑：基于 coinExchangeScript 的双向锁定**

   跨链交换需 Alice 和 Bob 分别在各自链上创建“锁定交易”, 并通过“赎回交易”完成资产转移, 核心依赖 coinExchangeScript（ScriptPubKey）实现两种赎回路径：

赎回场景	触发条件	实现逻辑
路径 1：接收者用秘密 x 赎回	接收者（如 Bob）已知秘密 x	接收者提供 x，脚本验证 $H(x)$ 与预设哈希值一致 → 验证通过，接收者获得资产
路径 2:双方签名赎回(或超时取回)	秘密 x 未被揭露（或交易异常）	发送者（如 Alice）与接收者（如 Bob）共同签名 → 验证通过，发送者取回原始资产（避免资产永久锁定）

表 2: coinExchangeScript 赎回路径对比

3. **交易执行顺序与资产流转**

步骤 1：Alice 创建 BTC 链锁定交易

Alice 在 BTC Testnet3 上发起交易，将自己的 BTC 锁定到 `coinExchangeScript`（ScriptPubKey），该交易仅能通过两种方式解锁：

(a) Bob 提供秘密 x

(b) Alice 与 Bob 共同签名

步骤 2：Bob 创建 BCY 链锁定交易

Bob 在 BCY Testnet 上发起交易，将自己的 BCY 锁定到相同逻辑的 `coinExchangeScript`，解锁条件与 Alice 的 BTC 锁定交易一致（依赖同一秘密 x 的哈希值 $H(x)$）。

步骤 3：赎回阶段（成功场景）

(a) **Bob 通过" 路径 1" 赎回 BTC**：Bob 提供秘密 x，生成 `coinExchangeScriptSig1`，解锁 Alice 在 BTC 链的锁定交易，获得 BTC；此时秘密 x 被公开。

(b) **Alice 通过" 路径 1" 赎回 BCY**：Alice 使用 Bob 公开的 x，生成 `coinExchangeScriptSig1`，解锁 Bob 在 BCY 链的锁定交易，获得 BCY。

(c) **最终流转**：Alice 的 BTC → Bob，Bob 的 BCY → Alice。

步骤 4：赎回阶段（失败场景）

(a) 若 Bob 始终不提供秘密 x（不赎回 BTC），Alice 可通过" 路径 2"（与 Bob 共同签名）解锁自己在 BTC 链的锁定交易，取回 BTC。

(b) 同理，Bob 也可通过" 路径 2" 取回 BCY。

(c) **最终流转**：Alice 取回 BTC，Bob 取回 BCY，资产回归初始状态。

4. **关键脚本验证逻辑**

3.2.1 coinExchangeScriptSig1（接收者用 x 赎回）

- **输入**：包含" 秘密 x + 接收者签名"

- **验证流程**：脚本先验证 x 的哈希值是否与预设 $H(x)$ 一致，再验证接收者签名合法性
- **结果**：双重验证通过则解锁交易

3.2.2 coinExchangeScriptSig2（双方签名赎回）

- **输入**：包含” 发送者签名 + 接收者签名”
- **验证流程**：脚本验证双方签名均合法
- **结果**：验证通过后解锁交易

3.2.3 本地验证方法

运行 `swap.py` 时，通过查询 BTC（`https://live.blockcypher.com/btc-testnet/`）和 BCY（`https://live.blockcypher.com/bcy/`）的实时区块高度，代入代码后检查 `ScriptSig + ScriptPK` 的执行结果是否为 `true`，确保脚本逻辑无误。

4 设计文档

4.1 代码内容及 coinExchangeScript 工作原理

本项目实现了 Alice 和 Bob 之间的跨链原子交换。核心代码位于 `swap.py` 中，其逻辑如下：

1. **初始化参数**：
   - 设置 Alice 的 BTC UTXO 及分币索引 `alice_txid_to_spend`、`alice_utxo_index` 和金额 `alice_amount_to_send`。
   - 设置 Bob 的 BCY UTXO 及分币索引 `bob_txid_to_spend`、`bob_utxo_index` 和金额 `bob_amount_to_send`。
   - 设置当前区块高度，用于时间锁（locktime）：`btc_test3_chain_height` 和 `bcy_test_chain_height`。
   - 配置交易等待时间（以区块数计）：`alice_locktime` 和 `bob_locktime`。
   - 交易手续费 `tx_fee`，保证交易能够被矿工打包。

- 交易广播与赎回标记：`broadcast_transactions` 和 `alice_redeems`。

2. **核心交换流程**：

   (a) Alice 生成一个秘密 x，并计算其哈希值 $H(x)$。哈希值在交易中公开，但 x 保密。

   (b) Alice 构建一笔锁定 BTC 的交易 `alice_swap_tx`，可由 Bob 使用 x 赎回，或者由 Alice 与 Bob 的双签赎回。

   (c) Alice 创建一个超时返回交易 `alice_return_coins_tx`，当锁定时间超过 `alice_locktime` 后，允许她取回 BTC。

   (d) Bob 签署 Alice 的返回交易，并在 Alice 广播锁定交易后生效。

   (e) 角色互换，Bob 构建锁定 BCY 的交易 `bob_swap_tx` 及超时返回交易 `bob_return_coins_tx`，并请求 Alice 签署。

   (f) Alice 与 Bob 广播各自锁定交易，并等待一段时间以保证交易确认。

3. **赎回阶段**：

   - 当 `alice_redeems=True` 时：

     (a) Alice 使用秘密 x 赎回 BCY，交易 `alice_redeem_tx` 被广播。

     (b) Bob 获取 x 后，使用其赎回 BTC，交易 `bob_redeem_tx` 被广播。

   - 当 `alice_redeems=False` 时：

     (a) 超时后，双方可使用超时返回交易取回各自的原始资产。

coinExchangeScript 工作原理：

- `coinExchangeScript` 是原子交换的核心智能脚本，嵌入在锁定交易的输出中。
- 它实现了两种赎回条件：

  1. 接收者（例如 Bob 或 Alice）提供与哈希值 $H(x)$ 对应的秘密 x，即可单方赎回交易。

2. 发送者与接收者共同签署交易，可在未提供 x 的情况下赎回，确保资产安全。

- 时间锁（locktime）结合 coinExchangeScript 确保：
  - 如果交易正常进行，双方可使用 x 完成资产交换。
  - 如果交易中断，超时后发送者可安全取回原资产。

通过该脚本设计，Alice 与 Bob 可在不同区块链上安全地交换资产，而无需信任对方。

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

4.2.1 Alice 如何确保拿回她的钱

- Alice 构建的锁定交易使用了 `coinExchangeScript`，其中包含两种赎回条件：
  1. 接收者（Bob）提供秘密 x 赎回。
  2. 发送者（Alice）与接收者（Bob）共同签署赎回。
- 如果 Bob 不使用 x 赎回交易，交易将无法被单方完成。
- 为了避免资产被卡住，Alice 事先构建了一个 **超时返回交易（return transaction）**，该交易在 locktime 到期后允许她取回锁定的 BTC。
- 因此，即使 Bob 不赎回硬币，Alice 仍能在 locktime 后通过超时交易安全地收回自己的资产。

4.2.2 为什么不能用简单的 1/2 multisig 解决

- 简单的 1/2 多重签名要求 Alice 和 Bob 都签名才能完成交易。
- 在跨链原子交换中，双方在不同区块链上操作，如果仅用 1/2 multisig：
  1. Alice 无法单独取回锁定的资产，因为只有她签名是不够的。
  2. Bob 也无法单独取回资产，如果 Alice 不配合，交易会被卡住。
- 使用 `coinExchangeScript` 结合秘密 x 和时间锁机制，可以：

1. 允许单方赎回（通过秘密 x）。
2. 防止交易被对方恶意占用。
3. 提供超时安全保障，使原资产可返回。

- 因此，单纯的 1/2 multisig 无法保证在跨链环境中交易的安全性和原子性。

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

在跨链原子交换中，为保证交易的安全性和原子性，Alice 和 Bob 需要按严格的顺序创建和签署交易。本节将解释关键交易及其设计原理。

4.3.1 交易顺序概览

1. **Alice 创建锁定交易 I（swap transaction）**：
   - 用于锁定 Alice 的 BTC。
   - 交易条件由 `coinExchangeScript` 控制，可被 Bob 赎回（提供秘密 x）或 Alice/Bob 共同签署赎回。
2. **Alice 创建超时返还交易 II（return transaction）**：
   - 用于在 locktime 到期后返还 BTC 给 Alice。
   - 该交易需要 Bob 的签名才能生效，确保 Alice 不会被卡住。
3. **Bob 创建锁定交易 III（swap transaction）**：
   - 用于锁定 Bob 的 BCY。
   - 交易条件同样由 `coinExchangeScript` 控制，可被 Alice 赎回（提供秘密 x）或 Bob/Alice 共同签署赎回。
4. **Bob 创建超时返还交易 IV（return transaction）**：
   - 用于在 locktime 到期后返还 BCY 给 Bob。
   - 该交易需要 Alice 的签名。

4.3.2 设计原理

- **原子性保证**：
  - 所有交易围绕 Alice 的秘密 x 构建，确保只有在秘密揭示后，双方才能完成资产交换。
  - 如果其中一方不赎回，超时返还交易可保证原资产安全返回。
- **先后顺序的重要性**：
  - Alice 先创建并广播交易 I，确保她的 BTC 被锁定并可被 Bob 赎回。
  - 在 Bob 签署超时返还交易后，Alice 可放心广播交易 I。
  - Bob 创建交易 III 后，Alice 执行赎回交易，实现跨链资产交换。
- **时间锁机制**：
  - Alice 的 locktime 必须大于 Bob 的 locktime，以保证 Bob 在 Alice 赎回前无法取回资产。
  - 时间锁与秘密 x 结合，实现安全的原子交换。

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？

在本作业的跨链原子交换实验中，涉及两种数字货币：BTC（Alice）和 BCY（Bob）。交易流程围绕 Alice 的秘密 x，结合锁定交易与时间锁机制，实现原子交换。

4.4.1 成功情况下的数字货币流转

1. Alice 创建锁定交易 I，将 BTC 锁定，交易条件要求 Bob 提供秘密 x 才能赎回。
2. Bob 创建锁定交易 III，将 BCY 锁定，交易条件要求 Alice 提供秘密 x 才能赎回。
3. Alice 赎回交易 III（BCY），此时需要提供秘密 x。
4. Bob 观察到区块链上公开的 x 后，赎回交易 I（BTC）。

结果：

- Alice 成功获得 Bob 的 BCY。
- Bob 成功获得 Alice 的 BTC。
- 秘密 x 公开，交易原子性得到保证。

4.4.2 失败情况下的数字货币流转

1. 若 Alice 或 Bob 不进行赎回，交易将超时。
2. Alice 的 BTC 会在 locktime 到期后，通过交易 II 返回 Alice。
3. Bob 的 BCY 会在 locktime 到期后，通过交易 IV 返回 Bob。

结果：

- 双方均保留原有资产，没有任何损失。
- 原子性仍然保证，交易失败不会造成资产丢失。

4.4.3 总结

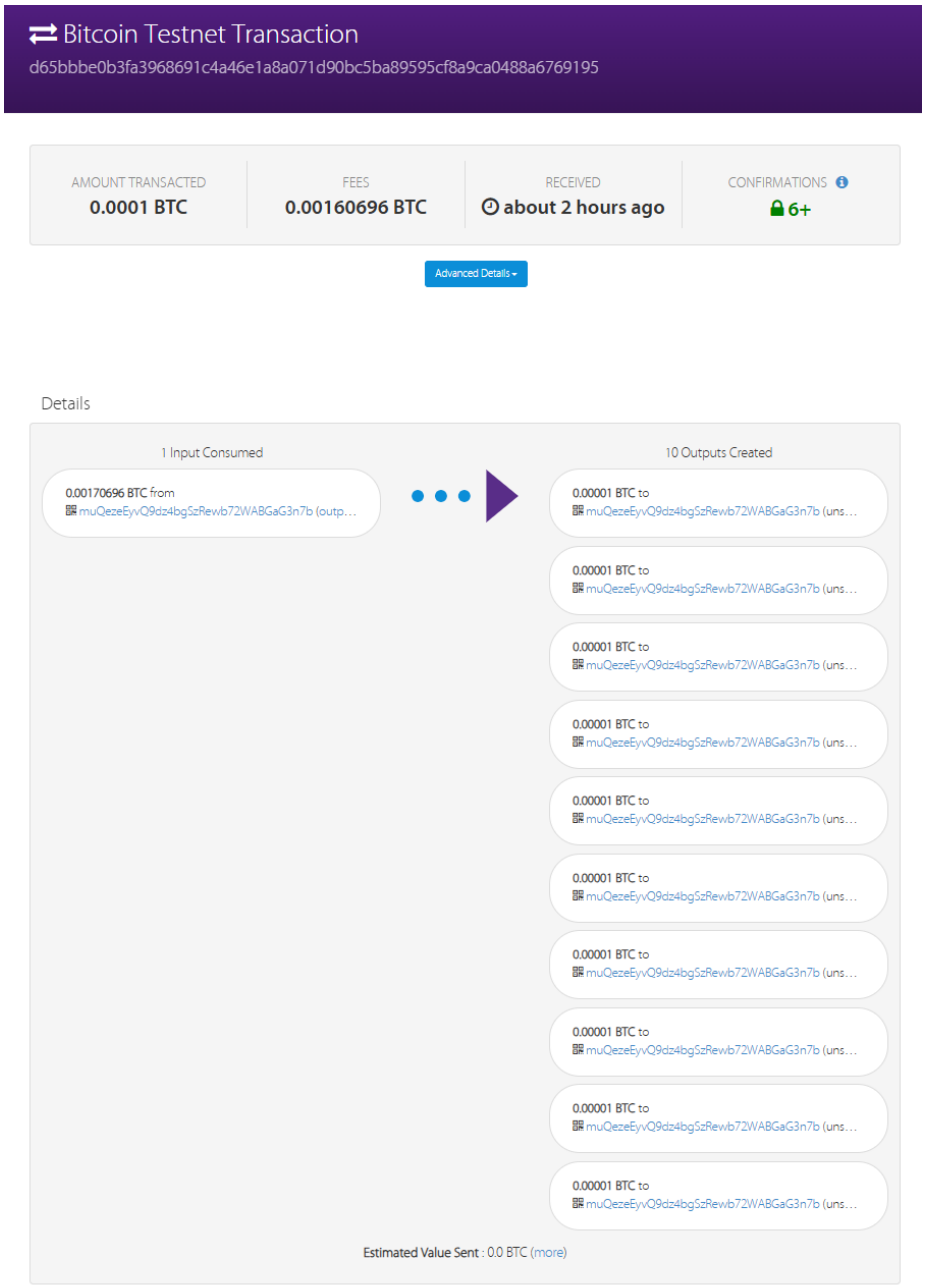
通过秘密 x 与时间锁结合，实现了：

- 成功交易时资产互换，且可追踪。
- 失败交易时资产安全返还，保证交易双方不会损失资金。

5 实验过程与结果分析

5.1 分币操作

按要求分别进行 BTC 和 BCY 分币，结果如下：



Bitcoin Testnet Transaction

d65bbbe0b3fa3968691c4a46e1a8a071d90bc5ba89595cf8a9ca0488a6769195

AMOUNT TRANSACTED	FEES	RECEIVED	CONFIRMATIONS
0.0001 BTC	0.00160696 BTC	about 2 hours ago	6+

Details

1 Input Consumed

0.00170696 BTC from
muQezeEyvQ9dz4bgSzRewb72WABGaG3n7b (outp…

10 Outputs Created

0.00001 BTC to
muQezeEyvQ9dz4bgSzRewb72WABGaG3n7b (uns…

0.00001 BTC to
muQezeEyvQ9dz4bgSzRewb72WABGaG3n7b (uns…

0.00001 BTC to
muQezeEyvQ9dz4bgSzRewb72WABGaG3n7b (uns…

0.00001 BTC to
muQezeEyvQ9dz4bgSzRewb72WABGaG3n7b (uns…

0.00001 BTC to
muQezeEyvQ9dz4bgSzRewb72WABGaG3n7b (uns…

0.00001 BTC to
muQezeEyvQ9dz4bgSzRewb72WABGaG3n7b (uns…

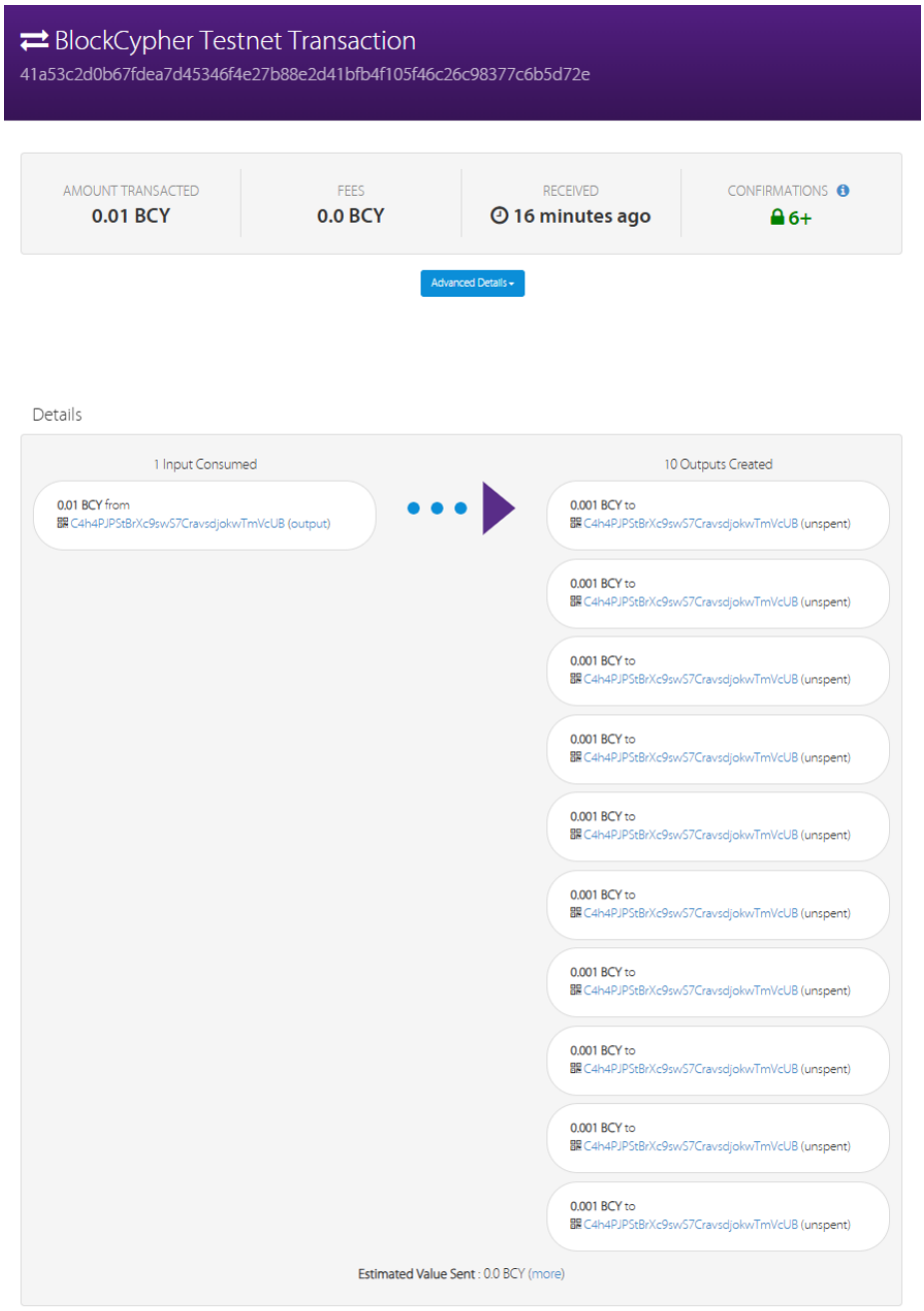
0.00001 BTC to
muQezeEyvQ9dz4bgSzRewb72WABGaG3n7b (uns…

0.00001 BTC to
muQezeEyvQ9dz4bgSzRewb72WABGaG3n7b (uns…

0.00001 BTC to
muQezeEyvQ9dz4bgSzRewb72WABGaG3n7b (uns…

0.00001 BTC to
muQezeEyvQ9dz4bgSzRewb72WABGaG3n7b (uns…

Estimated Value Sent : 0.0 BTC (more)


BlockCypher Testnet Transaction
41a53c2d0b67fdea7d45346f4e27b88e2d41bfb4f105f46c26c98377c6b5d72e
AMOUNT TRANSACTED
0.01 BCY
FEES
0.0 BCY
RECEIVED
16 minutes ago
CONFIRMATIONS
6+
Advanced Details
Details
1 Input Consumed
0.01 BCY from
C4h4PJPStBrXc9swS7CravsdjokwTmVcUB (output)
10 Outputs Created
0.001 BCY to
C4h4PJPStBrXc9swS7CravsdjokwTmVcUB (unspent)
Estimated Value Sent : 0.0 BCY (more)

5.2 交易操作

针对下面四种情况分别实验：

情况一	broadcast_transactions = False	alice_redeems = False
情况二	broadcast_transactions = False	alice_redeems = True
情况三	broadcast_transactions = True	alice_redeems = True
情况四	broadcast_transactions = True	alice_redeems = False

表 3: 四种运行 swap.py 的组合情况

5.2.1 情况一

运行结果如下：

```
(venv) john@John:~/john_libr/Blockchain2025/Ex4/code$ python 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!
```

这说明，Alice 和 Bob 在完成初始的交换交易后，并未广播实际交易，双方一致取消了交易，使用各自的 **超时赎回交易**（return coins transaction）将自己的资产安全地取回。即使交换未完成，双方的数字货币也不会丢失。

5.2.2 情况二

运行结果如下：

```
(venv) john@John:~/john_libr/Blockchain2025/Ex4/code$ python swap.py
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!
```

在此配置下，交易不会真正广播到区块链网络，而是在本地进行模拟验证。执行结果如下：

- **Alice swap tx (BTC) created successfully!**：Alice 创建了锁定 BTC 的交易，但尚未广播。

- **Bob swap tx (BCY) created successfully!**：Bob 创建了锁定 BCY 的交易，也未广播。
- **Alice redeem from swap tx (BCY) created successfully!**：Alice 使用 Bob 创建的 BCY 交易赎回了 BCY，并在本地公开了秘密 x。
- **Bob redeem from swap tx (BTC) created successfully!**：Bob 在 Alice 赎回 BCY 并公开秘密 x 后，可赎回 BTC，本地验证成功。

该组合主要用于**本地验证交易逻辑和赎回流程的正确性**，无需等待真实区块链确认。

5.2.3 情况三

结果查询如下：

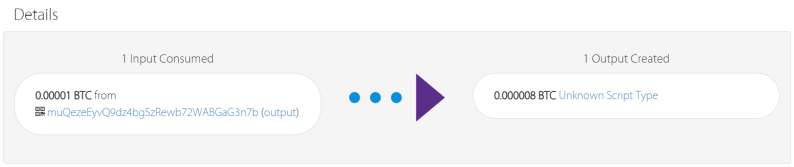


图 1: 开始 true+true 的 BTC

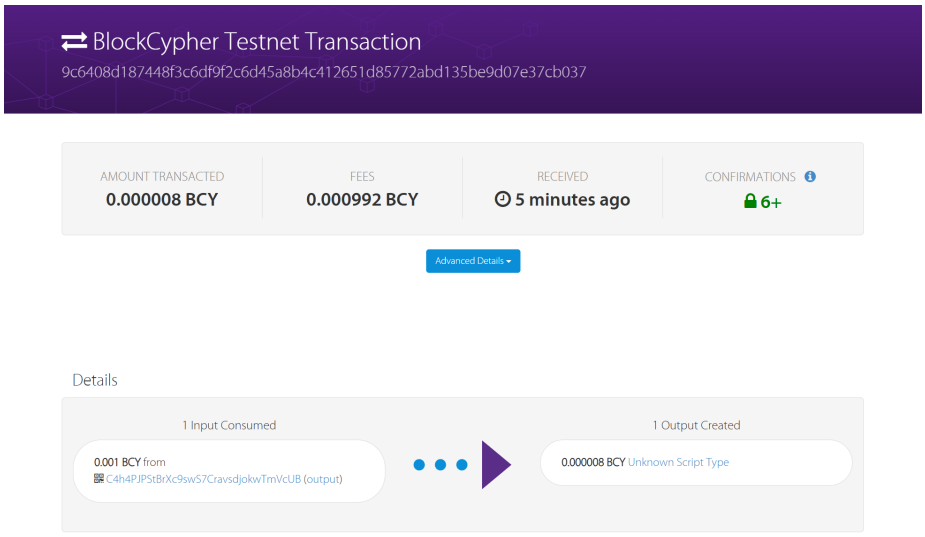


图 2: 开始 true+true 的 BCY

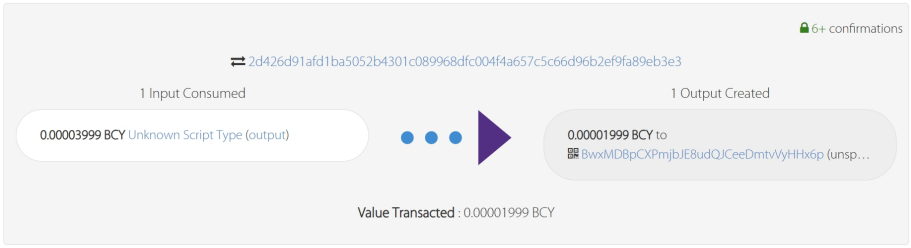


图 3: 结束 true+true 的 BCY

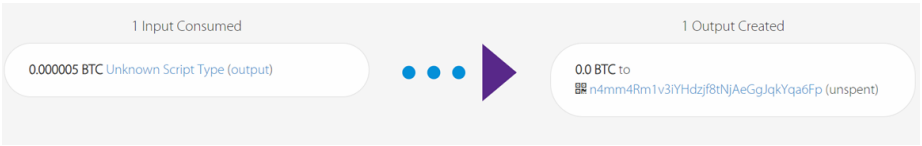


图 4: 结束 true+true 的 BTC

程序输出如下：

```
(venv) john@John:~/john_libr/Blockchain2025/Ex4/code$ python swap.py
Alice swap tx (BTC) created successfully!
201 Created
{
  "tx": {
    "block_height": -1,
    "block_index": -1,
    "hash":
        "faf4efd3df0884ed65fba1513cc7db9563411caf3446e34cd7acb53db903591b",
    "addresses": [
      "muQezeEyvQ9dz4bgSzRewb72WABGaG3n7b"
    ],
    "total": 800,
    "fees": 200,
    "size": 271,
    "vsize": 271,
    "preference": "low",
    "relayed_by": "117.131.219.48",
    "received": "2025-11-04T02:13:01.622582488Z",
    "ver": 1,
    "double_spend": false,
    "vin_sz": 1,
    "vout_sz": 1,
    "confirmations": 0,
    "inputs": [
      {
        "prev_hash":
            "d65bbbe0b3fa3968691c4a46e1a8a071d90bc5ba89595cf8a9ca0488a6769195",
        "output_index": 3,
        "script":
            "483045022100a759c2654b19e721a0a9a7e3cc2efe52d766ebfffa3fd54242561c68bf69525002207a19fa17a
        "output_value": 1000,
        "sequence": 4294967295,
        "addresses": [
          "muQezeEyvQ9dz4bgSzRewb72WABGaG3n7b"
        ],
        "script_type": "pay-to-pubkey-hash",
```

```
        "age": 4748331
      }
    ],
    "outputs": [
      {
        "value": 800,
        "script":
            "210348799ca496c6cd3fb06aaf05495b2a1406ee24e48a2e7e0cba307c0665de1c70ac6363a914853b7750792
        "addresses": null,
        "script_type": "unknown"
      }
    ]
  }
}
Bob swap tx (BCY) created successfully!
201 Created
{
  "tx": {
    "block_height": -1,
    "block_index": -1,
    "hash":
        "9c6408d187448f3c6df9f2c6d45a8b4c412651d85772abd135be9d07e37cb037",
    "addresses": [
      "C4h4PJPStBrXc9swS7CravsdjokwTmVcUB"
    ],
    "total": 800,
    "fees": 99200,
    "size": 270,
    "vsize": 270,
    "preference": "high",
    "relayed_by": "117.131.219.48",
    "received": "2025-11-04T02:13:02.386701421Z",
    "ver": 1,
    "double_spend": false,
    "vin_sz": 1,
    "vout_sz": 1,
    "confirmations": 0,
    "inputs": [
      {
```

```
        "prev_hash":
            "41a53c2d0b67fdea7d45346f4e27b88e2d41bfb4f105f46c26c98377c6b5d72e",
        "output_index": 3,
        "script":
            "473044022035f25acdc6449c5a77b98b4d5ed40fb280938428983748c039d5ae70a8bb1af9022043bc92723ea
        "output_value": 100000,
        "sequence": 4294967295,
        "addresses": [
          "C4h4PJPStBrXc9swS7CravsdjokwTmVcUB"
        ],
        "script_type": "pay-to-pubkey-hash",
        "age": 2088800
      }
    ],
    "outputs": [
      {
        "value": 800,
        "script":
            "21031c3b700580c864f77ea8fd1a6ab103ba6a1d0a39f880e422d2d05689db22ff75ac6363a914853b7750792
        "addresses": null,
        "script_type": "unknown"
      }
    ]
  }
}
Sleeping for 20 minutes to let transactions confirm...
```


5.2.4 情况四

结果查询如下：

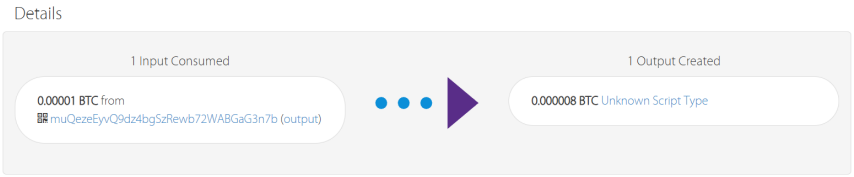
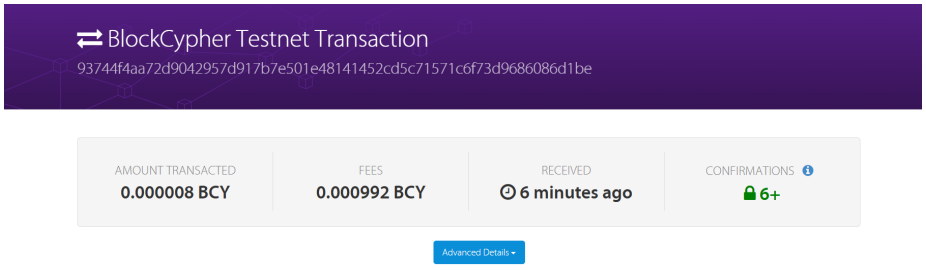


图 5: true+false 的 BTC



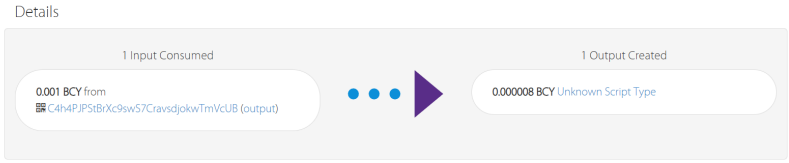


图 6: true+false 的 BCY

程序运行输出:

```
(venv) john@John:~/john_libr/Blockchain2025/Ex4/code$ python swap.py
Alice swap tx (BTC) created successfully!
201 Created
{
  "tx": {
    "block_height": -1,
    "block_index": -1,
    "hash":
```

```
    "918c0a70bacc91b16488995b7f5d3bd8689007d206b74f05dca29e7a7adeea61",
  "addresses": [
    "muQezeEyvQ9dz4bgSzRewb72WABGaG3n7b"
  ],
  "total": 800,
  "fees": 200,
  "size": 271,
  "vsize": 271,
  "preference": "low",
  "relayed_by": "117.131.219.48",
  "received": "2025-11-04T02:21:40.720532627Z",
  "ver": 1,
  "double_spend": false,
  "vin_sz": 1,
  "vout_sz": 1,
  "confirmations": 0,
  "inputs": [
    {
      "prev_hash":
          "d65bbbe0b3fa3968691c4a46e1a8a071d90bc5ba89595cf8a9ca0488a6769195",
      "output_index": 4,
      "script":
          "4830450221009565a45adbdc1486f0a631fbf6207f5a353646776420954e4b38a16a65f866d102201192e3c24
      "output_value": 1000,
      "sequence": 4294967295,
      "addresses": [
        "muQezeEyvQ9dz4bgSzRewb72WABGaG3n7b"
      ],
      "script_type": "pay-to-pubkey-hash",
      "age": 4748331
    }
  ],
  "outputs": [
    {
      "value": 800,
      "script":
          "210348799ca496c6cd3fb06aaf05495b2a1406ee24e48a2e7e0cba307c0665de1c70ac6363a914853b7750792
      "addresses": null,
      "script_type": "unknown"
```

```
      }
    ]
  }
}
Bob swap tx (BCY) created successfully!
201 Created
{
  "tx": {
    "block_height": -1,
    "block_index": -1,
    "hash":
        "93744f4aa72d9042957d917b7e501e48141452cd5c71571c6f73d9686086d1be",
    "addresses": [
      "C4h4PJPStBrXc9swS7CravsdjokwTmVcUB"
    ],
    "total": 800,
    "fees": 99200,
    "size": 270,
    "vsize": 270,
    "preference": "high",
    "relayed_by": "117.131.219.48",
    "received": "2025-11-04T02:21:41.490089816Z",
    "ver": 1,
    "double_spend": false,
    "vin_sz": 1,
    "vout_sz": 1,
    "confirmations": 0,
    "inputs": [
      {
        "prev_hash":
            "41a53c2d0b67fdea7d45346f4e27b88e2d41bfb4f105f46c26c98377c6b5d72e",
        "output_index": 4,
        "script":
            "47304402207edb385029718e71edda75a1878d45b6e9f3b904a8e3bf7c65993b1cdb50b1f6022000ed94df9f2
        "output_value": 100000,
        "sequence": 4294967295,
        "addresses": [
          "C4h4PJPStBrXc9swS7CravsdjokwTmVcUB"
        ],
```

```
        "script_type": "pay-to-pubkey-hash",
        "age": 2088800
      }
    ],
    "outputs": [
      {
        "value": 800,
        "script":
            "21031c3b700580c864f77ea8fd1a6ab103ba6a1d0a39f880e422d2d05689db22ff75ac6363a914853b7750792
        "addresses": null,
        "script_type": "unknown"
      }
    ]
  }
}
Sleeping for 20 minutes to let transactions confirm...
```

在此配置下，Alice 和 Bob 创建的交易会实际广播到区块链网络，但 Alice 并不主动赎回 Bob 的 BCY。执行结果如下：

- **Alice swap tx (BTC) created successfully!**：Alice 创建了锁定 BTC 的交易，并广播到 BTC 测试链。
- **Bob swap tx (BCY) created successfully!**：Bob 创建了锁定 BCY 的交易，并广播到 BCY 测试链。
- **Sleeping for 20 minutes to let transactions confirm...**：交易已广播，系统等待区块确认。

由于 Alice 不赎回 BCY，秘密 x 不会被公开，因此 Bob 无法赎回 BTC。最终，经过 locktime 到期后，双方可以通过时间锁回收各自的原始资产。这种情况用于测试交易广播和超时回退机制的正确性。

6 心得体会

通过本次跨链原子交换实验，我对区块链的交易机制和智能合约的应用有了更深入的理解。实验中需要同时操作 BTC 测试链和 BCY 测试链，

经历了账户创建、分币、交易构建、签名和广播等全过程，使我掌握了以下关键点：

- **跨链交易原理**：通过哈希锁和时间锁实现原子性，保证了在不同链上资产交换的安全性，避免了单链交易无法解决的风险。
- **交易设计与顺序**：理解了 Alice 和 Bob 交易创建的先后顺序及其背后的逻辑，学会了如何设计可回退的安全机制。
- **脚本与签名**：掌握了 coinExchangeScript 的基本工作原理，以及如何使用签名和秘密 x 控制交易赎回。
- **调试与测试**：通过四种不同的 broadcast/ redeem 情况，验证了交易的原子性和回退机制，提高了对链上调试和测试的能力。
- **实践经验**：对区块链交易的手续费计算、UTXO 选择、交易广播和确认有了直观的认识，理解了实际操作中可能遇到的问题和处理方法。

本次实验不仅加深了我对跨链原子交换的理解，也锻炼了我动手分析和调试区块链交易的能力，为后续更复杂的智能合约和多链应用开发打下了基础。