跨链原子交换设计文档

（a）代码实现与 coinExchangeScript 工作原理

核心脚本设计

`coinExchangeScript` 是本项目的核心，实现了哈希时间锁合约（HTLC）：

```
def coinExchangeScript(public_key_sender, public_key_recipient, hash_of_secret):
    return [
        OP_IF,
            # 路径1: 接收方提供密钥和签名
            OP_HASH160,
            hash_of_secret,
            OP_EQUALVERIFY,
            public_key_recipient,
            OP_CHECKSIG,
        OP_ELSE,
            # 路径2: 发送方和接收方都签名
            OP_2,
            public_key_recipient,
            public_key_sender,
            OP_2,
            OP_CHECKMULTISIG,
```

```
        OP_ENDIF
    ]
```

工作机制

1. **路径1（正常赎回）**：接收方提供密钥 x 和签名
   - 验证 `Hash160(x) == hash_of_secret`
   - 验证接收方签名
2. **路径2（退还机制）**：需要双方签名
   - 2-of-2 多重签名验证
   - 用于超时后退还资金

配套脚本

- **P2PKH 脚本**：标准的比特币地址支付脚本
- **签名创建**：Alice 和 Bob 各自的签名函数
- **交易广播**：支持 BTC Testnet3 和 BCY Testnet

（b）资金安全保障机制

Alice 为什么总能拿回她的钱?

1. **时间锁保护**：

```
alice_locktime = 5   # Alice 48小时后可退还
bob_locktime = 3     # Bob 24小时后可退还
```

2. **双重签名机制**：
   - Alice 创建退还交易时，要求 Bob 预先签名
   - 超时后，Alice 使用路径2（2-of-2 多重签名）取回资金
3. **时间优势**：Alice 的锁定时间更长，确保即使 Bob 恶意行为，Alice 也有足够时间应对

为什么不能用简单的 1/2 multisig?

1. **缺乏密钥验证**：1/2 multisig 无法验证密钥 x，无法实现原子性
2. **无条件赎回**：任何一方都可以随时取走资金，没有交换保障
3. **无时间约束**：缺乏超时退还机制，资金可能永久锁定

（c）交易创建顺序与设计原理

交易序列

1. **Alice 创建交换交易**：

```
alice_swap_tx =
alice.alice_swap_tx(txid, index,
amount)
```

2. **Alice 创建退还交易**：

```
alice_return_tx =
alice.return_coins_tx(amount,
alice_swap_tx, locktime)
```

3. **Bob 签名 Alice 的退还交易**：

```
bob_signature =
bob.sign_BTC(alice_return_tx,
scriptPubKey)
```

4. **Alice 广播交换交易**（仅在 Bob 签名后）
5. **Bob 创建对应的交换交易**：

```
bob_swap_tx = bob.bob_swap_tx(txid,
index, amount, hash_of_secret)
```

设计原理

- **先签名后广播**：确保退还机制在交换开始前就位
- **时间差设计**：Bob 锁定时间短于 Alice，防止竞争条件
- **秘密共享**：只有 Alice 知道 x，但 hash(x) 公开

（d）数字货币流转分析

成功的原子交换流程

```
初始状态:
Alice: 0.00009 BTC (Testnet3)
Bob:   0.01 BCY (BCY Testnet)

步骤1: Alice → coinExchangeScript (BTC)
Alice: 0 BTC
Bob:   0.01 BCY
合约: 0.00009 BTC (可被 Bob+x 或 Alice+Bob 赎回)

步骤2: Bob → coinExchangeScript (BCY)
Alice: 0 BTC
Bob:   0 BCY
合约1: 0.00009 BTC (Alice→Bob)
合约2: 0.01 BCY (Bob→Alice)

步骤3: Alice 赎回 BCY (使用密钥 x)
Alice: 0.01 BCY
Bob:   0 BCY
合约1: 0.00009 BTC (Alice→Bob)
合约2: 0 BCY

步骤4: Bob 赎回 BTC (使用从步骤3获得的 x)
最终状态:
Alice: 0.01 BCY
```

```
Bob:   0.00009 BTC
```

失败的交换流程

```
场景：Bob 创建交换交易后离线

步骤1-2: 同上，两个合约都创建

步骤3: Alice 等待超时
- Bob 24小时后可以退还 BCY
- Alice 48小时后可以退还 BTC

步骤4: 超时退还
Alice 使用预签名的退还交易取回 BTC
Bob 使用双重签名取回 BCY

最终状态:
Alice: 0.00009 BTC (原始状态)
Bob:   0.01 BCY (原始状态)
```