

区块链基础及应用实验报告

Ex2

网络空间安全学院 信息安全专业

2112492 刘修铭 0939

https://github.com/lxmliu2002/Blockchain_Fundamentals_and_Applications

Ex2a

1. 打开keygen.py文件，创建三个新账户，并将三个账户的私钥填写到ex2a.py文件中。

```
1 Private key: cTzY6AedUp2ki6yhKnCN5M4pLu7E7NZL44BaxxhxDZATueV4sF
2 Address: mmaRMk3sMBGvAxElb1nGP94SMK7GqwoaMK
```

```
1 Private key: CS8mXETAn6PZqM7cLnH7pMHcfLVHuzBoqnaCMXGE1KFVmoT9FYny
2 Address: mpqNgShTN4GRY9njKnjfeY8GeuZ6vdfDR6
```

```
1 Private key: cUUKg5w78M22jA5uQyRcoCwMZQKGSDeQMT4RgBiHEnAvVAGP9Nv
2 Address: mxKRmngw3sT6ZYGAYLNNybTzd6e5khgnaS
```

```
1 # 客户私钥
2 cust1_private_key =
  CBitcoinSecret('cTzY6AedUp2ki6yhKnCN5M4pLu7E7NZL44BaxxhxDZATueV4sF')
3 cust1_public_key = cust1_private_key.pub
4 cust2_private_key =
  CBitcoinSecret('CS8mXETAn6PZqM7cLnH7pMHcfLVHuzBoqnaCMXGE1KFVmoT9FYny')
5 cust2_public_key = cust2_private_key.pub
6 cust3_private_key =
  CBitcoinSecret('cUUKg5w78M22jA5uQyRcoCwMZQKGSDeQMT4RgBiHEnAvVAGP9Nv')
7 cust3_public_key = cust3_private_key.pub
```

2. 接着补全代码，创建多重签名脚本，完成多重签名交易的要求。

```
1 # 定义所需的签名数量
2 required_signatures = 3
3 # 创建包含cust1、cust2和cust3公钥的列表
4 pubkeys = [cust1_public_key, cust2_public_key, cust3_public_key]
5 # 创建多重签名脚本
6 ex2a_txout_scriptPubKey = CScript([required_signatures] + pubkeys +
  [len(pubkeys), OP_CHECKMULTISIG])
```

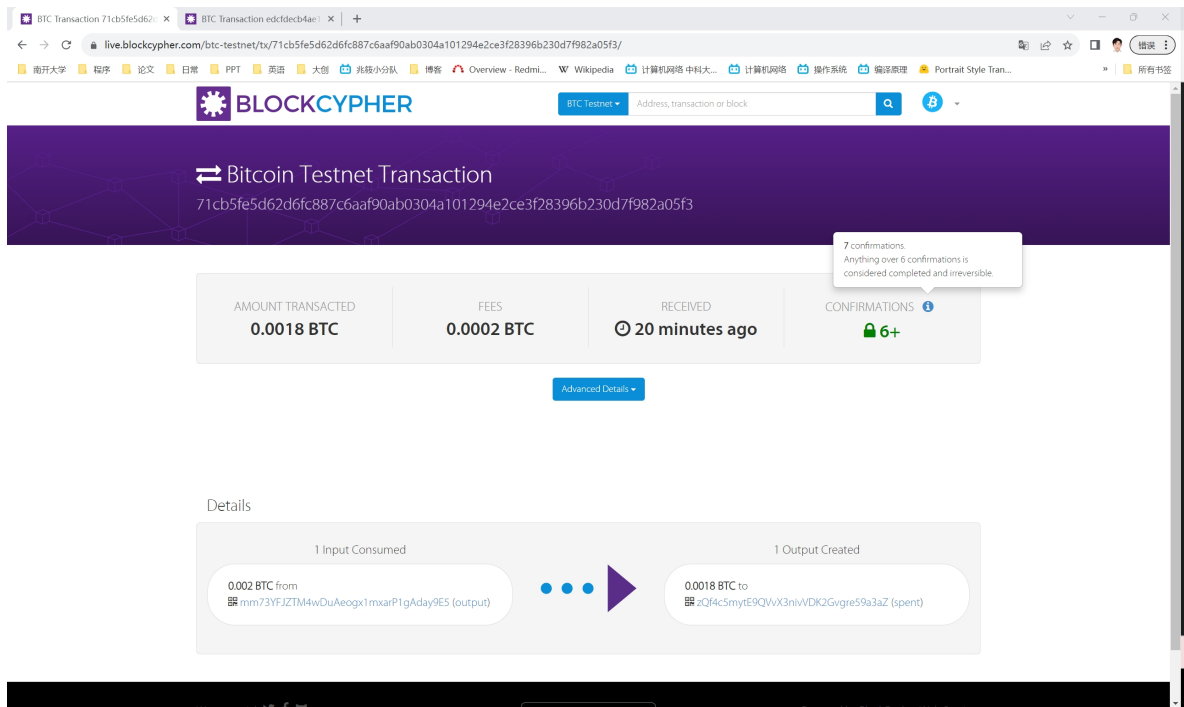
3. 填写好要交易的总金额、UTXO交易ID、UTXO索引等信息。

```
1 amount_to_send = 0.0018 # amount of BTC in the output you're splitting
  minus fee
2 txid_to_spend =
  ('8b89b2517a25e6694b17c52b2f37d595578911c928a9434aeb79ecbe7d976d')
3 utxo_index = 8
```

4. 输出信息

- 由于篇幅所限，输出信息将于附件output_ex2a.txt中展示。

5. faucet截图



Ex2b

1. 补全多重签名脚本的解锁脚本函数

- 使用银行私钥 (`my_private_key`) 和三个客户的私钥 (`cust1_private_key`、`cust2_private_key`、`cust3_private_key`) 来生成签名。然后，它将这些签名与操作码 `OP_0` 一起返回，以构建多重签名的解锁脚本。

```

1 def multisig_scriptSig(txin, txout, txin_scriptPubKey):
2     bank_sig = create_OP_CHECKSIG_signature(txin, txout,
        txin_scriptPubKey, my_private_key)
3     cust1_sig = create_OP_CHECKSIG_signature(txin, txout,
        txin_scriptPubKey, cust1_private_key)
4     cust2_sig = create_OP_CHECKSIG_signature(txin, txout,
        txin_scriptPubKey, cust2_private_key)
5     cust3_sig = create_OP_CHECKSIG_signature(txin, txout,
        txin_scriptPubKey, cust3_private_key)
6     return [OP_0, bank_sig, cust1_sig, cust2_sig, cust3_sig]

```

2. 填写好要交易的总金额、UTXO交易ID、UTXO索引等信息。

```

1 amount_to_send = 0.0015
2 txid_to_spend =
    '71cb5fe5d62d6fc887c6aaf90ab0304a101294e2ce3f28396b230d7f982a05f3'
3 utxo_index = 0

```

3. 输出信息

- 由于篇幅所限，输出信息将于附件output_ex2b.txt中展示。

4. faucet截图

The screenshot shows the BlockCypher website interface for a Bitcoin Testnet Transaction. The transaction ID is `edcfdec4ae17bd357c158a67e6423ce6f4e7434e77f8f646abf35ec558c1be`. The transaction details are as follows:

AMOUNT TRANSACTED	FEES	RECEIVED	CONFIRMATIONS
0.0015 BTC	0.0003 BTC	19 minutes ago	6+

A tooltip indicates: "7 confirmations. Anything over 6 confirmations is considered completed and irreversible."

The "Details" section shows the transaction flow:

- 1 Input Consumed:** 0.0018 BTC from `zQ4c5mytE9QVvX3niVVK2Gvgr59a3aZ` (output)
- 1 Output Created:** 0.0015 BTC to `mv4myY3Su5gjcDNzbMLK8Qk8icCtHUfB` (unspent)

The footer includes social media links, a GitHub link, and a note: "Powered by BlockCypher Web Services © 2023 - BlockCypher".