LAB02 - BLOCKCHAIN

Danh sách thành viên:

1. 22C11002 - Ngô Minh Đức
2. 22C11030 - Đặng Văn Hiển
3. 22C11035 - Đặng Trung Kiên

1. Cấu Trúc Dự Án:

+ File createP2PKH.py dùng để tạo private key, public key, bitcoin address. Lưu ý "testnet" để tạo giao dịch test từ key.
+ File spendLockFunds.py chi tiêu số tiền bị khóa dựa trên mã giao dịch phát sinh
+ File multiSignitureTransaction.py dùng để tạo nhiều private key, redeem script, địa chỉ giao dịch
+ File multisigAddress.py chi tiêu số tiền bị khóa dựa trên mã giao dịch phát sinh từ multi address
+ Phần mềm tạo ra giao dịch testnet: https://coinfaucet.eu/en/btc-testnet/ dùng để phát sinh giao dịch từ bitcoin address

2. Cấu trúc dữ liệu

a. File createP2PKH.py
CBitcoinSecret (private_key):
Loại: Đối tượng của lớp CBitcoinSecret.
Mô tả: Lớp này cung cấp một cách tiện lợi để làm việc với khóa riêng trong Bitcoin.
bytes (public_key):
Loại: Bytes.
Mô tả: Chuỗi bytes biểu diễn khóa công khai.
P2PKHBitcoinAddress (address):
Loại: Đối tượng của lớp P2PKHBitcoinAddress.
Mô tả: Lớp này biểu diễn địa chỉ Bitcoin dựa trên P2PKH Script và chứa thông tin như scriptPubKey.

b. File spendLockFunds.py

- Hàm create_signed_transaction:
  + Đầu vào:
    txins: Danh sách đầu vào giao dịch (đối tượng CTxIn).
    txouts: Danh sách đầu ra giao dịch (đối tượng CTxOut).
    private_keys: Danh sách các khóa riêng tương ứng với địa chỉ đầu vào.
  + Đầu ra:
    Giao dịch đã ký được chuỗi hóa theo định dạng bytes.
- Hàm create_txin:
  + Đầu vào:

utxo_txid: ID giao dịch của UTXO muốn chi trả.
utxo_index: Chỉ số của đầu ra trong giao dịch.

+ Đầu ra:
Tạo một từ điển biểu diễn đầu vào giao dịch (TxIn) sử dụng thông tin UTXO.

- Hàm create_txout:
  + Đầu vào:
    amount_to_send: Số lượng satoshis muốn gửi.
    destination_address: Địa chỉ của người nhận.
  + Đầu ra:
    Tạo một từ điển biểu diễn đầu ra giao dịch (TxOut).
- Biến:
  private_key: Khóa riêng của địa chỉ đầu vào.
  address: Địa chỉ Bitcoin tương ứng với khóa riêng.
  txid: ID giao dịch của UTXO cần chi trả.
  output_index: Chỉ số của đầu ra trong giao dịch.
- Thực thi: txin = create_txin(txid, output_index): Tạo đầu vào giao dịch sử dụng thông tin UTXO cung cấp.
  destination_address = 'mv4rnyY3Su5gjcDNzbMLKBQkBicCtHUtFB': Địa chỉ của người nhận cho đầu ra giao dịch.
  amount_to_send = 0.00395847: Số lượng satoshis muốn gửi cho đầu ra giao dịch.
  txout = create_txout(amount_to_send, destination_address): Tạo đầu ra giao dịch.
  tx = create_signed_transaction([txin], [txout], [private_key]): Tạo và ký giao dịch.
  broadcast_tx(tx): Phát sóng giao dịch đến mạng Bitcoin.

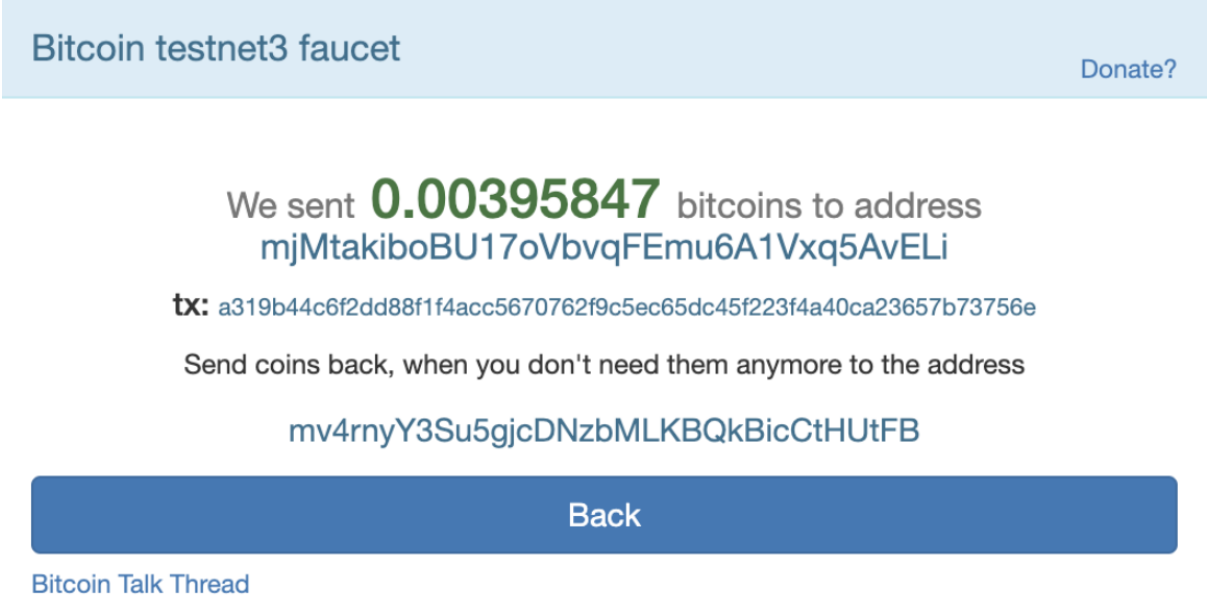
c. File multisigAddress.py
   - Biến:
     private_key1, private_key2: Khóa riêng ngẫu nhiên cho hai người tham gia.
     public_key1, public_key2: Khóa công khai tương ứng với khóa riêng trên.
     redeem_script: Skript chuẩn bị cho giao dịch Multisig 2-of-2.
     address: Địa chỉ Bitcoin được tạo từ skript Multisig.
   - Hàm:
     CBitcoinSecret.from_secret_bytes: Tạo một khóa riêng từ dữ liệu ngẫu nhiên.
     P2SHBitcoinAddress.from_redeemScript: Tạo một địa chỉ P2SH từ skript chuẩn bị.

d. File multisigAddress.py

3. Thao tác

Task 1:

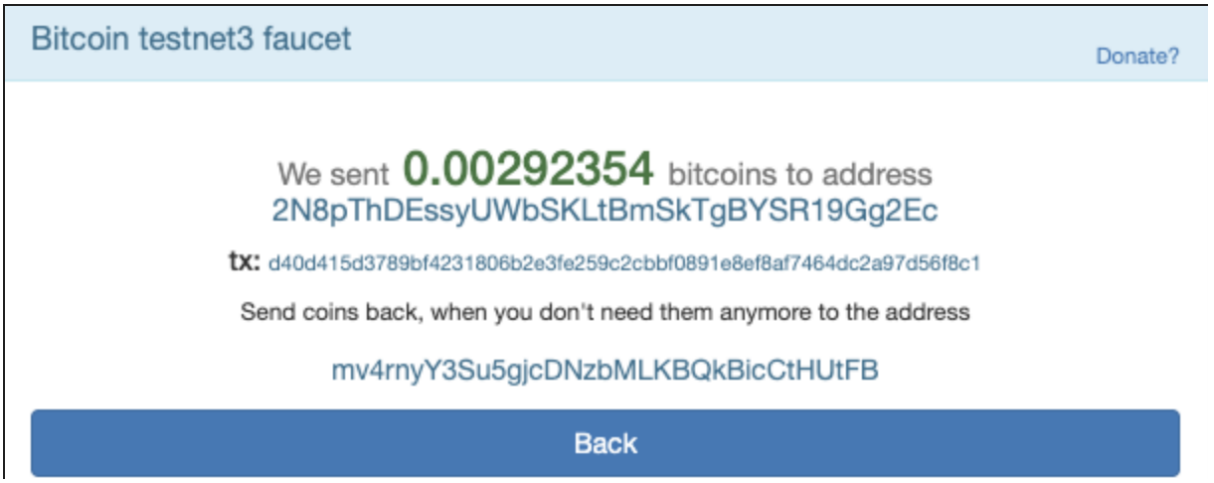
- Chạy file createP2PKH.py để lần lượt sinh ra private, public key và bitcoin address.
  Private Key:cSz37ppyonDHANjpv6pziLPKYEKXEiGrCLKRyqwLqkiTs2aZQKgx
  PublicKey:032f9beaec92ca3a291c53d1f25ec72984c6bd080ee0ad823cf2735e17514e3950
  Bitcoin Address: mjMtakiboBU17oVbvqFEmu6A1Vxq5AvELi
- Tạo giao dịch test



- Chạy file spendLockFunds.py chi tiêu số tiền bị khóa dựa trên mã giao dịch, địa chỉ đích được sinh ra

Task 2:

- Chạy file multisigAddress.py để sinh ra private key 1, private key 2, redeem script và Multisig address:
  Private Key 1: cTSebtTjRooKtvFpgznJiprVQAUTL7PHAyGgn6uJVGXhwwkx6jgp
  Private Key 2: cSovutdQDj1cXUxhB6p7GWEYYJBGDPXUcHpBErAUPuXWQ7WTRaTK
  Redeem Script:
  5221028e03b2a4270b3cea8d7d31cf7b7dc7df999a4064e1e988b3d25c08ff4ca400ca21027ce
  dea6a6db670360c123f1b74c6cdd661369792b346b2b9c24acdb6bb140e3d52ae
  Multisig Address: 2N8pThDEssyUWbSKLtBmSkTgBYSR19Gg2Ec
- Tạo giao dịch test



- Chạy file multiSignitureTransaction.py để chi tiêu số tiền bị khóa dựa trên mã giao dịch, địa chỉ đích được sinh ra

4. Nhận Xét về Thiết Kế và Thực Hiện

Đoạn mã trên có chức năng sinh ra một cặp khóa (private key và public key) và sau đó tạo địa chỉ Bitcoin từ khóa công khai đó. Dưới đây là nhận xét về thiết kế và thực hiện của mã nguồn:

- Sử Dụng Thư Viện Bitcoin Python: Việc sử dụng thư viện Bitcoin Python giúp giảm độ phức tạp của mã nguồn và cung cấp các công cụ tiện ích để làm việc với Bitcoin.
- Bảo Mật và Ngẫu Nhiên: Việc sử dụng os.urandom(32) để sinh khóa riêng từ nguồn ngẫu nhiên làm tăng tính bảo mật của mã nguồn.
- Hiển Thị Thông Tin Chi Tiết: Mã nguồn được thiết kế để hiển thị thông tin chi tiết về khóa riêng, khóa công khai và địa chỉ Bitcoin. Điều này giúp người phát triển và người đọc mã hiểu rõ hơn về quá trình sinh khóa.
- Kiểm Soát Định Dạng Đầu Ra: Việc sử dụng hex() để hiển thị giá trị của khóa công khai làm cho đầu ra trở nên dễ đọc và kiểm soát định dạng.
- Không Có Xử Lý Ngoại Lệ: Mã nguồn không xử lý các trường hợp ngoại lệ như lỗi khi sinh khóa hoặc các vấn đề khác liên quan đến định dạng dữ liệu.
- Thiếu Chú Giải và Hướng Dẫn: Mã nguồn không đi kèm với chú giải hoặc hướng dẫn, điều này có thể làm giảm khả năng hiểu của người đọc về mục đích và cách sử dụng của nó.

Tóm lại, đoạn mã thực hiện chức năng cơ bản của việc tạo khóa và địa chỉ Bitcoin. Thiết kế của nó tương đối đơn giản, phù hợp để sử dụng trong các tình huống đòi hỏi sự nhanh chóng và thuận tiện. Đối với mục đích này, mã nguồn đáp ứng tốt với yêu cầu cơ bản và cung cấp một cách đơn giản để sinh khóa Bitcoin.

5. Mô Tả Hoạt Động Hệ Thống

Hệ thống của chúng tôi bao gồm một đoạn mã thực hiện việc tạo địa chỉ Bitcoin dựa trên P2PKH Script. Dưới đây là mô tả chi tiết về cách hệ thống hoạt động:
Tạo một khóa riêng ngẫu nhiên (private_key) sử dụng thư viện Bitcoin.
Tính toán khóa công khai (public_key) từ khóa riêng đó.
Sử dụng địa chỉ Bitcoin (address) được tạo từ khóa công khai bằng cách ánh xạ nó qua P2PKH Script.

- Khởi tạo Môi Trường Testnet Bitcoin: Sử dụng bitcoin.SelectParams('testnet') để đảm bảo rằng hệ thống đang làm việc trên môi trường thử nghiệm Testnet của Bitcoin.
- Tạo Đầu Vào Giao Dịch (TxIn): Sử dụng hàm create_txin để tạo một đầu vào giao dịch (TxIn) sử dụng thông tin về đầu ra không được chi tiêu (UTXO) mà bạn muốn chi tiêu.
- Đầu vào này bao gồm thông tin như txid, chỉ mục (vout), và các thông số khác.
- Tạo Đầu Ra Giao Dịch (TxOut): Sử dụng hàm create_txout để tạo một đầu ra giao dịch (TxOut) với số lượng Bitcoin và địa chỉ người nhận mong muốn.
- Tạo và Ký Giao Dịch: Sử dụng hàm create_signed_transaction để tạo một giao dịch Bitcoin được ký. Mỗi đầu vào của giao dịch được ký bằng cách tạo một chữ ký số dựa trên khóa riêng của đầu vào và thông tin giao dịch. Một đoạn mã scriptSig (script khai báo chữ ký) được thêm vào mỗi đầu vào để xác minh chữ ký.
- Phát Sóng Giao Dịch: Gọi hàm broadcast_tx để phát sóng giao dịch đã tạo ra mạng Bitcoin Testnet.
- Kết Quả In Ra Màn Hình: In ra màn hình thông tin về đầu vào (txin), đầu ra (txout), và khóa riêng (private_key) sử dụng để ký giao dịch.
- Cuối cùng, in ra màn hình chuỗi hex của giao dịch đã ký để theo dõi và xác minh.

6. Thách Thức Trong Quá Trình Thực Hiện

- Bảo Mật:
    + Thách Thức: Bảo mật là một điểm quan trọng khi làm việc với khóa riêng và khóa công khai. Việc không đảm bảo an toàn đủ có thể gây rủi ro về bảo mật của hệ thống.
    + Giải Pháp: Sử dụng nguồn ngẫu nhiên đáng tin cậy như os.urandom(32) để sinh khóa riêng.
- Xử Lý Ngoại Lệ:
    + Thách Thức: Mã nguồn không xử lý các trường hợp ngoại lệ như lỗi khi sinh khóa hoặc các vấn đề khác liên quan đến định dạng dữ liệu.
    + Giải Pháp: Thêm các kiểm tra và xử lý ngoại lệ để đảm bảo tính ổn định của chương trình.
- Hiểu Rõ Về Bitcoin:

    + Thách Thức: Việc làm việc với Bitcoin đôi khi đòi hỏi hiểu biết sâu rộng về các khái niệm như UTXO, script, và quy trình hoạt động của mạng Bitcoin.
    + Giải Pháp: Cung cấp chú giải và tài liệu giải thích cơ bản về các khái niệm liên quan đến Bitcoin.
- Chú Giải và Hướng Dẫn:
    + Thách Thức: Thiếu chú giải và hướng dẫn có thể làm tăng độ khó của việc hiểu mã nguồn và cách sử dụng nó.
    + Giải Pháp: Thêm chú giải và hướng dẫn rõ ràng giúp người đọc hiểu rõ mục đích của mã nguồn và cách sử dụng nó.
- Thiếu Đa Dạng Hóa:

    + Thách Thức: Mã nguồn chỉ thực hiện một chức năng cụ thể (tạo khóa và địa chỉ Bitcoin), không đa dạng hóa nhiều chức năng.
    + Giải Pháp: Đối với mục đích học tập hoặc thử nghiệm, việc thêm các chức năng khác có thể giúp nâng cao kỹ năng lập trình.

7. Bài Học Học Được

- Bảo Mật là ưu tiên hàng đầu: Bảo mật là một trong những yếu tố quan trọng nhất khi làm việc với tiền điện tử. Sinh khóa và xử lý thông tin liên quan đến khóa riêng cần phải được thực hiện một cách an toàn.
- Xử lý Ngoại lệ và Kiểm Tra Đầu Vào: Việc kiểm tra và xử lý ngoại lệ là quan trọng để đảm bảo ổn định và an toàn của chương trình. Kiểm tra đầu vào cũng giúp tránh được các lỗi không mong muốn.
- Hiểu Biết Sâu Rộng về Bitcoin: Để làm việc với Bitcoin, đặc biệt là khi tạo và xử lý giao dịch, cần phải hiểu rõ về các khái niệm cơ bản như UTXO, script, và cách mạng Bitcoin hoạt động.
- Chú Giải và Hướng Dẫn Đầy Đủ: Việc bổ sung chú giải và hướng dẫn giúp người đọc hiểu rõ hơn về mục đích và cách sử dụng mã nguồn. Điều này quan trọng đối với cả người viết mã và người đọc mã nguồn.
- Kiểm Thử và Thử Nghiệm: Thử nghiệm mã nguồn với nhiều tình huống và dữ liệu đầu vào khác nhau giúp đảm bảo tính đúng đắn và ổn định của chương trình.
- Đa Dạng Hóa và Mở Rộng: Đối với mục đích học tập và nâng cao kỹ năng lập trình, việc thêm các chức năng mới hoặc mở rộng chức năng của mã nguồn có thể là một bài học quan trọng.

8. Tham Khảo

- https://www.slideshare.net/flowersj/introduction-to-bitcoins-scripting-language
- https://bitcoinvn.io/news/bitcoin-address-dia-chi-vi-bitcoin-la-gi/#:~:text=Bitcoin%20l%C3%A0%20g%C3%AC%3F%E2%80%9D-,%C4%90%E1%BB%8Ba%20ch%E1%BB%89%20v%C3%AD%20Bitcoin%20l%C3%A0%20g%C3%AC%3F,ch%C3%BAng%20ta%20g%E1%BB%ADi%20th%C6%B0%20v%E1%BA%ADy.

- https://viblo.asia/p/hieu-mot-cach-tong-quat-ve-cac-giao-dich-trong-bitcoin-maGK74oLZj2
- https://academy.binance.com/vi/articles/an-introduction-to-bitcoin-script
-