ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

LAB 2:

Understanding Bitcoin's Scripting Language

Sinh viên thực hiện

19120659 – Phạm Văn Thành

20120356 – Lê Minh Quân

20120382 – Hoàng Thu Thủy

20120386 – Lê Phước Toàn

20120389 – Nguyễn Thị Bích Trâm

Giảng viên hướng dẫn:

TS. Nguyễn Đình Thúc

Bộ môn: Blockchain và Ứng dụng

Thành phố Hồ Chí Minh, ngày 18 tháng 12, 2023

THÔNG TIN NHÓM

Nhóm 3			
Họ và tên	**MSSV**	**Phân công**	**Đánh giá**
Phạm Văn Thành	19120659	Multisignature Transactions (P2SH)	100%
Lê Minh Quân	20120356	P2PKH (Pay-To-Public-Key-Hash)	100%
Hoàng Thu Thủy	20120382	Analysis and Reflection	100%
Lê Phước Toàn	20120386	P2PKH (Pay-To-Public-Key-Hash)	100%
Nguyễn Thị Bích Trâm	20120389	Multisignature Transactions (P2SH)	100%

MỤC LỤC

1. Basic Script Execution (P2PKH)

1.1. Tổng quan về Bitcoin Script và P2PKH(Pay-To-Public-Key-Hash)

Bitcoin Script là ngôn ngữ lập trình không hoàn chỉnh dựa trên ngăn xếp được Bitcoin blockchain sử dụng để lập trình các giao dịch.

Có 2 loại Transaction scripts quan trọng, được dùng để xác thực các giao dịch của người tham gia:

Locking script (scirptPubKey): chỉ định các điều kiện cần phải được đáp ứng trước khi funds được người nhận sử dụng

Unlocking script (scriptSign): đáp ứng các điều kiện được đưa ra bởi Locking Script

P2PKH là một trong những locking script phổ biến được dùng cho hầu hết các giao dịch trên mạng Bitcoin. Các outputs bị khóa bởi P2PKH sẽ được mở bởi khóa công khai và chữ ký điện tử được tạo bởi khóa bí mật tương ứng

Trước P2PKH, đã tồn tại P2PK (Pay-To-Public-Key). Tuy nhiên, nó đã được thay đổi vì hai lý do:

+ Do máy tính lượng tử có thể lấy khóa riêng từ khóa công khai. Điều này có thể được giải quyết bằng cách chỉ sinh public key khi tiền được sử dụng, nghĩa là giả định rằng địa chỉ sẽ không được sử dụng lại. Và trong trường hợp này, chúng ta không thể lấy private key cụ thể.

+ Do kích thước của phiên bản trước khá lớn nên P2PKH ra đời để làm cho kích thước nhỏ hơn (20 byte). Bây giờ có thể nhúng nó vào một phương tiện như mã QR hoặc bản in.

1.2. Cơ chế hoạt động P2PKH script

Giả sử có 2 blockchain user là A và B. A muốn gửi cho B một vài bitcoins. A sẽ tạo một giao dịch mới dữ liệu phù hợp, và để xác nhận A sở hữu số bitcoins muốn gửi cho B thì ta phải tạo ra một script để xác nhận điều đó (một phần của P2PKH), gọi là scriptSig. Điều này được thực hiện bằng cách lấy public key của A đã nhận bitcoin và xác minh rằng nó có private key cần thiết để có thể chi tiêu bitcoin. Nếu A thực sự có thể chi tiêu bitcoin thì bước tiếp theo là tạo P2PKH script.

Lúc này ta tạo ra một locking script để đưa ra các điều kiện cần được đáp ứng để bitcoins có thể được chi tiêu. Kết hợp locking và unlocking script để hoàn thành P2PKH.

Những OP_CODES thường được sử dụng cho P2PKH script:

1. OP_DUP - Sao chép giá trị ở đỉnh của stack

2. OP_HASH160 - Thực hiện double hashing SHA-256 và RIPEMD-160 của giá trị đã được sao chép.

3. OP_DATA_X - Đẩy địa chỉ vào stack.

4. OP_EQUALVERIFY - Xác minh rằng giá trị hash đã được sao chép khớp với giá trị hash mong đợi đưa vào stack.

5. OP_CHECKSIG - Xác minh rằng stack chứa một public key và chữ ký và xác minh rằng chữ ký là hợp lệ cho public key tương ứng.

Ví dụ:

Xét P2PKH transaction script sau:

```
<sig> <pubKey> OP_DUP OP_HASH160 <pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG
```

Các bước thực thi script bên trên như sau:

B1: <sig> <pubKey> lần lượt được đưa vào stack

B2: OP_DUP sẽ lấy <pubKey> để sao chép và sau đó đưa lại vào stack

B3: OP_HASH160 sẽ lấy giá trị vừa sao chép để hash và đưa lại vào stack

B4: <pubKeyHash> được đưa vào stack

B5: OP_EQUALVERIFY sẽ so sánh <pubKeyHash> với giá trị đã hash trước đó, nếu hợp lệ sẽ đến B6

B6: OP_CHECKSIG sẽ kiểm tra stack có còn chứa <sig> và <pubKey> hay không, nếu còn thì xác minh xem <sig> có hợp lệ với <pubKey> hay không, nếu hợp lệ thì giao dịch hợp lệ

1.3. Ưu điểm và nhược điểm của P2PKH

Ưu điểm:

+ Đáng tin cậy trong các giao dịch vì người dùng có thể chia sẻ public keys của họ để trao đổi giá trị với nhau

+ Public keys được hash thành một chuỗi có kích thước cố định nhỏ hơn, tức là có thể lưu trữ trong các phương tiện nhỏ như QR codes.

+ Khóa có kích thước nhỏ đồng nghĩa với việc ít lỗi và băng thông mạng được sử dụng ít hơn

+ Đơn giản và dễ sử dụng so với các script tương ứng khác

Nhược điểm:

+ Tạo ra thêm độ phức tạp khi tạo ra các địa chỉ P2PKH tương ứng

+ P2PKH script được triển khai trước SegWit, do đó, các địa chỉ cũ không tương thích với các ví điện tử sử dụng SegWit. Người dùng vẫn có thể gửi giao dịch từ địa chỉ P2PKH đến địa chỉ SegWit nhưng phí giao dịch sẽ rất cao

2. Multisignature Transactions (P2SH)

2.1 Cơ chế hoạt động và bảo mật

P2SH (Pay-to-Script-hash) là một phương thức nhận bitcoin linh hoạt được giới thiệu vào năm 2012 dưới dạng một soft-fork thuộc BIP 16. Nó có thể được sử dụng để tạo địa chỉ đa chữ ký, địa chỉ SegWit và các loại địa chỉ khác. P2SH chỉ được tạo bằng cách băm một đoạn mã rút gốc (a redeem script) thay vì băm một khóa chung (a single public key).

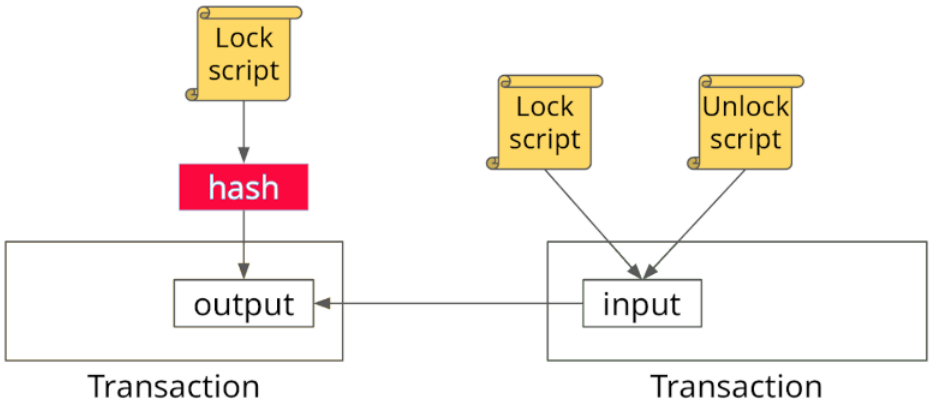
Đoạn mã rút gốc này xác định cách bitcoin nhận được đến địa chỉ P2SH có thể được chi tiêu trong tương lai. Người nhận xác định chi tiết đoạn mã và hướng dẫn về việc chi tiêu không được tiết lộ công khai cho đến khi bitcoin được chi tiêu khỏi địa chỉ.

Người dùng nâng cao có thể xây dựng các script phức tạp, nhưng các ứng dụng phổ biến nhất của P2SH là tạo địa chỉ đa chữ ký (multiple signature) và ví SegWit.

Hình ảnh dưới đây minh họa một giao dịch Bitcoin P2SH đơn giản trong đó:

+ **Lockscript:** Quy định điều kiện cần đáp ứng trước khi funds được chuyển tới đầu ra giao dịch (transaction output)

+ **UnlockScript:** Các điều kiện cần thiết cần đáp ứng để mở khóa Lockscript



P2SH cung cấp sự linh hoạt bởi vì người dùng có thể tạo ra một số tập lệnh tùy ý, như hình minh họa trên nếu Người dùng A muốn gửi bitcoins cho B. A chỉ cần gửi bản mã nhúng (hash) của LockScript chứ không cần gửi trực tiếp LockScript điều này đảm bảo sự bảo mật quyền riêng tư cho A. Khi Transaction được gửi cho B, B sẽ phải xây dựng lại bản mã (hash) được dùng bởi A và sau đó ký vào transaction với private keys.

Khác với P2PKH transaction(sử dụng input là các public key của người nhận), P2SH có input đầu vào phức tạp hơn chẳng hạn như nhiều chữ ký có thể được bao gồm(quy định bởi script). Sau đó script sẽ được băm và đoạn mã sau băm của P2SH sẽ có dạng:

```
          Output (scriptPubKey)
--------------------------------
OP_HASH160 <scripthash> OP_EQUAL
```

Để người nhận B có thể sử dụng tiền(funds) đã bị khóa trong transaction, B cần phải đảo ngược lại hàm băm để tìm ra đoạn mã gốc sử dụng để mở khóa funds, đoạn mã được tạo ra này chính là đoạn mã rút gốc (redeem script). Nó được sử dụng để nhận dạng rằng tất cả thông tin liên quan tới người nhận(mã băm, chữ ký,...) B là đúng, sau đó B có thể sử dụng số tiền.

Quá trình mở khóa P2SH sẽ có dạng:

```
          Input (scriptSig)
-----------------------------------------
              scriptSig       scriptPubKey (as a data push)
----------------------------- --------------
OP_0 <signature> <signature> <redeemscript>
```

Trong đó thì cả Locking và Unlocking Script đều được chứa trong scriptSig. Sau đó trong quá trình thực thi đoạn *Redeem script* sẽ được mang đi đối chiếu với *Scripthash*. Nếu chúng khớp nhau thì *Redeem script* đã được giải mã và chúng ta sẽ được:

```
     Input (scriptSig)
-----------------------------------------
             scriptSig       scriptPubKey
--------------------------- --------------
OP_0 <signature> <signature> OP_2 <pubkey> <pubkey> <pubkey> OP_3 OP_CHECKMULTISIG
```

Chúng ta có thể thấy đã các chữ ký cùng với các public key đều được bọc trong một đoạn đoạn mã P2SH, điều này nhằm giảm tỉ lệ phí phải chi trả đối với người gửi.

P2SH được thực thi thành hai phần, *standard execution* và *redeem script execution*.

- Trong *standard execution*, đoạn mã gốc(*redeem script*) được băm rồi so sánh với đoạn mã trong *locking script*(scriptPubKey). Thông thường, việc thực thi tập lệnh dừng ở đây, trong một số trường hợp có thể bổ sung thêm BIP-16.
- Trong *redeem script execution*, *redeem script* được giải mã và thực thi giống như tập lệnh khóa tiêu chuẩn(*standard execution*)

Tóm tắt lại, quá trình trong P2SH sẽ bao gồm:

1. Người nhận tạo ra một bộ mã điều kiện gốc - *redeem script*.
2. Người nhận tạo ra một địa chỉ Bitcoin bằng cách băm tập *redeem script* và mã hóa nó theo định dạng của địa chỉ.
3. Người gửi gửi quỹ đến địa chỉ với tiền (funds)bị khóa theo các điều kiện cụ thể ở bước 1.
4. Để người nhận có thể sử dụng Bitcoin được đã gửi, người nhận phải cung cấp *redeem script* hợp lệ có mã hash khớp với địa chỉ. R*edeem script* sau đó được thực thi như một *Locking script*

2.2 Ưu và nhược điểm

***) Ưu điểm:**

- Tính bảo mật và kiểm soát phi tập trung đối với tiền của ví, P2SH hỗ trợ ví chia sẻ multisig.
- P2SH hỗ trợ SegWit và Non-SegWit. Nested SegWit là ví bitcoin được tạo bằng khóa riêng ở định dạng P2SH.
- So với các tập lệnh giao dịch khác như tập lệnh P2MS, chúng nhỏ hơn nhiều, do đó, điều này dẫn đến ít dữ liệu giao dịch hơn, đồng nghĩa với việc mạng nhanh hơn và phí giao dịch nhỏ hơn.
- Người gửi bitcoin không phải viết các tập lệnh phức tạp vì giao dịch P2SH chỉ được thực hiện bằng cách gửi định dạng địa chỉ tiêu chuẩn bắt đầu bằng '3'.
- Băm đảm bảo tính bảo mật và bất biến của giao dịch và dữ liệu giao dịch.
- Công cụ khai thác cũng không yêu cầu nhiều tài nguyên tính toán để xử lý giao dịch P2SH, do đó giao dịch được xác thực nhanh hơn và phí rẻ hơn.
- Chi phí giao dịch của một tập lệnh dài được chuyển cho người nhận, trong đó người nhận phải bao gồm tập lệnh dài để tiêu số tiền đã gửi.

- Trong trường hợp một tập lệnh dài, việc lưu trữ nó sẽ bị hoãn lại cho đến khi nó được sử dụng trong tương lai.
- Việc lưu trữ một tập lệnh dài được chuyển từ đầu ra từ bộ UTXO sang đầu vào được lưu trữ trong chuỗi khối.

***) Nhược điểm:**

- So với Pay-To-MultiSig(P2MS), 2-of-3 mutlisignatured scripts sử dụng mã P2SH lớn hơn nhiều, điều này có nghĩa là chúng sẽ chiếm nhiều dung lượng hơn và sử dụng nhiều băng thông mạng hơn.
- Việc áp dụng nó bị hạn chế do P2SH được phát triển để hỗ trợ các giao dịch đa chữ ký, hầu hết người dùng blockchain chỉ sử dụng một chữ ký duy nhất để ký giao dịch.

Loại	Lần đầu ra mắt	Nguồn cung BTC*	Tính hữu ích*	Mã hóa	Tiền tố	Kí tự
P2PK	01/2009	9% (1.7triệu)	Lỗi thời			
P2PKH	01/2009	43% (8.3 triệu)	Giảm dần	Base 58	1	26-34
P2MS	01/2012	không đáng kể	Lỗi thời			
P2SH	04/2012	24% (4,6 triệu)	Giảm dần	Base 58	3	34
P2WPKH	08/2017	20% (3,8 triệu)	Tăng dần	Bech32	bc1q	42
P2WSH	08/2017	4% (0,8 triệu)	Tăng dần	Bech32	bc1q	62
P2TR	11/2021	0,1% (0,02 triệu)	Tăng dần	Bech32m	bc1p	62

3. Analysis and Reflection

3.1. Kiến thức và kết luận

Một số kiến thức và kỹ năng mới được học hỏi trong quá trình làm bài thực hành này, đó là Bitcoin Script, một ngôn ngữ lập trình dựa trên stack được Bitcoin sử dụng để quy định điều kiện mở khóa các giao dịch.

Bài tập thực hành gồm hai nhiệm vụ chính:

Nhiệm vụ đầu tiên là tạo một script Bitcoin đơn giản để khóa tiền vào một địa chỉ bằng cách sử dụng script Pay-to-Public-Key-Hash (P2PKH). Để làm được điều này, cần phải tạo ra một khóa riêng tư ngẫu nhiên, sau đó dùng nó để sinh ra khóa công khai và địa chỉ Bitcoin. Tiếp theo, cần phải sử dụng một Bitcoin Faucet testnet để nhận BTC testnet vào địa chỉ Bitcoin vừa tạo và viết một script Python để chi tiêu tiền đã khóa.

Nhiệm vụ thứ hai là tạo một script multisig 2-of-2 bằng cách sử dụng các khóa công khai đã cho trước. Để làm được điều này, cần phải tạo ra hai khóa riêng tư ngẫu nhiên, sau đó dùng chúng để sinh ra hai khóa công khai. Tiếp theo, cần phải tạo ra một Spending Script multisig 2-of-2 và một địa chỉ P2SH từ Spending Script. Cuối cùng, cần phải sử dụng một Bitcoin Faucet testnet để nhận BTC testnet vào địa chỉ multisig vừa tạo và viết một script Python để chi tiêu tiền đã khóa từ địa chỉ multisig.

3.2. Ưu điểm và hạn chế của Bitcoin Script

Bitcoin Script có một số ưu điểm và hạn chế như sau:

Ưu điểm:

- ***Đơn giản và dễ hiểu:*** Bitcoin Script chỉ có khoảng 200 mã lệnh, không có vòng lặp, không có biến, không có hàm. Người lập trình chỉ cần sử dụng các mã lệnh cơ bản để xây dựng các điều kiện chi tiêu.

- ***An toàn và bảo mật:*** Bitcoin Script không cho phép thực hiện các tính toán phức tạp, không cho phép truy cập vào bộ nhớ hoặc mạng, không cho phép gọi các hàm bên ngoài. Điều này giúp tránh được các lỗ hổng bảo mật, các cuộc tấn công hoặc các hành vi gian lận. Khi triển khai đúng cách, Bitcoin Script cung cấp một mức độ an toàn

cao đối với các giao dịch, đặc biệt là khi sử dụng tính năng multisig. Sự đồng thuận từ nhiều bên làm tăng khả năng ngăn chặn các tấn công từ bên ngoài hoặc bên trong. Ngoài ra, Bitcoin Script cũng có các mã opcode để kiểm tra tính hợp lệ của giao dịch và loại bỏ các giao dịch không mong muốn.

- ***Linh hoạt và sáng tạo:*** Bitcoin Script cho phép người lập trình tạo ra các loại giao dịch đa dạng, như giao dịch đa chữ ký, giao dịch khóa thời gian, giao dịch hợp đồng thông minh, v.v. Bitcoin Script cũng cho phép sử dụng các mã lệnh mới được thêm vào sau này để nâng cao khả năng của ngôn ngữ. Bitcoin Script còn cho phép người dùng tùy biến các điều kiện chi tiêu funds theo ý muốn, bằng cách kết hợp các mã opcode khác nhau. Ví dụ, người dùng có thể yêu cầu giao dịch chỉ được thực hiện khi đáp ứng một số điều kiện nhất định, như thời gian, địa chỉ, chữ ký, v.v. Điều này tạo ra nhiều loại giao dịch khác nhau, phù hợp với nhu cầu của người dùng.

- ***Multisig và đa chữ ký:*** Bitcoin Script hỗ trợ tính năng multisig, nơi một giao dịch yêu cầu sự đồng thuận từ nhiều bên trước khi được thực hiện. Ví dụ, người dùng có thể tạo ra một giao dịch yêu cầu chữ ký của ít nhất 2 trong 3 bên liên quan, hoặc 3 trong 5 bên, v.v. Điều này tăng cường tính bảo mật và ngăn chặn rủi ro từ mất mát hoặc sự phá vỡ của một khóa riêng.

Hạn chế:

- ***Hạn chế về tính năng:*** Bitcoin Script không thể thực hiện các tính toán phức tạp, không thể lưu trữ hay xử lý dữ liệu, không thể tương tác với các hệ thống bên ngoài. Điều này giới hạn khả năng của Bitcoin Script trong việc xây dựng các ứng dụng phức tạp hay tiên tiến.
- ***Hạn chế về hiệu năng:*** Bitcoin Script phải tuân theo các giới hạn về kích thước, chi phí và thời gian của giao dịch Bitcoin. Điều này có nghĩa là Bitcoin Script không thể sử dụng quá nhiều mã lệnh, không thể tạo ra quá nhiều đầu ra, không thể kéo dài quá nhiều thời gian. Điều này ảnh hưởng đến hiệu năng và khả năng mở rộng của Bitcoin Script.
- ***Phức tạp trong triển khai:*** Việc triển khai các điều kiện giao dịch phức tạp trong Bitcoin Script có thể đòi hỏi sự hiểu biết sâu rộng và kỹ năng lập trình cao. Điều này làm tăng ngưỡng đầu vào cho việc sử dụng nó đối với người không có kinh nghiệm.

Ngoài ra, Bitcoin Script cũng có một số giới hạn về kích thước và tính toán của các kịch bản.

- ***Rủi ro khi sử dụng không đúng cách:*** Việc sử dụng không đúng cách có thể tạo ra rủi ro bảo mật và làm mất funds. Ví dụ, người dùng có thể tạo ra một kịch bản giao dịch có lỗ hổng hoặc sai sót, hoặc quên mất hoặc để lộ các khóa riêng. Sự hiểu biết sâu rộng về cách triển khai và quản lý Bitcoin Script là quan trọng để tránh các vấn đề này.

3.3. Ứng dụng thực tế và đánh giá

- ***Quản lý tài khoản nhóm:***

  Mô *tả***:** Bitcoin Script có thể được sử dụng để triển khai các ví đa chữ ký (multisig wallets), nơi mà một giao dịch yêu cầu sự đồng thuận từ nhiều thành viên.

  Đánh *giá***:** Ứng dụng này mang lại tính an toàn cao cho quản lý tài khoản nhóm, ngăn chặn rủi ro từ việc mất mát khóa riêng và đồng thời tạo ra quá trình đồng thuận linh hoạt.

- ***Giao dịch đa bên tham gia:***

  Mô tả: Bitcoin Script có thể được áp dụng trong các thương vụ kinh doanh đòi hỏi sự tham gia từ nhiều bên để thực hiện một giao dịch.

  Đánh giá: Ứng dụng này mở ra khả năng xây dựng các hợp đồng thông minh phức tạp, nơi mà sự đồng thuận từ nhiều bên tham gia là quyết định quan trọng.

- ***Tạo ví chia sẻ (Shared Wallets):***

  Mô tả: Bitcoin Script cho phép tạo ra các điều kiện giao dịch đặc biệt, ví dụ như việc tạo ví chia sẻ giữa các thành viên của gia đình hoặc đồng nghiệp.

  *Đánh giá***:** Ứng dụng này hữu ích trong quản lý tài chính gia đình hoặc nhóm làm việc, tăng tính minh bạch và ngăn chặn việc chi tiêu không hợp lý.

- ***Thực tiện điều kiện giao dịch cụ thể:***

  Mô tả: Bitcoin Script có thể được sử dụng để xác định điều kiện cụ thể cho việc thực hiện một giao dịch, chẳng hạn như tự động kích hoạt giao dịch trong một thời điểm nhất định.

Đánh giá: Ứng dụng này hữu ích trong các kịch bản nơi việc kích hoạt tự động của giao dịch là quan trọng, chẳng hạn như trong các hợp đồng tương lai hoặc việc quản lý quảng cáo.

Bitcoin Script, với tính linh hoạt và tính năng multisig, mang lại nhiều ứng dụng thực tế có giá trị. Các ứng dụng này không chỉ cải thiện tính bảo mật trong quản lý tài chính mà còn mở ra các khả năng mới trong lĩnh vực thương mại và hợp đồng thông minh. Tuy nhiên, việc triển khai và hiểu biết sâu rộng về mã nguồn mở này là quan trọng để tận dụng đầy đủ tiềm năng của nó.

4. Tài liệu tham khảo

https://academy.binance.com/en/articles/an-introduction-to-bitcoin-script

https://iq.opengenus.org/p2pkh/

https://bitcoin.stackexchange.com/questions/36695/which-security-threats-in-p2pkh-lead-to-the-use-of-p2sh

https://iq.opengenus.org/p2sh-in-bitcoin/

https://notebook.community/1200wd/bitcoinlib/docs/bitcoinlib-10-minutes

https://bitcoinlib.readthedocs.io/en/latest/

https://github.com/1200wd/bitcoinlib