Mục lục

[1. Introduction 2](#_Toc82793180)

[1.1. Bitcoin 2](#_Toc82793181)

[1.2. Ví bitcoin 2](#_Toc82793182)

[1.3. Client 2](#_Toc82793183)

[2. How Bitcoin Works 2](#_Toc82793184)

[2.1. Giao dịch bitcoin (bitcoin transaction) 3](#_Toc82793185)

[2.2. Cấu thành một giao dịch (Constructing a Transaction) 4](#_Toc82793186)

[2.3. Ghi lại giao dịch vào sổ cái (ledger) 4](#_Toc82793187)

[2.4. Khai thác bitcoin (bitcoin mining) 4](#_Toc82793188)

[2.5. Mining transaction in blocks 4](#_Toc82793189)

[3. Bitcoin Core: The Reference Implementation 5](#_Toc82793190)

[4. Keys, Addresses 5](#_Toc82793191)

[4.1. Public Key Cryptography and Cryptocurrency 6](#_Toc82793192)

[4.2. Private and public keys 6](#_Toc82793193)

[4.3. Bitcoin addresses 8](#_Toc82793194)

[4.4. Key formats 11](#_Toc82793195)

[4.5. Advanced Keys and Addresses 13](#_Toc82793196)

[5. Wallet 14](#_Toc82793197)

[5.1. Overview 14](#_Toc82793198)

[5.2. Wallet Technology Details 16](#_Toc82793199)

[6. Transaction 22](#_Toc82793200)

[6.1. Transaction Outputs and Inputs 22](#_Toc82793201)

[6.2. Transaction Scripts and Script Language 25](#_Toc82793202)

[6.4. Bitcoin Addresses, Balances, and Other Abstractions 32](#_Toc82793203)

[7. Advanced Transactions and Scripting 32](#_Toc82793204)

[7.1. Multisignature 32](#_Toc82793205)

[7.2. Pay-to-Script-Hash (P2SH) 33](#_Toc82793206)

[7.3. Data Recording Output (RETURN) 35](#_Toc82793207)

[7.4. Timelocks 35](#_Toc82793208)

[8. The Bitcoin Network 37](#_Toc82793209)

[8.1. Peer-to-Peer Network Architecture 37](#_Toc82793210)

[8.2. The Extended Bitcoin Network 38](#_Toc82793211)

[8.3. Bitcoin Relay Networks 40](#_Toc82793212)

[8.4. Network Discovery 41](#_Toc82793213)

[8.5. Full Nodes 43](#_Toc82793214)

[8.6. Exchanging “Inventory” 43](#_Toc82793215)

[8.7. Simplifed Payment Verifcation (SPV) Nodes 44](#_Toc82793216)

[8.8. Bloom filter 45](#_Toc82793217)

[Chương 9, 10 (đọc sách hiểu rõ hơn). 48](#_Toc82793218)

1. Introduction
   1. Bitcoin

Bitcoin là một tập hợp các nền tảng và công nghệ cấu thành lên nền tảng hệ sinh thái tiền kỹ thuật số (digital money ecosysterm).

Những đơn vị bitcoin được sử dụng để lưu trữ và truyền tải những giá trị (value) thông qua các thành viên trong một mạng bitcoin.

Người sử dụng (bitcoin users) trao đổi với nhau bằng bitcoin protocol thông qua internet.

Bitcoin hoàn toàn ảo hóa, người sử dụng các keys của họ (được lư trong ví điện tử) để xác thực trong quá trình truyền tải bitcoin, họ dùng key để ký lên những giao dịch (transactions) giúp họ mở khóa để lấy giá trị, dùng nó bằng cách chuyển nó tới người sử dụng khác.

Bitcoin là một hệ thống phân phối, pear-to-pear. Mỗi Bitcoin được tạo bởi một tiến trình là “mining” (tìm lời giải cho một vấn đề toán học trong suốt quá trình xử lý giao dịch bitcoin).

Bitcoin mining giúp phân quyền phát hành tiền tệ (currency-issuance) và các chức năng thanh toán bù trừ (clearing function) của ngân hàng trung ương (central bank), thay thế nhu cầu bất kỳ ngân hàng trung ương nào.

Bitcoin bao gồm:

* Một mạng phân phối ngang hàng (blockchain protocal)
* Một số cái giao dịch công khai (blockchain)
* Một tập luật quy định xác thực giao dịch độc lập và phát hành tiền tệ (consensus rules)
* Một cơ chế để đạt được sự đồng thuận phi tập trung toàn cầu về một blockchain hợp lệ (Proof-of-Work algorithm).
  1. Ví bitcoin

Ví bitcoin tập chung vào một platform cụ thể cho những nhu cầu sử dụng cụ thể được phân loại như sau:

* Desktop wallet: chạy trên window hoặc macos, an toàn thấp và khả năng tính toán kém
* Mobile wallet: chạy trên thiết bị di động, đơn giản, dễ dàng sử dụng
* Web wallet: truy cập thông qua web browser lưu trữ ví trong server của bên thứ ba, dễ dàng sử dụng nhưng không nên lưu một lượng lớn bitcoin vào hệ thống bên thứ 3.
* Harware waller: thiết bị quản lý độc lập an toàn trên phần cứng có mục đích đặc biệt, rất an toàn và phù hợp lưu trữ lớn lượn bitcoin
* Paper wallet: các keys quản lý bitcoin được in vào một nơi lưu trữ dài hạn – cold storage (trên gỗ, nhựa,.. như thẻ ngân hàng), rất an toàn và yêu cầu kỹ thuật thấp.
  1. Client
* Full-node clident: lưu trữ hầu hết lịch sử giao dịch bitcoin, quản lý ví của user và có thể tạo các giao dịch trực tiếp trên bitcoin network.
* Lightweight client: kết nối với full-node để truy cập thông tin giao dịch nhưng lưu trữ cục bộ ví người dùng, khởi tạo, xác thực, truyền giao dịch độc lập.
* Third-party API client: kết nối với bitcoin thông qua hệ thống bên thứ ba.

1. How Bitcoin Works
   1. Giao dịch bitcoin (bitcoin transaction)

Một giao dịch thông báo với mạng rằng một lượng bitcoin, sở hữu bởi một người, đã được xác thưc chuyển tới một người sở hữu khác.

Mỗi giao dịch đều có một hay nhiều đầu vào – input – như các khoản ghi nợ vào tài khoản bitcoin cũng như có một hay nhiều đầu ra – output – như các khoản tín dụng thêm vào một tài khoản bitcoin. Tổng số output nhỏ hơn so với input vì phải trả thêm 1 khoản phí giao dịch là Transition fee.

Các giao dịch được định hình thành chuỗi nơi các đầu vào từ giao dịch gần nhất tương ứng với những đầu ra của giao dịch hiện thời.

Vì đầu vào giao dịch (Những ghi chú về tiền tệ) không thể chia được nên nếu xảy ra trường hợp khi trao đổi một giao dịch mà bên mua trả nhiều tiền hơn số tiền giao dịch thì đầu ra cần ghi lại hai địa chỉ để chuyển tiền là địa chỉ bên bán và địa chỉ nơi thu tiền thừa – được gọi là change address.

Một số mô hình giao dịch:

A picture containing diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

* 1. Cấu thành một giao dịch (Constructing a Transaction)

Bước 1: tìm đúng đầu vào (right inputs)

Đầu ra của giao dịch trước sẽ là đầu vào của giao dịch ngay sau đó, một ứng dụng ví bitcoin chạy dưới dạng full-node client sẽ chứa hầu hết bản sao của những đầu ra (ghi chú tiền tệ) chưa được sử dụng từ các giao dịch trong chuỗi blockchain và theo dõi những đầu ra thuộc về địa chỉ trong ví của mình. Từ đây sẽ chọn những output phù hợp để làm input cho giao dịch sau.

Bước 2: tạo output

Đầu ra sẽ được tạo dưới dạng một script (bên nhận chỉ nhận được giá trị khi đưa ra một giải pháp cho script đó), ngoài các giá trị giao dịch còn có một phí là transition fee. Phí này sẽ được thu bởi miner như một khoản chi cho việc xác thực cũng như ghi lại lịch sử giao dịch,…

Sau khi tạo đầu vào và đầu ra giao dịch được trao đổi trên mạng.

* 1. Ghi lại giao dịch vào sổ cái (ledger)

Mạng bitcoin là một mạng ngang hàng (p2p network) với mục tiêu lan truyền giao dịch và các block cho tất cả những người tham gia. Bất kỳ một bitcoin node nào nhận được một giao dịch đã được xác nhận mà nó chưa được thấy trước đó sẽ ngay lập tức truyền nó tới các node khác mà nó kết nối (flooding).

* 1. Khai thác bitcoin (bitcoin mining)

Một giao dịch sẽ không thể trở thành một phần của blockchain nếu nó khoogn được xác nhận và cho một block bởi một tiến trình là mining.

Sự tin cậy dựa trên sự tính toán:

+ Mining nodes sẽ xác nhận tất cả các giao dịch thông qua consensus rules.

+ Mining sẽ tạo bitcoin mới cho mỗi block.

Mining sẽ nhận được một khoản giữa phí và tiền thưởng (dưới dạng bitcoin mới và transaction fees), điều này chỉ đạt được khi xác nhận hết các giao dịch.

Qúa trình mining là một cuộc thi giữa các miners, tất cả cùng nhau tìm ra giải pháp xác nhận cho một các giao dịch, Proof-of-Work (PoW).

* 1. Mining transaction in blocks

Giao dịch sau khi được tạo mà chưa xác thực sẽ được chuyển vào một hồ (pool) gồm các giao dịch chưa xác thực.

Những miners sẽ tạo một khối mới và thêm những giao dịch chưa xác thực – ưu tiên về phí và những khía cạnh khác - vào khối đó, sau đó sẽ chứng minh xác thực cái khối này (PoW). Các khối là những candiate blocks, khi một miner tìm ra giải pháp – solution thì sẽ công bố nó trên mạng. Những miners khác xác nhận và bắt đầu một cuộc đua mới để tạo block tiếp theo. Block thắng cuộc sẽ thành một phần của blockchain.

1. Bitcoin Core: The Reference Implementation

Bitcoin Core là phần triển khai tham chiếu của hệ thống bitcoin, có nghĩa là nó là tài liệu tham khảo có thẩm quyền về cách thức từng phần của công nghệ nên được thực hiện.

Diagram

Description automatically generated

1. Keys, Addresses

Quyền sở hữu bitcoin được xác lập thông qua digital keys, bitcoin addresses, digital signatures. Digital keys (khóa điện tử) không được lưu trên network mà được khởi tạo, lưu trữ trên wallet (ví), chúng được phát triển, quản lý độc lập với bitcoin protocol bởi wallet software. Keys kích hoạt nhiều thuộc tính thú vị của bit coin bao gồm phi tập chung hóa sự tin cậy và kiểm soát (trust and control), chức thực quyền sở hữu (ownership attestation) và mô hình bảo mật mật mã (the cryptographic-proof security model).

Có hai loại keys là khóa bảo mật (private key) và khóa công khai (public key). Private key dùng để tạo digital signature giúp xác thực giao dịch, kiểm soát tài khoản, bitcoin được lưu trữ trong wallet file và quản lý bởi bitcoin wallet software. Bitcoin address được sử dụng để trừ tượng hóa nơi nhận tiền, làm linh hoạt các địa điểm của giao dịch, giống như một paper check: khoản phí được trả cho những tài khoản nào (của người, của công ty), trả bằng tiền mặt hay trả bằng hóa đơn. Bitcoin addresses được xây dựng từ public key tương ứng (public key của nơi nhận giao dịch) nó có thể đại diện cho public keys hoặc những người thụ hưởng (beneficiaries) khác dưới dạng scripts.

* 1. Public Key Cryptography and Cryptocurrency

Bitcoin sử dụng elliptic curve multiplication như nền tảng cho mật mã mã khóa của nó vì đây là một thuật toán chỉ tính toán dễ dàng theo một chiều và khó có thể tính toán ngược trở lại.

Từ một public key cryptography, chúng ta sẽ khở tạo một private key và từ private key này sẽ phát triển một public key duy nhất tạo thành một cặp keys. Public key dùng để nhận tiền (funds), private key được sử dụng để ký lên giao dịch cho phép chuyển tiền (spend the funds).

* 1. Private and public keys

Private key là một số được lấy ngẫu nhiên, từ private key chúng ta sử dụng elliptic curve multiplication để khởi tạo public key. Từ public key chúng ta sử dụng hàm băm mã hóa một chiều (one-way cryptographic hash function) để tạo lên bitcoin address.

Diagram, text, shape

Description automatically generated

Private key dùng để tạo chữ ký điện tử (digital signature) và những người có public key từ private key có thể xác nhận giao dịch của người sử hữu private key thông qua public key và digital signature. Như vậy bất kể ai cũng có thể xác nhận chữ ký của bất kỳ giao dịch nào trong khi chỉ duy nhất những người sở hữu private key mới có thể khởi tạo chữ ký để xác nhận.

Cách khởi tạo private key:

* Chọn nguồn entropy, randomness an toàn
* Private key có thể là một số ngẫu nhiên từ 1 đến n-1 (n = 1.158 \* 1077, nhỏ hơn một chút so với 2^256) với n được định nghĩa như là bậc (order) của đường cong elliptic sử dụng bởi bitcoin.
* Chúng ta chọn ngẫu nhiên 1 số 256 bit và kiểm tra xem nó có nhỏ hơn n-1 không, nếu có thì chúng ta có một public key hợp lệ, còn không thì chọn lại.

Public key được tính toán từ private key sử dụng elliptic curve multiplicationvới công thức K=k\*G (K là public key, k là private key còn G là một điểm cố định định trên ellipstic curve).

Elliptic Curve Cryptography:

* Bitcoin sẽ sử dụng một đường cong ellipstic đặc biệt và khở tạo những hằng số toán học, được định nghĩa dưới tiêu chuẩn secp256k1. Và đường cong secp256k1 được định nghĩa bởi công thức sau:

Text, letter

Description automatically generated

Trong đó mod p là chia lấy dư số nguyên tố p chỉ ra rằng đường cong này nằm trên một trường hữu hạn của bậc nguyên tố p. Có thể viết là Fp với p = 2256 – 232 – 29 – 28 – 27 – 26 – 24 – 1. Đường cong được định nghĩa bở bậc số nguyên tố nên có dạng là cá nút cách biệt.

Chart

Description automatically generated

* Các tính chất của ellipstic curve:

+ Có một điểm là ‘point at infinity’ gần tương ứng vai trò của số 0 trong phép cộng thỉnh thoảng biểu diễn là x=y=0.

+ Nếu 2 điểm P1 và P2 thuộc ellipstic curve thì điểm P3=P1+P2 cũng thuộc ellipstic curve.

+ P3 được tính bởi đường kẻ giữa P1 và P2, đường này sẽ giao với ellipstic curve tại 2 điểm là P’3(x, y), lấy đối xứng qua trục x ta được tọa độ P3(x, -y).

+ Nếu P1, P2 là cùng 1 điểm thì đường kẻ giữa hai điểm sẽ là đường tiếp tuyến của ellipstic curve.

+ Nếu P1 và P2 cùng tọa độ x nhưng khác y thì đường tiếp tuyến (tangent line) sẽ là một đường thẳng đứng, khi đó P3 là một ‘point at infinity’.

+ Nếu P1 là ‘point at infinity’ thì P1+P2=P2 và ngược lại

+ (A + B) + C = A + (B + C).

+ kP = P + P + P + … + P.

Khởi tạo public key K=k\*G, chúng ta chỉ có thể tính một chiều từ k sang K nhưng không thể tính ngược lại.

Diagram

Description automatically generated

* 1. Bitcoin addresses

Bitcoin address là một chuỗi chữ số và ký tự được chia sẻ với mọi người muốn gửi tiền cho bạn. Địa chỉ được tạo bở public key với đầu ra là một chuỗi các số, chữ cái bắt đầu bằng ‘1’.

Thuật toán được sử dụng để tính toán bitcoin address từ public key là Secure Hash Algorithm (SHA) và the RACE Integrity Primitives Evaluation Message Digest (RIPEMD), cụ thể là SHA256 and RIPEMD160.

A = RIPEMD160 (SHA256(K))

Bitcoin addresses hầu hết được mã hóa dưới dạng “Base58Check”, sử dụng 58 ký tự với 1 checksum giúp con người dễ đọc, tránh nhập nhằng và bảo vệ chống lỗi trong địa chỉ giao dịch và đường ra.

Diagram

Description automatically generated

Base58 and Base58Check Encoding:

* Base 64 được thể hiện bở 26 chữ cái viết thường, 26 chữ viết hoa, 10 số, 2 ký tự ‘+’ và ‘/’ để chuyển đổi dữ liệu nhị phân sang dạng text như email.
* Base 58 là tập con của Base 64 ngoại trừ 0 (number zero), O (capital o), l (lower L), I (capital i), and the symbols “+” and “/”.
* Base 58 check là một dạng mã hóa của Base 58, một checksum gồm 4 bytes được thêm vào cuối dữ liệu được mã hóa, checksum dùng để phát hiện, chống lại giao dịch lỗi. Phần mềm giải mã sẽ tính toán checksum và so sánh nó với checksum trong code, nếu không trùng nhau thì Base58check data là không xác thực.
* Để chuyển sang dạng Base58Check thì đầu tiên chúng ta thêm prefix (đầu) vào data – version byte – xác nhận kiểu dữ liệu mã hóa (bảng 4.1). Checksum sau đó được tính như sau

checksum = SHA256(SHA256(prefix+data))

Kết quả thu được 32 byte băm nhưng ta chỉ lấy 4 byte đầu là m checksum và gán vào cuối data.

A picture containing graphical user interface

Description automatically generated

Text

Description automatically generated with medium confidence

Graphical user interface, text

Description automatically generated

* 1. Key formats

Private key formats:

* Hexa‐ decimal and raw binary formats thường được sử dụng trong phần mềm và hiếm khi được đưa ra cho người dùng xem.
* WIF được sử dụng cho việc nhập/ xuất các keys giữa các ví và thường được sử dụng trong QR code (barcode) đại diện cho private keys.

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text

Description automatically generated

Public key formats:

* Được biểu diễn dưới hai dạng là compressed (nén) và uncompressed (không nén) key.
* Public key sau khi khởi tạo sẽ là 1 điểm điên ellipstic curve có tọa độ x, y

A screenshot of a computer

Description automatically generated with low confidence

* Khi đó nếu gán prefix là 04 thì đây là 1 uncompressed public key còn 02, 03 là compress public key

Text

Description automatically generated with medium confidence

* Do x, y đều được tính bởi công thức y^2 mod p = (x^3 + 7) mod p nên từ x chúng ta có thể tính ra y, việc này giúp chúng ta giảm thiểu dung lượng nhớ cho public key khi key chỉ cần lưu x, việc tại sao compressed key có 2 prefix là 02 và 03 là bởi y^2 cho 2 giá trị là âm và dương, trong tính toán ellipstic curve thì y sẽ là chẵn và lẻ. Vì vậy nếu prefix là 02 thì y là chẵn còn 03 thì y là lẻ.

Diagram

Description automatically generated

Khi đó với x và y là lẻ thì K sẽ là



Compressed private key:

* Nếu sử dụng 1 key cho cả uncompressed key và compressed key thì sẽ xảy ra việc từ 1 public key sẽ đưa ra hai giá trị khác nhau trỏ đến 1 địa chỉ gây nhầm lẫn, ngoài ra nếu private key để mặc định cho compressed public key thì khi di chuyển sang ví khác có version cũ hơn sử dụng key đó cho uncompressed public key thì cũng dẫn đến một nhầm lẫn như trên. Vì vậy compressed private key – private key chỉ được sử dụng cho compressed public key - sinh ra cùng với uncompressed private key - private key chỉ được sử dụng cho uncompressed public key.

Text

Description automatically generated

Hex-compressed private key có nhiều hơn 1 byte 01 cở cuối, WIF-compressed có K và L ở prefix, WIF thường chỉ có 5 ở prefix.

* 1. Advanced Keys and Addresses

Encrypted Private Keys (BIP-38): BIP38 đề xuất một tiêu chuẩn chung cho việc mã hóa private keys với một passphrase (mật khẩu) và mã hóa chúng với Base58Check vì vậy chúng có thể lưu trữ một cách bảo mật trong backup media, được vận chuyển an toàn giữa các ví và giữ trong mọi điều kiện mà key có thể bị tiết lộ.

Text

Description automatically generated

Pay-to-Script Hash (P2SH) and Multisig Addresses:

* Với pay-to-public-key-hash (P2PKH) thì bitcoin addresses bắt đầu với ‘1’ được dựng từ 1 public key – khởi tạo bởi 1 private key.
* Với Pay-to-Script Hash (P2SH) thì bitcoin addresses bắt đầu với ‘3’ thiết kế địa chỉ thụ hưởng 1 giao dịch bitcoin là 1 mã băm (hash) của 1 script thay vì chủ sở hữu public key.
* Một P2SH address được khởi tạo từ một transaction script, nơi chỉ định ai có thể sử dụng transaction output: script hash = RIPEMD160(SHA256(script))
* P2SH thường được sử dụng cho multi-signature address script (script cho một tập nhiều chữ ký để chứng thực chủ sở hữu). Từ N keys sẽ cần tạo ra M chữ ký (M-of-N multisig), M<=N.
  1. Vanity address:

Vanity address là bitcoin address được xác nhận, chứa ký tự mà con người đọc được

Vd: 1LoveBPzzD72PUXLzCkYAtGFYmK5vYNR33.

Vanity address yêu cầu xây dựng và thử nghiệm hàng tỷ ứng cử viên private keys cho đến khi một bitcoin address thỏa mã yêu cầu được tìm thấy

Khì tìm được vanity address thì private key từ nó sẽ được người sở hữu sử dụng.

Table

Description automatically generated

Vanity address security (xem sách).

1. Wallet
   1. Overview

Bitcoin wallet không chứa bitcoin mà chứa key. Coin được lưu trên blockchain trong bitcoin network, người sử dụng kiểm soát coin bằng việc ký lên giao dịch key trong ví của mình, bitcoin wallet là 1 keychain.

Có 2 loại ví là:

* Nondeterministic (Random) wallets: là một tập gồm các private key được xây dựng một cách ngẫu nhiên. Việc bất lợi là mình cần phải lưu bản sao cảu những key này để tránh sử dụng lại dễ gây mất an toàn bảo mật nghĩa là wallet phải được backed up thường xuyên.

A picture containing icon

Description automatically generated

* Deterministic (seeded) wallets: chứa private keys được xây dựng dựa trên 1 hạt chung (common seed) thông qua một hàm băm một chiều (one-way hash function). Seed là một số được chọn ngẫu nhiên kết hợn với những dữ liệu khác như một số chỉ mục (index number) hoặc ‘chain code’ để tạo private key. Từ seed có thể khôi phục được những key được tạo từ nó (a single backup at creation time).

A picture containing diagram

Description automatically generated

* HD wallet (BIP-32/BIP-44): là một dạng của deterministic wallets định nghĩa bởi chuẩn BIP-32. HD wallets chứa keys được tạo từ một cấu trúc cây, cụ thể một key cha mẹ có thể tạo ra một chuỗi những key con, mỗi một key con có thể tạo ra các key cháu, cứ thế và không giới hạn độ sâu của cây.

Diagram

Description automatically generated

HD wallets có 2 ưu điểm chính đó là:

+ cấu trúc cây có thể được sử dụng để thể hiện ý nghĩa tổ chức bổ sung, ví dụ một nhánh key dùng để nhận tiền đầu vào, một nhánh khác dùng để xuất tiền đầu ra trong giao dịch.

+ người sử dụng có thể tạo một chuỗi public keys mà không cần truy cập đến private key tương ứng.

* Seeds and Mnemonic Codes (BIP-39): HD wallet tốt cho việc quản lý nhiều keys và addresses. Nhưng việc kết hợp với một cách chuẩn hóa để tạo ra seed từ một chuỗi từ tiếng anh (mnemonic - một dạng chuẩn định nghĩa bởi BIP-39) thì càng dễ sử dụng khi có thể chếp lại, lưu trong paper, đọc, xuất, khôi phục,… dễ dàng hơn. So sánh trước và sau kết hợp chuẩn mnemonic:

Text

Description automatically generated

* Một số quy chuẩn về công nghệ bitcoin wallet:

• Mnemonic code words, based on BIP-39

• HD wallets, based on BIP-32

• Multipurpose HD wallet structure, based on BIP-43

• Multicurrency and multiaccount wallets, based on BIP-44

* 1. Wallet Technology Details

Mnemonic Code Words:

* Tạo mnemonic words: ví bắt đầu từ nguồn entropy, cộng thêm check-sum và kết nối entropy với 1 danh sách từ

+ Tạo 1 chuỗi ngẫu nhiên (entroy) từ 128 đến 256 bits.

+ Tạo một checksum của chuỗi ngẫu nhiên bằng việc lấy (entropy length/32) bits đầu tiên từ SHA256 hash của nó.

+ Cộng checksum vào phần cuối của chuỗi ngẫu nhiên.

+ Chia chuỗi thành mỗi phần 11 bits.

+ Kết nối mỗi 11 bit giá trị với 1 từ trong từ điển cho trước gồm 2048 chữ.

+ mnemonic code là chuỗi từ được tạo.

Graphical user interface, text, application

Description automatically generated

Table

Description automatically generated

* From mnemonic to seed: mnemonic words -> entropy -> seed (512-bit từ the key-stretching function PBKDF2). Hàm key-stretching gồm hai thành phần là mnemonic và salt. Salt được thêm vào để tăng độ khó cho việc xây dựng một bảng tra cứu (lookup table) cho phép một brute-force attack. Một vai trò khác của salt là cho phép một passphrase (mật khẩu) phục vụ như một yếu tố bảo mật bổ sung cho seed.

Graphical user interface, diagram, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

* Optional passphrase in BIP-39: nếu một passphrase được sử dụng thì stretching function sẽ tạo ra một seed khác từ một mnemonic so với seed đầu, không có passphrase sai (wrong). Mật khẩu tùy chọn (optional passphrase) tạo ra 2 đặc tính quan trọng:

+ Nhân tố thứ hai (thứ gì đó được lưu trữ) làm cho một mnemonic trở nên vô dụng, bảo vệ các bản sao lưu mnemonic khỏi sự xâm phạm của kẻ trộm.

+ Một hình thức từ chối hợp lý hoặc “duress wallet”, một passphrase được sử dụng cho ví có giá trị thấp giúp đánh lạc hướng những kẻ tấn công khỏi vì thật ‘real’ chứa nhiều giá trị.

Những rủi ro gặp phải:

+ Nếu chủ sở hữu ví không có khả năng hoặc đã chết và không ai khác biết passphrase, seed sẽ vô dụng và tất cả số tiền được lưu trữ trong ví sẽ bị mất vĩnh viễn.

+ Ngược lại, nếu chủ sở hữu sao lưu passphrase ở cùng một nơi với seed, nó

đánh bại mục đích của nhân tố thứ hai.

Creating an HD Wallet from the seed:

Diagram

Description automatically generated

Master private key (m) dùng để tạo những master public key (M) thông qua công thức ellipstic curve multiplication m\*G.

Chain code (c) được sử dụng để tạo entropy in trong hàm tạo child keys từ parent keys.

* Private child key derivation: HD wallets sử dụng hàm child key derivation (CKD) để tạo child keys từ parent keys, CKD dựa trên one-way hash function bao gồm tham số: một parent key hoặc public key (ECDSA uncompressed key), một seed hay còn gọi là chain code, một index number. Thay đổi index number giúp mở rộng nhánh cha và tạo nên các con khác theo một chuỗi như con 0, con 1, con 2,…

Diagram

Description automatically generated

* Using derived child keys: một khóa con không thể tính ngược lại khóa cha cũng như tính cách khóa con bên cạnh. Chỉ có parent key và chain code mới có thể tạo ra các child key. Một child key không thể tạo ra grandchildren key nếu không có child chain code, cần có cả hai để tạo ra một nhánh con mới.
* Extended keys: là sự kết hợp của (256 bits) key và (256 bits) chain code thành một chuỗi 512 bits. Extended private key có thể tạo ra các child private key từ đó tạo ra các child public key qua đó tạo ra một nhánh hoàn chỉnh, còn extended public key chỉ có thể tạo ra các child public key qua đó chỉ có thể tạo một nhánh chỉ gồm các public key. Extended key sử dụng Base58Check để mã hóa và gán prefix ‘xprv’ (private) và ‘xpub’ (public) để phân biệt hai loại extended keys.

A screenshot of a computer

Description automatically generated with medium confidence

* Public child key derivation: public child key có thể được dựng lên bới hai cách từ parent public key hoặc từ một child private key. Ngoài ra có thể sử dụng extended public key, việc sử dụng này tạo ra những triển khai khởi tạo khóa công khai một cách an toàn, cụ thể là server hoặc ứng dụng chỉ có bản sao của extended public key để tạo khóa và không có một private key nào.

Diagram

Description automatically generated

* Hardened child key derivation: việc tạo một nhánh mới gồm các public child key từ một xpuv (extended public key) tạo ra một lỗ hổng. Vì xpuv có chứa chain code, nếu một child private key bị tiết lộ thì nó có thể sử dụng chain code để tạo ra những child private keys khác. Để khắc phục điều này HD wallets sử dụng hàm hardened derivation, hàm này sẽ phá vỡ ‘break’ mối quan hệ giữa parent public key và child chain code bằng cash sử dụng parent private key để tạo ra child chain code thay vì parent public key.

Diagram

Description automatically generated

* Index numbers for normal and hardened derivation: index number là một số nguyên 32 bits dùng để phân biệt các keys được khở tạo từ derivation function, chúng được chia làm 2 khoảng, từ 0-2^31-1 (0x0 to 0x7FFFFFFF) được sử dụng cho khởi tạo bình thường, từ 2^31 đến 2^32 – 1 (0x80000000 to 0xFFFFFFFF) được sử dụng cho hardened derivation. Vì vậy index number nhỏ hơn 2^31 thì là normal child còn từ 2^31 trở lên sẽ là hardened child. Hardened child đầu tiên là 0x80000000 sẽ biểu diễn là 0’, từ hardened child thứ 2 trở đi sẽ biểu diễn là i’ như child normal và được hiểu là con thứ 2^31 + i.
* HD wallet key identifer (path): Các keys trong HD wallet được xác định bởi một đường với mỗi level của cây cách nhau bởi một ‘/’. Private keys lấy từ master private key bắt đầu bởi ‘m’, tương tự public key bắt đầu bởi ‘M’. Con đầu tiên của private key là m/0, public key là M/0,... Như vậy với một đường là m/x/y/z cho biết con thứ z của key m/x/y, là con thứ y của key m/x, là con thứ x của private key m.

Graphical user interface, text, application

Description automatically generated

* Navigating the HD wallet tree structure: do số lượng child key và độ sâu của cây là vô cùng lớn nên cần một tiêu chuẩn để có thể tìm kiếm và lấy thông tin của key. Ở đây ta sử dụng dạng BIP-44 có cấu trúc như sau:



Purpose luôn được đặt là 44, coin\_type chỉ định loại crytocurrency coin cho phép đa tiền tệ (multicurruncy) HD wallets nơi mà mỗi loại tiền tệ có một cây con riêng của nó. Account cho phép người sử dụng chia nhỏ ví của họ thành nhiều acoount con logic tách biệt được sử dụng với nhwunxg mục đích khác nhau. Change, HD wallet chia ra làm 2 cây con một cho khởi tạo những địa chỉ nhận, một cho khởi tạo thay đổi địa chỉ. Những địa chỉ được sử dụng được tạo bởi HD wallet như phần con của level Change, đây à phân address\_index.

Text

Description automatically generated

1. Transaction
   1. Transaction Outputs and Inputs

Đầu ra của giao dịch là một khối bitcoin curruncy không thể chia tách, được lưu trong blockchain và được xác nhận trong toàn bộ network. Bitcoin full nodes sẽ tìm tất cả những outputs sẵn sàng và có thể sử dụng được – unspent transaction output (UTXO). Qúa trình ví tiếp nhận ‘received’ bitcoin còn được hiểu là ví đã tìm được những UTXO có thể sử dụng ‘spent’. Số dư bitcoin của người dùng (user’s bitcoin ‘balance’) là tổng số UTXO mà người dùng được sử dụng, bằng cách quét toàn bộ blockchain và kết hợp những giá trị của bất kỳ UTXO nào mà nó được quyền sử dụng với key của nó, ví sẽ tính được số dư.

Mội transaction output có thể có một giá trị (số nguyên) tùy ý được gọi là bội số satoshi, như dollars được chia nhỏ thành cent. Do đầu ra là rời rạc và không thể chia nhỏ nên nếu bạn có 1 UTXO với 20 bitcoin và muốn chi cho 1 bitcon thì giao dịch của bạn phải có 20-bitcon UTXO (bằng cách kết hợp những cái con hoặc lấy hẳn cái có giá trị bằng, lớn hơn 20 bitcoin) với 2 đầu ra: 1 bitcoin cho nơi bạn muốn chi, 19 bitcoin còn lại chuyển lại vào ví.

Coinbase transaction - giao dịch (transaction) đầu tiên của một khối block, giao dịch này được đặt bởi ‘winning’ miner và tạo ra nhánh bitcoin mới mà miner này có thể nhận được phần thưởng cho mining. Coinbase transaction không chứa UTXO, thay vào đó đâu vào nó là ‘coinbase’.

* Transaction output: tất cả giao dịch bitcoin đều tạo outputs và chúng được lưu lại trong sổ cái bitcoin (bitcoin ladger). Những outputs này sẽ tạo ra một khối bitcoin có thể chi tiêu được - UTXO, được nhận biết bởi toàn bộ network và cho phép người sở hữu sử dụng trong những giao dịch tương lai. Transaction output gồm 2 phẩn:

+ Số lượng bitcoin, bội số của satoshis, the smallist bitcoin units.

+ Một crytographic puzzle (locking script, witness script, scriptPubkey) đưa ra điều kiện để sử dụng output.

* Khi các giao dịch được truyền qua mạng hoặc trao đổi giữa các ứng dụng, chúng sẽ được tuần tự hóa (serialize). Tuần tự hóa là quá trình chuyển đổi biểu diễn bên trong của cấu trúc dữ liệu thành một định dạng có thể được truyền từng byte một, còn được gọi là luồng byte (byte stream).

Text

Description automatically generated

Hầu hết các thư viện bitcoin và framework không thể lưu dữ liệu giao dịch theo byte stream mà lưu giao dịch thành một cấu trúc dữ liệu (data structures (usually object-oriented structures)). Qúa trình chuyển đổi từ byte stream về data structures của một giao dịch là deserialization hay transaction parsing.

Text

Description automatically generated

Transaction input chỉ định những UTXO nào sẽ được sử dụng và cung cấp minh chứng sở hữu thông qua unlocking script. Như vậy đầu vào của giao dịch bao gồm 4 phần. Đầu tiên là id của giao dịch chứa những UTXO có thể được sử dụng. Tiếp theo là output index, con trỏ trỏ tới một UTXO bằng tham chiếu tới mã băm giao dịch và số thứ tự nơi mà UTXO được lưu trong blockchain. Phần thứ 3 là unlocking script dùng để thỏa mãn tập điều kiện chi tiêu trong UTXO. Ulocking script là một chữ ký điện tử (có thể không có) và khóa công khai chứng minh quyền sở hữu bitcoin. Phần thứ 4 là số thứ tự (sequence number).

Khi viết phần mềm bitcoin, bất cứ lúc nào bạn giải mã một giao dịch với mục đích xác thực nó hoặc đếm các khoản phí hoặc kiểm tra tập lệnh mở khóa, code của bạn trước tiên sẽ phải truy xuất UTXO được tham chiếu từ blockchain để xây dựng ngữ cảnh (context) được ngụ ý nhưng không có trong UTXO tham chiếu của các đầu vào.

Transaction serialization—inputs

Graphical user interface, text

Description automatically generated with medium confidence

Text

Description automatically generated

Hầu hết các giao dịch đều bao gồm phí giao dịch, phí được trả cho bitcoin miners cho việc bảo mật network, khoản phí này đóng vai trò như 1 cơ chế tự bảo vệ vì nó không khả thi về mặt kinh tế đối với kẻ tất công khi tấn công theo kiểu tràn ngập (flood) network với những giao dịch. Phí khuyến khích miner tiếp tục mining cũng như hạn chế việc lạm dụng của hệ thống (chủ hệ thống áp đặt một khoản phí lên mỗi giao dịch). Phí được tính bởi kích thước giao dịch trong hàng kilobytes, giao dịch có phí hợp lý, cao sẽ được ưu tiên thực hiện trước và đưa vào block tiếp theo, còn những giao dịch có phí chưa hợp lý hoặc không có sẽ bị trì hoãn và được thực hiện sau một vài khối hặc có thể không thực hiện. Phí không hoàn toàn bắt buộc, những giao dịch không phí có thể thực hiện sau cùng. Trong giao dịch không chứa khu vực lưu trữ phí, thay vào đó phí được tính bằng độ chênh lệch giữa đầu vào và đầu ra.

Fees = Sum(Inputs) – Sum(Outputs)

Chart

Description automatically generated

* 1. Transaction Scripts and Script Language

Stateless Verifcation: Ngôn ngữ tập lệnh giao dịch bitcoin (bitcoin transaction script language) là không trạng thái, tức là không có trạng thái trước thực hiện tập lệnh cũng như không có trạng thái nào được lưu sau khi thực hiện. Thông tin cần thiết để thực hiện tập lệnh có trong tập lệnh.

Script Construction (Lock + Unlock): một locking script đưa ra những điều kiện để sử dụng đầu ra trong tương lai (scriptPubKey) còn unlocking script là một script có nhiệm vụ giải quyết, làm thỏa mãn những điều kiện được đặt ra trong 1 đầu ra bởi locking script cũng như cho phép đầu ra (output) được sử dụng (scriptSig).

Mỗi một đầu vào đều chứa một unlocking script và đề cập đén một UTXO. Phần mềm xác nhận sẽ phải sao chép unlocking script, lấy UTXO liên quan và copy locking script từ UTXO này. Input sẽ được xác nhận nếu unlocking script thỏa mãn locking script.

A picture containing table

Description automatically generated

The script execution stack: Bitcoin’s scripting language sử dụng cấu trúc dữ liệu là stack. Tập lệnh sẽ được triển khai bằng cách thực hiện từng vị trí từ trái qua ơhair. Những con số được đẩy vào stack. Bộ điều khiển sẽ đẩy (push) hoặc lấy ra (pop) một hay nhiều tham số từ stack, thực hiện lên chúng và đẩy kêt quả vào stack.

Graphical user interface, text, application

Description automatically generated

Separate execution of unlocking and locking scripts: unlocking script sẽ thực hiện trước cho vào stack tính toán. Nếu không có sai sót gì xảy ra và có kết quả trả về thì stack này sẽ sao chép locking script vào để tiếp tục thực hiện, nếu kết quả trả về thành công thì unlocking thỏa mã locking.

Pay-to-Public-Key-Hash (P2PKH): Đa số giao dịch được thực hiện trên bitcoin network sử dụng outputs được khóa bởi Pay-to-Public-Key-Hash or “P2PKH” script. Những đầu ra (outputs) này chứa một tập các lệnh dùng để kháo output với một mã băm public key (bitcoin address). Đầu ra được mở khóa bởi public key và chữ ký điện tử tạo bởi private key tương ứng.

Text

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

* 1. Digital Signatures (ECDSA)

Thuật toán chữ ký điện tử được sử dụng trọng bicoin là Elliptic Curve Digital Signature Algorithm, or ECDSA.

Chữ ký điện tử chứng minh sự sở hữu với private key. Chữ ký điện tử thể hiện bằng chứng ủy quyền (authorization) là không thể phủ nhận – undeniable (không thể chối bỏ - nonrepudiation). Chữ ký chứng minh rằng giao dịch (hoặc một phần đặc biệt của giao dịch) không thể thay đổi bởi bất kỳ một ai sau khi nó được ký.

Creating a digital signature:

Text

Description automatically generated

Hàm Fsig tạo ra chữ ký Sig bao gồm 2 giá trị là R và S ( Sig = (R, S) ). Hai giá trị này được tính toán và tuần tự hóa (serialize) thành byte stream dưới tiêu chuẩn mã hóa lược đồ Distinguished Encoding Rules, or DER.

Serialization of signatures (DER): format tuần tự hóa bao gồm 9 thành phần sau được trích từ ví dụ



• 0x30—indicating the start of a DER sequence

• 0x45—the length of the sequence (69 bytes)

• 0x02—an integer value follows

• 0x21—the length of the integer (33 bytes)

• R— 00884d142d86652a3f47ba4746ec719bbfbd040a570b1deccbb6498c75c4ae24cb

• 0x02—another integer follows

• 0x20—the length of the integer (32 bytes)

• S—4b9f039ff08df09cbe9f6addac960298cad530a863ea8f53982c09db8f6e3813

• A suffix (0x01) indicating the type of hash used (SIGHASH\_ALL)

Verifying the Signature: Thuật toán xác minh chữ ký nhận thông báo (mã băm của giao dịch hoặc các phần của nó), khóa công khai của người ký và chữ ký (giá trị R và S) và trả về TRUE nếu chữ ký hợp lệ cho thông báo và khóa công khai này.

Signature Hash Types (SIGHASH): Mỗi input có thể chứa một chữ ký trong tập lệnh mở khóa của nó. Vì vậy, một giao dịch chứa nhiều inputs có thể có nhiều chữ ký với những cờ SIGHASH khác nhau nhằm cam kết (commit) các phần khác nhau của giao dịch trong mỗi input. Ngoài ra inputs có thể đến từ những người sở hữu khác nhau, mỗi người ký một input và tập hợp đầy đủ những chữ ký này mới xác nhận được giao dịch. Nhiều loại cờ SIGHAG có tác dụng lớn nếu bạn nghĩ đến việc nhiều người tham gia cùng cộng tác bên ngoài mạng bitcoin và cập nhật giao dịch đã ký từng phần (a partially signed transaction).

Graphical user interface, text

Description automatically generated

Text

Description automatically generated

ECDSA Math: thuật toán chữ ký (signature algoritm) đầu tiên sẽ tạo ra một ephemeral (tam thời) private public key pair. Cặp khóa này sẽ được dùng để tính hai giá trị R và S, sau một quá quá trình chuyển đổi gồm với signing private key và transaction hash. Cặp khóa tạm thời cơ bản dựa trên một số ngẫu nhiên k (temporary private key). Từ k, sẽ tính ra temporary public key P = k\*G. Gía trị R là tọa độ x của P. Còn S được tính như sau:

Text, letter

Description automatically generated

Qúa trình xác nhận được tính toàn ngược lại như sau:

Text, letter

Description automatically generated

Nếu tọa độ x của P mà bằng R thì xác nhận signature. Chú ý rằng trong quá trình xác nhận chữ ký thì private key không được biết cũng như tiết lộ.

* 1. Bitcoin Addresses, Balances, and Other Abstractions

Giao dịch không chứa bitcoin addresses, mà thay vào đó là việc thực hiện các tập lệnh (script) dùng để khóa và mở khóa giá trị rời rạc của bitcoin. Số dư (Balance) không thể hiện ở bất kỳ đâu trong hệ thống nhưng ở bất kỳ ứng dụn ví nào cũng trình chiếu số dư đó cho người dùng.

Từ tập UTXO, blockchain explorer công tổng các giá trị của unspent output thuộc về key của người sở hữu qua đó đưa ra số dư cuối cùng ‘final balance’ cho người dùng.

Như vậy, mọi thông tin mà người dùng thấy được trên ứng dụng ví, blockchain explorers, và những giao diện bitcoin khác đều được tổng hợp bởi mức trừu tượng cao, thứ được tạo bở quá trình tìm kiếm những giao dịch khác nhau, phân tích nội dung của chúng và tổng hợp dữ liệu có trong chúng.

1. Advanced Transactions and Scripting
   1. Multisignature

Multisignature script thiết lập điều kiện nơi mà N khóa công khai (public keys) được ghi lại trong tập lệnh và có ít nhất M trong số N đó phải cung cấp chữ ký để mở khóa (unlock) funds (M-of-N scheme).



Ví dụ:

Text, letter

Description automatically generated

Có một lỗi trong quá trình thực hiện của CHECKMULTISIG, thay vì nó lấy M+N+2 thành phần trong stack thì nó sẽ lấp thêm 1 giá trị khác dự kiến dẫn đến stack thông báo lỗi và tập lệnh thực hiện sai. Vì vậy stack sẽ thêm 1 thành phần nữa thường là 0 vào cuối stack trước khi cho các thành phần tập lệnh vào.

Graphical user interface, text

Description automatically generated

* 1. Pay-to-Script-Hash (P2SH)

Multisignature script rất khó để sử dụng. Khi một giao dịch muốn thực hiện thì người đứng đầu phải xin liên hệ với những khách hàng đã ký lên giao dịch trước đó. Mỗi khách hàng sẽ có những phần mềm ví bitcoin đặc biệt với khả năng tạo các giao dịch của riêng họ, mỗi người phải hiểu rõ cách thức tạo giao dịch sử dụng tập lệnh cá nhân. Điều này dẫn đến giao dịch lớn gấp 5 lần so với giao dịch thông thường.

P2SH giải quyết vấn đề trên bằng việc thay thế complex locking script bằng một dấu vân tay điện tử (digital fingerprint), một mã hóa băm.Và giao dịch muốn thực hiện thì tập lệnh (script) của nó phải khớp (match) với mã băm này. Tập lệnh khóa (locking script) được thay thế bằng băm được gọi là tập lệnh đổi thưởng (redeem scripts) vì nó được hiển thị cho hệ thống tại thời điểm đổi thưởng (redemtion) thay vì dưới dạng tập lệnh khóa.

Text, letter

Description automatically generated

Tập lệnh phức tạp (complex script) đưa ra chi tiết điều kiện để sử dụng output (redeem script) không được thể hiện trong locking script. Thay vào đó là mã hóa băm trong locking script và redeem script như một phần của unlocking script khi sử dụng đầu ra.

Ví dụ: với một complex script không có P2SH thì ta có locking script

A picture containing text

Description automatically generated

Hay

Text

Description automatically generated

Sau khi áp dụng P2SH mã hóa băm 20 byte đầu của script trên được kết quả



Locking script sẽ là



Như vậy thay vì phải sử dụng 5 key thì giờ đây với P2SH chúng ta chỉ cần sử dụng mã băm tập lệnh của các key đó điều này rút ngắn locking script đi rất nhiều trong quá trình giao dịch. Khi người chủ sở hữu cũng như đối tác muốn sử dụng UTXO này thì họ phải đưa ra redeem script gốc ban đầu (tập lệnh được băm trở thành locking script) và chữ ký cần thiết để mở khóa.

A screenshot of a computer

Description automatically generated with medium confidence

P2SH addresses: P2SH có khả năng mã hóa một lệnh băm trở thành một address quy định trong BIP-13. Địa chỉ P2SH là mã hóa Base58Check của mã băm 20 byte của một tập lệnh, giống như địa chỉ bitcoin là mã hóa Base58Check của mã băm 20 byte của khóa công khai. P2SH address sử dụng version là prefix ‘5’, sau khi được Base58Check-encoded thì địa chỉ bắt đàu bởi số ‘3’.

Lợi ích của P2SH:

Text, letter

Description automatically generated

Redeem Script and Validation: redeem script được biểu diễn trong sử dụng giao dịch chỉ có thể là 1 trong những loại cơ bản sau: P2PK, P2PKH, or multisig nature, excluding RETURN and P2SH chính nó. Chú ý rằng không thể đặt một P2SH vào trong một P2SH redeem script vì đặc tả P2SH không thể đệ quy. Ngoài ra, bởi vì redeem script không được thể hiện trên network cho đến khi bạn sử dụng một P2SH output, nếu bạn muốn khóa một output với mã băm của một redeem script chưa xác nhận thì hệ thống vẫn thực hiện, UTXO vẫn được khóa thành cộng nhưng bạn không thể sử dụng nó.

* 1. Data Recording Output (RETURN)

Nhiều nhà phát triển cố gắng sử dụng transaction scripting language để tận dụng lợi thế về bảo mật và khả năng phục hồi của hệ thống ứng dụng nhằm hướng tới những mục đích khác nhau, một trong những mục đích là tạo ra một transaction output lưu trữ dữ liệu của blockchain.

Sử dụng blockchain của bitcoin nhằm lưu những dữ liệu không liên quan đến bitcoin payments là một chủ đề gây tranh cãi. Một số người phản đối vì lưu những dữ liệu không liên quan đến những khoản chi tiêu (nonpayment data) dẫn đến ‘blockchain bloat’, gây ra gánh nặng khi những running full bitcoin nodes phải tốn chi phí lưu trữ dữ liệu mà blockchain không có ý định mang theo. Những address của dữ liệu này sẽ không tương thích với một private key nào dẫn đến những UTXO không bao giờ được sử dụng (fake payment) gây ra bùng nổ kích thước cơ sở dữ liệu UTXO.

Việc sử dụng RETURN (cho phéo nhà phát triển thêm 80 bytes nonpayment data vào một transaction output), tuy nhiên khác với fake UTXO ở trên thì RETURN ghi đầu ra vào blockchain thay vì lưu vào tập UTXO. Điều này giúp giải quyết vấn đề bùng nổ kích thước cơ sở dữ liệu UTXO.

Text

Description automatically generated

Không có một ‘unlocking script’ nào liên quan tơi RETURN rằng có thể chi tiêu một RETURN ouput – Return là không thể chi tiêu (unspendable). Mọi bitcoin được cho vào đầu ra của Return sẽ bị mất vĩnh viễn. Nếu một RETURN được tham chiếu như một đầu vào trong một giao dịch, công cụ xác thực tập lệnh sẽ tạm dừng việc thực thi tập lệnh xác thực và đánh dấu giao dịch là không hợp lệ.

* 1. Timelocks

Timelocks (thời điểm khóa) những hạn chế đối với các giao dịch hoặc đầu ra chỉ cho phép chi tiêu sau một thời điểm. Timelocks hữu ích cho việc xác định thời hạn giao dịch và khóa tiền vào một thời điểm trong tương lai.

Transaction Locktime (nLocktime) là cài đặt cấp độ giao dịch (một trường trong cấu trúc dữ liệu giao dịch) xác định thời gian sớm nhất mà một giao dịch hợp lệ và có thể được chuyển tiếp trên mạng hoặc được thêm vào blockchain. Nếu nLocktime < 500 million thì nó chỉ độ cao của block, nghĩa là transaction chưa được xác nhận, không được chuyển tiếp hoặc đưa vào blockchain trước chiều cao khối được chỉ định. Nếu transaction được vẫn chuyển trước thời điểm nLocktime thì nó sẽ bị loại bỏ ngay khi đến node đầu tiên.

Transaction locktime limitations: hạn chế của locktime đó là người nhận giao dịch chỉ có thể sử dụng và trao đổi nó sau khi hết nLocktime, trước khoảng thời gian đó người nhận sẽ không thể biết rằng người chuyển giao dịch đó cho mình có tạo ra một giao dịch khác sử dụng những UTXO trong giao dịch đó hay không. Vì vậy timelock restriction phải được đặt cho UTXO thay vì transaction.

Check Lock Time Verify (CLTV) được cho vào tập lệnh dựa trên BIP-65 là timelock cho mỗi output thay vì mỗi transaction. CLTV không thay thế nLocktime mà nó sẽ hạn chế thời gian sử dụng cho từng UTXO một. CLTV được đưa vào redeem script trong quá trình tạo output để thực hiện.

Ví dụ ban đầu output gồm



Sau khi thêm CLTV



Khi nhận một transaction thì người nhận sẽ sử dụng chữ ký và public key để mở khóa và thấy nLocktime của transaction bằng hoặc lớn hơn CHECKLOCKTIMEVERIFY của các UTXO trong đó thì người nhận có thể chuyển tiếp transaction. Còn nếu không thì transaction sẽ là không xác nhận (invalid).

Graphical user interface, text, application

Description automatically generated

Nlocktime và CHECKLOCKTIMEVERIFY là thời gian tuyệt đối (absolut time).

Relative Timelocks: được định nghĩa là một điều kiện để sử dụng output, một khoảng thời gian đã trôi qua kể từ khi output được xác nhận trong blockchain. Relative Timelock cho phép một chuỗi gồm hai hoặc nhiều giao dịch phụ thuộc lẫn nhau được giữ lại ngoài chuỗi, trong khi thiết lập điều kiện thời gian cho 1 giao dịch phụ thuộc vào thời gian đã trôi qua kể từ khi xác nhận giao dịch trước đó. Nói cách kahcs là đồng hồ không đếm cho đến khi UTXO được ghi vào blockchain.

Relative Timelocks with nSequence:

* Vùng nSequence từng được sử dụng với mục đích cho phép giao dịch sửa đổi trong mempool. Cụ thể nếu nSequence dưới 2^32 (0xFFFFFFFF) thì giao dịch đó chưa phải cuối cùng, ngược lại thì giao dịch được coi là hoàn thiện và được khai thác (mined).
* nSequence as a consensus-enforced relative timelock: nếu như nSequence < 2^31 (bit 1<<31 chưa được thiết lập) thì transaction ở trạng thái ‘relative locktime’. Ngược lại (bit 1<<31 được thiết lập) thì nSequence sẽ được sử dụng với những mục đích khác như: hỗ trợ CHECKLOCKTIMEVERIFY, nLocktime, Opt-In-Replace- By-Fee, and other future developments.
* Khi nSequence chỉ định relative locktime thì nếu một giao dịch với một đầu vào relative locktime nSequence gồm 30 khối, giao dịch chỉ hợp lệ khi ít nhất 30 khối đã trôi qua từ thời gian UTXO được tham chiếu trong đầu vào được khai thác.
* Gía trị nSequence được định nghĩa bởi 2 loại thời gian là đếm theo số blocks, đếm theo giây. Cụ thể nếu type-flag được thiết lập trong khoảng dưới bit thứ 23 (ví dụ giá trị 1<<22) thì nSequence có giá trị là cấp số cộng của 512 giây, ngược lại tì giá trị này là số block.
* Khi số nSequence chỉ định là số relative locktime, chỉ 16 bit quan trọng nhất mới được xem xét. Khi các cờ (bit 32 và 23) được đánh giá, giá trị nSequence thường được "che" bằng mặt nạ 16 bit (masked with a 16 bit mask).

Timeline

Description automatically generated

Relative Timelocks with CSV:

* CHECKSEQUENCEVERIFY (CSV): giới hạn sử dụng UTXO cho đến khi một lượng blocks hoặc giây (seconds) trôi qua tương ứng với thời điểm UTXO được khai thác (mined). Gía trị CSV phải phù hợp tương thích với giá trị nSequence.
* Median-Time-Past: tron bitcoin có sự khác nhau giữa wall time và consensus time (thời gian đồng bộ). Độ trễ của network phải tính vào mỗi node và cuối cùng được đồng bộ hóa để tạo một sổ cái chung (ledger). Timestampes được thiết lập trong block heads bởi miners dẫn đến việc miners có thể làm giả thời gian trong block nhằm thu thêm phí từ những transactions thêm vào. Giair quyết vấn đề này thì một tiêu chuẩn khác ở đây là BIP-113 được thêm nào nhằm đưa ra một phương pháp tính đồng bộ thời gian mới là Median-Time-Past. Phương pháp này được tính bằng cách lấy timestamps của 11 blocks mới nhất và đưa ta thời gian trung bình. Thời gian này sẽ trở thành consensus time và được sử dụng cho tất cả phép tính timelock. Consensus time tính bở phương pháp này sẽ luôn chậm hơn 1 giờ so với wall clock time.
* Timelock Defense Against Fee Sniping: Fee-sniping là một kịch bản tấn công mà miners sẽ cố gắng sử dụng lại các khối trong quá khứ, lấy những giao dịch có chi phí cao từ các khối tương lại đưa lại vào khối quá khứ đó để làm tang lợi nhuận cho họ. Để ngăn chặn điều này, khi tạo một giao dịch chúng ta sẽ sử dụng một nLocktime nhằm giới hạn giao dịch chỉ được thêm vào block thứ nLocktime.
  1. Scripts with Flow Control (Conditional Clauses) (xem sách).

1. The Bitcoin Network
   1. Peer-to-Peer Network Architecture

Node Types and Roles: một bitcoin node là một tổ hợp của các hàm: routing, the blockchain database, mining và wallet service.

A picture containing diagram

Description automatically generated

Tất cả các nodes đều có hàm định tuyến (routing) để tham gia vào mạng và có thể bao gồm một số thành phần khác. Tất cả các nodes xác nhận và phân phối giao dịch, blocks, khám phá và duy trì kết nối ngang hàng.

Một số node được gọi là full node có thể duy trì sự đầy đủ, cập nhật mới nhất bản sao của blockchain, chúng tự động và có thẩm quyền xác nhận mọi giao dịch mà không cần thêm bất cứ thứ gì liên quan. Một số node chỉ duy trì một phần của blockchain và xác nhận giao dịch bằng phương thức simplifed payment verifcation, or SPV, đây là SPV nodes hay lightweight nodes.

Các Mining nodes sẽ cạnh tranh để tìm những blocks mới bằng việc chạy phần cứng chuyên dụng để tính toán giải quyết the Proof-of-Work algorithm.

User wallet thường là full node với trường hợp desktop bitcoin clients, là SPV nodes trong trường hợp thiết bị hạn chế dung lượng như smartphone,…

* 1. The Extended Bitcoin Network

Extended bitoin network bao gồm network chạy trên giao thức bitcoin P2P và những nodes chạy trên những giao thức đặc biệt khác. Được gắn với mạng chính bitcoin P2P là một lượng lớn pool servers và protocol gateways kết nối tất cả các nodes chạy trên giao thức khác. Những nodes này thường chủ yếu là pool mining nodes và lightweight wallet clients những nodes không thể chứa toàn bộ bản sao của blockchain.

Graphical user interface, text, application, chat or text message

Description automatically generated

Diagram

Description automatically generated

* 1. Bitcoin Relay Networks

Một Bitcoin relay networks là một mạng có nhiệm vụ thu hẹp nhỏ nhất độ trễ trong quá trình truyền tải blocks giữa những miners. Bitcoin Relay Network đầu tiên xây dựng vào năm 2015 cho phép đồng bộ hóa nhanh chóng các block giữa các miner với độ trễ rất thấp. Sau đó nó được thay thế bởi Fast Internet Bitcoin Relay Engine or FIBRE là một mạng chuyển tiếp dựa trên UDP chuyển tiếp các khối trong mạng các nút (network of nodes). FIBER thực hiện tối ưu hóa khối nhỏ gọn (compact block) để giảm lượng dữ liệu được truyền và độ trễ mạng. một mạng khác là Falcon. Falcon sử dụng ‘cut-through-routing’ thay vì ‘store-and-forward’ để giảm độ trễ bằng việc phân phát ngay các phần của các blocks mỗi khi chúng nhận được thay vì phân phát blocks khi nhận được toàn bộ phần của block.

* 1. Network Discovery

Khi một node được tiết lập, nó phải tìm kiếm những bitcoin nodes khác trong mạng để yêu cầu tham gia. Đầy tiên node mới phải tìm được tối thiểu một node trong mạng và kết nối đến nó (có thể tìm kiếm bằng chọn ngẫu nhiên). Nodes sẽ thiết lập một kết nối TCP tạo cổng 8333 và thực hiện bắt tay ‘handshake’ bằng cách truyền một vesion message, chứa những thông tin xác định cơ bản.

Text, application, email

Description automatically generated

Sau khi nhận được version message từ node mới thì node nhận được (local peer) sẽ kiểm tra nVersion của node mới và xem xét nó có tương thích hay không. Nếu tương thích thì local peer sẽ công nhận version message và thiết lập kết nối bằng việc gửi lại verack.

Node mới tìm các peers thông qua truy vấn DNS bằng một số ‘DNS seeds’. DNS seeds sẽ cung cấp một danh sách IP address của những stable bitcoin listening nodes. Một số seeds được tạo bở sử dụng BIND (Berkeley Internet Name Daemon) - chọn ngẫu nhiên một tập con từ danh sách địa chỉ thu thập bởi crawler hoặc long-running bitcoin node.

Diagram

Description automatically generated

Sau khi kết nối thì node mới sẽ gửi addr message chứa địa chỉ IP của nó cho những nodes lân cận (đã kết nối với node mới). Những nodes này sẽ tiếp tục chuyển tiếp message này tói những hàng xóm của chúng, đảm bảo rằng node mới kết nối tốt hơn và được biết rộng rãi. Node mới có thể gửi getaddr cho hàng xóm để lấy danh sách địa chỉ IP của những nodes khác.

Diagram

Description automatically generated

Một node cần kết nối đến nhiều node khác nhau để làm đa dạng đường truyền. Những đường chuyền là không đáng tin cậy vì một node có thể đến và đi vì vậy một node cần phải tiếp tục tìm những nodes mới khi mất những kết nối cũ cũng như hỗ trợ các node khác khi chúng khởi động. Sau khi khởi động, một node sẽ ghi nhớ các kết nối ngang hàng thành công gần đây nhất của nó, để nếu nó được khởi động lại, nó có thể nhanh chóng thiết lập lại các kết nối với mạng ngang hàng cũ của nó. Nếu không ai trong số các đồng nghiệp cũ phản hồi yêu cầu kết nối, nút có thể sử dụng các seed nodes để khởi động lại.

Nếu không có lưu lượng truy cập trên một kết nối, các nodes định kỳ sẽ gửi một thông báo để duy trì kết nối. Nếu một node không giao tiếp trên một kết nối trong hơn 90 phút, nó được coi là bị ngắt kết nối và một new peer sẽ được tìm kiếm.

* 1. Full Nodes

Full Nodes là gì?(đã nói ở trên).

Full blockchain node phụ thuộc vào network để nhận những nâng cấp về các blocks, giao dịch mới.

Chạy full blockchain node cho bạn trải nghiệm bitcoin thuần túy: độc lập xác nhận tất cả các giao dịch mà không cần dựa vào, tin tưởng bất kỳ một hệ thống. Việc chạy full node tiêu tốn một không gian lưu trữ cố định (disk space) vô cùng lớn lên đến hàng trăm gigabyte, và cần 2 đến 3 ngày để đồng bộ với mạng.

* 1. Exchanging “Inventory”

Việc đầu tiên mà full node sau khi kết nối với các peers là cấu trúc lên một blockchain hoàn chỉnh. Nếu là một full node mới hoàn toàn thì nó chỉ biết 1 block là genesis block (block #0).

Qúa trình đồng bộ hóa với blockchain bắt đầu với version message (chứa chiều cao blockchain mà node đang có). Với message này các node có thể biết được bao nhiêu blocks mà mỗi nodes có và so sánh với số blocks của mình. Các nodes được kết nối đồng đẳng (Peered) sẽ gửi getblock message chứa mã băm (fingersprint) của top block trong blockchain cục bộ của nó tới các nodes khác. Nhờ mã băm này thì một trong số các peers sẽ có thể biết được top block của node đó không phải top block của mình mà là một block cũ hơn trong chuỗi khối mình có suy ra mình chứa blockchain cục bộ dài hơn so với node gửi message. Node chứa blockchain dài hơn sẽ xác định những block mà node kia chưa có (khoảng 500 blocks đầu tiên) rồi gửi mã băm của các blocks qua inv (inventory) messeage cho node kia. Node thiếu blocks thông qua getdata message yêu cầu toàn bộ dữ liệu của block và xác định những block được yêu cầu sử dụng mã băm từ inv message.

Diagram

Description automatically generated

* 1. Simplifed Payment Verifcation (SPV) Nodes

SPV nodes chỉ tải về block headers và không tải toàn bộ giao dịch trong mỗi block. Vì vậy SPV nodes không thể cấu trúc lên đầy đủ thông tin về tất cả UTXO có thể được sử dụng, việc xác nhận giao dịch phải dựa vào các peers cung cấp từng phần thông tin liên quan đến các phần blockchain theo nhu cầu.

SPV xác nhận các giao dịch theo độ sâu (depth) của chúng trong blockchain thay vì độ cao (height) của nó. Cụ thể SPV sẽ xác nhận chuỗi của tất cả các block (chain of all blocks) và liên kết nó tới giao dịch được quan tâm.

Ví dụ xét một giao dịch trong block thứ 300000, với full node sẽ liên kết tất cả 300000 blocks đến tận genesis block và xây dựng một cơ sở dữ liệu UTXO, thiết lập xác nhận cho giao dịch bằng việc xác nhận UTXO đó chưa được dùng. Nhưng SPV lại khác, nó sẽ thiết lập liên kết giữa giao dịch với block chứa nó bằng merkle path. Rồi nó chờ khoảng 6 block được thiết lập từ 300001 đến 300006 đặt lên trên đầu của blockchain. Như vậy 6 block này đã được xác nhận và tất cả các node trong mạng đều chấp nhận block thứ 300000 qua đó gián tiếp xác nhận giao dịch trong block đó.

Một SPV có thể dễ dàng chứng minh một giao dịch tồn tại (thông qua merkle path proof và the Proof-of-Work trong chuỗi các blocks). Tuy nhiên nó không thể xác nhận giao dịch (có khả năng chi tiêu gấp đôi trên cùng một UTXO) vì nó không chứa toàn bộ bản sao của các giao dịch dẫn đến denial-of-service attack hoặc double-spending attack cho SPV nodes. Để chống lại các cuộc tấn công này thì SPV nodes cần kết nối ngẫu nhiên đến nhiều nodes với hy vọng có một node là trung thực. Việc kết nối ngẫu nhiên nhằm giảm thiểu xác xuất kết nối đến fake node hoặc fake network.

Để lấy block headers thì SPV nodes sử dụng một getheaders message, những node trả lơi sẽ gửi lên đến 2000 block heads thông qua sử dụng headers message. SPV node thiết lập bộ lọc trong các kết nối đến các peers nhằm lọc dòng những khối, giao dịch tương lại được các peers trả về. Những giao dịch yêu thích sẽ được yêu cầu lấy thông qua getdata message và peer sẽ tạo tx message chứa những giao dịch đó gửi lại cho SPV nodes.

Việc chỉ lấy những giao dịch cụ thể dấn đến một lỗ hổng bảo mật, việc SPV node yêu cầu dữ liệu đặc biệt có thể làm lộ addresses trong ví của họ.

Diagram

Description automatically generated

* 1. Bloom filter

Bloom filter là một bộ lọc cho phép SPV node truy xuất đến những giao dịch mong muốn mà không làm lộ địa chỉ cụ thể cũng như key hoặc giao dịch mà node đang tìm kiếm. Ví dụ mường tượng bloom filter như một câu hỏi đường ‘có con đường nào xung quanh đây có tên bắt đầu bằng chữ ‘h-g-m’?’.

How Bloom Filters Work?

Bloom filter cung cấp một mảng gồm N chữ số nhị phân và một biến M là số hàm băm được sử dụng. Kết quả mỗi hàm băm sẽ trả về giá trị từ 1 đến N liên quan đến mảng nhị phân. Điều chỉnh bloom filter bằng cách điều chỉnh M và N, mới một giá trị không đổi thì kết quả qua M hàm băm sẽ không thay đổi và tồn tại duy nhất một kết quả.

Graphical user interface, text, application

Description automatically generated

Đầu vào là những patterns (mẫu) được đưa qua M hàm băm, băm được M kết quả (1 đến N) mỗi vị trí trong mảng nhị phân N tương ứng với kết quả của M hàm băm sẽ được chuyển thành 1. Cứ thế làm vậy cho đến hết mẫu.

Diagram

Description automatically generated

Diagram

Description automatically generated

Khi muốn xác định một giá trị nào đó có thuộc vào tập mẫu (pattern) ban đầu không chúng ta chỉ cần cho giá trị đó qua M hàm băm thu được M giá trị, các vị trí trên mảng N tương ứng với M giá trị đó đều có giá trị là 1 thì giá trị trên thuộc mẫu, ngược lại thì không thuộc mẫu.

Diagram

Description automatically generated

Diagram

Description automatically generated

SPV nodes sử dụng bloom filter bằng cách khởi tạo một bloom filter rỗng sau đó khởi tạo một danh sách các addresses, keys và các mã băm mà nodes chú ý đến. Tiếp theo nodes sẽ đưa mỗi thông tin này như những patterns vào bloom filter. SPV node sau đó sẽ gửi filterload message chứa bloom filter đến các peers. Các peers sử dụng bloom filter để lấy những giao dịch mong muốn rồi gửi lại cho SPV.

* 1. SPV Nodes and Privacy (xem sách)
  2. Transaction Pools

Memory pool, mem pool hay transaction pool là nơi mà bitcoin network duy trì danh sách tạm thời những giao dịch chưa xác nhận. Nodes sử dụng những hồ chứa này để theo dõi những giao dịch đã công bố trên mạng nhưng chưa được thêm vào blockchain.

Một số node tạo ra orphan pool chứa những giao dịch mà đầu vào input lấy từ một giao dịch nào đó chưa biết. Khi một giao dịch thêm vào transaction pool, node sẽ kiểm tra giao dịch đó có liên quan đến một đàu vào của giao dịch nào trong orphan pool ko, nếu có thì loại giao dịch đó khỏi orphan pool và chuyển đến transaction pool.

UTXO pool là nơi chứa hàng triệu unspent transaction output. Nếu như transaction pool và orphan pool chưa những giao dịch chưa xác nhận thì UTXO pool chứa những outputs được xác nhận.

# Chương 9, 10 (đọc sách hiểu rõ hơn).