Để xác định một người dùng có nhiều địa chỉ, cần xác định người dùng đấy nắm tất cả các khóa riêng của các địa chỉ đó.

Để làm được như vậy, ta cần chứng minh người dùng có thể ký cùng một thông điệp để tạo nên các bản ký.

Và nếu phía contract xác thực là các bản chữ ký ứng với thông điệp chung kia là hợp lệ thì có thể xác định là người dùng nắm giữ các khóa riêng.

Câu hỏi

Hệ thống có gì để người dùng tin tưởng gửi tiết kiệm?

Người dùng cần cam kết gì để có thể vay tiền từ hệ thống? Dựa vào số tiền đang có trong tài khoản đủ đáp ứng điều kiện vay ???

Xử lý rủi ro của hệ thống do lập trình khắc phục kiểu gì? Bảo trì, tổ chức các cuộc thi hackathon có thưởng tìm ra lỗ hổng.

CryptoKitties trở nên quá phổ biến làm cho blockchain Ethereum bị bế tắc. Nên BSC có khả năng mở rộng quy mô.

Đường dẫn

Trava:

ToVChain is the world’s first hyper-linked datachain protocol that links both data and functionalities across distributed ledgers to drastically increase the adoption of blockchain technologies in real problems. While other projects and researches deal with simple data interoperability among multiple blockchain networks, we define the new term “semantic interoperability” and offer a unique solution for comprehensive and profound interoperability in both data and functionalities. Three distinctive features that differentiate our approach from the others are as follows: (i) the establishment of semantic links among the data elements in blockchain networks in a trustable, verifiable, and immutable manner, (ii) the automatic migration of a whole DApp including all of its components and data from a blockchain network to another network, and (iii) the automatic code generation of DApps that have components (e.g., smart contracts, processors) running on one or multiple blockchains to perform particular business processes.

Researchers at MIT in the research Toward an Interoperability Architecture for Blockchain Autonomous Systems defined blockchain interoperability as “a composition of distinguishable blockchain systems, each representing a unique distributed data ledger, where atomic transaction execution may span multiple heterogeneous blockchain systems, and where data recorded in one blockchain are reachable, verifiable, and referable by another possibly foreign transaction in a semantically compatible manner”. The definition focuses on the data level only.

To deal with blockchain data interoperability, sidechain, abstraction layer, and hub/adaptor have been three widely used techniques. Sidechain is the small blockchain network, which runs parallel with the main blockchain to synchronize data and increase users’ performance. Abstraction layer (message protocols) provides an abstraction layer for gathering and delivering data among different blockchain. Hub/adaptor is the coupling means for blockchains that enable transferring data from one to another.

These three techniques and others focus on data interoperability and develop their own data exchange protocols to create a vision of the Internet of blockchains. However, no projects or techniques have ever considered the semantics of data interoperability as mentioned in MIT's definition. Moreover, interoperability at the data level could not make the dream of the Internet of blockchains come true.

Security Leaderboard: https://www.certik.org/projects/trava

Deploy: https://www.binance.org/en/smartChain

Explorer:

https://etherscan.io/

<https://bscscan.com/>

https://api.bscscan.com/api?module=account&action=balance&address=0x70F657164e5b75689b64B7fd1fA275F334f28e18&apikey=FGKHMRH65UDHMRXY9DH5RNY4CBY9D1ZD2Z

<https://infura.io/dashboard>

<https://www.uport.me/>

<https://web3js.readthedocs.io/>

<https://learnmeabitcoin.com/beginners/getting-started>

<https://viblo.asia/p/ecdsa-he-mat-dua-tren-duong-cong-elliptic-va-ung-dung-trong-blockchain-XL6lA4oDZek>

https://besu.hyperledger.org/en/stable/Concepts/NetworkID-And-ChainID/

Integrated Encryption Scheme

- Pool Creation

- Data Analysis

- Cross-chain Lending

Thuật ngữ

Blockchain: chuỗi khối

Private key: mật mã bí mật

Public key: mật mã công khai

dApps: ???

???: Unknown

XXX: Link to write

Merkle Tree: cây Merkle (Ralph Merkle)

DApp: Decentralized Application

Công nghệ Blockchain tương đồng với có sở dữ liệu, chỉ khác ở việc tương tác với cơ sở dữ liệu. Để hiểu blockchain, cần nắm được năm định nghĩa sau: chuỗi khối (Blockchain), cơ chế đồng thuận phân tán đồng đẳng (Distributed), tính toán tin cậy (Trusted computing), hợp đồng thông minh (Smart contracts) và bằng chứng công việc (Proof of work). Mô hình tính toán này là nền tảng của việc tạo ra các ứng dụng phân tán.

- Cơ chế đồng thuận phân tán đồng đẳng (Hay còn gọi là cơ chế đồng thuận phân quyền)

- Chuỗi khối và dịch vụ chuỗi khối

Abilities:

- Proofing work

- Mining new blocks

- Creating transactions

- Validating the chain

- Retrieving address data and other functionalities

Blockchain là công nghệ chuỗi - khối, cho phép truyền tải dữ liệu một cách an toàn dựa trên hệ thống mã hóa phức tạp, tương tự như cuốn sổ cái kế toán của một công ty, nơi mà tiền được giám sát chặt chẽ và ghi nhận mọi giao dịch trên mạng ngang hàng.

Mỗi khối (block) đều chứa thông tin về thời gian khởi tạo và được liên kết với khối trước đó, kèm theo đó là một mã thời gian và dữ liệu giao dịch. Dữ liệu khi đã được mạng lưới chấp nhận thì sẽ không có cách nào thay đổi được.

Công nghệ Blockchain là sự kết hợp của 3 loại công nghệ:

- Mật mã học: Để đảm bảo tính minh bạch, toàn vẹn và riêng tư thì công nghệ Blockchain sử dụng public key và hàm hash function.

- Mạng ngang hàng: Mỗi nút trong mạng được xem như là một client và cũng là server để lưu trữ bản sao ứng dụng.

- Lý thuyết trò chơi: Tất cả các nút tham gia vào hệ thống đều phải tuân thủ luật chơi đồng thuận (giao thức PoW, PoS, ...) và được thúc đẩy bởi động lực kinh tế.

Hệ thống Blockchain chia thành 3 loại chính:

- Public: Bất kỳ ai cũng có quyền đọc và ghi dữ liệu trên Blockchain. Quá trình xác thực giao dịch trên Blockchain này đòi hỏi phải có nhiều nút tham gia. Vì vậy, muốn tấn công được vào hệ thống Blockchain này cần chi phí rất lớn và thực sự không khả thi. Ví dụ: Bitcoin, Ethereum,...

- Private: Người dùng chỉ được quyền đọc dữ liệu, không có quyền ghi vì điều này thuộc về bên tổ chức thứ ba tuyệt đối tin cậy. Vì đây là một Private Blockchain, cho nên thời gian xác nhận giao dịch khá nhanh vì chỉ cần một lượng nhỏ thiết bị tham gia xác thực giao dịch. Ví dụ: Ripple là một dạng Private Blockchain, hệ thống này cho phép 20% các nút là gian dối và chỉ cần 80% còn lại hoạt động ổn định là được.

- Permissioned (hay còn gọi là Consortium): Một dạng của Private nhưng bổ sung thêm một số tính năng khác, đây là sự kết hợp giữa Public và Private. Ví dụ: Các ngân hàng hay tổ chức tài chính liên doanh sẽ sử dụng Blockchain cho riêng mình.

Các phiên bản của công nghệ Blockchain:

- Công nghệ Blockchain 1.0 - Tiền tệ và Thanh toán. Ứng dụng chính của phiên bản này là tiền mã hóa bao gồm việc chuyển đổi tiền tệ, kiều hối và tạo lập hệ thống thanh toán kỹ thuật số. Đây cũng là lĩnh vực quen thuộc với chúng ta nhất mà đôi khi khá nhiều người lầm tưởng Bitcoin và Blockchain là một.

- Công nghệ Blockchain 2.0 - Tài chính và Thị trường. Ứng dụng xử lý tài chính và ngân hàng: mở rộng quy mô của Blockchain, đưa vào các ứng dụng tài chính và thị trường. Các tài sản bao gồm cổ phiếu, chi phiếu, nợ, quyền sở hữu và bất kỳ điều gì có liên quan đến thỏa thuận hay hợp đồng.

- Công nghệ Blockchain 3.0 - Thiết kế và Giám sát hoạt động: đưa Blockchain vượt khỏi biên giới tài chính, đi vào các lĩnh vực như giáo dục, chính phủ, y tế và nghệ thuật.

Đặc điểm nổi bật của Blockchain

- Không thể làm giả, không thể phá hủy các chuỗi Blockchain

- Bất biến

- Bảo mật

- Minh bạch

- Hợp đồng thông minh

Sổ cái: Ledger

- Viết 1 smart contract ERC20 đảm bảo những yêu cầu sau đây:

1. Khởi tạo 1 số lượng token nào đó

2. Smart contract cho phép tính năng "Burn" token

3. Smart contract cho phép "Transfer" token từ địa chỉ A -> B

4. Smart contract có tính năng "lock" token, không cho transfer

5. Smart contract định danh owner và đổi owner

Ngoài ra có thể thêm 1 số tính năng khác...

- Sau khi viết xong thì phải test dc toàn bộ các tính năng trên 1 mạng blockchain giả lập, sau đó là 1 mạng testnet

1 số khái niệm và tools cần nghiên cứu:

- Ý nghĩa của 5 hàm cơ bản trong ERC20

- 1 số mạng blockchain testnet phổ biến: ropstent, rinkeby

Tool viêt smart contract: Remix hoặc VS code + solidity extentions

Tool deploy: Truffle hoặc Remix compiler

Tool tạo mạng blockchain: Geth - local mode hoặc ganache

Tool test smart contract: YOUR CODE hoặc metamask

address(this) refers to the address of the instance of the contract where the call is being made.

msg.sender refers to the address where the contract is being called from.

ERC20 là một tiêu chuẩn kỹ thuật được sử dụng cho các hợp đồng thông minh trên Ethereum Blockchain khi phát hành Token. ERC20 là viết tắt của Ethereum Request for Comments

totalSupply

function totalSupply() public view returns (uint256): trả về tổng nguồn cung token mà hợp đồng nắm giữ

balanceOf

function balanceOf(address \_owner) public view returns (uint256 balance): trả về số dư token mà địa chỉ đó đang nắm giữ. Hãy nhớ rằng các tài khoản trên mạng Ethereum là công khai, vì vậy bạn có thể truy vấn số dư của bất kỳ người dùng nào mà bạn biết địa chỉ.

transfer

function transfer(address \_to, uint256 \_value) public returns (bool success): chuyển token từ người dùng này sang người dùng khác. Tại đây, bạn cung cấp địa chỉ muốn gửi và số tiền cần chuyển. Khi tính năng được gọi, transfer sẽ kích hoạt một thứ gọi là event.

transferFrom

function tranferFrom(address \_from, address \_to, uint255 \_value) public returns (bool success): thay thế tiện dụng hơn so với chức năng transfer. Nó cho phép lập trình nhiều hơn một chút trong các ứng dụng phi tập trung. Chuyển token, nhưng những token đó không nhất thiết phải thuộc về người gọi hợp đồng.

approve

function approve(address \_spender, uint256 \_value) public returns (bool success): là một chức năng hữu ích và có nhiều hướng để lập trình. Với chức năng này, bạn có thể giới hạn số lượng token mà một hợp đồng thông minh có thể rút từ số dư của bạn. Nếu không có nó, bạn phải đối mặt với nguy cơ hợp đồng bị trục trặc (hoặc bị lợi dụng). Sẽ kích hoạt sự kiện approval, ghi dữ liệu vào blockchain.

allowance

function allowance(address \_owner, address \_spender) public view returns (uint256 remaining): có thể sử dụng cùng với appove. Khi bạn đã cho phép hợp đồng quản lý các token của mình, bạn có thể sử dụng quyền này để kiểm tra xem nó vẫn có thể rút được bao nhiêu tiền. Ví dụ, nếu gói đăng ký của bạn đã sử dụng hết 12/20 token chấp thuận, thì việc gọi hàm allowance sẽ trả về tổng số là 8.

BSC:

Deploy on BSC: Binance Smart Chain. Blockchain chạy song song với Binance Chain. Không giống Binance Chain, BSC tự hào khi có chức năng hợp đồng thông minh và khả năng tương thích với máy ảo Ethereum (EVM). Mục tiêu của BSC là giữ nguyên thông lượng cao của Binance Chain và đồng thời có thể đưa các hợp đồng thông minh vào hệ sinh thái của Binance Chain.

Cách thức hoạt động của BSC:

- Đồng thuận: Binance Smart Chain có được thời gian khối xấp xỉ 3 giây với thuật toán đồng thuận Proof-of-Stake. Cụ thể, nó sử dụng thứ gọi là Proof of Staked Authority (PoSA). Với PoSA, người tham gia có thể đặt cược BNB để trở thành những người xác thực. Nếu họ cung cấp một khối hợp lệ, họ sẽ nhận được phí giao dịch từ các giao dịch trong đó.

- Khả năng tương thích chuỗi chéo: Có thể hình dung Binance Smart Chain như một hệ thống độc lập nhưng mang tính bổ trợ cho Binance Chain hiện có. Kiến trúc chuỗi kép đang được sử dụng nhằm giúp người dùng có thể chuyển tài sản của mình từ blockchain này sang blockchain khác một cách liền mạch. Bằng cách này, ta có thể giao dịch nhanh trên Binance Chain mà đồng thời, các ứng dụng phi tập trung cũng có thể được xây dựng trên BSC. Với khả năng tương thích như vậy, người dùng được tiếp xúc với một hệ sinh thái rộng lớn, có thể phục vụ người dùng trong nhiều trường hợp sử dụng.

- Các token BEP-2 và BEP-8 trên Binance Chain có thể được đổi lấy token BEP-20, tiêu chuẩn mới được Binance Smart Chain đưa ra. Nó sử dụng các chức năng tương tự như đối tác Ethereum của mình.

Bitoin là một tập hợp các khái niệm và công nghệ tạo thành nền tảng của hệ sinh thái tiền tệ kỹ thuật số. Đơn vị tiền tệ của Bitcoin là BTC, Satoshi.

Các khái niệm cơ bản: Bitcoin Protocol là giao thức liên lạc trong Bitcoin Network (mạng lưới), người dùng Bitcoin giao tiếp với nhau trên internet thông qua Bitcoin Protocol.

Điểm khác biệt của Bitcoin là ở chỗ không được lưu trữ hay quản lý tập trung như những thể mà ta thường thấy. Tất cả giao dịch của người dùng được ghi nhận vào các Block (khối), các Block này móc nối với nhau thành một chuỗi khối gọi là Blockchain (chuỗi khối). Block đầu tiên trong Blockchain là Genesis Block.

Blockchain giống như một cuốn sổ cái ghi lại toàn bộ nhật ký giao dịch từ buổi sơ khai cho đến hiện tại. Mỗi giao dịch được ghi nhận thành một Transaction.

Diagram

Description automatically generated

Diagram

Description automatically generated

Transaction (giao dịch) có hai thành phần chính, các input (đầu vào) và các output (đầu ra). Mỗi input là trạng thái ghi nhận quyền được spend (chi tiêu) một lượng BTC từ Wallet (các ví) khác chuyển nhượng đến. Mỗi output là trạng thái ghi nhận số lượng BTC mà người dùng đã chuyển nhượng (chuyển quyền spend) cho Wallet khác. Wallet là cái ví, nhưng điểm khác biệt là Wallet không chứa tiền (BTC) mà là chứa Key (khóa) để chứng minh quyền được spend BTC trên blockchain. Thực chất của việc chuyển tiền BTC chính là tạo ra Transaction mới, ghi nhận việc thay đổi quyền spend một lượng BTC của thực thể sở hữu Key liên quan. Key có hai loại chính là Private Key và Public Key. Public Key được tạo ra từ Private Key theo thuật toán mã hóa một chiểu. Người giữ Private Key có quyền Sign (ký) tạo ra chữ ký số, chứng minh quyền spend, và vì lý do đó họ có toàn quyền sở hữu BTC của Wallet. Public Key dùng để xác minh chữ ký số. Public Key cũng dùng để tạo ra Addres bằng cách sử dụng thuật toán mã hóa một chiều.

Address là một chuỗi ký tự đại diện cho Wallet trong việc chuyển và nhận BTC.

Khi một Transaction mới được tạo ra bởi một Wallet, nó có thể tạo offline rồi sau đó truyền tải lên Bitcoin Network khi Wallet online (giống như ta viết thư bỏ vào bao thư ở nhà, rồi sau đó mang thư đến mạng lưới chuyển phát \_ bưu điện). Transaction cần phải được Confirm (xác nhận hợp lệ) trước khi được đưa vào Block bởi các máy Mining (đào), các máy Mining có thể là máy vi tính, máy điện thoại, hoặc loại máy chuyên dụng … sử dụng tài nguyên của nó (CPU hoặc GPU) thực hiện các phép toán để xác minh các dữ liệu của Transaction. Các máy Mining tổng hợp đủ số lượng các Transaction vào một Block, sau đó nó sẽ thực hiện việc dò tìm ra một chuổi Hash thỏa mãn Difficulty Target (độ khó mục tiêu) mà mạng lưới quy định tại thời điểm đó, việc này gọi là Proof-of-work (bằng chứng làm việc). Sau khi tìm được chuỗi Hash thỏa mãn Difficulty Target, Block đó được xem là Mined (đã được đào), và đưa Block đó vào Blockchain.

Khi tạo ra Transaction, các Wallet sẽ duyệt toàn bộ Blockchain (hoặc từ vị trí Block nào đó mà nó đã ghi nhận trước các số liệu) để tổng hợp các Input và thiết lập các Output cho Transaction. Output có hai trạng thái Unspend (chưa chi tiêu) và Spent (đã chi tiêu). Đối với các Transaction đã được đưa vào Block và nằm trong Blockchain, thì các Ouput của nó sẽ có trạng thái Unspend, khi Output này được đưa vào thành Input của Transaction mới, và khi Transaction mới này được Confirm thì trạng thái của Output chuyển sang trạng thái Spent.

Tổng số BTC Input phải lớn hơn hoặc bằng tống số BTC Output. Trường hợp Input lớn hơn Output, phần chênh lệch được xem là Transaction Fee (phí giao dịch) trả cho các máy Mining và các máy Mining sẽ tự thu thập. Nếu Input lớn, trong khi chỉ thực hiện một giao dịch giá trị nhỏ thì phần thừa sẽ được tạo thành một Change Ouput coi như là trả ngược lại cho Wallet gửi. Wallet sẽ tự tính toán Fee để xác định Change Ouput, Fee càng cao thì Transaction càng được ưu tiên thực hiện trước, Fee được tính dựa trên dung lượng của Transaction theo kilobytes. Mỗi ứng dụng Wallet khác nhau có cách tính toán Fee khác nhau, nhưng mục tiêu là một con số hợp lý để Transaction được Confirm (sớm hay muộn). Nếu Fee thấp hơn 0.0001BTC thì được xem là không có Fee, và nhiều khi giao dịch sẽ không được thực hiện, nếu Wallet không tạo Change Ouput thì coi như tất cả phần BTC còn thừa là Fee và Miner (thợ mỏ) nào gặp Transaction này thì coi như gặp may.

Giá trị Fee là tất yếu phải có, và được xem là giải pháp chống Flood Transaction \_ tạo ra hàng loạt số lượng cực lớn các Transaction. Bạn có đủ số BTC để làm điều đó?

Về phần Mining, khi tạo ra được một Block, Miner (thợ mỏ) sẽ được Reward (thưởng) một số BTC. Số BTC thưởng này được ghi nhận thành một Coinbase Transaction \_ chính là Transaction đầu tiên của Block, Input của Coinbase Transaction không phải là Spent Output của Trasaction khác mà là một trường dữ liệu duy nhất gọi là Coinbase.

Cryptography Primer

Mật mã khóa công khai hay bất đối xứng, là một loại mật mã sử dụng các cặp khóa, mỗi cặp là duy nhất. Cặp khóa bao gồm khóa công khai và khóa riêng. Đây là loại tiền mã hóa mà Bitcoin sử dụng để kiểm soát tiền. Khóa công khai có thể được tạo từ khóa riêng, nhưng không thể ngược lại. Ngoài ra thứ gì đó được mã hóa bằng khóa riêng có thể được giải bằng khóa công khai và ngược lại, do đó chúng không đối xứng.

Hệ mật mã dựa trên đường cong Elliptic (ECDSA/ECC) là một giải thuật khóa công khai. Hiện nay, hệ mật RSA là giải thuật khóa công khai được sử dụng nhiều nhất, nhưng hệ mật dựa trên đường cong Elliptic (ECC) có thể thay thế cho RSA bởi mức an toàn và tốc độ xử lý cao hơn.

Ưu điểm của ECC là hệ mật mã này sử dụng khóa có độ dài nhỏ hơn so với RSA. Từ đó làm tang tốc độ xử lý một cách đáng kể, do số phép toán dung để mã hóa và giải mã ít hơn và yêu cầu các thiết bị có khả năng tính toán thấp hơn, nên giúp tăng tốc độ và giảm năng lượng cần sử dụng trong quá trình mã hóa và giải mã. Với cùng một độ dài khóa thì ECC có nhiều ưu điểm hơn so với các giải thuật khác, nên trong một vài năm tới có thể ECC sẽ là giải thuật trao đổi khóa công khai được sử dụng phổ biến nhất.

ECC thực hiện việc mã hóa và giải mã dựa trên tọa độ của các điểm trên đường cong Elliptic. Xét đẳng thức Q = kP, với Q, P là các điểm nằm trên đường cong Elliptic. Có thể khá dễ dàng tính Q nếu biết điểm k và P, nhưng rất khó xác định k nếu biết Q và P. Bên cạnh công thức của đường cong Elliptic, thì một thông số quan trọng khác của đường cong Elliptic là điểm G (còn gọi là điểm cơ sở), điểm G đối với mỗi đường cong Elliptic là cố định, trong hệ mật mã ECC thì một số nguyên lớn k đóng vai trò như một khóa riêng, trong khi đó kết quả của phép nhân giữa k với điểm G được coi như là khóa công khai tương ứng.

XXX: <https://en.wikipedia.org/wiki/Elliptic_Curve_Digital_Signature_Algorithm>

ECDSA là viết tắt của Elliptic Curve Digital Signature Algorithm – Thuật toán chữ ký điện tử dựa trên đường cong Elliptic. Đó là việc sử dụng một đường cong Elliptic và một trường hữu hạn để ký (sign) vào dữ liệu sao cho người khác có thể xác minh tính xác thực của chữ ký đó trong khi người ký vẫn giữ được độc quyền tạo ra các chữ ký. ECDSA có thủ tục riêng biệt cho việc ký kết và xác minh. Mỗi thủ tục là một thuật toán bao gồm một vài phép tính số học. Các thuật toán thực hiện việc ký sử dụng các Private Key, và quá trình xác minh sử dụng Public Key.

<https://cryptobook.nakov.com/digital-signatures/ecdsa-sign-verify-messages>

Decentralized Identifier

Giới thiệu đường cong Elliptic

Một đường cong Elliptic là một biểu diễn đại số của một phương trình có dạng

Phiên bản được sử dụng bởi Bitcoin có a = 0 và b = 7

Chart, line chart

Description automatically generated

Đường cong Elliptic có những tính chất hữu ích:

* Một đường thẳng không thẳng đứng, giao nhau tại hai điểm không tiếp tuyến của đường cong sẽ luôn giao nhau tại một điểm thứ ba trên đường cong.
* Một đường tiếp tuyến với đường cong tại một điểm và không thẳng đứng sẽ cắt nhau đúng một điểm khác trên đường cong.

Các phép toán trên đường cong Elliptic

Phép cộng điểm P + Q = R

Chart, line chart

Description automatically generated

Phép nhân đôi điểm R = 2P = P + P

Chart, line chart

Description automatically generated

Private key: bất kỳ giá trị 256 bit (32 byte). Tạo từ hạt giống (nguồn xáo trộn đủ phức tạp – gọi là entropy) như một nhóm từ hàm băm SHA256. Khóa riêng ở dạng thập lục phân. Cứ 2 chữ số đại diện cho 8 bit hoặc 1 byte.

Public key: khóa công khai được tạo từ các khóa riêng bằng secp256k1 phép nhân đường cong elliptic bằng K = k \* G, K là khóa công khai, k là khóa riêng tư và G là một hằng số.

K có tọa độ K(x, y) giá trị hex

G = 0x04 | x | y gọi là uncomporessed public key

G = 04 79BE667E F9DCBBAC 55A06295 CE870B07 029BFCDB 2DCE28D9 59F2815B 16F81798 483ADA77 26A3C465 5DA4FBFC 0E1108A8 FD17B448 A6855419 9C47D08F FB10D4B8

Nén khóa công khai thành 32 bit (compressed public key) với

0x02 nếu y > 0: 0x02 | x | y

0x03 nếu y < 0: 0x03 | x | y

Tạo địa chỉ từ việc thực hiện hash chuỗi public key (có thể là chuỗi uncompressed public key hoặc compressed public key tùy phiên bản wallet, các phiên bản wallet sau này toàn dùng compressed public key như là một cải tiến đã được chấp nhận). Address được tạo ra từ uncompressed public key thì gọi là uncompressed address, tương tự address được tạo ra từ compressed public key thì gọi là compressed address.

Tạo địa chỉ dạng P2PKH (pay to public key hash).

Địa chỉ được tạo bằng cách băm sha256, sau đó được băm bằng cách sử dụng ripemd160. Sau đó thực hiện base58check cho phép mã băm được hiển thị theo cách nhỏ gọn hơn (sử dụng nhiều chữ cái hơn trong bảng chữ cái).

Encrypted public key = ripemd160(sha256(K))

Checksum = first 4 bytes of sha256(sha256(Encrypted public key))

Hex address = base58(Mainnet encrypted public key + Checksum)

Tạo địa chỉ bằng Keccak256

Tính hash của public key và lấy 20 bytes cuối cùng làm địa chỉ

Address = last 20 bytes of Keccak256(K)

WIF và WIF-Compressed

Việc xuất hiện hai loại Address như đã nêu trên dẫn theo hệ lụy là làm sao khi import private key vào wallet, wallet sẽ sử dụng loại address nào. Do đó ta cần dùng đến wallet import format – là một định dạng cho phép Export / Import một private key giữa các wallet kèm theo flag (cờ) cho biết có sử dụng compressed public key hay không.

* WIF được tạo ra từ việc thực hiện Base58 Check Encoding trên k, kết quả là một chuỗi bắt đầu bằng 5.
* WIF-Compressed được tạo ra từ việc thực hiện Base58 Check Encoding trên k01 (nối thêm 01 vào sau k trước Encode) kết quả là một chuỗi bắt đầu bằng K hoặc L.

Base58 Encoding thực chất giống Base64 Encoding nhưng sử dụng bộ ký tự đã được loại bỏ bớt các ký tự dễ gây nhầm lẫn như: 0 (số 0), O (chữ o hoa), I (chữ L thường), I (chữ I hoa) và các ký tự “+” và “/”.

Base58 Check Encoding là phương thức Encoding mở rộng, thêm Version Prefix ở đầu chuỗi rồi tạo thêm Checksum vào cuối chuỗi mới sau đó mới áp dụng Base58 Encoding.

Dưới đây là sơ đồ mô tả cơ chế tạo ra Bitcoin Address

Diagram

Description automatically generated

Diagram

Description automatically generated

Secp256k1

XXX: <https://en.bitcoin.it/wiki/Secp256k1>

https://medium.com/witnet/announcing-our-own-elliptic-curve-cryptography-in-solidity-1df1c0ec90ef

Merkle Tree: là một cấu trúc dữ liệu dạng cây trong đó mọi nút lá được dán nhãn bằng giá trị băm của khối dữ liệu và mọi nút không phải là nút lá được dán nhãn bằng giá trị băm của các nút con của nó. Gốc của cây đóng vai trò là một bản băm đối với toàn bộ cây. Cây băm cho phép xác minh hiệu quả và an toàn nội dung của các cấu trúc dữ liệu lớn. Cây băm là một dạng tổng quát của danh sách băm và chuỗi băm.

Cây Merkle được ứng dụng vào công nghệ blockchain của Bitcoin để xác minh và lưu trữ dữ liệu giao dịch.

Khái niệm cây là một khái niệm phổ biến trong khoa học máy tính, đề cập đến một cấu trúc bao gồm các nút cha và các nút con phân nhánh từ các nút cha này.

Với dạng cây băm Merkle Tree này, số lượng phép tính toán hàm băm tỉ lệ với logarithm của số nút lá của cây. Trong khi với các danh sách băm thì số lượng này tỉ lệ với số lượng nút của chính nó. Chính điều này làm cho việc tính toán và xác minh, tìm kiếm giá trị băm trên khối diễn ra nhanh hơn trên các khối dữ liệu lớn.

Cây Merkel, theo cách hiểu đơn giản, lấy rất nhiều dữ liệu, nén nó thành một chuỗi ký tự đơn giản có thể chứng minh tính xác thực của dữ liệu được giữ trong đó, mà không tiết lộ dữ liệu đó là gì. Tương tự một tệp nén (.zip, .rar, …), nếu được đặt tên chính xác theo một tiêu chuẩn nhất định, người dung có thể nhận ra nội dung mà không cần phải giải nén và mở các tệp chứa. Chuỗi ký tự (tiêu đề) này được gọi là giá trị băm. Giá trị băm được tạo ra từ hàm băm là hàm một chiều, có nghĩa là nếu bạn đặt cùng một dữ liệu, bạn sẽ luôn nhận được cùng một giá trị băm, nhưng bạn không thể lấy giá trị băm đó và trích xuất ra được dữ liệu gốc.

XXX: https://cryptoviet.com/cay-merkle-la-gi

Đồ thị tri thức chuỗi chéo (Cross chain knowledge graph)

Đồ thị tri thức là một phần cơ bản nhất của hệ thống, nó có chức năng lưu trữ và trình bày dữ liệu tổng hợp từ nhiều mạng blockchain. Đồ thị và giao thức ??? làm các mạng blockchain độc lập liên kết với nhau và tạo nên mạng lưới siêu liên kết blockchain.

XXX: https://docs.trava.finance/whitepaper/cross-chain-knowledge-graph

Nhận dạng chuỗi chéo (Cross chain identification):

Giao thức nhận dạng chuỗi chéo nắm vai trò quan trọng trong kiến trúc hệ tri thức chuỗi chéo.

XXX: <https://docs.trava.finance/whitepaper/cross-chain-identification>

Private: 3edae309e75a778f88af5d8017d93f477297abaaae143439604ea08b38c0ec85

Address: 0x8B6ff17E6a61879661296CBA916BeC85F6649062

Public compressed: 03cc874cab4bc1daf7575a7370b874baf0a490dc11836d9af1c700d5d83297e367

Hash160 compressed: 830aa56eadc8132e044ee5446b3cf9178123e005

Public uncompressed: 04cc874cab4bc1daf7575a7370b874baf0a490dc11836d9af1c700d5d83297e3671ec38d2c4593c3801eda1d87dd5cd339e5f4ed9c5f33ce88124afd16e24af239

Hash160 uncompressed: f2a8c0e777c40ecbb60644e16ed8b78af2ca96f3

Hướng dẫn về ngôn ngữ Solidity

https://hackernoon.com/self-sovereign-identity-smart-contracts-and-web-30-4dm375d