**Bitcoin and Crytocurrency technology**

**Chapter 1: Introduction Crypto ( mã hóa ) and Cryptocurrencies ( tiền điện tử )**

1. Cryptographic hash function

Hash function là một hàm toán học gồm 3 thuộc tính sau:

+ input any string, any size.( dữ liệu đầu vào của hash funtion có thể là bất cứ một chuỗi nào với bất cứ một kích thước nào, không giời hạn dung lượng của dữ liệu đầu vào).

+ Fixed size output.( với input là tùy ý, thì output sẽ được mặc định là một chuỗi gồm 256 bit).

+ Efficiently computable.( với input đã cho thì chúng ta có thể thấy được output của hashfunction trong một khoảng thời gian hợp lý, thời gian ở đây có thể tính bằng hàm O(n) với n là số bit của input ).

Ba thuộc tính quan trong để tạo nên một hash function bảo mật về mặt mã hóa:

* Collision Resistance
* Hiding properties
* Puzzle Friendliness
  1. Collision Resistance

Collision Resistance ( không đụng độ ) ở đây có nghĩa là với mỗi input x!=y thì qua hash funtion chúng ta không thể tìm thấy H(x)=H(y).

* 1. Hiding properties (ẩn các giá trị của input).

Điều mà chúng ta muốn là: cho H(x), mà không thể tìm thấy được x.

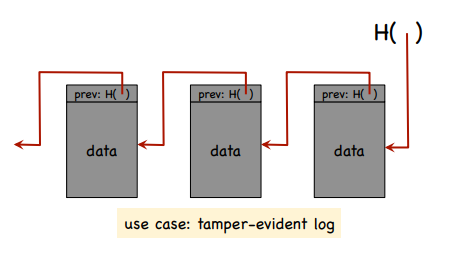
Tính chất này có nghĩa là khi đưa ra kết quả y của hàm hash H(x)=y thì không có cách nào tìm ra giá trị đầu vào x.

Trên thực tế để đạt được tính chất này khi tính hàm hash của biến đầu vào xngười ta sẽ đính thêm một biến đầu vào bí mật (secret value) r được chọn ngẫu nhiên để khi đưa ra H(r||x) thì không thể nào tìm được giá trị x.

* 1. Puzzle Friendliness: một hàm hash H được gọi là Puzzle Friendliness nếu với mọi giá trị n-bit đầu ra (output) y, khi chọn k từ một phân phối với min-entropy cao (hiểu là một giá trị ngẫu nhiên), thì không thể tìm dược giá trị x để H(k||x)=y trong thời gian ít hơn 2n .
  2. Hash Pointer.

Hash pointer là một con trỏ thông thường (pointer) nhưng có kèm theo giá trị hash của nội dung được trỏ tới.

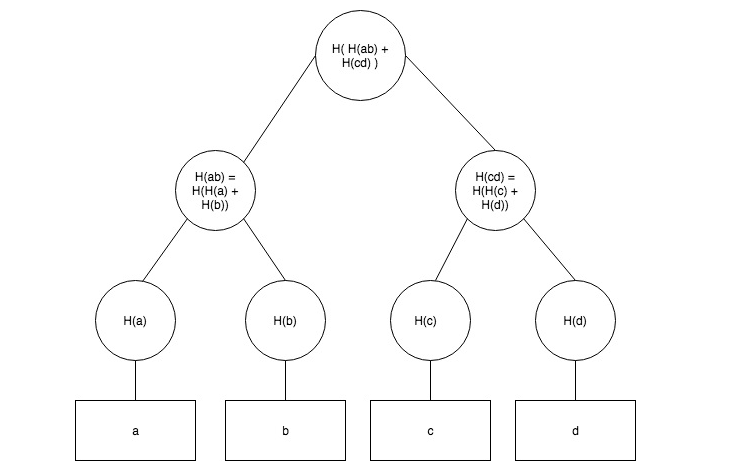
Trong blockchain, ngoài việc có thể trỏ tới block trước đó, mỗi block còn có thể lưu giá trị digest (hash value) của khối được trỏ tới. Thông qua việc kiểm tra giá trị hash, ta có thể nhận dạng khối được trở tới có bị thay đổi hay không.

Ví dụ:  Một người nào đó muốn thay đổi nội dung của dữ liệu của Blockchain ở một khối *k* trong danh sách; vì nội dung khối *k* này bị thay đổi, con trỏ hash của khối *k+1* sẽ không còn đúng. Lúc này khi ta tính giá trị hash của nội dung khối *k*, thì giá trị này sẽ không giống như giá trị hash được lưu ở con trỏ hash ở khối sau đó là khối *k+1*.

* 1. Merkle tree.

Merkle tree là 1 dạng mô hình dạng cây nhị phân trong đó các điểm ngọn là hash value của khối data, các điểm hash khác ngọn là các giá trị hash gộp.

Các input được đặt ở các lá (ở các node ko có node con) và sau đó các cặp giá trị của các child node sẽ được hash lại với nhau để tạo ra 1 parent node mới (internal node) cho tới khi đạt được 1 giá trị hash đơn được gọi là Merkle root



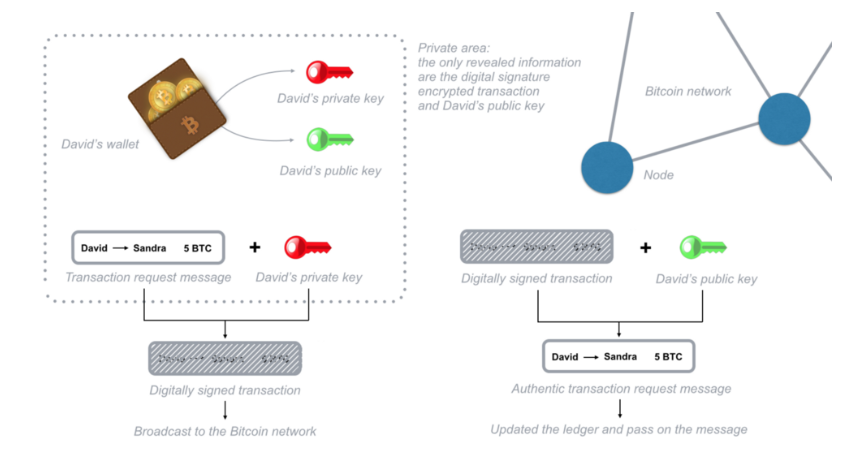
* 1. Public key và Private key
     1. Private key
* Private key là 1 số biểu diễn bởi 256-bit.
* Vậy thì theo lý thuyết, ta sẽ có 2256 private key nên việc “đoán” private key là điều không thể.
* Mất private key, tài khoản sẽ bị mất vĩnh viễn.
* Để sinh ra 1 private key, ta cần 1 số biểu diễn dưới dạng 256 bit thỏa mãn chuẩn secp256k1. Khi kết hợp 2 yếu tố đó, ta có thể viết 1 script ngẫu nhiên 1 số 256-bit trong khoảng từ 0x1 đến 0xFFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFE BAAE DCE6 AF48 A03B BFD2 5E8C D036 4140 thì private key đó hợp lệ.
* Thường 1 chuỗi private key sẽ bắt đầu với 5, ví dụ như:

5J3mBbAH58CpQ3Y5RNJpUKPE62SQ5tfcvU2JpbnkeyhfsYB1Jcn

* + 1. Public key.
* Công thức tính public key đơn giản hóa như sau:

**Public\_key = Private\_key \* G**

* Public key được tạo ra từ Private key theo một chiều, có nghĩa là không thể tìm ra Private key từ Public key. Thường dùng public key và chữ ký số để trao đổi Bitcoin với nhau.
* Public key được dùng như là địa chỉ người nhận trong các giao dịch trên blockchain.
* Thông qua việc dùng public key và chữ ký số, bất kỳ ai trong mạng lưới cũng có thể kiểm tra transaction đó có đúng hay không.



**Chapter 3: Mechanics of Bitcoin ( bitcoin cơ học )**

3.1 Bitcoin Transactions ( giao dịch bitcoin)

Để sử dụng bicoin cần phải có ví quản lý bitcoin ( bitcoin wallet ) , quản lý private key cũng như khởi tạo các giao dịch bitcoin. Ví của bitcoin gồm 3 loại: full node wallet, SPV wallet, light wallet.

Để tạo 1 transaction thì chúng ta sử dụng ví quản lý bitcoin để tạo transaction, nhờ full node gửi tới 1 full node khác từ đó broadcast lên mạng

Đặc điểm của transactions trong bitcoin là:

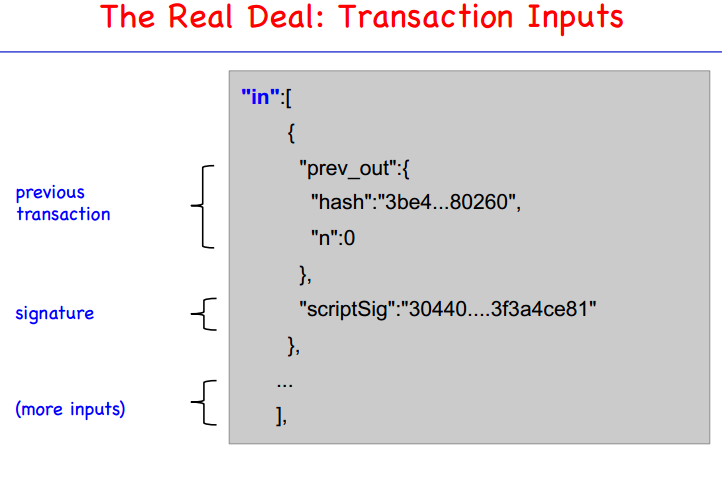
Bitcoin không lưu giá trị của tài khoản mà chỉ lưu các giao dịch của tài khoản đó. Các giao dịch được xác định rõ input ( do người gửi ) cũng như output(người nhận). Và các giao dịch cũng sẽ có một số nhân dạng duy nhất

Một giao dịch bitcoin thực sự thì nó trông giống như một chuỗi JSON, nó gồm có 3 phần: metadata (siêu dữ liệu), input ( các dữ liệu đầu vào ), output ( các dữ liệu đầu ra)

Trong phần metadata: có mã băm của giao dịch nó hoạt động giống như một ID duy nhất cho giao dịch cho phép thực hiện hash pointers, và một số parameter liên quan ( ver, vin\_sz, vout\_sz, lock\_time, size.).



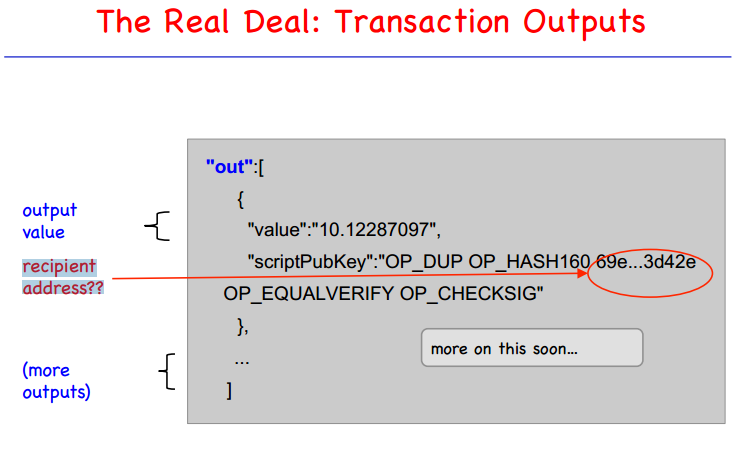
Phần input sẽ có giao dịch trước cụ thể, chúng ta cũng có được hash của giao dịch trước hoặc là hash pointers, tiếp theo index là đầu ra từ giao dịch mà bạn đang thực sự xác nhận quyền sở hữu, sau đó là signature vì nó đảm bảo cho việc những kết quả của giao dịch trước đó được xác nhận bởi chính người sở hữu



Và bây giờ là phần output gồm có 2 giá trị: value, scriptPubKey

Value: mỗi output có thể có 1 value khác nhau nhưng tỗng tất cả value trong output phải nhỏ hơn tổng của tất cả input

scriptPubkey: giống như địa chỉ của người nhận nó là một public key cụ thể xác định người nhận



3.2 Bitcoin script, Applications of Bitcoin Scripts

scriptPubKey là một script khóa được đặt trên đầu ra của một giao dịch Bitcoin, và yêu cầu các điều kiện nhất định phải được đáp ứng để người nhận chi tiêu bitcoin của mình. Ngược lại, scriptSig là một script không khóa, thỏa mãn các điều kiện được đặt trên đầu ra bởi scriptPubKey, và phải được cho phép chi tiêu.

Cụ thể, scriptPubKey còn được gọi là PubKey Script bên ngoài code của Bitcoin, còn scriptSig gọi là các script Chữ ký.

Cả hai scriptPubKey và scriptSig đều được viết bằng Script, một ngôn ngữ lập trình được sử dụng để xây dựng các giao dịch Bitcoin. Mặc dù thiếu nhiều chức năng của ngôn ngữ lập trình ngày nay, nhưng chính điều này đã làm cho script an toàn hơn do số lượng hoạt động bị hạn chế.

Hiểu đầu vào và đầu ra

Muốn nắm cách hàm scriptPubKey và scriptSig hoạt động trong một giao dịch Bitcoin, trước tiên bạn cần hiểu cơ cấu của các giao dịch.

Giao dịch bitcoin sử dụng các đầu vào UTXO chưa dùng từ các giao dịch trước đó làm đầu vào trong quá trình xây dựng một giao dịch mới. Để minh họa, hãy xem xét rằng Alice muốn gửi cho Bob 1 bitcoin, tuy nhiên, cô ấy biết rằng một khoản phí giao dịch bắt buộc là 0.25 bitcoin. Sử dụng các UTXO trước đây làm đầu vào cho giao dịch của cô với Bob, các đầu vào đó có thể là:

Đầu vào 1 – 0.25 BTC

Đầu vào 2 – 0.25 BTC

Đầu vào 3 – 0.25 BTC

Đầu vào 4 – 0.25 BTC

Đầu vào 5 – 0.25 BTC

Có tính đến phí giao dịch là 0.25 BTC, đầu ra của giao dịch, tức là số tiền bitcoin Bob thực sự nhận được, sẽ trông giống như sau:

Đầu ra 1 – 0.25 BTC

Đầu ra 2 – 0.25 BTC

Đầu ra 3 – 0.25 BTC

Đầu ra 4 – 0.25 BTC

Bob do đó sẽ nhận được một bitcoin duy nhất vào cuối giao dịch.

Tuy nhiên, để Bob chi tiêu số bitcoin của mình, mỗi đầu ra mà Bob nhận được sẽ chứa một script khóa – scriptPubKey, trước tiên phải được thỏa mãn bởi script không khóa – scriptSig.

Ví dụ, khi Alice quyết định bắt đầu mở giao dịch với Bob, các đầu ra mà Bob nhận được sẽ chứa một lượng bitcoin, và chỉ có thể được sử dụng khi scriptPubKey đính kèm thỏa mãn các điều kiện được trình bày. Khi Bob quyết định chi tiêu các đầu ra này, anh ta sẽ tạo một đầu vào bao gồm scriptSig. Đặc biệt, nó phải đáp ứng các điều kiện mà Alice đã đặt trong các đầu ra trước đó trước khi anh ta có thể sử dụng chúng.

Như có thể thấy từ hình ảnh, scriptSig, chứa một sig và PubK, hoặc chữ ký số và khóa công khai, phải được cung cấp để script khóa được thỏa mãn. Ngược lại, scriptPubKey, chứa PubKHash, còn được gọi là băm khóa công khai hoặc đơn giản hơn là địa chỉ Bitcoin.

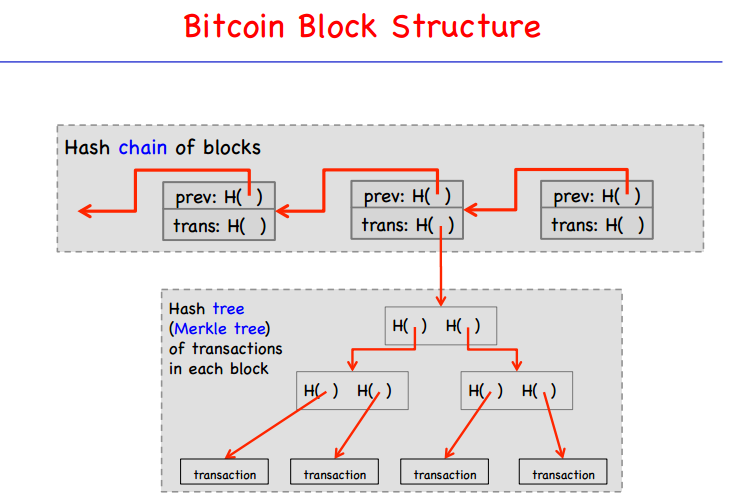
Quá trình hoạt động sẽ là, scriptSig và scriptPubKey được kết hợp và thực hiện theo thứ tự, với script mở khóa được thực hiện trước tiên. Ví dụ, trường hợp Bob quyết định chi 1 bitcoin mà anh ta nhận được từ Alice. Trước tiên, anh ta phải mở khóa các đầu ra, sau đó bị khóa khi người nhận nhận được 1 bitcoin.

Tóm lại, scriptPubKey là một script khóa được sử dụng trong máy lõi Bitcoin, và đặt điều kiện trên một đầu ra giao dịch phải được thỏa mãn trước khi nó có thể được sử dụng. Script không khóa, scriptSig, sẽ đáp ứng các điều kiện được đặt trên đầu ra bởi scriptPubKey

3.3 Bitcoin block

Cấu trúc dữ liệu chuỗi khối trông như thế nào?

Đó là một sự kết hợp khá thông minh của hai cấu trúc dữ liệu dựa trên băm khác nhau. chúng ta có một chuỗi băm của khối(hash chain of blocks.) và sau đó một con trỏ đến một số dữ liệu transaction cũng như một con trỏ đến khối trước đó và chuỗi. Và sau đó chúng ta có một cây của tất cả các transaction được bao gồm trong mỗi khối. Vì vậy, đây là một cây băm(hash tree )hoặc được gọi là một cây Merkle.



Đây là một khối Bitcoin. Block header chứa tất cả siêu dữ liệu cho khối đó và sau đó là cây giao dịch Merkle. Về cơ bản, một danh sách dài các giao dịch, tất cả các băm được bố trí trong cấu trúc cây này mang lại khả năng để chứng minh giao dịch nào được bao gồm trong một khối.

