**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.

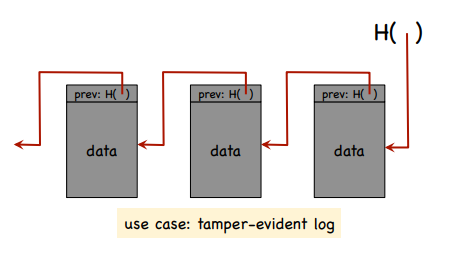
Tính chất này có nghĩa là khi đưa ra kết quả y của hàm hash H(x)=y thì không có cách nào tìm ra giá trị đầu vào x.

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.

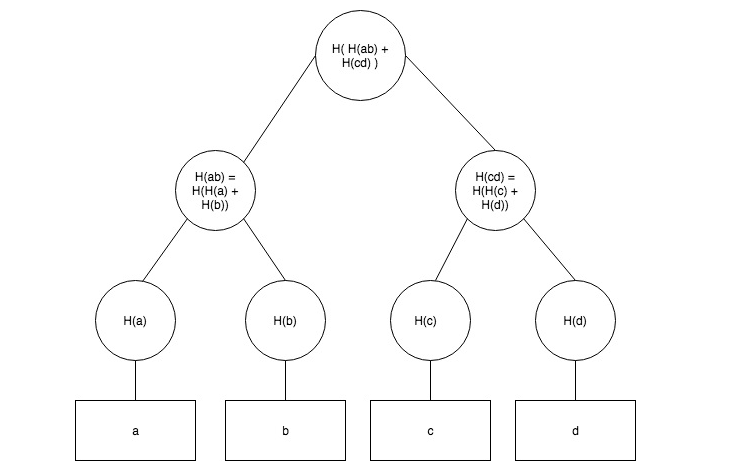
Trong blockchain, ngoài việc có thể trỏ tới block trước đó, mỗi block còn có thể lưu giá trị digest (hash value) của khối được trỏ tới. Thông qua việc kiểm tra giá trị hash, ta có thể nhận dạng khối được trở tới có bị thay đổi hay không.

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



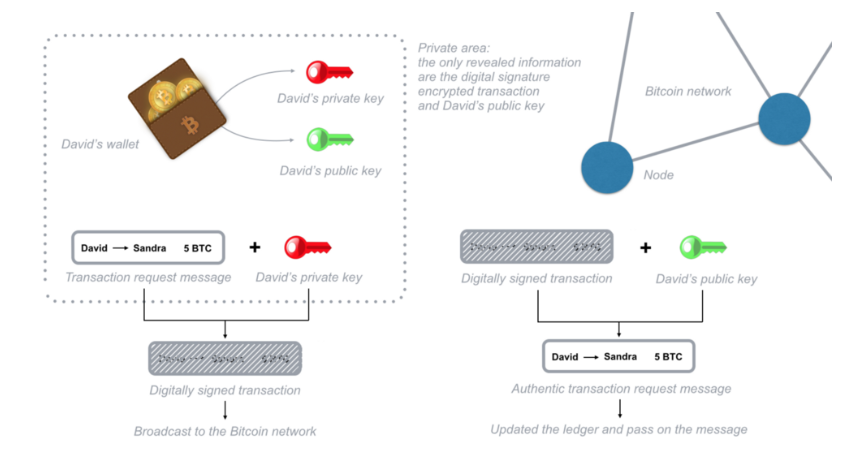
* 1. Public key và Private key
     1. Private key
* Private key là 1 số biểu diễn bởi 256-bit.
* Vậy thì theo lý thuyết, ta sẽ có 2256 private key nên việc “đoán” private key là điều không thể.
* Mất private key, tài khoản sẽ bị mất vĩnh viễn.
* Để sinh ra 1 private key, ta cần 1 số biểu diễn dưới dạng 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.



**.**