

HUST

ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

ONE LOVE. ONE FUTURE.



ĐẠI HỌC
BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY
OF SCIENCE AND TECHNOLOGY

Kinh tế chính trị Mác-Lênin

Đoàn Xuân Công Đạt – 20235030
Mã lớp: 155994

ONE LOVE. ONE FUTURE.

Giới thiệu

- Bài thuyết trình sẽ giúp chúng ta hiểu rõ hơn về Bitcoin, cách thức giao dịch Bitcoin, mối quan hệ của nó với nền kinh tế.





ĐẠI HỌC
BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY
OF SCIENCE AND TECHNOLOGY

Phần 1: Giới thiệu về Bitcoin

ONE LOVE. ONE FUTURE.

1.1 Bitcoin là gì ?

- Bitcoin là hệ sinh thái tiền kĩ thuật số, cho phép lưu trữ và truyền giá trị giữa người dùng qua mạng lưới phi tập trung, được tạo ra bởi việc “đào” bằng cách thực hiện những tính toán.
- Giao thức Bitcoin hoạt động chủ yếu trên Internet và dễ dàng tiếp cận qua các thiết bị như laptop, smartphone.
- Người dùng có thể sử dụng Bitcoin để mua bán, trao đổi hàng hóa và tiền tệ.

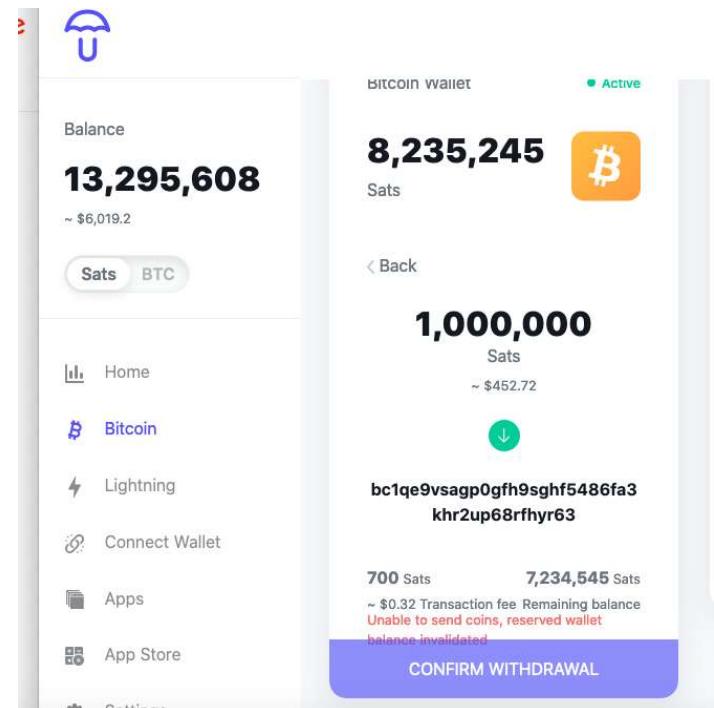


January 2023 - May 2024



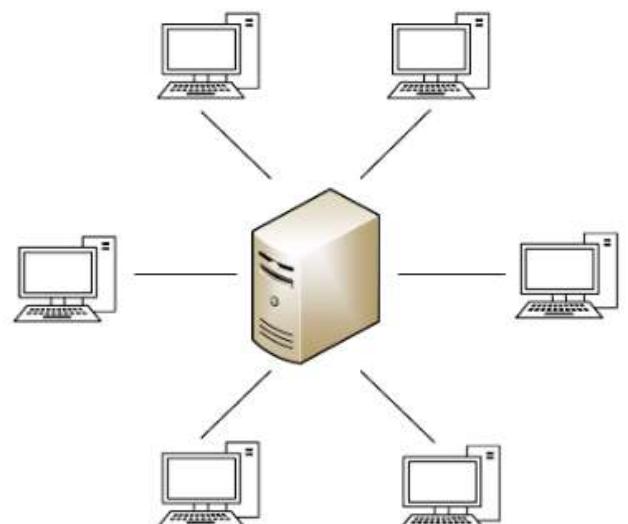
1.2 Đặc điểm chính của Bitcoin

- Bitcoin hoàn toàn là tiền ảo, không có đồng tiền vật lý.
- Quyền sở hữu Bitcoin được xác nhận qua các khóa cá nhân (private key), cho phép người dùng thực hiện giao dịch.
- Ví điện tử (Bitcoin wallet) sẽ là nơi lưu trữ các khóa này, thường là trên máy tính hoặc điện thoại (Một số phổ biến như: Satoshi client, Bitcoin core,...)

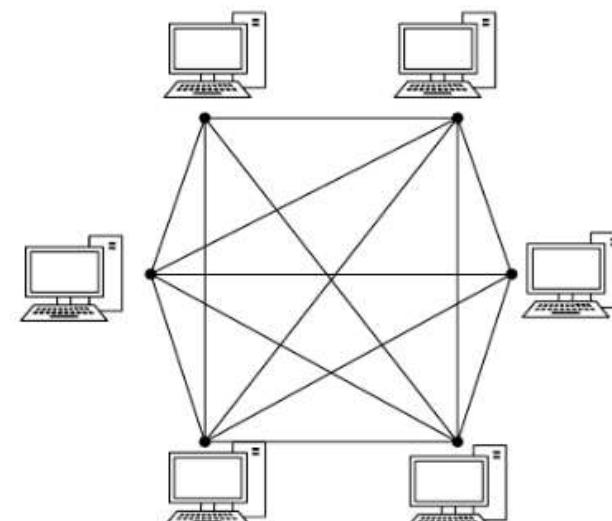


1.3 Mạng lưới phi tập trung và khai thác Bitcoin

- **Mạng phi tập trung:** Bitcoin không có máy chủ trung tâm hay điểm kiểm soát, mọi giao dịch được thực hiện qua mạng ngang hàng (P2P – Peer to Peer).
- **Khai thác (mining):** Khi chủ sở hữu các Node(thiết bị trên mạng Blockchain) đóng góp tài nguyên máy tính của họ để lưu trữ và xác thực giao dịch, họ có cơ hội nhận phần thưởng là các đồng tiền điện tử (Bitcoin) và phí giao dịch.



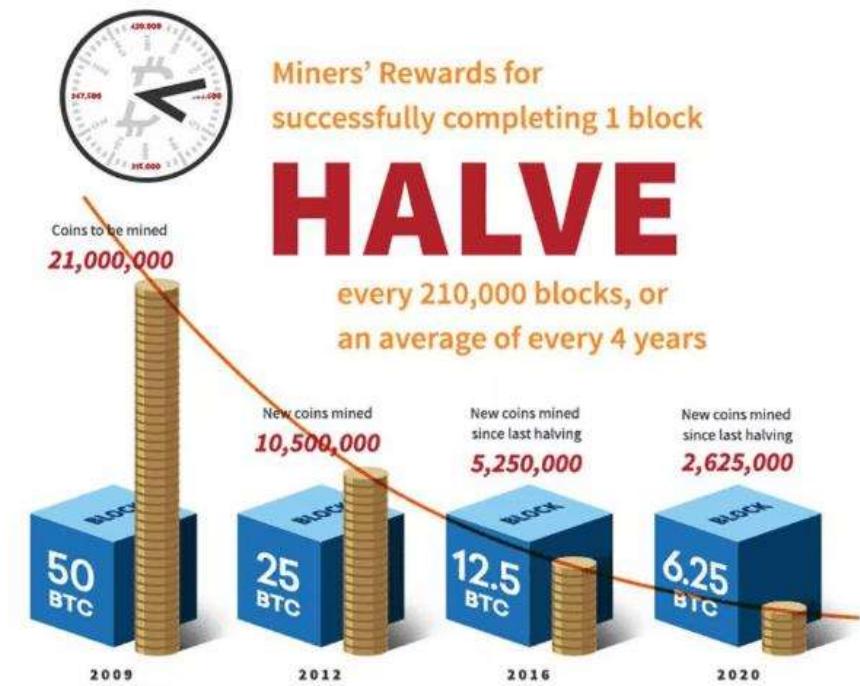
Mạng tập trung



Mạng ngang hàng

1.4 Quy tắc kiểm soát nguồn cung Bitcoin

- Giao thức Bitcoin tự điều chỉnh độ khó của việc khai thác để đảm bảo rằng cứ mỗi 10 phút sẽ có 1 thợ đào thành công.
- Cứ 4 năm, số lượng Bitcoin phát hành sẽ giảm một nửa (**halving**), và tổng số Bitcoin tối đa là 21 triệu.
 - Điều này khiến Bitcoin trở thành tiền tệ **giảm phát**, không thể bị lạm phát như tiền tệ truyền thống.



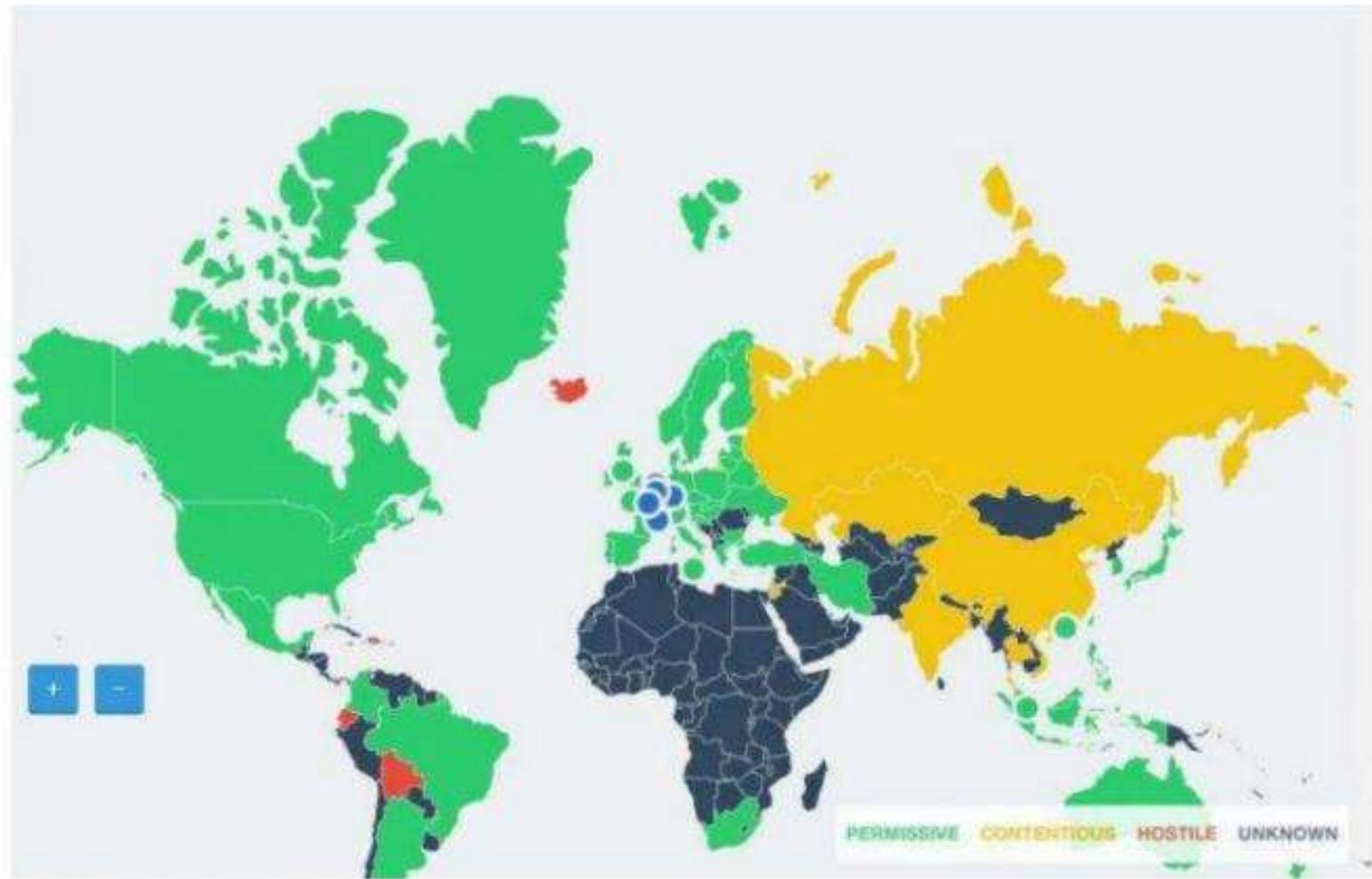
1.5 Các yếu tố cốt lõi của Bitcoin

- Bitcoin gồm 4 yếu tố chính:
 1. Mạng ngang hàng phi tập trung (P2P)
 2. Sổ cái công khai (Blockchain)
 3. Bộ quy tắc đồng thuận để xác thực giao dịch và phát hành tiền tệ (con-sensus rules)
 4. Thuật toán PoW (Proof-of-work) để đạt được sự đồng thuận.
- P/s: **Công nghệ Blockchain** là một loại chương trình để lưu, xác nhận, vận chuyển và truyền thông dữ liệu trong mạng thông qua các nút phân phối của riêng nó và không phụ thuộc vào bên thứ ba.

1.6 Lợi ích của Bitcoin

- **Tốc độ:** Giao dịch nhanh chóng, không cần qua trung gian.
- **Bảo mật:** Sử dụng **SHA-256** và **Proof-of-Work** để xác minh, đảm bảo khó bị tấn công hay làm giả. Cứ mỗi 10 phút, một khối giao dịch mới được xác nhận trên blockchain, giúp bảo vệ hệ thống khỏi các cuộc tấn công.
- **Tiết kiệm:** Việc không cần trung gian giúp giảm chi phí giao dịch, đặc biệt trong giao dịch xuyên biên giới. Việc không cần máy chủ trung tâm cũng sẽ đảm bảo an toàn và tiết kiệm chi phí bảo trì may chủ.
- **Không biên giới:** **Bitcoin** cho phép người dùng thực hiện giao dịch toàn cầu một cách dễ dàng mà không cần chuyển đổi tiền tệ hay phụ thuộc vào các hệ thống tài chính quốc gia.

1.6 Lợi ích của Bitcoin



Hình 2.5: Biểu đồ chấp nhận BTC trên thế giới

1.7 Kết luận

- Bitcoin là phát minh của nhiều thập kỷ nghiên cứu về mật mã học và hệ thống phân tán.
- Nó không chỉ là một loại tiền tệ, mà còn là nền tảng cho việc truyền giá trị và bảo vệ tài sản số thông qua hệ thống phi tập trung.
- Trong tương lai, Bitcoin có thể còn phát triển hơn với nhiều ứng dụng khác ngoài tiền tệ.





Phần 2: Giao dịch Bitcoin hoạt động thế nào?

Bitcoin có tác dụng gì trong kinh tế ?

2.1 Giới thiệu về giao dịch Bitcoin

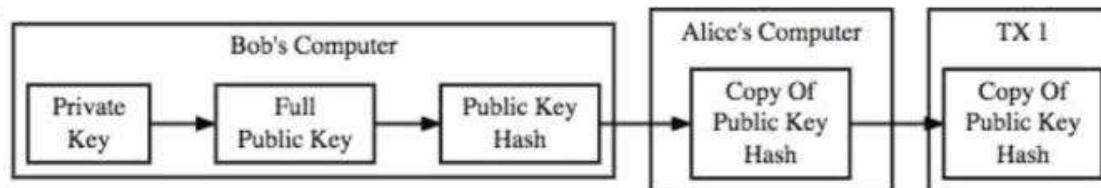
- **Giao dịch (Transaction):** là việc chuyển Bitcoin từ người này sang người khác.
- Mỗi giao dịch chứa:
 - Đầu vào (Input): Thông tin về nguồn Bitcoin(Previous.tx, Index, ScriptSig).
 - Đầu ra (Output): Địa chỉ người nhận và số lượng Bitcoin.(Value, ScriptPubKey).
- Các thành phần chính của Giao dịch:
 - Người dùng
 - Giao dịch (Transaction)
 - Thợ đào (Miners)
 - Blockchain (Sổ cái)

2.1 Giới thiệu về giao dịch Bitcoin

- Khóa trong giao dịch:

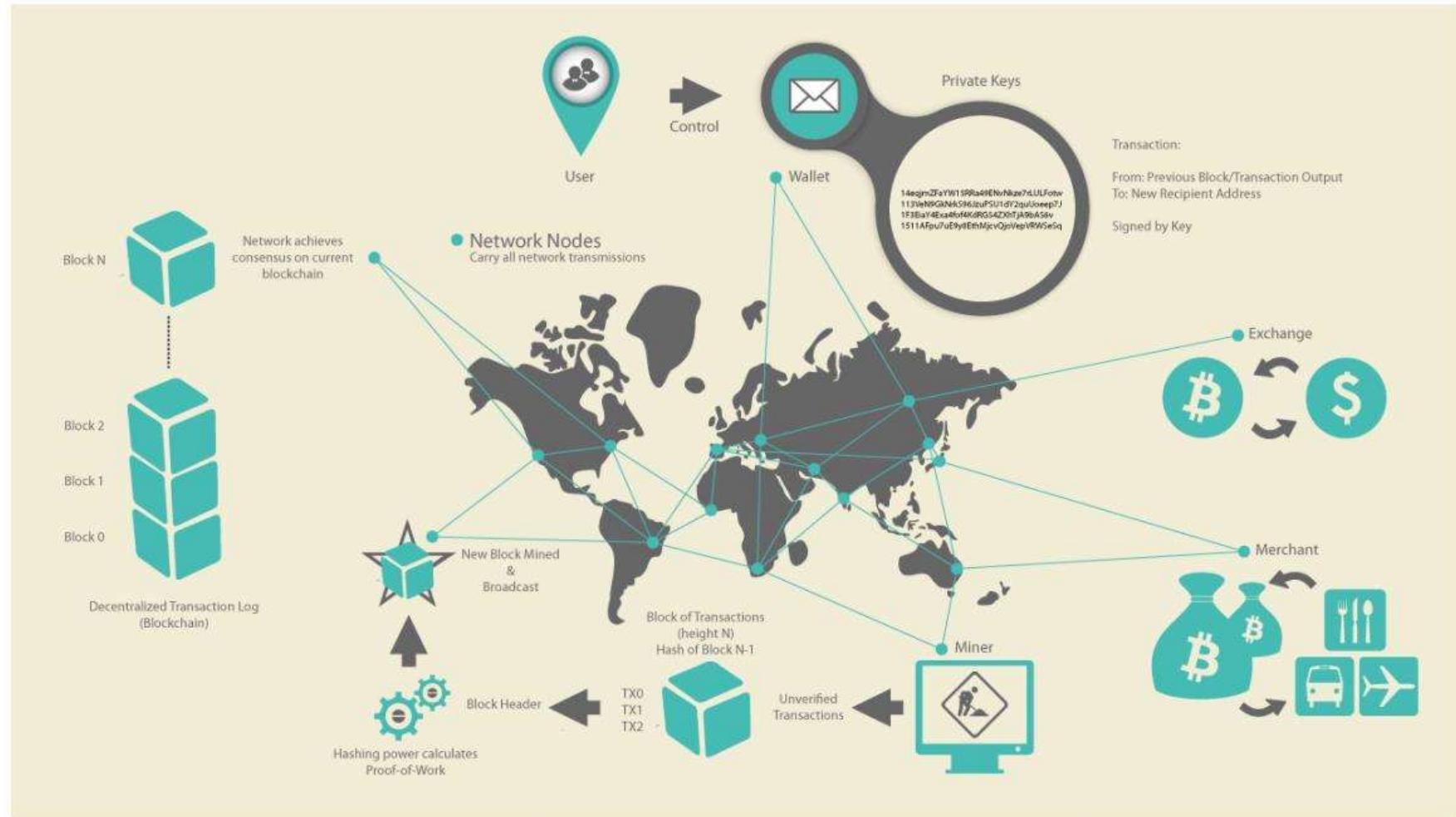
Như hình 2.9 Alice muốn gửi BTC cho Bob, Bob sẽ cần tạo ra cặp khóa gồm khóa riêng tư và khóa công khai, Bitcoin sử dụng thuật toán chữ ký số đường cong Elliptic (ECDSA) để thực hiện ký các giao dịch.

Địa chỉ ví của Bob chính là giá trị băm của khóa công khai được mã hóa base58. Alice gửi BTC vào ví của Bob bằng cách giã mã base58 để lấy giá trị băm khóa công khai của Bob.



Hình 2.9: Tạo khóa để thực hiện giao dịch trong bitcoin

2.1 Giới thiệu về giao dịch trong Blockchain



2.2 Quá trình giao dịch

1. Khởi tạo giao dịch

- Tạo cặp khóa:

- Khóa riêng tư: Người nhận tạo ra khóa này để ký các giao dịch và chứng minh quyền sở hữu BTC, sử dụng khóa này để chuyển BTC.

- Khóa công khai: Người dùng sử dụng để tạo địa chỉ BTC, nơi người khác có thể gửi BTC.

- Tạo địa chỉ ví:

- Địa chỉ ví của Bob là **giá trị băm** của khóa công khai, sử dụng **SHA-256** và **RIPEMD-160**, sau đó mã hóa bằng **Base58Check**.

2. Gửi BTC:

- Nhập địa chỉ ví của người nhận (mã hóa Base58) vào giao dịch.

- Hệ thống giải mã Base58 lấy giá trị băm khóa công khai của Bob.

2.2 Quá trình giao dịch

- 3. Xác nhận giao dịch:
 - **Thợ đào** xác thực giao dịch của để đảm bảo rằng người gửi có đủ BTC để gửi cho bạn và rằng giao dịch không phải là một giao dịch gian lận (Ví dụ: chi tiêu gấp đôi(Double-Spending),...).
 - Giao dịch được **ghi vào một khối** sau khi thợ đào giải bài toán Proof-of-Work.
- 4. Kết quả giao dịch
 - Sau khi giao dịch được xác nhận, BTC sẽ được gửi đến ví của bạn.
 - Người nhận có thể kiểm tra số dư trên ví điện tử (Bitcoin wallet).
 - Thông tin giao dịch được ghi nhận trên Blockchain, tạo ra một lịch sử giao dịch mà 2 bên có thể truy cập bất cứ lúc nào.

2.2 Quá trình giao dịch

- 3. Xác nhận giao dịch:
 - - **Thợ đào** xác thực giao dịch của để đảm bảo rằng người gửi có đủ BTC để gửi cho bạn và rằng giao dịch không phải là một giao dịch gian lận (Ví dụ: chi tiêu gấp đôi(Double-Spending),...).
 - - Giao dịch được **ghi vào một khối** sau khi thợ đào giải bài toán Proof-of-Work.
- 4. Kết quả giao dịch
 - - Sau khi giao dịch được xác nhận, BTC sẽ được gửi đến ví của bạn.
 - - Người nhận có thể kiểm tra số dư trên ví điện tử (Bitcoin wallet).
 - - Thông tin giao dịch được ghi nhận trên Blockchain, tạo ra một lịch sử giao dịch mà 2 bên có thể truy cập bất cứ lúc nào.

2.3 Kết luận về ứng dụng của Bitcoin với nền kinh tế

- **Mối Quan Hệ Của Bitcoin Với Nền Kinh Tế Tương Lai**

- 1. Phân cấp tài chính (Decentralization):** Bitcoin giúp giảm sự phụ thuộc vào các tổ chức tài chính trung gian, mở ra cơ hội cho một nền kinh tế phi tập trung.
- 2. Giá trị lưu trữ tài sản (Store of Value):** Bitcoin ngày càng được coi là “vàng kỹ thuật số” - một công cụ lưu trữ giá trị an toàn và chống lạm phát.
- 3. Tăng cường giao dịch toàn cầu:** Với chi phí giao dịch thấp và không biên giới, Bitcoin có tiềm năng cách mạng hóa thương mại quốc tế.
- 4. Ứng phó với các bất ổn kinh tế:** Trong bối cảnh khủng hoảng kinh tế, Bitcoin trở thành lựa chọn thay thế cho các loại tiền tệ truyền thống dễ bị lạm phát và kiểm soát.
- 5. Khả năng phát triển bền vững:** Dù đối mặt với các thách thức về quy định và môi trường, Bitcoin vẫn có vai trò quan trọng trong tương lai kinh tế kỹ thuật số.

Tổng kết

- **Bitcoin** là một hình thức tiền điện tử phi tập trung, ra đời nhằm thoát khỏi sự kiểm soát của các tổ chức tài chính truyền thống.
- Giá trị của **Bitcoin** không dựa trên giá trị lao động mà chủ yếu dựa vào cung – cầu thị trường và niềm tin của nhà đầu tư, trái ngược với quan điểm của Mác về giá trị lao động.
- **Bitcoin** thể hiện sự phát triển của công nghệ blockchain và tác động mạnh mẽ đến quá trình lưu thông tiền tệ, có tiềm năng thay đổi cách thức lưu thông giá trị trong nền kinh tế.
- Tuy nhiên, **Bitcoin** cũng gặp thách thức về tính bền vững, tính pháp lý, và sự biến động cao của giá trị, làm tăng nguy cơ bất ổn kinh tế.



ĐẠI HỌC
BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY
OF SCIENCE AND TECHNOLOGY

Lời cảm ơn

ONE LOVE. ONE FUTURE.



ĐẠI HỌC
BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY
OF SCIENCE AND TECHNOLOGY

Chương 3: Khóa, địa chỉ

ONE LOVE. ONE FUTURE.

3.1 Tổng quan về thuật toán băm SHA-256

- **SHA-256** (Secure Hash Algorithm 256-bit) là một hàm băm mật mã, tạo ra một giá trị băm cố định dài 256-bit (32 byte).
- **Nguyên lý hoạt động:** Mọi dữ liệu đầu vào (kích thước bất kỳ) sẽ được chuyển đổi thành một chuỗi băm 64 ký tự (hexadecimal).
- **Đặc điểm:**
 - Không thể đảo ngược: Không thể suy ngược từ giá trị băm để tìm lại dữ liệu gốc.
 - Đầu vào khác nhau tạo ra băm hoàn toàn khác biệt (ngay cả khi chỉ thay đổi một bit).
 - Xác suất trùng lặp băm là cực kỳ nhỏ, giúp bảo mật và chống giả mạo dữ liệu.



3.1 Tổng quan về thuật toán băm SHA-256

- Vai trò của SHA-256:
 - **Tạo khóa:** SHA-256 được dùng để tạo các khóa mã hóa trong hệ thống Bitcoin.
 - **Băm dữ liệu giao dịch:** Mỗi giao dịch được mã hóa thành một giá trị băm SHA-256 để bảo mật.
 - **Proof of Work:** Các thợ đào phải tìm một giá trị đầu vào sao cho giá trị băm của nó nhỏ hơn ngưỡng nhất định, quá trình này được gọi là "Proof of Work".
 - **Bảo vệ sự toàn vẹn:** Hàm băm đảm bảo dữ liệu giao dịch không bị thay đổi sau khi được xác nhận trong blockchain.

3.1 Tổng quan về thuật toán băm SHA-256

```
static const uint32_t K[64] = {
    0x428a2f98, 0x71374491, 0xb5c0fbcf, 0xe9b5dba5,
    0x3956c25b, 0x59f111f1, 0x923f82a4, 0xab1c5ed5,
    0xd807aa98, 0x12835b01, 0x243185be, 0x550c7dc3,
    0x72be5d74, 0x80deb1fe, 0x9bdc06a7, 0xc19bf174,
    0xe49b69c1, 0xefbe4786, 0xfc19dc6, 0x240ca1cc,
    0x2de92c6f, 0x4a7484aa, 0x5cb0a9dc, 0x76f988da,
    0x983e5152, 0xa831c66d, 0xb00327c8, 0xbf597fc7,
    0xc6e00bf3, 0xd5a79147, 0x06ca6351, 0x14292967,
    0x27b70a85, 0x2e1b2138, 0x4d2c6dfc, 0x53380d13,
    0x650a7354, 0x766a0abb, 0x81c2c92e, 0x92722c85,
    0xa2bfe8a1, 0xa81a664b, 0xc24b8b70, 0xc76c51a3,
    0xd192e819, 0xd6990624, 0xf40e3585, 0x106aa070,
    0x19a4c116, 0x1e376c08, 0x2748774c, 0x34b0bcb5,
    0x391c0cb3, 0x4ed8aa4a, 0x5b9cca4f, 0x682e6ff3,
    0x748f82ee, 0x78a5636f, 0x84c87814, 0x8cc70208,
    0x90befffa, 0xa4506ceb, 0xbef9a3f7, 0xc67178f2
};
```

3.1 Tổng quan về thuật toán băm SHA-256

Giải thuật 2. Giải thuật SHA256

SHA256: digest = SHA256(message in)

(M, N) = Padding (message in);

$H^{(0)}$ = IV Constants;

for ($t = 0$; $t < N$; $t++$) {

 W = BlockDecomposition($M^{(t)}$);

$H^{(t+1)}$ = HashComputation($H^{(t)}$, K Constants, W);

}

return digest = $H1^{(N)} \parallel H2^{(N)} \parallel H3^{(N)} \parallel H4^{(N)} \parallel H5^{(N)} \parallel H6^{(N)} \parallel H7^{(N)} \parallel H8^{(N)}$;

3.2 Khóa riêng tư (Private key)

- **Định nghĩa:** Khóa riêng tư là một số ngẫu nhiên, được sử dụng để kiểm soát và sở hữu Bitcoin liên quan đến một địa chỉ Bitcoin.
- **Vai trò:**
 - Dùng để tạo chữ ký số, chứng minh quyền sở hữu Bitcoin.
 - Người dùng cần giữ bí mật, nếu để lộ ra đồng nghĩa với việc mất quyền kiểm soát Bitcoin.
 - Không thể khôi phục nếu bị mất.

3.2 Khóa riêng tư (Private key)

- **Định nghĩa:** Khóa riêng tư là một số ngẫu nhiên, được sử dụng để kiểm soát và sở hữu Bitcoin liên quan đến một địa chỉ Bitcoin.
- **Vai trò:**
 - Dùng để tạo chữ ký số, chứng minh quyền sở hữu Bitcoin.
 - Người dùng cần giữ bí mật, nếu để lộ ra đồng nghĩa với việc mất quyền kiểm soát Bitcoin.
 - Không thể khôi phục nếu bị mất.

3.2 Khóa riêng tư (Private key)

Cách tạo khóa riêng tư:

- **Ngẫu nhiên hóa:** Khóa riêng tư được tạo từ 256-bit (0-9, A-F) ngẫu nhiên, thường lấy từ máy tính hoặc hệ điều hành.
Dùng SHA256 để tạo ra số 256-bit từ nguồn ngẫu nhiên.
- **Kích thước không gian khóa:** 2^{256} ($\approx 10^{77}$), lớn hơn số nguyên tử trong vũ trụ (10^{80}).
- **Lưu ý:** Cần sử dụng bộ tạo số ngẫu nhiên an toàn mật mã (CSPRNG), không nên tự viết mã tạo số ngẫu nhiên.

Ví dụ về khóa riêng tư:

6ddbc3b81729cc2a53ec5a468337d4b487df9fcff890567fd89258fa5d365f01

Nhận xét: Khóa này quá lớn để dự đoán hay tấn công Brute-force.

3.3 Tổng quan về Elliptic Curve Cryptography (ECC)

- **Mô tả:** ECC là một loại mã hóa khóa công khai, sử dụng bài toán logarithm rắc rối trên các điểm của đường cong elliptic.
- **Đường cong secp256:**
 - Hàm elliptic curve được Bitcoin sử dụng là:

$$y^2 \equiv x^3 + 7 \pmod{p}$$

$p = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$ là số nguyên tố lớn làm cho không gian điểm trên đường cong trở nên vô cùng rộng và khó đoán.

(x, y) là tọa độ của một điểm trên đường cong, và phép toán trong ECC dựa trên các phép cộng, nhân các điểm này.

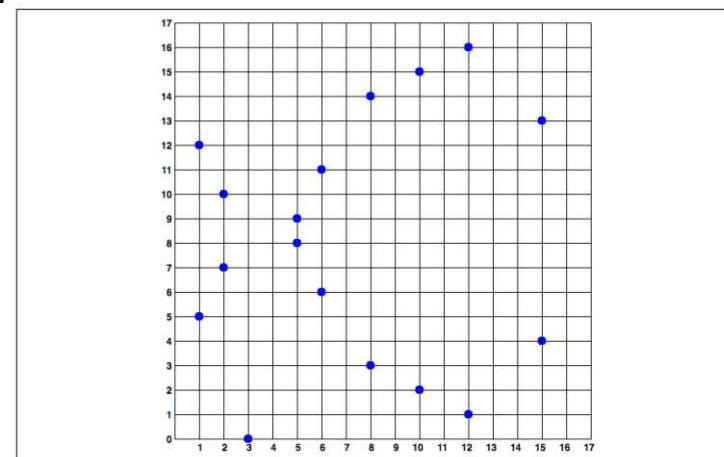
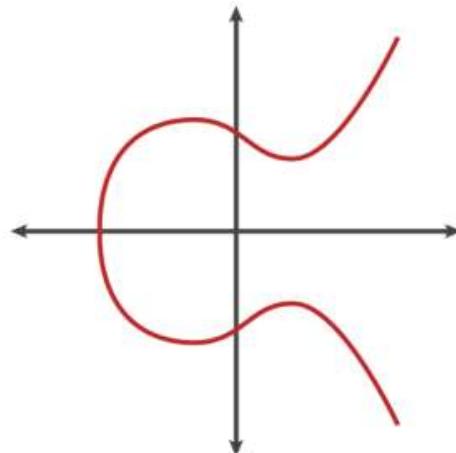


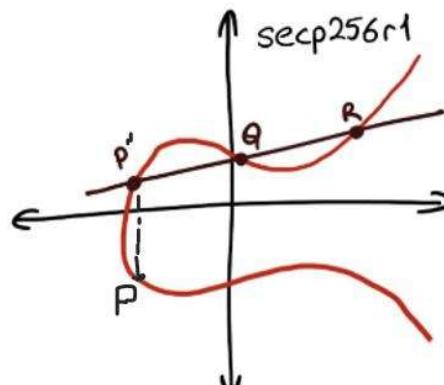
Figure 4-3. Elliptic curve cryptography: visualizing an elliptic curve over $F(p)$, with $p=17$

3.3 Tổng quan về Elliptic Curve Cryptography (ECC)

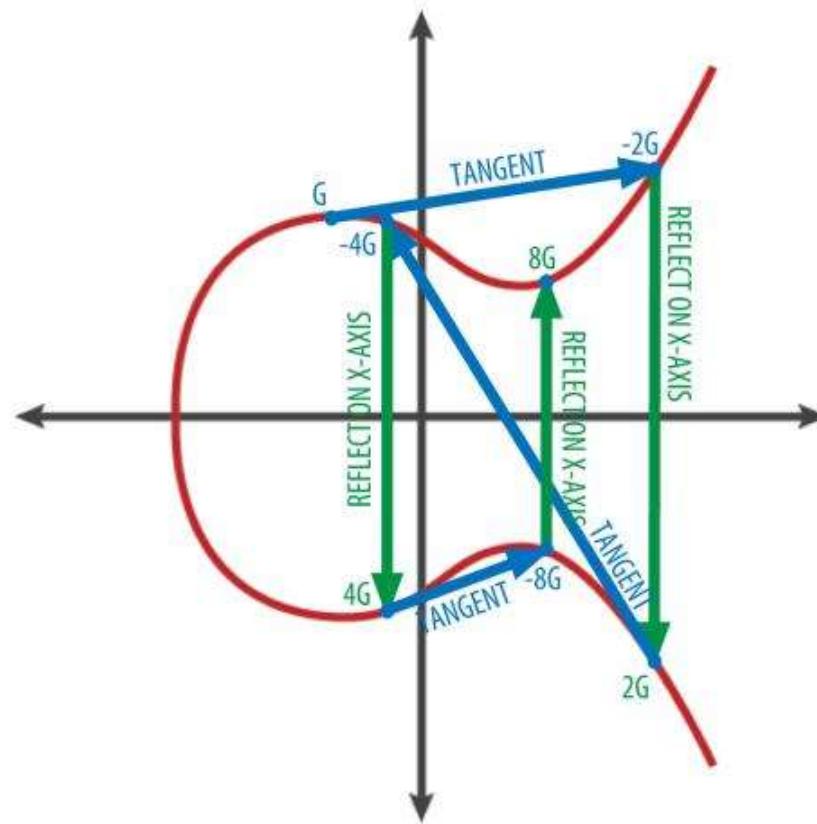
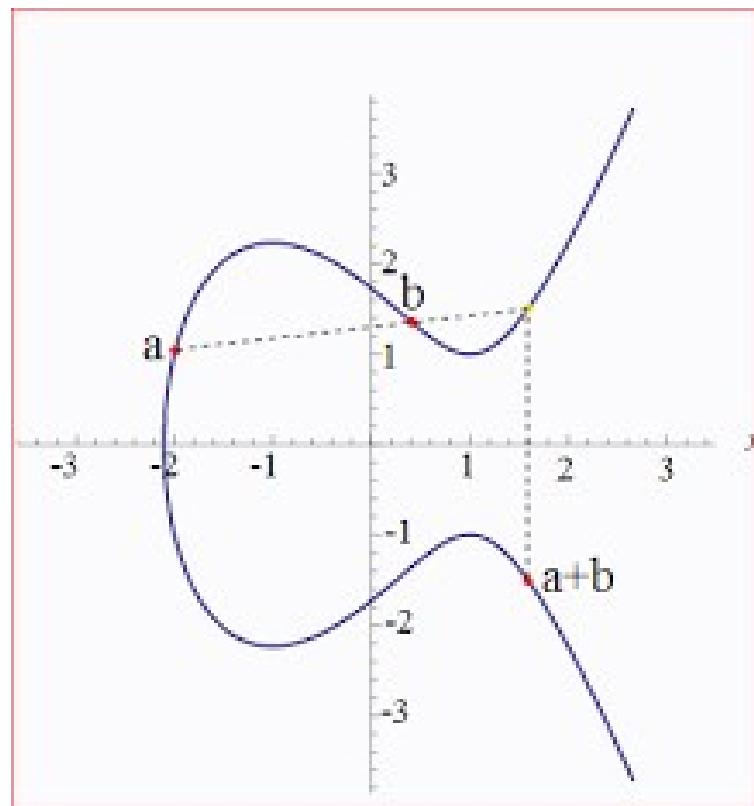
- Một số phép toán trên đường cong ECC:
- Cộng điểm:
 - Với hai điểm P_1 và P_2 trên đường cong, kẻ đường thẳng P_1P_2 đường thẳng này cắt đường cong tại một điểm P'_3 .

Chú ý: trường hợp một điểm cộng với chính nó thì ta sẽ lấy tiếp tuyến của đường cong tại điểm đó cắt đường cong tại điểm tiếp theo chính là P'_3 .

- Phản chiếu P'_3 qua trực hoành ta thu được $P_3 = P_1 + P_2$.
- Nhân điểm:
- Nhân một điểm P với một số nguyên k là cộng P với chính nó k lần.



3.3 Tổng quan về Elliptic Curve Cryptography (ECC)



3.4 Khóa công khai (Public key)

- **Định nghĩa:** Khóa công khai là một điểm trên đường cong elliptic được tính từ khóa riêng tư, dùng để tạo địa chỉ Bitcoin.
- **Vai trò:**
 - Được dùng để mã hóa dữ liệu và tạo địa chỉ Bitcoin.
 - Đóng vai trò xác minh chữ ký số, chứng minh giao dịch Bitcoin hợp lệ.
 - Có thể chia sẻ công khai mà không ảnh hưởng đến bảo mật của khóa riêng tư.
- **Tính chất:**
 - **Một chiều:** Dễ dàng tính toán Public Key từ Private Key, nhưng không thể quay ngược lại.
 - **Chống lại việc giải mã:** Để tìm Private Key từ Public Key, phải giải bài toán logarithm rời rạc, rất khó khăn (brute-force).

3.4 Khóa công khai (Public key)

- Cách tạo ra khóa công khai:

- Công thức:

$$K = k \times G$$

- Trong đó: +) K là khóa công khai.

- +) k là khóa riêng tư – một số ngẫu nhiên lớn.

- +) G là điểm sinh – điểm cố định trên đường cong

- Elliptic được chọn trước. (**secp256**)

Đặc điểm: Từ khóa công khai K , không thể tính ngược lại để tìm khóa riêng tư kkk do tính chất bảo mật của elliptic curve.

Kết luận:

- **Bảo mật:** Public Key có thể chia sẻ công khai mà không bị lộ Private Key.
- **Ứng dụng:** Cơ sở cho chữ ký điện tử và xác thực trong các giao dịch Bitcoin.

3.5 Tổng quan về mã hóa Base58

- **Base58 Encoding** là một phương pháp mã hóa số nhị phân (binary data) thành chuỗi ký tự ngắn gọn và dễ đọc hơn, phát triển đặc biệt cho Bitcoin và nhiều tiền điện tử khác.
- **Lý do sử dụng Base58:**
 - Giảm độ dài của chuỗi so với mã hóa thập phân và hex.
 - Giúp dễ đọc, giảm nhầm lẫn khi nhập tay.
 - Bỏ qua những ký tự dễ nhầm lẫn như 0 (số không), O (chữ O hoa), I (chữ I hoa), và l (chữ L thường).
- **Bộ ký tự của Base58:**
 - Sử dụng các chữ cái thường, chữ hoa và số, nhưng không bao gồm ký tự dễ nhầm lẫn.

VD:

```
Incoming Hexadecimal Number: 3b46511ae3b047b6ad2c7f25d0a995e1  
Base 58: 8KXmFY9SQtjKwqdiiMmrgg
```

3.5 Tổng quan về mã hóa Base58

- Quá trình mã hóa Base58:

1. Chuyển đổi từ số nguyên lớn: Đầu tiên, số nhị phân lớn (như địa chỉ hoặc khóa Bitcoin) được chuyển thành một số nguyên lớn.
2. Chia cho 58: Số nguyên này được chia cho 58, lưu lại phần dư cho mỗi lần chia.
3. Ánh xạ phần dư: Mỗi phần dư sẽ ánh xạ thành một ký tự trong bộ ký tự Base58.
4. Ghép nối các ký tự: Các ký tự kết quả được ghép lại theo thứ tự ngược lại để tạo thành chuỗi mã hóa Base58.

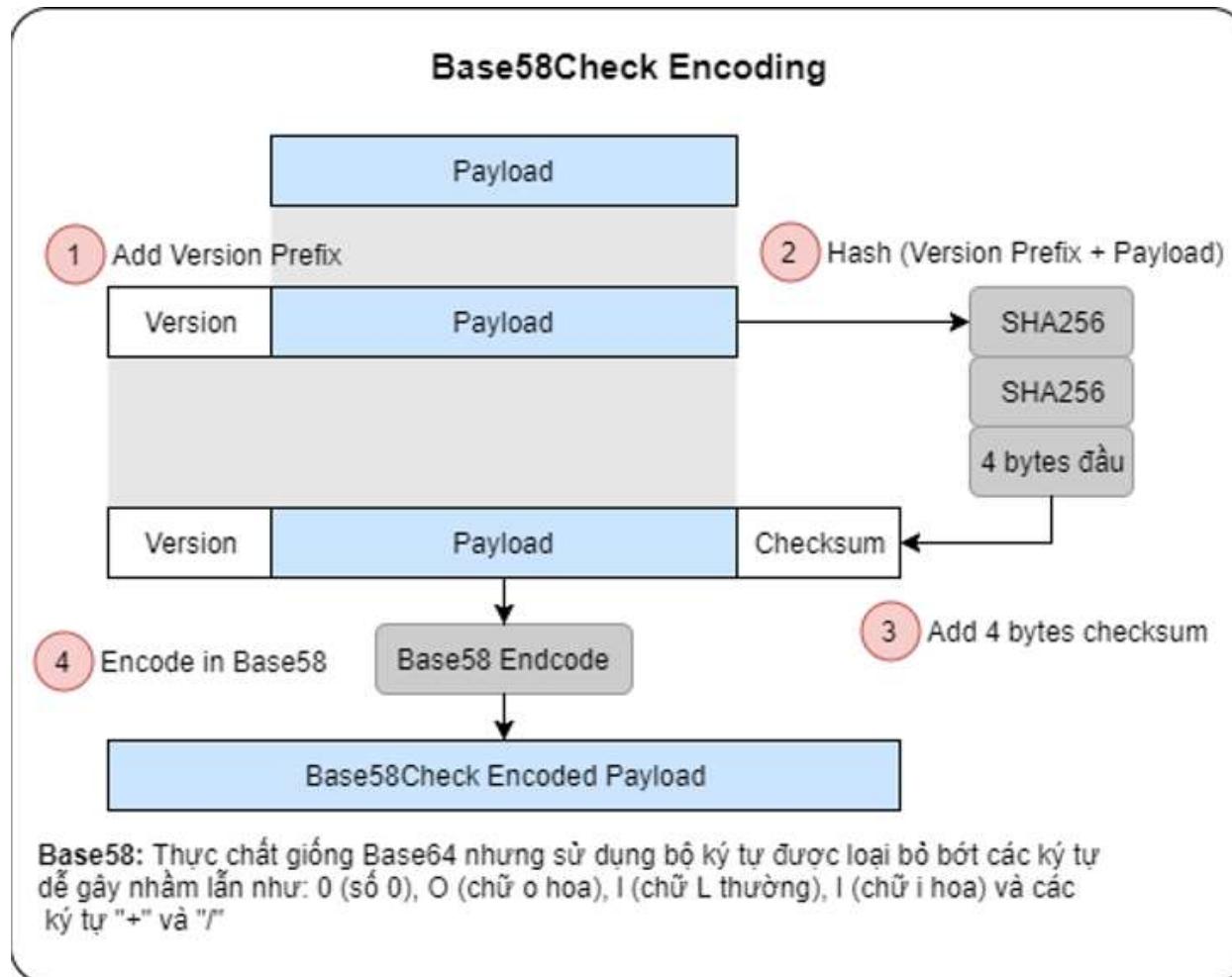
```
# Tạo địa chỉ Bitcoin từ khóa công khai
print("Bitcoin Address (b58check) is:", bitcoin.pubkey_to_address(public_key))
# Chuyển khóa công khai thành địa chỉ Bitcoin thông qua mã hóa Base58.
```

3.6 Tổng quan về Base58Check

- **Base58Check** là phiên bản mở rộng của Base58 với tính năng kiểm tra lỗi. Được sử dụng trong địa chỉ Bitcoin, khóa riêng tư (WIF), giúp dễ đọc và tránh lỗi nhập liệu.
- **Quy trình Base58Check:**
 - 1) Thêm version byte vào đầu dữ liệu (ví dụ: 0x00 cho địa chỉ Bitcoin).
 - 2) Tạo checksum bằng cách băm hai lần SHA-256, lấy 4 byte đầu của kết quả.
 - 3) Gắn checksum vào cuối dữ liệu.
 - 4) Mã hóa toàn bộ chuỗi bằng Base58.
- **Ví dụ:** Địa chỉ Bitcoin bắt đầu bằng "1" là mã hóa Base58Check.

```
Compressed Public Key (hex) is: 03ffe88374e0383b1598549f47ff23a55ae39f1331eca8babd1e0d00a77f867d57  
Bitcoin Address (b58check) is: 16k7Kxkkjy3WUjSATLcePzEdz8Ar1fs6sv
```

3.6 Tổng quan về Base58Check



3.6 Địa chỉ Bitcoin (Bitcoin Address)

- **Địa chỉ Bitcoin** là một chuỗi ký tự gồm các chữ số và ký tự, có thể được chia sẻ với bất kỳ ai muốn gửi tiền cho bạn.
- **Vai trò:**
 - Địa chỉ Bitcoin thường xuất hiện trong giao dịch như là "người nhận" tiền, tương tự như tên người thụ hưởng trên một tờ séc giấy.
 - Mối quan hệ giữa địa Chỉ Bitcoin và Khóa Công Khai
 - Địa chỉ Bitcoin: Không phải là khóa công khai, nhưng được dẫn xuất từ khóa công khai.
 - Quá trình tạo địa chỉ: Sử dụng các hàm băm một chiều (hash) từ khóa công khai.

VD:



3.7 Cơ chế băm trong tạo địa chỉ Bitcoin

- **Băm khóa công khai:**

- 1) Đầu tiên, tính hàm băm *SHA256* của khóa công khai.
- 2) Sau đó, tính hàm băm *RIPEMD160* của kết quả trên.
- 3) Kết quả cuối cùng là một chuỗi 160-bit (20-byte):

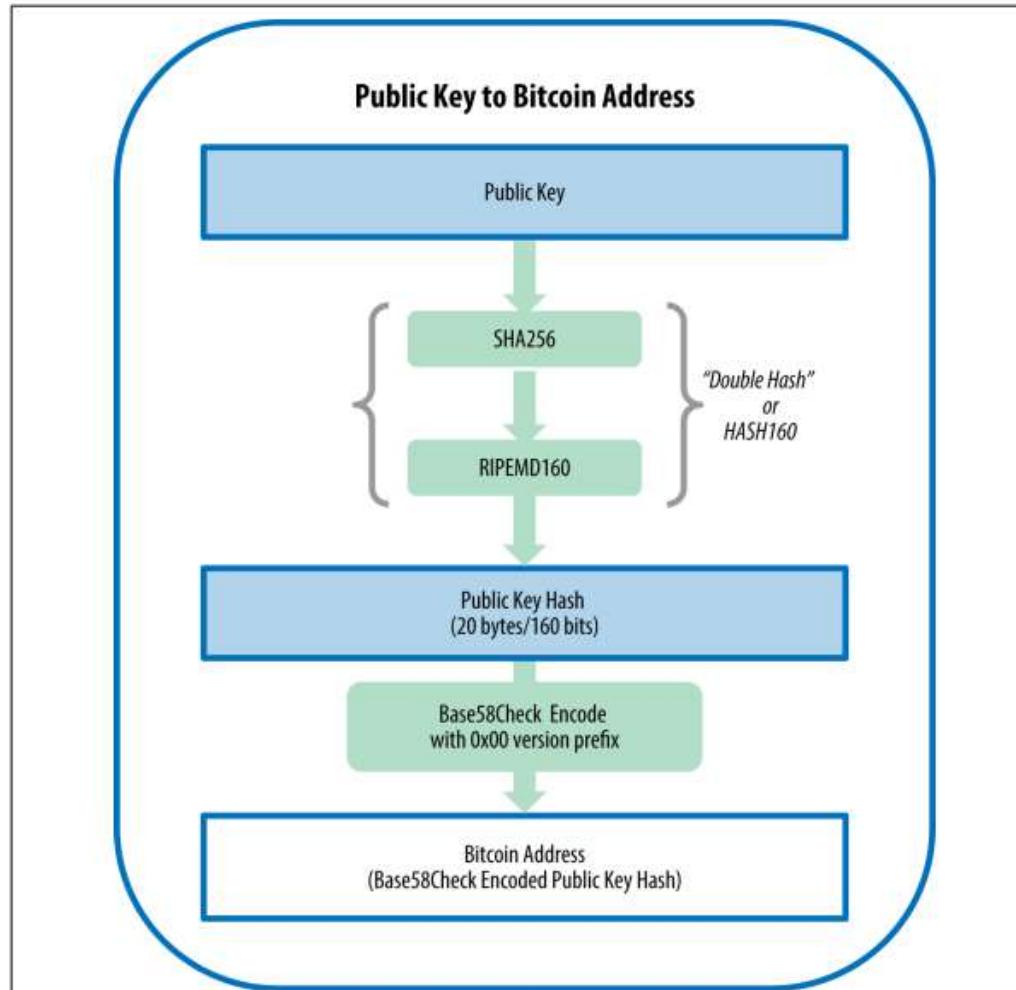
$$A = \text{RIPEMD160}(\text{SHA256}(K))$$

Trong đó, K : khóa công khai

A : Địa chỉ Bitcoin.

P/s: **RIPEMD-160** là hàm băm tạo ra chuỗi 160 bit (40 kí tự), thường dùng để rút ngắn khóa công khai trong việc tạo địa chỉ Bitcoin.

3.8 Mô hình tạo địa chỉ Bitcoin (Bitcoin Address)



3.9 Chương trình tạo Key, Adress bằng Python

```
import bitcoin # Thư viện 'bitcoin' được import để sử dụng các hàm và công cụ liên quan đến Bitcoin.

# Tạo khóa riêng tư ngẫu nhiên
valid_private_key = False # Khởi tạo biến để kiểm tra tính hợp lệ của khóa riêng tư.
✓ while not valid_private_key: # Vòng lặp kiểm tra tính hợp lệ của khóa riêng tư.
    private_key = bitcoin.random_key() # Tạo khóa riêng tư ngẫu nhiên dưới dạng chuỗi hexa.
    decoded_private_key = bitcoin.decode_privkey(private_key, 'hex') # Giải mã khóa riêng tư từ định dạng hex thành số thập phân.
    valid_private_key = 0 < decoded_private_key < bitcoin.N
    # Kiểm tra xem khóa riêng tư có hợp lệ không (phải nằm trong khoảng từ 1 đến N của Bitcoin).

    # In khóa riêng tư dưới dạng hex và decimal (hệ 10 và hệ 16)
    print("Private Key (hex) is: ", private_key) # In khóa riêng tư ở định dạng hex (mã hexa gồm 64 kí (gồm 32 byte, 1 byte = 8 bit))
    print("Private Key (decimal) is: ", decoded_private_key) # In khóa riêng tư ở dạng số thập phân.

    # Chuyển đổi khóa riêng tư sang định dạng WIF (Wallet Import Format)
    wif_encoded_private_key = bitcoin.encode_privkey(decoded_private_key, 'wif')
    # Mã hóa khóa riêng tư ở định dạng WIF để dễ dàng lưu trữ và sử dụng trong ví Bitcoin.
    print("Private Key (WIF) is: ", wif_encoded_private_key) # In khóa riêng tư dưới định dạng WIF.

    # Thêm hậu tố "01" để chỉ thị khóa riêng tư nén
    compressed_private_key = private_key + '01'
    # Khóa riêng tư nén được tạo bằng cách thêm chuỗi "01" vào cuối khóa riêng tư ban đầu.
    print("Private Key Compressed (hex) is: ", compressed_private_key) # In khóa riêng tư nén dưới dạng hex.

    # Tạo định dạng WIF từ khóa riêng tư nén (WIF-compressed)
    ✓ wif_compressed_private_key = bitcoin.encode_privkey(
        bitcoin.decode_privkey(compressed_private_key, 'hex'), 'wif')
    # Chuyển đổi khóa riêng tư nén sang định dạng WIF nén.
    print("Private Key (WIF-Compressed) is: ", wif_compressed_private_key) # In khóa riêng tư nén dưới định dạng WIF nén.
```

3.9 Chương trình tạo Key, Adress bằng Python

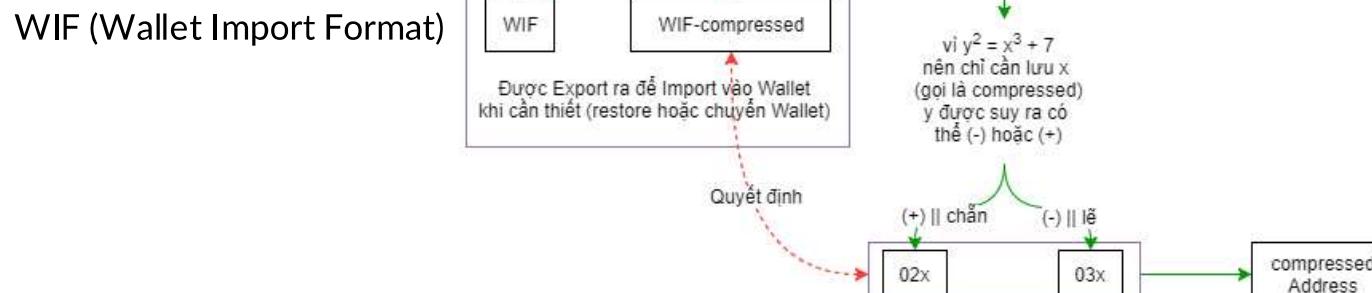
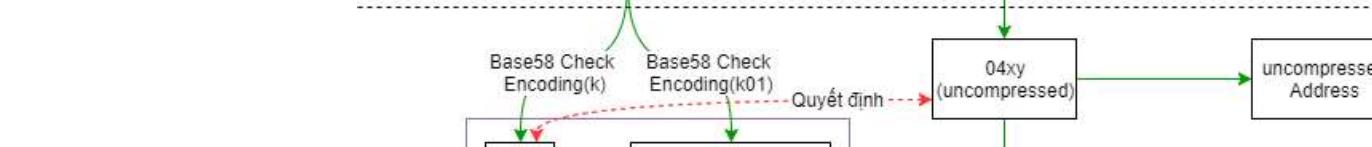
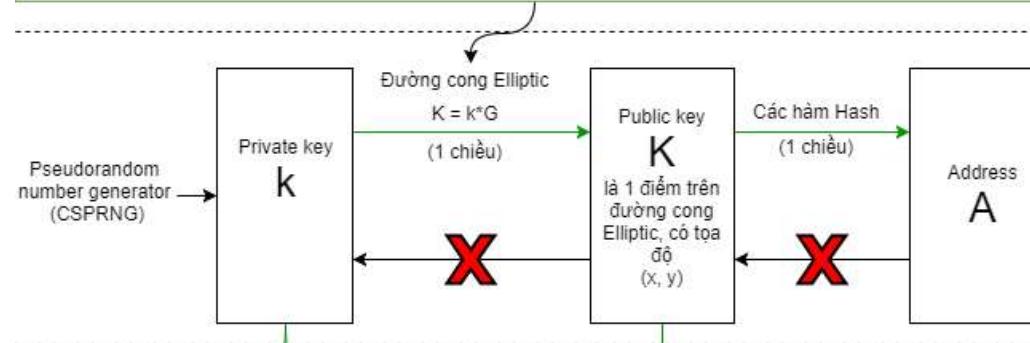
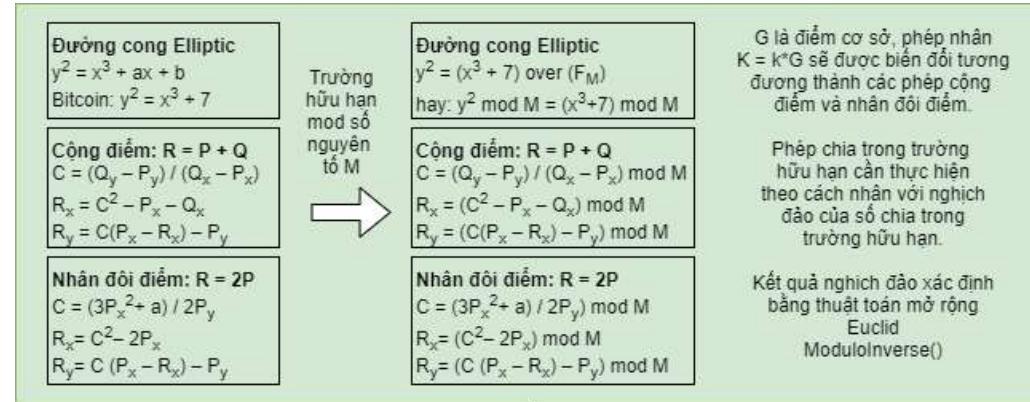
```
#Khóa công khai trong Bitcoin.  
# Nhân điểm sinh G của đường cong Elliptic (EC) với khóa riêng tư để tạo ra điểm khóa công khai  
public_key = bitcoin.fast_multiply(bitcoin.G, decoded_private_key)  
# Khóa công khai được tạo ra từ phép nhân giữa khóa riêng tư và điểm sinh của đường cong Elliptic.  
print("Public Key (x,y) coordinates is:", public_key) # In tọa độ x, y của khóa công khai.  
  
# Mã hóa khóa công khai dưới dạng hex, thêm tiền tố 04  
hex_encoded_public_key = bitcoin.encode_pubkey(public_key, 'hex')  
# Mã hóa khóa công khai dưới định dạng hex với tiền tố "04".  
print("Public Key (hex) is:", hex_encoded_public_key) # In khóa công khai dưới dạng hex.  
  
# Nén khóa công khai, thêm tiền tố phụ thuộc vào việc y là số chẵn hay lẻ  
(public_key_x, public_key_y) = public_key # Tách khóa công khai thành hai phần x và y.  
if (public_key_y % 2) == 0: # Nếu y là số chẵn, sử dụng tiền tố "02".  
    compressed_prefix = '02'  
else: # Nếu y là số lẻ, sử dụng tiền tố "03".  
    compressed_prefix = '03'  
  
hex_compressed_public_key = compressed_prefix + bitcoin.encode(public_key_x, 16)  
# Mã hóa khóa công khai nén dưới định dạng hex, thêm tiền tố tương ứng.  
print("Compressed Public Key (hex) is:", hex_compressed_public_key) # In khóa công khai nén dưới dạng hex.  
  
# Tạo địa chỉ Bitcoin từ khóa công khai  
print("Bitcoin Address (b58check) is:", bitcoin.pubkey_to_address(public_key))  
# Chuyển khóa công khai thành địa chỉ Bitcoin thông qua mã hóa Base58.  
  
# Tạo địa chỉ Bitcoin nén từ khóa công khai nén  
print("Compressed Bitcoin Address (b58check) is:", bitcoin.pubkey_to_address(hex_compressed_public_key))  
# Tạo địa chỉ Bitcoin từ khóa công khai nén và in ra địa chỉ đã nén.
```

3.9 Chương trình tạo Key, Adress bằng Python

- Output:

```
PS D:\University\DCS Lab\Coding> python -u "d:\University\DCS Lab\Coding\keyandadd_gen.py"
Private Key (hex) is: 6ddbc3b81729cc2a53ec5a468337d4b487df9fcff890567fd89258fa5d365f01
Private Key (decimal) is: 49690390806369690627673799149307657007334230506982544031124928999402178502401
Private Key (WIF) is: 5JefmWpU7KNsPjsx6Uq2vH6kRjewFp9ajX7swnKyFirW5gJvZE
Private Key Compressed (hex) is: 6ddbc3b81729cc2a53ec5a468337d4b487df9fcff890567fd89258fa5d365f0101
Private Key (WIF-Compressed) is: KzuG7UcWet7XRNFtBBq1UA4NgBGSUP6stfwMro4jHyjVK13s3nTk
Public Key (x,y) coordinates is: (115750592187499842495449544782693384344212111232307056464642916504053846801751, 8854199655891509366949099866330094480678254235023
5682100925407684650302456057)
Public Key (hex) is: 04ffe88374e0383b1598549f47ff23a55ae39f1331eca8babd1e0d00a77f867d57c3c0fe7e4e025e73c22d26669f237ea7386bb51752b633c0565593bdb67c40f9
Compressed Public Key (hex) is: 03ffe88374e0383b1598549f47ff23a55ae39f1331eca8babd1e0d00a77f867d57
Bitcoin Address (b58check) is: 16k7Kxkkjy3WUjSATLcePzEdz8Ar1fs6sv
Compressed Bitcoin Address (b58check) is: 1CvSKmQBwb8EPLKPdrGmcVnvd83pr4PVgf
```

3.10 Tổng kết





ĐẠI HỌC
BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY
OF SCIENCE AND TECHNOLOGY

Chương 4: Ví Bitcoin (Wallet)

ONE LOVE. ONE FUTURE.

4.1 Tổng quan về công nghệ ví Bitcoin

- Khái niệm cơ bản về ví Bitcoin:
 - Ví Bitcoin không chứa bitcoin mà chỉ chứa các khóa: Khóa riêng tư và khóa công khai.
- Công dụng: Người dùng kiểm soát Bitcoin trên mạng **Blockchain** bằng cách ký các giao dịch với khóa trong ví.
- Các “đồng Bitcoin” được lưu trữ trong **Blockchain** dưới dạng các đầu ra giao dịch (txt.out).

4.2 Phân loại công nghệ ví Bitcoin

- **Ví không định danh (Nondeterministic Wallet):**
 - + Các khóa được tạo ngẫu nhiên và độc lập với nhau.
 - + Yêu cầu sao lưu thường xuyên và khó quản lý, dễ dẫn đến mất khóa.

Lưu ý: Ví kiểu này ít được khuyến khích do tính phức tạp trong quản lý.

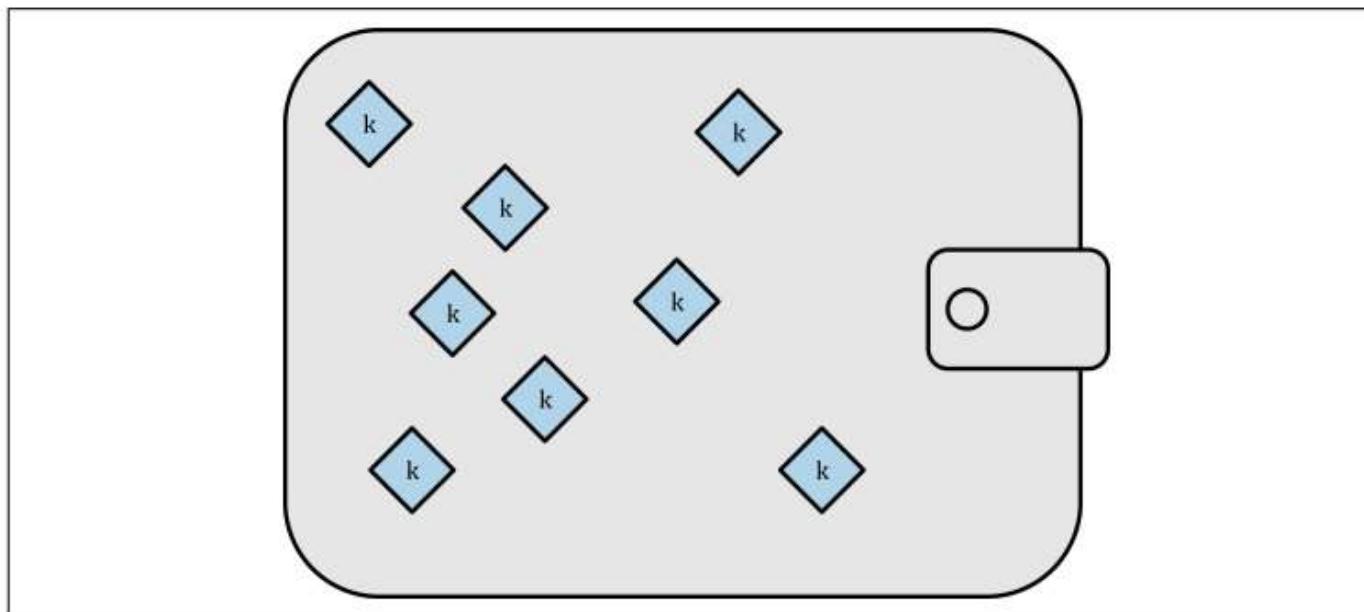


Figure 5-1. Type-0 nondeterministic (random) wallet: a collection of randomly generated keys

4.2 Phân loại công nghệ ví Bitcoin

- **Ví định danh (Deterministic Wallet):**
 - + Tất cả các khóa được sinh ra từ một khóa gốc chính duy nhất, gọi là “seed”.
 - + Chỉ cần sao lưu 1 lần và có thể khôi phục tất cả các khóa từ seed.

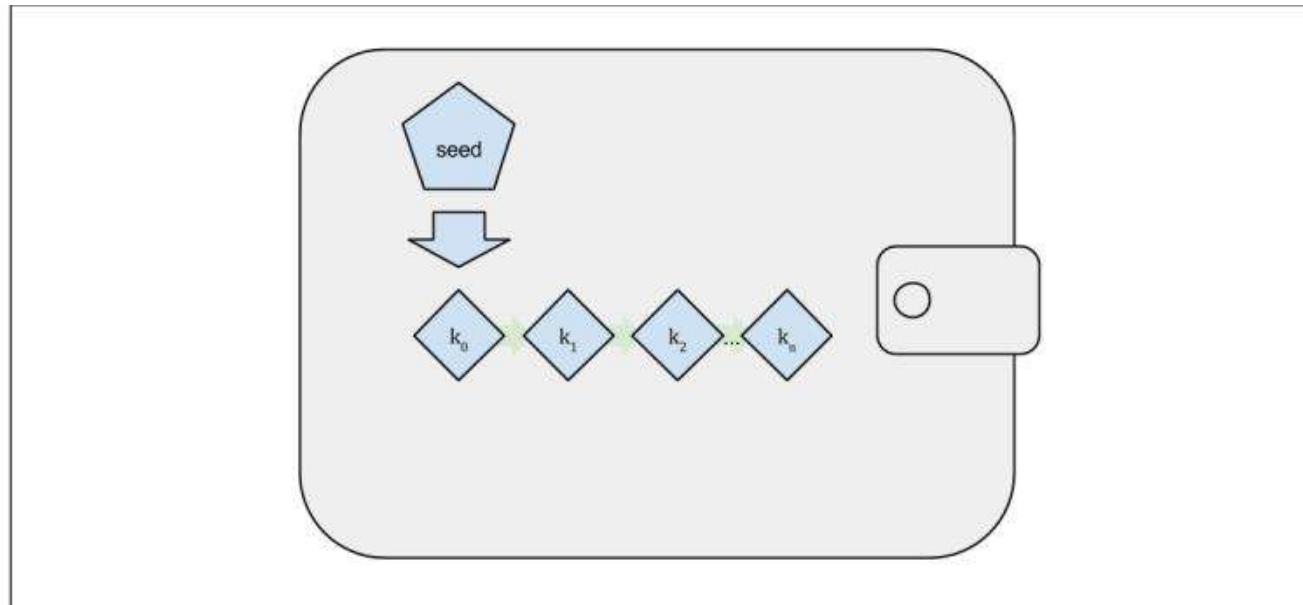


Figure 5-2. Type-1 deterministic (seeded) wallet: a deterministic sequence of keys derived from a seed

4.2 Phân loại công nghệ ví Bitcoin

- **Ví phân cấp (HD Wallet – Hierarchical Deterministic)**
 - Cấu trúc cây: Ví xác định cấp bậc (HD wallet) sinh ra các khóa dưới dạng cây, với các nhánh và khóa con. (Mỗi khóa con có thể tạo một địa chỉ nhận tiền khác nhau).
 - Ưu điểm: Có thể tổ chức khóa theo nhánh, phù hợp với nhu cầu tổ chức và quản lý lớn(các khóa con có chức năng khác nhau: nhận thanh toán, hoàn tiền,...)
 - BIP-32/BIP-44: Ví tiêu chuẩn HD được xác định bởi các BIP (Bitcoin Improvement Proposals) để hỗ trợ đa chức năng và bảo mật.

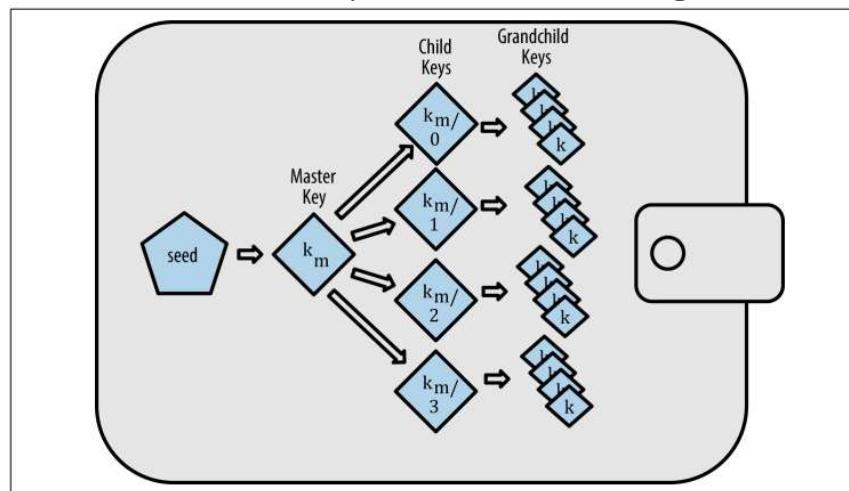


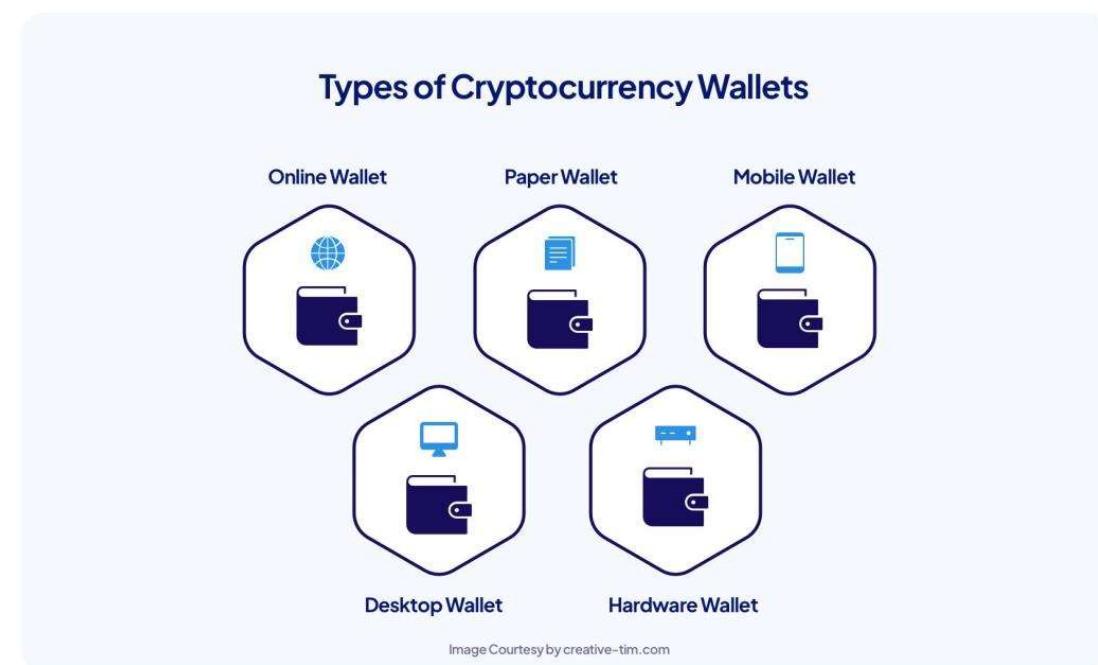
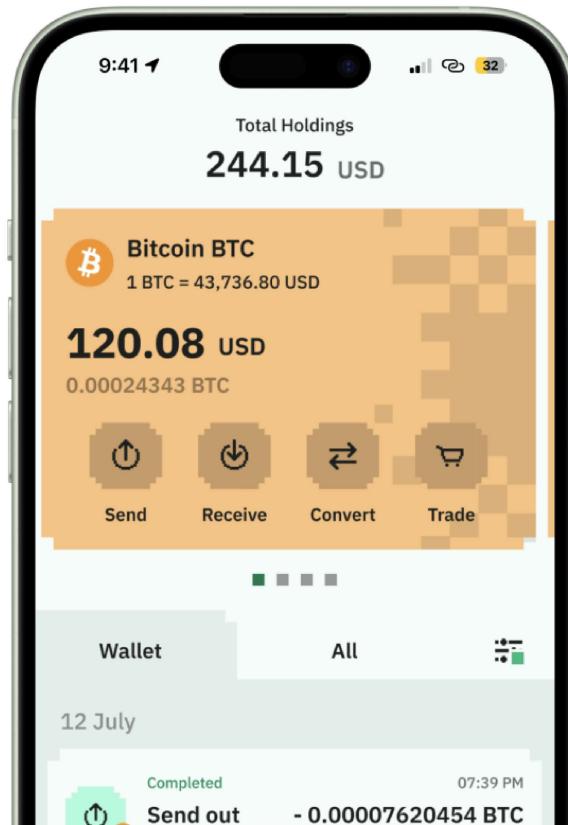
Figure 5-3. Type-2 HD wallet: a tree of keys generated from a single seed

4.3 Seed và Mnemonic trong Ví Bitcoin

- **Seed (Khóa Gốc):** một chuỗi số ngẫu nhiên dùng để tạo ra tất cả các khóa trong ví.
 - Chỉ cần có seed, người dùng có thể khôi phục toàn bộ ví và các khóa liên quan.
 - Seed là yếu tố quan trọng giúp đơn giản hóa việc sao lưu và quản lý ví.
- **Mnemonic Code (BIP-39):** là chuỗi các từ tiếng Anh đại diện cho seed của ví, giúp dễ dàng ghi nhớ và sao lưu.
 - Chuỗi mnemonic có thể có 12, 18, hoặc 24 từ, đảm bảo tính an toàn và tiện lợi.
 - Được sử dụng rộng rãi trong các ví hiện đại để khôi phục và bảo mật tài sản tiền điện tử.
- **Các Tiêu Chuẩn Ví Liên Quan:**
 - BIP-32: Quy định cấu trúc cây khóa HD, giúp tạo ra các nhánh khóa con từ khóa gốc.
 - BIP-44: Định dạng ví HD đa tài khoản và đa tiền tệ, phù hợp với nhu cầu tổ chức ví cho nhiều loại tiền điện tử khác nhau.

4.4 Kết luận

- Ví Bitcoin đa dạng về kiểu loại (có đặc điểm, chức năng riêng) đảm bảo tính an toàn và bảo mật của tài sản, giúp người dùng dễ dàng tương tác với mạng lưới Bitcoin mà không cần hiểu rõ các chi tiết kỹ thuật.





ĐẠI HỌC
BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY
OF SCIENCE AND TECHNOLOGY

Chương 5: Giao dịch (Transaction)

ONE LOVE. ONE FUTURE.

5.1 Giao dịch Bitcoin

- **Giao dịch (Transaction):** Là quá trình chuyển Bitcoin giữa các địa chỉ.
- **Vai trò:** Mỗi giao dịch xác thực việc sở hữu Bitcoin và ghi nhận trên blockchain.
- **Đặc điểm quan trọng:**
 - Minh bạch và công khai.
 - Không cần trung gian.

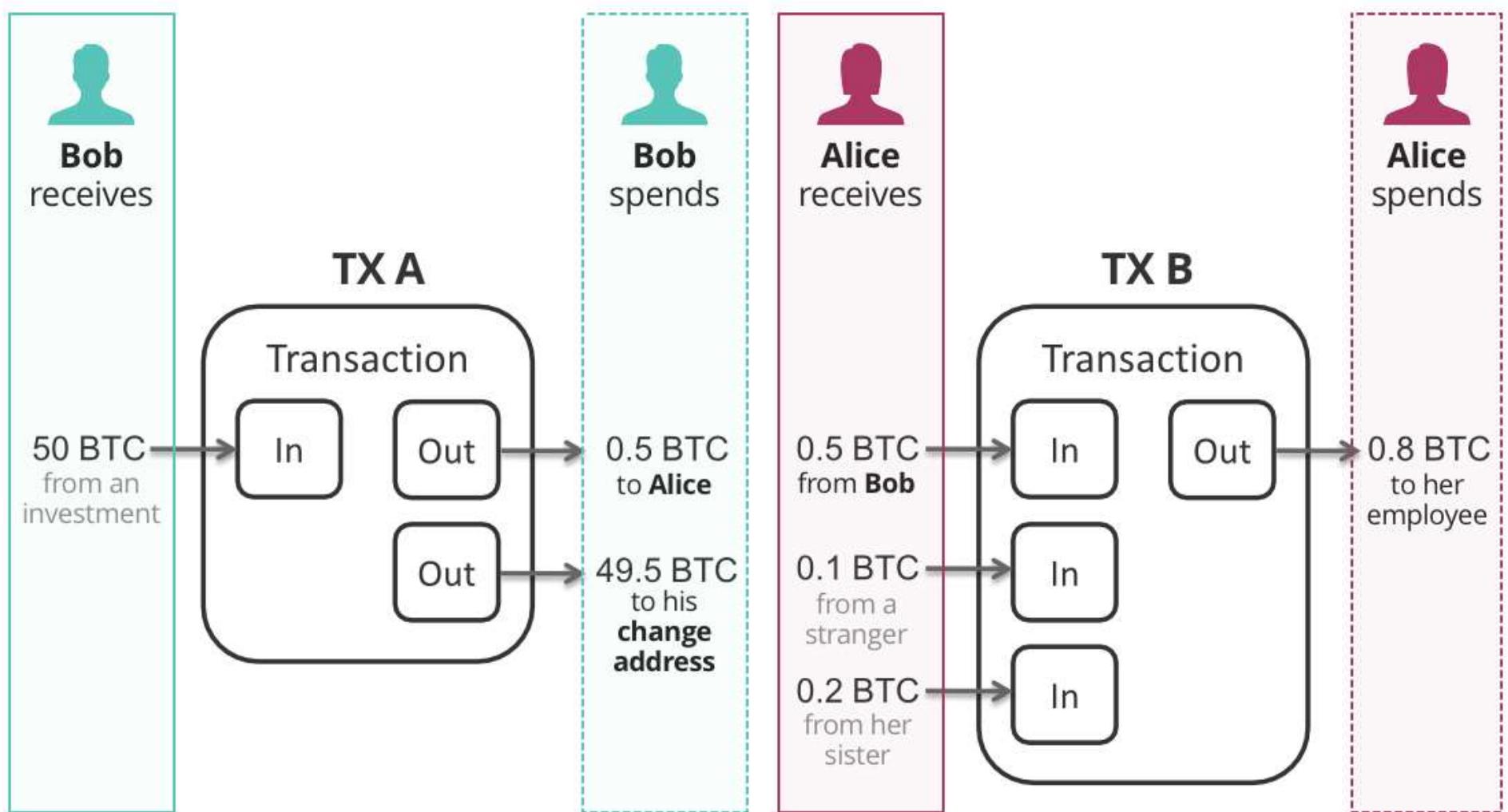
5.2 Cấu trúc và tầm quan trọng của giao dịch Bitcoin

- Thành phần chính của giao dịch:
 - **Đầu vào (Inputs):** Thông tin về nguồn Bitcoin (từ các giao dịch trước).
 - **Đầu ra (Outputs):** Thông tin về địa chỉ nhận và số lượng Bitcoin.
 - **Phí giao dịch (Transaction fee):** Nếu tổng giá trị đầu vào lớn hơn tổng đầu ra, một phí giao dịch nhỏ sẽ được trích từ lượng BTC của bạn dành cho thợ đào.
- Vai trò:
 - Cốt lõi của blockchain: Giao dịch là yếu tố xây dựng và duy trì sổ cái blockchain.
 - Bảo mật và minh bạch: Giao dịch giúp mạng lưới Bitcoin duy trì tính minh bạch và đảm bảo an ninh.
 - Không thể đảo ngược: Một khi đã xác nhận, giao dịch không thể bị thay đổi hoặc đảo ngược.

5.3 Đầu vào và đầu ra của giao dịch

- **Đầu vào của giao dịch:** chứa các Bitcoin chưa sử dụng từ các giao dịch trước đó.
 - Mỗi đầu vào bao gồm:
 - 1) ID giao dịch (Transaction ID): Liên kết với UTXO (Unspent Transaction Output – đại diện cho số tiền còn lại từ giao dịch trước).
 - 2) Chữ kí số (Signature): Chứng minh quyền sở hữu Bitcoin.
 - **Đầu ra của giao dịch:** Đầu ra xác định địa chỉ nhận và số lượng Bitcoin.
 - Mỗi đầu ra bao gồm:
 - 1) Địa chỉ nhận (Bitcoin Address): Địa chỉ nơi Bitcoin được gửi đến.
 - 2) Số lượng Bitcoin: Lượng Bitcoin cụ thể trong giao dịch.
- Nếu có dư thừa Bitcoin, nó sẽ được gửi lại vào một địa chỉ "thối" (change address) của người gửi.

5.3 Đầu vào và đầu ra của giao dịch



5.4 UTXO - Unspent Transaction Output

UTXO (Unspent Transaction Output) là các đầu ra từ giao dịch trước đó chưa được sử dụng.

Khi thực hiện giao dịch mới, UTXO được dùng làm đầu vào và sẽ được đánh dấu là đã chi tiêu.

- **Mỗi giao dịch mới** sẽ tạo ra UTXO mới dưới dạng đầu ra.
- **Giả sử bạn** có một UTXO trị giá **5 BTC**.

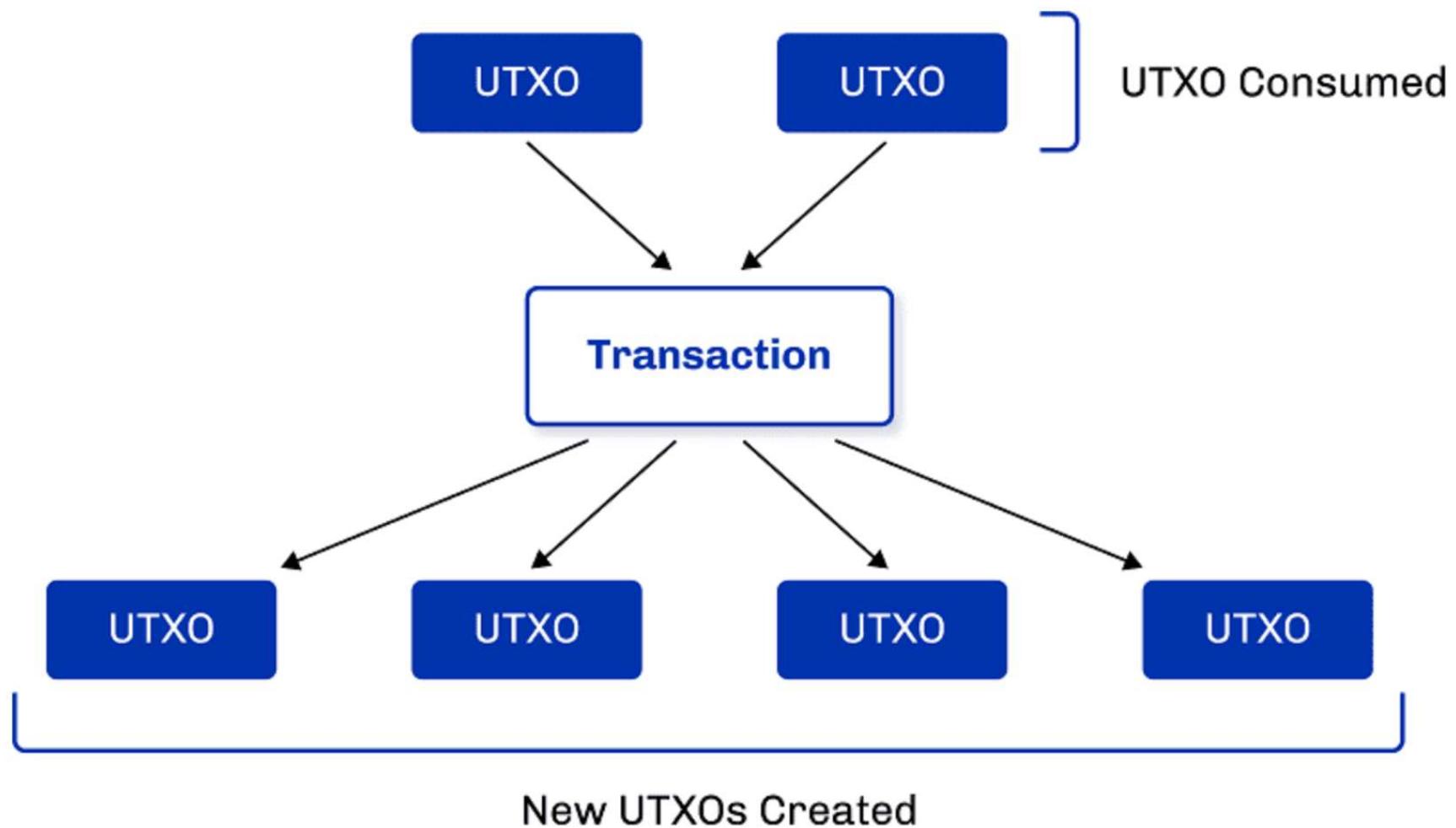
VD:

- Bạn muốn gửi **2 BTC** cho một người khác. Khi đó, bạn **không thể chỉ** sử dụng một phần của UTXO (2 BTC), mà phải **sử dụng toàn bộ 5 BTC** làm đầu vào. Trong phần **đầu ra**, bạn sẽ có:

- 2 BTC gửi cho người nhận (địa chỉ của người nhận).
- **3 BTC** còn lại sẽ được gửi trả lại **cho chính bạn**, nhưng thông qua một địa chỉ khác mà bạn sở hữu, gọi là **địa chỉ "thối"** (change address).

=> Qua đó, đảm bảo bạn không bị mất “phần dư” sau giao dịch và tăng tính bảo mật (do tạo địa chỉ mới).

5.4 UTXO - Unspent Transaction Output



5.5 Phí giao dịch (Transaction Fee)

- Phí giao dịch là **chênh lệch** giữa tổng giá trị **đầu vào** và **đầu ra**.

VD: Bạn có 2 BTC (đầu vào), muốn gửi 1.999 BTC (đầu ra), phần chênh lệch 0.001 BTC (dựa theo BitcoinCore) là phí giao dịch.

- Phí này khuyến khích thợ đào xử lý và nhận giao dịch trên blockchain để nhận thưởng.

5.7 Bitcoin Script và Cấu trúc

- **Bitcoin Script:** Ngôn ngữ lập trình đơn giản, kiểm tra điều kiện chi tiêu Bitcoin (không phải Turing hoàn thiện, tức là nó không có khả năng thực hiện mọi tính toán mà một ngôn ngữ Turing-complete có thể làm.).
- Mỗi giao dịch có 2 loại script:
- ScriptPubKey (Locking Script): Điều kiện khóa Bitcoin.
- ScriptSig (Unlocking Script): Giải khóa Bitcoin.
- Khi thực hiện giao dịch, ScriptSig và ScriptPubKey được kết hợp để kiểm tra tính hợp lệ.

5.8 P2PK và P2PKH - Các loại Giao dịch Cơ bản

- P2PK (Pay-to-Public-Key):

- Khóa Bitcoin bằng khóa công khai.

- ScriptPubKey:

<Public Key> OP_CHECKSIG. (check với chữ ký số)

- Kém bảo mật hơn P2PKH.

- P2PKH (Pay-to-Public-Key-Hash):

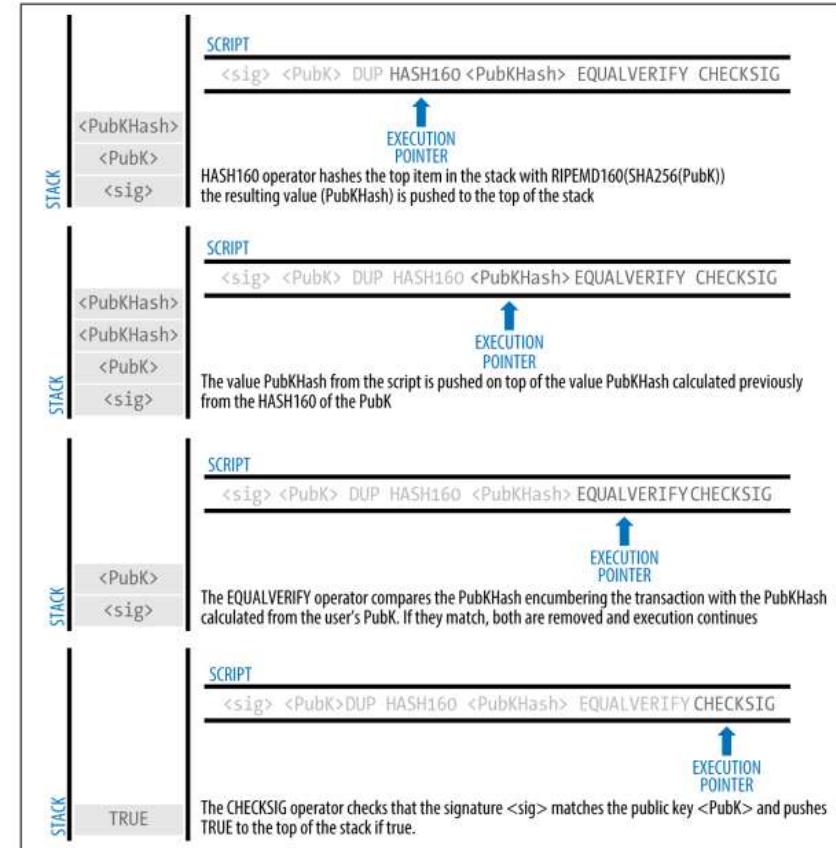
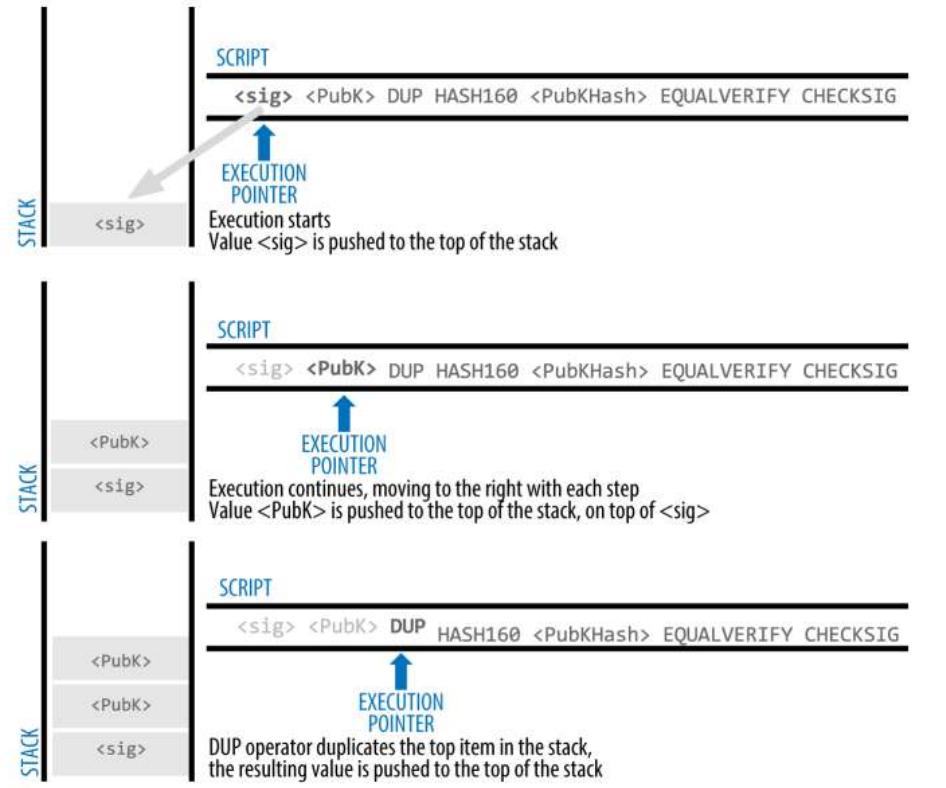
- Khóa bằng hash của khóa công khai, bảo mật hơn.

- ScriptPubKey:

OP_DUP OP_HASH160 <Public Key Hash> OP_EQUALVERIFY OP_CHECKSIG.

- P2PKH phổ biến hơn nhờ bảo mật cao.

5.8 P2PK và P2PKH - Các loại Giao dịch Cơ bản



5.9 Vai trò của Bitcoin Script và các loại giao dịch khác

- **Nội dung:** Bitcoin Script đảm bảo chỉ người sở hữu khóa riêng mới chi tiêu được Bitcoin.
- **Các loại giao dịch khác:**
 - **Multisig:** Yêu cầu nhiều chữ ký.
 - **P2SH (Pay-to-Script-Hash):** Khóa bằng script hash, yêu cầu cung cấp script khi chi tiêu.
 - **Timelock:** Giao dịch bị khóa đến một thời gian cụ thể.

5.10 Tổng quan về chữ ký số

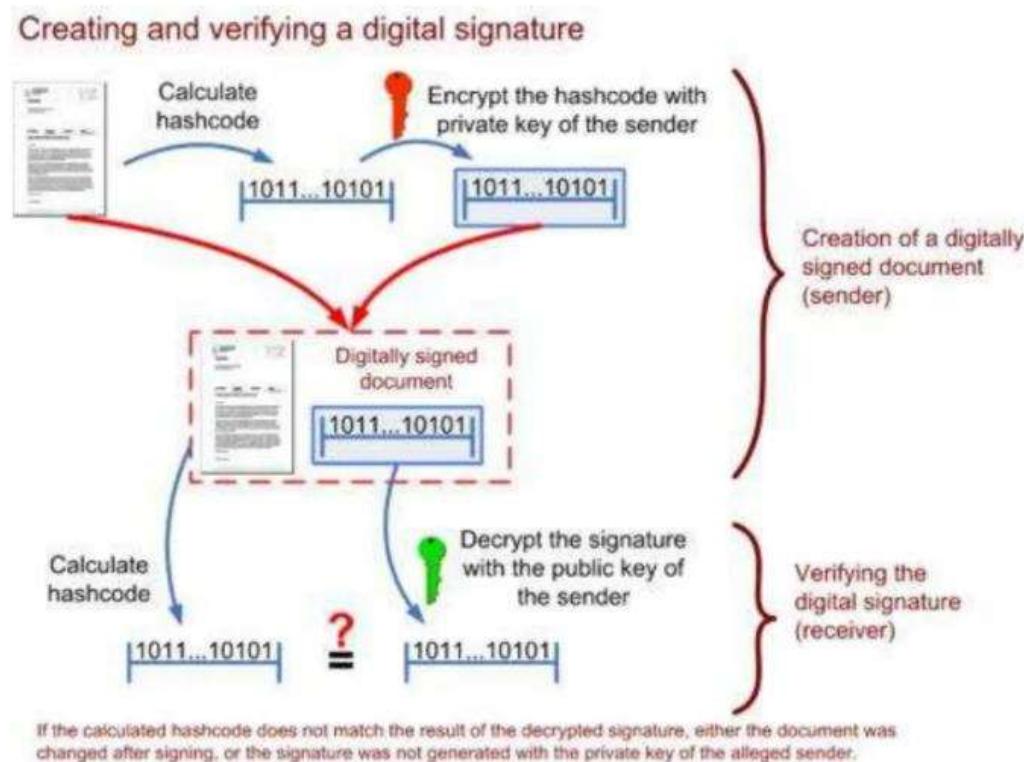
- **Chữ ký số:** Một cơ chế xác thực danh tính trong giao dịch Bitcoin bằng cách sử dụng khóa riêng tư và khóa công khai sử dụng thuật toán ECDSA (Elliptic Curve Digital Signature Algorithm).
- **Quy trình tạo chữ ký:**
 - 1) Dùng khóa riêng tư để tạo ra chữ ký số cho giao dịch.
 - 2) Chữ ký này được đính kèm vào giao dịch và gửi cùng khóa công khai.
- **Mục đích:**
 - Đảm bảo rằng giao dịch đến từ chủ sở hữu hợp lệ của khóa công khai.
 - Đảm bảo rằng giao dịch không thể bị thay đổi sau khi được ký.
 - Không thể giả mạo hoặc thay đổi giao dịch mà không phát hiện được.
- **Xác thực chữ ký:** Người nhận có thể sử dụng khóa công khai để xác minh chữ ký mà không cần biết khóa riêng tư.

5.11 ECDSA – Thuật toán chữ ký số trong Bitcoin

- **ECDSA** là viết tắt của Elliptic Curve Digital Signature Algorithm - thuật toán sinh chữ ký số dựa trên đường cong Elliptic (đã nói ở phần trước).
- Chữ ký số ECDSA gồm 2 thành phần chính:
 - Thuật toán tạo chữ ký từ khóa riêng tư (private key).
 - Thuật toán kiểm tra chữ ký với khóa công khai (public key).
- Thế mạnh của ECDSA:
 - Tiêu tốn ít tài nguyên hệ thống
 - Rút ngắn thời gian ký số
 - Tăng bảo mật

5.12 Quy trình tạo và xác minh chữ ký số

- Các bước tạo chữ ký số:
 - Dữ liệu ký: Giao dịch hoặc hash của các phần trong giao dịch.
 - Private key của người ký được sử dụng để tạo chữ ký.
 - Kết quả: Chữ ký được biểu diễn dưới dạng 2 giá trị R và S.



5.12 Quy trình tạo và xác minh chữ kí số

- Quy trình: Đầu tiên ta sẽ phải tạo chữ ký, chữ ký này có 2 phần ta gọi là R và S, để tạo chữ ký $ECDSA(R, S)$ ta sẽ có các bước sau:
 - Bước 1: Tạo ra một giá trị ngẫu nhiên gọi là K .
 - Bước 2: Ta tính $P(x, y) = K \times G$, lấy P_x được phần R của chữ ký.
 - Bước 3: Tính S. Đầu vào sẽ là tin nhắn đã được băm m cùng với dA , nghịch đảo của k và R .

Thuật toán:

$$S = k^{-1}(Hash(m) + dA * R) \pmod{p}$$

Chú thích: S là chữ kí số, dA là khóa riêng để ký, Qa là khóa công khai, m là dữ liệu giao dịch, k là khóa riêng tạm thời, R là tọa độ x của khóa công khai tạm thời, p là số nguyên tố của trường elip.

5.12 Quy trình tạo và xác minh chữ kí số

- Sau khi có được chữ ký, tiếp theo ta cần phải xác minh chữ ký ta có các bước sau đây:
 - Bước 1: Sử dụng khóa công khai Qa cùng điểm sinh G và tin nhắn đã băm.
 - Bước 2: Tính nghịch đảo của S trong chữ ký và gọi nó là S^{-1} .
 - Bước 3: Khôi phục điểm ngẫu nhiên P từ G , hàm đã băm m , S^{-1} , R và Qa .

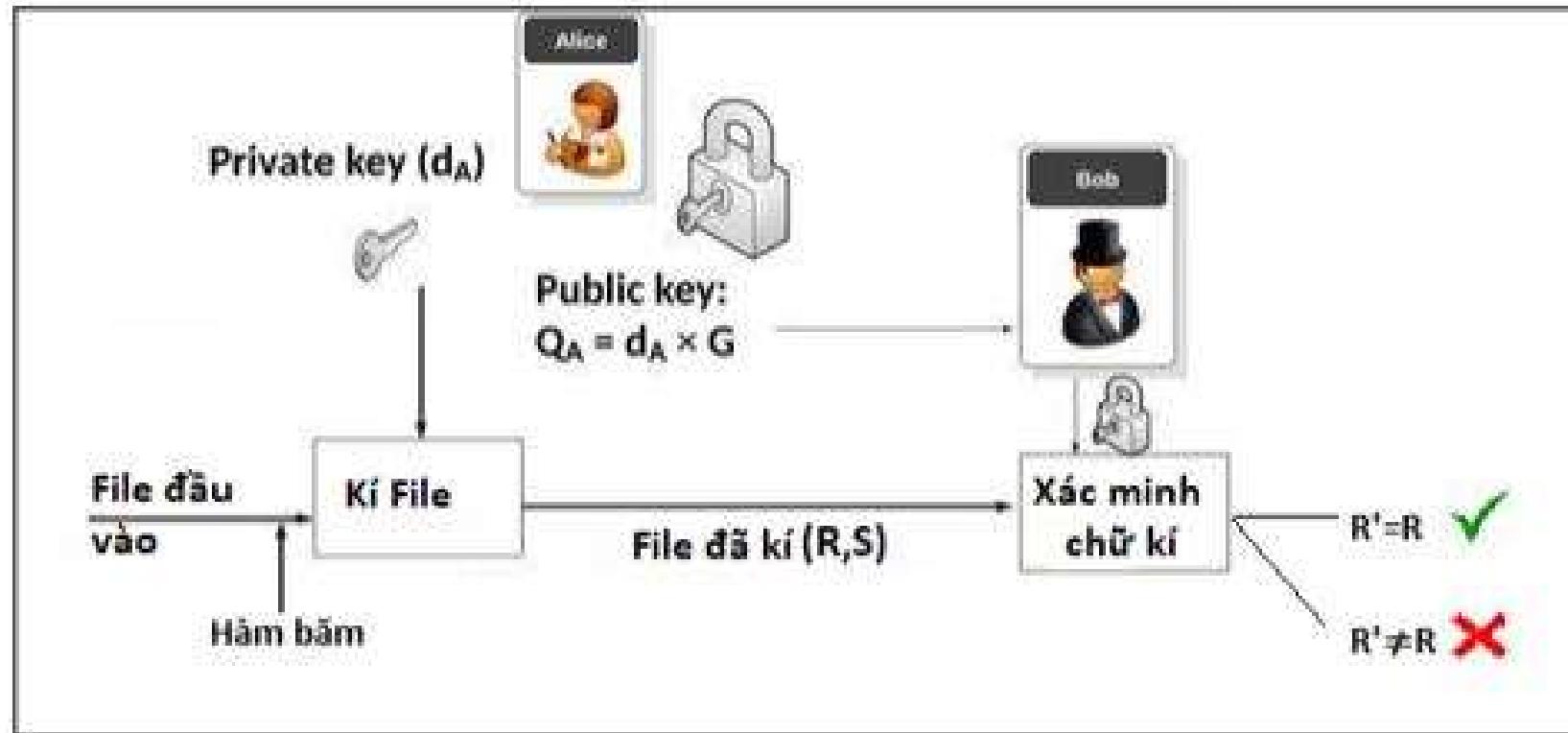
Thuật toán:

$$P = S^{-1} * \text{Hash}(m) * G + S^{-1} * R * Qa$$

Chú thích: - R và S hai giá trị của chữ ký số.

- Qa là khóa công khai
 - m là dữ liệu giao dịch đã được kí.
 - G là điểm sinh của đường cong Elliptic.
- Bước 4: So sánh nếu $P_x = R$ thì chữ ký hợp lệ, ngược lại là không hợp lệ.

5.12 Quy trình tạo và xác minh chữ kí số



5.13 Tâm quan trọng của khóa k ngẫu nhiên trong Chữ ký số

- Tại sao k ngẫu nhiên quan trọng?
 - Nếu sử dụng cùng k cho hai chữ ký khác nhau, khóa riêng có thể bị lộ.
 - Đã từng có trường hợp khóa riêng bị lộ do k được tái sử dụng.
- Giải pháp: Dùng thuật toán ngẫu nhiên quyết định (RFC 6979) để sinh k cho mỗi giao dịch nhằm tránh tái sử dụng.

5.14 Kết luận về chữ ký số

- Kết Luận:
 - Chữ ký số là một phần thiết yếu trong hệ thống Bitcoin, đảm bảo tính bảo mật và tính xác thực cho các giao dịch.
 - Nó cho phép nhiều bên tham gia ký vào giao dịch, tạo tính linh hoạt và khả năng tương tác cao.
- Ứng dụng của Chữ Ký Số
 - Hợp đồng thông minh (Smart Contract): Được sử dụng để xác minh và thực thi các điều khoản của hợp đồng tự động.
 - Quản lý danh tính: Chứng minh danh tính người dùng mà không tiết lộ thông tin nhạy cảm.
 - Tăng cường bảo mật: Sử dụng trong các giao dịch tài chính, bảo mật dữ liệu và nhiều ứng dụng khác trong công nghệ blockchain.

5.15 Địa chỉ Bitcoin

- **Địa chỉ Bitcoin:** Một định danh công khai được sử dụng để nhận Bitcoin, tương tự như số tài khoản ngân hàng.
- **Tạo địa chỉ:**
 - Bắt nguồn từ khóa công khai qua quá trình băm và mã hóa để bảo mật.
 - Đảm bảo rằng địa chỉ dễ sử dụng nhưng an toàn.
- **Bảo vệ thông tin:**
 - Địa chỉ Bitcoin không tiết lộ danh tính người dùng.
 - Tính ẩn danh cao nhưng vẫn có thể theo dõi qua chuỗi giao dịch công khai (blockchain).

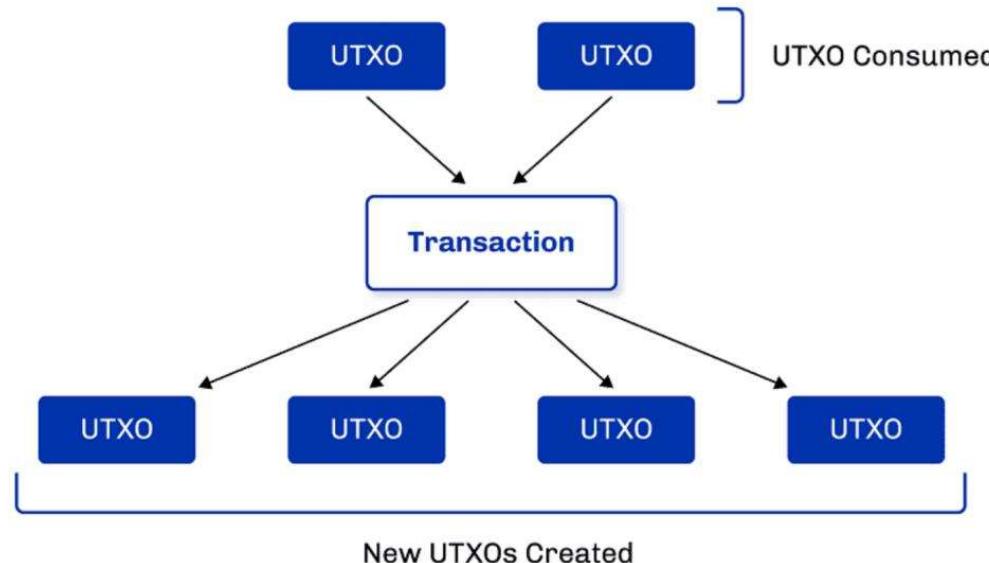
5.16 Số dư và tính trừu tượng khác

Số dư Bitcoin:

- Không phải lưu trữ dưới dạng số dư như trong tài khoản ngân hàng.
- Số dư là tổng của các đầu ra giao dịch chưa sử dụng (UTXOs).

Tính trừu tượng: Địa chỉ và số dư chỉ là các khái niệm trừu tượng, đại diện cho quyền kiểm soát tài sản số mà không tiết lộ thông tin cá nhân.

=> Bảo mật thông tin và giao dịch hiệu quả thông qua việc sử dụng địa chỉ và UTXO.





ĐẠI HỌC
BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY
OF SCIENCE AND TECHNOLOGY

Chương 6: Mạng lưới Bitcoin

ONE LOVE. ONE FUTURE.

6.1 Peer-to-peer Network Architecture

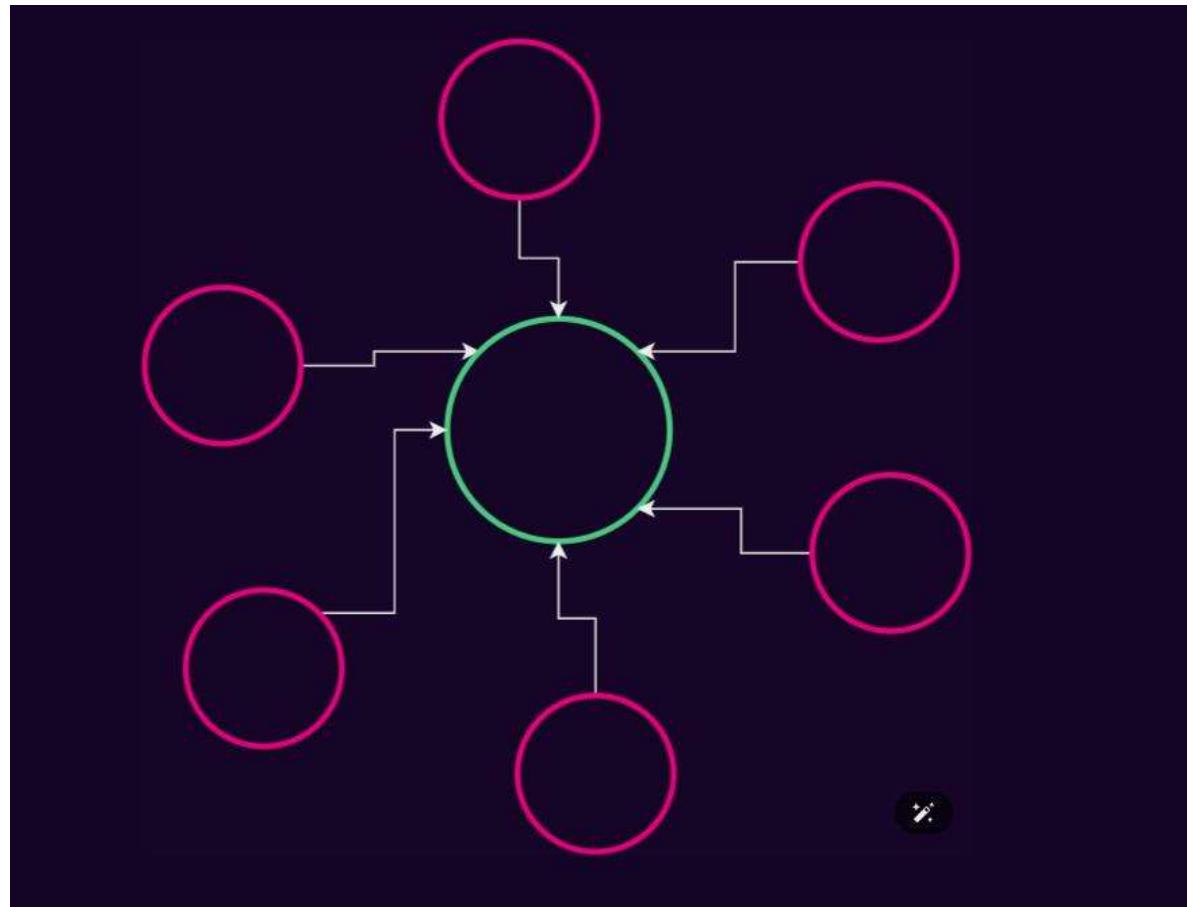
- Bitcoin được xây dựng trên kiến trúc mạng ngang hàng trên mạng Internet
- Peer-to-peer Network Architecture - Kiến trúc mạng ngang hàng - P2P
- Trong kiến trúc P2P, tất cả các node trong mạng đều ngang hàng và bình đẳng
- Khác biệt với kiến trúc Client – Server, các node không tập trung và không phân cấp
- Các node đều chịu trách nhiệm cung cấp cùng một lượng tài nguyên dịch vụ như nhau vào trong mạng
- Các node được kết nối dưới dạng mesh topology (Mạng lưới)
- Kiến trúc mạng là không tập trung
- Phân quyền là nguyên tắc lõi để thiết kế mạng Bitcoin theo kiến trúc P2P
- Thợ đào sử dụng stratum protocol

So sánh giữa mạng lưới Client và P2P

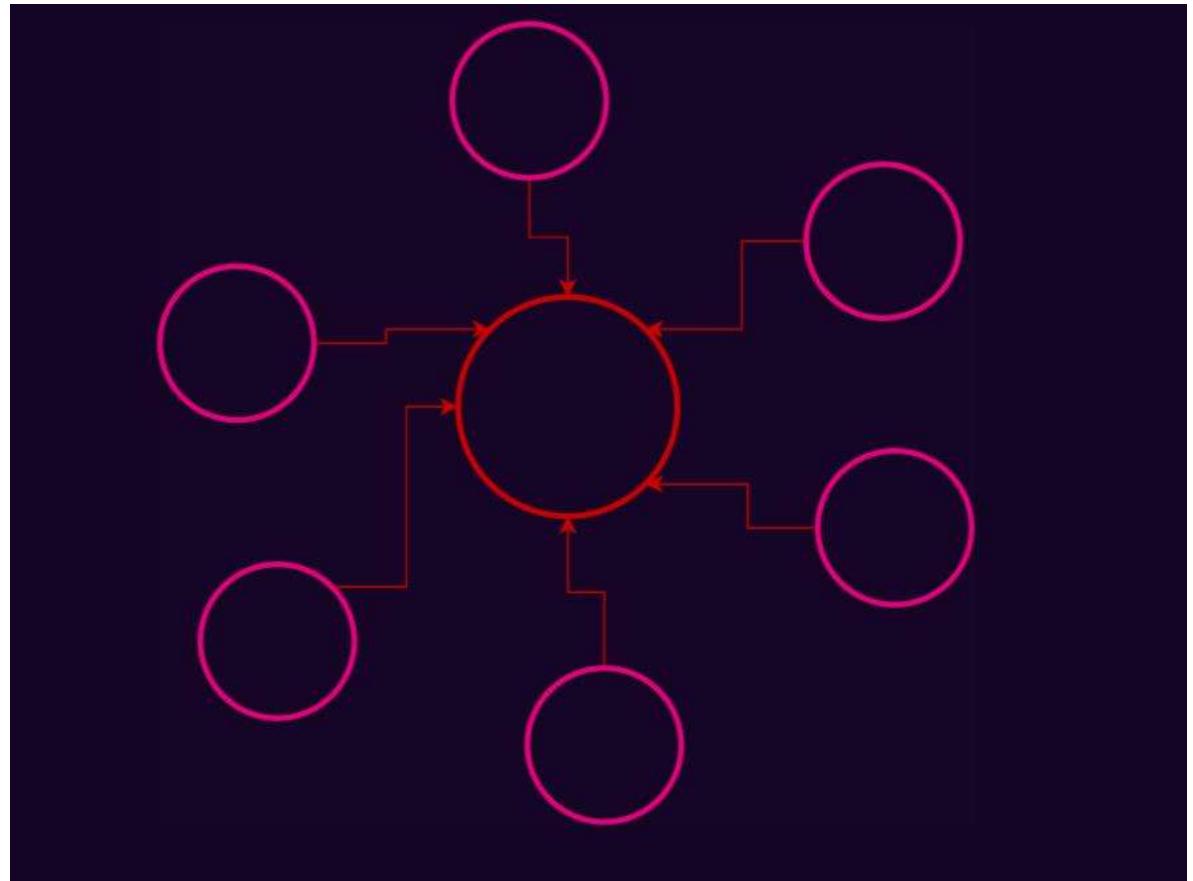
Client-Server vs Peer-to-Peer (P2P) Networks

	Client-Server Network	P2P Network
Structure	Centralized: One or more central servers control the network	Decentralized: All nodes (peers) participate equally
Data Flow	Server provides data to clients	Peers directly share data with each other
Resource Management	Servers manage resources and control access	Peers contribute resources including bandwidth, storage space, and processing power
Scalability	Can be limited by server capacity	Highly scalable due to the distribution of resources
Security	Centralized security measures, single point of failure	Potential for some security issues, malware(Depending on how it is implemented)

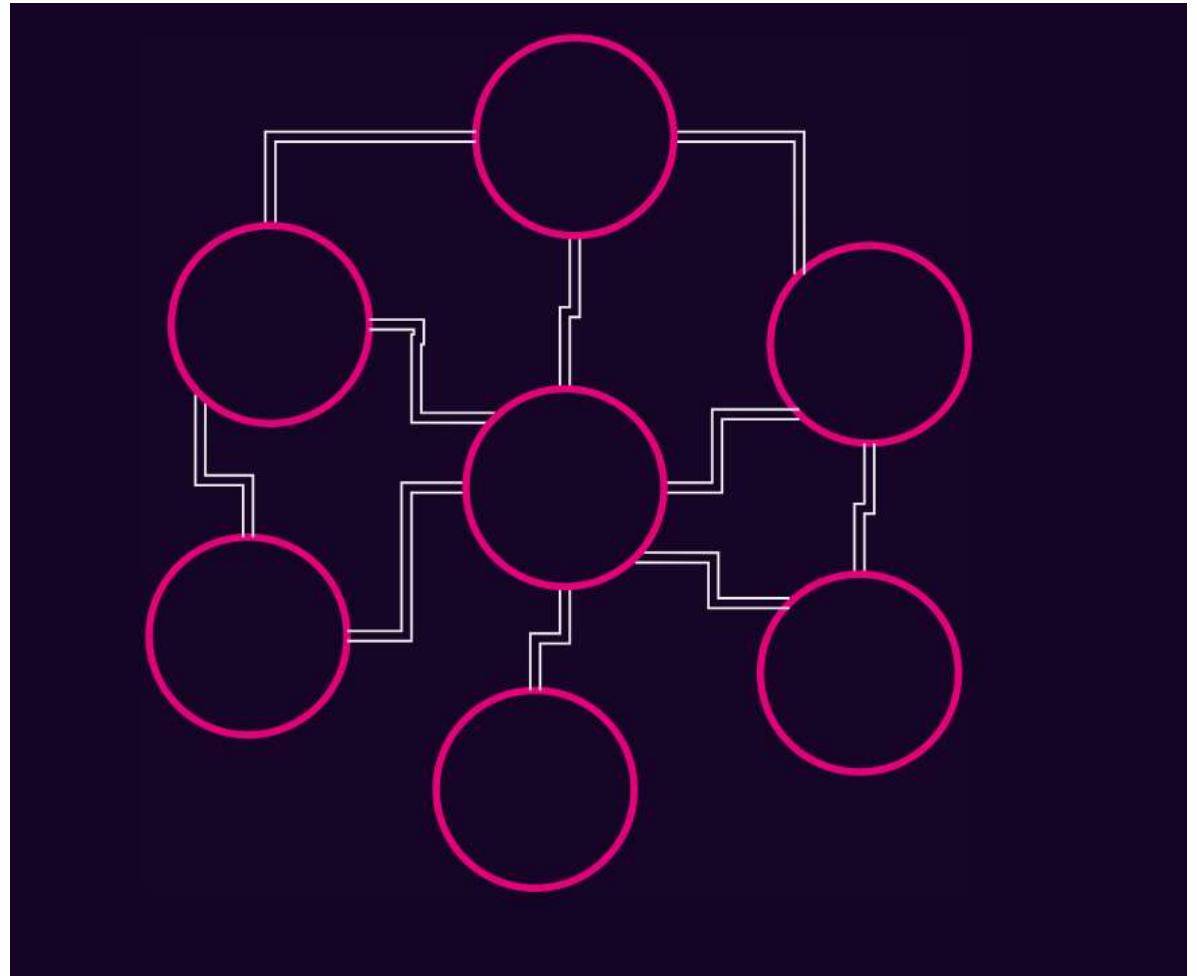
Centralized Network (Mạng tập trung)



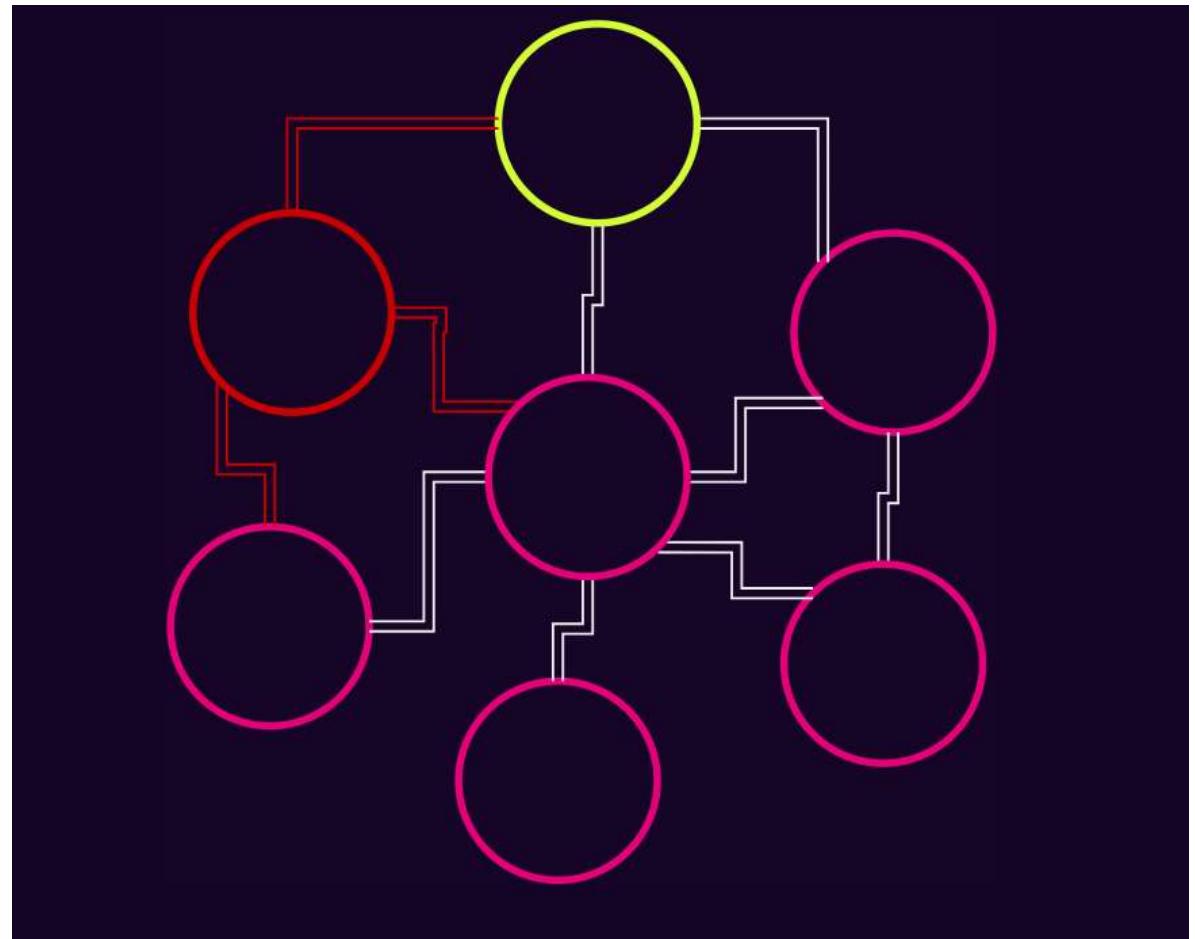
Centralized Network (Mạng tập trung)



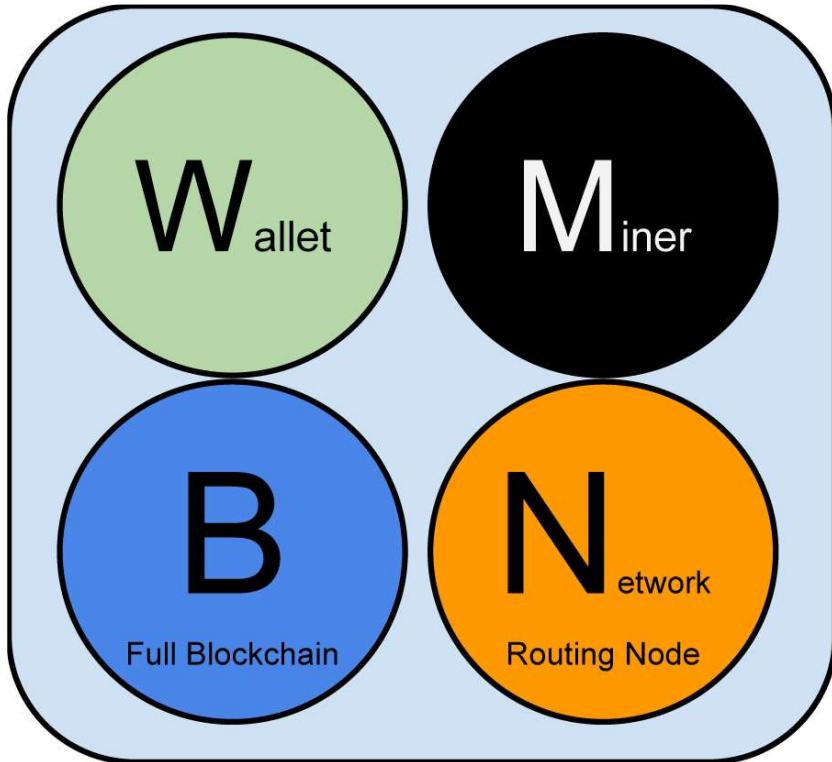
Decentralized
Network
(Mạng phi tập
trung)



Decentralized Network (Mạng phi tập trung)

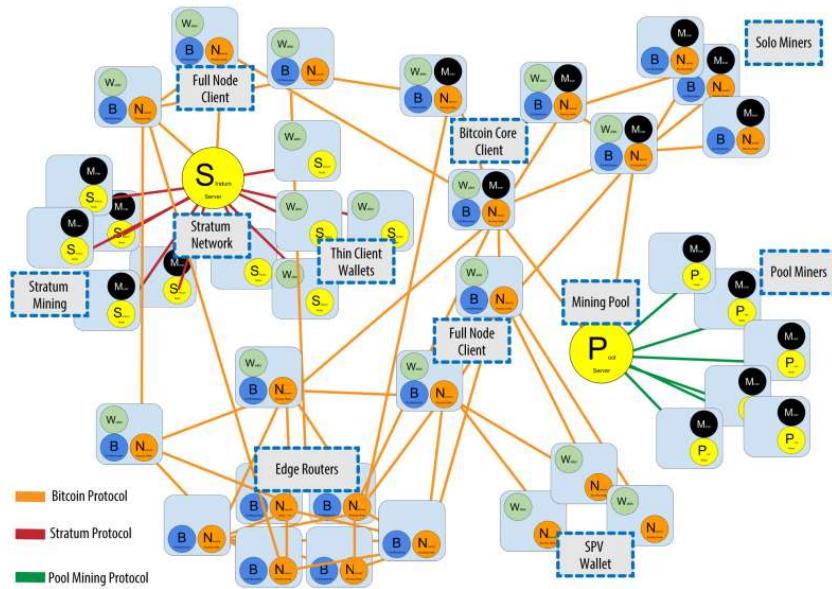


Bitcoin Node and Role



- **Bitcoin node**
- Các node trong Bitcoin Network ngang hàng, tuy vậy chúng vẫn đóng những vai trò khác nhau trong mạng tùy vào chức năng mà chúng hỗ trợ
- Các node có thể có những role
- Tất cả các node đều có chức năng định tuyến (routing function) để tham gia vào mạng

Mạng lưới Bitcoin mở rộng



- Bao gồm mạng chạy giao thức P2P và nút chạy các giao thức chuyên biệt
- Một số máy chủ nhóm và cổng giao thức kết nối các nút chạy các giao thức khác được đính vào mạng P2P.
- Các nút giao thức khác này chủ yếu là các nút khai thác nhóm và các máy khách ví nhẹ, không mang theo bản sao đầy đủ của chuỗi khối.
- Mạng Bitcoin hiển thị nhiều node, máy chủ, cổng, bộ định tuyến như sau:

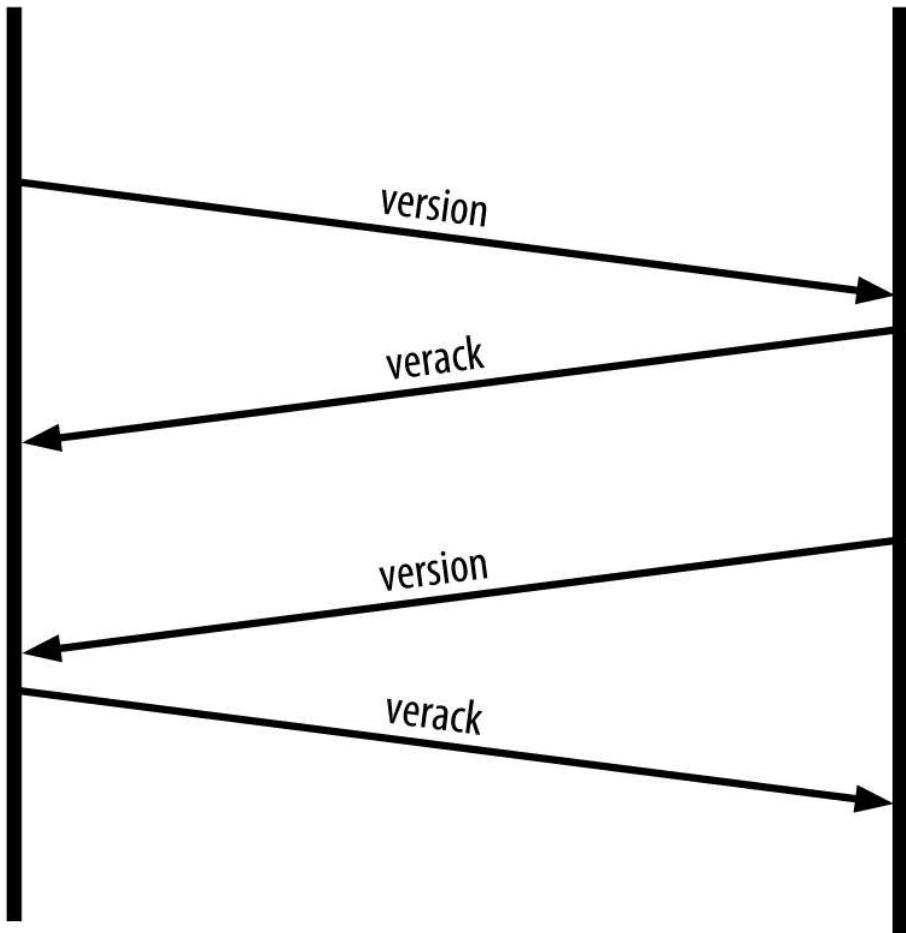
Mạng lưới chuyển tiếp Bitcoin

- Mạng P2P có độ trễ quá cao đối với miners. Thợ đào phải giảm thiểu thời gian lan truyền khối chiến thắng và thời gian bắt đầu vòng tiếp theo. Trong khai thác, độ trễ mạng liên quan trực tiếp đến biên lợi nhuận.
- Mạng chuyển tiếp bitcoin cố gắng giảm thiểu độ trễ trong quá trình truyền giữa các khối.
- Mạng chuyển tiếp Bitcoin gốc (2015) triển khai trên AWS, năm 2016 được thay thế bằng FIBRE (Fast Internet Bitcoin Relay Engine) - một mạng dựa trên UDP. Nó triển khai tối ưu hóa khối nhỏ gọn để giảm dữ liệu được truyền và độ trễ mạng.
- Falcon là một mạng chuyển tiếp mới sử dụng cut-through-routing thay vì store-and-forward
- Mạng chuyển tiếp là overlay network - để cung cấp kết nối giữa các nút có nhu cầu chuyên biệt. Đường tắt giữa các tuyến đường tắc nghẽn.
- => Mạng chuyển tiếp Bitcoin không phải để thay thế cho mạng P2P.

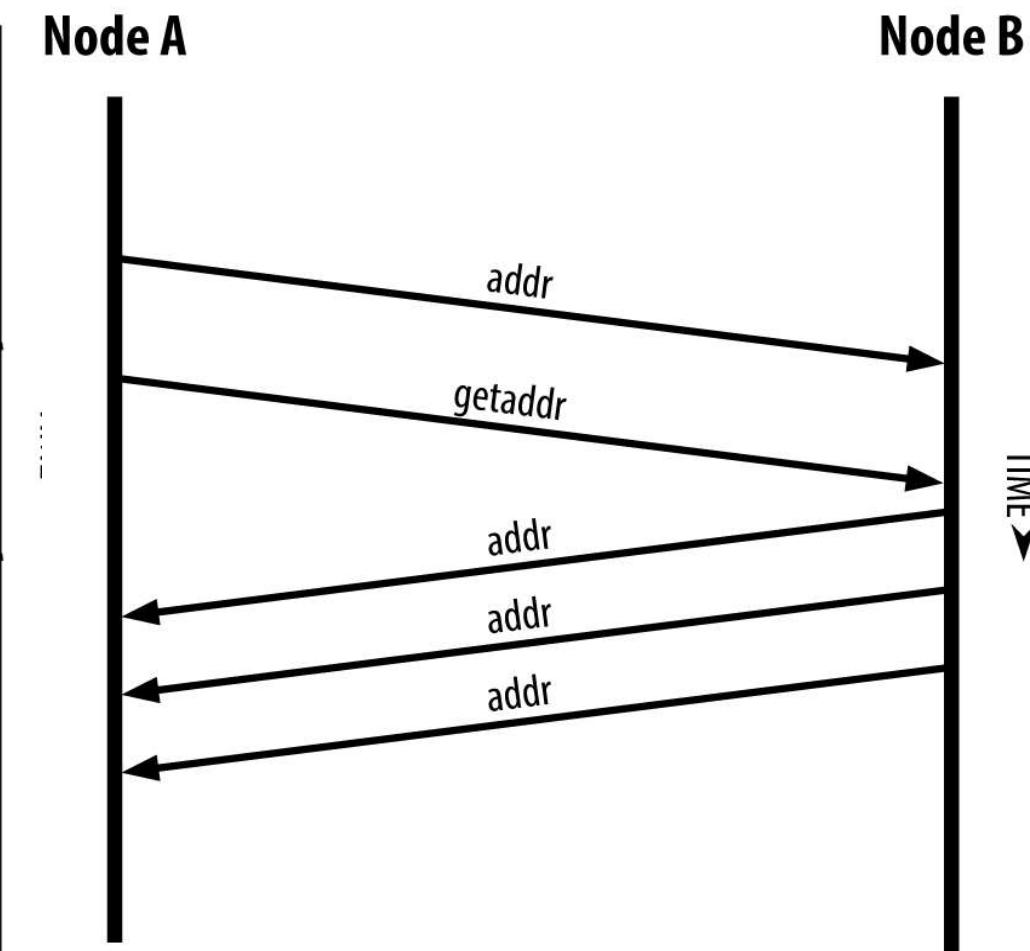
Explore Network

- Khi một nút mới khởi động - nó phải khám phá các nút bitcoin khác để tham gia. Vị trí địa lý không liên quan. Bất kỳ nút bitcoin hiện có nào cũng có thể được chọn ngẫu nhiên.
- Để kết nối với một đối tác đã biết - các nút kết nối qua TCP đến 8333. Để thiết lập kết nối, cần thực hiện bắt tay - chỉ định thông tin phiên bản và các thông tin cơ bản khác. (Truy vấn DNS thông qua DNS seed)
- nVersion - phiên bản mà giao thức bitcion nói đến
- nLocalServices - danh sách các dịch vụ được hỗ trợ bởi nút
- nTime - thời gian hiện tại
- addrYou - địa chỉ ip của nút từ xa
- addrMe - địa chỉ ip của nút cục bộ
- subver - sự phá hoại phần mềm đang chạy trên máy khách
- BestHeight - chiều cao của blockchain của nút này
- Nếu phiên bản tương thích, đối tác sẽ gửi một verack ??

Node A



Node B



Full Node: Node đầy đủ

- Node đầy đủ : full blockchain node
- Đồng thời chứa bản sao hoàn chỉnh của blockchain với toàn bộ giao dịch
- Node đầy đủ có thể thực hiện xác thực các giao dịch một cách độc lập
- Chờ đợi update các giao dịch từ mạng, sau đó xác minh và thêm vào bản sao cục bộ của blockchain được duy trì bởi node đầy đủ
- Nhược điểm: Mất nhiều thời gian để đồng bộ dữ liệu (ex:Running a node ves you the full experience - no reliance or trust in a third party is required. It is more than 400 GB now (August 2021) and it takes a few days to sync to the network - the price of independence and freedom.)
- Phổ biến nhất là Bitcoin core triển khai các client nút đầy đủ

Exchanging Inventory

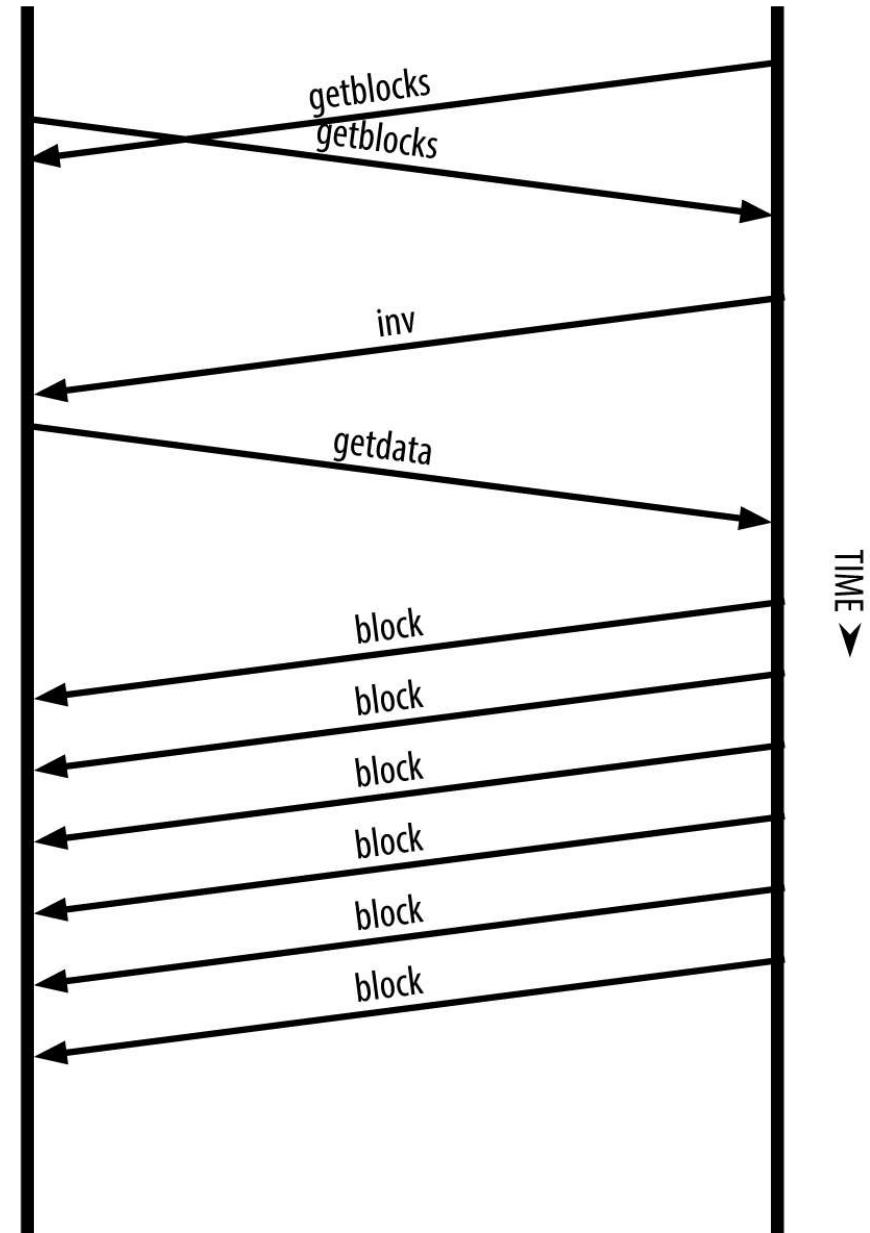
CÁCH THỨC TRAO ĐỔI "HÀNG TỒN"

- Một nút đầy đủ (full node) đầu tiên sẽ cố gắng xây dựng một blockchain hoàn chỉnh. Một nút hoàn toàn mới chỉ có khối khởi tạo (genesis block) được nhúng.
- Nút mới sẽ phải tải xuống và đồng bộ hóa tất cả các khối.
- Việc lấy thông tin về độ cao (BestHeight) của một peer cho phép so sánh số khối mà peer đó có.
- Một nút có độ cao cao hơn sẽ biết các nút cũ hơn cần khối nào. Nó sẽ xác định 500 khối đầu tiên mà nút cần và sẽ chuyển chúng với thông điệp inv (inventory). Nút thiếu khối sẽ gửi thông điệp getData để lấy chúng.
- Nút mới sẽ yêu cầu các khối từ các peer của nó - phân tán tải.

Continued...

Node đồng bộ hóa với blockchain bằng cách retrieving các khối từ một node ngang hàng

Node A **Node B**



Simple Payment Verification (SPV)

- Node SPV là gì ??
- Mỗi node chỉ bao gồm ví và node mạng
- Chỉ duy trì một phần của blockchain
- Các node sử dụng một dụng một phương thức xác thực giao dịch là SPV
- Chúng cũng được gọi là Light – weight Clients
- Các nút không lưu trữ toàn bộ khối mà chỉ lưu trữ tiêu đề (header) của từng khối

What is it?

This technique verifies whether a transaction has been included in the blockchain or not, without actually downloading the entire blockchain.

- SPV??
- Được sự dụng bởi một số thiết bị hiệu năng kém, hạn chế về tài nguyên và năng lượng và không thể download toàn bộ chuỗi khối
- Node SPV xác minh giao dịch bằng cách dựa vào các node ngang hàng để cung cấp những phần quan trọng và liên quan mà blockchain yêu cầu

Continue...

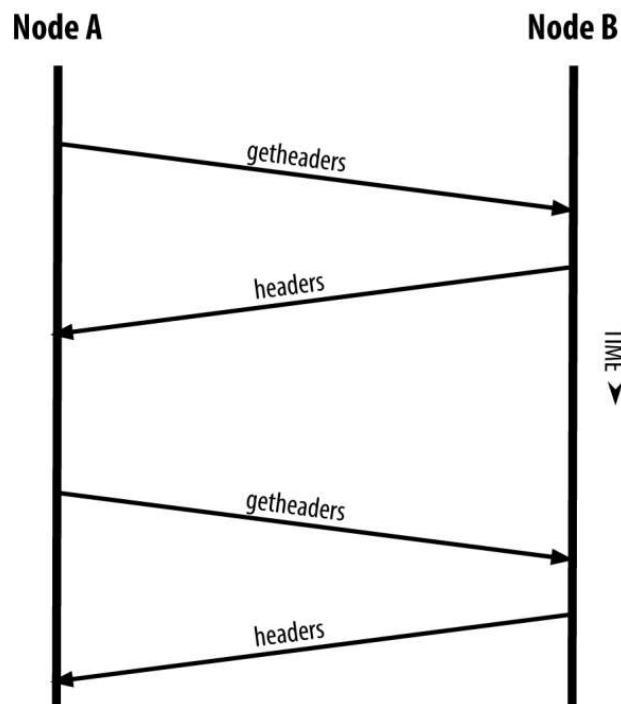
- Một node SPV cần tải về bản sao của tiêu đề khối từ chuỗi proof-of-work dài nhất.
- Sau đó, các giao dịch được xác minh bằng cách tham chiếu đến độ sâu của chúng trong blockchain thay vì chiều cao.
- Các node SPV thiết lập một liên kết giữa giao dịch và khối chứa nó bằng cách sử dụng đường dẫn Merkle.

****Lưu ý:**

- Node SPV không thể bị thuyết phục rằng một giao dịch tồn tại khi nó không có thật.
- Một node SPV có thể chắc chắn chứng minh rằng một giao dịch tồn tại, nhưng không thể xác minh rằng giao dịch đó, chẳng hạn như chi tiêu kép cùng một UTXO, không tồn tại vì nó không có bản ghi của tất cả các giao dịch. Loại tấn công này có thể được sử dụng như một cuộc tấn công từ chối dịch vụ (*Denial-of-Service*) hoặc một cuộc tấn công chi tiêu kép (*double-spending attack*) chống lại các node SPV.
- Để chống lại điều này, một node SPV cần kết nối ngẫu nhiên với nhiều nút khác nhau, nhằm tăng xác suất nó kết nối với ít nhất một nút trung thực.

Risk Involved in SPV

- Các nút SPV kiểm tra xem một giao dịch có tồn tại trong blockchain hay không bằng cách hỏi tất cả các nút khác mà nó kết nối.
- Điều này tiềm ẩn rủi ro về quyền riêng tư, vì nó có thể tiết lộ địa chỉ ví của người dùng.
- => Để khắc phục vấn đề về quyền riêng tư, một tính năng gọi là Bloom Filters được giới thiệu



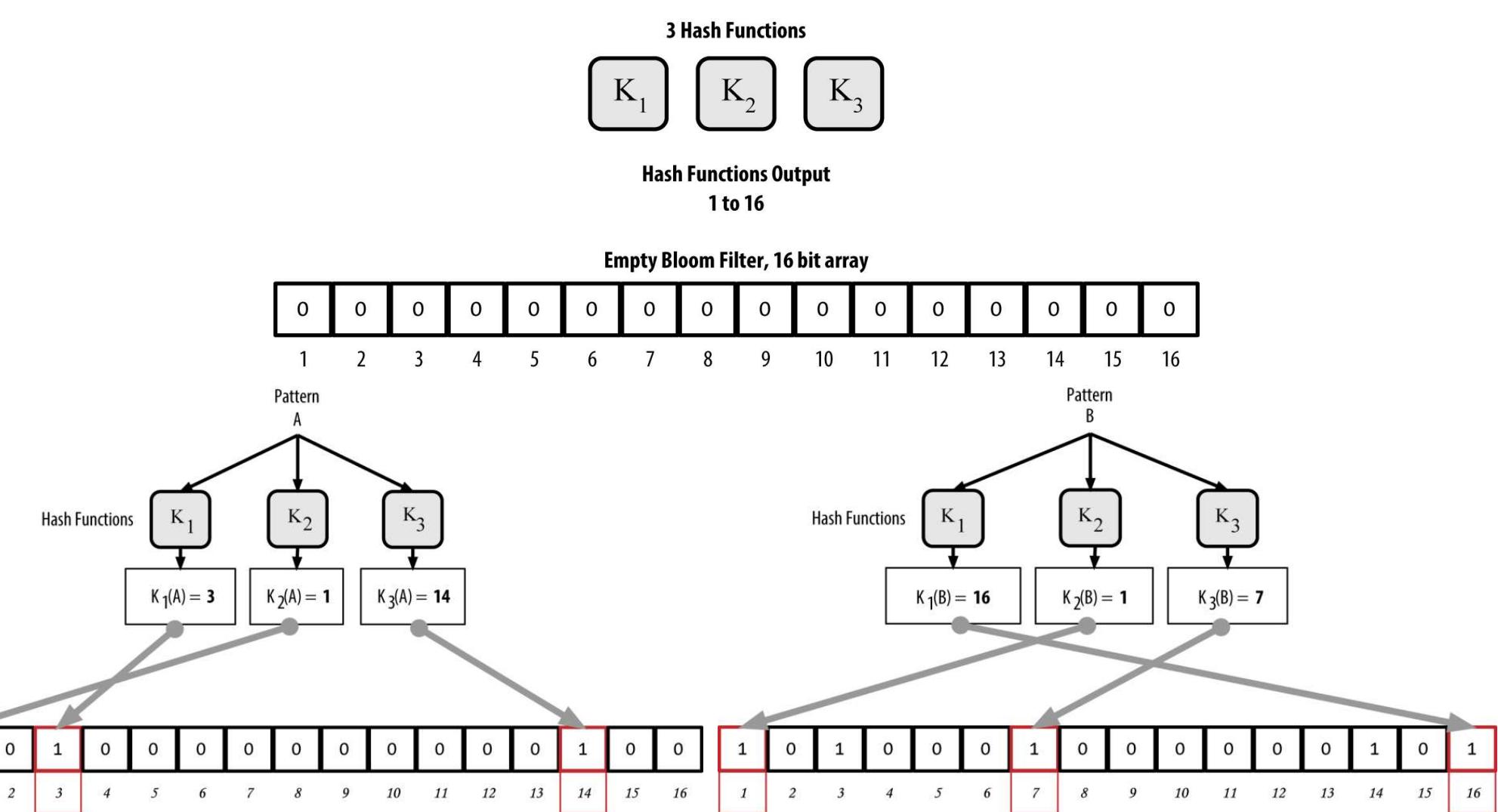
Bloom Filters

- Đặt vấn đề: Tại sao cần Bloom Filters??
- Bloom Filters (Bộ lọc Bloom) là một cấu trúc dữ liệu xác suất, đảm bảo hiệu quả về không gian, một cách để mô tả một mẫu mong muốn mà không cần chỉ định chính xác
- Chúng được các nút SPV sử dụng để yêu cầu các nút ngang hàng của chúng cung cấp các giao dịch khớp với một mẫu cụ thể, mà không tiết lộ chính xác địa chỉ, khóa hoặc giao dịch nào mà chúng đang tìm kiếm.
- Những kết quả sai (False Positive Result) có thể tồn tại do tính chất của hàm băm. Có thể 2 giá trị cùng được sinh ra với hàm băm, về nếu giá trị đó không tồn tại, kết quả từ bộ lọc vẫn có thể báo về là có tồn tại trong tập hợp đang tìm kiếm (Không có, có, có thể có)
- Nếu kết quả báo về là False Positive Result thì chắc chắn bộ lọc Bloom báo đúng
- Bộ lọc Bloom: Chính xác - Ít riêng tư và ngược lại

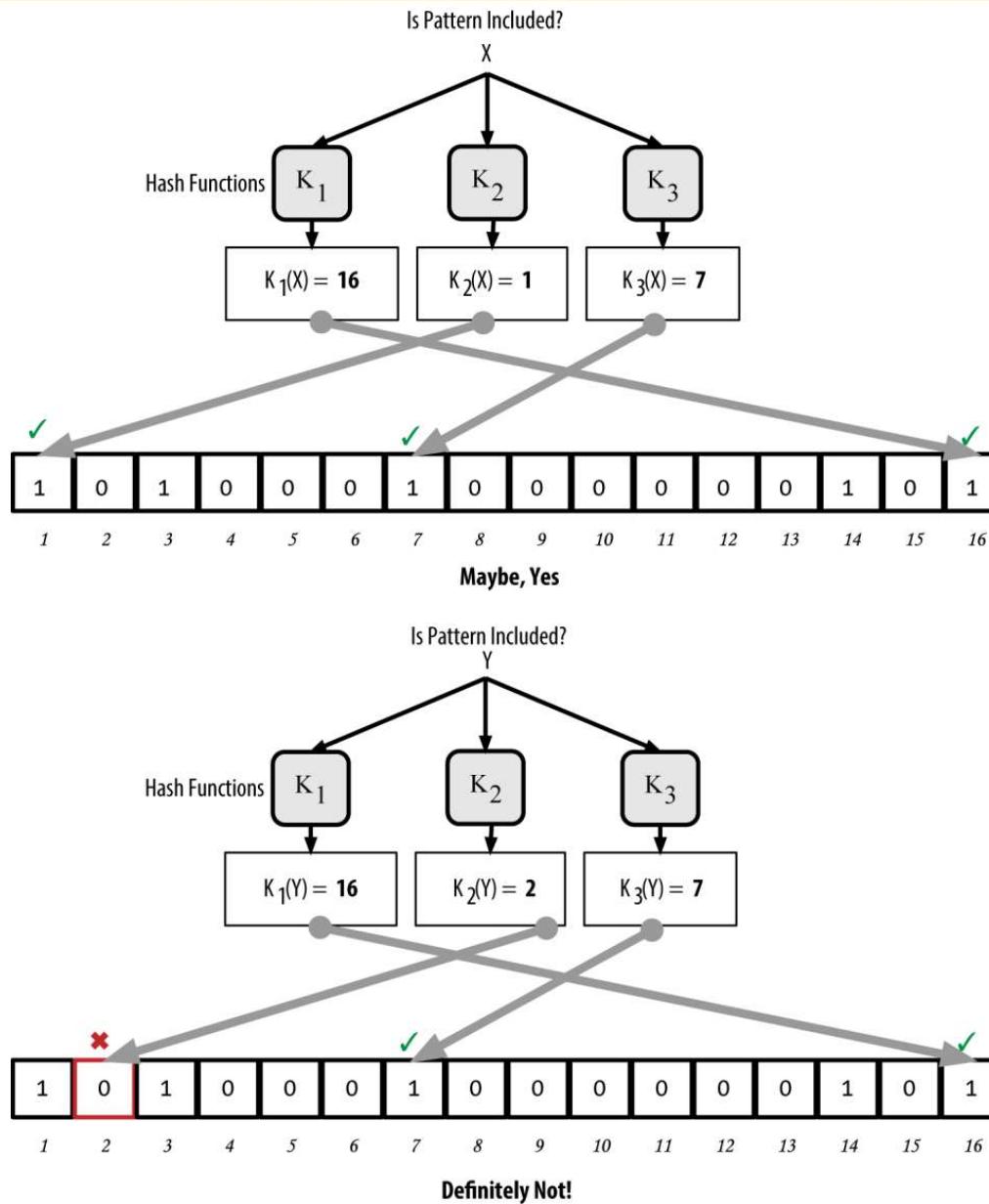
Bloom filters hoạt động như nào?

- **Khởi tạo:** Bloom filter bao gồm một mảng bit ban đầu với tất cả các giá trị bằng 0 và một số hàm băm (hash functions) khác nhau.
- **Thêm phần tử vào tập hợp:**
- Khi một phần tử (ví dụ, một chuỗi hoặc số) được thêm vào tập hợp, nó sẽ được xử lý bởi tất cả các hàm băm. Mỗi hàm băm sẽ trả về một chỉ số trong mảng bit.
- Các bit tại các vị trí chỉ số đó trong mảng bit sẽ được đặt thành 1.
- **Kiểm tra phần tử:**
- Để kiểm tra xem một phần tử có nằm trong tập hợp hay không, ta lại chạy nó qua các hàm băm.
- Nếu tất cả các bit tại các chỉ số do hàm băm trả về đều bằng 1, thì phần tử đó **có thể** đã tồn tại trong tập hợp (tức là có khả năng tồn tại false positive, nhưng không bao giờ có false negative).
- Nếu có bất kỳ bit nào tại vị trí chỉ số là 0, thì phần tử chắc chắn **không** tồn tại trong tập hợp.

Example of Bloom Filters



Example of Bloom Filters (continued...)



How are Bloom Filters used by SPV nodes

- SPV nodes yêu cầu các nút ngang hàng về những mẫu cụ thể mà "match" với giao dịch đang cần mà không cần tiết lộ toàn bộ địa chỉ công khai của giao dịch
- SPV node khởi tạo bộ lọc Bloom hoàn toàn rỗng và không "match" với bất kỳ mẫu nào của trạng thái
- Sau đó, SPV node sẽ tạo một danh sách các địa chỉ của giao dịch ở trong ví, khóa và hàm băm
- Sau đó, nút SPV thêm từng mục này vào bộ lọc bloom, để bộ lọc bloom sẽ "khớp" nếu các mẫu này có trong giao dịch, mà không tiết lộ chính các mẫu đó.
- Sau đó, nút SPV sẽ gửi một thông báo filterload đến đối tác, chứa bộ lọc bloom để sử dụng trên kết nối. Trên đối tác, bộ lọc bloom được kiểm tra đối với mỗi giao dịch đến. Nút đầy đủ kiểm tra một số phần của giao dịch đối với bộ lọc bloom, tìm kiếm sự khớp bao gồm:
 - ID giao dịch
 - Các thành phần dữ liệu từ các tập lệnh khóa của mỗi đầu ra giao dịch (mọi khóa và hàm băm trong tập lệnh)
 - Mỗi giao dịch đầu vào
 - Mỗi thành phần dữ liệu chữ ký đầu vào (hoặc tập lệnh chứng kiến)

Kết nối mã hóa ngang hàng và xác thực

- Ban đầu, tất cả các node truyền dữ liệu dưới dạng văn bản rõ ràng. Điều này không phải là vấn đề đối với các node đầy đủ (full node), nhưng lại là vấn đề đối với các SPV (Simplified Payment Verification - xác minh thanh toán đơn giản).
- Có hai giải pháp:
- **Vận chuyển qua Tor (Tor transport)**
- **Xác thực và mã hóa P2P (Peer-to-peer)**
- Tor Transport là gì?
- Tor viết tắt của "The Onion Routing", cung cấp mã hóa và đóng gói dữ liệu thông qua các đường truyền ngẫu nhiên trên mạng giúp đảm bảo tính ẩn danh, không thể truy vết, và bảo mật.
- Từ phiên bản Bitcoin Core 0.12, một node sẽ tự động cung cấp dịch vụ ẩn Tor.
- Bật chế độ log của Bitcoin Core cho Tor bằng lệnh:
bitcoind --daemon --debug=tor
- *Hiển thị : tor: ADD_ONION successful*

Continued...

- Kết nối và xác thực ngang hàng (P2P)
- Có hai đề xuất cải tiến Bitcoin (Bitcoin Improvement Proposal - BIP):
- **BIP-150:** Cung cấp tùy chọn xác thực đồng đẳng, cho phép các node xác thực danh tính của nhau bằng cách sử dụng ECDSA và khóa riêng.
- **BIP-151:** Kích hoạt mã hóa thỏa thuận cho tất cả các giao tiếp giữa hai node hỗ trợ BIP-151.
- Tất cả các BIP của Bitcoin có thể được xem trên GitHub.



Mathematics of Bloom Filter

- Let k be the number of hash functions, m be the total number of bits in the vector and n be the number of items in the set.
- Probability that certain bit will still be 0 after 1 insertion= $(1-1/m)^k$.
- Then, after n insertions the probability will be $(1-1/m)^{kn}$.
- So, the probability of false positives= $(1-(1-1/m)^{kn})^k$.

Nhóm giao dịch

- Danh sách tạm thời của các giao dịch chưa được xác nhận - gọi là **memory pool** hoặc **transaction pool**.
- Pool này theo dõi các giao dịch đã được mạng biết nhưng chưa được đưa vào blockchain. Các giao dịch được chuyển tiếp đến các node lân cận.
- **Giao dịch mồ côi** (orphaned transactions) tham chiếu đến các giao dịch mà node chưa biết đến. **UTXO pool** chứa hàng triệu mục của các giao dịch chưa chi tiêu. UTXO chỉ chứa các đầu ra đã được xác nhận và ít thay đổi giữa các node.

Chương 7: Blockchain

Introduction

- Blockchain data structure is an ordered, back linked list of block of transactions
- Lưu trữ dưới dạng tệp phẳng (flat file) hoặc cơ sở dữ liệu đơn giản
- Mỗi khối (block) được xác định bằng hàm băm SHA256.
- Các khối liên kết ngược, mỗi khối tham chiếu tới khối trước nó, được gọi là khối cha
- Chuỗi các hàm băm liên kết mỗi khối với khối cha của nó tạo ra một chuỗi quay ngược lại đến khối đầu tiên từng được tạo ra, được gọi là *khối genesis*.
- Bitcoin đã chuyển từ sử dụng Berkley DB sang LevelDB. Các khối xếp chồng lên nhau tạo nên thuật ngữ - chiều cao khối.
- Trường khối trước đó nằm trong tiêu đề và do đó ảnh hưởng đến hash của khối hiện tại. Nếu khối cha thay đổi - hash của nó sẽ thay đổi. Thay đổi con trở đến hash của khối trước đó sẽ buộc hash của con thay đổi. Tính toán cần thiết để thay đổi nó làm cho lịch sử của blockchain không thể thay đổi. \

Khối trong Blockchain

Mỗi block trong Blockchain bao gồm các thành phần sau:

- Index (Block #): Thứ tự của block (block gốc có thứ tự 0)
- Hash: Giá trị băm của block
- Previous Hash: Giá trị băm của block trước
- Timestamp: Thời gian tạo của block
- Data: Thông tin lưu trữ trong block
- Nonce: Giá trị biến thiên để tìm ra giá trị băm thỏa mãn yêu cầu của mỗi Blockchain.

Giá trị băm (Hash) sẽ băm toàn bộ các thông tin cần thiết như timestamp, previous hash, index, data, nonce.

Structure of a block

- Mỗi Khối bao gồm:

Size	Field	Description
4 bytes	Block Size	The size of the block, in bytes, following this field
80 bytes	Block Header	Several fields form the block header
1–9 bytes (VarInt)	Transaction Counter	How many transactions follow
Variable	Transactions	The transactions recorded in this block

Block Header (Tiêu đề khối)

- Tiêu đề khối bao gồm ba bộ siêu dữ liệu khối.
- Đầu tiên, có một tham chiếu đến một băm khối trước đó, kết nối khối này với khối trước đó trong chuỗi khối.
- Thứ hai, cụ thể là độ khó , *dấu thời gian* và *nonce* , liên quan đến cuộc thi khai thác,
- Phần siêu dữ liệu thứ ba là gốc cây merkle, một cấu trúc dữ liệu được sử dụng để tóm tắt hiệu quả tất cả các giao dịch trong khối.

Size	Field	Description
4 bytes	Version	A version number to track software/protocol upgrades
32 bytes	Previous Block Hash	A reference to the hash of the previous (parent) block in the chain
32 bytes	Merkle Root	A hash of the root of the merkle tree of this block's transactions
4 bytes	Timestamp	The approximate creation time of this block (seconds from Unix Epoch)
4 bytes	Difficulty Target	The Proof-of-Work algorithm difficulty target for this block
4 bytes	Nonce	A counter used for the Proof-of-Work algorithm

Mã định danh khối: Block Header Hash

- Mã định danh CHÍNH của một khối là giá trị băm mật mã của nó, một dấu vân tay điện tử được tạo ra bằng cách băm tiêu đề của khối (Block Header) 2 lần qua thuật toán SHA256.
- Kết quả băm 32 byte được gọi một cách chính xác là *Hàm băm tiêu đề khối*
- Hàm băm khối không thực sự nằm bên trong cấu trúc khối. Nó được tính toán bởi mỗi node khi khối được nhận từ mạng
- **Ví dụ:**
- 000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f là hàm băm khối của khối bitcoin đầu tiên từng được tạo ra.
- Hàm băm khối xác định một khối một cách duy nhất và rõ ràng và có thể được bất kỳ nút nào độc lập suy ra bằng cách chỉ cần băm tiêu đề khối.

Mã định danh khối: Block Height

- Cách thứ 2 để xác định một khối là theo vị trí của khối trong chuỗi, được gọi là **BLOCK HEIGHT** (*Chiều cao của khối*)
- **Khối đầu tiên:** Có chiều cao là 0 và được gọi là khối khởi tạo.
- **Các khối tiếp theo:** Mỗi khối mới được thêm vào sẽ có chiều cao tăng lên 1 đơn vị so với khối trước đó.
- **Ví dụ:** Vào ngày 1/1/2017, chuỗi khối đã có 446.000 khối, nghĩa là đã có 446.000 giao dịch được ghi nhận và xác thực.
- Chiều cao khối cũng có thể được lưu trữ dưới dạng siêu dữ liệu trong bảng cơ sở dữ liệu được lập chỉ mục để truy xuất nhanh hơn.
- **Giá trị Băm khối** của một khối luôn xác định một khối duy nhất. Một khối cũng luôn có *chiều cao khối cụ thể*. Tuy nhiên, không phải lúc nào chiều cao khối cụ thể cũng có thể xác định một khối duy nhất. Thay vào đó, hai hoặc nhiều khối có thể cạnh tranh cho một vị trí duy nhất trong chuỗi khối.



Khối Genesis

- Là khối đầu tiên trong chuỗi khối, là khối tổ tiên và được tạo ra vào năm 2009
- Mỗi nút luôn biết hàm băm và cấu trúc khối genesis.
- Do đó node đều có điểm khởi đầu cho blockchain, một "gốc" an toàn để xây dựng một blockchain đáng tin cậy.
- Được mã hóa tĩnh bên trong máy khách Bitcoin Core
- **Lấy mã băm khối genesis:**
- `// ubuntu@btc:~$ bitcoin-cli getblockhash 0
00000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f//`

Note: Khối genesis chứa một thông điệp ẩn bên trong. Đầu vào giao dịch coinbase chứa văn bản "The Times 03/Jan/2009 Chancellor on brink of second bailout for banks." Thông điệp này nhằm mục đích cung cấp bằng chứng về ngày sớm nhất khối này được tạo ra, bằng cách tham chiếu đến tiêu đề của tờ báo Anh *The Times*. Nó cũng đóng vai trò như một lời nhắc nhở dí dỏm về tầm quan trọng của một hệ thống tiền tệ độc lập, với sự ra mắt của bitcoin diễn ra cùng lúc với cuộc khủng hoảng tiền tệ toàn cầu chưa từng có. Thông điệp được nhúng vào khối đầu tiên bởi Satoshi Nakamoto, người tạo ra bitcoin.

Khối Genesis

Genesis Block	
 Previous Hash	0
 Timestamp	Thu, 27 Jul 2017 02:30:00 GMT
 Data	Welcome to Blockchain CLI!
 Hash	0000018035a828da0...
 Nonce	56551

Hình 1.3: Cấu trúc của block gốc trong blockchain

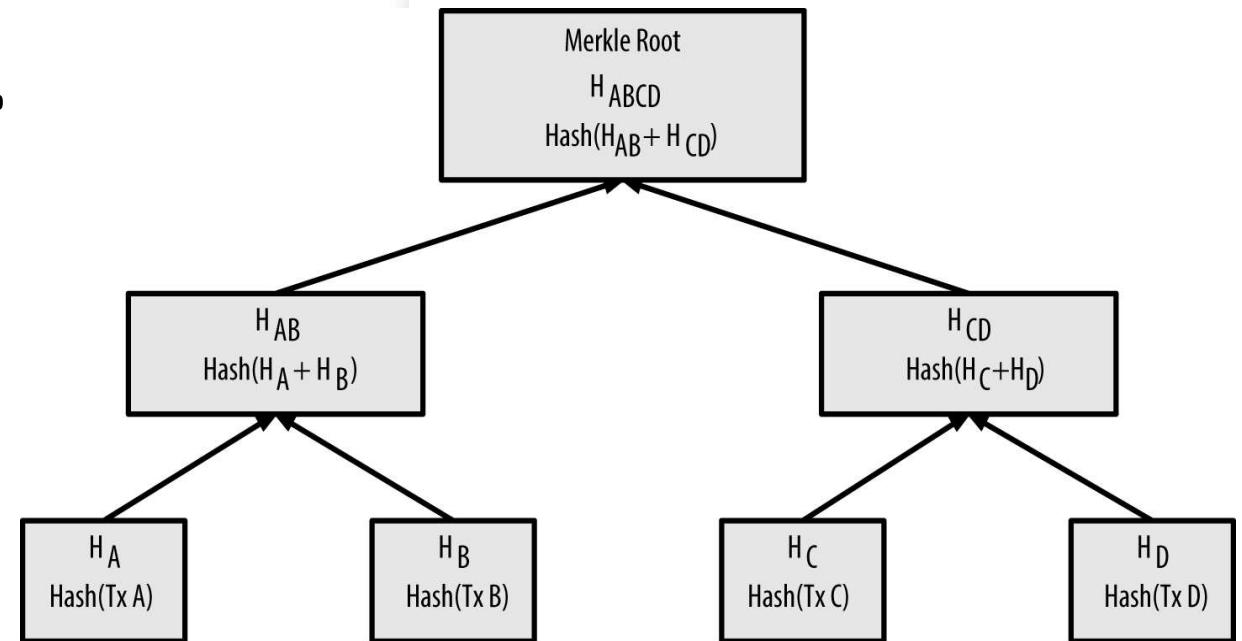
Liên kết các khối trong Blockchain

- Bản sao cục bộ của Blockchain liên tục cập nhập khi các chuỗi mới được tìm thấy và được sử dụng để mở rộng chuỗi
- Khi một nút nhận được các khối đến từ mạng, nó sẽ xác thực các khối này và sau đó liên kết chúng với blockchain hiện có.
- Để thiết lập liên kết, một nút sẽ kiểm tra tiêu đề khối đến và tìm kiếm "băm khối trước đó".
- **Previousblockhash:** Trường này chứa hàm băm của khối cha.

Cây Merkle

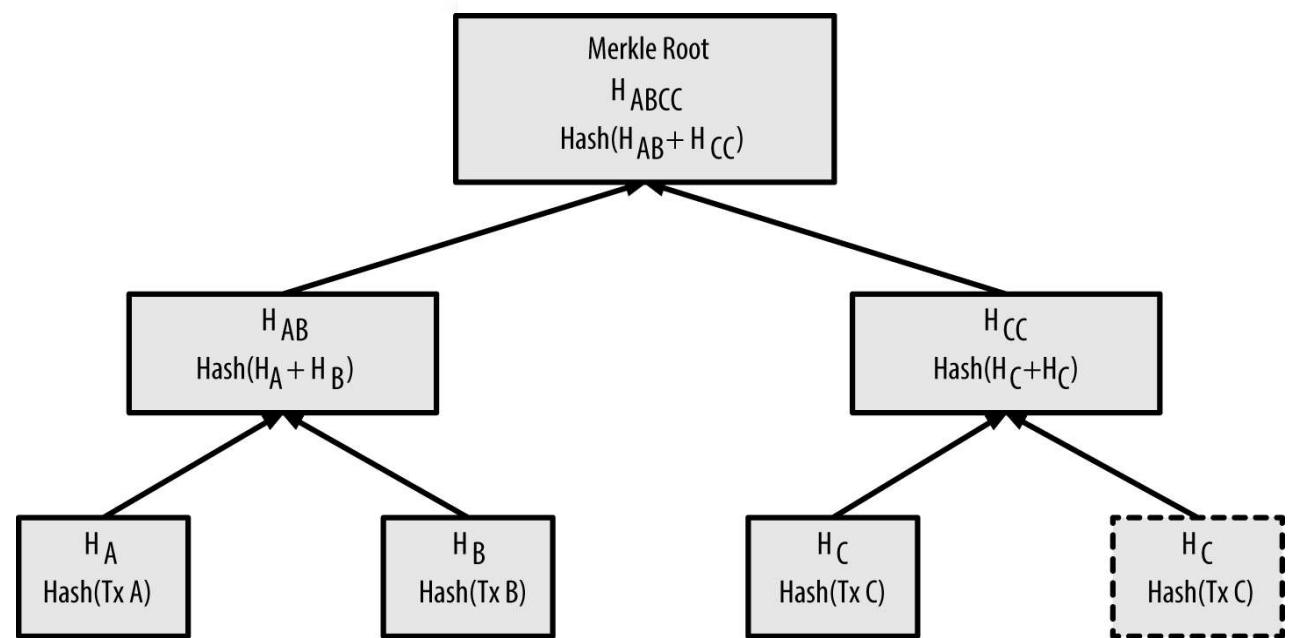
- Được sử dụng để chứa bản tóm tắt tất cả giao dịch trong khối
- Còn được gọi là *cây băm nhị phân*
- Cây Merkle được xây dựng bằng cách băm đệ quy các cặp nút cho đến khi chỉ còn một hàm băm, được gọi là *gốc merkle* .
- SHA256 được áp dụng hai lần, còn được gọi là SHA256 kép.
- Cấu trúc dữ liệu hiệu quả để xác minh giao dịch.
- Số phép tính tối đa để tìm kiếm: ($\log_2 n$)

Continued...



$$H_{AB} = \text{SHA256}(\text{SHA256}(H_A + H_B))$$

Số lượng nút lá phải chẵn => tạo bản sao ở giá trị băm giao dịch cuối cùng



Cây Merkle và Xác minh thanh toán đơn giản (SPV)

- Cây Merkle được các nút SPV sử dụng rộng rãi.
- Các nút SPV không có tất cả các giao dịch và không tải xuống toàn bộ các khối, chỉ có các tiêu đề khối.
- Để xác minh rằng một giao dịch được bao gồm trong một khối, mà không cần phải tải xuống tất cả các giao dịch trong khối, chúng sử dụng một đường dẫn xác thực hoặc đường dẫn merkle.

Bitcoin's Test Blockchain

- mainnet- Ngày 3 tháng 1 năm 2009
- testnet
- segnet
- regnet
- Testnet là tên của blockchain, mạng lưới và tiền tệ thử nghiệm. Đây là một ví P2P trực tiếp có đầy đủ tính năng, testcoin và khai thác.
- Khai thác Testnet vẫn phải dễ dàng và tiền xu sẽ không có giá trị
- Bảo vệ khỏi mất mát tiền bạc và bảo vệ mạng khỏi lỗi
- Thỉnh thoảng, mạng thử nghiệm lại bị thu thập dữ liệu.
- Lần lặp hiện tại là testnet3

Bitcoin's Test Blockchain

- Sử dụng testnet
- Run: *bitcoind -testnet*
- hoặc thiết lập *testnet=1* trong *bitcoin.conf*
- Segnet: Mạng thử nghiệm Segregated Witness
- Mục đích đặc biệt: phát triển và thử nghiệm - mục đích duy nhất
- Regtest: Blockchain cục bộ
- Một blockchain cục bộ cho mục đích thử nghiệm Dự định chạy trên các hệ thống cục bộ
- *bitcoind -regtest*
- của
- *regtest=1*
- Có thể tạo khối với
- *bitcoin-cli -regtest generate 500*

Mining and Consensus

What is mining?

khai thác không phải là tạo ra bitcoin mới. Đó là hệ thống khuyến khích. Khai thác là cơ chế mà tính bảo mật của bitcoin được phân cấp.

Đó là quá trình thực hiện nhiều lần việc băm (hash) một khối dữ liệu, sau khi thay đổi một tham số nào đó trong khối đó. Mục tiêu là để giá trị băm thu được khớp với một mẫu nhất định hoặc một giá trị đích.

- Mẫu băm được tạo ra không thể xác định trước được.
- Không thể tạo ra một mẫu để tạo ra một giá trị băm cụ thể.
- Điều này đảm bảo rằng giá trị băm chỉ có thể được tạo ra bằng cách sửa đổi giá trị trong khối và thử lại nhiều lần, cho đến khi giá trị băm trùng khớp với giá trị đích một cách ngẫu nhiên.
- Động lực chính cho việc khai thác là phần thưởng khối.
- Thợ đào nhận được hai loại phần thưởng để đổi lấy sự bảo mật do khai thác cung cấp: tiền mới được tạo ra với mỗi khối mới và phí giao dịch từ tất cả các giao dịch có trong khối
- Khai thác hỗ trợ cho trung tâm thanh toán phi tập trung mà qua đó các giao dịch được xác thực.
- Khai thác làm cho bitcoin trở nên đặc biệt - một cơ chế bảo mật phi tập trung là cơ sở cho tiền kỹ thuật số P2P.

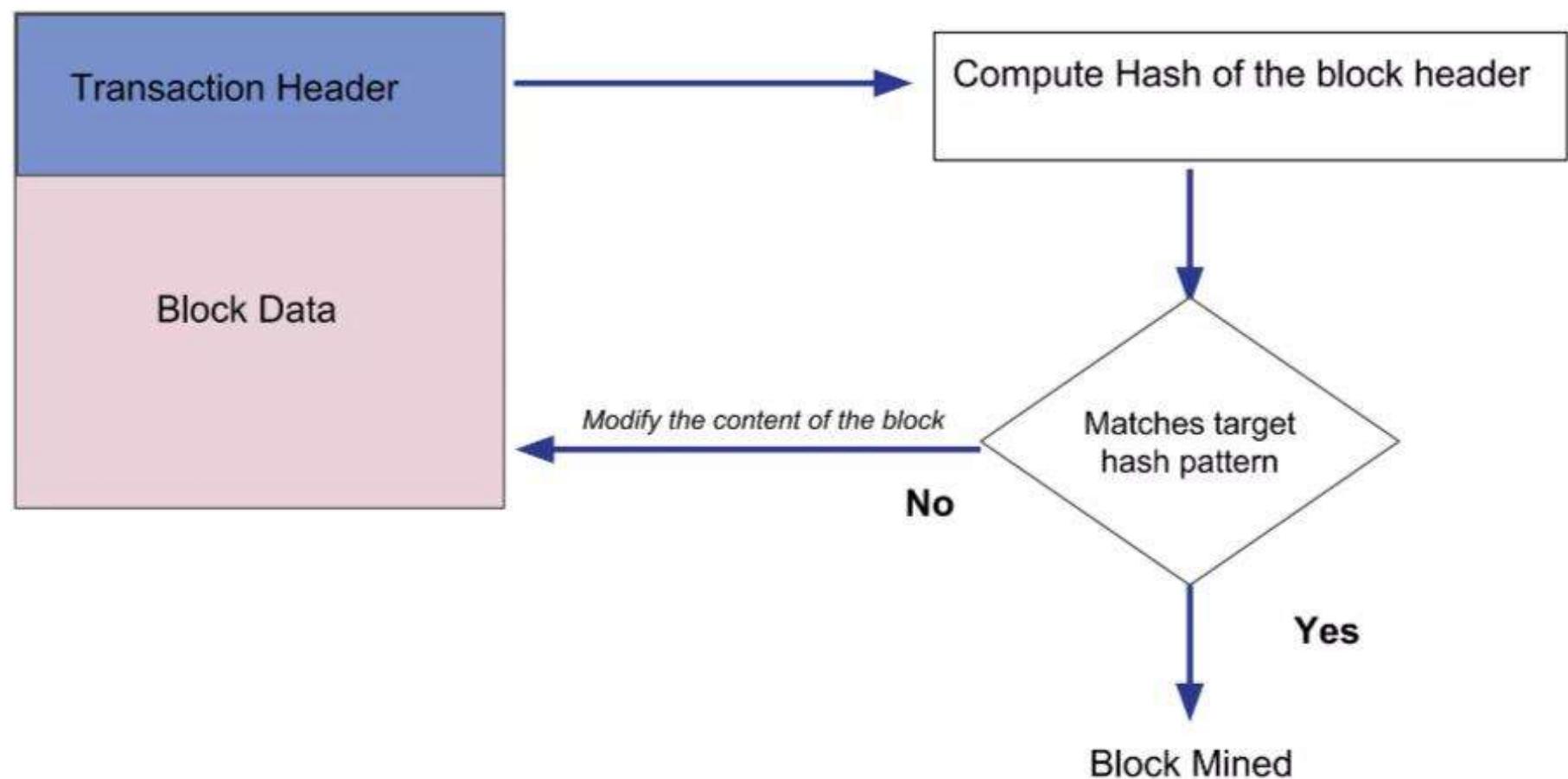


Let's start mining!

Kinh tế Bitcoin và việc tạo ra tiền tệ

- Bitcoin được "đúc" trong quá trình tạo ra mỗi khối với tỷ lệ cố định và giảm dần
- Người khai thác xác thực các giao dịch mới và ghi lại chúng trên sổ cái toàn cầu. Một khối được khai thác sau mỗi 10 phút.
- Các giao dịch là một phần của khối được coi là đã được xác nhận.
- Thợ đào cạnh tranh để giải một bài toán khó. Giải pháp cho bài toán này được gọi là *bằng chứng công việc* - bằng chứng cho thấy thợ đào đã bỏ ra nhiều công sức tính toán.
- Thuật ngữ *khai thác* được sử dụng để mô phỏng lợi nhuận giảm dần từ khai thác thực tế. Số lượng khai thác bitcoin tối đa (giao dịch coinbase) có thể thêm vào nguồn cung tiền giảm một nửa sau mỗi 4 năm (hoặc 210000 khối)
- Tháng 1 năm 2009: 50
- Tháng 11 năm 2012: 25
- Tháng 7 năm 2016: 12,5
- Tháng 5 năm 2020: 6,25
- Bitcoin không thể bị thổi phồng như việc in tiền của ngân hàng trung ương (và ngân hàng thương mại) - tạo ra tín dụng.
- Lạm phát là sự mất giá chậm nhưng không thể tránh khỏi của tiền tệ. Một loại thuế ẩn - trùng phạt những người tiết kiệm để cứu những người mắc nợ.
- Tiền giảm phát - tiền có sức mua cao hơn theo thời gian.

Mining Diagrammatically



Sự đồng thuận phi tập trung

Phát minh chính của Satoshi Nakamoto là cơ chế phi tập trung cho sự đồng thuận mới nổi

- Xuất hiện vì không có cuộc bầu cử hoặc thời điểm cố định khi sự đồng thuận xảy ra. Sự đồng thuận phát sinh thông qua tương tác không đồng bộ của hàng nghìn nút độc lập - tất cả đều tuân theo các quy tắc đơn giản.
- Bốn quá trình:
 1. Xác minh độc lập bởi mọi nút đầy đủ
 2. Tổng hợp độc lập các giao dịch thành các khối mới bằng các nút khai thác kết hợp với bằng chứng công việc
 3. Xác minh độc lập các khối mới và lắp ráp vào chuỗi khối
 4. Lựa chọn độc lập chuỗi có bằng chứng tính toán tích lũy nhiều nhất

Xác minh giao dịch độc lập

Phần mềm ví thu thập UTXO, cung cấp các tập lệnh mở khóa phù hợp và xây dựng các đầu ra mới. Giao dịch được gửi đến các nút lân cận để truyền bá trên toàn bộ mạng lưới bitcoin.

Trước khi truyền bá - mỗi nút xác minh giao dịch. Các giao dịch không hợp lệ sẽ không được truyền bá.

Danh sách kiểm tra:

- cú pháp giao dịch và cấu trúc dữ liệu phải chính xác
- không có danh sách đầu vào hoặc đầu ra nào là trống
- kích thước khối giao dịch nhỏ hơn MAX_BLOCK_SIZE
- mỗi đầu ra phải lớn hơn DUST và nhỏ hơn 21m
- Không có đầu vào nào có 0 hoặc -1- giao dịch coinbase không được chuyển tiếp
- NLockime bằng INT_MAX. NLocktime và nSequence được thỏa mãn theo MedianTimePast
- Kích thước chuyển đổi tính bằng byte là ≥ 100 byte
- ...v.v.

Khai thác các nút

Cuộc cạnh tranh giữa những người khai thác kết thúc hiệu quả bằng việc truyền bá một khối mới đóng vai trò như một thông báo về người chiến thắng.

Đối với những người khai thác, việc nhận được một khối mới hợp lệ có nghĩa là người khác đã chiến thắng cuộc thi và họ đã thua.

Tuy nhiên, kết thúc một vòng thi cũng là sự khởi đầu của vòng tiếp theo

Tổng hợp các giao dịch thành các khối

- Sau khi xác thực các giao dịch, chúng được thêm vào nhóm bộ nhớ. Các giao dịch chờ cho đến khi được đưa vào một khối.
- Các thợ đào sau khi nhận được khối đã xác nhận sẽ so sánh nó với tất cả các giao dịch trong nhóm bộ nhớ và xóa bất kỳ giao dịch nào được bao gồm trong khối mới
- . Bất kỳ giao dịch nào còn lại trong nhóm bộ nhớ đều chưa được xác nhận và đang chờ được ghi vào khối mới.. Thợ đào tạo ra một *khối ứng viên* .



Giao dịch Coinbase

- Giao dịch đầu tiên trong bất kỳ khối nào là một giao dịch đặc biệt, được gọi là *giao dịch coinbase*
- Giao dịch này được xây dựng bởi nút và chứa *phần thưởng* của anh ấy cho nỗ lực khai thác
- Các giao dịch Coinbase không sử dụng utxo làm đầu vào.
- Chỉ có một đầu vào, được gọi là *coinbase*, tạo ra bitcoin từ hư không.
- Tổng số tiền thưởng khi khai thác một khối = Phần thưởng Coinbase + Phí giao dịch



Phần thưởng và Phí của Coinbase

- Các khoản phí được tính như sau:

Tổng phí = Tổng (Đầu vào) - Tổng (Đầu ra)

- Phần thưởng được tính toán dựa trên chiều cao khối, bắt đầu từ 50 bitcoin cho mỗi khối và giảm một nửa sau mỗi 210.000 khối.
- Demo:

```
CAmount GetBlockSubsidy(int nHeight, const Consensus::Params& consensusParams)
{
    int halvings = nHeight / consensusParams.nSubsidyHalvingInterval;
    // Force block reward to zero when right shift is undefined.
    if (halvings >= 64)
        return 0;

    CAmount nSubsidy = 50 * COIN;
    // Subsidy is cut in half every 210,000 blocks which will occur approximately every .
    nSubsidy >>= halvings;
    return nSubsidy;
}
```



Cấu trúc giao dịch Coinbase

Bảng 1. Cấu trúc của đầu vào giao dịch "bình thường"

Kích cỡ	Cánh đồng	Sự miêu tả
32 byte	Giao dịch băm	Con trỏ đến giao dịch chứa UTXO cần chi tiêu
4 byte	Chỉ số đầu ra	Số chỉ mục của UTXO được chi tiêu, số đầu tiên là 0
1–9 byte (VarInt)	Mở khóa-Kích thước tập lệnh	Mở khóa-Độ dài tập lệnh tính bằng byte, để theo dõi
Biến đổi	Mở khóa-Script	Một tập lệnh đáp ứng các điều kiện của tập lệnh khóa UTXO
4 byte	Số thứ tự	Thường được đặt thành 0xFFFFFFFF để từ chối BIP 125 và BIP 68

Bảng 2. Cấu trúc của đầu vào giao dịch coinbase

Kích cỡ	Cánh đồng	Sự miêu tả
32 byte	Giao dịch băm	Tất cả các bit đều bằng không: Không phải là tham chiếu băm giao dịch
4 byte	Chỉ số đầu ra	Tất cả các bit đều là số 1: 0xFFFFFFFF
1–9 byte (VarInt)	Kích thước dữ liệu Coinbase	Độ dài của dữ liệu coinbase, từ 2 đến 100 byte
Biến đổi	Dữ liệu Coinbase	Dữ liệu tùy ý được sử dụng cho các thẻ nonce và khai thác bổ sung. Trong các khối v2; phải bắt đầu bằng chiều cao khối
4 byte	Số thứ tự	Đặt thành 0xFFFFFFFF

Dữ liệu Coinbase

- Tóm tắt và giải thích về dữ liệu Coinbase
- **Dữ liệu Coinbase** là một phần đặc biệt trong mỗi khối của blockchain Bitcoin, được sử dụng để chứa thông tin bổ sung ngoài các giao dịch. Nó giống như một "lời nhắn" mà thợ đào gửi đi kèm với mỗi khối được tạo ra.
- **Điểm nổi bật của dữ liệu Coinbase:**
- **Thay thế scriptSig:** Thay vì sử dụng trường scriptSig như các giao dịch thông thường, các khối sử dụng trường dữ liệu coinbase.
- **Độ dài và nội dung linh hoạt:** Dữ liệu coinbase có thể chứa từ 2 đến 100 byte và nội dung bên trong có thể do thợ đào tự định nghĩa.
- **Thông tin bổ sung:** Thợ đào có thể sử dụng dữ liệu coinbase để:
 - **Đánh dấu phiên bản khối:** Chỉ ra phiên bản của khối, ví dụ như hỗ trợ các tính năng mới của Bitcoin.
 - **Đính kèm thông điệp:** Thêm các thông điệp văn bản, ngày tháng hoặc các thông tin khác.
 - **Xác định nhóm khai thác:** Sử dụng các chuỗi ký tự đặc biệt để xác định nhóm khai thác đang tạo ra khối.
 - **Nonce bổ sung:** Thêm các giá trị ngẫu nhiên để hỗ trợ quá trình tìm kiếm giải pháp cho bài toán băm.
- **Ví dụ:**
 - **Khối genesis:** Satoshi Nakamoto đã sử dụng dữ liệu coinbase để ghi lại một thông điệp báo chí về cuộc khủng hoảng tài chính năm 2009.
 - **Khối hỗ trợ P2SH:** Thợ đào sẽ thêm chuỗi "/P2SH/" vào dữ liệu coinbase để biểu thị sự hỗ trợ cho tính năng P2SH (Pay-to-Script-Hash).



Xây dựng tiêu đề khối

- Sáu trường:
 - phiên bản (4 byte)
 - băm khối trước đó (32 byte)
 - gốc merkle (32 byte)
 - dấu thời gian (4 byte)
 - mục tiêu (4 byte) - thuật toán bằng chứng công việc mục tiêu
 - nonce (4 byte) - bộ đếm được sử dụng cho thuật toán bằng chứng công việc
- //Nút thợ đào bỏ phiếu cho chuỗi hợp lệ có độ dài nhất bằng cách chọn băm khối trước đó
- Thêm giao dịch coinbase làm giao dịch đầu tiên và đảm bảo có số lượng nút lá chẵn trong cây để tạo ra gốc Merkle.
- Thợ đào sẽ thêm dấu thời gian Unix epoch (số giây tính từ năm 1970).
- Sau đó, mục tiêu được đặt ra - bằng chứng công việc cần thiết để làm cho nó trở thành một khối hợp lệ.
- Mã hóa mantissa-exponent - một byte dành cho số mũ và ba byte dành cho hệ số (mantissa).
- Nonce được khởi tạo về 0.
- Với các trường này đã được điền, quá trình đào mỏ có thể bắt đầu...
- Mục tiêu bây giờ là tìm một giá trị cho nonce sao cho kết quả của hàm băm tiêu đề khối nhỏ hơn mục tiêu đã đặt. Node đào sẽ phải thử hàng tỷ hoặc hàng nghìn tỷ nonce trước khi tìm được một nonce thỏa mãn yêu cầu.
- Quá trình đào là việc băm liên tục tiêu đề khối - thay đổi một tham số cho đến khi kết quả hàm băm khớp với mục tiêu cụ thể. Kết quả của hàm băm không thể xác định trước - hàm băm là một chiều. Vì vậy, bạn phải sử dụng phương pháp thử sai.
- Thay đổi nonce cho đến khi xuất hiện hàm băm mong muốn.



Khai thác khối và thuật toán proof-of-work

- Khai thác là quá trình băm tiêu đề khối nhiều lần, thay đổi một tham số, cho đến khi băm kết quả khớp với một mục tiêu cụ thể.
- Kết quả của hàm băm không thể được xác định trước, cũng không thể tạo ra một mẫu sẽ tạo ra một giá trị băm cụ thể

Thuật toán Proof-of-work

Đầu ra của SHA256 luôn là 256 bit

```
import hashlib

TEXT = "I am Satoshi Nakamoto"
line = TEXT.encode('utf-8')

result = hashlib.sha256(line).hexdigest()
print(result)
```



Khai thác khối và thuật toán proof-of-work

- Khai thác là quá trình băm tiêu đề khối nhiều lần, thay đổi một tham số, cho đến khi băm kết quả khớp với một mục tiêu cụ thể.
- Kết quả của hàm băm không thể được xác định trước, cũng không thể tạo ra một mẫu sẽ tạo ra một giá trị băm cụ thể
- **Thuật toán Proof-of-work**
- **Đầu ra của SHA256 luôn là 256 bit**



```
import hashlib

TEXT = "I am Satoshi Nakamoto"
line = TEXT.encode('utf-8')

result = hashlib.sha256(line).hexdigest()
print(result)
```



5d7c7ba21cbbcd75d14800b100252d5b428e5b1213d27c385bc141ca6b47989e

Continued

- Thay đổi input sẽ tạo ra những giá trị băm khác nhau

```
import hashlib

TEXT = "I am Satoshi Nakamoto"

for i in range(1, 21):
    line = (TEXT + str(i)).encode('utf-8')
    result = hashlib.sha256(line).hexdigest()
    print(f"Hash for '{TEXT}{i}': {result}")

→ Hash for 'I am Satoshi Nakamoto1': f7bc9a6304a4647bb41241a677b5345fe3cd30db882c8281cf24fbb7645b6240
Hash for 'I am Satoshi Nakamoto2': ea758a8134b115298a1583ffb80ae62939a2d086273ef5a7b14fbfe7fb8a799e
Hash for 'I am Satoshi Nakamoto3': bfa9779618ff072c903d773de30c99bd6e2fd70bb8f2cbb929400e0976a5c6f4
Hash for 'I am Satoshi Nakamoto4': bce8564de9a83c18c31944a66bde992ff1a77513f888e91c185bd08ab9c831d5
Hash for 'I am Satoshi Nakamoto5': eb362c3cf3479be0a97a20163589038e4dbead49f915e96e8f983f99efa3ef0a
Hash for 'I am Satoshi Nakamoto6': 4a2fd48e3be420d0d28e202360cfbab410beddeeb8ec07a669cd8928a8ba0e
Hash for 'I am Satoshi Nakamoto7': 790b5a1349a5f2b909bf74d0d166b17a333c7fd80c0f0eeabf29c4564ada8351
Hash for 'I am Satoshi Nakamoto8': 702c45e5b15aa54b625d68dd947f1597b1fa571d00ac6c3dedfa499f425e7369
Hash for 'I am Satoshi Nakamoto9': 7007cf7dd40f5e933cd89fff5b791ff0614d9c6017fbe831d63d392583564f74
Hash for 'I am Satoshi Nakamoto10': c2f38c81992f4614206a21537bd634af717896430ff1de6fc1ee44a949737705
Hash for 'I am Satoshi Nakamoto11': 7045da6ed8a914690f087690e1e8d662cf9e56f76b445d9dc99c68354c83c102
Hash for 'I am Satoshi Nakamoto12': 60f01db30c1a0d4cbce2b4b22e88b9b93f58f10555a8f0f4f5da97c3926981c0
Hash for 'I am Satoshi Nakamoto13': 0ebc56d59a34f5082aaef3d66b37a661696c2b618e62432727216ba9531041a5
Hash for 'I am Satoshi Nakamoto14': 27ead1ca85da66981fd9da01a8c6816f54cfa0d4834e68a3e2a5477e865164c4
Hash for 'I am Satoshi Nakamoto15': 394809fb809c5f83ce97ab554a2812cd901d3b164ae93492d5718e15006b1db2
Hash for 'I am Satoshi Nakamoto16': 8fa4992219df33f50834465d30474298a7d5ec7c7418e642ba6eae6a7b3785b7
Hash for 'I am Satoshi Nakamoto17': dca9b8b4f8d8e1521fa4eaa46f4f0cdf9ae0e6939477e1c6d89442b121b8a58e
Hash for 'I am Satoshi Nakamoto18': 9989a401b2a3a318b01e9ca9a22b0f39d82e48bb51e0d324aaa44ecaba836252
Hash for 'I am Satoshi Nakamoto19': cda56022ecb5b67b2bc93a2d764e75fc6ec6e6e79ff6c39e21d03b45aa5b303a
Hash for 'I am Satoshi Nakamoto20': 063dfa8201be30fcda257be61c64a4de6305fa937e80b037fe4e126fe03e85a5c
```



Continued...

- Để thử thách, hãy tìm một dữ liệu đầu vào tạo ra giá trị băm bắt đầu bằng 0.
- Xác suất là 1 trên 16. Một trong 16 giá trị thập lục phân có thể có.
- Vì vậy, chúng ta muốn tìm một giá trị nhỏ hơn một mục tiêu cụ thể:
0x1000000000000000...
- Giảm mục tiêu sẽ làm cho việc tìm kiếm giá trị băm trở nên khó khăn hơn.
- Vì SHA256 mang tính xác định - bản thân dữ liệu đầu vào cung cấp bằng chứng cho thấy một lượng công việc nhất định đã được thực hiện.
- Chỉ cần 1 lần băm để xác minh - cần nhiều lần thử và tìm ra kết quả đúng.
- Demo (python 3): <https://onlinegdb.com/cloNn5TOM>
- tăng độ khó lên 1 bit sẽ làm tăng gấp đôi thời gian tìm ra giải pháp



Mục tiêu đại diện

Điều chỉnh độ khó bằng cách retargeting

- Tại sao có thể điều chỉnh được và ai là người điều chỉnh nó?
- Các khối Bitcoin được tạo ra trung bình 10 phút một lần. Đây là nhịp đập của Bitcoin và hỗ trợ tần suất phát hành tiền tệ và tốc độ giải quyết giao dịch.
- Sức mạnh máy tính thay đổi theo thời gian (cũng như thợ đào buộc phải ngoại tuyến như ở Trung Quốc)
- Để giữ thời gian tạo khối ở mức 10 phút - độ khó khai thác phải được điều chỉnh
- Nó được thực hiện như thế nào trên mạng phi tập trung: Việc nhắm mục tiêu lại diễn ra tự động và độc lập trên mọi nút
- Cứ sau 2.016 khối, tất cả các nút sẽ nhắm mục tiêu lại vào Proof-of-Work.
- Phương trình nhắm mục tiêu lại đo thời gian cần thiết để tìm 2.016 khối cuối cùng và so sánh với thời gian dự kiến là 20.160 phút (2.016 khối nhân với khoảng thời gian khối mong muốn là 10 phút). Tỷ lệ giữa khoảng thời gian thực tế và khoảng thời gian mong muốn được tính toán và điều chỉnh tương ứng (tăng hoặc giảm) được thực hiện theo mục tiêu. Nói một cách đơn giản: Nếu mạng tìm thấy các khối nhanh hơn sau mỗi 10 phút, độ khó sẽ tăng (mục tiêu giảm). Nếu phát hiện khối chậm hơn dự kiến, độ khó sẽ giảm (mục tiêu tăng).
- $\text{New Target} = \text{Old Target} * (\text{Actual Time of Last 2016 Blocks} / 20160 \text{ minutes})$
-

Continued...

- Có một lỗi chênh lệch 1% - dựa trên 2015 khối trước đó chứ không phải 2016. Độ khó tăng 0,05%.
- Hệ số điều chỉnh đạt cực đại tại⁴
- Sức mạnh băm không phụ thuộc vào giao dịch và việc áp dụng.
- Độ khó của việc khai thác liên quan đến chi phí điện. Giá của 1 kilowatt giờ là quan trọng.
- Nonce do thợ đào nhập vào - tạo ra một khối băm thấp hơn mục tiêu. Sau đó, khối được truyền đến các đồng nghiệp. Những thợ đào không trung thực sẽ bị từ chối khối của họ.
- “Chuỗi chính” tại bất kỳ thời điểm nào là bất kỳ chuỗi khối hợp lệ nào có Bằng chứng công việc tích lũy nhiều nhất liên quan đến nó. Trong hầu hết các trường hợp, đây cũng là chuỗi có nhiều khối nhất trong đó
- Trong một nhánh blockchain - chuỗi có bằng chứng công việc tích lũy nhiều nhất sẽ được chọn.
- Bằng cách chọn chuỗi hợp lệ có công việc tích lũy lớn nhất, cuối cùng tất cả các nút đều đạt được sự đồng thuận trên toàn mạng.

Khai thác khối thành công

- **Khai thác khối thành công** là quá trình trong đó một thợ đào (miner) tìm được giá trị nonce phù hợp để tạo ra một hàm băm (hash) của tiêu đề khối (block header) thỏa mãn điều kiện đặt ra, thường là nhỏ hơn một giá trị mục tiêu (target) cụ thể. Đây là cơ chế chính để thêm các khối mới vào blockchain trong các hệ thống như Bitcoin.
 - **Ví dụ:**

Khi được chèn vào tiêu đề khối, nonce 924,591,752 tạo ra băm khối là:

0000000000000001b6b9a13b095e96db41c4a928b97ef2d944a9b31b2c7bdc4

thấp hơn mục tiêu:

Xác thực một khối mới

- Bước thứ ba trong cơ chế đồng thuận của bitcoin là xác thực độc lập mỗi khối mới bởi mọi nút trên mạng
- Khi khối mới được giải quyết di chuyển qua mạng, mỗi nút thực hiện một loạt các bài kiểm tra để xác thực khối đó trước khi truyền đến các nút ngang hàng
- Xác thực độc lập cũng đảm bảo rằng những thợ đào hành động trung thực sẽ đưa khối của họ vào blockchain, do đó kiếm được phần thưởng.
- Các tiêu chí để xác thực khối trong ứng dụng khách Bitcoin Core trong các hàm CheckBlock và CheckBlockHeader và bao gồm:
 - Cấu trúc dữ liệu khối có giá trị về mặt cú pháp
 - Băm tiêu đề khối nhỏ hơn mục tiêu (thực thi Bằng chứng công việc)
 - Dấu thời gian khối ít hơn hai giờ trong tương lai (cho phép có lỗi thời gian)
 - Kích thước khối nằm trong giới hạn cho phép
 - Giao dịch đầu tiên (và chỉ giao dịch đầu tiên) là giao dịch coinbase



Lắp ráp và lựa chọn chuỗi

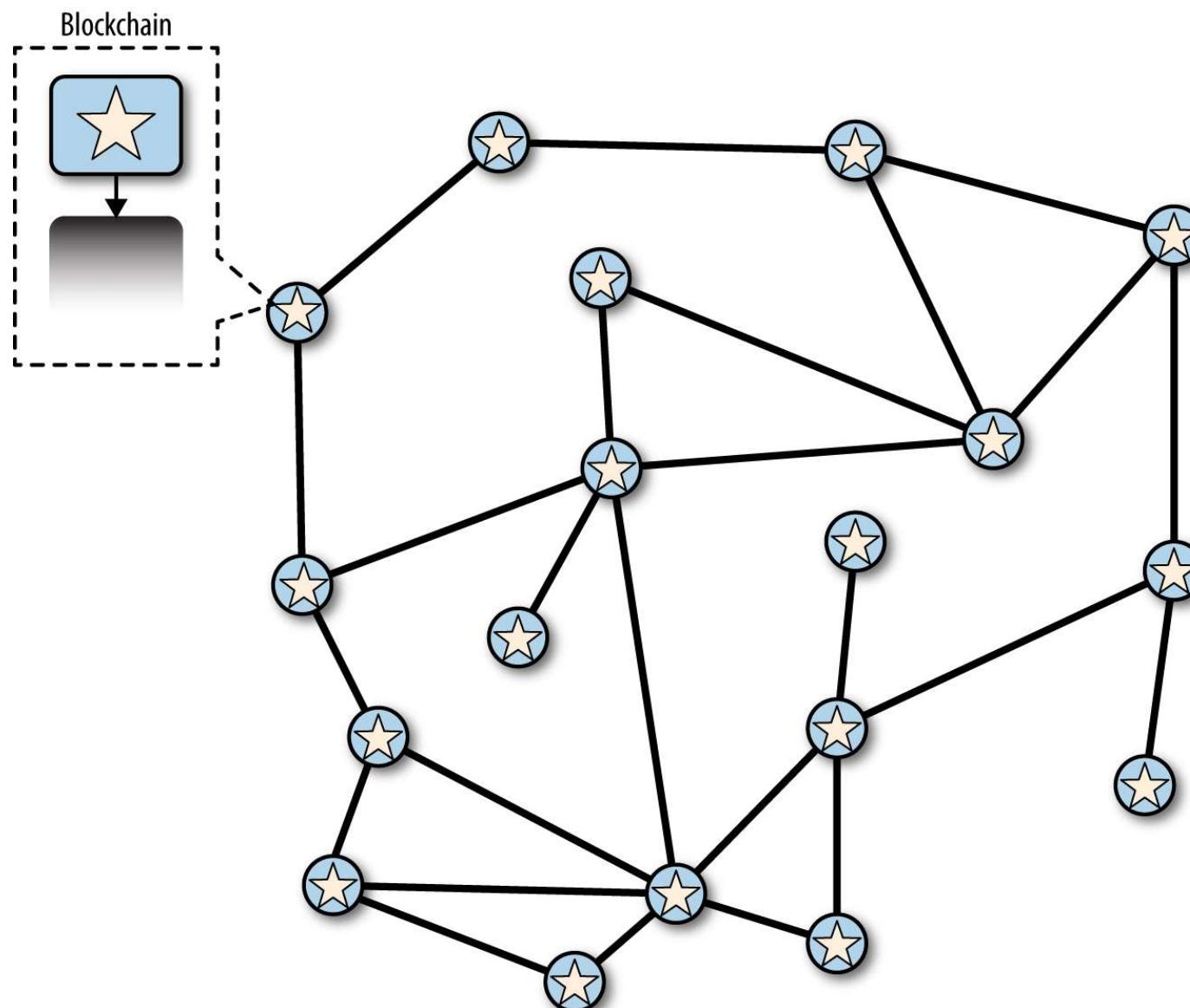
- Bước cuối cùng trong cơ chế đồng thuận phi tập trung
- Lựa chọn chuỗi có **Proof-of-work** cao nhất, không cần phải là chuỗi dài nhất
- Các nút duy trì ba loại khối: khối thuộc chuỗi chính, khối tạo thành nhánh từ chuỗi chính (chuỗi thứ cấp), và khối mồ côi (không có cha mẹ).
- Khi nhận khối mới, nút kiểm tra tham chiếu đến khối cha mẹ để tìm nơi kết nối. Nếu khối mở rộng chuỗi chính, nó sẽ được thêm vào chuỗi đó. Nếu mở rộng chuỗi thứ cấp và chuỗi này có PoW lớn hơn chuỗi chính, chuỗi thứ cấp sẽ trở thành chuỗi chính mới. Nếu khối không tìm thấy cha mẹ, nó sẽ được lưu dưới dạng khối mồ côi cho đến khi cha mẹ của nó được tìm thấy.
- **Cách giải quyết sự khác biệt giữa các chuỗi cạnh tranh (ngã ba) bằng cách lựa chọn độc lập chuỗi có công việc tích lũy lớn nhất.**



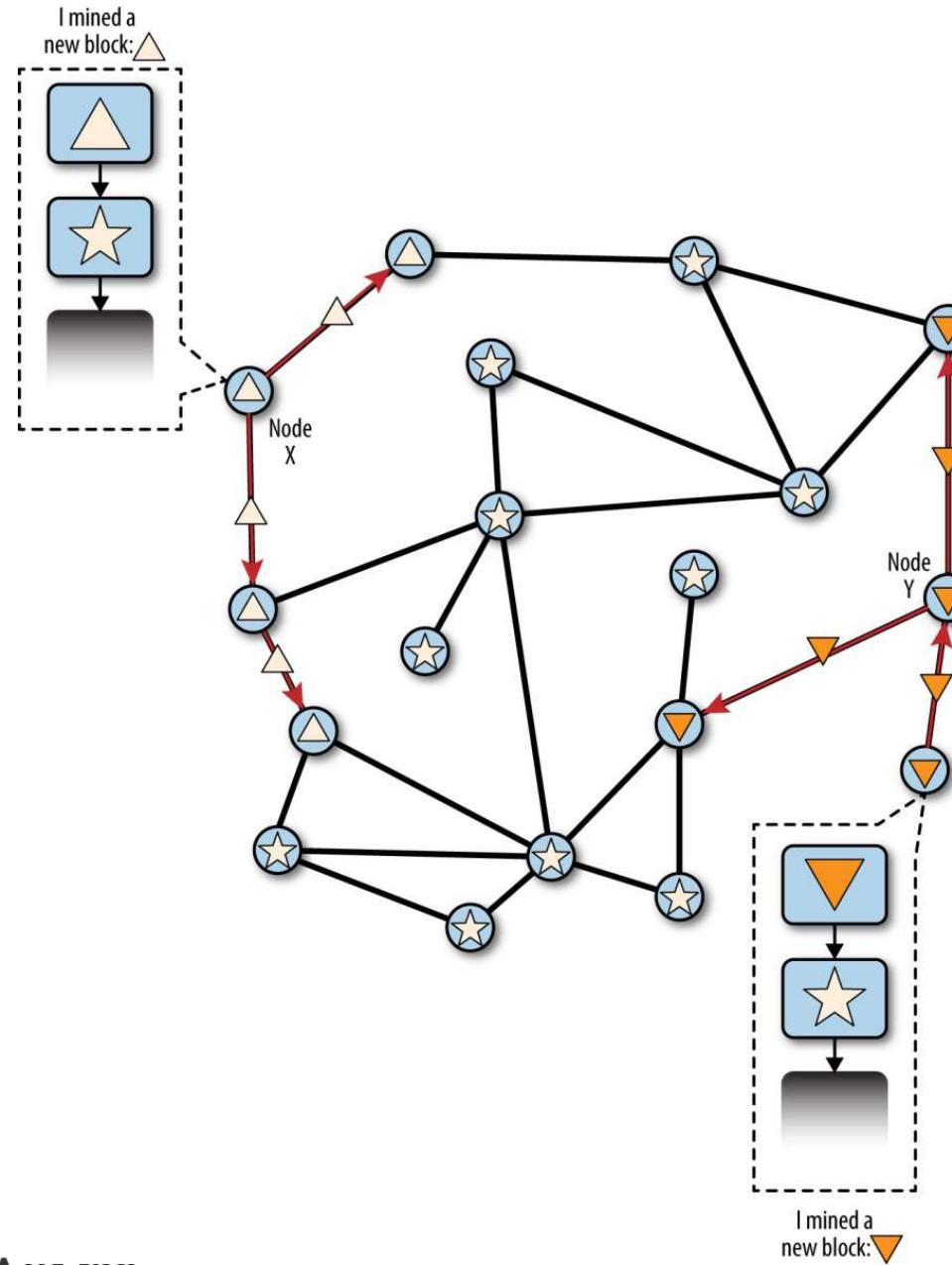
Phân nhánh Blockchain

- Hiện tượng phân nhánh (fork) trong blockchain, xuất hiện khi các nút trong mạng có góc nhìn khác nhau về chuỗi khối do sự chậm trễ trong truyền tải.
- Để giải quyết, mỗi nút chọn và mở rộng chuỗi có tổng công việc tích lũy lớn nhất, gọi là chuỗi dài nhất hoặc chuỗi tích lũy công việc lớn nhất.
- Các nhánh xuất hiện do sự không nhất quán tạm thời giữa các phiên bản blockchain, nhưng mạng sẽ hội tụ lại khi nhiều khối hơn được thêm vào một trong các nhánh
- Tiến trình phân nhánh (fork) trên toàn mạng blockchain được biểu diễn như sau:

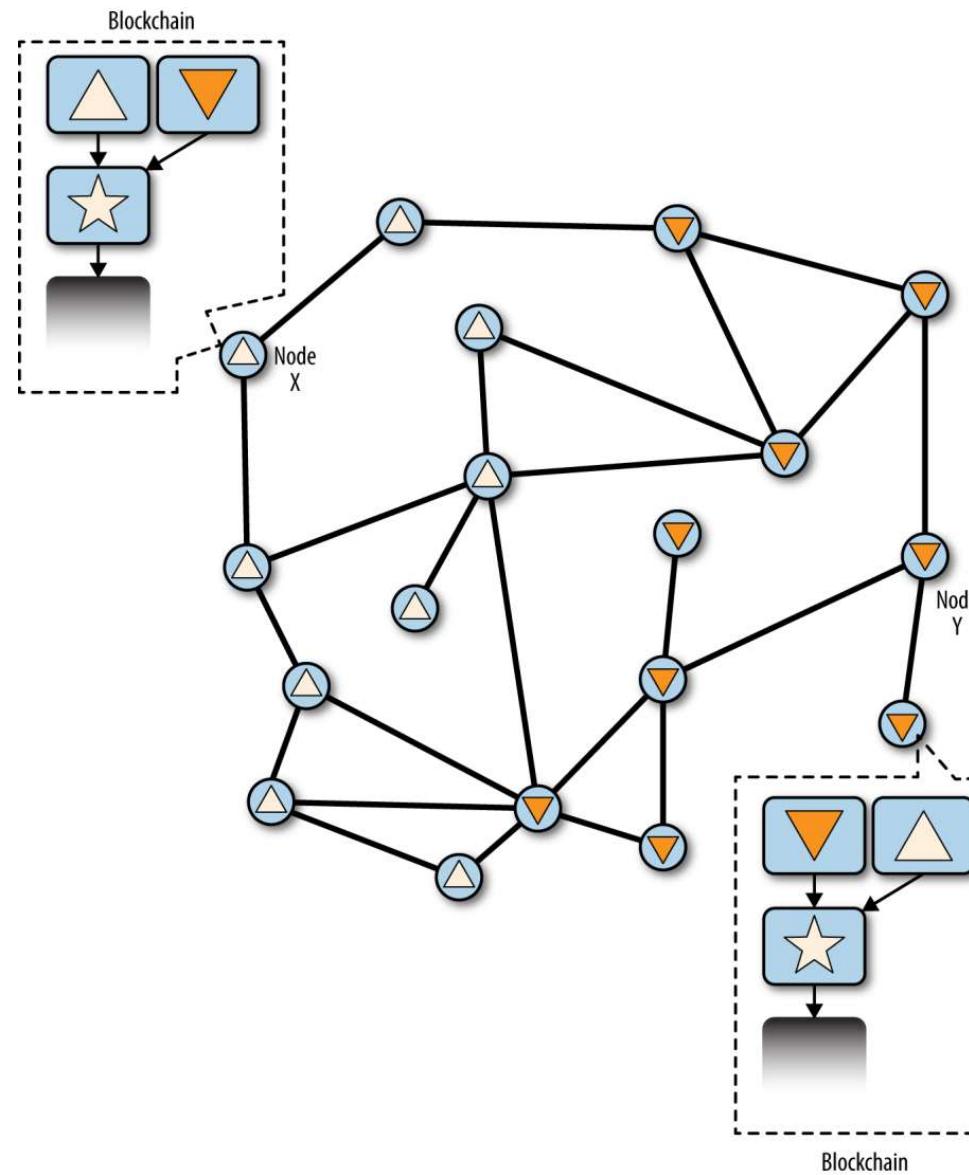
Trước khi phân nhánh, tất cả các nút đều có cùng góc nhìn



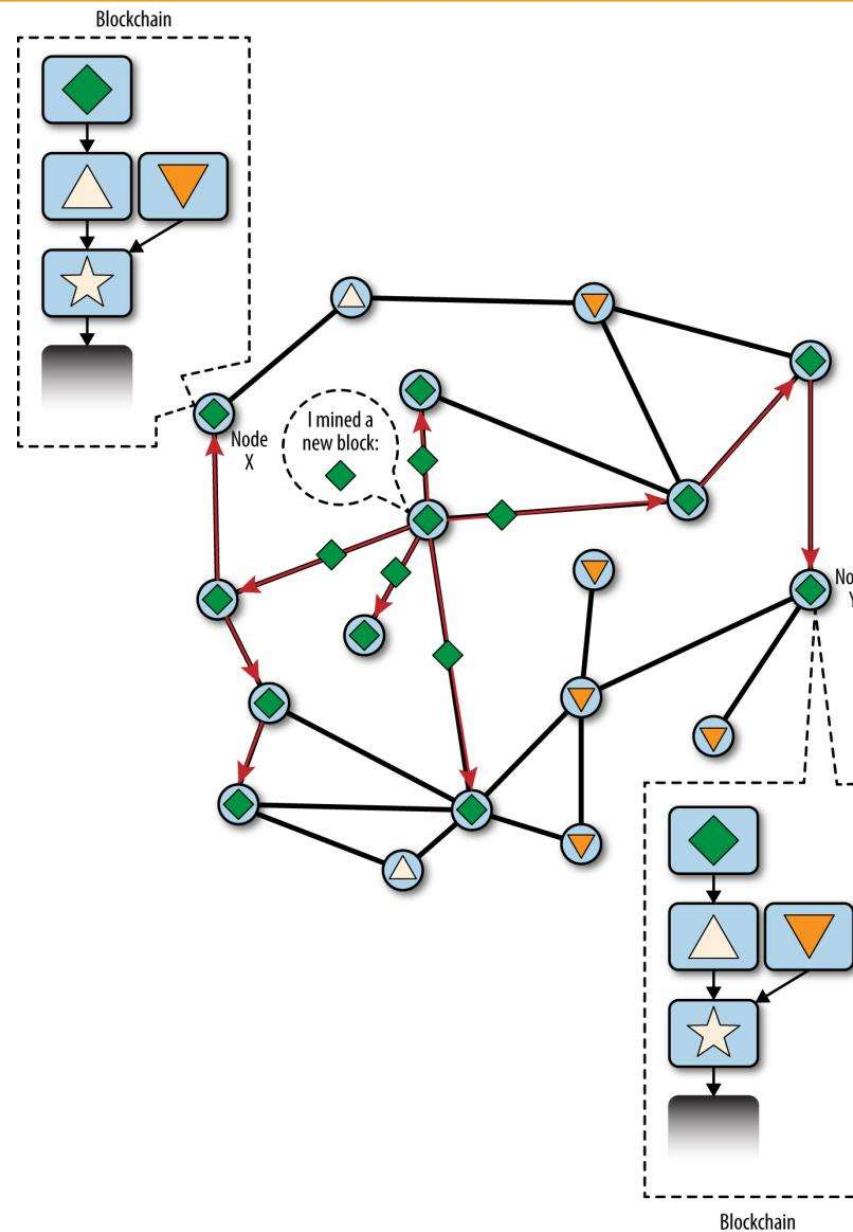
Hình ảnh trực quan của sự kiện phân nhánh blockchain: hai khối được tìm thấy cùng lúc



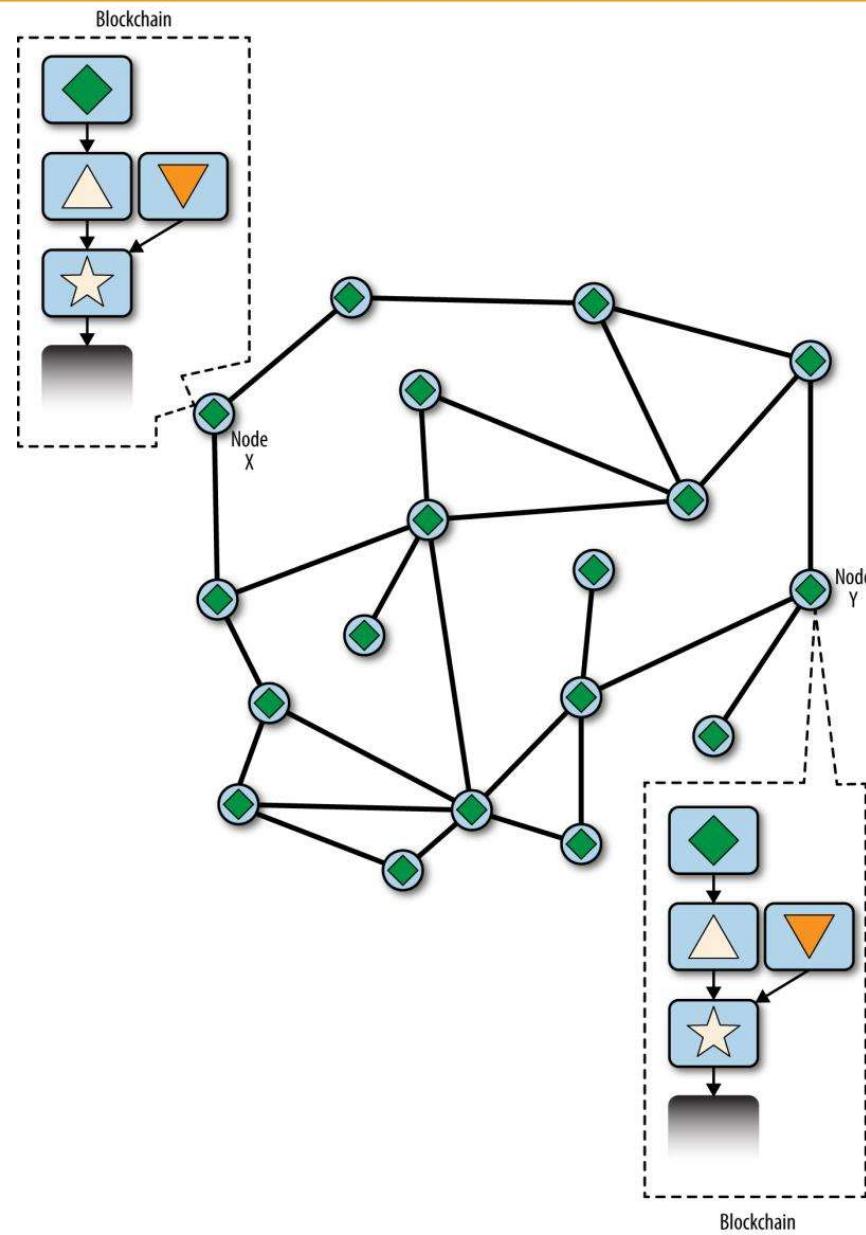
Hình ảnh trực quan của sự kiện phân nhánh blockchain: hai khối lan truyền, chia tách mạng



Hình ảnh trực quan của sự kiện phân nhánh blockchain: một khối mới mở rộng một phân nhánh, tái hội tụ mạng



Hình ảnh trực quan của sự kiện phân nhánh blockchain: mạng hội tụ lại trên một chuỗi dài nhất mới









HUST

THANK YOU !