**A triangle with words in it

Description automatically generated1 CÔNG NGHỆ CHUỖI KHỐI**

**CSDL tập trung và phân tán**

Cơ sở dữ liệu tập trung (**CDB** – *Centralized Data Base*) là cơ sở dữ liệu được định vị, lưu trữ và ***duy trì ở một vị trí duy nhất***. Vị trí này thường là máy tính trung tâm hoặc hệ thống cơ sở dữ liệu. Trong hầu hết các trường hợp, cơ sở dữ liệu tập trung sẽ được sử dụng bởi một cơ quan hay một tổ chức. Người dùng truy cập cơ sở dữ liệu tập trung thông qua mạng máy tính sau khi đã được cấp quyền truy cập vào hệ thống trung tâm, để tìm kiếm (*search*) hay duy trì (*maintain*) cơ sở dữ liệu đó.

Cơ sở dữ liệu phân tán (**DDB** – *Distributed Data Base*) là một loại cơ sở dữ liệu bao gồm nhiều cơ sở dữ liệu được kết nối với nhau và trải rộng trên các vị trí vật lý khác nhau. Dữ liệu được lưu trữ ở nhiều vị trí vật lý khác nhau có thể được quản lý độc lập với các vị trí vật lý khác.

Cụm từ *duy trì ở một vị trí duy nhất* trong khái niệm **CDB** hàm ý rằng ngay cả CSDL được lưu trữ phân tán ở nhiều vị trí vật lý hay địa lý khác nhau thì việc ***duy trì hay rộng hơn là giám sát vẫn được thực hiện ở một vị trí hay một người có quyền cao nhất***. Theo đó, ta vẫn có thể xem một DDB là CDB khi việc duy trì tập trung trong ‘một’ người. Ngược lại, ta gọi là CSDL phi-tập trung.

**CSDL phi-tập trung**

Cơ sở dữ liệu phi-tập trung (*Decentralized Data Base*), chính xác phải gọi là dữ liệu phi tập trung (*decentralized data*) là dữ liệu được tạo (*create*), lưu trữ (*storing*) và truy xuất (*access*) một cách độc lập từ nhiều vị trí khác nhau và không bị giám sát bởi bất kỳ tổ chức hay cá nhân nào.

Như vậy, CSDL phi-trập trung là cả lưu trữ lẫn xử lý phân tán. Điều này dẫn đến những thách thức trong việc duy trì CSDL chung, khi dữ liệu được phát triển thành CSDL – **Decentralized Database**.

**CSDL tập trung hay CSDL phi-tập trung?**

Cả CSDL tập trung lẫn phi-tập trung đều có những ưu thế và bất lợi riêng của chúng. Câu hỏi tự nhiên đặt ra là chọn CSDL nào: tập trung hay phi-tập trung?

Để trả lời, hãy xem hình vẽ minh họa về 2 tiêu chí quan trọng khi xem xét xây dựng một hệ thống thông tin: (**i**) hiệu năng (truy vấn) và (**ii**) bảo mật (security). Trước hết, cần lưu ý CSDL được tạo lập cho mục đích tìm kiếm là chính. Các thao tác khác như thêm (insert/append/add), xóa (delete/remove), hay cập nhật (edit) CSDL có thể không đòi hỏi phải xử lý cấp bách.

***Về tiêu chí (i): hiệu năng***. Do CSDL tập trung được lưu trữ và xử lý tại một điểm nên hiệu năng tìm kiếm cũng như duy trì CSDL là dễ dàng và hiệu quả. Truy vấn với CSDL tập trung đảm bảo tiêu chí hiệu năng. Ngược lại, với CSDL phi-tập trung, do dữ liệu được lưu trữ tại nhiều điểm khác nhau, CSDL chung chỉ là ý niệm với các thể hiện nằm ở các vị trí (vật lý cũng như địa lý) khác nhau nên việc đồng bộ (*synchronize*) để duy trì cho CSDL nhất quán là thách thức cho tiêu chí hiệu năng.

***Về tiêu chí (ii): bảo mật dữ liệu.*** Do tính tập, CSDL tập chỉ có một điểm tấn công vì thế dữ liệu không thực sự an toàn-bảo mật, đặc biệt, người có quyền cao nhất (*admin*) có toàn quyền và có thể làm sai lệch dữ liệu. Ngược lại, CSDL phi-tập trung, để tấn công, kẻ tấn công phải thay đổi được dữ liệu ở mọi điểm lưu trữ, mà sẽ bất khả thi nếu số điểm lưu trữ đủ nhiều.

|  |  |
| --- | --- |
| **Centralized Database** | **Decentralized Database** |
|  |  |

Tóm lại, việc chọn CSDL tập trung hay phi-tập trung cần phải được xem xét cẩn thận và tùy thuộc ứng dụng của hệ CSDL cần.

**CÔNG NGHỆ SỔ CÁI PHI TẬP TRUNG**

**Sổ cái.**

Thu nhỏ lại hoạt động của xã hội như những giao dịch (*transaction*) như trao đổi, vay mượn, buôn bán,… trực tiếp giữa các cá nhân trong công đồng với nhau. Để đảm bảo có vay thì phải trả, có bán thì mới mua,… các hoạt động cần có người làm chứng, nói cách khác, mọi giao dịch cần minh bạch, không có gì khuất tất

**Sổ cái phi tập trung**

Để đảm bảo sổ cái không bị phá hủy, vì một lý do nào đó như “chúa chổm” có thể hủy sổ cái để từ đó không còn bằng chứng nào chứng ninh hắn nợi-chúa-chổm. Vấn đề này có thể giải được bằng cách “phi tập trung sổ cái – **Decentralized Ledger**”, theo đó, sổ cái sẽ được giao cho n người giữ, mỗi người tự sao chép và cập nhật sổ cái mỗi khi có một giao dịch công khai nào xảy ra. Bằng cách đó, các cá nhân trao đổi tài sản với nhau chỉ phải công bố giao dịch của mình cho mọi người thấy. Những người đang giữ bản sao luận án sẽ xác minh (validate) các giao dịch đó, và nếu hợp lệ, sẽ ghi giao dịch hợp lệ (validated transcaction) vào sổ cái mình đang giữ. Một sổ cái phi tập trung như thế đảm bảo 3 tính chất sau cho mọi giao dịch công khai:

(•) Minh bạch: mọi giao dịch được thực hiện công khai với các nhân chứng là những cá nhân đạng giữ sổ cái.

(•) Toàn vẹn: không ai có thể sửa đổi nội dung của giao dịch đã được ghi xuống vì không thể thông đồng được với mọi người giữ sổ cái.

(•) An toàn: sổ cái không bị phá hủy (nếu còn lại một vài người đang giữ sổ cái).

Công nghệ hiện thực hóa khái niệm sổ cái thỏa 3 tính chất trên gọi là **công-nghệ-sổ-cái-phi-tập-trung**.

***Đồng bộ dữ liệu***

Tính nhất quán của CSDL chung được đảm bảo qua các thuật toán được biết với tên thuật-giải-đồng-thuận (*consensus algorithm*) mà được xây dựng dựa trên nguyên tắc đồng-thuận-số-đông, theo đó, block mới sẽ được thêm vào chain dài nhất hiện tại.

Một cách tổng quát, blockchain là một CSDL phi-tập trung (decentralized database).

Có 2 loại nút trong mạng chuỗi khối là nút sử dụng (*user node*) và nút duy trì (*miner node*) dữ liệu của CSDL blockchain.

***Nút sử dụng dữ liệu chuỗi khối – user node***

Là bất kỳ thiết bị nào có thể kết nối mạng để thực hiện các giao dịch liên quan đến thông tin được lưu giữ trong CSDL blockchain. Để giao dịch các dữ liệu của riêng mình, user node phải có khóa sử dụng tài khoản, như mã nhận dạng cá nhân (*Pin Code – Personal Identification Number*) được dùng để sử dụng thẻ ATM của ngân hàng. Khóa sử dụng tài khoản và số tài khoản có một quan hệ toán học chặt chẽ. Thực chất chúng là cặp khóa gốm khóa cá nhân (*private key*) và khóa công khai (*public key*) của một hệ mã công khai hiện đại (advanced public key cryptosystem), và địa chỉ (*address*) được phái sinh từ khóa công khai (nhiều mạng chuỗi khối, còn cho phép sử dụng trực tiếp khóa công khai làm địa chỉ tài khoản).

***Nút duy trì mạng chuỗi khối – miner nodes***

Là những máy chủ (*server*) đủ mạnh để có thể lưu giữ toàn bộ CSDL chuỗi khối, từ khối đầu tiên đến khối mới nhất và còn tiếp tục cập nhật thêm các khối tiếp nữa. Do tính chất bất biến của dữ liệu lưu trữ trong mạng chuỗi khối, các khối chỉ được thêm vào chuỗi chứ không thể xóa hay sủa đổi. Hơn nữa, chỉ các khối hợp lệ (*validated block*) mới được gắn thêm (*chain*) vào chuỗi. Một Các miner nodes có nhiệm vụ kiểm tra (*verify*) các giao dịch hợp lệ tập hợp trong khối hợp lệ và tranh đua cùng các miner nodes khác trong một trò chơi (*game*) được thiết kế riêng cho từng chuỗi khối để trở thành nút tạo được khối mới nhanh nhất. Khối mới này sẽ được gắn vào chuỗi khối và cuộc đua tạo khối mới lại tiếp tục.

Miner node cũng có thể có thông tin riêng của nó. Nghĩa là miner node hàm ý cũng là user node khi tham gia các giao dịch trao đổi thông tin của nó với các nút mạng khác trong cùng một mạng chuỗi khối.

**Các công nghệ cốt lõi của chuỗi khối**

Như ta đã thấy, chuỗi khối thực chất là một CSDL phi-tập trung được lưu trữ trên nhiều nút mạng trong mạng chuỗi khối (*blockchain network*). CSDL này chỉ được phép gắn thêm vào, mọi thao tác có thể làm thay đổi dữ liệu đã lưu như sửa đổi (*edit*) hay xóa (*delete*) đều không được phép. Nguyên do là khi một miner node thay đổi dữ liệu nó đã lưu đồng nghĩa nó tự loại mình ra khỏi mạng các nút duy trì chuỗi khối và chỉ có thể là nút sử dụng.

Dữ liệu trong CSDL chuỗi khối phải đảm bảo tính toàn vẹn (*integrity*).

Tính toàn vẹn của dữ liệu trong CSDL chuỗi khối được đảm bảo bởi công nghệ mật mã hiện đại bao gồm mã chứng thực thông điệp- và khóa sử dụng thông tin tài khoản.

Mọi dữ liệu được băm bằng một hàm băm mật mã (*cryptographic hash function*) và được ký (*sign*) bằng khóa cá nhân của người tạo ra dữ liệu đó.

CSDL chuỗi khối chỉ cho phép xem (*view*) và thêm các khối hợp lệ vào.

Một khối hợp lệ tạo là khối được bởi một miner node chiên thắng (*winner*) trong một trò chơi (*game*) được thiết kế riêng cho từng mạng chuỗi khối.

Như vậy, 3 lý thuyết cũng như công nghệ nền tảng làm trụ cột cho chuỗi khối gồm: lý thuyết trò chơi – mật mã hiện đại – mạng ngang hàng.

***Lý thuyết trò chơi***

Lý thuyết trò chơi là khoa học nghiên cứu các mô hình toán học về tương tác chiến lược giữa các tác nhân. Lý thuyết trò chơi có ứng dụng trong mọi lĩnh vực khoa học xã hội, cũng như logic, khoa học hệ thống và khoa học máy tính. Các khái niệm về lý thuyết trò chơi cũng được sử dụng rộng rãi trong kinh tế học. Lý thuyết trò chơi tiên tiến được áp dụng cho nhiều mối quan hệ hành vi và hiện tại được xem như một thuật ngữ chung cho khoa học về việc ra quyết định hợp lý ở con người, động vật cũng như máy tính.

Lý thuyết trò chơi được phát triển rộng rãi vào những năm 1950 và đã được áp dụng cho các quá trình tiến hóa vào những năm 1970, mặc dù những khái niệm tương tự đã có từ những năm 1930. Lý thuyết trò chơi đã được thừa nhận rộng rãi như một công cụ quan trọng trong nhiều lĩnh vực.

Tài liệu này không nghiên cứu nhiều về lý thuyết trò chơi cũng như công nghệ mạng ngang hàng mà tập trung chủ yếu vào công nghệ mật mã hiện đại.

**Các hệ mã hiện đại (*advanced cryptosystem*) có các đặc tính sau:**

**Tính bảo mật**: Thông tin chỉ có thể được truy cập bởi người hợp pháp và không người nào khác có thể truy cập thông tin mật này.

**Tính toàn vẹn**: Thông tin không thể được sửa đổi trong quá trình lưu trữ hoặc chuyển tiếp giữa người gửi và người nhận hợp pháp mà không phát sinh thông tin bổ sung.

**Chống từ chối**: Người tạo/người gửi thông tin không thể phủ nhận thông tin của mình đã gửi.

**Xác thực**: Danh tính của người gửi và người nhận được xác nhận. Cũng như điểm đến/nguồn gốc của thông tin được xác nhận.

Nói chung có ba loại mật mã: mã đối xứng (*symmetric cryptosystem*), hàm băm mật mã hay gọi tắt là hàm băm (*hash function*), và mã bất đối xứng (*asymmetric cryptosystem*).

**2 MÃ HÓA ỨNG DỤNG A triangle with red lines and black text

Description automatically generated**

**Phân loại các hệ mã**

***Bí mật hoàn toàn – Hàm mã hóa là ánh xạ đơn giản***

Như trong ví dụ 1, cho dù có bí mật khóa k thì chỉ bẳng cách thử (vét cạn) qua 26 trường hợp, bản rõ có thể được phục hồi từ bản mã được cho. Trong trường hợp nay, phải bí mật cả thuật toán (hàm mã hóa) lẫn khóa mã hóa.

***Công bố thuật toán và bí mật thông tin khóa – k ≡ k’***

Các ví dụ 2 và ví dụ 3 minh họa hệ mã này. Trong các hệ mã này, khóa bí mật là dãy số, mà là hoán vị của 10 chữ số, và hàm mã hóa chỉ là thay thế thông điệp m bằng hoán vị tương ứng σ(m) của nó. Trong ví dụ này, không gian phải vét cạn là 10! = 3628800 khả năng. Nếu hoán vị của 100 ký hiệu thì khả năng vét cạn 100! Trường hợp sẽ là không khả thi. Hệ mã dạng này được gọi là mã đối xứng (*symmetric cryptosystem*) khi khóa giải mã có thể được suy ra dễ ràng từ khóa mã hóa. Khóa mã hóa là bí mật chung giữa 2 đối tác trong hệ thống, vì thế phải được giữ bí mật, hệ mã có tính chất này còn được gọi là mã (khóa) bí mật (*secret key cryptosystem*). Rõ ràng khóa phải được quy ước trước giữa 2 đối tác, ta cũng có thể gọi là hệ mã quy ước khóa trước (*pre-share key cryptosystem*).

***Công bố thuật toán và một phần thông tin khóa – k ≠ k’***

k trong ví dụ 3 có tính chất đáng lưu ý là y = σ(x) = x3 % 11 và σ-1(y) = y11 % 11 = x. Như vậy, nếu ký hiêu e = k = 3 là khóa mã hóa thì ta có thể dùng d = k’ = 7 để giải mã. Theo đó, mã hóa bằng một khóa, khóa mã hóa e (*encryption key*), và giải bằng khóa khác, khóa giải mã (*decryption key*). Như vậy, sử dụng hệ mã này ta có thể công khai khoa mã hóa e, gọi là khóa công khai (*public key*), và giữ bí mật khóa giải mã d, gọi là khóa cá nhân (*private key*). Nên hệ mã dạng này được gọi là hệ mã (khóa) công khai (*public key cryptosystem*) hay còn có tên khác là hệ mã bất đối xứng (*asymmetric cryptosystem*) do tính khác biệt của các khóa mã và giải mã.

***Không sử dụng khóa – k = ‘’***

Ví dụ 4 minh họa cách hoạt động của hệ mã không sử dụng khóa. Hệ mã này đơn giản là biến đổi chuỗi thông điệp dài bất kỳ thành một chuỗi có chiều dài cố định. Hàm biến đổi E(.) này, giờ được ký hiệu là H(.), được gọi là hàm băm mật mã, và giá trị h = H(S) băm một chuỗi S được gọi là trị băm. Hàm này đơn giản chỉ là giấu thông tin (gốc) và thương được sử dụng trong các mục đích xác thực.

Các mạng xâu khối sử dụng chính hai loại hệ mã: mã công khai và hàm băm. Hệ mã bất đối xứng được sử dụng với mục đích bảo mật thông tin và không phải phổ biến.

**Nguyên lý thiết kế các hệ mã**

Về hình thức, ta thấy, để xây dựng một hệ mã, đơn giản chỉ là xác định 3 không gian: không gian các thông điệp (), không gian các bản mã (), không gian khóa (), và thiết kế thuật toán cho hàm khả nghịch E. Tuy nhiên, do yêu cầu bảo mật, các hệ mã phải được thiết kể để đảm bảo an toàn, bảo mật và hiệu quả khi dùng. Trong thực tế, các hệ mã thường được thiết kế theo nguyên lý Kerchkoff.

***Nguyên lý Kerchkoff***

. Hệ mã được chứng minh không thể bị thám mã về mặt toán học (*analyzing*), hơn nữa, phải không thể bị phá vỡ trong thực tế (*hacking*).

. Hệ mã nếu có bị kẻ xâm nhập khống chế cũng sẽ không bị bất kỳ sự xâm phạm nào vào hệ thống, và không gây bất kỳ sự bất tiện nào cho người dùng.

. Khóa mã hóa phải dễ trao đổi, dễ nhớ và có thể thay đổi.

. Bản mã có thể được truyền trên các kênh không an toàn.

. Thiết bị mã hóa và các văn bản phải có tính động và có thể sử dụng với người.

. Cuối cùng, hệ mã cần dễ sử dụng, không đòi hỏi phải có kiến thức và các quy tắc phức tạp phải tuân thủ.

**Về tính bảo mật**

Một hệ mã, trước hết phải bảo mật và kháng được mọi tấn công hệ thống (*hacking*) cũng như phân tích thuật toán (*analyzing*).

***Tấn công hệ mã***

Các cuộc tấn công thường được phân loại theo hành động được kẻ tấn công thực hiện. Theo đó, tấn công có thể là thụ động hoặc tấn công chủ động

*Tấn công thụ động*. Mục tiêu chính là giành được quyền truy cập (trái phép) vào thông tin.

*Tấn công tích cực*. Một cuộc tấn công tích cực liên quan đến việc thay đổi thông tin theo một cách nào đó qua việc tiến hành một số quy trình trên thông tin.

***Các giải định kẻ tấn công có thể có***

*Môi trường xung quanh hệ thống mật mã*. Kẻ tấn công có thể thâm nhập vào môi trường mà các thiết bị hay hệ mã đang hoạt động.

*Chi tiết về sơ đồ mã hóa*. Kẻ tấn công có được chi tiết về các lược đồ, các giao thức mật mã.

*Sự sẵn có của bản mã*. Kẻ tấn công giả thiết thu thập được không giới hạn các bản mã.

*Sự sẵn có của bản rõ và bản mã*. Kẻ tấn công được giải thiết thu thập được tập (vô hạn) các cặp bản rõ và bản mã.

Từ các giả định trên, các kỹ thuật tấn công có thể được tiến hành phù hợp.

***Phân loại tân công***

*Tấn công chỉ dựa vào bản mã*: **COA** – **Cipher text Only Attack**

Tấn công chỉ dựa vào bản mã (COA) là mô hình thám mã khi kẻ tấn công chỉ có một tập các bản mã. Mặc dù chỉ có các bản mã nhưng trong thực tế, kẻ tấn công có thể có một số tri thức về bản rõ. Chẳng hạn, kẻ tấn công có thể biết ngôn ngữ của bản rõ (tiếng Việt, tiếng Anh, …) hoặc/và biết được phân phối các ký tự trong bản rõ. Cũng vậy, dữ liệu cũng như các thông điệp trong các giao thức, trong nhiều hệ thống, thường là một phần của bản rõ và có thể đoán hoặc biết trước.

Tấn công thành công khi phục hồi được bản rõ tương ứng (tấn công một phần), hoặc phục hồi đươc khóa mã hóa (tấn công toàn phần). Thậm chí, thu thập được bất kỳ thông tin nào về bản rõ ngoài những gì kẻ tấn công đã biết vẫn được coi là thành công. Chẳng hạn, nhiều giao thức gửi bản mã liên tục nhằm duy trì trạng thái bảo mật đường truyền, việc phân biệt các tin nhắn thực với các tin nhắn rỗng có thể xem là thành công, thậm chí dự đoán được về sự tồn tại của một số thông điệp thực cũng sẽ tạo điều kiện thuận lợi cho việc phân tích lưu lượng truy cập.

*Tấn công biết bản rõ:* **KPA** *–* **Known Plaintext Attack**

Tấn công biết bản đã biết (KPA) là mô hình thám mã trong đó kẻ tấn công thu thập được tập các bản rõ và bản mã tương ứng. Tập dữ liệu nay có thể được sử dụng để phục hồi một số thông tin bí mật như khóa cá nhân bí mật và mã từ điển (code book).

*Tấn công chọn bản rõ*: **CPA – Chosen Plaintext Attack**

Tấn công chọn bản rõ (CPA) là mô hình thám mã, giả định rằng kẻ tấn công có thể thu thập được các bản mã của các bản gốc hắn muốn. Mục tiêu của tấn công này là lấy thông tin làm giảm tính bảo mật của sơ đồ mã hóa.

*Tấn công từ điển* – **Dictionary Attack**

Tấn công từ điển dựa vào việc thử tất cả các chuỗi trong danh sách được sắp xếp trước. Tấn công từ điển có thể sử dụng các danh sách có sẵn trên Internet chứa hàng trăm triệu mật khẩu được khôi phục từ các vụ vi phạm dữ liệu đã xảy ra. Ngoài ra có thể sử dụng phần mềm bẻ khóa áp dụng vào các danh sách đó để tạo ra các biến thể khác, như thay thế các chữ số cho các chữ cái. Tấn công từ điển thường thành công vì nhiều người có xu hướng chọn mật khẩu ngắn và là những từ thông thường hoặc mật khẩu thông dụng. Tấn công từ điển cũng có thể thành công vì nhiều kỹ thuật tạo mật khẩu thông dụng đều nằm trong danh sách có sẵn. Kết hợp với việc tạo mẫu của phần mềm bẻ khóa, khả năng thành công càng cao.

*Tấn công vét cạn*: **BFA – Brute Force Attack**

Tấn công vét cạn là tấn công mà kẻ tấn công kiểm tra một cách có hệ thống tất cả mật khẩu và cụm mật khẩu có thể có cho đến khi tìm thấy mật khẩu chính xác. Ngoài ra, kẻ tấn công có thể thử đoán khóa được tạo từ mật khẩu bằng cách sử dụng hàm phái sinh khóa.

Trên lý thuyết, tấn công vét cạn là tấn công thám mã có thể được áp dụng để thử giải mã bất kỳ bản mã nào. Tấn công như vậy được sử dụng khi không thể tận dụng các điểm yếu khác trong hệ thống mã hóa (nếu có).

Khi đoán mật khẩu, tấn công vét cạn hiệu quả khi kiểm tra tất cả mật khẩu ngắn, đối với mật khẩu dài, các phương pháp khác như tấn công từ điển hiệu quả hơn. Mật khẩu, cụm mật khẩu và khóa dài khó bị bẻ khóa hơn so với những mật khẩu ngắn hơn.

Tấn công vét cạn sẽ không hiệu quả với cách làm xáo trộn dữ liệu mã hóa khiến kẻ tấn công khó nhận ra khi nào mã đã bị bẻ khóa hoặc chí ít cũng khiến kẻ tấn công phải làm nhiều việc hơn để kiểm tra từng lần đoán. Tấn công vét cạn cũng được dùng để đánh giá độ án toàn của một hệ thống.

Tấn công vét cạn thực chất là áp dụng của tìm kiếm vét cạn. Kỹ thuật chung là liệt kê tất cả các ứng cử viên và kiểm tra từng ứng cử viên.

*Tấn công sinh nhật* – **Birth Attack**

Tấn công sinh nhật là một kiểu tấn công bằng cách khai thác tính chất toán học của bài toán sinh nhật trong lý thuyết xác suất. Tấn công sinh nhật dựa trên nguyên lý chuồng câu, theo đó, số chim nhiêu hơn số chuồng thì chắc chắn có chuồng có nhiều hơn 1 con, nghĩa là có đụng độ. Tấn công sinh nhật được dùng để tìm ra xung đột trong hàm băm. Có giả thuyết là máy tính lượng tử có thể thực hiện các cuộc tấn công sinh nhật hiệu quả. Tuy nhiên, giải thuyết này còn nhiều tranh cãi.

*Tân công qua trung gian*: **MIMA** – **Man in the Middle Attack**

Tấn công qua trung gian (MIM) là tấn công mạng trong đó kẻ tấn công bí mật chuyển tiếp và có thể thay đổi thông tin liên lạc giữa đối tác đang liên lạc mà họ không nhận ra có kẻ đứng giữa họ. Kẻ tấn công phải có khả năng chặn tất cả các tin nhắn truyền giữa hai nạn nhân và tiêm những tin nhắn mới vào. Việc này trong nhiều trường hợp có thể thực hiện dễ dàng. Chẳng hạn, trong khu vực phát song wifi không mã hóa, kẻ tấn công có thể phá vỡ tiến trình xác thực lẫn nhau.

*Tấn công kênh phụ*: **SCA – Side Chanel Attack**

Tấn công kênh phụ là tấn công nào trên thông tin bổ sung có thể được thu thập do cách thức cơ bản mà giao thức hoặc thuật toán máy tính được triển khai, không phải do sai sót trong thiết kế giao thức hoặc thuật toán hoặc do những sơ suất nhỏ nhưng có khả năng gây thiệt hại nghiêm trọng trong quá trình triển khai. Thông tin về thời gian, mức tiêu thụ điện năng, rò rỉ điện từ và âm thanh là những ví dụ về thông tin bổ sung có thể bị khai thác để tạo điều kiện cho các cuộc tấn công kênh phụ. Tấn công kênh phụ thường yêu cầu kiến thức kỹ thuật về hoạt động của hệ thống

*Tấn công định thời* – **Timming Attack**

Tấn công định thời là một dạng của tấn công kênh phụ trong đó kẻ tấn công thâm nhập hệ thống mật mã và phân tích thời gian thực hiện các thuật toán mật mã. Mọi thao tác logic trong máy tính đều cần có thời gian thực thi và thời gian này dài ngắn khác nhau phụ thuộc đầu vào; nếu đo chính xác được thời gian các thao tác, kẻ tấn công có thể suy ngược lại đầu vào. Việc phục hồi bí mật qua thông tin thời gian có thể dễ hơn nhiều việc sử dụng phương pháp thám mã đã biết ở trên. Đôi khi thông tin về thời gian cũng được kết hợp với thám mã để tăng tốc độ phục hồi thông tin.

Thông tin có thể bị rò rỉ qua việc đo thời gian cần thiết để phản hồi các truy vấn cụ thể. Tấn công theo thời gian có thể được áp dụng cho bất kỳ thuật toán nào có sự thay đổi thời gian phụ thuộc vào dữ liệu. Việc loại bỏ sự phụ thuộc vào thời gian là khó khăn trong một số thuật toán sử dụng các chỉ thị cấp thấp thường có thời gian thực hiện khác nhau.

Tấn công theo thời gian thường bị bỏ qua trong giai đoạn thiết kế vì chúng phụ thuộc quá nhiều vào việc triển khai và có thể được đưa vào một cách vô ý khi tối ưu hóa trình biên dịch. Vì thế, cần lưu ý tấn công khai thác thời gian thực thin gay trong quá trình thiết kế thuật toán, giao thức bảo mật.

*Tấn công phân tích năng lượng* – **Power Analysis Attacks**

Phân tích năng lượng là một hình thức tấn công kênh phụ trong đó kẻ tấn công nghiên cứu mức tiêu thụ năng lượng của thiết bị phần cứng mật mã. Tấn công này dựa vào các đặc tính vật lý cơ bản của thiết bị. Bằng cách đo dòng điện, có thể biết được một lượng nhỏ thông tin về dữ liệu đang được xử lý.

Phân tích năng lượng đơn giản (SPA – Simple Power Analysis) liên quan đến việc phân tích số liệu thống kê năng lượng hoặc biểu đồ hoạt động dòng điện theo thời gian. Phân tích năng lượng vi phân (DPA – Diferential Power Analysis) là một hình thức phân tích năng lượng hiện đại, cho phép kẻ tấn công tính toán các giá trị trung gian trong tính toán mật mã thông qua phân tích thống kê dữ liệu được thu thập từ nhiều hoạt động mật mã.

*Tấn công phân tích lỗi* – **Fault analysis Attacks**

Phân tích lỗi vi sai (DFA – Differential Fault Analysis) là một dạng tấn công kênh phụ vào các hê mã, cụ thể là thám mã. Nguyên tắc là tạo ra các lỗi cho các hoạt động mã hóa để tiết lộ trạng thái bên trong của chúng.

**HỆ MÃ KHÓA CÔNG KHAI**

**Mã khóa công khai**

Hệ mã khóa công khai, hay mã bất đối xứng, là các mã sử dụng cặp khóa có quan hệ toán học với nhau. Mỗi cặp khóa gồm một khóa công khai (*public key*) và một khóa cá nhân (*private key*). Các cặp khóa được tạo bằng thuật toán mật mã dựa trên một bài toán khó. Bài toán khó được sử dụng để tạo ra hàm một chiều. Tính bảo mật của mật mã khóa công khai phụ thuộc vào việc giữ bí mật khóa cá nhân; khóa công khai có thể được công bố mà không ảnh hưởng đến bảo mật của hệ mã.

Trong hệ mã khóa công khai, bất kỳ ai có khóa công khai đều có thể mã hóa tin nhắn, tạo ra bản mã, nhưng chỉ những người biết khóa cá nhân tương ứng mới có thể giải mã để phục hồi lại bản rõ.

**Chữ ký số**

Trong hệ thống chữ ký số, người gửi sử dụng khóa cá nhân cùng với thông điệp để tạo ra chữ ký. Bất kỳ ai có khóa công khai tương ứng đều có thể xác minh xem chữ ký có khớp với thông điệp hay không. Người giả mạo không biết khóa riêng sẽ không thể tìm ra bất kỳ cặp thông điệp/chữ ký nào vượt qua quá trình xác minh bằng khóa công khai.

**Ứng dụng của mã công khai**

. Ứng dụng rõ ràng nhất của hệ mã công khai là mã hóa thông tin liên lạc để cung cấp tính bảo mật - một tin nhắn mà người gửi mã hóa bằng khóa công khai của người nhận, chỉ có thể được giải mã bằng khóa nhân tương ứng của người nhận.

. Một ứng dụng khác của mã công khai là chữ ký số. Lược đồ chữ ký số có thể được sử dụng để xác thực người gửi. Hệ thống chống từ chối sử dụng chữ ký số để đảm quyền tác giả của một tài liệu hoặc thông tin liên lạc.

. Các ứng dụng khác được xây dựng trên nền tảng mã công khai có thể kể đến: tiền kỹ thuật số (cryptocurrency), trao đổi/thiết lập khóa cho các hệ mã đối xứng, chứng thực, và các dịch vụ đánh dấu thời gian và các giao thức chống từ chối.

**Điểm yếu của các hệ mã công khai**

Như mọi hệ thống liên quan đến bảo mật, điều quan trọng là xác định các điểm yếu tiềm ẩn. Ngoài việc lựa chọn thuật toán khóa bất đối xứng yếu hoặc độ dài khóa quá ngắn, rủi ro bảo mật chính là khóa cá nhân bị lộ.

Ngoài ra, với sự ra đời của điện toán lượng tử, nhiều thuật toán khóa bất đối xứng được coi là dễ bị tấn công và các kế hoạch kháng lượng tử mới đang được phát triển để khắc phục vấn đề này.

**HÀM BĂM MẬT MÃ**

**Hàm băm**

Hàm băm mật mã (**CHF** – Cryptographic Hash Function) là một thuật toán băm (ánh xạ của chuỗi nhị phân bất kỳ thành chuỗi nhị phân có kích thước cố định là n bit và có các tính chất:

. Xác suất tìm ra một chuỗi từ một trị băm n-bit cụ thể là 2-n (đối với bất kỳ hàm băm tốt nào). Do đó giá trị băm có thể được sử dụng làm đại diện cho thông báo;

. Kháng đụng độ một. Việc tìm chuỗi đầu vào khớp với giá trị băm cho trước (tiền ảnh) là bất khả thi.

. Kháng đụng độ hai. TVieecj tìm rah ai chuỗi có cùng trị băm là không thể

Hàm băm mật mã có nhiều ứng dụng bảo mật thông tin, đặc biệt là trong chữ ký số, mã xác thực thông điệp (MAC – *Message Authentication Code*) và các hình thức xác thực khác. Chúng cũng có thể được sử dụng để lập chỉ mục dữ liệu trong bảng băm, để lấy dấu vân tay, để phát hiện dữ liệu trùng lặp hoặc nhận dạng tính duy nhất các tập tin và kiểm tổng (*checksum*) để phát hiện lỗi dữ liệu. Trị băm mật mã đôi khi được gọi là dấu vân (*fingerprint*), tổng kiểm tra hoặc đơn giản là trị băm, cho thấy các ứng dụng có thể có của hàm băm.

Khac với hàm băm mật mã, hàm băm phi mật mã được sử dụng trong các bảng băm và để phát hiện các lỗi vô tình, việc xây dựng chúng thường không có khả năng chống lại các tấn công có chủ ý. Ví dụ: tấn công từ chối dịch vụ vào các bảng băm có thể xảy ra nếu các xung đột được tìm thấy dễ dàng.

**Tính chất**

Hàm băm mật mã thường được thiết kế để nhận đầu vào là một chuỗi có độ dài và tạo ra giá trị băm có độ dài cố định.

Hàm băm mật mã phải có khả năng chống lại tất cả các kiểu tấn công phân tích mật mã đã biết. Trong mật mã lý thuyết, mức độ bảo mật của hàm băm mật mã đã được xác định bằng các thuộc tính sau:

***Kháng tiền ảnh một***

Cho trước trị băm h, sẽ khó tìm được bất kỳ thông điệp m nào sao cho h = hash(m). Khái niệm này có liên quan đến hàm một chiều. Các hàm không có tính chất này có thể bị tấn công tiền ảnh.

***Kháng đụng độ yếu***

Với đầu vào m1, khó tìm được đầu vào m2 ≠ m1 sao cho hash(m1) = hash(m2). Tính chất này còn được gọi là khả năng kháng đụng độ yếu. Các hàm không có tính chất này dễ bị tấn công tiền ảnh loại hai.

***Kháng đụng độ mạnh***

Rất khó để tìm được hai thông điệp khác nhau m1 và m2 sao cho hash(m1) = hash(m2). Cặp thông điệp như vậy được gọi là độ độ. Tính chất này còn được gọi là khả năng kháng đụng độ mạnh. Trị băm cần dài ít nhất gấp đôi giá trị cần thiết cho khả năng kháng tiền ảnh; nếu không thì đụng độ có thể được tìm ra bằng tấn công sinh nhật.

Khả năng kháng đụng độ ngụ ý khả năng kháng tiền ảnh hai nhưng không bao hàm khả năng kháng tiền ảnh.

Các tính chất này có nghĩa là kẻ xấu không thể thay thế hoặc sửa dữ liệu đầu vào mà không thay đổi trị băm, gọi là bản tóm tắt của văn bản đầu vào. Vì vậy, nếu hai chuỗi có cùng một bản tóm tắt, ta có thể kết luận chúng giống hệt nhau. Khả năng kháng tiền ảnh thứ hai ngăn kẻ tấn công tạo một văn bản có cùng trị băm với văn bản mà kẻ tấn công không biết. Khả năng kháng đụng độ ngăn kẻ tấn công tạo hai văn bản khác nhau có cùng trị băm.

Một hàm đáp ứng các tiêu chí này vẫn có thể có những tính chất không mong muốn. Hiện tại, các hàm băm mật mã phổ biến rất dễ bị tấn công kéo dài độ dài: cho trước hash(m) và len(m) chứ không phải m, bằng cách chọn một m′ phù hợp, kẻ tấn công có thể tính toán hash(m ∥ m′), trong đó ∥ là phép ghép nối. Tính chất này có thể được sử dụng để phá vỡ các sơ đồ xác thực đơn giản dựa trên hàm băm. Việc tạo mã chứng thực thông điệp dùng hàm băm – HMAC (Hash Message Authentication Code) khắc phục được vấn đề này.

A black and white text with black text

Description automatically generated with medium confidenceTrong thực tế, khả năng kháng đụng độ không đủ cho nhiều mục đích sử dụng thực tế. Ngoài khả năng kháng đụng độ, kẻ tấn công không thể tìm thấy hai thông điệp có nội dung cơ bản giống nhau; hoặc suy ra được bất kỳ thông tin hữu ích nào về dữ liệu, mà chỉ dựa trên bản tóm tắt của dữ liệu. Cụ thể, hàm băm phải hoạt động giống hàm ngẫu nhiên nhất có thể (thường được gọi là sấm ngữ ngẫu nhiên (oracle) trong các bằng chứng về bảo mật) trong khi vẫn có tính xác định và tính toán hiệu quả. Điều này loại trừ các hàm như hàm băm xác suất dựa trên biên đổi Fourier– SWIFFT, có thể được chứng minh chặt chẽ là có khả năng kháng đụng độ với giả định rằng một số bài toán là khó tính toán, nhưng, với một hàm tuyến tính, không thỏa mãn các tính chất bổ sung này.

**Diffie-Hellman**

**A math equations and formulas

Description automatically generated with medium confidenceElliptic curvesA screenshot of a white page with red and blue text

Description automatically generated**

A math equation on a white background

Description automatically generatedA close-up of a document

Description automatically generated

A close up of a document

Description automatically generatedA screenshot of a computer program

Description automatically generatedA screenshot of a computer code

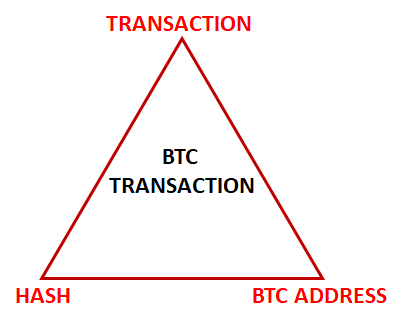
Description automatically generated

A white text with black text and red text

Description automatically generated

A white paper with black text and black text

Description automatically generated

**6 MẠNG CHUỖI KHỐI BITCOIN**

**CHUỖI KHỐI BITCOIN**

**Sổ cái bitcoin**

Bitcoin có thể coi là cài đặt sổ cái phi tập trung thành công nhất cho đến hiện nay. Bitcoin được Satoshi thiết kế năm 2007 từ ý tưởng sổ cái phi tập trung ở đó mọi thành viên có thể giao dịch mà không phải tin tưởng ai. Và 20/10/2008, Bitcoin được biết đến như một giao thức thanh toán ngang hàng (*peer-to-peer payment protocol*) của nhân vật bí ẩn Satoshi Nakamoto, và 9/01/2009 bắt đầu được sử dụng với khối khởi thủy (*genesis block*) làm quy chiếu cho mọi nút sau này. Giao dịch đầu tiên trị giá 10 Bitcoin do Satochi Nakamoto gửi cho Hal Finney, một nhà mật mã học, vào 12/01/2009. Và 5/10/2009 Bitcoin được ấn định trên sàn giao dịch (*exchanges*) với mức khởi điểm $1 = 1309.03 BTC (viết tắt của Bitcoin) hay 1 BTC = 0.00076 USD là chi phí tiêu thụ năng lượng cho một máy tính xác thực giao dịch thành công, gọi là máy đào (*minner*). Sau đó, khoảng giữa năm 2010, Bitcoin được xem như sổ cái phi tập trung (decentralized ledger) cho đến nay, và trở nên ngày càng phổ biến sau khi BTC được dùng để mua 2 chiếc bánh pizza bẳng 1000 BTC, tương đương $25 ngày đó. Tiếc thay, sự thành công bày làm mọi người hiều chuỗi khối như một loại tiền tệ, và thuật ngữ tiền mã (*cryptocurrency*) được dùng chính thức.

Nhấn mạng rằng Bitcoin hay BTC là một ứng dụng của chuỗi khối cài đặt sổ cái phi-tập trung.

**Mạng Bitcoin**

Tích hợp sẵn trong giao thức Bitcoin là công nghệ chuoỗi khối (blockchain technology). Mọi giao dịch đều được cập nhật trên hệ thống lưu trữ máy tính ngang hang (P2P – peerr to peer network), trong CSDL dạng chuỗi khối (blockchain), một sổ cái ghi lại số dư của mỗi tài khoản và lịch sử tất cả tài khoản tham gia giao dịch trước đó.

Khác biệt của mạng BTC với các giao dịch ‘ngân hàng’ là giao dịch trực tuyến, mạng BTC không cần đến bên trung gian với CSDL tập trung (centralized database) mà ta phải tin tưởng để xác minh các giao dịch nhằm chống giao dịch. Công nghệ chuỗi khối được giải quyết vấn đề xài thông tin một khối (tiền mã BTC) nhiều lần, gọi là vấn đề tiêu lại (double spent) này mà không cần bên trung gian thứ 3 đáng tin cậy:

Mọi thứ đều được ghi lại trên sổ cái phi-tập trung và không thể sửa đổi bất kỳ thông tin nào được ghi vào sổ cái.

**Các tính chất của ‘tiền tệ’ BTC**

Bitcoin, một loại tiền mật mã phi tập trung, an toàn, công khai, bình đẳng và ẩn danh.

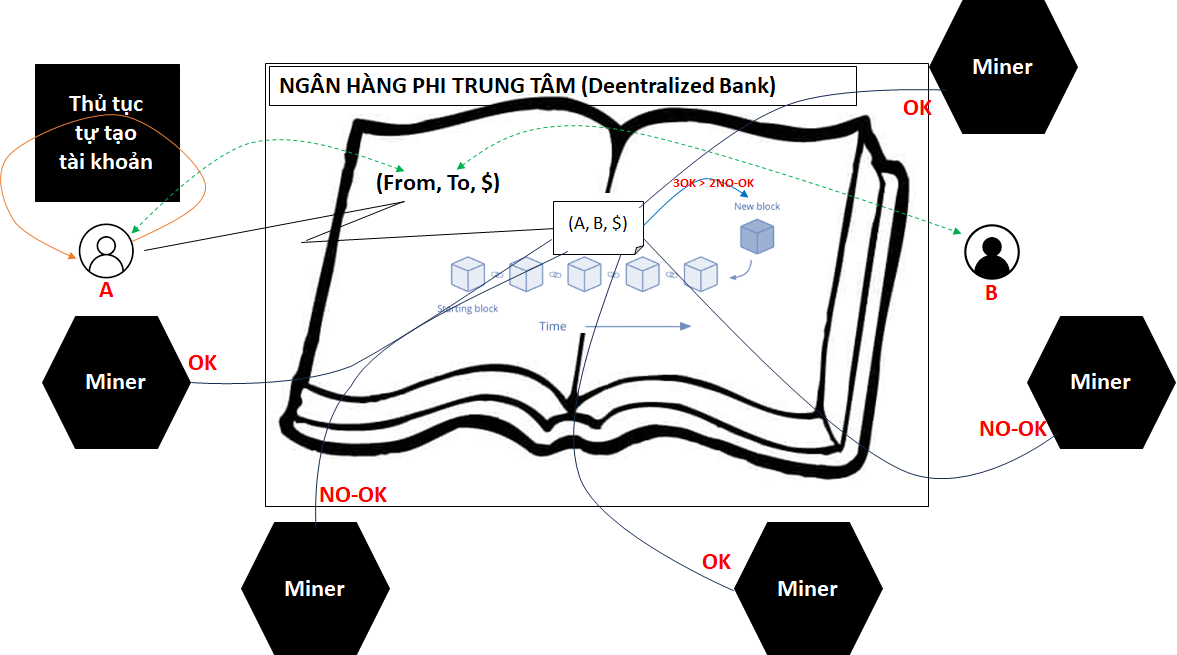
* ***Phi tập trung***. Không có cá nhân hay tổ chức nào sở hữu hoặc kiểm soát hoàn toàn được mạng bitcoin.
* ***Bảo mật***. Mạng bitcoin được xây dựng trên các thuật toán bảo mật hiện đại.
* ***Công khai***. Mạng bitcoin duy trì một sổ cái (ledger) ghi lại tất cả các giao dịch của người dùng tại mỗi nút mạng.
* ***Bình đẳng***. Bất kỳ ai hoặc bất kỳ tổ chức nào cũng có thể tham gia giao dịch trên mạng bitcoin.
* ***Ẩn danh***. Khi tham gia mạng bitcoin, không cần khai báo hay đăng ký thông tin cá nhân với bất kỳ tổ chức nào.

**MẠNG BITCOIN – NGÂN HÀNG PHI TẬP TRUNG**

Trước hết, cần lưu ý là chuỗi khối (*blockchain*) không phải là tiền mật mã (*cryptocurrency*), nhưng do mạng bitcoin (**BTC** *network*) sử dụng công nghệ chuỗi khối (*blockchain technology*) xây dựng ứng dụng cho tài chánh, mà tập trung chính vào giao dịch (**Tx** – *transaction*) nên ta mượn nhiều thuật ngữ ngân hàng để giải thích các thành phần cũng như nguyên lý hoạt động của mạng BTC.

**Ngân hàng phi tập trung**

Các khái niệm và hoạt động của ngân hàng phi tập chung là tương tự trong ngân hàng tập trung, chỉ khác ở điểm cốt lõi: không có và có bên trung gian thứ 3 đáng tin cậy. Nếu như ở ngân hàng tập trung, vai trò chính nằm ở bên thứ 3 đáng tin chính là ngân hàng, thậm chí, khách hàng cũng chỉ là phụ ngay cả tiền trong ngân hàng tai tài khoản của khách hàng thì có khi muốn rút cũng không thể rút nếu ngân hàng không cho. Hình dưới mô phỏng quy trình hoạt động của một ngân hàng phi tập trung, chủ yếu xây dựng xoay quanh 2 khái niệm số tài khoản ngân hàng, ở đây ta sẽ sử dụng thuật ngữ định danh người chủ tài khoản (*Identification*), và khái niệm giao dịch – **Tx** (*transaction*).



***Tóm tắt quy trình***

Người dùng tự do tham gia mạng BTC. Để tham gia, đơn giản người dùng chỉ tự tạo cho mình tài khoản BTC để chuyển và nhận BTC tới hay từ người khác cũng đã có tài khoản trên mạng BTC. Tài khoản là một định danh (Id) mà là một cặp (**khóa công khai**, **khóa cá nhân**), khóa công khai có thể được sử dụng trực tiếp hoặc có thể được chuyển sang **địa chỉ** (***address***) trên mạng BTC từ khóa công khai; và khóa cá nhân được sử dụng để chứng minh là chủ tài khoản hay chủ địa chỉ trên BTC. Khi ấy, người dùng với thiết bị nối mạng của mình hình thành một nút sử dung (*user node*) trên mạng BTC. User node có thể chỉ phải kích hoạt khi giao dịch.

Để thực hiện một giao dịch chuyển khoản, khác với giao dịch trên ngân hàng tập trung, ở BTC, người dùng tạo và phát tán (broadcast) giao dịch, gồm địa chỉ của mình, địa chỉ nhận, số tiền chuyển, và một số thông tin khác, lên mạng BTC. Trên mạng BTC có sẵn một số nút mạng đặc biệt gọi là nút khai thác (*miner node*), mà lưu toàn bộ CSDL sổ cái – block chain BTC và phải luôn ở trạng thái kích hoạt. Các miner nodes tập hợp các giao dịch được phát trên mạng BTC, xác minh tính hợp lệ và chấp nhận (OK) hay không (NO-OK). Giao dịch hợp lệ là giao dịch được đa số đông thuận, sẽ được 1 miner node nào đó đóng vào khối mới và gắn vào chuỗi khối.

***Địa chỉ và giao dịch BTC***

Cũng như trong ngân hàng tập trung, trong ngân hàng phi tập trung BTC, cũng có khái niệm ‘tài khoản’ và ‘giao dịch’.

. **Tài khoản BTC**. Là cặp (địa chỉ A, khóa cá nhân d), trong đó địa chỉ A được tạo từ khóa công khai e tương ứng với khóa cá nhân d, của một hệ mã công khai. BTC sử dụng mã đường cong elliptic (**ECC**).

. **Giao dịch BTC**. Cũng như giao dịch chuyển khoản trong ngân hàng truyền thống, chỉ khác ở 2 điểm chính (**i**) người dùng tự tạo giao dịch, và (**ii**) giao dịch được các nút khai thác xác minh.

**THỦ TỤC TẠO ĐỊA CHỈ CHO MỘT TÀI KHOẢN BTC**

Số tài khoản trên BTC được gọi là địa chỉ của tài khoản (*account address*). Địa chỉ một tài khoản cũng là địa chỉ định danh công khai (public *identification*) một nút trên mạng BTC, nút sử dụng (*user node*) cũng như nút khai thác (*miner node*). Địa chỉ BTC là một chuỗi 25 ký tự (*character*) mỗi ký tự 1 byte, được tạo từ khoa công khai của hệ mac đường cong elliptic (**ECC** – *Elliptic Curve Creyprosystem*).

Thủ tục tạo địa chỉ từ khóa công khai như trình bày dưới, trong thủ tục này, 2 hàm băm mật mã được sử dụng gồm:

. **Hàm băm SHA-2**. Hàm băm SHA-2, ký hiệu SHA256, nhận đầu vào là địa khóa công khai hệ mã ECC và trả về trị băm là một số 256 bits hay chuỗi 32 bytes.

. **Hàm băm RIPEMD-160**. RIPEMD-160, viết tắt là RIPMD160 (*RIPE message digest* - 160), Trả về trị băm là một chuỗi 160 bits hay 20 bytes.

**Thủ tục tạo địa chỉ BTC**

(**Bước 1**). Trước hết, sử dụng thủ tục tạo cặp khóa của ECC gồm khóa công khai e và khóa cá nhân d. Khóa công khai e là một điểm P = (x, y) trên nhóm đường cong elliptic (E, +).

Sử dụng kỹ thuật biểu diễn của SSL (*Secure Sockets Layer*), ta ghép thêm 1 byte vào điểm khóa công khai để biểu diễn 1 điểm của ECC bằng 65 bytes, trong đó, byte đầu tiên, tính từ trái qua, được dùng để mã ‘kiểu lưu’, 32 bytes tiếp theo biểu diễn hoành độ, và 32 bytes cuối biểu diễn tung độ của khóa công khai e:

PXY ← sparse(e) = [1 byte P | 32 bytes X | 32 bytes Y].

Các giá trị của byte P đầu tiên được quy ước như sau:

. Nếu có giá trị thập lục phân là 0x02 hay 0x03, thì chỉ lưu địa chỉ x. Địa chỉ y sẽ được tính từ x theo phương trình đương cong (E).

. Nếu giá trị là 0x04, thì lưu cả x và y.

(**Bước 2**). Chuỗi 65 bytes kết quả của bước 1 được băm thành chuỗi 32 byte.

Z ← SHA256(PXY) = [ 32 bytes Z]

(**Bước 3**). Chuỗi 32 bytes kết quả băm của SHA-2 lại được băm, bằng hàm băm RIPE MD-160, thành chuỗi 20 bytes.

U ← RIPMD160(Z) = [ 20 byte U]

(**Bước 4**) Sau đó, kết quả băm U được thêm 1 bytes vào bên trái để mã hóa ý nghĩa của địa chỉ.

V ← [1 byte Q | 20 bytes U].

Các giá trị của byte đầu tiên Q, được quy ước như sau:

. Nếu giá trị là 111 hay 196 thì sẽ biểu diễn địa chỉ của mạng thử (test net), là mạng BTC để cho các nhà phát triển ứng dụng mạng BTC có thể thử nghiêm các ý tưởng thuật toán mới. Trong đó

* **111** cho biết địa chỉ tạo ra là đị chỉ khóa, còn
* **196** là địa chỉ kịch bản, là địa chỉ bắt đầu của một ‘chương trình’ BTC.

(**Bước 5**) Chuỗi 21 bytes kết quả lại được băm thành giá trị băm SHA-2 gồm 32 bytes. Nhưng chỉ giữ lại 4 bytes đầu tiên bên trái để làm mã kiểm tổng (*checksum*).

CS ← SHA256(V) = [4 bytes CS].

(**Bước 6**) Ghép 4 bytes CS của kết quả với 21 bytes V của bước 5, được địa chỉ trung gian, là một chuỗi 25 bytes hay 200 bits nhị phân.

TIA ← [1 byte T | 20 bytes RIPEMD | 4 bytes SHA-2]

(**Bước 7**). Dãy 200 bits nhị phân gồm toàn giá trị 0, 1 này rất dễ nhầm lẫn khi phải nhập vào và rất khó kiểm tra kết quả nhập đúng hay sai. Vì thế nó dduocj chuyển sang chuỗi các ký tự ascii bằng thủ tục BASE256-2-BASE256. Thủ tục này chuyển chuỗi nhị phân sang dạng ký tự trong tập các ký tự {A,…, Z, a,…, z, 0… 9}\{+, /, O, 0} gồm 58 ký tự. Các ký tự {+, /, O, 0} được bỏ đi để tăng khả năng tránh nhầm lẫn khi nhập từ giấy vào máy tính.

Ta biết rằng, mọi giao dịch (**Tx**) đều có địa chỉ nguồn (**from**) là địa chỉ của người gửi, địa chỉ đích (**to**) là địa chỉ người nhận, và số tiền (**$**) cùng một số thông tin khác. Sai lầm của chuỗi ký tự **from** có thể được phát hiện khi xác minh **Tx**, nhưng chuỗi địa chỉ nhận **to** thì không được dùng trong cụ thể tiến trình xác minh. Vì thế, một khi **Tx** được ghi lên chuỗi khối thì không thể sửa chữa được nữa. Nghĩa là **$** chuyển từ **from** đến một **to** không đúng ý muốn. Vì thế, thủ tục ký tự hiển thị hóa nhằm hạn chế khả năng mất **$** vô lý này.

Add\_BTC ← BASE256-2-BASE256(TIA).

(**Bước 8**) Chuỗi kết quả của bước 6 chính là địa chỉ nút mạng BTC ứng với khóa công khai e ban đầu.

Return Add\_BTC.

Hình sau tóm tắt quy trình tạo địa chỉ BIT.

A screenshot of a computer

Description automatically generated

**GIAO DỊCH**

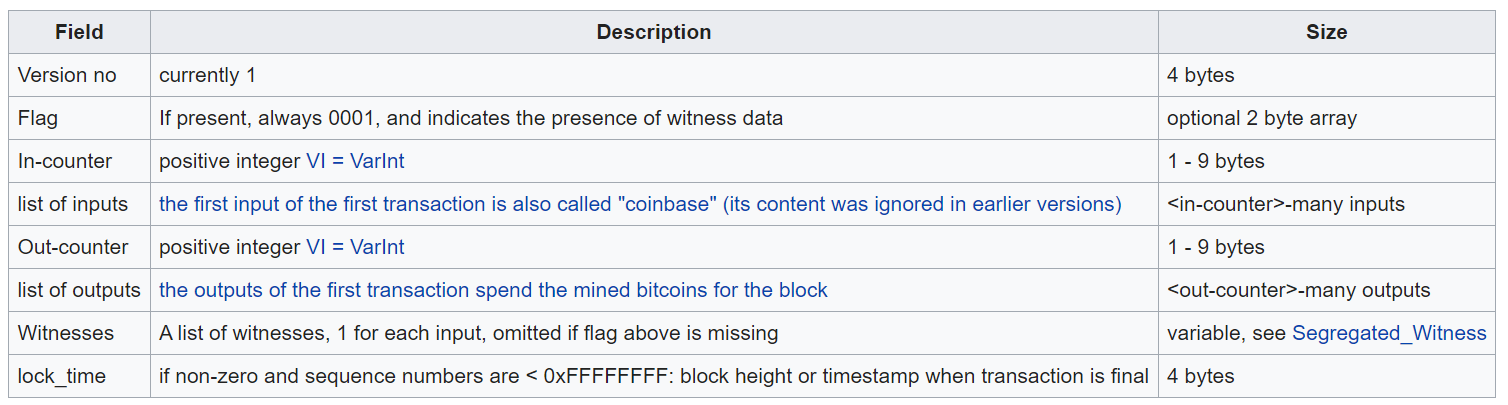
Bên cạnh khái niệm địa chỉ BTC (*BTC address*), giao dịch (**Tx** – *transaction*) và thực hiện (xác minh) giao dịch là khái niệm chính trong giao dịch bitcoin.

**Cấu trúc giao dịch BTC**

Giao dịch (**Tx**) là chuyển BTC được phát lên mạng BTC và được tập hợp trong các khối. Một giao dịch thường tham chiếu đến đầu ra các giao dịch đã xảy ra trước đó như đầu vào của giao dịch mới và chuyển tất cả giá trị BTC đầu vào cho đầu ra mới. Giao dịch BTC không được mã hóa nên có thể duyệt và xem nội dung mọi giao dịch trên chuỗi khối. Một khi giao dịch được ghi lên chuỗi khối sau khi dduocj xác minh, sẽ không thể thay đổi được nữa.

Đầu ra một giao dịch chuẩn chỉ đến địa chỉ và việc sử dụng (tiêu BTC) bất kỳ đầu vào nào trong tương lai đều cần có chữ ký liên quan, nghĩa là phải có khóa cá nhân, gọi là khóa ký (*signature key*) để có thể tiêu (*spend*) BTC đầu ra đó.

Mọi giao dịch đều hiển thị trong chuỗi khối (bằng giá tị thập lục phân – *hexa*) và có thể được xem bằng trình hiện thị giá trị **hex**. Trình duyệt chuỗi khối là một trang web nơi mọi giao dịch có trong chuỗi khối có thể được xem dưới dạng tự nhiên. Điều này giúp xem chi tiết kỹ thuật của các giao dịch đang hoạt động và để xác minh thanh toán. Cấu trúc tổng quát một giao dịch có dạng như mô tả trong hình dưới.



***Giao dịch 1 đầu vào – 1 đầu ra***

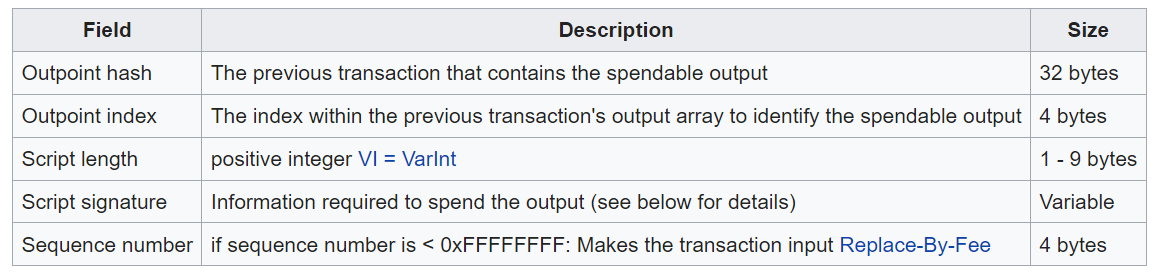
Hình sau minh họa nội dung một giao dịch BTC với 1 đầu vào và 1 đầu ra.



Theo đó, đầu vào giao dịch này nhập 50 BTC từ đầu ra số 0 trong giao dịch f5d8... Sau đó, đầu ra sẽ gửi 50 BTC đến địa chỉ BTC (được biểu thị ở đây bằng hệ thập lục phân 4043... thay vì cơ sở58). Khi người nhận muốn tiêu số tiền này, anh ta sẽ tham chiếu đầu ra số 0 của giao dịch này vào đầu vào của giao dịch của chính mình.

***Đầu vào giao dịch***

Cấu trúc đầu vào được mô tả trong hình:

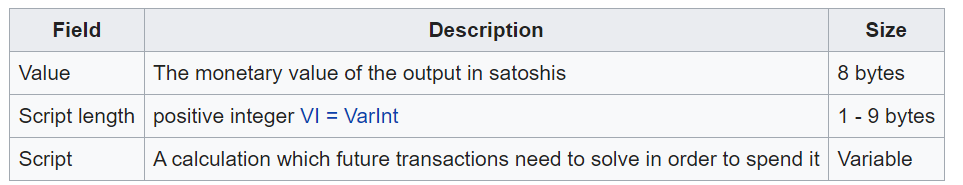


Đầu vào là tham chiếu đến đầu ra từ một giao dịch trước đó. Có thể có nhiều đầu vào trong một giao dịch. Tất cả các giá trị đầu vào của giao dịch mới (nghĩa là tổng giá trị BTC của các đầu ra trước đó được tham chiếu bởi đầu vào của giao dịch mới) được cộng lại và tổng (trừ đi phí giao dịch) được sử dụng hết cho đầu ra của giao dịch mới. Tx trước đó là trị băm của giao dịch trước đó. Chỉ số là đầu ra cụ thể trong giao dịch được tham chiếu. ScriptSig là nửa đầu của kịch bản (sẽ được thảo luận chi tiết hơn ở phần sau).

Tập lệnh chứa hai thành phần, chữ ký và khóa công khai. Khóa công khai phải tương úng với trị băm được cho trong tập lệnh của kết quả. Khóa công khai được sử dụng để xác minh chữ ký của người chuyển, là thành phần thứ hai. Cụ thể, thành phần thứ hai là chữ ký ECDSA trên trị băm của phiên bản giao dịch. Chữ ký kết hợp với khóa công khai, chứng minh giao dịch được tạo ra bởi chủ sở hữu thực sự của số BTC đó. Các cờ (*flag*) trong cấu trúc giao dịch tổng quát xác định cách giao dịch và có thể được sử dụng để tạo các loại thanh toán khác nhau.

***Đầu ra của giao dịch***

Cấu trúc đầu ra được mô tả như hình dưới.



Một đầu ra chứa mã lệnh chuyển BTC. Giá trị là số Satoshi (1 BTC = 100.000.000 Satoshi). ScriptPubKey là nửa sau của kịch bản (sẽ thảo luận sau). Có thể có nhiều đầu ra và chúng chia sẻ giá trị tổng hợp của các giá trị đầu vào. Vì mỗi đầu ra từ một giao dịch chỉ có thể được tham chiếu một lần bởi đầu vào của giao dịch sau đó, toàn bộ giá trị đầu vào kết hợp lại cần phải được gửi trong đầu ra, nếu không sẽ được xem là phí giao dịch (*transaction fee*). Chẳng hạn, nếu tổng đầu vào là 50 BTC nhưng chỉ muốn chuyển 25 BTC, mạng BTC sẽ tạo ra hai đầu ra trị giá 25 BTC: một đến đích cần gửi và một gửi lại (thối lại) cho nguồn (được gọi là "(tự thay đổi" dù gửi lại cho chính mình). Bất kỳ BTC đầu vào nào không được chuyển thành đầu ra đều được coi là phí giao dịch; nút tạo ra khối (*miner node*) có thể yêu cầu phí bằng cách chèn nó vào giao dịch coinbase của khối đó.

***Bằng chứng***

Đối với các giao dịch Segwit, có một danh sách các bằng chứng ngay đầu ra, với mỗi trường bằng chứng tương ứng với một đầu vào có cùng chỉ mục. Mỗi trường bằng chứng chứa một biến cho biết số phần tử trên ngăn xếp (*stack*) của bằng chứng và bản thân các phần tử stack chứa các cặp biến, biểu thị độ dài của dữ liệu và chính dữ liệu đó.

Điều quan trọng cần lưu ý là các trường bằng chứng không được tập hợp lại thành một trường "witness" ở đầu ra có cấu trúc, được thực hiện cách tuần tự, mà mỗi bằng chứng được đặt trong từ khóa "witness" riêng của nó bên dưới đầu vào tương ứng.

**Xác minh giao dịch**

Để xác minh rằng đầu vào hợp lệ với giá trị đầu ra được tham chiếu, mạng bitcoin sử dụng hệ kịch bản tựa-Forth. scriptSig đầu vào và scriptPubKey đầu ra sẽ được thực thi, với scriptPubKey sử dụng các giá trị còn lại trên stack bởi scriptSig. Đầu vào hợp lệ nếu scriptPubKey trả về kết quả true. Thông qua các kịch bản, người gửi có thể tạo ra những điều kiện phức tạp phải thỏa để có thể nhận được giá trị đầu ra. Ví dụ: có thể tạo một đầu ra mà không cần bất kỳ sự cho phép nào; cũng có thể yêu cầu đầu vào phải được ký bằng 3 khóa khác nhau hoặc có thể sử dụng được bằng mật khẩu thay vì khóa.

***Các kiểu giao dịch***

Mạng BTC cung cấp hai cặp kịch bản scriptSig/scriptPubKey khác nhau. Có thể thiết kế các loại giao dịch phức tạp hơn và liên kết chúng lại nhau thành các hợp đồng (contract) được thực thi bằng mật mã.

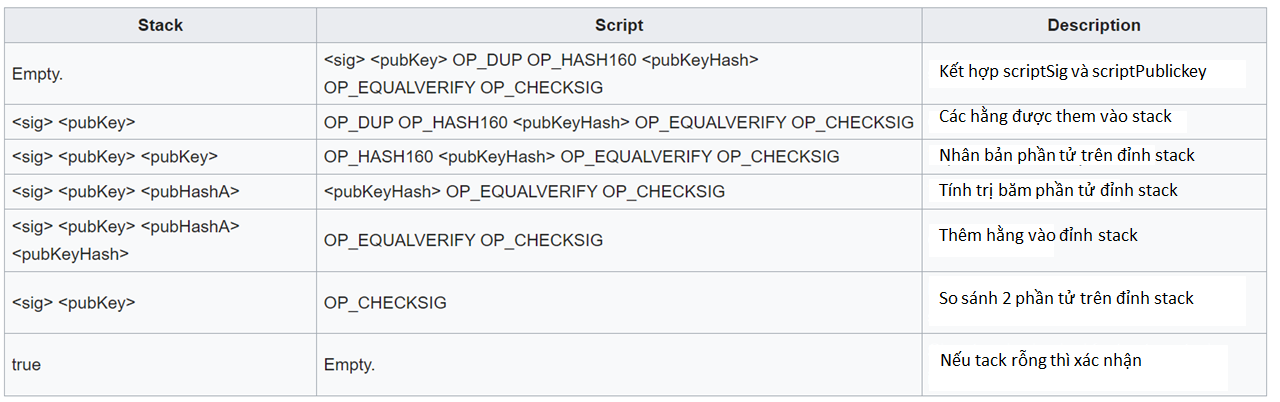
*Thanh toán cho một địa chỉ* – **Pay-to-Pubkey-Hash**

scriptPubKey: OP\_DUP OP\_HASH160 <pubKeyHash> OP\_EQUALVERIFY OP\_CHECKSIG

scriptSig: <sig> <pubKey>

Địa chỉ BTC, như ta đã biết, là 1 trị băm nên không thể cung cấp khóa công khai trong scriptPubKey. Khi chuyển BTC đã được gửi đến địa chỉ bitcoin, người nhận sẽ cung cấp cả chữ ký và khóa công khai. Tập lệnh xác minh khóa công khai đó sẽ được băm thành trị băm trong scriptPubKey, sau đó kịch bản cũng kiểm tra chữ ký tương ứng khóa công khai đó.

Quá trình xác minh được mô tả trong hình dưới.



*Thanh toán cho kịch bản băm* – **Pay**-**to-script-hash**

scriptPubKey: OP\_HASH160 <scriptHash> OP\_EQUAL

scriptSig: ..signatures... <serialized script>

m-of-n multi-signature transaction:

scriptSig: 0 <sig1> ... <script>

script: OP\_m <pubKey1> ... OP\_n OP\_CHECKMULTISIG

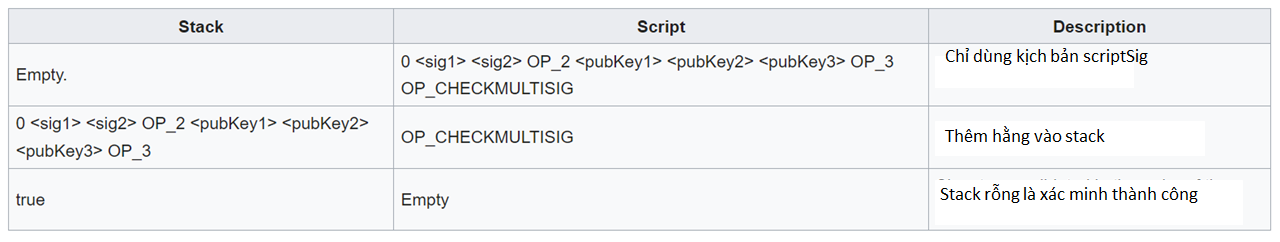
Địa chỉ P2SH (Pay-to-script-hash) được tạo ra với mục đích chuyển "trách nhiệm cung cấp các điều kiện để đổi giao dịch từ người gửi tiền sang người đổi. Chúng cho phép người gửi nạp tiền cho một giao dịch tùy ý, bất kể sự phức tạp, bằng cách sử dụng 20 byte hàm băm". Địa chỉ P2SHtương tự như trị băm 20 byte của khóa công khai.

P2SH là phương tiện cho các giao dịch phức tạp, khác với thanh toán qua địa chỉ trị băm khóa công khai, có scriptPubKey và scriptSig xác định. Thông số kỹ thuật không đặt ra giới hạn nào đối với kịch bản và do đó hoàn toàn có thể thực hiện bất kỳ hợp đồng nào bằng cách sử dụng các địa chỉ này.

scriptPubKey trong giao dịch cấp vốn là kịch bản đảm bảo rằng kịch bản trong giao dịch đổi thưởng sẽ băm thành kịch bản được sử dụng để tạo địa chỉ.

Trong scriptSig ở trên, '*chữ ký'* đề cập đến bất kỳ kịch bản nào đủ để đáp ứng kịch bản sau đây.

Quá trình xác minh được minh họa qua hình sau.



**7 DUY TRÌ CHUỖI KHỐI A triangle with text on it

Description automatically generated**

**CHUỖI KHỐI VÀ BITCOIN**

**Chuỗi khối**

Chuỗi khối (*blockchain*), như ta đã biết, là một công nghệ sổ cái phân tán và có thể đại diện tiêu biểu cho mạng Internet thế hệ hai. Nếu như trong Internet thế hệ thứ nhất, với công nghệ liên mạng toàn cầu và những cỗ máy tìm kiếm hiện đại, thông tin được lưu trữ và trao đổi một cách bình đẳng, dân chủ, thì chuỗi khối, Internet thế hệ hai, với khái niệm giao dịch (*transaction*) hứa hẹn sẽ dân chủ hóa việc trao đổi giá trị thực. Vào khoảng giữa năm 2008, giữa cuộc khủng hoảng tài chính toàn cầu, một giao thức mới cho “Hệ thống tiền mặt điện tử ngang hàng” và tạo ra một loại tiền kỹ thuật số Bitcoin (***BTC*** *cryptocurrency*) dựa trên công nghệ chuỗi khối, với giao dịch BTC đầu tiên được thực hiện vào ngày 12/1/2009. Khác với tiền tện truyền thống, tiền điện tử không được phát hành bởi một quốc gia nào; cũng không được lưu trữ trong kho tiền ngân hàng hoặc tín dụng được ghi trong tệp điện tử ở đâu đó mà là một bảng kê tất cả các giao dịch trên toàn cầu (*global* *transaction*) dựa trên lý thuyết và công nghệ mạng ngang hàng (**P2P** – *Peer-to-Peer*) để xác minh và phê duyệt từng giao dịch BTC. Nhưng chuỗi khối không chỉ là ‘tiền điện tử’, chuỗi khối có nhiều ứng dụng hơn và đang đượcc đầu tư phát triển với tốc độ nhanh.

***Tính chất của công nghệ chuỗi khối***

*Phân tán* – **distributed**

Giao thức cài đặt quy tắc tính toán phân tán nhằm đảm bảo tính toàn vẹn của dữ liệu được trao đổi giữa các thiết bị nối mạng mà không cần bên thứ ba đáng tin. Là CSDL được lưu trữ trên máy tính ‘đào’ tình nguyện viên trên toàn cầu, không có cơ sở dữ liệu trung tâm để có thể bị tấn công khai thác hay bị phá hủy hoặc tắt. Điều này có nghĩa là những thứ có giá trị cao – tiền, cổ phiếu, trái phiếu, quyền sở hữu trí tuệ, âm nhạc, xe hơi và thậm chí cả phiếu bầu – có thể được lưu trữ và trao đổi mà không cần một tổ chức hoặc người trung gian nào.

*Mã hóa* – **encrypted**

Chuỗi khối sử dụng mã công khai để cavs thực duy trì tính bảo mật.

*Đầy đủ và minh bạch* – **inclusive and transparence**

Với chuỗi khối công cộng, mọi người đều có thể xem tất cả các giao dịch trên mạng khiến không cá nhân hoặc tổ chức nào có thể che giấu giao dịch. Là nguồn mở nên ai cũng có thể tải xuống và sử dụng. Với chuỗi khối, mọi người đều có thể giao tiếp qua “chế độ xác minh thanh toán đơn giản hóa” và có thể được sử dụng trên thiết bị di động. Không cần phải có chứng từ để được tin hoặc được quyền truy cập.

*Bất biến* – **immutable**

Mọi giao dịch đều được xác minh và được lưu trữ trong một khối được ‘*xích – chain*’ với khối liền trước, tạo nên chuỗi khối (*blockcain*). Mỗi khối phải tham chiếu đến khối liền trước mới hợp lệ. Thông tin và các giao dịch được lưu ‘vĩnh viễn’, và không ai có thể thay đổi sổ cái.

*Lịch sử* – **historical**

Là một sổ cái phân tán thể hiện sự đồng thuận mạng của mọi giao dịch đã từng xảy ra nên chuỗi khối phải được bảo toàn trọn vẹn. Lưu trữ lịch sử cho phép truy xuất nguồn gốc của thông tin một cách dễ dàng.

Những tính chất này cho phép phát triển các ứng dụng dựa trên công nghệ chuỗi khối (*blockchain technology*) cho nhiều mục đích khác nhau.

***Thách thức***

Tuy nhiên, nếu các vấn đề về quản trị và quản lý không được giải quyết, công nghệ chuỗi khối có thể không đáp ứng được kỳ vọng.

*Nền tảng* – **platform**

Cần các giao thức để quản lý khả năng mở rộng (bằng chứng cổ phần – *proof of stake*), mức tiêu thụ năng lượng, tính ổn định (chuyển từ bằng chứng công việc – *proof of work*, sang *proof of stake*) và tính mạnh mẽ (*robustness*) của cơ sở hạ tầng mạng để đảm bảo hoạt động lâu dài. thành công (trong một thời kỳ).

*Ứng dụng* – **applications**

Cần có một số công cụ giao tiếp thân thiện và với người dùng và đào tạo các nhà phát triển lành nghề.

*Cấu trúc pháp lý để quản lý* – **legal structure for stewardship**

Cần một cơ quan quản trị để điều phối các vấn đề như khả năng tương tác, quyền riêng tư, bảo mật, bảo vệ danh tính và các hành động nhằm giảm mức độ không chắc chắn về mặt pháp lý xung quanh các công nghệ mới nổi.

**DUY TRÌ SỔ CÁI BTC**

Sổ cái BTC, hay chuỗi khối bất kỳ, là CSDL phi-tập trung, nghĩa là có nhiều nơi *lưu giữ cùng một CSDL*, gọi là các nút khai thác (*miner node*). Vì thế bên cạnh tính toàn vẹn (*integrity*) của CSDL sổ cái, tính đồng bộ (*synchronization*) cần phải đảm bảo để tai một thời điểm, CSDL các khối ở mọi thời điểm chỉ có thể khác nhau rất ít ở những khối cuối mới thêm vào. Có hai nhiệm vụ chính phải thực hiện trên mạng chuỗi khối:

* *Đúc khối* – **minted**. Tạo khối mới chứa tất cả các giao dịch đã được xác minh và gắn vào chuỗi.
* *Đồng bộ hóa* – **synchronization**. Duy trì tính nhất quán của CSDL phi tập trung.

Các nhiệm vụ này được đảm bảo thông qua cài đặt cơ chế đồng thuận số đông (*majority consensus*). Trong chuỗi khối, CSDL hiện hành là chuỗi có số khối nhiều nhất. Như vậy, các miner node luôn phải cập nhật để đảm bảo CSDL đang giữ trên máy chủ của mình là CSDL mới nhất. Tính đồng thuận cũng được thể hiện qua ‘cuộc đua’ đúc khối nhanh nhất. Miner nào có bằng chứng (*proof*) cho khối mới nhanh nhất sẽ là người chiến thắng và thêm khối đó vào CSDL, gắn khối vào chuỗi hiện hành và mọi miner khác phải cập nhật cho động bộ CSDL chung. Mạng BTC sử dụng khái niêm bằng chứng công việc – **PoW** (*Proof of Work*) cho nhiệm vụ này.

**Bằng chứng công việc**

***Mô tả bài toán công việc***

Bằng chứng công việc – PoW là minh chứng chứng minh đã có làm (và hoàn thành) công việc bằng cách giải một bài toán đố, và lời giải tìm được được dùng làm bằng chứng công việc giải bài.

Công việc trong chuỗi khối BTC, và nhiều chuỗi khối khác, là *tính hàm băm ngược* của hàm băm SHA-2 1-chiều. Do tính chất 1-chiều và nhất là tính ngẫu nhiên của SHA-2, cũng như nhiều hàm băm mật mã khác, việc tính hàm băm ngược buộc phải ‘vét cạn’. Ý tưởng này hình thành khái niệm ‘công việc’ (*work*) trong khai thác mạng BTC.

**Bài toán*.*** Cho trước mẫu (*template*) **T**, là một chuỗi nhị phân n bit, với k bit cố định, thường là k bit cao nhất có giá trị ‘0’, và một văn bản **D**. Bài toán là tìm ra một số ngẫu nhiên **nonce** (viết tắt của *number-once* số ngẫu nhiên dùng 1 lần)sao cho trị băm **H** là sốn bit, của chuỗi ghép **D||nonce** thỏa mẫu **T**: **H**(**D||nonce**) ∈ **T** = {N có n bit và 0 ≤ N < 2n-1}.

Trong mạng BTC, H được dùng là SHA-2, n = 256 bit, và số k được dùng để điều chỉnh ‘tốc độ làm việc’ sao cho trung bình một khối được đúc ra mất 10 phút. Hình sau mô tả bài toán tìm nonce.

A close up of a computer screen

Description automatically generated

***Ví dụ minh họa***

A computer code with red text

Description automatically generated with medium confidence

Mẫu **T** trong ví dụ này 12 bits đầu tiên của trị băm là 0 (3 ký tự thập lục phân).

**Bài toán**: Cho **SMS** = “Đây là tin nhắn #”, tìm một giá trị **nonce** sao cho SHA-2(**SMS||nonce**) khớp với một phần tử trong tập mẫu **T**, nghĩa là trị băm kết quả phải bắt đầu bằng ít nhất 3 số 0.

**Lời giải**: **nonce** = **6193**.

***Cài đặt***

Có 2 chiến lược cài đặt bài toán bằng chứng công việc: tuần tự hay ngaauc nhiên.

*Cách 1: tuần tự* – **sequence**

(1)

(2) If SHA-2(D||nonce) ∈ T: Return nonce

(3) Else nonce = nonce + 1

(4) If (chưa hết giờ): goto(1)

Như ta biết, trung bình 1 block được tạo ra cần 10 phút, nên thời gian để duyệt tuần tự tối đa phải ít hơn 10 phút.

*Cách 2: ngẫu nhiên* – **random**

(1)

(2) If SHA-2(D||nonce) ∈ T: Return nonce

(3) If (chưa hết giờ): goto(1)

***Cộng tác cùng làm việc***

Trong trường hợp có nhiều máy chủ, giả sử là m máy chủ, cùng hợp tác làm việc, tập chia được chia thành m tập con không giao nhau

T = T1 ∪ … ∪ Tm, Ti ∩ Tj≠I = ∅,

và giao cho mỗi máy làm việc trên một phần giao cụ thể trong thời gian cho phép. Thường thì việc chia này được tính theo năng lực làm việc của các máy chủ. Việc đo năng lực được thực hiện dựa trên hàm băm SHA-2.

**Cấu trúc của khối**

***Cấu trúc***

Mạng chuỗi khối BTC (*BTC* *blockchain*), như ta biết, là một chuỗi (*chain*) các khối (*block*). Mỗi Khối chứa tập các giao dịch (**Tx** – *transaction*) mới và được liên kết với Khối trước đó trong chuỗi khối. Mỗi khối được bảo vệ bằng bằng chứng công việc (**PoW** – *Proof of Work*) là đảo ngược hàm băm một phần của nội dung khối.

Mỗi khối gồm (**i**) các Tx hợp lệ, (**ii**) trị băm của khối liền trước và (**iii**) một nonce. Trong đó, nonce in a khối là lời giải của bài toán đảo ngược băm một phần. Hình sau mô tả các khối trong chuỗi khối.

|  |  |
| --- | --- |
| A diagram of a blockchain  Description automatically generated | . Mỗi khối có một giao dịch đặc biệt, được gọi là coinbase.  . Coinbase là giao dịch đầu tiên trong khối, Nó chỉ có 1 đầu vào ***TxIn***, và không có bất kỳ đầu ra ***TxOut*** nào trước đó.  . Một khối thường có nhiều giao dịch và một coinbase.  . Nút khai thác có thể chọn và đặt các giao dịch vào khối để khai thác (*mine*). Việc chọn thường được quyết định dựa trên mức phí (*fee*) mà giao dịch phải trả. |

***Nút khai thác***

Nút khai thác (*miner node*) là nút mạng giữ một bản sao đầy đủ của chuỗi khối, và giữ các cấu trúc dữ liệu thứ hai như bộ nhớ đệm lưu đầu ra các giao dịch chưa được sử dụng (*un-spent*) và nhóm bộ nhớ các giao dịch chưa được xác nhận (*un-validation*), để có thể nhanh chóng xác thực các giao dịch mới và khai thác khối mới.

Nếu giao dịch hoặc khối nhận được hợp lệ, nút sẽ cập nhật cấu trúc dữ liệu của nó và chuyển tiếp nó đến các nút mạng liên kết với nút này.

Điều quan trọng cần lưu ý là một nút mạng không cần phải tin cậy các nút mạng khác vì việc xác thực được thực hiện độc lập chỉ sử dụng các thông tin mà nó nhận được từ các nút mạng khác.

***Nguyên lý duiy trì chuỗi khối***

Khi nút khai thác (miner node) tìm thấy một khối mới, nó sẽ lan truyền. Mọi nút khai thác nhận được phải kiểm tra tính hợp lệ của khối, nghĩa là xem xem liệu nó có phải là lời giải cho bài toán công việc, nghĩa là đảo ngược băm một phần hay không. Nếu đúng, các nút khai thác đó sẽ nhật CSDL của mình để đồng bộ hóa với CSDL chung:

(1) Cập nhật bộ nhớ đệm đầu ra giao dịch chưa tiêu (UTXO)

(2) Cập nhật nhóm bộ nhớ giao dịch chưa được xác nhận.

Các nút mạng luôn duy trì một số kết nối nhất định tới các nút mạng khác trong mạng chuỗi khối.

***Tấn công chi tiêu nhiều lần***

Mạng BTC giải quyết vấn nạn chi tiêu lại (*double spend*) theo cách phân tán mà không cần cơ quan trung ương quyết định giao dịch nào hợp lệ. Kẻ tấn công muốn thay đổi chuỗi khối ở một khối nào đó, sẽ phải khai thác (giải lại) lại tất cả các khối tính từ khối muốn sửa đến khối đầu tiên của chuỗi BTC.

Cách duy nhất kẻ tấn công có thể thực hiện thành công điều này là có tốc độ băm nhanh hơn phần còn lại của toàn bộ mạng BTC. Tấn công này được gọi là tấn công 51% (*51% attack*).

***Tấn công khi đua***

Tấn công khi đua (*race attack*) xảy ra khi một nút nhận giao dịch chưa được xác nhận xung đột với giao dịch nằm trong bộ đệm chuyển đổi chưa được xác nhận của nó, tức là chi tiêu cùng một đầu ra, khi ấy, giao dịch mới sẽ bị loại bỏ.

Do đó, các nút chỉ giữ trong bộ đệm một bản sao của giao dịch nhận được đầu tiên. Race Attack thành công khi nhà cung cấp chấp nhận thanh toán cho giao dịch chưa được xác nhận khi chỉ kiểm tra một vài nút mạng. Kẻ tấn công có thể gửi một giao dịch đến các nút gần nhà cung cấp và một giao dịch khác tới nhiều nút khác trong mạng. Do đó, chỉ các nút ở gần nhà cung cấp mới thấy giao dịch gửi tiền cho nhà cung cấp, trong khi phần còn lại của mạng chứa trong bộ nhớ đệm của chúng một khoản chi tiêu lại (*double spent*). Để chống lại tấn công này, nhà cung cấp dịch vụ cần đợi giao dịch được đưa hẳn vào khối.

**Cây băm**

Để tối ưu quá trình xác minh tính toàn vẹn của các khối, gồm các giao dịch, dữ liệu được lưu trữ dạng cây băm, do Merkle đề xuất, gọi là cây Merkel (*Merkle tree*), là một cây trong đó mỗi nút lá, là nút ứng với các giao dịch, được gắn nhãn băm của khối dữ liệu và mỗi nút trung gian được gắn nhãn băm mật mã của nhãn các nút con của nó. Phần lớn cài đặt cây băm nhị phân (mỗi nút có hai nút con), nhưng tổng quát, chúng cũng có thể có nhiều nút con hơn. Cây Merkle trong Blockchain thực chất là gì và nó được sử dụng như thế nào trong chuỗi khối?

***Cây Merkle***

Cây Merkle (*Merkle tree*), còn được gọi là cây băm nhị phân, là một cấu trúc dữ liệu phổ biến trong khoa học máy tính. Chứng được áp dụng trong mạng BTC, và các mạng tiền điện tử khác, để mã hóa dữ liệu chuỗi khối cho hiệu quả và an toàn hơn.

Cây Merkle là cấu trúc dữ được tạo thành từ các trị băm của nhiều khối dữ liệu khác nhau nén tất cả các giao dịch trong một khối giúp minh nội dung nhanh chóng và an toàn trên các khối là tập dữ liệu lớn, đồng thời xác minh tính nhất quán và nội dung của dữ liệu. Hình dưới minh họa một cây Merkle với khối 4 giao dịch.

A diagram of a hash

Description automatically generated

Cây Merkle trong hình trên gồm 4 nút lá (*leaf*) được gán nhãn là trị băm H(Li), i =1, 2, 3, 4, của 4 dữ liệu lần lượt là Li, i = 1, 2, 3 ,4. Các nút tầng giữa, gọi là các nút trung gian, được gán nhãn là trị băm của 2 nút con tầng lá. Nút gốc, gọi là gốc cây băm Merkle (*Merkle root*), được gán nhãn là trị băm của 2 nút con tầng giữa.

***Gốc cây Merkle***

Gốc cây Merkle là một phương pháp toán học đơn giản để xác nhận các sự kiện trên cây Merkle. Chúng được sử dụng trong tiền điện tử để đảm bảo rằng các khối dữ liệu được gửi qua mạng ngang hàng là nuyên vẹn, không bị hư hại và không bị thay đổi.

Gốc cây Merkle cũng đóng một vai trò rất quan trọng trong việc tính toán cần thiết để duy trì hoạt động của các loại tiền điện tử như BTC. Hình dưới minh họa cây Merkle với 8 nút lá có giá trị lần lượt là 1, 2,…, 8, và gốc Root.

A diagram of a structure

Description automatically generated

***Hoạt động của cây Merkle***

Cây Merkle tính trị băm tóm tắt của tất cả các giao dịch trong một khối và tạo dấu vân kỹ thuật số (*fingerprint*) cho cả khối, cho phép người dùng xác minh xem nó có bao gồm giao dịch trong khối hay không. Tiến trình tạo cây băm Merkle như sau:

. Cây Merkle được tạo bằng cách băm các cặp nút liên tục cho đến khi chỉ còn lại một trị băm; trị băm này được gọi là Merkle Root hoặc Root Hash.

. Quá trinh băm được xây dựng từ dưới lên, sử dụng ID của giao dịch, là trị băm của các giao dịch riêng lẻ.

. Mỗi nút không phải nút lá là trị băm của trị băm trước đó và mỗi nút lá là trị băm của dữ liệu giao dịch.

Ví dụ: A, B, C và D là bốn giao dịch, tất cả đều được gom vào cùng một khối. Mỗi giao dịch sau đó sẽ được băm:

hA = H(A),

hB = H(B),

hC = H(C), và

hD = H(D).

Tiếp đến, các trị băm trên được ghép nối với nhau:

hAB = H(hA||hB),

hCD = H(hC||hD).

Cuối cùng, gốc cây Merkle được hình thành bằng cách kết hợp hai tri băm này:

Root = HABCD = H(hAB.hCD).

A diagram of a diagram

Description automatically generated

Trong thực tế, Merkle Tree phức tạp hơn nhiều (cụ thể, mỗi ID của giao dịch gồm 64 ký tự). Tuy nhiên, ví dụ trên giúp có cái nhìn tổng quan về cách các thuật toán hoạt động và lý do tại sao chúng lại hiệu quả đến vậy.

***Lợi ích của Merkle Tree trong Blockchain***

Cây Merkle có bốn lợi ích khi áp dụng cho chuỗi khối:

. Xác thực tính toàn vẹn của dữ liệu.

. Chiếm ít dung lượng đĩa.

. Thông tin nhỏ trên các mạng.

. Xác minh hiệu quả.

*Tính cần thiết của cây Merkle cho chuỗi khối*

Giả sử không sử dung cây Merkle thì mọi nút trên mạng sẽ phải giữ một bản sao của mọi giao dịch BTC từng được thực hiện. Như vậy, sẽ phải dữ lượng thông tin rất lớn.

Bất kỳ yêu cầu xác thực nào trên BTC sẽ yêu cầu một lượng dữ liệu khổng lồ phải truyền qua mạng: do đó, sẽ cần phải tự mình xác thực dữ liệu.

Để xác nhận tính toàn vẹn của mọi dữ liệu, sẽ cần rất nhiều sức mạnh thực hiện tính toán so sánh trên sổ cái.

Cây Merkle là một giải pháp cho vấn đề này. Chúng băm các bản ghi trong khối, từ đó tách bằng chứng dữ liệu khỏi chính dữ liệu đó. Diêu đó cho thấy chỉ cần cung cấp một lượng nhỏ thông tin trên mạng là tất cả những gì cần thiết để giao dịch có hiệu lực. Hơn nữa, có thể chứng minh rằng các biến thể sổ cái đều giống hệt nhau về sức mạnh máy tính danh nghĩa và băng thông mạng.

A diagram of blockchain technology

Description automatically generated

Cài đặt thực tế cây Merkle trong chuỗi khối

Có nhiều cách cài đặt cây Merkle hơn.

. **Git**, một hệ kiểm soát phiên bản phân tán, là một trong những hệ thống được sử dụng rộng rãi nhất. Nó được sử dụng để xử lý các dự án của các lập trình viên từ khắp nơi trên thế giới.

. **Hệ thống tệp liên hành tinh** – **IPFS** (*Interplanetary File System*), một giao thức phân tán ngang hàng. Là nguồn mở, cho phép các máy tính tham gia và sử dụng hệ thống tập tin phi tập trung.

. **Tạo chữ ký minh bạch**. Là một phần của kỹ thuật tạo ra nhật ký minh bạch của chứng chỉ có thể kiểm chứng được.

. **Amazon DynamoDB và Apache Cassandra** sử dụng cây Merkle trong quá trình sao chép dữ liệu. Các cơ sở dữ liệu phân tán No-SQL này sử dụng cây Merkle để kiểm soát sự khác biệt.

A screenshot of a computer code

Description automatically generatedA screenshot of a computer code

Description automatically generated

**Toán**

1. A screenshot of a math problem

   Description automatically generated**Ecuclide mở rộng:**

1. **CRT (Trung quốc):**

A screenshot of a math test

Description automatically generated

1. **Fermat nhỏ:**

**A screenshot of a math problem

Description automatically generatedA screenshot of a computer

Description automatically generated**