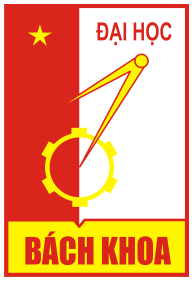
Trường Đại học Bách Khoa Hà Nội

Viện Toán ứng dụng và Tin học



ĐỒ ÁN TỐT NGHIỆP

Ứng dụng Chuỗi khối trong quản lý văn bằng giáo dục

Hướng dẫn:

TS. Đoàn Duy Trung

Thực hiện:

Đỗ Minh Tuấn - 20185419

Hà Nội, 2022

Nhận xét của giảng viên hướng dẫn

Mục đích và nội dung của Đồ án

Kết quả đạt được

Ý thức làm việc của sinh viên thực hiện

Hà Nội, ngày ... tháng ... năm ...

Giảng viên hướng dẫn

(Ký và ghi rõ họ tên)

Lời cảm ơn

Vậy là lễ tốt nghiệp sắp đến, có rất nhiều cảm xúc mà em không thể nói ra bằng lời. Em xin gửi lời cảm ơn đến *thầy Đoàn Duy Trung*, người đã định hướng và giúp đỡ em rất nhiều trong suốt chặng đường vừa qua. Em cảm ơn thầy.

Cùng với đó, em tự hào vì được là sinh viên CTTN Toán-Tin, hạnh phúc khi ở bên có những con người sẵn sàng chia sẻ từng chút "hiểu biết" cho nhau để ghép lên một bức tranh tuyệt đẹp về *tri thức của tình bạn*. Bạn bè giúp ta thấy được thời gian trôi nhanh và quý giá đến nhường nào, để ta cảm nhận được từng khoảnh khắc đáng nhớ trong cuộc sống vốn vội vã, xô bồ.

Tuy một học kỳ là ngắn ngủi, nhưng với định hướng rõ ràng, quả thực, nó mang lại cho em quá là nhiều kiến thức, đưa em đến những chân trời mới về công nghệ. Sau mỗi thành tựu mà ai đó đạt được, sẽ thật thiếu xót khi không nhìn vào sự nỗ lực của chính họ. Bản thân em cũng đã dành ra khá nhiều công sức tìm hiểu, thử nghiệm, triển khai, và mang đến kết quả phù hợp nhất cho đề tài mà mình đã chọn. Em tự tin với chất lượng của bài báo cáo này.

Mục lục

Chương 1

Tổng quan về bài toán quản lý văn bằng giáo dục

Từ lâu, các văn bằng giáo dục là một trong những mục tiêu của quá trình học tập không chỉ ở Việt Nam. Một "tấm bằng" sẽ ghi lại kết quả của một giai đoạn học và tích luỹ kiến thức của cá nhân, thể hiện bằng những điểm số qua các kỳ thi hay các bài kiểm tra cụ thể.

Với văn bằng ở các cấp độ giáo dục từ Trung học trở xuống, các thông tin thường được ghi là kết quả đánh giá tổng thể của một cấp học, đi kèm là một số giấy tờ bổ sung chi tiết về điểm số của học sinh và nhận xét của giáo viên, nhà trường đối với học sinh đó (thường được gọi là *Học bạ*). Đối với các văn bằng ở các cấp độ cao hơn như Trung cấp, Cao đẳng, Đại học, điểm số tổng kết và đánh giá năng lực của cơ sở giáo dục đối với học viên/sinh viên sẽ được thể hiện, cùng với đó là đính kèm bảng điểm chi tiết các học phần trong chương trình đào tạo mà học viên theo học tại cơ sở giáo dục đó (được gọi là *Phụ lục văn bằng*).

Ngoài văn bằng do các cơ sở giáo dục cấp phát, nhiều cơ quan, tổ chức cũng phát hành văn bằng đánh giá kết quả của cá nhân hoặc một nhóm cá nhân theo một lĩnh vực nào đó, thường dưới dạng các *chứng chỉ*, hay *giấy chứng nhận*.

Trong các hoạt động tuyển dụng nhân sự, văn bằng giáo dục (cùng với chứng chỉ) đem đến cho nhà tuyển dụng (các doanh nghiệp, cơ quan, tổ chức) cái nhìn tổng quát đầu tiên về năng lực của ứng viên tương ứng dựa trên đánh giá thể hiện qua thông tin trên văn bằng đó. Những ứng viên với một văn bằng cùng những thông tin tích cực có lợi thế rất lớn trong cuộc đua trở thành "người được chọn", bên cạnh việc thể hiện khả năng của mình trong quá trình làm việc. Việc có được sự đánh giá tốt từ những cơ sở giáo dục chất lượng, có uy tín cũng đem đến sự tin tưởng của nhà tuyển dụng đối với thông tin được ghi trong văn bằng.

Tuy nhiên, quy trình xác thực tính đúng đắn của một văn bằng giáo dục tại Việt Nam hiện nay gặp rất nhiều khó khăn. Phần nhiều là vì, các cơ sở giáo dục tại nước ta không công khai thông tin các văn bằng đã cấp phát bởi nhiều lý do. Thêm nữa, sự thiếu hụt các cổng tra cứu công cộng về văn bằng giáo dục cũng khiến các nhà tuyển dụng phải đặt niềm tin "tạm thời" vào một tờ bìa "có vẻ đáng tin cậy". Tính đến thời điểm bài báo cáo này được viết, *Bộ Lao động, Thương binh và Xã hội* đã cung cấp *Trang thông tin tra cứu văn bằng Giáo dục nghề nghiệp*. Tuy nhiên, đó vẫn là chưa đủ. Một hệ thống tra cứu thông tin về văn bằng với độ tin cậy cao và dễ sử dụng, áp dụng được trên mọi cơ sở giáo dục, tổ chức cấp phát văn bằng trở nên hết sức cần thiết.

Chương 2

Mật mã hoá khoá công khai và chữ ký số

2.1 Tổng quan về mật mã hoá khoá công khai

Mật mã hóa khóa công khai là một dạng mật mã hóa cho phép người sử dụng trao đổi các thông tin mật mà không cần phải trao đổi các khóa chung bí mật trước đó. Điều này được thực hiện bằng cách sử dụng một cặp khóa có quan hệ toán học với nhau là khóa công khai và khóa cá nhân (hay khóa bí mật).

Thuật ngữ mật mã hóa khóa bất đối xứng thường được dùng đồng nghĩa với mật mã hóa khóa công khai mặc dù hai khái niệm không hoàn toàn tương đương. Có những thuật toán mật mã khóa bất đối xứng không có tính chất khóa công khai và bí mật như đề cập ở trên mà cả hai khóa (cho mã hóa và giải mã) đều cần phải giữ bí mật.

Trong mật mã hóa khóa công khai, khóa cá nhân phải được giữ bí mật trong khi khóa công khai được phổ biến công khai. Trong 2 khóa, một dùng để mã hóa và khóa còn lại dùng để giải mã. Điều quan trọng đối với hệ thống là không thể tìm ra khóa bí mật nếu chỉ biết khóa công khai.

Hệ thống mật mã hóa khóa công khai có thể sử dụng với các mục đích:

- Mã hóa: giữ bí mật thông tin và chỉ có người có khóa bí mật mới giải mã được.
- Tạo chữ ký số: cho phép kiểm tra một văn bản có phải đã được tạo với một khóa bí mật nào đó hay không.
- Thỏa thuận khóa: cho phép thiết lập khóa dùng để trao đổi thông tin mật giữa 2 bên.

Thông thường, các kỹ thuật mật mã hóa khóa công khai đòi hỏi khối lượng tính toán nhiều hơn các kỹ thuật mã hóa khóa đối xứng nhưng những lợi điểm mà chúng mang lại khiến cho chúng được áp dụng trong nhiều ứng dụng.

2.1.1 Tính an toàn

Về khía cạnh an toàn, các thuật toán mật mã hóa khóa bất đối xứng cũng không khác nhiều với các thuật toán mã hóa khóa đối xứng. Có những thuật toán được dùng rộng rãi, có thuật toán chủ yếu trên lý thuyết; có thuật toán vẫn được xem là an toàn, có thuật toán đã bị phá vỡ... Cũng cần lưu ý là những thuật toán được dùng rộng rãi không phải lúc nào cũng đảm bảo an toàn. Một số thuật toán có những chứng minh về độ an toàn với những tiêu chuẩn khác nhau. Nhiều chứng minh gắn việc phá vỡ thuật toán với những bài toán nổi tiếng vẫn được cho là không có lời giải trong thời gian đa thức. Nhìn chung, chưa có thuật toán nào được chứng minh là an toàn tuyệt đối (như hệ thống mật mã sử dụng một lần). Vì vậy, cũng giống như tất cả các thuật toán mật mã nói chung, các thuật toán mã hóa khóa công khai cần phải được sử dụng một cách thận trọng.

2.1.2 Các ứng dụng

Ứng dụng rõ ràng nhất của mật mã hóa khóa công khai là bảo mật: một văn bản được mã hóa bằng khóa công khai của một người sử dụng thì chỉ có thể giải mã

với khóa bí mật của người đó.

Các thuật toán tạo chữ ký số khóa công khai có thể dùng để nhận thực. Một người sử dụng có thể mã hóa văn bản với khóa bí mật của mình. Nếu một người khác có thể giải mã với khóa công khai của người gửi thì có thể tin rằng văn bản thực sự xuất phát từ người gắn với khóa công khai đó.

Các đặc điểm trên còn có ích cho nhiều ứng dụng khác như: tiền điện tử, thỏa thuận khóa...

2.2 Hệ mật RSA

Thuật toán được Ron Rivest, Adi Shamir và Len Adleman mô tả lần đầu tiên vào năm 1977 tại Học viện Công nghệ Massachusetts (MIT). Tên của thuật toán lấy từ 3 chữ cái đầu của tên 3 tác giả. Đây là thuật toán đầu tiên phù hợp với việc tạo ra chữ ký điện tử đồng thời với việc mã hóa. Nó đánh dấu một sự tiến bộ vượt bậc của lĩnh vực mật mã học trong việc sử dụng khóa công cộng. RSA đang được sử dụng phổ biến trong thương mại điện tử và được cho là đảm bảo an toàn với điều kiện độ dài khóa đủ lớn.

Thuật toán RSA được MIT đăng ký bằng sáng chế tại Hoa Kỳ vào năm 1983 (Số đăng ký 4,405,829). Bằng sáng chế này hết hạn vào ngày 21 tháng 9 năm 2000. Tuy nhiên, do thuật toán đã được công bố trước khi có đăng ký bảo hộ nên sự bảo hộ hầu như không có giá trị bên ngoài Hoa Kỳ. Ngoài ra, nếu như công trình của Clifford Cocks đã được công bố trước đó thì bằng sáng chế RSA đã không thể được đăng ký.

Thuật toán dựa trên độ khó của bài toán phân tích một số thành nhân tử.

2.2.1 Quá trình tạo khoá cho hệ mật RSA

Giả sử hai người, An và Bình, cần trao đổi thông tin bí mật thông qua một kênh không an toàn (ví dụ như Internet). Với thuật toán RSA, An đầu tiên cần tạo ra cho mình cặp khóa gồm khóa công khai và khóa bí mật theo các bước sau:

1. Chọn hai số nguyên tố lớn khác nhau p và q.
2. Tính tích $n = pq$.
3. Tính một số giả nguyên tố của n bằng phi hàm Carmichael như sau:

$$\lambda(n) = \mathrm{GCD}(\lambda(p), \lambda(q)) = \mathrm{GCD}(p-1, q-1).$$

   Giá trị này sẽ được giữ bí mật.
4. Chọn một số tự nhiên e trong khoảng $(1, \lambda(n))$ sao cho $\mathrm{LCM(e, \lambda(n))} = 1$, tức là e và $\lambda(n)$ nguyên tố cùng nhau.
5. Tính toán số d sao cho $de \equiv 1 \pmod{\lambda(n)}$. Số d được gọi là số nghịch đảo modulo của e (theo modulo $\lambda(n)$).

Khoá công khai sẽ là bộ số (n, e), và khoá bí mật sẽ là bộ số (n, d). Chúng ta cần giữ khoá bí mật thật cẩn thận, cũng như các số nguyên tố p và q vì từ đó có thể tính ra các khoá khá dễ dàng.

2.2.2 Mã hoá và giải mã

Mã hoá

Giả sử Bình muốn gửi đoạn thông tin M cho An. Đầu tiên Bình chuyển M thành một số $m < n$ theo một hàm có thể đảo ngược (từ m có thể xác định lại M) được thỏa thuận trước. Quá trình này được mô tả ở phần **Chuyển đổi văn bản rõ**.

Lúc này Bình có m và biết n cũng như e do An gửi. Bình sẽ tính c là bản mã hóa của m theo công thức:

$$c \equiv m^e \pmod{n}.$$

Hàm trên có thể tính dễ dàng sử dụng phương pháp tính hàm mũ (theo modulo) bằng (thuật toán bình phương và nhân) Cuối cùng Bình gửi c cho An.

Giải mã

An nhận c từ Bình và biết khóa bí mật d. An có thể tìm được m từ c theo công thức sau:

$$c^d \equiv (m^e)^d \pmod{n} \equiv m^{de} \pmod{n}.$$

Do $de \equiv 1 \pmod{\lambda(n)}$, hay $de \equiv 1 \pmod{p-1}$ và $de \equiv 1 \pmod{q-1}$ (theo Định lý Fermat nhỏ), nên:

$$m^{de} \equiv m \pmod{p},\ m^{de} \equiv m \pmod{q}.$$

Lại do p và q nguyên tố cùng nhau, áp dụng Định lý Phần dư Trung Hoa, ta có:

$$m^{de} \equiv m \pmod{pq},$$

hay:

$$c^d \equiv m \pmod{n}.$$

Chuyển đổi văn bản rõ

Trước khi thực hiện mã hóa, ta phải thực hiện việc chuyển đổi văn bản rõ (chuyển đổi từ M sang m) sao cho không có giá trị nào của M tạo ra văn bản mã không an toàn. Nếu không có quá trình này, RSA sẽ gặp phải một số vấn đề sau:

- Nếu $m = 0$ hoặc $m = 1$ sẽ tạo ra các bản mã có giá trị là 0 và 1 tương ứng.
- Khi mã hóa với số mũ nhỏ (chẳng hạn $e = 3$) và m cũng có giá trị nhỏ, giá trị m^e cũng nhận giá trị nhỏ (so với n). Như vậy phép modulo không có tác dụng và có thể dễ dàng tìm được m bằng cách khai căn bậc e của c (bỏ qua modulo).

- RSA là phương pháp mã hóa xác định (không có thành phần ngẫu nhiên) nên kẻ tấn công có thể thực hiện tấn công lựa chọn bản rõ bằng cách tạo ra một bảng tra giữa bản rõ và bản mã. Khi gặp một bản mã, kẻ tấn công sử dụng bảng tra để tìm ra bản rõ tương ứng.

2.3 Hệ mật trên đường cong Elliptic

2.3.1 Đường cong Elliptic

Đường cong Elliptic có công thức tổng quát như sau:

$$y^2 \pmod p \equiv x^3 + ax + b \pmod p,$$

trong đó a, b là các số thực, p là một số nguyên dương ngẫu nhiên.

Viện Tiêu chuẩn và Kỹ thuật Quốc gia Mỹ (NIST) đặt ra tiêu chuẩn *secp256k1* với công thức của đường cong Elliptic như sau:

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

với p là một số nguyên tố rất lớn:

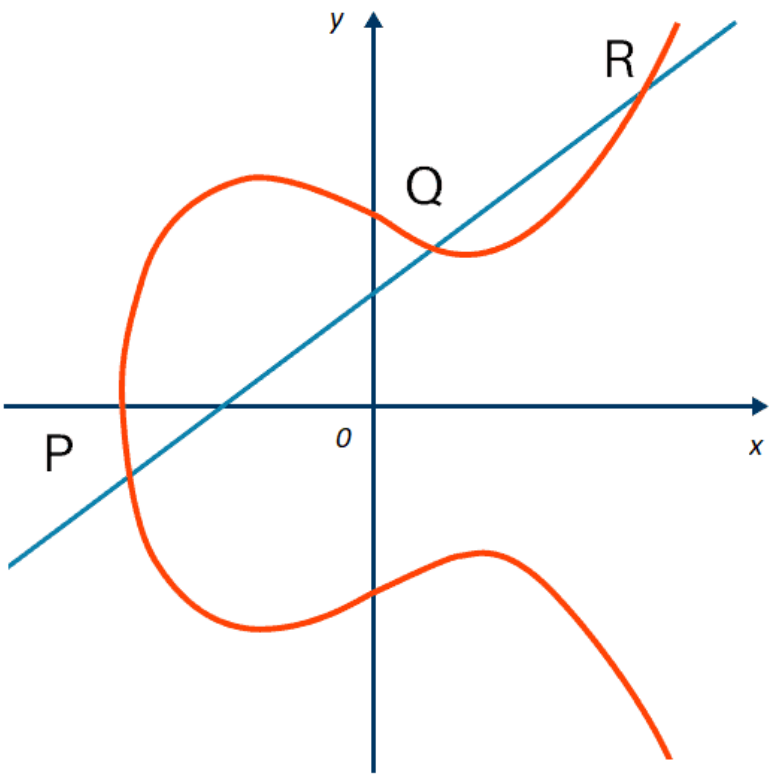
$$p = 2^{265} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1.$$

2.3.2 Các phép toán trên đường cong Elliptic

Có hai phép toán quan trọng trên đường cong Elliptic: Phép cộng và phép nhân.

Phép cộng

Đường cong Elliptic có một tính chất: "Nếu hai điểm P và Q nằm trên đường cong, thì điểm $P + Q$ cũng nằm trên đường cong đó.".



Hình 2.1: Hình dạng của đường cong Elliptic trên hệ trục toạ độ Oxy.

Điểm $P + Q$ được xác định như sau:

- Vẽ đường thẳng nối hai điểm P và Q, đường thẳng này cắt đường cong tại một điểm nữa, gọi là R.
- Lấy đối xứng của điểm R đó qua trục hoành, ta được điểm $P + Q$.

Và từ đó, ta có tính chất sau: "Nếu ba điểm trên đường cong Elliptic thẳng hàng, tổng của chúng bằng 0".

Phép nhân

Trên đường cong Elliptic, việc nhân một điểm với một hằng số không đơn thuần chỉ là lấy từng toạ độ rồi nhân là xong. Thực chất, phép nhân ở đây được thực hiện bằng cách lặp lại nhiều lần phép cộng.

Ví dụ trong phép tính $3P$, đầu tiên ta tính $2P$ bằng cách thực hiện $P + P$. Theo cách cộng ở bên trên, ta vẽ đường thẳng nối P với P (chính là tiếp tuyến của đường cong), nó cắt đường cong tại điểm $-2P$. Lấy đối xứng qua trục hoành, ta có điểm $2P$. Tiếp tục vẽ đường thẳng nối giữa $2P$ và P, cắt đường cong tại $-3P$, lấy đối xứng ta có $3P$.

Do cách tính toán trên, ta có thể dễ dàng tính toán được phép nhân kP khi biết k và P, nhưng hoàn toàn không thể tính toán được theo chiều ngược lại, tức phép chia. Đó cũng chính là tính chất đặc trưng thú vị của mã hoá bất đối xứng.

2.3.3 Tạo khoá công khai

Giả sử ta đã có khoá bí mật là một số ngẫu nhiên d_A. Trên đường cong Elliptic, ta chọn một điểm G, gọi là điểm sinh (generator point hay reference point).

Khoá công khai Q_A được sinh ra bằng kết quả của phép nhân:

$$Q_A = d_A \times G.$$

Tất nhiên Q_A cũng là một điểm trên đường cong Elliptic. Mối quan hệ giữa d_A và Q_A là cố định, và chỉ tính được theo một chiều từ d_A đến Q_A. Đó là lý do tại sao ta có thể sinh ra khoá công khai từ khoá bí mật và có thể chia sẻ khoá công khai này với tất cả mọi người, mà không thể dùng khoá công khai để tìm ngược lại về khoá bí mật.

2.4 Chữ ký số

Ứng dụng điển hình của mật mã khóa công khai là xác thực dữ liệu thông qua việc sử dụng chữ ký số. Chữ ký số thực chất là một chuỗi nhị phân đặc trưng cung cấp tính toàn vẹn, bằng chứng về nguồn gốc, danh tính và trạng thái của một tài liệu, giao dịch hoặc thông điệp điện tử. Chúng được sử dụng rộng rãi trong nhiều giao thức cho mục đích xác thực và đã được chứng minh là rất hữu ích và an toàn.

Chữ ký số là không thể sửa chữa và dễ dàng kiểm chứng nhờ sử dụng mật mã khóa công khai. Ngoài ra, do khóa riêng trong hệ mật khóa công khai chỉ liên kết đến một người dùng duy nhất, và chỉ có người dùng đó giữ khóa riêng, vì vậy chữ ký số cũng đảm bảo khả năng chống chối bỏ, có nghĩa là khi đã ký số lên dữ liệu thì người dùng không thể chối bỏ việc đã ký lên nó, và do vậy, chúng có thể ràng buộc về mặt pháp lý như chữ ký thông thường.

Bình thường, khi chúng ta sử dụng hệ mật mã khóa công khai để mã hóa và truyền dữ liệu, chúng ta sử dụng khóa công khai của người nhận để mã hóa, và do đó người nhận có thể dùng khóa riêng tương ứng của mình để giải mã dữ liệu, và chỉ duy nhất người nhận có khả năng làm việc đó. Tuy nhiên, trong chữ ký số, mọi chuyện lại ngược lại.

Chúng ta muốn xác nhận rằng thông điệp này đúng là do bản thân chúng ta ký vào nó, và mọi người đều có thể xác thực được. Do vậy, chữ ký số cần được giải mã bởi mọi người. Để đáp ứng điều này, chúng ta cần cho họ khóa chung (hay khóa công khai) và việc mã hóa diễn ra bằng khóa riêng bí mật của chính chúng ta.

Khi bạn mã hóa một cái gì đó bằng khóa riêng của mình, bất kỳ ai cũng có thể giải mã nó bằng khóa công khai, điều này nghe thật vô dụng, nhưng điều này đóng vai trò là bằng chứng cho việc bạn đã mã hóa dữ liệu. Nếu người khác

không thể giải mã đúng dữ liệu bằng khóa công khai của bạn, thì hoặc là dữ liệu đã bị thay đổi và mất tính toàn vẹn hoặc là bạn không hề ký và tạo ra dữ liệu đó. Đây được gọi là chữ ký số.

Trên thực tế, việc tạo và xác nhận chữ ký số phức tạp hơn rất nhiều, nó đòi hỏi có các thuật toán tạo chữ ký, sinh khóa, biến đổi dữ liệu, và thuật toán xác minh chữ ký. Mọi thứ tuân theo các lược đồ chữ ký số. Tuy nhiên về cơ bản nguyên lý hoạt động của chữ ký số đều tuân theo nguyên tắc trên.

Chữ ký số dựa trên đường cong Elliptic

Tạo chữ ký

Ta ký hiệu chữ ký số được biểu diễn bởi cặp số (r, s). Để tạo ra được cặp này, ta phải chọn ra một số ngẫu nhiên k (tất nhiên k khác với G trong công thức tìm khoá công khai Q_A từ khoá bí mật d_A ở trên).

Sau đó, ta nhân k với G để tạo ra một khoá công khai:

$$P = k \times G.$$

Lúc này, ta có được một điểm $P(x, y)$. Toạ độ x của P được lấy cho giá trị r.

Để tính được s, đầu tiên ta mã hoá nội dung thông điệp cần gửi m về dạng mã băm z. Khi đó:

$$s \equiv k^{-1}(z + d_A \times r) \pmod{p}.$$

Ta lưu ý, k^{-1} ở đây là nghịch đảo theo modulo p của k, tức là $k \times k^{-1} \equiv 1 \pmod{p}$.

Xác thực chữ ký

Để xác minh tính hợp lệ của chữ ký, ta chỉ cần khoá công khai Q_A là đủ. Ta tính:

$$P = s^{-1} \times z \times G + s^{-1} \times r \times Q_A.$$

Nếu toạ độ x của P bằng r trong chữ ký, thì chữ ký đó hợp lệ.

Ta có thể dễ dàng chứng minh điều này qua một số biến đổi sau:

$$\begin{aligned} P &= s^{-1} \times z \times G + s^{-1} \times r \times Q_A \\ &= s^{-1} \times z \times G + s^{-1} \times r \times (d_A \times G) \\ &= (z + r \times d_A) \times s^{-1} \times G \\ &= k \times s \times s^{-1} \times G \\ &= k \times G. \end{aligned}$$

Đây chính là công thức ta dùng để tính P khi tạo chữ ký.

Chương 3

Chuỗi khối và mạng Ethereum

3.1 Tổng quan về chuỗi khối

Một chuỗi khối là một sổ cái điện tử *phân tán*[1], *phi tập trung*[2], bao gồm các bản ghi được gọi là *khối* (block) thường được dùng để ghi lại các *giao dịch* (transaction) qua các máy tính.

Một chuỗi khối có thể bao gồm bảy lớp:

1. Cơ sở hạ tầng: Phần cứng
2. Mạng: Khám phá các nút mạng, chuyển tiếp thông tin, và xác thực
3. *Đồng thuận*[3]
4. Dữ liệu: Các khối, các giao dịch
5. Ứng dụng: *Hợp đồng thông minh*[4], *ứng dụng phi tập trung*[5]

[1]Distributed
[2]Decentralized
[3]Consensus
[4]Smart Contract
[5]Decentralized application (Dapp)

3.1.1 Khối

Các *khối* (block) nắm giữ các giao dịch hợp lệ đã được băm, quá trình mã hoá này sử dụng cấu trúc của một *cây Merkle*[6]. Mỗi khối bao gồm mã băm của khối liền trước trong chuỗi khối để liên kết với khối đó. Các khối liên kết với nhau định hình một *chuỗi* (chain). Quá trình tiếp diễn này xác minh tính toàn vẹn của khối trước đó, cho đến khối đầu tiên - được gọi là *khối bắt đầu*[7]. Để đảm bảo dữ liệu bên trong, các khối thường có *chữ ký số*[8].

Đôi khi các khối khác nhau được tạo ra đồng thời, tạo ra sự *phân nhánh tạm thời*[9]. Để bảo mật lịch sử băm, bất kỳ chuỗi khối nào đều có một thuật toán để "tính điểm" các phiên bản khác nhau, phần lịch sử của nhánh nào có "điểm số" lớn hơn sẽ được chọn cho chuỗi khối. Các khối không được chọn để đưa vào chuỗi khối được gọi là *khối mồ côi*[10]. Sự ngang hàng trong mạng khiến cho cơ sở dữ liệu có rất nhiều phiên bản/lịch sử theo thời gian, và phiên bản có "điểm số" cao nhất sẽ được giữ lại. Bất cứ khi nào một thành phần trong mạng nhận được một phiên bản có "điểm số" lớn hơn (thường là phiên bản cũ với một khối mới được thêm vào), nó sẽ mở rộng/ghi đè cơ sở dữ liệu và chuyển tiếp cho các thành phần ngang hàng khác ở trong mạng.

Thời gian khối

Thời gian khối[11] là thời gian trung bình để một mạng tạo ra một khối mới trong chuỗi khối. Ngay sau khi khối mới được tạo, dữ liệu trong nó sẽ được xác minh. Đối với tiền điện tử, các giao dịch diễn ra, một thời gian khối ngắn hơn đồng nghĩa với việc giao dịch sẽ nhanh hơn.

[6] Merkle tree
[7] Genesis block
[8] Digital signature
[9] Temporay fork
[10] Orphan block
[11] Block time

Phân nhánh hoàn toàn

Phân nhánh hoàn toàn[12] là việc thay đổi quy tắc trong chuỗi khối khiến các phần mềm xác thực dựa vào quy tắc cũ xác thực các chuỗi mới được tạo dựa trên quy tắc mới không hợp lệ. Trong trường hợp có một đợt phân nhánh hoàn toàn, tất cả các nút cần nâng cấp phần mềm để theo quy tắc mới. Nếu có một nhóm các nút tiếp tục sử dụng phần mềm cũ trong khi các nút còn lại sử dụng phần mềm mới, sự chia tách có thể xảy ra.

3.1.2 Phi tập trung

Bằng cách lưu dữ liệu thông qua *mạng ngang hàng*[13], chuỗi khối giảm được các rủi ro trong lưu trữ dữ liệu tập trung. Chuỗi khối phi tập trung sử dụng *truyền thông điệp tuỳ biến*[14] và *mạng phân tán*[15]. Khi thiếu đi tính phi tập trung, chuỗi khối có thể đối mặt với *tấn công 51%* (51% attack) - một thành phần trong mạng có thể kiểm soát nhiều hơn nửa phần còn lại của mạng và điều khiển các bản ghi trong chuỗi khối như ý muốn, trong đó có *chi tiêu gấp đôi* (double-spending).

Mạng chuỗi khối ngang hàng giảm thiểu khả năng bị khai thác tại các điểm tập trung dễ bị tấn công nào đó. Các phương pháp bảo mật trong chuỗi khối bao gồm việc sử dụng *mã hoá khoá công khai*[16]. Mỗi *khoá công khai* (public key), chuỗi ký tự dài ngẫu nhiên, là một địa chỉ (address) trong chuỗi khối. Các *token* gửi đi khắp mạng sẽ được ghi lại thuộc về một địa chỉ nào đó của chuỗi khối. Mỗi *khoá riêng tư* (private key) giống như một mật khẩu, giúp chủ sở hữu có thể truy cập vào *tài sản số*[17] của họ. Dữ liệu được lưu trong chuỗi khối được cho là không thể bị phá vỡ.

[12]Hard fork
[13]Peer-to-peer network (P2P network)
[14]Ad hoc message passing
[15]Distributed networking
[16]Public-key cryptography
[17]Digial asset

Mỗi *nút* (node) trong hệ thống phi tập trung giữ một bản sao của chuỗi khối. Chất lượng dữ liệu được duy trì bởi sự *nhân rộng cơ sở dữ liệu*[18] và sự tin cậy tính toán. Sẽ không có bản sao "chính" nào, cũng sẽ không có người dùng nào "đáng tin" hơn bất cứ ai trong hệ thống. Các giao dịch được gửi đi khắp/quảng bá (broadcasted) trong mạng qua phần mềm. Các thông điệp được gửi dựa trên *nền tảng nỗ lực tốt nhất*[19]. Các *nút đào*[20] xác thực giao dịch, thêm chúng vào khối mà nút đó đang tạo, và gửi khối đó đi khắp các nút trong mạng khi đã hoàn thành.

3.1.3 Tính mở

Các chuỗi khối mở *dễ dùng*[21] hơn so với các bản ghi truyền thống - mặc dù mở nhưng vẫn cần truy cập vật lý để xem. Tất cả các chuỗi khối trước đây đều *vô quyền* (permissionless). Nhiều tranh cãi đã nổ ra liên quan đến định nghĩa của chuỗi khối: Liệu một hệ thống riêng tư với xác thực viên được uỷ quyền bởi một *trung tâm uỷ quyền*[22] có được coi là một chuỗi khối hay không. Phía ủng hộ các chuỗi phân quyền hoặc riêng tư muốn thuật ngữ "chuỗi khối" có thể áp dụng với bất cứ cấu trúc dữ liệu phân dữ liệu thành các *khối đóng dấu thời gian*[23]. Phía phản đối điều này khẳng định các *hệ thống phân quyền* (permissioned system) giống như cơ sở dữ liệu truyền thống, không hỗ trợ xác thực dữ liệu phi tập trung, không thể chống lại sự giả mạo và sửa đổi.

Sự vô quyền

Một lợi thế của một mạng chuỗi khối mở, vô quyền, hoặc công khai là không cần bảo vệ chống lại các tác nhân xấu, không cần kiểm soát truy cập. Nghĩa là, các ứng dụng có thể được thêm vào mạng mà không cần sự chấp thuận và sự

[18] Database replication
[19] Best-effort basis
[20] Mining node
[21] User-friendly
[22] Central authority
[23] Time-stamped block

tin tưởng của các nút khác trong mạng, sử dụng chuỗi khối như là một *lớp vận chuyển*[24].

Chuỗi khối phân quyền/riêng tư

Các chuỗi khối phân quyền sử dụng một lớp điều khiển truy cập để quản lý những ai truy cập vào mạng. Trái với mạng chuỗi khối công cộng, xác thực viên trong mạng chuỗi khối riêng tư được kiểm tra chủ của mạng. Chủ của mạng không dựa vào các *nút ẩn danh*[25] để xác thực giao dịch hay các quyền lợi từ *hiệu ứng mạng*[26]. Các chuỗi khối phân quyền còn được biết đến với cái tên *chuỗi khối consortium*[27].

3.1.4 Các thuật toán đồng thuận phổ biến

Bằng chứng công việc - PoW

Bằng chứng công việc[28] (PoW) là một dạng của bằng chứng mã hoá, trong đó một bên chứng minh cho các bên khác (bên xác thực) rằng họ đã bỏ ra khối lượng tính toán nào đó. Bên xác thực sẽ tuần tự xác minh tính đúng đắn của bằng chứng này một cách dễ dàng.

PoW được đưa ra lần đầu bởi *Cynthia Dwork* và *Moni Naor* vào năm 1993 như một cách để xác định các cuộc *tấn công từ chối dịch vụ*[29], và các vấn đề liên quan đến lạm dụng dịch vụ như *spam* trong một mạng bằng cách yêu cầu một vài công việc bởi phía yêu cầu dịch vụ, thường là một quá trình tiêu tốn tài nguyên (thời gian, bộ nhớ) của máy tính. Thuật ngữ này được sử dụng lần đầu trong một bài báo của *Markus Jakobsson* và *Ari Juels*. Sau đó, nó được phổ biến bởi *Bitcoin*, như là một thuật toán đồng thuận đầu tiên trong mạng phi tập trung không phân

[24]Transport layer
[25]Anonymous node
[26]Network effect
[27]Consortium blockchain
[28]Proof of Work - PoW
[29]Denial-of-Service attack - DoS attack

quyền.

Trong mạng Bitcoin, các nút đào cần thực hiện giải một bài toán "khó" bằng cách tìm ra một con số được gọi là *nonce* sao cho sau khi kết hợp nó với các thông tin đã có của một khối trong chuỗi khối để thực hiện mã hoá băm, ta được một chuỗi băm bắt đầu với một chuỗi các ký tự "0" liên tiếp nhất định, và dãy số này cũng chính là địa chỉ của khối được tạo ra. Nút đầu tiên giải quyết được bài toán trên sẽ quảng bá khối đó lên toàn bộ mạng, các nút khác sẽ xác thực lại tính đúng đắn bằng cách sử dụng hàm băm để mã hoá lại thông tin của khối, so sánh với địa chỉ của khối: Nếu đúng, khối đó sẽ được thêm vào chuỗi, và nút tạo ra khối đó sẽ được nhận phần thưởng. Phần thưởng ở trong mạng Bitcoin chính là một lượng nhỏ đồng tiền mã hoá Bitcoin, và được gửi tới địa chỉ nút nhận ở khối nhất định sau khối vừa được thêm vào.

Do sự cải tiến về mặt công nghệ, các máy tính ngày nay có sức mạnh tính toán vô cùng lớn, tạo ra thách thức với các cơ chế đồng thuận dựa trên khối lượng tính toán, trong đó có PoW. Chính vì vậy, độ khó của bài toán mà các nút cần giải quyết ngày càng tăng lên. Tính tới thời điểm tháng 11 năm 2021, số lượng ký tự "0" liên tiếp trong phần đầu của địa chỉ khối trong mạng Bitcoin đã lên tới 7 chữ số, và thời gian để một giao dịch được thực thi trên mạng này xấp xỉ 10 phút. Thời gian thực thi giao dịch lâu khiến cho trải nghiệm của người dùng giảm thấp, đôi khi gây tắc nghẽn hệ thống mạng do nhiều giao dịch không được xử lý. Các cơ chế đồng thuận khác được đưa ra để giải quyết hạn chế này, trong đó có *Bằng chứng cổ phần* - PoS.

Bằng chứng cổ phần - PoS

Bằng chứng cổ phần[30] (PoS) là một lớp các cơ chế đồng thuận hoạt động bằng cách chọn ra các xác thực viên dựa trên tỷ lệ tiền mã hoá mà họ nắm giữ.

[30]Proof of Stake - PoS

Cơ chế PoS cho phép chủ sở hữu của các đồng tiền mã hoá "cọc" (stake) một lượng tiền để có thể trở thành một nút xác thực. *Cọc*[31] là khi một nút bỏ ra một số tiền để tham gia quá trình xác thực giao dịch. Số tiền này sẽ bị khoá khi *cọc*, và cần *huỷ cọc* (unstake) để có thể sử dụng giao dịch.

Khi một khối các giao dịch sẵn sàng để thực thi, cơ chế PoS sẽ chọn một nút xác thực (xác thực viên) để đánh giá khối đó. Xác thực viên sẽ kiểm tra thông tin các giao dịch trong khối có chính xác hay không, nếu đúng, khối đó sẽ được thêm vào chuỗi khối. Nút thêm khối mới vào chuỗi, tất nhiên, sẽ nhận được phần thưởng. Tuy nhiên, nếu nút đó thêm một khối với thông tin không chính xác, số tiền cọc của nút đó sẽ mất (một phần hoặc toàn bộ).

Mỗi cơ chế PoS có cách chọn xác thực viên khác nhau. Thông thường, quá trình lựa chọn ngẫu nhiên sẽ diễn ra, các yếu tố ảnh hưởng đến có thể kể ra như số lượng tiền mã hoá nút đó bỏ ra để cọc, nút đó tham gia quá trình cọc bao lâu rồi, vân vân. Mặc dù, ai cũng có thể tham gia cọc, nhưng tỷ lệ được chọn làm xác thực viên là rất thấp nếu số tiền bỏ ra để cọc. Vì lý do này, các thành viên tham gia quá trình cọc gia nhập vào các *bể cọc*[32]. Chủ của mỗi bể cọc sẽ thiết lập một nút tham gia quá trình xác thực, và các thành viên trong bể cọc sẽ "dồn tiền" để nút đó tham gia cọc để gia tăng cơ hội được chọn làm nút xác thực. Phần thưởng khi nút đó thêm khối mới vào chuỗi sẽ được chia cho các thành viên trong bể cọc đó.

Các cơ chế đồng thuận khác

Ngoài hai cơ chế đồng thuận phổ biến PoW và PoS, có rất nhiều các thuật toán đồng thuận khác. *Bằng chứng bộ nhớ*[33] (PoC) là một cơ chế cho phép chia sẻ không gian bộ nhớ của một nút cho mạng chuỗi khối. Nút nào có càng nhiều không gian bộ nhớ, nút đó càng có nhiều quyền duy trì mạng. *Bằng chứng hoạt động*[34] (PoA),

[31] Staking
[32] Staking pool
[33] Proof of Capacity - PoC
[34] Proof of Activity - PoA

được sử dụng trên chuỗi khối *Decred*, là một cơ chế kết hợp giữa PoW và PoS. *Bằng chứng tiêu thụ*[35] (PoB) lại yêu cầu các nút gửi lượng nhỏ tiền của chúng tới một địa chỉ ví không thể truy cập. Một cơ chế khác là *Bằng chứng lịch sử*[36] (PoH), được phát triển bởi *Solana Project*, tương tự như cơ chế *Bằng chứng thời gian còn lại*[37] (PoET), mã hoá thông tin thời gian trôi qua để đạt được sự đồng thuận mà không cần tiêu tốn nhiều tài nguyên.

3.2 Mạng Ethereum

3.2.1 Lịch sử

Ethereum là một nền tảng mã nguồn mở dựa trên chuỗi khối, hỗ trợ *Hợp đồng thông minh*[38]. Ethereum khá nổi với *đồng tiền mã hoá*[39] của nó với tên gọi là *Ether* (ký hiệu: ETH). Dựa vào sự phân tán của công nghệ chuỗi khối, Ethereum khá an toàn, và cũng nhờ bảo mật cao nên giá trị của đồng tiền ETH tích luỹ ngày càng lớn trên thị trường tiện điện tử.

Bắt đầu ý tưởng từ năm 2013 bởi lập trình viên *Vitalik Buterin* và một số cộng sự, công việc phát triển Ethereum được vận hành và kêu gọi vốn từ cộng đồng vào năm sau đó. Mạng Ethereum chính thức "lên sóng" vào ngày 30 tháng 7 năm 2015.

3.2.2 Các điểm khác biệt so với Bitcoin

Về nguồn gốc, Bitcoin được tạo ra như một loại tiền tệ và để lưu trữ giá trị. Còn Ethereum được tạo ra như một nền tảng giao dịch hợp đồng thông minh phân tán. Lưu ý rằng Bitcoin cũng có thể xử lý được hợp đồng thông minh, và

[35]Proof of Burn - PoB
[36]Proof of History - PoH
[37]Proof of Elapsed Time - PoET
[38]Smart contract
[39]Cryptocurrency

Ethereum cũng có thể được sử dụng như một loại tiền tệ. Ngoài ra, giữa Bitcoin và Ethereum còn có những điểm khác biệt cơ bản sau:

- Bitcoin có thể sử dụng để thanh toán hàng hóa và dịch vụ tại bất cứ nơi nào đồng tiền này được chấp nhận, còn đồng tiền Ether của mạng lưới Ethereum không được thiết kế như một giải pháp thanh toán thay thế, mà là để thúc đẩy các lập trình viên và các tổ chức sáng tạo và vận hành các ứng dụng phi tập trung trong mạng Ethereum.
- Thời gian tạo khối Ethereum mới là 14 tới 15 giây thay vì 10 phút trong Bitcoin.
- Việc sử dụng giao thức Ghost giúp giao dịch Ether nhanh hơn Bitcoin.
- Số lượng Bitcoin bị giới hạn ở mức 21 triệu với phần thưởng giảm còn một nửa sau mỗi 4 năm. Còn Ethereum thì không giới hạn số lượng ether. Lượng lạm phát ether hàng năm không được xác định rõ. Các ngân hàng trung ương thường thích Ethereum hơn vì cách phát hành tiền này.
- Phí giao dịch của Ethereum được trả bằng Gas (quy đổi được ra ether), được tính dựa trên khối lượng tính toán, băng thông, lưu trữ. Còn phí giao dịch Bitcoin bị cạnh tranh trực tiếp với nhau để vào được khối của Bitcoin mà bị giới hạn.
- Ethereum cho phép chạy mã Turing-complete, cho phép mọi tính toán được thực thi nếu có đủ khả năng tính toán và thời gian. Tuy nhiên điều này cũng mang lại nhiều rủi ro bị tấn công hơn cho Ethereum so với cấu trúc đơn giản hơn của Bitcoin.
- Có 13% số ether được bán cho lượng người đã tài trợ dự án ban đầu. Còn những người đầu tiên đào Bitcoin nắm giữ số lượng lớn lượng Bitcoin đang phát hành.
- Ethereum chống lại việc sử dụng ASIC như Bitcoin. Người đào Ethereum phải sử dụng card đồ họa vì hàm băm của Ethereum yêu cầu sử dụng bộ nhớ.

- Ethereum chống lại việc đào mỏ tập trung bằng cách sử dụng giao thức Ghost.
- Bitcoin đã có một lịch sử chưa bao giờ can thiệp vào dữ liệu trên sổ cái. Còn Ethereum đã phải chia nhánh sau khi DAO bị tấn công.

3.2.3 Kiến trúc

Máy ảo Ethereum

Máy ảo Ethereum (EVM) là một môi trường chạy các hợp đồng thông minh Ethereum. Định nghĩa chính thức của EVM được quy định trong Ethereum Yellow Paper của Gavin Wood. Nó được hoàn toàn cô lập từ mạng, hệ thống tập tin và các quá trình khác của hệ thống máy chủ. Mỗi nút Ethereum trong mạng chạy một EVM và thực hiện các hướng dẫn giống nhau. Ethereum Virtual Machines đã được lập trình trong C++, Go, Haskell, Java, Python, Ruby, Rust và WebAssembly (hiện đang được phát triển).

Hợp đồng thông minh

Nền tảng Ethereum còn hỗ trợ mạng lưới các *ứng dụng phi tập trung*[40]. Mạng này vận hành xoay quanh các hợp đồng thông minh. Phần lớn các ứng dụng sử dụng hợp đồng thông minh để liên kết với công nghệ chuỗi khối. Có thể nói, hợp đồng thông minh chính là nhân tố trung tâm của nền tảng Ethereum.

Hợp đồng thông minh là *hợp đồng tự thực thi*[41] với các điều khoản được viết bởi các dòng lệnh hay các đoạn mã lập trình. Các đoạn mã này tồn tại khắp các nút trong mạng chuỗi khối, điều hành sự thực thi các giao dịch, và không thể thay đổi. Hợp đồng thông minh mang đến các giao dịch đáng tin cậy, sự đồng ý với các điều khoản trong hợp đồng tới các bên "ẩn danh" mà không cần qua một bên

[40]Decentralized applications (DApps)
[41]Self-executed contract

trung gian hay một cơ chế thực thi bên ngoài.

Trên Ethereum, các đoạn mã của hợp đồng thông minh được viết bằng ngôn ngữ lập trình *Solidity* hoặc *Vyper*. Solidity là ngôn ngữ lập trình hướng đối tượng bậc cao dựa theo *C++*, *JavaScript*, *Python*, và được thiết kế để tích hợp được với *Máy ảo Ethereum*[42] (EVM). Vyper là ngôn ngữ đang trong quá trình thử nghiệm.

Tài khoản

Mỗi tài khoản Ethereum được đại diện bởi 20 ký tự. Các thông số sau được lưu trong dữ liệu trạng thái (state) của Ethereum cho mỗi tài khoản:

- Số nonce, để đảm bảo mỗi giao dịch chỉ được xử lý một lần.
- Số dư tài khoản.
- Mã nguồn hợp đồng (nếu có).
- Phần lưu trữ của tài khoản (mặc định là trống).

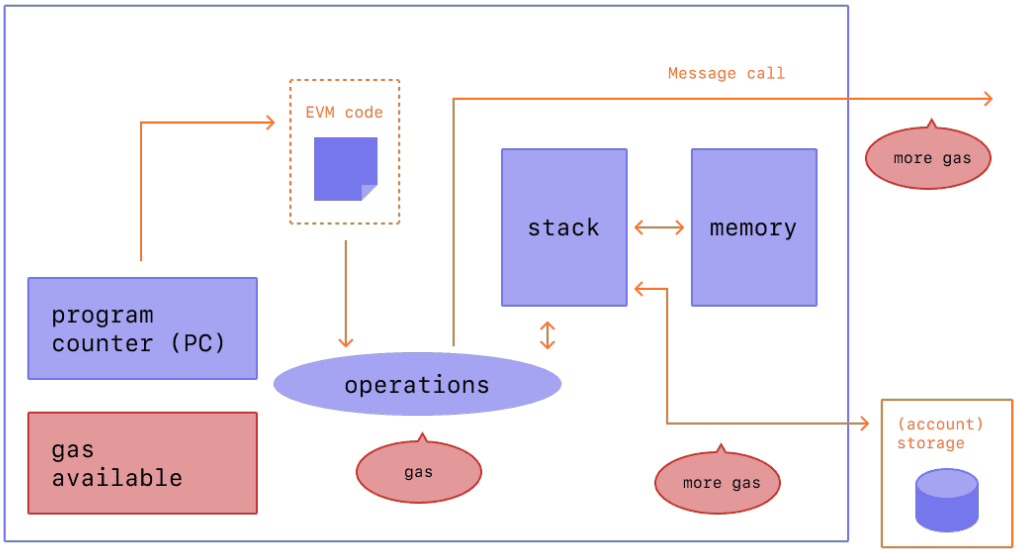
Các giao dịch giữa các tài khoản được trả tiền bằng Ether. Có hai loại tài khoản: Tài khoản ngoại vi được quản lý bởi khóa riêng tư, và tài khoản hợp đồng được quản lý bởi mã hợp đồng. Tài khoản ngoại vi không chứa mã hợp đồng, có thể gửi thông điệp đi bằng cách tạo và ký kết một giao dịch, giống như tài khoản Bitcoin. Về phía tài khoản hợp đồng, mỗi khi nó nhận được 1 thông điệp, mã hợp đồng sẽ chạy và cho phép đọc và ghi vào phần lưu trữ của nó, kèm theo việc gửi thông điệp đi và tạo ra hợp đồng khác lần lượt.

Lưu ý rằng "hợp đồng" trong Ethereum không phải là một cái gì đó phải "hoàn thành" hoặc "tuân thủ". Thay vào đó, nó giống như các "thực thể tự trị" sống bên trong môi trường Ethereum, luôn thực hiện một đoạn mã cụ thể khi được tác động bởi một thông điệp hoặc giao dịch, và có quyền kiểm soát trực số Ether và dữ liệu trong phần lưu trữ của nó.

[42]Ethereum Virtual Machine - EVM

Chi phí giao dịch

Gas là đơn vị thể hiện cho khối lượng tính toán để thực hiện một hành động nào đó trên mạng Ethereum. Do mỗi giao dịch trên mạng Ethereum đều cần tài nguyên tính toán để được thực thi, vì thế mà nó phát sinh ra *chi phí giao dịch*. Khi đó, *gas* thể hiện chi phí để thực hiện giao dịch thành công trên mạng.



Gas được trả bằng đồng *ether* (ETH). *Giá gas*[43] có đơn vị là *gwei*, mỗi *gwei* tương ứng với một phần một tỷ của một *ether*: 1 *gwei* $= 10^{-9}$ *ether*. Vì vậy, thay vì nói chi phí giao dịch là $0,000000001$ *ether*, ta có thể nói giao dịch đó tiêu tốn 1 *gwei*. Ngoài ra, 1 *gwei* chính là một tỷ *wei*; *wei* (được đặt tên theo *Wei Dai* - nhà khoa học máy tính nổi tiếng đưa ra lý thuyết về thanh toán bằng tiền mã hoá) là đơn vị nhỏ nhất trên Ethereum.

[43] Gas price

Chương 4

Hệ thống tập tin phân tán IPFS

4.1 Tổng quan về IPFS

IPFS là viết tắt của từ Interplanetary File System, một hệ thống tập tin phân tán ngang hàng kết nối tất cả các thiết bị máy tính với nhau. Cụ thể hơn, nó sẽ phân phối dữ liệu được lưu trữ theo hình thức P2P, hay còn gọi là mạng ngang hàng (mạng đồng đẳng).

Trong đó, các hoạt động của IPFS chủ yếu dựa vào khả năng tính toán băng thông của tất cả các máy tham gia chứ không tập trung vào một phần nhỏ các máy chủ trung tâm như giao thức HTTP. Nói cách khác, IPFS là mạng lưới chuyển phát nội dung hoàn toàn phi tập trung cho phép quản lý và lưu trữ dữ liệu một cách linh hoạt. Mỗi máy tính tham gia trong mạng lưới đảm nhận nhiệm vụ download và upload dữ liệu mà không cần sự can thiệp của máy chủ trung tâm.

4.2 Kiến trúc của IPFS

4.2.1 Đối tượng IPFS

Một đối tượng IPFS (IPFS Object hay còn được ký hiệu là IPLD) là một cấu trúc dữ liệu với hai trường:

- `Data`: Dữ liệu nhị phân không có cấu trúc có kích thước nhỏ hơn 256kB.
- `Links`: Chứa các liên kết (`Link`) đến các đối tượng IPFS khác.

Cấu trúc của `Link` gồm ba trường:

- `Name`: Tên của liên kết.
- `Hash`: Hàm băm của đối tượng IPFS được liên kết tới.
- `Size`: Kích thước tích luỹ của đối tượng IPFS được liên kết tới, bao gồm cả các liên kết sau đó nữa.

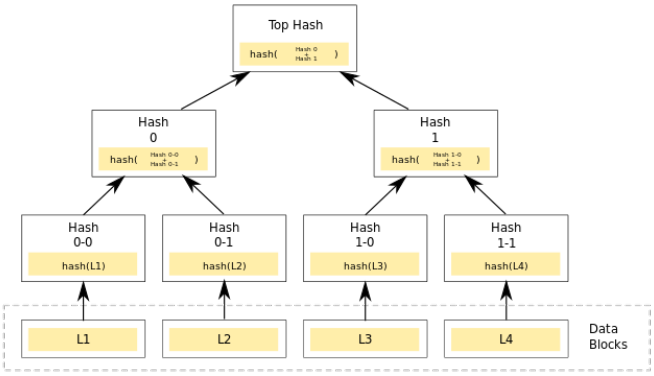
Trong đó, trường `Size` chủ yếu được sử dụng cho việc tối ưu hoá mạng P2P.

4.2.2 Merkle-DAG

DAG (Directed Acyclic Graph) là một dạng đồ thị có hướng, trong đó mỗi nút sẽ liên kết với các nút khác và không cho phép tạo thành chu trình có hướng. Một nút mà không là con của một nút nào khác trong đồ thị được gọi là nút gốc.

Merkle-DAG là một DAG trong đó mỗi nút có một định danh (identity hay id) là kết quả của việc mã hoá nội dung của nút đó. Điều này mang lại một số lưu ý:

- Các nút con phải được sinh trước thì các nút cha mới có id để liên kết tới.
- Mỗi nút trong Merkle-DAG là một nút gốc của một Merkle-DAG con nào đó.
- Các nút trong Merkle-DAG là không thể thay đổi. Bất kỳ thay đổi nào trong một nút sẽ làm thay đổi id của nút đó, và ảnh hưởng đến tất cả các nút khác.



Hình 4.1: Ví dụ về một Merkle-DAG.

4.2.3 Hệ thống tập tin

IPFS dễ dàng biểu diễn một hệ thống các tập tin và thư mục.

Các tập tin nhỏ

Một tập tin nhỏ được định nghĩa có kích thước nhỏ hơn 256kB, được biểu thị bằng một đối tượng IPFS với trường `Data` chứa nội dung của nó và trường `Links` là một danh sách rỗng.

Do tên tập tin không phải là một phần của đối tượng IPFS nên nếu có hai tập tin có cùng nội dung, chúng sẽ được biểu diễn bởi cùng một đối tượng IPFS.

Các tập tin lớn

Một tập tin lớn được định nghĩa có kích thước không dưới 256kB, được biểu thị bởi một Merkle-DAG của các đối tượng IPFS sao cho mỗi đối tượng có kích thước dữ liệu nhỏ hơn 256kB.

4.3 Cơ chế hoạt động

Đầu tiên mọi dữ liệu sẽ được mã hoá và được lưu dưới dạng mã băm (còn gọi là đối tượng IPFS). Ý tưởng chủ đạo là nếu trình duyệt của bạn muốn truy cập một trang nào đó trên IPFS thì chỉ cần đưa ra mã băm rồi mạng sẽ tìm máy có lưu trữ dữ liệu khớp với mã băm và sau đó tải dữ liệu, trang đó về từ máy tính đấy về cho bạn.

Cách thức hoạt động của IPFS sẽ tương tự như BitTorrent, đồng nghĩa với mỗi máy tính tham gia trong mạng lưới của nó sẽ đảm nhận cả việc tải xuống lẫn tải lên dữ liệu mà không cần có sự có mặt của một máy chủ trung tâm.

Tổng quan, cách hoạt động của IPFS sẽ có 2 phần chính:

- Xác định tệp có địa chỉ nội dung (giá trị băm của tệp đó).
- Tìm dữ liệu được lưu trữ và tải xuống: khi bạn có đoạn hash của tệp hay trang cần tải, mạng sẽ tìm và kết nối tới máy tốt nhất để tải dữ liệu xuống cho bạn.

4.4 Điểm độc đáo của IPFS

Nếu được triển khai đúng, IPFS mang lại tiềm năng lớn nhờ cải thiện được tốc độ truyền tải dữ liệu, tránh sự phụ thuộc vào các máy chủ và tiết kiệm chi phí.

4.4.1 Tránh sự phụ thuộc vào máy chủ

Trong các mô hình Máy khách - Máy chủ như HTTP, khi các máy chủ đang gặp phải sự cố thì chúng sẽ không thể hồi đáp thông tin cho người dùng. Đây cũng là vấn đề lớn nhất mà giao thức HTTP gặp phải khi nó phụ thuộc vào một máy chủ tập trung, điều mà nó không thể cải thiện cũng như khắc phục.

Với IPFS, nó hoàn toàn bỏ qua khái niệm máy chủ, mà chỉ quan tâm tới nội dung tìm kiếm. Điều này không chỉ giúp chúng ta rút ngắn con đường tới thông tin, mà lại không lo gặp phải các máy chủ kém chất lượng, kém tin cậy.

4.4.2 Mô hình phi tập trung

Với một mô hình tập trung, số lượng lớn dữ liệu được tập trung trong tay một số tên tuổi lớn trong lĩnh vực như Facebook, Amazon, Google,... Điều này vô tình khiến chúng trở thành tâm điểm cho các tin tặc tấn công. Trong lịch sử, không ít lần chúng ta chứng kiến những vụ rò rỉ thông tin liên quan đến những tên tuổi lớn.

Với mô hình phi tập trung của IPFS, các vấn đề này hoàn toàn được khắc phục và không còn chế độ quản lý phân cấp. Các dữ liệu được lưu trữ phân tán và không có một máy chủ tập trung để tấn công, càng nhiều người tham gia vào IPFS thì mạng sẽ càng bảo mật và khó có thể thao túng hơn.

4.4.3 Giảm bớt chi phí

Ưu điểm tiếp theo của mô hình IPFS đó là giảm bớt chi phí đối với cả người cung cấp nội dung và người dùng thông thường. IFPS sẽ cho phép đoạn video trên được tải hoàn toàn về mạng nội bộ IFPS dù bạn là ai và đang ở đâu. Do đó loại bỏ sự cần thiết của hàng loạt trạm kết nối và máy chủ Internet, giúp chi phí tổng thể giảm một cách rõ rệt.

Chương 5

Quản lý văn bằng giáo dục trên mạng Ethereum

5.1 Đề xuất giải pháp

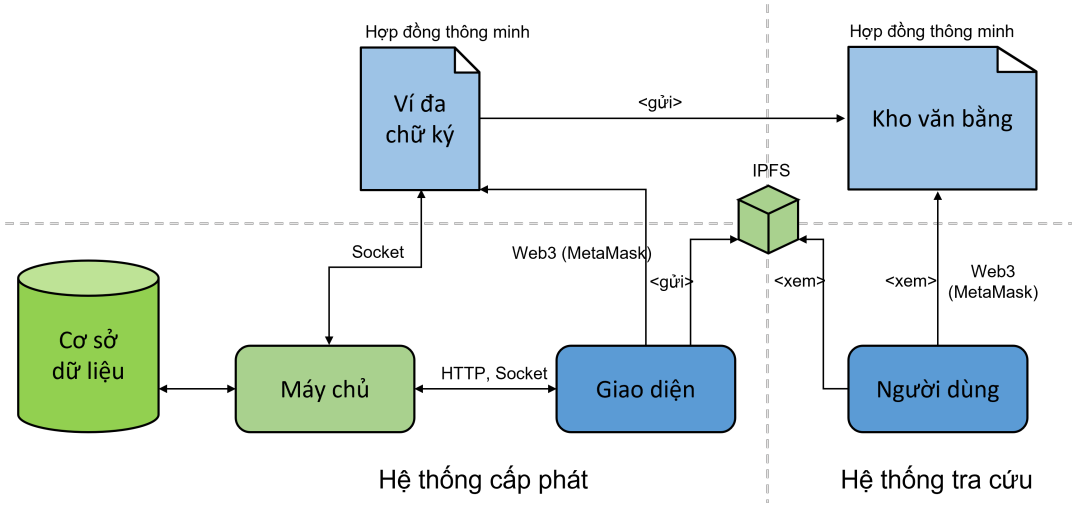
Giải pháp phổ biến hiện nay là xây dựng cơ sở dữ liệu chung về văn bằng giữa các cơ sở giáo dục. Nhà tuyển dụng có thể dễ dàng tra cứu thông tin văn bằng khi truy cập vào hệ thống sử dụng cơ sở dữ liệu này. Tuy nhiên, vấn đề hiện hữu là sử dụng cơ sở dữ liệu chung truyền thống tiềm ẩn rất nhiều rủi ro về dữ liệu. Việc nhiều bên truy cập và chỉnh sửa dữ liệu (các cơ sở giáo dục có quyền như nhau đối với cơ sở dữ liệu này) có thể phát sinh mất mát và ảnh hưởng đến dữ liệu của các bên khác. Ngoài ra, nếu giao quyền cập nhật dữ liệu cho một hoặc một số lượng hạn chế các bên tham gia, quy trình rà soát và sửa sai có thể kéo dài và các yêu cầu thay đổi dữ liệu được gửi từ các bên không có quyền cập nhật sẽ không được xử lý kịp thời.

Không phải nói quá, chuỗi khối giải quyết quá tốt các bài toán về cơ sở dữ liệu chung, đặc biệt là khi tính minh bạch của dữ liệu được ưu tiên. Đặc biệt, tương tác với chuỗi khối ngày càng trở nên đơn giản với *DApp* - các ứng dụng phi tập trung với giao diện thân thiện, dễ dùng, đồng thời tốc độ truy xuất thông tin

chấp nhận được, việc ứng dụng chuỗi khối càng được ưa chuộng. Việc lưu trữ thông tin văn bằng giáo dục trên chuỗi khối đảm bảo được dữ liệu không thể bị chỉnh sửa, mọi người đều dễ dàng truy cập. Nhiệm vụ cấp phát văn bằng được đưa về phía từng cơ sở giáo dục, lưu trữ trên mạng chuỗi khối chung, không thể chỉnh sửa. Đó là ưu điểm khi việc quản lý văn bằng sẽ không quá tập trung vào một số bên nhất định, nhưng cũng là thách thức cho các cơ sở giáo dục trong vấn đề xây dựng hệ thống tích hợp với mạng chuỗi khối một cách hợp lý và tốn không quá nhiều chi phí.

5.2 Xây dựng hệ thống

Với yêu cầu của bài toán, ta cần xây dựng hệ thống cấp phát văn bằng cho các cơ sở giáo dục, và hệ thống tra cứu thông tin văn bằng cho phía doanh nghiệp (hoặc người có nhu cầu tra cứu).



Hình 5.1: Sơ đồ tổng quan các hệ thống và mạng chuỗi khối

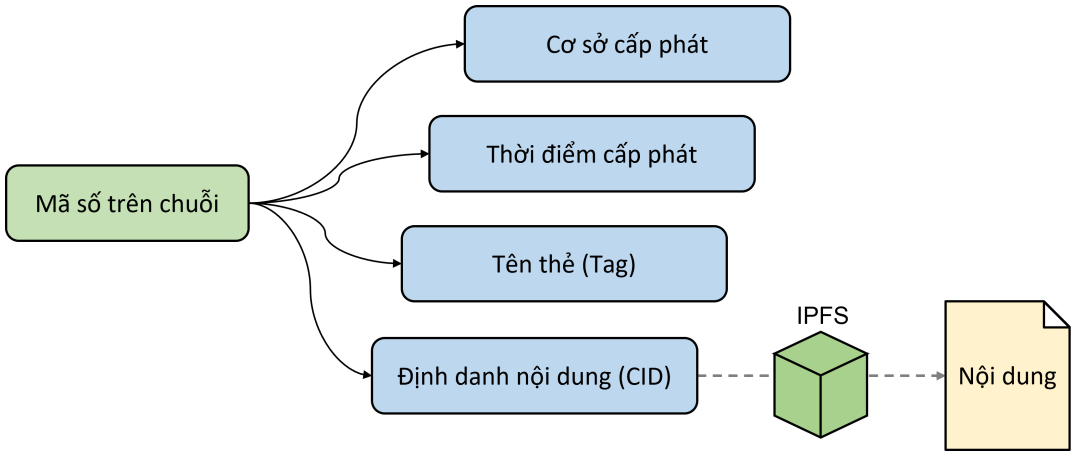
Những hệ thống này sẽ tương tác với các hợp đồng thông minh trên mạng chuỗi khối, bao gồm *Kho văn bằng* và *Ví đa chữ ký*. Trong bài báo cáo này, em xin phép tập trung trình bày về phần thiết kế các hợp đồng thông minh được triển khai cùng hệ thống.

5.2.1 Kho văn bằng

Kho văn bằng (hay *kho chứng chỉ*) là một hợp đồng thông minh nắm giữ thông tin các văn bằng được lưu trữ trên mạng chuỗi khối. Thông tin được tổ chức theo cấu trúc dạng cây, mỗi mã số văn bằng trên mạng chuỗi khối sẽ tương ứng với các thông tin liên quan đến văn bằng đã được cấp phát.

Để lưu trữ văn bằng trên mạng chuỗi khối, các cơ sở giáo dục cần sử dụng một

địa chỉ ví để tương tác với hợp đồng thông minh được triển khai trên mạng đó. Mỗi cơ sở phát hành văn bằng sẽ có một địa chỉ ví xác định, địa chỉ này sẽ được cơ quan như *Bộ Giáo dục và Đào tạo* hoặc *Chính phủ* ghi nhận là chữ ký đại diện cho cơ sở cấp phát văn bằng đó.



Hình 5.2: Cấu trúc lưu trữ thông tin trong *kho văn bằng*

Kho văn bằng cung cấp một số chức năng liên quan đến lưu thông tin, tra cứu thông tin văn bằng, phụ lục văn bằng, thông tin của các cơ sở giáo dục, tổ chức cấp phát văn bằng, chứng chỉ. Cụ thể, ba chức năng hiện có ở đây là:

- Lưu thông tin văn bằng
- Xem thông tin văn bằng
- Khoá văn bằng

Đối với việc *lưu thông tin*, địa chỉ ví của cơ sở (hay người) gửi yêu cầu lưu thông tin văn bằng sẽ được lấy làm *Cơ sở cấp phát*, và thông tin đi kèm sẽ được sử dụng làm *Tên thẻ* cho văn bằng. Như vậy, sẽ không có tình huống một cơ sở giáo dục phát hành văn bằng với địa chỉ của cơ sở khác, trường hợp nhầm lẫn do vô ý hoặc có chủ đích không thể xảy ra.

Để *xem thông tin*, người tra cứu chỉ cần cung cấp mã số văn bằng đã được cấp phát. Những thông tin được hợp đồng thông minh trả về bao gồm cả định danh nội dung trên mạng IPFS, hệ thống tra cứu sẽ tự động lấy nội dung văn bằng (bao gồm cả phụ lục văn bằng) dựa trên định danh này.

Đối với trường hợp văn bằng đã được cấp phát nhưng thông tin bị sai hoặc không phù hợp, phía cơ sở cấp phát văn bằng có thể lựa chọn *khoá văn bằng* lại. Người tra cứu không thể *xem thông tin* đối với những văn bằng đã bị khoá.

5.2.2 Ví đa chữ ký

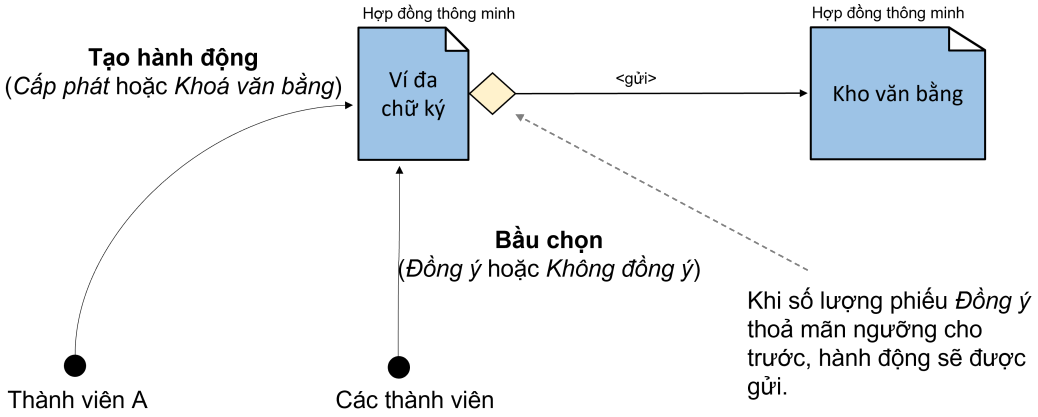
Ví đa chữ ký là một hợp đồng thông minh với mục đích tăng tính bảo mật cho quá trình tương tác thay đổi thông tin văn bằng trên chuỗi khối.

Với việc "đẩy" thông tin văn bằng lên mạng chuỗi khối một cách thông thường, mỗi cơ sở giáo dục sử dụng địa chỉ ví của một cá nhân đại diện để tương tác, hoặc lựa chọn một địa chỉ ví và sử dụng chung cho cá nhân trong cơ sở. Điều này đảm bảo mỗi cơ sở cấp phát chứng chỉ có một địa chỉ `School Address` duy nhất. Tuy nhiên, khi nhiều cá nhân cùng dùng một địa chỉ ví, khả năng mất cắp tài sản liên kết với địa chỉ này càng lớn, đặc biệt khi nó còn được sử dụng trong các giao dịch khác có giá trị về tài chính (như địa chỉ sở hữu tiền mã hoá với giá trị cao trên các *sàn giao dịch*[1], hay liên kết với các *DApp* khác). Do đó, một cơ chế giúp giảm thiểu khả năng nhiều người cùng sở hữu một địa chỉ ví và có thể sử dụng địa chỉ ví để xác thực thông tin cơ sở cấp phát văn bằng là vô cùng cần thiết. *Ví đa chữ ký* ra đời để giải quyết vấn đề này.

Không giống với các *hệ thống xác thực đa chữ ký*[2] khi ít nhiều phụ thuộc vào các cơ chế xác thực phức tạp, *ví đa chữ ký* sử dụng các tính năng, lợi thế của hợp đồng thông minh và mạng chuỗi khối. Ở *ví đa chữ ký*, mỗi hành động cần thực thi (ở đây là việc cấp phát văn bằng) yêu cầu một số lượng nhất định sự đồng ý từ cá

[1] Exchange
[2] Multi-signature authentication system

nhân. Địa chỉ ví của các cá nhân này đã được thêm vào danh sách "thành viên" ngay từ khi hợp đồng thông minh này được triển khai, và họ được coi như các "cổ đông" của "doanh nghiệp" cấp phát văn bằng khi có "tiếng nói" trong các "hoạt động" ở đây. Mỗi cơ sở cấp phát văn bằng sử dụng một *ví đa chữ ký* duy nhất, và địa chỉ của hợp đồng thông minh này đại diện cho địa chỉ ví của cả cơ sở đó. Các văn bằng cần được đẩy lên *kho văn bằng* sẽ được một cá nhân trong cơ sở gửi lên "ví" này. Các thành viên khác trong cơ sở có thể xem thông tin các văn bằng được gửi lên, và đưa ra biểu quyết "đồng ý" hay "không đồng ý" trên hợp đồng thông minh. Khi số lượng sự đồng ý đạt ngưỡng nhất định (được thiết lập từ đầu), các văn bằng đó được đẩy lên "kho", và thông tin được lưu trữ trên mạng chuỗi khối.



Hình 5.3: Cơ chế bầu chọn trong hệ thống

5.2.3 Các yêu cầu liên quan

Với *hệ thống tra cứu*, không có có quá nhiều yêu cầu cần thực hiện. Hiện nay, các tiện ích, phần mềm được tạo ra, đáp ứng nhu cầu kết nối đến mạng chuỗi khối, trong số đó có thể kể đến *Web3.js*. *Web3.js* là một dự án mã nguồn mở của *ChainSafe*, được viết chủ yếu bởi ngôn ngữ lập trình *JavaScript*, cung cấp các công cụ giúp người dùng tương tác với mạng Ethereum và các hợp đồng thông minh

trên mạng này. *Web3.js* có thể được tích hợp cho các *DApp* dạng *web*[3] (webapp). Việc tra cứu thông tin văn bằng trở nên đơn giản và hết sức thân thiện, khi người dùng có thể trực tiếp thao tác qua giao diện trên trình duyệt.

Hệ thống cấp phát cũng yêu cầu sử dụng cơ sở dữ liệu đơn giản để lưu trữ thông tin hành động (như cấp phát hay khoá văn bằng) trước khi chúng được thực thi trên mạng Ethereum. Ở đây, em sử dụng bảng `actions` với một số cột sau:

- `id`: Định danh của hành động trên *Ví đa chữ ký*, là số nguyên (tự động tăng trên hợp đồng thông minh), được sử dụng làm khoá chính cho bảng này.
- `executed`: Trạng thái thực thi của hành động, mang giá trị `true` (đã thực thi) hoặc `false` (chưa thực thi).
- `cancelled`: Trạng thái huỷ của hành động, mang giá trị `true` (đã bị huỷ) hoặc `false` (chưa bị huỷ).

Đồng thời, thông tin của quá trình bầu chọn cũng cần được ghi lại, tránh trường hợp một người bầu chọn nhiều lần cho một hành động (trường hợp này cũng không thể xảy ra do hợp đồng thông minh cũng đã kiểm tra các điều kiện phù hợp trước khi một người tham gia bầu chọn). Bảng `votes` được thiết kế cho việc này, bao gồm:

- `action`: Định danh cho hành động được bầu chọn.
- `voter`: Địa chỉ hay định danh của người bầu chọn.
- `affirmed`: Trạng thái đồng tình của người bầu chọn, mang giá trị `true` nếu người đó đồng ý với hành động, ngược lại là `false`.

Cuối cùng chính là thông tin các văn bằng. Bảng `contents` được thiết kế cho phần này:

- `id`: Mã số văn bằng trên `Kho văn bằng`, là số nguyên (tự động tăng trên hợp đồng thông minh).

[3]Website

- `cid`: Định danh nội dung hay địa chỉ nội dung của văn bằng trên mạng IPFS.
- `tag`: Chuỗi khác rỗng, được sử dụng cho mục đích tra cứu nội bộ và để ánh xạ giữa văn bằng trên hợp đồng thông minh với cơ sở dữ liệu. Trường này thường được gán giá trị bởi mã số học viên.

Ngoài ra, thông tin các thành viên được phép tham gia vào quá trình bầu chọn cũng như những thiết lập khác liên quan đến mạng Ethereum đều được lưu trữ trong cơ sở dữ liệu. Và ta cũng cần lưu ý rằng, phía máy chủ sẽ tự động cập nhật thông tin từ hợp đồng thông minh vào cơ sở dữ liệu; người dùng chỉ có quyền đọc dữ liệu từ đây (không có thao tác ghi vào cơ sở dữ liệu từ phía người dùng, người dùng tương tác trực tiếp với hợp đồng thông minh qua giao diện).

5.3 Kết quả triển khai và đánh giá

Triển khai

Giao diện *hệ thống quản lý* được xây dựng với thư viện *React* (được phát triển bởi đội ngũ *Facebook* với sự đóng góp của cộng đồng) sử dụng ngôn ngữ lập trình *TypeScript*. Đồng thời, người dùng cần đồng ý kết nối ví mã hoá *MetaMask* với hệ thống để sử dụng các tính năng tương tác với hợp đồng thông minh. Phía máy chủ, *Node.js* được lựa chọn, ta sử dụng *Express* để tạo các *API*[4]. Thông tin cần thiết được lấy từ cơ sở dữ liệu, và việc trao đổi giữa máy chủ và giao diện được thực hiện qua kết nối *HTTP* và *Web Socket*, các cập nhật từ một người sẽ được thông báo ngay lập tức tới các cá nhân khác trong cùng cơ sở cấp phát văn bằng.

Hệ thống tra cứu sử dụng cấu trúc *trang tĩnh*[5], triển khai trên *GitHub Pages* với mã nguồn công khai, cung cấp cho người dùng một công cụ tra cứu thông tin văn bằng nhanh chóng, đáng tin cậy. Các doanh nghiệp có thể lấy danh sách *địa chỉ ví* của các cơ sở giáo dục tại trang thông tin (website) của cơ sở giáo dục đó, hoặc lấy từ một cơ quan có độ tin cậy lớn (như *Bộ Giáo dục và Đào tạo* chẳng hạn). Thông tin số hiệu văn bằng sẽ được ứng viên cung cấp.

Dưới đây là một số hình ảnh khi người dùng trải nghiệm hệ thống, và em xin lưu ý *đây chưa phải là hình ảnh của hệ thống hoàn thiện*.

`Đang hoàn thiện...`

Đánh giá kết quả

Nhìn chung, các hệ thống ta đã trình bày ở trên đáp ứng tốt các yêu cầu của bài toán đã nêu. Tuy nhiên, một số hạn chế vẫn có thể chỉ ra, như:

[4] Application Programming Interface
[5] Static web

1. Hiện tại, các hợp đồng thông minh đang được triển khai trên *mạng kiểm thử*[6] (testnet) nên chi phí giao dịch chưa được đề cập. Nếu triển khai trên *mạng chính*[7] của Ethereum, phí giao dịch khá là cao.

2. Chưa hỗ trợ đầy đủ các tính năng cần có của một hệ thống cấp phát nội dung (như chỉnh sửa tại trang, thông báo thời gian thực, vân vân), cũng như giao diện chưa được bắt mắt.

Đối với *hạn chế về chi phí giao dịch*, đây là một bài toán khá đau đầu với những *DApp* triển khai trên mạng Ethereum. Tuy nhiên, trong những năm gần đây, rất nhiều giải pháp giảm chi phí và tăng tốc độ xác thực giao dịch trên mạng chuỗi khối đã được đưa ra, trong số đó có thể kể đến như *Plasma*, *Matic*.

Các phần bổ sung sẽ được em cân nhắc kỹ lưỡng cho vào thiết kế, phụ thuộc vào mức độ phù hợp với môi trường triển khai thực tế.

[6]Testnet
[7]Mainnet

Kết luận

Tính tới thời điểm này, chuỗi khối không còn gì là mới mẻ, nhưng còn rất nhiều vấn đề ta cần giải quyết để khai thác được tiềm năng tối đa của nó. BitCoin với mang đến sự thịnh hành của tiền mã hoá, Ethereum lại phổ biến hợp đồng thông minh và *DApp*, đem tới nhiều cái nhìn tích cực hơn về công nghệ này. Và rồi nay mai đây thôi, những giải pháp cải tiến mạng chuỗi khối tiếp tục ra đời, đồng thời sự phổ cập kiến thức về nó cũng sẽ được đẩy mạnh. Em tin rằng, không lâu nữa, rất nhiều bài toán sẽ có thêm lời giải hợp lý khi đi cùng chuỗi khối.

Cảm ơn thầy, cô, và mọi người đã theo dõi những gì em trình bày trên đây. Em rất mong nhận được sự góp ý, và cũng mong, những gì em viết ra, tạo ra sẽ sớm góp phần nhỏ vào sự phát triển chung, ít nhất là tại đất nước Việt Nam thân yêu này.

Như thường lệ, mọi mã nguồn được công khai và lưu trữ tại GitHub:

`https://github.com/DCerts`

Tài liệu tham khảo

[1] Diniel Drescher, *Blockchain Basics: A Non-Technical Introduction in 25 Steps*, 2017

[2] Investopedia, *Blockchain Definition: What You Need To Know*

[3] Investopedia, *Consensus Mechanism (Cryptocurrency) Definition*

[4] 101 Blockchains, *Blockchain Technology Explained: A Decentralized Ecosystem*

[5] R. C. Hansdah, *A Multisignature Scheme for Implementing Safe Delivery Rule in Group Communication Systems*

[6] Ethereum, *Ethereum Developement Documentation*

[7] ConsenSys Academy, *MultiSig. Wallet Excercise*

[8] ChainSafe, *Web3.js Documentation*

[9] Internet, *các thông tin về Chuỗi khối*