Decentralized Public Key Infrastructure - DPKI

Mục lục

[Chương 1: Hạ tầng khóa công khai PKI 4](#_Toc54031832)

[1.1 Cơ sở mật mã học 4](#_Toc54031833)

[1.1.1 Mã hóa khóa công khai. 4](#_Toc54031834)

[1.1.2 Hàm băm 5](#_Toc54031835)

[1.1.3 Chữ ký số. 6](#_Toc54031836)

[1.2 Hạ tầng khóa công khai PKI. 8](#_Toc54031837)

[1.2.1 Giới thiệu. 8](#_Toc54031838)

[1.2.2 Các thành phần của PKI. 10](#_Toc54031839)

[1.2.3 Các mô hình hạ tầng khóa công khai hiện nay. 13](#_Toc54031840)

[1.2.4 Chứng chỉ số và chuẩn x509. 15](#_Toc54031841)

[1.2.5 Độ an toàn của các mô hình PKI hiện nay. 18](#_Toc54031842)

[Chương 2: Decentralized Public Key Infrastructure 20](#_Toc54031843)

[2.1 Tổng quan. 20](#_Toc54031847)

[2.2 Công nghệ Blockchain 21](#_Toc54031849)

[2.2.1 Giới thiệu 21](#_Toc54031850)

[2.2.2 Quá trình xử lý giao dịch 23](#_Toc54031851)

[2.2.3 Cơ chế đồng thuận. 24](#_Toc54031852)

[2.3 Nền tảng Ethereum 27](#_Toc54031853)

[2.3.1 Ethereum 27](#_Toc54031854)

[2.3.2 A Vitural Machine và ngôn ngữ Solidity 28](#_Toc54031855)

[2.4 Mô hình triển khai chứng chỉ số trên Ethereum. 30](#_Toc54031856)

Lời mở đầu

Ngày nay với sự phát triển của công nghệ blockchain, thế giới đang có xu hướng đưa mọi thứ lên nền tảng này để tận dụng được những đặc tính ưu việt của nó. Đặc biệt là những lĩnh vực đòi hỏi tính minh bạch cao như tiền tệ, xử lý giao dịch, công chứng, danh tính và chứng minh nguồn gốc. Không ngoài xu hướng đó hạ tầng khóa công khai PKI, hạ tầng cho việc truyền thông an toàn trên internet cũng đã có những ý tưởng mon men về việc này và chắc chắn sẽ được hiện thực trong tương lai.

Mô hình hạ tầng khóa công khai hiện nay đang được chịu kiểm soát của các bên thứ 3. Cụ thể các thực thể này được gọi là CA (Certificate Authority) có nhiệm vụ quản lý, cấp phép và phân phối các chứng chỉ số chứa thông tin về các khóa công khai và danh tính tương ứng. Đặc biệt các hệ thống này đều là các hệ thống tập trung centralized, do vậy chúng được xem là mắt xích yếu nhất trong hệ thống, trở thành mục tiêu dễ dàng trong các cuộc tấn công, trở thành bàn đạp cho các Page | 3Page | 3cuộc tấn công sau này. Do vậy chúng ta cần một mô hình tiên tiến hơn để giảm thiểu rủi ro cho điều này,

Bản đồ án này sẽ đề cập đén Decentralized public key infrastructure DPKI, một mô hình hạ tầng khóa công khai mới tận dụng được sức mạnh của blockchain, xóa bỏ các bên thứ 3 giảm thiểu rủi ro tăng khả năng chịu lỗi của cả hệ thống.

Nội dung đồ án bao gồm

Chương 1: Giới thiệu về mật mã khóa công khai đây là các thành phần cốt lõi mà hạ tầng khóa công khai được xây dựng phục vụ xung quanh nó để đảm bảo an toàn. Tiếp theo chúng ta sẽ giới thiệu các thành phần của PKI và các mô hình PKI hiện tại. Phân tích các điểm yếu của các mô hình đó.

Chương 2: Giới thiệu về mô hình hạ tầng khóa công khai mới, Decentralized Public Key Infrastructure, trước đó ta sẽ đề cập đến công nghệ blockchain và ethereum và cách áp dụng chúng lên PKI, giải thích tại sáo mô hình mới này được xem là tân tiến hơn so với các mô hình truyền thống.

Chương 3: Cuối cùng là phần cài đặt, đây là một bản caì đặt demo thể hiện cách mà một PKI được triển khai trên Ethereum, cách nó được phân phối đến người dùng, và một kịch bản giả định cách mà DPKI được ứng dụng trong việc truyền thông an toàn.

# Chương 1: Hạ tầng khóa công khai PKI

## Cơ sở mật mã học

### Mã hóa khóa công khai.

Mã khóa công khai là một hệ mã hóa dùng 2 cặp khóa được gọi là public key và private key. Chỉ private key được giữ bí mật còn public key thì được công bố rộng rãi. 2 khóa này có quan hệ toán học với nhau nhưng rất khó để tìm được khóa còn lại khi biết một trong 2 khóa. Khi truyền thông tin người gửi sẽ sử dụng khóa bí mật mã hóa và gửi bản tin đã mã hóa cho người nhận. Người nhận sử dụng khóa công khai được người gửi công khai trên mạng để giả mã. Quá trình này có thể thực hiện ngược lại tùy vào mục đích sử dụng.



Cở sở toán học của hệ mật này là những bài toán khó. Vấn đề khó đây chính là việc số lượng phép tính cần thiết để tìm ra một lời giải cho bài toán là rất lớn. Trong lịch sử của ngành mã hóa bất đối xứng đã có nhiều đề xuất khác nhau cho dạng bài toán như vậy, tuy nhiên chỉ có hai trong số các đề xuất đó còn tồn tại vững đến ngày này. Hai bài toán đó bao gồm: bài toán Logarit rời rạc (discrete logarithhm problem) và bài toán phân tích thừa số của số nguyên.

Do cơ chế trên mà hệ mật mã hóa bất đối xứng có tính bảo mật cao nhưng nhược điểm là kích thước khóa lớn lên đến hàng ngàn bit và một nhược điểm khác là tốc độ xử lý tốn nhiều thời gian. Một số hệ mật mã hóa bất đối xứng như RSA,ECC.

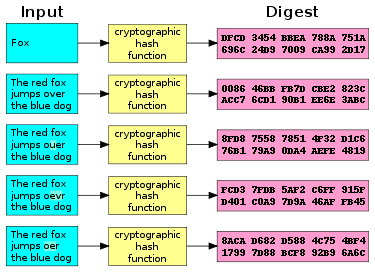
Mã hóa khóa công khai thường được sử dụng để.

* [Mã hóa](https://vi.wikipedia.org/wiki/M%C3%A3_h%C3%B3a): giữ bí mật thông tin và chỉ có người có khóa bí mật mới giải mã được.
* Tạo [chữ ký số](https://vi.wikipedia.org/wiki/Ch%E1%BB%AF_k%C3%BD_s%E1%BB%91): cho phép kiểm tra một văn bản có phải đã được tạo với một khóa bí mật nào đó hay không.
* [Thỏa thuận khóa](https://vi.wikipedia.org/w/index.php?title=Th%E1%BB%8Fa_thu%E1%BA%ADn_kh%C3%B3a&action=edit&redlink=1): cho phép thiết lập khóa để trao đổi thông tin mật giữa 2 bên

### Hàm băm

Hàm băm là một giải thuật chuyển một dữ liệu có kích thước tùy ý về dữ liệu có kích thước cố định. Giá trị băm đóng vai gần như một khóa để phân biệt các khối dữ liệu, tuy nhiên, người ta chấp hiện tượng trùng khóa hay còn gọi là [đụng độ](https://vi.wikipedia.org/w/index.php?title=%C4%90%E1%BB%A5ng_%C4%91%E1%BB%99_b%C4%83m&action=edit&redlink=1) và cố gắng cải thiện giải thuật để giảm thiểu sự [đụng độ](https://vi.wikipedia.org/w/index.php?title=%C4%90%E1%BB%A5ng_%C4%91%E1%BB%99_b%C4%83m&action=edit&redlink=1) đó.

Hàm băm thường được dùng trong [bảng băm](https://vi.wikipedia.org/wiki/B%E1%BA%A3ng_b%C4%83m) nhằm giảm chi phí tính toán khi tìm một khối dữ liệu trong một tập hợp (nhờ việc so sánh các giá trị băm nhanh hơn việc so sánh những khối dữ liệu có kích thước lớn). Hàm băm thường là hàm một chiều có nghĩa là ta dễ dàng tạo ra mã băm từ dữ liệu ban đầu nhưng để dịch ngược từ dữ liệu từ mã băm thường là rất khó. Việc giải mã dữ liệu từ giá trị băm tốn rất nhiều thời gian và tài nguyên đế mức bất khả thi.



Hình 1. 2 Ví dụ minh họa hàm Hash

Một số hàm băm thường dùng là các dòng SHA và MD như SHA1, SHA2, MD1, MD5.

### Chữ ký số.

Chữ ký số là một loại chữ ký điện tử được tạo ra bằng kỹ thuật mã hóa. Theo ngôn ngữ chuyên ngành chữ ký số được định nghỉa như sau:”Chứ ký số là một chuỗi dữ liệu được liên kết với một thông điệp và thực thể tạo ra thông điệp cho phép người nhận xác định được chủ sở hữu của thông điệp đó nhằm tránh việc thông điệp bị giả mạo”. Chữ ký số sử dụng mật mã khóa công khai để giảu mã cũng như mã hóa. Người gửi sẽ sử dụng private key của mình để ký còn người nhận sẻ sử dụng public key của người gửi để xác minh chữ ký. Do chữ ký số được tạo bằng kỹ thuật mã hóa nên độ an toàn của chữ ký phụ thuộc vào giải thuật mã hóa nhưng ta có thể khằng định chữ ký số có mức độ an toàn cao hơn hẳn chữ ký thường và một số chữ ký điện tử khác.

Bên cạnh định nghĩa về chữ ký số ta có một số định nghĩa theo sau như:

* Giải thuật tạo chữ ký số (Digital Signature generation algorithm) là một phương pháp sinh chữ ký số;
* Giải thuật kiểm tra chữ ký số (Digital Signature verification algorithm) là một phương pháp xác minh tính xác thực của chữ ký số, có nghĩa là nó thực sự được tạo ra bởi 1 bên chỉ định;
* Một hệ chữ ký số (Digital Signature Scheme) bao gồm giải thuật tạo chữ ký số và giải thuật kiểm tra chữ ký số.
* Quá trình tạo chữ ký số (Digital signature signing process) bao gồm:
  + Giải thuật tạo chữ ký số, và
  + Phương pháp chuyển dữ liệu thông điệp thành dạng có thể ký được.
* Quá trình kiểm tra chữ ký số (Digital signature verification process) bao gồm:
  + Giải thuật kiểm tra chữ ký số
  + Phương pháp khôi phục dữ liệu từ thông điệp.

Các bước của quá trình ký thông điệp

* Tính toán chuỗi đại diện của thông điệp cụ thể là mã hash của thông điêp.

Do quá trình ký sử dụng hệ mật mã hóa bất đối xứng nên dữ liệu ban đầu nếu có giá trị quá lớn thì tốc độ mã hóa và giải mã sẽ rất lâu mặt khác khi sử dụng mã hash dữ liệu ban đầu được nén vào một chuỗi cố định có độ lớn phù hợp để mã hóa. Việc này cung làm tăng tốc quá trình so khớp dữ liệu.

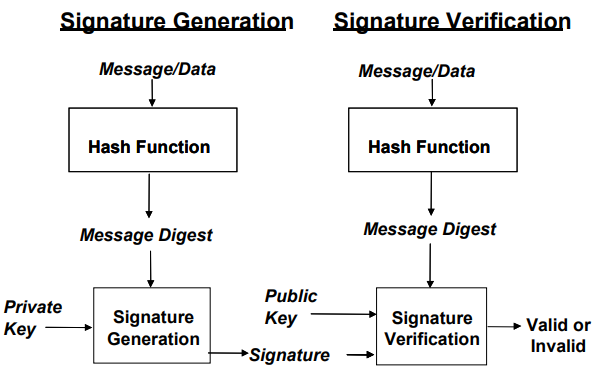
* Chuỗi đại diễn sẽ được ký với khóa riêng của người gửi bằng giải thuật tạo chứ ký.

Ở đây có thể hiểu là mã hóa chuỗi địa diện bằng thuật toán mã hóa bất đối xứng sử dụng khóa bí mật của người gửi.

* Thông điệp ban đầu được ghép vơi chữ ký tạo thành bản thông điệp được ký và gửi cho người nhận

Các bước của quá trình kiểm tra chữ ký:

* Tách chữ ký và thông điệp gốc ra khỏi thông điệp đã ký
* Tính toán chuỗi đại diện của thông điệp gốc ta được chuỗi đại diện 1
* Sử dụng khóa công khai của người gửi để giải mã chữ ký ta được chuỗi đại diện 2
* So sánh 2 chuỗi đại diện xem có khớp với nhau hay không.



Hình 1. 3 Quá trình ký và xác thực chữ ký

Trong cả quá trình ký và kiểm tra ta có thể thấy lợi ích của việc sử dụng hàm hash để tăng tốc độ mã hóa cũng như quá trình kiểm tra so khớp. Việc sử dụng hàm hash kết hợp với mã hóa khóa công khai giúp cho chữ ký số đảm bảo những tính chất sau.

* Tính toàn vẹn: Như đã nói ở trên việc sử dụng hàm hash để so sanh giúp chúng ta kiểm chứng tính toàn vẹn của thông điệp. Bất kỳ một thay đổi nào trong thông điệp được ký thì đều làm thay đổi giá trị hash của thông thông điệp nên ta có thể kiểm tra dễ dàng tính toàn vẹn của dữ liệu.
* Tính xác thưc: Bởi vì thông điệp được ký bởi khóa bí mật và chỉ được giải mã bới khóa công khai của người gửi nên ta có thể xác thực xem thông điệp có phải giả mạo hay không.
* Tính không chối bỏ: Rõ ràng, khóa bí mật dùng để giải mã chỉ có người giải mã giữ. Chính vì vậy, khi có vấn đề xảy ra có cơ sở để quy trách nhiệm các bên liên quan.

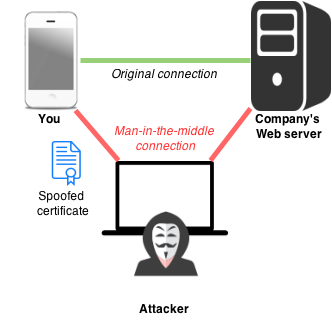
## Hạ tầng khóa công khai PKI.

### Giới thiệu.

Như chúng ta đã được đọc phía trên về mã hóa khoá công khai. Việc chúng ta quan tâm về giả thuật mã hóa ở đây không chỉ là độ an toàn của các hệ mật mà còn việc quản lý các khóa công khai. Chúng ta cần phải biết chính xác khóa công khai kia thuộc về ai, việc này hết sức quan trọng do nếu chúng ta sử dụng khóa công khai sai thì một bên thứ 3 có thể đọc được nội dung dữ liệu.

Gỉa sử Alice muốn truyền tin bí mật cho Bob. Việc đầu tiên mà Alice cần làm là lấy được public key (PK) của Bob, sau đó mã hóa dữ liệu với khóa PK này và gửi cho Bob. Bob sẽ sử dụng khóa secret key (SK) của mình để giải mã. Đối với các hệ mã công khai, có một kiểu tấn công rất phổ biến đó là Man-in-The-Middle (MiTM) attack. Trong đó, Eve sẽ chặn ở giữa phiên truyền tin của Alice và Bob:

* B1: Alice request khóa công khai PK của Bob
* B2: Bob gửi khóa công khai PK của mình về cho Alice.
* B3: Eve chặn ở giữa và gửi khóa công khai giả mạo PK' của mình về cho Alice.
* B4: Alice nhận được khóa giả mạo PK', mã hóa tin nhắn cần gửi M: C' = E(PK', M) và gửi C' cho Bob. Eve nhận được bản mã C', sử dụng khóa bí mật SK' của mình để giải mã và lấy được M = D(SK', C'). Sau đó mã hóa M với khóa PK của Bob: C = E(PK, M), rồi gửi bản mã C cho Bob.



Từ trên ta thấy rằng Bob và Alice vẫn nghĩ rằng phiên làm việc của mình an toàn, nhưng thực ra mọi thông điệp đều bị kẻ thứ 3 Eve nghe lén. Vấn đề cốt lõi ở đây là khóa công khai PK của Bob đã bị giả mạo bởi PK' của Eve, và Alice không có cách nào để xác minh liệu khóa mình nhận được có đúng là của Bob hay không?

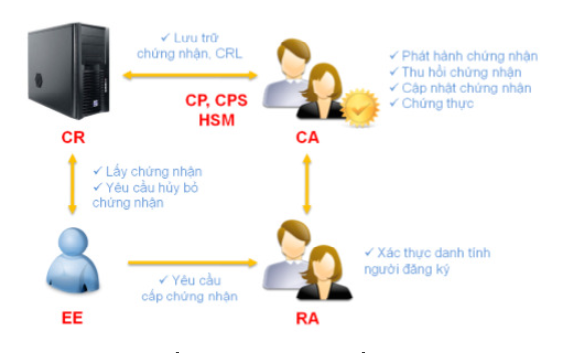
Do vậy hạ tầng khóa công khai được thiết lập để cung cấp một cơ chế để xác định khóa công khai thuộc về ai.

Theo định ngĩa Hạ tầng khóa công khai PKI là một tập hợp các tiêu chuẩn chính sách, phần cững cũng như phần mềm cần thiết để tạo, quản lý, lưu trữ, phân phối cũng như thu hồi chứng chỉ số. Mục đích của việc này là tạo điều kiện cho việc truyền thông an toàn trong môi trưòng không an toàn như internet, cung cấp bằng chứng xác định danh tính của các bên liên quan trong việc trao đổi thông tin. Từ đó cho phép sử dụng các công việc có tính bảo mật cao như xác thực, mã hóa giải mã văn bản, tạo chữ ký số lên văn bản ...

Theo định nghĩa ở trên ta có để cập đến chứng chỉ số. Ta có thể hiểu chứng chỉ số giống như chứng minh nhân dân (CMND) nhưng thay vì chứng minh một người là ai, nó chứng minh khóa công khai PK thuộc về ai. Tương tự, CMND được cấp phép bởi một cơ quan có thẩm quyền và tin cậy (để không bị làm giả) là bộ công an, chứng chỉ số được phát hành (được ký) bởi một bên thứ 3 tin cậy và có uy tín mà ta gọi là Certificate Authority (CA). Đây chính là 2 thành phần chính của hạ tầng khóa công khai.

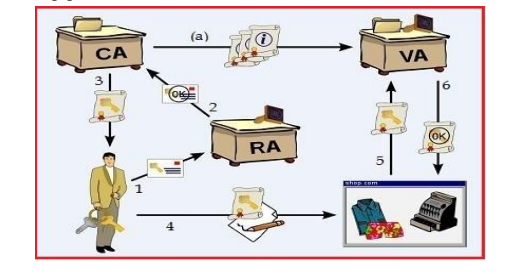
Chúng ta sẽ tìm hiểu chi tiết về chứng chỉ số và các thành phần cơ bản của PKI ở các phần tiếp theo.

### Các thành phần của PKI.



PKI gồm các thành phần chính sau:

* Thực thể cuối (End Entity – EE):
  + Đối tượng sử dụng chứng nhận (chứng thư số): có thể là một tổ chức, một người cụ thể hay một dịch vụ trên máy chủ, …
* Tổ chức chứng nhận(Certificate Authority – CA):
  + Có nhiệm vụ phát hành, quản lý và hủy bỏ các chứng thư số
  + Là thực thể quan trọng trong một PKI mà được thực thể cuối tín nhiệm
  + Gồm tập hợp các con người và các hệ thống máy tính có độ an toàn cao
* Chứng nhận khoá công khai (Public Key Certificate):
  + Một chứng nhận khóa công khai thể hiện hay chứng nhận sự ràng buộc của danh tính và khóa công khai của thực thể cuối
  + Chứng nhận khóa công khai chứa đủ thông tin cho những thực thể khác có thể xác nhận hoặc kiể m tra danh tính của chủ nhận chứng nhận đó
  + Định dạng được sử dụng rộng rãi nhất của chứng nhận số dựa trên chuẩn IETF X.509.
* Tổ chức đăng kí chứng nhận (Registration Authority – RA): Mục đích chính của RA là để giảm tải công việc của CA
  + Xác thực cá nhân, chủ thể đăng ký chứng thư số.
  + Kiểm tra tính hợp lệ của thông tin do chủ thể cung cấp.
  + Xác nhận quyền của chủ thể đối với những thuộc tính chứng thư số được yêu cầu.
  + Kiểm tra xem chủ thể có thực sự sở hữu khóa riêng đang được đăng ký hay không (chứng minh sở hữu).
  + Tạo cặp khóa bí mật, công khai. (nếu chủ thể yêu cầu)
  + Phân phối bí mật được chia sẻ đến thực thể cuối (ví dụ khóa công khai của CA).
  + Thay mặt chủ thể thực thể cuối khởi tạo quá trình đăng ký với CA.
  + Lưu trữ khóa riêng.
  + Khởi sinh quá trình khôi phục khóa
  + Phân phối thẻ bài vật lý (thẻ thông minh)
* Kho lưu trữ chứng nhận (Certificate Repository – CR):
  + Hệ thống (có thể tập trung hoặc phân tán) lưu trữ chứng thư và danh sách các chứng thư bị thu hồi
  + Cung cấp cơ chế phân phối chứng thư và danh sách thu hồi chứng thư (CRLs - Certificate Revocatio Lists).



Quá trình cấp chứng chỉ số.

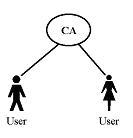
* (1) : Người dùng gửi yêu cầu phát hành thẻ chứng thư số và khóa công khai của nó đến RA;
* (2) : Sau khi xác nhận tính hợp lệ định danh của người dùng thì RA sẽ chuyển yêu cầu này đến CA;
* (3) : CA phát hành thẻ chứng thư số cho người dùng;
* (4) : Sau đó người dùng “ký” thông điệp trao đổi với thẻ chứng thư số mới vừa nhận được từ CA và sử dụng chúng (thẻ chứng thực số + chữ ký số) trong giao dịch;
* (5) : Định danh của người dùng được kiểm tra bởi đối tác thông qua sự hỗ trợ của VA (validation authority) : Cơ quan xác thực của bên thứ ba có thể cung cấp thông tin thực thể này thay mặt cho CA.)
* (6) : Nếu chứng thư số của người dùng được xác nhận tính hợp lệ thì đối tác mới tin cậy người dùng và có thể bắt đầu quá trình trao đổi thông tin với nó (VA nhận thông tin về thẻ chứng thư số đã được phát hành từ CA (a)

### Các mô hình hạ tầng khóa công khai hiện nay.

Tùy vào các yêu cầu, quy mô và khả năng của từng tổ chức mà có thể chọn triển khai một trong 2 mô hình PKI phổ biến sau:

* Hierarchical PKI
* Single CA

**Single CA**

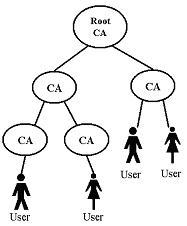
[](https://manthang.files.wordpress.com/2012/04/sing-ca.jpg)

Đây là mô hình PKI cơ bản nhất phù hợp với các tổ chức nhỏ trong đó chỉ có một CA cung cấp dịch vụ cho toàn hệ thống và tất cả người dùng đặt sự tin cậy vào CA này. Mọi thực thể muốn tham gia vào PKI và xin cấp chứng chỉ đều phải thông qua CA duy nhất này. Mô hình này dễ thiết kế và triển khai nhưng cũng có các hạn chế riêng. Thứ nhất là ở khả năng co giãn – khi quy mô tổ chức được mở rộng, chỉ một CA thì khó mà quản lý và đáp ứng tốt các dịch vụ. Hạn chế thứ hai là CA này sẽ là điểm chịu lỗi duy nhất, nếu nó ngưng hoạt động thì dịch vụ bị ngưng trệ. Cuối cùng, nếu nó bị xâm hại thì nguy hại tới độ tin cậy của toàn bộ hệ thống và tất cả các chứng chỉ số phải được cấp lại một khi CA này được phục hồi.

Nếu có nhiều CA đơn lẻ trong tổ chức nhưng lại không có các trust relationship giữa các CA được tạo ra thì bằng cách sử dụng trust list người dùng vẫn có thể tương tác với tất cả các CA. Lúc này các người dùng sẽ duy trì một danh sách các CA mà họ tin cậy. Các CA mới về sau có thể dễ dàng được thêm vào danh sách. Phương thức này tuy đơn giản nhưng cũng sẽ tốn thời gian để cập nhật hết các CA cho một lượng lớn người dùng, mặt khác nếu một CA nào đó bị thỏa hiệp thì không có một hệ thống cảnh báo nào báo cho những người dùng mà tin cậy CA đó biết được sự cố này.

**Hierarchical PKI**

Đây là mô hình PKI được áp dụng rộng rãi trong các tổ chức lớn. Có một CA nằm ở cấp trên cùng gọi là root CA, tất cả các CA còn lại là các Subordinate CA (gọi tắt là sub. CA) và hoạt động bên dưới root CA. Ngoại trừ root CA thì các CA còn lại trong đều có duy nhất một CA khác là cấp trên của nó. Hệ thống tên miền DNS trên Internet cũng có cấu trúc tương tự mô hình này.

[](https://manthang.files.wordpress.com/2012/04/hierarchical-pki.jpg)

Tất cả các thực thể (như người dùng, máy tính) trong tổ chức đều phải tin cậy cùng một root CA. Sau đó các trust relationship được thiết lập giữa các sub. CA và cấp trên của chúng thông qua việc CA cấp trên sẽ cấp các chứng chỉ cho các sub. CA ngay bên dưới nó. Lưu ý, root CA không trực tiếp cấp chứng chỉ số cho các thực thể mà chúng sẽ được cấp bởi các sub. CA. Các CA mới có thể được thêm ngay dưới root CA hoặc các sub. CA cấp thấp hơn để phù hợp với sự thay đổi trong cấu trúc của tổ chức. Sẽ có các mức độ tổn thương khác nhau nếu một CA nào đó trong mô hình này bị xâm hại.

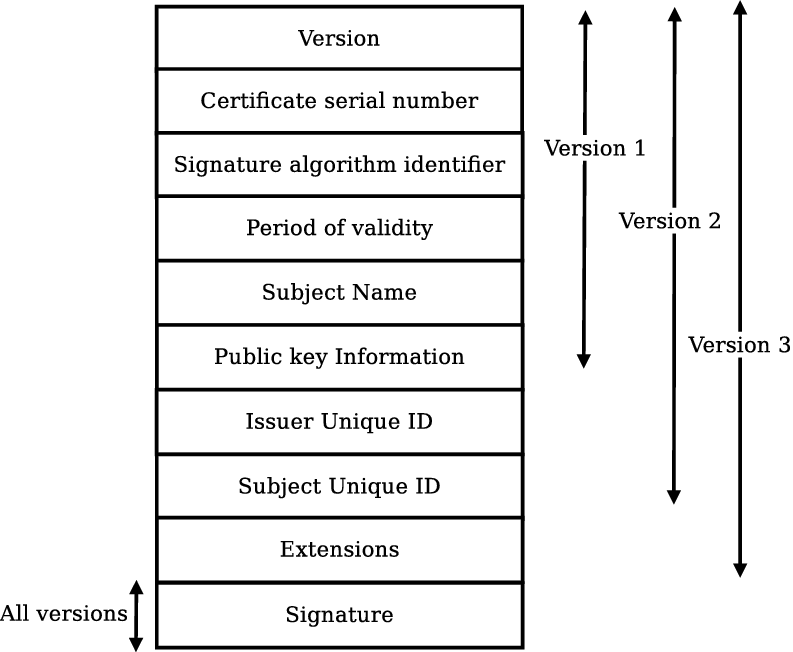
Trường hợp một sub. CA bị thỏa hiệp thì CA cấp trên của nó sẽ thu hồi chứng chỉ đã cấp cho nó và chỉ khi sub. CA đó được khôi phục thì nó mới có thể cấp lại các chứng chỉ mới cho người dùng của nó. Cuối cùng, CA cấp trên sẽ cấp lại cho nó một chứng chỉ mới.

Nếu root CA bị xâm hại thì đó là một vấn đề hoàn toàn khác, toàn bộ hệ thống PKI sẽ chịu ảnh hưởng. Khi đó tất cả các thực thể cần được thông báo về sự cố và cho đến khi root CA được phục hồi và các chứng chỉ mới được cấp lại thì không một phiên truyền thông nào là an toàn cả. Vì thế, cũng như single CA, root CA phải được bảo vệ an toà

### Chứng chỉ số và chuẩn x509.

Chứng chỉ số digital certificate hay public key certificate la một loại dữ liệu kỹ thuật số chứng minh quyền sở hữu của một cá nhận hay tổ chức đối với một khóa công khai. Chứng chỉ số sẽ chứa những thông tin xác đinh danh của chủ sở hữu, khóa công khai và chữ ký số của bên xác thực cấp phép lên chứng chỉ này. Nếu chữ ký số hợp lệ có nghĩa là chữ ký của bên cung cấp đối với chúng chỉ này đúng thì người dùng cuối có thể tin tưởng khóa công khai của đối tượng này và sử dụng chúng trong việc trao đổi thông tin an toàn như sử dụng trong chữ ký số, ssl/tls, email encryption. Ngoài ra còn có một số thông tin khác giúp hỗ trợ cho việc xác thực. Việc phân phối xác thực và thu hổi chứng chỉ số chính là công việc của hạ tầng khóa công khia PKI. Đơn vị thực hiện việc này được gọi là CA certificate authority.

Chuẩn chữ ký số được dùng phổ biến nhất hiện nay đó là x509. Chuẩn này có 3 phiên bản v1, v2 và v3. Chuẩn này gồm các trường.



**X.509 version 1**

* Version: chứa giá trị cho biết đây là chứng chỉ X.509 version 1
* Serial Number: cung cấp một mã số nhận dạng duy nhất cho mỗi chứng chỉ được phát hành bởi CA
* CA Signature Algorithm: tên của thuật toán mà CA sử dụng để ký lên nội dung của chứng chỉ số.
* Issuer Name: tên phân biệt (distinguished name) của CA phát hành chứng chỉ. Thường thì tên phân biệt này được biểu diễn theo chuẩn X.500 hoặc định dạng theo đặc tả của X.509 và RFC 3280.
* Validity Period: khoảng thời gian mà chứng chỉ được xem là còn hiệu lực, bao gồm 2 trường là: Valid From và Valid To.
* Subject Name: tên của máy tính, người dùng, thiết bị mạng sở hữu chứng chỉ. Thường thì tên chủ thể này được biểu diễn theo chuẩn X.500 hoặc định dạng theo đặc tả của X.509, nhưng cũng có thể bao gồm các định dạng tên khác như được mô tả trong RFC 822.
* Subject Public Key Info: khóa công khai của đối tượng nắm giữ chứng chỉ. Khóa công khai này được gửi tới CA trong một thông điệp yêu cầu cấp chứng chỉ (certificate request) và cũng được bao gồm trong nội dung của chứng chỉ được phát hành sau đó. Trường này cũng chứa nhận dạng của thuật toán được dùng để tạo cặp khóa công khai và khóa bí mật được liên kết với chứng chỉ.
* Signature Value: chứa giá trị của chữ ký.

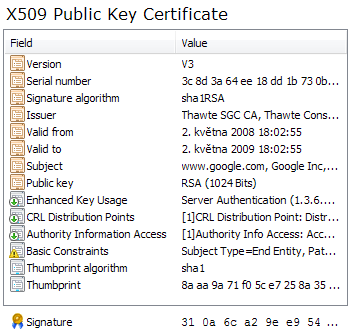
**X.509 version 2**

Mặc dù chứng chỉ X.509 version 1 cung cấp khá đầy đủ những thông tin cơ bản về người nắm giữa chứng chỉ nhưng nó lại có ít thông tin về tổ chức cấp phát chứng chỉ khi chỉ bao gồm *Issuer Name, CA Signature Algorithm* và *Signature Value*

* Issuer Unique ID: là một trường không bắt buộc, chứa chuỗi giá trị ở hệ 16, mang tính duy nhất và dành để nhận dạng CA. Khi CA thay mới chứng chỉ của chính nó, một Issuer Unique ID mới được khởi tạo cho chứng chỉ đó.
* Subject Unique ID: là một trường không bắt buộc, chứa chuỗi giá trị ở hệ 16, mang tính duy nhất và dùng để nhận dạng chủ thể của chứng chỉ. Nếu chủ thể này cũng chính là CA thì trường này sẽ giống với Issuer Unique ID.

**X.509 version 3**

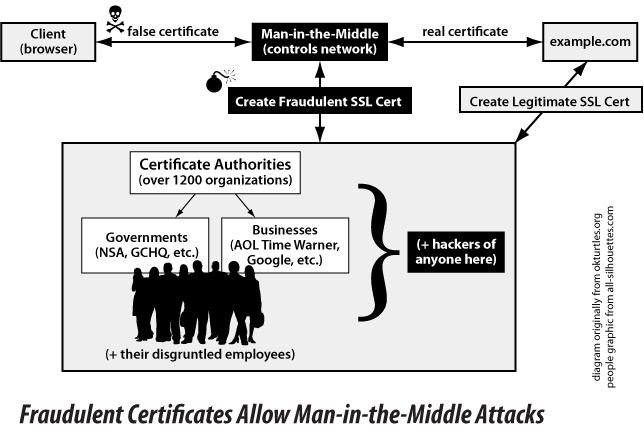
Được ra đời vào năm 1996, định dạng X.509 version 3 được bổ sung thêm các phần mở rộng (*extension*) để khắc phục các vấn đề liên quan tới việc so khớp Issuer Unique ID và Subject Unique ID cũng như là các vấn đề về xác thực chứng chỉ. Một chứng chỉ X.509 version 3 co thể chứa một hoặc nhiều extension



Ta có thể dễ dàng truy cập chứng chỉ số của môt trang web khi truy cập chorme bẳng các kích chọn vào hình ổ khóa bên trái màn hình.

### Độ an toàn của các mô hình PKI hiện nay.

Như chúng ta đã biết để quản lý các chứng chỉ số, hạ tầng khóa công khai được hình thành để đảm bảo các chứng chỉ này được xác thực một cách đúng đắn, nòng cốt trong việc này chính là các CA. Nhưng chính CA được xem là điểm yếu nhất tromg cả hạ tầng khóa công khai. Với một hệ thống được quản lý bởi một bên thứ 3 có thể là một công ty hay một tổ chức chính phủ nòa đó và được thiết kế theo mô hình tập trung việc có thể bị tấn công cũng như bị thỏa hiệp là một điều không thể tránh khỏi. Từ đó cho phép kẻ tấn công thực hiên các cuộc tấn công Man in the Middle mạo danh để thu thập các dữ liệu quan trọng, và đặc biệt kiểu tấn công này rất khó để mà phát hiện nhưng ảnh hưởng của nó đến tính bí mật cảu các thông tin rất cao.



Kịch bản tấn công Man in the middle khá đơn giản. Kẻ tấn công làm cách nào đó chiếm được đường truyền giữa 2 bên và thỏa hiệp được với một CA. Từ đó chúng có thể tạo ra các chứng chỉ số giả chứa thông tin định dạng của tổ chức đó và một khóa công khai khác của kẻ tấn công, và gửi chứng chỉ này cho người truy cập. Do chứng chỉ này hợp lệ do được cấp bởi một bên hợp lệ, nên quá trình truy vết chứng chỉ đến node root đểu được hợp pháp. Vì vậy người dùng tin tưởng mà sử dụng khóa. Từ đó kẻ tấn công có thể truy cập nội dung truyền tải với cơ chế Man in the middle như đã đề cập ở trên.

Trong thực tế các cuộc tấn công vào CA đã từng xảy ra. Năm 2011 một cuộc tấn công vào DigiNotar một công ty cung cấp chứng chỉ số của Hà Lan đã bị tấn công và qua đó cung cấp các chứng chỉ giả để thực hiện các cuộc tán công lên hơn 300.000 người dùng gmail tại Iran. Sau đó các công ty công nghệ như google, firefox, micrsoft đã rút chứng chỉ số của mình tại công ty này và đưa công ty vào dánh sách đen. Một vụ việc diễn ra năm 2017, khi tin tặc chiếm quyền kiểm soát máy chủ DNS của các ngân hàng Brazil và lừa CA cung cấp các chứng chỉ giả cho tin tặc.

Từ đây ta có thể thấy các thiết kế PKI hiện tại đang chứa nhiều rủi ro bảo mật cao khi chỉ cần chiếm được một CA thì có thể tấn công và mở khóa bất kì giao thức mã hóa nào. Vì vậy chúng ta cần một mô hình PKI mới phù họp với mức độ phát triển của công nghệ hiện tại.

# Chương 2: Decentralized Public Key Infrastructure



## Tổng quan.

Như chúng ta đã thấy mô hình hạ tầng khóa công khai cũ tồn tại nhiều điểm yếu và xem như đã không còn phù hợp với môi trường hiện tại. Vì vậy chúng ta cần một mô hình mới ít rủi ro và an toàn hơn cho công nghệ này. Trong số các hướng tiếp cận mới cho PKI hướng đi phát triển một một mô hình Pki phi tập trung được xem như là một hướng đi phù hợp nhất để xây dựng một mô hình Pki đảm bảo tính an toàn cho người dùng. Các mô hình này được gọi là Decentralized Public Key Infrastructure. Nhưng việc đạt được sự phi tập trung trên môi trường Internet được xem là rất khó cho đến khi blockchain xuất hiện tạo một tiền đề cho các ứng dụng đòi hỏi tính phi tập trung.

Trước đây khi chưa có sự xuất hiện của blockchain, phần mềm PGP Pretty Good Privacy được phát triển bởi Phil Zimmermann đã cố thiết lập một mô hình phân tán cho việc trao đổi thông tin và đã đạt được một số thành công nhưng vẫn gặp phải nhiều nghi hoặc trong các vân đề an toàn. Chỉ khi blockchain xuất hiện mô hình phân tán PKI mới tìm được một hướng đi mới đủ sức để thiết lập một DPKi an toàn.

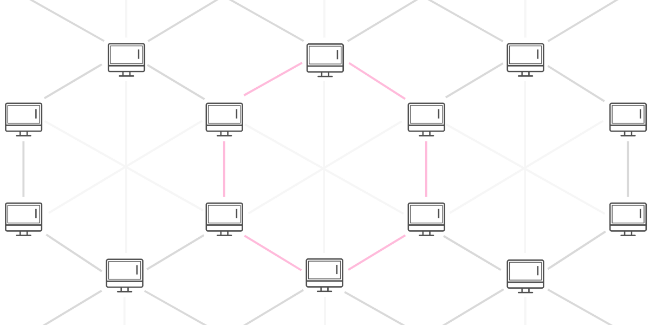
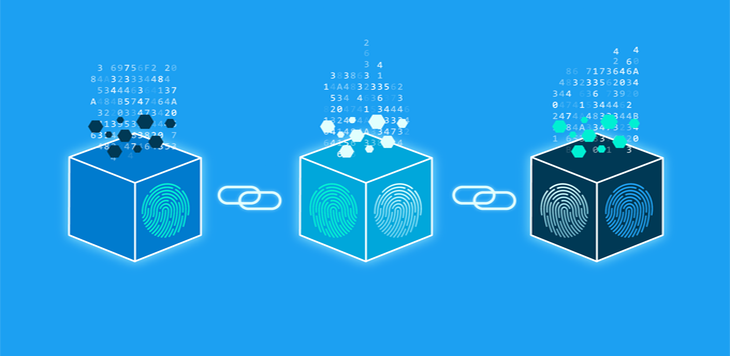
Phần này chúng ta sẽ đề cập đến hướng phát triển DPKI trên Ethereum một nền tảng blockchain, phân tích tìm hiểu ưu và nhược điểm của nó. Trước khi đi vào mô hình chính chúng ta sẽ đề cập qua về công nghệ Blockchain và vì sao nó có thể giúp xây dụng một ứng dụng phân tán.



## Công nghệ Blockchain

### Giới thiệu

Blockchain hay chuổi khối là một cơ sở dữ liệu phân tán phi tập trung trong đó dữ liệu được chia thành các khối được kết nối với nhau bở các giải thuật mã hóa cụ tạo thành một chuỗi dũ liệu được gọi là blockchain. Các chuỗi dữ liệu này được chia sẻ bởi một mạng lưới máy chủ ngang hàng P2P mà ở đó các máy đều có quyền hạn như nhau. Do đó hệ thống không chịu sự quản lý bởi một bên trung gian nào, và được gọi là hệ thống phi tập trung. Để đưa ra một quyết định hay cụ thể hơn đó là việc thêm một khối dữ liệu vào trong hệ thống, hệ thống cần phải có sự đồng thuận của hầu hết các node trong mạng, để giải quyết điều này blockchain sẻ dụng các giải thuật đông thuân.

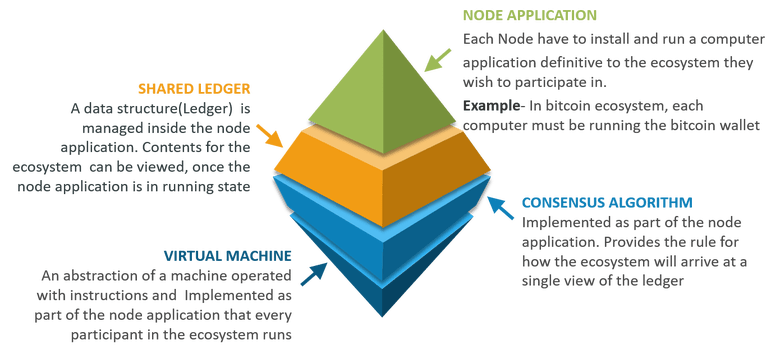
 

Với việc thiết kế như vậy hệ thống blockchain sẽ có những đặc điểm như sau.

* Decentralized hay phi tập trung. Như ta đã hệ thống blockchian được quản lý bởi một mạng máy tính ngang hàng do đó nó không chịu sự quản lý bởi một tổ chức nay đơn vị nào tạo tính minh bạch cho hệ thống.
* Tính trong suất:Do dữ liệu được chia sẽ bởi tất cả các node nên bất kỳ ai đều có quyền truy xuất đển toàn bộ dữ liệu được lưu vào hệ thống, nếu nó không được mã hóa.
* Tính bất biến. Blocchian được thiết kế sao cho việc thay đổi dữ liêu được xem như không thể một khi đã đưuọc đưa vào hệ thống, điều này đạt được do việc thiết kế dữ liệu dưới dạng chuỗi liên kết với nhau và các cơ chế đồng thuận. Nếu muốn thay đổi dữ liệu hacker phải tạo ra một chuỗi hợp lệ, nhưng khi thay đổi dữ liệu kể cả 1 ít giả trị hash của toàn chuỗi sẽ bị thay đổi. Nếu kẻ tấn công tạo ra một chuỗi hợp lệ thì cơ chế đồng thuận cũng không cho kẻ tấn công cập nhật chuỗi mới vào hệ thống.
* Khả năng chịu lỗi cao. Do việc thiết kế phân tán nên khả năng chịu lỗi của hệ thống rất cao. Nếu một máy bị sập hay bị tấn công thì cũng không thể làm sập tất cả hệ thống. Hệ thống blockchian chỉ chết khi không có một ai tham gia vào hệ thống.

Vì vậy blockchain phù hợp cho việc ghi lại những sự kiện, xử lý giao dịch, công chứng, danh tính và chứng minh nguồn gốc. Giúp loại bỏ các bên trung gian thứ 3, giảm thiểu khả năng gian lân dữ liệu. Đây cũng chính là lý do cho việc xây dựng hệ thống trên nền tảng này.

Một hệ thống blockchain sẽ gồm có nhiểu nhành phần nhưng cơ bản sẽ gồm 4 thành phần chính từ tầng thấp nhất đến tàng cao nhất như sau.



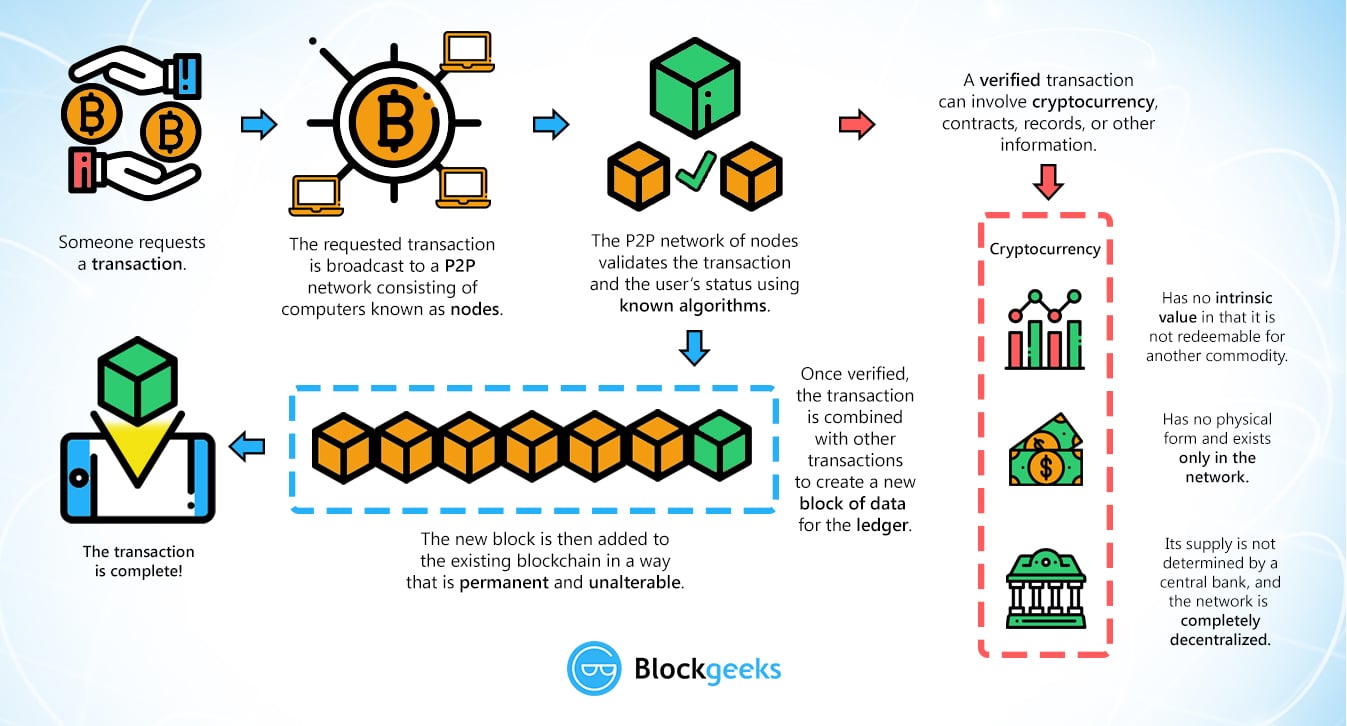
* A P2P Network: Một mạng lưới máy chủ ngang hàng p2p được gọi là các node, giao tiếp với nhau bằng một giao thức nhất định.
* A Shared Ledger: Một sổ cái được chia sẻ bởi các node, gồm các khối dữ liệu được liên kết với nhau bở các hàm hash.
* A Consensus Algorithm: Cơ chế đồng thuận, các cơ chế để các node trong mạng đạt được thỏa thuận về một trạng thái của sổ cái. Các cơ chế đồng thuân phổ biến hiện nay là PoW, PoS
* A virtual machine: Đây là lớp phía trên cùng của các node chịu trách nhiệm xử lý các logic của chúng tùy theo nền tảng mà cơ chế xử lý khác nhua. Ví dụ như ethereum là các máy ảo xử lý các hợp đồng thông minh.

### Quá trình xử lý giao dịch

Giao dịch là những thành phần cơ bản của hệ thống blockchain, người dùng giao tiếp với hệ thống bằng các gửi các giao dịch lên trên hệ thống và được xử lý bởi các node, quá trình xử lý các node được xác thực dựa vào chữ ký số được đính kèm theo giao dịch. Chữ ký số cung cấp cho chúng ta khả năng xác định xem giao dịch đó đến từ ai và đi đến đâu và có bị chỉnh sửa trong quá trình truyền tin hay không. Một giao dịch gồm có.

* Payload: chứa nội dung của giao dịch tùy theo hệ thống mà payload sẽ khác
* Hash value. Một giá trị hash của giao dịch.
* Address. Địa chỉ của tài khoản đến đây cũng chính là một biến thể của Pubkey
* Chữ ký số trên hàm hash của giao dịch được mã hóa bưởi Private key của người gửi.

Tùy vào mục đích mà các hệ thống blockchain sẽ có những cách hoạt động khác nhau nhưng cơ bản các giao dịch sẽ được xử lý như sau.

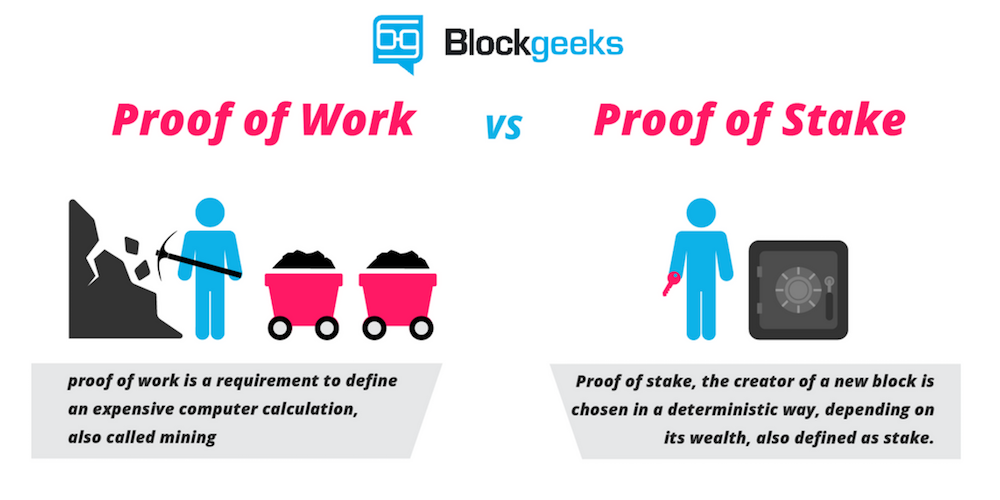


1. Giao dịch được tạo thông qua các phần mềm được gọi là ví và được gửi lên hệ thống thông qua một node.
2. Các node sẽ xác thực chữ ký số cuả các giao dịch để xem giao dịch có hợp lệ hay không, sau đó xử lý payload tùy theo từng hệ thống.
3. Sau khi xác minh tính hợp lệ, các giao dịch được các node tập hợp lai thành các block và tiến hành quá trình mineing.
4. Block mới sau khi tìm được sẽ được broadcasting lên hệ thống để được xác minh tính hợp lê. Khi block được đưa vào sổ cái thì quá trình kết thúc.

### Cơ chế đồng thuận.

Như ta đã biết blockchain là một cuốn sổ cái phân tán được chia sẻ bởi tất cả các node trong mạng. Trong hệ thống này không có một bên nào quản lý sự vân hành của hệ thống mà thay vào đó các node hoạt động riêng rẽ và giao tiếp với nhau để đưa ra các quyết định đồng thuận. Khi chúng ta muốn thêm một block vào khối, khối này phải đước sự đồng thuận của phần lớn các node trong mạng thì hệ thống mới đạt được tính nhất quán. Nếu chúng ta ủy thác việc này cho bên thứ 3 hay chỉ phụ thuộc vào một node thì sẽ làm mất tính phi tập trung của hệ thống. Hay nói cách khác, một hệ thống blockchain cần có khả năng chịu lỗi Byzantine.

Thuật ngữ khả năng chịu lỗi Byzantine xuất phát từ bài toán các vị tướng Byzantine được đề xuất năm 1982 của ba nhà khoa học máy tính là Leslie Lamport, Robert Shostak và Marshall Pease và họ cũng đề xuất một số giải thuật để giải quyết vấn đề này nhưng khả năng xử lý cũng như việc truyền tin trong môi trường không an toàn vẫn còn là một trở ngại. Với việc xuất hiện của chữ ký số đã giúp việc đảm bảo an toàn trong việc truyền tin trong môi trường internet. Điều này đã tạo điều kiện cho các giải thuật đồng thuận trên blockchain được phát triển. Vào năm 2008 với sự xuất hiện của bitcoin và chuỗi khối của nó, tác giả Nakamoto Satoshi cũng đã giới thiệu một một phương pháp để giải quyết bài toán các vị tướng quân, được gọi dưới cái tên Proof-of-Work (PoW), hay "bằng chứng công việc”. Ngoài PoW chúng ta cũng có một giải thuật đồng thuận khác như PoS hay DPoS.



Blockchian sử dụng các giải thuật đồng thuận để chỉ ra một node người mà sẽ đề xuất một khối mới để đưa vào chuỗi khối chính.

PoW là giải thuật đông thuân được dùng phổ biến nhất và được sử dụng trong các chuỗi lớn như Bitcoin hay Ethereum. Trong Proof of Word để được bầu thành người chọn khối tiếp theo, các node phải giải một bai toán. Phía dưới chỉ là một phiên bản của bài toán thể hiện giải thuật Proof of Work chúng ta có nhiều phiên bản bài toán khác nhau tùy vào khối sử dụng, nhưng chúng phải tuân thủ một số tiêu chí về độ phức tạp.

Bài toán được phát biểu như sau:

Với một dữ liệu X, tìm một số tự nhiên nhỏ nhất n sao cho giá trị hash của X + n có giá trị bé hơn một số Y cho trước.

Example - hash is a hypothetical hash function that has the outputs listed as below

Y = 10, X = 'test'hash(X) = hash('test') = 0x0f = 15 > 10

hash(X+1) = hash('test1') = 0xff = 255 > 10

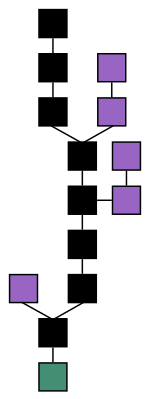
hash(X+2) = hash('test2') = 0x09 = 9 < 10 **OK, Solved.**

Với gỉả thiết rằng các hàm hash được sử dụng được xem là an toàn. Thì cách duy nhất để tìm kết quả này là thực hiện giải thuật vét cạn bruteforce, thử từ giá trị 0 đến khi nào tìm được giá trị n thỏa mãn. Vì vậy để giả được bài toán này các node tham gia phải có một sức mạnh tính toán đáng kế. Quá trình này thường được gọi là quá trình mining và tốn rất nhiều năng lượng của các máy đào và được trả công bằng các đông coin của hệ thống. Ngược lại với quá trình giải quyết quá trình xác thực lại dễ dàng được thực hiện do chỉ cần thử hàm hash với giá trị tìm được xem có thỏa mãn hay không.

Giữ liệu X trong bài toán chính là giá trị của hàm hash của header của khối trước đó và giá trị hash của khối hiện tại được thành lập bởi các giao dịch chưa được xác định lấy từ sổ cái của hệ thống. Do mỗi node sẽ lấy những giao dịch khác nhau để làm thành khối mới nên giá trị n (nonce) tìm được sẽ khác nhua và chi phí tính toán sẽ khác nhau cho mỗi node. Vì vậy không phải node nào có khả năng tính toán cao thì sẽ tìm được khối mới phù hợp trước.

Proof of Work cung cấp một giải thuật hiểu quả và đảm bả tính bảo mật cho hệ thống trong môi trường nhất định. Nhưng đổi lại chi phí tính toán cao dẫn đến tiêu thụ điện năng cao của nó. Đây chính là điểm yếu khiến các hệ thống blockchian bị chỉ trích rất nhiểu tại thời điểm hiện. Khi mà vấn đề bảo vệ môi trường đang được quan tâm hàng đầu. Điểm yếu thứ 2 của blockchian đó là giải thuật này cần một lượng lớn máy tham gia để đảm bảo tính an toàn. Nếu hệ thống blockchain xây dựng có quy mô nhỏ thì khả năng bị chiếm quyên điều khiển nếu hacker chiếm hơn 51% khả năng tính toán của hệ thống, đây được biết đến vưới kiểu tấn công 51%.

Sau khi chọn ra khối được đề xuất khối này sẽ được broadcasting ra hệ thống để các node khác xác thực tính hợp lệ của khối, nếu khối này hợp lệ thì các node sẽ cập nhật vào chuỗi khối của mình và tiến hành quá trình đồng thuận mới.



Trong quá trình cập nhật chuỗi mới chúng ta sẽ gặp trường hợp, trong một lúc sẽ có 2 khối hợp lệ được tìm ra gần như cùng lúc và tạo thành 2 chuỗi hợp lệ. Ví dụ 2 chuỗi này là A và B. Khi đó 2 khối này vẫn được chấp nhận vào hệ thống và tạo thành 2 chuỗi khác nhau, một phần hệ thống sẽ chập nhận chuỗi A và phần còn lại chập nhận chuỗi B. Lúc này những node chấp nhận chuỗi A vẫn tiếp tục đào trên chuỗi A và B cũng tương tư.Tại đây nếu một khôi mới được tìm thấy và được cập nhật vào chuỗi A thì chuỗi A sẽ dài hơn chuỗi B và được xem là chuỗi chính do có độ dài dài hơn và sẽ được cập nhật trên toàn hệ thống. Nhưng nếu trường hợp ban đầu lại tiếp tục xảy ra trên 2 chuỗi A và B thì quá trình sẽ diễn ra tiếp diễn đến khi nào có một chuỗi dài hơn và được xem là chuỗi chính. Ngoài chuỗi chính ra các chuỗi còn lại được xem là không hợp lệ và không được sử dụng.

Đây cũng là lí do vì sao một khối không thể được sử dụng cho đến khi có ít nhất một số lượng khối nhất định đã được thêm vào chuỗi đó.

## Nền tảng Ethereum

### Ethereum

Ethereum là một platform blockchain mã nguồn mở được xây dựng để thực hiện các chương trình phân tán được gọi là smartcontract. Ethereum cung cấp một loại tiền mã hóa được gọi là ether được sử dụng để giao dịch cũng như trả công cho các máy đào trong hệ thống. Không giống với Bitcoin được xây dựng là một loại tiền tệ và lưu các giao dịch của nó. Ethereum có thể làm được nhiều hơn thế, được xây dựng là một máy tính phân tán, nơi mà lập trình viên có thể lập trình trên đó.

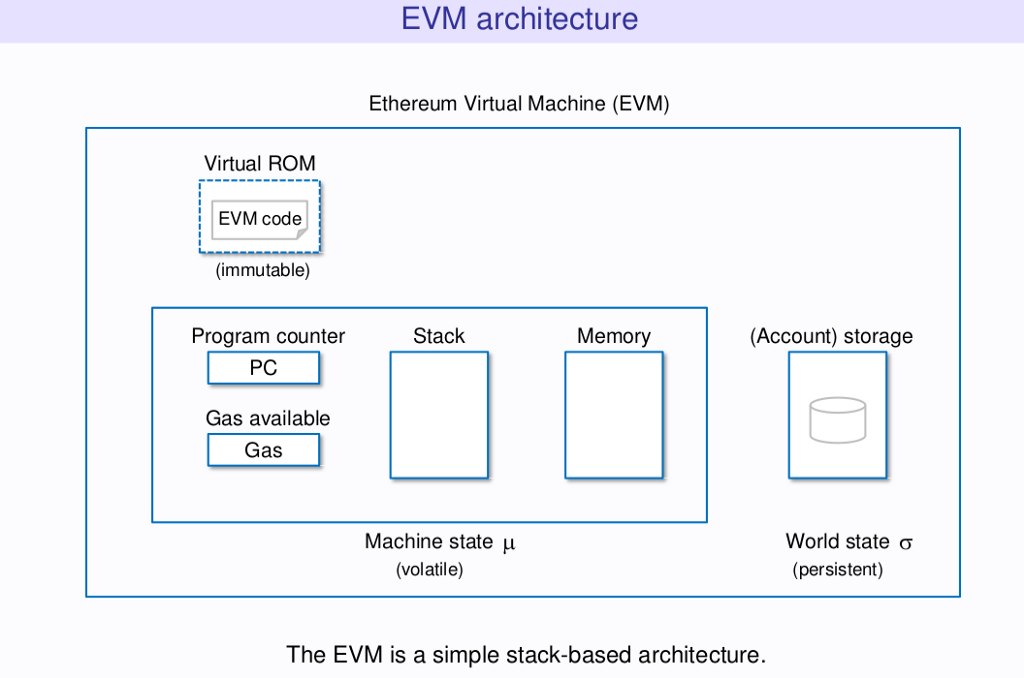
Các chương trình được tạo ra trên nền tảng này được gọi là Dapps (decentralized applications). Chúng là những chương trình tin cậy và có thể dự đoán được, có nghĩa là khi một hợp đồng được upload lên hệ thống thì mã nguồn của nó không thể thay đổi và chúng luôn chạy đúng như đã được lập trình. Do là một hệ thống phân tán phi tập trung nên không có một tổ chức nào có thể kiểm soát được chúng. Ngày nay có hàng ngàn lập trình viên và hàng ngàn dapps đã được upload lên hệ thống trên nhiều lĩnh vực khác nhau từ tài chính, bảo hiểm hay hệ thống nhắn tin trong đồ án này.

Giống với các hệ thống blockchian khác ethereum gồm có các thành phần như sau.

* P2P Network, Ethereum chạy trên một mạng lưới được gọi là Ethereum main network, điều này để phân biệt với một số mạng ethereum dùng để test trong việc lập trình. Mạng lưới này chạy trên cổng TCP 30303 và giao tiếp với nhau bởi giao thức ĐEVp2p.
* A Consensus Algorithm, Hiện tại ethereum sử dụng cơ chế PoW và đang có kế hoạch đổi thành PoS trong thời gian gần đây.
* A Shared Ledger: Sổ cái của ethereum lưu giữ nhiều thông tin bao gồm trạng thái của các tài khỏan chứa số dư tài khoản, đối với các tài khoản smartcontract còn chứa mã code của smart contract đó ngoài ra còn có một số thông tin khac. Cac giao dịch và các biên lai khi thực thi chương trình smartcontract.
* A Vitural Machine: Ethereum sử dụng một máy ảo Ethereum Virtual Machine để xử lý các giao dịch

### A Vitural Machine và ngôn ngữ Solidity

Ethereum sử dụng một máy ảo Ethereum Virtual Machine để xử lý các giao dịch. Ethereum Virtual Machine hay EVM là một môi trường chạy các đoạn mã của hợp đồng thông minh. Được xây dựng phía trên chuỗi khối, với mỗi node trong hệ thống sẽ có một máy ảo độc lập do đó người ta thường nói Etherrum là một máy tính phi tập trung toàn cầu gồm nhiều máy trên toàn thế giới với mỗi máy đều có bộ nhớ riêng và thực hiện chung một công việc. EVM là một máy trạng thái turing gần như hoàn hảo do quá trình thực hiện bị giới hạn bởi số gas có sẵn được cấp cho mỗi giao dịch, khi máy ảo xử lý hết gas thì quá trình tính toán cũng tạm dừng do vậy nó không hoàn toàn.



Máy ảo EVM là một máy ảo dựa trên kiến trúc stack, có nghĩa là quá trình thực thi sẽ dựa vào cơ chế stack để thực hiện các opcode. Ngoài stack EVM còn có các phân vùng bộ nhớ khác. Các phân vùng này đều được chia thành các khối thành phần có giá trị là 256 bit, giá trị này để thuận tiện cho việc thực thi các câu lệnh mã hóa ECDSA.

* ROM: Đây là một phân vùng bộ nhớ chứa những opcode sẽ được hiện trong máy ảo. Khi có một giao dịch gọi hàm từ smartcontract những đoạn mã trong hàm đó sẽ được load vào ROM
* Memory: Đây là phân vùng nhớ khi thực hiện các hàm, là phân vùng nhớ tạm thời có nghĩa là sẽ được reset lại với mỗi lượt gọi hàm khác nhau. Đây là bộ nhớ truy xuất trực tiếp.
* Storage: Đây là bộ nhớ chính là bản sao trạng thái của contract hiện tại.

Bộ đếm chương trinh PC tương tự với các máy ảo khác, ngoài ra còn có Gas bộ đếm gas để đêm lượng gas còn lại trong mỗi giao dịch.

Giống như các nền tảng platform khác chúng ta khó có thể viết trục tiếp smartcontrct từ mã opcode của máy ảo thay vào đó ta cần một ngôn ngữ bậc cao hơn để xử lý. Solidity hiện nay đang là ngôn ngữ phổ biến nhất để viết smart contract, bên cạch đó còn có Vyper và Bamboo và trước đó là Serpent và Mutan. Solidity là một ngôn ngữ lập trình bậc cao hướng contract. Được xây dựng dựa trên chuẩn ECMAScript nên có nhiểu nét tương đồng với Javascript. Là một ngôn ngữ biên dịch, dịch trực tiếp từ mã solidity sang mã opcode. Khác với các ngôn ngữ khác Solidity hỗ trợ đầy đủ các tính chất của một ngôn ngữ hướng đối tượng như kế thừa, tính đa hình hay đóng gói. Cung cấp các cấu trúc dữ liệu phức tạp như Structs hay Mapping. Có thể nói hiện nay Solidity là ngôn ngữ hoàn thiện nhất trên nền tảng này

Do công nghệ cũng như ngôn ngữ này mới được phát triển trong những năm gần đây nên nó thay đổi khá nhiều và thưởng xuyên được cập nhật với nhiều nhiểu tính năng được thêm vào mỗi phiên bản. Trong bản báo cáo này cũng như trong dự án chúng ta sẽ sử dụng phiên bản Solidity v0.6.8 để giải thích cũng như thực hiện đồ án.

**pragma solidity** ^0.6.8;

**contract** SimpleStorage {

uint storedData;

address **public** seller;

**function** set(uint x) **public** {

storedData = x; *// State variable*

}

**function** get() **public** **view** **returns** (uint) { *// Function*

**return** storedData;

}

**modifier** onlySeller() { *// Modifier*

require(

msg.sender == seller,

);

**\_**;

}

**function** abort() **public** **view** onlySeller { *// Modifier usage*

}

**event** HighestBidIncreased(address bidder, uint amount); // Event

**function** bid() public payable {

emit HighestBidIncreased(msg.sender, msg.value); // Triggering event

}

**struct** Voter { // Struct

uint weight;

bool voted;

address delegate;

uint vote;

}

**enum** State { Created, Locked, Inactive } // Enum

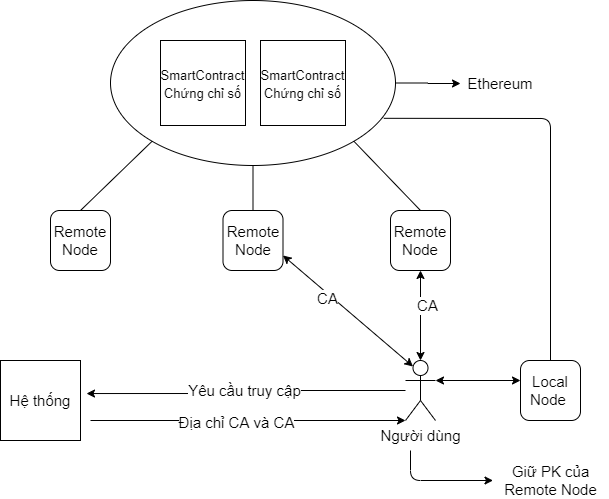
}

Do Solidity là ngôn ngữ phổ biến nhất hiện nay, hỗ trợ nhiều tính năng oop và có các cú pháp dựa trên chuẩn ECMAScript quen thuộc nên được chọn làm ngôn ngữ để thực hiện đồ án này

## Mô hình triển khai chứng chỉ số trên Ethereum.

Hiện nay có nhiều cách thức để đạt được sự phân tán trong PKI, trong bản đồ án này chúng ta sẽ để cập đến mô hình sử dụng Blockchian cụ thể hơn là nền tảng Ethereum để đạt được sự phân tán trong việc triển khai các chứng chỉ số.

Vậy với DPKI được triển khai trên Ethereum là như thế nào. Đơn giản đó là việc các chứng chỉ số sẽ được biểu diển thành các Contract và được deploy lên nền tảng của Ethereum. Mọi việc xử lý đối với PKI sẽ được thực hiện bởi các contract chạy trên nền hạ tầng mạng lưới P2P của Ethereum, chính mô hình Ethereum này sẽ đảm bảo các tiêu chí an toàn cho PKI.



Hình vẽ trên đã mô tả khá rõ mô hình hoạt động của DPKi. Người dùng các bên sẽ thông qua cơ sở hạ tầng của Ethereum để tạo, truy cập hay sử dụng các dịch vụ khác trên đó. Các bên có thể tự mình chạy các node hoặc thông qua các node trung gian để thực hiện các giao dịch việc này người dùng có thể tùy chọn các node mình muỗn không phụ thuộc hoàn toàn vào bên nào.

Khi một người dùng yêu cầu khóa công khai của một bên, bên đó sẽ gửi chứng chỉ số của mình kèm theo địa chỉ của PKCC tương ứng trên ethereum.

Người dùng sẽ truy cập hệ thống Ethereum thông qua các node tin tưởng với địa chỉ Ethereum tương ứng để lấy chứng chỉ số trên Ethereum. Và so sánh 2 chứng chỉ này với nhau.

Câu hỏi đặt ra đó là làm sao đảm bảo đường truyền giữa người dùng và các node Ethereum là an toàn và không bị sửa đối. Câu hỏi này tương tự với việc làm sao để xác định khóa công khai của CA là đúng đối với các mô hình truyền thống. Và đối với mô hình Hierarchical PKI thì sẽ được xử lý thông qua một chuỗi các xác thực CA và cuối cùng đó là sử dụng chính key của Root CA được tích hợp trong máy để sử dụng. Tương tự mô mình đó DPKi cần phải tích hợp các khóa công khai của các node tin tưởng trong máy để trao đổi thông tin. Điểm khác biệt ở đây đó là chúng ta có thể chọn bất kì các node nào mình muốn để xác thực, chúng ta có thể có nhiều khóa của nhiều node để tín hành quá trình xác thực chứ không bị cố định một RootCA như mô hình trên. Việc xác thực có thể kết hợp nhiều node để đảm bảo thêm an toàn. Và việc dữ liệu trên Ethereum là không thể sửa đổi nên việc xác thực các chứng chỉ số là an toàn.

Phần tiếp theo chúng ta sẽ đi sâu đến nội dung được triển khai trên Ethereum.

Đầu tiên đó là nội dung các thông tin được triển khai trên DPKI. Với chuẩn x509 chúng ta có khá nhiều trường thông tin khác nhau, nhưng tựu chung lại là có các nội dung chính như sau.

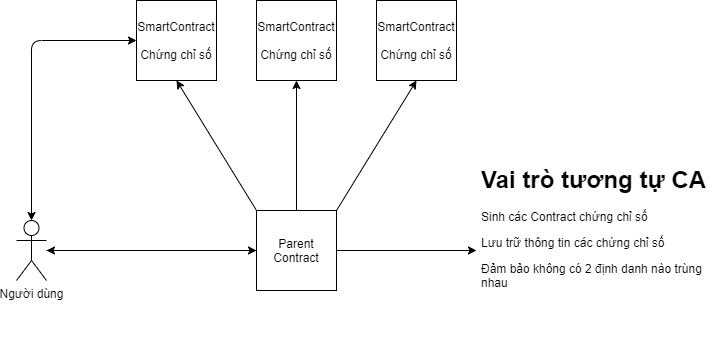
* Thông tin định danh đơn vị sở hữu chúng chỉ só.
* Thông tin về khóa và các cơ chế mã hóa.
* Thông tin về đơn vị cung cấp chứng chỉ số.
* Chữ kí số của CA lên chứng chỉ.

Còn đối với DPKI chúng ta chỉ cần 2 thông tin ban đầu đó là định danh chủ sở hữu và thông tin về khóa công khia và các cơ chế mã hóa của nó. Điều này có thể hiểu được dễ dàng do ở DPKI không có một CA nào nên ta sẽ không có các thông tin về CA cũng như chữ kí số của CA trong đó. Công việc của các thực thể này sẽ được tự động hóa và thay thể bởi các Smart Contract.

Cụ thể nội dung của một Public key Certificate Contract (PKCC) như sau.

* Thông tin đinh danh: Tùy vào môi trường mà DPKi được sử dụng mà định danh sẽ khác nhau.
  + Ví dụ với môi trường truyền tin trong internet ta sẽ lấy cặp Domain:IP là thông tin định danh. Với mỗi định danh chỉ có một khóa công khai và ngược lại.
* Khóa công khai và thuật toán mã hóa
* Một số thông tin khác.
  + Thời gian hiệu lực của khóa
  + Ethereum Address của người tạo.

Các công việc phát hành, phân phối, lưu trữ và thu hồi được giải thích như sau.



* Phát hành:

Việc phát hành các chứng chỉ số sẽ được tự động hóa bởi một Smart Contract cha trên Ethereum. Contract này sẽ được triển khai đầu tiên trên hệ thống, nó sẽ cung cấp các API cho người dùng để thông qua đó tạo ra các PKCC.

Người dùng sẽ có toàn quyền sinh khóa và hệ thống không có quyền này, điều này để đảm bảo không ai ngoài chủ sở hữu có thể biết được khóa bí mật. Hệ thống chỉ tiếp nhận thông tin và tạo ra PKCC tương ứng và PSC chỉ đảm bảo không có 2 định danh cũng như khóa công khai trùng nhau trong hệ thống. Như vậy hệ thống sẽ không biết khóa này đúng hay sai. Việc cho người dùng toàn quyền sinh khóa nhằm tăng tính an toàn cho khóa và cũng không có điểm yếu nào đến hệ thống nếu người dùng sinh khóa sai thì chỉ ảnh hưởng đến các bên dùng khóa thôi. Vì vậy người dùng phải có trách nhiệm sinh khóa đúng và an toàn. Và một điều nữa là nền tảng Ethereum chưa có cách để xác định một khóa có khả năng đúng hay sai.

Việc đăng ký các định danh và khóa công khai trên đây hiện tại sẽ được thả trôi, có nghĩa là ai cũng có quyền đăng ký một định danh và khóa mà không phải xác thực danh tính có đúng không miễn là nó chưa có trên hệ thống là được. Đây là một điểm yếu đối với các thương hiệu đã có trên thị trường khi mà kẻ xấu có thể chiếm quyền định danh trước khi mà bên hợp pháp chưa kịp đăng ký.

* Phân phối:

Việc phân phối các chứng chỉ số sẽ được thực hiện thông qua hạ tầng của Ethereum, Người dùng có thể chọn bất cứ node nào trên hệ thống để truy cập vào DPKI hoặc chính người dùng sẽ chạy một node Ethereum để có thể truy cập vào Ethereum. Điều này tăng thêm quyền cho người dùng và giảm thiểu sự phụ thuộc vào bên thứ 3.

* Lưu trữ:

Các PKCC sẽ được lưu vào sổ cái của Ethereum, với cơ chế của Ethereum sẽ đảm bảo không ai có quyền chỉnh sử PKCC ngoài trừ chủ sở hữu của nó.

* Tắt kích hoạt

Thay vì khái niệm thu hồi với PKI truyền thống DPKI chỉ có khái niệm tắt kích hoạt. Gọi là tắt kích hoạt thay vì thu hồi là do các chứng chỉ này vẫn sẽ được truy cập trên hệ thống việc có sử dụng hay không đó là quyền của người dùng. Chức năng này sử dụng trực tếp trên Contract CA không qua Parent Contract. Có 3 điều kiện để một PKI tắt kích hoạt đó là.

* PKCC sẽ hết hạn khi quá thời gian được xác định trên chính PKCC đó
* Chủ sở hữu có quyền chủ động tắt kích hoạt đối với PKCC của mình.
* Khi PKCC đang trong quá trình sang nhượng.
* Sang nhượng

Như đã đề cập ở trên PKCC có thể được sang nhượng cho bên khác. Khi trong quá trình sang nhượng thì khóa công khai sẽ được reset và chuyển qua trạng thái chờ và chủ sở hữu sẽ được đặt mới. PKCC này sẽ được tái kích hoạt khi khóa mới được đặt lại. Việc này sẽ được đảm bảo trên PKCC chỉ có chủ sở hữu mới được dùng API này. Chức năng này sử dụng trực tếp trên Contract CA không qua Parent Contract.