Sổ cái có khả năng chống lại các tính toán lượng tử (Quantum Resistant Ledger - QRL)

peterwaterland@gmail.com

Tháng 11 năm 2016

Tóm tắt

Để có thể tồn tại lâu dài, các đồng tiền kỹ thuật số phải được bảo mật chống lại các tiến bộ của máy tính. Bài viết này sẽ trình bày về thiết kế và sự phát hành sổ cái của một loại tiền mã hóa, sử dụng các chữ ký kỹ thuật số dựa trên mã băm, có khả năng chống lại các cuộc tấn công truyền thống cũng như các cuộc tấn công của tính toán lượng tử.

1. Giới thiệu

Khái niệm về một sổ cái có giá trị với các giao dịch trực tiếp, ngang hàng trên mạng Internet, được ghi dưới dạng chuỗi khối (blockchain) và được bảo mật bằng phương pháp bằng chứng về công việc (proof of work) được đưa ra lần đầu tiên vào năm 2008 [11]. Cho đến nay Bitcoin là loại tiền mã hóa được lưu hành rộng rãi nhất. Sau Bitcoin, có hàng trăm các loại sổ cái khác đã được sáng tạo ra; phần lớn các loại sổ cái này dựa trên hệ bảo mật địa chỉ công khai dựa trên đường cong elliptic (ECDSA) để tạo ra các chữ ký kỹ thuật số giúp xác minh các giao dịch một cách an toàn. Về mặt lý thuyết, các cơ chế bảo mật bằng chữ ký được sử dụng phổ biến nhất ngày nay như ECDSA, DSA và RSA không có khả năng chống lại các cuộc tấn công của tính toán lượng tử. Bởi vậy ở đây chúng ta sẽ khám phá về thiết kế và cách xây dựng của một sổ cái hoạt động theo giao thức blockchain chống lượng tử, để chống lại một cuộc tấn công bất ngờ phi tuyến của tính toán lượng tử.

2. Sự bảo mật của các giao dịch bitcoin

Hiện nay, cách duy nhất để sử dụng (các kết quả từ các giao dịch trước chưa được sử dụng) từ một địa chỉ bitcoin là tạo ra một giao dịch có chứa chữ ký đường cong elliptic có hiệu lực (secp256k1) từ khóa riêng tư $(x \in N | x < 2^{256})$ của chính địa chỉ bitcoin đó. Nếu khóa riêng tư được tạo ngẫu nhiên đó được bảo mật hay bị thất lạc, thì số tiền trong địa chỉ đó sẽ không bao giờ có thể gửi đi được.

Khả năng xảy ra sự xung đột khóa riêng tư bitcoin chỉ là 1 trong 2^{256}. Có thể ước lượng được xác suất xung đột một địa chỉ bitcoin bất kỳ bằng cách sử dụng Bài toán Ngày sinh. Số lượng các địa chỉ bitcoin cần tạo ra để xảy ra 0,1% xung đột là $5{,}4 \times 10^{23}$ địa chỉ [14].

Tuy nhiên, khi một giao dịch được ký kết, khóa công khai ECDSA của địa chỉ gửi đã bị lộ và được lưu trữ trên blockchain. Cách bảo mật tốt nhất là không sử dụng lại các địa chỉ đó nữa, tuy

nhiên tính đến tháng 11 năm 2016, có tới 49,58% tổng số dư trên sổ cái của bitcoin lại được chứa trong các địa chỉ đã bị lộ khóa công khai [1].

3. Các phương hướng tấn công của tính toán lượng tử

Các hệ thống bảo mật RSA, DSA và ECDSA vẫn an toàn nhờ vào sự khó khăn về mặt tính toán của bài toán tìm thừa số của một số nguyên lớn, bài toán giải thuật rời rạc và bài toán giải thuật rời rạc đường cong elliptic. Thuật toán lượng tử của Shor (1994) đã giải được bài toán phân tích các số nguyên lớn và các thuật toán rời rạc trong thời gian đa thức. Bởi vậy, về mặt lý thuyết thì một máy tính lượng tử có thể khôi phục khóa riêng tư nếu được cung cấp khóa công khai ECDSA. Do đó ECDSA được coi là dễ tổn thương trước các tấn công lượng tử hơn RSA vì ECDSA sử dụng các khóa có kích thước nhỏ hơn, vì một máy tính lượng tử 1300 và 1600 qubit (2^{11}) có thể giải được ECDSA 228 bit.

Theo các thông tin công khai, máy tính lượng tử vẫn chưa vượt qua được 2^5 qubit hoặc tìm thừa số của các số nhỏ (15 hoặc 21). Tuy nhiên, tháng 8 năm 2015, NSA đã đưa ra một sự phản đối với phương thức mã hóa đường cong elliptic dựa trên các lo ngại về tính toán lượng tử. Hiện nay không ai biết các tính toán lượng tử đã phát triển đến đâu hoặc các bước đột phá trong lĩnh vực này có được công khai để các giao thức mã hóa đang được sử dụng nhiều trên mạng Internet sẽ được chỉnh sửa để bảo mật trước các tính toán lượng tử hay không. Nếu không được cải thiện, bitcoin sẽ trở thành mục tiêu tấn công của một địch thủ có trong tay một chiếc máy tính lượng tử.

Trong trường hợp các tiến bộ quan trọng về tính toán lượng tử được công bố, các nhà lập trình nút giao có thể tiến hành các phương thức chữ ký mã hóa chống lại lượng tử vào bitcoin và khuyến khích tất cả những người dùng bitcoin chuyển số dư của họ từ các địa chỉ dựa trên ECDSA sang các địa chỉ mới, an toàn trước lượng tử. Để giảm thiểu số lượng của các địa chỉ bị ảnh hưởng, cần phải ngăn chặn việc tái sử dụng khóa công khai ở cấp độ toàn giao thức. Công cuộc nâng cấp này có thể bao gồm cả việc di chuyển 1 triệu xu thuộc sở hữu của Satoshi Nakamoto và do đó sẽ làm giá của đồng tiền dao động dữ dội.

Trong tình huống xấu hơn, các tính toán lượng tử phi tuyến được phát triển âm thầm sẽ làm tiền đề cho một cuộc tấn công tính toán lượng tử vào các địa chỉ bitcoin đã bị lộ khóa công khai. Sự tấn công hay lấy cắp này sẽ có ảnh hưởng vô cùng xấu tới tỷ giá bitcoin, do áp lực bán tăng mạnh và sự mất niềm tin hoàn toàn vào hệ thống xảy ra khi các thông tin về các cuộc tấn công lan rộng. Bitcoin sẽ không còn được tin tưởng như một phương thức dự trữ giá trị nữa ("một loại vàng kỹ thuật số") và điều này sẽ có những hậu quả tai hại cho toàn thế giới. Trong trường hợp này, chúng tôi cho rằng cần phải thử nghiệm các chữ ký mã hóa chống lượng tử trong một sổ cái của đồng tiền mã hóa và có thể phải tạo ra một phương án dự trữ giá trị dự phòng để đề phòng trường hợp xấu nhất xảy ra.

4. Chữ ký chống lượng tử

Có một số hệ thống mã hóa quan trọng được xem là có khả năng chống lại lượng tử: hệ mã hóa dựa trên mã băm, mã hóa dựa trên code, mã hóa dựa trên lưới, mật mã của các phương trình bậc hai nhiều biến và mật mã dựa trên khóa bí mật. Tất cả các phương thức này đều được xem là có khả năng chống lại các cuộc tấn công thông thường cũng như tấn công lượng tử, nếu được cung cấp các kích thước khóa đủ dài.

Các phương thức chữ ký kỹ thuật số dựa trên mã băm, theo kiểu bảo mật chuyển tiếp, chỉ có các yêu cầu bảo mật ở mức tối thiểu và chỉ dựa trên tính năng chống xung đột của hàm băm mật mã học. Các chữ ký kỹ thuật số dựa trên mã băm đã được nghiên cứu kỹ và là một lựa chọn tiềm năng cho các chữ ký thời kỳ hậu lượng tử trong tương lai. Do vậy, chúng đã được chọn làm chữ ký QRL trong thời kỳ hậu lượng tử.

5 Các chữ ký kỹ thuật số dựa trên mã băm

Các chữ ký dựa trên mã băm có khả năng chống lượng tử hoạt động nhờ vào tính bảo mật của một hàm băm mật mã học một chiều. Tính năng này cho phép hàm băm đó tiếp nhận một thông điệp, *m* và xuất ra một mã băm tóm tắt, h có chiều dài cố định, *n*, chẳng hạn như SAH-256; SHA-512. Với hàm băm mật mã hóa, xét về mặt tính toán, hầu như là không thể đoán được *m* khi biết *h* (khả năng chống lại việc tìm *m* khi biết *h*), hay đoán được *h* từ h_2, với $h_2 = hash(h)$ (khả năng chống lại việc tìm *h* khi biết h_2), đồng thời cũng rất khó để có thể tìm được hai thông điệp khác nhau $(m1 \neq m2)$ mà tạo ra cùng một *h* (khả năng chống xung đột).

Có thể sử dụng thuật toán lượng tử của Grover để cố gắng tìm ra một sự xung đột mã băm hay để thực hiện tấn công để tìm *m* từ *h*, điều đó đòi hỏi $O(2^{n/2})$ thao tác. Do đó để duy trì tính bảo mật 128 bit, chiều dài của mã băm tóm tắt, phải chọn n có ít nhất 256 bit – giả thiết có một hàm băm mật mã hóa hoàn hảo.

Các chữ ký kỹ thuật số dựa trên mã băm đòi hỏi phải có một khóa công khai, *pk*, để xác thực và một khóa riêng tư, *sk* để ký xác nhận thông điệp. Chúng tôi sẽ thảo luận về khả năng sử dụng các chữ ký một lần dựa trên mã băm (OTS) trong một phần của một sổ cái theo phương thức blockchain.

5.1 Chữ ký một lần của Lamport-Diffie

Năm 1979 nhà bác học Lamport đề xuất một kiểu chữ ký một lần dựa trên hàm băm cho một thông điệp có độ dài *m* bit (thường là kết quả của một hàm băm chống xung đột). Đầu tiên, một thuật toán tạo cặp khóa được sử dụng để tạo ra *m* cặp khóa bí mật ngẫu nhiên, $sk_j^m \in \{0,1\}^n$ trong đó $j \in \{0,1\}$, tức là khóa riêng tư có dạng: $sk = ((sk_0^1, sk_1^1), .., .., (sk_0^m, sk_1^m))$. Chúng ta có *f* là một hàm băm một chiều $\{0,1\}^n \rightarrow \{0,1\}^n$ với *m* cặp khóa công khai được tạo ra $pk_j^m = f(sk_j^m)$, tức là khóa công khai có dạng $pk = ((pk_0^1, pk_1^1), .., .., (pk_0^m, pk_1^m))$. Để ký vào thông điệp, mỗi bit trong mã băm của thông điệp được gắn với một sk_j (tức là, nếu

$bit = 0,\ sk_j = sk_0,\ bit = 1,\ sk_j = sk_1$), và từ đó ta có chữ ký $s = (sk_j^1, .., sk_j^m)$, tức là đã sử dụng một nửa số khóa bí mật. Để kiểm định chữ ký, kiểm tra các bit trong mã băm của thông điệp $(j \in \{0,1\}$ để xác nhận $(pk_j = f(sk_j))^m$.

Giả sử sau khi đạt được bảo mật của giải thuật 128 bit Grover, trong đó thông điệp được tạo bởi hàm băm SHA256, với m=256 và n=256, do đó ta có pk = sk = 16kb, và mỗi OTS được tạo một chữ ký có kích thước 8kb. Chữ ký Lamport chỉ được sử dụng một lần ; nó được tạo ra rất nhanh chóng, tuy nhiên kích thước khóa, chữ ký và giao dịch lớn, do vậy không thích hợp để sử dụng trong một cuốn sổ cái hoạt động theo hình thức chuỗi khối công khai.

5.2 Chữ ký một lần của Winternitz

Với một cốt thông điệp, M, có chiều dài M bit, với các khóa công khai và khóa bí mật có chiều dài n bit, và hàm một chiều $f : \{0,1\}^n \to \{0,1\}^n$ và tham số Winternitz $w \in N | w > 1$, ý tưởng chung của chữ ký một lần của Winternitz là áp dụng một hàm băm lặp đối với một danh sách các khóa bí mật ngẫu nhiên, $sk \in \{0,1\}^n\ sk = (sk_1, .., sk_{m/w})$ tạo ra các chuỗi mã băm có chiều dài $w - 1$, kết thúc bằng các khóa công khai $(pk \in N | \{0,1\}^n)$ $pk_x = f^{2^{w-1}}(sk_x),\ pk = (pk_1, .., pk_{m/w})$.

Để tạo chữ ký, nếu trong chữ ký của Lamport, từng bit trong cốt của thông điệp được gắn với một số trong các khóa bí mật ngẫy nhiên, thì trong chữ ký của Winternitz, thì w bit của thông điệp được phân tích để xuất được một số $i \in N, i < 2^w - 1$ và tạo ra chữ ký từ số đó, $s_x = f^i(sk_x)\ s = (s_1, ..s_{m/w})$. Số w càng lớn thì kích thước các khóa và chữ ký lại càng nhỏ, tuy nhiên, thời gian và chi phí tính toán sẽ lớn hơn. [10].

Để xác thực chữ ký, đơn giản chỉ cần tạo $pk_x = f^{2^{w-1}-i}(s_x)$ từ M, s và đối chiếu sự trùng hợp giữa các khóa công khai.

Sử dụng SHA-2 (SHA-256) như là một hàm số băm mã hóa một chiều, $f:\ m = 256$ và n=256, với w=8 chúng ta có pk=sk=s với kích thước $\frac{(m/w)n}{8}$ byte=1kb.

Để tạo được pk chúng ta phải có f^i bước lặp băm, trong đó $i = \frac{m}{w}2^{w-1} = 8160$ với mỗi cặp khóa OTS được tạo ra. Với w=16, các khóa và chữ ký chỉ còn bằng ½ kích thước, nhưng không thể có $i = 1048560$.

5.3 Một biến thể của khóa OTS của Winternitz (W-OTS+)

Buchmann đề xuất một biến thể của khóa OTS của Winternitz bằng cách áp dụng hàm lặp một chiều với một số ngẫu nhiên, x, một cách lặp đi lặp lại, nhưng lần này được tham số hóa bởi một

khóa, k, được tạo ra từ phép lặp $f_k(x)$ trước đó. Kẻ tấn công không thể làm giả được chữ ký này khi sử dụng một hàm giả ngẫu nhiên (PRF) và có thể tính được một bằng chứng bảo mật với các thông số đã cho [3]. Do vậy không cần phải có một hàm băm chống xung đột do chúng ta đã sử dụng hàm thay vì chỉ lặp lại. Huelsing đưa ra một biến thể W-OTS+ khác, cho phép tạo các chữ ký nhỏ hơn với độ bảo mật tương đương bằng cách bổ sung bitmask XOR trong hàm chuỗi lặp [6]. Một khác biệt nữa giữa W-OTS (biến thể năm 2011)/W-OTS+ và W-OTS- nằm ở chỗ thông điệp được phân tích theo $\log_2(w)$ bit mỗi lần thay vì w bit, làm giảm số lần lặp của hàm băm nhưng làm tăng kích thước của khóa và chữ ký.

Bây giờ chúng tôi sẽ mô tả ngắn gọn W-OTS+. Với thông số bảo mật, $n \in N$, tương ứng với chiều dài của thông điệp (m), các khóa và chữ ký theo bit, được xác định bởi hàm băm mật mã hóa được chọn và thông số Winternitz, $w \in N | \ w > 1$, (thường là{4;16}), chúng ta tính được l. l là số lượng các yếu tố trong một khóa hoặc chữ ký WOTS+, trong đó $l=l_1 + l_2$:

$$l_1 = \lceil \frac{m}{log_2(w)} \rceil, \quad l_2 = \lfloor \frac{log_2(l_1(w-1))}{log_2(w)} \rfloor + 1$$

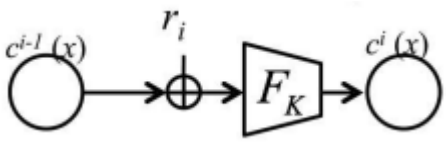
Một hàm băm nhập số vào được sử dụng: $f_k : \{0,1\}^n \rightarrow \{0,1\}^n \mid k \in \{0,1\}^n$. Dưới dạng mã giả:

$$f_k(M) = Hash(Pad(K)||Pad(M))$$

Trong đó $Pad(x) = (x||10^{b|x|1})$ với $|x| < b$.

Hàm chuỗi, $c_k^i(x,r)$: với $x \in \{0,1\}^n$, số lần lặp i, khóa $k \in K$ và các yếu tố ngẫu nhiên $r = (r_1, .., r_j) \in \{0,1\}^{n*j}$, với $j \geq i$, được xác định như sau:

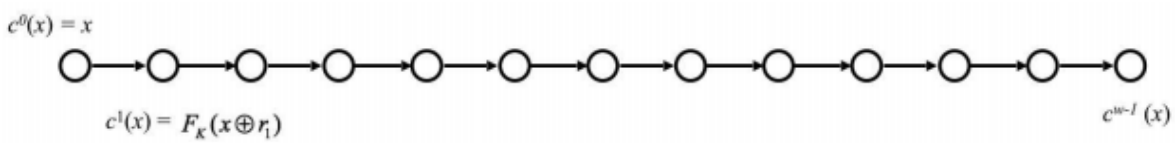
Hình 1. Hàm chuỗi W-OTS+



trong đó

$$c^i(x,r) = \begin{cases} x & \text{if } i = 0; \\ f_k(c_k^{i-1}(x,r) \oplus r_i) & \text{if } i > 0; \end{cases}$$

Hình 2. Ví dụ về sự tạo chuỗi băm



Đó là xor ở từng bit của phép lặp trước của c_k và yếu tố ngẫu nhiên, sau đó là f_k trên kết quả, và kết quả này được đưa vào phép lặp tiếp của c_k.

5.3.1 Khóa chữ ký

Để tạo được khóa bí mật, sk, các chuỗi $l+w-1\ n\ bit$ được chọn ngẫu nhiên (với PRF), trong l đầu tiên tạo thành khóa bí mật, $sk = (sk_1, ..sk_l)$ và các chuỗi $w-1\ n\ bit$ còn lại trở thành $r = (r_1, .., r_{w-1})$. Một khóa của hàm số, k được chọn ngẫu nhiên.

5.3.2 Khóa xác thực

Khóa công khai là:

$$pk = (pk_0, pk_1, .., pk_l) = ((r, k), c_k^{w-1}(sk_1, r), c_k^{w-1}(sk_2, r), .., c_k^{w-1}(sk_l, r))$$

Lưu ý rằng pk_0 có chứa r và k.

5.3.3 Ký

Để thực hiện một chữ ký: thông điệp, M, có độ dài m được phân tích để $M = (M_1, ..M_{l_1}), M_i \in \{0, w-1\}$ (tạo một biểu thức về M dựa trên w)/

Sau đó, kiểm tra tổng, C, chiều dài l_2 được tính toán và thêm vào:

$$C = \sum_{i=1}^{l_1} (w - 1 - M_i)$$

Để mà: $M + C = b = (b_0, ..b_l)$

Ta được chữ ký có dạng:

$$s = (s_1, .., s_l) = (c_k^{b_i}(sk_1, r), .., c_k^{b_l}(sk_l, r))$$

5.3.4 Xác thực

Để xác thực rằng chữ ký $b = (b_1, .., b_l)$ là được tái tạo từ M.

Nếu $pk = (c_k^{w-1-b_1}(s_1), .., c_k^{w-1-b_l}(s_l))$ thì chữ ký có hiệu lực.

W-OTS+ cung cấp một mức độ bảo mật tối thiểu $n - w - 1 - 2log(lw)$ bit [3]. Một chữ ký điển hình trong đó w=16 sử dụng SHA-256 (n=m=256) có ln bit hay 2,1 kb.

6. Sơ đồ chữ ký hình cây Merkle

Mặc dù các chữ ký sử dụng một lần có mức độ bảo mật cao, phù hợp để bảo mật việc ký và xác thực các giao dịch, nhưng chúng có một nhược điểm lớn đó là chỉ có thể được sử dụng một lần. Nếu một địa chỉ trên sổ cái phụ thuộc vào sự biến đổi với khóa công khai của một cặp khóa OTS duy nhất, điều này sẽ tạo ra một sổ cái chuỗi khối hạn chế, trong đó tất cả các quỹ ở một địa chỉ gửi cần phải chuyển sang địa chỉ khác sau mỗi giao dịch – nếu không chúng có nguy cơ bị đánh cắp.

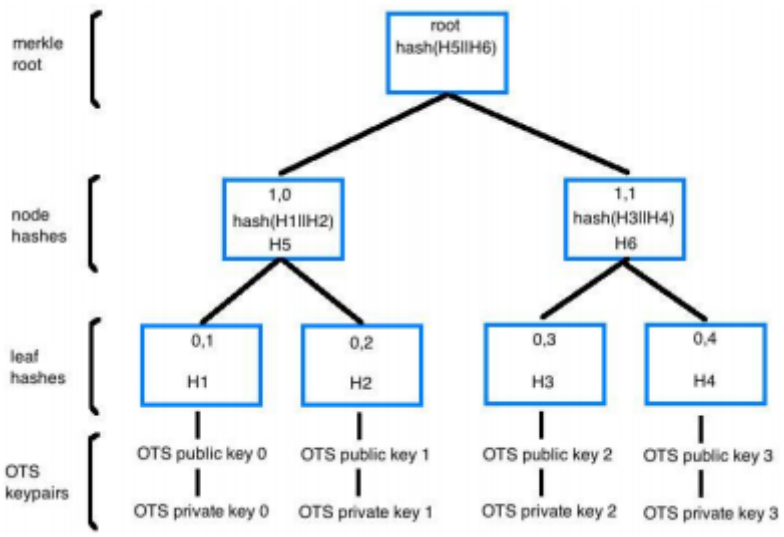
Do vậy, cần mở rộng sơ đồ chữ ký để sử dụng được nhiều hơn một chữ ký OTS có hiệu lực với mỗi địa chỉ trên sổ cái, để cho phép tạo ra số lượng tối đa các cặp khóa OTS. Một cây mã băm nhị phân, hay còn gọi là cây merkle chính là một phương pháp để đạt được điều này.

6.1 Cây mã băm nhị phân

Cây merkle là một cây có hình dạng đảo ngược, trong đó các nút con ở bên dưới được ghép với nhau, sử dụng hàm băm để mã hóa và tạo thành nút cha nằm ở trên, và cứ thế lên cao dần cho đến khi lên đến gốc. Có thể tính được các nút hoặc lá của cây được mã hóa bằng cách tính gốc.

Cây merkle được tạo từ n lớp lá và có chiều cao từ các lá được mã băm (lớp 0) qua từng lớp nút, lên đến gốc là h $(n = 2^h)$. Mỗi nút lá được tạo ra trong cuốn sổ cái giả thiết của chúng tôi bằng cách mã băm hóa một khóa công khai OTS đã được tạo ra từ trước. Ở cây dưới đây, chúng ta có thể thấy mỗi nút nằm ở phía trên của các lá đã được mã hóa được tạo ra bằng cách mã hóa các mã băm của các nút con được ghép với nhau.

Hình 3. Ví dụ về sơ đồ chữ ký cây Merkle



Quá trình này tiếp diễn từ dưới lên trên qua các lớp của cây cho đến khi lên tới gốc của cây đã được mã băm hóa, còn được gọi là gốc merkle.

Trong ví dụ ở trên, chúng ta coi gốc merkle là một khóa công khai, chúng ta có thể sử dụng 4 cặp khóa OTS để tạo ra bốn chữ ký sử dụng một lần được mã hóa an toàn. Gốc merkle của cây mã băm nhị phân có thể trở thành một địa chỉ trên sổ cái (có thể bằng phép băm lặp lại với tổng kiểm tra đã thêm vào). Một chữ ký S ở dạng đầy đủ của thông điệp M, với một cặp khóa OTS cho trước sẽ bao gồm: chữ ký s, số khóa ots, n và đường dẫn xác thực merkle, ví dụ, với cặp khóa OTS 0 (n=0):

S = s, n, khóa OTS công khai 0, H1, H2, H5, H6, gốc

Biết rằng có thể suy ra khóa công khai OTS và mã băm của lá từ s, và có thể tính các nút cha từ các nút con, nên chúng ta có thể rút gọn trở thành:

S=s, n, H2, H6, gốc

trong đó S có hiệu lực bằng cách xác thực khóa công khai OTS từ s và M, sau đó việc kiểm tra các mã băm từ đường dẫn xác thực merkle tái tạo ra một gốc merkle tương thích (khóa công khai).

6.2 Trạng thái

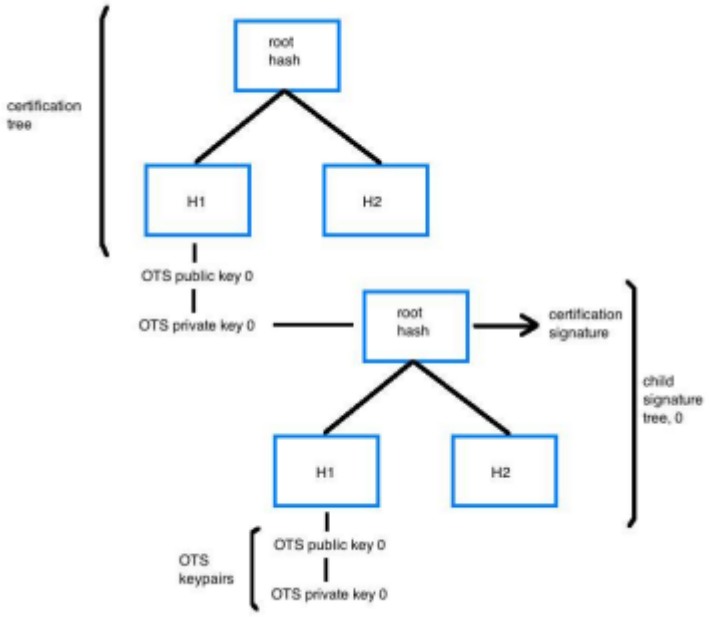
Để sử dụng sơ đồ chữ ký merkle (MSS) một cách an toàn, thì các khóa OTS phải không được sử dụng lại. Do đó, nó còn phụ thuộc vào trạng thái của chữ ký hay các giao dịch đã ký được ghi lại. Trong thế giới thông thường, đây có thể là một vấn đề, tuy nhiên một sổ cái chuỗi khối công khai bất biến là một phương tiện lý tưởng để lưu trữ các sơ đồ chữ ký mã hóa có các lịch sử giao dịch. Năm 2015, người ta đã giới thiệu một sơ đồ chữ ký mã hóa dựa trên mã băm gọi là SPHINCS, đưa ra các chữ ký không có lịch sử giao dịch với mức bảo mật 2^{128} bit [2].

6.3 Các siêu cây

Vấn đề của một MSS cơ bản là số lượng các chữ ký có thể tạo ra là giới hạn và tất cả các cặp khóa OTS phải được tạo ra trước khi tính toán cây merkle. Thời gian tạo ra khóa và chữ ký tăng lên theo số mũ với chiều cao của cây, h, nghĩa là để tạo ra các cây có nhiều hơn 256 cặp khóa cần sử dụng nhiều thời gian và chi phí tính toán hơn.

Một cách để làm chậm các tính toán trong quá trình tạo ra khóa và cây đồng thời tăng số lượng các cặp khóa OTS là sử dụng một cây mà bản thân nó đã bao gồm các cây merkle, gọi là một siêu cây. Tức là chúng ta sẽ ký vào gốc merkle của một cây con bằng một khóa OTS từ lá của một cây merkle cha, gọi là cây xác thực.

Hình 4. Kết nối các cây merkle



Ở dạng thức đơn giản nhất (chiều cao h=2), có thể tính trước cây xác thực bằng 2^1 cặp khóa OTS và khi chữ ký đầu tiên được yêu cầu, một cây merkle chữ ký mới (cây chữ ký 0) được tính toán và ký bởi một trong các cặp khóa OTS của cây xác thực. Cây chữ ký bao gồm n lá với các cặp khóa OTS tương ứng và những cặp này được dùng để ký vào các thông điệp như yêu cầu. Khi mỗi cặp khóa OTS trong cây chữ ký đã được sử dụng, thì cây chữ ký tiếp theo (cây chữ ký 1) được ký bởi cặp khóa OTS thứ hai từ cây xác thức và chúng ta có một đợt các chữ ký mới.

Chữ ký, S, của cấu trúc siêu cây này trở nên phức tạp hơn và sẽ bao gồm:

1. từ cây chữ ký: s, n, đường dẫn merkle, gốc

2. từ mỗi cây xác thực: s (của gốc cây merkle con), n, đường dẫn merkle, gốc

Về mặt lý thuyết chúng ta có thể ghép các lớp của cây từ cây xác thực để tăng MSS ban đầu ra đến vô hạn. Kích thước của chữ ký tăng tuyến tính với mỗi một cây được ký thêm, trong khi khả năng chữ ký của siêu cây tăng lên theo số mũ.

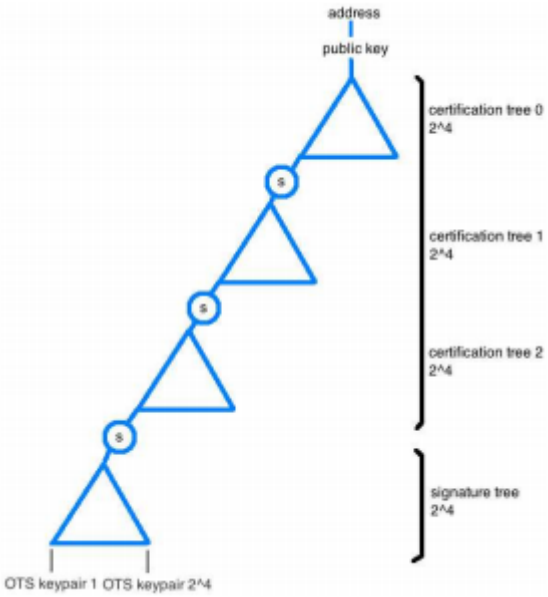
6.3.1 Các ví dụ về siêu cây

Để thấy MSS có thể được mở rộng ra một cách dễ dàng như thế nào với cấu trúc siêu cây, chúng ta hãy xem một cây xác thực có chiều cao h_1=5, với 2^5 lá và các cặp khóa OTS tương ứng. Gốc merkle của cây này được biến đổi để tạo ra một địa chỉ sổ cái. Một cây merkle khác, cây chữ ký có kích thước tương tự (h_2=5, 2^5 lá và các cặp khóa OTS) được tạo ra. Chúng ta thu được 32 chữ ký trước khi tạo ra một cây chữ ký tiếp theo. Tổng số chữ ký có thể có là $2^{h_1+h_2}$ trong trường hợp này là $2^{10} = 1024$.

Sử dụng một máy tính Macbook pro 2,7Ghz i5, 8gb ram, tạo các cặp khóa OTS và một cây xác thực merkle với các kích thước khác nhau, chúng ta thu được kết quả sau (mã python chưa tối ưu hóa, Winternitz OTS): 2^4 = 0,5 giây, 2^5 = 1,2 giây, 2^6 = 3,5 giây, 2^8 = 15,5 giây. Thời gian tạo chữ ký của một siêu cây gồm hai cây 2^4 là khoảng 1 giây so với 15,5 giây của một cây MSS 2^8 thông thường với cùng công suất tạo chữ ký.

Tăng chiều cao của siêu cây sẽ khiến công suất của nó phát triển theo hướng này. Một siêu cây gồm bốn cây xác thực 2^4 gắn với nhau và một cây chữ ký có kích thước 2^4 có khả năng tạo ra 2^{20} = 1.048.576 chữ ký, trong đó kích thước chữ ký tăng lên nhưng thời gian tạo chữ ký chỉ là 2,5 giây.

Hình 5. Cấu tạo của siêu cây

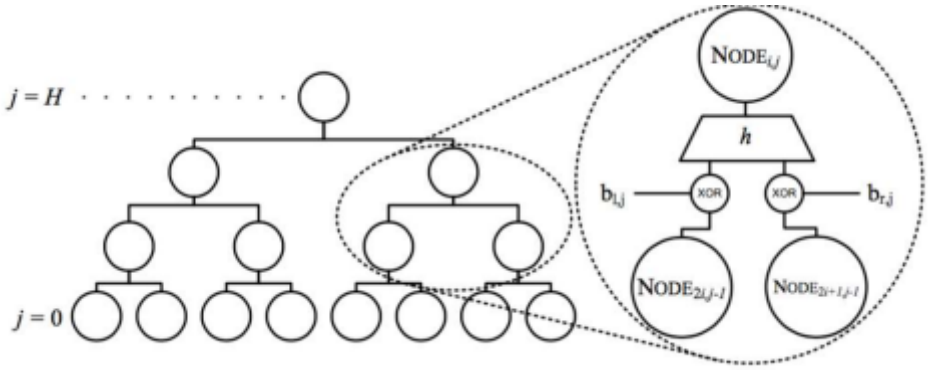


Một siêu cây không cần thiết phải có cấu trúc đối xứng, do vậy nếu ban đầu siêu cây gồm hai cây, sau đó nó có thể được mở rộng bằng cách ký thêm các lớp cây khác. Do đó các chữ ký từ một địa chỉ sổ cái phải bắt đầu từ nhỏ và cuối cùng sẽ tăng lên cùng với chiều dài của cây. Nếu sử dụng một siêu cây merkle để tạo và ký các giao dịch từ một địa chỉ sổ cái, chúng ta thường không bao giờ phải sử dụng nhiều hơn 2^{12} giao dịch. Do đó, việc có thể ký các giao dịch một cách an toàn với mức độ tính toán được giảm bớt 2^{20} lần với một siêu cây có chiều dài h=5 là hoàn toàn đủ.

6.4 Cây XMSS

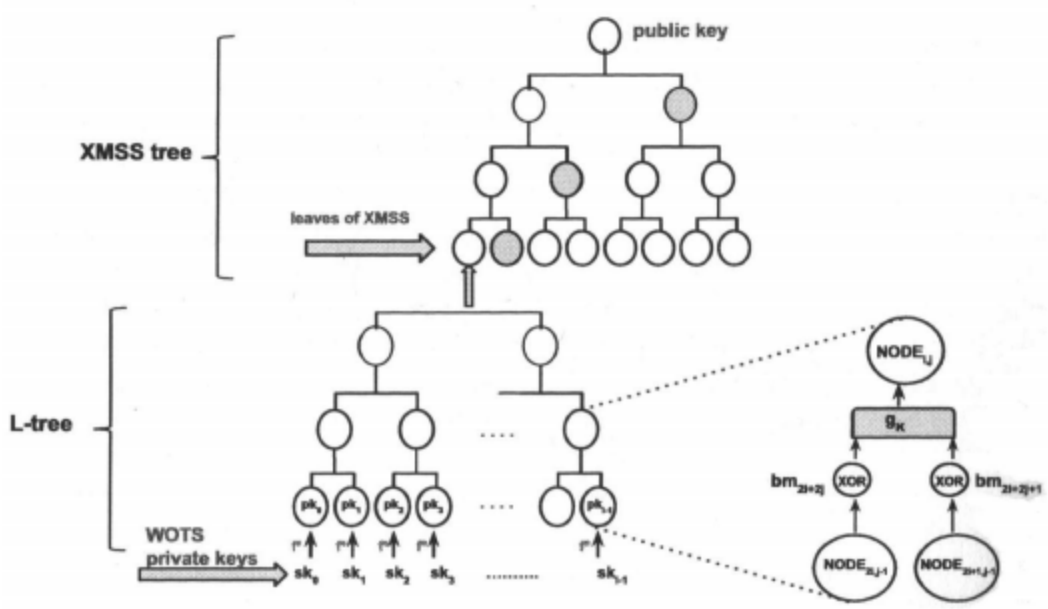
Sơ đồ chữ ký merkle mở rộng (XMSS) được đề xuất lần đầu bởi Buchmann và các cộng sự trong năm 2011 và được xuất bản dưới dạng một bản thảo IETF vào năm trước [4][7]. Sơ đồ này mang tính bảo mật chuyển tiếp và kẻ tấn công không thể làm giả các thông điệp, chỉ với các yêu cầu bảo mật tối thiểu: một PRF và một hàm băm chống đoán được h từ h_2 (với h_2 là hàm băm của h). Sơ đồ này cho phép mở rộng các chữ ký sử dụng một lần qua một cây merkle, mà sự khác biệt chính là ở chỗ, sử dụng bitmask XOR của các nút con trước khi kết hợp các mã băm vào nút cha. Việc sử dụng bitmask XOR cho phép khả năng có thể thay thế hàm băm chống xung đột.

Hình 1. Cấu trúc cây XMSS



Các lá của cây không phải là các mã băm cặp khóa OTS, mà gốc của các cây con hình chữ L nắm giữ các khóa OTS công khai với l miếng tạo thành các lá ở phần dưới của cây. Winternitz OTS+ được sử dụng làm các chữ ký sử dụng một lần (mặc dù phiên bản 2011 được mô tả đầu tiên).

Hình 7. Cấu tạo XMSS [8]



Chiều dài bit của khóa công khai XMSS là $(2(H+\lceil logl \rceil)+1)n$, chữ ký XMSS có chiều dài $(l+H)n$, và chiều dài của khóa chữ ký bí mật XMSS là <2n.

Buchmann đã đưa ra kết quả chạy trên máy Intel(R) i5 2,5 Ghz với cây XMSS có chiều cao h=20, trong đó w=16 và hàm băm mã hóa được sử dụng là SHA-256 (m=256) với gần một triệu chữ ký. Với các tham số và phần cứng tương tự, việc ký tên mất 7 mili giây, xác thực mất 0,52 mili giây và việc tạo khóa là 466 giây. Mức độ bảo mật mà các tham số này tạo ra là 196 bit với một khóa công khai có kích thước 1.7kb, khóa riêng tư là 280 bit và chữ ký 2,8kb. XMSS là một sơ đồ tiềm năng mà nhược điểm chính của nó là thời gian tạo khóa vô cùng lâu.

6.5 Hiệu suất của cây XMSS

Sử dụng một thư viện python chưa tối ưu hóa được xây dựng cho một cấu trúc kiểm nghiệm QRL gồm một cây XMSS có 4096 lá (h=12) trong đó tất cả các khóa và bitmask được tạo ra từ một PRF dựa trên hàm băm sẽ mất 32 giây trên cùng một phần cứng như mô tả ở trên (Macbook pro 2,7Ghz i5, 8gb ram). Thời gian này đã bao gồm thời gian tạo ra qua PRF hơn 8000 bitmask và trên 300,000 mảnh khóa riêng tư (sk). Một giải thuật giao nhau trên cây merkel hiệu quả hơn và nhu cầu chỉ cần tạo ra w-1 mã băm với mỗi hàm chuỗi khóa bí mật trong WOTS+ thay vì 2^{w-1} với WOTS sẽ góp phần cải thiện hiệu suất của một MSS thông thường.

Cấu trúc này có thể đạt được một kích thước chữ ký hoàn chỉnh là 5,75 kb (mã hóa chuỗi thập lục phân 11,75kb) bao gồm: cặp khóa OTS n, chữ ký, đường dẫn XMSS, khóa công khai OTS và khóa công khai của cây XMSS (bao gồm seed của khóa công khai PRF và gốc cây XMSS).

Với các cây có các công suất tạo chữ ký khác nhau bằng PRF và một seed ngẫu nhiên, chúng ta có được hiệu suất sau: (h=9) 512 4.2 giây, (h=10) 1024 8.2 giây, (h=11) 2048 16.02 giây.

7. Sơ đồ chữ ký được đề xuất

7.1 Các yêu cầu về bảo mật

Trong thiết kế của đồng QRL, khả năng bảo mật mã hóa của sơ đồ chữ ký phải đảm bảo chống lại được các cuộc tấn công thông thường cũng như tấn công của tính toán lượng tử hiện tại cũng như trong nhiều thập kỷ sắp tới. XMSS sử dụng SHA-256, với w=26, mang lại sự bảo mật ở mức 196 bit và được dự đoán có khả năng chống lại các cuộc tấn công làm giả cho đến tận năm 2164 [9].

7.2 Các chữ ký của QRL

Đồng QRL được đề xuất sử dụng một sơ đồ chữ ký dạng siêu cây bất đối xứng, có thể mở rộng được bao gồm các cây XMSS tạo thành chuỗi. Sơ đồ này có tác dụng kép là sử dụng một sơ đồ chữ ký đã xác thực và vừa tạo ra các địa chỉ trên sổ cái có khả năng ký các giao dịch, nhờ vậy tránh được thời gian trễ trước tính toán xảy ra trên các cấu trúc XMSS khổng lồ. Trong sơ đồ này W-OTS+ đã được chọn là chữ ký dùng một lần dựa trên mã băm vì các lý do bảo mật cũng như do hiệu suất của nó.

7.3 Cấu trúc của siêu cây

7.3.1 Các kích thước của khóa và chữ ký

Khi số cây trong một siêu cây tăng, kích thước khóa và chữ ký cũng tăng lên tuyến tính – nhưng công suất tạo chữ ký tăng theo hàm mũ. Kích thước của các khóa công khai và chữ ký được tạo ra từ các cây XMSS (theo mô tả năm 2011), trong đó w = 16, m = 256, h là chiều cao của cây là SHA-256 là giải thuật băm mã hóa, là:

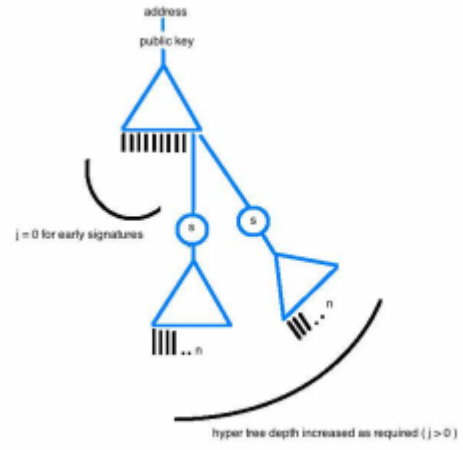
- h = 2, 2^2 chữ ký: khóa công khai 0,59kb, chữ ký 2,12kb (0.4 giây)
- h = 5, 2^5 chữ ký: khóa công khai 0,78kb, chữ ký 2,21kb (0.6 giây)
- h = 12, 2^{12} chữ ký: khóa công khai 1,23kb, chữ ký 2,43kb (32 giây)
- h = 20, 2^{20} chữ ký: khóa công khai 1,7kb, chữ ký 2,69kb (466 giây [3])

Chúng ta cũng có thể chấp nhận đánh đổi để tạo ra một siêu cây XMSS (4 cây, j=3, h=5) với công suất tạo chữ ký cuối cùng là 2^{20} trong vòng 3 s và chữ ký có kích thước 8,84kb, thay vì 466s và kích thước chữ ký là 2,69kb.

7.3.2 Cấu trúc bất đối xứng

Tạo một cây bất đối xứng cho phép tạo ra các chữ ký sớm trong một cấu trúc cây XMSS, cấu trúc này có thể được mở rộng khi cần thiết để tạo ra các chữ ký sau này; và điều này có ảnh hưởng tới tổng công suất tạo chữ ký. Lý do của việc này là điều này không có ảnh hưởng tới ứng dụng của sổ cái của chuỗi khối và ví tiền có thể cho phép người dùng lựa chọn giữ khả năng tạo chữ ký mạnh và kích thước của chữ ký/khóa. Chiều cao cây tối đa j=2 là đủ cho mọi hoàn cảnh.

Hình 8. Siêu cây có cấu trúc bất đối xứng



7.4 Thông số của siêu cây QRL

Các thông số mặc định sau đây được áp dụng cho một cấu trúc siêu cây tiêu biểu:

j = 0 ($j \in \{0 \leq x \leq 2\}$), h = 12 ($h \in \{1 \leq x \leq 14\}$), giới hạn trên của số lượng chữ ký có thể tạo ra: 2^{36}, kích thước chữ ký tối thiểu: 2,21kb, kích thước chữ ký tối đa: 7,65kb.

ví dụ, một cây XMSS, h=12 với 4096 chữ ký có sẵn, có thể được mở rộng thêm nhiều cây lên tới h=14 nếu được yêu cầu. Với hầu hết mọi người dùng, có thể không cần phải sử dụng thêm cây nữa.

7.4.1 Ví dụ chữ ký QRL

Giả sử cấu trúc siêu cây phức tạp nhất trong đó j = 2 và h = 14, chữ ký của thông điệp giao dịch, m, trong đó n là vị trí của cặp khóa OTS với mỗi cây XMSS đòi hỏi:

- Cây chữ ký j = 2: chữ ký OTS của m, n, bằng chứng xác thực merkle, gốc merkle của cây chữ ký
- Cây xác thực, j = 1: chữ ký OTS của gốc merkle từ cây chữ ký (j = 2), n, bằng chứng xác thực merkle, gốc merkle
- Cây XMSS ban đầu, j = 0: chữ ký OTS của gốc merkle (j = 1), n, bằng chứng xác thực của merkle, gốc merkle

Việc xác thực sẽ bao gồm việc tạo ra khóa OTS công khai từ m và chữ ký, sau đó việc xác nhận bằng chứng xác nhận merkle được cung cấp sẽ tạo ra gốc merkle của cây chữ ký. Đây chính là thông điệp của chữ ký OTS tiếp theo và từ đây, khóa OTS công khai tiếp theo được tạo ra, bằng chứng xác nhận merkle được cung cấp được sử dụng để tái tạo ra gốc merkle của cây xác thực, trở thành thông điệp cho chữ ký OTS của cây xác thực tiếp theo, và tiếp tục như vậy. Chữ ký chỉ có hiệu lực nếu gốc merkle của cây cao nhất, cây XMSS ban đầu (j=0) đã được tạo ra đúng cách.

Chú ý rằng không cần phải có các khóa OTS công khai để xác thực chữ ký của cây XMSS. Thực ra có thể suy đoán được gốc merkle của mỗi cây và do đó có thể loại bỏ với việc xác thực chữ ký siêu cây nếu địa chỉ gửi trên sổ cái đã được biết (do đây là một đạo hàm được tính từ gốc merkle của cây XMSS xác thực cao nhất (j=0) trong chữ ký QRL – xem phần Tải khoản ở dưới.

7.5. PRF

PRF từ seed. HMAC DRBG.

7.6 Ví tất định

Sử dụng một SEED đơn lẻ, chúng ta có thể tạo được một cây XMSS lớn đủ cho phần lớn người dùng trong một thời gian dài. SEED này được tạo ra từ một nguồn entropy bảo đảm, sau đó được đưa qua một hàm PRF bảo mật để tạo ra một bộ các khóa giả ngẫu nhiên tạo ra cây. Một nhược điểm của việc dùng chính cây XMSS đó là người dùng bị hạn chế ở một địa chỉ duy nhất (mặc dù với MSS việc bị lộ khóa công khai không phải là một vấn đề đáng lo ngại).

Một địa chỉ bitcoin hoặc ethereum được dẫn xuất từ khóa công khai liên quan và do vậy, một khóa riêng tư hay khóa công khai chỉ có thể tạo ra một địa chỉ duy nhất. Tuy nhiên, một địa chỉ XMSS được tạo ra từ khóa công khai, PK, có chứa gốc merkle và SEED công khai. Nếu SEED không đổi nhưng số lượng các cặp khóa OTS để tính ra cây thay đổi, thì với mỗi biến thể, gốc merkle sẽ thay đổi. Do đó với mỗi lần cộng hoặc trừ một cặp khóa OTS, địa chỉ thu được sẽ thay đổi.

Phần mềm ví/nút có thể sử dụng đặc điểm này để tạo ra vô vàn các biến thể của cây XMSS (mở rộng/rút gọn cây khi cần thiết qua việc sử dụng SEED ban đầu), nhờ đó có thể tạo ra tối đa số lượng các địa chỉ cần thiết. Để ghi lại thông tin này một cách an toàn, tóm tắt cũng như lịch sử các giao dịch là một vấn đề dễ dàng về mặt tính toán.

8 Các tham số thiết kế của tiền điện tử

Phần còn lại của bài viết này nhằm mô tả các thông số thiết kế đề xuất cho cuốn sổ cái của QRL. Cuốn sổ cái này chính là một chuỗi khối (blockchain) công khai, có tính bảo mật cao, có khả năng chống lại các cuộc tấn công thông thường cũng như các tính toán lượng tử. Bài viết này là bản thảo đầu tiên do đó bất kỳ nội dung nào cũng có thể được thay đổi.

8.1 Bằng chứng về cổ phần (proof-of-stake)

QRL sẽ là một đồng tiền có sổ cái là một blockchain công khai, với hình thức bảo mật dựa trên giải thuật bằng chứng cổ phần. Một giai đoạn kéo dài trong 10.000 khối. Những người xác nhận tính hiệu lực của cổ phần được xác định từ các giao dịch cổ phần trong giai đoạn trước. Ở đây, mỗi người xác nhận tính hiệu lực của cổ phần ký vào một giao dịch có chứa mã băm cuối cùng của một chuỗi lặp có chiều dài 10.000 mã băm (có thể áp dụng bitmask XOR trong mỗi lần lặp để giảm các yêu cầu bảo mật của hàm băm). Khi giao dịch cổ phần được xác nhận trên chuỗi,

mỗi nút trong mạng lưới bây giờ có thể gắn định dạng mã hóa của địa chỉ cổ phần vào chuỗi mã băm trong giai đoạn tiếp theo.

8.1.1 Thiết kế và sự ngẫu nhiên

Với mỗi khối, mỗi nút xác thực đang góp cổ phần trong giai đoạn hiện sẽ tiết lộ mã băm kế tiếp trong chuỗi để chứng minh về mặt mã hóa về sự tham gia và bỏ phiếu để trở thành block được chọn.

HMAC DRBG được chọn để tạo ra một chuỗi số giả ngẫu nhiên gồm 32 byte đầu ra từ dữ liệu seed được lấy từ chuỗi khối (ban đầu là khối gốc, sau đó entropy được bổ sung lấy từ các mã tiêu đề khối ghép với nhau gần nhất trong mỗi giai đoạn sau).

Do vậy mỗi khối mà người xác thực cổ phần đã chọn để trở thành khối được chọn được xác định bằng cách trở thành mã băm tiết lộ có số gần nhất với đầu ra PRF của khối đó. Việc này rất khó tính toán bởi những người nắm cổ phần không biết PRF tại thời điểm tạo ra chuỗi. Ngoài ra, một chuỗi băm lặp (nhập giá trị) về bản chất là một chuỗi số ngẫu nhiên. Cuối cùng, kể cả nếu những người xác thực cổ phần cấu kết với nhau bằng một cách nào đó thì họ cũng không thể biết nội dung chuỗi băm của những người xác thực cổ phiếu khác vì chúng vẫn chưa bị tiết lộ.

Để ngăn cản một cuộc tấn công vào khối, nếu một địa chỉ không thể tạo được một khối có hiệu lực sau khi đã nộp một mã băm tiết lộ, thì địa chỉ đó sẽ mất phần thưởng khối, và địa chỉ đó bị cấm tham gia trong một quãng thời gian.

Để giảm thiểu cuộc tấn công từ các nút chủ yếu hoặc các địa chỉ của người xác thực cổ phiếu có số dư trong tài khoản thấp, một ngưỡng danh sách người xác thực linh hoạt được sử dụng. Phần thưởng khối được chi trả theo tỉ lệ phần trăm dựa trên số dư của tài khoản người nắm cổ phiếu. Với thông điệp băm bị tiết lộ, mỗi nút cũng để lộ một mã băm gốc cây merkle gòm một danh sách các mã băm tx đã phân loại trong các tập hợp các giao dịch của họ cùng với số lượng các giao dịch đang chờ khối. Mỗi nút lấy một tỉ lệ phần trăm ở trên và ở dưới để xem bao nhiêu giao dịch được mong đợi trong khối kế tiếp. Nếu khối đó rỗng hoặc có số lượng giao dịch ít hơn mong đợi, thì số lượng những người xác thực cổ phiếu được phép sẽ giảm dần (loại trừ những người xác thực cổ phiếu từ nghèo tới giàu) trên mỗi khối. Nếu các nút lựa chọn khối đang hoạt động chân thật, thì điều ngược lại là đúng và số lượng những người xác thực cổ phiếu đang tham gia sẽ tăng. Các tài khoản có thể sẽ không di chuyển trong giai đoạn mà chúng đang góp vốn – điều này sẽ tránh được các nỗ lực để khoan sự lựa chọn khối bằng việc tạo ra vô vàn các địa chỉ người xác thực cổ phiếu.

8.2 Các khoản phí

Do các kích thước giao dịch lớn hơn so với các loại tiền khác , nên mỗi giao dịch phải có một khoản phí giao dịch. Tác giả của bài viết này cho rằng các thị trường phí nhân tạo (xem bitcoin) là không cần thiết và phản tác dụng với ý tưởng về một sổ cái có blockchain công khai. Mỗi giao

dịch đều ngang bằng với các giao dịch khác nếu chúng trả một khoản phí tối thiểu. Chi phí tối thiểu mà những người đào có thể chấp nhận phải được thả nổi và do thị trường quyết định, tức là các nút/người đào sẽ cạnh tranh và đặt ra giới hạn dưới của các khoản phí. Một giá trị tối thiểu tuyệt đối sẽ được áp đặt ở cấp độ giao thức. Do đó, những người đào sẽ yêu cầu những giao dịch từ tập hợp để bao gồm trong một khối do họ quyết định.

8.3 Khối

8.3.1 Thời gian giữa hai khối được tạo

Khoảng thời gian để một khối mới được tạo ra trong Bitcoin là khoảng 10 phút, nhưng với các yếu tố tự nhiên, khoảng thời gian này đôi khi có thể kéo dài hơn. Các đồng tiền mới tạo ra như ethereum đã cải tiến điểm này và có thời gian giữa các khối ngắn hơn nhiều (15 giây) mà không ảnh hưởng đến tính bảo mật hay sự tập trung hóa người đào từ tỉ lệ cao các khối tách biệt khỏi chuỗi (orphan). Ethereum sử dụng một phiên bản được chỉnh sửa của giao thức Greedy Heaviest Observed Subtree cho phép các khối tách biệt đó được tham gia vào chuỗi khối và được nhận thưởng [13,5].

Do đồng QRL dự định sử dụng thuật toán bằng chứng về cổ phiếu ngay từ đầu, nên chúng tôi hy vọng thời gian tạo khối mới trong QRL là từ 15 đến 30 giây.

8.3.2 Phần thưởng khi tạo được khối

Mỗi khối mới được tạo ra sẽ bao gồm một giao dịch “dựa trên xu” đầu tiên gồm một địa chỉ đào trong đó phần thưởng bằng tổng phần thưởng của xu và tổng các phí giao dịch trong khối. Phần thưởng khối được phân chia dựa trên số dư của địa chỉ của người xác thực cổ phiếu.

Phần thưởng khối được tính lại bởi nút đào trong mỗi khối và tuân theo lịch nhả xu.

8.3.3 Kích thước khối

Để tránh gây tranh cãi, chúng tôi sử dụng một giải pháp dựa trên đề xuất Bitpay để tăng kích thước của khối dựa trên một bội số, x, của kích thước trung bình, y, của khối z cuối cùng [12]. Việc sử dụng số trung bình tránh việc những người thợ đào đưa các khối rỗng hoặc các khối quá lớn vào để thay đổi kích thước khối trung bình. Khi đó x và z là những quy tắc đồng thuận mà mạng lưới phải tuân thủ.

Do đó, kích thước tối đa, b có thể được tính bằng:

$$b=xy$$

8.4 Đơn vị và mệnh giá đồng tiền

QRL sử dụng đơn vị tiền tệ, quantum (số nhiều quanta) làm đơn vị tiền tệ cơ sở. Mỗi quantum có thể được chia nhỏ thành các đơn vị như sau:

•1: Shor
• 10^3 : Nakamoto
• 10^6 : Buterin
• 10^{10} : Merkle
• 10^{13} : Lamport
• 10^{16} : Quantum

Do đó, mỗi giao dịch có sử dụng một phần của quantum thực ra đang sử dụng một số nguyên Shor rất lớn. Các chi phí giao dịch được thanh toán và tính bằng đơn vị Shor.

8.5 Tài khoản

Số dư của người dùng được giữ trong các tài khoản. Mỗi tài khoản đơn giản chỉ là một địa chỉ trên sổ cái có thể tái sử dụng và được mô tả bằng một chuỗi bắt đầu với chữ "Q".

Có thể tạo địa chỉ bằng cách chạy một hàm SHA-256 trên gốc merkle của cây xác thực XMSS cao nhất. Một kiểm tra tổng 4 byte được bổ sung vào đây (được hình thành từ bốn byte đầu tiên của một mã băm SHA-256 kép của gốc merkle) và chữ cái "Q" đã tính trước, nghĩa là trong mã giả python:

Q + sha256(merkle root) + sha256(sha256(merkle root))[: 4]

Một địa chỉ tài khoản điển hình:

Qcea29b1402248d53469e352de662923986f3a94cf0f51522bedd08fb5e64948af479

Mỗi tài khoản có một số dư được biểu hiện bằng đơn vị tiền tệ quanta mà có thể được chia nhỏ xuống đơn vị Shor.

Mỗi giao dịch sử dụng một cặp khóa OTS mới đều được ghi lại vào lịch sử của mỗi địa chỉ và QRL chứa tất cả các khóa công khai đã từng sử dụng (có thể được cắt xén do nó có thể được tái tạo trong khi đang hoạt động từ chữ ký và thông điệp của giao dịch nhưng sẽ mang tính cấp tốc) với mỗi tài khoản. Một bộ đếm giao dịch gọi là một nonce sẽ tăng lên sau khi mỗi giao dịch được gửi đi từ một tài khoản. Việc này cho phép các ví không chứa toàn bộ chuỗi khối có thể theo dõi vị trí của chúng trong một sơ đồ chữ ký siêu câu merkle có lịch sử giao dịch.

8.6 Phát hành xu

8.6.1 Các cân nhắc mang yếu tố lịch sử

Bitcoin là đồng tiền điện tử phi tập trung hóa đầu tiên và ban đầu thử nghiệm việc không có giá trị tiền tệ, do đó có thể phân phối toàn bộ số tiền qua việc đào. Gần đây, đồng Zcash đã chọn một quy trình tương tự trong đó một tỉ lệ % phần thưởng từ việc đào xu trong giai đoạn đầu của giai đoạn được chuyển sang dự án mở - do đó giá biến động lớn.

Thay vào đó, các đồng tiền khác như ethereum đã bán một tỉ lệ % lớn của nguồn tiền xu cuối cùng như một phần của lần Phát hành xu lần đầu ra công chúng (ICO). Điều này có ưu điểm là những người sử dụng đồng tiền đầu tiên vẫn có được lợi nhuận từ việc hỗ trợ dự án, nhưng sau đó dự án có thể tạo ra đủ số tiền để tiếp tục phát triển dự án từ trứng nước. Cách tiếp cận theo kiểu ICO đồng thời cũng cho phép một thí trường có thể phát triển dễ dàng hơn khi các nhà đầu tư có thể tiếp cận một số lượng lớn các đồng xu để mua và bán từ khối ban đầu.

Đồng Auroracoin (2014) có một cách tiếp cận khác bằng cách trao cho mọi người ở Iceland một phần bằng nhau trong ICO, trong khi những nhà phát triển sẽ giữ 50% toàn bộ số xu cho bản thân họ.

Các loại tiền điện tử khác hoặc là sao chép hoàn toàn đồng bitcoin hoặc là sử dụng một chuỗi mới trên một nền tảng mã khác.

8.6.2 Sự chuyển số dư trong chuỗi

Có thể phát hành QRL dựa trên sự nắm bắt tình trạng của sổ cái bitcoin hiện tại đang được đưa vào khối gốc của QRL. Mục đích là cho phép những người sử dụng được tạo ra các giao dịch "nhập khẩu" một lần có chứa một thông điệp và chữ ký độc nhất (nghĩa là một địa chỉ ví QRL được tạo ngẫu nhiên được ký bởi một khóa bitcoin riêng tư từ một địa chỉ có số dư bitcoin tại thời điểm nắm bắt tình trạng). Tính năng này sẽ hoạt động cho tới một chiều cao khối nhất định và sau đó phần còn lại của số xu sẽ được đào như thông thường. Phần phình tra trong khối ban đầu sẽ được cắt đi ở cùng một độ cao khối. Một nhược điểm của điều này là mặc dù đảm bảo tính công bằng, nhưng nó lại gây khó khăn cho những người sử dụng tiền điện tử khác không phải là đồng bitcoin và điều này khá khó khăn về mặt kỹ thuật cho những người sử dụng mới. Một lo ngại về mặt kỹ thuật là có thể khôi phục một khóa công cộng ECDSA từ chữ ký và thông điệp. Điều này sẽ tiết lộ vĩnh viễn những khóa công cộng cho các địa chỉ bitcoin được sử dụng trong quá trình và sẽ cần phải di chuyển số dư tới một địa chỉ bitcoin được tạo ra ngẫu nhiên khác để phòng tránh.

(? có thể cung cấp tính năng này cho những người sử dụng đồng ethereum)

8.6.3 Đề xuất về sự phát hành – bản thảo

Lần phát hành đầu tiên đồng QRL sẽ như sau:

ICO trị giá 1 triệu quanta (4,7% tổng số xu) trước khi vận hành

Nắm bắt trạng thái của tất các các số dư của các địa chỉ bitcoin có trên 0,01 btc được sử dụng để hình thành khối ban đầu của QRL. Bất kỳ ai mong muốn chuyển với tỉ lệ 1:1 từ sổ cái bitcoin sang sổ cái QRL sẽ có thể làm được điều này cho tới chiều cao khối đạt 518400 (trong vòng 3 đến 6 tháng) qua ví nút.

Một triệu quanta khác sẽ được giữ trong địa chỉ khối ban đầu để sử dụng để làm cơ sở

Phần xu còn lại sẽ được đào (21, 000, 000 − (2, 000, 000+btc được chuyển vào cho tới khi chiều cao khối đạt 5184000))

8.7 Quy trình phát hành xu

Một đặc điểm nổi bật của bitcoin là sự khan hiếm và tổng giá trị của các đồng xu được phát hành là có giới hạn. QRL cũng sẽ giống với bitcoin ở điểm này và giá trị của số lượng xu được giới hạn ở $21x10^6$ quanta. Sự phân rã số mũ trong khoản tiền thưởng tìm block chỉ đến mức trần của số lượng xu. Điều này sẽ giúp giảm sự biến động do hiện tượng "chia nửa" của bitcoin.

Tổng số xu x = $21x10^6$, trừ số xu tạo ra ở khối ban đầu, y, sẽ giảm theo số mũ từ Z_0 và đi xuống mãi mãi. Đường cong phân rã được tính toán để phân phối các phần thưởng đào trong khoảng 200 năm (cho đến 2217AD, 420480000 blocks với thời gian giữa hai block được tạo ra là 15giây) cho tới khi chỉ còn 1 quanta là chưa được đào (mặc dù việc đào vẫn tiếp tục sau đó).

Số lượng xu còn lại ở khối t, Z_t có thể được tính bằng:

$$Z_t = Z_0 e^{-\lambda t}$$

Hệ số λ được tính từ: $\lambda = \frac{\ln Z_0}{t}$. Trong đó t là tổng số các khối trong lịch trình nhả khối cho tới quanta cuối cùng. Cho tới khối 518400, λ = $3{,}98590885111x10^{-08}$. Phần thưởng tìm được khối, b, được tính cho từng khối bằng:

$$b = Z_{t-1} - Z_t$$

Giữa khối ban đầu và khối số 518400, số dư bitcoin có thể được chuyển sang sổ cái qua các giao dịch nhập khẩu. Ở khối 518401 lịch trình nhả khối sẽ thay đổi mục tiêu, khóa chặt số dư mới được nhập, giảm Z_t và thay đổi phần thưởng tìm ra khối một cách phù hợp.

Tài liệu tham khảo:

[1] http://oxt.me/charts..
[2] D. Bernstein. Sphincs: practical stateless hash-based signatures. 2015.
[3] J Buchmann. On the security of the winternitz one-time signature scheme.
[4] J. Buchmann. Xmss – a practical forward secure signature scheme based on minimal security assumptions. 2011.
[5] V Buterin. Ethereum whitepaper. 2013.
[6] A. Hulsing. W-ots+ - shorter signatures for hash-based signature schemes. 2013.
[7] A. Hulsing. Xmss: Extended hash-based signatures. 2015.
[8] A Karina. An efficient software implementation of the hash-based signature scheme mss and its variants.

2015.
[9] A. Lenstra. Selecting cryptographic key sizes. 2001.
[10] R. Merkle. A certified digital signature. CRYPTO, 435, 1989.
[11] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.
[12] S Pair. A simple, adaptive blocksize limit. 2016.
[13] Yonatan Sompolinsky. Accelerating bitcoin's transaction processing fast money grows on trees, not
chains. 2014.
[14] A. Toshi. The birthday paradox. 2013.