Введение в перевод
Перевод данного документа на русский язык был выполнен Андреем Коваленко и заказан инвестором QRL Джастином Пуарье, основателем Tokenized Capital. Пожалуйста, рассматривайте исходный документ на английском языке как официальный, и ссылайтесь на него в случае каких-либо расхождений с текстом перевода. Также было сделано дополнение, предоставляющее читателям информацию об изменениях, касающихся выпуска монет, которые вступили в силу после появления документа. Была проведена успешная предпродажа и привлечены средства в размере 4,16 млн. долларов США в системах bitcoin и ether. Исходное содержание документа было оставленно неизменным для справки.

Квантовоустойчивый блок цепочки транзакций (QRL)

peterwaterland@gmail.com

Ноябрь 2016 г.

Абстракт

Условием длительного существования частных цифровых денег является их защищенность в свете развития компьютерных технологий. В данной работе представлены дизайн и конструкция блока цепочки транзакций криптовалюты с использованием хешированных цифровых подписей, устойчивых как к классическим атакам, так и тем, для осуществления которых используются квантовые вычисления.

1 Введение

Концепция одноранговой интернет блок цепочки транзакций валюты, записанной в виде цепочки блоков и защищенной механизмом доказательства выполнения работы (proof-of-work), была впервые опубликована в 2008 году [11]. Биткойн по сегодняшний день остается наиболее широко используемой криптовалютой. Сотни аналогичных систем блоков цепочки транзакций криптовалюты были созданы впоследствии, но за некоторыми исключениями все они полагаются на одно и то же использование криптографии с открытым ключом, базирующейся на эллиптической кривой (ECDSA) для генерации цифровых подписей, которые позволяют безопасно проверять транзакции. Наиболее часто используемые схемы подписей сегодня, такие как ECDSA, DSA и RSA, теоретически уязвимы для атаки с использованием квантовых вычислений. Было бы полезно исследовать возможности дизайна и конструкции квантовустойчивой системы блок цепочки транзакций, чтобы противостоять потенциальному проявлению неожиданного развития в нелинейных квантовых вычислениях.

2 Безопасность транзакций биткойн

В настоящее время существует возможность тратить (неиспользованные транзакции) адреса биткойна, создавая транзакцию, содержащую действительную подпись на основе эллиптической кривой (secp256k1) с использованием закрытого ключа ($x \in N \mid x < 2^{256}$) для этого конкретного адреса биткойна. Если по-настоящему случайно сгенерированный закрытый ключ хранится в секрете или потерян, тогда было бы разумно ожидать, что никакие средства с этого адреса уже никогда не смогут быть переведены.

Вероятность конкретного совпадения с частным ключом биткойна равна 1 из 2^{256}. Вероятность совпадения ключа биткойн-адреса может быть показана на примере решения проблемы дня рождения. Количество адресов биткойнов, которые должны быть сгенерированы, чтобы привести к коллизии с вероятностью в 0,1%, составляет $5{,}4\times10^{23}$ [14].

Тем не менее, когда транзакция подписана, открытый ключ ECDSA посылающего адреса становится доступен и сохраняется в цепочке блоков. Наилучшей практикой было бы не использовать адреса повторно, но по состоянию на ноябрь 2016 года 49,58% всего баланса биткойнов хранится в адресах с доступными открытыми ключами [1].

3 Векторы квантовых вычислительных атак

RSA, DSA и ECDSA остаются защищенными на основе вычислительной сложности факторизации больших целых чисел, решения задач дискретного логарифмирования и дискретного логарифмирования эллиптической кривой. Квантовый алгоритм Шора (1994) решает факторизацию больших целых чисел и дискретные логарифмы за полиномиальное время. Таким образом, квантовый компьютер теоретически может восстановить закрытый ключ по открытому ключу ECDSA. Считается, что ECDSA более уязвима для квантовой атаки, чем RSA, из-за использования меньшей длины ключей, от 1300 до 1600 кубитов (2^{11}), и квантовый компьютер способен решить 228-битный ECDSA.

Развитие квантового компьютера, если базироваться на общедоступной информации, пока не превысило 2^5 кубитов или факторизацию малых чисел (15 или 21). Однако в августе 2015 года Агентство национальной безопасности США выступило против использования криптографии эллиптической кривой, основываясь именно на проблематике квантовых вычислений. Неизвестно, насколько продвинутый уровень квантовых вычислений может существовать в настоящее время или будут ли опубликованы материалы о любых возможных прорывах в исследованиях в этой области, что позволило бы обеспечить пост-квантовую безопасность для криптографических протоколов, используемых на данный момент в Интернете. В связи с тем, что природа происхождения его некоторым образом направленна на подрыв устоявшегося строя, биткойн может оказаться одной из первых целей для оппонента с квантовым компьютером.

Если бы существенное продвижение в сфере квантовых вычислений происходило публично, разработчики узлов могли бы осуществить реализацию квантовоустойчивые схем криптографической подписи для биткойна и побудить всех пользователей переносить свои балансы с адресов на основе ECDSA на новые квантовобезопасные адреса. Для того, чтобы уменьшить долю адресов, которые могут подвергнуться внешнему воздействию, было бы разумно прекратить возможность повторного использования открытых ключей на уровне протокола. Такое запланированное обновление также привело бы к возможному перемещению 1 миллиона монет, принадлежащих Сатоши Накамото с соответствующей ценовой волатильностью.

Менее благоприятным сценарием было бы развитие нелинейных квантовых вычислений без огласки, с последующей нюансированной квантовой компьютерной атакой на адреса биткойнов с доступными открытыми ключами. Изъятие средств с подобных адресов может оказать разрушительное влияние на обменную цену биткойна в связи с возникновением нового фактора, оказывающего мощное давление на необходимость продаж и полной потери уверенности в системе, поскольку масштабы изъятий станут известны. Роль биткойна как хранилища валюты («цифрового золота») в результате очень сильно пострадает с крайне негативными последствиями для всего мира. В этом контексте авторы считают целесообразной необходимость экспериментировать с квантовоустойчивыми криптографическими подписями для блока цепочки транзакций криптовалюты, а также, возможно, создания хранилища резервных запасов на случай событий "черного лебедя".

4 Квантоустойчивые подписи

Существует несколько основных криптографических систем, которые считаются квантовоустойчивыми: криптография на основе хешей, криптография на основе кода, криптография на основе матрицы, криптография, основанная на многомерных квадратичных системах, и криптография на базе секретного ключа. Считается, что все эти схемы могут противостоять как классическим атакам, так и тем, для осуществления которых используются квантовые вычисления, при наличие достаточно длинных ключей.

Существуют передовые защищенные схемы цифровой подписи на основе хеширования при соблюдении минимальных требований к безопасности, которые полагаются только

на устойчивость к коллизиям криптографической хеш-функции. Изменение избранной хеш-функции создаст новую схему цифровой подписи на основе хеша. Цифровые подписи на основе хеширования хорошо изучены и являются наиболее приемлемыми для постквантовых подписей в будущем. Таким образом, мы можем говорить о них, как об избранном классе постквантовой подписи в рамках квантовоустойчивой системы блока цепочки транзакций.

5 Цифровые подписи на основе хеша

Квантовоустойчивые подписи на основе хеша базируются на безопасности односторонней криптографической хеш-функции, которая принимает сообщение m и выдает итоговое значение хеша h фиксированной длины n, например, SHA-256, SHA-512. Использование криптографической хеш-функции должно сделать невыполнимыми вычисления путем простого перебора m по значению h (устойчивость к обнаружению прообраза) или h исходя из h_2, где $h_2 = hash(h)$ (устойчивость к обнаружению второго прообраза), в то же время должно быть очень трудно найти два сообщения ($m1 \neq m2$), на основании которых может быть создан один и тот же h (устойчивость к коллизиям).

Квантовый алгоритм Гровера может быть использован для попытки найти хеш-коллизию или выполнить предварительную атаку, чтобы найти m, для чего потребуется $O(2^{n/2})$ операций. Таким образом, чтобы поддерживать 128-битную безопасность, необходимо выбрать длину итогового значения хеша, n не менее 256 бит - предполагая создание идеальной криптографической хеш-функции.

Для цифровых подписей на основе хеша требуется открытый ключ pk для проверки и закрытый ключ, sk для подписания сообщения. Различные базирующиеся на хеше одноразовые подписи (OTS) будут обсуждены далее относительно их пригодности для включения в качестве части цепочки блоков.

5.1 Одноразовая подпись Лампорта-Диффи

В 1979 году Лампорт описал одноразовую подписи у на основе хеша для сообщения длиной m бит (обычно это результат функции хеширования, устойчивой к коллизиям). Генерация ключей создает m пар случайных секретных ключей, $sk_j^m \in \{0, 1\}^n$, где $j \in \{0, 1\}$, например закрытый ключ: $sk = ((sk_0^1, sk_1^1), .., .., (, sk_0^m, sk_1^m))$. Пусть f - это односторонняя хеш-функция $\{0, 1\}^n \rightarrow \{0, 1\}^n$, с m пар сгенерированных открытых ключей, $pk_j^m = f(sk_j^m)$, т.е. открытым ключем будет: $pk = ((pk_0^1, pk_1^1), ..., .., (pk_0^m, pk_1^m))$. Подписание включает побитовое обнаружение хеша сообщения для выбора sk_j (т.е. если бит = 0, $sk_j = sk_0$, бит = 1, $sk_j = sk_1$), создающее подпись: $s = (sk_j^1, .., sk_j^m)$, которая показывает половину закрытого ключа. Чтобы проверить подпись, побитовый ($j \in \{0, 1\}$) просмотр хеша сообщения проверяет, что $(pk_j = f(sk_j))^m$.

Предполагая, что с учетом алгоритма Гровера требуется 128-битная защита, где длина сообщения является фиксированным хеш-результатом из SHA256, m = 256 и n = 256, в результате чего получается pk = sk = 16 кб и подписи в 8 кб для каждой используемой одноразовой подписи. Подпись Лампорта должна использоваться только один раз, может быть сгенерирована очень быстро, но она страдает от большого размера ключей, подписи и увеличивает объем обмена информации, что делает ее нецелесообразной для публичного блока цепочки транзакций.

5.2 Одноразовая подпись Винтерница

Для итогового значения сообщения (M) длины m бит с закрытыми и открытыми ключами длины, n бит, односторонняя функция, f: $\{0, 1\}^n \rightarrow \{0, 1\}^n$ и параметр Винтерница, $W \in N \mid w > 1$, общая идея одноразовой подписи Винтерница заключается в применении итерационной хеш-функции в списке случайных закрытых ключей $sk \in \{0, 1\}^n$, $sk = (sk_1, .., Sk_{m/w})$, создавая цепочки

хешей длины, w - 1, заканчивающихся открытыми ключами ($pk \in N \mid \{0, 1\}^n$), $pk_x = f^{2^{w-1}}(sk_x)$, $pk = (pk_1, .., pk_{m/w})$.

В отличие от побитового осмотра итогового значения сообщения в подписи Лампорта сообщение анализируется w бит за раз, чтобы извлечь число, $i \in N$, $i < 2^w$ - 1, из которого генерируется подпись, $s_x = f^i(sk_x)$, $s = (s_1, ..s_{m/w})$. С ростом w, обеспечивающим компромисс между ключами меньшего размера и подписями для увеличения необходимых вычислительных усилий [10].

Проверка включает простую генерацию $pk_x = f^{2^{w-1}-i}(s_x)$ из M, s и подтверждение совпадения открытых ключей.

Использование SHA-2 (SHA-256) в качестве односторонней криптографической хеш-функции, f: m = 256 и n = 256, с w = 8, приводит к $pk = sk = s$ размеру $\frac{(m/w)n}{8}$ байт = 1 кб и для генерации pk требуются f^i итераций хеша, где $i = \frac{m}{w} 2^{w-1} = 8160$ для генерации каждой пары ключей одноразовой подписи. При w = 16 количество ключей и подписей уменьшается вдвое, но при этом значение i = 1048560, что становится непрактичным.

5.3 Дополненная одноразовая подпись Винтерница (W-OTS+)

Бахман представил вариацию исходной одноразовой подписи Винтерница, изменяя повторяющуюся одностороннюю функцию, применяя ее, в отличае от исходного варианта, к случайному числу x, многократно, но на этот раз с параметризованным ключом k, который генерируется из предыдущей итерации $f_k(x)$. Такую подпись уже практически невозможно подделать при адаптивных избирательных атаках сообщений с использованием псевдослучайной функции (PRF), и доказательство безопасности может быть вычислено для заданных параметров [3]. Это устраняет необходимость в семействе хеш-функции, устойчивых к коллизиям, путем случайного прохода через функцию вместо простой итерации. Халсинг представил еще одну вариацию дополненной одноразовой подписи Винтерница, позволяющую создавать более мелкие подписи для обеспечения эквивалентной битовой безопасности посредством добавления побитовой маски эксклюзивной дизъюнкции (XOR) в итеративной функции цепочки. [6] Другое различие между дополненной одноразовой подписью Винтерница (вариант 2011) и исходной заключается в том, что парсинг сообщения происходит $log_2(w)$ бит за раз, а не w, уменьшая тем самым итерации хеш-функций, но увеличивая размер ключей и подписи.

Теперь коротко опишем дополненную одноразовую подпись Винтерница. С параметром безопасности $n \in N$, соответствующим длине сообщения (m), ключам и подписи в битах, определяется выбранной криптографической хеш-функцией и параметром Винтерница, $w \in N \mid W > 1$ (обычно {4, 16}), вычисляется l, которое является количеством n битовых строчных элементов в ключе или подписи W-OTS+, где $l = l_1 + l_2$:

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

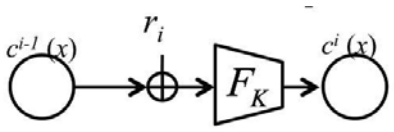
Используется ключевая хеш-функция, f_k: $\{0, 1\}^n \rightarrow \{0, 1\}^n \mid k \in \{0, 1\}^n$. В псевдокоде:

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

Где $Pad(x) = (x||10^{b|x|1})$ для $|x| < b$.

Функция цепочки, $c_k^i(x, r)$: при вводе $x \in \{0, 1\}^n$, счетчик итерации i, ключ, $k \in K$ и элементы рандомизации, $r = (r_1, ..., r_j) \in \{0,1\}^{n*j}$, с $j \geq i$, определяется следующим образом:

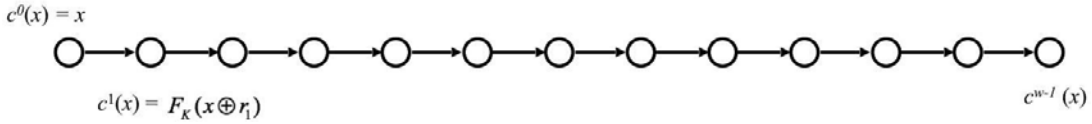
Рисунок 1. Функция цепочки W-OTS+



Где:

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

Рисунок 2. Пример генерации хеш-цепочки



Это побитовая эксклюзивная дизъюнкция предыдущей итерации c_k и элемента рандомизации, за которым следует f_k результата, который затем передается в следующую итерацию c_k.

5.3.1 Ключ подписи

Для создания закрытого ключа sk, $l + w - 1$ n битовых строк выбираются единообразно случайным образом (с помощью PRF), из которых первый l составляет секретный ключ, $sk = (sk_1,..sk_l)$ и остальные $w-1$ n битовых строк становятся $r = (r_1,..,r_{w-1})$. Функциональный ключ, k выбирается равномерно случайным образом.

5.3.2 Ключ подтверждения

Открытый ключ:

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

Обратите внимание, что pk_0 содержит r и k.

5.3.3 Подписание

Для выполнения подписи: сообщение M длины m анализируется таким образом, чтобы $M = (M_1, ..M_{l1})$, $M_i \in \{0, w - 1\}$ (создавая представление base-w для M).
Затем вычисляется контрольная сумма C, длины l_2, и добавляется:

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

Таким образом, что $M + C = b = (b_0,..b_l)$
Подписью является:

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

5.3.4 Верификация

Для проверки того, что подпись $b = (b_1,.., b_l)$ воссоздается из M.
Если $pk = \left(c_k^{w-1-b_i}(s_1), .., c_k^{w-1-b_l}(s_l)\right)$ то подпись действительна.

W-OTS+ обеспечивает уровень безопасности не менее $n - w - 1 - 2log(lw)$ бит [3]. Типичная подпись, где $w = 16$, использующая SHA-256 ($n = m = 256$), равна ln бит или 2,1 кб.

6 Схемы подписи дерева Меркла (MSS)

В то время как одноразовые подписи обеспечивают удовлетворительную криптографическую безопасность для подписания и проверки транзакций, для них характерен существенный недостаток - их можно использовать безопасно только один раз. Если адрес блока транзакций базируется на некотором преобразовании открытого ключа единственной пары ключей одноразовой подписи, это приводит к чрезвычайно ограничительному по функциональности блоку цепочки транзакций, в этом случае все средства с отправляющего адреса должны будут перемещаться с каждой выполненной транзакцией или эти средства окажутся под угрозой кражи.

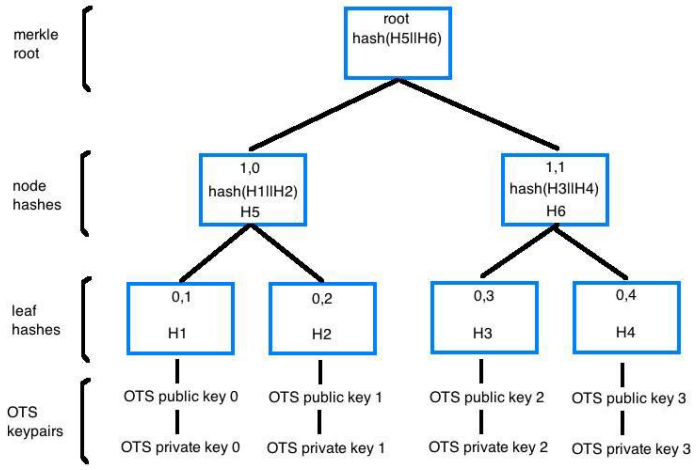
Решение заключается в расширении схемы подписи для включения более чем одной действительной одноразовой подписи для каждого адреса блока транзакций, что позволит сформировать предварительно столько подписей, сколько будет пар ключей одноразовых подписей. Логичным путем достижения этого является построение двоичного хеш-дерева, известного как дерево Меркла.

6.1 Двоичное дерево хешей

Общая идея дерева Меркла - это инвертированное дерево, состоящее из родительских узлов, вычисленное путем хеширования конкатенации дочерних узлов по слоям к корню. Существование любого узла или листа может быть криптографически доказано путем вычисления корня.

Дерево Меркла образовывается из n базовых листьев и имеет высоту до корня Меркла, h ($n = 2^h$) - начиная с хешей листьев (нулевой слой) и просчитывается вверх по каждому слою узлов. Каждый листовой узел в нашем гипотетическом использовании блока цепочки транзакций создается хешированием случайно сгенерированного открытого ключа одноразовой подписи. На дереве ниже видно, что узел над каждой парой хешей листьев сам формируется путем хеширования конкатенации дочерних хешей.

Рисунок 3. Пример схемы подписи дерева Меркла



Этот процесс продолжается вверх по слоям дерева до слияния с корневым хешем дерева, известным как корень Меркла.

В примере дерева на диаграмме возьмем корень Меркла в качестве открытого ключа, тогда можно использовать четыре предварительно вычисленных ключа одноразовой подписи для создания четырех криптографически безопасных действительных одноразовых подписей. Корень Меркла двоичного дерева хешей можно преобразовать в адрес блока транзакций (возможно, путем итеративного хеширования с добавленной контрольной суммой). Полная подпись S сообщения M для данной пары ключей одноразовой подписи включает в себя: подпись, s, номер ключа одноразовой подписи, n и путь аутентификации Меркла, например, для пары ключей одноразовой подписи 0 (то есть $n = 0$):

$$S = s,\ n,\ открытый\ ключ\ одноразовой\ подписи\ 0,\ H1,\ H2,\ H5,\ H6,\ корень$$

Учитывая, что открытый ключ одноразовой подписи и хеш листа могут быть выведены из s, а родительские узлы могут быть вычислены из их детей, на самом деле это может быть сокращено до:

$$S = s,\ n,\ H2,\ H6,\ корень$$

Если S является действительным путем проверки открытого ключа одноразовой подписи из s и M, тогда проверка хешей пути аутентификации Меркла воссоздает соответствующий корень Меркла (открытый ключ).

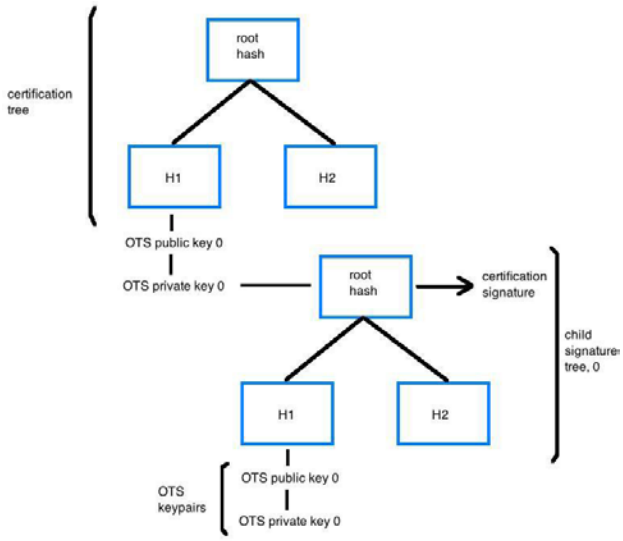
6.2 Состояние

Использование вышеуказанной схемы подписи Меркла (MSS) безопасно основывается на неиспользовании повторно ключей одноразовой подписи. Таким образом, это зависит только от состояния подписей или записей о подписанных транзакциях. Как правило, в реальном мире это потенциально может быть проблемой, но неизменяемый публичный блок цепочки транзакций является идеальным хранилищем для криптографической схемы подписи с учетом состояния. В 2015 году сообщалось о новой схеме криптографической подписи на основе хешей под названием SPHINCS, которая предлагает практически не зависящие от состояния подписи с 2^{128}-битной защитой [2].

6.3 Гипердеревья

Проблема с базовой MSS заключается в том, что количество доступных подписей ограничено, и все пары ключей одноразовых подписей должны быть предварительно сгенерированы до вычисления дерева Меркла. Генерация ключей и время подписания растут экспоненциально относительно высоты дерева, *h*, что означает, что деревья, превышающие 256 ключей одноразовой подписи, становятся затратными по параметрам времени и вычислительной мощности, необходимых для генерации. Стратегия отсрочки вычислений при генерации ключей и деревьев, а также расширение количества доступных пар ключей одноразовой подписи заключается в использование дерева, которое само состоит из деревьев Меркла, называемого гипердеревом. Общая идея состоит в том, что корень дочернего дерева Меркла подписывается ключом одноразовой подписи из хеша листа родительского дерева Меркла, известного как дерево сертификации.

Рисунок 4. Соединение деревьев Меркла



В самой простой форме (высота, $h = 2$) дерево сертификации предварительно вычисляется с помощью 2^1 пары ключей одноразовой подписи, и в момент, когда требуется первая подпись, новое дерево подписи Меркла (дерево подписи 0) вычисляется и подписывается одной из пар ключей одноразовой подписи дерева сертификатов. Дерево подписи состоит из *n* хешей листьев с соответствующими ключами одноразовой подписи, и они служат для подписи сообщений по мере необходимости. Когда каждая пара ключей одноразовой подписи в дереве подписи использована, следующее дерево подписи (дерево подписи 1) подписывается второй парой ключей одноразовой подписи дерева сертификации, и становится доступной следующая партия подписей.

Подпись *S* такой конструкции гипердерева становится несколько более сложной и состоит:
1. из дерева подписи: *s*, *n*, *путь Меркла*, *корень;*
2. из каждого дерева сертификации: *s* (корня дочернего дерева Меркла), *n*, *путь Меркла*, *корень*.

Теоретически можно складывать слои деревьев из дерева сертификации, чтобы бесконечно расширять исходную MSS. Размер подписей растет линейно для каждого дополнительного дерева, которое подписывается, в то время как объём подписей гипердерева увеличивается экспоненциально.

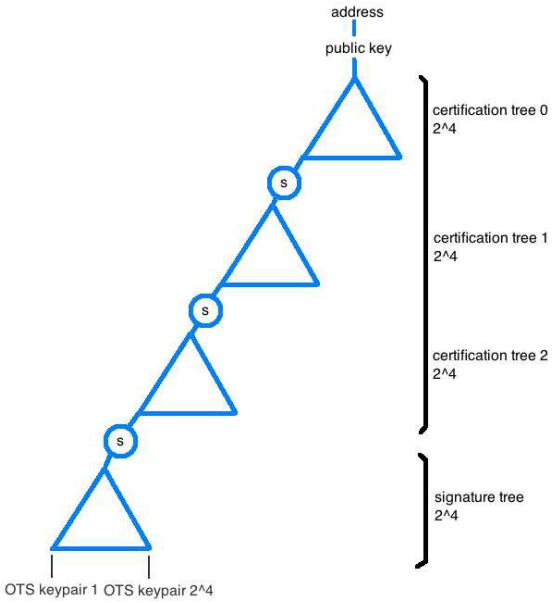
6.3.1 Примеры гипердеревьев

То Чтобы продемонстрировать, как легко MSS может быть расширена с помощью конструкции гипердерева, рассмотрим начальное дерево сертификации с высотой, $h_1 = 5$, с 2^5 листовыми хешами и связанными парами ключей одноразовой подписи. Корень Меркла этого дерева преобразуется для создания адреса блока транзакций. Возьмем для примера другое дерево Меркла, дерево подписи идентичного размера ($h_2 = 5$, то есть 2^5 листов и пар ключей одноразовой подписи). Существование 32 подписей возможно до того момента, как будет создано новое дерево подписей. Общее количество доступных подписей - $2^{h_1+h_2}$, которое в данном случае равно $2^{10} = 1024$.

Создание на Macbook Pro (2,7 ГГц i5, 8 гигабайт оперативной памяти), пар ключей одноразовой подписи и дерева сертификации Меркла разных размеров дало следующие результаты (неоптимизированный код на Python, одноразовая подпись Винтерница): 2^4 = 0,5 с, 2^5 = 1,2 с, 2^6 = 3,5 с, 2^8 = 15,5 с. Создание гипердерева, состоящего из начальной генерации двух 2^4 деревьев, занимает около 1 с по сравнению с 15,5 с, требующимися для генерации стандартного 2^8 дерева MSS для одного и того же объема подписей.

Увеличение глубины (или высоты) гипердерева продолжает эту тенденцию. Гипердерево, состоящее из четырех соединенных 2^4 деревьев сертификации и дерева подписи размером 2^4, может содержать 2^{20} = 1 048 576 подписей с увеличенным размером подписи, но при этом время создания составляет всего 2,5 с.

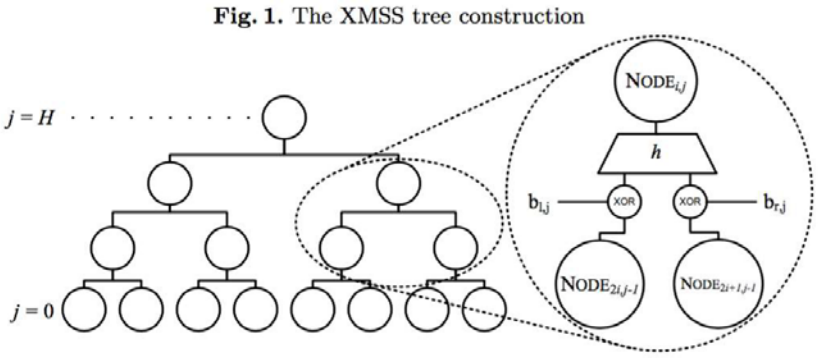
Рисунок 5. Создание гипердерева



Нет необходимости, чтобы гипердерево было симметричным, и поэтому, если оно состояло первоначально из двух деревьев, оно может быть расширено впоследствии путем присоединения дополнительных слоев деревьев. Таким образом, подписи блока транзакций будут изначально небольшого размера, который будет постепенно возрастать по мере увеличения глубины гипердерева. Использование гипердерева Меркла для создания и подписи адреса блока транзакций вряд ли потребуется для количества транзакций превышающего 2^{12}. Таким образом, возможность создать с вычислительной легкостью 2^{20} защищенных подписей для глубины гипердерева $h = 5$ является более чем достаточной.

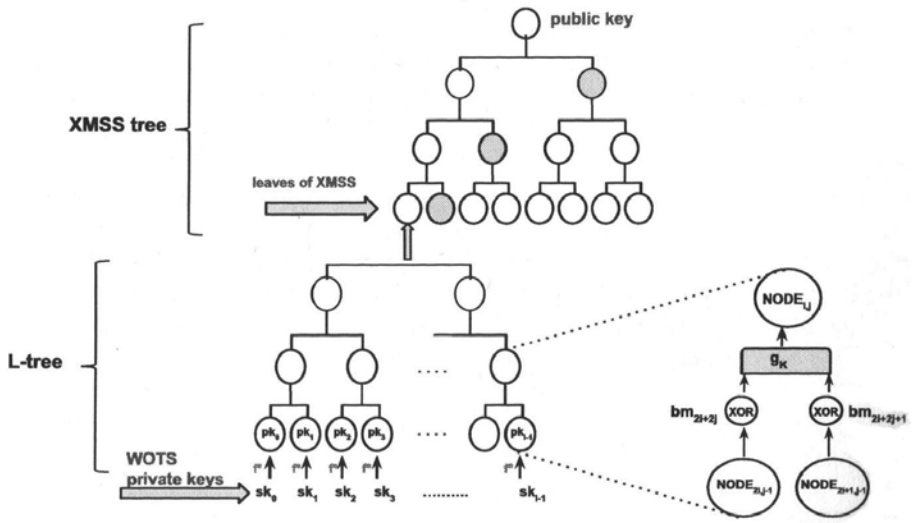
6.4 Расширенная схема подписи Меркла (XMSS)

Расширенная схема подписи Меркла (XMSS) была впервые обнародована Бахманом и группой соавторов в 2011 году и опубликована в качестве проекта IETF в прошлом году [4] [7]. Она доказуемо защищена в дальнейшем, и ее совершенно невозможно подделать при адаптивных избирательных атаках сообщений при соблюдении минимальных требований к безопасности: использование псевдослучайной функции (PRF) и дополнительной хеш-функции, обеспечивающей устойчивость к обнаружению второго прообраза. Эта схема позволяет расширять одноразовые подписи с помощью дерева Меркла, при этом существенное отличие заключается в использовании побитовой эксклюзивной дизъюнкции дочерних узлов до конкатенации хешей в родительский узел. Использование побитовой маски эксклюзивной дизъюнкции позволяет заменить семейство хеш-функций, устойчивых к коллизиям.



Листья дерева также не являются хешами для пар ключей одноразовой подписи, эту функцию выполняет корень дочерних L-деревьев, которые содержат открытые ключи одноразовой подписи, с l частями, образующими базовые листья. Дополненная одноразовая подпись Винтерница используется для одноразовых подписей (хотя впервые схема была описана с использованием вариации 2011 года).

Рисунок 7. Конструкция XMSS [8]



Длина бита открытого ключа XMSS равна $(2(H+\lceil logl \rceil+1)n$, подпись XMSS имеет длину $(l+H)n$, а длина секретного ключа подписи XMSS $<2n$.

Бахман сообщает о производительности компьютера с Intel (R) i5 2.5 Ghz для дерева высоты XMSS, h = 20, где w = 16, а используемая криптографическая хеш-функция - SHA-256 (m = 256) - около миллиона подписей. При тех же параметрах и оборудовании подписание занимает 7 мс, проверка 0,52 мс и генерация ключа 466 секунд. Уровень безопасности, достигнутый с такими параметрами, составил 196 бит для открытого ключа размером 1,7 кб, закрытого ключа 280 бит и подписи 2.8 кб. XMSS - привлекательная схема, основным недостатком которой является чрезвычайно длительное время генерации ключа.

6.5 Производительность дерева XMSS

Используя неоптимизированную библиотеку python, созданную для QRL формирования тестового уровня дерева XMSS размером 4096 (h = 12) со всеми ключами и побитовыми масками, генерируемыми с помощью базирующейся на хеше псевдослучайной функции (PRF), заняло 32 с на аппаратном обеспечении, описанном выше (Macbook Pro, 2.7 ГГц i5, 8 гигабайт оперативной памяти). Это включало генерацию с помощью псевдослучайной функции (PRF) более 8000 побитовых масок и более 300 000 *sk* фрагментов. Более эффективный алгоритм обхода дерева Меркла и необходимость только выполнять только w-1 хешей для функции цепочки закрытых ключей для дополненной одноразовой подписи Винтерница, а не 2^{w-1} для исходной одноразовой подписи Винтерница, способствовали значительному улучшению производительности по сравнению с обычным MSS.

В этой конструкции был достигнут полный размер подписи около 5.75 кб (кодировка шестнадцатеричной строки 11.75 кб), которая включала в себя: пару ключей одноразовой подписи, подпись, маршрут авторизации XMSS, открытый ключ одноразовой подписи и открытый ключ дерева XMSS (включая первичный ключ псевдослучайной функции и корень дерева XMSS).

Для деревьев с различным объемом подписей, созданных с использованием псевдослучайной функции и заданным начальным состоянием (seed), были получены следующие результаты: (h = 9) 512 4.2 с, (h = 10) 1024 8.2 с, (h = 11) 2048 16.02 с.

7 Предлагаемая схема подписи

7.1 Требования безопасности

В дизайне QRL важно, чтобы криптографическая безопасность схемы подписи была защищена от классических и квантовых компьютерных атак как в наши дни, так и в будущие десятилетия. XMSS, использующий SHA-256, где w = 16, предлагает 196-битную защиту с прогнозируемой безопасностью против вычислительной атаки путем простого перебора до 2164 года [9].

7.2 Подписи QRL

Предлагается расширяемая состоятельная асимметричная схема подписи на основе гипердерева, состоящего из связанных деревьев XMSS. Такой подход дает двойную выгоду: он позволяет использовать утвержденную схему подписи и генерировать адреса блок цепочки транзакций с возможностью подписи, избегая длительной вычислительной задержки, наблюдаемой при создании гигантских конструкций XMSS. Также выбор в данной схеме падает на дополненную одноразовую подпись Винтерница на основе хэша, как по соображениям безопасности, так и производительности.

7.3 Конструкция гипердерева

7.3.1 Размеры ключей и подписи

По мере роста количества деревьев в рамках гипердерева размер ключей и подписи растут линейно, но объем подписей растет экспоненциально. Размеры для различных ключей и подписи, построенных на основе дерева XMSS (базируясь на описании 2011 года), где w = 16, m = 256, h - высота дерева, а SHA-256 выбрана в качестве криптографического алгоритма хеширования:

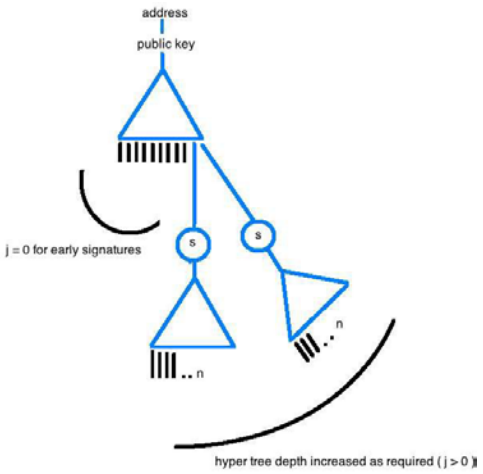
- $h = 2$, 2^2 подписей: открытый ключ 0.59 кб, подпись 2.12 кб (0.4 с)
- $h = 5$, 2^5 подписей: открытый ключ 0.78 кб, подпись 2.21 кб (0.6 с)
- $h = 12$, 2^{12} подписей: открытый ключ 1.23 кб, подпись 2.43 кб (32 с)
- $h = 20$, 2^{20} подписей: открытый ключ 1.7 кб, подпись 2.69 кб (466 с [3])

Компромисс для создания гипердерева XMSS (4 дерева, j = 3, h = 5) с возможным объемом в 2^{20} подписей, конструкция которого занимает менее чем 3 с по сравнению с 466, для подписи 8,84 кб вместо 2,69 кб может оказаться приемлемым.

7.3.2 Асимметричность

Создание асимметричного дерева позволяет получать ранние подписи с помощью единственной конструкции дерева XMSS, которая по мере необходимости будет расширяться для получения последующих подписей, соответственно изменяя общий объем доступных подписей. Обоснование основывается на том, что это, вероятно, не будет иметь никакого значения для приложения, работающего с блоком цепочки транзакций, и на уровне кошелька пользователю может предоставляться выбор возможности иметь больший объем подписей в зависимости от размеров подписи/ключа. Представляется, что максимальная глубина дерева j = 2 окажется достаточной для всех возможных обстоятельств.

Рисунок 8. Асимметричное дерево



7.4 Спецификация гипердерева QRL

Следующие параметры по умолчанию должны быть приняты для стандартной конструкции гипердерева:

- j = 0 ($j \in \{0 \leq x \leq 2\}$), h = 12 ($h \in \{1 \leq x \leq 14\}$), возможна верхняя граница количества подписей: 2^{36}, минимальный размер подписей: 2.21 кб, максимальный размер подписи: 7.65 кб.

Например, одно дерево XMSS, h = 12 с доступными 4096 подписями, которые могут быть расширены с помощью добавления последующих деревьев вплоть до h = 14 по мере необходимости. Для большинства пользователей дополнительные деревья вряд ли потребуются.

7.4.1 Пример подписи QRL

Предполагается, что для самой сложной конструкции гипердерева, где j = 2 и h = 14, m - подпись для сообщения транзакции, а n является позицией пары ключей одноразовой подписи для каждого дерева XMSS, потребуется:

- дерево подписи j = 2: одноразовая подпись для m, n, проверка аутентификации Меркла, корень дерева подписи Меркла
- дерево сертификации, j = 1: одноразовая подпись корня Меркла из дерева подписи (j = 2), n, проверка аутентификации Меркла, корень Меркла
- Исходное дерево XMSS, j = 0: одноразовая подпись корня Меркла (j = 1), n, проверка аутентификации Меркла, корень Меркла

Верификация включает в себя генерацию открытого ключа одноразовой подписи из m и самой подписи, а затем подтверждение того, что предоставленное доказательство проверки аутентификации Меркла генерирует корень дерева подписи Меркла. Это станет сообщением для следующей одноразовой подписи, и из этого будет сгенерирован следующий открытый ключ одноразовой подписи, предоставленное доказательство проверки аутентификации Меркла используется для воссоздания корня дерева сертификации Меркла, которое становится сообщением для следующей одноразовой подписи дерева сертификации и т.д. Подпись действительна только в том случае, если корень Меркла старшего дерева, исходного дерева XMSS (j = 0) правильно сгенерирован.

Обратите внимание, что открытые ключи одноразовой подписи не требуются для проверки подписи дерева XMSS. Фактически корень Меркла для каждого дерева также может быть выведен и поэтому пропускается во время проверки подписи гипердерева, если известен адрес блока транзакций (так как является вычисляемой производной от корня Меркла для наивысшего дерева сертификации XMSS (j = 0) в рамках подписи QRL - см. Аккаунты ниже).

Поскольку схема подписи является состоятельной, реализация кошелька должна сохранять и обновлять n для каждого дерева XMSS, сгенерированного в гипердереве для заданного адреса.

7.5 Псевдослучайная функция (PRF)

Для псевдослучайной функции начальное состояние (seed) генерируется с помощью HMAC DRBG.

7.6 Детерминистический кошелек

Используя один SEED, можно создать очень большое дерево XMSS, размеров которого будет достаточно для большинства пользователей в течение длительного периода. Защищенный источник энтропии используется для генерации этого SEED, он передается через защищенную функцию PRF для генерации набора псевдослучайных ключей, на основании которых генерируется дерево. Один из недостатков использования одного и того же дерева XMSS состоит в том, что пользователь ограничен одним адресом (хотя в то же время видимость открытого ключа не вызывает беспокойства в рамках MSS).

Адреса биткойнов или эфириума можно получить на основании связанного с ними открытого ключа, и, таким образом, отдельный закрытый или открытый ключ может создавать только один адрес. Тем не менее, адрес XMSS получается из открытого ключа PK, который содержит корневой ключ Меркла и открытый SEED. Если SEED остается постоянным, но количество ключей для одноразовой подписи для вычисления дерева

меняется, тогда корень Меркла будет изменяться для каждого варианта. Таким образом, для каждого отдельного сложения или вычитания одной пары ключей одноразовой подписи производный адрес будет изменяться.

Эта функция может использоваться программным обеспечением кошелька/узла для генерации многочисленных вариаций дерева XMSS (и его расширения/сокращения по мере необходимости с использованием того же самого начального SEED), что позволяет создавать столько уникальных адресов, сколько потребуется. Запись этой информации в безопасном, состоятельном и компактном виде будет вычислительно тривиальной.

8 Параметры дизайна криптовалюты

В оставшейся части данного информационного документа будут указаны предлагаемые параметры дизайна для блока цепочки транзакций QRL. Основное внимание уделяется тому, что он должен быть публичным блоком цепочки транзакций, который надежно защищен от классических и квантовых компьютерных атак. Это - первый проект документа, и, следовательно, любой аспект описания может потенциально претерпеть изменения в будущем.

8.1 Подтверждение доли

QRL должен быть открытым публичным блоком цепочки транзакций, обеспеченным алгоритмом подтверждения доли (proof-of-stake). Длина эпохи 10 000 блоков. Валидаторы доли определяются на основании долевых транзакций в предыдущую эпоху. Общая идея заключается в том, что каждый валидатор доли подписывает транзакцию, содержащую конечный хеш итеративной цепочки длиной 10 000 хешей (побитовая маска эксклюзивной дизъюнкции может применяться во время каждой итерации для уменьшения требований безопасности хеш-функции). При подтверждении долевой транзакции в цепочке каждый узел сети теперь может привязывать криптографический идентификатор адреса к хеш-цепочке для следующей эпохи.

8.1.1 Дизайн и случайность

For Для каждого блока каждый проверяющий узел доли текущей эпохи показывает хеш в цепочке непосредственно перед текущим, чтобы криптографически доказать участие и проголосовать за возможность быть выигравшим селектором блока.

HMAC DRBG используется для генерации последовательности псевдослучайных чисел из 32-байтовых результатов из данных начального состояния (seed), взятых из цепочки блоков (сначала блок генезиса, затем добавленная энтропия, взятая из конкатенированных последних хешей заголовков блока для каждой последующей эпохи).

Таким образом, каждый валидатор доли, выбранный для того, чтобы стать селектором блока, определяется по возможности продемонстрировать хеш ближайший по значению к результату псевдослучайной функции этого блока. Этим сложно манипулировать, поскольку псевдослучайная функция будет неизвестна в момент создания хеш-цепочки. Кроме того, итеративная (ключевая) хеш-цепочка представляет собой последовательность случайных чисел. Наконец, даже при существовании сговора между валидаторами, они не будут знать содержимое хеш-цепочки другого валидатора доли, поскольку оно еще не будет доступно.

Чтобы предотвратить атаку путем удержания блока, неспособность создать действительный блок после представление действительного хеша будет сопровождаться потерей всей награды блока по этому адресу, и он не будет участвовать в течение периода наказания.

Чтобы уменьшить атаку пустых блока от предсказываемых узлов или адресов валидаторов доли с низкими остатками на счете, используется гибкий порог для списка валидаторов доли. Награда за блок выплачивается взвешенным образом, основываясь на балансе адреса. С открытием хеш-сообщения каждый узел также разглашает хеш корня дерева Меркла отсортированного списка tx-хешей в пулах транзакций вместе с количеством транзакций, ожидающих блок. Каждый узел отрезает процент сверху и снизу, чтобы узнать, сколько транзакций ожидается в следующем блоке. Если блок пуст или имеет меньше ожидаемого значения доли, разрешается нескольким валидаторам заключать в дальнейшем договоренности (исключая валидаторов доли от бедных до богатых) на каждый блок. Если узлы селектора блоков ведут себя честно, то обратное истинно, и разрешенное число участвующих валидаторов доли растет. Средства не могут перемещаться в эпоху, которую они создают - это предотвращает попытки мошенничества с выбором блока с помощью предсказательного создания многочисленных адресов валидаторов доли.

8.2 Оплата

Большие размеры транзакций по сравнению с другими блоками цепочки транзакций требуют оплаты для каждой транзакции. Автор придерживается мнения, что рынки с искусственной оплатой (см. Биткойн) не нужны и противоречат идеалу открытого публичного блока цепочки транзакций. Каждая транзакция, если она сопровождается минимальной оплатой, должна быть такой же действительной, как и любая другая. Размер минимальной оплаты, приемлемой для майнеров, должны быть плавающим и устанавливаться рынком. То есть узлы/майнеры конкурентно устанавливают нижнюю границу оплаты между собой. Абсолютное минимальное значение будет соблюдаться на уровне протокола. Таким образом, майнеры будут заказывать транзакции из мемпула для включения в блок по своему усмотрению.

8.3 Блоки

8.3.1 Время нахождения блоков

Биткойн имеет время между блоками примерно 10 минут, но при естественной дисперсии это может привести к довольно длительным периодам до момента запуска следующего блока. Более новые схемы блока цепочки транзакций, такие как эфириум, улучшены в этом аспекте и выигравают засчет более короткого времени нахождения блока (15 секунд) без потери в безопасности или централизации майнеров из-за высокой скорости появления осиротевших/устаревших блоков. Эфириум использует модифицированную версию протокола Greedy Heaviest Observed Subtree, который позволяет включить блокированные/сиротские блоки в блок-цепочку и получать вознаграждение [13, 5].

Поскольку QRL планирует использовать алгоритм подтверждение доли с самого начала, мы ожидаем безопасного времени нахождения блоков от 15 до 30 секунд.

8.3.2 Вознаграждение за блок

Каждый новый созданный блок будет включать в себя первую транзакцию «coinbase», содержащую адрес майнинга, в который вознаграждение равно сумме вознаграждения за монетную ставку и суммарную сумму комиссий за транзакции внутри блока. Вознаграждение за блок взвешивается на основе баланса адреса валидатора доли, выбранного селектором блоков.

Вознаграждение за блок пересчитывается майнинг-узлом в каждом блоке и соответствует графику выпуска монет.

8.3.3 Размер блока

Чтобы избежать возможных споров было смоделировано готовое адаптивное решение, базирующееся на предложении Bitpay, которое использует для увеличения размера блока множетель *x* средней величины *y* последних *z* блоков [12]. Использование средней величины не позволяет майнерам манипулировать, включая либо пустые, либо переполненные блоки для изменения среднего размера блока. *x* и *z* будут тогда жесткими консенсусными правилами для сети, которым придется подчиняться.

Таким образом, максимальный размер блока *b* можно просто вычислить как:

$$b = xy$$

8.4 Валютная единица и наименования

QRL будет использовать денежный токен, *квант* (*кванты* во множественном числе), как базовую единицу валюты. Каждый *квант* делится на наименьшие элементы следующим образом:

- 1 : Шор
- 10^3 : Накамото
- 10^6 : Батерин
- 10^{10} : Меркл
- 10^{13} : Лампорт
- 10^{16} : Квант

Таким образом, каждая транзакция с участием части *кванта* на самом деле очень большое число единиц *Шора*. Плата за транзакцию оплачивается и рассчитывается в единицах *Шора*.

8.5 Аккаунты

Балансы пользователя хранится на аккаунтах. Каждый аккаунт является просто уникальным многократно используемым адресом блока цепочки транзакций, обозначенным строкой, начинающейся с «Q».

Адрес создается посредством выполнения SHA-256 по корню Меркла самого высокого дерева сертификации XMSS. К этому добавляется четырехбайтная контрольная сумма (образованная из первых четырех байтов двойного хеша SHA-256 корня Меркла) и буквы «Q». То есть в python псевдокоде это будет описано следующим образом:

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

Типичный адрес аккаунта:

$$Qcea29b1402248d53469e352de662923986f3a94cf0f51522bedd08fb5e64948af479$$

Каждый аккаунт имеет баланс, деноминированный в квантах, делимый вплоть до единственной единицы Шора. Адреса состоятельны с каждой транзакцией, использующей новую пару ключей одноразовой подписи и QRL, и сохраняют каждый открытый ключ, когда-либо используемый (это возможно сократить, поскольку может быть повторно сгенерировано «на лету» из подписи транзакции и сообщения, но такой подход будет операционно интенсивным) для каждого аккаунта. Счетчик транзакций, называемый nonce, будет увеличиваться с каждой транзакцией, отправленной с аккаунта. Это позволяет кошелькам, которые не хранят всю цепочку блоков, отслеживать их местоположение в схеме подписи гипердерева с сохранением состояния.

8.6 Выпуск монет

8.6.1 Исторические соображения

Биткойн был первой децентрализованной криптовалютой и первоначально экспериментальной, без ассоциированной денежной стоимости, поэтому было бы целесообразно распределять валюту целиком посредством майнинга. Совсем недавно Zcash выбрал сходный процесс с % от вознаграждения за добычу монеты в начале периода эмиссии, передаваемых в проект с открытым исходным кодом, что привело к невероятной волатильности цен.

Другие системы блоков цепочки транзакций, такие как эфириум, вместо этого продали большую часть конечной суммы монет в качестве части первоначального монетного предложения (ICO). В этом есть преимущество того, что ранние пользователи по-прежнему могут выигрывать, поддерживая развитие проекта, но, кроме этого, сам проект может генерировать средства для продолжения разработки и поддержки начальной нагрузки и развития проекта с самого начального периода становления. Подход ICO также позволяет рынку легко развиваться, поскольку инвесторам предлагается большое количество монет из блока генезиса для покупки и продажи.

Auroracoin (2014) придерживался другого подхода, предлагая всем в Исландии равную долю первоначального монетного предложения, в то время как разработчики оставили 50% всей суммы монет для себя.

Другие криптовалюты либо просто клонировали биткойн целиком, либо стартовали заново с новой цепочкой, но другой кодовой базой.

8.6.2 Трансфер баланса между цепочками

Возможно создать QRL на основе текущего состояния блока цепочки транзакций биткойна, интегрированного в блок генезиса QRL. Общая идея заключалась бы в том, чтобы позволить пользователям создавать транзакцию «импорта» для однократного использования, содержащую уникальное сообщение и подпись (то есть случайно созданный адрес кошелька QRL, подписанный закрытым ключом биткойна с адреса с балансом биткойна во время фиксации текущего состояния). Эта функция может оставаться активной до определенной высоты блока, а затем остальная часть монет будет добываться обычным способом. Первоначальное раздувание блока генезиса было бы остановлено при той же высоте блока. Недостатком подобного подхода является то, что, честно говоря, он наказывает владельцев других криптовалют, отличных от биткойна, и технически потенциально сложен для новых пользователей. Также техническая озабоченность может вызвана тем фактом, что будет возможно восстановить открытый ключ ECDSA только по подписи и сообщению. Таким образом будут постоянно раскрываться открытые ключи для адресов биткойнов, используемых в процессе, и было бы важно переводить средства впоследствии на новый случайно сгенерированный биткойн адрес, для смягчения этого фактора.

(? Можно ли допустить существование подобного функционала для держателей эфириума)

8.6.3 Предлагаемая эмиссия - проект

Первоначальная эмиссия QRL будет следующей:

- Первоначальное монетное предложение в 1 миллион квантов (4,7% конечной суммы монет) до запуска.
- Текущее состояние всех адресов биткойнов с балансами выше 0.01 btc используется для формирования первоначального блока генезиса QRL. Любой, кто хочет напрямую перевести монеты в соотношении 1:1 из блока цепочки транзакций

биткойнов в блок цепочки транзакций QRL, сможет сделать это до тех пор, пока не будет достигнута высота блока 518400 (3-6 месяцев) через кошелек узла.

- Еще 1 миллион квантов будет храниться в адресе блока генезиса для использования фондом.
- Оставшееся количество будет добываться майнингом (21 000 000 - (2 000 000 + btc, балансов импортированных до высоты блока 518400)).

8.7 График выпуска монет

Определяющей чертой биткойна является дефицит и фиксированный верхний предел для выдачи основного денежного токена. QRL последует практике биткойна в этом отношении, зафиксировав верхний предел выпуска монет на уровне $21x10^6$ *квантов*. Равномерное экспоненциальное снижение в ставке вознаграждения за блок будет предпочтительнее жесткого потолка для выпуска монет. Это устранит волатильность, связанную с явлением «ополовинивания» биткойна.

Общая сумма монет, $x = 21x10^6$, минус монеты, созданные в блоке генезиса, y, будет всегда экспоненциально постоянно снижаться с Z_0. Кривая затухания рассчитывается для распределения вознаграждений за майнинг в течение приблизительно 200 лет (до 2217AD, 420480000 блоков в 15 сек.) до того момента, пока только один *квант* не останется необнаруженным (хотя майнинг сможет продолжаться и после этого).

Оставшаяся сумма монет в блоке t, Z_t может быть рассчитана следующим образом:

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

Коэффициент λ вычисляется по формуле: $\lambda = \frac{lnZ_t}{t}$. Где t, - общее количество блоков в графике выпуска до последних *квантов*. До блока 518400, $\lambda = 3.98590885111x10^{-08}$. Вознаграждение за блок, b рассчитывается для каждого блока:

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

Между блоком генеза и блоком биткойна 518400 балансы могут быть перенесены в блок цепочки транзакций с помощью транзакций импорта. На этапе 518401 график выпуска переназначит блокировку новых импортированных балансов, уменьшая Z_t и соответствующим образом корректируя вознаграждение блока.

Ссылки

[1] http://oxt.me/charts.

[2] Д. Бернштейн. Сфинкс: практические подписи, основанные на хешах. 2015.

[3] Дж. Бахман. О безопасности однократной схемы подписи Винтерница.

[4] Дж. Бахман. Xmss - практическая схема прямой защищенной подписи, основанная на минимальных предположениях о безопасности. 2011.

[5] В. Бутерин. Технический документ Эфириума. 2013.

[6] А. Халсинг. W-ots+ - более короткие подписи для схем подписи на основе хеша. 2013.

[7] А. Халсинг. Xmss: расширенные хеш-подписи. 2015.

[8] А. Карина. Эффективная программная реализация схемы подписи на основе хеш-кода mss и ее вариантов. 2015.

[9] А. Ленстра. Выбор размеров криптографических ключей. 2001.

[10] Р. Меркл. Сертифицированная цифровая подпись. CRYPTO, 435, 1989.

[11] С. Накамото. Биткойн: электронная наличная система одноранговой сети. 2008.

[12] С. Пейр. Простой, адаптивный предел размерности блока. 2016.

[13] Йонатан Сомполински. Ускорение обработки биткойн транзакций быстрых денег, которые растут на деревьях, а не на цепочках. 2014.

[14] А. Тоши. Парадокс дня рождения. 2013.

Дополнение

Обновление, касающееся выпуска монет:

Ниже приведено обновление, предоставленное уже во время перевода технического документа и отсутствующее в исходном тексте. Обновление относится к общей сумме предпродажного выпуска, которая были изменена в сравнении с исходным документом. Это дополнение следует рассматривать как заменяющее разделы 8.6: «Выпуск монет» и 8.7: «График выпуска монет». Оно основывается на следующем сообщении Питера Уотерленда в блоге, которое можно найти по адресу: https://medium.com/the-quantum-resistant-ledger/seed-investment-strategy-pos-algorithm-updates-3b3854e83a4a.

Выпуск монет

Определяющей чертой биткойна является дефицит и фиксированный верхний предел для выдачи основного денежного токена. QRL последует практике биткойна в этом отношении, зафиксировав верхний предел выпуска монет на уровне $21x10^6$ *квантов*. Равномерное экспоненциальное снижение в ставке вознаграждения за блок будет предпочтительнее жесткого потолка для выпуска монет. Это устранит волатильность, связанную с явлением «ополовинивания» биткойна.

Первоначальная эмиссия QRL будет следующей:

- Первоначальное монетное предложение составит 65 миллионов монет (квантов).
  Окончательный выпуск составит 105 миллионов монет в течении 200 лет (сглаженная экспоненциальная кривая затухания с жестким верхним пределом).
- 20% предпродажи (13 миллионов квантов) будет проведено командой, из которых 65% будут начислены фонду QRL.
- 100% финансирования в предпродажный период будет проводиться фондом QRL в рамках поэтапной реализации программы, связанной с фазами развития и целями исследований.
- Адреса bitcoin и ether с мультиподписями будут приняты в виде депозитов предпродажного финансирования (2-из-3 и m-из-n мультиподписи с ключами, независимыми держателями которых будут 3 разных члена-основателя QRL).
- Предлагаемый запуск сети (ожидается с синхронным обменным списком) запланирован на 4-й квартал 2017 года.