RaiBlocks: Криптовалютная Сеть Без Комиссий

Colin LeMahieu
clemahieu@gmail.com

Аннотация—В последнее время при высоком спросе и ограниченной масштабируемости увеличилось среднее время транзакций и комиссий в популярных криптовалютах, что привело к неудовлетворительному опыту. Здесь мы познакомим с RaiBlocks, криптовалютой с новой архитектурой блочной структуры, где каждый аккаунт имеет свою собственную блок-цепочку, обеспечивая почти мгновенную скорость транзакции и неограниченную масштабируемость. У каждого пользователя есть своя блок-цепочка, позволяя им асинхронно обновляться его для остальной сети, что приводит к быстрым транзакциям с минимальными издержками. Транзакции отслеживают остаток на аккаунтах, а не суммы в транзакции, позволяя агрессивную обрезку базы данных без ущерба для безопасности. На сегодняшний день сеть RaiBlocks обработала более 4,2 млн транзакций имея полный объем базы чуть более 1,7 ГБ. Безкомиссионность RaiBlocks и транзакции в доли секунд делают его главной криптовалютой для потребительских транзакций.

Index Terms—криптовалюта, блокчейн, raiblocks, распределенная база, цифровой, транзакции

I. ВСТУПЛЕНИЕ

С МОМЕНТА появления Bitcoin в 2009 году наблюдается растущий уход от традиционных, поддерживаемых правительством валют и финансовых систем, к современным системам платежей, основанных на криптографии, которые предлагают возможность хранить и переводить средства надежным и безопасным способом [1]. Для эффетивного функционирования, валюта должна быть легко передаваемой, неотменяемой и иметь некоторые ограничения или вовсе без комиссий. Увеличенное время транзакций, большие комиссии и сомнительная масштабируемость сети вызвали вопросы о практичности Bitcoin как повседневной валюты.

В этой статье мы знакомим с RaiBlocks - криптовалютой с малой задержкой, основанной на инновационной блочной структуре данных, предлагающей неограниченную масштабируемость и отсутствие транзакционных комиссий. RaiBlocks разработан как простой протокол и с исключительной целью быть высокопроизводительной криптовалютой. Протокол RaiBlocks может работать на маломощном оборудование, позволяя быть практичной и децентрализованной криптовалютой для повседневного использования.

Статистика криптовалюты, представленная в настоящем документе, является точной на дату публикации.

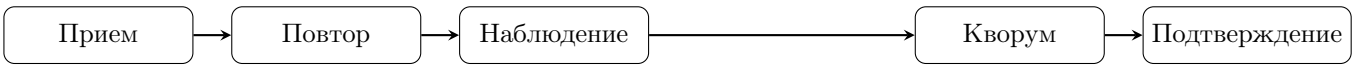
II. СПРАВОЧНАЯ ИНФОРМАЦИЯ

В 2008 году аноним под псевдонимом Satoshi Nakamoto опубликовал белую бумагу, описывающую первую в мире децентрализованную криптовалюту Bitcoin [1]. Ключевым нововведением Bitcoin стал блокчейн, публичная, неизменяемая и децентрализованная структура данных, которая используется в качестве регистра операций с валютой. К сожалению, по мере того как Биткойн развивался, некоторые проблемы в протоколе сделали Bitcoin недоступным для многих приложений:

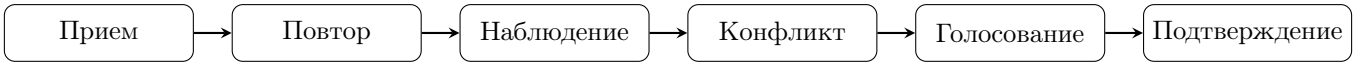
1) Плохая масштабируемость: Каждый блок блокчейна может хранить ограниченное количество данных, что означает, что система может обрабатывать только столько транзакций в секунду, сколько хватает места в блоке для записи транзакций. В настоящее время средний сбор за транзакцию составляет $10,38 [2].
2) Высокая задержка: среднее время подтверждения составляет 164 минуты [3].
3) Энерго неэффективна: Сеть Биткоин потребляет около 27.28 млрд. кВт / ч в год, используя в среднем 260KWh на транзакцию [4].

Биткоин и другие криптовалюты функционируют путем достижения консенсуса в отношении своего глобального блокчейна, с тем чтобы проверить законность операции, сопротивляясь злоумышленникам. Биткоин достигает консенсуса с помощью экономической меры, называемый доказательством работы (PoW). В системе PoW участники конкурируют за вычисление числа, называемого nonce, так что хэш всего блока находится в целевом диапазоне. Этот допустимый диапазон обратно пропорционален совокупной вычислительной мощности всей сети Bitcoin для поддержания согласованного среднего времени, затрачиваемого на поиск допустимого значения nonce. Тогда нашедшей действительный номер nonce может добавить блок в блокчейн; поэтому те, кто обладает огромным вычислительным ресурсом для вычисления nonce, играют большую роль в состоянии блокчейна. PoW обеспечивает устойчивость к атаке Sybil, где атакующий ведет себя как несколько объектов, чтобы получить дополнительную мощность в децентрализованной системе, снижает уровень состояния гонки, которое по своей природе существуют при доступе к глобальной структуре данных.

Альтернативный консенсусный протокол это подтверждающий долю участия (PoS), впервые был введен Peercoin в 2012 году [5].В системе PoS участники голосуют с весом, эквивалентным количеству средств, которое они имеют в данной криптовалюте. С помощью этого механизма те, кто имеет большие финансовые инвестиции, получают больше влияния и по своей сути стимулируются поддерживать честность системы



(a) Если конфликт не обнаружен, то никаких дополнительных деействиий не требуется.



(b) В случае конфликтующей транзакции ноды голосуют за допустимую транзакцию.

Рис. 1. RaiBlocks не требует дополнительных затрат для типичных транзакций. В случае конфликта транзакций ноды должны голосовать за допустимые транзакции

или риск потерять свои инвестиции. PoS избавляет от расточительной конкуренции мощности вычислений, требуя только легкого программного обеспечения, работающего на малой мощности.

Оригинальная бумага RaiBlocks и первая бета-реализация были опубликованы в декабре 2014 года, что делает ее одним из первых криптовалют использующию Направленный Ациклический Граф (DAG) [6]. Вскоре начали развиваться другие криптовалюты DAG, в частности DagCoin/Byteball и IOTA [7], [8]. Эти криптовалюты на основе DAG сломали форму блокчейна, увеличив производительность системы и безопасность. Byteball достигают консенсуса, опираясь на «главную цепочку» состоящия из честных, авторитетных и пользователей-надежных «свидетелей», в то время как IOTA достигает консенсуса по PoW сложенным транзакциям. RaiBlocks достигает консенсуса посредством взвешенного голосования по конфликтующим транзакциям. Эта система консенсуса обеспечивает более быстрые и более детерминированные транзакции, сохраняя при этом сильную децентрализованную систему. RaiBlocks продолжает эту разработку и позиционирует себя как одну из самых высокопроизводительных криптовалют.

III. КОМПОНЕНТЫ RAIBLOCKS

Прежде чем описывать общую архитектуру системы RaiBlocks, мы определим отдельные ее компоненты.

A. Аккаунт

Аккаунт - это часть публичного ключа от пары ключей цифровой подписи. Публичный ключ, также именуемый адресом, передается другим участникам сети, в то время как закрытый ключ хранится в секрете. Пакет данных с цифровой подписью гарантирует, что содержимое было одобрено владельцем закрытого ключа. Один пользователь может управлять многими аккаунтами, но для каждого аккаунта может существовать только один публичный адрес.

B. Блок/Транзакции

Термин «блок» и «транзакция» часто используются взаимозаменяемо, когда блок содержит одну транзакцию. Транзакция конкретно относится к действию, тогда как блок относится к цифровому кодированию транзакции. Транзакции подписываются закрытым ключом, принадлежащим аккаунту, на котором выполняется транзакция.

C. Ledger

Ledger - это глобальный набор аккаунтов, где каждый аккаунт имеет свою собственную цепочку транзакций (рис 2). Это ключевой компонент дизайна, который подпадает под категорию замены соглашения о времени выполнения с соглашением времени разработки; каждый соглашается с помощью проверки подписи, что только владелец аккаунта может изменить свою собственную цепочку. Это преобразует, судя по виду общую структуру данных, распределенный ledger, в набор не общих (частных) структур пользования.

D. Нода

Нода представляет собой часть программного обеспечения, работающего на компьютере, который соответствует протоколу RaiBlocks и участвует в сети RaiBlocks. Программное обеспечение управляет ledger и любыми учетными записями, которыми может управлять нода, если таковые имеется. Нода может либо хранить весь ledger, либо ее обрезанную историю,

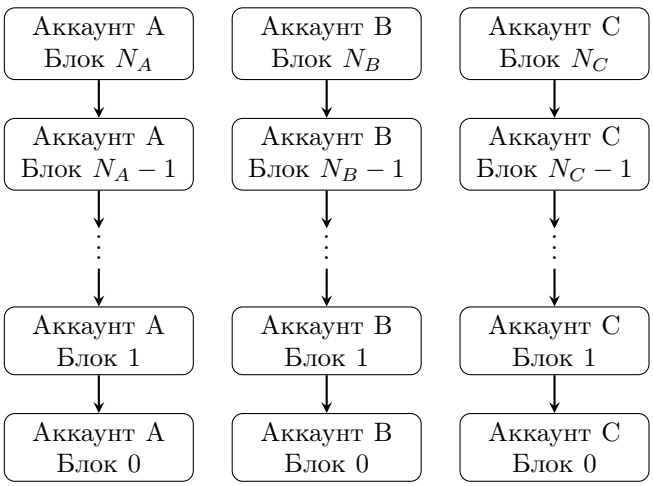


Рис. 2. Каждый аккаунт имеет свой собственный блокчейн, содержащий историю изменения баланса. Блок 0 должен быть открывающий транзакцией (Секц IV-B)

содержащую только последние несколько блоков цепочки каждого аккаунта. При настройке новой ноды рекомендуется проверять всю историю и делать срез локально.

IV. ОБЗОР СИСТЕМЫ

В отличие от блокчейнов используемых во многих других криптовалютах, RaiBlocks использует блок-решетчатую структуру. Каждый аккаунт имеет свой собственный блокчейн (аккаунт- цепочка), эквивалент истории транзакций/баланса аккаунта (Рис. 2). Каждая цепочка аккаунта может быть обновлена только владельцем аккаунта; это позволяет каждой цепочке аккаунта быть обновленной незамедлительно и асинхронно к остальной блок-решетке, что приводит к быстрым транзакциям. Протокол RaiBlocks является чрезвычайно легким; каждая операция вписывается в требуемый минимальный размер пакета udp для передачи через интернет. Требования к оборудованию для нод также минимальны, так как ноды должны только записывать и ретранслировать блоки для большинства транзакций (Рис 1).

Система инициируется с аккаунтом генезиса, содержащего баланс генезиса. Баланс генезиса является фиксированным количеством и никогда не может быть увеличен. Баланс генезиса делится и отправляется на другие аккаунты через транзакции отправки зарегистрированные в цепочке аккаунта генезиса. Сумма остатков на всех счетах никогда не превысит первоначальный баланс генезиса, что дает системе верхнюю границу по количеству и не позволяет увеличить его.

В этом разделе описывается создание и распространение различных типов транзакций в сети.

A. Транзакции

Перевод средств с одного аккаунта на другоий требует двух операций: отправки списания средств с баланса отправителя и приема добавления средств на аккаунт получателя (рис. 3). Перенос сумм в виде отдельных операций на аккаунтах отправителя и получателя служит для нескольких важных целей:

1) Последовательность входящих передач, которые по своей сути асинхронны.
2) Сохранить транзакцию небольшой чтобы поместиться в udp-пакеты.
3) Облегчать ledger обрезку путем минимизации вывода данных.
4) Изоляция принятых транзакций от еще не принятых.

Более одного аккаунта, отправляющего на один и тот же является асинхронной операцией, задержка в сети и отправляющие аккаунты не обязательно находятся в связи друг с другом, это означает, что нет универсального приемлемого способа узнать, какая транзакция произошла в первую очередь. Поскольку добавление ассоциативно, порядок последовательных вводимых данных не имеет значения, и поэтому нам

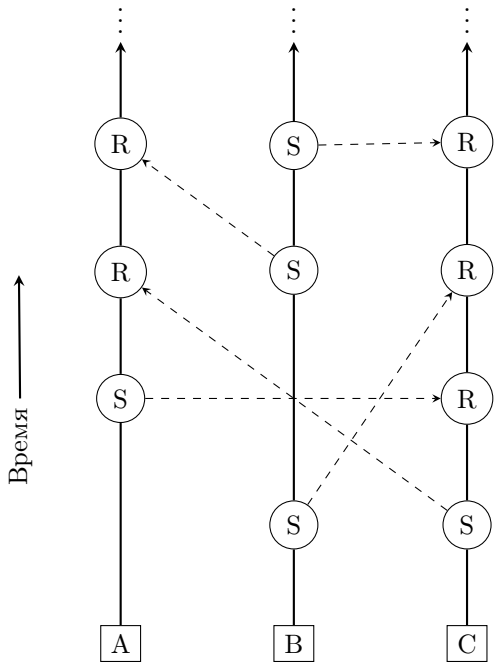


Рис. 3. Визуализация блок-решетки. Каждый перевод средств требует блока отправки (S) и блока получения (R), каждый подписанный владельцем аккаунта-цепочки (A,B,C)

просто нужно глобальное соглашение. Это ключевой компонент дизайна, который преобразует соглашение о времени выполнения в соглашение о времени разработки. Принимающий аккаунт имеет контроль над тем, чтобы решить, какой перевод пришел первым и тем самым установить своий порядок входящих блоков.

Если аккаунт хочет сделать большой перевод, который был получен как набор небольших переводов, мы хотим представить это таким образом, который вписывается в пакет UDP. Когда получающий аккаунт устанавливает последовательность входящих переводов, он сохраняет общую сумму своего баланса счета, так что в любое время он имеет возможность перевести любую сумму с фиксированным размером транзакции. Это отличается от модели транзакции ввода/вывода используемой Bitcoin и другими криптовалютами.

Некоторые ноды не заинтересованы в расходовании ресурсов на хранение полной истории транзакций аккаунта; они заинтересованы только в текущем балансе каждого аккаунта. Когда аккаунт совершает транзакцию, он кодирует накопленный баланс, и эти ноды должны отслеживать только последний блок, который позволяет им отбрасывать исторические данные, сохраняя при этом правильность.

Даже с акцентом на соглашения о времени разработки, есть окно задержки при проверке транзакций из-за выявления и обработки плохих участников в сети. Так как соглашения в RaiBlocks достигаются быстро, от миллисекунд до секунд, мы можем представить пользователю две знакомые категории входящих транзакций: принятые и не принятые. Принятые транзакции - это транзакции, в которых аккаунт создал блок полу-

чения. Не принятые транзакции еще не были включены в совокупный баланс получателя. Это замена более сложной и непривычной метрики подтверждения в других криптовалютах.

B. Создание Аккаунта

Чтобы создать аккаунт, Вам необходимо создать Open транзакцию (Рис. 4). Open транзакция всегда является первой транзакцией каждой цепочки аккаунтов и может быть создана при первом поступлении средств. Поле account хранит открытый ключ (адрес), производный от закрытого ключа, который используется для подписи. Поле source содержит хэш транзакции, которая направила средства. При создании аккаунта, представитель должен быть выбран для голосования от вашего имени, он может быть изменен позже (раздел IV-F). Аккаунт может объявить себя своим представителем.

```
open {
    account: DC04354B1...AE8FA2661B2,
    source: DC1E2B3F7C...182A0E26B4A,
    representative: xrb_1anr...posrs,
    work: 0000000000000000,
    type: open,
    signature: 83B0...006433265C7B204
}
```

Рис. 4. Пример open транзакции

C. Баланс Аккаунта

Остаток на счете записывается в ledger. Вместо того, чтобы записывать сумму транзакции, подтверждение (раздел IV-I) требует проверки разницы между балансом в блоке отправки и балансом предыдущего блока. Получающий аккаунт может затем увеличить предыдущий остаток, измеренный в конечном остатке, указанном в новом блоке приема. Это делается для повышения скорости обработки при загрузке больших объемов блоков. При запросе истории счета, суммы уже даны.

D. Отправка с Аккаунта

Чтобы отправить с адреса, на адресе должен быть уже открытый блок, а также и баланс (Рис. 5). В previous поле содержится хэш предыдущего блока в цепочке аккаунта. Поле destination содержит адрес для которого отправляются средства. Блок отправки является неизменяемым после подтверждения. После передачи в сеть средства немедленно вычитаются из баланса счета отправителя и находятся в статусе pending для получателя, пока он не подпишет блок чтобы принять эти средства. Отложенные (Pending) средства не следует считать ожидающими подтверждения, поскольку они уже потрачены с аккаунта отправителя и отправитель не может аннулировать эту транзакцию.

```
send {
    previous: 1967EA355...F2F3E5BF801,
    balance: 010a8044a0...1d49289d88c,
    destination: xrb_3w...m37goeuufdp,
    work: 0000000000000000,
    type: send,
    signature: 83B0...006433265C7B204
}
```

Рис. 5. Пример send транзакции

E. Получение Транзакции

Для завершения транзакции получатель отправленных средств должен создать блок приема на собственной аккаунт-цепочке (Рис. 6). Поле source ссылается на хэш транзакции отправки. Как только блок создан и транслирован в сеть, баланс аккаунта обновляется и средства официально зачисляются на счет получателя.

```
receive {
    previous: DC04354B1...AE8FA2661B2,
    source: DC1E2B3F7C6...182A0E26B4A,
    work: 0000000000000000,
    type: receive,
    signature: 83B0...006433265C7B204
}
```

Рис. 6. Пример receive транзакции

F. Назначение представителя (Representative)

Владельцы аккаунтов, имеют возможность выбрать представителя для голосования от своего имени. Это являются мощным инструментом децентрализации, которая не имеет сильного аналога в протоколе Proof of Work или Proof of Stake. В обычных системах PoS у владельца аккаунта должна быть запущена нода для участия в голосовании. Постоянный запуск ноды нецелесообразен для многих пользователей; передача представителю права голоса от имени аккаунта, что ослабляет это требование. Владельцы аккаунтов имеют возможность переназначить представителя аккаунта в любое время. Транзакция change изменяет представителя аккаунта, вычитая вес голосования от старого представителя и добавив вес для нового представителя (рис. 7). Денежные средства в данной операции не перемещаются, и представитель не имеет возможности тратить средства на счет.

G. Форк и Голосование

Форк возникает, когда j подписал блоки $b_1, b_2, \ldots, b_j$ и утвердил те же блоки, что и их предшественник (рис. 8). Эти блоки вызывают противоречивое представление о состоянии аккаунта и должны быть устранены. Только владелец аккаунта имеет возможность

```
change {
    previous: DC04354B1...AE8FA2661B2,
    representative: xrb_1anrz...posrs,
    work: 0000000000000000,
    type: change,
    signature: 83B0...006433265C7B204
}
```

Рис. 7. Пример change транзакции о смене представителя

подписывать блоки в свою цепочку аккаунтов, поэтому форк должен быть результатом плохого программирования или злонамеренного намерения (двоийного расходования) владельцем аккаунта.

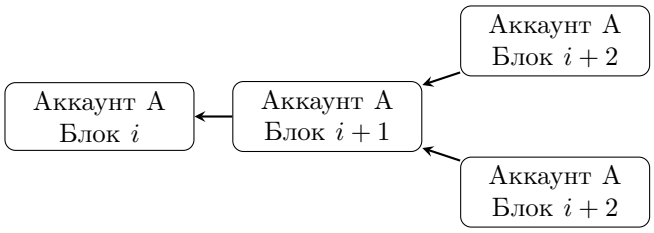


Рис. 8. Форк возникает, когда два (или более) подписанных блока ссылаются на тот же предыдущий блок. Старые блоки слева, а новые блоки справа

После обнаружения представитель создаст голосование, ссылающееся на блок $\hat{b}_i$ в его ledger и передаст его в сеть. Вес голосования ноды, w_i, является суммой балансов всех счетов, которые назвали его своим представителем. Нода будет наблюдать входящие голоса от других M онлайн представителей и хранить накопительное соотвествие на 4 периода голосования, 1 минута на все, и подтвердит победивший блок (уравнение 1).

$$v(b_j) = \sum_{i=1}^{M} w_i \mathbb{1}_{\hat{b}_i = b_j} \tag{1}$$

$$b^* = \arg\max_{b_j} v(b_j) \tag{2}$$

Самый популярный блок b^* будет иметь большинство голосов и будет сохранен в ledger ноды (уравнение 2). Блок(и), что потеряли голоса, удаляются. Если представитель заменит блок в своей ledger, он создаст новое голосование с более высоким порядковым номером и передаст новое голосование в сеть. Это единственный сценарий, когда голосуют представители.

В некоторых случаях краткие проблемы подключения к сети могут привести к тому, что передаваемый блок не будет принят всеми пирами (peers). Любой последующий блок на этом аккаунте будет проигнорирован как недопустимый пирами, которые не видели начальную передачу. Повторная трансляция этого блока будет принята остальными пирами и последующие блоки будут извлечены автоматически. Даже при возникновении форка или отсутствующего блока затрагиваются только аккаунты, указанные в транзакции; остальная часть сети продолжает обработку транзакций для всех других аккаунтов.

H. Proof of Work

Все четыре типа транзакций имеют рабочее поле, которое должно быть правильно заполнено. Поле work позволяет создателю транзакции вычислить nonce такое, что хэш-код nonce объединялся с предыдущем полем в receive/send/change транзакциях или в поле аккаунта в открывающей транзакции, ниже определенного порогового значения. В отличие от Биткоин, PoW в RaiBlocks используется как анти-спам инструмент, аналогичная система Hashcash, и может быть вычислен за несколько секунд [9]. После отправки транзакции PoW для последующего блока можно предварительно вычислить, так как известно предыдущее поле блока. Это делает транзакции мгновенными для конечного пользователя до тех пор, пока время между транзакциями превышает время, необходимое для вычисления PoW.

I. Проверка Транзакций

Чтобы блок считался допустимым, он должен иметь следующие атрибуты:

1) Блок уже не должен быть в ledger (повторяющиеся операции).
2) Должен быть подписан владельцем аккаунта.
3) Предыдущий блок является главным в цепочке аккаунта. Если он существует, но не главный, то это форк.
4) Аккаунт должен иметь блок открытия.
5) Вычисляемый хэш соответствует пороговому значению PoW.

Если это блок получения, проверьте, ожидает ли хэш исходного блока, то есть он еще не был погашен. Если это блок отправки, текущий баланс должно быть меньше предыдущего значения баланса.

V. ВЕКТОРЫ АТАК

RaiBlocks, как и все децентрализованных криптовалюты, могут подвергнуться нападению со стороны злоумышленников в попытке получения финансовой выгоды или разрушения системы. В этом разделе мы описываем несколько возможных сценариев атаки, последствия таких атак, и какие превентивные меры принимает протокол RaiBlock.

A. Синхронизации Блоков

В Разделе IV-G мы обсудили сценарий, когда блок может не передаваться должным образом, в результате чего сеть игнорирует последующие блоки. Если нода наблюдает за блоком, у которого нет указанного предыдущего блока, она имеет два варианта:

1) Игнорировать блок, так как это может быть вредоносный мусорный блок.

2) Запрос повторной синхронизации с другой нодой. В случае повторной синхронизации TCP-соединение должно быть сформировано с загрузочной нодой, чтобы снизить увеличение объема трафика, необходимого для повторной синхронизации. Однако, если блок на самом деле был плохим блоком, то повторная синхронизация является ненужным, что ведет к увеличение ненужного трафика в сети. Это атака на сеть и приводит к отказу в обслуживании.

Чтобы избежать ненужной повторной синхронизации, ноды будут ждать, пока не будет достигнут определенный порог голосов для потенциально вредоносного блока, прежде чем инициировать соединение с нодой для начала синхронизации. Если блок не получает достаточного количества голосов, его можно считать нежелательным.

B. Флуд Транзакциями

Злоумышленник может отправить много ненужных, но действительных транзакций между учетными записями под его контролем, пытаясь насытить сеть. Без комиссий за транзакции они могут продолжать эту атаку очень долго. Тем не менее, PoW, требуемый для каждой транзакции, ограничивает скорость транзакции, которую может создать злонамеренная организация, без значительного инвестирования в вычислительные ресурсы. Даже при такой атаке, пытаясь раздуть ledger, ноды, которые не используют полную историю блоков, способны обрезать старые транзакции из своей цепочки, это обезопасит использование ledger от такого типа атаки почти для всех пользователей.

C. Sybil Атака

Атакующий может создать сотни нод RaiBlocks на одном компьютере, однако, поскольку система голосования основана на балансе счетов, добавление дополнительных узлов в сеть не даст злоумышленнику дополнительных голосов. Поэтому нет никаких преимуществ, которые будут получены через атаку Sybil.

D. Атака пенни-траты

Атака на пенни - это то, где атакующий тратит бесконечно малые средства на большое количество аккаунтов, чтобы засорить запоминающие устройства нод. Публикации блоков ограничена по скорости в PoW, поэтому это ограничивает создание аккаунтов и транзакций в определенной степени. Ноды, которые не являются полными историческими узлами, могут обрезать аккаунты ниже статистической метрики, где аккаунт, скорее всего, не рабочий. Наконец, RaiBlocks настроен на использование минимального постоянного пространства для хранения, поэтому пространство, необходимое для хранения одного дополнительного аккаунта, пропорционален размеру открытого блока + индексирование = 96B+32B = 128B. Что позволяет в 1 ГБ хранить 8 миллионов аккаунтов за счет пенни. Если узлы захотят обрезать более агрессивно, они могут рассчитать распределение на основе частоты доступа и делегировать нечасто используемые аккаунты для более медленного хранения.

E. Предвычисленная Атака PoW

Поскольку владелец аккаунта будет единственным лицом, добавляющим блоки в цепочку аккаунта, последовательные блоки можно вычислить вместе с их PoW, прежде чем передавать их в сеть. Здесь злоумышленник генерирует множество последовательных блоков, каждый из которых имеет минимальное значение, в течение длительного периода времени. В определенный момент злоумышленник выполнит Denial of Service (DoS) путем флуда сети с большим количеством допустимых транзакций, которые другие ноды будут стараться обработать как можно быстрее. Это усовершенствованная версия флуда в транзакциях, описанных в разделе V-B. Такая атака будет действовать недолго, но может использоваться в сочетании с другими атаками такими как >50% Атаки (раздел V-F) для повышения эффективности. В настоящее время расследуются ограничения скорости передачи и другие методы для смягчения атак.

F. >50% Атака

Показателем консенсуса для RaiBlocks является взвешенная система голосования. Если злоумышленник может набрать более 50% голосов, это может привести к тому, что сеть будет колебаться в результате сбоя системы. Злоумышленник может снизить сумму баланса, которую они должны потерять, не допуская, чтобы хорошие ноды голосовали используя реализацию сетевой DoS. RaiBlocks принимает следующие меры для предотвращения такой атаки:

1) Первичная защита от такого типа атаки - это вес голосования, привязанный к инвестициям в систему. Владелец аккаунта по своей сути стимулируется поддерживать честность системы для защиты своих инвестиций. Попытка изменить ledger будет разрушительной для системы в целом, которая разрушит и их инвестиции.
2) Стоимость такой атаки пропорциональна рыночной капитализации RaiBlocks. В системах PoW можно изобрести технологию, которая дает непропорциональный контроль по сравнению с денежными вложениями, и если атака будет успешной, эта технологию можно направить на другие цели. У RaiBlocks стоимость атаки на систему зависит от самой же системы и, если атака должна быть успешной, инвестиции в атаку не могут быть восстановлены.
3) Чтобы сохранить максимальный кворум голосующих, следующая линия защиты является репрезентативное голосование. Владельцы аккаунтов, которые не могут участвовать в голосовании по причинам подключения к сети, могут назначить

представителя, который будет голосовать с весом их баланса. Максимизация числа представителей повышает устойчивость сети.

4) Форки в RaiBlocks никогда не бывают случайным, поэтому ноды могут принимать правила о том, как взаимодействовать с раздвоенными блоками. Единственный раз, когда неатакующие аккаунты уязвимы для блокировки форк - если они получают баланс от атакующего аккаунта. Аккаунты, которые хотят быть защищенными от блочных форков, могут подождать некоторое время, прежде чем получить от аккаунта, который сгенерировал форк, или никогда не получать вообще этот блок. Получатели также могут создать отдельные аккаунты, которые будут использоваться при получении средств с сомнительных аккаунтов, чтобы изолировать другие свои аккаунты.
5) Последняя мера защиты, которая еще не реализована - block cementing. RaiBlocks делает все возможное, чтобы быстро устранить блок-форки путем голосования. Ноды могут быть настроены для block cementing, что предотвратить их откат после определенного периода времени. Сеть достаточно защищена путем фокусировки на быстрое время предотвращения неоднозначных форков.

Более сложная версия атаки $> 50\%$ подробно описана на рисунке 9. «Offline» - это процент представителей, которые находятся не онлайн в момент голосования. «Stake» - это сумма инвестиции, с которой злоумышленник голосует. «Active» - это представители, которые находятся в режиме онлайн и голосуют по протоколу. Злоумышленник может компенсировать сумму, которую они должны потерять, выбив других голосующих в автономном режиме через сетевую DoS- атаку. Если эта атака станет устойчивой, атакуемые представители станут несинхронизированными и это демонстрирует «Unsync». Наконец, злоумышленник может получить короткий разрыв в относительной силы голосования, переключив атаку DoS для нового набор представителей, в то время как старый набор повторно синхронизирует свой ledger, это демонстрирует «Attack».

Рис. 9. Потенциальный механизм голосования, который может снизить требования к атаке на 51%.

Если злоумышленник может вызвать Stake $>$ Active путем объединения этих обстоятельств, он сможет успешно опрокинуть голоса в ledger за счет своей доли. Мы можем оценить, насколько этот тип атаки дорог, исследуя рыночную капитализацию других систем. Если мы предположим, что 33% представителей находятся в offline режиме или атакованы через DoS, злоумышленнику необходимо будет купить 33% от рыночной капитализации, чтобы атаковать систему путем голосования.

G. Bootstrap Poisoning

Чем дольше злоумышленник будет хранить старый закрытый ключ с балансом, тем выше вероятность того, что балансы, существовавшие в то время, не будут иметь представителей, потому что их балансы или представители перешли на более новые аккаунты. Это означает, что если нода загружает старого представителя сети, где злоумышленник имеет большенство для голосования по сравнению с представителями в тот момент времени, они смогут варьировать решения о голосовании на этой ноде. Если этот новый пользователь хочет взаимодействовать с кем-либо, кроме атакующеий ноды, все его транзакции будут отклоняться, так как они имеют разные заглавные блоки. Конечным результатом является то, что ноды могут тратить время на новые узлы в сети, передавая им плохую информацию. Чтобы предотвратить это, ноды могут быть сопряжены с исходной базой данных аккаунтов и с хорошо известными главными блоками, это замена для загрузки базы данных вплоть до блока генезиса. Чем ближе загрузка будет к актуальноий версии, тем выше вероятность полной защиты от этой атаки. В конце концов, эта атака, вероятно, не хуже, чем загрузка нежелательных данных в ноды при загрузке блоков, поскольку они не смогут совершать сделки с кем-либо, у кого есть актуальная база данных.

VI. РЕАЛИЗАЦИЯ

В настоящее время справочная исполнение реализовано на C++ и выпускает релизы с 2014 года на Github [10].

A. Особенности дизайна

Реализация RaiBlocks соответствует стандарту архитектуры, описанному в этой статье. Дополнительные характеристики описаны здесь.

1) Алгоритм подписи: RaiBlocks использует модифицированный алгоритм эллиптической кривой ED25519 с хешированием Blake2b для всех цифровых подписей [11]. ED25519 был выбран для быстрой подписи, быстрой проверки и высокой безопасности.

2) Алгоритм Хеширования: Поскольку алгоритм хэширования используется только для предотвращения спама сети, выбор алгоритм менее важен по сравнению с криптовалютами на основанных на майнинге. Наша реализация использует Blake2b как цифровой алгоритм против содержимого блока [12].

3) Функция деривации ключа: В кошельке ключи шифруются паролем, а пароль передается через функцию деривации ключей для защиты от попыток взлома ASIC. В настоящее время Argon2 [13] является победителем единственного публичного конкурса, направленного на создание отказоустойчивой функции деривации ключей.

4) Блочныий интервал: Поскольку каждый аккаунт имеет свою собственную цепочку, обновления могут выполняться асинхронно с состоянием сети. Поэтому интервалы между блоков нет и транзакции могут быть опубликованы мгновенно.

5) Протокол сообщенииий UDP: Наша система рассчитана на бессрочную работу с минимальным объемом вычислительных ресурсов. Все сообщения в системе были разработаны таким образом, чтобы они были без состояния и помещались в один пакет UDP. Это также облегчает для облегченных пиров с прерывистым подключением участия в сети без повторного восстановления краткосрочных TCP-соединений. TCP используется только для новых узлов, когда они хотят загрузить цепочки блоков массовым способом.

Ноды могут быть уверены, что их транзакции поступили в сеть при соблюдении транзакций широковещательного трафика другими нодами, а это должны увидеть несколько копий эхом вернувшимся к себе.

B. Протокол IPv6 и multicast

Создание поверх UDP без установления соединения позволяет в будущим использовать многоадресную рассылку по IPv6 в качестве замены традиционного флуда транзакций и передачи голосований. Это уменьшит потребление пропускной способности сети и даст больше гибкости правил для нод.

C. Производительность

На момент написания этой статьи сеть RaiBlocks обработала 4,2 миллиона транзакций, получив размер блокчейна размером 1,7 ГБ. Время транзакции измеряется в секундах. Текущая эталонная реализация, замеренная на SSD, может обрабатывать более 10 000 транзакций в секунду, в первую очередь это связано с ограничением IO.

VII. ИСПОЛЬЗОВАНИЕ РЕСУРСОВ

Это обзор ресурсов, используемых нодой RaiBlocks. Кроме того, мы переходим к идеям уменьшению использования ресурсов для конкретных случаев использования. Упрощенные ноды обычно называются легкими, обрезанными или упрощенными для верификации платежа.

A. Сеть

Объем сетевой активности зависит от того, насколько сеть способствует здоровью сети.

1) Представительная: Представительная нода требует максимальных сетевых ресурсов, поскольку она следит за трафиком голосов от других представителей и публикует свои собственные голоса.

2) Без доверия: Недоверенная нода похожа на представительная ноду, но является только наблюдателем и не содержит представительного закрытого ключа аккаунтам, и а также не публикует собственные голоса.

3) С доверием: Доверенная нода наблюдает за трафиком голосов от одного представителя, которому он доверяет, чтобы правильно выполнить консенсус. Это сокращает количество входящего трафика голосования от представителей, ссылающихся на эту ноду.

4) Легкая: Легкая нода также является доверенным узлом, которыий отслеживает трафик только для аккаунтов, в которых он заинтересован, что позволяет минимальное использование сети.

5) Загрузочная: Загрузочная нода обслуживает часть или весь ledger для нод, которые подключены к сети. Это делается через TCP-соединение, а не UDP, из-за с большого объема передаваемых данных.

B. Занимаемое место на диске

В зависимости от требований пользователя разные конфигурации нод требуют разных условий к хранению.

1) Историческая: Нода, заинтересованная в сохранении полной исторической записи всех транзакций, потребует максимального места хранения.

2) Текущиий: В связи с дизайном сохранения накопленных балансов с блоками, ноды должны держать только последние или главные блоки для каждого аккаунта для того, чтобы участвовать в консенсусе. Если нода не заинтересована в сохранении полной истории, она может сохранять только главные блоки.

3) Легкая: Легкая нода не сохраняет данные из локального ledger, но участвует только в сети, чтобы наблюдать за деятельностью на аккаунтах, в которых она заинтересована, или, возможно, создавать новые транзакции с закрытыми ключами.

C. Процессор (CPU)

1) Генерация Транзакции: Нода заинтересована в создании новых транзакций, которые должны произвести Proof of Work nonce, чтобы пройти механизм регулирования RaiBlocks. Вычисление различных аппаратных средств приведено в приложении A.

2) Представительная: Представитель должен проверять подписи для блоков, голосов, а также создавать свои собственные подписи для участия в консенсусе. Объем используемых ресурсов CPU для представительной ноды значительно меньше, чем при генерации транзакции и может работать с любым процессором на современном компьютере.

3) Наблюдательная: Наблюдательная нода не генерирует свои собственные голоса. Поскольку используемые ресурсы на подпись минимальны, требования к процессору почти идентичны работе с представительной нодой.

VIII. ЗАКЛЮЧЕНИЕ

В этом документе мы представили фреймворк для доверенной, безкомиссионной и с низкой задержкой

криптовалюты, которая использует новую структуру блок-решетки и делегированного PoS голосования. Сеть требует минимальных ресурсов, не требует мощного оборудования для интеллектуального анализа данных и может обрабатывать высокую пропускную способность транзакций. Все это достигается за счет наличия отдельных блоков для каждого аккаунта, устранение проблем доступа и неэффективности глобальной структуры данных. Мы идентифицировали возможные атаки в системе и представили аргументы в отношение того, насколько RaiBlocks устойчив к этим формам атак.

Приложение A
POW АППАРАТНЫЕ ПОКАЗАТЕЛИ

Как упоминалось ранее, PoW в RaiBlocks - это для сокращения сетевого спама. Наша реализация нод обеспечивает ускорение, которое может использовать преимущества совместимых с OpenCL графических процессоров. В таблице I приведено сравнение производительности различных аппаратных средств в реальных условиях. В настоящее время порог PoW фиксирован, но адаптивный порог может быть реализован по мере достижения средней вычислительной мощности.

Таблица I
ПРОИЗВОДИТЕЛЬНОСТЬ ОБОРУДОВАНИЯ ДЛЯ POW

Device	Transactions Per Second
Nvidia Tesla V100 (AWS)	6.4
Nvidia Tesla P100 (Google,Cloud)	4.9
Nvidia Tesla K80 (Google,Cloud)	1.64
AMD RX 470 OC	1.59
Nvidia GTX 1060 3GB	1.25
Intel Core i7 4790K AVX2	0.33
Intel Core i7 4790K,WebAssembly (Firefox)	0.14
Google Cloud 4 vCores	0.14-0.16
ARM64 server 4 cores (Scaleway)	0.05-0.07

БЛАГОДАРНОСТЬ

Мы хотели бы поблагодарить Брайан Пью за компиляцию и форматирование этой статьи.

Список литературы

[1] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” 2008. [Online]. Available: http://bitcoin.org/bitcoin.pdf
[2] “Bitcoin median transaction fee historical chart.” [Online]. Available: https://bitinfocharts.com/comparison/bitcoin-median_transaction_fee.html
[3] “Bitcoin average confirmation time.” [Online]. Available: https://blockchain.info/charts/avg-confirmation-time
[4] “Bitcoin energy consumption index.” [Online]. Available: https://digiconomist.net/bitcoin-energy-consumption
[5] S. King and S. Nadal, “Ppcoin: Peer-to-peer crypto-currency with proof-of-stake,” 2012. [Online]. Available: https://peercoin.net/assets/paper/peercoin-paper.pdf
[6] C. LeMahieu, “Raiblocks distributed ledger network,” 2014.
[7] Y. Ribero and D. Raissar, “Dagcoin whitepaper,” 2015.
[8] S. Popov, “The tangle,” 2016.
[9] A. Back, “Hashcash - a denial of service counter-measure,” 2002. [Online]. Available: http://www.hashcash.org/papers/hashcash.pdf
[10] C. LeMahieu, “Raiblocks,” 2014. [Online]. Available: https://github.com/clemahieu/raiblocks
[11] D. J. Bernstein, N. Duif, T. Lange, P. Shwabe, and B.-Y. Yang, “High-speed high-security signatures,” 2011. [Online]. Available: http://ed25519.cr.yp.to/ed25519-20110926.pdf
[12] J.-P. Aumasson, S. Neves, Z. Wilcox-O’Hearn, and C. Winnerlein, “Blake2: Simpler, smaller, fast as md5,” 2012. [Online]. Available: https://blake2.net/blake2.pdf
[13] A. Biryukov, D. Dinu, and D. Khovratovich, “Argon2: The memory-hard function for password hashing and other applications,” 2015. [Online]. Available: https://password-hashing.net/argon2-specs.pdf