RaiBlocks: Крыптавалютная Сетка Без Камісій

Colin LeMahieu
clemahieu@gmail.com

Анатацыя—У апошні час пры высокім попыце і абмежаванай маштабаванасці павялічыўся сярэдні час транзакцый і камісій у папулярных крыптавалютах, што прывяло да нездавальняючага досведу. Тут мы пазнаёмім з RaiBlocks, крыптавалютай з новай архітэктурай блокавай структуры, дзе кожны акаўнт мае свой ўласны ланцуг блок, забяспечваючы амаль імгненную хуткасць транзакцыі і неабмежаваную маштабаванасць. У кожнага карыстальніка ёсць свой ланцуг блок, дазваляючы ім асінхронна абнаўляцца яго для астатняй сеткі, што прыводзіць да хуткіх транзакцый з мінімальнымі выдаткамі. Транзакцыі адсочваюць астатак на акаўнтах, а не сумы ў транзакцыі, дазваляючы агрэсіўнае абразанне базы даных без шкоды для бяспекі. На сённяшні дзень сетка RaiBlocks апрацавала больш 4,2 млн транзакцый маючы поўны аб'ём базы крыху больш за 1,7 ГБ. Тое, што RaiBlocks без камісіі і транзакцыі ў долі секунд робяць яго галоўнай крыпртавалютай для спажывецкіх транзакцый.

Індэкс Умовы—крыптавалюта, блокчэйн, raiblocks, размеркаваная база, лічбавы, транзакцыі

I. УСТУПЛЕННЕ

З МОМАНТУ з'яўлення Bitcoin ў 2009 годзе назіраецца сыход ад традыцыйных, якія падтрымліваюцца ўрадам валют і фінансавых сістэм, да сучасных сістэм плацяжоў, заснаваных на крыптаграфіі, якія прапануюць магчымасць захоўваць і пераводзіць грошы надзейным і бяспечным спосабам [1]. Для эфектыўнага функцыянавання, валюта павінна лёгка перадавацца, не адмяняцца і мець некаторыя абмежаванні або зусім без камісій. Павялічаны час транзакцый, вялікія камісіі і сумніўная маштабаванасць сеткі выклікалі пытанні аб практычнасці Bitcoin як паўсядзённым валюты.

У гэтым артыкуле мы знаёмім з RaiBlocks - крыптавалютай з малай затрымкай, заснаванай на інавацыйнай блокавай структуры даных, якая прапануе неабмежаваную маштабаванасць і адсутнасць транзакцыйных камісій. RaiBlocks распрацаваны як просты пратакол з выключнай мэтай быць высокапрадукцыйнай крыптавалютай. Пратакол RaiBlocks можа працаваць на маламагутным абсталяванне, дазваляючы быць практычнай і дэцэнтралізаванай крыптавалютай для паўсядзённага выкарыстання.

Статыстыка крыптавалюты, прадстаўленая ў гэтым дакуменце, з'яўляецца дакладнай на дзень публікацыі.

II. ДАВЕДАЧНАЯ ІНФОРМАЦЫЯ

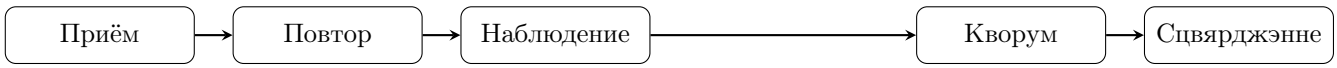
У 2008 годзе ананім пад псеўданімам Satoshi Nakamoto апублікаваў белую паперу, якая апісвае першую ў свеце дэцэнтралізаваную крыптавалюту Bitcoin [1]. Ключавым новаўвядзеннем Bitcoin стаў блокчэйн, публічная, нязьменная і дэцэнтралізаваная структура дадзеных, якая выкарыстоўваецца ў якасці рэгістра аперацый з валютай. На жаль, па меры таго як Біткоін развіваўся, некаторыя праблемы ў пратаколе зрабілі Bitcoin недаступным для многіх прыкладанняў:

1) Дрэнная маштабаванасць: Кожны блокэйн можа захоўваць абмежаваную колькасць дадзеных, што азначае, што сістэма можа апрацоўваць толькі столькі транзакцый у секунду, колькі хапае месца ў блоку для запісу транзакцый. У цяперашні час сярэдні збор за транзакцыю складае $10,38 [2].
2) Высокая затрымка: сярэдні час пацверджання складае 164 хвіліны [3].
3) Энерга неэфектыўная: Сетка Біткоін спажывае каля 27.28 млрд. КВт/г у год, выкарыстоўваючы ў сярэднім 260KWh на транзакцыю [4].

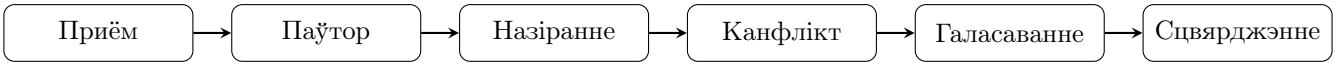
Біткоін і іншыя крыптавалюты функцыянуюць шляхам дасягнення кансэнсусу ў дачыненні да свайго глабальнага блокчейна, з тым каб праверыць законнасць аперацыі, супраціўляючыся зламыснікам. Біткоін дасягае кансэнсусу з дапамогай эканамічнай меры, званы доказам працы (PoW). У сістэме PoW ўдзельнікі канкуруюць за вылічэнне колькасці, званага nonce, так што хэш ўсяго блока знаходзіцца ў мэтавым дыяпазоне. Гэты дапушчальны дыяпазон зваротна прапарцыйны сукупнай вылічальнай магутнасці ўсёй сеткі Bitcoin для падтрымання ўзгодненага сярэдняга часу, затрачвае на пошук дапушчальнага значэння nonce. Тады якая знайшла сапраўдны нумар nonce можа дадаць блок у блокчейн; таму той, хто валодае велізарным вылічальным рэсурсам для вылічэнні nonce, маюць вялікую ролю ў стане блокчэйна. PoW забяспечвае ўстойлівасць да атакі Sybil, дзе атакуючы паводзіць сябе як некалькі аб'ектаў, каб атрымаць дадатковую магутнасць у дэцэнтралізаванай сістэме, зніжае ўзровень стану гонкі, якія па сваёй прыродзе існуюць пры доступе да глабальнай структуры дадзеных.

Альтэрнатыўны кансэнсусны пратакол які пацвярджае долю ўдзелу (PoS), упершыню быў уведзены Peercoin ў 2012 годзе [5].У сістэме PoS ўдзельнікі галасуюць з вагой, эквівалентным колькасці сродкаў, якія яны маюць у дадзенай крыптавалюце. З дапамогай гэтага механізму тыя, хто мае вялікія фінансавыя інвестыцыі, атрымліваюць больш уплыву і па па сваёй сутнасці стымулююцца падтрымліваць сумленнасць сістэмы або рызыка страціць свае інвестыцыі. PoS пазбаўляе ад марнатраўнай канкурэнцыі магутнасці вылічэнняў, патрабуючы толькі лёгкага праграмнага забеспячэння, які працуе на малай магутнасці.

Арыгінальная папера RaiBlocks і першая бэта-



(a) Калі канфлікт не знойдзены, то ніякіх дадатковых дзеянній не патрабуецца.



(b) У выпадку канфліктах транзакцыі Ноды галасуюць за дапушчальную транзакцыю.

Рыс. 1. RaiBlocks не патрабуе дадатковых выдаткаў для тыповых транзакцый. У выпадку канфлікту транзакцый ноды павінны галасаваць за дапушчальныя транзакцыі

рэалізацыя былі апублікаваны ў снежні 2014 года, што робіць яе адным з першых крыптавалют якія выкарыстоўваюць Накіраваны Ацыклічны Граф (DAG) [6]. Неўзабаве пачалі развівацца іншыя крыптавалюты DAG, а менавіта DagCoin / Byteball і IOTA [7], [8]. Гэтыя крыптавалюты на аснове DAG зламалі форму блокчэйна, павялічыўшы прадукцыйнасць сістэмы і бяспеку. Byteball дасягаюць кансэнсусу, абапіраючыся на «галоўны ланцуг» які складаецца з сумленных, аўтарытэтных і карыстальнікаў - надзейных «сведак», у той час як IOTA дасягае кансенсусу па PoW складзеным странзакціям. RaiBlocks дасягае кансэнсусу пасродкам ўзважанага галасавання па канфліктуючым транзакцыям. Гэтая сістэма кансэнсусу забяспечвае больш хуткія і больш дэтэрмінаваныя транзакцыі, захоўваючы пры гэтым моцную дэцэнтралізаваную сістэму. RaiBlocks працягвае гэтую распрацоўку і пазіцыянуе сябе як адну з самых высокапрадукцыйных крыптавалют.

III. КАМПАНЕНТЫ RAIBLOCKS

Перш чым апісваць агульную архітэктуру сістэмы RaiBlocks, мы вызначым асобныя яе кампаненты.

A. Акаўнт

Акаўнт - гэта частка публічнага ключа ад пары ключоў лічбавага подпісу. Публічны ключ, таксама названы адрасам, перадаецца іншым удзельнікам сеткі, у той час як закрыты ключ захоўваецца ў сакрэце. Пакет дадзеных з лічбавым подпісам гарантуе, што змесціва было адобрана уладальнікам закрытага ключа. Адзін карыстальнік можа кіраваць многімі акаўнтамі, але для кожнага акаўнта можа існаваць толькі адзін публічны адрас.

B. Блок/транзакцыі

Тэрмін «блок» і «транзакцыя» часта выкарыстоўваюцца ўзаемазамяняема, калі блок змяшчае адну транзакцыю. Транзакцыя канкрэтна ставіцца да дзеяння, тады як блок ставіцца да лічбавага кадаванню транзакцыі. Транзакцыі падпісваюцца закрытым ключом, якія належаць акаўнту, на якім выконваецца транзакцыя.

C. Ledger

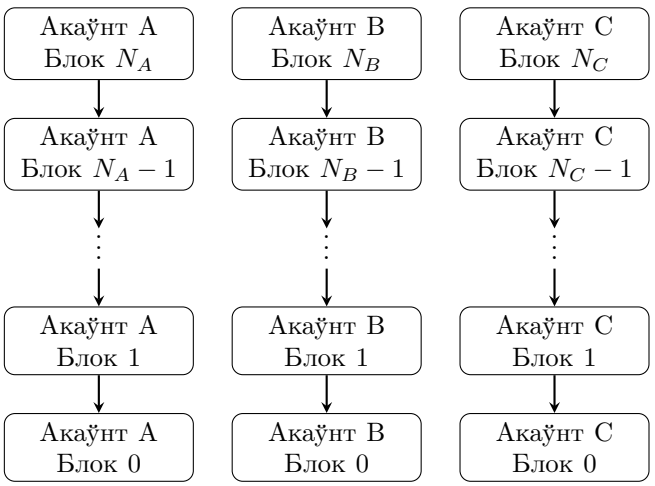
Ledger - это глобальный набор аккаунтов, где каждый аккаунт имеет свою собственную цепочку транзакций (рис 2). Гэта ключавы кампанент дызайну, які падпадае пад катэгорыю замены пагаднення аб часе выканання з пагадненнем часу распрацоўкі; кожны згаджаецца з дапамогай праверкі подпісы, што толькі ўладальнік акаўнта можа змяніць свай ўласны ланцуг. Гэта пераўтварае, мяркуючы па выглядзе агульнай структуру дадзеных, размеркаваны ledger, у набор не агульных (прыватных) структур карыстання.

D. Нода

Нода ўяўляе сабой частка праграмнага забеспячэння, які працуе на кампутары, які адпавядае пратаколу RaiBlocks і ўдзельнічае ў сетцы RaiBlocks. Праграмнае забеспячэнне кіруе ledger і любымі ўліковымі запісамі, якімі можа кіраваць Нода, калі такія ёсць. Нода можа альбо захоўваць увесь ledger, альбо яе абрэзаную гісторыю, якая змяшчае толькі апошнія некалькі блокаў ланцуга кожнага акаўнта. Падчас усталявання новай Ноды рэкамендуецца правяраць ўсю гісторыю і рабіць зрэз лакальна.

IV. АГЛЯД СІСТЭМЫ

У адрозненне ад блокчэйнов, якія выкарыстоўваюцца ў многіх іншых крыптавалютах, RaiBlocks выкары-

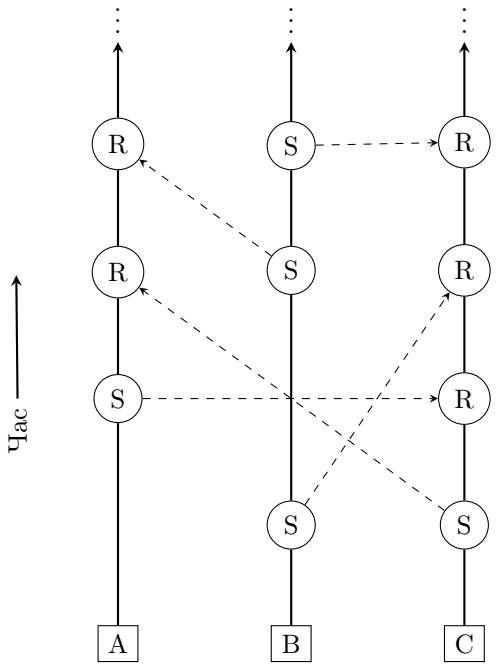


Рыс. 2. Кожны акаўнт мае свой уласны блокчэйн, які змяшчае гісторыю змены балансу. Блок 0 павінен быць які адкрывае транзакцыяй (Секц IV-B)

стоўвае блок-рашэцістую структуру. Кожны рахунак мае свой уласны блокчэйн (аккаунт- ланцуг), эквівалент гісторыі транзакцый/балансу акаўнта (Рис. 2). Кожны ланцуг акаўнта можа быць абноўлён толькі уладальнікам акаўнта; гэта дазваляе кожнаму ланцугу акаўнта быць неаднаразова абноўленым і асінхронна да астатняй блок-рашотцы, што прыводзіць да хуткім транзакцый. Пратакол RaiBlocks з'яўляецца надзвычай лёгкім; кожная аперацыя ўпісваецца ў патрабаваны мінімальны памер пакета udp для перадачы праз інтэрнэт. Патрабаванні да абсталявання для нод таксама мінімальныя, так як ноды павінны толькі запісваць і рэтрансляваць блокі для большасці транзакцый (Рис 1).

Сістэма ініцыюецца з акаўнтам генезісу, які змяшчае баланс генезісу. Баланс генезісу з'яўляецца фіксаваныай колькасцю і ніколі не можа быць павялічаны. Баланс генезісу дзеліцца і адпраўляецца на іншыя акаўнты праз транзакцыі адпраўкі зарэгістраваныя ў ланцугу акаўнта генезісу. Сума астаткаў на ўсіх рахунках ніколі не перавысіць першапачатковы баланс генезісу, што дае сістэме верхнюю мяжу па колькасці і не дазваляе павялічыць яго. У гэтым раздзеле апісваецца стварэнне і распаўсюджванне розных тыпаў транзакцый у сеткі.

У гэтым раздзеле апісваецца стварэнне і распаўсюджванне розных тыпаў транзакцый у сеткі.



Рыс. 3. Візуалізацыя блок-рашоткі. Кожны пераклад сродкаў патрабуе блока адпраўкі (S) і блока атрымання (R), кожны падпісаны уладальнікам акаўнта-ланцуга (A, B, C)

A. Транзакцыі

Пераклад сродкаў з аднаго акаўнта на другі патрабуе двух аперацый: адпраўкі спісання сродкаў з балансу адпраўніка і прыёму дадання сродкаў на рахунак атрымальніка (рис. 3). Перанос сум у выглядзе асобных аперацый на акаўнтах адпраўніка і атрымальніка служыць для некалькіх важных мэтаў:

1) Паслядоўнасць ўваходзячых перадач, якія па сваёй сутнасці асінхронны.
2) Захаваць транзакцыю невялікай каб змясціцца ў udp-пакеты.
3) Палягчаць ledger абразанне шляхам мінімізацыі высновы дадзеных.
4) Ізаляцыя прынятых транзакцый ад яшчэ не прынятых.

Больш за аднаго акаўнта, які адпраўляе на адзін і той жа з'яўляецца асінхроннай аперацыяй, затрымка ў сеткі і якія адпраўляюць акаўнты не абавязкова знаходзяцца ў сувязі адзін з адным, гэта азначае, што няма універсальнага прымальнага спосабу даведацца, якая транзакцыя адбылася ў першую чаргу. Паколькі даданне асацыятыўна, парадак паслядоўных уводных дадзеных не мае значэння, і таму нам проста трэба глабальнае пагадненне. Гэта ключавы кампанент дызайну, які пераўтворыць пагадненне аб часе выканання ў пагадненне аб часе распрацоўкі. Які прымае рахунак мае кантроль над тым, каб вырашыць, які пераклад прыйшоў першым і тым самым ўсталяваць свае і парадак ўваходных блокаў.

Калі акаўнт хоча зрабіць вялікі пераклад, які быў атрыманы як набор невялікіх перакладаў, мы хочам прадставіць гэта такім чынам, які ўпісваецца ў пакет UDP. Калі атрымлівальны акаўнт ўсталёўвае паслядоўнасць ўваходных перакладаў, ён захоўвае агульную суму свайго балансу рахунку, так што ў любы час ён мае магчымасць перавесці любую суму з фіксаваным памерам транзакцыі. Гэта адрозніваецца ад мадэлі транзакцыі ўводу/высновы выкарыстоўванай Bitcoin і іншымі крыптавалютамі.

Некаторыя ноды не зацікаўленыя ў выдаткоўванні рэсурсаў на захоўванне поўнай гісторыі транзакцый акаўнта; яны зацікаўленыя толькі ў бягучым балансе кожнага акаўнта. Калі рахунак здзяйсняе транзакцыю, ён кадуе набыты баланс, і гэтыя ноды павінны адсочваць толькі апошні блок, які дазваляе ім адкідаць гістарычныя дадзеныя, захоўваючы пры гэтым правільнасць.

Нават з акцэнтам на пагаднення аб часе распрацоўкі, ёсць акно затрымкі пры праверцы транзакцый з-за выяўлення і апрацоўкі дрэнных удзельнікаў у сеткі. Бо пагадненне ў RaiBlocks дасягаюцца хутка, ад мілісекунд да секунд, мы можам прадставіць карыстачу дзве знаёмыя катэгорыі ўваходных транзакцый: прынятыя і ня прынятыя. Прынятыя транзакцыі - гэта транзакцыі, у якіх акаўнт стварыў блок атрымання. Ня прынятыя транзакцыі яшчэ не былі ўключаны ў сукупны баланс атрымальніка. Гэта замена больш складанай і нязвыклай метрыкі пацверджання ў іншых крыптавалютах.

B. Стварэнне Акаўнта

Каб стварыць акаўнт, Вам неабходна стварыць Open транзакцыю (Рис. 4). Open транзакцыя ёсць найпершай транзакцыяй кожнага ланцуга ак і можа

быць створана пры першым паступленні сродкаў. Поле account захоўвае адкрыты ключ (адрас), вытворны ад закрытага ключа, які выкарыстоўваецца для подпісу. Поле source змяшчае хэш транзакцыі, якая накіравала сродкі. Пры стварэнні акаўнта, прадстаўнік павінен быць абраны для галасавання ад вашага імя, ён можа быць зменены пазней (раздел IV-F). Акаўнт можа аб'явіць сябе сваім прадстаўніком.

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

Рыс. 4. Прыклад open транзакцыі

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

Рыс. 5. Прыклад send транзакцыі

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

Рыс. 6. Прыклад receive транзакцыі

C. Баланс Акаўнта

Астатак на рахунку запісваецца ў ledger. Замест таго, каб запісваць суму транзакцыі, пацверджанне (раздел IV-I) патрабуе праверкі розніцы паміж балансам ў блоку адпраўкі і балансам папярэдняга блока. Які атрымлівае рахунак можа затым павялічыць папярэдні астатак, вымераны ў канчатковым астатку, паказаным у новым блоку прыёму. Гэта робіцца для павышэння хуткасці апрацоўкі пры загрузцы вялікіх аб'ёмаў блокаў. Пры запыце гісторыі рахункі, сумы ўжо дадзены.

D. Адпраўка з Акаўнта

Каб адправіць з адрасу, на адрасе павінен быць ужо адкрыты блок, а таксама і баланс (Рис. 5). У previous поле ўтрымлівае хэш папярэдняга блока ў ланцугу акаўнта. Поле destination ўтрымлівае адрас для якога адпраўляюцца сродкі. Блок адпраўкі з'яўляецца нязменлівым пасля пацверджання. Пасля перадачы ў сетку сродкі неадкладна адымаюцца з балансу рахунку адпраўшчыка і знаходзяцца ў статусе pending для атрымальніка, пакуль ён не падпіша блок каб прыняць гэтыя сродкі. Адкладзеныя (Pending) сродкі не варта лічыць якія чакаюць пацверджання, паколькі яны ўжо выдаткаваныя з акаўнта адпраўніка і адпраўнік не можа ануляваць гэтую транзакцыю.

E. Атрыманне Транзакцый

Для завяршэння транзакцыі атрымальнік адпраўленых сродкаў павінен стварыць блок прыёму на ўласнай акаўнт-ланцугу (Рис. 6). Поле source спасылаецца на хэш транзакцыі адпраўкі. Як толькі блок створаны і трансляваны ў сетку, баланс акаўнта абнаўляецца і сродкі афіцыйна залічваюцца на рахунак атрымальніка.

F. Прызначэнне прадстаўніка (Representative)

Уладальнікі акаўнтаў, маюць магчымасць выбраць прадстаўніка для галасавання ад свайго імя. Гэта з'яўляюцца магутным інструментам дэцэнтралізацыі, якая не мае моцнага аналага ў пратаколе Proof of Work або Proof of Stake. У звычайных сістэмах PoS ва ўладальніка акаўнта павінна быць запушчана Нода для ўдзелу ў галасаванні. Пастаянны запуск Ноды немэтазгодны для многіх карыстальнікаў; перадача прадстаўніку права голасу ад імя акаўнта, што аслабляе гэта патрабаванне. Уладальнікі акаўнтаў маюць магчымасць перапрызначыць прадстаўніка акаўнта ў любы час. Транзакцыя change змяняе прадстаўніка акаўнта, адымаючы вагу галасавання ад старога прадстаўніка і дадаўшы вагу для новага прадстаўніка (рис. 7). Грашовыя сродкі ў дадзенай аперацыі не перамяшчаюцца, і прадстаўнік не мае магчымасці марнаваць сродкі на рахунак.

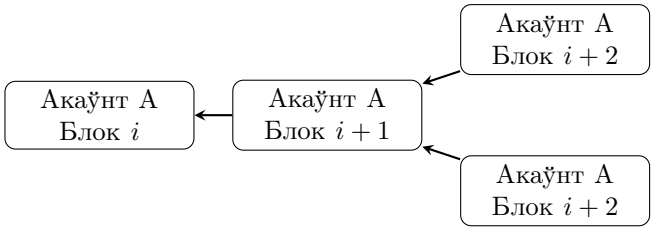
```
change {
    previous: DC04354B1...AE8FA2661B2,
    representative: xrb_1anrz...posrs,
    work: 0000000000000000,
    type: change,
    signature: 83B0...006433265C7B204
}
```

Рыс. 7. Прыклад change транзакцыі аб змене прадстаўніка

G. Форк і Галасаванне

Форк ўзнікае, калі j падпісаў блокі $b_1, b_2, \ldots, b_j$ и утвердил те же блоки, что и их предшественник (рис. 8). і зацвердзіў тыя ж блокі, што і іх папярэднік (мал. 8). Гэтыя блокі выклікаюць супярэчлівае ўяўленне

аб стане акаўнта і павінны быць ліквідаваны. Толькі ўладальнік акаўнта мае магчымасць падпісваць блокі ў свой ланцуг акаўнтаў, таму форк павінен быць вынікам дрэннага праграмавання або зламыснай намеры (падвойнага расходавання) уладальнікам акаўнта.



Рыс. 8. Форк ўзнікае, калі два (ці больш) падпісаных блока спасылаюцца на той жа папярэдні блок. Старыя блокі злева, а новыя блокі справа

Пасля выяўлення прадстаўнік створыць галасаванне, якое спасылаецца на блок $\hat{b}_i$ ў яго ledger і перадасць яго ў сетку. Вага галасавання ноды, w_i, з'яўляецца сумай балансаў ўсіх рахункаў, якія назвалі яго сваім прадстаўніком. Нода будзе назіраць ўваходныя голасы ад іншых M онлайн прадстаўнікоў і захоўваць накапляльную адпаведнасць на 4 перыяду галасавання, 1 хвіліна на ўсе, і пацвердзіць які перамог блок (ураўненне 1).

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

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

Самы папулярны блок b^* будзе мець большасць галасоў і будзе захаваны ў ledger ноды (ураўненне 2). Блок (і), што страцілі галасы, выдаляюцца. Калі прадстаўнік заменіць блок у сваёй ledger, ён створыць новае галасаванне з больш высокім парадкавым нумарам і перадасць новае галасаванне ў сетку. Гэта адзіны сцэнар, калі галасуюць прадстаўнікі.

У некаторых выпадках кароткія праблемы падлучэння да сеткі могуць прывесці да таго, што перадаваны блок не будзе прыняты ўсімі пірамі (peers). Любы наступны блок на гэтым запісе будзе праігнараваны як недапушчальны пірамі, якія не бачылі пачатковую перадачу. Паўторная трансляцыя гэтага блока будзе прынятая астатнімі пірамі і наступныя блокі будуць вынятыя аўтаматычна. Нават пры ўзнікненні форка або адсутнага блока закранаюцца толькі акаўнты, названыя ў транзакцыі; астатняя частка сеткі працягвае апрацоўку транзакцый для ўсіх іншых акаўнтаў.

H. Proof of Work

Усе чатыры тыпу транзакцый маюць працоўнае поле, якое павінна быць правільна запоўнена. Поле work дазваляе стваральніку транзакцыі вылічыць nonce такое, што хэш-код nonce аб'ядноўваўся з папярэднім полем у receive/send/change транзакцыях або ў поле акаўнта ў адкрывалай транзакцыі, ніжэй пэўнага парогавага значэння. У адрозненне ад Біткоін, PoW ў RaiBlocks выкарыстоўваецца як антыспам інструмент, аналагічная сістэма Hashcash, і можа быць вылічаны за некалькі секунд [9]. Пасля адпраўкі транзакцыі PoW для наступнага блока можна папярэдне вылічыць, бо, як вядома папярэдняе поле блока. Гэта робіць транзакцыі імгненнымі для канчатковага карыстальніка да таго часу, пакуль час паміж транзакцыямі перавышае час, неабходнае для вылічэнні PoW.

I. Праверка транзакцый

Каб блок лічыўся дапушчальным, ён павінен мець наступныя атрыбуты:

1) Блок ўжо не павінен быць у ledger (аперацыі паўтараюцца).
2) Павінен быць падпісаны уладальнікам акаўнта.
3) Папярэдні блок з'яўляецца галоўным у ланцугу акаўнта. Калі ён існуе, але не галоўны, то гэта форк.
4) Акаўнт павінен мець блок адкрыцця.
5) Вылічаемы хэш адпавядае парогаваму значэнню PoW.

Калі гэта блок атрымання, праверце, ці чакае хэш зыходнага блока, гэта значыць ён яшчэ не быў згашаны. Калі гэта блок адпраўкі, бягучы баланс павінны быць менш папярэдняга значэння балансу.

V. ВЕКТАРЫ НАПАДУ

RaiBlocks, як і ўсе дэцэнтралізаваным крыптавалюты, могуць падвергнуцца нападу з боку зламыснікаў у спробе атрымання фінансавай выгады або разбурэння сістэмы. У гэтым раздзеле мы апісвае некалькі магчымых сцэнарыяў атакі, наступствы такіх нападаў, і якія прэвентыўныя меры прымае пратакол RaiBlock.

A. Сінхранізацыі Блокаў

У Раздзеле IV-G мы абмеркавалі сцэнар, калі блок можа не перадавацца належным чынам, у выніку чаго сетку ігнаруе наступныя блокі. Калі нода назірае за блокам, у якога няма названага папярэдняга блока, яна мае два варыянты:

1) Не зважаць на блок, так як гэта можа быць шкоднасны смеццевы блок.
2) Запыт паўторнай сінхранізацыі з другой нодай.

У выпадку паўторнай сінхранізацыі TCP-злучэнне павінна быць сфармавана з загрузнай нодай, каб знізіць павелічэнне аб'ёму трафіку, неабходнага для паўторнай сінхранізацыі. Аднак, калі блок на самай справе быў дрэнным блокам, то паўторная сінхранізацыя з'яўляецца непатрэбнай, што вядзе да павелічэнне непатрэбнага трафіку ў сеткі. Гэта атака на сетку і прыводзіць да адмовы ў абслугоўванні.

Каб пазбегнуць непатрэбнай паўторнай сінхранізацыі, ноды будуць чакаць, пакуль не будзе дасягнуты пэўны парог галасоў для патэнцыйна шкоднаснага блока, перш чым ініцыяваць злучэнне з нодай для пачатку сінхранізацыі. Калі блок не атрымлівае дастатковай колькасці галасоў, яго можна лічыць непажаданым.

B. Флуд Транзакцыямі

Зламыснік можа адправіць шмат непатрэбных, але сапраўдных транзакцый паміж ўліковымі запісамі пад яго кантролем, спрабуючы насыціць сетку. Без камісій за транзакцыі яны могуць працягваць гэтую атаку вельмі доўга. Тым не менш, PoW, патрабаваны для кожнай транзакцыі, абмяжоўвае хуткасць транзакцыі, якую можа стварыць зламысная арганізацыя, без значнага інвеставання ў вылічальныя рэсурсы. Нават пры такой атацы, спрабуючы раздуць ledger, ноды, якія не выкарыстоўваюць поўную гісторыю блокаў, здольныя абрэзаць старыя транзакцыі з свайго ланцуга, гэта засцеражэ выкарыстанне ledger ад такога тыпу атакі амаль для ўсіх карыстальнікаў.

C. Sybil Атака

Атакуючы можа стварыць сотні нод RaiBlocks на адным кампутары, аднак, паколькі сістэма галасавання заснавана на балансе рахункаў, даданне дадатковых вузлоў у сетку не дасць зламысніку дадатковых галасоў. Таму няма ніякіх пераваг, якія будуць атрыманы праз атаку Sybil.

D. Атака пені траты

Атака на пені - гэта тое, дзе атакуючы марнуе бясконца малыя сродкі на вялікую колькасць акаўнтаў, каб засмеціць запамінальныя прылады нод. Публікацыі блокаў абмежаваная па хуткасці ў PoW, таму гэта абмяжоўвае стварэнне акаўнтаў і транзакцый у пэўнай ступені. Ноды, якія не з'яўляюцца поўнымі гістарычнымі вузламі, могуць абрэзаць акаўнты ніжэй статыстычнай метрыкі, дзе рахунак, хутчэй за ўсё, не працоўны. Нарэшце, RaiBlocks настроены на выкарыстанне мінімальнай пастаяннай прасторы для захоўвання, таму прастора, неабходная для захоўвання аднаго дадатковага акаўнта, прапарцыйны памеры адкрытага блока + індэксаванне = 96B + 32B = 128B. Што дазваляе ў 1 ГБ захоўваць за 8 мільёнаў акаўнтаў за кошт пені. Калі вузлы захочуць абрэзаць больш агрэсіўна, яны могуць разлічыць размеркаванне на аснове частоты доступу і дэлегаваць нячаста выкарыстоўваныя акаўнты для больш павольнага захоўвання.

E. Прадвылічальная Атака PoW

Таму што ўладальнік акаўнта будзе адзінай асобай, дадавалым блокі ў ланцуг акаўнта, паслядоўныя блокі можна вылічыць разам з іх PoW, перш чым перадаваць іх у сетку. Тут зламыснік генеруе мноства паслядоўных блокаў, кожны з якіх мае мінімальнае значэнне, на працягу доўгага перыяду часу. У пэўны момант зламыснік выканае Denial of Service (DoS) шляхам флуда сеткі з вялікай колькасцю дапушчальных транзакцый, якія іншыя ноды будуць старацца апрацаваць як мага хутчэй. Гэта ўдасканаленая версія флуда ў транзакцыях, апісаных у раздзеле V-B. Такая атака будзе дзейнічаць нядоўга, але можа выкарыстоўвацца ў спалучэнні з іншымі нападамі такімі як >50% Напады (раздел V-F) для павышэння эфектыўнасці. У цяперашні час высвятляюцца абмежаванні хуткасці перадачы і іншыя метады для змякчэння нападаў.

F. >50% Атака

Паказчыкам кансэнсусу для RaiBlocks з'яўляецца ўзважаная сістэма галасавання. Калі зламыснік можа набраць больш за 50% галасоў, гэта можа прывесці да таго, што сетка будзе вагацца ў выніку збою сістэмы. Зламыснік можа знізіць суму балансу, якую яны павінны страціць, не дапушчаючы, каб добрыя ноды галасавалі выкарыстоўваючы рэалізацыю сеткавай DoS. RaiBlocks прымае наступныя меры для прадухілення такога нападу:

1) Першасная абарона ад такога тыпу атакі - гэта вага галасавання, прывязаны да інвестыцый у сістэму. Уладальнік акаўнта па сваёй сутнасці стымулюецца падтрымліваць сумленнасць сістэмы для абароны сваіх інвестыцый. Спроба змяніць ledger будзе разбуральнай для сістэмы ў цэлым, якая разбурыць і іх інвестыцыі.
2) Кошт такога нападу прапарцыйная рынкавай капіталізацыі RaiBlocks. У сістэмах PoW можна прыдумаць тэхналогію, якая дае непрапарцыйны кантроль у параўнанні з грашовымі ўкладаннямі, і калі атака будзе паспяховай, гэтую тэхналогію можна накіраваць на іншыя мэты. У RaiBlocks кошт атакі на сістэму залежыць ад самой жа сістэмы і, калі атака павінна быць паспяховай, інвестыцыі ў атаку не могуць быць адноўлены.
3) Каб захаваць максімальны кворум тых, хто галасуе, наступная лінія абароны з'яўляюцца рэпрэзентатыўнае галасаванне. Уладальнікі акаўнтаў, якія не могуць удзельнічаць у галасаванні па прычынах падлучэння да сеткі, могуць прыхначыць прадстаўніка, які будзе галасаваць з вагой іх балансу. Максімізацыя ліку прадстаўнікоў павышае ўстойлівасць сеткі.
4) Форкі ў RaiBlocks ніколі не бываюць выпадковым, таму ноды могуць прымаць правілы аб тым, як узаемадзейнічаць з раздвоенымі блокамі. Адзіны раз, калі неатакуючыя акаўнты ўразлівыя для блакавання форк - калі яны атрымліваюць баланс ад атакавалага акаўнта. Акаўнты, якія хочуць быць абароненымі ад блокавых форк, могуць пачакаць некаторы час, перш чым атрымаць ад акаўнта, які згенераваў форк, ці ніколі не атрымліваць наогул гэты блок. Атрымальнікі таксама могуць стварыць асобныя акаўнты, якія будуць выкарыстоўвацца пры атрыманні сродкаў з сумніўных акаўнтаў, каб ізаляваць іншыя свае акаўнты.
5) Апошняя мера абароны, якая яшчэ не рэалізаваная - block cementing. RaiBlocks робіць усё магчымае, каб хутка ліквідаваць блок-форкі шляхам галасавання. Ноды могуць быць настроены для

block cementing, што прадухіліць іх адкат пасля пэўнага перыяду часу. Сетка дастаткова абаронена шляхам факусоўкі на хуткае час прадухілення неадназначных форкі.

Больш складаная версія атакі > 50% падрабязна апісана на рисунке 9. «Offline» - гэта працэнт прадстаўнікоў, якія знаходзяцца не анлайн ў момант галасавання. «Stake» - гэта сума інвестыцыі, з якой зламыснік галасуе. «Active» - гэта прадстаўнікі, якія знаходзяцца ў рэжыме онлайн і галасуюць па пратаколе. Зламыснік можа кампенсаваць суму, якую яны павінны страціць, выбіўшы іншых, хто галасуе ў аўтаномным рэжыме праз сеткавую DoS- атаку. Калі гэтая атака будзе ўстойлівай, атакаванага прадстаўнікі стануць несінхранізаваныя і гэта дэманструе «Unsync». Нарэшце, зламыснік можа атрымаць кароткі разрыў у адноснай сілы галасавання, пераключыўшы атаку DoS для новага набор прадстаўнікоў, у той час як стары набор паўторна сінхранізуе свой ledger, гэта дэманструе «Attack».

Offline	Unsync	Attack	Active	Stake

Рыс. 9. Патэнцыйны механізм галасавання, які можа знізіць патрабаванні да атакі на 51%.

Калі зламыснік можа выклікаць Stake> Active шляхам аб'яднання гэтых абставінаў, ён зможа паспяхова перакуліць галасы ў ledger за кошт сваёй долі. Мы можам ацаніць, наколькі гэты тып атакі дарог, даследуючы рынкавую капіталізацыю іншых сістэм. Калі мы выкажам здагадку, што 33% прадстаўнікоў знаходзяцца ў offline рэжыме або атакаваныя праз DoS, зламысніку неабходна будзе купіць 33% ад рынкавай капіталізацыі, каб атакаваць сістэму шляхам галасавання.

G. Bootstrap Poisoning

Чым даўжэй зламыснік будзе захоўваць стары закрыты ключ з балансам, тым вышэй верагоднасць таго, што балансы, якія існавалі ў той час, не будуць мець прадстаўнікоў, таму што іх балансы або прадстаўнікі перайшлі на больш новыя акаўнты. Гэта азначае, што калі нода загружае старога прадстаўніка сеткі, дзе зламыснік мае большасць для галасавання ў параўнанні з прадстаўнікамі ў той момант, яны змогуць вар'іраваць рашэння аб галасаванні на гэтай нодзе. Калі гэты новы карыстальнік хоча ўзаемадзейнічаць з кім-небудзь, акрамя атакуючай ноды, усе яго транзакцыі будуць адхіляцца, так як яны маюць розныя загалоўныя блокі. Канчатковым вынікам з'яўляецца тое, што ноды могуць марнаваць час на новыя вузлы ў сетцы, перадаючы ім дрэнную інфармацыю. Каб прадухіліць гэта, ноды могуць быць спалучаныя з зыходнай базай дадзеных акаўнтаў і з добра вядомымі галоўнымі блокамі, гэта замена для загрузкі базы дадзеных аж да блока генезісу. Чым бліжэй загрузка будзе да актуальнай версіі, тым вышэй верагоднасць поўнай абароны ад гэтай атакі. У рэшце рэшт, гэтая атака, верагодна, не горш, чым загрузка непажаданых дадзеных у ноды пры загрузцы блокаў, паколькі яны не змогуць здзяйсняць здзелкі з кім-небудзь, у каго ёсць актуальная база дадзеных.

VI. РЭАЛІЗАЦЫЯ

У цяперашні час даведачная выкананне рэалізавана на C ++ і выпускае рэлізы з 2014 года на Github [10].

A. Асаблівасці дызайну

Рэалізацыя RaiBlocks адпавядае стандарту архітэктуры, апісанаму ў гэтым артыкуле. Дадатковыя характарыстыкі апісаны тут.

1) Алгарытм подпісу: RaiBlocks выкарыстоўвае мадыфікаваны алгарытм эліптычнай крывой ED25519 з Хэшаванне Blake2b для ўсіх лічбавых подпісаў [11]. ED25519 быў абраны для хуткай подпісы, хуткай праверкі і высокай бяспекі.

2) Алгарытм Хэшавання: Паколькі алгарытм хэшавання выкарыстоўваецца толькі для прадухілення спаму сеткі, выбар алгарытм менш важны ў параўнанні з крыптавалютамі заснаваных на майнінге. Наша рэалізацыя выкарыстоўвае Blake2b як лічбавы алгарытм супраць змесціва блока [12].

3) Функцыя дэрывацыі ключа: У кашальку ключы шыфруюцца паролем, а пароль перадаецца праз функцыю дэрывацыі ключоў для абароны ад спробаў узлому ASIC. У цяперашні час Argon2 [13] з'яўляецца пераможцам адзінага публічнага конкурсу, накіраванага на стварэнне ад збояў функцыі дэрывацыі ключей.

4) Блочны інтэрвал: Таму што кожны акаўнт мае свой ўласны ланцуг, абнаўлення могуць выконвацца асінхронна са станам сеткі. Таму інтэрвалаў паміж блокаў няма і транзакцыі могуць быць апублікаваны імгненна.

5) Пратакол паведамленняў UDP: Наша сістэма разлічана на бестэрміновую працу з мінімальным аб'ёмам вылічальных рэсурсаў. Усе паведамленні ў сістэме былі распрацаваны такім чынам, каб яны былі без стану і змяшчаліся ў адзін пакет UDP. Гэта таксама палягчае для палегчаных піроў з перарывістым падключэннем ўдзелу ў сеткі без паўторнага аднаўлення кароткатэрміновых TCP-злучэнняў. TCP выкарыстоўваецца толькі для новых вузлоў, калі яны хочуць загрузіць ланцугі блокаў масавым спосабам.

Ноды могуць быць упэўнены, што іх транзакцыі паступілі ў сетку пры выкананні транзакцый шырокавяшчальнага трафіку іншымі нодамі, а гэта павінны ўбачыць некалькі копій рэхам якія вярнуліся да сябе.

B. Пратакол IPv6 и multicast

Стварэнне па-над UDP без усталявання злучэння дазваляе ў будучым выкарыстоўваць шматадрасныую

рассылку па IPv6 ў якасці замены традыцыйнага флуда транзакцый і перадачы галасаванняў. Гэта паменшыць спажыванне прапускной здольнасці сеткі і дасць больш гнуткасці правілаў для нод.

C. Прадукцыйнасць

На момант напісання гэтага артыкула сетку RaiBlocks апрацавала 4,2 мільёна транзакцый, атрымаўшы памер блокчэйна памерам 1,7 ГБ. Час транзакцыі вымяраецца ў секундах. Бягучая эталонная рэалізацыя, замеру на SSD, можа апрацоўваць больш за 10 000 транзакцый у секунду, у першую чаргу гэта звязана з абмежаваннем IO.

VII. ВЫКАРЫСТАННЕ РЭСУРСАЎ

Гэта агляд рэсурсаў, якія выкарыстоўваюцца нодой RaiBlocks. Акрамя таго, мы пераходзім да ідэй памяншэння выкарыстання рэсурсаў для канкрэтных выпадкаў выкарыстання. Спрошчаныя ноды звычайна называюцца лёгкімі, абрэзанымі або спрошчанымі для верыфікацыі плацяжу.

A. Сеть

Аб'ём сеткавай актыўнасці залежыць ад таго, наколькі сетку спрыяе здароўю сеткі.

1) Прадстаўнічая: Прадстаўнічая Нода патрабуе максімальных сеткавых рэсурсаў, паколькі яна сочыць за трафікам галасоў ад іншых прадстаўнікоў і публікуе свае ўласныя галасы.

2) Без даверу: Недавераная нода падобна на прадстаўнічую ноду, але з'яўляецца толькі назіральнікам і не ўтрымлівае прадстаўнічага закрытага ключа акаўнтаў, і таксама не публікуе ўласныя галасы.

3) З даверам: Даверaная нода назірае за трафікам галасоў ад аднаго прадстаўніка, якому ён давярае, каб правільна выканаць кансэнсус. Гэта скарачае колькасць ўваходнага трафіку галасавання ад прадстаўнікоў, якія б спасылаліся на гэтую ноду.

4) Лёгкая: Лёгкая нода таксама з'яўляецца давераным вузлом, які адсочвае трафік толькі для акаўнтаў, у якіх ён зацікаўлены, што дазваляе мінімальнае выкарыстанне сеткі.

5) Загрузная: Загрузная нода абслугоўвае частку або ўвесь ledger для нод, якія падлучаны да сеткі. Гэта робіцца праз TCP-злучэнне, а не UDP, з-за з вялікага аб'ёму перадаваных дадзеных.

B. Займанае месца на дыску

У залежнасці ад патрабаванняў карыстальніка розныя канфігурацыі нод патрабуюць розных умоў да захоўвання.

1) Гістарычная: Нода, зацікаўленая ў захаванні поўнай гістарычнай запісы ўсіх транзакцый, запатрабуе максімальнага месцы захоўвання.

2) Бягучый: У сувязі з дызайнам захавання набытых балансаў з блокамі, ноды павінны трымаць толькі апошнія або галоўныя блокі для кожнага акаўнта для таго, каб удзельнічаць у кансэнсусе. Калі Нода не зацікаўленая ў захаванні поўнай гісторыі, яна можа захоўваць толькі галоўныя блокі.

3) Лёгкая: Лёгкая нода не захоўвае дадзеныя з лакальнага ledger, але ўдзельнічае толькі ў сетцы, каб назіраць за дзейнасцю на акаўнтах, у якіх яна зацікаўлена, або, магчыма, ствараць новыя транзакцыі з зачыненымі ключамі.

C. Працэсар (CPU)

1) Генерацыя транзакцый: Нода зацікаўлена ў стварэнні новых транзакцый, якія павінны вырабіць Proof of Work nonce, каб прайсці механізм рэгулявання RaiBlocks. Вылічэнне розных апаратных сродкаў прыведзена ў дадатку A.

2) Прадстаўнічая: Прадстаўнік павінен правяраць подпісы для блокаў, галасоў, а таксама ствараць свае ўласныя подпісы для ўдзелу ў кансэнсусе. Аб'ём выкарыстаных рэсурсаў CPU для прадстаўнічай ноды значна менш, чым пры генерацыі транзакцыі і можа працаваць з любым працэсарам на сучасным кампутары.

3) Назіральная: Назіральная нода не генеруе свае ўласныя галасы. Таму што выкарыстоўваныя рэсурсы на подпіс мінімальныя, патрабаванні да працэсара амаль ідэнтычныя працы з прадстаўнічай нодой.

VIII. ЗАКЛЮЧЭННЕ

У гэтым дакуменце мы прадставілі фреймворк для даверанай, безкамісійнай і з нізкай затрымкай крыптавалюты, якая выкарыстоўвае новую структуру блокрашоткі і дэлегаванага PoS галасавання. Сетка патрабуе мінімальных рэсурсаў, не патрабуе магутнага абсталявання для інтэлектуальнага аналізу дадзеных і можа апрацоўваць высокую прапускную здольнасць транзакцый. Усё гэта дасягаецца за кошт наяўнасці асобных блокаў для кожнага акаўнта, ліквідацыю праблем доступу і неэфектыўнасці глабальнай структуры дадзеных. Мы ідэнтыфікавалі магчымыя атакі ў сістэме і прадставілі аргументы на стаўленне таго, наколькі RaiBlocks устойлівы да гэтых формаў нападаў.

Дадатак A
POW АПАРАТНЫЯ ПАКАЗЧЫКІ

Як згадвалася раней, PoW ў RaiBlocks - гэта для скарачэння сеткавага спаму. Наша рэалізацыя нод забяспечвае паскарэнне, якое можа выкарыстаць перавагі сумяшчальных з OpenCL графічных працэсараў. У табліцы I прыведзена параўнанне прадукцыйнасці розных апаратных сродкаў у рэальных умовах. У цяперашні час парог PoW фіксаваны, але адаптыўны парог можа быць рэалізаваны па меры дасягнення сярэдняй вылічальнай магутнасці.

Табліца I
ПРАДУКЦЫЙНАСЦЬ АБСТАЛЯВАННЯ 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

УДЗЯЧНАСЦЬ

Мы хацелі б падзякаваць Браян П'ю за кампіляцыю і фарматаванне гэтага артыкула.

Спіс літаратуры

[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