Nano: Криптовалутна мрежа без такси

Colin LeMahieu
clemahieu@nano.co

Абстракт—В последно време, големият публичен интерес и ограничената мащабируемост на популярните криптовалути станаха причина за по-дълги транзакции и по-големи такси, доставяйки незадоволителен опит. Затова ние ви представяме Nano, криптовалута с нова мрежова архитектура, наречена block-lattice (блок-решетка), в която всеки акаунт има свой блокчейн, като така се гарантират мигновени транзакции и неограничена мащабируемост. Всеки потребител има собствен блокчейн, който може да се обновява самостоятелно и несинхронно от останалата мрежа и това допринася за бързи транзакции с минимално натоварване на системата. Транзакциите съдържат информация относно баланса на акаунтите, вместо на сумите в тях, позволявайки агресивно съкращаване на базата с данни без да се застрашава сигурността ѝ. Досега, мрежата на Nano е обработила 4.2 милиона транзакции с пълен обем на тефтера от само 1.7GB. Безтаксовите, мигновени транзакции на Nano я издигат на челно място сред криптовалутите за потребителски транзакции.

Термини—криптовалута, блокчейн, Nano, разпределен тефтер, дигитален, транзакция

I. Въведение

СЛЕД въвеждането на Bitcoin през 2009, започна да се наблюдава постепенно изместване от традиционните валути, зад които стоят правителствата и финансовите дружества, към модерни системи на разплащания, базирани върху криптографията, която предоставя начин на съхраняване и пренасяне на средства по сигурен и неподлежащ на доверие (бездоверителен) начин [1]. За да функционира, една валута трябва да се пренася лесно, да е необратима и да има минимални или никакви такси. Увеличилото се време на транзакции, огромните такси и съмнителната мащабируемост на Bitcoin повдигна въпроса за потенциала ѝ да се използва като всекидневна валута.

В този документ ние ще ви представим Nano, нисколатентна криптовалута, построена върху иновативна структура от данни, предлагаща неограничена мащабируемост и никакви такси. По дизайн, Nano има прост и разбираем протокол, чиято единствена цел е да бъде криптовалута с висока производителност. Протоколът може да бъде използван от хардуер с ниска консумация на енергия, като така се позволява на валутата да бъде практична, децентрализирана и най-добрият избор за всекидневно използване.

Статистиките относно криптовалутите, упоменати в този документ, са точни спрямо датата на публикуване.

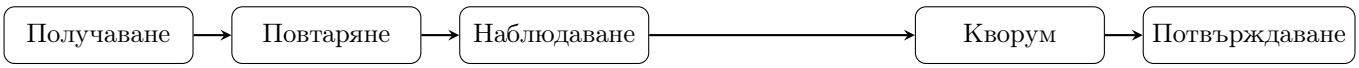
II. Предистория

През 2008 анонимно лице под псевдонима Сатоши Накамото (Satoshi Nakamoto) публикува документ, в който описва първата децентрализирана криптовалута в света, Bitcoin [1]. Ключова иновация, която Bitcoin представя, е блокчейн технологията или непроменяема и децентрализирана структура от данни, която се използва като „тефтер“, включващ всички транзакции на валутата. За съжаление, след като Bitcoin започна да се развива, няколко проблеми в протокола на валутата я възпряха от следните ѝ приложения:

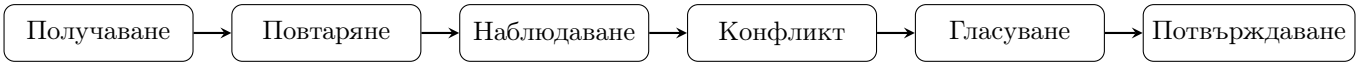
1) Ниска мащабируемост. Всеки блок в блокчейна може да съдържа ограничено количество данни, което означава, че системата може да записва толкова транзакции в секунда, колкото празни места има в блока за записване. В момента на писане средната такса е $10.38 [2].
2) Висока латентност: средното време за потвърждение на един блок е 164 минути [3].
3) Енергична неефективност: мрежата на Bitcoin консумира средно 22.78 млрд. kW·h, използвайки средно 260 kW·h за транзакция [4].

Bitcoin, както и други криптовалути, работят като използват общо съгласие върху техните глобални тефтери, за да потвърдят легитимни транзакции, докато отхвърлят злонамерени такива. Bitcoin постига консенсус чрез икономически мерки, наречени доказателство за работа (PoW). В такава PoW система участниците се конкурират да изчислят число, наречено nonce, по такъв начин, че хешът на целия блок да е в целеви диапазон. Този валиден диапазон е обратно пропорционален на съвкупната изчислителна мощност на цялата Bitcoin мрежа, за да поддържа постоянно средно време за намирането на валиден nonce. Този, който намери валиден nonce, след това може да добави блока към блокчейна; следователно тези, които използват повече изчислителни ресурси за изчислението на nonce, имат по-голяма роля в развитието на блокчейна. PoW предлага устойчивост срещу атака Sybil, където един участник може да се представи като много участници, за да вземе допълнителна мощност в децентрализираната система, и също силно намалява условията на конкуренция, които по начало съществуват, ставайки дума за глобална структура от данни.

Алтернативен протокол на консенсус, наречен доказателство за залог (PoS) бе първо предложен в Peercoin през 2012 [5]. В една такава PoS система участниците трябва да гласуват и тежестта на гласа им зависи от количеството криптовалута, която те притежават. По този начин тези, които имат по-голяма финансова инвестиция, имат по-голяма тежест и по своята същност са длъжни да поддържат честността в системата или



(а) Когато няма конфликт, процесът протича без необходими допълнителни действия.



(б) В случай на конфликт се гласува за валидна транзакция.

Фигура 1. Nano не се нуждае от допълнителни действия при транзакции. В случая на конфликт, възлите гласуват коя транзакция да проведат

рискуват да загубят своите инвестиции. PoS премахва конкуренцията и се възпроизвежда на лек софтуер върху хардуер с ниска мощност.

Оригиналният документ на криптовалутата Nano (RaiBlocks) и първото ѝ бета изпълнение бяха публикувани през декември 2014, правейки я една от първите криптовалути с „Насочен Ацикличен Граф“ (DAG) видове архитектура [6]. Малко след това, други криптовалути DAG започнаха да се разработват, едни от по-известните – DagCon/Byteball и IOTA [7], [8]. Тези криптовалути, базирани на DAG, разбиха стария модел на блокчейна, подобрявайки производителността на системата и сигурността. Byteball постига консенсус, разчитайки на „основна верига“, съдържаща в себе си честни и авторитетни „свидетели“, докато IOTA постига консенсус чрез съвкупни PoW от насъбрани транзакции. Nano постига консенсус чрез претеглени гласове върху конфликтни транзакции. Тази система на консенсус осигурява по-бързи и по-детерминирани транзакции, докато в същото време поддържа силна и децентрализирана система. Nano продължава да се развива и се поставя на едно от челните места в най-добре представящите се криптовалути.

III. Компоненти на Nano

Преди да опишем цялостната архитектура на Nano, ще определим елементите, които съставляват системата.

A. abstract

Акаунтът представлява публичния ключ от двойката-ключ цифров подпис. Публичният ключ, също така наричан и адрес, се споделя с други участници в мрежата, докато скритият (личният) ключ се пази само от собственика. Дигитално подписан пакет от данни гарантира, че съдържанието им е потвърдено от собственика на този скрит ключ. Един потребител може да има много акаунти, но само един публичен адрес може да съществува за акаунт.

Б. Блок/Транзакция

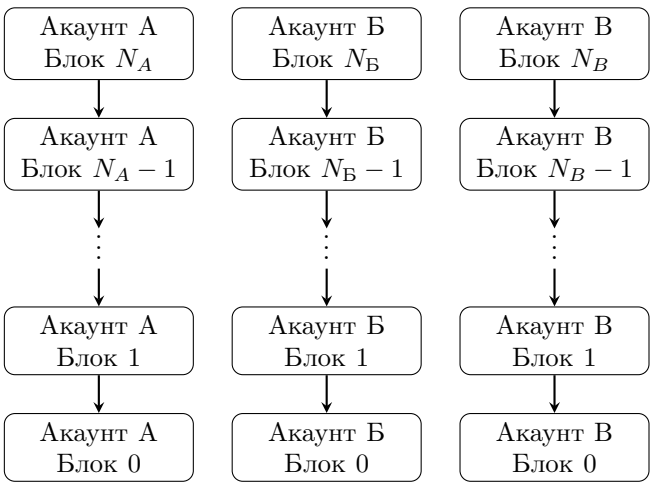
Термините “блок” и “транзакция” често се използват взаимозаменяемо, където блокът съдържа една транзакция. Транзакцията специфично се отнася за самото действие, докато блокът е всъщност дигиталното кодиране/записване на транзакцията. Транзакциите се подписват от скритият ключ, притежател на който е акаунтът, провел транзакцията.

В. Тефтер

Тефтерът е глобален архив от акаунти, в който всеки акаунт има своя собствена верига от транзакции (фиг. 2). Това е ключов елемент от дизайна, който попада в категорията за замяна на споразумението за изпълнение със споразумение за развитие; всеки участва в съгласието чрез подпис, който потвърждава, че само собственикът на акаунта може да променя своята верига. Това конвертира очевидно споделена структура от данни, разпределен тефтер, в такава от несподелена.

Г. Възел (Node)

Възелът или още "node" е част от софтуер, който се изпълнява на компютър, използващ протокола на Nano и участващ в неговата мрежа. Софтуерът отговаря за тефтера и всички акаунти, които възелът би управлявал, ако има такива. Един възел може да съхранява или целия тефтер, или съкратена версия с история, съдържаща само последните няколко блокове на блокчейна на всеки акаунт. Когато се настройва нов възел, е препоръчително първо да се потвърди цялата история, след което да се съкращава локално.



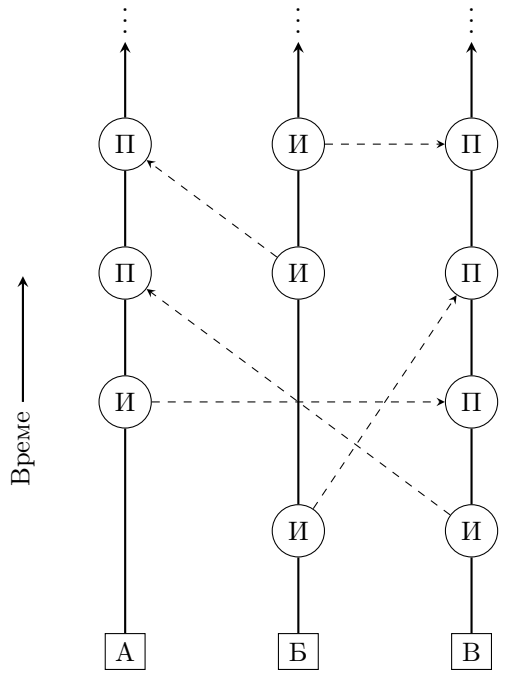
Фигура 2. Всеки акаунт има своя блокчейн, съдържащ историята на баланса на акаунта. Блок 0 трябва да е отворена транзакция (Раздел IV-Б)

IV. Преглед на системата

За разлика от блокчейн технологията, използвана в много други криптовалути, Nano използва структура от блок-решетка. Всеки акаунт има свой собствен блокчейн (от английски „blockchain“ - верига от блокове) или акаунт-верига, еквивалентен на историята от транзакции/баланс в акаунта (фиг.2). Всеки акаунт-верига може да се обновява само от собственика на акаунта; това позволява на всеки акаунт-верига да бъде обновяван моментално и асинхронно от останалата структура от блок-решетка, което допринася за много бързи транзакции. Протоколът на Nano е изключително лек и ненатоварващ, всяка транзакция попада между минималния размер на UDP пакетите за пренасяне в Интернет. Хардуерните изисквания за възел (node) са също минимални, тъй като възлите трябва само да записват и изпращат блокове за повечето транзакции (фиг.1).

Системата е стартирана с генезис акаунт, съдържащ генезис баланс. Генезис балансът е строго определен и не може да бъде увеличен. Генезис балансът е разделен и разпратен до други акаунти чрез изпращащи транзакции, регистрирани на генезис акаунт-веригата. Общата сума на балансите на всички акаунти никога няма да надвиши тази на първоначалния генезис баланс, който поставя горния лимит на баланса на системата без способността да се увеличава.

Този раздел ще ви запознае как различните видове транзакции са построени и разпратени през мрежата.



Фигура 3. Визуализация на блок-решетката. Всеки трансфер на средства изисква изпращащ блок (И) и получаващ блок (П) всеки от които е подписан от собственика на акаунт-веригата (А, Б, В)

A. Транзакции

Пренасянето на средства от един акаунт към друг изисква две транзакции: изпращаща, която взема определената сума от акаунта на изпращача и получаваща, която добавя същата сума към акаунта на получателя (фиг. 3).

Пренасянето на суми в отделни транзакции между акаунтите на изпращача и получателя спомага за следното:

1) Подреждане на входящите транзакции, които са асинхронни.
2) За малки размери транзакции, побиращи се в UDP пакети.
3) Улесняване на скъсяването на тефтера, минимизирайки отпечатъка от данни.
4) Изолиране на установени транзакции от неустановени.

Повече от един акаунт, изпращащ до същия акаунт е асинхронна операция; латентността на мрежата и изпращащите транзакции не комуникират задължително помежду си, което означава, че няма универсален начин на съгласяване коя транзакция е протекла първа. Тъй като добавянето е асоциативно, ходът, по който транзакциите са подредени, не е от значение, затова се нуждаем от глобално съгласие. Това е ключов елемент от дизайна, който конвертира споразумението във време за работа на мрежата във споразумение във време за разработка. Получаващият акаунт има контрол върху избора да определи кой трансфер е пристигнал първи и се изразява чрез подписания ред на входящите блокове.

Ако акаунт иска да направи голям трансфер, който е разделен на много малки трансфери, бихме искали това да се случи по начин, който побира информацията в UDP пакети. Когато получаващ акаунт подреди трансфери, той пази общия си баланс по такъв начин, че по всяко едно време да може да изпрати какъвто и да е размер средства в определен размер на транзакцията. Това се разграничава от модела на транзакции вход/изход при Bitcoin и други криптовалути.

Някои възли не желаят да харчат ресурси за съхраняването на цялостната история на даден акаунт, те се интересуват само от текущия им баланс. Когато акаунт направи транзакция, той записва своя натрупан баланс и тези възли се нуждаят само да следят последния блок, което им позволява да не приемат цели исторически данни, поддържайки така коректността в системата.

Дори и с фокус върху споразумението във време за разработка, съществува малък прозорец от забавяне, когато се потвърждават транзакциите, заради разпознаването и справянето със злонамерени участници в мрежата. След като споразуменията в Nano се изпълняват бързо, в рамките от милисекунди до секунди, можем да представим на потребителя две познати категории от входящи транзакции: установени и неустановени. Установените транзакции са транзакции,

за които даден акаунт е създал получаващи блокове. Неустановените транзакции са тези, които все още не са добавени към общия баланс на получателя. Това е заместител на по-сложните и непознати метрични начини на потвърждение в другите криптовалути.

Б. Създаване на акаунт

За да се създаде акаунт, ще трябва да се направи отворена open транзакция (фиг. 4). Отворена транзакция е винаги първата транзакция на всяка акаунт-верига и може да се създаде след първото получаване на средства. Полето account съдържа публичния ключ (адрес), създаден от личния ключ, използван за вход. Полето source съдържа хеша (hash) на протеклата транзакция. При създаване на акаунт, трябва да се избере представител representative, който да гласува за вас в случай на конфликт; това може да се промени на по-късен етап (Раздел IV-Е). Акаунтът може да избере и себе си за представител.

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

Фигура 4. Структура на отворена транзакция

В. Баланс по акаунт

Балансът по акаунта бива записван в самия тефтер. Вместо да се записва големината на средствата в транзакцията, се проверява (Раздел IV-И) балансовата разлика между изпращащия блок и предишния блок. Получаващият акаунт тогава може да увеличи предишния баланс според финалния баланс, изчислен в новия получаващ блок. Това се прави, за да се подобри скоростта на изчисляване при изтеглянето на голям обем от блокове. Когато се поиска историята на акаунта, информацията е вече предоставена.

Г. Изпращане от акаунт

За да се изпрати от адрес, адресът трябва да има съществуващ отворен блок и следователно баланс (фиг. 5). Полето previous съдържа хеша на предишния блок в акаунт-веригата. Полето destination съдържа акаунта, към който ще се изпратят средства. Веднъж изпратен, изпращащият блок е неотменим. Веднъж изпратени по мрежата, средствата са приспаднати веднага от баланса на изпращащия акаунт и са в статус на изчакване, докато получателят не подпише блока и не приеме тези средства. Средствата в изчакване не трябва да се смятат като такива, които изчакват потвърждение, защото са също толкова валидни, колкото тези, които са вече изпратени от акаунта и изпращачът не може да върне транзакцията.

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

Фигура 5. Структура на изпращаща транзакция

Д. Получаване на транзакция

За да се завърши транзакция, получателят на изпратените средства трябва да създаде получаващ блок на собствената си акаунт-верига (фиг. 6). Полето source съдържа хешът на асоцииранатa изпратена транзакция. След като блокът е създаден и изпратен към мрежата, балансът по акаунта е обновен и средствата са официално преместени в акаунта.

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

Фигура 6. Структура на получаваща транзакция

Е. Настройване на представител

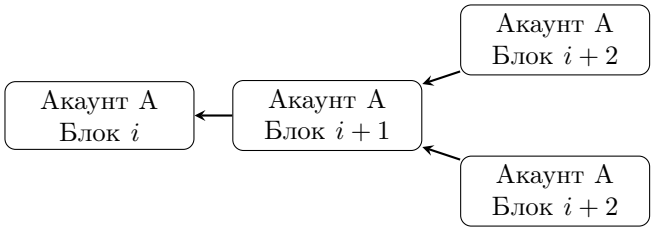
Собствениците на акаунт имат способността да избират представител, който да гласува вместо тях и това е мощен инструмент за децентрализация, който няма аналог в протоколите „доказателство за работа (PoW) и доказателство за залог (PoS)“. В традиционните PoS системи, възелът на акаунта трябва да е онлайн, за да участва в гласуването. Непрестанното работене на тези възли е доста непрактично за много потребители; давайки способността на представител да гласува вместо даден акаунт до определена степен освобождава и улеснява това задължение. Собствениците на акаунт могат да променят представителя по всяко едно време. Сменяща транзакция change променя представителя, отменяйки силата на гласуване от стария представител и връчвайки я на нов представител (фиг. 7). В такава транзакция не се извършва трансфер на средства и представителят няма контрол върху средствата на акаунта.

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

Фигура 7. Структура на сменяща транзакция

Ж. Разклонения и гласуване

Разклонение се получава тогава, когато j-подписани блокове $b_1, b_2, \ldots, b_j$ претендират същия блок за свой предшественик (фиг.8). Тези блокове създават конфликтно мнение върху статуса на акаунта и трябва да се разрешат. Само собственикът на акаунта има силата да подписва блокове в своята акаунт-верига, затова възникналото разклонение трябва да е резултат от недобро програмиране или е направено със злонамерена цел (двойно харчене) от собственика на акаунта.



Фигура 8. Разклонение се получава тогава, когато две (или повече) подписани блокове са свързани с един и същ предшественик. По-старите блокове са отляво, а по-новите – отдясно

Преди откриване, определен представител ще гласува за блок bi в своя тефтер и ще го излъчи към мрежата. Тежестта на гласа на този възел w_i, е сумата от балансите на всички акаунти, които са го избрали за представител. Възелът ще наблюдава пристигащи гласове от други M онлайн представители и ще прецени кумулативно съвпадение от 4 периода на гласуване, които са общо 1 минута, след което ще потвърди печелившия блок (формула 1).

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

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

Най-известният блок b^* ще има мнозинството от гласове и ще бъде запазен в тефтера на възела (формула 2). Блоковете, които губят в гласуването се отписват. Ако представител замени блок в тефтера си, ще създаде нов глас с по-висок пореден номер и ще излъчи новия глас към мрежата. Това е единствения случай, в който представителите гласуват.

В някои случаи, леки спадове в мрежовата връзка може да попречи на даден излъчен блок да не бъде приемен от всички участници. Всеки последващ блок на този акаунт ще бъде игнориран като инвалиден от участниците, невидяли първоначалното му излъчване. Повторно излъчване на същия блок ще бъде прието от останалите участници и последващите блокове ще се възвърнат автоматично. Дори и такова разклонение да се получи или да се пропусне даден блок, само акаунтите, свързани с транзакцията, ще бъдат повлияни; останалата част от мрежата продължава да работи и извършва транзакции за останалите акаунти.

З. Доказателство за работа (PoW)

Всички четири вида транзакции имат работно поле, което трябва да се попълни правилно. Работното поле позволява на създателя на транзакцията да изчисли число nonce така, че хешът, свързан с предишното поле в получаващите/изпращащите/сменящите транзакции или полето на акаунта в отворена транзакция, да е под определен праг. За разлика от Bitcoin, PoW при Nano се използва просто като анти-спам инструмент, подобен на Hashcash, и може да се изчисли в рамките на секунди [9]. След като дадена транзакция е изпратена, PoW за следващия блок може да бъде повторно изчислен, тъй като предишният блок се знае; това спомага транзакциите да изглеждат като мигновени за крайния потребител, стига времето между транзакциите да е повече от времето за извършването на PoW.

И. Потвърждение на транзакция

За да бъде счетен един блок за валиден, то той трябва да има следните атрибути:

1) Блокът не трябва да съществува в тефтера (дублираща се транзакция).
2) Трябва да е подписан от собственика на акаунта.
3) Предишният блок да е по-горе в йерархията на акаунт-веригата. Ако съществува, но не е над/преди него, то тогава се е случило разклонение.
4) Акаунтът трябва да има отворен блок.
5) Изчисленият хеш спазва изискванията на PoW и е над определен праг.

Ако е получаващ блок, проверете дали хешът е в изчакване, в случай, че вече вече не е погасен. Ако е изпращащ блок, балансът трябва да е по-малко от предишния баланс.

V. Вектори за атака

Nano, като всички децентрализирани криптовалути, може да бъде атакувана от злонамерени трети лица с цел финансова изгода или фалит на системата. В този раздел ще разграничим някои възможни сценарии на атаки, възможните последствия и какви мерки са предприети от протокола на Nano.

А. Синхронизиране на пропуски между блокове

В раздел IV-Ж разгледахме сценарий, в който блок може да не бъде правилно излъчен в мрежата, което кара мрежата да игнорира последвали блокове. Ако определен възел наблюдава блок, който няма реферирян предишен блок, то той има два избора:

1) Да игнорира блока, тъй като може да е зловреден.
2) Да поиска ресинхронизация с друг възел.

В случай на ресинхронизация, трябва да се осъществи TCP връзка с вече активен възел, който да улесни увеличилият се трафик, от който се нуждае ресинхронизацията. Въпреки това, ако блокът всъщност е бил зловреден, тогава ресинхронизацията е била ненужна и не е било нужно да се натоварва трафика на мрежата. Това е т. нар. Network Amplification Attack (атака чрез усилване на мрежата) и следва в атака за „отказ на услуга“.

За да се предотврати ненужно ресинхронизиране, възлите ще чакат докато определен брой от гласове определят за вероятно зловреден блок, преди да осъществят връзка с активен възел за синхронизация. Ако даден блок не получи достатъчно гласове, то информацията му се счита за нежелани данни.

Б. Преливане от транзакции

Злонамерени лица могат да изпратят множество ненужни, но валидни транзакции между акаунти под свой контрол в опит да пренасити мрежата. Безтаксовата природа на криптовалутата им позволява да извършват тази атака безкрайно. Въпреки това, изискваната PoW за всяка една транзакция ограничава скоростта на изчисление на злонамереното лице, което не е задължително да инвестира в големи изчислителни ресурси. Дори под такава атака в опит да се „надуе“ тефтера, възлите, които не са цели исторически възли, могат да съкращават стари транзакции от своята верига; това пристяга потреблението на хранилището от тази атака за почти всички потребители.

В. Атака Sybil

Дадено лице може да създаде стотици възли Nano на една машина; въпреки това, след като системата за гласуване е базирана върху баланса по акаунта, добавянето на допълнителни възли в мрежата няма да подари на атакуващия допълнителна гласувателна сила. Следователно няма полза от извършването на атака Sybil.

Г. Атака чрез дребно харчене

Атака от дребно харчене (penny-spend attack) е когато атакуващият харчи минимални стойности от средствата си и ги препраща към огромен брой свои акаунти с цел да изхаби съхраняващите способности на възлите. Публикуването на блокове е ограничено от PoW, така че това ограничава създаването на акаунти до определена степен. Възли, които не са пълни исторически възли, могат да съкращават акаунти под определен статистически фактор, където акаунтът най-вероятно не е валиден. Накрая, Nano се настройва да използва минималното от перманентното съхраняващо пространство, така че пространството, необходимо за съхраняването на още един последващ акаунт, да бъде с размера на отворен блок+индексиране = 96B+32B = 128B. Това се равнява на 1GB пространство, в което могат да се поберат 8 милиона акаунта от този тип атака чрез дребно харчене. Ако възлите пожелаят съкращават по-агресивно, могат да преизчислят дистрибуция, базирана на честота използване и делегиране на рядко използвани акаунти за по-бавно съхранение.

Д. Преизчислена атака PoW

След като собственикът на акаунта ще бъде единственото лице, което ще добавя блокове до акаунт-веригата, последвалите блокове могат да бъдат преизчислени, заедно с тяхната PoW, още преди да са излъчени в мрежата. Тук атакуващият генерира безброй последващи блокове, всеки от които има минимална стойност, в голям период от време. В определен момент, атакуващият извършва атака „отказ на услуга“ – Denial of Service (DoS), натоварвайки мрежата с много валидни транзакции, които другите възли ще приемат и излъчат възможно най-бързо. Това е по-разширена версия на пренасищане с транзакции, описана в раздел V-Б. Такава атака ще работи за кратко време, но може да се използва в тандем с други атаки, като $>50\%$ атака (раздел V-Е), за да увеличи успеваемостта. В момента се разучават начини за ограничаване на транзакциите и други техники за справяне на с атаки.

Е. $>50\%$ Атака

Измерването на консенсуса в Nano е чрез баланс-ориентирана система на гласуване. Ако атакуващ успее да придобие повече от 50% от гласувателната сила, тогава ще може да разколебае консенсуса, по този начин разбивайки цялата система. Атакуващият ще има правото да гласува за зловредни блокове чрез мрежов DoS. Nano предприема следните мерки за предотвратяване на такава атака:

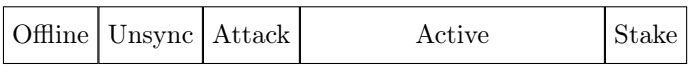
1) Основната защита срещу такъв тип атака е гласовете да бъдат свързани към инвестицията в системата. Собственик на акаунт е стимулиран по подразбиране да поддържа честния дух на системата, за да запази своята инвестиция в нея. Опитът да се преобърне тефтера ще бъде пагубно за цялата система и следователно за вложената инвестиция.
2) Цената на такава атака е пропорционална на пазарната капитализация на Nano. В PoW системите, технологията може да се подобри и това дава непропорционален контрол в сравнение с паричните инвестиции и ако атаката е успешна, тази технология може да се преориентира в други

сфери, след като атаката се извърши. С Nano цената на атакуване на системата се увеличава със самата система и ако атаката е успешна, инвестицията в самата атака не може да бъде възстановена.

3) За да се запази максималният кворум от гласуващи, следващата защита бива гласуване от представители. Собственици на акаунти, които не могат да участват в гласуването заради проблеми с връзката, могат да изберат представител, който да гласува със силата на техния баланс. Максимизирането на броя гласуващи и разновидността от представители увеличава резистентността на мрежата.
4) Разклоненията в Nano не са случайни, затова различните възли могат да стигнат до консенсус как да взаимодействат в разклонени блокове. Единственият период, в който не-атакуващи акаунти са уязвими, е когато получат баланс от атакуващ акаунт. Акаунти, които искат да са защитени от разклонения на блокове, могат да изчакат известно време, преди да получат средства от акаунт, който е предизвикал разклонения или въобще да не получават средства. Получаващите могат също да генерират други акаунти, за да използват при получаване на средства от съмнителни акаунти с цел да изолират останалите си акаунти.
5) 5) Последна мярка на защита, която все още не е въведена, е циментиране на блок или от английски - „block cementing“. Nano стига до големи решения, за да разреши бързо блоковите разклонения чрез гласуване. Възлите също могат да бъдат конфигурирани да „циментират блокове“, което ще ги предпази от това да бъдат върнати след определен период от време. Мрежата е достатъчно предпазена чрез фокусиране върху бързо време на разрешаване, за да предотврати двусмислени разклонения.

По-изискана версия на $> 50\%$ атака е описана във фигура 9. „Offline“ е процентът от представители, които могат да гласуват, но не са на линия да го направят. „Stake“ е инвестицията, с която атакуващият разполага и гласува. „Active“ са представители, които са онлайн и гласуват, спрямо протокола. Атакуващ може измести количеството залог, който трябва да фалшифицират, изкарвайки останалите гласуващи офлайн чрез мрежова DoS атака. Ако тази атака бъде продължителна, атакуваните представители ще се десинхронизират и това се демонстрира чрез „Unsync“. Накрая, атакуващият ще придобие краткотраен прилив на гласуваща сила, превключвайки своята атака от отказ на услуга към нов сет от представители, докато старият сет ресинхронизира своя тефтер, това е показано с „Attack“.

Ако атакуващ успее да обърне Stake >Active чрез комбинация от тези обстоятелства, тогава ще може да обърне гласовете в тефтера на цената на своя залог.



Фигура 9. Потенциален модел на споразумение за гласуване, който може да намали изискванията за атака от 51%.

Можем да определим колко ще струва тази атака, като разгледаме пазарната капитализация на други системи. Ако сметнем 33% от представителите, че са офлайн или атакувани чрез DoS, атакуващото лице ще трябва да закупи 33% от пазарната капитализация, за да атакува системата чрез гласуване.

Ж. Bootstrap Poisoning

Колкото по-дълго атакуващият държи стар скрит ключ с баланс, толкова по-голям е шансът от балансите, съществували по това време да нямат активни или участващи представители, защото техните баланси ще са преместени в по-нови акаунти. Това означава, че ако възел е вързан към стар представител на мрежа, в която атакуващият има кворум от гласуващ залог, за разлика от представителите от този период от време, то тогава атакуващият ще може да разколебае решенията от гласовете в този възел. Ако този нов потребител иска да взаимодейства с всеки, но не и с атакуващия възел, всички от неговите транзакции ще бъдат отхвърлени, тъй като имат различни начала на блоковете. Резултатът е, че възлите могат да пилеят времето на нови възли в мрежата, давайки им невалидна информация. За да се предотврати това, възлите могат да се държат към първоначална база от данни с акаунти и познати добри блокове; това заменя нуждата от теглене на цялата база от данни чак до първоначалния (генезис) блок. Колкото по-близко е изтеглянето до текущия блок, толкова по-голям е шансът от адекватна защита срещу тази атака. Накрая, тази атака вероятно не е по-различна от това да се споделя невалидна информация към възли, тъй като те няма да имат възможността да обменят с когото и да било, имащ съвременна база от данни.

VI. Имплементация

В момента реферираната имплементация е въведена в C++ и произвежда нови версии от 2014 насам в Github [10].

А. Функции

Имплементацията на Nano се придържа към архитектурата, описана в този документ. Останалите специфики са описани по-долу.

1) Алгоритъм на подписване: Алгоритъм на подписване: Nano използва модифициран алгоритъм на елиптичната крива ED25519 с хешинг Blake2b за всички дигитални подписи [11]. ED25519 бе избран за бързо подписване, бързо потвърждаване и висока сигурност.

2) Алгоритъм на хешинг: Тъй като алгоритъма на хешинг се ползва само за предотвратяването на спам, изборът на алгоритъм е по-маловажен от избора на алгоритъм при миниранитe криптовалути. Нашата имплементация използва Blake2b като алгоритъм за усвояване срещу съдържанието на блока [12].

3) Получаване на ключ: В реферирания портфейл, ключовете са криптирани с парола и паролата преминава през функция за деривация на ключове, за да се защити срещу пробив от ASIC. Победител от единствената публична надпревара е Presently Argon2 [13], създавайки еластична функция за деривация на ключове.

4) Интервал на блокове: Тъй като всеки акаунт има свой блокчейн, обновления могат да се правят асинхронно от мрежата Следователно, няма интервал, през който блоковете се създават, което спомага за мигновени транзакции.

5) Протокол от съобщения UDP: Нашата система е създадена да работи безкрайно с минимални изисквания за сила на процесора. Всички съобщения в системата са създадени да бъдат без залог и да могат да бъдат побрани в един UDP пакет. Това също го прави по-лесно за леки и малки участници в мрежата да участват в нея без да преустановяват TCP връзки. TCP се ползва само за нови участници, когато искат да държат блоковите вериги в натрупан модел.

Възлите могат да са сигурни, че техните транзакции са приети от мрежата, наблюдавайки други възли, като видят няколко копия, върнати обратно към тях.

Б. IPv6 и Multicast

Изграждането върху безконтактния UDP позволява бъдещи версии да използват IPv6 multicast като заместител на традиционните излъчвания от транзакции и гласувания. Това ще намали This will reduce консумацията на широчина на честотната лента на мрежата и ще даде по-свободна политика на възлите да продължават напред.

В. Производителност

В момента на писане на този документ, 4.2 милиона транзакции са извършени от мрежата на Nano, правейки размера на тефтера 1.7GB. Времето за транзакция се измерва в рамките на секунди. Текущата реферирана имплементация, работеща върху стокови SSD може да произведе на 10,000 транзакции за секунда, като основно е IO свързана.

VII. Външни препратки

Това е оглед на ресурсите, използвани от един възел на Nano. В допълнение, разглеждаме и някои идеи за намаляване на мястото за съхранение при определени случаи. Намалените възли по принцип се наричат леки, съкратени или опростени възли за платежно потвърждение -simplified payment verification (SPV) възли.

А. Мрежа

Големината на активността на мрежата зависи от това колко мрежата допринася здравето на мрежата.

1) Представителен: Представителният възел използва най-много ресурси, тъй като наблюдава трафика между други представители и преценя гласовете.

2) Бездоверителни: Бездоверителен възел е подобен на представителен възел, но е само наблюдател, не съдържа скрития ключ на представителя и не гласува.

3) Доверчиви: Доверчив възел наблюдава трафика от гласове от един представител, на който се е доверил, за да извърши правилно консенсус. Това се свежда до броя от входящия трафик от гласове към този възел.

4) Леки: Лек възел е също и доверчив възел, който само наблюдава трафика на акаунти, в които позволява минимална използваемост на мрежата.

5) Вързани: Вързан възел възпроизвежда части или целия тефтер за възли, които тепърва се включват онлайн. Това се прави през TCP връзка, вместо през UDP, тъй като включва голям брой от данни, изискващи посложен контрол на потока.

Б. Капацитет на диска

В зависимост от различните потребления на потребителя, различните възли изискват различни конфигурации.

1) Исторически: Историческите възли ще се нуждаят от максимално дисково пространство, за да пазят цялата история на тефтера.

2) Текущи: Тъй като тези пазят само натрупаните баланси с блокове, възлите се нуждаят да пазят само последните или главните блокове за всеки акаунт, за да участват в консенсуса. Ако възелът е незаинтересован да пази цялата история, може да премине към пазене само на главните блокове.

3) Леки: Леките възли не пазят никакви данни от тефтера и само участват в мрежата, наблюдават активността в акаунти, в които са заинтересовани и провеждат транзакции със скрития ключ на акаунта, който притежават.

В. Процесор (CPU)

1) Генериране на транзакция: Възел, желаещ да създаде нова транзакция трябва да създаде число nonce чрез доказателство за извършена работа - Proof of Work (PoW), за да премине успешно през дроселиращи механизъм на Nano. Направен е бенчмарк с различен хардуер за изчисление в приложение А.

2) Представител: Представителят трябва да потвърждава подписите за блокове, гласове и също да създава свои собствени подписи, за да участва в консенсуса. Броят на използваните CPU ресурси за представителен възел са сравнително по-малко от тези за генериране на транзакция и би трябвало да работят при всеки процесор в потребителски компютър.

3) Наблюдател: Наблюдателен възел не генерира свои собствени гласове. Тъй като използването на ресурси за подписване е минимално, изискванията за процесор са почти идентични до тези за представителен възел.

VIII. Заключение

В този документ представихме структурата на бездоверителна, безтаксова криптовалута с ниска латентност, която използва нововъведена блокова архитектура и делегирано гласуване чрез доказателство за залог – Proof of Stake (PoS). Мрежата изисква минимални ресурси и никакви мощни хардуери за миниране, може в същото време и да провежда висок поток от транзакции. Всичко това е постигнато чрез индивидуални блокчейн вериги за всеки акаунт, елиминирайки проблеми с достъпа и неефективността на глобалната структура от данни. Определихме възможните вектори на атака на системата и представихме аргументи за резистентност на Nano срещу тази форма от атаки.

Приложение А
PoW Бенчмарк на хардуер

Както бе упоменато по-горе, PoW в Nano се използва за намаляването на спам в мрежата. Нашата имплементация на възли има предимство при видеокарти, поддържащи OpenCL. Таблица I показва реално проведен бенчмарк на различен хардуер. Засега прагът на PoW е фиксиран, но променлив праг е вероятно да бъде въведен, когато средно-статистическата скорост на процесорите прогресира.

Таблица I
Производителност на хардуер PoW

Хардуер	Транзакции в секунда
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

Благодарности

Бихме искали да благодарим на Браян Пю (Brian Pugh) за компилирането и форматирането на този документ.

Литература

[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