Nano: O rețea răspândită de monedă digitală fără taxă

Colin LeMahieu
clemahieu@nano.co

***Rezumat*—Recent, creșterea cererii și scalabilitatea limitată a crescut timpul obișnuit de tranzacționare și taxele binecunoscutelor criptomonede, rezultând astfel în experiențe nesatisfăcătoare. Aici introducem Nano, o criptomonedă cu un nou algoritm arhitectural, unde fiecare cont deține propriul registru, livrând astfel transacții aproape instantanee ca viteză și scalabilitate nelimitată. Fiecare utilizator are propriul registru, care îi permite să actualizeze în mod ascrionizat la restul rețelei, rezultând astfel în tranzacții rapide cu cheltuieli generale minime. Transacțiile țin cont de balanța contului în loc de cantitatea tranzacțiilor, permițând astfel o prelucrare a bazei de date mai agresivă fără însă a compromite securitatea. Până la această dată, rețeaua Nano a procesat 4.2 milioane de tranzacții cu un spațiu netrunchiat de doar 1.72GB. Tranzacțiile fără de taxă și la secundă a tehnologiei Nano o transformă principala criptomonedă pentru tranzacțiile consumatorilor.**

***(Index Termeni*—criptomonedă, rețea tip blockchain, Nano, bază de date distribuită, digital, tranzacții**

I. Introducere

DE la implementare Bitcoin in 2009, s-a început îndepărtarea de la moneda tradițională sau cea susținută de către autorități și a sistemului financiar în general, migrându-se astfel spre sisteme de plăți mai moderne, bazate pe criptografie, care dă abilitatea de a înmagazina și a transfera fonduri într-un mod securizat și de încredere [1]. Pentru a funcționa într-un mod eficient, moneda trebuie să fie ușor de transferat, nereversibilă și care să aibă o taxă minimă sau inexistentă. Creșterea timpului de tranzacționare, marile taxe și scalabilitatea rețelei îndoielnice au început să ridice semne de întrebare cu privire la practicabilitatea Bitcoin pentru tranzacțiile obișnuite.

În acest document, introduceți Nano, o criptomonată ușor construită pe o structură de rețea de date de inovație care oferă scalabilitate nelimitată, fără taxe tranzacționare. Nano, prin concept este un protocol simplu cu scopul principal de a fi criptomonată cu perfomanțe superioare. Printeaza Nano poate rula hardware-ul de joasă putere, permițând-i să fie practică, descentralizată monedă pentru utilizări zilnice.

Statisticile criptomonelor raportate în acest document sunt exacte ale datei acestei publicații.

II. Fondul

În 2008, un individ anonim sub pseudonimul Satoshi Nakamoto a publicat o foaie albă care evidențiază prima criptomonedă descentralizată din lumii, Bitcoin [1]. O inovație cheie introdusă de către Bitcoin a fost rețeaua de tip blockchain, o structură de date publică, neschimbătoare și descentralizată care este utilizată ca un registru al tranzacțiilor valutare. Din păcate, odată ce Bitcoin a ajuns la maturitate, mai multe probleme din protocol au făcut ca Bitcoin să fie interzis pentru mai multe aplicații:

1) Scalabilitate slabă: Fiecare bloc din blockchain poate stoca o cantitate limitată de date, ceea ce înseamnă că sistemul poate procesa o cantitate limitată de multe tranzacții pe secundă, făcând loc într-un bloc unui produs. În prezent, taxa mediană de tranzacție este de $10.38 [2].
2) Latență ridicată: Timpul mediu de confirmare este 164 minute [3].
3) Energie ineficientă: Rețeaua Bitcoin consumă o valoare estimate 27,28 TWh pe an, folosind în medie 260 KWh pe tranzacție [4].

Bitcoin, și alte criptomonede, funcționează prin atingerea unui consens asupra contabililor lor globali pentru a verifica tranzacții veritabile concomitant ținând frâu participanților rău intenționați. Bitcoin atinge un consens printr-o măsură economică numită Dovada muncii (POW). Într-un sistem de tip PoW participanții concurează pentru calcula un număr, numit un *nonce*, astfel încât hash-ul întregului bloc este într-un interval țintă. Această gamă valabilă este invers proporțională cu puterea de calcul cumulativă a întregei rețele Bitcoin pentru a menține o durată medie constantă pentru găsirea unui nonce valid. Descoperitul unui nonce valid îi este apoi permis să adăuge un bloc lanțului de blocuri; prin urmare, cei ce epuizaeazăă mai multe resurse computaționale pentru a calcula un nonce joacă un rol mai important în cadrul lanțului de blocului. PoW oferă rezistență împotriva unui atac Sybil, unde o entitate se comportă ca entități multiple pentru a obține o putere suplimentară într-un system descentralizat, și, de asemenea, reduce foarte mult condițiile de rasă care există în mod inerent în timp ce accesează o structură globală de date.

Un protocol de consens alternativ, Dovada de Acțiuni (PoS), a fost introdus pentru prima dată de Peercoin în 2012 [5]. Într-un sistem PoS, participanții votează cu o greutate echivalentă cu suma de afacere pe care o posedă într-o criptomonedă dată. Cu acest aranjament, cei care au o investiție financiară mai mare li se dă mai multă putere și sunt în mod inerent stimulați să mențină onestitatea sistemului sau riscă în a-și pierde investiția. PoS se îndepărtează de competiția de calcul a puterii, solicitând doar software-uri ușore care rulează pe un hardware de putere redusă.

Hârtia originală Nano (RaiBlocks) și prima implementare beta au fost publicate în decembrie 2014, devenind una dintre ele primele criptomonede bazate pe Grafice Direcționate acy-

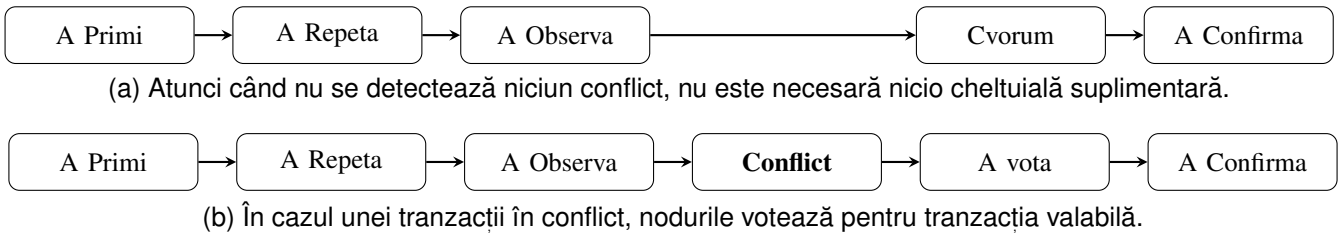

Figura 1. Nano nu necesită cheltuieli suplimentare pentru tranzacții tipice. În cazul tranzacțiilor în conflict, nodurile trebuie să voteze pentru păstrarea tranzacției

clic (DAG) [6]. Curând după aceea, au început să se dezvolte alte criptomonede, cea mai notabilă fiind DagCoin / Byteball și IOTA [7], [8]. Aceste criptomonede bazate pe DAG a schimbat tiparul lanțului de blocuri, îmbunătățind performanța și securitatea sistemului. Byteball atinge consens, bazându-se pe un "lanț principal," alcătuit din "martori" onești, cu reputație bună și care se bucură de încredere utilizatorilor, în timp ce IOTA atinge consens prin intermediul PoW cumulativ al tranzacțiilor acțiunilor. Nano realizează un consens printr-un vot gândit pe tranzacții conflictuale. Acest sistem consensensual oferă soluții mai rapide, tranzacții mai bine determinate, menținând în același timp un sistem puternic și descentralizat. Nano continuă această dezvoltare și s-a poziționat ca unul dintre cele mai performante cryptomonede.

III. Componente Nano

Înainte de a descrie arhitectura generală Nano, definim componentele individuale care alcătuiesc sistemul.

A. Cont

Un cont este partea cheie publică a unei semnături digitale cheie – care este perechea. Cheia publică, denumită și adresa, este impărtășită celorlalți participanți la rețea în timp ce cheia privată este păstrată în secret. Un pachet de date semnat digital asigură că acel conținutul a fost aprobat de titularul de cheie privată. Un utilizator poate controla mai multe conturi, dar numai o singură adresă publică poate exista pe cont. .

B. Block/tranzacție

Termenii "bloc" și "tranzacție" sunt adesea utilizați interschimbabil, unde un bloc conține o singură tranzacție. Tranzacția se referă în mod specific la acțiune în timp ce blocul se referă la codificarea digitală a tranzacției. Tranzacțiile sunt semnate de cheia privată aparținând contului pe care tranzacția este efectuată.

C. Registrul

Registrul este setul global de conturi unde fiecare cont are propriul său lanț de tranzacții (Figura 2). Acesta este un componentă de design cheie care se încadrează în categoria de înlocuire a unui acord de timp de execuție acordul cu un acord de proiectare; toată lumea este de acord prin intermediul semnării pe care numai proprietarul contului o poate modifica la propriul lanț. Aceasta convertește o structură de date aparent împărtășită, un registru distribuit, într-un set de registre neîmpărtășite.

D. Nod

Un *nod* este o parte de software care rulează pe un computer care se conformează protocolului Nano și participanților la rețeaua Nano. Software-ul gestionează registrul și conturile toate conturile pe care nodul le-ar putea controla, dacă este cazul. Un nod poate să stocheze intregul registru sau o istoricul de mecanisme efectuate care contine doar ultimele cateva blocuri din lanțul de blocuri ale fiecărui cont. Când se crează un nod nou se recomandă verificarea întregii istorii și a mecanismului realizat la nivel local.

IV. Prezentare Sistemului

Spre deosebire de blocurile folosite în multe alte criptomonede, Nano folosește o structură *block-lattice* (structură multiplicată). Fiecare cont are propriile sale lanțuri de blocuri (lanț de conturi) echivalent cu istoricul tranzacției/ soldului (Figura 2). Fiecare lanț de conturi poate fi actualizat doar de către proprietarul contului; acest lucru permite fiecărui cont de conturi să fie actualizat imediat și asincron cu restul blocului, rezultând tranzacții rapide. Protocolul Nano este extrem de ușor; fiecare tranzacție se potrivește în dimensiunea minimă a pachetului UDP necesar pentru a fi transmisă pe internet. Cerințele hardware pentru noduri sunt de asemenea minimal, deoarece nodurile trebuie doar să înregistreze și să retransmită blocuri pentru cele mai multe tranzacții (Figura 1).

Sistemul este inițiat cu un *cont de geneză* care conține *soldul genesis*. Soldul genezei este o cantitate fixă și nu poate

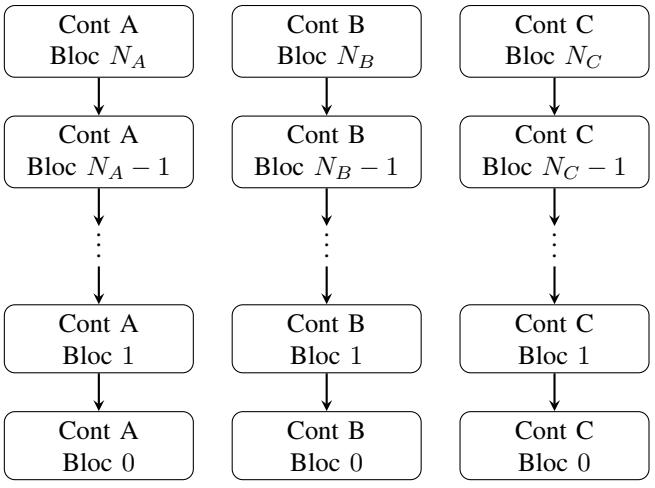

Figura 2. Fiecare cont are propria sa istorie blockchain care conține soldul contului. Blocul 0 trebuie să fie o tranzacție deschisă (Section IV-B)

fi niciodată mărită. Soldul genezei este împărțit și expediat către alte conturi prin intermediul tranzacțiilor de expediere înregistrat pe lanțul de conturi geneză. Suma soldurilor tuturor conturilor nu va depăși niciodată echilibrul inițial de geneză care dă sistemului o limită superioară a cantității și nici o capacitate de creștere a aceasteia.

Această secțiune va trece prin modul în care diferite tipuri de tranzacțiile sunt construite și propagate pe întreaga rețea.

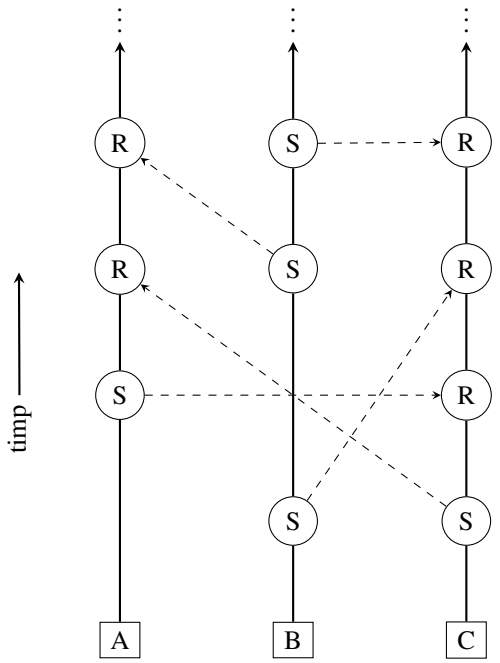

Figura 3. Vizualizarea blocului. Fiecare transfer de fonduri necesită un bloc de expediere (S) și un bloc de primire (R), fiecare semnată de proprietarul lanțului său de cont (A, B, C)

A. Tranzacții

Transferul de fonduri de la un cont la altul necesită două tranzacții: o *trimitere* deducând suma din soldul expeditorului și o *primire* primire care adaugă suma către soldul contului care primește (Figura 3).

Transferarea sumelor ca tranzacții separate în conturile expeditorului și a destinatarului servesc la câteva scopuri importante:

1) Secvențierea transferurilor de intrare care sunt în mod inerent asincron.
2) Menținerea tranzacțiilor mici pentru a se încadra în pachetele UDP.
3) Facilitarea tăierii registrelor prin minimizarea datelor de amprentare.
4) Izolarea tranzacțiilor procesate de cele nerezolvate.

Mai mult de un cont care transferă la aceeași cont de destinație este o operație asincronă; latența rețelei și conturile de trimitere care nu sunt în mod obligatoriu în comunicare unul cu altul înseamnă că nu există un mod universal valabil de a se ști ce tranzacție s-a finalizat mai întâi. Deoarece adăugirea este asociativă, ordinea ăn care intrările sunt secvențiate nu contează, și, prin urmare, avem nevoie pur și simplu de un acord global. Aceasta este o componentă cheie de proiectare care convertește un acord de execuție în acord de design. Contul de primire deține controlul de a hotărâ care transfer a sosit mai întâi și este exprimat de prin ordinul semnat a blocurilor primite.

Dacă un cont dorește să facă un transfer mai mare, dar care a fost primit ca un set de transferuri mici, dorim să le reprezentăm acest lucru într-un mod care se potrivește într-un pachet UDP. Când se primește secvențele de transferuri de intrare în cont, se păstrează un total funcțional al soldul contului, astfel încât, în orice moment, are capacitatea de a transfera orice sumă cu o tranzacție de dimensiune fixă. Acest lucru diferă din modelul tranzacției de intrare / ieșire utilizat de Bitcoin și alte criptomonede.

Unele noduri nu sunt interesate să cheltuiască resurse pentru a stoca istoricul complet al tranzacțiilor unui cont; ele sunt interesate numai de soldul curent al fiecărui cont. Când un cont efectuează o tranzacție, codifică soldul acumulat și aceste noduri trebuie doar să țină evidența ultimului bloc, ceea ce le permite să renunțe la istoricul datelor, menținând în același timp corectitudinea.

Chiar și cu accent pe acordul design-timp, există o fereastră de întârzierea la validarea tranzacțiilor din cauza identificării și manevrarea actorilor răi în rețea. Deoarece acordurile din Nano sunt atinse rapid, de la ordinul milisecundelor la secunde, putem introduce utilizatorulor două categorii familiare din tranzacții primite: soluționate și nesoluționate. S-au efectuat tranzacții sunt tranzacțiile în care un cont a generat primi blocuri. Operațiunile nesoluționate nu au fost încorporate încă în soldul cumulativ al receptorului. Acesta este un înlocuitor pentru confirmările mai complexe și necunoscute în metrice alte criptocuritate.

B. Crearea unui cont

Pentru a crea un cont, trebuie să emiteți o tranzacție *deschisă* (Figura 4). O tranzacție deschisă este întotdeauna prima tranzacție din fiecare lanț de cont și poate fi creată la prima primire din fonduri. Câmpul *contului* stochează cheia publică (adresa) care este derivată din cheia privată utilizată pentru semnare. Câmpului *sursei* conține hash-ul tranzacției care a trimis fondurile. La crearea contului, trebuie ales un reprezentant care să voteze în numele dumneavoastră; acest lucru poate fi modificat ulterior (secțiunea IV-F). Contul se poate declara ca reprezentant propriu.

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

Figura 4. Anatomia unei tranzacții deschise

C. Balanța contului

Soldul contului este înregistrat în registrul propriu-zis. În loc să înregistreze valoarea unei tranzacții, verificarea (Secțiunea IV-I) necesită cererea diferenței dintre balanța la blocul de expediere și soldul precedentului bloc. Contul de primire poate apoi să crească soldul precedent măsurat în soldul final dat în noul bloc primit. Aceasta se face pentru a îmbunătăți viteza de procesare atunci când se descărcă volume mari de blocuri. Când se solicită istoricul contului, sumele sunt deja date.

D. Trimiterea dintr-un cont

TPentru a trimite de la o adresă, adresa trebuie să aibă deja un bloc deschis, și, prin urmare, un sold (Figura 5). Câmpul *anterior* conține hash-ul blocului precedent din cont-chain. Câmpul de *destinație* conține contul pentru care fondurile sunt trimise. Blocul de expediere este imuabil, odată confirmat. Odată difuzate în rețea, fondurile sunt deduse imediat din soldul contului expeditorului și e în *așteptare* până când partea care primește semnează un bloc de accept ale acestor fonduri. Fondurile în așteptare nu ar trebui să fie considerate confirmări în așteptare, deoarece acestea sunt la fel de bune ca cele cheltuite din contul expeditorului și expeditorul nu poate revoca tranzacția.

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

Figura 5. Anatomia unei tranzacții de trimitere

E. Primirea unei tranzacții

Pentru a finaliza o tranzacție, destinatarul fondurilor trimise trebuie să creeze un bloc de primire pe propriul lor lanț de conturi (Figura 6). Câmpul sursă face referire la hash-ul tranzacţiei trimise asociate. Odată ce acest bloc este creat și difuzat, soldul contului este actualizat, iar fondurile s-au mutat official în contul lor.

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

Figura 6. Anatomia unei tranzacții de primire

F. Atribuirea unui reprezentant

Titularii de cont au capacitatea de a alege un reprezentant care să voteze în numele lor este un puternic instrument de descentralizare care nu are nici un analog solid în Dovada muncii sau Dovada mizei protocolare. În sistemele PoS convenționale, nodul proprietarului contului trebuie să se grăbească pentru a participa la vot. Rularea continua a unui nod este nepractică pentru mulți utilizatori; oferind puterea unui reprezentant de a vota în numele unui cont se relaxează cerința. Titularii de cont au abilitatea de a reevalua acceptul către oricare cont la orice moment. O tranzacție *schimbare* schimbă reprezentantul unui cont prin scăderea greutății votului pentru vechiul reprezentant și adăugarea greutății la noul reprezentant (Figura 7). Nu sunt mutate fonduri pentru această tranzacție, iar reprezentantul nu are puterea de a cheltuieli din fondurile contului.

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

Figura 7. Anatomia unei tranzacții de schimbare

G. Forks and Voting

Un furt apare atunci când j blocuri semnate $b_1, b_2, \ldots, b_j$ pretinde același bloc ca și predecesorul lor (Figura 8). Aceste blocuri cauzează o viziune conflictuală asupra stării unui cont și trebuie fi rezolvată. Numai proprietarul contului are abilitatea de a semna blocuri în lanțul lor de conturi, deci furtul trebuie să fie rezultatul unei programări precare sau rău intenționată (cheltuilă-dublă) de către proprietarul contului.

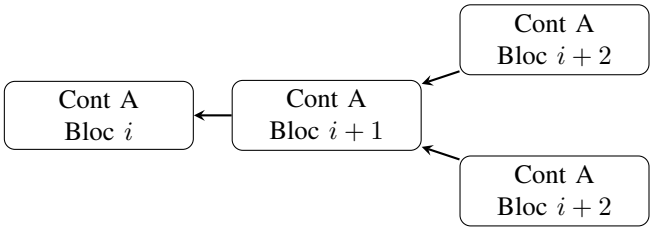

Figura 8. O furculiță apare atunci când două (sau mai multe) blocuri semnate fac referire la același bloc anterior. Blocurile mai vechi sunt pe partea stângă; blocurile mai noi sunt pe dreapta

La detectare, un reprezentant va crea o referință la votul blocul $\hat{b}_i$ în registrul său și îl difuzează în rețea. Ponderea votului unui nod, w_i, este suma soldurilor tuturor conturilor care l-au denumit ca reprezentant. Nodul va observa voturile primite de la ceilalți M reprezentanți online și să păstreze o înregistrare cumulativă pentru 4 perioade de vot, un toatal de 1 minut și confirmă blocul câștigător (Ecuația 1).

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

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

Cel mai popular bloc b^* va avea majoritatea voturilor și va fi păstrat în registrul nodului (Ecuația 2). Blocul (blocurile) care pierd votul este eliminat. Dacă un reprezentant înlocuiește un bloc în registrul său, va crea un nou vot cu un număr de ordine mai mare și va difuza noul vot către rețea. Acesta este **singurul** scenariu în care reprezentanții votează.

În anumite circumstanțe, problemele conectivității scurte la rețea poate cauza ca un bloc difuzat să nu fie acceptat de către toți colegii. Orice bloc ulterior din acest cont va fi ignorat ca invalid de către colegii care nu au văzut difuzarea inițială. O retransmitere a acestui bloc va fi acceptată de restul colegilor și blocurile ulterioare vor fi preluate automat. Chiar și atunci când apare un furt sau un bloc lipsă, doar conturile menționate în tranzacție sunt afectate; restul rețelei realizează tranzacții de procesare pentru toate celelalte conturi.

H. Dovada muncii

AToate cele patru tipuri de tranzacții au un câmp de lucru care trebuie să fie populat corect. Câmpul de lucru permite ca tranzacția creatorlui să calculeze un număr nonce cum ar fi un hash de număr nonce alăturat cu un câmp anterior într-o tranzacție de tip „primește / trimite / schimbă" sau câmpul contului într-o tranzacție deschisă care să foe sub un anumit prag de valoare. Spre deosebire de Bitcoin, PoW din Nano este pur și simplu folosit ca un instrument anti-spam, similar cu Hashcash, și poate fi calculat pe ordinea de secunde [9]. Odata ce tranzacția este trimisă, PoW pentru blocul ulterior poate fi precomprimat, deoarece este cunoscut câmpul blocului anterior; acestea vor face ca tranzacțiile să apară instantaneu unui utilizator final atât timp cât timpul dintre tranzacții este mai mare decât timpul necesar pentru a calcula PoW.

I. Verificarea tranzacțiilor

Pentru ca un bloc să fie considerat valabil, acesta trebuie să aibă următoarele atribute:

1) Blocul nu trebuie să fie deja în registru (tranzacţie duplicată).
2) Trebuie să fie semnat de proprietarul contului.
3) Blocul anterior este blocul principal al lanțuilui de conturi. Dacă există, dar nu este capul, este un furt.
4) Contul trebuie să aibă un bloc deschis.
5) Hash-ul calculat îndeplinește cerința de prag PoW.

Dacă este un bloc de primire, verificați dacă blocul sursă are hash-ul în așteptare, adică nu a fost deja cerut. Dacă este un bloc trimis, soldul trebuie să fie mai mic decât soldul anterior.

V. Vectorii De Atac

Nano, la fel ca toate criptomonedere descentralizate, poate fi atacat de către părțile rău intenționate care încercă să aibă un câștig financiar sau ca sistemul să cedeze. În această secțiune vom sublinia câteva posibile scenario de atac, consecințele unui astfel de atac și cum este protocolul Nano ia măsuri preventive.

A. Blocarea sincronizării blocurilor

În secțiunea IV-G, am discutat scenariul în care un bloc nu pot fi difuzat în mod corespunzător, cauzând ca rețeaua să ignore blocurile ulterioare. Dacă un nod observa că un bloc nu are referință la blocul anterior, are două opțiuni:

1) Ignorarea blocul deoarece ar putea fi un bloc gunoi rău intenționat.
2) Solicitați o resincronizare cu un alt nod.

În cazul unei resincronizări, trebuie să se formeze o conexiune TCP cu un nod începător pentru a facilita creșterea cantității din trafic necesară unei resicronizări. Cu toate acestea, dacă blocul a fost efectiv un bloc rău, atunci resincronizarea nu a fost necesară și a crescut traficul în rețea în mod inutil. Aceasta este un Atac de amplificare rețelei și dă naștere unui refuz al serviciului.

Pentru a evita reacționarea inutilă, nodurile vor aștepta până când un anumit prag de voturi au fost observate pentru un potențial rău înainte de a iniția o conexiune la un nod începător în sincronizare. Dacă un bloc nu primește suficiente voturi se poate presupune că sunt date nedorite.

B. Inundarea tranzacțiilor

O entitate răuvoitoare ar putea trimite multe tranzacții inutile, dar valide între conturile aflate sub controlul său într-o încercare de a duce rețeaua la o saturație. Fără taxe de tranzacție sunt astfel capabili să continue acest atac pe termen nelimitat. Cu toate acestea, PoW cere ca pentru fiecare tranzacție să limiteaze rata de tranzacție pe care entitatea răuvoitoare ar putea-o genera fără a investi în mod semnificativ resurse computaționale. Chiar și sub un astfel de atac într-o încercare de a mări registrul, nodurile care nu sunt noduri istorice complete sunt capabile să taie tranzacțiile vechi din lanțul lor; acesta fixează stocarea utilizată de acest tip de atac pentru aproape toți utilizatorii.

C. Atacul Sybil

O entitate ar putea crea sute de noduri Nano pe un singur aparat; cu toate acestea, deoarece sistemul de votare este ponderat pe baza soldul contului, adăugarea de noduri suplimentare în rețea nu va obține voturi suplimentare pentru un atacator. Prin urmare, nu există nici un avantajul ce poate fi câștigat printr-un atac Sybil.

D. Atac prin cheltuirea monedelor

Un atac de tip penny-spend este cazul în care un atacator cheltuie cantități infinitezimale la un număr mare de conturi pentru a epuiza resursele de stocare a nodurilor. Publicarea blocurilor este limitată de către PoW, ceea ce limitează crearea de conturi și tranzacții într-o anumită măsură. Nodurile care nu sunt noduri istorice mature pot schimba conturile sub o valoare statistică unde contul nu este cel mai probabil un cont

valid. In cele din urma, Nano este reglat să utilizeze un spațiu pemanent de stocare minimal, deci spațiul necesar pentru a stoca un cont suplimentar este proporțional la dimensiunea unui open block + indexing = 96B + 32B = 128B. Acest lucru echivalează cu 1GB fiind capabil să stocheze 8 milioane de monede chltuite de un cont. Dacă nodurile doreau să diminueze mai agresiv, pot calcula o distribuție bazată pe frecvența de acces și să delege conturile utilizate mai puțin frecvent pentru stocarea mai lentă.

E. Atacul prealabil PoW

Deoarece proprietarul unui cont va fi singura entitate care poate adăuga blocuri în lanțul de conturi, blocurile secvențiale pot fi calculate, împreună cu PoW-urile lor, înainte de a fi difuzate la rețea. Aici atacatorul generează o mulțime de blocuri secvențiale, fiecare având o valoare minimă, pe o perioadă extinsă de timp. La un anumit punct, atacatorul execută un Refuz de Servicu (DoS) prin inundarea rețelei cu multe tranzacții valide, pe care alte noduri le vor procesa și le vor replica cât mai repede posibil. Aceasta este o versiune avansată a tranzacție descrisă în secțiunea V-B. Un astfel de atac ar funcționa doar pe termen scurt, dar ar putea fi utilizat împreună cu alte atacuri, cum ar fi un atac de peste >50% (Secțiunea V-F) să crească eficacitate altor atacuri. Tranzacțiile de limitare a ratei precum și alte tehnici sunt în curs de investigare pentru a atenua atacurile.

F. >50% Atac

Consensul de metrică pentru Nano este un sistem de votare cu sold ponderat. Dacă un atacator este capabil să câștige peste 50% din puterea de vot, poate cauza oscilarea consensului în rețea redând un sistem stricat. Un atacator poate reduce soldul de echilibru pe care trebuie să-l sustragă prin prevenirea nodurilor bune de a vot prin intermediul unei rețele DoS. Nano ia următoarele măsuri pentru a preveni un astfel de atac:

1) Principala apărare împotriva acestui tip de atac este greutatea la vot fiind legat de investițiile în sistem. Un titularul de cont este în mod inerent stimulat să mențină onestitatea sistemului pentru a-și proteja investițiile. Încercarea de a răsturna registrul ar fi distrugătoare pentru sistem în ansamblul său, care ar distruge investiția.
2) Costul acestui atac este proporțional cu piața de capitalizarea Nano. În sistemele PoW, tehnologia poate fi inventată care oferă un control disproporționat în comparație cu investițiile monetare și, dacă atacul are succes, această tehnologie ar putea fi refolosită după ce atacul este gata. Cu Nano costul de a ataca crește odată cu sistemul însuși și dacă ar fi ca un atac să fie de succes investiția din acel atac nu poate fi recuperată.
3) Pentru a menține cvorumul maxim al votanților, următoarea linie de apărare este votul reprezentativ. Titularii de cont care nu pot participa în mod fiabil la vot din motive de conectivitate poate numi un reprezentant care poate vota cu ponderea soldului lor. Maximizarea numărul și diversificarea reprezentanților cresc rezistența rețelei.
4) Furturile în Nano nu sunt niciodată accidentale, astfel încât nodurile pot lua deciziile de politică privind modul de interacțiune cu blocurile furate. Singura data în care conturile care nu atacă sunt vulnerabile la blocurile furate este atunci când primesc un sold de la un cont atacant. Conturile care doresc să se ferească de furturile de blocurile pot aștepta puțin sau mult înainte de a primi un cont care a făcut furturi sau pot opta să nu primească niciodată. Receptorii ar putea genera și conturi separate pe care să le folosească atunci când primesc fonduri din conturi dubioase cu scopul de a izola alte conturi.
5) O linie finală de apărare care nu a fost încă implementată este *izolarea blocului*. Nano se extinde foarte mult pentru a soluționa rapid furturile de blocuri prin intermediul votului. Nodurile ar putea fi configurate să blocheze blocurile, ceea ce le-ar împiedica de la revenirea după o anumită perioadă de timp. Rețeaua este suficient de protejată prin focalizare restabilirea rapidă a timpului pentru a preveni furturile ambigue.

O versiune mai sofisticată a unui atac de peste $>50\%$este detaliată în Figura 9. “Offline” este un procentul reprezentanților care au fost numiți, dar nu sunt online pentru a vota. “Miza” este valoarea investiției de care se folosește atacatorul pentru a vota. “Activ” sunt reprezentanții care sunt online și votează în conformitate cu protocolul. Un atacator poate să compenseze suma de mizei pe care o trebuiesă o pierdă prin a induce în offline alegătorii prin intermediul unui atac la rețeaua DoS. Dacă acest atac poate fi susținut, reprezentanții atacați devin nesincronizați și acest lucru este demonstrat de către “Unsync”. În cele din urmă, un atacator poate obține un mic avantaj în puterea votul relativ prin schimbarea atacului DoS cu un nou set de reprezentanți în timp ce vechiul set resincronizează registrul lor, iar acest lucru este demonstrat de “atac”.

Deconectat	Unsync	**Atac**	Activ	țăruș

Figura 9. Un aranjament de vot potențial care ar reduce cerințele de atac de 51%.

Dacă un atacator poate provoca ca Miza >Activă prin combinarea acestor circumstanțe, ei ar fi capabili să reușească cu succes să răstoarne voturi pe registru în detrimentul mizei lor. Putem estima cât de mult ar putea costa acest tip de atac examinând capacitatea de piață a altor sisteme. Dacă estimăm că 33% dintre reprezentanți sunt offline sau atacați prin intermediul DoS, un atacatorul ar trebui să cumpere 33% din plafonul de piață pentru a ataca sistemul prin vot.

G. Otrăvirea începătorilor

Cu cât un atacator poate deține o cheie privată cu debit mai mult, cu atât este mai mare probabilitatea de debitul existent la acel moment nu va avea reprezentanți participanți deoarece soldul sau reprezentanții lor au fost transferați către conturi mai noi. Aceasta înseamnă că dacă un nod este reînoit la o reprezentare veche a rețelei unde un atacator are un cvorumul de mize de votare, comparativ cu reprezentanții acestora în

timp, ar putea să oscileze deciziile de vot la acel nod. Dacă acest nou utilizator dori să interacționeze cu oricine înafară de nodul de atac, toate tranzacțiile lor ar fi respinse deoarece au capuri de blocuri diferite. Rezultatul net este că nodurile pot pierde timpul nodurilor noi din rețea prin întreținerea cu informațiilor lor rele. Pentru a preveni acest lucru, nodurile pot să fie asociate cu o bază de date inițială a conturilor și bine cunoscute capete de bloc; acesta este un subtuitor pentru descărcarea bazei de date de la blocul de geneză. Cu cât descărcarea este mai actuală, cu atât apărarea este mai bună împotriva acestui tip de atac. În final, acest tip de atacul nu este, probabil, mai rău decât alimentarea datelor nefolositoare către noduri în timp ce aceștia sunt în formare, deoarece acestea nu ar fi în măsură să tranzacționeze cu nimeni care are o bază de date actualizată.

VI. Implementare

În prezent, implementarea referințelor este implementată în C ++ și produce versiuni începând cu 2014 pe Github [10].

A. Caracteristici de proiectare

Implementarea Nanro aderă la standadul de arhitectură prezentate în această lucrare. Specificații suplimentare sunt descrise aici.

1) Algoritmul de semnare: Nano utilizează un algoritm curbă ED25519 modificată eliptic cu hashtag Blake2b pentru toate semnăturile digitale [11]. ED25519 a fost ales pentru semnarea rapidă, verificare rapidă și securitate ridicată.

2) Algoritmul Hashing: Deoarece algoritmul hashing este numai utilizat pentru a preveni spam-ul de rețea, alegerea algoritmului este mai puțin importantă în comparație cu criptomonedele cu bază minieră. Implementarea noastră folosește Blake2b ca algoritm de asimilare împotriva conținutului blocului [12].

3) Funcția de Derivare a cheii: În portofelul de referință, tastele sunt criptate de o parolă și parola este alimentată printr-o funcție de derivare a cheii pentru a proteja împotriva unei încercări a ASIC de a o sparge. În prezent, Argon2 [13] este câștigătorul singurului concurs public care vizează crearea unei funcții de derivare a unei cheie rezistente.

4) Intervalul bloc: Deoarece fiecare cont are propriul lanț de blocuri, actualizările pot fi efectuate asincron cu starea reţelei. Prin urmare, nu există intervale de bloc și tranzacțiile pot fi publicate instantaneu.

5) Protocolul de mesaje UDP: Sistemul nostru este conceput pentru a funcționa pe o perioadă nedeterminată, folosind cantitatea minimă de resurse pentru a calcula. Toate mesajele din sistem au fost proiectate a fi apatrid și se încadrează într-un singur pachet UDP. Acest facilitează, de asemenea, ca colegii ușori cu conectivitate intermitentă să participe la rețea fără restabilirea pe termen scurt a Conexiuni TCP. TCP este utilizat numai pentru noii colegi atunci când aceștia doresc să încărce lanțurile de bloc într-o manieră interconectată.

Nodurile pot fi sigure că tranzacția lor a fost primită de către rețea prin observarea traficului de difuzare a tranzacțiilor de la alte rețele noduri, unde ar trebui să vadă mai multe copii, redifuzate către sine.

B. IPv6 și Multicast

Construită deasupra conexiunii puține UDP permite viitoarele implementări pentru utilizarea multicastului IPv6 ca înlocuitor pentru abundența tranzacțiilor tradiționale și difuzarea voturilor. Asta va reduce consumul de lățime de bandă a rețelei și oferă o politică de flexibilitate față de noduri de aici încolo.

C. Performanță

La momentul acestei scrieri, au avut loc 4,2 milioane de tranzacții procesate de către rețeaua Nano, generând un lanț de blocuri de 1,7 GB. Timpii de tranzacție sunt măsurați la ordinul secundelor. O referință actală implementată care funcționează la SSD-urile de marfuri poate procesa peste 10.000 de tranzactii pe secundă fiind în principal legat de IO.

VII. Utilizarea Resurselor

Aceasta este o prezentare generală a resurselor utilizate de un nod nano. În plus, vom trece peste idei pentru reducerea utilizării resurselor pentru cazuri specifice de utilizare. Nodurile reduse sunt de obicei numite lumini, tăiate sau simplificate de nodurile de verificare a plăților (SPV).

A. Rețeua

Activitatea de rețea a unui nod depinde de cât de mult nodul contribuie la sănătatea unei rețele.

1) Reprezentant: Un nod reprezentativ necesită maximum de resurse de rețea, deoarece observă traficul de vot de la alții și publică propriile sale voturi.

2) De neîncredere: un nod de neîncredere este similar cu un nod reprezentant, dar este doar un observator, nu conține un cheie private de cont a reprezentantului și nu publică voturi proprii.

3) De încredere: Un nod de încredere observă traficul de vot de la un reprezentant în care are încredere și care realizează corect consensul. Acest lucru reduce cuantumul traficului de voturi de intrare de la reprezentanții care aderă la acest nod.

4) Lumina: Un nod luminos este, de asemenea, un nod de încredere care numai observă traficul pentru conturile de care acesta este interesat permițând utilizarea minimă a rețelei.

5) Bootstrap: Un nod bootstrap servește părți sau toate registrele pentru nodurile care se aduc online. Aceasta este făcut peste o conexiune TCP, în loc de UDP, deoarece implică o cantitate mare de date care necesită un control avansat al debitului.

B. Capacitatea discului

În funcție de cerințele utilizatorilor, configurații diferite de noduri necesită cerințe de stocare diferite.

1) Istoric: un nod interesat de păstrarea unui istoric complet al înregistrărilor tuturor tranzacțiilor va necesita o cantitate maximă de stocare.

2) Curentul: datorită design-ului de păstrare al debitului acumulat cu blocurile, nodurile trebuie doar să păstreze cele mai recente sau capetele de blocuri pentru fiecare cont pentru a participa consens. Dacă un nod nu este interesat să păstreze un istoric complet se poate opta să păstreze numai capetele de blocuri.

3) Ușor: un nod ușor nu păstrează date de la registrii locali și doar participă la rețea pentru a observa activitatea conturilor despre care este interesat sau opțional să creeze noi tranzacții cu cheile private pe care le deține.

C. CPU

1) Generarea tranzacțiilor: Un nod interesat de crearea tranzacțiilor noi trebuie să producă un PoW de tip nonce pentru a trece de mecanismul de reglaj al lui Nano. Calculul diverselor hardware-ul este notat în Anexa A.

2) Reprezentant: Un reprezentant trebuie să verifice semnăturile pentru blocuri, voturi și, de asemenea, să producă propriile semnături pentru a participa la un consens. Cantitatea de resurse CPU pentru un nodul reprezentativ este semnificativ mai mic decât generarea de tranzacții și ar trebui să lucreze cu orice CPU ale unui calculator actual.

3) Observator: Un nod observator nu generează propriile sale voturi. De vreme ce generare de semnături este minimă, Cerințele procesorului sunt aproape identice cu funcționarea unui nod reprezentant.

VIII. Concluzie

În această lucrare am prezentat cadrul pentru o criptomonedă fără de încredere, lipsit de taxe, cu latență redusă, care folosește o structura vastă, cu structură de tip block și PoS bazat pe vot. Rețeaua necesită resurse minime, fără a necesita o putere mare a hardware-ului, și poate care șă poată procesa tranzacții în număr mare. Toate acestea s-au putut realiza prin folosirea unor lanuțuri de blocuri individuale pentru fiecare cont, eliminând problemele de acces și ineficiențele unui serviciu de structura globală a datelor. Am identificat posibilii vectori de atac ale sistemul și am prezentat argumente privind modul în care Nano este rezistent la aceste forme de atac.

Anexa A
PoW Hardware De Referință

După cum sa menționat anterior, PoW în Nano este pentru a reduce spam –urile de rețea. Implementarea nodului nostru asigură accelerarea de care pot beneficia OpenCL compatabile cu GPU. Tabelul I oferă o comparație reală a referințelor la diferitele componente hardware. În prezent, pragul PoW este fix, dar este adaptabil și poate fi implementat ca putere medie de calcul progresează.

Tabela I
Hardware de performanță PoW

Dispozitiv	Tranzacții pe secundă
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

Recunoştinţă

Am dori să-i mulțumim lui Brian Pugh pentru compilarea și formatarea acestei lucrări.

Bibliografie

[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