Nano: Bezpłatna rozproszona sieć kryptowalutowa

Colin LeMahieu
clemahieu@nano.co

***Streszczenie*—Współcześnie, duży popyt i limitowana skalowalność zwiększyły czas i koszta transakcji popularnych kryptowalut, prowadząc do niezadowalających doświadczeń. Tutaj przedstawiamy Nano - kryptowalutę z innowacyjną architekturą block-lattice, gdzie każde konto ma swój własny łańcuch bloków (blockchain), dostarczając niemal natychmiastowe transakcje i nieskończoną skalowalność. Każdy użytkownik posiada własny łańcuch bloków, który może aktualizować asynchronicznie względem reszty sieci, czego wynikiem są szybkie transakcje bez zbędnych kosztów. Transakcje raczej śledzą balans konta niż kwoty transakcji, pozwalając na agresywne zmniejszenie bazy danych bez narażania na niebezpieczeństwo. Jak dotąd sieć Nano przetworzyła 4.2 miliona transakcji, bez zmniejszania głównego rejestru, o rozmiarze jedynie 1.7GB. Nano jest bez opłat, a transakcje w ułamki sekund czynią je pierwszą kryptowalutą przeznaczoną do transakcji konsumenckich.**

Zbiór określeń - kryptowaluta, łańcuch bloków (blockchain), Nano, główny rejestr (ledger), transakcje

I. WPROWADZENIE

OD czasu wdrożenia waluty Bitcoin w 2009 roku, nastąpiło wyraźne odejście od wspieranych przez rząd, tradycyjnych walut i systemów finansowych na rzecz nowoczesnych systemów płatności opartych o kryptografię, które oferują możliwość przechowywania i transferu funduszy w sposób bezpieczny i niewymagający zaufania osobom trzecim. [1]. Aby waluta działała efektywnie musi być łatwo transferowalna, nieodwracalna i posiadać znikome opłaty lub ich brak. Wydłużone czasy transakcji, ogromne opłaty i kwestionowana skalowalność sieci postawiły znaki zapytania na temat praktyczności Bitcoina jako waluty codziennego użytku.

W ninieszym opracowaniu, przedstawiamy Nano jako kryptowalutę o niskim opóźnieniu, zbudowaną na innowacyjnej strukturze danych block-lattice, oferującą nieograniczoną skalowalność i brak opłat transakcyjnych. Projekt Nano to prosty protokół, którego jedynym celem jest bycie kryptowalutą o wysokiej wydaności. Protokół Nano może pracować na sprzęcie o niskiej mocy, tworząc praktyczą i zdecentralizowaną kryptowalutę codziennego użytku.

Statystyki dotyczące kryptowaluty przedstawione w tym dokumencie są dokładne dla dnia publikacji.

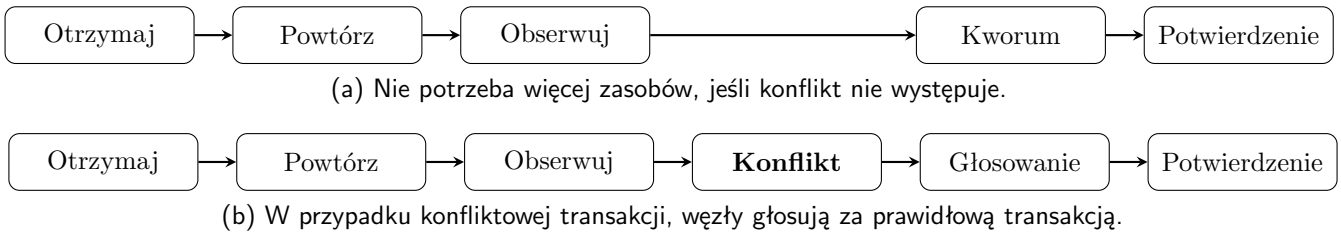
II. PODSTAWY

W roku 2008 anonimowa jednostka pod pseudonimem Satoshi Nakamoto opublikowała zarys nakreślający pierwszą na świecie kryptowalutę: Bitcoin. Kluczową innowacją wprowadzoną przez Bitcoina był łańcuch bloków- publiczna, stała i zdecentralizowana struktura danych używana jako główny rejestr dla kryptowalutowych transakcji. Niestety, gdy Bitcoin dojrzał, pojawiło się kilka problemów z jego nieprzystępnym dla wielu zastosowań protokołem:

1) Słaba skalowalność: Każdy blok w łańcuchu bloków może przechować ograniczoną ilość danych, co oznacza, że system może przetworzyć ograniczoną liczbę transakcji na sekundę, przez co przestrzeń wewnątrz bloku jest towarem. Obecnie mediana kosztów transakcji wynosi $10.38 [2].
2) Duże opóźnienie: Średni czas potwierdzenia transakcji wynosi 164 minuty [3].
3) Niewydajność energetyczna: Sieć Bitcoin wykorzystuje około 27.28TWh na rok, zużywając średnio 260KWh na transakcję [4].

Bitcoin i inne kryptowaluty funkcjonują dzięki osiąganiu konsensusu w ich globalnych rejestrach w celu weryfikowania prawowitych transakcji, jednocześnie opierając się złośliwym atakom. Bitcoin osiąga konsensus poprzez środek ekonomiczny zwanym dowodem wykonanej pracy (Proof of Work: PoW). W systemie PoW uczestniczy rywalizują o obliczenie liczby, zwanej *nonce*, której skrót całego bloku znajduje się w docelowym przedziale. Ten ważny zakres docelowy jest odwrotnie proporcjonalny do skumulowanej mocy obliczeniowej całej sieci Bitcoin, w celu konsekwentnego utrzymania średniego czasu potrzebnego do znalezienia poprawnej liczby nonce. Znalazca poprawnej liczby nonce jest później uprawniony do dodania bloku do łańcucha bloków. W związku z tym, ci, którzy wyczerpią więcej zasobów obliczeniowych, by obliczyć liczbę nonce, pełnią większą rolę w określaniu łańcucha bloków. Dowód wykonanej pracy zapewnia odporność na ataki Sybil, w trakcie których jednostka zachowuje się jak wiele jednostek, aby zyskać dodatkową moc w zdecentralizowanym systemie i bardzo ograniczyć warunki z natury istniejącego wyścigu podczas zyskiwania dostępu do globalnej struktury danych.

Alternatywny protokół konsensusu, dowód stawki (Proof of Stake: PoS), został zaprezentowany przez Peercoin w 2012 roku [5]. W systemie dowodu stawki, uczestnicy głosują z wagą głosów równoważną do ilości posiadanego bogactwa, mierzonego w danej kryptowalucie. Dzięki takiemu układowi, osoby, które zainwestowały więcej, otrzymują większą władzę i są bardziej zachęcone do utrzymania rzetelności systemu, bowiem ryzykują



(a) Nie potrzeba więcej zasobów, jeśli konflikt nie występuje.

(b) W przypadku konfliktowej transakcji, węzły głosują za prawidłową transakcją.

Rysunek 1. Nano nie wymaga dodatkowych zasobów dla typowych transakcji. W przypadku konfliktowych transakcji, węzły muszą głosować na transakcję, którą chcą zachować.

utracenie ich pierwotnej inwestycji. Dowód stawki (PoS) eliminuje niepotrzebną konkurencję w zakresie mocy obliczeniowej, wymagając jedynie lekkiego oprogramowania działającego na sprzęcie o niskim poborze mocy.

Opublikowanie projektu Nano (poprzednio RaiBlocks) oraz implementacja miały miejsce w grudniu 2014, sprawiając, że Nano zostało jedną z pierwszych walut opartych na technologii skierowanego grafu acyklicznego (Directed Acyclic Graph: DAG) [6]. Wkrótce później, inne waluty DAG zaczęły być rozwijane, przede wszystkim DagCoin/Byteball i IOTA [7], [8]. Wymienione kryptowaluty oparte na DAG, złamały monopol łańcucha bloków, poprawiając wydajność i bezpieczeństwo systemu. Byteball osiąga konsensus polegając na "głównym łańcuchu" zawierającym renomowanego i zaufanego "świadka", natomiast IOTA osiąga konsensus poprzez łącznemu dowodowi pracy (PoW) kolejnych transakcji. Nano osiąga konsensus przez głosowanie saldem ważonym przy konfliktowych transakcjach. Ten system konsensusu zapewnia szybkie, bardziej deterministyczne transakcje, przy jednoczesnym utrzymaniu silnego, zdecentralizowanego systemu. Nano kontynuuje ten rozwój i zyskuje pozycję jednej z najbardziej wydajnych kryptowalut.

III. KOMPONENTY NANO

Przed opisaniem architektury Nano, zdefiniujemy poszczególne komponenty, które składają się na cały system.

A. Konto

Konto jest częścią klucza publicznego składającą się z pary kluczy podpisu cyfrowego. Publiczny klucz, zwany również adresem, jest udostępniany innym uczestnikom sieci, a klucz prywatny jest trzymany w tajemnicy. Podpisany cyfrowo pakiet danych gwarantuje, że treść została zatwierdzona przez właściciela klucza prywatnego. Jeden użytkownik może kontrolować wiele kont, ale tylko jeden adres publiczny może istnieć dla jednego konta.

B. Blok/Transakcja

Terminy "blok" i "transakcja" są często używane zamiennie, gdyż blok zawiera pojedynczą transakcję. Transakcja w szczególności odnosi się do akcji, podczas gdy blok odnosi się do cyfrowego kodowania transakcji. Transakcje są podpisywane przez klucz prywatny należący do konta, na którym przeprowadzana jest transakcja.

C. Główny rejestr (ledger)

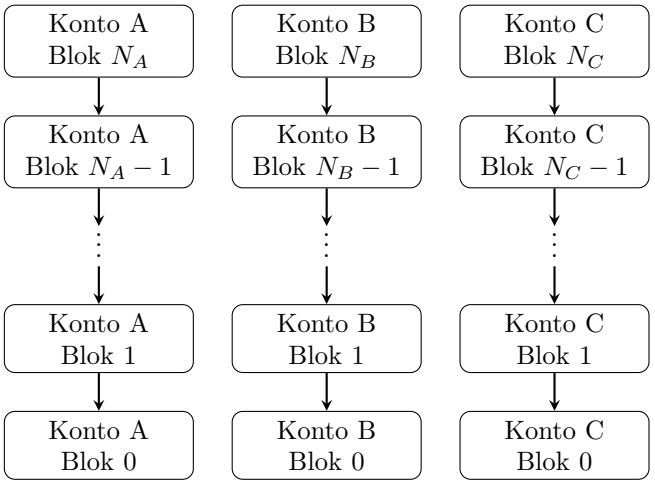
Główny rejestr jest globalnym zbiorem kont, w którym każde konto ma swój własny łańcuch transakcji (Rysunek 2). Jest to kluczowy element projektu, który należy do kategorii zastąpienia zgody wykonawczej zgodą projektową; przy sprawdzaniu podpisu każdy zgadza się, że tylko właściciel konta może modyfikować własny łańcuch. To przekształca pozornie dzieloną strukturę danych, rozproszony główny rejestr, w zestaw niedzielonych danych.

D. Węzeł (node)

Węzeł jest częścią oprogramowania działającego na komputerze zgodnym z protokołem Nano i uczestniczącym w sieci Nano. Oprogramowanie zarządza głównym rejestrem i wszelkimi kontami, które może kontrolować węzeł, jeśli takie istnieją. Węzeł może przechować cały główny rejestr lub zmniejszoną historię, zawierającą tylko kilka ostatnich bloków łańcucha bloków każdego konta. Podczas konfigurowania nowego węzła zaleca się zweryfikowanie całej historii i lokalne zmniejszenie.

IV. PRZEGLĄD SYSTEMU

W przeciwieństwie do łańcuchów bloków używanych w wielu innych kryptowalutach, Nano używa struktury nazywanej *block-lattice*. Każde konto posiada swój własny łańcuch bloków równoważny historii transakcji/salda

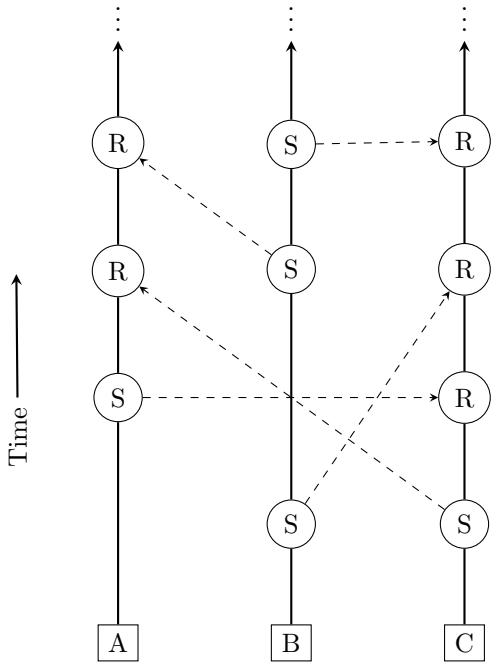


Rysunek 2. Każde konto posiada swój własny łańcuch bloków zawierający historię salda konta. Blok 0 musi być transakcją otwierającą. (Sekcja IV-B)

konta (Rysunek 2). Każdy łańcuch konta może być aktualizowany tylko przez właściciela konta; dzięki temu każdy łańcuch kont może zostać natychmiast zaktualizowany i asynchronicznie powiązany z resztą sieci block-lattice, co sprawia, że transakcje są szybkie. Protokół Nano jest wyjątkowo lekki; każda transakcja mieści się w wymaganym minimalnym rozmiarze pakietu UDP w celu przesłania go przez Internet. Wymagania sprzętowe dla węzłów są również minimalne, ponieważ węzły muszą tylko rejestrować i retransmitować bloki dla większości transakcji (Rysunek 1).

System zostaje zainicjowany *kontem genezy* zawierającym *początkowe saldo*. Saldo początkowe jest stałą wielkością i nigdy nie może zostać zwiększone. Jest ono dzielone i wysyłane na inne konta za pośrednictwem transakcji zarejestrowanych na początkowym łańcuchu konta. Suma sald wszystkich rachunków nigdy nie przekroczy początkowego salda, które daje systemowi górną granicę w stosunku do ilości, bez możliwości jej zwiększenia.

W tej sekcji omówiono sposób konstruowania i propagowania różnych typów transakcji w całej sieci.



Rysunek 3. Wizualizacja block-lattice. Każdy transfer funduszy wymaga bloku nadawczego (S) i bloku odbiorczego (R). Każdy blok podpisany jest przez właściciela łańcucha konta (A,B,C).

A. Transakcje

Przesłanie środków z jednego konta na drugie wymaga dwóch transakcji: *wysłania* - odjęcia kwoty z salda nadawcy i *odbioru* - dodania kwoty do salda konta odbierającego (Rysunek 3).

Przesyłanie kwot jako oddzielnych transakcji na kontach nadawcy i odbiorcy służy kilku ważnym celom:

1) Sekwencjonowanie transferów przychodzących, które są z natury asynchroniczne.
2) Utrzymanie transakcji małymi, tak by mieściły się w pakietach UDP.
3) Ułatwienie zmniejszania głównego rejestru przez minimalizowanie śladu danych.
4) Izolowanie transakcji rozliczonych od nierozliczonych.

Więcej niż jedno konto przesyłające do tego samego konta docelowego jest operacją asynchroniczną; opóźnienie sieci i konta wysyłające niekoniecznie komunikują się ze sobą, co oznacza, że nie ma uniwersalnego akceptowalnego sposobu, aby wiedzieć, która transakcja miała miejsce jako pierwsza. Ponieważ dodawanie jest asocjacyjne, kolejność danych wejściowych nie ma znaczenia, a zatem po prostu potrzebujemy globalnego porozumienia. Jest to kluczowy element konstrukcyjny, który przekształca zgodę wykonawczą w zgodę projektową. Konto odbierające ma kontrolę nad wyborem, który transfer dotrze pierwszy i jest to wyrażone przez kolejność podpisania przychodzących bloków.

Jeśli konto chce wykonać duży transfer, który został odebrany jako zestaw wielu małych transferów, chcemy przedstawiać to w sposób, który pasuje do pakietu UDP. Gdy konto odbierające sekwencyjnie przesyła dane, utrzymuje bieżącą sumę salda na koncie, dzięki czemu w każdej chwili może przenieść dowolną kwotę przy transakcji o stałym rozmiarze. Różni się to od modelu transakcji wejścia/wyjścia wykorzystywanego przez Bitcoin i inne kryptowaluty.

Niektóre węzły nie są zainteresowane wydatkowaniem zasobów, aby przechowywać pełną historię transakcji konta; interesuje je tylko bieżące saldo każdego konta. Kiedy konto dokonuje transakcji, koduje ono zgromadzone saldo, a węzły muszą jedynie śledzić ostatni blok, co pozwala im odrzucić dane historyczne jednocześnie zachowując poprawność.

Nawet przy skupieniu się na zgodach projektowych, istnieje okno opóźnienia podczas sprawdzania transakcji z powodu identyfikacji i postępowania złośliwych podmiotów w sieci. Ponieważ uzgodnienia w Nano są osiągane szybko, rzędu milisekund do sekund, możemy przedstawić użytkownikowi dwie znane kategorie transakcji przychodzących: rozliczone i nierozliczone. Rozliczone transakcje to transakcje, w przypadku których konto wygenerowało bloki odbiorcze. Nierozliczone transakcje nie zostały jeszcze włączone do łącznego salda odbiorcy. Jest to zamiennik dla bardziej złożonej i nieznanej miary potwierdzenia w innych kryptowalutach.

B. Tworzenie konta

Aby stworzyć konto, musisz wyemitować *otwierającą* transakcję (open) (Rysunek 4). Otwierająca transakcja jest zawsze pierwszą transakcją każdego łańcucha konta i może zostać utworzona przy pierwszym otrzymaniu środków. Pole *account (konto)* przechowuje klucz publiczny (adres) uzyskany z klucza prywatnego używanego do podpisywania (signature). Pole *source (źródło)* zawiera hash transakcji konta, które wysłało fundusze. Podczas tworzenia konta, należy wybrać przedstawiciela do głosowania w

Twoim imieniu; można go zmienić później (Sekcja IV-F). Konto może zadeklarować się jako jego własny przedstawiciel.

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

Rysunek 4. Anatomia transakcji otwierającej.

C. Saldo konta

Saldo konta jest rejestrowane wewnątrz głównego rejestru. Zamiast rejestrować kwotę transakcji, weryfikacja (Sekcja IV-I) wymaga sprawdzenia różnicy między saldem w bloku nadawczym a saldem bloku poprzedzającego. Konto odbierające może następnie zwiększyć poprzednie saldo, mierzone w saldzie końcowym, podanym w nowym bloku odbiorczym. Odbywa się to w celu zwiększenia szybkości przetwarzania podczas pobierania dużych ilości bloków. Gdy żądasz historii konta, kwoty są już podane.

D. Wysyłanie z konta

Aby wysłać z adresu, adres musi mieć już istniejący otwarty blok, a zatem też i saldo (Rysunek 5). Pole *previous (poprzednie)* zawiera hash poprzedniego bloku w łańcuchu konta. Pole *destination (cel)* zawiera konto, na które środki mają zostać wysłane. Blok nadawczy jest niezmienny po potwierdzeniu. Po transmisji do sieci, środki są natychmiast odejmowane od salda rachunku nadawcy i czekają, dopóki strona odbierająca nie podpisze bloku, aby zaakceptować te fundusze. Środki oczekujące nie powinny być uważane za oczekujące na potwierdzenie, ponieważ są one praktycznie wysłane na koncie nadawcy, a nadawca nie może odwołać transakcji.

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

Rysunek 5. Anatomia wysłanej transakcji.

E. Otrzymywanie transakcji

Aby ukończyć transakcję, odbiorca wysłanych środków musi utworzyć blok odbiorczy na swoim własnym łańcuchu konta (Rysunek 6). Pole source (źródło) odnosi się do hasha powiązanej transakcji wysyłającej. Po utworzeniu i przesłaniu tego bloku saldo konta zostaje zaktualizowane, a środki zostają oficjalnie przeniesione na konto odbierające.

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

Rysunek 6. Anatomia transakcji odbierającej.

F. Przypisywanie przedstawiciela

Właściciele kont mający możliwość wyboru przedstawiciela do głosowania w ich imieniu, to potężne narzędzie decentralizacyjne, które nie ma silnego odpowiednika w protokołach Proof of Work lub Proof of Stake. W konwencjonalnych systemach PoS, węzeł właściciela konta musi być uruchomiony, aby uczestniczyć w głosowaniu. Ciągłe działanie węzła jest niepraktyczne dla wielu użytkowników; udzielenie pełnomocnikowi upoważnienia do głosowania w imieniu konta rozluźnia ten wymóg. Właściciele kont mają możliwość ponownego przypisania konsensusu do dowolnego konta, w dowolnym momencie. Transakcja *change (zmiana)* zmienia przedstawiciela konta, odejmując wagę głosu od starego przedstawiciela i dodając wagę do nowego przedstawiciela (Rysunek 7). Żadne fundusze nie są przenoszone w ramach tej transakcji, a przedstawiciel nie ma siły wydatkowania środków z konta.

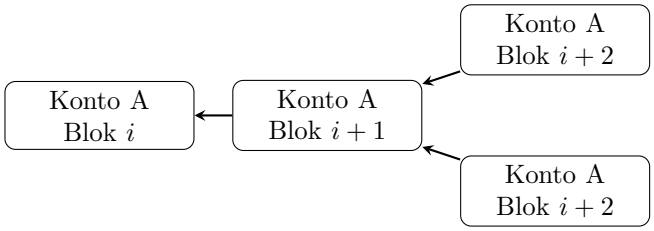
```
change -
    previous: DC04354B1...AE8FA2661B2,
    representative: xrb_1anrz...posrs,
    work: 0000000000000000,
    type: change,
    signature: 83B0...006433265C7B204
"
```

Rysunek 7. Anatomia transakcji change.

G. Rozgałęzienia i głosowanie

Rozgałęzienie występuje gdy j podpisze bloki $b_1, b_2, \ldots, b_j$ twierdząc, że którekolwiek z nich posiadają tego samego poprzednika (Rysunek 8). Bloki te powodują sprzeczny widok stanu konta i muszą zostać rozwiązane. Tylko właściciel konta ma możliwość podpisywania bloków w łańcuchu konta, więc rozgałęzienie musi wynikać ze złego oprogramowania lub złośliwego działania (podwójnego wydatku) przez właściciela konta.

Po wykryciu, przedstawiciel utworzy głosowanie odnoszące się do bloku $\hat{b}_i$ w jego głównym rejestrze i wyemituje



Rysunek 8. Rozgałęzienie występuje, gdy dwa (lub więcej) podpisane bloki odnoszą się do tego samego bloku. Starsze bloki są po lewej stronie; nowsze bloki znajdują się po prawej stronie.

je do sieci. Waga w głosowaniu węzła, w_i, jest sumą sald wszystkich kont, które nazwały go swoim przedstawicielem. Węzeł będzie obserwował nadchodzące głosy od innych przedstawicieli online M i będzie utrzymywał skumulowaną liczbę głosów przez 4 okresy głosowania, 1 minutę łącznie i potwierdzi zwycięski blok (Równanie 1).

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

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

Najpopularniejszy blok b^* będzie posiadał większość głosów i zostanie zachowany w głównym rejestrze węzła (Równanie 2). Bloki, które utraciły głos są odrzucane. Jeśli przedstawiciel zastąpi blok w swoim głównym rejestrze, utworzy nowe głosowanie z wyższym numerem porządkowym i przekaże je do sieci. Jest to jedyny scenariusz, w którym przedstawiciele głosują.

W niektórych okolicznościach krótkie problemy z łącznością sieciową mogą spowodować, że nadawany blok nie zostanie zaakceptowany przez wszystkich użytkowników. Każdy kolejny blok na tym koncie zostanie zignorowany jako nieważny przez osoby, które nie widziały początkowej transmisji. Powtórne przesłanie tego bloku, zostanie zaakceptowane przez pozostałych uczestników, a kolejne bloki zostaną automatycznie pobrane. Nawet w przypadku pojawienia się rozgałęzienia lub brakującego bloku, dotyczy to tylko kont przywoływanych w transakcji; reszta sieci kontynuuje przetwarzanie transakcji dla wszystkich innych kont.

H. Dowód wykonanej pracy

Wszystkie cztery typy transakcji mają pole robocze, które musi być poprawnie wypełnione. Pole robocze pozwala twórcy transakcji obliczyć liczbę nonce tak, że jej hash powiązany jest z poprzednim polem w transakcjach odbioru/wysłania/zmiany lub pole konta w otwartej transakcji jest poniżej pewnej wartości progowej. W przeciwieństwie do Bitcoina, dowód wykonanej pracy w Nano jest po prostu używany jako narzędzie antyspamowe, podobnie do HashCash, i może być obliczany w parę sekund [9]. Po wysłaniu transakcji, wartość Proof of Work dla następnego bloku może być wstępnie obliczona, ponieważ znane jest poprzednie pole bloku; spowoduje to, że transakcje będą widoczne natychmiast dla użytkownika końcowego, o ile czas pomiędzy transakcjami będzie dłuższy niż czas wymagany do obliczenia PoW.

I. Weryfikacja transakcji

Aby blok mógł być uznany za poprawny, musi posiadać następujące cechy:

1) Blok nie może już znajdować się w głównym rejestrze (zduplikowana transakcja).
2) Musi być podpisany przez właściciela konta.
3) Poprzedni blok jest głównym blokiem łańcucha konta. Jeśli blok istnieje, ale nie jest blokiem głównym, jest rozgałęziony.
4) Konto musi mieć otwarty blok.
5) Obliczony hash spełnia wymagania dotyczące progu PoW.

Jeśli jest to blok odbiorczy, sprawdź, czy wartość hasha bloku źródłowego jest oczekująca, co oznacza, że nie została jeszcze odebrana. Jeśli jest to blok nadawczy, saldo musi być mniejsze niż poprzednie saldo.

V. TYPY ATAKÓW

Nano, podobnie jak wszystkie zdecentralizowane waluty, może zostać zaatakowane przez złośliwe strony celem finansowego zysku lub upadku sieci. W tym rozdziale wyszczególniliśmy możliwe scenariusze ataków, ich konsekwencje oraz to, w jaki sposób protokół Nano potrafi im zapobiegać.

A. Synchronizacja Odstępów Blokowych

W rozdziale IV-G, rozpatrywaliśmy scenariusz, w którym blok mógł nie zostać poprawnie rozgłoszony, powodując to, że sieć zignorowała następne bloki. W wypadku gdy węzeł zaobserwował blok, który nie posiadał poprzedniego bloku odniesienia, węzeł posiada dwie opcje:

1) Zignorować blok, który może być złośliwym, śmieciowym blokiem
2) Zażądać ponownej synchronizacji z innym węzłem.

W wypadku ponownej synchronizacji, połączenie TCP musi zostać stworzone z węzłem ładowania początkowego (bootstrapping node), by ułatwić resynchronizację poprzez zwiększony ruch. Jednakże, jeśli blok ten był błędny, wtedy resynchronizacja nie była konieczna i niepotrzebnie zwiększała ruch w sieci. Ten typ ataku nosi nazwę Atak Wzmocnieniowy (Network Amplification Attack) i skutkuje odmową usługi (DoS - denial-of-service).

By uniknąć niepotrzebnych resynchronizacji, węzły będą czekać do momentu zaobserwowania pewnego progu głosów dla potencjalnie szkodliwego bloku, zanim zainicjują połączenie z węzłem ładowania początkowego służące synchronizacji. Jeśli blok nie otrzyma wystarczającej liczby pozytywnych głosów, można przyjąć, że dane skojarzone z tym blokiem są śmieciowe.

B. Powódź Transakcyjna

Złośliwa jednostka może wysłać wiele niepotrzebnych, ale poprawnych transakcji pomiędzy kontami, które kontroluje, próbując przesycić sieć. Przy braku opłat transakcyjnych mogą taki atak kontynuować w nieskończoność. Jednakże dowód wykonanej pracy (PoW) potrzebny dla każdej transakcji ogranicza tempo transakcji, które złośliwy podmiot mógłby wygenerować bez znaczących inwestycji w zasoby obliczeniowe. Nawet w przypadku takiego ataku próbującego nadąć główny rejestr (ledger), węzły, które nie są pełnymi węzłami historycznymi, mają zdolność do zmniejszenia starych transakcji z ich łańcucha; to ogranicza użycie pamięci masowej pochodzącej z tego typu ataków dla prawie wszystkich użytkowników.

C. Atak Sybil

Jednostka może stworzyć setki węzłów sieci Nano na pojedynczej maszynie, jednakże odkąd system głosujący jest ważony na podstawie salda konta, dodanie kolejnych węzłów do sieci nie przyzna atakującemu dodatkowych głosów. W związku z tym nie można odnieść żadnej korzyści za pomocą ataku Sybil.

D. Atak Wydaj-Grosz

Atak wydaj-grosz (penny-spend) ma miejsce wtedy, gdy atakujący wydaje nieskończenie małe ilości środków dużej liczbie kont w celu zmarnowania zasobów pamięciowych węzłów. Generowanie bloków jest ograniczone przez algorytm dowodu wykonanej pracy (PoW), co w pewnym stopniu ogranicza liczbę kont i transakcji. Węzły, które nie są pełnymi węzłami historycznymi, mogą zmniejszać konta, które znajdują się poniżej metryki statystycznej i najprawdopodobniej nie są prawidłowymi kontami. Ostatecznie, Nano jest przystosowane do korzystania z minimalnej stałej przestrzeni dyskowej, toteż ilość miejsca do przechowywania jednego dodatkowego konta jest proporcjonalna do wielkości otwarcie bloku + indeksowanie = 96B + 32B = 128B. Oznacza to, że 1GB przestrzeni dyskowej może przechowywać 8 milionów kont powstałych przez atak wydaj-grosz. Jeśli węzły chcą bardziej agresywnie zmniejszać konta, mogą wyliczyć dystrybucję opartą o częstotliwość dostępu i oddelegować rzadko używane konta do wolniejszej pamięci.

E. Atak Wcześniej Wyliczonego PoW

Skoro właściciel konta jest jedynym podmiotem dodającym bloki do łańcucha bloków konta, bloki wraz z dowodem pracy (PoW) mogą zostać policzone sekwencyjnie, zanim zostaną rozgłoszone w sieci. W tym przypadku atakujący generuje niezliczoną ilość sekwencyjnych bloków, każdy o minimalnej wartości, przez dłuższy okres. W pewnym momencie atakujący wykonuje atak odmowy usługi (DoS), zalewając sieć wieloma prawidłowymi transakcjami, które będą przetwarzane i odbijane przez pozostałe węzły tak szybko, jak to tylko możliwe. Atak stanowi zaawansowaną wersję ataku powodzi transakcyjnej (transaction flooding) opisaną w rozdziale V-B. Może on działać tylko przez krótki czas, ale może zostać użyty w połączeniu z innymi atakami, takimi jak Atak >50% (Rozdział V-F), by podnieść efektywność. Obecnie badane są metody ograniczania szybkości transakcji i pozostałe techniki, aby złagodzić skutki takich działań.

F. Atak >50%

Miarą konsensusu w sieci Nano jest system ważenia głosów na podstawie posiadanego salda. Jeśli atakujący jest w stanie zyskać ponad 50% siły głosowania, może on wprawić sieć w chwiejny konsensus, powodując załamanie systemu. Osoba atakująca może obniżyć saldo, które musi stracić, by uniemożliwić dobrym węzłom głosowanie w sieci DoS. Nano podejmuje następujące działania, aby zapobiec takiemu atakowi:

1) Podstawową formą obrony przeciwko tego typu atakowi jest ciężar głosowania ściśle powiązany z inwestowaniem w system. Posiadacz konta jest z natury motywowany do utrzymania uczciwości systemu w celu ochrony swoich inwestycji. Próba odwrócenia głównego rejestru (ledger) miałaby destrukcyjny wpływ na system, który zniszczyłby również inwestycje atakującego.
2) Koszt takiego ataku jest proporcjonalny do kapitalizacji rynkowej Nano. W systemach PoW można wynaleźć technologię, która zapewnia nieproporcjonalną kontrolę w stosunku do inwestycji monetarnych, a jeśli atak się powiedzie, technologia ta może zostać również wykorzystana do innych celów po zakończeniu ataku. W Nano, koszt przeprowadzenia ataku skaluje się wraz z samą siecią i jeśli atak się powiedzie, inwestycja poczyniona w tym celu nie może zostać odzyskana.
3) Aby utrzymać maksymalne kworum głosujących, kolejną linią obrony jest głosowanie reprezentatywne. Posiadacze kont, którzy nie mogą rzetelnie uczestniczyć w głosowaniu w związku z połączeniem, mogą wskazać przedstawiciela, który będzie mógł głosować z wagą ich salda. Maksymalizacja liczby i różnorodności przedstawicieli zwiększa odporność sieci.
4) Rozgałęzienia w Nano nie są nigdy przypadkowe, więc węzły mogą podejmować decyzje dotyczące sposoby interakcji z rozgałęzionymi blokami. Jedynym przypadkiem, w którym konta nieuczestniczące bezpośrednio w ataku są podatne na blokadę, jest ten, w którym otrzymają one środki z atakującego konta. Konta, które chcą zostać zabezpieczone przed rozgałęzieniem bloku, mogą poczekać trochę lub znacznie dłużej przed otrzymaniem środków z konta, które generuje rozgałęzienie lub też zdecydować się nigdy je nie otrzymywać.
5) Ostatnią linią obrony, która nie została jeszcze zaimplementowana, jest *cementowanie blokowe (block cementing)*. Nano dokłada wszelkich starań, aby szybko ustawić rozgałęzienie bloku poprzez głosowanie. Węzły mogą zostać skonfigurowane do cementowania bloków, co uniemożliwiłoby ich wycofanie

po pewnym czasie. Sieć jest wystarczająco zabezpieczona poprzez skupienie się na szybkim czasie ustalania, aby zapobiec niejednoznacznym rozgałęzieniom.

Bardziej wyszukaną wersją ataku > 50% jest sytuacja szczegółowo przedstawiona na Rysunku 9. Pole "Offline" oznacza procent przedstawicieli, którzy zostali nazwani, ale nie są online, aby zagłosować. Pole "Udział" jest to ilość poczynionych inwestycji przez atakującego, z którą głosuje. "Aktywni" są to reprezentanci, którzy pozostają online i głosują zgodnie z protokołem. Atakujący może przesunąć ilość udziałów celem wyłączenia pozostałych głosujących poprzez atak sieciowy DoS. Jeśli ten atak może zostać utrzymany, atakowani przedstawiciele pozostaną niezsynchronizowani, co zostało pokazane poprzez pole "Unsync". Ostatecznie, atakujący może uzyskać krótki impuls we względnej sile głosu poprzez atak odmowy serwisu (DoS) na nowych zestaw reprezentantów, podczas gdy stary zestaw ponownie synchronizuję główny rejestr (ledger) - jest to zaprezentowane jako pole "Atak".

Offline	Unsync	**Atak**	Aktywni	Udział

Rysunek 9. Potencjalny układ głosowania, który mógłby obniżyć wymagania ataku 51%.

Jeśli atakujący jest w stanie sprawić by Udział >Aktywni poprzez kombinacje tych warunków, będzie zdolny odwrócić głosowanie nad rejestrem głównym (ledger) poprzez poniesienie kosztu posiadanego udziału. Możemy oszacować, ile będzie kosztował ten typ ataku, badając kapitalizację rynku innych systemów. Jeśli oszacujemy, że 33% reprezentantów jest offline lub zaatakowanych przez DoS, atakujący będzie potrzebował wydać 33% kapitalizacji rynkowej w celu zaatakowania systemu poprzez głosowanie.

G. Zatrucie Ładowania Początkowego

Im dłużej atakujący jest w stanie zatrzymać stary klucz prywatny wraz z bilansem, tym wyższe prawdopodobieństwo, że salda, które istnieją w tym czasie, nie będą mieć uczestniczących przedstawicieli, ponieważ ich salda lub salda przedstawicieli zostaną przeniesione na nowe konta. Oznacza to, że jeśli węzeł jest początkowo ładowany do starej reprezentacji sieci, w której atakujący posiada kworum udziałów głosowych w porównaniu do przedstawicieli w danym momencie, będą oni w stanie balansować decyzjami głosowania dla tego węzła. Jeśli nowy użytkownik chciałby oddziaływać z kimkolwiek oprócz węzła atakującego, wszystkie jego transakcje zostałyby odrzucone z powodu różnic w blokach nadrzędnych. Ostateczny wynik jest taki, że węzły mogą tracić czas na nowe węzły w sieci, dostarczając im złych informacji. Aby temu zapobiec, węzły mogą zostać sparowane z początkową bazą danych kont i z dobrze znanymi blokami nadrzędnymi; jest to alternatywa dla pobierania bazy danych aż do bloku genezy. Im pobieranie jest bliższe bieżącemu, tym większe prawdopodobieństwo dokładnej obrony przed atakiem. Ostatecznie, atak ten nie jest prawdopodobnie gorszy od przesyłania niepotrzebnych danych do węzłów podczas ładowania początkowego, ponieważ nie będą one w stanie przeprowadzić transakcji z kimkolwiek, kto ma aktualną bazę danych.

VI. IMPLEMENTACJA

Obecna implementacja referencyjna jest napisana w C++ i wydaje wersje począwszy od 2014 r. w serwisie Github [10].

A. Cechy Projektu

Implementacja Nano jest zgodna ze standardem architektury opisanym w tym dokumencie. Dodatkowe specyfikacje są opisane poniżej.

1) Algorytm Podpisu: Nano używa zmodyfikowanego algorytmu krzywych eliptycznych ED25519 z funkcją haszującą Blake2b dla wszystkich cyfrowych sygnatur. [11] ED25519 został wybrany z uwagi na szybkie działanie i weryfikacje, a także duże bezpieczeństwo.

2) Algorytm Haszujący: Jako, że algorytm haszujący jest wykorzystywany tylko do zapobiegania spamowi sieciowemu, wybór algorytmu jest mniej istotny w porównaniu do kryptowalut, które można kopać. Nasza implementacja używa Blake2b jako algorytmu funkcji skrótu na zawartość bloku [12].

3) Funkcja Wyprowadzenia Klucza: W referencyjnym portfelu klucze są szyfrowane hasłem, które jest podawane poprzez funkcje wyprowadzania klucza (key derivation function) celem zabezpieczenia przed próbami łamania układami ASIC. Obecnie algorytm Argon2 [13] jest zwycięzcą publicznego konkursu mającego na celu stworzenie bezpiecznej funkcji wyprowadzania klucza.

4) Interwał Czasowy Bloków: Z uwagi na to, że każde konto ma swój własny łańcuch bloków, aktualizacje mogą zostać wykonywane asynchronicznie z punktu widzenia sieci. W związku z tym nie ma interwałów między blokami i transakcje mogą być natychmiast publikowane.

5) Protokół UDP: Nasz system został zaprojektowany do pracy ciągłej przy użyciu, jak tylko możliwie minimalnej ilości zasobów obliczeniowych. Wszystkie wiadomości w systemie zostały zaprojektowane tak, aby były bezstanowe i mieściły się w pojedynczym pakiecie UDP. Ułatwia to również lekkim peerom z nieciągłym połączeniem, aby uczestniczyć w sieci bez ustanawiania krótkoterminowych połączeń TCP. TCP jest używany tylko dla nowych peerów, kiedy chcą początkowo załadować łańcuch bloków w masowy sposób.

Węzły mogą być pewne, że ich transakcja została odebrana przez sieć, obserwując ruch rozgłaszanych transakcji pochodzących z innych węzłów i powinny zobaczyć, że kilka kopii powróciło z powrotem do nich.

B. IPv6 i Multicast

Opierając się na bezpołączeniowym protokole UDP, przyszłe implementacje mogą wykorzystywać multicast IPv6 jako zamiennik dla tradycyjnej powodzi transakcji i

rozgłaszania głosów. Zmniejszy to zużycie przepustowości sieci i zapewni w przyszłości większą politykę elastyczności węzłom.

C. Wydajność

W chwili pisania tego tekstu, sieć Nano przetworzyła 4.2 miliona transakcji, dając rozmiar łańcucha bloków na poziomie 1.7 GB. Czasy transakcyjne wyrażane są w pojedynczych sekundach. Obecna implementacja referencyjna działająca na rynkowych dyskach SSD może przetwarzać ponad 10 000 transakcji na sekundę, ograniczona przez operacje IO.

VII. UŻYCIE ZASOBÓW

Tutaj znajduje się przegląd zasobów używanych przez węzeł Nano. Ponadto, omawiamy pomysły zmniejszenia zużycia zasobów dla konkretnych przypadków. Zredukowane węzły są zwykle nazywane lekkimi, przyciętymi lub uproszczonymi węzłami weryfikacji płatności (SPV).

A. Sieć

Aktywność sieciowa węzła zależy od tego w jakim stopniu węzeł przyczynia się do zdrowia sieci.

1) Przedstawiciel: Węzeł-przedstawiciel potrzebuje maksymalnych zasobów sieciowych, ponieważ obserwuje ruch głosowania innych przedstawicieli i publikuje własne głosy.

2) Niewymagający Zaufania: Węzeł, który nie wymaga zaufania jest podobny do węzła-przedstawiciela, z tą różnicą, że jest tylko obserwatorem, nie zawiera on klucza prywatnego konta przedstawiciela i nie publikuje własnych głosów.

3) Zaufany: Węzeł zaufania obserwuje ruch głosowania od jednego przedstawiciela, któremu ufa, aby prawidłowo osiągnąć konsensus. Zmniejsza to ilość przychodzącego ruchu głosów od przedstawicieli do tego węzła.

4) Lekki: Lekki węzeł jest także węzłem zaufania, który obserwuje ruch tylko dla kont, którymi jest zainteresowany, umożliwiając minimalne wykorzystywanie sieci.

5) Ładowanie Początkowe: Węzeł ładowania początkowego obsługuje część lub całość rejestru głównego dla węzłów, które same przenoszą się do trybu online. Odbywa się to za pośrednictwem połączenia TCP zamiast UDP, ponieważ wiąże się to z dużą ilością danych, które wymagają zaawansowanej kontroli przepływu.

B. Pojemność Dysku

W zależności od potrzeb użytkownika różne konfiguracje węzłów wymagają różnych wymagań dotyczących pamięci.

1) Historyczny: Węzeł zainteresowany przechowywaniem pełnego historycznego rejestru wszystkich transakcji będzie wymagał maksymalnej ilości pamięci.

2) Obecny: Ze względu na zaprojektowane utrzymanie zgromadzonych sald w blokach, węzły muszą jedynie przechowywać najnowsze lub główne bloki dla każdego konta, aby móc uczestniczyć w konsensusie. Jeśli węzeł nie jest zainteresowany utrzymaniem pełnej historii, może zdecydować się zachować tylko bloki główne.

3) Lekki: Lekki węzeł nie przechowuje lokalnie danych z rejestru głównego, a tylko uczestniczy w sieci, aby obserwować aktywność na kontach, którymi jest zainteresowany lub opcjonalnie tworzy nowe transakcje z kluczami prywatnymi, które przechowuje.

C. CPU

1) Generowanie Transakcji: Węzeł zainteresowany tworzeniem nowych transakcji musi wygenerować dowód wykonanej pracy (PoW), aby przejść przez mechanizm blokujący sieć Nano. Obliczenia na różnych sprzętach zostały porównane w Dodatku A.

2) Przedstawiciel: Przedstawiciel musi weryfikować sygnatury bloków, głosować, a także tworzyć swoje własne podpisy, aby uczestniczyć w konsensusie. Ilość zasobów procesora dla węzła-przedstawiciela jest znacznie mniejsza niż potrzebna do generacji transakcji. Wymagania te powinny zostać spełnione dla dowolnego pojedynczego procesora współczesnego komputera.

3) Obserwator: Węzeł-obserwator nie generuje własnych głosów. Skoro narzut generacji podpisu jest minimalny, wymagania procesora są prawie identyczne, jak w wypadku węzła-przedstawiciela.

VIII. KONKLUZJA

W niniejszym dokumencie przedstawiliśmy strukturę całkowicie darmowej, błyskawicznej i niewymagającej zaufania stron trzecich kryptowaluty, która wykorzystuje nowatorską strukturę block-lattice i delegowane głosowanie przez dowód stawki (Proof of Stake). Sieć wymaga minimalnych zasobów, nie potrzebuje sprzętu górniczego o wysokiej mocy i zapewnia wysoką przepustowość transakcji. Wszystko to zostało osiągnięte za pomocą indywidualnego łańcucha bloków dla każdego konta, eliminując problemy z dostępnością i nieefektywności globalnej struktury danych. Zidentyfikowaliśmy możliwe rodzaje ataków w systemie i przedstawiliśmy argumenty na temat tego, w jaki sposób Nano jest odporne na te formy ataków.

Dodatek A
Sprzętowe testy porównawcze

Jak wcześniej wspomniano, PoW w Nano ma na celu zmniejszenie spamu sieciowego. Nasza implementacja węzła zapewnia akcelerację, która może zostać wykorzystana przez GPU zgodne z OpenCL. Tablica I dostarcza rzeczywistego porównania różnych urządzeń. W tej chwili próg PoW jest ustalony, lecz próg adaptacyjny może zostać zaimplementowany, gdy średnia moc obliczeniowa się zwiększy.

PODZIĘKOWANIA

Dziękujemy Brianowi Pugh za skompilowanie i sformatowanie tego dokumentu.

Tablica I
Wydajność Sprzętowa PoW

Urządzenie	Transakcje Na Sekundę
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

Literatura

[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