Nano: Ücretsiz Dağıtılmış Kriptopara Ağı

Colin LeMahieu
clemahieu@nano.co

***Özet*—Son zamanlarda, yüksek talep ve sınırlı ölçeklenebilirlik, piyasada bulunan popüler kriptoparaların işlem süresini ve ücretlerini arttırarak tatmin etmeyen bir deneyim yaşatmaya başladı. Sizleri, özgün block-lattice mimarisi ile her hesabın kendine ait block zincirinin olduğu ve neredeyse anlık işlem hızı ve sınırsız ölçeklenebilirlik sunan Nano ile tanıştırıyoruz. Her kullanıcının kendine has blok zinciri olması, kullanıcıların ağın geri kalanı ile eşzamansız olarak senkronize olabilmesini sağlamakta, bu da hızlı transfer zamanları ve minimum ek işlem masrafı ortaya çıkartmaktadır. İşlemler, işlem tutarı yerine hesap bakiyelerini takip ederek güvenlikten ödün vermeden veritabananın ileri düzeyde budanmasına olanak sağlar. Bugüne kadar, Nano ağı sadece 1.7GB boyutunda budanmamış hesap defteri ile 4.2 milyon işlem gerçekleştirdi. Nano'nun ücretsiz, saliselik işlemleri onu, tüketici işlemleri için en önemli kriptopara yapmaktadır.**

***Anahtar Kelimeler*—kriptopara, block zinciri, nano, dağıtılmış defter, dijital işlemler**

I. Giriş

BITCOIN'IN 2009 yılında ortaya çıkmasından beri, geleneksel, devlet destekli para birimleri ve finansal sistemlerden, kriptografi üzerine kurulu, karşılıklı güvene bağımlı olmayan ve güvenli bir biçimde sermaye depolama ve transfer etme olanağı sunan modern ödeme sistemlerine doğru büyüyen bir kaçış başladı [1]. Etkili bir şekilde çalışması için, bir para biriminin kolayca transfer edilebilmesi, tek yönlü çalışması, işlem ücretinin sınırlı veya ücretsiz olması gerekmektedir. İşlem sürelerinin artışı, yüksek işlem ücretleri ve tartışılabilir ağ ölçeklenebilirliği Bitcoin'in günlük yaşamda kullanılabilir bir para birimi olmasıyla ilgili sorular oluşturdu.

Bu makalede, size, yenilikçi block-lattice veri yapısı üzerine kurulu, sınırsız ölçeklenebilirlik ve ücretsiz işlemler sunan, düşük gecikme süreli kriptopara Nano'yu sunuyoruz. Nano, dizayn olarak, tek amacı yüksek performanslı bir kriptopara olmak olan basit bir protokoldür. Nano protokolünün düşük güçteki donanımlarda çalışabilir olması onu pratik, merkezi olmayan, günlük kullanıma uygun bir kriptopara haline getirir.

Bu makalede bildirilen kriptopara istatistikleri, makalenin yayınlanma tarihi itibariyle doğrudur.

II. Arka plan

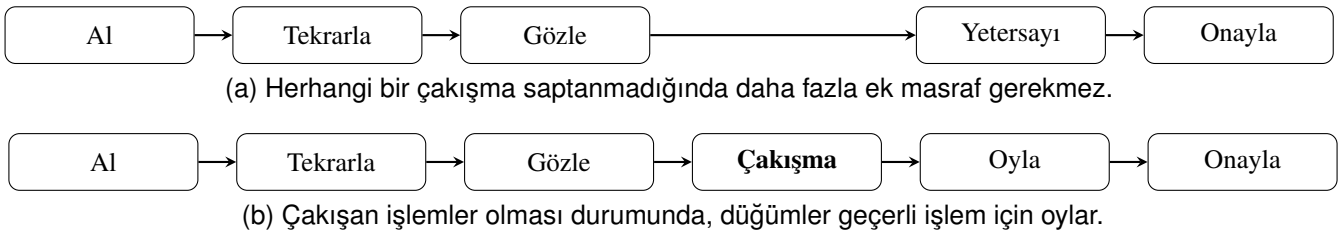
2008 yılında, Satochi Nakamoto takma adında anonim bir şahıs dünyanın ilk merkezi olmayan kriptoparası Bitcoin'i özetleyen bir makale yayınladı. [1]. Bitcoin'in beraberinde getirdiği en temel yenilik, kamuya açık, değişmez ve merkezsiz bir veri yapısı olan ve para biriminin işlemlerinin defteri olarak kullanılan blok zinciriydi (blockchain). Ne yazık ki, Bitcoin olgunlaştıkça, protokolde yer alan bazı konular Bitcoin'in birçok uygulama için yetersiz kalmasına sebep oldu:

1) Düşük ölçeklenebilirlik: Blok zinciri içerisindeki her blok sınırlı miktarda veri saklayabilir, bu da sistemin saniyede sadece bu miktarda işlem yapabileceği anlamına gelir ki bu durum blok zincirindeki bir yeri emtia haline getirir. Şu anki medyan işlem ücreti $10.38'dir [2].
2) Yüksek gecikme süresi: Ortalama onay süresi 164 dakikadır [3].
3) Güç israfı: Bitcoin ağı işlem başına ortalama 260KWh harcayarak yılda yaklaşık 27.28TWh harcamaktadır [4].

Bitcoin ve diğer kriptoparalar, kötü niyetli aktörlere karşı direnip meşru işlemleri doğrulamak için global defterlerinde fikir birliğine vararak işlev görürler. Bitcoin, fikir birliğini, "Proof of Work" (POW) denen ekonomik bir ölçü ile elde eder. PoW sistemlerinde kullanıcılar, *nonce* adı verilen bir sayıyı hesaplamak için rekabet ederler, öyle ki bütün bloğun hashi hedef aralığındadır. Bu geçerli aralık, bütün Bitcoin ağının geçerli bir nonce bulmaya harcadığı sürekli bir ortalama zamanı devam ettirmek için kullandığı kümülatif hesaplama gücü ile ters orantıldır. Geçerli bir nonce'un bulucusuna bloğu ekleme izni verilir, bu nedenle nonce hesaplamak için daha fazla bilişimsel kaynağı tüketenler blok zincirinin durumunda daha büyük bir rol oynamaktadır. PoW, bir varlığın, merkezsiz bir sistemde fazladan güç kazanmak için birden fazla varlık gibi davrandığı Sybil saldırılarına karşı koruma sağlar ve global bir veri yapısına erişim sağlarken doğal olarak ortaya çıkan yarış durumunu büyük ölçüde azaltır.

Alternatif bir fikir birliği protokolü olan "Proof of Stake" (PoS), ilk defa Peercoin tarafından 2012 de ortaya çıkarıldı [5].PoS bir sistemde katılımcılar, belli bir kriptopara içinde sahip oldukları servet miktarına eşit bir ağırlıkla oy kullanırlar. Bu düzenlemeyle, daha büyük bir finansal yatırıma sahip olanlara daha fazla güç verilir ve doğal olarak sistemin dürüstlüğünü korumaları için teşvik edilir, aksi takdirde yatırımlarını kaybetmeyi göze alacaklardır. PoS sistemi ayrıca düşük donanımda çalışan hafif bir yazılım gerektirerek güç israfı sorununu ortadan kaldırır.

Orijinal Nano (Raiblocks) raporu ve ilk beta uygulaması Aralık 2014 tarihinde yayınlandı ve Nano, Directed Acyclic Graph üzerine kurulu ilk kriptoparalardan biri haline geldi [6]. Kısa süre sonra, en bilinen örnekleri DagCoin/Byteball [7] ve IOTA [8] olan başka DAG kriptoparalar da geliştirilmeye başlandı. Bu DAG tabanlı kriptoparalar alışılagelmiş blok zinciri kalıplarını yıkarak sistem performansını ve güvenliği arttırdı. Byteball, dürüst, saygın ve kullanıcı tarafından güvenilir "şahitler" den oluşan bir "ana-zincir" e güvenerek konsensüse varırken, IOTA, istiflenmiş işlemlerin kümülatif PoW'i vasıtasıyla konsensüs sağlar. Nano, çakışan işlemler üzerinde hesap bakiyesi ağırlıklı bir oy birliği ile konsensüs sağlar. Bu çözüm sistemi daha güçlü, merkezi olmayan bir sistemi korurken daha hızlı, daha deterministik işlemler sağlar. Nano gelişimine devam etmektedir ve şimdiden kendini en



Skl. 1. Nano tipik işlemler için ek yük gerektirmez. Çakışan işlemler durumunda, düğümler hangi işlemi tutacaklarını oylarlar.

yüksek performans gösteren kriptoparalardan biri olarak konumlandırmıştır.

III. Nano Bileşenleri

Nano mimarisini size anlatmadan önce, sistemi oluşturan bileşenleri tanıyalım.

A. Hesap

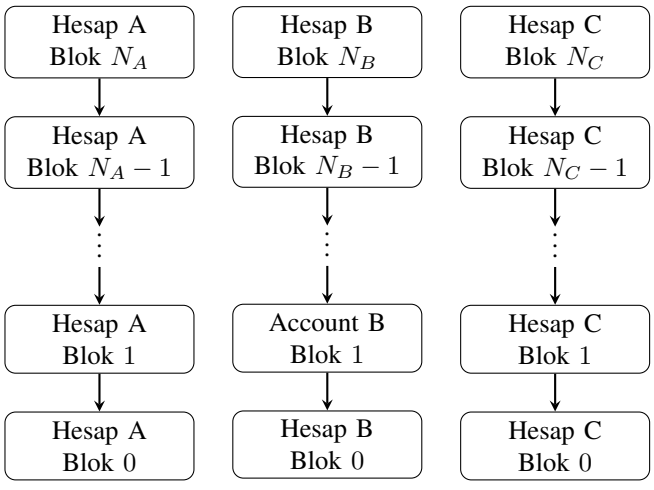
Hesap, anahtar çifti dijital imzasının açık-anahtar kısmıdır. Adres olarak da bilinen açık-anahtar diğer ağ katılımcıları ile paylaşılırken özel-anahtar gizli tutulur. Dijital olarak imzalanmış bir veri paketi, içeriğin özel-anahtar sahibi tarafından onaylandığını garantiler. Bir kullanıcı birçok hesabı kontrol edebilir, ancak her hesap için yalnızca bir açık adres olabilir.

B. Blok/İşlem

"Blok" ve "işlem" terimleri, bir bloğun tek bir işlemi içerdiği durumda genellikle birbirinin yerine kullanılır. İşlem spesifik olarak eylem belirtirken blok işlemin dijital olarak kodlanmasını belirtir. İşlemler, işlemin yapıldığı hesaba ait özel anahtar tarafından imzalanır.

C. Hesap Defteri

Hesap Defteri, her hesabın kendi işlem zincirinin bulunduğu global hesap dizisidir. (Şekil 2). Bu, tasarım zamanı anlaşmasıyla çalışma zamanı anlaşmasının değiştirilmesi kategorisine giren kilit bir tasarım bileşenidir; herkes, sadece bir hesap sahibinin kendi zincirini değiştirebileceğini dijital imza denetlemesi yolu ile kabul eder. Bu, görünüşte paylaşılan bir veri yapısı olan dağıtılmış hesap defterini, paylaşılmayanlar dizisine dönüştürür.



Skl. 2. Her hesapta, hesabın bakiye geçmişi içeren kendi blok zinciri bulunur. Blok 0 açık işlem olmalıdır. (Bölüm IV-B)

D. Düğüm

Düğüm, Nano protokolüne uyan ve Nano ağına katılan bir bilgisayarda çalışan yazılım parçasıdır. Yazılım, hesap defterini ve eğer varsa düğümün kontrol edebileceği hesapları yönetir. Düğüm, tüm hesap defterini veya her bir hesabın blok zincirinin yalnızca son birkaç bloğunu içeren budanmış geçmişini saklar. Yeni bir düğüm oluştururken, geçmişin tamamını doğrulamanız ve yerel olarak budamanız önerilir.

IV. Sistem Görünümü

Diğer şifreleme para birimlerinde kullanılan blok zincirlerin aksine, Nano *block-lattice* yapısını kullanır. Her hesabın, hesabın işlem / bakiye geçmişine eşdeğer kendi blok zinciri (hesap zinciri) vardır.(Şekil 2). Her bir hesap zinciri yalnızca hesap sahibince güncellenebilir; bu, her bir hesap zincirinin anında ve blok-lattice'in geri kalanı ile eşzamansız olarak güncellenmesini ve böylece işlemlerin hızlı gerçekleşmesini sağlar. Nano protokolü son derece hafiftir; her işlem, internet üzerinden iletilmek üzere gerekli minimum UDP paket boyutuna uymaktadır. Düğümlerin çoğu işlem için blokları sadece kaydetmek ve yeniden yayınlamak zorunda olmaları sayesinde düğümler için donanım gereksinimleri de minimaldir .(Şekil 1).

Sistem bir *başlangıç bakiyesi*'ne sahip *başlangıç hesabı* tarafından başlatılır. Başlangıç bakiyesi sabit bir miktardır ve asla arttırılamaz. Başlangıç bakiyesi bölünüp başlangıç hesab-zincirinde kayıtlı gönderme işlemleri ile diğer hesaplara gönderilir. Tüm hesapların bakiyelerinin toplamı ilk başlangıç hesap bakiyesini asla geçmeyerek sisteme, arttırılması mümkün olmayan nicelik üst sınırı verir .

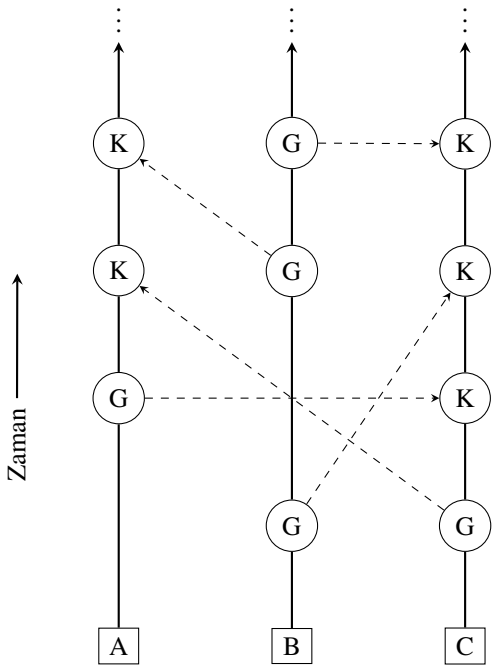
Bu bölüm, farklı işlem türlerinin ağ boyunca nasıl oluşturulduğunu ve yayıldığını anlatacaktır.

A. İşlemler

Bir hesaptan diğerine para aktarma işlemi iki işlem gerektirir: gönderilen miktarı gönderenin hesabından düşen *gönderme* ve bu miktarı alıcının hesabına ekleyen *alma*. (Şekil 3).

Gönderenlerin ve alıcının hesaplarında tutarların ayrı işlemler olarak aktarılması, birkaç önemli amaca hizmet eder:

1) Doğal olarak asenkron olmayan gelen aktarmaları sıralamak.



Skl. 3. Block-lattice teknolojisinin görseli. Her bir para transferi, hesap zincirinin sahibi (A, B, C) tarafından imzalanmış bir gönderme bloğu (G) ve bir alma bloğu (K) gerektirir.

2) İşlemleri UDP paketlerine uyacak şekilde küçük tutmak.
3) Veri izini en aza indirgeyerek defteri budama işlemini kolaylaştırmak.
4) Yerleşik işlemleri, yerleşik olmayan işlemlerden izole etmek.

Birden fazla hesabın aynı hedef hesaba göndermesi, eşzamansız bir işlemdir; ağ gecikmesi ve gönderen hesapların birbirleriyle iletişim halinde olmamaları, hangi işlemin daha önce gerçekleştiğinin bilinemeyeceği anlamına gelir. Toplama işleminin birleşme özelliği olduğundan dolayı lemlerin gerçekleşme sırasının bir önemi yoktur, bu nedenle tek ihtiyacımız olan küresel anlaşmadır. Bu, çalışma zamanı anlaşmasını tasarım zamanı anlaşmasına dönüştüren önemli bir tasarım bileşenidir. Alıcı hesap, hangi aktarımın önce geldiğine karar vermede kontrol sahibidir ve bunu gelen blokların imzalı sırası ile ifade eder.

Bir hesap birçok küçük transfer dizisi olarak alınacak büyük bir transfer yapmak isterse, bunu bir UDP paketi içine uyacak şekilde yansıtırız. Bir alıcı hesaba gelen transferleri sıraladığında, herhangi bir zamanda sabit bir boyutta işlemle herhangi bir miktarı aktarabilmek için hesap bakiyesinin toplamını tutar. Bu, Bitcoin ve diğer kripto para birimleri tarafından kullanılan girdi / çıktı işlem modelinden farklıdır.

Bazı düğümler, bir hesabın tam işlem geçmişi kaydetmek için kaynakları harcamaya ilgisizdir; sadece her hesabın şu andaki bakiyesiyle ilgilenirler. Bir hesap bir işlem yaparken, kümülatif bakiyesini kodlar ve bu düğümler sadece yeni bloğu takip ederek, doğruluğun korunup geçmiş verilerinin atılmasına olanak sağlar.

Tasarım-zaman sözleşmelerine odaklanmış olsa dahi, ağdaki kötü aktörleri tanımlama ve işleme tabi tutma nedeniyle işlemleri onaylarken bir gecikme penceresi vardır. Nano'daki sözleşmelere milisaniye ile saniye arasında hızlı bir şekilde ulaşıldığından, kullanıcıya iki farklı kategoride gelen işlemler sunuyoruz: yerleşik ve yerleşik olmayan. Yerleşik işlemler, bir hesabın alma blokları oluşturduğu işlemlerdir. Yerleşik olmayan işlemler henüz alıcının kümülatif dengesine dahil edilmemiştir. Bu, diğer kripto para birimlerindeki daha karmaşık ve yabancı teyit metriğinin yerine geçer.

B. Hesap Oluşturma

Hesap oluşturabilmek için, *açma* işlemi oluşturmalısınız (Şekil 4). Açma işlemi her hesap zincirinin daima ilk işlemidir ve fonlar ilk alındığında oluşturulabilir. *Hesap* alanı imzalama için kullanılan özel anahtardan türetilen ortak anahtarı (adres) depolar. *Kaynak* alanı ise , fonları gönderen işlemin hash'ini içerir. Hesap yaratmada, sizin adınıza oy verecek bir temsilci seçilmelidir; bu daha sonra istenirse değiştirilebilir. (Bölüm IV-F). Hesap kendisini kendi temsilcisi olarak ilan edebilir.

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

Skl. 4. Açma işleminin anatomisi

C. Hesap Bakiyesi

Hesap bakiyesi hesap defterinin kendisi içinde kaydedilir. Bir işlem tutarını kaydetmek yerine, (Bölüm IV-I) gönderme bloğundaki bakiye ve önceki bloğun bakiyesi arasındaki fark kontrol edilerek doğrulama yapılır. Alıcı hesap, daha sonra, yeni alınan blokta verilen son bakiye olarak ölçülen önceki bakiyeyi arttırabilir. Bu, yüksek miktarda blok indirirken işlem hızını arttırmak için yapılır. Hesap geçmişi talep edilirken tutarlar verilmiş olur.

D. Bir Hesaptan Başka Hesaba Gönderme

Bir adresten gönderme işlemi yapabilmeniz için hesabın açma bloğu olması ve dolayısıyla hesap bakiyesi olması gerekmektedir (Şekil 5). *Önceki* alan, hesap zincirindeki önceki bloğun hash'ini içerir. *Hedef* alanı, fonların gönderileceği hesabı içerir. Gönderme bloğu onaylandıktan sonra değiştirilemez. Bir kez ağa yayınlandığında, para gönderenin hesap bakiyesinden kesilir ve alıcı taraf bu fonları kabul etmek için bir blok imzalayana kadar *askıda kalır* . Bekleyen fonlar, gönderenin hesabından düştüğü ve gönderen işlemi iptal edemediğinden, işlem onay bekliyor gibi düşünülmemelidir.

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

Skl. 5. Gönderim işleminin anatomisi

E. İşlem Alma

Bir işlemi tamamlamak için, gönderilen fonların alıcısı, kendi hesap zincirinde bir alma bloğu oluşturmalıdır (Şekil 6). Kaynak alanı, ilişkili gönderme işleminin hash'ini belirtir. Bu blok oluşturulup yayınlandığında, hesabın bakiyesi güncellenir ve fonlar resmen hesaplarına geçer.

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

Skl. 6. Alma işleminin anatomisi

F. Temsilci Atama

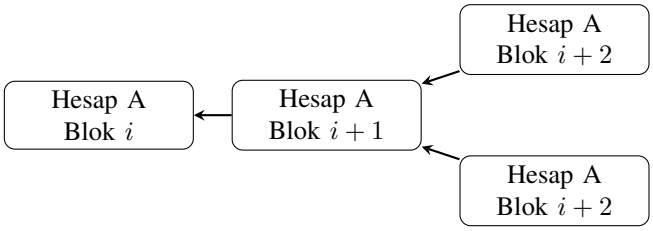
Hesap sahiplerinin kendi adına oy kullanacak bir temsilci seçme imkanı bulmaları Proof of Work ya da Proof of Stake protokollerinde karşılığı olmayan güçlü bir merkezsizleştirme aracıdır. Geleneksel PoS sistemlerinde, hesap sahibinin düğümü oylamaya katılmak için çalışıyor olmalıdır. Bir düğümü sürekli olarak çalıştırmak birçok kullanıcı için pratik değildir; bir temsilci olarak bir hesaba oy verme yetkisi bu şartı rahatlatır. Hesap sahipleri, konsensüsü herhangi bir hesaba yeniden atama hakkına sahiptir. Bir *değiştirme* işlemi, eski temsilciden oy ağırlığının çıkarılması ve ağırlığın yeni temsilciye eklenmesiyle bir hesabın temsilcisini değiştirir (Şekil 7). Bu işlemde fonlar yer değiştirmez ve temsilci, temsil ettiği hesabın fonlarını harcayamaz.

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

Skl. 7. Bir değişim işleminin anatomisi

G. Çatallar ve Oylama

İmzalı j ile $b_1, b_2, \ldots, b_j$ bloklarının selefleri ile aynı bloğu talep etmeleri durumunda çatallanma oluşur (Şekil 8). Bu bloklar, bir hesabın durumuyla çelişkili bir görüş oluşturur ve bunların çözülmesi gerekir. Yalnızca hesap sahibinin blokları hesap zincirine ekleme yeteneği vardır, dolayısıyla, bir çatal, başarısız bir programlamanın veya hesap sahibinin kötü niyetinin (çift harcama - double spending) sonucunda oluşur.



Skl. 8. Bir çatal, iki (veya daha fazla) imzalanmış blokların aynı önceki bloğa referans vermeleri durumunda oluşur. Eski bloklar solda; Yeni bloklar sağda

Tespit edildikten sonra, bir temsilci $\hat{b}_i$ bloğuna referans eden bir oyu kendi hesap defterine yaratır ve bunu ağa yayınlar. Bir düğümün oy ağırlığı w_i, onu temsilcisi olarak atayan tüm hesapların bakiyelerinin toplamıdır. Düğüm, diğer M çevrimiçi temsilcilerinden gelen oyları gözlemler ve toplam 4 oy periodu boyunca toplamda 1 dakika kümülatif bir çetele tutar ve kazanan bloğu onaylar. (Denklem 1).

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

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

En popüler blok b^* oyların çoğuna sahip olacak ve düğümün hesap defterinde tutulacaktır (Denklem 2). Oylamayı kaybeden blok(lar) atılır. Bir temsilci, hesap defterindeki bir bloğun yerini değiştirirse, daha yüksek bir sıra numarası olan yeni bir oylama yaratır ve yeni oyu ağa yayınlar. Bu temsilcilerin oy verdiği **tek** senaryodur.

Bazı durumlarda, kısa ağ bağlantı sorunları yayınlanan bir bloğun tüm ağ tarafından kabul edilmemesine neden olabilir. Bu hesaptaki sonraki tüm bloklar, ilk yayını görmeyen emsalleri tarafından geçersiz sayılır. Bu bloğun yeniden yayınlanması ile kalan emsalleri tarafından kabul edilecek ve sonraki bloklar otomatik olarak alınacaktır. Çatallanma veya eksik blok oluştuğunda bile, yalnızca işlemde rol alan hesaplar etkilenir; ağın geri kalanı, diğer tüm hesaplar için işlemleri işlemeye ile devam eder.

H. Proof of Work

Dört işlem türünün de doğru doldurulması gereken bir çalışma alanı vardır. Çalışma alanı, işlem yaratıcısının, alıcı / gönderme / değiştirme işlemlerinde önceki alana veya açma işlemindeki hesap alanına hash'i eklenecek olan nonce'ın belli bir eşik değerinin altında kalmasına olanak tanır. Bitcoinden farklı olarak PoW, Hashcash'e benzer şekilde anti-spam aracı olarak kullanılır ve saniyeler içerisinde hesaplanabilir [9]. Bir

işlem gönderildiğinde, önceki blok alanı bilindiği için sonraki bloğun PoW'u da önceden hesaplanabilir; işlemler arasındaki zaman PoW'yi hesaplamak için gereken zamandan daha uzun olduğu sürece, işlemler son kullanıcıya anında görünür hale gelecektir.

I. İşlem Doğrulaması

Bir bloğun geçerli sayılması için aşağıdaki niteliklere sahip olması gerekir:

1) Blok, önceden hesap defterinde olmamalıdır (yinelenen işlem).
2) Hesabın sahibi tarafından imzalanmış olmalıdır.
3) Önceki blok hesap zincirinin baş bloğudur. Önceden var fakat baş değilse, çataldır.
4) Hesabın bir açma bloğu olması gerekir.
5) Hesaplanan hash PoW eşiği gereksinimini karşılamaktadır.

Bir alıcı bloğuysa, kaynak bloğu hash'inin beklemede olup olmadığını, yani geri ödenmediğini kontrol edin. Bir gönderme bloğu ise, hesap bakiyesi, önceki bakiyeden daha az olmalıdır.

V. Saldırı Vektörleri

Nano, tüm merkezi olmayan kripto para birimleri gibi, kötü niyetli kişiler taraflar tarafından mali kazanç veya sistem kaybı gibi sebeplerle saldırıya uğrayabilir. Bu bölümde, olası olası saldırı senaryolarını, böyle bir saldırının sonuçlarını ve Nano protokolünün önleyici tedbirleri nasıl aldığını özetledik.

A. Boşluk-Engel Senkronizasyonu

IV-G bölümünde, bir bloğun düzgün şekilde yayınlanamadığı, ağın sonraki blokları yoksaymasına neden olan senaryoyu tartıştık. Bir düğüm, beliritilen önceki bloğun bulunmadığı bir blok tespit ederse, iki seçeneğe sahiptir:

1) Kötü amaçlı bir çöp bloğu olabileceği için bloğu yok sayma.
2) Başka bir düğümle yeniden senkronizasyon isteğinde bulunma.

Yeniden senkronizasyon durumunda, yeniden senkronun gerektirdiği artan trafik miktarını kolaylaştırmak için bir TCP bağlantının bir önyükleme (bootstrapping) düğümü ile oluşturulması gerekir. Bununla birlikte, eğer blok aslında kötü bir bloksa, o zaman yeniden senkronizasyon gereksizdir ve gereksiz yere ağdaki trafiği arttırmış olur. Bu, Bir Ağ Yükseltme Saldırısıdır ve hizmet reddine neden olur.

Gereksiz tekrar başlatılmasını önlemek için düğümler, senkronizasyon amacıyla önyükleme düğümüyle bağlantı kurmaya başlamadan önce, potansiyel olarak kötü amaçlı bir blok için belirli bir oy eşiğinin gözlemlenmesini bekleyecektir. Bir blok yeterince oy almıyorsa, önemsiz veri olduğu varsayılabilir.

B. İşlem Seli

Kötü amaçlı bir varlık, ağın doyurulması amacıyla kendi kontrolü altındaki hesaplar arasında gereksiz fakat geçerli işlemler gönderebilir. Hiçbir işlem ücreti olmadan bu saldırıyı süresiz devam ettirebilirler. Bununla birlikte, her işlem için gereken PoW, kötü niyetli varlığın hesaplama kaynaklarına önemli miktarda yatırım yapmaksızın verebileceği işlem oranını sınırlar. Defteri şişirmek için böyle bir saldırıda bile tam tarihsel düğüm olmayan düğümler eski işlemlerini kendi zincirlerinden temizleyebilir; bu, hemen hemen tüm kullanıcılar için depolama kullanımını bu tip bir saldırıdan etkisiz hale getirir.

C. Sybil Saldırısı

Bir varlık tek bir makinede yüzlerce Nano düğümü oluşturabilir; ancak, oylama sistemi hesap bakiyesine dayanılarak ağırlıklandırıldığından, ağa fazladan düğüm eklenmesi bir saldırganın ekstra oy kazanmasına neden olmaz. Dolayısıyla Sybil saldırısı ile kazanılan bir avantaj yoktur.

D. Kuruş-Harcama Saldırısı

Kuruş harcama saldırısı, bir saldırganın düğümlerin depolama kaynaklarını boşa harcamak için çok sayıdaki hesaba çok küçük miktarda para harcamasıdır. Blok yayıncılığı PoW tarafından sınırlandırılmıştır, bu durum hesapların ve işlemlerin oluşturulmasını belirli bir ölçüde sınırlandırır. Tam tarihsel olmayan düğümler, hesapların büyük olasılıkla geçerli bir hesap olmadığı istatistiksel bir metrikin altına budama yapabilir. Son olarak, Nano, minimum kalıcı depolama alanını kullanacak şekilde ayarlanmıştır; bu nedenle, bir ek hesap depolamak için gereken alan open block + indexing = 96B + 32B = 128B'dır. Bu, 8 milyon kuruş harcama hesabı saklayabilecek 1 GB'a eşittir. Düğümler daha agresif bir budama yapmak istediklerinde, erişim sıklığına dayalı bir dağılım hesaplayabilir ve nadiren kullanılan hesapları daha yavaş depolara devredebilirler.

E. Önceden hesaplanmış PoW Saldırısı

Hesap sahibinin hesap zincirine blok ekleyen tek varlık olması nedeniyle, sıralı bloklar, PoW ile birlikte ağa yayınlanmadan önce hesaplanabilir. Burada saldırgan, her biri düşük değere sahip sayısız ardışık bloğu uzun süre üretir. Belirli bir noktada, saldırgan, çok sayıda geçerli işlem üreterek ağda mümkün olduğunca çabuk yankılanan ve diğer düğümler tarafından da işlenen bir Hizmet Reddi (DoS) gerçekleştirir. Bu, V-B bölümünde tasvir edilen işlem selinin gelişmiş bir versiyonudur Bölüm V-B.Böyle bir saldırı sadece kısa süreliğine çalışır, ancak etkinliği artırmak için >50% Attack (Bölüm V-F) gibi diğer saldırılarla birlikte kullanılabilir. İşlem oranını sınırlayanlar ve diğer teknikler, şu anda saldırıları azaltmak için araştırılmaktadır.

F. >50% Saldırısı

Nano için konsensüs metriği bakiye ağırlıklı oylama sistemidir. Bir saldırgan oylama gücünün 50%'sinden fazlasına sahip olabilirse, ağın oybirliğinin salınıp sistemin bozulmasına neden olabilir. Bir saldırgan, iyi bir düğümün a DoS aracılığıyla oy vermesini engelleyerek, kaybedilen denge miktarını düşürebilir. Nano, böyle bir saldırıyı önlemek için aşağıdaki önlemleri alır:

1) Bu türden saldırılara karşı birincil savunma, oy ağırlığının sisteme yapılan yatırımla bağlantılı olmasıdır. Bir hesap sahibi kendi yatırımını korumak amacıyla sistemin dürüstlüğünü devam ettirmeye teşvik edilir. Defteri değiştirmeye çalışmanın tahrip edici etkisi, hesap sahibinin yatırımlarını da yok edecektir.
2) Bu saldırının maliyeti Nano'ların piyasa kapitalizasyonu ile orantılıdır. PoW sistemlerinde, parasal yatırımla karşılaştırıldığında orantısız denetim sağlayan teknoloji icat edilebilir ve eğer saldırı başarılı olursa, saldırı tamamlandıktan sonra bu teknoloji yeniden hazırlanabilir. Nano ile sisteme saldırmanın maliyeti sistemin kendisiyle orantılıdır ve bir saldırı başarılı olursa, saldırıya yapılan yatırım geri alınamaz.
3) Azami seçmen yetersayısını korumak için bir sonraki savunma hattı temsili oylamadır. Bağlantı nedenleriyle oylamaya güvenilir şekilde katılamayan hesap sahipleri, hesap bakiyelerinin ağırlığıyla oy kullanabilecek bir temsilci seçebilirler. Temsilcilerin sayısını ve çeşitliliğini en üst düzeye çıkarmak, ağ esnekliğini artırır.
4) Nano'taki çatallar kazara olmaz, bu nedenle düğümler çatallı bloklarla nasıl etkileşim kuracağına dair politika kararları verebilir. Saldırgan olmayan hesapların blok çatallarına karşı savunmasız olduğu tek durum, saldırgan bir hesaptan bakiye almalarıdır. Blok çatallarından güvenli olmak isteyen hesaplar, çatal üreten bir hesaptan almadan önce biraz bekleyebilir veya hiçbir zaman hiç almamayı tercih edebilir. Alıcılar, diğer hesapları izole etmek için şüpheli hesaplardan para alırken kullanılacak ayrı hesaplar da üretebilirler.
5) Henüz uygulanmayan son bir savunma hattı*blok çimentolama*dır. Nano, blok çatalılarını oylamayla hızla halletmek için büyük çaba harcar. Düğümler, blokların belirli bir süre sonra geri çevrilmesini önleyecek şekilde blok çimentolama için yapılandırılabilir. Ağ, belirsiz çatalları önlemek için hızlı yerleşim süresine odaklanarak yeterince güvenli bir durumda olur.

$> 50\%$ saldırısının daha sofistike bir hali burdadır Şekil 9. "Çevrimdışı", temsilci olarak seçilen ancak oy vermek için online olmayan temsilciler yüzdesidir."Stake", saldırganın birlikte oylayacağı yatırım miktarıdır."Aktive", çevrimiçi olan ve protokole göre oy kullanan temsilcilerdir. Bir saldırgan, diğer bir seçmenleri DoS saldırısı yoluyla offline duruma getirerek tahakkuk etmesi gereken miktarı denk getirebilir. Bu saldırı devam ederse, saldırıya uğramış temsilciler senkronizasyonlarını kaybederler ve bu "Unsync" ile gösterilir. Son olarak, bir saldırgan, önceki grup hesap defterini yeniden senkronize ederken Hizmet Reddi saldırılarını yeni bir temsilci grubuna geçirerek göreceli oylama gücünde kısa sürede bir kazanç elde edebilir; bu da, "Attack" tarafından gösterilir.

Offline	Unsync	**Attack**	Active	Stake

Skl. 9. 51% saldırı gereksinimlerini düşürebilecek olası bir oylama düzenlemesi.

Bir saldırgan, bu şartların bir kombinasyonu ile Stake >Active durumuna neden olabilirse, kendi paylarının bedeli karşılığında, hesap defterinin üstünde oyları başarıyla değiştirebilirler. Diğer sistemlerin piyasa kapitalizasyonunu inceleyerek bu tür saldırıların maliyetinin ne kadar olacağını tahmin edebiliriz. Temsilcilerin 33%'ü DoS aracılığıyla saldırıya uğramış veya çevrimdışı ise, bir saldırgan saldırabilmek için market kapitalizasyonunun 33%'üne sahip olmalıdır

G. Önyükleme Zehirlenmesi

Temsilcilerin veya bakiyelerin zamanla yeni hesaplara geçmesinden dolayı, bir saldırgan, eski özel anahtarı ne kadar uzun süre dengede tutabilirse, sahip olduğu hesap bakiyesinin katılımcı temsilcisinin olmama olasılığı da o kadar artar. Bir düğüm, saldırganın o sırada temsilcilere kıyasla oylama payı yetersayısı olan eski bir ağ temsilciliğine önyüklenebilirse, o düğümün oylama kararlarını değiştirilebilir. Bu yeni kullanıcı, saldıran düğümün dışında birisiyle etkileşim kurmak isterse, işlemlerinin tamamı, farklı baş bloklarına sahip olmalarından dolayı reddedilecektir. Net sonuç, düğümlerin ağdaki yeni düğümlerin zamanını, onları kötü bilgiler ile besleyerek, harcayabilmesidir. Bunu önlemek için düğümler ilk hesap veritabanı ve bilinen iyi blok başları ile eşleştirilebilir; bu, veritabanını başlangıç bloğuna kadar tekrar yüklemenin rine geçer. İndirme güncele ne kadar yakınsa, bu saldırıya karşı doğru şekilde savunma olasılığı da o kadar yüksektir. Sonuç olarak bu saldırı, güncel bir veritabanına sahip olan herhangi biriyle işlem yapamayacakları için, önyükleme sırasında düğümü önemsiz verileri beslemekten öte bir şey değildir.

VI. Uygulama

Şu anda referans uygulaması C ++'da yazılmıştır ve 2014'ten bu yana Github'da yeni sürümleri üretilmektedir. [10].

A. Dizayn Özellikleri

Nano uygulaması, bu yazıda özetlenen mimari standardına bağlı kalmaktadır. Ek spesifikasyonlar burada açıklanmaktadır.

1) İmzalama Algoritması: Nano, tüm dijital imzalar için değiştirilmiş bir ED25519 eliptik eğri algoritması ile Blake2b hashleme kullanır [11]. Hızlı imzalama, hızlı doğrulama ve yüksek güvenlik özellikleri için ED25519 seçildi.

2) Hashing Algoritması: Karma algoritması yalnızca ağ spamini önlemek için kullanıldığından algoritma seçimi madenciliğe dayalı kripto para birimleri ile karşılaştırıldığında daha az önemlidir. Uygulamamız blok içeriğine karşı Blake2b'yi bir özet algoritması olarak kullanmaktadır. [12].

3) Anahtar Türetme Fonksiyonu: Referans cüzdanında, anahtarlar bir parola ile şifrelenir ve ASIC kırma girişimlerine karşı koruma sağlamak için parola bir anahtar türetme fonksiyonu ile beslenir. Şu an Argon2 [13]esnek bir anahtar türetme fonksiyonu yaratmayı amaçlayan tek kamusal yarışmanın birincisidir.

4) Blok Aralığı: Her hesabın kendi blok zinciri olduğundan, güncellemeler ağ durumu ile eşzamansız yapılabilir. Bu nedenle blok aralıkları yoktur ve işlemler anında yayınlanabilir.

5) UDP Mesaj Protokolü: Sistemimiz mümkün olan en düşük miktarda bilgi işlem kaynağı kullanarak sınırsız olarak çalışacak şekilde tasarlanmıştır. Sistemdeki tüm iletiler durumsuz ve tek bir UDP paketine sığacak biçimde tasarlanmıştır. Bu ayrıca, kesintili bağlantıya sahip hafif eşlerinin, kısa vadeli TCP bağlantılarını yeniden kurmadan ağa katılmasını kolaylaştırır. TCP, yalnızca yeni eşler için, blok zincirlerini toplu haliyle önyüklemek istediklerinde kullanılır.

Düğümler, diğer düğümlerden gelen işlem yayın trafiğini gözlemleyerek, birkaç kopyanın kendine tekrar yansıtılması sayesinde, işlemlerinin ağ tarafından alındığından emin olabilir.

B. IPv6 ve Multicast

Bağlantısız UDP'nin üstünde kurulması, geleneksel çoklu trafik akışı ve oy yayınının yerini alması için gelecekte olabilecek IPv6 entegrasyonuna olanak tanır. Bu, ağ bant genişliği tüketimini azaltacak ve gelişmekte olan düğümlere daha fazla politika esnekliği sağlayacaktır.

C. Performans

Bu yazının yazıldığı tarihte, Nano ağı tarafından 4.2 milyon işlem gerçekleştirildi ve 1.7GB boyutunda bir blok zinciri elde edildi. İşlem süreleri saniyeler ile ifade edilecek cinsdedir. SSD'lerinde çalışan mevcut bir referans uygulaması, başlıca IO ile sınırlı olarak saniyede 10.000'den fazla işlemi işleyebilir.

VII. Kaynak Kullanimi

Bu bölüm, Nano düğümü tarafından kullanılan kaynakların bir özetidir. Ayrıca, özel kullanım örnekleri için kaynak kullanımını azaltmak için fikirler üzerinde duruyoruz. İndirgenmiş düğümlere genellikle hafif, budanmış veya basitleştirilmiş ödeme doğrulama (SPV) düğümleri denir.

A. Ağ

Bir düğümün ağ etkinliği miktarı, düğümün ağın sağlığına ne kadar katkıda bulunduğuna bağlıdır.

1) Temsilci: Temsilci bir düğüm, diğer temsilcilerin oy pusulasını gözlemleyip kendi oylarını yayınladığı için maksimum ağ kaynağı gerektirir.

2) Güvenilmez: Güvenilmez düğüm temsili bir düğüme benzer ancak yalnızca bir gözlemcidir, temsilci bir hesabın özel anahtarını içermez ve kendi oylarını yayınlamaz.

3) Güvenilir: Güvenilir bir düğüm, doğru bir şekilde konsensüs sağlanması için güvendiği bir temsilcinin oy trafiğini gözlemler. Bu, temsilcilerden bu düğüme giden oy trafiği miktarını azaltır.

4) Hafif: Hafif düğüm, en az ağ kullanımına izin veren, yalnızca ilgilendiği hesapların trafiğini gözlemleyen güvenilir bir düğümdür.

5) Önyükleme: Önyükleme düğümü, kendilerini çevrimiçi duruma getiren düğümler için defterin tamamın veya bir kısmın sunar. Gelişmiş akış denetimini gerektiren büyük miktarda veri içerdiğinden, UDP yerine TCP bağlantısı üzerinden yapılır.

B. Disk Kapasitesi

Kullanıcı taleplerine bağlı olarak, farklı düğüm yapılandırmaları farklı depolama gereksinimlerini gerektirir.

1) Tarihsel: Tüm işlemlerin tam bir geçmiş kaydını tutmak isteyen bir düğüm, maksimum miktarda depolama alanı gerektirir.

2) Güncel: Bloklarla birikmiş bakiyelerin tutulması tasarımı nedeniyle, düğümlerin fikir birliğine katılabilmesi için her bir hesap için en yeni veya en başta gelen blokları tutması yeterlidir. Eğer bir düğüm tam geçmişi saklamaya ilgisiz ise, yalnızca baş bloğunu tutmayı tercih edebilir.

3) Hafif: Hafif bir düğüm, yerel bir defter ile veri saklamaz ve sadece ilgilenen hesaplarda etkinliği gözlemlemek veya isteğe bağlı olarak tuttuğu özel anahtarlarla yeni işlemler oluşturmak için ağa katılır.

C. CPU

1) İşlem Yaratma: Yeni işlemler oluşturmak isteyen bir düğüm, Nano'nun kısma mekanizmasını geçebilmek için Proof of Work nonce'u üretmelidir. Çeşitli donanımların hesaplanması Ek A'da değerlendirilmiştir.

2) Temsilci: Bir temsilci blokların imzalarını doğrulamak, oy kullanmak ve fikir birliğine katılmak için kendi imzalarını üretmek zorundadır. Temsilci bir düğüm için gereken CPU kaynaklarının miktarı, işlem üretmekten çok daha azdır ve günümüz bilgisayarlarından herhangi bir CPU ile çalışmalıdır.

3) Gözlemci: Bir gözlemci düğümü kendi oylarını üretemez. İmza oluşturma için gerekli ek masraf çok düşük olduğundan, işlemci gereksinimleri temsilci düğüm çalıştırmakla hemen hemen aynıdır.

VIII. Sonuç

Bu yazıda, yeni block-lattice yapısı ve temsilcili Proof of Stake oylaması kullanan ve aracısız, işlem ücreti olmayan, düşük gecikmeli bir kriptoparayı temel hatları ile sizlere sunduk. Ağ, minimum kaynak gerektirir, yüksek güç gerektiren madencilik donanımı gerektirmez ve işlem hacmi yüksektir. Tüm bunlar, her hesabın kendi blok zincirine sahip olması ile sağlanır ve erişim sorunları ve küresel veri yapısının verimsizliklerini ortadan kaldırır. Sistem üzerinde olası saldırı vektörlerini tespit ettik ve Nano'nun bu saldırı türlerine karşı nasıl direneceğini gösteren argümanları sizlere sunduk.

Ek A
PoW Donanim Testleri

Daha önce belirtildiği gibi, Nano'daki PoW'un amacı ağ spamini azaltmaktır. Düğüm uygulamanız OpenCL uyumlu GPU'lardan yararlanabilecek ivme sağlar. Tablo I'de bazı donanımların karşılıklı değerlendirmeleri verilmiştir. Şu anda PoW eşiği sabittir, ancak uyarlanabilir bir eşik, ortalama bilgi işlem gücü arttıkça entegre edilebilir.

Teşekkür

Bu yazıyı düzenlediği için Brian Pugh'a teşekkür ederiz.

TABLO I
DONANIM POW PERFORMANSI

Cihaz	Saniyedeki İşlem Sayısı
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

REFERANSLAR

[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