Bitcoin: Eşlerarası Elektronik Nakit Sistemi[1]

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

Özet. Tamamen eşten eşe çalışan bir elektronik nakit sistemi çevrimiçi ödemeleri herhangi bir finansal kuruluştan geçmeden bir taraftan diğerine doğrudan gönderilmesini mümkün kılar. Dijital imzalar çözümün bir parçasını sağlar ancak çifte harcamaları önlemek için hala güvenilir bir üçüncü tarafa ihtiyaç duyuluyorsa asıl avantajlarını kaybederler. Çifte harcama problemine eşler arası ağ kullanarak bir çözüm öneriyoruz. Ağ, işlemleri sürekli büyüyen bir kriptografik özet (hash) tabanlı iş kanıtı zincirine ekleyerek zaman damgasıyla mühürler ve iş kanıtını tekrarlamadan değiştirilemez bir kayıt meydana getirir. En uzun zincir sadece tanıklık edilen olaylar dizisinin kanıtı olarak hizmet etmez aynı zamanda en büyük işlemci gücü havuzundan geldiğini de kanıtlar. İşlemci gücünün çoğunluğu ağa saldırmak için işbirliği yapmayan düğümlerin (node) kontrolünde olduğu sürece en uzun zinciri üretecekler ve saldırganların önüne geçecekler. Ağın kendisi asgari yapıya ihtiyaç duyar. Mesajlar en üst çabayla yayımlanır ve düğümler dilediklerinde ağdan ayrılabilirler ve dışarıda geçirdikleri sürede yapılan işlemlerin kanıtı olan en uzun iş kanıtı zincirini kabullenerek yeniden katılabilirler.

1. Giriş

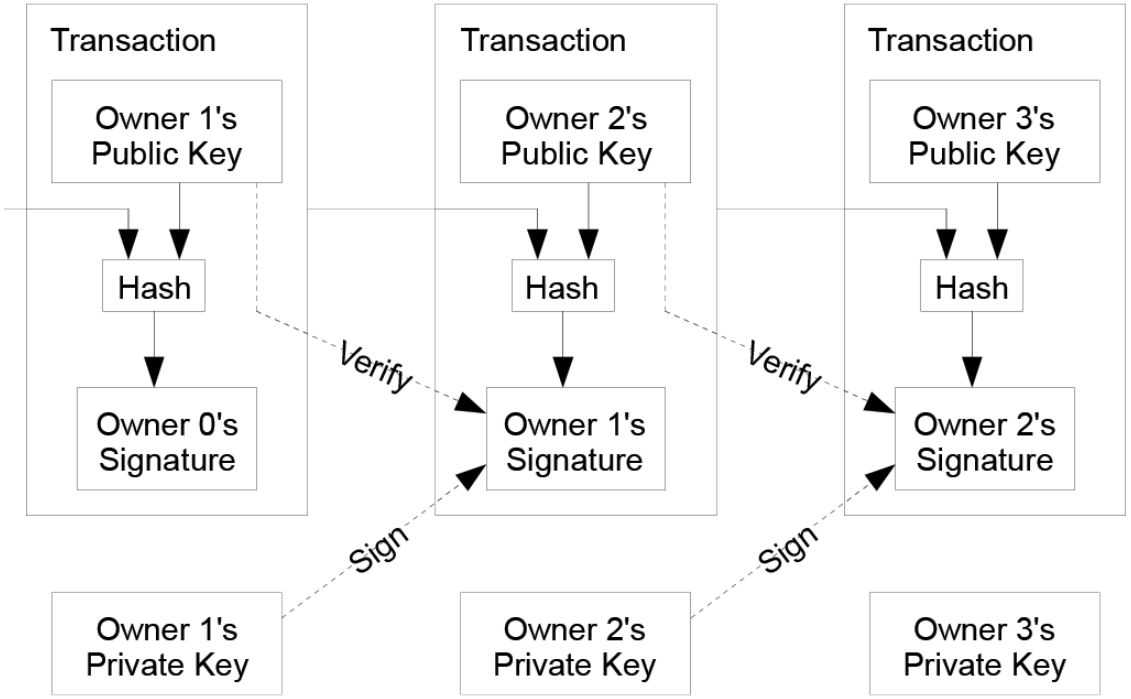
İnternet üzerinde ticaret, elektronik ödemeleri işleyen güvenilir üçüncü partiler olarak hizmet veren finansal kuruluşlara bağlıdır. Sistem, çoğu işlem için yeterince iyi çalışıyor olsa da, güven temelli modelde bulunan aslında var olan zayıf yönleri hala taşımaktadır. Finansal kuruluşlar uyuşmazlıklarda uzlaşıdan kaçındıklarından, tamamen geri döndürülemez işlemler mümkün değildir. Uzlaşmanın maliyeti işlem maliyetlerini artırır, minimum uygulanabilir işlem ölçüsünü sınırlandırır ve küçük rastgele ödemelerin mümkün olması olasılığının önünü keser ve geri döndürülemez hizmetler için geri döndürülemez ödemeler yapma imkanının olmamasının da önemli maliyetleri vardır. İşlemi geri döndürme ihtimali güvenme ihtiyacını artırır. Tüccarlar, ihtiyaç duymayacakları bilgileri aldıkları için rahatsızlık duyan müşterilerine karşı dikkatli olmalıdır. Belirli ölçülerde sahtekarlık kaçınılmaz olarak kabul edilir. Bu maliyetler ve ödeme belirsizlikleri fiziksel para kullanımı ile önlenebilir, ancak güvenilir bir üçüncü taraf olmadan iletişim kanalları üzerinden ödeme yapma mekanizması yoktur.

İhtiyaç duyulan şey; güven yerine şifreleme kanıtı üzerine kurulu, iki tarafın birbiri ile doğrudan bağlantılı olduğu elektronik bir ödeme sistemidir. işlemi tersine çevirmek için hesaplaması zor olan işlemler tarafları dolandırıcılıktan koruyacaktır. Bu makalede, çifte harcama sorununa, işlemlerin kronolojik sıralamasının hesaplama kanıtı oluşturmak için eşler arası dağıtık bir zaman damgası sunucusu kullanarak bir çözüm öneriyoruz. Bu sistem, ağdaki dürüst düğümler, ortak çalışan saldırgan düğümlerden daha fazla işlemci gücünü kontrol ettiği sürece güvenlidir.

[1] Satoshi Nakamoto rumuzuyla 30 Ekim 2008 tarihinde yayınlanan "Bitcoin: A Peer-to-Peer Electronic Cash System" başlıklı makale, aslına sadık kalınarak Türkçeleştirilmiştir. Orijinal dokümana https://bitcoin.org/bitcoin.pdf adresinden ulaşabilirsiniz

2. İşlemler

Bir elektronik parayı bir dijital imza zinciri olarak tanımlıyoruz. Her işlem sahibi parayı, bir önceki işlemin kriptografik özeti ve bir sonraki sahibinin açık anahtarıyla işlemi imzayıp bunu paranın sonuna ekleyerek, bir sonrakine aktarır. Ödemeyi alan sahiplik zincirini doğrulamak için imzaları doğrulayabilir.

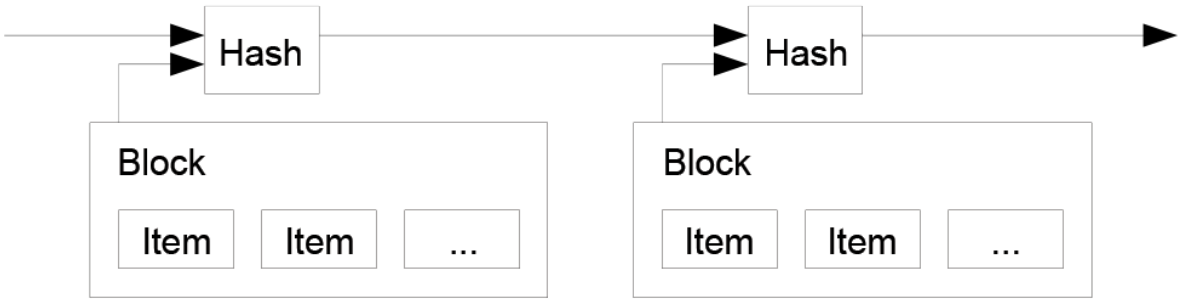


Buradaki sorun, tabii ki ödemeyi alanın, birinin parayı çifte harcama yapmadığını doğrulayamamasıdır. Bu sorun için yaygın çözüm, mükerrer harcamaları kontrol eden bir merkezi otoriteyi ya da darphaneyi dahil etmektir. Her işlemden sonra para, yenisinin üretilmesi için darphaneye geri döndürülmelidir ve ancak sadece direkt olarak darphane tarafından üretilen paraların çifte harcanmadığına güvenilir. Ancak bu çözümdeki sorun; tüm para sisteminin kaderinin, tüm işlemlerin tıpkı bir banka gibi, darphaneyi işleten şirketin üzerinden geçmesi zorunluluğuna bağlı olmasıdır.

Ödeme alıcısının, önceki sahiplerinin daha önceki işlem imzalamamış olduğunu bilmesi için bir yola ihtiyacımız var. Bizim için geçerli olan işlem, en önce gerçekleşendir, bu sayede sonraki çifte harcama girişimlerini önemsemeyiz. Bir işlemin yokluğunu doğrulamanın tek yolu tüm işlemlerin farkında olmaktır. Darphane tabanlı modelde, darphane tüm işlemlerin farkındadır ve hangisinin önce geldiğine karar verir. Bunu güvenilir bir otorite olmadan yapabilmek için, tüm işlemlerin halka açık yayınlanmalıdır [1] ve katılımcılar için her işlemin geliş sırasından birinde uzlaştıkları bir sisteme ihtiyacımız vardır. Ödemeyi alan, her işlem gerçekleştiği an, düğümlerin çoğunluğunun işlemin ilk önce geldiğini kabul ettiği kanıta ihtiyaç duyar.

3. Zaman Damgası Sunucusu

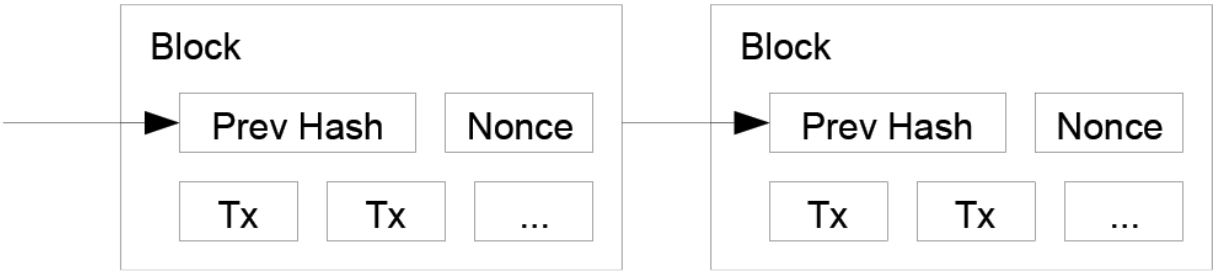
Önerdiğimiz çözüm, bir zaman damgası sunucusu ile başlar. Bir zaman damgası sunucusu, zaman damgalanacak mesaj bloğunun kriptografik özetinin alınması ve kriptografik özetin, ağ geneline tıpkı bir gazete ya da Usenet gönderisi gibi [2-5] yayınlanmasıyla çalışır. Zaman damgası, o verinin, o zamanda, kriptografik özeti alabilmek için açıkça sıralı olarak, var olması gerektiğini kanıtlar. Her bir zaman damgası, kriptografik özetinin içerisinde, zincir oluşturacak şekilde bir önceki zaman damgasını içerir ve eklenen her zaman damgası kendinden öncekileri güçlendirir.



4. İş Kanıtı

Dağıtık bir zaman damgası sunucusunu eşten eşe esasıyla uygulamak için, gazeteden veya Usenet gönderilerinden ziyade Adam Back'in Hashcash'ine [6] benzer bir İş Kanıtı (PoW) sistemi kullanmamız gerekecek. İş kanıtı, bir değerin SHA-256 gibi şifreleme metoduyla kriptografik özeti alındığında, bu kriptografik özet, belirli sayıda sıfır biti ile başlamasını tarar. Gerek duyulan ortalama iş, gerek duyulan sıfır bit sayısı ile üstel olur ve tek bir kriptografik özet çalıştırılarak doğrulanabilir.

Bizim zaman damgası ağımızda, iş kanıtını, bloğun kriptografik özetinde istenen sıfır değerine ulaşana kadar arttırdığımız bir "nonce" değeri ile sağlarız. İşlemci (CPU) iş kanıtını sağlayacak işlemleri bir kez yaptıktan sonra, aynı işlemler yapılmadan blok bir daha değiştirilemez. Sonraki bloklar zincire eklendikten sonra, önceki blok değiştirildiğinde, sonra gelen tüm blokların da aynı şekilde değiştirilmesi gerekir.



İş kanıtı, aynı zamanda çoğunluğun kararının benimsenmesi sorununu da çözer. Eğer çoğunluk, bir IP adresinin bir oy verebildiği bir sisteme bağlı olsaydı, bu, çok sayıda IP adresi edinebilen herhangi biri tarafından altüst edilebilirdi. İş kanıtı, esas olarak "bir işlemci bir oy"dur. Çoğunluk kararı, en büyük kanıtlama çalışması eforuna sahip en uzun zincir tarafından temsil edilir. CPU gücünün büyük bir çoğunluğu dürüst düğümlerle kontrol edilirse, dürüst zincir en hızlı büyüyecek ve rakip zincirleri aşacaktır. Geçmişteki bir bloğu değiştirmeye çalışan bir saldırganın, o bloğun ardından gelen tüm blokları da değiştirmesi gerekecek, ve akabinde mevcutta dürüst düğümlerden kurulu bir zincirden daha uzun bir zincir oluşturması gerekecektir. Daha sonra inceleyeceğimiz üzere, daha yavaş kalacak olan bir saldırganın bloklar eklendikçe başarılı olma ihtimali katlanarak azalacaktır. Zaman içinde artan donanım hızları ve ve artan düğüm sayısına uygun şekilde, her bir blok için iş kanıtı zorluk seviyesi hareketli ortalama metodu kullanılarak saatlik olarak yeniden hesaplanır. Çok hızlı oluşturulursa zorluk da artar.

5. Ağ

Ağı çalıştırmak için gereken adımlar şunlardır:

1. Yeni işlemler tüm düğümlere yayınlanır.
2. Her düğüm yeni işlemleri bir blok içine toplar.
3. Her düğüm kendi bloğu içinde zor bir iş-kanıtı bulmaya çalışır.
4. İş-kanıtını bulan düğüm, diğer tüm düğümlere bu kanıtı yollar.

5. Düğümler, sadece blok içindeki tüm işlemler geçerliyse ve daha önce harcanmamışsa bloğu kabul eder.
6. Düğümler, bloğun kabulünü, zincirdeki bir sonraki bloğu oluşturmak için, kabul edilmiş önceki bloğun kriptografik özetiyle (hash) çalışarak ortaya koyarlar.

Düğümler her zaman en uzun zincirin doğru olduğunu varsayar ve onu genişletmek için çalışmaya devam eder. Eğer iki farklı düğüm, bir sonraki bloğun farklı versiyonlarını eş zamanlı olarak yayınlarsa, bazı düğümler bunlardan herhangi birini diğerinden önce alabilir. Bu durumda düğümler ilk aldıkları üzerinde çalışmaya devam ederler, ancak daha fazla uzama ihtimaline karşı diğer çatalı da saklarlar. Beraberlik, bir sonraki iş-kanıtı bulunduğunda ve çatallardan birisi daha uzun olduğunda bozulmuş olacaktır; diğer çatalda çalışan düğümler de uzun olana geçecektir.

Yeni işlem yayınlarının tüm düğümlere ulaşmasına gerek yoktur. Birçok düğüme eriştikleri sürece çok beklemeden bir bloğa dahil olacaklardır. Blok yayınları mesaj kayıplarına karşı toleranslıdır. Eğer bir düğüm bir bloğu alamazsa, onu, bir sonraki bloğu aldığında talep eder ve bir tanesini kaçırdığını fark eder.

6. Teşvik

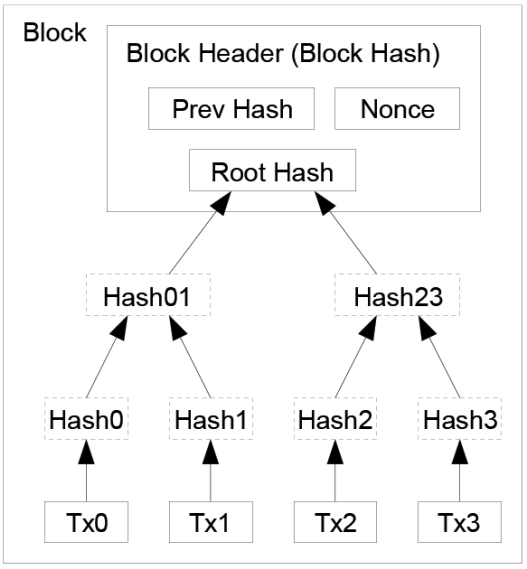
Kurallara göre, blok içindeki ilk işlem, bloğu yaratanın, sahip olduğu yeni kripto parayı ortaya çıkaran özel bir işlemdir. Bu, düğümlere ağı desteklemek için iş yapma isteği verir ve üretecek herhangi bir merkezi otorite olmadığından, kripto paraları, ilk defa dolaşıma sokmak için bir yol sağlar. Sabit miktarda yeni kripto paraların düzenli olarak dolaşıma eklenmesi, altın madencilerinin altını dolaşıma ekleyebilmek için kaynaklarını kullanmasına benzemektedir. Bizim olayımızda, bu, harcanan işlemci gücü ve elektriktir.

Teşvik, işlem ücretleri ile de karşılanabilir. Bir işlemin çıktı değeri girdi değerinden düşükse, aradaki fark, işlemi içeren bloğun teşvik değerine eklenen bir işlem ücretidir. Önceden belirlenen sayıda kripto para dolaşıma girdiğinde, işlemler sadece işlem ücretleri ile gerçekleşebilir ve tamamen enflasyondan bağımsız hale gelebilir.

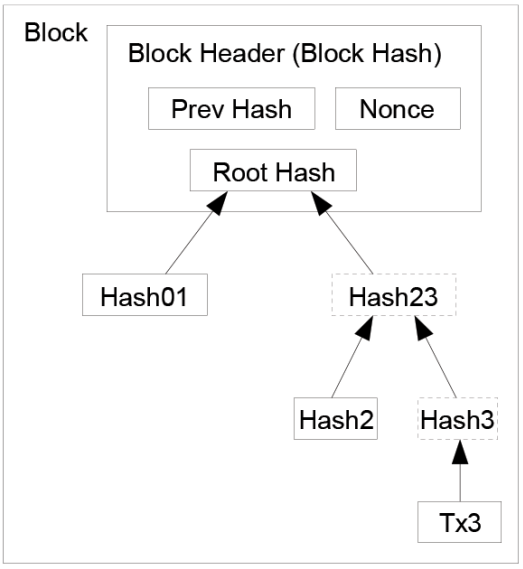
Teşvik, düğümlerin dürüst kalması için cesaretlendirmeye yardımcı olabilir. Açgözlü bir saldırgan tüm dürüst düğümlerden daha fazla işlemci gücü toplayabilirse, ödemelerini geri çalarak insanları dolandırabilir veya yeni kripto paralar üretmek için onu kullanabilir. Aslında sistemin ve kendi zenginliğinin geçerliliğini zayıflatmak yerine, ona diğerlerine göre daha çok kripto para ile ödüllendirecek kurallara göre oynamayı daha kârlı bulmalıdır

7. Disk Alanından Tasarruf

Bir kripto paranın son işlemi yeteri kadar bloğun altında kaldığında, önceki harcama işlemleri boş disk alanını korumak için silinebilir. Bunu bloğun kriptografik özetini bozmadan kolayca gerçekleştirebilmek için işlemlerin sadece köklerini içeren kriptografik özeti Merkle Ağacı'na [7][2][5] alınır. Eski bloklar, ağacın dalları boşaltılarak sıkıştırabilir. Dahili kriptografik özetlerin depolanması gerekmez.



İşlemlerin kriptografik özetinin Merkel Ağacı'na alınması

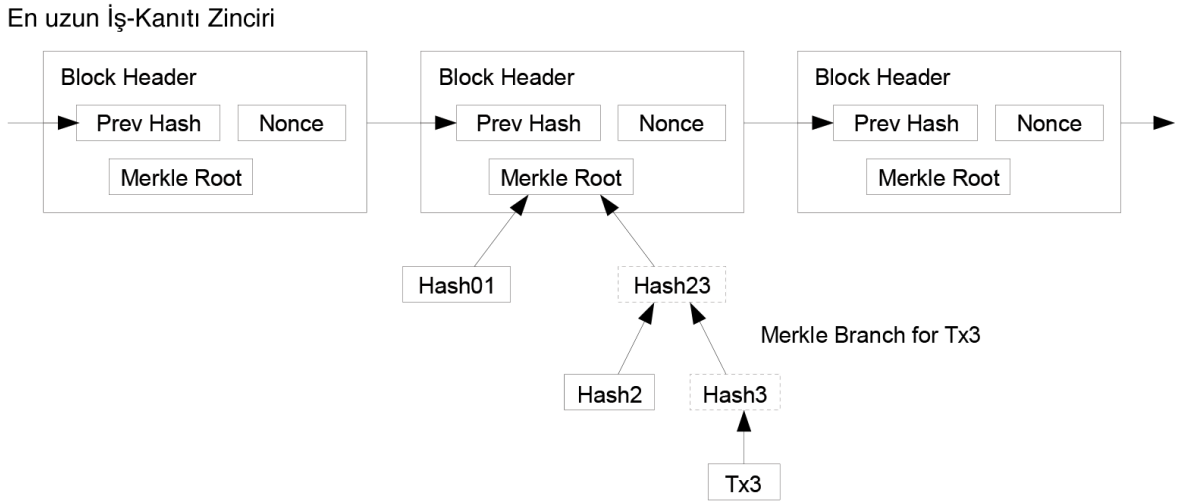


Sıkıştırmanın ardından Bloktaki 0-2 İşlemleri

İşlem içermeyen blok başlığı yaklaşık 80 bayt olacaktır. Her 10 dakikada bir blok üretildiğini varsayarsak, 80 bayt * 6 * 24 * 365 = yılda 4,2MB. 2008 yılı itibariyle 2GB RAM ile satılan bilgisayar sistemleri ve Moore Yasası'yla öngörüldüğünde şu an yıllık 1,2GB büyüme ile blok başlıkları bellekte saklansa bile depolama sorun olmayacaktır.

8. Basitleştirilmiş Ödeme Doğrulaması

Tam düğüm çalıştırmadan ödemeleri doğrulamak mümkündür. Kullanıcının sadece, en uzun zincir olduğundan emin oluncaya kadar ağ düğümlerini sorgulayarak bulacağı ve işlemin zaman damgalanmış bloğa bağlandığı Merkle dalını elde edeceği, ağdaki en uzun iş kanıtı doğrulama zincirinin blok başlıklarının bir kopyasını saklaması yeterlidir. İşlemi kendisi kontrol edemez, ancak zincirdeki bir yere bağlayarak ağdaki düğümlerin bunu onayladığını ve ağın onu kabul ettiğini doğruladıktan sonra blokları eklediğini görebilir.

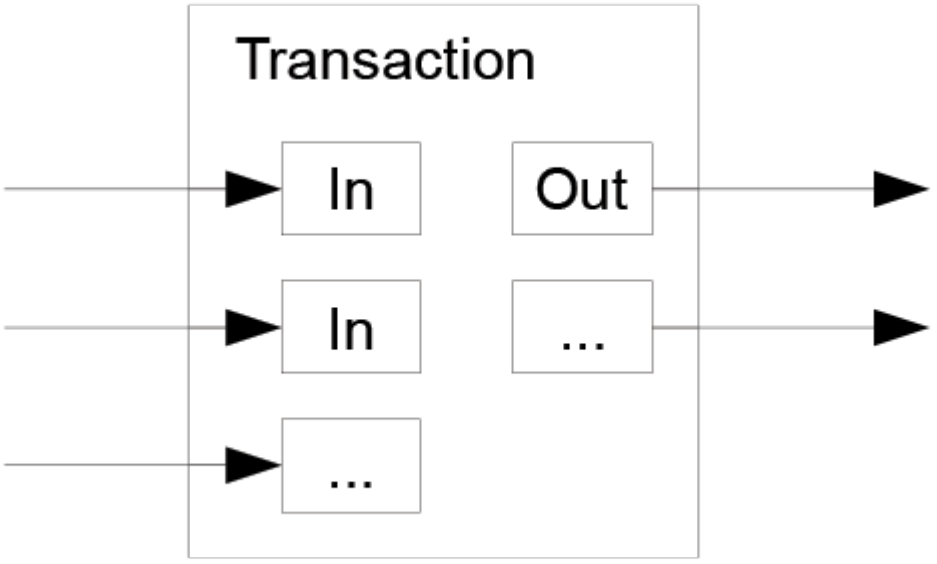


Bu nedenle, dürüst düğümler ağı kontrol ettiği sürece doğrulama güvenilirdir fakat ağ, bir saldırgan tarafından daha fazla işlemci gücüyle etkisiz hale getirilirse tehlikeye daha açıktır. Ağ düğümleri işlemleri kendi başlarına doğrulayabiliyorken, basitleştirilmiş metod, saldırgan daha fazla işlemci gücüyle ağa hakim olmaya devam ettiği sürece saldırganın uydurduğu işlemlerle saptırılabilir. Ağ düğümleri hatalı bir blok ile karşılaştığında gelen alarmları kabul etmek, kullanıcı yazılımlarından alarm verilen işlemleri ve tüm bloğu yüklemesini istemek ve

tutarsızlığı doğrulamak bundan korunmak için bir karşı strateji olabilir. Daha sık ödeme alan işletmeler, muhtemelen, daha hızlı onay ve daha bağımsız güvenlik için kendi düğümlerini çalıştırmayı isteyeceklerdir.

9. Değerleri Birleştirme ve Bölme

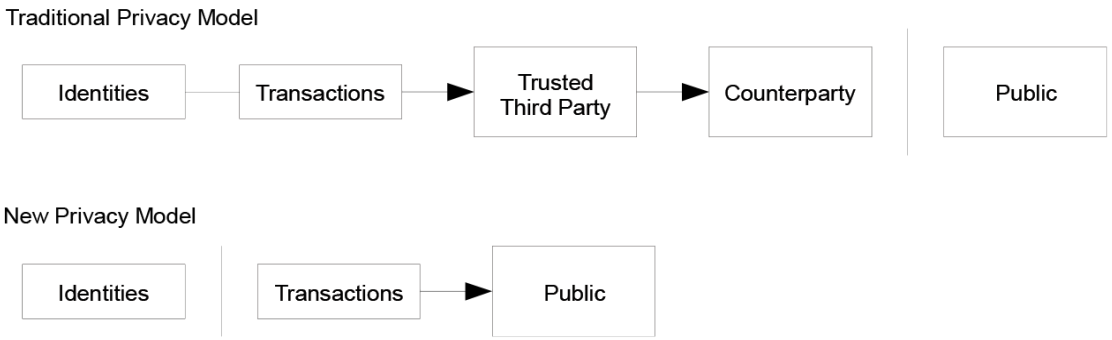
Kripto paraları ayrı ayrı işlemek mümkün olsa da, bir transferde her kuruş için ayrı bir işlem yapmak kullanışsız olurdu. Değerin bölünmesi ve birleşmesine izin vermek için işlemler, çoklu giriş ve çıkışlar içerir. Normalde, önceki daha büyük işlemden gelen tek bir giriş veya daha küçük miktarları birleştiren çoklu girişler ve en fazla iki çıkış olur: biri ödeme için ve eğer varsa gönderene geri giden para üstü iadesi.



Bu yelpazede, bir işlemin birden çok işleme ve bu işlemlerin de daha bir çoğuna bağlı olmasının bir sorun olmadığı dikkate alınmalıdır. Bu işlem geçmişinin eksiksiz bir dökümünün çıkarılmasına ihtiyaç yoktur.

10. Gizlilik

Geleneksel bankacılık modeli, taraflara ve güvenilir üçüncü partiye bilgi erişimi sınırlandırarak bir gizlilik seviyesine ulaşır. Tüm işlemlerin herkesin erişimine açılması gerekliliği bu metodu bertaraf eder ancak gizlilik, bilgi akışını başka bir yerde kırarak sağlanabilir: genel anahtarları anonim tutarak. Herkes, işlemi herhangi biriyle ilişkilendirmeden, birinin başka birine bir tutar gönderdiğini görebilir Bu, borsalar tarafından yayınlanan, işlemlerin ayrı ayrı zaman ve büyüklük bilgisinin, bilgi bandının, tarafların kim olduğu söylenmeden herkese açık olduğu işlem bilgileri seviyesine benzerdir.



Ek bir güvenlik duvarı olarak, ortak bir sahiple ilişkilendirilmesinin önüne geçmek amacıyla her işlem için yeni bir anahtar çifti kullanılmalıdır. Birden çok girdili işlemlerde, girdilerin aynı sahip tarafından sahiplenildiğinin mecburen açıklandığı işlemlerde bu kaçınılmazdır. Buradaki risk, bir anahtarın sahibi ortaya çıkarsa, ilişkilendirme, aynı sahibe ait diğer işlemleri de ortaya çıkarabilir.

11. Hesaplamalar

Bir saldırganın, dürüst zincirden daha hızlı alternatif bir zincir oluşturmaya çalıştığı bir senaryo ele alalım. Eğer bu gerçekleşse bile, bu durum sistemi, saldırganın bir anda ortaya çıkan bir değer yaratması ya da kendine ait olmayan bir parayı üretmesi gibi keyfi değişikliklere açık hale getirmez. Düğümler, geçersiz bir işlemi ödeme olarak kabul etmeyecek ve dürüst düğümler bunları içeren bir bloğu asla kabul etmeyecektir. Bir saldırgan, yalnızca yakın zamanda harcadığı parayı geri alabilmek için kendi işlemlerinden birini değiştirmeye çalışabilir.

Dürüst zincir ve bir saldırganın zinciri arasındaki yarış Binomiyal Rassal Yürüyüş (Binomial Random Walk) ile karakterize edilebilir. Başarı durumu, dürüst zincirin liderliğinin +1 artırılması ile bir blok genişlemesi, başarısızlık durumu ise, saldırgan zincirin aradaki farkın -1 azaltılarak bir blok genişlemesidir

Saldırganın belirli bir açığı yakalama olasılığı, bir Kumarbazın İflası (Gambler's Ruin) problemine benzetilebilir. Sınırsız kredisi olan bir kumarbazın bir borçla oyuna başladığını ve başabaş noktasına ulaşmaya çalışmak için sonsuz deneme hakkı olduğunu varsayalım. Kumarbazın başabaş noktasına ulaşabilme ihtimalini ya da saldırganın dürüst zinciri yakalayabilme ihtimalini şu şekilde hesaplayabiliriz[8]:

p= dürüst bir düğümün bir sonraki bloğu bulma ihtimali

q = saldırgan bir sonraki bloğu bulma ihtimali

qz = saldırganın z blok geriden gelip yakalama ihtimali

$$q_z = \begin{cases} 1 & if\ p \le q \\ (q/p)^z & if\ p > q \end{cases}$$

Belirtilen p > q varsayımına göre, saldırganın yakalamak zorunda olduğu blok sayısı arttıkça yakalama olasılığı katlanarak düşer. Ona karşı ihtimaller nedeniyle, eğer zincirin ilk halkalarında şanslı bir atak yapmazsa, dürüst zincir uzadıkça şansı giderek azalır ve çok geride kalır.

Yeni bir işlemin alıcısının, gönderenin işlemi değiştiremeyeceğinden yeterince emin olana kadar ne kadar beklemesi gerektiğini şimdi ele alalım. Bir süreliğine gönderenin, alıcıyı, ona ödeme yaptığına inandırmak isteyen ve bir süre sonra kendisine geri ödeme yapmayı amaçlayan bir saldırgan olduğunu varsayalım. Böyle bir durumda alıcı uyarılacaktır ancak saldırgan da bunun çok geç olacağını ümit etmektedir.

Alıcı yeni bir anahtar çifti üretir ve imzalamadan kısa bir süre önce genel anahtarı gönderene yollar. Bu, gönderenin, daha önceden öne geçmek için şanslı bir ana kadar üzerinde devamlı çalışarak zincir halinde uzun bir blok oluşturmasının ve o anda işlemi gerçekleştirmesinin önüne geçmiş olur. İşlem bir kere gönderildikten sonra, dürüst olmayan gönderici işleminin alternatif sürümünü içeren paralel bir zincir üzerinde gizli bir şekilde çalışmaya başlar.

Alıcı, işlem bir bloğa bağlanana ve ardından “z” sayısı kadar blok da bu bloğa eklenene kadar bekler. Alıcı, saldırganın ne kadar ilerleme kaydettiğini bilmez, ama dürüst blokların her blok başına ortalama beklenen süreyi aldığını varsayarak, saldırganın potansiyel ilerlemesi “Poisson Dağılımı”na göre beklenen bir değer alacaktır:

$$\lambda = z\frac{q}{p}$$

Saldırganın hala yetişebilme olasılığına ulaşmak için, o noktadan yetişebileceği ihtimali ile yapabileceği her ilerleme miktarının Poisson yoğunluğunu çarpmak gerekir:

$$\sum_{k=0}^{\infty} \frac{\lambda^k e^{-\lambda}}{k!} \cdot \begin{cases} (q/p)^{(z-k)} & \text{if } k \le z \\ 1 & \text{if } k > z \end{cases}$$

Dağılımın sonsuz kuyruğunun toplamından korunmak için tekrar düzenliyoruz:

$$1 - \sum_{k=0}^{z} \frac{\lambda^k e^{-\lambda}}{k!} \left(1 - (q/p)^{(z-k}\right.$$

C koduna dönüştürelim:

```
#include <math.h>
   double AttackerSuccessProbability(double q, int z)
   {
       double p = 1.0 - q;
       double lambda = z * (q / p);
       double sum = 1.0;
       int i, k;
       for (k = 0; k <= z; k++)
       {
           double poisson = exp(-lambda);
           for (i = 1; i <= k; i++)
```

```
                poisson *= lambda / i;
            sum -= poisson * (1 - pow(q / p, z - k));
        }
    return sum; }
```

Bazı sonuçlar için çalıştırdığımızda, olasılığın “z” ile katlanarak düştüğünü görebiliriz.

```
q=0.1
  z=0    P=1.0000000
  z=1    P=0.2045873
  z=2    P=0.0509779
  z=3    P=0.0131722
  z=4    P=0.0034552
  z=5    P=0.0009137
  z=6    P=0.0002428
  z=7    P=0.0000647
  z=8    P=0.0000173
  z=9    P=0.0000046
  z=10   P=0.0000012

  q=0.3
  z=0    P=1.0000000
  z=5    P=0.1773523
  z=10   P=0.0416605
  z=15   P=0.0101008
  z=20   P=0.0024804
  z=25   P=0.0006132
  z=30   P=0.0001522
  z=35   P=0.0000379
  z=40   P=0.0000095
  z=45   P=0.0000024
  z=50   P=0.0000006
```

P’yi %0.1'den daha küçük alarak çözünce:

```
P < 0.001
  q=0.10   z=5
  q=0.15   z=8
  q=0.20   z=11
  q=0.25   z=15
  q=0.30   z=24
  q=0.35   z=41
  q=0.40   z=89
  q=0.45   z=340
```

12. Sonuç

Elektronik işlemler için güvene bağımlı olmayan bir sistem önerdik. Güçlü bir sahiplik kontrolü sağlayan, fakat çifte harcamayı önleyen bir yol olmadan eksik olan, dijital imzalardan oluşan paraların bilindik yapısı ile başladık. Bunu çözmek için, işlem dökümünü herkesin erişimine açık şekilde kaydetmek için, dürüst düğümler işlemci gücünün çoğunluğunu kontrol ettiğinde saldırganlar tarafından hızlıca hesaplanması mümkün olmayan, iş-kanıtını (proof-of-work) kullanan eşler arası bir ağ önerdik. Ağ dağıtık basitliği içinde sağlam bir yapıdadır. Düğümler hep birlikte çok az koordinasyon ile çalışırlar. Mesajlar, belirli bir yere bir yere yönlendirilmediğinden ve yalnızca en iyi çaba esasına göre dağıtılması gerektiğinden, düğümlerin kimliklerinin belirtilmesi gerekmez. Düğümler, kendileri ağda yokken iş-kanıtı zincirini, ne olduysa onun kanıtı olarak kabul ederek dilediklerinde ağdan ayrılabilir ve tekrar bağlanabilirler. Düğümler, blokları onayladıklarını, onları uzatmak için çalışarak; hatalı blokları reddettiklerini de bloklar üzerinde çalışmayı reddederek göstermek için işlemci güçleri ile oy kullanırlar. İhtiyaç duyulan tüm kurallar ve teşvikler bu uzlaşma mekanizması ile desteklenebilir.

Referanslar

[1] W. Dai, "b-money," http://www.weidai.com/bmoney.txt, 1998.

[2] H. Massias, X.S. Avila, and J.-J. Quisquater, "Design of a secure timestamping service with minimal

trust requirements," In 20th Symposium on Information Theory in the Benelux, May 1999.

[3] S. Haber, W.S. Stornetta, "How to time-stamp a digital document," In Journal of Cryptology, vol 3, no

2, pages 99-111, 1991.

[4] D. Bayer, S. Haber, W.S. Stornetta, "Improving the efficiency and reliability of digital time-stamping,"

In Sequences II: Methods in Communication, Security and Computer Science, pages 329-334, 1993.

[5] S. Haber, W.S. Stornetta, "Secure names for bit-strings," In Proceedings of the 4th ACM Conference

on Computer and Communications Security, pages 28-35, April 1997.

[6] A. Back, "Hashcash - a denial of service counter-measure,"

http://www.hashcash.org/papers/hashcash.pdf, 2002.

[7] R.C. Merkle, "Protocols for public key cryptosystems," In Proc. 1980 Symposium on Security and

Privacy, IEEE Computer Society, pages 122-133, April 1980.

[8] W. Feller, "An introduction to probability theory and its applications," 1957.