Bitcoin: Eşler Arası Elektronik Bir Ödeme Sistemi

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

Translated in Turkish from bitcoin.org/bitcoin.pdf by Efe Cini
(https://www.linkedin.com/in/efecini/)

Özet. Tamamen eşler arası çalışan bir elektronik ödeme sistemi, çevrimiçi ödemelerin bir finans kurumundan geçmeden, doğrudan bir taraftan diğerine gönderilmesine olanak sağlamaktadır. Dijital imzalar çözümün bir parçasıdır ancak çift harcamayı önlemek için yine de üçüncü bir tarafa güvenmek lazım ise, asıl fayda kaybolmuş demektir. Çift harcama problemine eşler arası ağ kullanarak bir çözüm önermekteyiz. Ağ, işlemleri süregelen özet tabanlı bir iş kanıtı zincirine ekleyerek zaman damgasıyla işaretlemekte ve iş kanıtı tekrarlanmadan değiştirilemeyen bir kayıt oluşturmaktadır. En uzun zincir, sadece şahit olunan olayların sırasının kanıtı olarak hizmet etmemekte, ayrıca kendisinin en büyük işlemci gücüne sahip havuzdan geldiğinin de kanıtı olmaktadır. İşlemci gücünün çoğunluğu, ağa saldırmak için işbirliği yapmayan düğümler tarafından kontrol edildiği sürece, en uzun zinciri oluşturacak ve saldırganları geride bırakacaktır. Ağın kendisi minimal bir yapıya gereksinim duymaktadır. Mesajlar elden gelenin en iyisi temeline göre yayınlanmakta ve düğümler istedikleri zaman ağdan ayrılıp, ağ dışında kaldıkları süre içerisinde olanların kanıtı olarak en uzun iş kanıtı zincirini kabul etmek şartıyla ağa tekrar katılabilmektedirler.

1. Giriş

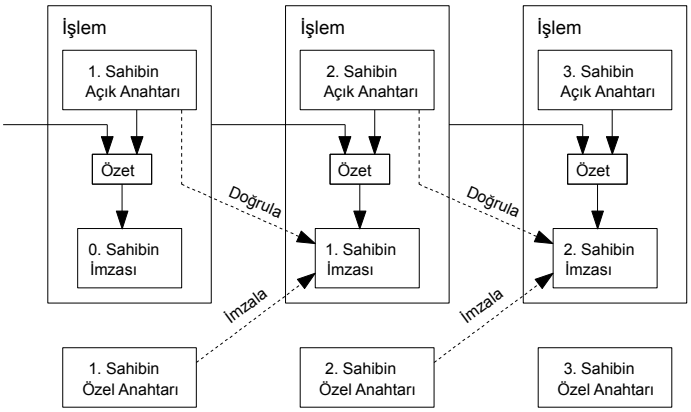
İnternet'te ticaret neredeyse tamamen elektronik ödemeleri işleyerek güvenilen bir üçüncü taraf olarak hizmet veren finansal kurumlara bağımlı hale gelmiştir. Sistem çoğu işlem için yeteri kadar iyi çalışıyor olsa da, güvene dayalı bir model olmanın doğasında var olan zaaflardan hala muzdariptir. Finansal kurumlar anlaşmazlık durumlarında arabuluculuktan kaçınamadıkları için geri döndürülemez işlemler aslında mümkün değildir. Arabuluculuk maliyeti işlem maliyetlerini yükseltmekte, mümkün olan minimum işlem büyüklüğünü sınırlamakta, sıradan küçük işlemleri engellemekte, dahası geri döndürülemez hizmetler için geri döndürülemez ödeme yapma imkanının olmaması masrafları arttırmaktadır. İşlemin geri döndürülme ihtimali ile birlikte güven ihtiyacı artmaktadır. Satıcılar normalde ihtiyaç duymadıkları ekstra bilgiyi müşteriden isterken dikkatli olmalıdır çünkü bu müşterileri rahatsız edecektir. Belli bir orandaki dolandırıcılık kaçınılmaz olarak kabul edilir. Bu maliyetler ve ödeme belirsizlikleri yüzyüze alışverişte fiziksel para birimi kullanımıyla giderilebilir ancak bir iletişim kanalı üzerinden, güvenilir bir taraf olmaksızın ödeme yapılabilen bir mekanizma bulunmamaktadır.

Asıl gereken, güven yerine kriptografik kanıta dayalı, iki istekli tarafın, üçüncü bir güvenilir tarafa gerek duymadan doğrudan birbiriyle işlem yapabilmelerini mümkün kılan bir elektronik ödeme sistemidir. Geri döndürülmesi hesaba dayalı olarak mümkün olmayan işlemler satıcıları dolandırıcılığa karşı koruyacak ve alıcıları koruyacak rutin emanetçi mekanizmalar kolayca uygulanabilecektir. Bu makalede, eşler arası dağıtık bir zaman damgası sunucusu ile işlemlerin kronolojik sırasının hesaba dayalı kanıtını oluşturarak, çift harcama problemine bir çözüm önermekteyiz. Sistem, dürüst düğümler işbirliği halindeki herhangi bir saldırgan düğüm

grubundan daha fazla işlemci gücünü toplu olarak kontrol ettiği sürece güvenlidir.

2. İşlemler

Elektronik parayı dijital bir imza zinciri olarak tanımlamaktayız. Her para sahibi, paranın bir önceki işlemiyle birlikte, parayı alacak kişinin açık anahtar özetini dijital olarak imzalayıp paranın sonuna ekleyerek, bir sonraki sahibe o parayı yollamaktadır. Ödemeyi alan kişi mülkiyet zincirini doğrulamak için imzaları doğrulayabilir.

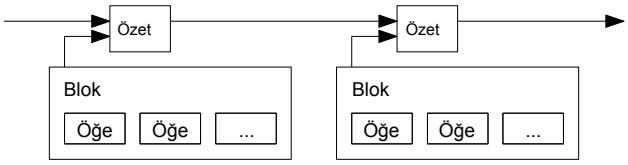


Elbette buradaki sorun, ödeme alan kişinin paranın önceki sahiplerinden birinin parayı iki kez harcamadığını doğrulayamamasıdır. Yaygın bir çözüm, her işlemde çift harcamayı denetleyen güvenilir, merkezi bir otorite veya darphane sunmaktır. Her işlemden sonra para, yeni para basımı için darphaneye iade edilmelidir ve sadece doğrudan darphanede basılmış paraların iki kez harcanmadığına güvenilmektedir. Bu çözüm ile ilgili problem, bütün işlemlerin bir bankaymış gibi darphaneyi işleten şirketten geçme zorunluluğu ve tüm para sisteminin kaderinin bu şirketin elinde olmasıdır.

Ödemeyi alan kişinin, paranın önceki sahiplerinin daha önce hiçbir işlemi imzalamadığını bildiği bir yol bulmamız gerekmektedir. Amaçlarımız doğrultusunda en önemli işlem, en eski işlemdir ve böylece sonraki çift harcama girişimleri bizi ilgilendirmemektedir. Bir işlemin gerçekleşmediğini onaylamanın tek yolu, tüm işlemlerden haberdar olmaktır. Darphaneye dayalı modelde, darphane tüm işlemlerden haberdardı ve hangisinin önce geldiğine o karar veriyordu. Güvenilir bir üçüncü taraf olmadan bunu başarabilmek için, işlemlerin alenen ilan edilmesi gerekmektedir [1] ve katılımcıların aldıkları siparişin tek bir geçmişi olduğu konusunda hemfikir olacakları bir sisteme ihtiyaç duyulmaktadır. Ödeme yapılan kişi her işlem sırasında, düğüm çoğunluğunun aldıkları ilk işlemin o işlem olduğu konusunda hemfikir olduğu bir kanıta ihtiyaç duymaktadır.

3. Zaman Damgası Sunucusu

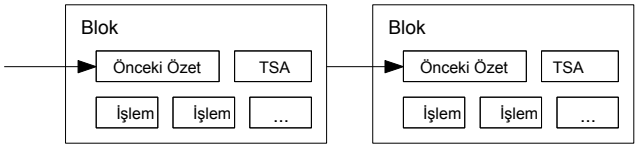
Önerdiğimiz çözüm bir zaman damgası sunucusuyla başlamaktadır. Bir zaman damgası sunucusu, damgalanmayı bekleyen öğelerden oluşan bloğun özetini alarak ve bu karışımı bir gazete veya Usenet gönderisi[2-5] gibi her tarafa yayımlayarak çalışmaktadır. Zaman damgası, özete girmek amacıyla verinin o anda var olması gerektiğinin kanıtıdır. Özetinde bir önceki zaman damgasını bulunduran da dahil olmak üzere her zaman damgası, eklenen her zaman damgası ile kendisinden öncekileri sağlamlaştırarak bir zincir oluşturmaktadır.



4. İş Kanıtı

Eşler arası temelde dağıtılmış bir zaman damgası sunucusunu uygulamak için, gazete veya Usenet yayınlarından ziyade, Adam Back'in geliştirdiği Hashcash [6] benzeri bir iş kanıtı sistemi kullanmamız gerekecektir. İş kanıtı, -örneğin SHA-256 ile- özetleme işlemi sonucunda çıkan özetin başında belirli sayıda 0 bit olan bir değerin taranmasını içermektedir. Yapılması gereken ortalama iş, bulunması gerekli 0 bitlerin sayısı ile üssel olarak orantılıdır ve tek bir özet hesabıyla doğrulanabilmektedir.

Zaman damgası ağımızdaki iş kanıtı modelini, blok özetindeki gerekli sıfır değerli bitleri veren bir değer bulunana kadar, bloğun tek seferlik anahtarını arttırarak uygulamaktayız. İşlemci gücü bir kez iş kanıtını yerine getirmek için harcandığında, blok aynı iş tekrarlanmadan değiştirilememektedir. Sonraki bloklar zincire eklendikçe, bloğu değiştirmek için gereken iş kendisinden sonra zincire eklenen bütün blokları değiştirmeyi de kapsayacaktır.



İş kanıtı aynı zamanda çoğunluğun karar alma sürecindeki temsilin belirlenme sorununu da çözmektedir. Eğer çoğunluk IP adresine dayalı olarak belirlenseydi, kendine çok sayıda IP adresi tahsis edebilen herkes bu çoğunluğu bozabilirdi. İş kanıtında esas, bir işlemcinin bir oyu temsil etmesidir. Çoğunluk kararı, kendisine yatırılan en büyük iş kanıtı çabasına sahip olan en uzun zincir tarafından temsil edilmektedir. İşlemci gücünün büyük bir kısmı dürüst düğümler tarafından kontrol edilirse, dürüst zincir en hızlı şekilde büyüyecek ve rakip zincirlerin önüne geçecektir. Bir saldırganın geçmişteki bir bloğu değiştirmesi için, bloğun ve ondan sonra gelen tüm blokların iş kanıtını yeniden yapması ve ardından dürüst düğümlerin yaptığı işi yakalayarak, onları geçmesi gerekmektedir. İleride, daha yavaş bir saldırganın ağı yakalama olasılığının bloklar eklendikçe üssel olarak azaldığını göstereceğiz.

Artan donanım hızını ve zamanla değişen düğüm çalıştırma ilgisine ayak uydurmak için iş kanıtı zorluğu, saatte ortalama blok sayısını hedefleyen bir hareketli ortalama ile belirlenir. Bloklar çok hızlı üretilirlerse, zorluk seviyesi artmaktadır.

5. Ağ

Ağı çalıştırmak için gerekli adımlar şöyledir:

1) Yeni işlemler tüm düğümlere yayınlanır.
2) Her düğüm yeni işlemleri bir blok içinde toplar.
3) Her düğüm kendi bloğu için zor bir iş kanıtı üzerinde çalışır.

4) Bir düğüm iş kanıtını bulduğunda, bu bloğu tüm düğümlere yayınlar.
5) Düğümler bloğu yalnızca içindeki tüm işlemler geçerliyse ve halihazırda harcanmamışsa kabul ederler.
6) Düğümler bloğu kabul ettiklerini, zincirdeki bir sonraki bloğu oluşturmak için çalışırken önceki özet olarak bu bloğun özetini kullanarak gösterirler.

Düğümler her zaman en uzun zincirin doğru zincir olduğunu düşünmekte ve onu uzatmak için çalışmaya devam edeceklerdir. İki düğüm, bir sonraki bloğun farklı versiyonlarını aynı anda yayınlıyorsa, bazı düğümler birini, bazı düğümler diğerini alabilir. Bu durumda ilk aldıkları üzerinde çalışırlar ancak diğer dalı da daha uzun olma ihtimaline karşılık saklarlar. Beraberlik bir sonraki iş kanıtı bulunduğunda ve bir dal daha uzun hale geldiğinde kırılacak, diğer dalda çalışmış olan düğümler uzun olana geçecektir.

Yeni işlem yayınlarının tüm düğümlere ulaşması gerekmez. Bu yayınlar birçok düğüme eriştiği sürece, çok geçmeden bir bloğa gireceklerdir. Blok yayınları aynı zamanda bırakılmış mesajlara karşı da dayanıklıdır. Eğer bir düğüm bir bloğu almazsa, bir sonraki bloğu teslim aldığında eksik bloğu fark edecek ve bu bloğu talep edecektir.

6. Teşvik

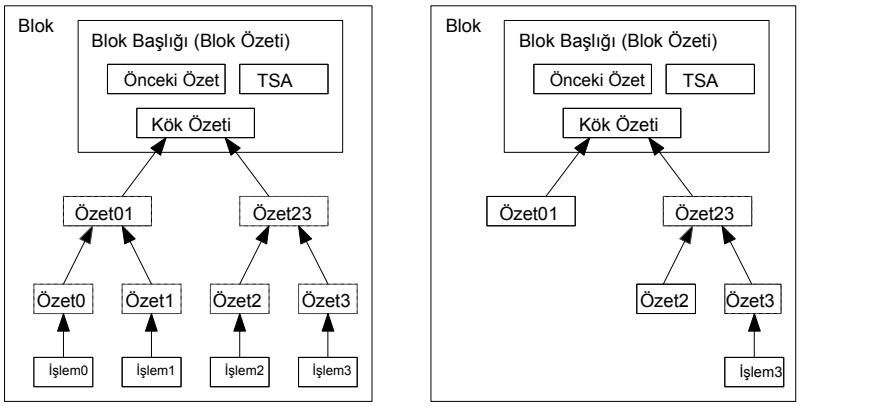
Kural gereği bloktaki ilk işlem, bloğun yaratıcısına ait olan paranın yaratıldığı özel bir işlemdir. Bu, düğümlerin ağı desteklemesini teşvik etmekte ve paranın ilk kez dolaşıma girmesini sağlamaktadır. Zira, para basacak başka bir merkezi otorite bulunmamaktadır. Sabit miktardaki yeni basılan paranın düzenli olarak dolaşıma girmesi, altın madencilerinin kaynaklarını harcayarak dolaşıma altın sürmesine benzer. Bizim durumumuzda bu, harcanan işlemci zamanı ve elektriktir.

Teşvik, işlem ücretleri ile de finanse edilebilir. Bir işlemin çıktı değeri girdisinden düşük ise, aradaki fark işlemi içeren bloğun teşviğine eklenen işlem ücretidir. Önceden belirlenmiş miktardaki para dolaşıma girdiğinde, teşvik tamamiyle işlem ücretlerine dönüşebilir ve tamamen enflasyondan arınmış olabilir.

Teşvik, düğümlerin dürüst kalmasının özendirilmesine yardımcı olabilir. Eğer açgözlü bir saldırgan dürüst düğümlerin tümünden daha fazla işlemci gücünü toplayabilirse, bunu ödemelerini geri alarak insanları dolandırmak için mi, yoksa yeni para üretmek için mi kullanacağına karar vermesi gerekecektir. Saldırgan oyunu kuralına göre oynamayı daha kârlı bulmalıdır. Böylece sistemi ve kendi servetinin geçerliliğini baltalamaktansa, diğer herkesin toplamından daha fazla yeni para elde edecektir.

7. Disk Alanı Geri Kazanımı

Bir paranın kullanıldığı son işlem yeterince blok altında kaldıktan sonra, o paranın önceki işlemleri disk alanından tasarruf etmek için atılabilir. Blok özetini bozmadan bunu yapabilmek için, işlemler bir Merkle ağacında[7][2][5] sadece kök dahil olacak şekilde özetlenmiştir. Eski bloklar daha sonra ağacın dalları budanarak sıkıştırılabilir. İç özetlerin depolanmasına gerek yoktur.



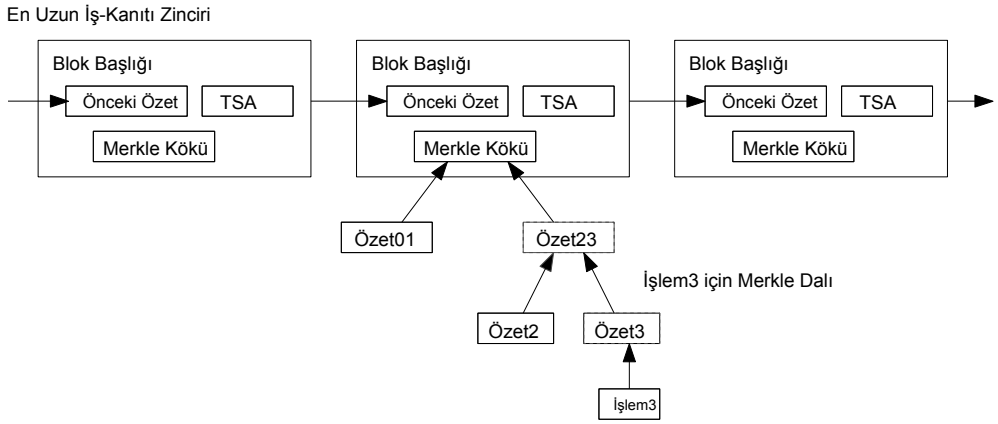
Merkle Ağacında Özetlenmiş İşlemler

0-2 arasındaki işlemler bloktan budandıktan sonra

Bir blok başlığı işlem olmadan yaklaşık 80 bayttır. 10 dakikada bir blok üretildiğini varsayarsak, 80 bayt * 6 * 24 * 365 = yılda 4.2MB yapar. 2008 itibariyle tipik bilgisayar sistemlerinin 2GB RAM ile satıldığını ve yıllık 1.2GB büyümeyi öngören Moore yasasını göz önünde bulundurursak, blok başlıkları bellekte tutulsa bile depolama bir sorun teşkil etmeyecektir.

8. Basitleştirilmiş Ödeme Doğrulaması

Ödemeleri tam bir ağ düğümü çalıştırmadan doğrulamak mümkündür. Bunun için bir kullanıcının yalnızca en uzun iş kanıtı zincirindeki blok başlıklarının bir kopyasını saklaması ve işlemi, içinde zaman damgalandığı bloğa bağlayan Merkle dalını bulması gerekmektedir. Bu kopyayı en uzun zincirin kendisinde olduğuna ikna olana dek ağ düğümlerine sorgu yaparak elde edebilir. İşlemi kendisi kontrol edemez ancak zincirdeki bir yere bağlayarak bir ağ düğümünün bu işlemi kabul ettiğini görebilir ve kendisinden sonraki bloklar, ağın işlemi kabul ettiğini onaylayabilir.

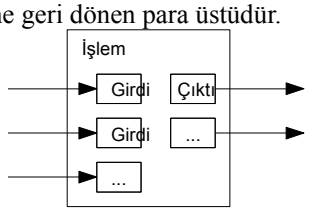


Bu sebeple, doğrulama işlemi dürüst düğümler ağı kontrol ettiği sürece güvenilirken, ağ bir saldırgan tarafından güçlendirilmişse savunmasız hale gelir. Ağ düğümleri, işlemleri kendileri doğrulayabilse de, saldırgan uydurma işlemler ile ağı güçlendirmeye devam edebildiği sürece basitleştirilmiş yöntem kandırılabilir. Bundan korunmak için bir strateji, geçersiz bir blok fark ettiğinde ağ düğümlerinden gelen uyarıları alarak, tutarsızlığı onaylamak için kullanıcının yazılımını tüm bloğu ve uyarı alan işlemleri indirmeye yönlendirmektir. Sık ödeme alan

işletmeler muhtemelen daha bağımsız güvenlik ve daha hızlı doğrulama için yine de kendi düğümlerini çalıştırmak isteyeceklerdir.

9. Değer Birleşimi ve Ayrımı

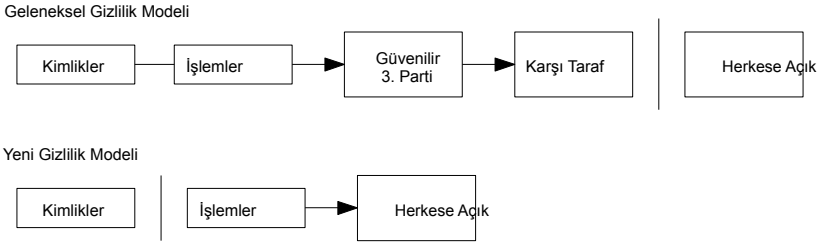
Paraları tek tek idare etmek mümkün olsa da, bir transferdeki her kuruş için ayrı birer işlem yapmak kullanışsız olmaktadır. Değerin bölünmesine ve birleştirilmesine olanak sağlamak için, işlemler birden çok girdi ve çıktı içerir. Normalde, ya önceki büyük işlemden gelen tek bir girdi ya da daha küçük miktarları birleştiren birden fazla girdi ve en fazla iki çıktı olacaktır: biri ödeme tutarı, diğeri -eğer varsa- gönderene geri dönen para üstüdür.



Unutulmamalıdır ki yayılma, yani bir işlemin bir kaç işleme bağlı olması ve onların da başka işlemlere bağlı olma durumu bir sorun yaratmaz. Bir işlemin geçmişinin tek başına kopyasını çıkarma hiçbir zaman gerekmemektedir.

10. Gizlilik

Geleneksel bankacılık modeli bilgiye erişimi ilgili taraflar ve güvenilir bir üçüncü tarafla sınırlayarak, belli bir seviyeye kadar gizliliği sağlamaktadır. Tüm işlemleri herkese açık bir şekilde duyurma zorunluluğu bu yöntemi engeller ancak gizlilik, yine de bilgi akışını başka bir yerde kırarak, açık anahtarları anonim tutarak sürdürülebilir. Herkes birisinin başka birine bir miktar gönderdiğini görebilir ancak işlemin kimle bağlantılı olduğu bilgisine ulaşamaz. Bu borsaların yayınladığı bilgi seviyesine benzer; bireysel işlemlerin zamanı ve büyüklüğü herkese açıkken, tarafların kimlik bilgileri gizlenmektedir.



Ek bir güvenlik duvarı olarak, işlemlerin ortak bir sahiple ilişkilendirilmesini önlemek için her işlemde yeni bir anahtar çifti kullanılmalıdır. Girdilerin aynı kişiye ait olması gereken çoklu girdili işlemlerde, bazı ilişkilendirmeler kaçınılmazdır. Buradaki risk, anahtar sahibi ortaya çıktığında, ilişkilendirme yapılarak başka işlemlerin aynı sahibe ait olduğunun ortaya çıkabilmesidir.

11. Hesaplamalar

Dürüst zincirden daha hızlı bir şekilde alternatif bir zincir oluşturmaya çalışan bir saldırganın senaryosunu ele almaktayız. Bu başarılsa bile, hiç yoktan değer yaratılan ya da saldırgana hiç ait olmamış paranın alınabileceği şekilde, sistemi keyfi değişikliklere açık bir hale getirmez. Düğümler geçersiz bir işlemi ödeme olarak kabul etmeyecek ve dürüst düğümler asla bu işlemleri içeren bir bloğu kabul etmeyecektir. Bir saldırgan yalnızca yakın zamanda harcadığı parayı geri almak için kendi işlemlerinden birini değiştirmeye çalışabilir.

Dürüst zincir ile bir saldırgan zinciri arasındaki yarış Binom Rastgele Yürüyüşü olarak nitelendirilebilir. Başarılı olma durumu dürüst zincirin bir blok uzatılarak öncülüğünü 1 arttırması, başarısızlık durumu ise saldırganın zincirinin bir blok uzatılarak aralığın 1 azalmasıdır.

Bir saldırganın belirli bir açığı yakalama olasılığı Kumarbazın İflası problemine benzemektedir. Sınırsız krediye sahip bir kumarbazın oyuna borçla başladığını ve başabaş noktasına gelebilmek için sonsuz sayıda deneme oyunu oynadığını varsayın. Başabaş noktasına ulaşabilme olasılığını veya saldırganın dürüst zinciri yakalayabilme olasılığını şu şekilde hesaplayabiliriz [8]:

p = dürüst bir düğümün sonraki bloğu bulma olasılığı
q = saldırganın bir sonraki bloğu bulma olasılığı
q_z = saldırganın z blok geriden gelip yakalama ihtimali

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

P > q varsayımımıza göre, saldırganın yakalaması gereken blok sayısı arttıkça, olasılık katlanarak düşmektedir. Bütün olanaksızlıklarla beraber, saldırgan başlarda eğer bir dizi şanslı hamle yapmazsa, kazanma ihtimali zincirin gerisinde kaldıkça hızla azalmaktadır.

Şimdi yeni bir işlemdeki alıcının, gönderenin işlemi değiştiremeyeceğinden yeterince emin olmadan önce ne kadar beklemesi gerektiğini ele almaktayız. Gönderenin, alıcıyı belli bir süre için kendisine ödeme yaptığına inandırmak isteyen, belli bir zaman geçtikten sonra ise ödemeyi kendisine geri döndürmek isteyen bir saldırgan olduğunu varsaymaktayız. Bu olduğunda alıcı uyarılacaktır ancak gönderen bunun için artık çok geç olacağını ümit etmektedir.

Alıcı yeni bir anahtar çifti oluşturur ve imzalamadan kısa bir süre önce gönderene açık anahtarı verir. Bu, gönderenin önceden sürekli olarak üstünde çalışarak, bloklardan oluşan bir zinciri hazırlamasını ve bu zincirin öne geçtiği şanslı bir anda işlemi çalıştırmasını önlemektedir. İşlem gönderildikten sonra dürüst olmayan gönderici, işleminin alternatif bir versiyonunu içeren paralel bir zincir üzerinde gizlice çalışmaya başlar.

Alıcı, işlem bir bloğa eklenene ve bunun arkasına z kadar blok bağlanana kadar bekler. Saldırganın ne kadar ilerleme kaydettiğini tam olarak bilmez ancak dürüst blokların blok başına beklenen ortalama bir süre aldığını varsayarsak, saldırganın yaptığı potansiyel ilerleme beklenen değerli bir Poisson dağılımı olacaktır:

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

Saldırganın herşeye rağmen yetişebilme ihtimalini elde etmek için, o noktadan zinciri yakalayabilme olasılığı ile elde edebileceği her ilerleme için Poisson yoğunluğunu çarparız:

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

Sonsuz dağılım kuyruk toplamını almaktan kaçınmak için yeniden düzenliyoruz...

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

C koduna dönüştürüyoruz...

```
#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 değerler vererek, olasılığın z ile katlanarak azaldığını 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 düşük değerler için çözüyoruz...

```
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

Güvene dayalı olmayan elektronik işlemler için bir sistem önermiş bulunmaktayız. Dijital imzalardan oluşan alışılmış bir para sistemi ile başladık. Bu sistem mülkiyet üzerinde güçlü bir kontrol sağlasa da, çift harcamayı engellemenin bir yolunu bulmadıkça eksik kalmaktadır. Bunu çözmek için, işlem geçmişini açık bir şekilde kaydetmek için iş kanıtı kullanan, dürüst düğümler işlemci gücünün çoğunluğunu kontrol ettiği takdirde bir saldırganın değiştirmesinin hesaplanabilir olarak mantıksız olduğu, eşler arası bir ağ sunduk. Bu ağın sağlamlığı yapılandırılmamış basitliğinden kaynaklanmaktadır. Düğümler çok az koordinasyon ile aynı anda çalışmaktadır. Mesajlar belirli bir yere yönlendirilmediği ve yalnızca elden gelenin en iyisi temelinde iletilmesi gerektiği için, düğümlerin tanımlanmasına gerek kalmamaktadır. Düğümler ağdan ayrıldıkları süreçte olanların kanıtı olarak iş kanıtı zincirini kabul etme şartıyla istedikleri zaman ağdan ayrılabilir ve yeniden ağa katılabilirler. İşlemci güçleriyle oy kullanırlar. Geçerli blokları kabul ettiklerini, onları yaymaya çalışarak ve geçersiz blokları reddettiklerini ise onların üzerinde çalışmayı geri çevirerek gösterirler. Gerekli tüm kural ve teşvikler bu fikir birliği mekanizmasıyla uygulanabilmektedir.

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.