Bitcoin: Eşler Arası Elektronik Nakit Sistemi

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

Öz. Tamamen eşler arası bir elektronik nakit sürümü, çevrimiçi
doğrudan bir taraftan diğerine gönderilecek ödemeler
finansal kurum. Dijital imzalar çözümün bir parçasını sağlar, ancak ana
Çift harcamayı önlemek için hala güvenilir bir üçüncü taraf gerekiyorsa faydalar kaybolur.
Eşler arası bir ağ kullanarak çift harcama sorununa bir çözüm öneriyoruz.
Ağ, işlemleri devam eden bir zincir haline getirerek işlemlerin zaman damgasını
karma tabanlı çalışma kanıtı, yeniden yapmadan değiştirilemeyen bir kayıt oluşturur
işin kanıtı. En uzun zincir, yalnızca dizinin kanıtı olarak hizmet etmez.
Olaylara tanık oldu, ancak bunun en büyük CPU gücü havuzundan geldiğinin kanıtı. Gibi
CPU gücünün çoğu, işbirliği yapmayan düğümler tarafından kontrol edildiği sürece
ağa saldırırsa, en uzun zincire ve saldırganların ötesine geçerler. The
ağın kendisi minimum yapı gerektirir. Mesajlar en iyi çabayla yayınlanır
temel ve düğümler istediği zaman ağdan ayrılabilir ve en uzun süreyi kabul ederek ağa yeniden katılabilir.
onlar yokken olanların kanıtı olarak iş kanıtı zinciri.

1. Giriş

İnternet üzerinden ticaret, neredeyse yalnızca, şu şekilde hizmet veren finansal kurumlara dayanmaktadır:
elektronik ödemeleri işlemek için güvenilir üçüncü şahıslar. Sistem aşağıdakiler için yeterince iyi çalışsa da
çoğu işlemde, güvene dayalı modelin içsel zayıflıklarından hala muzdariptir.
Tamamen geri döndürülemez işlemler gerçekten mümkün değildir, çünkü finansal kuruluşlar bunu yapamaz.
anlaşmazlıklarda arabuluculuk yapmaktan kaçının. Arabuluculuğun maliyeti, işlem maliyetlerini artırarak,
minimum pratik işlem boyutu ve küçük geçici işlemler olasılığını ortadan kaldırma,
ve olmayanlar için geri döndürülemez ödemeler yapma kabiliyetinin kaybının daha geniş bir maliyeti vardır.
tersine çevrilebilir hizmetler. Tersine dönme olasılığı ile güven ihtiyacı yayılır. Tüccarlar,
Müşterilerine karşı dikkatli olun, aksi takdirde ihtiyaç duyacaklarından daha fazla bilgi için onları zorlayın.
Dolandırıcılığın belirli bir yüzdesi kaçınılmaz olarak kabul edilmektedir. Bu maliyetler ve ödeme belirsizlikleri
fiziksel para kullanarak şahsen önlenebilir, ancak ödeme yapmak için bir mekanizma yoktur
güvenilir bir tarafı olmayan bir iletişim kanalı üzerinden.
İhtiyaç duyulan güven yerine kriptografik kanıta dayalı elektronik ödeme sistemidir,
herhangi iki istekli tarafın güvenilir bir kuruluşa ihtiyaç duymadan birbirleriyle doğrudan işlem yapmasına izin vermek
üçüncü şahıs. Tersine çevirmek hesaplama açısından pratik olmayan işlemler satıcıları koruyacaktır
dolandırıcılıktan ve alıcıları korumak için rutin emanet mekanizmaları kolayca uygulanabilir. İçinde
bu makalede, eşler arası dağıtılmış bir kullanarak çift harcama sorununa bir çözüm öneriyoruz.
işlemlerin kronolojik sırasının hesaplamalı kanıtını oluşturmak için zaman damgası sunucusu. The
dürüst düğümler toplu olarak herhangi bir işlemciden daha fazla CPU gücünü kontrol ettiği sürece sistem güvenlidir.
işbirliği yapan saldırgan düğümleri grubu.

---

2. İşlemler

Elektronik parayı dijital imzalar zinciri olarak tanımlıyoruz. Her sahip, madeni parayı
sonraki işlemin bir karma değerini ve sonraki sahibin genel anahtarını dijital olarak imzalayarak
ve bunları madalyonun sonuna eklemek. Alacaklı, zincirini doğrulamak için imzaları doğrulayabilir.
mülkiyet.
Elbette sorun, alacaklının sahiplerden birinin iki kez harcama yapmadığını doğrulayamamasıdır.
madeni para. Yaygın bir çözüm, her şeyi kontrol eden güvenilir bir merkezi otorite veya darphane
sunmaktır.
çift harcama için işlem. Her işlemden sonra, bozuk paranın darphaneye iade edilmesi gerekir.
yeni bir madeni para çıkarır ve yalnızca doğrudan darphaneden çıkarılan madeni paraların iki kez
harcanmamasına güvenilir.
Bu çözümle ilgili sorun, tüm para sisteminin kaderinin,
Tıpkı bir banka gibi, her işlemin gerçekleştirilmesi gereken darphane şirketi.
Alacaklının, önceki sahiplerin daha önce imzalamadığını bilmesi için bir yola ihtiyacımız var.
işlemler. Amaçlarımız açısından, en erken işlem önemli olandır, bu yüzden umursamıyoruz
sonraki iki kez harcama girişimleri hakkında. Bir işlemin olmadığını doğrulamanın tek yolu,
tüm işlemlerden haberdar olun. Darphane tabanlı modelde, darphane tüm işlemlerin farkındaydı ve
hangisinin önce geldiğine karar verdi. Bunu güvenilir bir taraf olmadan gerçekleştirmek için işlemler,
kamuya açıklandı [1] ve katılımcıların tek bir tarihçe üzerinde anlaşmaları için bir sisteme ihtiyacımız
var.
alındıkları sipariş. Alacaklının, her işlem sırasında
düğümlerin çoğu bunun ilk alındığı konusunda hemfikirdi.

3. Zaman Damgası Sunucusu

Önerdiğimiz çözüm, bir zaman damgası sunucusuyla başlar. Bir zaman damgası sunucusu, bir
zaman damgası alınacak ve hash'i geniş çapta yayınlayacak bir öğelerin karması, örneğin bir
gazete veya Usenet gönderisi [2-5]. Zaman damgası, verilerin şu anda mevcut olması gerektiğini
kanıtlıyor
Zaman, belli ki, hash'e girmek için. Her zaman damgası şuradaki önceki zaman damgasını içerir:
hash'i bir zincir oluşturur ve her ek zaman damgası kendisinden öncekileri güçlendirir.
2
Blok
Öğe
Öğe
...
Hash
Blok
Öğe
Öğe
...
Hash
İşlem
Sahip 1'ler
Genel anahtar
Sahip 0'lar
SiTnature
Hash
İşlem
Sahip 2'ler
Genel anahtar
Sahip 1'ler
SiTnature
Hash
Doğrulayın
İşlem
Sahip 3
Genel anahtar

Sahip 2'ler
SiTnature
Hash
Doğrulayın
Sahip 2'ler
Özel anahtar
Sahip 1'ler
Özel anahtar
SiTn
SiTn
Sahip 3
Özel anahtar

---

4. Çalışma Kanıtı

Eşler arası temelde dağıtılmış bir zaman damgası sunucusu uygulamak için, bir kanıt kullanmamız gerekecek.
gazete veya Usenet gönderileri yerine Adam Back'in Hashcash [6] 'sına benzer bir çalışma sistemi.
İşin kanıtı, SHA-256 gibi hashing uygulandığında,
karma bir dizi sıfır bit ile başlar. Gerekli ortalama iş, sayı olarak üsteldir
sıfır bit gereklidir ve tek bir hash çalıştırılarak doğrulanabilir.
Zaman damgası ağımız için, işin kanıtını,
bloğun hash'ına gerekli sıfır biti veren bir değer bulunana kadar blok. Bir kez CPU
iş kanıtı tatmin etmek için çaba sarf edildi, blok değiştirilemez
işi yeniden yapmadan. Daha sonraki bloklar zincirlendiğinden, bloğu değiştirme işi
ondan sonraki tüm blokların yeniden yapılmasını içerir.
İş kanıtı aynı zamanda çoğunluk kararında temsilin belirlenmesi sorununu da çözer.
yapımı. Çoğunluk bir IP adresi-bir oyuna dayanırsa, herkes tarafından tersine çevrilebilir.
birçok IP tahsis edebilir. İşin kanıtı esasen bir CPU-bir oydur. Çoğunluk
karar, yatırılan en büyük çalışma kanıtı çabasına sahip en uzun zincir tarafından temsil edilir
içinde. CPU gücünün çoğunluğu dürüst düğümler tarafından kontrol ediliyorsa, dürüst zincir
en hızlı ve rakip zincirleri geride bırakın. Geçmiş bir bloğu değiştirmek için bir saldırganın
bloğun ve ondan sonraki tüm blokların çalışma kanıtını yeniden yapın ve ardından
dürüst düğümlerin çalışması. Daha sonra daha yavaş bir saldırganın yakalama olasılığını göstereceğiz.
sonraki bloklar eklendikçe üssel olarak azalır.
Artan donanım hızını ve zaman içinde çalışan düğümlere olan ilgiyi telafi etmek için,
iş kanıtı zorluğu, ortalama bir sayı hedefleyen hareketli bir ortalama ile belirlenir.
saatte blok. Çok hızlı üretilirlerse zorluk artar.

5. Ağ

Ağı çalıştırma adımları aşağıdaki gibidir:
1) Yeni işlemler tüm düğümlere yayınlanır.
2) Her düğüm yeni işlemleri bir blok halinde toplar.
3) Her düğüm, bloğu için zor bir çalışma kanıtı bulmaya çalışır.
4) Bir düğüm bir çalışma kanıtı bulduğunda, bloğu tüm düğümlere yayınlar.
5) Düğümler, bloğu yalnızca içindeki tüm işlemler geçerliyse ve henüz harcanmamışsa kabul eder.
6) Düğümler, bloğu kabul ettiklerini, bir sonraki bloğu oluşturmak için çalışarak ifade eder.
zincir, kabul edilen bloğun karmasını önceki karma olarak kullanarak.
Düğümler her zaman en uzun zincirin doğru zincir olduğunu düşünür ve üzerinde çalışmaya devam eder.
genişletmek. İki düğüm bir sonraki bloğun farklı sürümlerini aynı anda yayınlarsa, bazıları
düğümler ilk olarak birini veya diğerini alabilir. Bu durumda, aldıkları ilk ürün üzerinde çalışırlar.
ancak diğer şubeyi daha uzun olması durumunda saklayın. Beraberlik bir sonraki kanıt olduğunda bozulur.
of-work bulunur ve bir dal uzar; diğerinde çalışan düğümler
şube daha sonra uzun olana geçecektir.

3
Blok
Önceki Karma
Nonce
Tx
Tx
...
Blok
Önceki Karma
Nonce
Tx
Tx
...

Yeni işlem yayınlarının tüm düğümlere ulaşması gerekmez. Ulaştıkları sürece
birçok düğüm, çok geçmeden bir bloğa girecekler. Blok yayınlar da düşmeye toleranslıdır
mesajlar. Bir düğüm bir blok almazsa, bir sonraki bloğu aldığında bunu talep edecek ve
birini kaçırdığının farkına varır.

6. Teşvik

Geleneksel olarak, bir bloktaki ilk işlem, sahip olunan yeni bir madeni parayı başlatan özel bir
işlemdir.
bloğun yaratıcısı tarafından. Bu, düğümlerin ağı desteklemesi için bir teşvik ekler ve
madeni paraları ilk başta dolaşıma sokmanın bir yolu, çünkü bunları basacak merkezi bir otorite yok.
Sabit miktarda yeni madeni paranın sabit bir şekilde eklenmesi, harcayan altın madencilerine benzer.
dolaşıma altın eklemek için kaynaklar. Bizim durumumuzda, harcanan CPU zamanı ve elektriktir.
Teşvik, işlem ücretleriyle de finanse edilebilir. Bir işlemin çıktı değeri
Girdi değerinden daha düşük olan fark, teşvik değerine eklenen bir işlem ücretidir.
işlemi içeren blok. Önceden belirlenmiş sayıda jeton girildiğinde
dolaşım, teşvik tamamen işlem ücretlerine geçebilir ve tamamen enflasyon olabilir
Bedava.
Teşvik, düğümleri dürüst kalmaya teşvik etmeye yardımcı olabilir. Açgözlü bir saldırgan,
tüm dürüst düğümlerden daha fazla CPU gücü toplarsa, kullanmak arasında seçim yapmak zorunda
kalırdı.
ödemelerini geri çalarak veya yeni madeni paralar yaratmak için kullanarak insanları
dolandırmak. Yapmalı
kurallara göre oynamayı daha karlı buluyor, bu tür kurallar onu daha fazla yeni jetonla destekleyen
kurallar
sistemin ve kendi servetinin geçerliliğinin altını oymaktan başka herkes birleşti.

7. Disk Alanını Geri Kazanmak

Bir madeni paradaki en son işlem yeterli bloğun altına gömüldüğünde,
disk alanından tasarruf etmek için atılabilir. Bunu bloğun hash değerini bozmadan kolaylaştırmak için,
işlemler bir Merkle Ağacında [7] [2] [5] karma haline getirilir ve yalnızca bloğun karmasına eklenen
kök vardır.
Daha sonra eski bloklar, ağacın dalları kesilerek sıkıştırılabilir. İç karmalar işe yarar
saklanmasına gerek yoktur.
İşlem içermeyen bir blok başlığı yaklaşık 80 bayt olacaktır. Blokların olduğunu varsayarsak
her 10 dakikada bir oluşturulur, 80 bayt * 6 * 24 * 365 = yılda 4,2 MB. Bilgisayar sistemleri ile
tipik olarak 2008 itibariyle 2GB RAM ile satış ve şu anki büyümeyi öngören Moore Yasası
Yılda 1,2 GB, blok başlıklarının saklanması gerekse bile depolama bir sorun olmamalıdır
hafıza.
4
Blok
Blok
Üstbilgiyi Engelle (Karma Blok)

Önceki Karma
Nonce
Hash01
Hash0
Hash1
Hash2
Hash3
Hash23
Kök Hash
Hash01
Hash2
Tx3
Hash23
Üstbilgiyi Engelle (Karma Blok)
Kök Hash
Bir Merkle Ağacında Karıştırılan İşlemler
Bloktan PruninT Tx0-2'den sonra
Önceki Karma
Nonce
Hash3
Tx0
Tx1
Tx2
Tx3

5.Sayfa

8. Basitleştirilmiş Ödeme Doğrulaması

Tam bir ağ düğümü çalıştırmadan ödemeleri doğrulamak mümkündür. Bir kullanıcının yalnızca tutması gerekir
en uzun çalışma kanıtı zincirinin blok başlıklarının bir kopyası, sorgulayarak elde edebilir
en uzun zincire sahip olduğuna ikna olana ve Merkle şubesini elde edene kadar ağ düğümleri
işlemi, zaman damgası bulunan bloğa bağlamak. İşlemi için kontrol edemez
kendisi, ancak onu zincirdeki bir yere bağlayarak, bir ağ düğümünün bunu kabul ettiğini görebilir,
ve ağın bunu kabul ettiğini doğruladıktan sonra eklenen bloklar.

Bu nedenle, dürüst düğümler ağı kontrol ettiği sürece doğrulama güvenilirdir, ancak daha fazlasıdır.
ağ bir saldırgan tarafından etkisiz hale getirilirse savunmasız kalır. Ağ düğümleri doğrulayabilirken
basitleştirilmiş yöntem, bir saldırganın uydurması tarafından kandırılabilir.
saldırgan ağa üstün gelmeye devam ettiği sürece işlemler. Bir strateji
buna karşı koruma, geçersiz bir
engelleme, kullanıcının yazılımından tüm bloğu indirmesini ve uyarı verilen işlemleri
tutarsızlığı onaylayın. Sık ödeme alan işletmeler büyük olasılıkla yine de
daha bağımsız güvenlik ve daha hızlı doğrulama için kendi düğümlerini çalıştırır.

9. Değeri Birleştirmek ve Bölmek

Madeni paraları ayrı ayrı ele almak mümkün olsa da,
bir transferdeki her kuruş için ayrı işlem. Değerin bölünmesine ve birleştirilmesine izin vermek için,
işlemler birden çok girdi ve çıktı içerir. Normalde tek bir giriş olacaktır
daha büyük bir önceki işlemden veya daha küçük miktarları birleştiren birden çok girdiden ve en fazla iki
çıktılar: biri ödeme için, diğeri ise değişikliği gönderene iade ediyor.

Bir işlemin birkaç işleme bağlı olduğu yayılmanın ve
işlemler çok daha fazlasına bağlıdır, burada sorun değil. Asla bir
bir işlem geçmişinin tam bağımsız kopyası.

5
İşlem
İçinde

...
İçinde
Dışarı
...
Hash01
Hash2
Hash3
Hash23
Üstbilgiyi Engelle
Merkle Kökü
Önceki Karma
Nonce
Üstbilgiyi Engelle
Merkle Kökü
Önceki Karma
Nonce
Üstbilgiyi Engelle
Merkle Kökü
Önceki Karma
Nonce
Tx3 için Merkle Şubesi
LonTest İş Kanıtı Zinciri
Tx3

Sayfa 6

10. Gizlilik

Geleneksel bankacılık modeli, bilgiye erişimi, bilgi erişimiyle sınırlandırarak bir mahremiyet düzeyine ulaşır.
ilgili taraflar ve güvenilir üçüncü taraf. Tüm işlemlerin kamuya duyurulması zorunluluğu
bu yöntemi engeller, ancak mahremiyet yine de bilgi akışını bozarak korunabilir.
başka bir yer: açık anahtarları anonim tutarak. Halk, birinin gönderdiğini görebilir
başka birine bir miktar, ancak işlemi kimseyle ilişkilendiren bilgi olmadan. Bu
borsalar tarafından yayınlanan bilgi düzeyine benzer şekilde, burada zaman ve boyut
bireysel esnaf, "kaset" halka açıklanır, ancak tarafların kim olduğu söylenmez.
Ek bir güvenlik duvarı olarak, bunları korumak için her işlem için yeni bir anahtar çifti kullanılmalıdır.
ortak bir sahibe bağlanmaktan. Çoklu giriş ile bazı bağlantılar hala kaçınılmazdır
girdilerinin aynı mal sahibine ait olduğunu ortaya çıkaran işlemler. Risk
bir anahtarın sahibinin ortaya çıkması durumunda bağlantı oluşturmanın,
aynı sahip.

11. Hesaplamalar

Dürüst olandan daha hızlı bir alternatif zincir oluşturmaya çalışan bir saldırganın senaryosunu düşünüyoruz.
Zincir. Bu başarılsa bile sistemi keyfi değişikliklere açık bırakmaz,
hiçbir zaman saldırgana ait olmayan parayı almak veya havadan değer yaratmak olarak. Düğümler
Geçersiz bir işlemi ödeme olarak kabul etmeyecek ve dürüst düğümler asla bir engellemeyi kabul etmeyecektir
onları içeren. Bir saldırgan, geri almak için yalnızca kendi işlemlerinden birini değiştirmeyi deneyebilir.
son zamanlarda harcadığı para.
Dürüst zincir ile bir saldırgan zinciri arasındaki yarış, Binom olarak tanımlanabilir
Rastgele yürüyüş. Başarı olayı, dürüst zincirin bir blok genişletilerek,
+1 ile başlar ve başarısızlık olayı, saldırganın zincirinin bir blok genişlemesidir.
-1 ile aralık.
Bir saldırganın belirli bir açığı yakalama olasılığı, bir Kumarbazınkine benzer
Yıkım sorunu. Sınırsız krediye sahip bir kumarbazın açıkla başladığını ve potansiyel olarak

başabaş noktasına ulaşmaya çalışmak için sonsuz sayıda deneme. Onun şimdiye kadarki olasılığını hesaplayabiliriz
başabaş noktasına ulaşırsa veya bir saldırganın dürüst zinciri yakaladığına, aşağıdaki gibi [8]:
p = dürüst bir düğümün bir sonraki bloğu bulma olasılığı
q = saldırganın bir sonraki bloğu bulma olasılığı
$q\ z$ = saldırganın arkadaki z bloktan yakalama olasılığı
$q\ z$ = { 1
eğer $p \leq q$
□ q / p □ z
eğer p □ q }
6
Kimlikler
İşlemler
Güvenilir
Üçüncü şahıs
Karşı taraf
halka açık
Kimlikler
İşlemler
halka açık
Yeni Gizlilik Modeli
Geleneksel Gizlilik Modeli

---

7. Sayfa

$P > q$ varsayımımıza göre , olasılık, blok sayısı arttıkça üssel olarak düşer.
saldırgan artışları yakalamak zorundadır. Onun aleyhine olan ihtimallerle, eğer şanslı değilse erkenden ileri atılırsa, daha da geride kaldıkça şansı ortadan kaybolur.
Şimdi, yeni bir işlemin alıcısının gerçekleşmeden önce ne kadar beklemesi gerektiğini düşünüyoruz. gönderenin işlemi değiştiremeyeceğinden yeterince emin. Gönderenin bir saldırgan olduğunu varsayıyoruz
alıcıyı bir süreliğine ödediğine inandırmak isteyen, sonra geri ödeme yapmak için
Bir süre sonra kendisi geçti. Bu olduğunda alıcı uyarılacaktır, ancak
gönderen çok geç olacağını umuyor.
Alıcı, yeni bir anahtar çifti oluşturur ve genel anahtarı gönderene kısa bir süre önce verir.
imzalama. Bu, gönderenin üzerinde çalışarak önceden bir blok zinciri hazırlamasını engeller.
yeterince ileri gidebilecek kadar şanslı olana kadar sürekli olarak, ardından işlemi
o an. İşlem gönderildikten sonra, dürüst olmayan gönderen, bir
kendi işleminin alternatif bir versiyonunu içeren paralel zincir.
Alıcı, işlem bir bloğa eklenene ve z blokları eklenene kadar bekler.
ondan sonra bağlantılı. Saldırganın ne kadar ilerleme kaydettiğini tam olarak bilmiyor ama dürüst blokların blok başına ortalama beklenen süreyi aldığını varsayarsak, saldırganın potansiyeli ilerleme, beklenen değere sahip bir Poisson dağılımı olacaktır:
□ = z
q
p
Saldırganın şimdi yakalama olasılığını elde etmek için Poisson yoğunluğunu çarpıyoruz
o noktadan itibaren yakalayabileceği olasılıkla elde edebileceği her ilerleme miktarı:
$\sum$
$k = 0$
∞
□ k
e
$-$□
k !

· { □ *q* / *p* □ □ *z* - *k* □
k ≤ *z ise*
1
k □ *z* } ise
Dağılımın sonsuz kuyruğunu toplamamak için yeniden düzenleme ...
1− ∑
k = 0
z
□ *k e*
−□
k !
□ 1 − □ *q* / *p* □ □ *z* - *k* □ □
C koduna dönüştürülüyor ...
#include <math.h>
double AttackerSuccessProbability (double q, int z)
{
çift p = 1.0 - q;
çift lambda = z * (q / p);
çift toplam = 1.0;
int i, k;
için (k = 0; k <= z; k ++)
{
çift poisson = exp (-lambda);
için (i = 1; i <= k; i ++)
poisson * = lambda / i;
toplam - = poisson * (1 - üs (q / p, z - k));
}
getiri toplamı;
}
7

Bazı sonuçları çalıştırdığımızda, z ile üssel olarak düşme olasılığını 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 küçük çözme ...
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 dayanmadan elektronik işlemler için bir sistem önerdik. İle başladık
güçlü kontrol sağlayan dijital imzalardan yapılmış sikkelerin olağan çerçevesi
mülkiyet, ancak çift harcamayı önlemenin bir yolu olmadan eksiktir. Bunu çözmek için biz
halka açık bir işlem geçmişini kaydetmek için çalışma kanıtı kullanan bir eşler arası ağ önerdi
dürüst düğümler varsa, bir saldırganın değişmesi için hızla hesaplama açısından pratik olmayan
CPU gücünün çoğunu kontrol eder. Ağ, yapılandırılmamış basitliği ile sağlamdır. Düğümler
aynı anda çok az koordinasyonla çalışın. Mesajlar olduğu için tanımlanmalarına gerek yoktur.
belirli bir yere yönlendirilmez ve yalnızca en iyi çaba temelinde teslim edilmesi gerekir. Düğümler
şunları yapabilir:
iş kanıtı zincirini neyin kanıtı olarak kabul ederek ağdan istediğiniz zaman ayrılıp yeniden katılın.
onlar yokken oldu. CPU güçleriyle oy verirler ve kabul ettiklerini ifade ederler.
geçerli blokları genişletmeye çalışarak ve geçersiz blokları üzerinde çalışmayı reddederek reddederek
onları. Bu fikir birliği mekanizması ile ihtiyaç duyulan tüm kurallar ve teşvikler uygulanabilir.

Sayfa 9

Referanslar

[1] W. Dai, "b-money," http://www.weidai.com/bmoney.txt, 1998.
[2] H. Massias, XS Avila ve J.-J. Quisquater, "Minimum düzeyde güvenli bir zaman damgası hizmeti tasarımı
güven gereksinimleri, " *Benelüks'te Bilgi Teorisi üzerine 20. Sempozyumda* , Mayıs 1999.
[3] S. Haber, WS Stornetta, "Dijital bir belgeye zaman damgası nasıl eklenir", *Journal of Cryptology* , cilt 3, no
2, sayfalar 99-111, 1991.
[4] D. Bayer, S. Haber, WS Stornetta, "Dijital zaman damgasının verimliliğini ve güvenilirliğini artırmak,"
In *Diziler II: İletişim, Güvenlik ve Bilgisayar Bilimi Yöntemleri* , sayfa 329-334, 1993.
[5] S. Haber, WS Stornetta, "Bit dizgileri için güvenli adlar," *4. ACM Konferansı Bildirilerinde Bilgisayar ve İletişim Güvenliği üzerine* , sayfalar 28-35, Nisan 1997.
[6] A. Back, "Hashcash - bir hizmet reddi karşı önlemi,"
http://www.hashcash.org/papers/hashcash.pdf, 2002.
[7] RC Merkle, "Açık anahtarlı şifreleme sistemleri için protokoller", *Proc. 1980 Güvenlik Sempozyumu ve*
Gizlilik , IEEE Computer Society, sayfalar 122-133, Nisan 1980.
[8] W. Feller, "Olasılık teorisine ve uygulamalarına giriş," 1957.