Tezos - kendi kendini değiştiren bir kripto defter

Beyaz kağıt

LM Goodman

2 Eylül 2014

Orijinal belge ile mevcut uygulamamız arasındaki değişiklikler
kırmızıyla belirtilmiştir.

"Bizim iddiamız düz değil
dairesel, ancak buna benzer bir şey. "
- Willard van Orman Quine

Öz

Genel ve kendi kendini değiştiren bir kripto defteri olan Tezos'u sunuyoruz. Tezos
herhangi bir blok zinciri temelli defteri başlatabilir. Düzenli bir operasyon
blok zinciri, tamamen işlevsel bir modül olarak uygulanmaktadır.
ağ işlemlerinden sorumlu bir kabuk. Bitcoin, Ethereum, Cryptonote,
vb. uygun girişler uygulanarak Tezos içinde temsil edilebilir.
ağ katmanına terface.
En önemlisi, Tezos meta yükseltmeleri destekler: protokoller,
kendi kodlarını değiştirerek gelişirler. Bunu başarmak için Tezos,
paydaşların değişikliği onaylaması için bir prosedür tanımlayan bir başlangıç protokolü-
Oylama prosedüründeki değişiklikler de *dahil olmak üzere* protokolde yapılan değişiklikler
kendisi. Bu, filozof Peter Suber'in inşa edilmiş bir oyun olan Nomic [3] 'e benzemez.
tamamen içe dönük bir kurallar dizisi etrafında.
Ek olarak, Tezos'un tohum protokolü saf bir risk kanıtı üzerine kuruludur.
sistemi ve Turing eksiksiz akıllı sözleşmelerini destekler. Tezos uygulanıyor
güçlü bir işlevsel programlama dili olan OCaml'de
hız, kesin bir sözdizimi ve anlam ve bir ekosistem oluşturma
Tezos, resmi doğruluk kanıtları için iyi bir aday.
Bitcoin protokolüne ve temel kriptografik ilkelere aşinalık
Bu yazının geri kalanında çatallar varsayılmıştır.

İçindekiler

---

3. Sayfa

1. Giriş

Bu yazının ilk bölümünde, soyut blok zincirleri kavramını tartışacağız.
ve kendi kendini değiştiren bir kripto defterinin uygulanması. İkinci bölümde,
önerilen tohum protokolümüzü açıklayacağız.

2 Kendini düzelten kriptolu hesaplayıcı

Bir blok zinciri protokolü üç farklı protokole ayrılabilir:
- Ağ protokolü, blokları keşfeder ve işlemleri yayınlar.
- İşlem protokolü, bir işlemi neyin geçerli kıldığını belirtir.
- Konsensüs protokolü, benzersiz bir zincir etrafında fikir birliği oluşturur.

Tezos, genel bir ağ kabuğu uygular. Bu kabuk,
işlem protokolü ve fikir birliği protokolü. İşleme atıfta bulunuyoruz
protokol ve fikir birliği protokolü bir "blok zinciri protokolü" olarak birlikte. Biz
önce bir blok zinciri protokolünün matematiksel bir temsilini verecek ve sonra
Tezos'taki bazı uygulama seçeneklerini açıklar.

2.1 Matematiksel gösterim

Bir blok zinciri protokolü, temelde eşzamanlı bir monadik uygulamasıdır.
küresel bir devletin mutasyonları. Bu, operatörler olarak "blokları" tanımlayarak elde edilir
bu küresel duruma göre hareket ediyor. Oluşum durumuna etki eden serbest tek bloklu blok
bir ağaç yapısı oluşturur. Küresel, kanonik bir durum, minimal yaprak olarak tanımlanır
belirli bir sipariş için.

Bu, aşağıdaki soyut temsili önermektedir:
- ($\mathbf{S}$, $\leq$) tamamen sıralı, sayılabilir, olası durumlar kümesi olsun.
- $\varnothing$ / $\in \mathbf{S}$ özel, geçersiz bir durumu temsil etsin .
- $\mathbf{B} \subset \mathbf{S}$ $\mathbf{S} \cup \{\varnothing\}$ bloklar kümesi olsun. *Geçerli* bloklar kümesi $\mathbf{B} \cap \mathbf{S}$ **S'dir** .
$\mathbf{S}$ üzerindeki toplam sıralama $\forall s \in \mathbf{S}$, $\varnothing <$*s olacak* şekilde genişletilir . Bu emir caydırıcı
Blok ağacında yaprak oluşturan mayınlar kanonik maden olarak kabul edilir. Bloklar
içinde **B** durumuna aktörleri olarak görülür.
Sonuç olarak, herhangi bir blockchain protokolü1 (Bitcoin, Litecoin, Peercoin, Ethereum,
Cryptonote, vb.), Tuple tarafından tamamen belirlenebilir:
(
$\mathbf{S}$, $\leq$ $\varnothing$, $\mathbf{B} \subset \mathbf{S}$ $\mathbf{S} \cup \{\varnothing\}$
)
1 HAYALET, ağacın özelliklerine göre yaprakları sıralayan bir yaklaşımdır. Bu tür bir
yaklaşım hem teorik hem de pratik nedenlerle sorunludur. Neredeyse her zaman daha iyidir
ana zincire madencilik kanıtları ekleyerek onu taklit etmek.

Ağ protokolü, bu blok zincirleri için temelde aynıdır.
"Madencilik" algoritmaları, ağın ortaya çıkan bir özelliğidir,
blok oluşturma için teşvikler.
Tezos'ta, blokların harekete geçmesine izin vererek bir blok zinciri protokolünü iç gözlem yapıyoruz.
protokolün kendisinde. Daha sonra protokol kümesini yinelemeli olarak şu şekilde ifade edebiliriz:
$P =$
{(
$\mathbf{S}$, $\leq$ $\varnothing$, $\mathbf{B} \subset \mathbf{S}$ ($\mathbf{S} \times P$) $\cup \{\varnothing\}$
)}

2.2 Ağ kabuğu

Bu resmi matematiksel açıklama bize blok ağacını *nasıl* inşa edeceğimizi söylemez .
Bu, bir dedikodu arasında bir arayüz görevi gören ağ kabuğunun rolüdür.
ağ ve protokol.
Ağ kabuğu, istemci tarafından bilinen en iyi zinciri koruyarak çalışır.
Üç tür nesnenin farkındadır. İlk ikisi işlemler ve bloklardır.
yalnızca geçerli kabul edilirse ağ üzerinden yayılır. Üçüncüsü
protokoller, mevcut protokolü değiştirmek için kullanılan OCaml modülleri. Olucaklar
daha sonra daha ayrıntılı olarak açıklanacaktır. Şimdilik işleme odaklanacağız ve
bloklar.
Ağ kabuğunun en zorlu kısmı, düğümleri bunlara karşı korumaktır.
hizmet reddi saldırıları.

2.2.1 Saat

Her blok, ağ kabuğunun görebildiği bir zaman damgası taşır. Görünen bloklar
zaman damgaları birkaç dakika içindeyse gelecekten gelmek için ara belleğe alınır
sistem zamanı ve aksi takdirde reddedilir. Protokol tasarımı tolerans göstermelidir
istemcilerde makul saat sapmaları ve zaman damgalarının olabileceğini varsaymalıdır.
tahrif edildi.

2.2.2 Zincir seçim algoritması

Kabuk, tam bir blok ağacı yerine tek bir zincir tutar. Bu zincir
yalnızca müşteri kesinlikle daha iyi bir zincirin farkına varırsa üzerine yazılır.
Bir ağacın bakımı, ağ iletişimi açısından daha cimri olacaktır.
ancak bir saldırganın hizmet reddi saldırılarına
çok sayıda düşük puanlı ancak geçerli çatal üretir.
Yine de, bir düğümün belirli bir zincirin skoru hakkında yalan söylemesi mümkün olmaya devam
ediyor:
müşterinin ancak potansiyel olarak büyük bir işlem yaptıktan sonra ortaya çıkarabileceğini yalan

blok sayısı. Bununla birlikte, böyle bir düğüm daha sonra göz ardı edilebilir.
Neyse ki, bir protokol, düşük puanlı zincirlerin sergilediği özelliğe sahip olabilir.
düşük blok oluşturma oranı. Bu nedenle, müşteri yalnızca birkaç bloğu dikkate alacaktır.
Açıklanan skorun bir yalan olduğu sonucuna varmadan önce "zayıf" bir çatallaşma.
4

5.Sayfa

2.2.3 Ağ düzeyinde savunma
Ek olarak, kabuk "savunma" niteliğindedir. Birçok eşe bağlanmaya çalışır.
çeşitli IP aralıkları. Bağlantısı kesilen eşleri tespit eder ve kötü niyetli düğümleri yasaklar.
Protokol, belirli hizmet reddi saldırılarına karşı koruma sağlamak için
blokların ve işlemlerin boyutuna göre bağlama bağlı sınırlara sahip kabuk.
2.3 Fonksiyonel gösterim
2.3.1 Zincirin doğrulanması
Soyut blok zincirimizin neredeyse tüm genelliğini verimli bir şekilde yakalayabiliriz
aşağıdaki OCaml türleri ile yapı. Başlangıç olarak, bir blok başlığı
şu şekilde tanımlanır:
tip {raw_block_header =
pred: Block_hash.t;
başlık: Bytes.t;
işlemler: Operation_hash.t listesi;
zaman damgası: float;
}
Başlık alanını kasıtlı olarak daha güçlü bir şekilde yazmıyoruz, böylece
keyfi içeriği temsil eder. Ancak, bunun için gerekli alanları yazıyoruz
kabuğun çalışması. Bunlar, önceki bloğun karmasını içerir, bir liste
işlem karmaları ve bir zaman damgası. Uygulamada, operasyonlar bir
blok, ağ seviyesinde bloklarla birlikte iletilir. Operasyonlar
kendileri keyfi lekeler olarak temsil edilir.
tip raw_operation = Bytes.t
Durum, çevreleyen bir **Bağlam** modülü yardımıyla temsil edilir.
disk tabanlı değişmez bir anahtar-değer deposunu sulates. Bir anahtar / değer çiftinin yapısı
mağaza çok yönlüdür ve çok çeşitli durumları verimli bir şekilde temsil etmemize olanak tanır.
module Context = **sig**
türü t
tip anahtar = dize listesi
val get: t -> anahtar -> Bytes.t seçeneği Lwt.t
val set: t -> key -> Bytes.t -> t Lwt.t
val del: t -> anahtar -> t Lwt.t
(... *)*
son
Disk işlemlerini engellemekten kaçınmak için, işlevler eşzamansız
monad Lwt [4] . Bağlam üzerindeki işlemlerin tamamen işlevsel olduğuna dikkat edin:
get , **ayarlıyken** bir istisna atmak yerine monad **seçeneğini** kullanır ve
del her ikisi de yeni bir **Bağlam** döndürür . **Bağlam** modülü bir kombinasyonunu kullanır
bellek önbelleğe alma ve disk depolamanın verimli bir şekilde
değişmez mağaza.
Artık rastgele bir blockchain protokolünün modül türünü tanımlayabiliriz:
5

Sayfa 6

type score = Bytes.t listesi
modül türü PROTOCOL = **sig**
tip işlem

val parse_block_header: raw_block_header -> block_header seçeneği
val parse_operation:
Bytes.t -> işlem seçeneği
val uygula:
Context.t ->
block_header seçeneği ->
(Operation_hash.t * operation) listesi ->
Context.t seçeneği Lwt.t
değer puanı: Context.t -> puan Lwt.t
(... *)*
son
Artık durumları matematiksel modeldeki gibi doğrudan karşılaştırmıyoruz, bunun yerine
biz proje **Bağlam** kullanarak bayt listesi üzerine **skor** fonksiyonu. Listesi
baytlar önce uzunluğa, ardından sözlüksel sıraya göre sıralanır. Bu oldukça
yazılım sürümlemesinde kullanılana benzer genel yapı, oldukça
çeşitli sıralamaları temsil etmede çok yönlü.
Neden protokol modülleri içinde bir karşılaştırma işlevi tanımlamıyorsunuz? İlk
kapalı, böyle bir işlevin bir
toplam sipariş. Skor projeksiyonu her zaman bunu doğrular (bağlar,
son bloğun karması). İkincisi, prensip olarak karşılaştırma yeteneğine ihtiyacımız var
farklı protokoller arasında devletler. Spesifik protokol değişiklik kuralları muhtemelen
bunun gerçekleşmesi son derece düşük bir ihtimaldir, ancak ağ kabuğu
biliyorum.
Operasyonlar **parse_block_header** ve **parse_operation** maruz
kabuğa ve tamamen yazılı işlemleri ve blokları protokole geçirmesine izin verin
aynı zamanda bu işlemlerin ve blokların iyi biçimlendirilip biçimlendirilmediğini kontrol etmek için
işlemleri aktarmaya veya yerel blok ağacı veritabanına bloklar eklemeye karar verme.
Uygulama işlevi, protokolün kalbidir:
- Bir blok başlığı ve ilişkili işlemler listesi iletildiğinde,
bağlamda yapılan değişiklikleri hesaplar ve değiştirilmiş bir kopya döndürür.
Dahili olarak, bir versiyonlama sisteminde olduğu gibi, yalnızca fark saklanır.
bir sürüm tanıtıcısı olarak bloğun karması.
- Sadece bir operasyonlar listesinden geçtiğinde, açgözlülükle başvurmaya çalışır.
mümkün olduğunca çok işlem. Bu fonksiyon, cihaz için gerekli değildir.
protokolün kendisi, ancak geçerli bloklar oluşturmaya çalışan madenciler için çok faydalıdır.

2.3.2 Protokolün değiştirilmesi

Tezos'un en güçlü özelliği, yapabilen protokolleri uygulama yeteneğidir.
kendini düzeltme. Bu, iki prosedür işlevinin
protokol:
- **set_test_protocol, test ağında** kullanılan protokolü şu şekilde değiştirir:

---

yeni bir protokol (tipik olarak bir paydaş aracılığıyla kabul edilen bir protokol)
seçmen).
- **promosyon_test_protocol** , mevcut protokolü ile değiştiren
şu anda test edilen protokol
Bu işlevler, ilişkili protokolü değiştirerek bir Bağlamı dönüştürür.
Yeni protokol, zincire aşağıdaki blok uygulandığında etkili olur.
module Context = **sig**
türü t
(... *)*
val set_test_protocol: t -> Protocol_hash.t Lwt.t
val promot_test_protocol: t -> Protocol_hash.t -> t Lwt.t
son

Protocol_hash olan **sha256** bir arşivini karma **.ml** ve **.mli** dosyaları.
Bu dosyalar anında derlenir. Küçük bir standart kitaplığa erişimleri var
ancak korumalı alanlıdır ve herhangi bir sistem çağrısı yapmayabilir.
Bu işlevler , protokolün **uygulama** işlevi aracılığıyla çağrılır.
yeni **Bağlamı** döndürür .
Birçok koşul bir protokol değişikliğini tetikleyebilir. En basit haliyle,
bir paydaş oyu, bir protokol değişikliğini tetikler. Daha karmaşık kurallar,
aşamalı olarak oylanmalıdır. Örneğin, paydaş isterse geçebilir.
bir bilgisayar sağlamak için daha fazla değişiklik gerektirecek bir değişiklik
Yeni değişikliğin belirli özelliklere saygı duyduğunun kontrol edilebilir kanıtı. Bu
"anayasanın" etkili ve algoritmik kontrolü.

2.3.3 RPC

GUI oluşturma işini kolaylaştırmak için protokol bir JSON-
RPC API. API'nin kendisi, türleri gösteren bir json şeması ile tanımlanır.
çeşitli prosedürlerin. Tipik olarak **get_balance** gibi işlevler **şu** şekilde olabilir:
RPC'de uygulanmaktadır.

type service = {
isim: dize listesi;
girdi: json_schema seçeneği;
çıktı: json_schema seçeneği;
uygulama: Context.t -> json -> json seçeneği Lwt.t
}

Ad, prosedürlerde ad alanlarına izin veren bir dizgi listesidir. Giriş
ve çıktı isteğe bağlı olarak bir json şeması ile açıklanır.
Çağrının, genellikle yeni bir bağlamda yapıldığına dikkat edin.
en yüksek puan alan yaprağın atası. Örneğin, bağlam altı sorgulama
en yüksek puanlı yaprağın üzerindeki bloklar, defterin durumunu altı ile gösterir.
onaylar.
Kullanıcı arayüzünün kendisi, protokolün belirli bir sürümüne veya genel
JSON belirtiminden elde edilmiştir.
7

---

8. Sayfa

3 Tohum protokolü

Blok zincirlerinin bir genesis hash'ından başlaması gibi, Tezos da bir tohum pro-
tocol. Bu protokol, herhangi bir blok zinciri tabanlı sanal olarak yansıtacak şekilde değiştirilebilir.
algoritması.

3.1 Ekonomi

3.1.1 Madeni paralar

Başlangıçta 100000000000 (on milyar) madeni para vardır (
token arzı, kitle satışı sırasında verilen token sayısı olacaktır,
özellikle "10 milyar", sadece bir yer tutucuydu. Bu boyut değişikliği
eldeki ana öğe üzerinde hiçbir etkisi yoktur) , iki ondalık basamağa kadar bölünebilir (için
Doğruluk uğruna, gerçekte ondalıktan sonra sekiz basamak kullanıyor olabiliriz) .
Tek bir madeni paranın "tez" olarak adlandırılmasını ve en küçüğünün
birim basitçe bir kuruş olarak. Ayrıca ꜩ (\ ua729, "Latince
küçük harf tz ") tez temsil eder. Bu nedenle 1 cent = ꜩ 0 . 01 = bir saatte bir
bir tez.

3.1.2 Madencilik ve imza ödülleri

İlke Herhangi bir merkezi olmayan para biriminin güvenliğinin yeniden
katılımcıları maddi bir ödül ile teşvik etmek istiyor (biz
şu anda ödül programını tamamlama süreci) . Açıklandığı gibi
pozisyon belgesi, yalnızca işlem maliyetlerine dayanarak, bir trajediden muzdariptir.
ortak. Tezos'ta, bir tahvil ve ödül kombinasyonuna güveniyoruz.
Tahviller bir yıldır (tahviller, yüksek

fırsat maliyeti ve bağlanma süresini uzatmanın güvenliğine çok az fayda
Geçtiğimiz bir döngü) madenciler tarafından satın alınan güvenlik mevduat (cirant da olacak
tahvil satın almak için gereklidir) . Çifte imza durumunda bu tahviller
kaybedildi.
Bir yıl (döngü) sonra, madenciler (ve ciro edenler) ile birlikte bir ödül alırlar.
fırsat maliyetlerini telafi etmek için bağları. Güvenlik öncelikle
tahvilin değerinden ve ödülün sadece küçük bir
bu değerin yüzdesi.
Tahvillerin amacı, ihtiyaç duyulan ödül miktarını azaltmak ve
belki kayıptan kaçınma etkisini ağın avantajına kullanmak için.
Özellikler Çekirdek protokolünde, bir blok madenciliği ʇ 512 ve
bir bağ gerektiren ʇ bir blok teklifler 32Δ bir ödül imzalanması 1536 *T*
- 1 tez nerede
Δ *T* , imzalanan blok ile blok arasındaki dakika cinsinden zaman aralığıdır.
selef. Blok başına 16 adede kadar imza vardır ve imzalama,
bağ. Bu rakamlar 10 milyar jeton arzına dayanıyordu ve biz
buna göre ince ayar yapın. Blok başına imza sayısını artırabiliriz
simülasyonlarda bulduğumuz gibi,
çatallar.
8

Sayfa 9

Bu nedenle, dakikada bir blok madencilik oranı varsayarsak,
ilk para kütlesi, ilkinden sonra emniyet bonosu şeklinde tutulmalıdır.
yıl. yukarıdaki parametrelerin ayarlanmasına bağlı olarak değişebilir .
Ödül programı en fazla% 5,4 *nominal* enflasyon oranını (
toplam blok ödülleri yine de yılda yaklaşık% 5'ten başlayacak, ancak bir
toplam token sayısına asimptotik sınır. Alakasız olduğunu düşünüyoruz
yönetişim modeli, token sahibinin menfaati ile uyumludur, ancak önemlidir
bazı insanlara bu yüzden gönülsüzce düşünüyoruz) . *Nominal* enflasyon nötrdür,
kimseyi ne kızdırır ne de yoksullaştırır2 .
Bir yılın döneminin bloğun zaman damgalarından belirlendiğini unutmayın,
blok sayısından değil. Bu, uzunluğa ilişkin belirsizliği ortadan kaldırmak içindir.
madenciler tarafından yapılan taahhüdün
İleriye bakma Önerilen ödül, madencilere işlerinden% 33 getiri sağlar.
tahvil (şu anda bu parametreleri revize ediyoruz, ancak yakında bir yöntemi tamamlayacağız
bu tüm taraflar için mantıklıdır) . İlk günlerde bu geri dönüşün yüksek olması gerekiyor
madenciler ve imzalayanlar, potansiyel olarak değişken bir varlığın tamamı için ellerinde tutmayı
taahhüt ettikçe
yıl (tahviller tam bir yıl değil, yalnızca bir döngü boyunca sürecek) .
Ancak Tezos olgunlaştıkça bu dönüş kademeli olarak
geçerli faiz oranı. % 1'in altındaki nominal bir enflasyon oranı güvenli bir şekilde
elde edildi, ancak bunu yapmanın herhangi bir anlamı olmayacağı net değil.

3.1.3 Kayıp paralar

Parasal kütleye ilişkin belirsizliği azaltmak için, gösterilen adresler
bir yıldan uzun süredir hiçbir aktivite (zaman damgalarına göre belirlendiği üzere) boyunca yok
edilmez
içerdikleri paralarla (aktif olmayan adresler artık paralarını kaybetmeyecekler)
Beyaz kitapta başlangıçta önerildiği gibi bir yıl sonra, yalnızca
tekrar aktif hale gelene kadar stake etme hakları. Anlamı, eğer bir
adres etkin değil, bloklar oluşturmak için seçilmeyecek (bu yavaşlayacaktır)
mutabakat algoritmasını aşağıya doğru) ve kabul edilene kadar oy kullanmasına izin verilmeyecektir.
yeniden etkinleştirildi (katılım oranıyla ilgili belirsizliği önlemek için) .

3.1.4 Değişiklik kuralları

Değişiklikler, *N* = 2 17 = 130072 blok süren seçim döngüleri üzerinden kabul edildi.

her biri. Bir dakikalık blok aralığı verildiğinde, bu yaklaşık üç takvim ayıdır.
Seçim döngüsü, 2 15 = 32 768 bloğun dört çeyreğine bölünmüştür . Bu
döngü, erken iyileştirmeleri teşvik etmek için nispeten kısadır, ancak beklenmektedir
daha fazla değişiklik, döngünün uzunluğunu artıracaktır (protokol yükseltme oyları
hızlı yinelemeye izin vermek için ilk yıl çok daha sık olacaktır.
Bir güvenlik önlemi olarak, Tezos vakfının veto yetkisi,
oylama prosedüründeki herhangi bir karışıklığı ortadan kaldırana kadar on iki ay) . Benimseme
belirli bir çoğunluğun karşılanmasını gerektirir. Bu yeter sayı Q =% 80 ile başlar, ancak
2 Buna karşılık, Bitcoin'in madencilik enflasyonu, Bitcoin sahiplerini bir bütün olarak ve merkezi
Bankacılık tasarruf sahiplerinin pahasına finans sektörünü canlandırıyor

---

Sayfa 10

ortalama katılımı yansıtacak şekilde dinamik olarak uyarlanır. Bu gerekli ise
sadece kayıp paralarla uğraşmak için.
İlk çeyrek Protokol değişiklikleri, karma değerinin gönderilmesiyle önerilmektedir.
yeni bir protokolü temsil eden .ml ve .mli dosyalarından oluşan bir tarball. Paydaşlar olabilir
bu protokollerin herhangi bir sayısının onaylanması. Bu, "onay oylaması" olarak bilinir,
özellikle sağlam bir oylama prosedürü.
İkinci çeyrek Değişiklik ilk **çeyrekte** en çok onay alan
Çeyrek artık oylamaya tabi. Paydaşlar lehine, aleyhine veya
açıkça çekimser kalmayı seçebilir. Çekimser yeter sayıya sayılır.
Üçüncü çeyrek Yetersayı karşıladıysa (açık çekimserlikler dahil) ve
değişiklik yayların% 80'ini aldı, değişiklik onaylandı ve
test protokolü. Aksi takdirde reddedilir. Ulaşılan yeter sayının *q olduğunu* varsayarsak ,
minimum çekirdek Q şu şekilde güncellenir:
$Q \leftarrow 0 . 8 S + 0 . 2 q.$
Bu güncellemenin amacı, oylama prosedürüne neden olan madeni paraların kaybolmasını önlemektir.
zamanla sıkışmak. Minimum yeter sayı üstel bir harekettir
önceki her seçimde ulaşılan nisabın ortalaması.
Dördüncü çeyrek Değişikliğin onaylandığını varsayarsak,
üçüncü çeyreğin başından beri test ağında çalışıyor. Paydaşlar
test protokolünü ana mecraya tanıtmak istediklerini onaylamak için ikinci kez oylayın.
protokol. Bu aynı zamanda yeterli çoğunluğun karşılanmasını ve% 80 üstünlüğü gerektirir.
Değişikliklere karşı kasıtlı olarak muhafazakar bir yaklaşım seçtik. Ancak,
paydaşlar bu politikayı gevşeten veya sıkılaştıran değişiklikleri kabul etmekte özgürdür
bunu yararlı bulmalılar mı

3.2 Proof-of-Stake mekanizması

3.2.1 Genel Bakış

Teminat kanıtı mekanizmamız, Slasher [1] dahil olmak üzere birkaç fikrin bir karışımıdır ,
aktivite zinciri [2] ve yanma kanıtı. Aşağıdakiler kısa bir genel bakıştır.
algoritması, bileşenleri aşağıda daha ayrıntılı olarak açıklanmıştır.
Her blok rastgele bir paydaş (madenci) tarafından çıkarılır ve birden çok
rastgele paydaşlar tarafından sağlanan önceki bloğun hafif imzaları (
imzalayanlar). Madencilik ve imzalama hem küçük bir ödül sunar hem de
bir yıl (bağsızlık bir döngüden sonra olacak ve başlangıçta olduğu gibi bir yıl olmayacak)
önerildi. Sürenin bir döngüden daha uzun süre uzatılması, gerçekten iyileşme sağlamadı.
çok miktarda sermayeyi hareketsiz hale getirme pahasına teminat) kaybedilecek güvenlik depozitosu
çift madencilik veya çift imzalama durumunda.

Sayfa 11

Protokol, 2048 blokluk döngülerde ortaya çıkar. Her döngünün başında,
rastgele bir tohum, madencilerin seçtiği ve taahhüt ettiği sayılardan türetilir

sondan bir önceki döngüde ve sonda ortaya çıktı. Bu rastgele tohumu kullanarak, bir
madeni para stratejisini takip edin, migning haklarını ve imzalama haklarını tahsis etmek için
kullanılır.
sonraki döngü için belirli bir adres. Şekil 1'e bakınız .

3.2.2 Saat

Protokol, bloklar arasında minimum gecikmeler uygular. Prensip olarak, her blok
herhangi bir paydaş tarafından çıkarılabilir. Ancak, belirli bir blok için her bir paydaş
rastgele bir minimum gecikmeye tabidir. En yüksek pay sahibi
öncelik, önceki bloktan bir dakika sonra bloğu mayınlayabilir. Pay-
İkinci en yüksek önceliği alan tutucu, bloğu iki dakika boyunca mayınlayabilir
önceki bloktan sonra üçüncü, üç dakika vb.
Bu, paydaşların yalnızca küçük bir kısmının
haraç, düşük bir blok oluşturma oranı sergileyecektir. Durum bu değilse, bir CPU
hizmet reddi saldırıları, düğümleri bir
çok uzun zincirin çok yüksek bir puana sahip olduğu iddia edildi.

3.2.3 Rastgele tohum oluşturma

Çıkarılan her blok, tarafından seçilen rastgele bir sayıya bir hash taahhüdü taşır.
madenci. Bu sayılar bir sonraki döngüde ceza altında açıklanmalıdır.
güvenlik bağını kaybetme. Bu sert ceza, seçici davranmayı önlemek içindir.
entropisine saldırmak için dava edilebilecek sayıların tutulması
tohum.
Bir sonraki döngüdeki kötü niyetli madenciler bu tür açıklamaları sansürlemeye çalışabilir.
ancak tek bir blokta birden fazla sayı ortaya çıkabileceğinden, bunlar çok
başarılı olma olasılığı düşük.
Bir döngüde açığa çıkan tüm sayılar bir hash listesinde birleştirilir ve
tohum, scrypt anahtarı türetme işlevi kullanılarak kökten türetilir.
anahtar türetme, tohumun türetilmesinin sırasını alması için ayarlanmalıdır.
tipik bir blok için ortalama doğrulama süresinin yüzde bir kısmı
masaüstü bilgisayar.
Şekil 1: Proof of Stake mekanizmasının dört döngüsü

3.2.4 Madeni parayı takip etme prosedürü

Rastgele bir paydaş seçmek için madeni para prosedürünü takip ediyoruz.
İlke Fikir, bitcoin'de satoshi'yi takip etmek olarak bilinir. Prosedürler
şimdiye kadar basılan her satoshi'nin benzersiz bir seri numarası varmış gibi çalışır. Satoshiler
oluşturma zamanına göre dolaylı olarak sıralanır, rastgele bir satoshi çizilir ve izlenir
blok zinciri aracılığıyla. Tabii ki, bireysel sentler doğrudan takip edilmiyor.
Bunun yerine, girdiler birleştirildiğinde ne olacağını açıklamak için kurallar uygulanır.
ve birden fazla çıktı için harcandı.
Sonunda, algoritma, aşağıdakilerle ilişkili bir dizi aralığı izler:
her anahtar. Her aralık, bir satoshi "aralığını" temsil eder. Maalesef bitti
zaman geçtikçe, veritabanı gittikçe daha parçalı hale gelir ve veri tabanındaki şişkinliği artırır.
müşteri tarafı.
Madeni Para Ruloları Büyük "madeni para" oluşturarak önceki algoritmayı optimize ediyoruz
10 000 tezden oluşan rulo ”. Bu nedenle, yaklaşık bir milyon rulo var.
Bir veritabanı, her ruloyu mevcut sahibiyle eşler.
Her adres, belirli bir dizi özel rulonun yanı sıra bazı gevşek değişiklikleri de içerir.
Tam rulonun bir kısmını harcamak istediğimizde, rulo bozulur ve seri
numara LIFO yuvarlanma sırasında gönderilir, bir tür "belirsizlik". Her işlem
kırılan rulo sayısını en aza indirecek şekilde işlenir. Ne zaman bir
adres bir rulo oluşturmaya yetecek kadar bozuk para tutar,
sıra ve rulo yeniden oluşturulur.
LIFO önceliği, gizli bir çatal üzerinde çalışan bir saldırganın

hesaplar arasında değişiklik yaparak elinde tuttuğu paraları değiştirebilir.
Bu yaklaşımın küçük bir dezavantajı, hissenin aşağı yuvarlanmasıdır.
en yakın tam sayı rulo sayısı. Bununla birlikte, bu büyük bir gelişme sağlar
Satoshi'yi takip etme yaklaşımına göre verimlilikte.
Rulolar numaralandırılırken, bu yaklaşım,
Zerocash gibi esnekliği koruyan protokoller. Bu tür protokoller aynı şeyi kullanabilir
"Limbo" kuyruk tekniği.
Motivasyon Bu prosedür, işlevsel olarak sadece çizim yapmaktan farklıdır.
teraziye göre ağırlıklandırılmış rastgele adres.
Aslında, gizli bir çatalda, bir madenci nesli kontrol etmeye çalışabilirdi.
rastgele çekirdekten ve oluşturarak kendisine imzalama ve basım haklarını atamak için
önceden uygun adresler. Rulo halinde bunu başarmak çok daha zordur
gizli çatal belirli bir sahiplik taklit edemeyeceği için rastgele seçilir.
yuvarlanır ve bu nedenle, tohuma uygulanan hash işlevinin ön görüntüsünü almaya çalışmalıdır.
imza ve basım haklarını kendisine verir.
Gerçekten de, uzunlukta bir döngü içinde , *N* = 2048, birisi bir fraksiyon tutma *f* arasında
valsler ortalama olarak *fN* madencilik haklarına sahip olacak ve efektif fraksiyon alınacak ,

---

Sayfa 13

f 0'ın standart sapması
√
1
N
√
1 - *f*
f
.
Bir saldırgan *W* farklı tohumlar aracılığıyla kaba kuvvetle arama yapabilirse ,
o zaman beklenen avantajı en fazla 3'tür
(√
2 günlük (*W*)
N
√
1 - *f*
f
)
fN
bloklar. Örneğin, saldırgan kontrol *f* silindirlerinin =% 10 olmalıdır
döngü başına yaklaşık 205 blok çıkarılması bekleniyor. Denediği gizli bir çatalda
bir trilyon karma üzerinden hesapladığını varsayarak tohumu kontrol etmek için
kendisi yaklaşık 302 blok veya yaklaşık 14 blok . Blokların% 7'si. Dikkat:
- Tohumun türetildiği karma pahalı bir anahtar türetmedir
işlevi, kaba kuvvet aramasını pratik olmayan hale getirir.
- Çıkarılan bloklarda doğrusal kazançlar elde etmek için, saldırıya uğrayanın bir
ikinci dereceden üstel çaba.

3.2.5 Maden blokları

Rastgele tohum, bir ruloyu tekrar tekrar seçmek için kullanılır. İlk atış seçildi
paydaşının bir dakika sonra, ikincisi sonra bir blok madenciliği yapmasına izin verir.
iki dakika - vb.
Bir paydaş tohumu gözlemlediğinde ve yüksek bir öncelik verebileceğini fark ettiğinde
bir sonraki döngüde blok, güvenlik depozitosu yapabilir.
Potansiyel olarak sorunlu bir durumdan kaçınmak için hiçbir paydaş bir
16 dakikalık bir gecikmeden sonra belirli bir bloğu çıkarmak için güvenlik depozitosu, blok
depozito olmadan çıkarılabilir.

Tahviller, herhangi bir zincirde derhal alıcılarına iade edilir
bloğu mayınlamadıkları yer.

3.2.6 İmzalama blokları

Olduğu gibi, neredeyse çalışan bir kazık kanıtı sistemimiz var. Bir tanımlayabiliriz
zincirin ağırlığı blok sayısı olacaktır. Ancak bu kapıyı açar
bencil bir madencilik biçimine.
Bu nedenle bir imzalama planı sunuyoruz. Bir blok basılırken,
rastgele tohum, 16 ruloya rastgele 16 imzalama hakkı atamak için kullanılır.
İmza hakkı alan paydaşlar, basılan blokları gözlemler.
ve sonra bu blokların imzalarını gönderin. Bu imzalar daha sonra dahil edilir
3 bu, normal olarak dağıtılan maksimum W beklentisine ilişkin standart bir sınırdır.
değişken
13

Sayfa 14

bir sonraki blokta, ebeveynlerinin
blok zinciri.
İmzalayanlar tarafından alınan imza ödülü, zamanla ters orantılıdır
blok ve selefi arasındaki aralık.
İmzalayan bu nedenle, gerçekten inandıkları şeyi imzalamak için güçlü bir teşvike sahiptir.
bir noktada üretilen en iyi blok olun. Ayrıca güçlü bir teşvikleri var.
İmza ödülleri olarak hangi bloğu imzalayacakları konusunda anlaşmaya varmak için yalnızca blok
blok zincirine dahil edilir.
En yüksek öncelikli blok çıkarılmamışsa (belki de madenci açık olmadığı için)
satır), imzalayanların bir süre beklemesi için bir teşvik olabilir, sadece
madenci geç kaldı. Ancak, diğer imzalayanlar daha sonra en iyi önceliği imzalamaya karar verebilir
blok ve yeni bir blok, bu imzaları kapsayabilir ve uzatmaları dışarıda bırakabilir.
Bu nedenle, madencilerin bu stratejiyi izleme olasılığı düşüktür.
Tersine, işaretçilerin paniğe kapılıp
Diğer imzalayanların bunu yapacağından korktukları için gördükleri ilk bloğu imzalamak ve
yeni blok hemen inşa edilecektir. Ancak bu çok uydurma bir durumdur
bu kimseye fayda sağlamaz. İmzalayanların bu dengeyi düşünmeleri için hiçbir teşvik yok
bırakın programlarının kodunu bu şekilde davranacak şekilde değiştirmeyi bırakın. Bir
kötü niyetli paydaşın operasyonları aksatmaya çalışması sadece zarar verir
Başkalarının takip etme olasılığı düşük olacağından, bu stratejiyi takip etmeye çalışarak
takım elbise.

3.2.7 Zincirin ağırlığı

Ağırlık, imza sayısıdır.

3.2.8 İhbarlar

Bir bloğun çift basımını veya bir bloğun çift imzalanmasını önlemek için,
bir madenci bloğuna bir ihbar ekleyebilir.
Bu ihbar, iki imza şeklini alır. Her basım imzası
veya blok imzası bloğun yüksekliğini işaretler ve suistimalin kanıtı olur
oldukça özlü.
Herhangi birinin suiistimali kınamasına izin verebilirken, gerçekten
blok madencisinin dışındaki herhangi birine izin vermeyi işaret edin. Nitekim bir madenci basitçe
yapabilir
herhangi bir kötüye kullanım kanıtını kopyalayın ve bunu kendi keşfi olarak sunun.4
Bir taraf, çift basım veya çift imzalamadan suçlu bulunduktan sonra,
güvenlik bağı kaybedilir.

3.3 Akıllı sözleşmeler

3.3.1 Sözleşme türü

Harcanmamış çıktılar yerine Tezos durum bilgili hesaplar kullanır. O hesaplar ne zaman
çalıştırılabilir kodu belirtir, daha genel olarak sözleşmeler olarak bilinir. Bir
4 Sıfır bilgi kanıtı, herkesin suiistimalleri ihbar etmekten yararlanmasına izin verir, ancak

Bunun çok fayda sağladığı özellikle açık değil.

Sayfa 15

hesap bir sözleşme türüdür (çalıştırılabilir kodu olmayan), ikisine de şu şekilde atıfta bulunuruz:
Tam genel olarak "sözleşmeler".
Her sözleşmenin bir "yöneticisi" vardır ve bir hesap söz konusu olduğunda
Sahip. Sözleşme harcanabilir olarak işaretlenmişse, yönetici harcayabilir
sözleşmeyle ilişkili fonlar. Ek olarak, her bir sözleşme şunları belirtebilir:
Proof-of-stake protokolünde blokları imzalamak veya madencilik yapmak için kullanılan bir genel anahtarın karması.
Özel anahtar yönetici tarafından kontrol edilebilir veya edilmeyebilir.
Resmi olarak, bir sözleşme şu şekilde temsil edilir:
sözleşme **türü** = {
counter: int; *(* tekrarlanan saldırıları önlemek için karşı *)*
yönetici: id; *(* sözleşmenin yönetici genel anahtarının karması *)*
denge: Int64.t; *(* tutulan bakiye *)*
imzalayan: kimlik seçeneği; *(* imzalayanın kimliği *)*
kod: işlem kodu listesi; *(* işlem kodlarının listesi olarak sözleşme kodu *)*
depolama: veri listesi; *(* sözleşmenin saklanması *)*
harcanabilir: bool; *(* para yönetici tarafından harcanabilir mi? *)*
yetki verilebilir: bool; *(* yönetici imzalama anahtarını değiştirebilir mi? *)*
}
Bir sözleşmenin tanıtıcısı, başlangıç içeriğinin karmasıdır. Denemek
karması mevcut bir sözleşmeyle çakışacak bir sözleşme oluşturmak geçersizdir
işlem ve geçerli bir bloğa dahil edilemez.
Verilerin birleşim türü olarak temsil edildiğini unutmayın.
türü veri =
| STRING / dize
| INT **of** int
INT, 64 bitlik işaretli bir tam sayıdır ve dize, 1024'e kadar bir dizidir
bayt. Depolama kapasitesi 16 384 bayt ile sınırlıdır ve tamsayılar şu şekilde sayılır:
sekiz bayt ve uzunlukları olarak dizeler.

3.3.2 Oluşturma

Oluşturma işlemi yeni bir sözleşme oluşturmak için kullanılabilir,
sözleşmenin kodu ve sözleşmenin depolanmasının ilk içeriği. Eğer
tanıtıcı zaten mevcut bir sözleşmenin tanıtıcısıdır, oluşturma reddedilir
(Yanlışlık veya kötü niyet olmadıkça bunun olması için hiçbir neden yoktur).
Bir sözleşmenin aktif kalması için minimum bakiye 1 olmalıdır. Bakiye
bu sayının altına düşerse sözleşme feshedilir.

3.3.3 İşlemler

Bir işlem, bir sözleşmeden başka bir sözleşmeye gönderilen bir mesajdır, bu
mesajlar şu şekilde temsil edilir:
işlem **türü** = {
miktar: miktar; *(* gönderilen tutar *)*
parametreler: veri listesi; *(* betiğe aktarılan parametreler *)*
(tekrarlanan saldırıları önlemek için sayaç (fatura kimliği) *)*
counter: int;
hedef: sözleşme karması;
}

Sayfa 16

Böyle bir işlem, yöneticinin imzası kullanılarak bir sözleşmeden gönderilebilir.

anahtar veya sözleşmede kod çalıştırılarak programlı olarak gönderilebilir. Ne zaman
işlem alınır, tutar varış sözleşmesinin hesabına eklenir.
bakiye ve hedef sözleşmenin kodu yürütülür. Bu kod yapabilir
kendisine aktarılan parametrelerin kullanımı, sözleşmenin deposunu okuyup yazabilir,
imza anahtarını değiştirin ve işlemleri diğer sözleşmelere gönderin.
Sayacın rolü, tekrar saldırılarını önlemektir. Bir işlem sadece
sözleşmenin sayacı işlemin sayacına eşitse geçerlidir. Birkez
işlem uygulandığında sayaç bir artarak işlemi engelliyor
yeniden kullanılmaktan.
İşlem aynı zamanda müşterinin son bloğun blok hashini de içerir.
geçerli kabul eder. Bir saldırgan uzun bir yeniden yapılanmayı zorlamayı başarırsa
bir çatalla, bu tür işlemleri dahil edemeyecek ve çatalı yapacaktır.
belli ki sahte. Bu son bir savunma hattı, TAPOS,
uzun süreli yeniden yapılanmaları önleyin, ancak kısa vadeyi önlemek için çok iyi bir sistem değil
çift harcama.
(Account_handle, counter) çifti, kabaca harcanmamış bir
Bitcoin'de çıktı.

3.3.4 Depolama ücretleri

Depolama ağa bir maliyet getirdiği için minimum ꜩ 1 ücret değerlendirilir
depolamadaki her bayt artışı için. Örneğin, eğer bir
işlem, depoya bir tamsayı eklendi ve on karakter var
depodaki mevcut bir dizeye eklendiğinde, ꜩ 18 geri alınacaktır
sözleşmenin bakiyesinden ve tahrip edildi.

3.3.5 Kod

Dil, yüksek seviyeli veri türleri ve ilkelleri ile yığın tabanlıdır ve katı
statik tip kontrolü. Tasarımı Forth, Scheme, ML ve Cat tarafından desteklenmiştir. Bir
talimat setinin tam özellikleri [5] 'te mevcuttur. Bu şartname verir
tam talimat seti, tip sistemi ve dilin anlambilimi. Bu
kolay bir giriş değil, kesin bir referans el kitabı anlamına geliyordu.

3.3.6 Ücretler

Şimdiye kadar, bu sistem Ethereum'un işlem yapma şekline benziyor. Ancak,
ücretleri ele alma şeklimizde farklılık gösteririz. Ethereum, keyfi olarak uzun programlara izin verir
programın çalıştırılmasıyla doğrusal olarak artan bir ücret talep ederek yürütmek
ing zaman. Ne yazık ki, bu bir madencinin
işlemi doğrulayın, diğer madencilere böyle bir teşvik sağlamaz,
bu işlemi de doğrulaması gerekir. Pratikte, ilginç profesyonellerin çoğu
Akıllı sözleşmeler için kullanılabilecek gramlar çok kısadır. Böylece sadeleştiriyoruz
izin verdiğimiz adımların sayısına sert bir sınır koyarak inşaat
çalıştırılacak programlar.

Sabit başlık bazı programlar için çok sıkı olursa, yürütmeyi bozabilirler.
Birden çok adımda işlem yapın ve tam olarak yürütmek için birden çok işlem kullanın. Tezos'tan beri
değiştirilebilir, bu sınır gelecekte değiştirilebilir veya gelişmiş ilkeller
yeni işlem kodları olarak tanıtılabilir.
Hesap izin veriyorsa, imza anahtarı imzalı bir imza verilerek değiştirilebilir.
değişikliği isteyen mesaj.

4. Sonuç

Çekici bir tohum protokolü oluşturduğumuzu düşünüyoruz. Ancak Tezos'un gerçek potansiyeli
paydaşları, kendilerinin
onlara hizmet edenin en iyisi olduğunu hissediyorum.

Referanslar

[1] Vitalik
Buterin.

Slasher:
Bir
cezalandırıcı
risk kanıtı
al-
gorithm.
https://blog.ethereum.org/2014/01/15/
slasher-a-punitive-proof-of-stake-algoritması /, 2014.
[2] Ariel Gabizon Iddo Bentov ve Alex Mizrahi. Olmayan kripto para birimleri
iş kanıtı . http://www.cs.technion.ac.il/~idddo/CoA.pdf, 2014.
[3] Peter Suber. Nomic: Kendini değiştirme oyunu . http: //legacy.earlham.
edu / ~ peters / writing / nomic.htm, 1982.
[4] Jérôme Vouillon. Lwt: ortak bir iş parçacığı kitaplığı. 2008.
[5] Tezos projesi. Tezos akıllı sözleşme dilinin biçimsel özellikleri.
https://tezos.com/pages/tech.html, 2014.