DÜMF Mühendislik Dergisi
web: http://dergipark.gov.tr/dumf

Araştırma Makalesi / Research Article

Blokzincir teknolojisi: uzlaşma protokolleri

Süleyman KARDAŞ *
Batman Üniversitesi, Bilgisayar Mühendisliği Bölümü, Batman
suleyman.kardas@batman.edu.tr ORCID: 0000-0002-6197-578X, Tel: (488) 217 40 06

Geliş: 24.05.2018, Kabul Tarihi: 24.07.2018

Öz

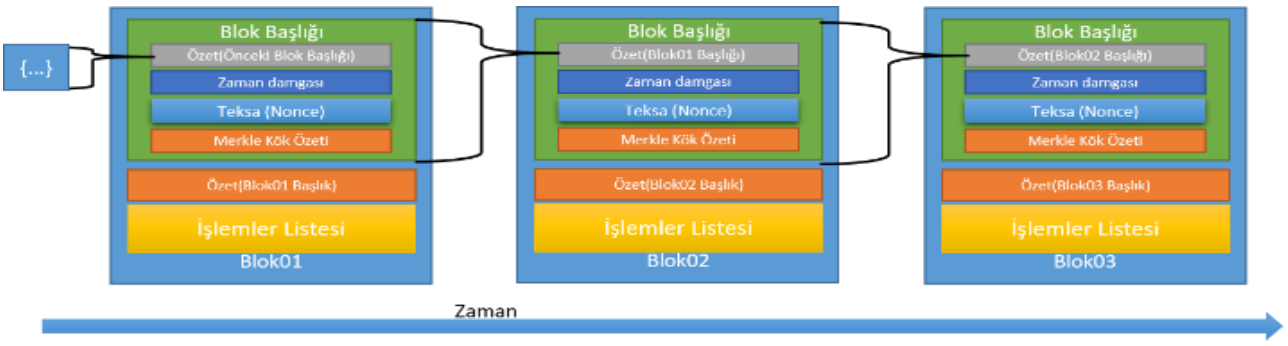
Blokzincir, veri akışını yöneten herhangi bir merkezi yetkiye sahip olmayan düğümler (eşler, sunucu/istemci) arası bir ağ sistemi olarak tanımlanır. Tüm düğümler aynı haklara sahiptirler ve düğümler birbirlerine güvenmek zorunda değildirler. Düğümler kendilerini yalnızca açık adresleriyle ifşa ederler ve kazanç arzusu ile motive olurlar, diğer madencilik düğümlerinin refahını düşünmek veya bir bütün olarak hareket etmek zorunda değildirler. Böyle bir durumda, bir düğüm neden başka bir düğüm tarafından hazırlanmış bir bloğu yayınlamak istesin? Ayrıca, birden fazla madencilik düğümü bir bloğu yaklaşık olarak aynı anda oluşturduğunda çatışmaları kim çözecektir? Bu sebeplerden ötürü, karşılıklı güvensiz kullanıcı gruplarının birlikte çalışmasına olanak tanıyan çeşitli uzlaşma protokolleri ortaya çıktı. Her bir uzlaşma modeli farklı varsayımlar ile güvenlik sorunlarını ortadan kaldırmaya çalışmaktadır. Bu çalışmada literatürde ve pratikte yer alan bu uzlaşma protokollerin avantajları ve dezavantajları kıyaslı bir biçimde verilmiştir.

Anahtar Kelimeler: *Blokzincir; Uzlaşma Protokolleri; Kripto Para*

* Yazışmaların yapılacağı yazar
DOI: 10.24012/dumf.426805

Giriş

Blokzincirler sayısal işlemlerin sahipleri tarafından elektronik olarak imzalanmasından sonra bloklara eklenmesi ve her bir bloğun bir önceki yayınlanan bloğa kriptografik olarak bağlanması ile ortaya çıkan dijital defterlerin dağıtık bir ağda kopyalarının saklanması olarak tanımlanmaktadır. Herhangi bir blokzincirin yapısı incelendiğinde özet fonksiyonları, açık anahtar sistemi, adresler, işlemler, bloklar, defterler, düğümler ve uzlaşma protokolleri olmak üzere sekiz farklı bileşenden oluştuğu görülmektedir. Özet fonksiyonları değişken uzunluktaki açık metinleri sabit uzunluktaki bir özet değerine dönüştüren fonksiyonlardır (NIST, 2015). Özet değerini bir metnin parmak izi, DNA'sıdır. Bu fonksiyonların hesaplanması çok kolay olurken, çakışma ve ters görüntü elde edilmesine karşı güvenli olmalıdırlar. Blokzincirlerde kullanılan en popüler ve güvenli özet fonksiyonlar SHA-256, Keccak-256, RIPEMD-160, Scrypt, X11 algoritmalarıdır. Açık anahtar şifrelemesi veya asimetrik kriptografi, matematiksel olarak ilişkili ve birbirinden farklı iki anahtar kullanan (açık ve özel/gizli anahtarlar) bir şifreleme protokolüdür. Açık anahtarlar açık bir metni şifreleme için kullanılırken özel anahtar ise şifreli metnin çözülmesinde kullanılmaktadır (NIST, 2013). Açık anahtar sistemi ile aynı zamanda metinler imzalanabilmektedir. Özel anahtar ile bir metnin kriptografik özeti şifrelenmesi ile sayısal imza oluşturulur. Metin ve imza değerleri açık anahtarlar ile doğrulanabilmektedir. Özel anahtarın, açık anahtarı kullanarak hesaplanması sayısal olarak mümkün değildir. Bu nedenle, açık anahtarlar serbestçe paylaşılabilir ve kullanıcılara içeriği şifrelemek ve dijital imzaları doğrulamak için kolay ve kullanışlı bir yöntem sağlar. Bu sebeple, özel anahtarlar yalnızca sahiplerinin içeriğin şifresini çözebilmesini ve sayısal imzalar oluşturmasını sağlayarak gizli tutulması gerekmektedir. Bir işlem, ağa yayınlanan ve bloklar halinde toplanan kripto para değerinin ve/veya mesajların bir adresten başka bir adrese transferi olarak tanımlanmaktadır. Kripto para transferleri dikkate alındığında bir işlem tipik olarak önceki işlem çıkışlarını yeni işlem girdileri olarak referans alır ve tüm giriş kripto para değerlerini yeni çıktılara ayırır. Blokzincirlerde genellikle işlemler şifrelenmez, bu yüzden bir blokta toplanan her işleme göz atmak ve görüntülemek mümkün olmaktadır. Her işlem, işlemi gerçekleştiren kişilerce işlemde kullanılan adrese ait özel anahtar ile imzalanmaktadır. Bunu yapmakla işlemin doğru kişilerce yapıldığı doğrulanması kolay olmaktadır. Aynı zamanda işlemi yapan kişinin bu işlemi gerçekleştirme durumunu inkâr etmesini engellemektedir. İşlemler, sistemde bulunan düğümler tarafından kontrol edilip onay verilmektedir. İşlemler yeterli onay aldığında geri dönüşümsüz sayılmaktadır. E-posta adresleri gibi, blokzincir adreslerinden birine güvenli bir şekilde kripto-para/mesaj gönderebilirsiniz. Ancak, e-posta adreslerinden farklı olarak, kullanıcıların birçok farklı blokzincir adresi olabilmektedir ve her işlem için benzersiz bir adres kullanılmalıdır. Çoğu blokzincir yazılımları ve web siteleri, her fatura veya ödeme talebi oluşturduğunuzda yepyeni bir adres üreterek bu konuda yardımcı olmaktadır. Blokzincir adresleri açık anahtarların birden fazla farklı özet fonksiyonlarından geçirilmesi ile oluşturulmaktadır. Adresler çevrimdışı olarak güvenli bir yerde açık anahtarlar kullanılarak oluşturulmaktadır. İşlemler, blok adı verilen dosyalara kalıcı olarak kaydedilir. Bloklar zaman içinde doğrusal bir dizi halinde düzenlenir ve büyür (blok zinciri olarak da bilinir). Yeni işlemler, madenci düğümler tarafından sürekli olarak zincirin sonuna eklenen yeni bloklara eklenir. Bazı blokzincir sistemlerinde (örneğin Bitcoin), her bloğa özgü matematiksel bir bulmaca önerilir ve yayınlanmadan önce bulmacanın çözümünün madenciler tarafından gerçekleştirilmesi beklenir. Öte yandan her yeni blokta bir önceki blok başlığının özeti eklenerek blokların kriptografik olarak birbirine bağlanması sağlanır (Bkz. Şekil 1). Bir blok başlığı genellikle bir önceki blok başlığının özeti, zaman damgası, *merkle* kök özeti (bloktaki işlemlerin özeti) ve o bloğa özgü tek kullanımlık sayı (teksa). Blokların birbirine bağlanması ve uzamasıyla defter oluşmaktadır. Bir blokzincir sisteminde tek bir



Şekil 1.Blokların Zincirlenmesi

tane zincir devam ettirilmekte olup tek bir defter tutulmaktadır. Sistemde birden fazla zincir oluşması durumunda uzun dönemde uzun olan zincir benimsenmekte olup diğer zincirlerde bulunan işlemler iptal edilmektedir. Defterin kopyaları birçok farklı düğümlerde tutulmaktadır.

Bir blokzincir, veri akışını yöneten herhangi bir merkezi yetkiye sahip olmayan eşler (düğümler) arası bir sistem olarak tanımlanmaktadır. Bu dağıtık ağdaki tüm düğümler herhangi bir merkezi bir denetime tabi değildirler. Her bir düğüm aynı zamanda istemci ve sunucu olabilmektedir. Her düğüm, aynı haklara sahiptir ve düğümler karşılıklı olarak birbirlerine güvenmezler. Ayrıca, ağdaki kullanıcılar kendilerini yalnızca kendi açık adresleriyle ifşa ederler. Düğümler yalnızca kazanç elde etme arzusu ile motive olur, diğer düğümlerinin refahını veya bir ağı bütün olarak düşünmezler. Sistemin şeffaf ve doğru bir şekilde yürütülmesini sağlamak için düğümler neden dürüst bir şekilde kendilerine verilen görevleri yerine getirsin? Bir düğüm, neden başka bir düğüm tarafından hazırlanmış bir bloğu kendisine bağlı diğer düğümler ile paylaşarak yaysın? Ayrıca, birden fazla madenci düğümü bir bloğu yaklaşık olarak aynı anda çözdüğünde çatışmaları kim çözecektir? Bütün bu soruların cevabı önceden tanımlanmış uzlaşma modelinde yer almaktadır. Literatürde birçok uzlaşma modeli bulunmaktadır ve her bir uzlaşma modeli blokzincirdeki sorunlara farklı açılardan bakıp farklı çözümler sunmaktadır. Her bir uzlaşma modelinin varsayımları farklı olduğundan sorunlara karşı üretmiş olduğu çözümlerin önemi farklı olabilmektedir.

Bir kullanıcı blokzincir sistemine katıldığında, sistemin başlangıç durumunu kabul etmek zorundadır. Bu, yalnızca önceden yapılandırılmış blok, yani 'genesis' bloğu içine kaydedilir. Her blokzincir yayınlanmış bir 'genesis' bloğuna sahiptir ve her blok, mutabık kalınmış mutabakata dayalı bir yöntem temelinde blokzincirine sonra eklenmelidir. Bununla birlikte, yöntem ne olursa olsun, her bir bloğun geçerliliği vardır ve bu nedenle, blokzincir ağındaki her bir kullanıcı tarafından bağımsız olarak onaylanabilir olması gerekir. Başlangıç durumunu ve o zamandan beri her bloğu doğrulama yeteneğini birleştirerek, kullanıcılar blokzincirinin mevcut durumunu kabul ederler. Blokzincirlerde birden fazla zincirin oluşması durumunda, temel varsayım herkesin en uzun zincirin benimsenmesidir. Daha sonra aşağıdaki özellikler uygulanır:

- Her kullanıcı sistemin ilk durumu üzerinde anlaşmaya varır.
- Kullanıcılar, blokların sisteme eklendiği uzlaşma modelini kabul etmiş sayılırlar.
- Her blok, bir kriptografik özet alma ile önceki bloğa bağlanır (önceki bloğu olmayan ve genellikle önceki blok için her 0'ın bir özet değerine sahip olan ilk 'genesis' bloğu hariç).
- Kullanıcılar her bloğu doğrulayabilir.

Blokzincir yaklaşımının temeli, sistemin durumunu bildirmek için güvenilir üçüncü bir şahsa gereksinim duymadan işlemleri gerçeklemesidir. Sistem içindeki tüm kullanıcılar sistemin bütünlüğünü doğrulayabilirler. Blokzincirine yeni bir blok eklemek için, katılan tüm düğümlerin zamanla ortak bir anlaşmaya varmaları gerekir, bu nedenle bazı geçici anlaşmazlıklara izin verilir. Uzlaşma metni (veya fikir birliği), muhtemel kötü amaçlı kullanıcıların blokzincirini bozmaya veya devretmeye çalışması durumunda bile çalışmalıdır. Bitcoin'de kullanılan Emek kanıtı uzlaşma

modelinden sonra onlarca farklı uzlaşma modelleri ortaya çıktı. Tüm bu uzlaşma modellerini üç farklı uzlaşma protokolü ve varyantları olarak görebiliriz. Bu uzlaşma modelleri:

- Emek Kanıtı (Proof of Work) Uzlaşma Protokolü
- Hisse Kanıtı (Proof of Stake) Uzlaşma Protokolü
- Bizans Hata Toleransı (BHT) Uzlaşma Protokolü

Bu makalenin temel amacı, blokzincir sistemlerinin en önemli parçası olan literatürde ve pratikte yer alan uzlaşma protokolleri inceleyip karşılaştırmalı olarak vermektir. Bu amaçla, öncelikle blokzincirlerde farklılıkları sağlayan önemli görülen kriterler tanımlanacaktır. Daha sonra emek kanıtı uzlaşma protokolü ve dezavantajları anlatılacaktır. Daha sonra hisse kanıtı uzlaşma protokolü ve avantajları anlatılacaktır. Daha sonra Bizans hata toleransı ve farklı sürümleri anlatılacaktır. Son olarak makalede anlatılan uzlaşma protokollerinin kıyaslı tablosu verilerek tartışmalara yer verilecektir.

Sınıflandırma Kriterleri

Blokzincirler birçok farklı şekilde oluşturulup herkes tarafından erişilebilir kılınabilirler. Örneğin, en popüler kripto para madenciliği olan Bitcoin'de, herkes tam düğüm olabildiği gibi aynı zamanda madenci olarak sisteme katkıda bulunabilmektedir. Herkes, tüm verileri okuma hakkına sahip olabilmektedir ve aynı zamanda tüm zincirin bir kopyasını kendisinde saklayarak blokzincirde yasal olan değişiklikleri yapabilirler. Tüm katılımcılar eşit ve halka açık haklara sahiptirler. Bu tarz zincirlere *halka açık* ve *izinsiz* blokzincirler denilmektedir.

Özel izne tabi tutularak erişim, okuma ve yazma haklarına sahip kılınan blokzincirlerde bulunmaktadır. Bu şekilde oluşturulan zincirlerde yer alan konfigürasyon katılımcıların işlemlerini kontrol eder ve her bir katılımcılının hangi erişimlere sahip olduğu rolleri tanımlar. Ayrıca, katılımcıların kimlik bilgilerini ağda tutabilir. Bu şekilde oluşturulan zincirlere, *izinli* blokzincir denilmektedir.

Sadece bilinen ve tanımlı katılımcıların ağa katılmasına izin veren zincirlere *gizli/özel* blokzincir olarak adlandırılırlar. İzinli blokzincirler özel blokzincirlerden farklıdırlar. Örneğin, bir banka, bankanın içinde bulunan belirli sayıda düğüm yoluyla çalıştırılan özel bir blokzincir çalıştırabilir. Buna karşılık, izin verilen blokzincirler, kimlik ve rol tanımlandıktan sonra herhangi birinin bir ağa katılmasına izin verebilir.

Bir diğer kıstas olarak zincirlerin *değişmezlik* (immutability) seviyesidir. Blokzincirler, dağıtık ağ altında verilerin yüksek oranda *değişmezliğini* ve güvenilirliğini sağlayabilen bir veri tabanı olarak tanımlanmaktadır. Bu, blokzincirine eklendikten sonra verilerin değiştirilemeyeceği anlamına gelmemektedir. Örneğin, Bitcoin ağında, madenciler her on dakikada bir blok oluştururlar. İki madencinin matematiksel denklemi çözebildikleri sürece aynı anda iki farklı blok üretmesi düşük bir ihtimalde olsa mümkündür ve bu farklı işlem kayıtlarına sahip iki zincirin oluşturulmasına yol açar. Bununla birlikte, zaman geçtikçe, iki zincirin aynı oranda uzamaya devam etme olasılığı düşer. Aynı anda iki bloğun üretilme olasılığının %30 olduğunu varsayalım. Daha sonra iki ardışık 10 dakikalık periyotta iki bloğun aynı anda üretilme olasılığı %9'tür (=% 30 *% 30). Üç ardışık 10 dakikalık periyot için olasılık daha da %0.27'e düşer (=% 30 *% 30 *%30). Başka bir deyişle, bir zaman noktasında, sadece bir zincir sürekli büyüyebilir ve bir diğeri düğümler tarafından kabul görmeyecektir. Terk edilmiş zincir içindeki veriler, diğer zincirdeki verilerle değiştirilir. Bu nedenle, veriler tamamen değiştirilebilir ve hatta silinebilir ve başka verilerle değiştirilebilirler. Bu nedenle, bir Bitcoin işlemi için bir "altı onaylama kuralı" vardır. Bir onay, yeni oluşturulan blokta görünen bir işlem anlamına gelir. Altı onay, mevcut bloktan beş blok uzaklıkta olan blokta görünen bir işleme karşılık gelir. Bu, o Bitcoin işleminin de dâhil edilmemesini engelleyebilir. Buradan şu sonuca varılır, uzun dönemde Bitcoin'de verilerin değişmesi zordur.

Blokzincir sisteminde işlemlerinin sayısı arttıkça, zincirler hızlı bir şekilde büyürler ve bu da işlemlerin yavaşlanmasına sebebiyet verir. Bunun nedeni, bir blokzincir öğesinin kaldırabileceği işlemlerin sayısı hiçbir zaman ağda bulunan düğümlerin kapasitesini aşamaz. Ağa yeni düğümler eklendikçe zincirin yönetilmesi zorlaşmaktadır çünkü düğümler arası logaritmik bir gecikme söz konusudur. Bu sebeple daha fazla işlemleri işlemek için blokzincirler *ölçeklendirilmesi* (scalability) zorlaşmaktadır. Farklı çözümler kullanılarak bazı blokzincirlerin ölçeklendirilmesi yapılabilmektedir.

Emek Kanıtı Uzlaşma Protokolü

Emek Kanıtı (Proof of Work, PoW) kavramı ilk defa Cynthia Dwork ve Moni Naor tarafından istenmeyen e-postalar ile mücadelede kullanıldı (Naor ve Dwork, 1993). Daha sonra Markus Jakobsson and Ari Juels tarafından 1999'daki makalelerinde bu yöntem resmileştirildi (Jakobsson ve Juels, 1999). Hizmet saldırılarını ve diğer hizmet ihlallerini engellemek için ekonomik bir önlem olarak tanımlanmaktadır. Burada istekte bulunan servis ve/veya kişilerden bazı hesaplamaya dayalı işlemlerin yapılması istenir. Bu yöntemin en önemli özelliği asimetri olmasıdır, istekte bulunan taraf için orta derecede zor ama hesaplanması mümkün olmasına karşın, servis sağlayıcının işlemi kontrol etmesi oldukça kolaydır. Bu protokol, aynı zamanda bir CPU maliyet fonksiyonu, istemci bulmaca çözümü, hesaplamalı bir bulmaca veya CPU fiyatlandırma işlevi olarak tanımlanabilir. Emek Kanıtı, bilgisayarlar tarafından çözülmesi beklendiğinden insanlar tarafından çözülmesi beklenen güvenlik kodlarından (captcha) farklılık gösterir. İki türlü Emek ispatı protokolü bulunmaktadır; sorgu-cevap protokolü ve çözüm-doğrulama protokolü.

Sorgu-Cevap Protokolü

Sorgu-cevap protokolleri, istekte bulunan (istemci) ve sağlayıcı (sunucu) arasında doğrudan bir etkileşimli bağlantı olduğunu varsayar. Sağlayıcı önceden tanımlanmış bir küme ile ilgili rasgele bir sorgu oluşturur, istemci ise ilgili doğru yanıtı sağlayıcıya geri gönderir ve yanıt kontrol edilir (Bkz. *Şekil 2.Sorgu-Cevap Protokolü*).



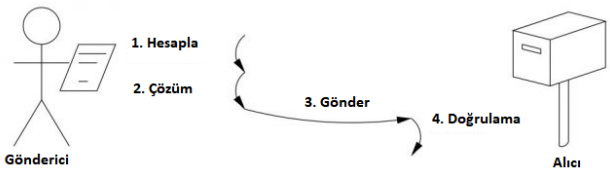
Şekil 2.Sorgu-Cevap Protokolü

Sorgular sağlayıcı tarafından oluşturulduğundan sorgunun zorluk derecesini duruma göre ayarlayabilir. Sorgu-cevap protokolünün sağlayıcı tarafında bilinen bir çözüme sahip olması ya da sınırlı bir arama alanı içinde var olması halinde istemcideki hesaplama alanı sınırlandırılabilir.

Çözüm-Doğrulama Protokolü

Bu protokolde istemci ve sağlayıcı arasında herhangi bir direkt bağlantıya ihtiyaç duymamaktadır. Talepte bulunan kişi tarafından çözüm aranmadan önce problem kendiliğinden yüklenir ve sağlayıcı hem problem seçimini hem de bulunan çözümü kontrol eder (Bkz. *Şekil 3.Çözüm Doğrulama Protokolü*). Bu tür protokollerin çoğu, 1997'de Adam Back tarafından geliştirilen, iş miktarını seçilebilir kılan ve işin kanıtını etkili bir şekilde doğrulayabilen Hashcash gibi sınırsız olasılıklı yineleme algoritmasına sahiptir (Back, 2018). Hashcash, özet fonksiyonlarını kullanarak e-posta kullanıcıları için istenmeyen e-posta mekanizmasını sağlamaktadır. Hashcash'te bir kullanıcı e|-posta göndermek istediğinde e-posta sunucusu daha önceden hesaplamış olduğu ve özet değerini bildiği bir kriptografik özet fonksiyonunu kullanıcıya hesaplattırır. Normal bir kullanıcı için bu hesaplama işlemi ortalama birkaç saniye kadar sürmektedir ve bu süre kullanıcı için problem teşkil etmemektedir. Fakat bu işlem her bir gönderi için yapılması gerektiğinden kötü niyetli bir kullanıcı 1 milyon tane istenmeyen e-posta yollamak istediğinde 1 milyon saniyeden fazla beklemek durumunda kalır. Böylece, kriptografik özet fonksiyonlarının

belli bir emek kanıtı olarak kullanılmasının güzel bir örneği olarak gösterilir.



Şekil 3.Çözüm Doğrulama Protokolü

Emek kanıtında kullanılan işlevlerin hesaplama süresi CPU, bellek ve/veya ağ iletişimi tarafından sınırlanabilir. CPU tarafından sınırlandırılmış fonksiyonların hızı düşük maliyetli taşınabilir cihazlarda çok yavaş olurken yüksek uçlu sunucularda çok hızlı olabilmektedir. Öte yandan bellek tarafından sınırlandırılan fonksiyonların hızı donanımların değişiminden çok fazla etkilenmemektedir (Abadi vd., 2005; Dwork vd., 2013). Dahası, ağ iletişimi sınırlamalı işlevlerde, istemcinin birkaç hesaplama yapması gerekiyorsa, son hizmet sağlayıcısını sorgulamadan önce uzak sunuculardan bazı jetonları alması gerekir. Jeton alma işlevi istemci ile sağlayıcı arasında belli bir gecikmeler olmasını sağlar (Abliz ve Znati, 2009).

Blokzincir teknolojilerinde çözüm-doğrulama protokolleri kullanılmaktadır ve ilk defa Bitcoin tarafından kullanılmaya başlandı. Bir sonraki bölümde Bitcoin'de Emek kanıtının nasıl kullanıldığı anlatılacaktır.

Bitcoin'de Emek Kanıtı

Literatürde Emek kanıtı birçok farklı uygulamalarda kullanılmakta olup her bir uygulamada farklı fonksiyonlar kullanılmaktadır. Kripto para madenciliğinde emek kanıtı ilk defa Bitcoin (BTC) üretiminde uzlaşma protokolünde kullanıldı ve bu protokolde SHA256 özet algoritması emek kanıtı için kullanılmaktadır (Nakamoto, 2009). Bitcoin ’de bir blok yayınlanmadan önce o bloğun güvenliğini belirleyen ve bir önceki blokların güvenliğini arttıran kriptografik özet fonksiyonu kullanılarak bir bulmaca tanımlanır ve bu bulmacayı çözen ilk düğüm bu bloğu yayınlama hakkına sahip olmaktadır. Bu bulmacanın çözümünde yoğun miktarda enerji harcanmakta olup ve bu çözümün kendisi 'kanıt/ispat' olarak tanımlanmaktadır. Emek kanıtı protokolünde, hesaplamaya dayalı zor çözülen bulmacalar belirlenirken, bulunan çözümlerin ise geçerli olup olmadığını kontrol etmeyi kolay olacak şekilde tasarlanmaktadır. Bu durum, diğer madenci düğümlerinin önerilen tüm sonraki blokları kolaylıkla doğrulamasını ve bulmaca çözümü doğrulanamayan önerilmiş blokların reddedilmesini sağlar. Ortak bir bulmaca yönteminde, bloğun özetinin belirli bir değerden daha düşük olmasını gerektirir. Belirlenen değer bulmacaların zorluk derecesini belirlemektedir. Madenci düğümler, gereksinimi karşılayan bir blok özeti bulmaya çalışırken, bloğa eklenen tek kullanımlık sayılar (*teksa*) üzerinde çok küçük değişiklikler yaparlar. Her girişim için madenci düğümü, yoğun hesaplama gerektiren bir işlem olan tüm blok başlığı için özeti hesaplar. Gerekli değer, blokların ne sıklıkta yayınlandığını etkileme zorluğunu ayarlamak için zaman içinde değiştirilebilir. Örneğin, Bitcoin'de, iki haftada bir yeni önerilen bulmacaların zorluk derecesi otomatik olarak ayarlanır ve blok yayın oranını her on dakikada bir kez olması sağlanır.

Bu protokolün en önemli özelliklerden biri, bulmaca çözümünde yapılan geçmiş işlerin, gelecekteki bulmacaları çözme olasılığını etkilememesidir. Diğer bir deyişle, aday bloğu bin veya bir milyon kez farklı teksa değerleri ile özet almak, mevcut bulmaca çözme olasılığını artırır, fakat kullanıcının gelecekteki bulmacalar için herhangi bir çözüm bulma şansını arttırmaz. Bu nedenle, bir blok için tanımlanan çözülmemiş her bulmaca birbirinden bağımsızdır ve çözümü benzer miktarda iş gerektirir. Bu, bir kullanıcı tamamlanmış bir bloğu başka bir kullanıcıdan aldığında, yeni blok eklenmesi konusunda teşvik edilir; çünkü diğer blokların ekleneceğini bilirler ve bu bloktan devam ederler. Yeni bloğu kabul etmeyi reddetmeleri durumunda daha kısa bir blok zinciri inşa edeceklerdir ve bu da istenmeyen durum olacaktır çünkü protokolde varsayılan

olarak en uzun geçerli zincirle devam edileceği varsayımıdır.

Bu modeli daha iyi anlamak için örnek olarak, SHA-256 algoritmasını kullanarak bir bilgisayarın aşağıdaki hedef ölçütlerini karşılayan bir özet değer bulması gereken bir bulmaca düşünelim:

- SHA256 ('blokzincir' + teksa) = '0000000' ile başlayan özet değer bulmacası (Yedi tane sıfır ile başlayan özet değeri).

Bu örnekte, 'blokzincir' metin dizesinin sonuna bir teksa değeri eklenir ve daha sonra özet değeri hesaplanır. Kullanılan teksa değerleri yalnızca sayısal değerlerdir. Bu çözülmesi nispeten kolay bir bulmaca olup bazı örnek çıktıları şöyledir:

- SHA256('blokzincir0') =0x11dda4b2f8409049aedb06247f0423f45194bdad1c23308b69565e49a671ebab (çözülmedi)
- SHA256('blokzincir1') =0xb2efee8229b6fabd4dad44351ec327c6a3295b19e47e1a239ec381919fcf90ea (çözülmedi)
- …
- SHA256('blokzincir33758449') =0x**0000000**67d668d96d3415bcd2909fed7242cf75438596b38f675e8a926a5e30c (çözüm!)

Bu bulmacayı çözmek için 33.758.449 tane farklı teksa değerlerinin denenmesi ile bulundu. Bu işlemler için I7 işlemcili Windows 10'da Java Kütüphanesi kullanılarak tek bir iş parçacığı ile optimize edilmeden 25 saniye içinde tamamlanırken *teksa'lar* 0'dan başlayıp bir defada bir değer test edildi. Bununla birlikte, her ek 'önde gelen sıfır' değeri bulmacanın zorluğu arttırır. Hedefi bir ön ek sıfır ('**00000000**', sekiz tane sıfır ile başlayan özet) artırarak aynı donanımla, bulmacayı çözmek için 9.264.471.446 farklı tahmin gerektirdi (1 saat 53 dakika, 13 saniye içinde tamamlandı):

- SHA256('blokzincir9264471446') =0x**00000000**8252af91fb84d5c0cbb5103e169892c8b56334bfa4666bb6ff49b17e

Bu işlemlerin çözümünde kısa yol bulunmamaktadır. Madenci düğümler, hedef değer için doğru olan özet değeri bulmak için yoğun bir hesaplama yapmak ve bunun için de zaman ve kaynakları harcamak zorundadır. Bir kullanıcı bulmaca çözme işini yaptıktan sonra, ağdaki diğer düğümlere geçerli bir not ile aday bloğunu gönderir. Alıcı düğümler, bu işin düzgün yapıldığını doğruladıktan sonra aday bloğu kendi zincirine ekleyip bloğu diğer düğümlere gönderirler. Bu şekilde, yeni aday bloklar ağda bulunan tüm düğümlere hızla dağıtılır. Bulmacaların çözümünde bulunan teksa değerlerinin doğrulanması kolaydır, çünkü bulmacanın çözülüp çözülmediğini kontrol etmek için yalnızca tek bir özet alınması yeterlidir.

Havuz Yöntemi

Emek kanıtı uzlaşma protokolü, sistem kullanıcıları arasında çok az güvenin olduğu veya birbirlerine tamamen güvenmediği durumlar için tasarlanmıştır. Madenci düğümlerin, bulmacaları çözerek ve eklenen bloklardaki işlemleri de kontrol ederek sistem üzerinde beklenmeyen değişiklik yapılmasını önler. Yazılımların ve donanımların sürekli gelişmesi bulmacaların daha verimli bir şekilde çözülmesini sağlamaktadır. Buna ek olarak ağdaki kullanıcıların sayısının artması bulmacaların çözme hızlarını daha da arttırmaktadır. Fakat sistemde bazı kötü niyetli kullanıcıların bir araya gelerek sisteme zarar vermesini engellemek için bulmacaların zorluk dereceleri zamanla arttırılmaktadır.

Bitcoin ve benzeri kripto para madenciliğinde tanımlanan blok üretme zorluk derecesinin artması nedeniyle herhangi bir bilgisayar için bir bulmacayı çözmek daha da zorlaşmaktadır. Bu nedenle, madenci düğümleri kendilerini 'havuz' ya da 'kolektif' olarak örgütleyerek bulmacaları topluca çözmektedirler. Bunun nedeni, iş yükü ve ödülleri paylaşmaktır. Bunun için çalışmayı iki veya daha fazla düğüm arasında bir kolektif arasında dağıtarak havuz oluşturulur.

Havuz sisteminin daha iyi anlamak 4 madenci düğümden oluşan bir havuzu deney örneği şöyle gerçekleştirildi. Hedef bulmaca 8 tane sıfır ile başlayan bir özet değeri bulmak olarak tanımlansın. Toplamda 40.000.000.000 tane farklı teksa denenmesi amaçlandı. Sınanması gereken değerler kümesi dörde bölündü ve her

düğüm kendi aralığındaki değerleri dikkate aldı. Şöyle ki:

1. Düğüm: 0000000000'nolu *teksa* değerinden 10000000000'e kadar kontrol eder.
2. Düğüm: 10000000001'nolu *teksa* değerinden 20000000000'e kadar kontrol eder.
3. Düğüm: 20000000001'nolu *teksa* değerinden 30000000000'e kadar kontrol eder.
4. Düğüm: 30000000001'nolu *teksa* değerinden 40000000000'ye kadar kontrol eder.

Bu bulmacanın çözümlerinden birini 3. düğüm şöyle hesapladı:

- SHA256('blokzincir30458017781') =0x**00000000**d4fa2df066b145aeda4e7a975e0049cba2a18f683c51d83f60f74db9

Bu çözümde toplamda 1.835.009.602 farklı teksa tahminleri kullanılarak hesaplandı (toplam 8 dakika ve 5 saniye içinde tamamlandı). İşi çok daha fazla makine arasında bölmek, emek modelinin ispatında daha tutarlı ödüllerin yanı sıra daha iyi sonuçlar vermesini sağladığı sonucuna varıldı.

Emek Kanıtı Dezavantajları

Emek kanıtı uzlaşma protokolüne yapılan en büyük eleştiri doğal enerji kaynaklarının israfıdır. Bu modeli kullanan ağlar küçük iken önemli ve tartışılan bir konu değildi. Fakat, Bitcoin ve benzeri kripto para madenciliklerin ağı son yıllarda katlanarak büyümektedir. Bu büyüme bulmacaların zorluk derecelerinin arttırılmasına sebebiyet vermektedir ve bulmacaların çözümünde çok fazla işlemlerin yapılması ve bu da çok fazla enerji kullanımını gerektirmektedir. Bu enerji ihtiyacı dikkate alınacak kadar hayati öneme sahiptir. Örneğin, hâlihazırda Bitcoin ‘de blok üretmek için İrlanda'nın ihtiyacı olan elektrikten daha fazla elektriği kullanmaktadır ve 2020 yılına kadar Danimarka'nın ihtiyacı olduğu elektrik miktarını tüketeceği düşünülmektedir. Emek kanıtı modelinin güvenliği, hiçbir kuruluşun/düğümün işlem gücünün %50'sinden fazlasını toplayamaması ilkesine dayanmaktadır, çünkü böyle bir durumda, gücü elinde tutan kuruluş en uzun zinciri sürdürerek sistemi etkin bir şekilde kontrol edebilir. Aynı zamanda, bu gücü elde eden bir saldırgan:

- Çift harcama saldırısını gerçekleştirebilir. Aynı harcamayı iki farklı bloklarda gerçekleştirir ve istediği blokların onaylanmasını ve istemediği blokların ret edilmesini sağlamasından ötürü bu saldırıyı gerçekleştirmesi kolay olmaktadır.
- Hizmet engelleme saldırısını gerçekleştirebilirler.

Emek kanıtı uzlaşma modeline yapılacak son eleştiri madencilere verilen ödüllerin miktarı/değeri düştüğünde madenci sayıları azalacak olmasıdır. Bu eksi yönlü değişim ağın hesaplama gücünü azaltacaktır ve bu da kullanıcıların sisteme olan güvenini sarsacaktır.

Hisse Kanıtı (Proof of Stake) Uzlaşma Protokolü

Hisse kanıtı kavramı ilk defa Bitcoin'de kullanılmak üzere parayı elde tutma süresi (coin age) olarak 2011 yılının başlarında bitcointalk.org'ta ortaya atıldı ve bu özellik gerçekleşecek işlemlerin önceliklerini belirlemek için kullanıldı. Sunny King ve Scott Nadal 2012 yılında yayımlanan bir makalede hisse kanıtının kripto para madenciliğinde uzlaşma modeli olarak kullanılabileceğini resmi olarak gösterdiler (King ve Nadal, 2012). Bu modelde, cüzdanında para bulunduran her kullanıcı aynı zamanda doğrulayıcı olabilmektedir. Bu modelin ilk sürümü emek kanıtı modeli ile birlikte düşünülmüştür, yani madenciliğin ilk başlarında emek kanıtının kullanılması ve zamanla sisteme olan etkisinin kaldırılması öngörüldü. Böylece emek kanıtında gereken yüksek enerji tüketimi sorununu kaldırılmış olacaktır.

Emek kanıtında harcanan enerjiye dayalı güven yerine, Sunny ve Scott, düğümlerin sahip olduğu hisse miktarlarına (kripto para miktarı) göre bir sonraki bloğu yayınlayacak düğümü seçecek algoritmayı önerdiler. Diğer bir deyişle, bir

düğümün sistemde (cüzdanda) ne kadar çok hissesi (kripto parası) var ise bir sonraki bloğu yayınlama hakkını o kadar çok kazanma olasılığına sahip olmaktadır. Ayrıca, hissesi çok olan bir düğüm, bir o kadar daha az ihtimal ile sistemin çalışmasını bozmak isteyecektir. Hisse ispatını kullanan uygulamalardaki en önemli temel yapı taşlardan biri işlemlerde kullanılan para yaşı kavramıdır.

Para Yaşı ve Parayı Elde Tutma

Paranın yaşı ölçütü, esasen kripto paraya sahip olan kişinin elindeki parayı harcamadan veya hareket ettirmeden elinde tutma süresi olarak tanımlanır. Herhangi bir işleme tabi tutulan para miktarının yaşı transfer işlemin başarı ile tamamlanmasından sonra sıfırlanır. Örneğin, paranın yaşının doğrusal olarak arttığı düşünülürse 200 gün boyunca 365 adet kripto parayı elde tutan bir kişi 7.300 "para günü" veya yaklaşık 200 "para yıl" biriktirdiği düşünülür. Öte yandan paranın yaşının doğrusal değil de farklı bir fonksiyona kullanılarak ta değişebilir. Paranın yaşının aktif olarak kullanılabilmesi amacı ile tüm işlemlerde zaman damgası kullanılmaktadır. Blokların yayınlanma zamanı ile işlemlerin zaman damgası paranın yaşının güvenli bir şekilde hesaplanmasında yardımcı olmaktadır.

Daha fazla para yaşına sahip düğümlerin bir bloğu imzalama şansı o kadar oranda fazla olur. Dolaşımdaki toplam paraların %1'ine sahipseniz ve bunu elinizde bulundurursanız, herhangi bir bloğu sizin imzanız ile yayınlanma olasılığı %1 olacaktır. Diğer bir deyişle, her 100 bloktan 1'ini imzalamayı bekleyebilirsiniz. Bir bloğu imzalama şansını kazandığınızda, cüzdanınız otomatik olarak bir "staking" (hisse tutma) işlemini tamamlar. Bu bazen "minting" (para basma) olarak da adlandırılır. Cüzdanınız kazandığında, para yaşınızın bir kısmını veya tamamını tüketir ve karşılığında bir miktar ödül alırsınız. Örneğin, bir kripto paranın %5 yıllık enflasyona sahip olduğunu ve 10 yıllık para yaşını (yukarıdaki örnekte olduğu gibi) tükettiğinizi bildirirseniz, size ödül olarak 5 adet kripto para verilir (100 yıl * 0.05 para/yıl = 5. para). Bu işlemden sonra elinizdeki paraların yaşı sıfırlanır ve paranız tekrardan yaşlanmaya başlar.

Sıfır Hisse Problemi

Bazı düğümlerin birden blok oluşturup ve ayrı işlem ücreti talep etmesi ve imzalayan düğümlerin de tüm blokları imzalama problemidir. Emek ispatında madencilik maliyetli olurken, hisse ispatında işlemler çok ucuz olması bu olayın gerçekleşmesini kolaylaştırmaktadır. Diğer bir deyişle çatallaşma durumlarında imzalayan düğümlerin hangi çatalın kazanmasına bakmaksızın her iki çataldaki blokları imzalamasıdır. Bu durumda uzlaşma protokolü beklenildiği gibi çalışmamaktadır ve öyle ki bu çatallaşmalarda çift harcama problemi gerçekleşme olasılığı çok büyüktür. Hisse kanıtı uzlaşma modelinin en önemli sorunu olarak tanımlanmaktadır. Bu probleme karşı farklı çözümler üretilmiştir. Örneğin, Casper hisse kanıtı uzlaşma modelinde aynı anda iki bloğu imzalayanları cezalandırmaktadır. Hisse ispatı modelinde kullanılabilecek farklı görecelerin var olması, blokzincirlerin farklı amaçlar için oluşturulması ve farklı problemlere odaklanmasından ötürü çeşitli uzlaşma modelleri ortaya çıkmıştır.

Temsili Hisse Kanıtı (Delegated Proof Of Stake)

Daniel Larimer tarafından geliştirilen temsili hisse kanıtı var olan hisse kanıtı yöntemine göre daha adil, iki katmanlı demokratik ve esnek bir uzlaşma yaklaşımı sergilemektedir ve bu yaklaşım ile işlemler daha verimli ve hızlı yapılmaktadır (Larimer, 2014). Bu modelde, itibar koruma sistemi ve gerçek zamanlı oylama kullanılır ve iki katmanlı temsili bir demokrasi gerçekleşir. Daha açık olmak gerekirse, güvenilir kripto parası olan kullanıcılardan oluşan bir tanık paneli, oylama yöntemi ile oluşturulur. Bu panelin kontrolünde bloklar oluşturulur ve panele güvenilir olmayan kişilerin engellenmesi sağlanır. Panelde bulunan tanık heyeti blokların oluşturulmasından sorumludurlar ve bu işlemlerden ötürü ödüllendirilmektedir. Öte yandan, blokların içerisinde yer alan işlemleri değiştiremezler. Fakat bloklara uygun olmayan kötücül işlemlerin eklenmesini engellerler. Bu işlev kötü niyetli tanık heyetinde bulunan kullanıcılara yanlış görünen bir güç veriyor gibi görünebilir. Fakat kötü niyetli tanıkların

davranışları zamanla halka açık hale gelmesi muhtemeldir. Öte yandan, bu modelde yer alan tüm üyeler topluca herhangi bir vekilin temsil heyetinden çıkarılmasını önerebilirler ve oylayabilirler. Bu sebeple, tanık heyetinde bulunan/seçilen kullanıcılar sahip oldukları itibarlarını kaybetmemek uğruna işlemleri doğru ve gerçek zamanlı yaparlar.

Bu modelde ayrıca sistem ve ağ parametrelerine karar verecek bir temsil heyeti, tanık heyeti oluşturulmasına benzer bir demokratik yöntemle oluşturulur. Bu heyettekiler işlemlerin/blokların boyutu ve blok oluşturulma ödül ücretlerine karar verirler. Bu parametreler sistemde çok sık değişmemektedir. Bu heyette yer alan kullanıcılara herhangi bir ödeme yapılmamaktadır. Bu uzlaşma modelini kullanan popüler kripto para olarak BitShares, Steem, EOS, List ve Ark'ı örnek olarak gösterilir.

Kiralık Hisse İspatı (Leased Proof of Stake)

Geleneksel hisse ispatı uzlaşma modelinde az sayıda hisseye sahip olanlar küçük bir ihtimalle blok oluştururlar ve bu ihtimalin gerçekleşmesi uzun bir süre gerektirir. Bu problem emek kanıtı uzlaşma modelinde düşük işlem gücüne sahip olanlarda da bulunmaktadır. Çok düşük bir ihtimalle şanslı bir kullanıcı yıllar sonra blok oluşturmaya hak kazanabilir. Bu durum, çok az sayıda hisse oranına sahip kullanıcıların düğüm oluşturmayacağı ve sistemi genellikle büyük hisseye sahip kullanıcıların yöneteceği anlamına gelmektedir. Fakat sistemi ne kadar fazla katılımcı ile yönetilirse sistem o kadar merkezi olmayan güvenilir bir duruma gelecektir. Bu durumu sağlamak için az sayıda hisseye sahip kişilere teşvik verilmesi önem arz etmektedir.

Kiralık hisse ispatı uzlaşma modeli ile bu kişilerin hisselerini düğümlere kiralamalarını sağlamaktadır (Platform Waves, 2018). Kiralanan tüm hisselerin kontrolü sahibinin kontrolünde olup istediği zaman harcayabilir veya başka bir yere gönderebilir. Kiralanan hisseler düğümlerin sistem içerisindeki ağırlıklarını arttırır ve bu da kiralayan düğümlerin blok oluşturma olasılıklarını arttırır. Blok oluşturmada elde edilen ödül kiralanan hisse oranında paylaşılır. Bu uzlaşma modeli Waves kripto para madenciliği tarafından gerçekleştirilmektedir. Ağın güvenliği cüzdanında en az 10.000 WAVE bulunduran aktif düğümler tarafından sağlanmaktadır. Cüzdanında en az bu miktar kadar para bulunduran herkes düğüm olabilmektedir ve başkalarının hisselerini kiralayabilmektedir.

Önemin Kanıtı (Proof of Importance)

NEM tarafından geliştirilen bu modelde her bir hesap için bir güven derecelendirme notu verilmektedir. Önemli olarak kabul edilen kullanıcılar blok oluşturabilir ve karşılığında ödül almaktadırlar. Bir NEM kullanıcısının önemi, sahip oldukları hissenin sayısı ve cüzdanlarına yapılan işlem sayısı ile ölçülmektedir. Bu yöntemle herkese benzer fırsatlar sunulmaktadır ve temel ise amaç düzenli bir şekilde işlem yapan kullanıcıları teşvik etmektir. Hisse kanıtına dayalı sistemlerde, bir kişinin bir blok oluşturmak için çok sayıda hisseye sahip olması gerekirken NEM işlemlerinde hacim ve güven faktörleri önem arz etmektedir. Böylece, NEM kullanıcıları sadece XEM'lerini cüzdanlarında tutmayacak, ayrıca aktif olarak işlem hacimlerini gerçekleştirecektir (Beikverdi, Cointelegraph, 2015) (Beikverdi, Cointelegraph, 2015).

NEM'de blok oluşturmaya ve karşılığında verilen ödüle ekin biçme/hasat denilmektedir. Önemin kanıtı modeli hangi bloğun geçerli olduğunu belirler. NEM'de kullanıcılar, her 1440 blokta bir cüzdanlarındaki bakiyenin $\frac{1}{10}$ oranında XEM miktarı üzerinde hak elde edilmiş sayılırlar. Herhangi bir kullanıcının önem derecesinin hesaplanabilmesi ve ekin hasat edebilmesi için cüzdanında en az 10.000 tane XEM üzerinde hak elde edilmiş olması lazım. XEM'de blok üretmede yeni paralar üretilmemektedir, bloklardaki işlem ücretleri blok üreticisine verilmektedir.

Ouroboros Hisse Kanıtı Protokolü

Cardano için geliştirilen Ouroboros hisse kanıtı modelinde zaman çağlara ve her bir çağ n tane zaman dilimine (örnek. 20sn) bölünmüştür (Kiayias vd., 2016). Her bir zaman diliminde bir tane blok oluşturur. Her zaman aralığının bir lideri vardır. Lider belirli bir orandan fazla

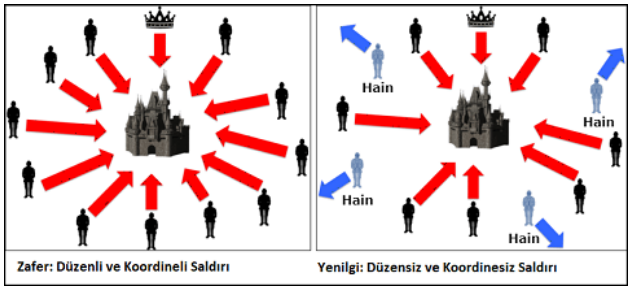
hisseye sahip kişilerden seçilir. Birden fazla zaman aralığı için aynı lider seçilebilir yani Hissesi çok olan daha fazla liderlik yapacaktır. Bu arada, her lider kendi zaman dilimlerinde çevrimiçi olmak zorundadır. Aksi durumda o blok boş geçecek üretilemeyecektir.
Lider seçiminin tarafsız ve bağımsız olması için seçmenler arasında Güvenli Çoklu Hesaplamalar ile Kamu tarafından doğrulanabilir Gizli paylaşım protokolleri koşturulur (Blum, 1983) (Feldman, 1987). Protokolde, her seçmen rasgele bir madeni para atması (yazı-tura gibi) beklenir ve bu protokoller aracılığı ile final bir çekirdek değer elde edilir ve bu değerle zaman dilimlerinin liderleri belirlenir.

Casper Hisse Kanıtı Protokolü
Vitalik Buterin ve Virgil Griffith tarafından 2017'de Ethereum için geliştirilen ve yakın zamanda uygulanacak olan bir uzlaşma modelidir (Buterin ve Griffith, 2017). Bu model, aslında Hisse kanıtı ile birazdan anlatacağım Bizans hata toleransı uzlaşma modelinin birleşmesinden oluşmaktadır. Hali hazırda yürütülen emek ispatı ile birlikte yürütülecektir ve zamanla emek ispatının etkisi azaltılarak tamamen ortadan kaldırılacaktır. Blokların oluşturulması ve onaylanmasını doğrulayıcılar tarafından gerçekleştirilir. Bir miktar ETH depozit olarak Casper akıllı sözleşmelerine yatırmaları gerekir. Depozitle, kötü doğrulayıcılar cezalandırılır. Bununla Sıfır Hisse Problemi çözülmüş olur. İki fazda oylama ile bloklar hazırlanır ve yayınlanır. Her iki fazda da en az 2/3 oranında onay gerekir.
Bu protokolün birkaç güzel özelliği vardır:

- Herkes düğüm olabilir.
- Uzlaşmaya hızlıca varılır.
- Geri dönüşü olmayan bir uzlaşma üzerinde anlaşılır
- Herhangi bir politika güdülmez.

Bizans Hata Toleransı Modeli

Dağıtık hesaplama sistemlerinde kullanılan Bizans Hata Toleransı (BHT), iki general probleminin genelleştirilmiş hali olan Bizans generalleri probleminin başarısızlık durumlarını belirli bir düzeye kadar tolere eden sistemin bir özelliğidir. Bu problemin çözülemeyen bir ispatı bulunmaktadır. Bizans generalleri hikâyesi şöyle özetlenmiştir (Bkz. Şekil 4. Bizans Generalleri Problemi).



Şekil 4. Bizans Generalleri Problemi

- Bizans ordusunun birkaç bölüğünün düşman kentinin dışında kamp yaptığını ve her bir bölüğün kendi generalinin komutasında olduğunu düşünelim. Generaller, yalnızca haberci tarafından birbirleriyle iletişim kurabilirler. Düşmanı gözlemledikten sonra ortak bir eylem planı üzerinde karar vermeleri gerekir. Bununla birlikte, generallerden bazıları, sadık generallerin anlaşmaya varmalarını önlemeye çalışan hainler olabilir. Generaller şehre ne zaman saldırı gerektiğine karar vermeli fakat aynı zamanda hücum etmek için ordularının güçlü bir çoğunluğuna ihtiyaç duyuyorlar. Generallerin, (a) tüm sadık generallerin aynı eylem planına karar verdiklerini ve (b) az sayıda haince sadık generallerin kötü bir plan hazırlamasına izin vermeyecek bir algoritma olmalıdır. Sadık generaller, algoritmanın gerektiğini söylediklerini yapacak, ancak hainler istedikleri her şeyi yapabilecek. Algoritma, hainlerin ne yaptıklarına bakılmaksızın (a) koşulunu garanti altına almalıdır. Sadık generaller sadece anlaşmaya varmakla kalmamalı, makul bir plan üzerinde anlaşmalıdır.

Hikâyede yer alan 'generaller', söz konusu blokzincirin üyeleridir. Birbirlerine gönderdikleri haberciler, ağ üzerinden gönderilen mesajlardır. "Sadık generallerin" ortak amacı, blokzincire gönderilen bir bilginin geçerli olup olmadığını kabul etmeye karar vermektir. Geçerli bir bilgi parçası, hikâyede saldırı lehine karar vermek için doğru bir fırsat olacaktır. Sadık generaller, blokzincirin

bütünlüğünü sağlamak ile ilgilenen ve yalnızca doğru bilginin kabul edilmesini sağlamak için sadık blokzinciri katılımcılarıdır. Diğer taraftan hain generaller, blokzincirdeki bilgileri tahrif etmek isteyen kötü niyetli uçlardır.

Blokzincir sistemlerinde BHT problemi için tek veya resmi bir çözüm bulunmamaktadır. Aslında emek kanıtı yeterince büyük bir katılımcıya sahip olursa etkili bir şekilde BHT'yi sağlar. Yeterince katılımcı olması ve zamanla zorluk derecesinin artmasıyla, örneğin Bitcoin'de eski bir bloğu değiştirmek neredeyse imkânsız hale gelmiştir. Emek kanıtında saldırgan düğüm, saldırı esnasında çatallaşmaya neden olacaktır ve saldırı esnasında yeterince güçlü bir hesaplamaya (en az %51 güce) sahip olmadığı durumda çatallaşmaya neden olduğu zincirin devamını getiremeyecektir. Tüm düğümler uzun zincirin arkasından gitmeye meyilli oldukları için saldırganın etkisi ortadan kalkacaktır.

Diğer çözümlerde kullanılan BFT algoritmalarında her general(düğüm) bünyesinde bir durum bilgisi tutar. Bir general bir ileti aldığında, hesaplama veya işlem yürütmek için iletiyi kendi durum bilgisiyle birlikte kullanırlar (Lamport vd., 1982). Bu hesaplama, o kişiye söz konusu mesaj hakkında ne düşünülmesi gerektiğini söyler. Daha sonra, yeni mesajla ilgili bireysel kararına ulaştıktan sonra, general bu kararı sistemdeki diğer generallerle paylaşır. Bir uzlaşma kararı, tüm generaller tarafından verilen çoğulcu kararlara dayanarak belirlenir.

Pratik BHT Uzlaşma Modeli

Hyperledger Fabric platformunda kullanılan bu uzlaşma modelinde katılımcıların bazı kesimlerinin kötü niyetli davranışlarına rağmen fikir birliği sağlanmaktadır (Castro ve Liskov, 1999). Hyperledger Farbric'te kullanıcılar önceden kayıtlı olduğundan bu model izinli blokzincirlerde kullanılmaktadır. Bu modelde, her doğrulayıcı düğümde bir adet açık ve özel anahtar çifti bulunmaktadır. Her düğüm diğer doğrulayıcı düğümlerin açık anahtar bilgisine sahiptir. Blokzincire eklenmek amacı ile yeni oluşturulan bloğun kriptografik özet değeri hesaplanır ve ağdaki diğer düğümlere gönderilir ve gönderen düğüm gelen yanıtları saymaya başlar. Doğrulama yapanların sayısının en az 2/3'ünde de aynı özet değeri görüldüğünde oluşturulan blok muhasebe defterine kopyalanır. Bir işlem belirli bir sayıda düğüm tarafından onaylanmış ise uzlaşma sağlanmış olup yapılan işlem geçerli hale gelmiş demektir. Tüm katılımcılar merkezi bir yapının onayı ile sisteme dâhil olması nedeniyle bu model dağıtık izinsiz blokzincir sistemlerinde kullanılmamaktadır. Bu sebeple, izinli yapılar içinde kullanılmaktadır. Bu modelin en büyük eksisi, $n > 20$ olunca düğümler arası mesajlaşma üstel olarak artmaktadır.

Temsili BHT Uzlaşma Modeli

Castro ve Liskov tarafından önerilen PBHT uzlaşma modelinin temsili demokrasi ile geliştirilmiş sürümü olup NEO platformunda uygulanmaktadır. Bu modelde, düğümler iki ayrılmaktadır; defter tutucular ve sıradan düğümler. Defter tutucular, tüm ağ için muhasebe hizmeti sağlar ve muhasebeyi sürdürür. Sıradan düğümler ise transferleri sağlar, para değişimi yapabilirler ve defter tutuculardan gelen verileri kabul ederler. NEO hissesine sahip olanlar kimlerin defter tutucu olacağına oylama yöntemi ile karar verirler ve defter tutucuların kimler olduğu kamuya açık hale getirilir. Modelin şu varsayımlara yer verilmiştir; mesajlar kaybolabilir, değişebilir, gecikerek ulaşabilir veya tekrar gönderilebilir. Düğümlerin davranışları sürekli değişebilir, ağa katılabilir veya ağdan kopabilirler, doğrulama yapabildikleri gibi yanılsayabilirler. Gönderilen verilerin bütünlüğü ve kimlik doğrulaması kriptografik işlemler ile yapılmaktadır, gönderici göndereceği verinin özet değerini alır imzaladıktan sonra gönderir.

Uzlaşma algoritmasında sıradan düğümler uzlaşmaya katılmazken, sistemin toplam defteri defter tutucular tarafından yönetilmektedir. Tüm uzlaşma düğümleri, mevcut uzlaşma durumunu tutmak için bir durum tablosunu kaydetmek zorundadırlar. Blokzincirin oluşumundan son haline kadar bir uzlaşma için kullanılan veri kümesine bir *Görünüm* adı verilir. Mevcut *Görünüm* içerisinde fikir birliğine ulaşılamıyorsa, bir değişikliğin gidilecektir.

Tablo 1. Uzlaşma Modellerinin Sınıflandırılması

Uzlaşma Modeli	Türü	İşlem Kesinliği	İşlem Hızı	Düğüm Ağının Büyümesi	Düğüme Güven
Emek İspatı	İzinsiz	İhtimale Bağlı	Yavaş	Yüksek	Yok
Temsili Hisse Kanıtı	İzinli/ İzinsiz	İhtimale Bağlı	Hızlı	Yüksek	Yok
Kiralık Hisse Kanıtı	İzinli/ İzinsiz	İhtimale Bağlı	Hızlı	Yüksek	Yok
Önemin Hisse Kanıtı	İzinli/ İzinsiz	İhtimale Bağlı	Hızlı	Yüksek	Yok
Casper Hisse Kanıtı	İzinli/ İzinsiz	İhtimale Bağlı	Hızlı	Yüksek	Yok
Ouroboros Hisse Kanıtı	İzinli/ İzinsiz	İhtimale Bağlı	Hızlı	Yüksek	Yok
PBHT	İzinli	Kesin	Hızlı	Düşük	Yarı-Güven
TBHT	İzinli/ İzinsiz	Kesin	Hızlı	Yüksek	Yarı-Güven
FEDERE BHT	İzinli/ İzinsiz	Kesin	Hızlı	Yüksek	Yarı-Güven

Her bir *Görünümü*, 0'dan başlayarak v ile bir sayı ile tanımlanır ve fikir birliğine varıncaya kadar arttırılabilir.

Her bir uzlaşma düğümü 0'dan başlayarak bir sayı ile tanımlanır, son düğüm $n-1$ olarak numaralandırılır. Uzlaşma gerçekleşmesinin her turu için, bir düğüm kongre konuşmacısı olurken diğer düğümler kongre dinleyicileri rolünü oynamaktadır. Konuşmacı seçimi (seçilen kişinin numarası p olsun) şöyle yapılmaktadır. Mevcut bloğun boyutu h olsun, $p = mod\ (h - v)\ n$ olarak hesaplanır ve p'nin değeri 0 ile $n-1$ arasında olur. Uzlaşmada en fazla $f = \lfloor \frac{n-1}{2} \rfloor$ tane düğüme tolerans gösterilir, mutabakat düğümlerinden en az $n-f$ imzası ile her uzlaşma turunda yeni bir blok oluşturulacaktır. Bir blok oluşturulduğunda, yeni bir uzlaşma turu başlayacak ve $v = 0$ sıfırlanacaktır.

Federe BHT Uzlaşma Modeli

İlk defa Ripple (Schwartz vd, 2014) tarafından ortaya konan bu modelin çalıştığını Stellar'da (MAZIERES, 2016) formalize edilerek resmileştirildi. Temel olarak bu modelde Her düğüm belirli bir sayıdaki düğümlere güven duymaktadır. Bu model birden fazla turdan oluşmaktadır. İlk turda uzlaşma küçük gruplar arasında gerçekleşir. Sonraki turlarda gruplar arasında kesişmenin olması nedeniyle genel uzlaşma sağlanmış olur. Ripple'de katılımcılar önceden belirlenmişken Stellar'da ağa sonradan katılım yapılabilmektedir. Bu iki protokolü teknik olarak şöyle özetleyebiliriz:

- Ripple protokolü, her düğümün Benzersiz Uç Listesi tanımlamasını gerektirir(UNL). UNL, güvenilen diğer Ripple uçlarını içerir. Ripple ağında uzlaşma UNL'deki diğer düğümlere danışılarak her düğüm tarafından sağlanır. Her UNL'nin Ripple ağındaki diğer uçlarla % 40 örtüşme sağlaması gerekir. Her düğüm, işlemleri "aday kümesi" adı verilen bir veri yapısında toplar ve aday kümesini UNL'deki düğümlere yayınlar. Düğümler işlemlerin geçerliliğini onaylar, oy verir ve oylarını yayınlar. Her düğüm aday kümesindeki en çok oyu alan düğüm işlemleri seçer ve bir sonraki tur için tekrar yayınlar. Bir aday kümesi düğümlerin %80'inden fazlasının çoğunluğunu aldığında aday kümesi geçerli bir blok haline gelir veya Ripple'da bir "ledger" haline gelir. Bu ledger "Last Closed Ledger (LCL)" olarak kabul edilir ve her düğüm tarafından Ripple blok zincirine eklenir.
- Stellar protokolü yeterli çoğunluk (quorum) prensibine dayalıdır. Quorum dilimi, belli bir düğümün uzlaşmada ikna olabilmesi için yeterli olacak quorum'un bir alt kümesidir.
- Tek bir düğüm birden fazla quorum dilimde yer alabilir. Her düğüm işlemlerdeki ilk oylamayı yapar. Bu, ortak oylama sürecinin ilk adımıdır. Her düğüm kendi seçimini gerçekleştirir ve seçimini tersine çeviren

başka bir bildirimde oy kullanmayacaktır. Bununla birlikte, eğer içinde bulunduğu dilim farklı bir seçimi kabul etmişse farklı bir seçimi kabul edebilir. İkinci adım, kabul aşamasıdır. Bir düğüm daha önce ters bir seçim yapmadıysa ve v-bloğundaki diğer uçlar seçimi kabul ediyorsa seçimi kabul eder. V-bloğu mevcut ucun üye olduğu quorum dilimlerinin her birine üye olan uçlardan oluşan bir kümedir. Quorum dilimlere anlaşmaya varmak için böylece birbirlerini etkilemiş olurlar. Quroumdaki bütün üyeler seçimi kabul ettiği zaman onaylama aşaması gerçekleşmiş olur. Düğümler birbirlerine onaylama mesajları göndererek ortak bir son duruma ulaşırlar.

Tartışma ve Sonuç

Yukarıda anlatılan uzlaşma protokollerinin genel özellikleri Tablo 1'de özet olarak verilmiştir. Tabloya baktığımızda, emek ispatı hariç diğer tüm modeller, izinli ve izinsiz blokzincir yapılarında duruma göre kullanılmaktadır. İşlemlerin kesinliği sadece Bizans hata toleransı uzlaşma modellerinde gerçekleşmektedir. Emek ispatında her bir blok oluşturmada belirli bir süre içerisinde madenciler tarafında yüksek miktarda işlemlerin yapılması istenildiğinden transfer işlemleri yavaş gerçekleşirken diğer modellerde bu işlemler çok hızlı yapılmaktadır. Blok oluşturan düğümlere olan güven sadece Bizans hata toleransı modellerde bulunmaktadır.

Kaynakça

Abadi, M., Burrows, M., Manasse, M., Wobber, T. (2005). Moderately hard, memory-bound functions. ACM Transactions on Internet Technology, 5(2), 299-327.

Abliz, M., Znati, T. (2009). A guided tour puzzle for denial of service prevention. Annual Computer Security Applications Conference, ACSAC '09 (s. 279288). Washington, DC, USA: IEEE Computer Society.

Back, A. (2018, Nisan 13). A partial hash collision based postage scheme. http://www.hashcash.org/papers/announce.txt adresinden alındı

Beikverdi, A. (2015, Mart 13). Cointelegraph. Proof-of-Importance: How NEM is Going to Add Reputations to the Blockchain.: https://cointelegraph.com/news/proof-of-importance-nem-is-going-to-add-reputations-to-the-blockchain adresinden alındı

Beikverdi, A. (2015, Nisan 1). Cointelegraph. NEM Launches, Targets Old Economy with Proof-of-Importance: https://cointelegraph.com/news/nem-launches-targets-old-economy-with-proof-of-importance adresinden alındı

Blum, M. (1983). Coin flipping by telephone a protocol for solving impossible problems. ACM SIGACT News - A special issue on cryptography, 15(1), 23-27.

Buterin, V., Griffith, V. (2017). Casper the friendly finality gadget. https://arxiv.org/abs/1710.09437 adresinden alındı

Castro, M., Liskov, B. (1999). Practical byzantine fault tolerance. Proceedings of the Third Symposium on Operating Systems Design and Implementation, OSDI '99 (s. 173-186). Berkeley, CA, USA: USENIX Association.

Dwork, C., Goldberg, A., Naor, M. (2013). On memory-bound functions for fighting spam. Advances in Cryptology (s. 426-444). Berlin, Heidelberg: Springer Berlin Heidelberg.

Feldman, P. (1987). A practical scheme for noninteractive verifiable secret sharing. Annual Symposium on Foundations of Computer Science (s. 427–438). Washington, DC, USA: IEEE Computer Society.

Jakobsson, M., Juels, A. (1999). Proofs of Work and Bread Pudding Protocols (Extended Abstract). Secure Information Networks (s. 258-272). Boston: Springer US.

Kiayias, A., Russell, A., David, B., Oliynykov, R. (2016). Ouroboros: A provably secure proof-of-stake blockchain protocol. https://eprint.iacr.org/2016/889 adresinden alındı

King, S., Nadal, S. (2012, Ağustos 19). Ppcoin: Peer-topeer crypto-currency with proof-of-stake.

Lamport, L., Shostak, R., Pease, M. (1982). The byzantine generals problem. ACM Transactions on Programming Languages and Systems, 4(3), 382–401.

Larimer, D. (2014). Delegated Proof of Stake.

Mazieres, D. (2016, Şubat 25). The stellar consensus protocol: A federated model for internet level consensus.

Nakamoto, S. (2009). Bitcoin: A peer-to-peer electronic cash system.

Naor, M., Dwork, C. (1993). Pricing via processing or combatting junk mail. Advances in Cryptology

-CRYPTO' 92 (s. 139-147). Berlin, Heidelberg: Springer Berlin Heidelberg.

NIST. (2013). Federal Information Processing Standards (FIPS) Publication 186-4, Digital Signature Standard(DSS). National Institute of Standards and Technology.

NIST. (2015). Federal Information Processing Standards (FIPS) Publication 180-4, Secure Hash Standard (SHS). National Institute of Standards and Technology.

Platform Waves. (2018, Mart 3). Waves launches balance leasing in lite client.

Schwartz, D., Youngs, N., Britto, A. (2014). The ripple protocol consensus algorithm.

Blockchain technology: consensus protocols

Extended abstract

A blockchain is defined as an inter-node system that does not have any central authority to manage the data stream. A block is a copy of digital ledger stored in a distributed network. These occur when the owners digitally sign their transactions after they have been allocated to the blocks. Each block is cryptographically linked to the previous block. This link makes the blockchain unbreakable in the long run. More specifically, when we examine the structure of any blockchain, it is seen that there are eight different composites including hash functions, public key cryptography, addresses, operations, blocks, books, nodes and consensus protocols. All nodes have the same rights, and the nodes do not have to trust each other. The nodes only disclose themselves with their open addresses and become motivated by the desire for profit, not the welfare of other mining nodes or the need to act as a whole. In such a case, why would a node want to publish a block prepared by another node? In addition, who will resolve conflicts when more than one mining node resolves a block at about the same time? For these reasons, various consensus protocols emerged that allow mutually insecure user groups to work together.

In this study, the advantages and disadvantages of the existing consensus protocols in the literature and in practice are given in a comparative way. In this respect, we first define the important criteria that provide the differences in the existing blockchain technology. These criteria are significant in order for classify the effectiveness of the consensus algorithms. We also classify all existing consensus algorithm into three classes:

- *Proof of Work*
- *Proof of Stake*
- *Byzantine Fault Tolerance*

The consensus algorithm of the blockchain, which is first implemented by Bitcoin in its core, is the proof of work algorithm. This algorithm solves cryptographic puzzles during the block generations and requires a great amount of computation. This consensus protocol along with its advantages and disadvantages are explained in detail.

Moreover, the second class of consensus algorithm is the proof of stake protocol. This consensus algorithm mainly focuses on the amount of coin hold by the owner. In addition, the advantages and disadvantage of this algorithm are explained in detailed.

Furthermore, Byzantine fault tolerance along with it is the problem of general problem. There are three different versions arose in the literature. These are slightly different from each other but this difference made a great impact on the trust of blockchain. In addition, the advantages and disadvantage of these algorithms are explained in detailed.

Finally, the general characteristics of the consensus protocols described above are summarized in Table 1. When we look at the table, all the other models except for the proof of work are used according to the situation in the permission and unauthorized blockchain. The accuracy of the transactions is only realized in the Byzantine fault tolerance compromise models. In the proof of work, the transfer process is slow because the miners are required to make a high amount of transactions within a certain period of time in forming each block, while in other models, these processes are done very fast. The confidence in the nodes that make up the block is only found in the Byzantine fault tolerance models.

Keywords: *blockchain, consensus protocols, cryptocurrency*