T.C.

TRAKYA ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

BLOK ZİNCİRİ TEKNOLOJİSİNDE KULLANILAN KONSENSÜS ALGORİTMALARININ İNCELENMESİ VE YENİ BİR ALGORİTMA TASARIMI

TUĞBA ÇALIŞ

DOKTORA TEZİ

HESAPLAMALI BİLİMLER ANABİLİM DALI

Tez Danışmanı: Prof. Dr. Muharrem Tolga SAKALLI

EDİRNE-2025

Tuğba ÇALIŞ'ın hazırladığı **"Blok Zinciri Teknolojisinde Kullanılan Konsensüs Algoritmalarının İncelenmesi ve Yeni Bir Algoritma Tasarımı"** başlıklı bu tez, tarafımızca okunmuş, kapsam ve niteliği açısından Hesaplamalı Bilimler Anabilim Dalında bir **Doktora tezi** olarak kabul edilmiştir.

Jüri Üyeleri	İmza
Prof. Dr. Muharrem Tolga SAKALLI (Tez Danışmanı)	……
Prof. Dr. Metin AYDOĞDU	……
Prof. Dr. Hacı Ali GÜLEÇ	……
Prof. Dr. Erdinç UZUN	……
Dr. Öğr. Üyesi Bora ASLAN	……

Tez Savunma Tarihi: 03/01/2025

Bu tezin Doktora tezi olarak gerekli şartları sağladığını onaylarım.

İmza

Prof. Dr. Muharrem Tolga SAKALLI

Tez Danışmanı ……

Trakya Üniversitesi Fen Bilimleri Enstitüsü onayı

……

Doç. Dr. Filiz UMAROĞULLARI

Fen Bilimleri Enstitüsü Müdürü

T. Ü. FEN BİLİMLERİ ENSTİTÜSÜ

HESAPLAMALI BİLİMLER ANABİLİM DALI DOKTORA PROGRAMI

DOĞRULUK BEYANI

Trakya Üniversitesi Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında, tüm verilerin bilimsel ve akademik kurallar çerçevesinde elde edildiğini, kullanılan verilerde tahrifat yapılmadığını, tezin akademik ve etik kurallara uygun olarak yazıldığını, kullanılan tüm literatür bilgilerinin bilimsel normlara uygun bir şekilde kaynak gösterilerek ilgili tezde yer aldığını ve bu tezin tamamı ya da herhangi bir bölümünün daha önceden Trakya Üniversitesi ya da farklı bir üniversitede tez çalışması olarak sunulmadığını beyan ederim.

03 / 01 / 2025

Tuğba ÇALIŞ

İmza

Doktora Tezi

Blok Zinciri Teknolojisinde Kullanılan Konsensüs Algoritmalarının İncelenmesi ve Yeni Bir Algoritma Tasarımı

T.Ü. Fen Bilimleri Enstitüsü

Hesaplamalı Bilimler Anabilim Dalı

ÖZET

Blok zinciri, dijital verilerin güvenli, şeffaf ve merkeziyetsiz bir şekilde kaydedilmesine imkân tanıyan, birbirine bağlı veri bloklarından oluşan bir teknoloji olarak tanımlanmaktadır. Her blok, belirli bir zaman dilimi içerisinde gerçekleşen bir dizi işlemi içermektedir ve bir önceki bloğun verilerini içeren şifrelenmiş bir özet (hash) ile birbirine bağlanmaktadır. Bu zincirli yapı, her bloğun bir önceki bloğun şifrelenmiş bir özetini içermesi ve bu özete dayanarak bir bağ oluşturması nedeniyle, verilerin değiştirilmesini veya silinmesini son derece zorlaştırmaktadır. Merkezi bir otoriteye ihtiyaç duymaksızın işlemlerin doğrulanması ve kaydedilmesi amacıyla geliştirilen blok zinciri teknolojisi, özellikle kripto para birimleri, akıllı sözleşmeler ve dijital kimlik doğrulama gibi alanlarda yaygın biçimde kullanılmaktadır.

Blok zinciri teknolojisi, her bir işlemin tanımlanması, doğrulanması, depolanması, paylaşılabilmesi için dijital olarak kaydedilmesi ve imzalanması gibi aşamaları içeren bir teknolojidir. Bu protokole göre, katılımcıların belirli bir veri ve işlem üzerinde ortak bir karara varması (mutabakata ulaşması) gerekmektedir. Konsensüs algoritmaları, blok zincir ağı içinde tüm ağ katılımcılarının aynı verileri kabul etmesini sağlamak, ağda gerçekleşen işlemlerin geçerli olduğunu doğrulamak, ağda tek bir merkezi otorite olmaksızın güvenli ve tutarlı işlem kaydı oluşturmak ve dağıtık katılımcılar arasında fikir birliği sağlamak amacıyla kullanılmaktadır. Dağıtık bir sistemde veri bütünlüğünü koruyan bu algoritmalar, tüm katılımcıların aynı bilgiye sahip olmasını ve zincirin güvenliğini sağlamaktadır. Farklı blok zinciri ağları tarafından güvenlik, merkeziyetsizlik ve performans dengelerini sağlamak için İş Kanıtı (PoW),

Hisse Kanıtı (PoS), Yetkilendirilmiş Hisse Kanıtı (DPoS), Pratik Bizans Hata Toleransı (pBFT), Federatif Bizans Anlaşması (FBA), Kapasite Kanıtı (PoC), Yetki Kanıtı (PoA), Yakma Kanıtı (PoB), Geçen Zaman Kanıtı (PoET), Önem Kanıtı (PoI), Ripple Protokol Konsensüs Algoritması (RPCA) gibi çeşitli algoritmalar geliştirilmiştir. Bu tez çalışmasında, mevcut konsensüs algoritmalarının incelenmesi ve çift katmanlı yeni bir algoritma tasarlanarak, ilk katmanda rastgele doğrulayıcı seçimi ve ikinci katmanda rastgele seçilen 3 doğrulayıcıdan birinin oylama yoluyla belirlenmesi ile doğrulama süreçlerinin dağıtılması ve blok zinciri sisteminde merkeziyetsizliğin güçlendirilmesi amaçlanmıştır.

Yıl : 2025

Sayfa Sayısı : 104

Anahtar Kelimeler : Blok Zinciri, Konsensüs Algoritmaları, PoW, PoS.

PhD Thesis

Analysis of Consensus Algorithms Used in Blockchain Technology and the Development of a Novel Algorithm

Trakya University Institute of Natural and Applied Sciences

Department of Computational Sciences

ABSTRACT

Blockchain is defined as a technology consisting of interconnected data blocks that allows digital data to be recorded securely, transparently, and in a decentralized manner. Each block contains a series of transactions that occurred within a specific time frame and is linked to the previous block with an encrypted hash, making it extremely difficult to alter or delete the data. Blockchain technology, developed to verify and record transactions without the need for a central authority, is widely used in areas such as cryptocurrencies, smart contracts, and digital identity verification.

Blockchain technology encompasses the stages of defining, verifying, storing, and sharing each transaction, digitally recording and signing them. According to this protocol, participants are required to reach a consensus on a specific data and process. Consensus algorithms are used to ensure that all participants in the blockchain network accept the same data, validate the transactions that occur in the network, create a secure and consistent transaction record without a central authority, and achieve consensus among distributed participants. These algorithms, which ensure data integrity in a distributed system, guarantee that all participants have the same information and ensure the security of the chain. Various consensus algorithms, such as Proof of Work (PoW), Proof of Stake (PoS), Delegated Proof of Stake (DPoS), Practical Byzantine Fault Tolerance (pBFT), Federated Byzantine Agreement (FBA), Proof of Capacity (PoC), Proof of Authority (PoA), Proof of Burn (PoB), Proof of Elapsed Time (PoET), Proof of Importance (PoI),

and Ripple Protocol Consensus Algorithm (RPCA), have been developed to achieve a balance between security, decentralization, and performance in different blockchain networks. This thesis aims to analyze existing consensus algorithms and design a novel two-layered algorithm to enhance decentralization in blockchain systems.

Year : 2025

Number of Pages : 104

Keywords : Block Chain, Consensus Algorithms, PoW, PoS

TEŞEKKÜR

Bu tezin hazırlanması sürecinde akademik birikimi ile bana yol gösteren, yardımcı olan, bilgisini esirgemeyen değerli danışman hocam Prof. Dr. M. Tolga SAKALLI ’ya çok teşekkür ederim. Doktora tezi çalışmamı titizlikle takip eden ve tecrübeleriyle yol gösteren değerli hocalarım Prof. Dr. Hacı Ali GÜLEÇ’e ve Prof. Dr. Metin AYDOĞDU’ya çok teşekkür ederim. Tüm çalışma arkadaşlarıma ve hocalarıma anlayışları, destekleri için ayrı ayrı çok teşekkür ederim.

Tez yazım aşamasında desteklerini esirgemeyen ağabeyim Arş. Gör. Dr. İbrahim Uğur ÇALIŞ’a ve yengem Dr. Öğr. Üyesi Özge Alev SÖNMEZ ÇALIŞ’a çok teşekkür ederim. Dünyaya gelmesiyle hayatıma renk katan ve beni hala yaparak büyük bir mutluluk hediye eden biricik yeğenim Bernis Lila ÇALIŞ’a çok teşekkür ederim. Çalışmalarım sırasında ve hayatımın hemen her aşamasında yanımda olan dostlarım Önder AÇIKGÖZ’e ve Hivda HOCAOĞLU’na çok teşekkür ederim.

Son olarak, beni bu dünyaya getiren, büyüten, okutan, maddi manevi desteklerini esirgemeyen, annem Ümmühan ÇALIŞ’a, babam Adnan ÇALIŞ’a çok teşekkür ederim.

İÇİNDEKİLER

ÖZET iv
ABSTRACT vi
TEŞEKKÜR viii
İÇİNDEKİLER ix
SİMGELER VE KISALTMALAR DİZİNİ xii
ŞEKİLLER DİZİNİ xiv
ÇİZELGELER DİZİNİ xv
BÖLÜM 1 1
GİRİŞ 1
1.1. Literatür Özeti 2
1.2. Tezin Amacı 9
1.3. Tezin Hipotezi 9
BÖLÜM 2 11
BLOK ZİNCİRİ TEKNOLOJİSİ 11
2.1. Blok Zincirinin Yapısı ve Çalışma Sistemi 12
2.2. Blok Zinciri Türleri 17
2.2.1. Genel Blok Zincirleri 17
2.2.2. Özel Blok Zincirleri 19
2.2.3. Konsorsiyum Blok Zincirleri 20
2.2.4. Hibrit Blok Zincirleri 21
2.3. Blok Zinciri Uygulama Alanları 23

BÖLÜM 3 25
KONSENSÜS ALGORİTMALARI 25
3.1. Konsensüs Algoritması Türleri 25
3.1.1. İş Kanıtı Algoritması 27
3.1.2. Hisse Kanıtı Algoritması 31
3.1.3. Yetkilendirilmiş Hisse Kanıtı 35
3.1.4. Pratik Bizans Hata Toleransı 37
3.1.5. Federatif Bizans Anlaşması 39
3.1.6. Kapasite Kanıtı 40
3.1.7. Yetki Kanıtı 41
3.1.8. Yakma Kanıtı 43
3.1.9. Geçen Zaman Kanıtı 44
3.1.11. Ripple Protokol Konsensüs Algoritması 47
3.1.12. Diğer Konsensüs Algoritmaları 48
3.2. Konsensüs Algoritmalarının Karşılaştırılması 49
3.2.1. Güvenlik 49
3.2.2. Ölçeklenebilirlik 50
3.2.3. Enerji Verimliliği 50
3.2.4. Gecikme Süresi 50
BÖLÜM 4 53
YENİ BİR KONSENSUS ALGORİTMASI 53
4.1. Uygulama Amaç ve Kapsamı 53
4.2. Uygulamanın Planlanması ve Hazırlık Süreci 53
4.3. Uygulamada Kullanılan Araç ve Yöntemler 54
4.4. Uygulama Süreci 54

4.4.1. Blok Zincirinin Başlatılması 56
4.4.2. Başlangıç Bloğunun Oluşturulması 57
4.4.3. Blok Yönetimi 57
4.4.4. Doğrulayıcı Seçimi 58
4.4.5. Doğrulayıcı Oylaması 58
4.4.6. Özet Hesaplama 59
4.4.7. Ana İşlem Akışı 59
4.4.8. Blok Zincirinin Geçerliliğinin Kontrolü 59
4.5. Karşılaşılan Zorluklar ve Çözümleri 60
BÖLÜM 5 62
SONUÇLAR VE TARTIŞMA 62
5.1. Uygulama Sonuçları ve Çıkarımlar 62
5.2. Tartışma 62
5.3.Performans Test Sonuçları 63
5.4. Algoritmanın Zaman ve Bellek Karmaşıklığı 64
5.5. Gelecek Çalışmalar İçin Öneriler 65
KAYNAKLAR 67
EKLER 81
EK-A: ALGORİTMA UYGULAMA KODU 81
EK-B: ALGORİTMA UYGULAMA TEST KODU 86
ÖZGEÇMİŞ 89

SİMGELER VE KISALTMALAR DİZİNİ

ASIC:	Uygulamaya Özgü Entegre Devre
BFT:	Bizans Hata Toleransı
CPU:	Merkezi İşlem Birimi
DeFi:	Merkeziyetsiz Finans
DBFT:	Yetkilendirilmiş Bizans Hata Kanıtı
DPoS:	Yetkilendirilmiş Hisse Kanıtı
FBA:	Federatif Bizans Anlaşması
GPU:	Grafik İşleme Birimi
LPoS:	Kiralanmış Hisse Kanıtı
NEM:	Yeni Ekonomi Hareketi
PoA:	Yetki Kanıtı
PoB:	Yakma Kanıtı
PoC:	Kapasite Kanıtı
PoET:	Geçen Zaman Kanıtı
PoH:	Tarih Kanıtı
PoI:	Önem Kanıtı
PoP:	Olasılık Kanıtı
PoR:	İtibar Kanıtı
PoRand:	Rastgelelik Kanıtı
PoS:	Hisse Kanıtı
PoSpace:	Alan Kanıtı
PoW:	İş Kanıtı

RPCA:	Ripple Protokol Kanıtı
SCP:	Stellar Konsensüs Protokolü
SGX:	Yazılım Koruma Uzantıları
TEE:	Güvenilir Yürütme Ortamı
UNL:	Benzersiz Düğüm Listesi

ŞEKİLLER DİZİNİ

Şekil 2.1. Merkle Ağacı.....17

Şekil 3.1. Konsensüs Algoritmalarının Sınıflandırılması.....26

Şekil 3.2. PoW algoritması akış şeması.....28

Şekil 3.3. PoS algoritması akış şeması.....32

Şekil 3.4. DPoS algoritması akış şeması.....35

Şekil 3.5. PBFT algoritması akış şeması.....38

Şekil 3.6. PoET algoritması akış şeması.....45

Şekil 4.1. Tasarlanan algoritmanın akış şeması.....56

ÇİZELGELER DİZİNİ

Çizelge 2.1. SHA-256 özet fonksiyonu örnek bir girdi ve çıkış değeri 12

Çizelge 2.2. Temel bir blok yapısı 14

Çizelge 3.1. Konsensus algoritmalarının güçlü ve zayıf yanlarının, kullanıldığı platformların ve blok zinciri türlerine göre karşılaştırılması 51

Çizelge 3.2. Konsensus algoritmalarının güvenlik, ölçeklenebilirlik, enerji verimliliği ve gecikme süresi kriterlerine göre karşılaştırılması 52

BÖLÜM 1

GİRİŞ

Son yıllarda dijitalleşme sürecinin hızlanmasıyla birlikte, veri güvenliği ve merkeziyetsizlik kavramları önem kazanmıştır. Bu bağlamda, blok zinciri teknolojisi, veri güvenliğini sağlamak, merkezi otoritelerden bağımsız bir yapı oluşturmak ve şeffaf bir şekilde işlemleri kayıt altına almak amacıyla geliştirilen bir çözüm olarak dikkat çekmektedir. Blok zinciri teknolojisi, işlemlerin güvenilirliğini ve şeffaflığını sağlarken, aynı zamanda katılımcılar arasında merkezi otoriteye ihtiyaç duymaksızın güvenli bir şekilde veri transferi gerçekleştirmektedir (Zheng, Xie, Dai, Chen & Wang, 2017).

Blok zinciri, dağıtık bir defter yapısı ile çalışmakta ve işlemleri katılımcılar arasında güvenli, şeffaf ve değişmez bir şekilde kaydetmektedir. Blok zincirinin en önemli avantajlarından biri, merkezi otoriteleri ortadan kaldırarak, taraflar arasında doğrudan güvenilir işlemler yapılmasını sağlamasıdır (Yaga, Mell, Roby & Scarfone, 2019). Bu nedenle, finans, tedarik zinciri ve sağlık gibi birçok sektörde blok zinciri teknolojisinin kullanımı hızla artmaktadır (Crosby, Pattanayak, Verma & Kalyanaraman, 2016).

Blok zinciri sistemlerinin temelinde konsensüs algoritmaları yer almaktadır. Konsensüs algoritmaları, ağdaki tüm katılımcıların belirli bir işlem üzerinde anlaşmasını sağlamak amacıyla kullanılan yöntemlerdir. İş Kanıtı ve Hisse Kanıtı gibi yaygın algoritmalar, bu teknolojinin temel taşlarını oluşturmaktadır. Ancak, bu algoritmalarda performans, enerji tüketimi ve ölçeklenebilirlik gibi çeşitli zorluklar bulunmaktadır (Zheng vd., 2017). Bu zorlukların aşılması ve yeni nesil konsensüs algoritmalarının geliştirilmesi, blok zinciri teknolojisinin geleceği açısından kritik bir öneme sahiptir (Yaga vd., 2019).

Bu tez çalışmasında, ilk olarak blok zinciri teknolojisinde kullanılan literatürdeki mevcut konsensüs algoritmalar incelenmiştir. Ardından, ikinci bölümde blok zinciri

teknolojisi, çalışma sistemi, blok zinciri türleri hakkında bilgi verilmiştir. Üçüncü bölümde blok zinciri teknolojisinde kullanılan konsensüs algoritmaları, performans, merkeziyetsizlik, enerji tüketimi bakımından karşılaştırılmıştır. Dördüncü bölümde ise blok zinciri teknolojisi üzerine yeni bir bakış açısı kazandıracak iki katmanlı bir konsensüs algoritması önerilmiş ve bu algoritmanın Go programlama dili kullanılarak simülasyonu gerçekleştirilmiştir. Son olarak, önerilen algoritmanın uygulanmasına ilişkin sonuçlar değerlendirilmiş ve tartışılmıştır.

1.1. Literatür Özeti

Blok zinciri teknolojisinin gelişiminde ve yaygınlaşmasında konsensüs algoritmalarının kritik bir rol oynadığı görülmektedir. Konsensüs algoritmaları, merkezi olmayan blok zincir ağlarında işlemlerin doğrulanması ve güvenliğin sağlanması amacıyla tüm düğümlerin belirli bir işlem üzerinde mutabakata varmasına olanak tanımaktadır. Bu algoritmaların temel hedefi, merkeziyetsizlik ilkesine bağlı kalarak güvenilir, hızlı ve enerji verimli bir doğrulama süreci sağlamaktır. Literatürde farklı yapıda birçok konsensüs algoritması incelenmiş olup, her bir algoritmanın kendi avantajları ve kısıtlamaları bulunmaktadır. İş Kanıtı gibi ilk nesil algoritmalar güvenlik odaklıyken, Hisse Kanıtı gibi yeni nesil modeller daha çok enerji verimliliğine odaklanmıştır. Bunun yanı sıra, Federatif Bizans Anlaşması ve Yetki Kanıtı gibi algoritmalar da kurumsal uygulamalara uyumlu çözümler sunmaktadır. Bu bölümde, literatürde geniş yer bulan başlıca konsensüs algoritmaları, avantajları ve dezavantajları, ilgili araştırmalarla birlikte ele alınmaktadır.

İş Kanıtı algoritması, blok zinciri teknolojisinin ilk konsensüs algoritmalarından biri olarak ele alınmaktadır. Nakamoto'nun Bitcoin makalesinde tanıtılan İş Kanıtı algoritması, yüksek güvenlik sağlaması algoritmanın en güçlü yanı olarak literatürde yer bulmuştur (Nakamoto, 2008). Ancak İş Kanıtı algoritmasının çevresel etkileri ve enerji tüketimini azaltmaya yönelik alternatifler üzerine yapılan çalışmalar, blok zinciri teknolojisinin sürdürülebilirliği açısından büyük önem taşımaktadır. Örneğin, Shi, Chen ve Liu 2021'deki çalışmasında, PoW tabanlı blok zincir platformlarının enerji tüketimi ve karbon ayak izi detaylı bir şekilde incelenmiş ve enerji tüketimini azaltmak için Hisse Kanıtı gibi alternatif konsensüs mekanizmalarının benimsenmesi önerilmiştir. Benzer şekilde, Ibañez ve Freier, 2023 yılında araştırmasında, Bitcoin madenciliğinin karbon

ayak izi ele alınmış ve yenilenebilir enerji kaynaklarının kullanımının artırılmasıyla çevresel etkinin azaltılabileceği vurgulanmıştır. Bu çalışmalara ek olarak, Dubrovsky, Wei ve Goldstein'ın 2019'daki araştırmasında, enerji tüketimini azaltmak amacıyla Optik İş Kanıtı (Optical Proof of Work) adlı yeni bir konsensüs algoritması önerilmiştir. Bu algoritma, enerji tüketimini azaltırken güvenlikten ödün vermemeyi hedeflemektedir. Bu çalışmalar, PoW algoritmasının enerji tüketimi ve çevresel etkilerini azaltmaya yönelik alternatif konsensüs mekanizmalarının ve yenilenebilir enerji kaynaklarının kullanımının önemini ortaya koymaktadır.

Hisse Kanıtı algoritması, İş Kanıtı algoritmasının enerji yoğunluğuna karşılık daha verimli bir alternatif olarak geliştirilmiş ve literatürde karşılaştırmalı olarak incelenmiştir. King ve Nadal (2012)'ın geliştirdiği PoS modeli, PoW'a kıyasla daha az enerji gereksinimi ile öne çıkmaktadır. Kumar ve Gupta, 2024 yılında PoS algoritmasının ölçeklenebilirlik avantajlarını ele alarak, bu algoritmanın daha demokratik ve sürdürülebilir bir yapı sunduğunu göstermiştir. Benzer şekilde, Fang, 2023 yılında PoS'un enerji verimliliğini ve güvenlik özelliklerini inceleyerek PoW'a kıyasla daha sürdürülebilir bir çözüm sunduğunu belirtmiştir. Bu çalışmalar, PoS'un enerji verimliliği ve çevresel sürdürülebilirlik açısından literatürde yaygın olarak tercih edilmesini açıklar niteliktedir.

Pratik Bizans Hata Toleransı algoritması, özellikle özel blok zincirlerde kullanılan bir güvenlik mekanizması olarak literatürde geniş yer bulmuştur. PBFT, dağıtık sistemlerde Bizans hata toleransını sağlamak amacıyla geliştirilmiştir. Nguyen ve Cronje (2020)'nin çalışmalarında, PBFT algoritmasının güvenlik ve hız açısından sunduğu avantajlar ele alınarak, özellikle düşük gecikme süresi ve yüksek işlem doğrulama hızı vurgulanmıştır. Luo, Zhang, Yu, Sun ve Xu, 2023 araştırması ise PBFT'nin özel blok zincirler üzerindeki uygulamalarını analiz etmiş ve güvenlik özellikleri üzerinde durmuştur. Bu araştırmalar, PBFT'nin güvenli ve hızlı bir konsensüs mekanizması olarak özel blok zincirlerde kullanımını desteklemektedir. Bununla beraber, PBFT konsensüs mekanizması, özellikle büyük ölçekli ağlarda (çok sayıda düğümle) bir performans darboğazı oluşturabilmektedir (Sukhwani, Martinez, Chang, Trivedi & Rindos, 2017). Bu hipotez doğrultusunda, Sukhwani vd. PBFT mekanizmasının performansını değerlendirerek, ağ büyüklüğü, mesaj iletim süresi ve diğer sistem parametrelerinin konsensüs süresi üzerindeki etkilerini incelemeyi amaçlamıştır (Sukhwani vd., 2017).

Yetkilendirilmiş Hisse Kanıtı, Hisse Kanıtı algoritmasının demokratik bir varyantı olarak öne çıkmaktadır. Literatürde, DPoS algoritmasının özellikle yüksek işlem hacmine sahip blok zincir platformlarında uygulandığı ve başarılı sonuçlar elde ettiği belirtilmiştir. EOS, TRON ve BitShares gibi platformlar, DPoS'un avantajlarından yararlanarak saniyede binlerce işlemi gerçekleştirme kapasitesine ulaşabilmiştir (Flarecompare, t.y.; Kaur vd., 2021). Bu durum, DPoS'un yüksek verimliliğe sahip blok zincir sistemleri için ideal bir yapı sunduğunu göstermektedir. Ayrıca, DPoS'un oylama yoluyla doğrulayıcı seçme mekanizması, ağdaki katılımcıların güvenlik süreçlerine dolaylı yoldan katılmalarını sağlarken, merkeziyetsizlik ilkesini de korumaktadır. Zhang ve Lee (2023) tarafından yapılan çalışmada, DPoS'un demokratik yönetim yapısının avantajları vurgulanmış; bu algoritmanın hem ölçeklenebilirliği artırdığı hem de enerji verimliliğini sağladığı belirtilmiştir. Kaur vd. (2021) çalışmasında, DPoS'un yüksek işlem hacmi ve hızlı doğrulama süreleri sunan yapısının, finansal işlemler gibi hızlı onay gerektiren alanlarda büyük fayda sağladığı belirtilmiştir. Ayrıca, düşük enerji tüketimi sayesinde DPoS, çevresel sürdürülebilirlik açısından da avantaj sunmaktadır. Bu özellik, PoW tabanlı sistemlerin yoğun enerji tüketimi sorununa alternatif olarak DPoS'un popülaritesini artırmaktadır. Ancak, DPoS'un merkeziyetsizlik düzeyini sınırlayabileceği ve belirli grupların ağ üzerinde hakimiyet kurma riskini artırabileceği belirtilmektedir (Hossain, Nirob, Islam, Rakin & Al-Amin, 2024). Örneğin, yoğun oy dağılımı, belirli grupların veya kullanıcıların oyları üzerinde hakimiyet kurarak, ağın güvenliğine potansiyel tehdit oluşturabilmektedir. Hossain vd., bu durumun güvenlik açısından zayıflık yaratabileceğini ve saldırılara karşı ağın kırılganlığını artırabileceğini öne sürmüştür. DPoS algoritması, sunduğu avantajlar kadar zorlukları da içeren bir konsensüs algoritması olarak değerlendirilmektedir. Gelecekteki araştırmalarda, DPoS'un mevcut kısıtlarını aşmaya yönelik yeni yaklaşımlar önerilmektedir. Özellikle, ağdaki demokratik yapının güçlendirilmesi ve merkeziyetsizliğin artırılması için oy dağılımının daha geniş bir tabana yayılması, delegelerin seçiminde daha kapsayıcı yöntemlerin benimsenmesi gibi çözümler gündeme gelmektedir (Zhang & Lee, 2023).

Federatif Bizans Anlaşması, dağıtık sistemlerde güvenlik ve ölçeklenebilirlik sağlamak amacıyla geliştirilmiş bir konsensüs algoritmasıdır. FBA'nın kökenleri, Lamport, Shostak ve Pease'nin önerdiği Byzantine Fault Tolerance algoritmalarına dayanmaktadır (Lamport vd., 1982). BFT'nin temel prensiplerini genişleten FBA,

düğümlerin kendi güvenilir setlerini (quorum slices) belirleyebilmesini sağlayarak merkeziyetsizlik ve esneklik avantajları sunmaktadır. FBA'nın ilk önemli uygulaması, Stellar Konsensüs Protokolü (Stellar Consensus Protocol- SCP) olarak bilinmektedir. 2015 yılında Mazières tarafından önerilen bu protokol, FBA modelini kullanarak finansal işlemleri hızlı ve düşük maliyetle gerçekleştirmeyi hedeflemiştir (Mazières, 2015). Stellar protokolü, açık katılımlı ağlarda güvenlik ve performans dengesini sağlayan yenilikçi bir yapı sunmaktadır. FBA'nın teorik temelleri ve uygulama detayları, Mazières'in seminal (öncü) çalışmasında detaylandırılmıştır. Bu çalışma, FBA'nın geleneksel BFT algoritmalarına kıyasla güvenlik, ölçeklenebilirlik ve esneklik gibi avantajlarını ortaya koymaktadır (Mazières, 2015). FBA, özellikle genel (public) ağlar için uygun bir konsensüs mekanizması olarak tanımlanmıştır. Sonraki yıllarda, FBA üzerine yapılan araştırmalar hızla artmıştır. Örneğin, 2021 yılında yayımlanan bir derleme çalışması, FBA ile diğer konsensüs algoritmalarını karşılaştırmış ve FBA'nın avantajlarını tartışmıştır (Bouraga, 2021). Benzer şekilde, Zhou, Huang, Zheng ve Bian tarafından yapılan bir analizde, FBA'nın farklı uygulama senaryolarındaki performansı incelenmiştir (Zhou vd., 2020). Bu çalışmalar, FBA'nın dağıtık sistemlerdeki geniş uygulama potansiyelini vurgulamaktadır.

Kapasite Kanıtı, blok zinciri ağlarında kullanılan enerji verimli bir konsensüs algoritmasıdır. PoC, madencilerin hesaplama gücü yerine depolama alanlarını kullanarak blok üretmelerine olanak tanımakta ve bu özelliğiyle İş Kanıtı algoritmasından farklılık göstermektedir. Bu mekanizma, bir madencinin sahip olduğu depolama alanını kullanarak önceden birçok potansiyel çözümü hesaplaması ve bu çözümleri sabit disklerinde saklamasını gerektirmektedir. Yeni bir blok oluşturma ihtiyacı doğduğunda, madenciler tarafından bu önceden hesaplanmış çözümler taranmakta ve en uygun çözüm kullanılmaktadır (Jiang & Wu, 2020). PoC'nin teorik temelleri, depolama alanının bir kanıt mekanizması olarak kullanılabileceğini öne süren araştırmalara dayanmaktadır (Dziembowski, Faust, Kolmogorov & Pietrzak, 2015). İlk uygulaması ise 2014 yılında Burstcoin (şimdiki adıyla Signum) tarafından gerçekleştirilmiştir (Signum Network, 2021). Burstcoin, PoC'yi kullanarak enerji verimliliği sağlamayı ve madencilik sürecini daha erişilebilir hale getirmeyi hedeflemiştir. Daha yakın zamanda, PoC algoritması, Chia gibi projelerde popülerlik kazanmıştır (Hern, 2021). Jiang ve Wu, PoC'nin mobil cihazlarda uygulanabilirliğini incelemiş ve bu süreçte bulut depolama (cloud storage)

kullanımı gibi yenilikçi çözümler önermiştir. Bu yöntem, madencilerin sabit disk kapasitelerinin sınırlı olduğu durumlarda bulut depolama sağlayıcılarını kullanarak madencilik yapmalarına olanak tanımaktadır. Ayrıca Gate.io, PoC'nin enerji verimliliği ve düşük maliyet avantajlarına dikkat çekerken, merkeziyetsiz madencilik sürecine katkısını da vurgulamaktadır (Gate.io, 2022). Bunlara ek olarak, Park vd., SpaceMint adlı bir kripto para birimi önererek PoC'nin uygulanabilirliğini ve güvenliğini artırmayı amaçlamışlardır. Bu çalışma, PoC tabanlı bir blok zinciri sisteminin saldırılara karşı nasıl korunabileceğini ve daha adil bir madencilik sürecinin nasıl sağlanabileceğini detaylandırmaktadır (Park vd., 2018).

Yetki Kanıtı, özel blok zincir ağları için geliştirilmiş, güvenilirlik ve verimliliği ön planda tutan bir konsensüs mekanizmasıdır. Doğrulayıcı seçimi kimlik ve itibar esasına dayanmaktadır. PoA'nın düşük enerji tüketimi ve hızlı işlem doğrulama gibi avantajları bulunmaktadır (Lu, Lei & Tsai, 2024). Bununla beraber, bu algoritmanın merkeziyetsizlikten ödün vermesi ile karşılaşılan zorlukları ve güvenlik zafiyetlerini ele alan çeşitli akademik çalışmalar gerçekleştirilmiştir. Wang, Li, Wang, Chen ve Xiang çalışmasında, PoA mekanizmasının adaletsizlik sorunlarını ele almış ve işlem sıralaması manipülasyonlarına odaklanmıştır. Araştırma, doğrulayıcıların işlem sırasını manipüle ederek kendi çıkarları için kullanabileceğini ve bu tür saldırılara karşı önerilen çözüm yollarını incelemiştir. Çalışma, PoA'nın özel ağlarda bile güvenlik sorunlarıyla karşılaşabileceğini ve daha güçlü doğrulayıcı seçim süreçlerinin önemini vurgulamaktadır (Wang vd., 2022). Ekparinya, Gramoli ve Jourjon, PoA algoritmasının güvenlik açıklarını daha da genişleterek klonlama saldırılarını incelemiştir. Bu saldırılarda, kötü niyetli aktörler doğrulayıcı düğümleri kopyalayarak ağ üzerinde kontrol sağlamaya çalışmaktadır. Bu çalışmada, bu tür saldırıların etkilerini analiz edilmiş ve savunma mekanizmaları önerilmiştir. Çalışma, PoA'nın güvenilir ortamlar için uygunluğunu değerlendirirken, bu tür tehditlere karşı dayanıklılığının artırılması gerektiğini göstermiştir (Ekparinya vd., 2019). Joshi, PoA'nın uygulanabilirliğini PoW ve PoS gibi diğer konsensüs algoritmalarıyla karşılaştırmalı olarak ele almıştır. Çalışma, PoA'nın düşük enerji tüketimi ve yüksek işlem hızı gibi avantajlarını vurgularken, merkeziyetçiliğe dayalı yapısının, özellikle genel (public) blok zincirler için bir dezavantaj olduğunu belirtmiştir. Bu analiz, PoA'nın kurumsal uygulamalar için uygunluğunu destekler niteliktedir (Joshi, 2021). Lu vd., PoA'nın güvenlik açıklarına

çözüm getirmek amacıyla klonlama saldırılarına karşı geliştirilmiş bir model önermiştir. Çalışma, protokolün güvenliğini ve verimliliğini artıran bir yaklaşım sunarak, yüksek güven ve düşük hesaplama gereksinimi olan ortamlarda PoA'nın nasıl daha etkin kullanılabileceğini göstermektedir (Lu vd., 2024). Bu çalışmalar, PoA algoritmasının hem güçlü yanlarını hem de geliştirilmesi gereken yönlerini ortaya koymaktadır.

Yakma Kanıtı konsensüs algoritması, blok zincir ağlarında enerji tüketimini azaltmak ve güvenilir bir doğrulama süreci sağlamak amacıyla geliştirilmiştir. Bu mekanizma, kullanıcıların ağdaki belirli miktarda kripto parayı yakarak, yani erişilemez hale getirerek işlem doğrulama hakkı kazanmasını sağlamaktadır. PoB algoritması üzerine yapılan çalışmalar, bu sistemin avantajları, dezavantajları ve uygulama alanlarını geniş bir perspektiften ele almıştır. Menon, Saranya, Sureshbabu ve Mahesh, PoB'nin PoET (Proof of Elapsed Time) ve PoA (Proof of Authority) gibi diğer konsensüs algoritmaları ile karşılaştırmasını yapmışlardır. Çalışma, PoB'nin enerji verimliliği ve güvenlik açısından avantajlarını vurgularken, uygulama senaryolarına uygunluğunu değerlendirmiştir. Özellikle PoB'nin enerji tüketimini düşürme potansiyeli, diğer mekanizmalarla kıyaslandığında öne çıkan bir özelliktir (Menon vd., 2021). Rebello vd., PoB ve diğer kanıta dayalı konsensüs algoritmalarının güvenlik ve performansını analiz etmiştir. Bu çalışma, PoB'nin merkeziyetsizlik ve işlem hızını iyileştirme potansiyelini değerlendirirken, güvenlik açısından sağladığı avantajlara vurgu yapmıştır (Rebello vd., 2022). Vyas ve Deshmukh tarafından gerçekleştirilen çalışma, blok zincir teknolojisi ve konsensüs algoritmalarını genel olarak ele almıştır. PoB'nin düşük enerji tüketimi, maliyet etkinliği ve güvenilirliğiyle öne çıktığını belirten bu çalışma, PoB'nin kurumsal ve bireysel uygulamalarda nasıl etkili bir şekilde kullanılabileceğine dair kapsamlı bir değerlendirme sunmaktadır (Vyas & Deshmukh, 2023). Zhou vd., blok zincir konsensüs mekanizmalarının sistematik bir incelemesini yapmıştır ve PoB'nin küçük ölçekli projeler için uygunluğunu tartışmıştır. Çalışma, PoB'nin uzun vadeli sürdürülebilirlik açısından diğer mekanizmalarla olan farklılıklarını ele almıştır (Zhou vd., 2023).

Geçen Zaman Kanıtı, Intel tarafından geliştirilmiş ve Hyperledger Sawtooth platformunda kullanılan enerji verimliliği odaklı bir konsensüs algoritmasıdır. PoET, katılımcılara rastgele bir bekleme süresi atayarak, en kısa süreye sahip olanın yeni bloğu oluşturmasını sağlamaktadır. Bu yöntem, İş Kanıtı algoritmasına kıyasla enerji tüketimini önemli ölçüde azaltmakta ve çevre dostu bir alternatif sunmaktadır (Asadauskas, 2021).

PoET, adil lider seçimi sağlamak için donanım tabanlı güvenlik özellikleri kullanmaktadır. Intel'in Yazılım Koruma Uzantıları (Software Guard Extensions- SGX) sayesinde, bekleme sürelerinin manipülasyonu engellenmektedir (Chen vd., 2017). Bu yaklaşım, blok zincirinin güvenilirliğini artırmakla birlikte, merkeziyetçilik sorunu ortaya çıkarmıştır. Merkezi bir otorite tarafından sağlanan donanım güvenliği, sistemin bu otoriteye bağımlı hale gelmesine neden olabilmektedir (Rebello vd., 2022). PoET'nin işlem hızını artırması ve düşük enerji tüketimiyle ölçeklenebilirliği desteklemesi, bu algoritmayı birçok uygulama için cazip bir seçenek haline getirmektedir. Bununla birlikte, güvenlik açıkları ve merkeziyetçilik gibi kısıtlamaların giderilmesi, PoET'nin daha geniş bir kullanım alanı bulmasını sağlamaktadır (Bowman, Das, Mandal & Montgomery, 2021; Corso, 2019).

Önem Kanıtı, kullanıcıların ağdaki etkinliklerini ve katkılarını değerlendirerek, blok oluşturma ve işlem onaylama süreçlerinde daha fazla katılımı teşvik etmeyi hedeflemektedir. Bu algoritma, kullanıcıların sahip oldukları varlık miktarının yanı sıra, ağdaki işlem sıklığı ve diğer kullanıcılarla etkileşimleri gibi faktörleri de dikkate almaktadır. PoI'nin en bilinen uygulaması, NEM (New Economy Movement) blok zinciri platformunda görülmektedir. NEM, PoI algoritmasını kullanarak, ağın güvenliğini ve verimliliğini artırmayı amaçlamıştır (NEM, 2018). Bu yaklaşım, yalnızca varlık miktarına dayalı olan İş Kanıtı ve Hisse Kanıtı algoritmalarına alternatif olarak geliştirilmiştir. Sonuç olarak, Önem Kanıtı konsensüs algoritması, blok zinciri ağlarında daha adil ve katılımcı bir yapı oluşturmayı hedefleyen yenilikçi bir yaklaşımdır. PoI üzerine yapılan bilimsel çalışmalar sınırlı olmakla birlikte, bazı araştırmalar bu algoritmanın etkinliğini ve güvenliğini incelemiştir. Örneğin, Xiao vd., dinamik yetkilendirmeye dayalı bir Önem Kanıtı konsensüs algoritması önererek, ağın güvenliğini ve verimliliğini artırmayı hedeflemişlerdir.

Ripple Protokol Konsensüs Algoritması, Ripple ağı tarafından kullanılan Bizans Hata Toleranslı (BFT) bir konsensüs mekanizmasıdır. Bu mekanizma, benzersiz düğüm listesi (Unique Node List- UNL) üzerinden ağdaki düğümlerin fikir birliğine varmasını sağlamaktadır (Mundhra & Rebeiro, 2020) (Chase & MacBrough, 2018). RPCA'nın güvenliği, ölçeklenebilirliği ve performansı üzerine çeşitli araştırmalar yapılmıştır. Örneğin, Amores-Sesar, Cachin ve Mićić, protokolün güvenlik ve canlılık özelliklerini

analiz etmişlerdir. Ayrıca, Mundhra ve Rebeiro, RPCA'da hız ve güvenlik gibi konularda iyileştirmeler önermiştir.

Bunlara ek olarak, İtibar İspatı (PoR), Alan İspatı (PoSpace), Tarih İspatı (POH), Kiralanmış Hisse İspatı (LPoS), Rastgelelik İspatı (PoRand), Olasılık İspatı (PoP) birden fazla algoritmanın birleşiminden oluşarak, her birinin güçlü yönlerini kullanmayı amaçlan Hibrit konsensüs algoritmaları gibi algoritmalar da çeşitli makalelerde ele alınmış ve farklı yönleriyle incelenmiştir.

1.2. Tezin Amacı

Bu tez çalışmasında, mevcut konsensüs algoritmalarının incelenmesi ve bunların kısıtlarını aşacak yeni bir algoritmanın tasarlanması hedeflenmiştir. Blok zinciri teknolojisinin temel prensipleri ile konsensüs algoritmalarının birleştirilmesi sonucunda, ağ merkeziyetsizliğini artırmayı ve manipülasyonu azaltmayı amaçlayan bu tez, aynı zamanda yeni bir algoritmanın geliştirilme aşamalarını ve performans değerlendirmelerini de içermektedir. Bu çalışmanın sonuçları, blok zinciri teknolojisinde merkeziyetsizliği artıran ağın katılımcılarının neredeyse eşit söz sahibi olabildiği bu sebeple manipülasyonu azaltan daha demokratik sistemler tasarlamak için önemli bir adım niteliğinde olup, literatüre katkı sağlamayı hedeflemektedir.

1.3. Tezin Hipotezi

Blok zinciri teknolojisi, başlangıçta merkeziyetsizliği temel alarak şeffaf ve güvenilir bir dijital altyapı sunmayı hedeflemiştir. Ancak, zamanla geliştirilen yeni konsensüs algoritmaları, hız, ölçeklenebilirlik ve işlem kapasitesi gibi gereksinimlere yanıt vermek için merkeziyetsizlikten ödün verilmesine yol açmıştır. Örneğin Yetki Kanıtı algoritmasında doğrulayıcılar genellikle az sayıda güvenilir kişi veya kuruluş tarafından belirlenmektedir. Buna ek olarak, Yetkilendirilmiş Hisse Kanıtı algoritmasında yalnızca seçilen delegelerin yetki sahibi olması merkeziyet seviyesini düşürmektedir. Bu durum, blok zincirinin temel felsefesinden sapmalar göstermesine ve manipülasyona daha açık sistemlerin ortaya çıkabilmesine neden olmuştur. Bu tezde önerilen çift katmanlı konsensüs algoritmasıyla, merkeziyetsizlik ve güvenilirlik ilkeleri mümkün olduğunca korunarak hız ve güvenlik gibi gereksinimlere yanıt verebilecek bir çözüm sunulması amaçlanmıştır. Birinci katmanda, rastgele doğrulayıcı seçimiyle hızlı ve öngörülemez bir yapı oluşturulması, ikinci katmanda ise doğrulayıcılar arasında eşit oy hakkına dayalı

demokratik bir yapı sağlanması hedeflenmiştir. Önerilen bu çift katmanlı algoritmanın hipotezi, mevcut konsensüs algoritmalarında gözlemlenen sınırlamaları azaltabilecek bir alternatif sunabileceği ve blok zinciri teknolojisinin merkeziyetsizlik ilkesine daha yakın bir yapı sağlayabileceği yönündedir.

BÖLÜM 2

BLOK ZİNCİRİ TEKNOLOJİSİ

Blok zinciri teknolojisi, merkezi olmayan bir dijital defterdir ve birbirine bağlı bloklar şeklinde organize edilmiş veri yapıları kullanarak bilgilerin güvenli ve değiştirilemez bir şekilde saklanmasını sağlamaktadır. Her blok, önceki blokla kriptografik olarak bağlanmıştır ve bu bağ, zincirdeki verilerin değiştirilmesini veya manipüle edilmesini neredeyse imkânsız hale getirmektedir. Blok zinciri, Bitcoin gibi kripto paraların yanı sıra, elektronik oylama, tedarik zinciri yönetimi ve sağlık gibi birçok alanda kullanılabilecek potansiyele sahiptir (Zheng vd., 2017).

İlk olarak 2008 yılında Satoshi Nakamoto tarafından Bitcoin'in temel altyapısı olarak tanıtılan blok zinciri teknolojisi, 21. yüzyılın en önemli teknolojik yeniliklerinden biri haline gelmiştir. Blok zinciri teknolojisi, başlangıçta finans sektöründe devrim yaratmayı hedeflemiş olsa da merkeziyetsiz, şeffaf, izlenebilir ve güvenli yapısı sayesinde zamanla sağlık, tedarik zinciri yönetimi, bankacılık, tarım ve yönetim gibi farklı sektörlerde de benimsenmiştir. Blok zinciri sisteminin temel rollerinden biri, geleneksel sistemlerin aksine işlemleri merkeziyetsiz hale getirerek bir aracıya ihtiyaç duymadan taraflar arasında doğrudan işlem yapılmasını sağlamaktır. Konsensüs mekanizmalarıyla bu işlemler doğrulanmaktadır ve onaylanmaktadır. Bu özellik sayesinde birçok aracıyı ortadan kaldırarak işlem maliyetlerini düşürmektedir (Guo & Liang, 2016). Blok zinciri sisteminin önemli rollerinden biri, ağda gerçekleştirilen tüm işlemleri açık, şeffaf ve izlenebilir bir dijital kayıt sistemi olarak sunarak, verilerin değiştirilmesini veya manipüle edilmesini zorlaştırmasıdır. Bu özellik sayesinde blok zinciri, gizlilik, bütünlük ve erişilebilirlik gibi bilgi güvenliği hedeflerini de sağlamaktadır. Ayrıca, sistemin sürekli erişilebilir olması, işlemlerin kesintisiz bir şekilde gerçekleşmesine olanak tanımakta ve ağdaki veri güvenliğini bütüncül bir yapıda desteklemektedir (Crosby vd., 2016). Bölüm 2.1'de blok zincirinin yapısı ve çalışma sistemi detaylandırılmaktadır.

2.1. Blok Zincirinin Yapısı ve Çalışma Sistemi

Blok zinciri teknolojisi, her bir işlemin tanımlanması, doğrulanması, depolanması, paylaşılabilmesi için dijital olarak kaydedilmesi ve imzalanması gibi aşamaları barındırmaktadır. Adını, verilerin bloklar halinde saklanması ve bu blokların birbirine zincir gibi bağlanmasından almaktadır. Blok, verilerin toplandığı ve saklandığı temel birimdir. Bir blok, belirli bir veri veya işlem kümesini içermektedir ve blok zincirine eklenmektedir. Bu bloklar bir araya gelerek bir zincir oluşturulmakta ve bu zincirli yapı, verilerin güvenli ve şeffaf bir şekilde saklanmasını sağlamaktadır (Nakamoto, 2008).

Bir blok veri, zaman damgası (timestamp), önceki bloğa ait özet, blok özeti, işlemler gibi temel bileşenlerden oluşmaktadır. Veri bileşeni, belirli bir veri kümesini içermektedir. Örneğin, Bitcoin blok zincirinde her blok, belirli bir kaynağın belirli bir alıcıya ne kadar Bitcoin transfer ettiğine dair finansal işlemleri içermektedir. Zaman damgası bileşeni, bloğun hangi tarihte ve saatte oluşturulduğu bilgisini göstermektedir. Bu bilgiyle, blokların kronolojik olarak sıralanması sağlamaktadır. Blok özeti bileşeni, her bloğun kendine ait benzersiz bir özet değerini içermektedir. Bu bileşen, bloktaki tüm verilerin bir özetidir ve blok zincirinde o bloğun kimliğini belirlemektedir. Bu veriler, belirli bir kriptografik özet fonksiyonu kullanılarak işlenmekte ve blok için bir özet değeri üretilmektedir. En yaygın kullanılan özet fonksiyonlarından biri SHA-256'dır. Bitcoin gibi birçok blok zincirinde bu özet fonksiyonu kullanılmaktadır. SHA-256, girdiyi (bloğun tüm verilerini) alıp sabit uzunlukta (256 bit) bir çıkış değeri üretmektedir (Yaga vd., 2019). Örnek bir girdi ve çıkış değeri (özeti) Çizelge 2.1'de verilmiştir (md5hashing.net, t.y.).

Çizelge 2.1. SHA-256 özet fonksiyonu örnek bir girdi ve çıkış değeri

Düz Metin	Hesaplanmış Özet Değeri
Tuğba	041d8247d5ac3d953b71e8d37e88a116f77ca7e3273aea3bc74fbe2a16d408b1

Çıkış değeri şu şekilde hesaplanmaktadır:

- Blok verileri özet fonksiyonuna (örneğin, SHA-256) girdi olarak verilmektedir.
- Özet fonksiyonu, girdiyi işleyerek sabit uzunlukta bir özet değeri üretmektedir.

Örnek bir blok ve bir özet değeri üretimi aşağıdaki gibidir:

Blok verileri

- Blok numarası: 1001
- Önceki blok özet değeri: 0000abc123...,
- Zaman damgası: 2024-09-18 12:00:00
- İşlemler: Birkaç işlem verisi

Bu veriler birleştirildikten sonra, SHA-256 özet fonksiyonun girdisi oluşmuştur ve bu verilerin özet değeri hesaplanmıştır. Bu değer, örneğin aşağıdaki gibi hesaplanmıştır: 1a79a4d60de6718e8e5b326e338ae533...

Blok zincirinde özet değeri, veri bütünlüğünü koruma ve güvenliği sağlama açısından kritik bir işlev üstlenmektedir. Herhangi bir blok üzerinde yapılan bir değişiklik, o bloğun özet değerini değiştirmektedir. Bu nedenle blok zincirinde yapılan bir değişiklik, zincirin devamlılığını bozmakta ve fark edilmektedir. Bloklar, özet değerleri ile birbirine bağlı olduğu için eğer bir kişi önceki bir bloğu değiştirmeye çalışırsa, bu, tüm sonraki blokların özet değerlerini etkileyecek ve zincirde bir tutarsızlık yaratacaktır. Özet değeri, bu açıdan, blok zincirinin değiştirilemezliğini sağlamaktadır. Blok zinciri özet değerleri, zincirin güvenliğini sağlamak için kritik bir mekanizmadır ve blok zincirinin temel yapı taşlarından biridir (Zheng vd., 2017). Önceki bloğa ait özet bilgisi bileşeni ise, her bloğun bir önceki bloğa ait benzersiz bir kimliğinin bilgisini içermektedir. Bu bilgi sayesinde blokların birbirine zincir gibi bağlanması sağlanmaktadır. İşlemler bileşeni ise, blok içerisindeki işlemleri barındırmaktadır. Bu işlemler, kullanıcılar arasındaki dijital para transferleri veya dijital belgeler olabilmektedir. (Crosby vd., 2016).

Bir blok, blok başlığı ve işlem bloğu olmak üzere iki temel bileşenden oluşmaktadır (Yaga vd., 2019) ve Çizelge 2.2'de temel bir blok yapısı gösterilmiştir.

Çizelge 2.2. Temel bir blok yapısı

BlokID
Blok başlığı Blok özeti Önceki blok özeti Zaman damgası Nonce Zorluk Hedefi
İşlem bloğu İşlem1 İşlem2 . . . .

Blok zincirini oluşturan her blok, blok başlığındaki verilerin özeti olarak hesaplanan bir blok tanımlayıcısı (BlockID) taşımaktadır. Bloklar, ilk blok temel olacak şekilde üst üste katmanlanmakta ve blok zincirinin sonuna ulaşılana ve dizi tamamlanana kadar yükseklikleri büyümektedir. Zincirdeki ilk blok aynı zamanda "genesis bloğu" olarak da bilinmektedir.

Blok başlığı, blok meta verisini (üst verisini) içeren bir yapıya sahiptir. Bu başlık, 80 byte uzunluğunda bir dizi karakterden oluşmaktadır. İçeriğinde, 4 byte uzunluğunda bir Bitcoin sürüm numarası, 32 byte uzunluğunda bir önceki bloğun özet değeri, 32 byte uzunluğunda Merkle kökü, 4 byte uzunluğunda bir zaman damgası, 4 byte uzunluğunda blok için belirlenmiş zorluk hedefi ve 4 byte uzunluğunda madencilerin kullandığı nonce değeri (bir kez kullanılan sayı, rastgele sayı) bulunmaktadır. (Zheng vd., 2017). Bu bileşenler, her bloğun kimliğini ve bütünlüğünü güvenli bir şekilde korumak için kritik öneme sahiptir. Her bir bloğun birincil tanımlayıcısı, içerdiği kriptografik özettir. Esasen dijital bir parmak izidir ve blok başlığının uygulanabilir algoritma aracılığıyla iki kez karıştırılmasıyla oluşturulmaktadır. Bitcoin sürüm numarası, protokol boyunca değişiklikleri ve güncellemeleri takip etmek için kullanışlıdır. Önceki blok özeti, önceki bloğa veya üst bloğa bağlanmaktadır ve zinciri etkin bir şekilde güvence altına almaktadır. Merkle kökü, işlem içindeki tüm işlem özetlerinden oluşmakta, her bir özet işlemi daha fazla özet işlemine tabi tutulmaktadır (Crosby vd., 2016). Zaman damgası, proje üzerinde çalışan herkesin belirli bir olayın ne zaman gerçekleştiğine dair kalıcı, kodlanmış bir kaydı görebilmesi için dahil edilmiştir. Tipik olarak, söz konusu olay için günün tarihini ve saatini sağlamaktadır. Zorluk hedefi, basitçe, çalışan madencilerin

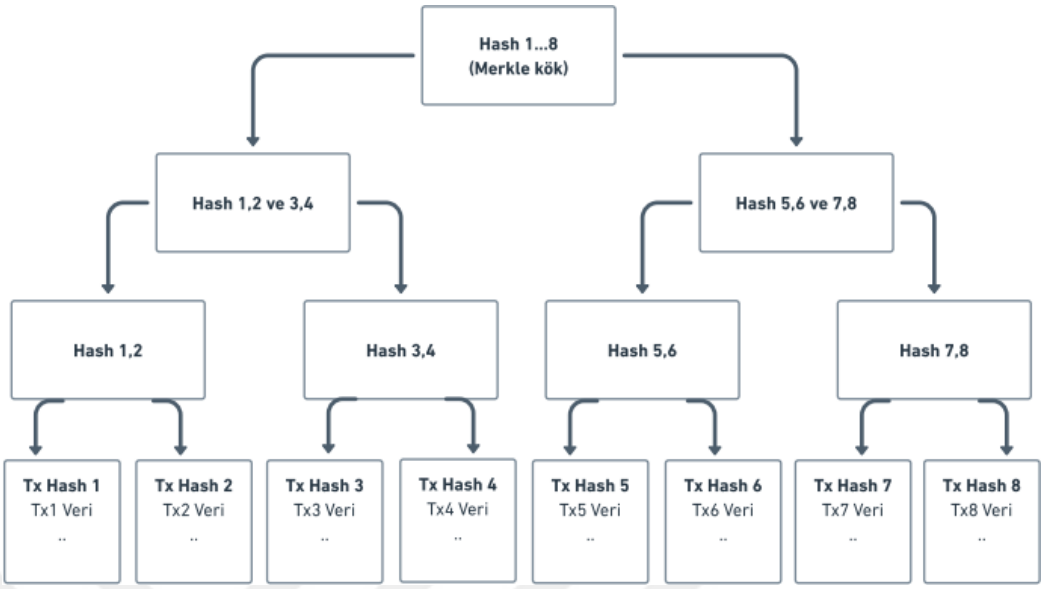
bloğu çözmenin ne kadar zor olduğunu ayarlamak için kullanılmaktadır. Son olarak nonce değeri, özellikle madencilik sürecinde, istenilen özet sonucunu elde etmek için sürekli değiştirilmekte ve yalnızca bir kez kullanılmak üzere tasarlanmaktadır. Nonce değeri, bir bloğun özeti istenen zorluk seviyesine ulaştırmak için gereken deneme sayısını temsil etmekte ve blok zincirinde güvenli, doğrulanabilir işlemler sağlamaktadır. Başka bir tabirle bu değer, madencilerin farklı permütasyonlar oluşturmak ve doğru bir özet oluşturmak için değiştirebilecekleri değerdir (Bezuidenhout, Nel & Maritz, 2022).

Bir blok zinciri sisteminin temel bileşenlerinden biri, kriptografik olarak birbirine bağlanmış ve sıralı bir işlem kaydı olarak işlev gören, dağıtılmış bir defterdir. Dağıtılmış defterin özellikleri, tarafların birbirlerine güvenmek zorunda kalmadan ve merkezi bir otoritenin müdahalesi veya gözetimi olmaksızın işlem kaydının tek bir doğru versiyonu üzerinde anlaşmalarını sağlamak için tasarlanmıştır. Bir blok zinciri sisteminin amacı, değişmez ve reddedilemez, güvenli, şeffaf ve erişilebilir olan dağıtılmış bir deftere işlemleri kaydetmektir. Bu sistem, eşler arası (peer-to-peer network) bir temelde çalışmaktadır ve katılımcıların blok zincirinin nihai doğru durumu üzerinde bir anlaşmaya varmaları için merkezi olmayan bir konsensüs (fikir birliği) mekanizması gereklidir (Bezuidenhout vd., 2022). Bloklar oluşturulduktan ve zincire eklendikten sonra bir blok zinciri ile etkileşime girmek isteyen herhangi bir paydaş bunu blok zincirince bir işlem başlatarak yapmaktadır. Bu süreç 3 temel adımda gerçekleşmektedir:

- 1.adım, işlem verilerinin oluşturulmasını içermektedir. Basitlik açısından, her işlemin, işlemi başlatan kullanıcının blok zinciri adresini içerdiği düşünülmektedir. Bu adres, genel anahtar şifreleme şemasında sahibinin genel anahtarını temsil etmektedir.
- 2.adım, işlem verilerinin özet hesaplamasını barındırmaktadır. TxHash olarak belirtilen bu özet, işlem verilerine eklenmektedir ve işlem için benzersiz bir tanımlayıcı olarak hizmet etmektedir.
- 3.adım, kullanıcının özel anahtarını kullanarak TxHash'i imzalamasını gerektirmektedir.

Anahtar oluşturma, şifreleme, şifre çözme, mesaj imza oluşturma ve imza doğrulama gibi aşamalar homomorfik şifreleme şemalarıyla yapılmaktadır. Homomorfik şifreleme, bir gizli şifre çözme anahtarına erişim gerektirmeden şifrelenmiş veriler

üzerinde hesaplamaların yapılmasına izin veren özel bir şifreleme tekniğini ifade etmektedir. Hesaplamaların sonuçları şifrelenmekte ve yalnızca gizli anahtarın sahibi tarafından açıklanabilmektedir. İmza, işlem verilerine eklenmektedir. Bu son adımın nedeni, herhangi bir tarafın, örneğin bir teklif verenin, ilk olarak işlemin adresin (AddrPubK) gerçek sahibi tarafından imzalandığını ve ikinci olarak işlem verilerinin bu süre boyunca değiştirilmediğini doğrulamasını sağlamaktır. İşlemler, sonunda blok zincirine kaydedildikleri blok zinciri ağı boyunca yayılmaktadır. Blok zinciri işlemleri, bir Merkle ağacındaki blok zincirine kaydedilmektedir. Merkle ağacı, her bir işlem verisinin kriptografik özetlerinin yapısal bir şekilde organize edildiği bir veri yapısıdır. Blok zincir sisteminde her işlem, bir özet değeri ile temsil edilmektedir. Bu özetler, çiftler halinde birleştirilip tekrar özet alınarak daha üst seviyedeki düğümler oluşturulmaktadır. Bu süreç, bir kök özet (Merkle Root) elde edilene kadar devam etmektedir. Bir Merkle ağacında, yaprak düğümleri, Şekil 2.1'de gösterildiği gibi her bir işlemin özetiyle (TxHash) tanımlandığı işlem verilerini içermektedir (Bezuidenhout vd., 2022). Yaprak düğümleri çiftler halinde gruplanmakta ve her bir kardeş çiftinin TxHash'i bir ana özet üretmek için özetlenmektedir. Bu işlem, tek bir kök özet olan Merkle kökü üretilene kadar özyinelemeli olarak tekrarlanmaktadır. Bu yapı, bloktaki tüm işlemlerin doğrulanabilirliğini sağlamakta ve verilerin değiştirilmediğini garanti etmektedir. Bloktaki herhangi bir işlemin geçerliliği ve değişmezliği, yalnızca işlem özetinin değil, Merkle ağacındaki özetlerin de kontrol edilmesiyle doğrulanabilmektedir. Bu nedenle, bir işlemin değiştirilip değiştirilmediğini anlamak için tüm ağaç boyunca özetlerin izlenmesi gerekmemektedir, sadece Merkle kökünden itibaren belirli bir yoldaki özetler kontrol edilerek işlem doğrulanabilmektedir.



Şekil 2.1. Merkle Ağacı

2.2. Blok Zinciri Türleri

Blok zinciri türleri temel olarak genel (public / permissionless), özel (private / permissioned), konsorsiyum (consortium) ve hibrit (hybrid) olmak üzere 4 ana grupta toplanmaktadır (Pilkington, 2016) (Zheng vd., 2017) ve alt başlıklar halinde açıklanmaktadır.

2.2.1. Genel Blok Zincirleri

Genel blok zinciri ağı, tüm katılımcıların ağa katılım, işlem gerçekleştirme ve blok zincirini okuma yetkisine sahip olduğu, merkeziyetsiz ve izin gerektirmeyen bir ağdır. Bu ağlar, herhangi bir merkezi otoritenin kontrolü olmadan çalışmaktadır ve tamamen şeffaf bir yapıya sahiptir. Herhangi bir kullanıcı, ağa katılarak düğüm olabilmekte ve ağın bir parçası olarak işlem yapabilmektedir. Ağa katılan herkes blok zincirini görebilmekte, doğrulayabilmekte ve blok zincirine yeni bloklar ekleyebilmektedir (Zheng vd., 2017). Tüm işlemlerin herkes tarafından izlenebilir oluşu tam bir şeffaflık sağlamaktadır. Ağdaki işlemler, kriptografik algoritmalar ve konsensüs mekanizmaları (örneğin, Proof of Work) kullanılarak doğrulanmaktadır. Bu sayede taraflar arasında güven inşa edilmekte ve güvene dayalı bir üçüncü taraf gereksinimi ortadan kaldırılmaktadır (Yaga vd., 2019).

Genel blok zinciri ağlarının merkeziyetsizlik, şeffaflık, yüksek güvenlik gibi avantajları vardır. Ağa katılan tarafların birbirine güven duymasına ihtiyaç yoktur, ağdaki işlemler matematiksel algoritmalar ve konsensüs mekanizmaları ile doğrulanmaktadır. Yapılan işlemlerin geri alınamaz ve değiştirilemez bir yapıda oluşu ise veri güvenliği ve bütünlüğü için önemlidir. Kriptografik algoritmalar kullanılarak işlemler güvence altına alınmaktadır. Ağa katılan kötü niyetli bir aktörün, ağdaki işlemleri manipüle etmesi, çoğunluk kontrolünü ele geçirmediği sürece oldukça zordur (Crosby vd., 2016). Ancak, bir veya daha fazla dürüst olmayan düğüm, ağın işlem gücünün %50'sinden fazlasını ele geçirirse, bu durumda çoğunluk üzerinde etkili hale gelip ağı yönlendirebilmekte ve manipüle edebilmektedir. Özellikle genel blok zincir ağları, bu %51 saldırısı olarak bilinen tehdit riski altındadır (Azeem, Mahmood, Aslam & Shahzadi, 2022). Bununla beraber, bazı genel blok zinciri ağlarının yüksek enerji tüketimi mevcuttur. Bu bazı genel blok zinciri ağları, genellikle düşük işlem hızlarına sahiptir ve ölçeklenebilirlik sorunlarıyla karşı karşıyadır. Her düğümün işlemleri doğrulaması gerektiği için, büyük bir ağda işlem başına harcanan süre artabilmektedir. Özellikle İş Kanıtı konsensüs algoritması kullanan ağlarda (örneğin Bitcoin ağı) madencilik işlemleri büyük miktarda enerji tüketmektedir (Zheng vd., 2017). Buna ek olarak, tüm işlemler herkes tarafından görülebildiğinden kişisel gizlilik konusunda endişelere yol açabilmektedir. İşlemler genel anahtarlar ile yapılmasına rağmen, bu anahtarlar ile kullanıcı kimlikleri ilişkilendirilebilmektedir (Yaga vd., 2019).

En bilinen genel blok zinciri ağı Bitcoin blok zinciri ağıdır (Nakamoto, 2008). Diğer bir genel blok zinciri olan Ethereum ağı ise belirli koşullar karşılandığında otomatik olarak yürütülen, kendi kendini doğrulayan ve blok zinciri üzerinde çalışan kod parçalar olan akıllı sözleşmeleri destekleyen bir platformdur. Ethereum ağı, dijital para birimi işlemlerinin gerçekleştirilmesine ek olarak merkeziyetsiz uygulamaların (DApps) geliştirilmesine de olanak tanımaktadır. (Buterin, 2013). Genel blok zinciri ağlarında en yaygın kullanılan konsensüs mekanizmaları Proof of Work (PoW) ve Proof of Stake (PoS) algoritmalarıdır (Zheng vd., 2017). Bölüm 3'te konsensüs algoritmaları detaylı olarak verilmektedir.

2.2.2. Özel Blok Zincirleri

Özel blok zinciri ağı, yalnızca belirli bir grup katılımcının ağa erişim, işlem yapma ve blok zincirine katkıda bulunma yetkisine sahip olduğu merkeziyetli veya izinli bir ağdır. Bu tür blok zincirleri, genellikle özel şirketler, kuruluşlar veya belirli bir grup tarafından kullanılmak üzere tasarlanmıştır ve genel blok zincirlerinden daha sınırlı ve kontrollüdür (Androulaki vd., 2018). Özel blok zincirleri, katılımcıları sınırlayarak daha fazla gizlilik ve işlem verimliliği sağlamayı amaçlamaktadır. Özel blok zincirleri, genellikle merkezi bir otorite veya kuruluş tarafından kontrol edilmektedir. Bu otorite, kimin ağa katılabileceğini, kimin işlem yapabileceğini ve kimin blok ekleyebileceğini belirlemektedir. Özel blok zincirlerine katılmak, işlem yapmak veya madencilik yapmak için özel izinler gereklidir. Her katılımcı, genellikle bir onay sürecinden geçmekte ve belirli rollere sahip olabilmektedir (Xu, Liu & Khan, 2020). Genel blok zincirlerine kıyasla daha az sayıda katılımcı olduğu için, işlem doğrulama süreçleri daha hızlıdır. Bu da ağın genel işlem kapasitesini artırmakta ve ölçeklenebilirlik sorunlarını en aza indirmektedir. Özel blok zincirlerinde işlem verileri genellikle daha gizlidir ve yalnızca ağa katılma izni olan kullanıcılar işlemleri görebilmektedir. Bu ağı kullanan, şirketler ve kuruluşlar için ağa katılma izni veri güvenliği açısından önemli bir avantaj sağlamaktadır (Nofer, Gomber, Hinz & Schiereck, 2017). Ayrıca, özel blok zinciri ağlarında genellikle enerji yoğun konsensüs mekanizmaları kullanılmamaktadır. Örneğin, İş Kanıtı yerine daha enerji verimli konsensüs algoritmaları, Practical Byzantine Fault Tolerance (PBFT) gibi, tercih edilebilmektedir (Xie, Cai, Zhang & Yu, 2019). Özel blok zincirlerinin avantajları arasında gizlilik, güvenlik, yüksek performans, ölçeklenebilirlik, kontrollü ortam ve daha düşük maliyetler yer almaktadır. Gizlilik ve veri güvenliği gereksinimi olan şirketler için, yalnızca belirli katılımcıların işlemleri görebildiği ve ağa katılabildiği bu yapılar önemli bir avantajdır (Nofer vd., 2017). Bu ağlarda daha az sayıda düğüm bulunduğundan, işlemler daha hızlı işlenmektedir. Ağın performansı genel blok zincirlerine göre çok daha yüksektir. Ayrıca, ağın ölçeklenebilirliği daha yönetilebilir düzeydedir. Ağdaki tüm katılımcılar, merkezi otorite veya kuruluş tarafından denetlendiği için, ağa katılan kötü niyetli aktörlerin olma olasılığı çok daha düşüktür. Bu da işlem güvenliğini artırmaktadır (Androulaki vd., 2018). Genellikle genel blok zincirlerinden daha düşük enerji tüketimine sahip oldukları için maliyetlerin azalmasına yardımcı olmaktadırlar. Ayrıca, ağdaki düğümler arasında güven sağlandığı için, ağır

madencilik süreçlerine ihtiyaç duyulmamaktadır (Xu vd., 2016). Bununla birlikte, özel blok zincirlerinin bazı dezavantajları da vardır. Genellikle merkezi bir otorite tarafından kontrol edilen ağlarda tam anlamıyla merkeziyetsizlik özelliği sunmadığı için, bazı kullanıcılar için güvenilirlik ve güven konularında endişelere yol açabilmektedir. İşlemler sadece izin verilen katılımcılar tarafından görülebilmesi sebebiyle genel blok zincirlerinde sunulan şeffaflığın eksik olmasına neden olmaktadır. Genel blok zincirlerinin aksine sansüre karşı daha az dayanıklıdırlar. Merkezi bir otorite, işlemleri engelleyebilmekte veya manipüle edebilmektedir. Özel blok zincirleri, yalnızca sınırlı sayıda katılımcıya izin verdiği için geniş katılımlı veya tamamen açık bir ekosistem oluşturulması için engel teşkil edebilmektedir (Johar, Ahmad, Asher, Cruickshank & Durrani, 2021).

En bilinen özel blok zinciri ağlarından biri olan Hyperledger Fabric, şirketlerin ve kuruluşların kendi izinli ağlarını kurmalarına olanak tanımaktadır. Bu platform, tedarik zinciri yönetimi ve finansal hizmetler gibi alanlarda yoğun olarak kullanılmaktadır. Hyperledger Fabric, modüler yapısı sayesinde işletmelere esneklik sunarak farklı kullanım senaryolarına uyum sağlayabilmektedir (Androulaki vd., 2018). Diğer bir özel blok zinciri ağı olan Corda R3 çeşitli bankalar ve finansal kuruluşlar tarafından kullanılan bir platformdur. İşlemlerin yalnızca ilgili taraflar arasında gerçekleşmesini sağlayarak gizliliği ön planda tutmaktadır. Corda, işlem doğrulama ve veri paylaşımını yalnızca gerekli taraflarla sınırlayarak yüksek güvenlik standartlarını korumaktadır (Nextrope, 2022). Son olarak Quorum ise JPMorgan tarafından geliştirilen Ethereum tabanlı bir özel blok zinciri platformudur. Quorum, gizlilik ve veri kontrolü gibi kurumsal ihtiyaçlara uygun bir şekilde tasarlanmıştır. Özellikle finans sektörü tarafından benimsenen bu platform, yüksek işlem hızı ve ölçeklenebilirlik avantajları sunmaktadır (Blockchain Council, 2022).

2.2.3. Konsorsiyum Blok Zincirleri

Konsorsiyum blok zinciri, bir grup kuruluşun veya birliğin ortak kontrolünde olan, katılımın sınırlı olduğu ve yalnızca izin verilen düğümlerin blokları doğrulayıp işlem gerçekleştirebildiği bir blok zinciri ağı türüdür. Özel blok zincirlerine benzer, ancak tek bir otorite yerine bir grup tarafından yönetilmektedir. Bu, güvenlik ve merkezsizleşme arasında bir denge sağlamaktadır. Yalnızca yetkilendirilmiş katılımcılar işlemleri

görebildiği ve ağ üzerinde işlem yapabildiği için özel blok zincirlerine göre daha fazla gizlilik sunmaktadırlar (Hussein, Salama & El-Rahman, 2023). Ayrıca, bu ağlarda daha hızlı ve enerji verimli konsensüs mekanizmaları kullanılmaktadır (Sun, Liu, Yue & Wang, 2020). Birden fazla kuruluşun ortak olarak ağa katılması, iş birliği ve denetim süreçlerinin daha etkili olmasını sağlamaktadır. Bir konsorsiyum blok zinciri, bir grup şirketin veya kuruluşun ortak iş süreçlerini güvenli ve şeffaf bir şekilde yürütmesini sağlamaktadır (Hussein vd., 2023).

Konsorsiyum blok zincirlerinin avantajları arasında merkezi olmayan kontrol, yüksek verim ve performans, gizlilik, güvenlik ve daha az enerji tüketimi yer almaktadır (Sun vd., 2020). Tek bir kuruluşun ağı kontrol etmemesi, ağa katılan tüm paydaşların dengeleyici bir kontrol mekanizmasına sahip olmasını sağlamaktadır. İzinli ve sınırlı sayıda katılımcı olduğundan, işlemler daha hızlı doğrulanabilmekte ve işleme kapasitesi yüksek olmaktadır. İşlemler yalnızca izin verilen katılımcılar tarafından görülebildiği için, özellikle gizlilik gerektiren işlemler için uygundur. Dezavantajları ise merkeziyetsizlik eksikliği, kısıtlı katılım ve güç dengesi sorunlarıdır. Ağ bir grup tarafından kontrol edildiği için özel blok zincirlerinin tam anlamıyla merkeziyetsiz yapısını sunmamaktadırlar. Yine yalnızca belirli üyeler tarafından kullanılabildiği için ağın dışındaki katılımcılar için erişimi sınırlamaktadır. Konsorsiyum blok zincirinde yer alan kuruluşlar arasındaki güç dengesizliği sorunlarına yol açabilmekte ve büyük kuruluşlar, ağı daha fazla kontrol edebilmektedir (Hussein vd., 2023). En bilinen Konsorsiyum blok zincir ağları Corda R3, EWF, B3i and Quorum ağlarıdır (Perera vd., 2020).

Özel blok zincirlerinde kullanılan enerji yoğun Proof of Work gibi konsensüs mekanizmalarına ihtiyaç yoktur. Bunun yerine, Practical Byzantine Fault Tolerance (PBFT) gibi daha verimli ve daha az enerji tüketen algoritmalar kullanılabilmektedir (Hussein vd., 2023). Bölüm 3'te konsensüs algoritmalarına detaylı değinilmektedir.

2.2.4. Hibrit Blok Zincirleri

Hibrit blok zinciri ağı hem izinli hem de izinsiz erişimlere olanak tanıyarak, açık ve özel blok zincirlerinin özelliklerini birleştiren bir blok zinciri türüdür. Bu ağlar, belirli işlemleri ve verileri halka açık olarak sunarken, diğer işlemler ve veriler yalnızca yetkilendirilmiş katılımcılar tarafından erişilebilmektedir. Bu esnek yapı, açık blok

zincirlerinin şeffaflık ve merkeziyetsizlik avantajlarını korurken, özel blok zincirlerinin gizlilik ve güvenlik özelliklerini de sunmaktadır (Roets, 2024; XinFin Network, 2024).

Hibrit blok zincirleri, verileri ikiye bölmekte; hassas veriler özel blok zinciri üzerinde işlenirken, doğrulama ve denetim için gerekli olan genel veriler açık blok zincirinde saklanabilmektedir (Kumari, Kumar & Dwivedi, 2024). Bu şekilde, sistemde gizlilik ihlali olmadan kamu tarafından doğrulama yapılabilmektedir. Açık blok zincirlerinde yaygın olarak görülen ölçeklenebilirlik ve hız sorunları, hibrit blok zincirlerinde özel bölümün verimliliği ile dengelenmektedir. Özel işlemler daha hızlı doğrulanırken, daha geniş şeffaflık gerektiren işlemler açık zincir üzerinden gerçekleştirilebilmektedir.

Hibrit blok zincirlerinin avantajları arasında gizlilik, güvenlik, şeffaflık, güven, daha hızlı ve düşük maliyetli işlemler ve kontrol edilebilir merkeziyet yer almaktadır. Özel blok zincirlerinin gizlilik avantajlarını koruyarak hassas veriler ve işlemler yalnızca izinli düğümler tarafından erişilebilir hale getirilmektedir. Genel blok zincirlerinin sunduğu şeffaflık ve merkeziyetsizlik özellikleri de korunmaktadır böylece kamuya açık işlemler herkes tarafından doğrulanabilmekte ve izlenebilmektedir (Roy & Shyamasundar, 2023). Özel bölümde yer alan işlemler, genellikle daha düşük maliyetle ve daha hızlı bir şekilde gerçekleştirilmektedir. Ağı kontrol eden yöneticiler, hangi verilerin ve işlemlerin kamuya açık olacağını, hangilerinin özel kalacağını belirleyebilmektedir. Bununla birlikte, hibrit blok zincirlerinin bazı dezavantajları da bulunmaktadır. Özel kısmı, genellikle bir kuruluş veya konsorsiyum tarafından kontrol edildiği için, tam anlamıyla merkeziyetsiz bir yapı oluşturmayı engelleyebilmekte ve belirli bir otoritenin müdahale etme olasılığını artırabilmektedir. Hibrit blok zinciri ağları hem açık hem de özel blok zincirleri teknolojilerini birleştirdiği için teknik bilgi ve uzmanlık gerektirmekte ve entegrasyon süreçlerini zorlaştırabilmektedir. Ayrıca, bu ağlar, tam şeffaflık sunmamaktadır. Özel bölümler yalnızca belirli katılımcılar tarafından erişilebilir olduğu için, kamu denetimi ve izleme kabiliyeti kısıtlı olabilmektedir.

En bilinen hibrit blok zinciri ağlarından biri olan Disney tarafından geliştirilen ve daha sonra açık kaynaklı hale getirilen Dragonchain, hibrit bir blok zinciri olarak tasarlanmıştır. Dragonchain, işletmelere işlemlerini özel tutma esnekliği sunarken, gerektiğinde bu işlemleri açık blok zinciri üzerinden doğrulama imkânı sağlamaktadır

(Roets, 2024). Diğer hibrit blok zinciri kullanan ağ ise XDC Network 'tür. XDC Network, hibrit bir blok zinciri olarak hem özel hem de açık blok zincirlerinin avantajlarını birleştiren bir platformdur. Kurumsal kullanım için optimize edilmiş XDC, finansal kuruluşlar için güvenli ve verimli bir blok zinciri ağı sunmaktadır (XinFin Network, 2024).

Hibrit blok zincirleri, her iki tür blok zincirinin avantajlarından faydalanarak konsensüs mekanizmalarını seçebilmektedir. Açık blok zincirlerinde enerji yoğun Proof of Work kullanılırken, özel bölümlerde daha verimli konsensüs algoritmaları kullanılabilmektedir. Hibrit blok zincirlerinde yaygın olarak Hisse Kanıtı ve Yetkilendirilmiş Hisse Kanıtı kullanılmaktadır (Venkatesan & Rahayu, 2024). Bölüm 3'te konsensüs algoritmalarına detaylı değinilmektedir.

2.3. Blok Zinciri Uygulama Alanları

Blok zinciri teknolojisi, genellikle finans sektörüyle ilişkilendirilse de sunduğu avantajlar sayesinde birçok sektörde köklü değişimler yapma potansiyeline sahiptir. Merkeziyetsizlik, güvenlik, şeffaflık ve verimlilik gibi özellikleriyle blok zinciri, verilerin güvenli bir şekilde depolanması, paylaşılması ve doğrulanmasını sağlayan, değiştirilemez bir dijital defter teknolojisidir (Tapscott & Tapscott, 2016). Bu teknoloji, özellikle verilerin güvenliği ve doğruluğunun kritik olduğu alanlarda önemli yenilikler sunmaktadır. Finans ve bankacılık sektöründe blok zinciri, sınır ötesi para transferlerinde aracılara olan ihtiyacı ortadan kaldırarak işlem sürelerini kısaltmakta ve akıllı sözleşmelerle finansal anlaşmaları otomatik hale getirmektedir (Catalini & Gans, 2016). Ayrıca, blok zincirinin şeffaf yapısı sayesinde daha iyi risk yönetimi ve denetim süreçleri sağlanmaktadır.

Tedarik zinciri yönetiminde blok zinciri, ürünlerin kaynağını ve kalitesini doğrulamayı kolaylaştırarak izlenebilirlik sağlamaktadır (Hazen, Boone, Ezell & Jones-Farmer, 2014). Sahtecilikle mücadelede ürünlerin orijinalliğini ve bütünlüğünü doğrulamak için kullanılabilmektedir. Ayrıca, tedarik zincirindeki tüm işlemlerin blok zinciri üzerinde tutulması, bilgi paylaşımını hızlandırmakta ve maliyetleri düşürmektedir.

Sağlık sektöründe blok zinciri, hasta verilerinin güvenli bir şekilde saklanması ve paylaşılması için kullanılabilmektedir. İlaç takibi ve tıbbi araştırmalarda verilerin

güvenilirliğini ve şeffaflığını artırarak sahte ilaçların önlenmesi ve araştırma sonuçlarının doğrulanması gibi konularda fayda sağlamaktadır (Huang & Foysal, 2021).

Gayrimenkul sektöründe blok zinciri, tapu sicil işlemleri, mülkiyet devri ve finansman süreçlerinin şeffaflığını artırmakta ve maliyetleri düşürmektedir (Kouhizadeh, Saberi & Sarkis, 2020). Bu sektörde blok zinciri teknolojisi tapu kayıtlarının güvenilirliğini artırmakta, mülkiyet devri süreçlerini hızlandırmakta ve akıllı sözleşmelerle satış ve kiralama işlemlerini otomatikleştirmektedir. Oylama sistemlerinde blok zinciri, seçmen katılımını, seçim güvenliğini ve şeffaflığını artırarak seçmenlerin oylarının doğru bir şekilde kaydedilmesini ve sayılmasını sağlamaktadır (Hajian Berenjestanaki, Barzegar, El Ioini & Pahl, 2024).

Sonuç olarak, blok zinciri teknolojisi, finans sektörünün ötesinde birçok farklı sektörde köklü değişim yaratma potansiyeline sahiptir. Güvenlik, şeffaflık, merkeziyetsizlik ve değiştirilemezlik gibi özellikleri, blok zincirini geleceğin dijital altyapısı haline getirmektedir. Daha fazla sektör bu teknolojiyi benimseyerek iş süreçlerini optimize etmekte, verimliliklerini artırmakta ve maliyetlerini düşürmektedir. Blok zincirinin kullanım alanları genişlemeye devam ettikçe, teknolojiye duyulan ilgi ve inovasyon potansiyeli de artmaktadır.

BÖLÜM 3

KONSENSÜS ALGORİTMALARI

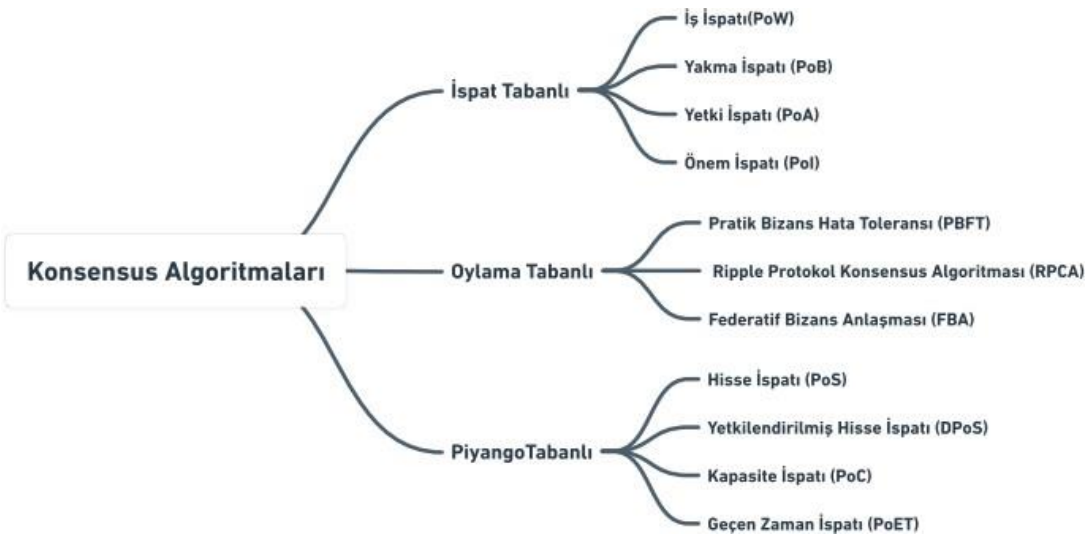
Konsensüs algoritmaları, blok zinciri ağlarının temel yapı taşlarından biridir. Merkezi bir otoritenin yokluğunda, blok zinciri ağındaki tüm düğümlerin aynı veri üzerinde uzlaşmalarını sağlayan mekanizmalardır (Dinh vd., 2018; Nakamoto, 2008). Bu algoritmalar, ağın güvenilirliğini ve veri bütünlüğünü garanti altına almaktadır. Blok zinciri ağlarında düğümler birbirinden bağımsız çalışmakta ve veri paylaşımı merkezi bir kontrol olmadan gerçekleşmektedir. Örneğin, farklı düğümlerde farklı veriler bulunabilmekte veya hatalı işlemler önerilebilmektedir. Konsensüs algoritmaları, tüm düğümlerin aynı veri üzerinde anlaşmalarını sağlayarak bu sorunları çözmekte ve işlemlerin doğruluğunu temin etmektedir (Zheng vd., 2017). Blok zincirinde işlemler, bloklar halinde gruplandırılmakta ve bu bloklar birbirine zincirlenerek değiştirilemez bir kayıt oluşturulmaktadır. Bu zincirin güvenilirliği ve bütünlüğü, her yeni bloğun eklenmeden önce düğümler tarafından doğrulanmasına dayanmaktadır (Cachin & Vukolić, 2017). Konsensüs algoritmaları, ağın bu süreci güvenli bir şekilde yönetmesini sağlamaktadır. Her blok eklenmeden önce düğümler arasında mutabakat sağlanmakta ve böylece blok zinciri sürekli olarak güvenli kalmaktadır (Zhang & Lee, 2019). Farklı blok zinciri sistemlerinin ihtiyaçlarına göre çeşitli konsensüs algoritmaları geliştirilmiştir. Her bir algoritma, kendi avantaj ve dezavantajlarına sahiptir ve özel kullanım senaryolarına uygun olarak seçilmektedir (Yaga vd., 2019). Bölüm 3.1'de mevcut konsensüs algoritmalarına detaylı bir şekilde değinilmektedir.

3.1. Konsensüs Algoritması Türleri

Konsensüs algoritmaları, bir blok zinciri ağında bulunan düğümlerin bir veri ya da işlem üzerinde aralarında fikir birliğine varmayı sağlayan yöntemlerden oluşmaktadır (Cachin & Vukolić, 2017). Bu algoritmalar, dağıtık sistemlerin güvenliğini sağlamak, kötü niyetli katılımcıları ya da faaliyetleri önlemek ve işlemlerin doğruluğunu garanti

altına almak için geliştirilmiştir (Dinh vd., 2018). Blok zincirinde güvenlik, merkeziyetsizlik ve ölçeklenebilirlik gibi temel unsurlar konsensüs algoritmaları sayesinde optimize edilmektedir (Zheng vd., 2017).

Konsensüs algoritmaları, rastgelelik mekanizmalarını kullanarak doğrulayıcı seçimi yapan piyango tabanlı algoritmalar, belirli bir kaynağın (örneğin, enerji, zaman veya depolama alanı) kullanımını veya harcanmasını kanıtlamaya dayanan kanıt tabanlı algoritmalar ve ağ katılımcılarının oy birliği ile anlaşmaya varmalarını temel alan oylama tabanlı algoritmalar olmak üzere 3 ana başlık altında incelenebilmektedir (Hossain vd., 2024). Bu sınıflandırmada bazı algoritmaların her iki sınıfın özelliklerini de barındırabileceğinden keskin bir ayrım söz konusu değildir. Örneğin Hisse Kanıtı algoritmasında daha fazla hisseye sahip olan doğrulayıcıların, blok doğrulama sürecine seçilme olasılığı daha yüksektir. Ancak bu tamamen kesin değildir; rastgelelik, herkesin bir şansa sahip olmasını sağlamaktadır. Dolayısıyla Hisse Kanıtı, rastgelelik kavramı barındırdığı için piyango tabanlı algoritmalar içinde sınıflandırılmaktadır. Yetkilendirilmiş Hisse Kanıtı, Hisse Kanıtı'nın bir türevi olması sebebiyle piyango tabanlı olarak sınıflandırılabilmekte ancak oylama mekanizmasına sahip olduğu için de oylama tabanlı sınıflandırmaya da dahil edilebilmektedir. Şekil 3.1'de konsensüs algoritmaları sınıflandırılması bu tasnifi görselleştirmektedir (Hossain vd., 2024).



Şekil 3.1. Konsensüs Algoritmalarının Sınıflandırılması

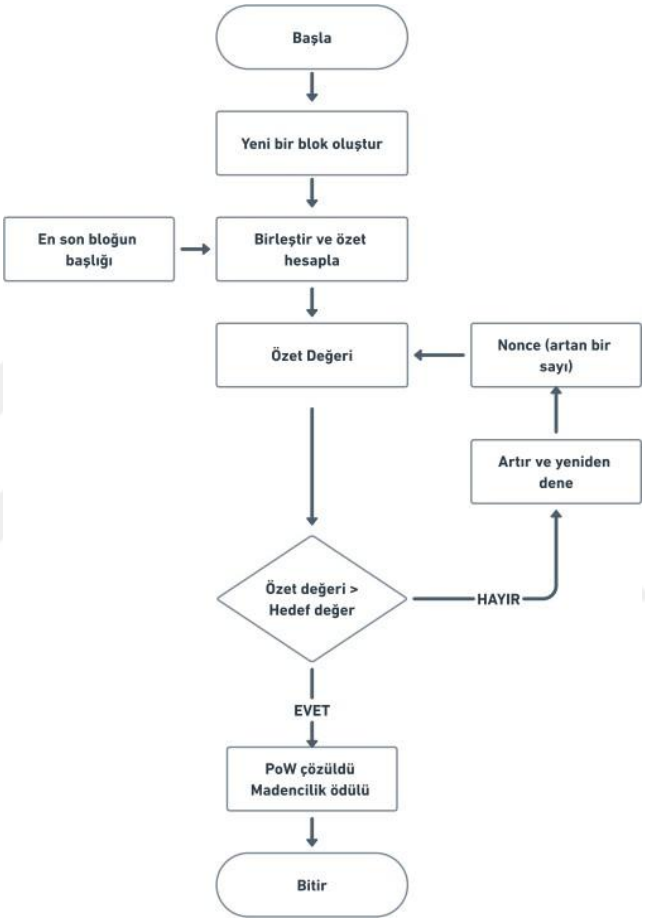
Bu algoritmaların her birinin güçlü ve kısıtlı olduğu yanları mevcuttur. Temel olarak ölçeklenebilirlik, performans ve güvenlik gibi blok zinciri ağlarının gereksinimlerine göre seçilip kullanılmaktadır (Yaga vd., 2019). Alt başlıklarda en çok bilinen ve yaygın olarak kullanılan konsensüs algoritmalarına dair detaylar ile literatürde daha az yer bulan diğer konsensüs algoritmaları birlikte verilmiştir.

3.1.1. İş Kanıtı Algoritması

Blok zinciri teknolojisinde, İş Kanıtı, merkezi olmayan bir ağda güvenliği ve fikir birliğini sağlamak için kullanılan önemli bir konsensüs algoritmasıdır. PoW fikri ilk olarak Dwork ve Naor tarafından 1993 yılında makalelerinde sunulmuştur. E-posta spam'iyle mücadele etmek için bu algoritmayı kullanma fikrini öne sürmüşlerdir (Dwork & Naor, 1993). Önerilerine göre, bir e-posta göndericisinin bir matematik bulmacasını çözmesi ve çözümü görevin yerine getirildiğinin bir kanıtı olarak e-postaya eklemesi gerekmiştir. E-posta alıcısı, yalnızca çözüm başarıyla doğrulanabiliyorsa bu e-postayı kabul etmiştir (Dwork & Naor, 1993). Blok zinciri ortamında buna benzer bir konsept benimsenmiştir.

İş Kanıtı mekanizması kanıtlayıcı ve doğrulayıcı olmak üzere iki farklı düğüm içermektedir. Kanıtlayıcı, bir hedefe ulaşmayı amaçlayan yoğun kaynak gerektiren bir hesaplama görevi gerçekleştirmektedir ve bunu çok daha az kaynak gerektiren doğrulama için bir doğrulayıcıya veya bir doğrulayıcı grubuna sunmaktadır (Yusoff, Mohammad & Anuar, 2022). PoW, temelde, ağdaki madencilerin karmaşık matematiksel problemleri çözmek için hesaplama gücü harcayarak (madencilik) yeni bloklar oluşturma ve doğrulama hakkını kazanmalarını gerektirmektedir. Bu süreçte, madenciler bir blok içindeki verileri almakta ve rastgele bir sayı olan "nonce" ile birleştirmektedir. Ardından, bu birleştirilmiş veriye kriptografik bir özet fonksiyonu uygulamaktadırlar (Nakamoto, 2008). Amaç, belirli bir koşulu örneğin, belirli sayıda sıfırla başlayan karşılayan bir özet değeri bulmaktır. Doğru özet değerini bulan ilk madenci, bloğu blok zincirine ekleme hakkını kazanmakta ve bunun için bir ödül (kripto para birimi değerinde) almaktadır (Eyal & Sirer, 2018). Şekil 3.2'de PoW algoritmasının akış şeması gösterilmiştir (Yusoff vd., 2022). Süreç, başlangıç aşamasıyla başlamakta ve yeni bir blok oluşturularak devam etmektedir. Bu yeni blok, mevcut zincirdeki son bloğun başlığıyla birleştirilmekte ve bu birleşim üzerinden bir özet değeri hesaplanmaktadır. Hesaplanan özet değeri, belirlenmiş

bir hedef değerle karşılaştırılmaktadır. Eğer özet değeri hedef değerden küçükse, bir nonce değeri artırılarak işlem yeniden tekrarlanmaktadır. Bu döngü, hedef değeri aşan bir özet değeri elde edilene kadar devam etmektedir. Özet değeri hedef değerden büyük olduğunda süreç başarıyla tamamlanmaktadır. Bunun sonucunda, PoW çözülmüş kabul edilmekte ve madenciye ilgili ödül tahsis edilmektedir. Tüm bu işlemlerin ardından süreç sonlandırılarak yeni blok zincire eklenmektedir.



Şekil 3.2. PoW algoritması akış şeması

PoW, ayrıca çift harcamayı (double spending), önleyerek işlemlerin kronolojik sırasını güvence altına almaktadır (Nakamoto, 2008). Çift harcama aynı dijital paranın birden fazla kez harcanması anlamına gelmektedir. Bir kullanıcı aynı parayı birden fazla kişiye göndererek birden fazla işlem gerçekleştirmeye çalışırsa dijital para biriminin güvenilirliği ve değeri tehlikeye girebilmektedir. Bitcoin gibi blok zinciri tabanlı sistemler, çift harcama sorununu çözmek için İş Kanıtı ve dağıtık yapısını kullanmaktadır (Bonneau vd., 2015). Her işlem, düğümler tarafından doğrulanmakta ve onaylandıktan sonra bir bloğa eklenmektedir. Bloklar birbirine zincirlenmekte ve en uzun zincir en

geçerli işlemleri içermektedir. Bu mekanizma, ağın çoğunluğu tarafından doğrulanmış olan işlemlerin zincire dahil edilmesini sağlamakta ve çift harcama yapılmasını önlemektedir. Bu algoritma, ağdaki düğümlerin İş Kanıtı ile çoğunluğu sağlayan bir zincir oluşturmalarını sağlamaktadır. PoW'un en önemli özelliklerinden biri, en uzun zincirin, en fazla İş Kanıtı yapılmış zincir olmasıdır. Bu yapı, düğümlerin işlem gücüne dayalı çoğunluk kararının temsil edilmesi anlamına gelmektedir. Dürüst düğümler, en uzun zinciri oluşturarak saldırganların oluşturduğu zincirleri geride bırakmakta ve onları geçersiz kılmaktadır.

PoW algoritmasının temel avantajları şunlardır:

- Merkeziyetsizlik: Herhangi bir merkezi otoriteye ihtiyaç duymadan, ağdaki tüm katılımcılar arasında fikir birliği sağlamaktadır.
- Güvenlik: Ağın %51'inden fazlasını kontrol etmedikçe, kötü niyetli kişilerin blok zincirini manipüle etmesini zorlaştırmaktadır.

Ancak, PoW algoritmasının bazı dezavantajları da vardır:

- Enerji Tüketimi: Karmaşık matematiksel problemleri çözmek için çok fazla enerji gerektirmekte, bu da çevresel endişelere yol açmaktadır.
- Ölçeklenebilirlik: İşlem hızları sınırlıdır ve ağ büyüdükçe daha da yavaşlayabilmektedir.
- 51% Saldırısı: Ağın %51'ini kontrol eden bir kişi veya grup, blok zincirini manipüle edebilmektedir (Yusoff vd., 2022).

PoW, Bitcoin gibi birçok popüler kripto para biriminde kullanılan köklü bir konsensüs algoritmasıdır. Ancak, enerji tüketimi ve ölçeklenebilirlik gibi dezavantajları nedeniyle, Hisse Kanıtı gibi daha enerji verimli ve ölçeklenebilir alternatifler geliştirilmektedir (De Vries, 2018).

İş Kanıtı Matematiksel Altyapısı

Tanım 3.1 (İş Kanıtı). PoW, bir problemi çözmek için hesaplama gücünü kullanarak bir "iş kanıtı" üretme sürecidir. Bu süreçte, belirli bir hedef özet değerini sağlayan bir nonce bulunmaktadır (Nakamoto, 2008).

Önerme 3.1 (Özet fonksiyonunun deterministikliği). *Bir özet fonksiyonu H(x) deterministik bir yapıya sahiptir; yani aynı girdi için her zaman aynı çıktıyı üretmektedir* (Schneier, 1996).

$$\forall x \in X, H(x) = y \; ve \; H(x') = y \; ise \; x = x' \tag{3.1}$$

Önerme 3.2 (Özet fonksiyonunun pseudo-rastgeleliği*). Bir özet fonksiyonunun çıktısı, girdi üzerindeki küçük değişiklikler için tamamen farklılaşmaktadır. Bu, özet fonksiyonunun rastgele görünümlü bir dağılım sağladığı anlamına gelmektedir* (Krawczyk, Bellare & Canetti, 1997).

$$x \neq x' \Longrightarrow Pr[H(x) = H(x')] \approx 0 \tag{3.2}$$

Önerme 3.3 (Zorluk koşulunun sağlanabilirliği). Özet fonksiyonunun çıktısının belirli bir zorluk seviyesini (d) sağlaması, özet fonksiyonunun hedef aralığında bulunması anlamına gelmektedir. PoW algoritması, özetin ilk d bitinin sıfır olmasını gerektirmektedir (Nakamoto, 2008).

$$Pr[H(x) < 2^{n-d}] = \frac{1}{2^d} \tag{3.3}$$

Önerme 3.4 (Ortalama çözüm süresi). *PoW algoritmasında, bir özet çıktısının zorluk koşulunu sağlaması için gereken nonce denemelerinin sayısı, 2d büyüklüğünde bir çözüm alanına sahiptir. Ortalama olarak* 2^{d-1}*deneme yapılması gerekmektedir* (Antonopoulos, 2017).

$$E[Deneme] = \mathbf{2^{d-1}} \tag{3.4}$$

Önerme 3.5 (Özet fonksiyonunun tek yönlülüğü). *Bir özet çıktısından (y) girdiyi (x) tahmin etmek, hesaplama açısından zordur* (Schneier, 1996).

$$\text{Verilen y} = \text{H(x), x'i bulmanın işlem karmaşıklığı O}(2^n)\text{'dir.} \tag{3.5}$$

Önerme 3.6 (Zorluk seviyesi ile enerji tüketimi arasındaki ilişki*). Zorluk seviyesi (d) arttıkça, PoW algoritmasının çözümü için gereken toplam işlem sayısı ve buna bağlı enerji tüketimi üstel olarak artmaktadır* (Antonopoulos, 2017).

$$T_{madencilik} \propto 2^d \cdot T_{özet} \tag{3.6}$$

Teorem 3.1 (Blok zinciri güvenliği). *PoW, bir bloğu değiştirmek veya sahte bir blok oluşturmak için gereken işlem gücünün, ağın toplam işlem gücünün %50'sinden fazlasını gerektirdiğini garanti etmektedir* (Nakamoto, 2008).

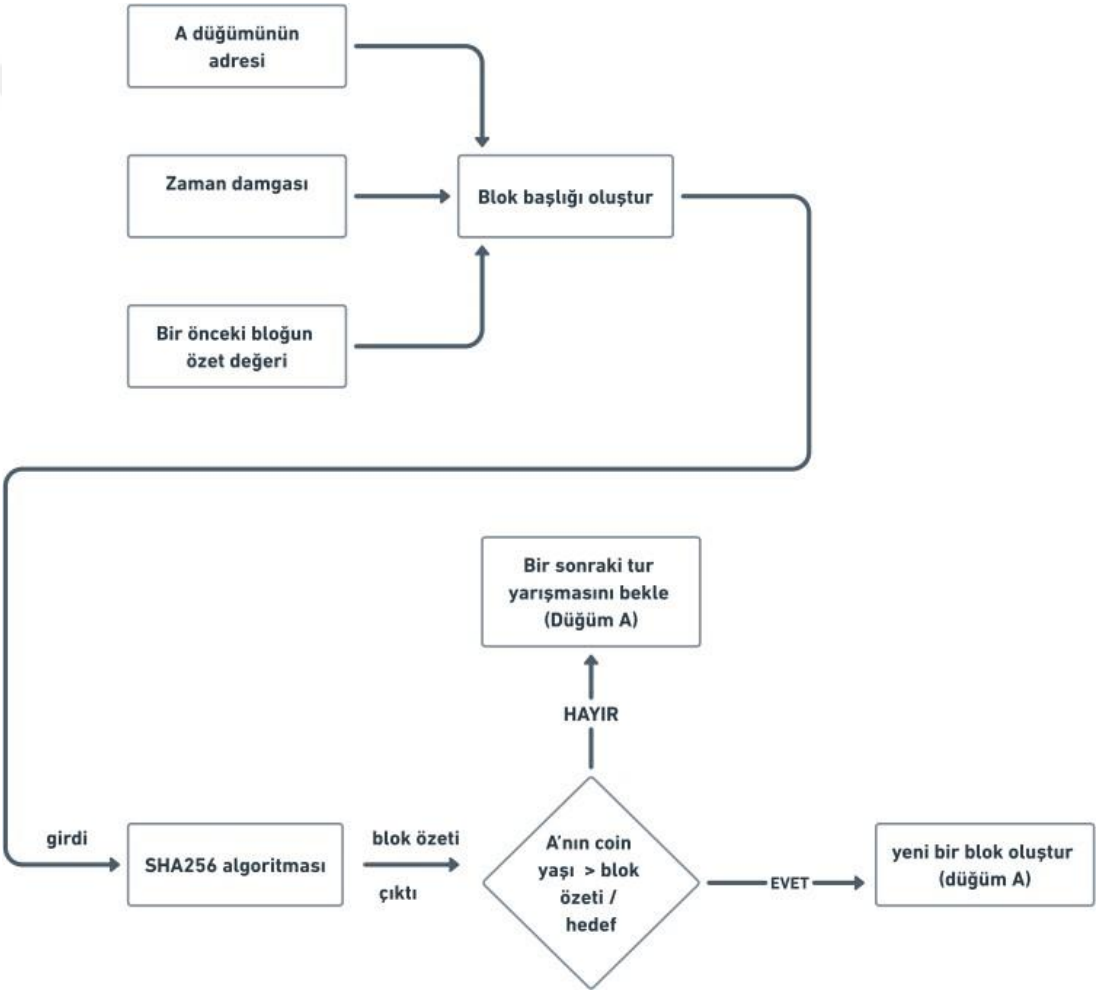
Bir saldırganın sahte blok zinciri oluşturma olasılığı: $\mathrm{P} < (1 - p)^t$ (3.7)

Kanıt. Ağın geri kalanı tarafından üretilen bloklar ile saldırganın üretebileceği bloklar arasındaki yarış, Poisson dağılımına dayanmaktadır (Antonopoulos, 2017). Saldırganın başarısız olma olasılığı, her bir blok doğrulama döngüsü için *(1 - p)* ile ifade edilir. Toplam başarısızlık olasılığı *t* denemeden sonra üstel olarak düşmektedir.

3.1.2. Hisse Kanıtı Algoritması

Hisse Kanıtı algoritması, blok zinciri ağlarında kullanılan bir konsensüs mekanizmasıdır ve İş Kanıtı algoritmasına alternatif olarak tasarlanmıştır. İş Kanıtı algoritmasında madenciler karmaşık matematik problemlerini çözerek blokları doğrulayarak yeni bloklar eklerken, Hisse Kanıtı algoritmasında blok doğrulayıcıları (validators) rastgele seçilmekte ve sahip oldukları kripto para birimi miktarına (stake) göre ağırlıklandırılmaktadır (Garcia Ribera, 2018). Blok oluşturma sürecine katılabilmeleri için, hisse (stake) adı verilen para birimlerinin belirli bir miktarını bir emanet hesabına kilitlemeleri gerekmektedir. Bu sistem, ağın güvenliğine katkıda bulunmayı taahhüt ettikleri anlamına gelmektedir. Yeni bir blok oluşturulacağı zaman, algoritma rastgele bir doğrulayıcı seçmektedir. Seçim, hisse edilen kripto para miktarına göre ağırlandırılmaktadır. Yani, daha fazla kripto para hisse eden birinin doğrulayıcı olarak seçilme olasılığı daha yüksektir. Seçilen doğrulayıcı, bloktaki işlemleri doğrulamakta ve bloğu blok zincirine eklemektedir. Doğrulayıcı, bloğu başarıyla doğrulayıp eklediği için bir ödül (kripto para birimi değerinde) almaktadır (Garcia Ribera, 2018; King & Nadal, 2012). Şekil 3.3'te PoS algoritmasının akış seması gösterilmiştir (Yusoff vd., 2022). Bu akış, bir düğümün yeni bir blok oluşturmak için nasıl rekabet ettiğini ve hangi adımları takip ettiğini göstermektedir. Öncelikle, düğüm adresi, zaman damgası ve önceki blok özet değeri bilgileri toplanmaktadır. Toplanan bu bilgiler kullanılarak bir blok başlığı oluşturulmaktadır. Bu blok başlığı, daha sonra özet girdi olarak verilmektedir. Blok başlığı, SHA-256 algoritması ile işlenmekte ve blok başlığının özet değeri elde edilmektedir. Bu özet değeri elde edilme aşaması, blok oluşturma sürecinin önemli bir parçasıdır. Sonraki aşamada, düğüm A'nın kripto para yaşı ile blok

özeti ve belirlenen bir hedef değere göre bir karşılaştırma yapılmaktadır. Kripto para yaşı, bir düğümün kripto para birimini ne kadar süre elinde tuttuğunu ifade etmektedir. Bu yaş, blok özet değeriyle birlikte değerlendirilmektedir. Karşılaştırma sonucunu olumsuz çıkarsa yani eğer düğüm A'nın kripto para yaş değeri "blok özeti / hedef değer" den küçükse, düğüm yeni bir blok oluşturamamakta ve bir sonraki tur için beklemektedir (Saleh, 2020). Bu durumda düğüm tekrar rekabete katılmak için beklemektedir. Karşılaştırma sonucu olumlu çıkarsa yani eğer düğüm A'nın kripto para yaş değeri, blok karması / hedef değerinden büyükse, bu düğüm yeni bir blok oluşturma hakkını kazanmakta ve yeni bir blok yaratmaktadır (King & Nadal, 2012; Yusoff vd., 2022).



Şekil 3.3. PoS algoritması akış şeması

PoS algoritmasının temel avantajları şunlardır:

Enerji Verimliliği: PoW'daki gibi yoğun hesaplama gücü gerektirmediği için çok daha az enerji tüketmektedir (Saleh, 2020).

Daha Hızlı İşlemler: Blok doğrulama süreci daha hızlı olduğu için işlemler daha hızlı gerçekleştirilmektedir (King & Nadal, 2012).

Ölçeklenebilirlik: PoW'a göre daha ölçeklenebilirdir, yani daha fazla işlem hacmini daha kolay yönetebilmektedir (Garcia Ribera, 2018).

Daha Güçlü Güvenlik: 51% saldırısı PoS sistemlerinde zordur çünkü saldırganın ağdaki toplam hissenin çoğunluğunu kontrol etmesi gerekmektedir (Saleh, 2020).

Ancak, PoS algoritmasının bazı dezavantajları da vardır:

Ağda daha fazla kripto paraya sahip olanlar, daha fazla hisse edebilmekte ve daha fazla ödül kazanabilmektedir. Bu durum da zenginlik eşitsizliğini artırabilmekte, tekelleştirmeye sebep olabilmektedir yani "zengini daha fazla zenginleştirebilme" riski taşımaktadır (Saleh, 2020).

Hisse edilen kripto paralar belirli bir süre için kilitlenmektedir, bu durum da likidite sorunlarına yol açabilmektedir (King & Nadal, 2012).

Hisse Kanıtı Matematiksel Altyapısı

Tanım 3.2 (Hisse Kanıtı). Hisse Kanıtı, bir katılımcının sahip olduğu hisseler oranında blok doğrulayıcı olarak seçilme olasılığına dayanan bir konsensüs algoritmasıdır. PoS, enerji tüketimi yerine hisse oranlarını temel almaktadır (King & Nadal, 2012; Saleh, 2020).

Blok Seçilme Olasılığı

Bir katılımcının, sahip olduğu hisselere (S_i) göre bir sonraki blok doğrulayıcı olarak seçilme olasılığı:

$$P_i = \frac{S_i}{S_{total}} \tag{3.8}$$

Burada:

- P_i: i-nci katılımcının seçilme olasılığı.
- S_i: i-nci katılımcının hisse miktarı.

- S_{total}: Ağdaki toplam hisse miktarı (King & Nadal, 2012).

Güvenlik ve Ağın Çoğunluk Kontrolü

PoS algoritmasının güvenliği, ağdaki toplam hisse'nin çoğunluğunu (%50'den fazlasını) elinde bulunduran katılımcıların dürüstlüğüne bağlıdır (Saleh, 2020).

Bir saldırganın ağın toplam hisselerinin en az yarısını ele geçirme maliyeti:

$$C_{attack} = \frac{S_{total}}{2} \times P_{token} \quad (3.9)$$

Burada:

- C_{attack}: Saldırıyı gerçekleştirmek için gereken toplam maliyet.
- P_{token}: Bir token'ın fiyatı (Bentov, Lee, Mizrahi & Rosenfeld, 2014).

Rastgelelik Mekanizması

Hisse sahipleri arasında rastgele bir seçim yapmak için şu şart kullanılmaktadır:

$$H(S_i, t) < \frac{D}{S_i} \quad (3.10)$$

Burada:

• $H(Si , t)$: i-nci doğrulayıcıya ait özet değeri.

• D: Zorluk seviyesi.

• t: Zaman damgası (Micali, 2017).

Ortalama Blok Üretim Süresi

Bir PoS ağındaki blok üretim süresi şu şekilde ifade edilmektedir:

$$T_{blok} = \frac{1}{\lambda . P_{valid}} \quad (3.11)$$

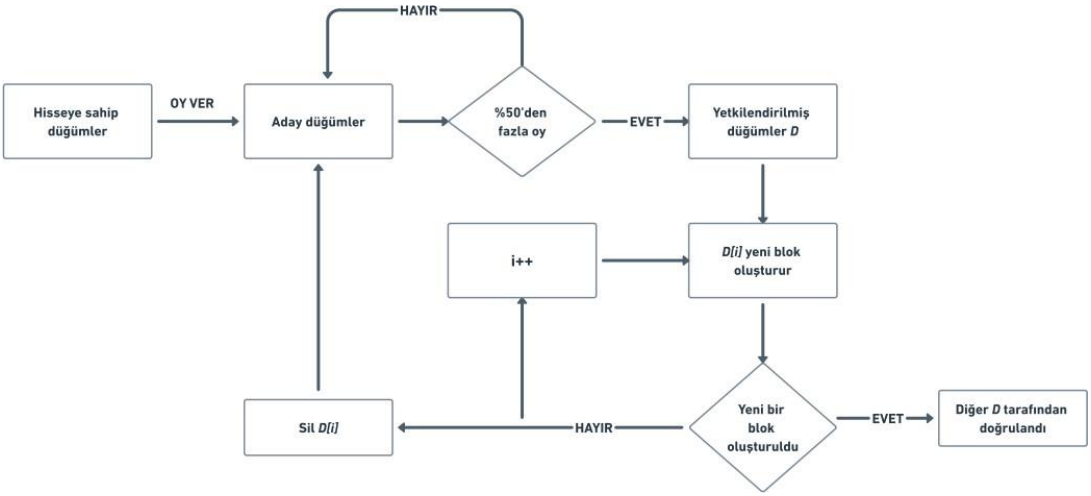
Burada:

- T_{blok}: Ortalama blok üretim süresi.
- λ: Ağdaki toplam doğrulayıcı sayısı.
- P_{valid}: Bir doğrulayıcının, belirli bir süre içinde blok üretme olasılığı (Gervais vd., 2016).

3.1.3. Yetkilendirilmiş Hisse Kanıtı

Yetkilendirilmiş Hisse Kanıtı, blok zinciri ağlarının güvenliğini sağlamak ve blok zincirindeki işlemleri doğrulamak için kullanılan bir konsensüs algoritmasıdır. DPoS, özellikle Bitcoin gibi PoW temelli sistemlerdeki verimsizliklere ve yüksek enerji tüketimi problemine çözüm bulmak amacıyla geliştirilmiştir. Bu sistemde, ağ katılımcıları tarafından seçilen bir grup blokları üretmekte ve onaylamaktadır. DPoS, oy verme mekanizmasına dayalı olarak çalışmaktadır. Daha merkeziyetsiz bir yapıya sahip olmasına ek olarak işleme hızı ve verimlilik açısından oldukça başarılı bir algoritmadır (Xu vd., 2020).

DPoS algoritması, katılımcıların oy kullanarak ağda blok üretme yetkisi olan grupları seçmesiyle çalışmaktadır. Ağdaki her katılımcı (token sahibi), sahip olduğu token miktarıyla orantılı olarak oy kullanma hakkına sahiptir. Yeterli oy alan gruplar, belirli aralıklarla blok üretme yetkisine sahip olmaktadırlar. Blok üreticileri, ağın işleyişini sürdürebilmek için işlemleri doğrulamakta ve yeni bloklar oluşturmaktadırlar. Bu sistem, oy alan bu grupların kötü niyetli davranmasını önlemek amacıyla dinamik bir yapıya sahiptir. Katılımcılar, kötü performans gösteren ya da kötü niyetli hareket eden grupları sistemden çıkarabilmekte ve yerlerine yenilerini seçebilmektedirler (Yang vd., 2019). Şekil 3.4'te DPoS algoritmasının aşamaları, akış şemasında gösterilmiştir (Yusoff vd., 2022).



Şekil 3.4. DPoS algoritması akış şeması

DPoS'in en önemli özelliklerinden biri, blok üretimi ve doğrulama süreçlerini hızlandırmasıdır. Blok üreticilerinin sayısının sınırlı olması, sistemin ölçeklenebilirliğini

artırmıştır ve ağın genel işlem hızını yükseltmiştir (Luo, Chen, Chen & Liang, 2018). Bu nedenle, DPoS algoritması, yüksek işlem kapasitesine sahip olan blok zinciri uygulamaları için ideal bir çözümdür.

DPoS algoritmasının temel avantajları şunlardır:

- Yüksek Verimlilik ve Düşük Gecikme Süresi: DPoS, PoW gibi yoğun hesaplama gerektiren algoritmalara kıyasla çok daha hızlıdır. Blok üreticileri belirli aralıklarla seçildiği için, blok üretimi ve doğrulama süreçleri hızlı bir şekilde gerçekleşmektedir (Sun vd., 2020).
- Enerji Verimliliği: PoW algoritmasında olduğu gibi enerji yoğun hesaplama işlemleri gerektirmemekte ve bu da enerji maliyetlerinin düşmesini sağlamaktadır (Xu vd., 2020). Böylece çevresel etkileri de PoW'a göre daha düşük olmaktadır.
- Merkeziyetsiz Yönetim: DPoS algoritması, tüm ağ katılımcılarına oy hakkı vererek merkeziyetsiz bir yapı sağlamaktadır. Bu sistem, kullanıcıların doğrudan karar verme süreçlerine katılmalarına olanak tanımakta ve böylece blok üreticileri topluluğun güvenini kazanmak zorunda kalmaktadır (Kang vd., 2019).
- Esnek ve Güncellenebilir Sistem: Gruplar, kötü niyetli ya da düşük performanslı olduğunda hızlı bir şekilde sistemden çıkarılabilmekte ve yerlerine yenileri seçilebilmektedir. Bu özellik, sistemin güvenliğini ve dayanıklılığını artırmaktadır (Yazdinejad vd., 2021).

Ancak, DPoS algoritmasının bazı dezavantajları da vardır:

- Merkeziyetleşme Riski: Her ne kadar DPoS merkeziyetsiz bir sistem olarak tasarlanmış olsa da blok üreticilerinin sayısının sınırlı olması ve token sahibi büyük katılımcıların daha fazla oy hakkına sahip olması, sistemde merkeziyetleşme riskini artırabilmektedir (Xu vd., 2020). Bu durum, özellikle büyük token sahiplerinin sistemi manipüle edebilmesine olanak tanımaktadır.
- Oylama Katılımı: DPoS sistemlerinde token sahiplerinin aktif olarak oy kullanması beklenmektedir. Ancak, düşük katılım oranları, blok üreticilerinin etkin bir şekilde seçilmemesine neden olabilmekte ve sistemin güvenliği tehlikeye girebilmektedir (Luo vd., 2018).
- Güvenlik Açıkları: DPoS, kötü niyetli grupların sistemin işleyişini bozmasına karşı bazı riskler barındırmaktadır. Özellikle az sayıda grup seçildiğinde, bu

grupların birbiriyle iş birliği yaparak blok doğrulama süreçlerinde manipülasyon yapma olasılığı bulunmaktadır (Yang vd., 2019).

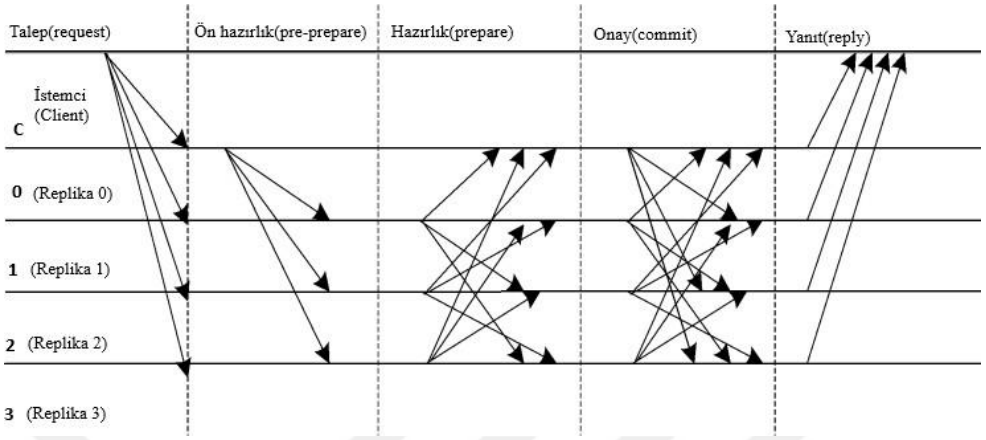
Sonuç olarak, DPoS algoritması, blok zinciri teknolojisinin daha verimli ve enerji dostu bir şekilde çalışmasını sağlayan önemli bir yeniliktir. Ancak, merkeziyetleşme riski ve güvenlik açığı gibi zorluklarla başa çıkmak için titizlikle uygulanması gerekmektedir. DPoS, özellikle hız ve verimlilik gerektiren blok zinciri uygulamaları için ideal bir konsensüs mekanizması sunmaktadır.

3.1.4. Pratik Bizans Hata Toleransı

Pratik Bizans Hata Toleransı algoritması, kötü niyetli saldırılara maruz kalabilen dağıtık sistemlerde güvenilirlik ve tutarlılık sağlama zorluklarını ele almak için tasarlanmış bir konsensüs mekanizmasıdır. PBFT, Castro ve Liskov tarafından 1999'da tanıtılmış olup, blok zinciri teknolojisi alanında temel bir algoritma olarak kabul edilmektedir. Bu algoritma, düğümlerin hatalı veya kötü niyetli davranabileceği sistemler için özel olarak geliştirilmiştir. PBFT, sistemdeki bazı düğümlerin hatalı olabileceği veya çıkarları doğrultusunda hareket edebileceği varsayımına dayanmaktadır. Algoritma, f kadar hatalı düğüm toleransına sahip bir ağda, en az n düğüm bulunması gerektiğini belirtmektedir ve $n \geq 3f + 1$ koşulunu gerektirmektedir. Yani, üç düğüm başına en fazla bir tanesi hatalı olabilmektedir. Protokol, hatalı düğümler olsa bile tüm sağlıklı düğümlerin işlem sırası konusunda fikir birliğine varmasını sağlamakta ve böylece dağıtık defterin bütünlüğünü korumaktadır (Sukhwani vd., 2017).

PBFT konsensüs süreci, üç ana aşamadan oluşur: ön hazırlık, hazırlık ve onay. Ön hazırlık aşamasında lider (primary) düğüm, diğer düğümlere yeni bir işlem bloğu (replica) önermektedir. Hazırlık aşamasında her düğüm, önerilen bloğu diğer düğümlere kabul ettiğini yaymaktadır. Düğüm, liderin önerisini aldığında, bu öneriyi onaylamak için diğer düğümlerle iletişime geçmektedir. Her düğüm, liderin önerisini alıp almadığını kontrol etmekte ve diğer düğümlerin onaylarını toplamaktadır. Onay aşamasında ise yeterli sayıda düğüm (en az $2f + 1$) bloğu onayladıktan sonra, bu blok zincire eklenmektedir. Bu üç aşamalı protokol, tüm kopyaların konsensüse ulaşmasını sağlamak için kritik

öneme sahiptir (Li vd., 2021; Wang vd., 2019). Şekil 3.5'te bu aşamalar algoritmanın akış şemasında gösterilmektedir (Yao, Ye, Murimi & Wang, 2021).



Şekil 3.5. PBFT algoritması akış şeması

PBFT'nin birçok avantajı bulunmaktadır. Bunlar arasında yüksek işlem hacmi, düşük gecikme süresi, güçlü hata toleransı ve madencilik gereksiniminin olmaması yer almaktadır. PBFT, geleneksel PoW mekanizmalarına göre daha hızlı işlem yapma kapasitesine sahiptir. Ayrıca, gerçek zamanlı işlem gerektiren uygulamalar için ideal bir çözüm sunmaktadır. Algoritma, kötü niyetli davranışları ve düğüm hatalarını etkili bir şekilde tolere ederek dağıtık sistemin bütünlüğünü korumaktadır. Bunun yanı sıra, PBFT enerji tüketimini azaltarak işletme maliyetlerini de düşürmektedir (Golan Gueta vd., 2019; Meshcheryakov, Melman, Evsutin, Morozov & Koucheryavy, 2021). Ancak, PBFT'nin bazı dezavantajları da bulunmaktadır. Özellikle ölçeklenebilirlik sorunları, uygulama karmaşıklığı, ağ gecikmesine duyarlılık ve kısıtlı hata toleransı gibi zorluklar söz konusudur. Düğüm sayısının artması durumunda iletişim yükü önemli ölçüde artmakta ve bu da PBFT'nin performansını bozabilmektedir. Ayrıca, algoritmanın lider düğüm bağımlılığı ve konsensüs sağlamak için minimum düğüm sayısına ihtiyacı, dinamik ortamlarda uygulanmasını zorlaştırmaktadır. Ağ gecikmeleri, PBFT'nin performansını olumsuz etkileyebilmektedir çünkü zamanında mesaj iletimi, konsensüs süreci için kritik öneme sahiptir. Hatalı düğüm sayısının üçte birden fazla olması durumunda ise konsensüs sağlama yeteneği etkilenebilmektedir. Bu durum her senaryoda kabul edilebilir olmayabilmektedir (Hao, Yu, Zhiqiang, Zhen & Dawu, 2018; Meshcheryakov vd., 2021).

Sonuç olarak, Pratik Bizans Hata Toleransı algoritması, dağıtık sistemler ve blok zinciri teknolojisi alanında önemli bir ilerleme temsil etmektedir. Hatalı düğümlerin varlığında konsensüs sağlama yeteneği, birçok uygulama için değerli bir seçenek olmasını sağlamaktadır. Ancak, PBFT'yi bir konsensüs mekanizması olarak seçerken ölçeklenebilirlik ve uygulama karmaşıklığı gibi hususların dikkate alınması gerekmektedir.

3.1.5. Federatif Bizans Anlaşması

Federatif Bizans Anlaşması algoritması, dağıtık sistemlerde güvenilir bir konsensüs sağlamak amacıyla tasarlanmış yenilikçi bir mekanizmadır. Geleneksel Bizans Hata Toleransı (BFT) algoritmalarının kısıtlamalarını aşmak için geliştirilmiştir ve özellikle genel blok zincirleri için daha uygun bir alternatif sunmaktadır (Kim, Kwon & Kim, 2019). FBA, kullanıcıların herhangi bir zamanda ağa katılabilmesine olanak tanırken, yalnızca diğer düğümler tarafından güvenilir kabul edilen katılımcıların konsensüs süreçlerine dahil olmasına olanak vermektedir (Fattahi, 2023). Bu özellik, ağın dinamikliğini ve ölçeklenebilirliğini artırmaktadır. FBA, her düğümün kendi güvenilir olarak tanımladığı bir alt küme olan çoğunluk dilimi (quorum slice) adı verilen bir grup düğümden mesaj alarak bireysel kararlar almasına olanak tanımaktadır (Nugroho, 2023). Konsensüs süreci, her düğümün belirli bir çoğunluk diliminden belirli mesajları alarak kendi kararlarını oluşturmasıyla başlamaktadır. Bu yapı, her düğümün kendi güvenilirliğine dayalı olarak bir karar almasını sağlarken, daha büyük bir grup içinde bu kararların doğruluğunu sağlamak için bir çoğunluk (quorum) gereklidir. Çoğunluk, her düğümün en az bir çoğunluk dilimine sahip olduğu ve bir araya gelerek karar alabilen düğümlerden oluşan bir grubu ifade etmektedir. Bu mekanizma, katılımcı düğümlerin çoğunluğunun onayını alarak konsensüs sağlamaya çalışırken, aynı zamanda ağın merkezi olmayan yapısını korumaktadır (Hayes, Aneiba, Gaber, Islam & Abozariba, 2023).

FBA'nın en önemli özelliklerinden biri, düğümler arasındaki güvenilirliğin bireysel düzeyde tanımlanabilmesidir. Düğümler, kendileri için güvenilir kabul ettikleri diğer düğümlerle iletişim kurarak, sistemin genel güvenliğini ve sürekliliğini sağlamak için gereken kararları almaktadır (Thanujan, Rajapakse & Wickramaarachchi, 2021). Bu yapı, ağda esneklik ve dayanıklılık sağlarken, aynı zamanda katılımcıların kendilerini

korumalarına ve kendi güvenilirliklerini artırmalarına yardımcı olmaktadır. FBA'nın temel bileşenlerinden biri olan çoğunluk dilimi kavramı, bir düğümün kendi kararlarını oluşturabilmesi için gereken güvenilir düğümler grubunu ifade etmektedir ve bu da algoritmanın etkin bir şekilde çalışabilmesi için kritik bir unsurdur (Trestioreanu vd., 2024).

FBA, sistemin genel güvenliğini artırmak ve karar alma sürecini hızlandırmak için farklı stratejiler ve algoritmalar kullanmaktadır. Bununla birlikte, çoğunluk dilimlerinin seçimi ve yapılandırılması, sistemin güvenliğini ve işleyişini doğrudan etkileyen önemli bir faktördür (Fattahi, 2023). Bu nedenle, hatalı yapılandırmalar ağın güvenliği için tehdit oluşturabileceğinden ağ tasarımcıları ve yöneticileri tarafından çoğunluk dilimi yapılandırmaları dikkatli seçilmeli ve izlenmelidir. FBA algoritması, dağıtık sistemler için güçlü bir konsensüs mekanizması sunarken, yapılandırma ve yönetim açısından titiz bir yaklaşım gerektirmektedir (Nugroho, 2023).

FBA'nın sunduğu bu avantajların yanı sıra, ağın güvenliği için çoğunluk dilimi grubunun doğru yapılandırılması, yönetilmesi ve sistemin genel performansına olan etkileri gözetilmelidir (Kim vd., 2019).

3.1.6. Kapasite Kanıtı

Geleneksel olarak kullanılan İş Kanıtı ve Hisse Kanıtı algoritmaları, enerji tüketimi ve merkeziyetçilik gibi bazı sınırlamalara sahiptir. Bu sınırlamaları aşmak amacıyla geliştirilen Kapasite Kanıtı algoritması, madencilik sürecinde depolama alanını kullanarak daha enerji verimli ve erişilebilir bir alternatif sunmaktadır. Kapasite Kanıtı, İş Kanıtı algoritmasına benzemektedir ancak madencilerin işlem gücüne değil, sabit disk kapasitelerine dayanmaktadır. Bu nedenle, PoW'da kullanılan ASIC (Application-Specific Integrated Circuit) madenciliğine kıyasla önemli ölçüde daha enerji verimlidir. PoC'deki blok oluşturma süresi 4 dakikadır. PermaCoin ve SpaceMint, PoC kullanan iki kripto para birimidir (Salimitari & Chatterjee, 2018). PoW'da özet işlemlerini tekrar etmek yerine, PoC'de düğümler madencilik etkinliği başlamadan önce sabit disklerini boşaltarak olası çözümlerin listesini depolamaktadırlar. Sonuç olarak, sistem daha hızlı çalışmaktadır.

PoC, iki ana bileşene sahiptir:

- Plotlama (plotting)

- Madencilik (mining)

Plotlama(çizim) aşamasında, madenciler sabit disklerinde kriptografik özet fonksiyonları kullanarak olası çözüm değerlerini önceden hesaplamakta ve depolamaktadırlar. Bu değerler, gelecekteki madencilik işlemlerinde kullanılmak üzere saklanmaktadır. Bu konsensüs algoritması, aynı zamanda Alan ve Zaman Kanıtı (Proof of Space and Time) olarak da adlandırılmaktadır (K & S, 2023).

Kapasite Kanıtı mekanizmasını kullanan BURST, büyük miktarda rastgele veri depolamayı ve bunları işlemeyi gerektiren bir Kapasite Kanıtı sistemine örnektir (Debus, 2017). BURST, işlemci (CPU) veya grafik kartı (GPU) yerine kullanılmayan sabit disk alanını madencilik için kullanmaktadır. Madenciler, çizim (plot) olarak bilinen veri parçalarını önceden oluşturmakta ve bunları diske kaydetmektedir. Depolanan çizim sayısı, madencilik hızını belirlemektedir. Her blokta madenci, kaydedilen çizimleri taramakta ve başka bir blok henüz bulunmadıysa bir blok çıkarabilmesi için gereken süreyi belirlemektedir. Çizimlerin yalnızca bir kez, bilgisayarın işlemcisi (CPU) veya grafik kartı (GPU) ile oluşturulması gereklidir, bu da POC algoritmasını donanım ve enerji açısından verimli kılmaktadır (Burstcoin, t.y.).

PoC algoritmasının avantajlarının yanı sıra bazı dezavantajları da bulunmaktadır. Bu konsensüs sistemi, büyük miktarda sabit disk alanı gerektirdiği için depolama verimsizliğine yol açabilmekte ve bu alan başka bir amaçla kullanılamamaktadır. Ayrıca, büyük sabit disklere sahip olan sistem aktörleri avantajlı hale gelerek ağdaki merkeziyetsizlik ilkesine zarar verebilmektedir. Güvenlik açısından, çizim dosyalarının manipüle edilmesi ve kötü amaçlı yazılımların boş disk alanlarını hedef alması tehdit oluşturmaktadır (Investopedia, 2024; MakeUseOf, 2024).

3.1.7. Yetki Kanıtı

Yetki Kanıtı algoritması, özel blok zincir ağlarında kullanılan ve belirli doğrulayıcıların (otoriteler) blok oluşturma ve onaylama işlemlerinden sorumlu olduğu bir konsensüs mekanizmasıdır. Bu algoritma, enerji verimliliği ve işlem kapasitesi açısından İş Kanıtı gibi yoğun hesaplama gücü gerektiren mekanizmalara göre önemli avantajlar sunmaktadır. PoA, özellikle otorite güvenilirliğinin ön planda olduğu ve tüm katılımcıların önceden doğrulandığı özel ve konsorsiyum blok zincir ağları için tasarlanmıştır (Islam, Merlec & In, 2022).

PoA algoritması, otoriteler arasında blok oluşturma görevlerinin sırayla paylaşılması üzerine kuruludur. Bu modelde her otorite, belirli bir sırayla blok önerme hakkına sahip olmaktadır. Bu süreç, merkezi olmayan bir güven modeli sunmakla birlikte, ağın güvenilir bir çoğunluğa dayanmasını gerektirmektedir. PoA'nın Aura ve Clique gibi popüler uygulamaları, farklı işleyiş özellikleri ve avantajlar sunmaktadır. Aura algoritması, lider tarafından bir bloğun önerilmesini ve diğer otoritelerin bu bloğu kabul etmesini gerektirirken Clique, aynı anda birden fazla otoritenin blok önermesine olanak tanımaktadır. Aura modeli, tutarlılığı önceliklendirirken daha fazla iletişim yükü oluşturabilirken, Clique modeli daha düşük gecikme süreleri sağlamakla birlikte geçici çatallanmaların ortaya çıkmasına yol açabilmektedir (Yang, Dai, Gooi, Nguyen & Paudel, 2022).

PoA algoritmasının temel avantajları arasında enerji verimliliği, yüksek işlem kapasitesi, deterministik blok oluşturma ve artırılmış güvenlik bulunmaktadır. Enerji verimliliği açısından, karmaşık matematiksel problemlerin çözülmesini gerektirmeyen bu algoritma, özellikle mikro şebekeler gibi enerji kısıtlamalarının ön planda olduğu sistemler için uygundur. Ayrıca, PoA algoritması saniyede 20.000 işlem kapasitesine ulaşarak, yüksek işlem hacmi gerektiren uygulamalar için güçlü bir çözüm sunmaktadır. Blok üretiminde belirsizliği ortadan kaldıran deterministik yapısı, ağdaki katılımcılar arasında eşit fırsatların sağlanmasına olanak tanımaktadır. Aynı zamanda, özel ağlarda otorite kimliklerinin belirgin olması, kötü niyetli faaliyetlerin caydırılmasında önemli bir rol oynamaktadır (Asad, Elahi, Hasan & Yousuf, 2020).

Bununla birlikte, PoA algoritmasının dezavantajları da bulunmaktadır. Merkeziyetçilik riski, algoritmanın yalnızca sınırlı sayıda doğrulayıcıya dayanmasından kaynaklanmaktadır. Bu durum, otoritelerin iş birliği yaparak ağın güvenliğini tehlikeye atabileceği anlamına gelmektedir. Ayrıca, PoA, blok zincir teknolojisinin merkeziyetsizlik ilkesiyle tamamen uyumlu değildir, çünkü otoritelere duyulan güven sistemin temel dayanağıdır. Daha büyük ağlarda, iletişim yükünün artması ölçeklenebilirliği sınırlayabilmektedir. Clique modelinde ise geçici çatallanmaların çözülmesi için ek mekanizmalara ihtiyaç duyulması başka bir kısıtlama olarak öne çıkmaktadır (Islam vd., 2022).

PoA algoritması, enerji verimliliği ve hesap verebilirlik gereksinimlerinin ön planda olduğu uygulamalarda etkili bir çözüm sunmaktadır. Sağlık verilerinin güvenli bir şekilde paylaşımı, mikro şebekelerin siber saldırılardan korunması ve finansal hizmetler gibi özel uygulama alanlarında kullanımı yaygındır. Ancak, merkeziyetçilik riski ve ölçeklenebilirlik kısıtlamalarının göz önünde bulundurulması gerekmektedir.

3.1.8. Yakma Kanıtı

Yakma Kanıtı algoritması, blok zincir ağlarında enerji tüketimini azaltmak ve güvenliği artırmak amacıyla geliştirilmiş bir konsensüs mekanizmasıdır. PoB, kullanıcıların belirli bir miktar kripto parayı geri alınamaz şekilde yakmalarını (yani, erişilemeyen bir adrese göndermelerini) gerektirmektedir. Bu işlem, kullanıcılara ağda yeni blok oluşturma ve ödül kazanma hakkı tanımaktadır. Yakılan kripto para miktarı arttıkça, kullanıcının yeni bir blok oluşturma olasılığı da artmaktadır (Khobragade & Turuk, 2023). PoB algoritmasının çalışma prensibi, kullanıcıların mevcut kripto paralarını feda ederek ağın güvenliğine katkıda bulunmalarına dayanmaktadır. Bu yaklaşım, enerji yoğun İş Kanıtı algoritmasına alternatif olarak geliştirilmiştir. PoW'da madenciler, karmaşık matematiksel problemleri çözmek için yüksek enerji tüketen donanımlar kullanırken, PoB'da enerji tüketimi önemli ölçüde düşmektedir (Katal, Sethi & Lamba, 2020). Tripathi, Jain, Kumar, Singh ve Shukla, PoB'nin enerji verimliliği açısından önemli bir gelişme sunduğunu ve çevresel sürdürülebilirliği desteklediğini ifade etmektedir (Tripathi vd., 2024).

PoB algoritmasının avantajları arasında düşük enerji tüketimi ve çevre dostu olması bulunmaktadır. Ayrıca, kullanıcıların ağın güvenliğine doğrudan katkıda bulunmalarını teşvik etmektedir. Ancak, bazı dezavantajları da mevcuttur. Örneğin, PoB, varlıklı kullanıcıların daha fazla kripto para yakarak ağ üzerinde daha fazla kontrol sahibi olmalarına yol açabilmektedir. Bu durum, ağın merkeziyetsizlik ilkesine aykırı olabilmektedir (Hussein vd., 2023). Ayrıca, yakılan kripto paraların geri alınamaması, kullanıcılar için finansal risk oluşturabilmektedir.

Sonuç olarak, Yakma Kanıtı algoritması, blok zincir ağlarında enerji verimliliğini artırmak ve güvenliği sağlamak için geliştirilmiş bir konsensüs mekanizmasıdır. Avantajları ve dezavantajları göz önünde bulundurularak, belirli kullanım senaryolarında PoB, PoW'a alternatif olarak değerlendirilebilmektedir.

3.1.9. Geçen Zaman Kanıtı

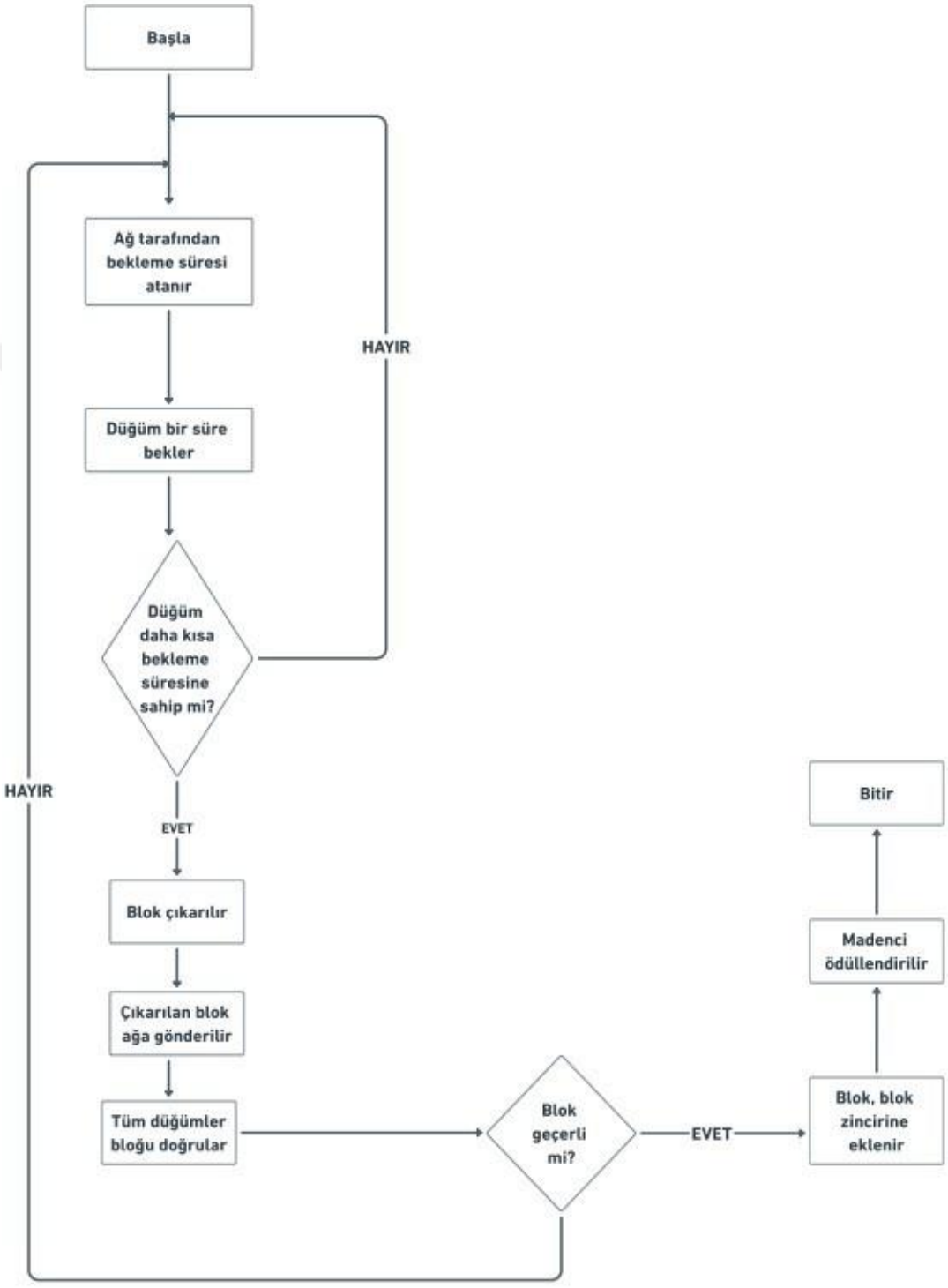
Geçen Zaman Kanıtı, özellikle özel blok zincirlerinde enerji verimliliği sağlayan, lider seçimini rastgeleleştiren ve güvenli donanım teknolojilerine dayanan bir konsensüs algoritmasıdır. POET, katılımcı düğümlere rastgele bir bekleme süresi atayarak çalışmaktadır. Bu süre sonunda bekleme süresi ilk dolan düğüm, lider olarak seçilmektedir. Bu süreç, Intel SGX gibi Trusted Execution Environment (TEE) teknolojileri tarafından doğ rulanarak manipülasyonlar engellenmekte ve lider seçiminin adil olması sağlanmaktadır (Bowman vd., 2021; Chen vd., 2017).

Algoritmanın başlıca avantajları arasında enerji tasarrufu, adil lider seçimi, yüksek işlem verimliliği ve düşük güç tüketimiyle ölçeklenebilirlik yer almaktadır. Bu özellikler, POET'i özellikle enerji verimliliğinin ve performansın kritik olduğu kurumsal blok zinciri uygulamaları için uygun hale getirmektedir (Asadauskas, 2021; Corso, 2019). Bununla birlikte, algoritmanın Intel SGX gibi özel donanımlara bağımlılığı, sınırlı erişim ve donanım açıkları nedeniyle güvenlik riskleri yaratabilmektedir. Ayrıca, POET'in genellikle özel blok zincirlerde kullanılabilir olması, tamamen merkeziyetsiz sistemlerde uygulama alanını sınırlandırmaktadır (Chen vd., 2017; Corso, 2019).

POET, genel olarak, enerji verimliliği ve adil lider seçimi konularında öne çıkan bir konsensüs algoritmasıdır. Ancak, donanım güvenliği teknolojilerindeki ilerlemeler, bu algoritmanın güvenlik açıklarını giderebileceği ve daha geniş bir uygulama yelpazesi sunabileceği anlamına gelmektedir.

POET algoritmasının çalışma süreci, yeni bir lider seçimi ve blok oluşturma süreciyle başlamaktadır. Ağ, her bir düğüme rastgele bir bekleme süresi atamakta ve düğümler, bu süre boyunca beklemektedir. Bekleme süresi, güvenilir bir yürütme ortamında (TEE) doğrulanmaktadır. Bekleme süreleri sona erdiğinde, düğümler arasında en kısa süreye sahip olan belirlenmekte ve bu düğüm lider olarak seçilmektedir. Lider seçilen düğüm, bir blok oluşturmakta ve bu bloğu ağdaki diğer düğümlere göndermektedir. Tüm düğümler, oluşturulan bloğun geçerliliğini doğrulamakta ve blok geçerli bulunursa blok zincire eklenmektedir. Geçersiz bir blok durumunda süreç sona ermekte ve yeni bir döngü başlatılmaktadır. Geçerli blok oluşturan lider düğüm, ödüllendirilmekte ve algoritma, bir sonraki lider seçimi için süreci yeniden başlatmaktadır. Bu süreç, POET algoritmasının enerji verimliliği ve adil lider seçimi

sağlamasını mümkün kılarken, blok zincirinin güvenliğini ve işlem kapasitesini artırmaktadır. Şekil 3.6'da PoET algoritmasının tüm aşamaları akış şemasında gösterilmektedir. (Hossain vd., 2024)



Şekil 3.6. PoET algoritması akış şeması

3.1.10. Önem Kanıtı

Önem Kanıtı, blok zinciri ağlarında daha adil ve katılımcı bir yapı oluşturmayı hedefleyen yenilikçi bir konsensüs algoritmasıdır. Bu algoritma, yalnızca kullanıcıların sahip oldukları varlık miktarını değil, aynı zamanda ağdaki etkinliklerini, işlem sıklıklarını ve diğer kullanıcılarla olan etkileşimlerini de değerlendirmektedir. PoI'nin en bilinen uygulaması, NEM (New Economy Movement) blok zinciri platformunda görülmektedir. NEM, PoI algoritmasını entegre ederek, ağın güvenliğini ve verimliliğini artırmayı amaçlamıştır (NEM, 2018). Bu yaklaşım, varlık miktarına dayalı olan İş Kanıtı ve Hisse Kanıtı algoritmalarına alternatif olarak geliştirilmiştir.

PoI algoritması, bir kullanıcının ağ üzerindeki önemini belirlemek için birkaç farklı faktörü birleştirmektedir. Bunlar arasında kullanıcının sahip olduğu varlık miktarı, ağda gerçekleştirdiği işlemlerin sıklığı ve diğer kullanıcılarla olan etkileşimleri yer almaktadır. Bu faktörler, algoritma tarafından bir önem puanı olarak hesaplanmakta ve blok oluşturma ya da işlem doğrulama süreçlerinde öncelikli olarak kullanılmaktadır. Böylece yalnızca büyük varlık sahipleri değil, aktif kullanıcılar da ödüllendirilmektedir. NEM platformunda bu mekanizma, kullanıcıların ağ aktivitelerine katılımını artırarak daha merkeziyetsiz ve güvenilir bir ağ yapısı sağlamayı hedeflemiştir (NEM, 2018).

PoI algoritmasının avantajları arasında adil bir katılımın teşvik edilmesi, ağdaki merkeziyetçiliğin azaltılması ve enerji verimliliği bulunmaktadır. Kullanıcıların ağdaki etkinliklerine dayalı bir önem puanıyla ödüllendirilmesi, hem bireysel katılımı artırmakta hem de yalnızca varlık miktarına dayalı ödüllendirme mekanizmalarının neden olduğu adaletsizlikleri azaltmaktadır. Ayrıca PoW gibi yüksek enerji tüketimine yol açan mekanizmalardan farklı olarak PoI, enerji verimli bir çözüm sunmaktadır. Ancak, bu algoritmanın bazı kısıtlamaları da bulunmaktadır. Örneğin, PoI algoritması üzerine yapılan bilimsel çalışmalar sınırlı olmakla birlikte, algoritmanın geniş çapta uygulanabilirliği ve güvenilirliği tam anlamıyla kanıtlanmış değildir. Xiao vd., 2022 yılında, dinamik yetkilendirmeye dayalı bir PoI konsensüs algoritması önererek, ağ güvenliğini ve işlem verimliliğini artırmayı hedeflemişlerdir. Bu öneri, PoI'nin mevcut sınırlamalarının üstesinden gelmek için önemli bir adım olarak değerlendirilmektedir.

Sonuç olarak, Önem Kanıtı konsensüs algoritması, blok zinciri ağlarında daha adil, katılımcı ve enerji verimli bir yapı oluşturmayı hedeflemektedir. NEM gibi

platformlarda uygulanması, bu algoritmanın potansiyelini ortaya koymaktadır. Bununla birlikte, PoI'nin güvenliği ve etkinliği üzerine daha geniş kapsamlı bilimsel çalışmalar yapılması gerekmektedir. PoI'nin yenilikçi yaklaşımı, gelecekte blok zinciri teknolojilerinin gelişi minde önemli bir rol oynayabileceğini göstermektedir.

3.1.11. Ripple Protokol Konsensüs Algoritması

Ripple Protokol Konsensüs Algoritması, Ripple ağı tarafından kullanılan ve Bizans Hata Toleranslı bir konsensüs mekanizmasıdır. Bu algoritma, ağdaki düğümlerin belirli bir işlem seti üzerinde fikir birliğine varmasını sağlayarak, dağıtık defterin tutarlılığını ve güvenilirliğini temin etmektedir. RPCA'nın temel yapı taşı, her düğümün kendi Benzersiz Düğüm Listesi'ni (Unique Node List - UNL) oluşturmasıdır. UNL, düğümün güvenilir bulduğu diğer düğümleri içermekte ve konsensüs sürecinde yalnızca bu düğümlerin oyları dikkate alınmaktadır (Mundhra & Rebeiro, 2020). Bu yapı, ağdaki her düğümün kendi perspektifinden güvenilir bir alt ağ oluşturmasını sağlamaktadır. RPCA, her birkaç saniyede bir tekrarlanan turlar halinde çalışmaktadır. Süreç, aday işlemlerin yayınlanması, bu işlemlerin UNL üzerinden birleştirilmesi ve ardından her düğümün bu işlemler üzerinde oy kullanması adımlarından oluşmaktadır. Belirli bir eşik değerin üzerinde "evet" oyu alan işlemler kabul edilmekte ve deftere eklenmekte, eşik altındaki işlemler ise bir sonraki turda yeniden değerlendirilmekte veya tamamen reddedilmektedir (Chase & MacBrough, 2018). Bu süreç, ağdaki tüm düğümlerin defterin son hali üzerinde mutabık kalmasını sağlamaktadır.

RPCA'nın güvenliği üzerine yapılan analizlerde, UNL seçiminin kritik bir unsur olduğu vurgulanmaktadır. Eğer UNL'ler arasında yeterli örtüşme yoksa, ağda çatallanma veya tutarsızlıklar meydana gelmektedir. Bu durum, protokolün güvenlik garantilerini ihlal edebilmektedir (Amores-Sesar vd., 2020). Ayrıca, RPCA'nın hız ve güvenliğini artırmak amacıyla UNL'lerin daha dinamik ve esnek bir şekilde yönetilmesi gerektiği belirtilmektedir (Mundhra & Rebeiro, 2020).

Performans açısından RPCA, enerji yoğun madencilik süreçlerine ihtiyaç duymadan hızlı işlem onayları sağlamaktadır ve bu özelliğiyle finansal işlemler için düşük gecikme süresi ve yüksek verimlilik sunmaktadır. Ancak, ağın büyüklüğü ve UNL'lerin yönetimi, ölçeklenebilirlik üzerinde doğrudan etkili olmaktadır. Bilgi

yayılımını optimize etmek ve kötü niyetli davranışları tespit etmek için UNL yapılandırmalarının iyileştirilmesi gerekmektedir (Amores-Sesar vd., 2020).

Sonuç olarak, RPCA, Ripple ağı için hızlı ve güvenilir bir konsensüs mekanizması sunmaktadır. Ancak, güvenlik ve ölçeklenebilirlik açısından başarılı bir şekilde uygulanabilmesi için UNL'lerin doğru seçimi ve yönetimi büyük önem taşımaktadır.

3.1.12. Diğer Konsensüs Algoritmaları

Bölüm 3.1'de değinilen mevcut konsensüs algoritmalarına ek olarak, literatürde görece daha az yer bulmuş algoritmalar da mevcuttur. Bu algoritmaların çalışma sistemi, kısıtlı ve güçlü yönleri, uygulanabilirliği tartışılmıştır. İtibar Kanıtı (PoR), Alan Kanıtı (PoSpace), Tarih Kanıtı (POH), Kiralanmış Hisse Kanıtı (LPoS), Rastgelelik Kanıtı (PoRand), Olasılık Kanıtı (PoP) birden fazla algoritmanın birleşiminden olun her birinin güçlü yönlerini kullanmayı amaçlan hibrit konsensüs algoritmaları gibi algoritmalar da çeşitli makalelerde ele alınmış ve farklı yönleriyle karşılaştırılarak incelenmiştir.

İtibar Kanıtı algoritmasında düğümlerin geçmişteki davranışları değerlendirilmiştir. Güvenilir düğümler, daha fazla doğrulama şansı elde etmekte ve ağın bütünlüğünü korunmaktadır (Gai, Wang, Deng & Peng, 2018).

Alan Kanıtı algoritmasında katılımcılar, blok zincirine katkıda bulunmak için depolama alanlarını tahsis etmektedirler. Daha fazla alan ayıran katılımcılar, blok oluşturma şansı elde etmektedirler (Dziembowski vd., 2015). Chia blok zinciri ağı, bu algoritmayı kullanmaktadır (Cohen & Network, 2021).

Tarih Kanıtı algoritmasında her işlemin bir zaman damgası alınması sağlanarak işlemler sıralanmaktadır. Bu, blok zincirinde işlemler, hızlı bir şekilde doğrulanmakta ve enerji verimliliği sağlanmaktadır. Solana ağı, bu algoritmayı kullanan bir blok zinciri ağıdır (Yakovenko, 2018).

Kiralanmış Hisse Kanıtı algoritmasında kullanıcılar, tokenlerini doğrulayıcı düğümlere kiralamaktadır. Doğrulayıcılar, kullanıcıların hisselerini kullanarak blok oluşturmakta ve ödülleri paylaşmaktadırlar. Küçük kullanıcılar da ödüllerden yararlanabilmektedir. Waves platformu, bu algoritmayı kullanan bir blok zinciri ağıdır (Waves Documentation, n.d.).

Rastgelelik Kanıtı algoritmasında, yeni blokların oluşturulması ve işlemlerin doğrulanması görevi, rastgele seçilen doğrulayıcılara atanmaktadır. Bu seçim süreci, bir rastgele sayı üreteci kullanılarak gerçekleştirilmektedir (Hossain vd., 2024).

Olasılık Kanıtı algoritmasında hem orijinal şifrelenmiş özet değeri hem de bir dizi sahte özet değeri depolanmaktadır. Orijinal özet değerini başarıyla çözen ilk düğüm, yeni bir blok üretmekten sorumlu olmaktadır (Hossain vd., 2024). Yoğun hesaplama rekabetini yönetmek için, ardışık özet çözümleri arasında bir bekleme süresi tanınmaktadır (Kim & Kim, 2018).

Son olarak, POW ve POS konsensüs algoritmalarının güçlü yanlarını birleştirerek hibrit yaklaşımla tasarlanan algoritmalar da bulunmaktadır. PoW madencileri blok oluştururken, PoS doğrulayıcıları işlemleri onaylamaktadır (Kumari vd., 2024). Dash ağı, bu hibrit konsensüs algoritmayı kullanan bir blok zinciri ağıdır.

3.2. Konsensüs Algoritmalarının Karşılaştırılması

Konsensüs algoritmaları, blok zincir teknolojilerinde performans, güvenlik, enerji verimliliği, gecikme süresi ve ölçeklenebilirlik ölçütleriyle değerlendirilmiştir.

3.2.1. Güvenlik

- PoW algoritması, yüksek güvenlik sağlamasına rağmen %51 saldırılarına teorik olarak açık bir yapıya sahiptir. Ancak, özellikle Bitcoin gibi büyük ağlarda bu tür bir saldırıyı gerçekleştirmek, son derece yüksek maliyetler gerektirdiği için pratikte neredeyse imkansızdır.
- PoS algoritması, Sybil saldırılarına karşı dayanıklı olmasına karşın ekonomik eşitsizlik eleştirisi almaktadır.
- PBFT algoritması, lider seçimine bağımlı olduğu için güvenlik riskleri barındırmaktadır.
- PoA ve diğer özel algoritmalar, merkezi yapıları nedeniyle güvenlik açıkları taşıyabilmektedir.
- FBA ve RPCA gibi hibrit veya konsorsiyum tabanlı algoritmalar, yapılandırmaya bağlı olarak güvenlik sağlamaktadır.

3.2.2. Ölçeklenebilirlik

- PoW algoritmasının işlem kapasitesi düşük olmakla birlikte, enerji tüketimi oldukça yüksektir.
- PoS algoritması, daha ölçeklenebilir ve enerji verimli bir yapı sunmaktadır.
- PBFT ve PoA algoritmaları, daha küçük ağlarda yüksek ölçeklenebilirlik sağlamaktadır.
- FBA ve hibrit algoritmalar, büyük ağlarda ölçeklenebilirliği artırmakla birlikte, doğru yapılandırma gerektirmektedir.

3.2.3. Enerji Verimliliği

- PoW algoritması, çok yüksek enerji tüketimine sahip olup çevresel etkileri nedeniyle eleştirilmektedir.
- PoS, PoET ve PoA algoritmaları, düşük enerji tüketimleriyle çevre dostu bir yapı sunmaktadır.
- PoC algoritması, enerji tasarrufu sağlamasına rağmen depolama maliyetleri oluşturmaktadır.

3.2.4. Gecikme Süresi

- PBFT, PoET ve RPCA algoritmaları, düşük gecikme süreleriyle öne çıkmaktadır.
- PoW algoritmasında işlem onayında yüksek gecikme süreleri yaşanabilmektedir.
- PoS ve diğer hibrit algoritmalar, orta düzey gecikme süreleri sunmaktadır.

Bölüm 3'te detaylandırılan konsensüs algoritmalarının bu farklılıkları Çizelge 3.1'de blok zinciri türüne göre, güçlü yanları açısından, zayıf yanları açısından değerlendirilmiş ve kullanıldığı platformlar ile birlikte detaylandırılmıştır. Bu algoritmalar, Çizelge 3.2'de güvenlik, ölçeklenebilirlik, enerji verimliliği ve gecikme süresi açılarından karşılaştırılmıştır.

Çizelge 3.1. Konsensus algoritmalarının güçlü ve zayıf yanlarının, kullanıldığı platformların ve blok zinciri türlerine göre karşılaştırılması

Algoritma	**Blok Zinciri Türü**	**Güçlü Yanları**	**Zayıf Yanları**	**Kullanıldığı Platformlar**
PoW (Nakamoto, 2008; De Vries, 2018)	Genel	Güvenli, merkeziyetsiz	Enerji verimsiz, düşük işlem kapasitesi	Bitcoin, Litecoin
PoS (King, 2012; Saleh, 2020)	Genel	Enerji verimli, ölçeklenebilir	Zenginleşme, likidite sorunu	Ethereum 2.0, Cardano
PBFT (Castro & Liskov, 1999; Wang vd., 2019)	Özel	Yüksek işlem hızı, düşük gecikme süresi	Ölçeklenebilirlik, lider bağımlılığı	Hyperledger Fabric, Corda
PoA (Islam vd., 2022; Yang vd., 2022)	Özel	Enerji verimli, yüksek işlem kapasitesi	Merkeziyetçilik, ölçeklenebilirlik	Özel ve konsorsiyum blok zincirleri
PoB (Katal vd., 2020; Hussein vd., 2023)	Genel	Enerji verimli, çevre dostu	Finansal ve merkeziyetçilik riski	Yeni nesil blok zinciri platformları
PoC (Salimitari vd., 2018; Debus, 2017)	Genel	Donanım dostu, düşük enerji tüketimi	Depolama verimsizliği	Chia, Burstcoin
PoET (Chen vd., 2017; Bowman vd., 2021)	Özel	Enerji tasarrufu, adil lider seçimi	Donanım bağımlılığı, güvenlik riskleri	Kurumsal blok zinciri uygulamaları
FBA (Nugroho vd., 2023; Kim vd., 2019)	Hibrit	Dinamik, ölçeklenebilir	Çoğunluk dilimlerinin yapılandırılması	Stellar, konsorsiyum ağları
RPCA (Mundhra, 2020; Chase, 2018)	Konsorsiyum	Hızlı işlem onayı, enerji verimliliği	UNL yapılandırılması, ölçeklenebilirlik	Ripple

Çizelge 3.2. Konsensus algoritmalarının güvenlik, ölçeklenebilirlik, enerji verimliliği ve gecikme süresi kriterlerine göre karşılaştırılması

Algoritma	**Güvenlik Özellikleri**	**Ölçeklenebilirlik**	**Enerji Verimliliği**	**Gecikme Süresi**
PoW (Nakamoto, 2008; De Vries, 2018)	Yüksek- ancak %51 saldırısına açık	Düşük ölçeklenebilirlik	Çok yüksek enerji tüketimi	Yüksek gecikme süresi
PoS (King, 2012; Saleh, 2020)	"Zengin daha zengin" eleştirisi	Daha iyi ölçeklenebilirlik	Enerji verimli	Daha düşük gecikme süresi
PBFT (Castro & Liskov, 1999; Wang vd., 2019)	Düşük saldırı riski- lider seçimi bağımlı	Yüksek işlem hızı	Enerji verimli (lider düğümlere bağımlı)	Çok düşük gecikme süresi
PoA (Islam vd., 2022; Yang vd., 2022)	Merkezi bir kontrol- güvenlik riski	Küçük ağ ile yüksek ölçeklenebilirlik	Enerji açısından çok verimli	Düşük gecikme süresi
PoB (Katal vd., 2020; Hussein vd., 2023)	Varlık bazlı saldırılara açık	Orta düzey ölçeklenebilirlik	Çevre dostu	Orta düzey gecikme süresi
PoC (Salimitari vd., 2018; Debus, 2017)	Donanım- güvenlik	Orta düzey ölçeklenebilirlik	Daha az enerji tüketimi	Orta düzey gecikme süresi
PoET (Chen vd., 2017; Bowman vd., 2021)	Donanımına bağımlı	Orta düzey ölçeklenebilirlik	Çok düşük enerji tüketimi	Çok düşük gecikme süresi
FBA (Nugroho vd., 2023; Kim vd., 2019)	Çoğunluk dilim yapılandırılması	Yüksek ölçeklenebilirlik	Orta düzey enerji verimliliği	Düşük gecikme süresi
RPCA (Mundhra, 2020; Chase, 2018)	UNL yapılandırılması	Düşük ölçeklenebilirlik	Enerji verimli	Çok düşük gecikme süresi

BÖLÜM 4

YENİ BİR KONSENSUS ALGORİTMASI

4.1. Uygulama Amaç ve Kapsamı

Bu çalışmanın amacı, mevcut konsensüs algoritmalarının belirli kısıtlamalarını aşabilecek yeni bir algoritma tasarımıdır. Bu bağlamda:

- Blok zinciri teknolojisinin temel ilkeleriyle mevcut algoritmalar analiz edilmiştir.
- Tasarlanan yeni algoritma ile:
  - Merkeziyetsizlik seviyesi artırılmış,
  - Manipülasyon riski (rastgelelik artırılarak) azaltılmış,
  - Doğrulayıcı seçiminde eşit oy hakkı ile demokratik sistem desteklenmiştir.

Tasarlanan bu yeni algoritmanın, literatürde blok zinciri teknolojisinde kullanılan konsensus algoritmalarının odaklanılan belli gereksinimleri sağlayabilmek amacıyla merkeziyetsizlikten ödün verme, demokratikleşmeden uzaklaşma ve manipülasyon gibi kısıtlarına yönelik önemli katkılar ve iyileştirmeler sunması beklenmektedir.

4.2. Uygulamanın Planlanması ve Hazırlık Süreci

Planlama süreci iki temel aşamadan oluşmuştur:

1. Dokümantasyon ve Literatür Taraması: Çalışmanın ilk 6 aylık sürecinde, mevcut konsensüs algoritmaları detaylı şekilde incelenmiş ve bu algoritmalara yönelik kısıtlamalar tespit edilmiştir. Bu süreç sonunda elde edilen bulgular tez izleme kuruluna sunulmuştur.
2. Algoritma Analizi ve Tasarım: Mevcut algoritmaların analizine dayalı olarak yeni algoritmanın yapısı geliştirilmiş ve tasarımı gerçekleştirilmiştir. Algoritmanın her aşaması algoritma akış şeması ile planlanmış ve süreç görselleştirilmiştir.

4.3. Uygulamada Kullanılan Araç ve Yöntemler

Bu çalışmada kullanılan araçlar ve yöntemler şunlardır:

Programlama Dili: Go programlama dili, hızlı ve verimli bir geliştirme ortamı sağladığı için tercih edilmiştir. Go, derlenmiş bir dil olup, yüksek performanslı bir dil olarak bilinmektedir. Derleme süreci, yazılan kodun doğrudan makine koduna çevrilmesini sağlamakta, bu da yüksek performans elde edilmesine olanak tanımaktadır. Go'nun basit sözdizimi, kodun okunabilirliğini artırmaktadır. Dil tasarımı, programcılara daha az karmaşıklık ve daha fazla anlaşılabilirlik sunmaktadır. Go, otomatik çöp toplama (garbage collection) özelliği ile bellek yönetimini kolaylaştırmakta, böylece programcılar bellek sızıntılarına karşı daha dirençli kodlar yazabilme imkanı sunmaktadır. Go'nun zengin standart kütüphanesi, birçok temel görevi kolayca gerçekleştirmeyi sağlamaktadır (Donovan & Kernighan, 2015).

Veri Yapıları: Zincir (chain) ve doğrulayıcı oyları (voteCounts) gibi veri yapıları kullanılmıştır.

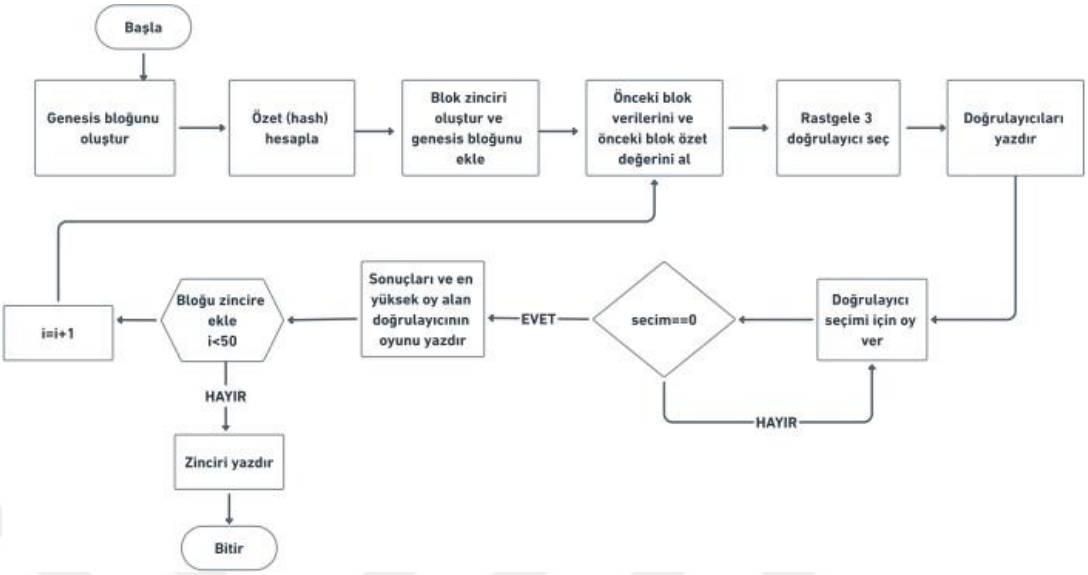
Kriptografik Fonksiyonlar: SHA-256 algoritması, blok zinciri verilerinin güvenliğini sağlamak için kullanılmıştır.

4.4. Uygulama Süreci

Mevcut konsensüs algoritmalarının kullanılabilirliğini, avantajlarını, dezavantajlarını birlikte değerlendirerek yeni bir konsensüs algoritması tasarlanmıştır. Bir blok zinciri simülasyonu tasarlanarak bloklar arasında oylama ve doğrulayıcı seçimi süreçleri gösterilmiştir. Uygulama sürecinde, tasarlanan algoritmanın ara aşamalarının her biri belirli bir amaca yönelik olarak geliştirilmiş ve zincirin işlevselliği sağlanmıştır. Algoritma iki katmandan oluşmaktadır. Birinci katmanda 50 doğrulayıcı adayı içinden rastgele 3 doğrulayıcı seçilmektedir. Rastgele seçilen doğrulayıcılar, blok zincir ağının daha dağıtılmış olmasını sağlamaktadır. Bu, ağın güvenilirliğini artırabilmekte ve tek bir noktanın hedef alınmasını zorlaştırabilmektedir. Rastgele seçim, blok zincir ağının merkeziyetsizliğini korumaya yardımcı olabilmektedir. Belirli bir grup doğrulayıcıya sürekli olarak güvenmek yerine, doğrulayıcıların rastgele seçilmesi, gücün tek bir noktada toplanmasını önleyebilmektedir. Bunlara ek olarak, rastgele seçim, tüm doğrulayıcıların eşit şartlarda işlem yapmasını sağlamaktadır, ağdaki herhangi bir doğrulayıcının ağa katılma ve işlemlere katılma şansının eşit olduğu anlamına gelmektedir. Diğer bir yandan,

bazen ağın performansını etkileyebilmekte, özellikle büyük bir ağda, rastgele seçilen doğrulayıcıların bazıları ağdakilerle uyumsuz olabilmektedir veya işlemleri geciktirebilmektedirler. Yeni katılan doğrulayıcılar veya kötü niyetli doğrulayıcılar, ağa zarar verebilmekte veya güvenlik açıklarını sömürebilmektedir. İkinci katmanda ise bu rastgele seçilen doğrulayıcılar, ağ katılımcılarının oylamasına sunulmaktadır. En yüksek oy alan doğrulayıcı seçilmektedir. Oylama, blok zincir ağının katılımcılarına katılım ve katkı sağlamaktadır. Kullanıcılar, doğrulayıcıları seçme sürecine aktif olarak katılabilmekte ve bu şekilde ağın yönetimine katkıda bulunabilmektedir. Kullanıcıların blok zincir ağının işleyişine katılma konusunda daha fazla özgürlük, daha adil temsiliyet ve esneklik sunmaktadır. Buna karşın, oylama sürecinde, manipülasyon veya oy hırsızlığı gibi güvenlik riskleriyle karşı karşıya kalınabilmektedir. Kötü niyetli aktörler, oylamayı etkileyebilmekte veya sonuçları değiştirebilmektedir. Oylama sonuçlarına dayalı olarak seçilen doğrulayıcılar, ağın denge ve kararlılığını etkileyebilmektedirler. Eğer seçim süreci düzensiz veya adil olmayan bir şekilde gerçekleşirse, ağın güvenilirliği ve performansı olumsuz etkilenebilmektedir. Oylama mekanizması karmaşık olabilmekte ve bunun uygulanması maliyetli olabilmektedir. Ayrıca, oy verme ve doğrulayıcıların seçimi için gereken süre, ağın işleyişini yavaşlatabilmekte veya geciktirebilmektedir. Tüm bu avantaj ve dezavantajlar değerlendirilerek algoritma gerçeklenmelidir. Bu çalışma, algoritmanın test edilmesi için Go (Golang) programa diliyle gerçeklenmiştir. Google tarafından geliştirilen bu programlama dili, derlenmiş bir dil olup, yüksek performansı ile bilinmektedir. Derleme süreci, yazılan kodun doğrudan makine koduna çevrilmesini sağladığı için yüksek performans elde edilmesine olanak tanımaktadır. Paralel programlama yapmayı kolaylaştırması, çoklu çekirdekli sistemlerde etkili performans sağlaması, basit ve temiz sözdizimi ile kodun okunabilirliğini artırması ve programcılara daha az karmaşıklık ve daha fazla anlaşılabilirlik sunması sebepleriyle tercih edilmiştir. Algoritmanın tüm aşamaları, algoritma Go dili ile simüle edilmesiyle ve algoritmanın akış şemasıyla açıklanmaktadır.

Tasarlanan bu yeni algoritmanın (RandVoteChain) tüm aşamalarını barındıran akış şeması Şekil 4.1'de verilmiştir.



Şekil 4.1. Tasarlanan algoritmanın akış şeması

RandVoteChain algoritmasının tasarımının simülasyonunun tüm aşamaları alt başlıklar halinde açıklanmaktadır.

4.4.1. Blok Zincirinin Başlatılması

Algoritma.1'de blok zinciri başlatılması aşamasının sözde kodu (pseudocode) verilmektedir. Algoritmanın bu aşaması, bir blok zinciri (Blockchain) sınıfını başlatmak için kullanılmaktadır. Zincir, tüm blokları saklamak için oluşturulan bir liste (chain) ile başlamaktadır ve başlangıçta boştur. Ayrıca, doğrulayıcılar için rastgele seçim yapılacak doğrulayıcı listesi (validators) tanımlanmaktadır.

Algoritma 1: Blok Zincirinin Başlatılması

Require: Yok

Ensure: Başlatılmış blok zinciri örneği class Blockchain:

procedure Initialize():

chain ← Boş liste

validators ← Tüm doğrulayıcıları içeren liste

4.4.2. Başlangıç Bloğunun Oluşturulması

Algoritma 2'de, blok zincirine başlangıç bloğunun (genesis block) eklenmesi aşamasının sözde kodu (pseudocode) verilmektedir. Bu aşamada, ilk blok sıfır indeks ile oluşturulmakta ve zincire eklenmektedir.

Algoritma 2: Başlangıç Bloğunun Oluşturulması

procedure CreateGenesisBlock():

index ← 0

timestamp ← Şu anki zaman

data ← "Genesis Block"

previousHash ← ""

validator ← "Genesis"

hash ← CalculateHash(index, previousHash, data, timestamp, validator)

block ← Block(index, timestamp, data, previousHash, hash, validator)

chain.append(block)

4.4.3. Blok Yönetimi

Algoritma 3'te blok yönetimi aşamasının sözde kodu (pseudocode) verilmektedir. Yeni blok oluşturma ve zincire ekleme işlemleri bu aşamada gerçekleştirilmektedir. Zincir üzerinde eklenen her blok doğrulama sürecinden geçmektedir.

Algoritma 3: Blok Yönetimi

procedure CreateBlock(data, validators):

prevBlock ← Son blok

selectedValidators ← getRandomValidators(validators)

timestamp ← Şu anki zaman

hash ← CalculateHash(prevBlock.index + 1, prevBlock.hash, data, timestamp, selectedValidators)

block ← Block(prevBlock.index + 1, timestamp, data, prevBlock.hash, hash, selectedValidators)

IF isBlockValid(block, prevBlock)THEN

chain.append(block)

RETURN block

procedure AddBlock(newBlock):

IF isBlockValid(newBlock, Son blok) THEN

chain.append(newBlock)

4.4.4. Doğrulayıcı Seçimi

Algoritma 4'te doğrulayıcı seçim işleminin sözde kodu (pseudocode) verilmektedir. Rastgele bir doğrulayıcı listesi seçilmektedir ve bu liste blok oluşturma yetkisi için oylanmak üzerine katılımcılara sunulacaktır.

Algoritma 4: Doğrulayıcı Seçimi

procedure getRandomValidators(allValidators):

shuffledValidators ← Randomize(allValidators)

RETURN shuffledValidators[:3]

4.4.5. Doğrulayıcı Oylaması

Algoritma 5'te, doğrulayıcı oylama aşamasının sözde kodu (pseudocode) verilmektedir. Kullanıcı oylamasıyla doğrulayıcılar arasından bir seçim yapılmakta ve en çok oy alan doğrulayıcı seçilmektedir.

Algoritma 5: Aşama 5: Doğrulayıcı Oylaması

procedure voteForValidator(validators):

Göster doğrulayıcıları

WHILE giriş != 0 DO

Kullanıcıdan giriş al

Eğer geçerli ise

Oy sayısını artır

EN YÜKSEK oy sayısına sahip doğrulayıcıyı belirle

RETURN doğrulayıcı

4.4.6. Özet Hesaplama

Algoritma 6'da blok zincirindeki özet hesaplama aşamasının sözde kodu verilmektedir. Bu aşama, blokların özetlerini oluşturmaktadır.

Algoritma 6: Özet Hesaplama

procedure CalculateHash(index, previousHash, data, timestamp, validators):

hashValue ← Concatenate(index, previousHash, data, timestamp, validators)

RETURN SHA256(hashValue)

4.4.7. Ana İşlem Akışı

Algoritma 7'de ana işlem akışının sözde kodu verilmektedir. Bu işlem tüm blokların oluşturulması, doğrulanması ve zincire eklenmesi için sıralı işlemleri tanımla maktadır.

Algoritma 7: Ana İşlem Akışı

procedure Main():

Initialize() CreateGenesisBlock()

FOR i ← 1 TO 50 DO

newBlock ← CreateBlock("Data" + i, validators)

AddBlock(newBlock)

END FOR

Tüm blokları yazdır

4.4.8. Blok Zincirinin Geçerliliğinin Kontrolü

Algoritma 8'de, blok zincirinin geçerliliğini kontrol eden sözde kodu (pseudo-code) verilmektedir. Bu aşama, her bir bloğun özet değerlerinin doğruluğunu ve önceki özet ile bağlantısını kontrol ederek zincirin bütünlüğünü doğrulamaktadır.

Algoritma 8: Blok Zincirinin Geçerliliğinin Kontrolü

```
procedure isValid(blockchain):
FOR i ← 1 TO length(blockchain.chain) - 1 DO
    currentBlock ← blockchain.chain[i] prevBlock ← blockchain.chain[i-1]
    calculatedHash ← CalculateHash(currentBlock.index, currentBlock.timestamp,
      currentBlock.data, currentBlock.prevHash, currentBlock.validator)
    IF currentBlock.hash != calculatedHash THEN
        PRINT Blok i: Hash hatalı.
        RETURN false
    END IF
    IF currentBlock.prevHash != prevBlock.hash THEN
        PRINT Blok i: Önceki hash hatalı.
        RETURN false
    END IF
END FOR
RETURN true
```

4.5. Karşılaşılan Zorluklar ve Çözümleri

Uygulama sürecinde karşılaşılan başlıca zorluklar ve çözümleri maddelendirilmiştir.

- **Performans Sorunları:** Büyük ölçekli veri setlerinde işlem doğrulama süreleri uzamıştır. Go programlama dilinin paralel işlem yetenekleri ile bu sorun optimize edilmiştir.
- **Hash Çakışmaları:** Hash hesaplamaları sırasında oluşabilecek çakışma riski, verilerin yeniden düzenlenmesi ile giderilmiştir.
- **Doğrulayıcı Seçiminde Rastgelelik Sorunları:** Rastgelelik algoritmalarının yeterince adil olmaması nedeniyle doğrulayıcı seçim sürecinde merkeziyetçi bir eğilim oluşmuştur. Daha güvenilir bir rastgelelik kaynağı kullanılarak bu sorun çözülmüştür.

- **Blok Boyutu Optimizasyonu:** Büyük bloklar nedeniyle veri işleme süreleri artmıştır. Blok boyutları, işlem hacmi göz önünde bulundurularak optimize edilmiştir.
- **Güvenlik ve Manipülasyon Riskleri:** Manipülasyon riskleri, doğrulayıcıların oylama sürecindeki ilk aşamada rastgelelik sorununun çözülmesiyle azaltılmıştır.
- **Ağ Gecikmeleri:** Dağıtık ağdaki düğümler arasında iletişim gecikmeleri yaşanmıştır. Daha hızlı iletişim protokolleri kullanılarak bu sorun azaltılmıştır.
- **Veri Saklama Kapasitesi:** Zincirin büyüklüğü arttıkça veri saklama gereksinimleri artmıştır. Daha verimli depolama çözümleri araştırılmıştır.
- **Enerji Tüketimi Sorunları:** Blok zincirinin işlem süreçlerinde enerji tüketimi yüksek olmuştur. Enerji verimliliği sağlayan algoritmaların entegrasyonu ile bu sorun çözülmüştür.

BÖLÜM 5

SONUÇLAR VE TARTIŞMA

Bölüm 4'te yer verilen RandVoteChain konsensüs algoritmasının tasarımı ile ilgili tartışma, sonuç ve çıkarımları ile gelecek çalışmalar için öneriler alt başlıklar halinde açıklanmaktadır.

5.1. Uygulama Sonuçları ve Çıkarımlar

Bu bölümde, uygulama sonuçları ve çalışmanın blok zinciri literatürüne katkıları ele alınmaktadır.

- **Merkeziyetsizlik Seviyesi Artışı:** Tasarlanan algoritma, katılımcılar arasında eşit söz hakkı sağlamaktadır ve merkezi bir otoriteye bağımlılığı azaltması beklenmektedir.
- **Performans İyileştirmeleri:** Yeni algoritma, işlem doğrulama sürelerini kısaltarak mevcut konsensüs algoritmalarına göre daha yüksek bir işlem kapasitesi sağlaması beklenmektedir.
- **Manipülasyon Riskinin Azaltılması:** Doğrulayıcı seçim sürecindeki rastgelelik ve oylama mekanizması, ağın güvenilirliğini artırabilmektedir.

5.2. Tartışma

Çalışmanın sonuçları, mevcut literatürde yer alan konsensüs algoritmalarına kıyasla şu katkıları sağlayabileceğini göstermektedir:

- Merkeziyetsizlik ve Adalet: Doğrulayıcı seçim mekanizmasının, katılımcılar arasında eşit haklar sağlama ve demokratik bir yapı oluşturma potansiyeli bulunmaktadır.
- Performans ve Güvenlik: Yeni algoritmanın, hızlı işlem doğrulama süreçleri ve iki aşamalı konsensüs mekanizması ile rastgeleliğin öngörülemezliği sayesinde

belirli doğrulayıcılara doğrudan saldırı yapılamamasını sağlayarak güvenlik açısından blok zinciri teknolojisini daha işlevsel hale getirebileceği düşünülmektedir.

- Literatüre Katkı: Çalışma, konsensüs algoritmalarında merkeziyetsizlik ve adalet temelli yaklaşımlar konusunda eksikliği hissedilen alanlara yenilikçi bir çözüm sunabilmektedir.

5.3.Performans Test Sonuçları

Algoritmanın performansı, çeşitli senaryolar altında test edilmiş ve elde edilen bulgular aşağıda özetlenmiştir:

- Tek İş Parçacıklı Performans: ‘generateBlock‘ fonksiyonu, tek iş parçacıklı modda yaklaşık 1019 ns/op ± 335 ns/op hızında çalışarak hızlı ve optimize bir yapıya sahip olduğunu göstermiştir.
- Paralel İşleme Yeteneği: Paralel modda aynı fonksiyon, yaklaşık 226.1 ns/op ± 125.1 ns/op ile yaklaşık 4.5 kat daha hızlı çalışmış ve yoğun iş yükleri altında iyi bir performans sergilemiştir.
- Toplam Çalışma Süresi: Testlerin toplam süresi yaklaşık 2.863 s ± 1.468s olarak ölçülmüş, bu da sistemin verimli çalıştığını ve hızlı yanıt verebildiğini göstermektedir.

Bu testler sırasında CPU başka işlemlerle meşgul olabilmekte ve bu da ölçümlerde sapmaya neden olabilmektedir. Çoklu işlemci çekirdeklerinin kullanımı (özellikle paralel testlerde) daha fazla değişkenlik yaratabilmektedir. Cache kullanımı veya bellek erişim gecikmeleri (latency) performansı etkileyebilmektedir. İşlemcinin sıcaklığı arttıkça hız sınırlamaları (thermal throttling) devreye girebilmekte, bu da performans düşüşüne yol açabilmektedir.

- *Ölçeklenebilirlik:* 1.000.000 blok içeren veri setiyle gerçekleştirilen testlerde, algoritmanın büyük veri setleri altında istikrarlı bir performans gösterdiği gözlemlenmiştir.
- *Enerji Verimliliği:* Testler sırasında enerji tüketiminin düşük olduğu gözlemlenmiş, bu da tasarlanan algoritmanın enerji verimliliği açısından avantajlı olabileceğini göstermektedir. Enerji tasarrufu, özellikle büyük ağlarda çevresel sürdürülebilirlik açısından önemli bir katkı sunabilmektedir.

- *Güvenlik:* Algoritma, SHA-256 kullanılarak özet hesaplama yapmaktadır. Bu, güçlü bir kriptografik güvenlik sağlamakta ve zincirin bütünlüğünü korumaktadır.
- *Blok Doğrulama:* isValid fonksiyonu, her bir bloğun özet değerini ve zincir üzerindeki bağlantısını kontrol etmektedir. Bu, zincirin herhangi bir manipülasyona karşı korunmasını sağlayabilmektedir.
- *Doğrulayıcı Seçimi:* Rastgelelik ve oylama mekanizması ile bir doğrulayıcı seçilmektedir. Rastgelelik mekanizmasının öngörülemez doğası, dışarıdan gelecek müdahalelere veya kötü niyetli doğrulayıcıların manipülasyon girişimlerine karşı güvenliği artırmakta ve sistemin bütünlüğünü koruyabilmektedir.
- *Oylama Mekanizması:* Oylama sırasında sadece 1, 2 ve 3 girişleri kabul edilmekte ve başka girdiler hatalı sonuç uyarısı verip oylamaya dahil edilmemektedir. Eşit oy varsa rastgele seçilen doğrulayıcı listesinin ilk elemanı seçilmektedir.
- *Eşit Oy Durumu:* Eğer iki veya daha fazla doğrulayıcı eşit oy aldıysa, farklı bir senaryoda kazanan doğrulayıcı rastgele seçilebilmektedir. Diğer bir senaryoda deterministik bir çözüm olarak ise eşit oy durumunda alfabetik olarak en küçük doğrulayıcı seçilebilmektedir. Bir başka senaryoda ise, eşit oy durumunda önceki blokta seçilen doğrulayıcıyı tekrar seçmek mümkündür. Bu, zincirin tutarlılığını artırabilmektedir.

Bu bulgular ile algoritmanın yüksek performanslı sistemlerde uygulanabilir olabileceğini ve büyük veri setleriyle çalışmaya uygun olabileceği analiz edilmektedir. Ayrıca, çift katmanlı yapının hem merkeziyetsizliği hem de güvenilirliği artırarak literatürdeki diğer algoritmalara alternatif bir çözüm sunabileceği değerlendirilmektedir.

5.4. Algoritmanın Zaman ve Bellek Karmaşıklığı

Tasarlanan RandVoteChain algoritmasının zaman ve bellek karmaşıkları incelenmiştir.

Zaman Karmaşıklığı

Blok oluşturma (generateBlock):

Her yeni blok oluşturmak için calculateHash fonksiyonu çağrılmaktadır.

calculateHash fonksiyonu, bir string üzerinde sabit uzunlukta bir SHA-256 hash hesaplamaktadır, bu nedenle $O(1)$'dir.

Blok sayısı n olduğundan, tüm bloklar için karmaşıklık $O(n)$ olmaktadır.

Doğrulayıcı seçimi (voteForValidator):

Oylama sistemi, kullanıcı girişine bağlıdır ve maksimum k doğrulayıcıyı işleme almaktadır (burada $k = 3$).

Her turda bir kullanıcı girişi işlenmektedir ve toplamda $O(k \times v)$, v kullanıcı giriş sayısı ile sınırlıdır. Pratikte sabit sayıda doğrulayıcıyla sınırlandırıldığı için $O(k)$ olarak kabul edilebilmektedir.

Blok zincirinin doğrulanması (isValid):

Her blok için özet kontrolü yapılmaktadır ($O(1)$).

Blok sayısı n olduğundan, toplam doğrulama süresi $O(n)$'dir.

Toplam zaman karmaşıklığı: $O(n)$ (blok oluşturma ve doğrulama işlemlerinin birleşimiyle).

Bellek Karmaşıklığı

Blok zinciri saklama:

Her blok sabit boyutlu bilgileri içerir
($Index, Timestamp, Data, Hash, PrevHash, Validator$).

n blok için toplam bellek karmaşıklığı $O(n)$'dir.

Doğrulayıcı listesi:

Tüm doğrulayıcılar listesi saklanmaktadır (m doğrulayıcı varsa $O(m)$).

Ancak sabit bir doğrulayıcı sayısı kullanıldığından bu genellikle $O(1)$ 'dir.

Toplam bellek karmaşıklığı: $O(n)$ (blok zincirinin saklanmasıyla).

Bu analiz, giriş boyutunun (örneğin blok sayısı n) büyümesine göre performansı özetlemektedir.

5.5. Gelecek Çalışmalar İçin Öneriler

Bu çalışmanın sonuçlarına dayanarak, gelecekte yapılabilecek araştırmalar için aşağıdaki önerilerde bulunulmaktadır:

- Daha Büyük Ölçekli Testler: Algoritmanın performansı, daha büyük ağlarda ve farklı veri türleriyle test edilmelidir.
- Farklı Blok Zinciri Uygulamaları: Algoritmanın merkeziyetsiz finans (DeFi), akıllı sözleşmeler ve tedarik zinciri yönetimi gibi farklı alanlara uygulanabilirliği incelenmelidir.
- Karmaşık Doğrulayıcı Seçim Mekanizmaları: Doğrulayıcı seçim sürecine sosyal, ekonomik veya diğer karmaşık kriterler entegre edilerek algoritmanın daha çeşitli uygulamalara uyum sağlaması test edilmelidir.
- Enerji Verimliliği Çalışmaları: Algoritmanın enerji tüketimi optimizasyonu üzerine çalışmalar yapılmalıdır.
- Gerçek Dünya Uygulamaları: Algoritmanın simülasyon ortamından çıkarılarak gerçek dünyadaki uygulamalar üzerinde test edilmesi gerekmektedir.
- Gecikme Süresi Optimizasyonu: Algoritmanın gecikme süresini azaltmak için paralel işleme mekanizmalarının optimize edilmesi gereklidir.

KAYNAKLAR

Amores-Sesar, I., Cachin, C., & Mićić, J. (2020). Security Analysis of Ripple Consensus. *arXiv preprint*. https://arxiv.org/abs/2011.14816

Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., & vd. (2018). Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains. *Proceedings of the Thirteenth EuroSys Conference*, 1–15. https : //doi.org/10.1145/3190508.3190538

Antonopoulos, A. M. (2017). Mastering Bitcoin: *Unlocking Digital Cryptocurrencies* (2nd). O'Reilly Media.

Asad, N. A., Elahi, M. T., Hasan, A. A., & Yousuf, M. A. (2020). Permission-based blockchain with proof of authority for secured healthcare data sharing. 2020 2nd *International Conference on Advanced Information and Communication Technology (*ICAICT), 35–40. https://doi.org/10.1109/ICAICT51780.2020.9333488

Asadauskas, M. P. (2021). PoET: An Eco-Friendly Alternative to PoW [Bachelor Thesis, University of Bern]. https://crypto.unibe.ch/archive/theses/2021.bsc.markus. asadauskas.pdf

Azeem, M., Mahmood, J., Aslam, T., & Shahzadi, K. (2022). A Systematic Literature Review on Blockchain Attacks. 2022 *International Conference on Frontiers of Information Technology* (FIT). https://doi.org/10.1109/FIT57066.2022.00050

Bentov, I., Lee, C., Mizrahi, A., & Rosenfeld, M. (2014). Proof of activity: Extending bitcoin's proof of work via proof of stake. *arXiv preprint*. https://arxiv.org/abs/ 1406.5694

Bezuidenhout, R., Nel, W., & Maritz, J. M. (2022). Embedding tamper-resistant, publicly verifiable random number seeds in permissionless blockchain systems. *IEEE Access*, 10, 39911–39925. https://doi.org/10.1109/ACCESS.2022.3165616

Blockchain Council. (2022). Hyperledger vs Corda vs Ethereum: A detailed comparison [Blockchain Council].

Bonneau, J., Miller, A., Clark, J., Narayanan, A., Kroll, J. A., & Felten, E. W. (2015). SoK: Research Perspectives and Challenges for Bitcoin and Cryptocurrencies.

2015 *IEEE Symposium on Security and Privacy*, 104–121. https://doi.org/10.1109/SP.2015.14

Bouraga, S. (2021). A taxonomy of blockchain consensus protocols: A survey and classification framework. *Expert Systems with Applications*, 168, 114384. https://doi. org/10.1016/j.eswa.2020.114384

Bowman, M., Das, D., Mandal, A., & Montgomery, H. (2021). On Elapsed Time Consensus Protocols. *IACR Cryptology ePrint Archive*, 2021, 086. https://eprint.iacr. org/2021/086

Burstcoin. (T.y.). Burstcoin: Official Website [Accessed: 2024-11-21].

Buterin, V. (2013). Ethereum White Paper: A Next Generation Smart Contract & Decentralized Application Platform [Ethereum.org].

Cachin, C., & Vukolić, M. (2017). Blockchain consensus protocols in the wild. *arXiv preprint*, arXiv:1707.01873. https://arxiv.org/abs/1707.01873

Castro, M., & Liskov, B. (1999). Practical Byzantine fault tolerance. *Proceedings of the Third Symposium on Operating Systems Design and Implementation*, 173–186.

Catalini, C., & Gans, J. S. (2016). *Some Simple Economics of the Blockchain* (NBER Working Paper No. 22952). National Bureau of Economic Research. https://doi.org/10.3386/w22952

Chase, B., & MacBrough, E. (2018). Analysis of the XRP Ledger consensus protocol.

arXiv preprint. https://arxiv.org/abs/1802.07242

Chen, L., Xu, L., Shah, N., Gao, Z., Lu, Y., & Shi, W. (2017). On Security Analysis of Proof-of-Elapsed-Time (PoET). *International Symposium on Stabilization, Safety, and Security of Distributed Systems*, 282–297. https://www.researchgate.net/ publication/320246838

Cohen, B., & Network, C. (2021). Chia Blockchain Consensus Algorithm: Proof of Space and Time [Accessed: 2024-10-02].

Corso, A. (2019). *Performance Analysis of Proof-of-Elapsed-Time (PoET) Consensus in the Sawtooth Blockchain Framework* [yük. lis. tezi, University of Oregon]. https://1library.net/document/zk630oey-performance-analysis-proof-elapsed-consensus-sawtooth-blockchain-framework.html

Crosby, M., Pattanayak, P., Verma, S., & Kalyanaraman, V. (2016). Blockchain technology: Beyond bitcoin. *Applied Innovation Review*, (2), 6–19.

De Vries, A. (2018). Bitcoin's growing energy problem. *Joule*, 2(5), 801–805. https://doi.org/10.1016/j.joule.2018.04.016

Debus, J. (2017). *Consensus methods in blockchain systems* (Technical Report). Frankfurt School of Finance & Management, Blockchain Center.

Dinh, T. T. A., Liu, R., Zhang, M., Chen, G., Ooi, B. C., & Wang, J. (2018). Untangling blockchain: A data processing view of blockchain systems. *IEEE Transactions on Knowledge and Data Engineering*, 30(7), 1366–1385. https://doi.org/10.1109/TKDE.2017.2781227

Donovan, A. A., & Kernighan, B. W. (2015). *The Go Programming Language*. Addison-Wesley Professional.

Dubrovsky, I., Wei, Z., & Goldstein, J. (2019). Optical Proof of Work: A Low Energy Consensus Protocol for Blockchain. *IEEE Access*, 7, 9991–10002. https://doi.org/10.1109/ACCESS.2019.2893428

Dwork, C., & Naor, M. (1993). Pricing via processing or combatting junk mail. *Advances in Cryptology—CRYPTO'92*, 139–147. https://doi.org/10.1007/3-540-48071-4_10

Dziembowski, S., Faust, S., Kolmogorov, V., & Pietrzak, K. (2015). Proofs of space. *Advances in Cryptology* – CRYPTO 2015, 585–605. https://doi.org/10.1007/978-3-662-48000-7_29

Ekparinya, P., Gramoli, V., & Jourjon, G. (2019). The attack of the clones against Proof-of-Authority. *arXiv preprint*. https://arxiv.org/abs/1902.10244

Eyal, I., & Sirer, E. G. (2018). Majority is not enough: Bitcoin mining is vulnerable. *Communications of the ACM*, 61(7), 95–102. https://doi.org/10.1145/3212998

Fang, L. (2023). Proof of Stake: Energy Efficiency and Security in Blockchain. *International Journal of Blockchain Studies*, 9, 301–316. https://doi.org/10.1016/j.ijbs. 2023.01.005

Fattahi, J. (2023). A Federated Byzantine Agreement Model to Operate Offline Electric Vehicle Supply Equipment. *IEEE Transactions on Smart Grid*, PP(1), 1–1. https: //doi.org/10.1109/TSG.2023.3307679

Flarecompare. (T.y.). TRON vs EOS vs BitShares [Accessed: 2024-11-17].

Gai, F., Wang, B., Deng, W., & Peng, W. (2018). Proof of Reputation: A Reputation-Based Consensus Protocol for Peer-to-Peer Network. Içinde J. Pei, Y. Manolopoulos, S. Sadiq & J. Li (Ed.), *Database Systems for Advanced Applications*. *DASFAA 2018* (ss. 666–681, C. 10828). Springer, Cham. https://doi.org/10.1007/978-3-319- 91458-9_41

Garcia Ribera, E. (2018). *Design and implementation of a proof-of-stake consensus algorithm for blockchain* [Master's thesis]. Universitat Politècnica de Catalunya [Universitat Politècnica de Catalunya Repository].

Gate.io. (2022). Proof of capacity consensus algorithm [Retrieved from https://www.gate. io/blog/872/proof-of-capacity-consensus-algorithm]. https://www.gate.io/blog/ 872/proof-of-capacity-consensus-algorithm

Gervais, A., Karame, G. O., Wüst, K., Glykantzis, V., Ritzdorf, H., & Capkun, S. (2016). On the security and performance of proof of work blockchains [Provides a comparative view of PoS and PoW in terms of security.]. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 3–16. https://doi.org/10.1145/2976749.2978341

Golan Gueta, G., Abraham, I., Grossman, S., Malkhi, D., Pinkas, B., Reiter, M. K., Tamir, O., & Tomescu, A. (2019). SBFT: A Scalable and Decentralized Trust Infrastructure. *2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 568–580. https://doi.org/10.1109/DSN.2019.00063

Guo, Y., & Liang, C. (2016). Blockchain application and outlook in the banking industry.

Financial Innovation, 2(24). https://doi.org/10.1186/s40854-016-0034-9

Hajian Berenjestanaki, M., Barzegar, H. R., El Ioini, N., & Pahl, C. (2024). Blockchain-Based E-Voting Systems: A Technology Review. *Electronics*, 13(1), 17. https://doi.org/10.3390/electronics13010017

Hao, X., Yu, L., Zhiqiang, L., Zhen, L., & Dawu, G. (2018). Dynamic Practical Byzantine Fault Tolerance. *2018 IEEE Conference on Communications and Network Security (CNS)*, 1–8. https://doi.org/10.1109/CNS.2018.8433150

Hayes, J., Aneiba, A., Gaber, M., Islam, M. S., & Abozariba, R. (2023). FBA-SDN: A Federated Byzantine Approach for Blockchain-Based Collaborative Intrusion Detection in Edge SDN. *2023 IEEE International Conference on Communications Works- hops (ICC Workshops)*, 427–433. https://doi.org/10.1109/ICCWorkshops57953. 2023.10283805

Hazen, B. T., Boone, C. A., Ezell, J. D., & Jones-Farmer, L. A. (2014). Data Quality for Data Science, Predictive Analytics, and Big Data in Supply Chain Management: An Introduction to the Problem and Suggestions for Research and Applications. *International Journal of Production Economics*, 154, 72–80. https://doi.org/10. 1016/j.ijpe.2014.04.018

Hern, A. (2021). New cryptocurrency Chia blamed for hard drive shortages [Retrieved from https : / / www. theguardian . com / technology / 2021 / may / 26 / new - cryptocurrency- chia- blamed- for- hard- drive- shortages]. *The Guardian*. Https://www.theguardian.com/technology/2021/may/26/new- cryptocurrency- chia-blamed-for-hard-drive-shortages

Hossain, M. R., Nirob, F. A., Islam, A., Rakin, T. M., & Al-Amin, M. (2024). A Comprehensive Analysis of Blockchain Technology and Consensus Protocols Across Multilayered Framework. *IEEE Access*, 12, 63087–63129. https://doi.org/10. 1109/ACCESS.2024.3395536

Huang, G., & Foysal, A. (2021). Blockchain in Healthcare. *Technology and Investment*, 12, 168–181. https://doi.org/10.4236/ti.2021.123010

Hussein, Z., Salama, M. A., & El-Rahman, S. A. (2023). Evolution of blockchain consensus algorithms: A review on the latest milestones of blockchain consensus algorithms. *Cybersecurity*, 6(30). https://doi.org/10.1186/s42400-023-00163-y

Ibañez, J. I., & Freier, A. (2023). Reducing the Carbon Footprint of Bitcoin Mining. *Environmental Impact Assessment Review*, 55, 101–114. https://doi.org/10.1016/j.eiar.2023.03.005

Investopedia. (2024). *Proof of Capacity (Cryptocurrency)* [Accessed: 2024-11-21]. https://www.investopedia.com/terms/p/proof-capacity-cryptocurrency.asp

Islam, M. M., Merlec, M. M., & In, H. P. (2022). A comparative analysis of proof- of-authority consensus algorithms: Aura vs Clique. *2022 IEEE International Conference on Services Computing (SCC)*, 327–332. https://doi.org/10.1109/SCC55611.2022.00054

Jiang, S., & Wu, J. (2020). A game-theoretic approach to storage offloading in PoC-based mobile blockchain mining. *Proceedings of the Twenty-first ACM International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing (Mobihoc '20)*, 10 pages. https://doi.org/10.1145/3397166.3409136

Johar, S., Ahmad, N., Asher, W., Cruickshank, H., & Durrani, A. (2021). Research and applied perspective to blockchain technology: A comprehensive survey. *Applied Sciences*, 11(14), 6252. https://doi.org/10.3390/app11146252

Joshi, S. (2021). Feasibility of Proof of Authority as a Consensus Protocol Model. *arXiv preprint.* https://arxiv.org/abs/2109.02480

K, H. P., & S, G. K. (2023). Evolution of Consensus Algorithms in Blockchain Technology. *2023 International Conference on Intelligent Systems for Communication, IoT and Security (ICISCoIS)*, 493–498. https://doi.org/10.1109/ICISCoIS56541.2023. 10100388

Kang, J., Xiong, Z., Niyato, D., Ye, D., Kim, D. I., & Zhao, J. (2019). Toward Secure Blockchain-Enabled Internet of Vehicles: Optimizing Consensus Management Using Reputation and Contract Theory. *IEEE Transactions on Vehicular Technology*, 68(3), 2906–2920. https://doi.org/10.1109/TVT.2019.2894944

Katal, A., Sethi, V., & Lamba, S. (2020). Blockchain consensus algorithms: Study and challenges. *Blockchain applications in IoT ecosystem* (ss. 45–64). Springer. https://doi.org/10.1007/978-3-030-65691-1_4

Kaur, M., Khan, M. Z., Gupta, S., Noorwali, A., Chakraborty, C., & Pani, S. K. (2021). MBCP: Performance Analysis of Large Scale Mainstream Blockchain Consensus Protocols. *IEEE Access*, 9, 80931–80944. https://doi.org/10.1109/ACCESS.2021.3085187

Khobragade, P., & Turuk, A. K. (2023). Blockchain consensus algorithms: *A survey. Blockchain and applications,* 4th international congress (ss. 198–210). Springer. https://doi.org/10.1007/978-3-031-21229-1_19

Kim, M., Kwon, Y., & Kim, Y. (2019). Is Stellar As Secure As You Think? *2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, 377–385. https://doi.org/10.1109/EuroSPW.2019.00048

Kim, S., & Kim, J. (2018). POSTER: Mining with Proof-of-Probability in Blockchain. *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, 841–843. https://doi.org/10.1145/3196494.3201592

King, S., & Nadal, S. (2012). PPCoin: Peer-to-Peer Cryptocurrency with Proof-of-Stake.

Kouhizadeh, M., Saberi, S., & Sarkis, J. (2020). Blockchain technology and the sustainable supply chain: Theoretically exploring adoption barriers. *International Journal of Production Economics*, 231, 107831. https://doi.org/10.1016/j.ijpe.2020.107831

Krawczyk, H., Bellare, M., & Canetti, R. (1997). HMAC: Keyed-hashing for message authentication. *RFC 2104*. https://doi.org/10.17487/RFC2104

Kumar, R., & Gupta, A. (2024). Scalability and Sustainability of PoS. *IEEE Transactions on Blockchain Technology*, 12, 55–68. https://doi.org/10.1109/TBT.2024.1234567

Kumari, T., Kumar, R., & Dwivedi, R. K. (2024). Blockchain-based secure and smart healthcare IoT system using hybrid consensus mechanism with an honest block.

Health Services and Outcomes Research Methodology. https://doi.org/10.1007/s10742-024-00330-9

Lamport, L., Shostak, R., & Pease, M. (1982). The Byzantine generals problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 4(3), 382–401. https://doi.org/10.1145/357172.357176

Li, W., Feng, C., Zhang, L., Xu, H., Cao, B., & Imran, M. A. (2021). A Scalable Multi-Layer PBFT Consensus for Blockchain. *IEEE Transactions on Parallel and Distributed Systems*, 32(5), 1146–1160. https://doi.org/10.1109/TPDS.2020.3042392

Lu, S.-P., Lei, C.-L., & Tsai, M.-H. (2024). An efficient Proof-of-Authority consensus scheme against cloning attacks. *Computer Communications*, 228, 107975. https://doi.org/10.1016/j.comcom.2024.107975

Luo, H., Zhang, Q., Yu, H., Sun, G., & Xu, S. (2023). Applications and Security of PBFT in Private Blockchains. *IEEE Blockchain Letters*, 10, 88–99. https://doi.org/10.1109/IBL.2023.3299320

Luo, Y., Chen, Y., Chen, Q., & Liang, Q. (2018). A New Election Algorithm for DPos Con- sensus Mechanism in Blockchain. 2018 7th International Conference on Digital Home (ICDH), 116–120. https://doi.org/10.1109/ICDH.2018.00029

MakeUseOf. (2024). *What Is Proof of Capacity? The Green Consensus Algorithm* [Accessed: 2024-11-21]. https://www.makeuseof.com/proof- of- capacity- green-consensus-algorithm

Mazières, D. (2015). The Stellar consensus protocol: A federated model for internet-level consensus (tek. rap.). Stellar Development Foundation. https://www.stellar.org/papers/stellar-consensus-protocol.pdf

md5hashing.net. (T.y.). SHA-256 hash of "b4056df6691f8dc72e56302ddad345d65fead3ead9299609a [Erişim tarihi: 2024-11-02].

Menon, A. A., Saranya, T., Sureshbabu, S., & Mahesh, A. S. (2021). A comparative analysis on three consensus algorithms: Proof of Burn, Proof of Elapsed Time,

Proof of Authority. *Advances in Intelligent Systems and Computing* (ss. 369–383, C. 75). Springer. https://doi.org/10.1007/978-981-16-3728-5_28

Meshcheryakov, Y., Melman, A., Evsutin, O., Morozov, V., & Koucheryavy, Y. (2021). On Performance of PBFT Blockchain Consensus Algorithm for IoT-Applications With Constrained Devices. *IEEE Access*, 9, 80559–80570. https://doi.org/10.1109/ACCESS.2021.3085405

Micali, S. (2017). Algorand: The efficient and democratic ledger [Explores innovative approaches to consensus in PoS systems.]. *arXiv preprint*. https://arxiv.org/abs/1607.01341

Mundhra, M., & Rebeiro, C. (2020). SISSLE in consensus-based Ripple: Some Improvements in Speed, Security and Last Mile Connectivity. *arXiv preprint*. https://arxiv.org/abs/2008.01742

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system [Bitcoin.org].

NEM. (2018). Proof-of-Importance [NEM Technical Reference. Accessed: Nov. 12, 2019].

Nextrope. (2022). 4 most popular blockchains: Analysis and comparison of Ethereum, Hyperledger Fabric, Corda and Quorum [Nextrope].

Nguyen, Q., & Cronje, A. (2020). Practical Byzantine Fault Tolerance: Security and Performance Analysis. *IEEE Transactions on Distributed Systems*, 8, 800–820. https://doi.org/10.1109/TDS.2020.3005432

Nofer, M., Gomber, P., Hinz, O., & Schiereck, D. (2017). Blockchain. *Business & Information Systems Engineering*, 59(3), 183–187. https://doi.org/10.1007/s12599-017-0467-3

Nugroho, E. W. (2023). Analyzing the Federated Byzantine Agreement Blockchain Network for Liveness Using Multiple Online Analytical Processing. *2023 7th International Conference on Information Technology (InCIT)*, 135–140. https://doi. org/10.1109/InCIT60207.2023.10413163

Pan, H., Lin, Y., & Wang, Y. (2023). Improvement of Hierarchical Byzantine Fault Tolerance Algorithm in RAFT Consensus Algorithm Election. *Applied Sciences*, 13(16), 9125. https://doi.org/10.3390/app13169125

Park, S., Kwon, A., Fuchsbauer, G., Gaži, P., Alwen, J., & Pietrzak, K. (2018). SpaceMint: A cryptocurrency based on proofs of space [Retrieved from https://dci.mit.edu/ dci-news/2019/3/20/spacemint-a-cryptocurrency-based-on-proofs-of-space]. *22nd International Conference on Financial Cryptography and Data Security (Financial Crypto 2018).* https://dci.mit.edu/dci-news/2019/3/20/spacemint-a- cryptocurrency-based-on-proofs-of-space

Perera, S., Nanayakkara, S., Rodrigo, M. N. N., Senaratne, S., & Weinand, R. (2020). Blockchain technology: Is it hype or real in the construction industry? *Journal of Industrial Information Integration*, 17, 100125. https://doi.org/10.1016/j.jii.2020.100125

Pilkington, M. (2016). Blockchain technology: Principles and applications. *Research Handbook on Digital Transformations* (ss. 225–253). Edward Elgar Publishing.

Rebello, G. A. F., Camilo, G. F., Guimarães, L. C. B., de Souza, L. A. C., Thomaz, G. A., & Duarte, O. C. M. B. (2022). A security and performance analysis of proof-based consensus protocols. *Annals of Telecommunications*, 77(4-5), 517–537. https://doi.org/10.1007/s12243-021-00896-2

Roets, J. (2024). How Joe Roets built Dragonchain, a hybrid blockchain platform [HackerNoon].

Roy, S., & Shyamasundar, R. K. (2023). An analysis of hybrid consensus in blockchain protocols for correctness and progress. *Springer*. https://doi.org/10.1007/ 978-3-031-37586-6_24

Saleh, F. (2020). Blockchain without waste: Proof-of-stake. *The Review of Financial Studies*, 34(3), 1156–1190. https://doi.org/10.1093/rfs/hhaa075

Salimitari, M., & Chatterjee, M. (2018). A survey on consensus protocols in blockchain for IoT networks. *arXiv preprint* arXiv:1809.05613.

Schneier, B. (1996). *Applied Cryptography: Protocols, Algorithms, and Source Code in C* (2nd). John Wiley & Sons.

Shi, W., Chen, X., & Liu, F. (2021). Energy Consumption and Carbon Emission of Blockchain Technology. *Journal of Sustainable Computing*, 17, 10–22. https : //doi.org/10.1000/j.suscom.2021.03.001

Signum Network. (2021). About Signum [Retrieved from https://signum.network/about]. https://signum.network/about

Sukhwani, H., Martinez, J., Chang, X., Trivedi, K., & Rindos, A. (2017). Performance Modeling of PBFT Consensus Process for Permissioned Blockchain Network (Hyperledger Fabric). *2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS)*, 253–255. https://doi.org/10.1109/SRDS.2017.36

Sun, W., Liu, J., Yue, Y., & Wang, P. (2020). Joint resource allocation and incentive design for blockchain-based mobile edge computing. *IEEE Transactions on Wireless Communications*, 19(9), 6050–6062. https://doi.org/10.1109/TWC.2020.2999721

Tapscott, D., & Tapscott, A. (2016). *Blockchain Revolution: How the Technology behind Bitcoin Is Changing Money, Business, and the World.* Penguin.

Thanujan, T., Rajapakse, C., & Wickramaarachchi, D. (2021). A community-based hybrid blockchain architecture for the organic food supply chain. *2021 International Research Conference on Smart Computing and Systems Engineering (SCSE)*, 77–83. https://doi.org/10.1109/SCSE53661.2021.9568325

Trestioreanu, L., Scheidt, F., Shbair, W., Francois, J., Magoni, D., & State, R. (2024). To Squelch or not to Squelch: Enabling Improved Message Dissemination on the XRP Ledger. *NOMS 2024 - 2024 IEEE Network Operations and Management Symposium*, 1–7. https://doi.org/10.1109/NOMS59830.2024.10575886

Tripathi, A. K., Jain, A., Kumar, A., Singh, S., & Shukla, B. (2024). Exploring consensus algorithms: A comprehensive examination and comparative analysis. *AIP Conference Proceedings*, 3168(1), 020036. https://doi.org/10.1063/5.0221104

Venkatesan, K., & Rahayu, S. B. (2024). Blockchain security enhancement: An approach towards hybrid consensus algorithms and machine learning techniques. *Scientific Reports*, 14(1149). https://doi.org/10.1038/s41598-024-51578-7

Vyas, K., & Deshmukh, A. (2023). A Survey Paper on Blockchain Technology and Consensus Algorithms. *2023 11th International Conference on Emerging Trends in Engineering & Technology - Signal and Information Processing (ICETET-SIP)*, 1–6. https://doi.org/10.1109/ICETET-SIP58143.2023.10151598

Wang, Q., Li, R., Wang, Q., Chen, S., & Xiang, Y. (2022). Exploring unfairness on Proof of Authority: Order manipulation attacks and remedies. *arXiv preprint*. https://arxiv.org/abs/2203.03008

Wang, Y., Cai, S., Lin, C., Chen, Z., Wang, T., Gao, Z., & Zhou, C. (2019). Study of Blockchain's Consensus Mechanism Based on Credit. *IEEE Access*, 7, 10224–10231. https://doi.org/10.1109/ACCESS.2019.2891065

Waves Documentation. (n.d.). *Leased Proof of Stake* (LPoS) [Accessed: 2024-10-28]. https://docs.waves.tech/en/blockchain/waves-protocol/leased-proof-of-stake-lpos#core-concepts-of-waves-lpos

Xiao, B., Jin, C., Li, Z., Zhu, B., Li, X., & Wang, D. (2022). Proof of importance: A consensus algorithm for importance based on dynamic authorization. *In IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WIIAT '21)* (pp. 510–513). Association for Computing Machinery. https://doi.org/10.1145/3498851.3499007

Xie, W., Cai, Z., Zhang, Y., & Yu, J. (2019). A survey of blockchain technology applied to smart cities: Research issues and challenges. *IEEE Communications Surveys & Tutorials*, 21(3), 2794–2830. https://doi.org/10.1109/COMST.2019.2899617

XinFin Network. (2024). XDC Network-XDC Consensus Algorithm White Paper.

Xu, G., Liu, Y., & Khan, P. W. (2020). Improvement of the DPoS consensus mechanism in blockchain based on vague sets. *IEEE Transactions on Industrial Informatics*, 16(6), 4252–4260. https://doi.org/10.1109/TII.2019.2955719

Xu, X., Pautasso, C., Zhu, L., Gramoli, V., Ponomarev, A., Chen, S., & Muhlberg, J. T. (2016). The blockchain as a software connector. *2016 13th Working IEEE/IFIP Conference on Software Architecture (WICSA)*, 182–191. https://doi.org/10.1109/WICSA.2016.21

Yaga, D., Mell, P., Roby, N., & Scarfone, K. (2019). *Blockchain technology overview* (tek. rap.). National Institute of Standards ve Technology (NIST). https://doi.org/10. 6028/NIST.IR.8202

Yakovenko, A. (2018). Solana: A New Architecture for a High Performance Blockchain [Accessed: 2024-12-02].

Yang, F., Zhou, W., Wu, Q., Long, R., Xiong, N. N., & Zhou, M. (2019). Delegated proof of stake with downgrade: A secure and efficient blockchain consensus algorithm with downgrade mechanism. *IEEE Access*, 7, 118541–118555. https://doi.org/10. 1109/ACCESS.2019.2935149

Yang, J., Dai, J., Gooi, H. B., Nguyen, H. D., & Paudel, A. (2022). A proof-of-authority blockchain-based distributed control system for islanded microgrids. *IEEE Transactions on Industrial Informatics*, 18(11), 8287–8297. https://doi.org/10.1109/ TII.2022.3142755

Yao, W., Ye, J., Murimi, R., & Wang, G. (2021). A Survey on Consortium Blockchain Consensus Mechanisms. *arXiv preprint*, arXiv:2102.12058. https://doi.org/10. 48550/arXiv.2102.12058

Yazdinejad, A., Parizi, R. M., Dehghantanha, A., Karimipour, H., Srivastava, G., & Aledhari, M. (2021). Enabling Drones in the Internet of Things With Decentralized Blockchain-Based Security. *IEEE Internet of Things Journal*, 8(8), 6406–6415. https://doi.org/10.1109/JIOT.2020.3015382

Yusoff, J., Mohammad, Z., & Anuar, M. (2022). A Review: Consensus Algorithms on Blockchain. *Journal of Computer and Communications*, 10, 37–50. https://doi. org/10.4236/jcc.2022.109003

Zhang, S., & Lee, J.-H. (2019). Double-spending with a Sybil attack in the Bitcoin decentralized network. *IEEE Transactions on Industrial Informatics*, 15(10), 5715–5722. https://doi.org/10.1109/TII.2019.2921566

Zhang, Y., & Lee, J. H. (2023). Improving Voting of Block Producers for Delegated Proof- of-Stake with Quadratic Delegate. *IEEE Transactions on Blockchain*, 4, 45–56. https://doi.org/10.1109/TBC.2023.10255193

Zheng, Z., Xie, S., Dai, H., Chen, X., & Wang, H. (2017). An overview of blockchain technology: Architecture, consensus, and future trends. *2017 IEEE 6th International Congress on Big Data (BigData Congress)*, 557–564. https://doi.org/10. 1109/BigDataCongress.2017.85

Zhou, Q., Huang, H., Zheng, Z., & Bian, J. (2020). Solutions to scalability of blockchain: A survey. *IEEE Access*, 8, 16440–16455. https://doi.org/10.1109/ACCESS.2020. 2967218

Zhou, S., Li, K., Xiao, L., Cai, J., Liang, W., & Castiglione, A. (2023). A Systematic Review of Consensus Mechanisms in Blockchain. *Mathematics*, 11(10), 2248. https://doi.org/10.3390/math11102248

EKLER

EK-A: ALGORİTMA UYGULAMA KODU

```
package main import (

"bufio" "crypto/sha256" "encoding/hex" "fmt" "math/rand" "os"

"strconv" "strings" "sync"

"time"

)

type Block struct { Index       int Timestamp string Data      string

Hash    string PrevHash string Validator string

}

type Blockchain struct { Chain []Block

Mutex sync.Mutex

}

func calculateHash(index int, timestamp string, data string, prevHash string, validator string) string { value := fmt.Sprintf("%d%s%s%s%s", index, timestamp, data, prevHash, validator)

hash := sha256.New() hash.Write([]byte(value))

return hex.EncodeToString(hash.Sum(nil))

}

func generateBlock(prevBlock Block, data string, selectedValidator string) Block { timestamp := time.Now().Format(time.RFC3339)

newBlock := Block{

Index:  prevBlock.Index + 1, Timestamp: timestamp,
```

```
Data:    data,

PrevHash: prevBlock.Hash, Validator: selectedValidator,

}

newBlock.Hash = calculateHash(newBlock.Index, newBlock.Timestamp,
newBlock.Data, newBlock.PrevHash, newBlock.Validator)

return  newBlock

}

func getRandomValidators(allValidators []string) []string {
rand.Seed(time.Now().UnixNano())

shuffledValidators := make([]string, len(allValidators)) copy(shuffledValidators,
allValidators) rand.Shuffle(len(shuffledValidators), func(i, j int) {

shuffledValidators[i],  shuffledValidators[j]  =  shuffledValidators[j],
shuffledValidators[i]

})

return shuffledValidators[:3]

}


func voteForValidator(validators []string) string { reader := bufio.NewReader(os.Stdin)

voteCounts := make(map[string]int)

fmt.Println("Seciminizi yapin [", strings.Join(validators, ", "), "] arasindan 0’a basarak
oylamayi sonlandirabilirsiniz.")

for {

for i, validator := range validators { fmt.Printf("%d: %s\n", i+1, validator)

}

fmt.Print("Seciminizi yapin (1, 2, 3 veya 0): ") choiceStr, _ := reader.ReadString(’\n’)
choiceStr = strings.TrimSpace(choiceStr)
```

```
choice, err := strconv.Atoi(choiceStr)

if err == nil {

if choice == 0 { break

} else if choice >= 1 && choice <= len(validators) { selectedValidator :=
validators[choice-1] voteCounts[selectedValidator]++

fmt.Printf("%s icin oy verdiniz.\n", selectedValidator)

} else {

fmt.Println("Gecersiz secim. Lutfen 1, 2, 3 veya 0 girin.")

}

} else {

fmt.Println("Gecersiz giris. Lutfen bir sayi girin.")

}

}

maxVotes := 0 selectedValidator := ""

for validator, votes := range voteCounts { if votes > maxVotes {

maxVotes = votes selectedValidator = validator

}

}

fmt.Printf("Oylama tamamlandi. Toplam oylar: %v\n", voteCounts) fmt.Printf("Secilen
dogrulayici: %s (%d oy)\n", selectedValidator, maxVotes) return selectedValidator

}

func getInitialValidators(numValidators int) []string { allValidators := make([]string,
numValidators)

for i := 0; i < numValidators; i++ { allValidators[i] = fmt.Sprintf("Node%d", i+1)

}

return allValidators
```

```
}

func isValid(blockchain Blockchain) bool {

for i := 1; i < len(blockchain.Chain); i++ { currentBlock := blockchain.Chain[i]
prevBlock := blockchain.Chain[i-1]

calculatedHash := calculateHash(currentBlock.Index, currentBlock.Timestamp,
currentBlock.Data, currentBlock.PrevHash, currentBlock.Validator)

if currentBlock.Hash != calculatedHash {

fmt.Printf("Blok %d: Hash hatali. Beklenen: %s, Bulunan: %s\n", currentBlock.Index,
calculatedHash, currentBlock.Hash)

return false

}

if currentBlock.PrevHash != prevBlock.Hash {

fmt.Printf("Blok %d: Onceki hash hatali. Beklenen: %s, Bulunan: %s\n",
currentBlock.Index, prevBlock.Hash, currentBlock.PrevHash)

return false

}

}

return true

}

func main() {

allValidators := getInitialValidators(50)

genesisBlock := Block{ Index:          0,

Timestamp: time.Now().Format(time.RFC3339), Data:        "Genesis Blok",

PrevHash: "", Validator: "Genesis",

}
```

```
genesisBlock.Hash = calculateHash(genesisBlock.Index, genesisBlock.Timestamp, genesisBlock.Data, genesisBlock.PrevHash, genesisBlock.Validator)

fmt.Println("Genesis Blok Bilgileri:") fmt.Printf("Index: %d\n", genesisBlock.Index)

fmt.Printf("Zaman Damgasi: %s\n", genesisBlock.Timestamp) fmt.Printf("Veri: %s\n", genesisBlock.Data) fmt.Printf("Hash: %s\n", genesisBlock.Hash) fmt.Printf("Onceki Hash: %s\n", genesisBlock.PrevHash) fmt.Printf("Dogrulayici: %s\n", genesisBlock.Validator) fmt.Println("            ")

blockchain := Blockchain{

Chain: []Block{genesisBlock},

}

for i := 0; i < 50; i++ {

selectedValidators := getRandomValidators(allValidators) selectedValidator := voteForValidator(selectedValidators)

fmt.Printf("Blok %d icin secilen dogrulayici: %s\n", i+1, selectedValidator)

newBlock := generateBlock(blockchain.Chain[len(blockchain.Chain)-1], fmt.Sprintf("Data%d", i+1), selectedValidator)

blockchain.Chain = append(blockchain.Chain, newBlock) fmt.Printf("Blok %d olusturuldu:\n", i+1) fmt.Printf("Index: %d\n", newBlock.Index) fmt.Printf("Zaman Damgasi: %s\n", newBlock.Timestamp) fmt.Printf("Veri: %s\n", newBlock.Data) fmt.Printf("Hash: %s\n", newBlock.Hash) fmt.Printf("Onceki Hash: %s\n", newBlock.PrevHash) fmt.Printf("Dogrulayici: %s\n", newBlock.Validator) fmt.Println("
            ")

}

if isValid(blockchain) { fmt.Println("Blok zinciri gecerli.")

} else {

fmt.Println("Blok zinciri gecersiz!")

}}
```

EK-B: ALGORİTMA UYGULAMA TEST KODU

```
package main

//Testler

//go test -v

//go test -bench=.

//go test -bench=. -cpuprofile=cpu.out -memprofile=mem.out

//go tool pprof mem.out

//go tool pprof cpu.out

import ( "testing" "time"

)

func TestCalculateHash(t *testing.T) { index := 1

timestamp := "2024-12-21T12:00:00Z"

data := "Test Data" prevHash := "abc123" validator := "Node1"

expectedHash := calculateHash(index, timestamp, data, prevHash, validator)
calculatedHash := calculateHash(index, timestamp, data, prevHash, validator)

if expectedHash != calculatedHash {

t.Errorf("Beklenen: %s, Hesaplanan: %s", expectedHash, calculatedHash)

}

}

func TestGenerateBlock(t *testing.T) { prevBlock := Block{

Index:  0,

Timestamp: "2024-12-21T12:00:00Z",

Data:    "Genesis Blok",
```

```
Hash:   "genesisHash", PrevHash: "",
Validator: "Genesis",
}
newBlock := generateBlock(prevBlock, "New Block Data", "Node1") if
newBlock.Index != prevBlock.Index+1 {
t.Errorf("Blok index hatası. Beklenen: %d, Bulunan: %d", prevBlock.Index+1,
newBlock.Index)
}
if newBlock.PrevHash != prevBlock.Hash {
t.Errorf("Önceki hash hatası. Beklenen: %s, Bulunan: %s", prevBlock.Hash,
newBlock.PrevHash)
}
}
func TestBlockchainValidity(t *testing.T) { blockchain := Blockchain{
Chain: []Block{
{
Index:  0,
Timestamp: "2024-12-21T12:00:00Z",
Data:    "Genesis Blok",
Hash:   "genesisHash", PrevHash: "",
Validator: "Genesis",
},
},
}
newBlock := generateBlock(blockchain.Chain[0], "Test Data", "Node1")
blockchain.Chain = append(blockchain.Chain, newBlock)
```

```
if !isValid(blockchain) {
t.Errorf("Blok zinciri geçerlilik testi başarısız oldu")
}
}
func BenchmarkGenerateBlock(b *testing.B) { prevBlock := Block{
Index:  0,
Timestamp: time.Now().Format(time.RFC3339), Data:       "Genesis Blok",
Hash:   "genesisHash", PrevHash: "",
Validator: "Genesis",
}
for i := 0; i < b.N; i++ {
_ = generateBlock(prevBlock, "Benchmark Data", "Node1")
}
}
func BenchmarkGenerateBlockParallel(b *testing.B) { prevBlock := Block{
Index:  0,
Timestamp: time.Now().Format(time.RFC3339), Data:       "Genesis Blok",
Hash:   "genesisHash", PrevHash: "",
Validator: "Genesis",
}
b.RunParallel(func(pb *testing.PB) { for pb.Next() {
_ = generateBlock(prevBlock, "Parallel Data", "Node1")
}
})
}
```

ÖZGEÇMİŞ

Tuğba ÇALIŞ, Lise öğrenimini Ankara/Mamak- Cumhuriyet Anadolu Lisesi'nde tamamladı. 2008 yılında girdiği Çankaya Üniversitesi Bilgisayar Mühendisliği (İngilizce -Tam burslu) Bölümü'nden, 2013 yılında mezun oldu. 2018 yılında Trakya Üniversitesi Bilgisayar Mühendisliği Bölümünde Yüksek Lisans eğitimini tamamladı. 2015 yılından beri Trakya Üniversitesi Bilgisayar Mühendisliği Bölümünde Araştırma Görevlisi olarak çalışmaktadır.