T.C.

BİLECİK ŞEYH EDEBALİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

ELEKTRONİK VE BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

BLOKZİNCİR TABANLI AKILLI SÖZLEŞME KULLANARAK GÜVENLİ VERİ SAKLAMA VE VERİ DOĞRULAMA

DOKTORA TEZİ

SEFA TUNÇER

TEZ DANIŞMANI

PROF. DR. CİHAN KARAKUZU

BİLECİK, 2023

10549014

T.C.

BİLECİK ŞEYH EDEBALİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

ELEKTRONİK VE BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

BLOKZİNCİR TABANLI AKILLI SÖZLEŞME KULLANARAK GÜVENLİ VERİ SAKLAMA VE VERİ DOĞRULAMA

DOKTORA TEZİ

SEFA TUNÇER

TEZ DANIŞMANI

PROF. DR. CİHAN KARAKUZU

BİLECİK, 2023

10549014

BEYAN

"Blokzincir Tabanlı Akıllı Sözleşme Kullanarak Güvenli Veri Saklama ve Veri Doğrulama" adlı doktora tezi hazırlık ve yazımı sırasında bilimsel araştırma ve etik kurallarına uyduğumu, başkalarının eserlerinden yararlandığım bölümlerde bilimsel kurallara uygun olarak atıfta bulunduğumu, kullandığım verilerde herhangi bir tahrifat yapmadığımı, tezin herhangi bir kısmının Bilecik Şeyh Edebali Üniversitesi veya başka bir üniversitede başka bir tez çalışması olarak sunulmadığını, aksinin tespit edileceği muhtemel durumlarda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Bu çalışmanın,
Bilimsel Araştırma Projeleri (BAP), TÜBİTAK veya benzeri kuruluşlarca desteklenmesi durumunda; projenin ve destekleyen kurumun adı proje numarası ile birlikte, ETİK KURUL onayı alınması durumunda ise ETİK KURUL tarih karar ve sayı bilgilerinin beyan edilmesi gerekmektedir.

DESTEK ALINMIŞTIR		**DESTEK ALINMAMIŞTIR**	**X**

Destek alındı ise;	
Destekleyen kurum;	
Desteğin Türü	**Proje Numarası**
1- BAP (Bilimsel Araştırma Projesi)	
2- TÜBİTAK	
Diğer;	
ETİK KURUL onayı var ise;	
ETİK KURUL karar tarih/sayı:	

Sefa TUNÇER

Tarih

İmza

ÖN SÖZ

Bu tez çalışmasının yazılmasında ve akademik gelişim sürecimde, desteğini hiçbir zaman eksik etmeyen ve karşılaştığım her problemin çözümünde bilgi ve tecrübelerinden istifade ettiğim, benim için bir danışmandan daha fazlası olan değerli hocam Prof. Dr. Cihan KARAKUZU ’ya; tezimin bilimsel olarak ilerlemesine olan katkılarından dolayı değerli jüri üyeleri sayın Prof. Dr. Uğur YÜZGEÇ ve Prof. Dr. Turgay Tugay BİLGİN ’e sonsuz teşekkürlerimi sunuyorum. Bu tez çalışması dolayısıyla TÜBİTAK 2211/C Yurt İçi Öncelikli Alanlar Doktora Burs Programı kapsamında Bilgi Güvenliği alanında desteklenen bir bursiyer olduğumdan dolayı TÜBİTAK’a teşekkürlerimi sunuyorum.

Ayrıca okul ve meslek hayatım boyunca maddi-manevi her türlü desteğini benden esirgemeyen çok değerli anneme, çok değerli babama ve çok değerli kardeşlerime sonsuz teşekkürlerimi sunuyorum.

Sefa TUNÇER

2023

ÖZET

BLOKZİNCİR TABANLI AKILLI SÖZLEŞME KULLANARAK GÜVENLİ VERİ SAKLAMA VE VERİ DOĞRULAMA

Hyperledger Fabric ve Ethereum blokzincir platformları, merkezi olmayan, güvenli ve şeffaf bir ortam sağlayarak öne çıkan ve farklı kullanım senaryolarına uygun olan akıllı sözleşme kullanan platformlardır. Özellikle kurumsal uygulamalarda tercih edilen Hyperledger Fabric modüler yapısı, yetkilendirme ve kimlik yönetimi mekanizmaları, kanal sistemi ve işlemle ilgili özel akıllı sözleşmelerin kullanımına imkan veren yapısıyla dikkat çekmektedir. Askeri, tıbbi görüntülerin veya gizliliği üst seviyede önemli olan görüntülerin/belgelerin saklanması bazı kurumlar açısından büyük önem arz etmektedir. Bu tez kapsamında sunulan HFSecImg çalışması veri güvenliği ve veri doğrulama gibi önemli problemlere çözüm önerisi olarak oluşturulmuştur. Bu çalışma, blokzincir teknolojisinin kullanımı ile veri güvenliği ve doğrulama açısından etkin bir çözüm sunmaktadır. Bu çözümde, görüntü verilerinin güvenli bir şekilde şifrelenerek paydaşlar arasında dağıtılması ve güvenilir bir şekilde saklanmasının yaynında paydaş imzalarının ve şifrelenen görüntünün veri doğrulamasının akıllı sözleşme kullanılarak sağlanması da veri doğrulama problemini etkin çözmeye yöneliktir. Bu yaklaşımın ilk adımında kaotik sistem tabanlı görüntü şifreleme yöntemiyle görüntü şifrelenmiştir. Şifrelemede kullanılan anahtar Shamir'in sır paylaşım yöntemiyle belirlenerek paydaşlar arasındaki özel bilginin güvenliğinin artırılması hedeflenmiştir. Şifrelenen görüntüye ait özet bilgisi ve paydaşların dijital imzaları Hyperledger Fabric blokzincir ağında saklanmaktadır. Ayrıca paydaşların imzalarında bulunan özel ve açık anahtarlar RSA şifreleme algoritması kullanılarak oluşturulmuştur. Veri bütünlüğü ve güvenlik temelli oluşturulan çalışmada akıllı sözleşme kullanımı vurgulanmaktadır. Dağıtık defterde tutulan paydaşların imzaları ve şifrelenen görüntü Hyperledger akıllı sözleşme mekanizmasıyla doğrulama sürecinden geçmektedir. Bu sayede blokzincir teknolojisi kullanılarak güvenlik ve veri bütünlüğü sağlanmıştır. Şifrelenen görüntünün bulut sisteminde depolanması sağlanarak ağda bulunan herkese açık, erişimi kolay ve güvenlik riskinin düşük olması amaçlanmıştır. Yapılan analizler ve kullanılan mekanizmalar başarılı bir güvenli veri saklama ve veri doğrulama mekanizmasının oluşturulduğunu göstermiştir. Ethtrace adı verilen ikinci çalışmada, blokzincir teknolojisinin şeffaf bağış uygulamalarında kullanımı ve Ethereum tabanlı akıllı sözleşmelerin kullanımı ile oluşacak avantajlar konusu ele alınmıştır. Çalışmada bu konuya ilişkin bir örnek

uygulama geliştirilmiştir. Bağış sürecinin otomatikleştirilmesi, bağışçılar ve alıcı kuruluşlar arasındaki anlaşmazlıkları gidermek ve güvenli bağış yönetiminin sağlanması amaçlanmıştır. Uygulamanın şeffaflık, izlenebilirlik ve güvenlik açısından değerlendirildiğinde başarılı sonuçlar verdiği görülmüştür. Bu çalışma, bağış sürecinin şeffaf ve güvenli hale getirilmesini ve bağışçıların bağışlarının etkili bir şekilde kullanıldığından emin olmalarını sağlayarak toplumsal faydayı artırmayı hedeflemektedir. Aynı zamanda, birden fazla kuruluşa aynı anda bağış yapma imkanı sunularak daha geniş bir bağışçı kitlesine hitap edebilmesi, bağış uygulamaları ve blokzincir teknolojisi arasında olası bir sinerjiyi ortaya koyması açısından literatüre katkı sağlanmıştır. Bu tez çalışması kapsamında gerçeklenen HFSecImg ve Ethtrace blokzincir uygulamalarının ortak yönü blokzincir akıllı sözleşmelerinin kullanılmasıdır. Ayrıca iki farklı uygulamaya ait akıllı sözleşmelerin birbirlerine göre üstünlükleri, sakıncaları ve kullanılması gereken alanlar detaylı bir şekilde incelenmiştir. Uygulamaların farklılıkları gizlilik, güvenlik, maliyet vb. birçok açıdan incelenmiştir. Sonuç olarak geliştirilmesi gereken uygulamaya göre seçilmesi gereken platform ile ilgili bilgilere yer verilmiştir. Blokzincir platformu seçiminin geliştirilen uygulamanın yapısına bağlı olarak çok önemli olduğu gösterilmiştir.

Anahtar Kelimeler: Blokzincirde güvenlik, Hyperledger Fabric, Akıllı sözleşme, Görüntü şifreleme, Ethereum, Sır paylaşımı, Dijital imza.

ABSTRACT

SECURE DATA STORAGE AND DATA VERIFICATION USING BLOCKCHAIN BASED SMART CONTRACT

Hyperledger Fabric and Ethereum blockchain platforms are platforms that use smart contracts that stand out by providing a decentralized, secure and transparent environment and are suitable for different usage scenarios. Hyperledger Fabric, which is especially preferred in corporate applications, draws attention with its modular structure, authorization and identity management mechanisms, channel system and its structure that allows the use of special smart contracts related to the transaction. It is of great importance for some institutions to keep military, and medical images or images/documents with high confidentiality. The HFSecImg study presented within the scope of this thesis was created as a solution proposal to important problems such as data security and data validation. This study offers an effective solution in terms of data security and verification with the use of blockchain technology. This solution is aimed to effectively solve the data validation problem by using a smart contract to securely encrypt the image data, distribute it among the stakeholders and store it reliably, ensuring stakeholder signatures and data verification of the encrypted image in the broadcast. In the first step of this approach, the image is encrypted with the chaotic system-based image encryption method. It is aimed to increase the security of private information between stakeholders by determining the key used in encryption by Shamir's Secret Sharing method. The summary information of the encrypted image and the digital signatures of the stakeholders are stored in the Hyperledger Fabric blockchain network. In addition, the private and public keys in the signatures of the stakeholders were created using the RSA encryption algorithm. The use of smart contracts is emphasized in the study, which is based on data integrity and security. The signatures of the stakeholders and the encrypted image kept in the distributed ledger go through the verification process with the Hyperledger smart contract mechanism. In this way, security and data integrity are ensured by using blockchain technology. It is aimed that the encrypted image is stored in the cloud system, open to everyone on the network, easy to access and low-security risk. The analyzes and the mechanisms used have shown that a successful secure data storage and data validation mechanism has been established. In the second study, called Ethtrace, the use of blockchain technology in transparent donation applications and the advantages of using Ethereum-based smart contracts are discussed. In this study, a

sample application has been developed for this issue. It is aimed to automate the donation process, resolve disputes between donors and recipient organizations, and provide secure donation management. When the application is evaluated in terms of transparency, traceability and security, it has been seen that it gives successful results. This study aims to increase the social benefit by making the donation process transparent and secure, and by ensuring that donors' donations are used effectively. At the same time, by offering the opportunity to donate to more than one organization at the same time, it has contributed to the literature in terms of appealing to a wider donor mass and revealing a possible synergy between donation applications and blockchain technology. The common aspect of HFSecImg and Ethtrace blockchain applications implemented within the scope of this thesis is the use of blockchain smart contracts. In addition, the advantages, disadvantages and areas of use of smart contracts belonging to two different applications are examined in detail. The differences between the applications are privacy, security, cost, etc. have been studied from many perspectives. As a result, information about the platform that should be selected according to the application to be developed is given. It has been shown that the choice of blockchain platform is very important depending on the structure of the developed application.

Keywords: Security in blockchain, Hyperledger Fabric, Smart contract, Image encryption, Ethereum network, Secret sharing, Digital signature.

İÇİNDEKİLER

Sayfa

KISALTMALAR VE SİMGELER LİSTESİ

ABE : Öznitelik Tabanlı Şifreleme (Attribute-Based Encryption)

BFT : Bizans Hata Toleransı (Byzantine Fault Tolerance)

CA : Sertifika Yetkilisi (Certificated Authority)

CCM : Kaotik Cat Haritası (Chaotic Cat Map)

CFT : Çökme Hatası Toleransı (Crash Fault Tolerans)

CLI : Komut Satırı Arayüzü (Command Line Interface)

CPU : Merkezi İşlem Birimi (Central Processing Unit)

DApps : Merkezi Olmayan Uygulamalar

DPoS : Yetkilendirilmiş Hisse Kanıtı (Delegated Proof of Stake)

DSA : Dijital İmza Algoritması

ECDSA : Eliptik Eğri Dijital İmza Algoritması

ERC-20 : Kriptopara Standardı (Ethereum Request For Comments)

ESCC/VSCC : Onay Sistemi/Doğrulama Sistemi Zincir Kodu

Ethtrace : İzlenebilir Ethereum Bağış Uygulaması

EVM : Ethereum Sanal Makinesi

GPU : Grafik İşlem Birimi (Graphical Processing Unit)

Hash : Özet, Karma Değer

HLF : Hyperledger Fabric

HFSecImg : Hyperledger Fabric'te Veri Güvenliği Uygulaması

IoT : Nesnelerin İnterneti (Internet of Things)

IIoT : Endüstriyel Nesnelerin İnterneti

IoMT : Tıbbi Nesnelerin İnterneti

IPFS : Gezegenler Arası Dosya Sistemi

NPCR : Piksel Değişim Hızı (Number of Pixel Change Rate)

PAEKS : Açık Anahtar Kimlik Doğrulamalı Aranabilir Şifreleme

PBFT : Pratik Bizans Hata Toleransı

PoA : Yetki Kanıtı (Proof of Authority)

PoC : Kapasite Kanıtı (Proof of Capacity)

PoET : Geçen Sürenin Kanıtı (Proof of Elapsed Time)

PoH : Tarih Kanıtı (Proof of History)

PoS : Hisse Kanıtı (Proof of Stake)

PoW : İş Kanıtı (Proof of Work)

P2P : Eşten Eşe (Peer to peer)

RSA : RSA Şifreleme Algoritması

SHA-256 : 256 Bit Güvenli Özet Algoritması (Secure Hash Algorithm)

SSS : Shamir'in Sır Paylaşımı (Shamir's Secret Sharing)

TLS : Taşıma Katmanı Güvenliği (Transport Layer Security)

TPS : Saniyedeki İşlem Sayısı (Transaction Per Second)

UACI : Birleşik Ortalama Değiştirilmiş Yoğunluk

zk-SNARK : Sıfır-Bilgi Özlü İnteraktif Olmayan Bilginin Argümanı

ŞEKİLLER LİSTESİ

Sayfa

TABLOLAR LİSTESİ

Sayfa

1. GİRİŞ

Günümüzde veri güvenliği ve gizliliğinin büyük önem taşıması, özellikle dijital medya dosyalarının paylaşımı ve saklanması söz konusu olduğunda büyük önem taşımaktadır. Ayrıca blokzincirde veri güvenliği ve gizliliği konularının günümüzde tartışılmaya devam ettiği görülmektedir. Gelişen teknoloji ile birlikte bu verilerin güvenli bir şekilde paylaşılması ve saklanması gerekliliği blokzincir teknolojisini kullanımını önemli bir noktaya taşımıştır. Bundan dolayı HFSecImg çalışmasında sır paylaşım şeması, RSA şifreleme ve kaotik tabanlı şifreleme gibi güçlü kriptografik algoritmaların birleşiminden oluşan bir güvenli veri paylaşım mekanizması ve blokzincir tabanlı bir doğrulama sistemi geliştirilerek veri güvenliği üzerinde yoğunlaşılmıştır. Güvenli veri paylaşımı ve veri saklama konusunda blokzincir tabanlı çözümlerin sağladığı potansiyeli vurgulayarak gelecekteki dijital veri güvenliği uygulamaları için değerli bir katkı sağlamak amacıyla bu çalışma yapılmıştır.

Bağış uygulamaları, yardımseverlerin ve bağışçıların ihtiyaç sahiplerine destek olmasını sağlayan önemli araçlardır. Özellikle bağışların izlenebilirliği ve şeffaflığı açısından mevcut bağış sistemleri bazı zorluklarla karşı karşıya kalmaktadır. Ethtrace çalışması blokzincir tabanlı akıllı sözleşmeleri kullanarak şeffaf bağış uygulamalarının nasıl geliştirilebileceğini ve bu alanda nasıl kullanılabileceğini örnek bir uygulama üzerinden sunmayı amaçlamaktadır. Çalışma aynı anda birden fazla kuruluşa bağış yapılmasına olanak sağlayarak daha geniş bir bağışçı kitlesine hitap edebilmesi, bağışların izlenebilirliğini artırması ve güvenilirlik sağlaması bakımından önemli avantajlar sunmaktadır. Bağış uygulamalarının gelecekteki gelişimine yön verebilecek blokzincir teknolojisi ve akıllı sözleşmelerin bağış alanındaki potansiyelini ortaya koyarak toplumsal yardım ve dayanışma alanında önemli bir katkı sağlamayı hedeflemektedir.

Blokzincir ve özellikle akıllı sözleşme teknolojisinin halen gelişmekte olması ve veri güvenliği ile ilgili eksiklerin bulunması sebebiyle; blokzincirde veri güvenliği ile ilgili çalışmalara katkı sağlamak amaçlanmıştır. Bu açıdan akıllı sözleşmeler ve farklı blokzincir tabanlı akıllı sözleşme platformları kullanılarak blokzincir teknolojisinin yeniliklerini takip etmek, platformları karşılaştırmak ve farklılıklarını uygulamalı bir şekilde görmek amaçlanmıştır. Ayrıca akademik anlamda ülkemizde blokzincir alanında yapılan çalışmaların azlığı ve eksikliği nedeniyle bu alanda veri güvenliği ile ilgili bir çalışma yapma isteği ve motivasyonu oluşmuştur. Bu sayede hem bu alandaki açıkları

giderebilmek hem de bu alanda kayda değer bir çalışma yapmaya çalışarak gelecekteki araştırmacılar için bir başvuru kaynağı olunması hedeflenmişir. Bu bölümde tezin amacı, tezle ilgili hipotez, tez çalışmasının katkısı ve literatürdeki yerinden bahsedilmektedir.

1.1.Tezin Amacı

HFSecImg çalışması, güvenli veri depolama ve doğrulama yöntemlerini kullanarak bir görüntünün güvenli bir şekilde saklanmasını amaçlamaktadır. Günümüzde, özellikle finans, sağlık ve devlet yönetimi gibi hassas verilerin saklanması gereken alanlarda, veri güvenliği büyük önem taşımaktadır. Bu nedenle, Hyperledger Fabric tabanlı bir blokzincir kullanılarak, bir görüntünün şifrelenerek güvenli bir şekilde saklanması, veri güvenliğini sağlamak açısından oldukça önemlidir. Çalışmada kullanılan şifreleme anahtarı Shamir'in sır paylaşım yöntemiyle belirlenmektedir. Bu yöntem, bir verinin şifrelenmesinde kullanılan anahtarın parçalara bölünmesi ve bu parçaların farklı paydaşlarda saklanmasıdır. Bu sayede, anahtarı elde etmek için tüm parçaların bir araya getirilmesi gerekmektedir. Bu yöntem güvenli bir şekilde veri paylaşımını sağlayarak veri kaybını önlerve kötü amaçlı saldırılara karşı koruma sağlar. Ayrıca görüntünün özet bilgisi ve paydaşların dijital imzaları RSA asimetrik şifreleme yöntemi kullanılarak oluşturulmaktadır. Bu dijital imzalar paydaşların kimlik doğrulama sürecinde kullanılarak veri bütünlüğünü ve güvenliğini sağlamaktadır. Chaincode kullanılarak blokzincirinde saklanan imzalar, her bir paydaşın katılımı ile oluşan bir ağ vasıtasıyla korunmaktadır. Görüntü, şifrelenerek bulut sisteminde saklanmaktadır. Bu sayede güvenli bir şekilde depolanarak kötü niyetli saldırılardan korunmaktadır. Ayrıca paydaşların imzaları ve görüntüye erişim chaincode vasıtasıyla doğrulanmaktadır. Bu doğrulama işlemi, veri bütünlüğünü ve veri güvenliğini sağlamaktadır. Hyperledger Fabric tabanlı bir blokzincir ile güvenli veri saklama ve doğrulama yöntemleri kullanarak bir görüntüyü korumayı amaçlamaktadır. Kullanılan yöntemler veri güvenliği açısından oldukça önemli olan bütünlük, gizlilik ve doğrulama prensiplerini temel almaktadır. Bu çalışma hassas verilerin saklanması gereken alanlarda, güvenli veri depolama ve doğrulama için önemli bir temel sağlamaktadır.

Ethtrace şeffaf bağış çalışmasının amacı Ethereum blokzinciri ve akıllı sözleşme teknolojilerini kullanarak şeffaf ve güvenli bir bağış uygulaması geliştirmektir. Blokzincir teknolojisi bağış işlemlerinin tüm aşamalarının şeffaf ve izlenebilir olmasını sağlamaktadır. Böylece bağışçılar nereye ve ne kadar bağış yaptıklarını takip edebilmektedir. Akıllı sözleşme bağışçıların bağışlarına belirli koşullar ve kısıtlamalar

koyarak bağışlarının nasıl kullanılacağına dair karar verme yetkisi vermektedir. Bu sayede bağışçılar, bağışlarının istedikleri şekilde kullanıldığından emin olabilmektedir. Ayrıca bağışlar aynı anda birden fazla kuruluşa yapılabilir, böylece bağışçılar farklı kuruluşlara farklı miktarlarda bağış yapabilirler. Bu çalışma, bağış sürecinin şeffaf ve güvenli hale getirilmesini ve bağışçıların bağışlarının etkili bir şekilde kullanıldığından emin olmalarını sağlayarak toplumsal faydayı artırmayı hedeflemektedir.

1.2.Hipotez

HFSecImg çalışması için, şifreleme anahtarı Shamir'in sır paylaşım yöntemiyle belirlenen ve paydaşlara dağıtılan bir gizli anahtar mevcuttur. Bu anahtar kullanılarak kaotik sistem tabanlı şifreleme yöntemiyle şifrelenen bir görüntünün özet bilgisinin ve paydaşların RSA şifreleme yöntemiyle ile oluşturulan dijital imzalarının chaincode kullanılarak Hyperledger Fabric tabanlı blokzincirde saklanması, veri bütünlüğü ve doğruluğu açısından etkili olacaktır. Çalışmada blokzincir teknolojisi ve dağıtık sistemler kullanarak veri bütünlüğünü ve güvenliğini sağlamak amaçlanmaktadır. Şifreleme anahtarları Shamir'in sır paylaşım yöntemi ile belirlenmekte ve paydaşlar arasında paylaşılmaktadır. Bu sayede, herhangi bir paydaşın bilgiye erişimi tek başına doğrudan sağlanmamaktadır. Ayrıca, paydaşların RSA şifreleme yöntemiyle oluşturulan dijital imzaları blokzincirinde saklanmaktadır. Bu da veri bütünlüğü ve doğruluğunu sağlamakta ve verinin değiştirilmesini zorlaştırmaktadır. Hipotez, bu yöntemlerin kullanımının veri bütünlüğü ve doğruluğunu artıracağını öne sürmekedir. Bu şekilde veri güvenliği sağlanacak ve güvenilir bir veri depolama ve paylaşım platformu oluşturulacaktır.

Ethtrace uygulamasının hipotezi; Ethereum tabanlı akıllı sözleşme kullanılarak oluşturulan blokzincir şeması, şeffaf bağış uygulamaları için geleneksel yöntemlere kıyasla daha güvenli, şeffaf ve etkili bir çözüm olacağıdır. Geleneksel yöntemlerde veri güvenliği ve bütünlüğü sağlansa da şeffaflık tam anlamıyla hiçbir zaman sağlanamamaktadır. Birden fazla kuruluşa yapılan bağışların şeffaflığı, kaynakların takibi ve bağışların güvenliği açısından Ethereum blokzincir teknolojisi ile sağlanan akıllı sözleşme uygulamaları, mevcut merkezi sistemlere kıyasla daha verimli sonuçlar sunacaktır.

1.3.Tez Çalışmasının Katkısı

Bu çalışma blokzincir teknolojisi ve dağıtık sistemler konusunda literatüre katkı sağlamaktadır. Çalışmada görüntü verilerinin güvenliği için Shamir'in sır paylaşım

yöntemiyle belirlenen şifreleme anahtarı kullanılmıştır. Bu yöntem veri bütünlüğünü koruyarak veri gizliliğini artırmaktadır. Ayrıca RSA şifreleme yöntemiyle oluşturulan dijital imzaların kullanımı verilerin güvenilirliğini artırmaktadır.

Bu çalışmada, blokzincir teknolojisinden faydalanarak Hyperledger Fabric tabanlı bir blokzincir uygulaması geliştirilmiştir. Bu uygulama sayesinde paydaşlar arasında güvenli bir şekilde veri paylaşımı sağlanmaktadır. Blokzincir teknolojisi kullanarak, verilerin güvenliği ve doğruluğu artırılmaktadır. Ayrıca bu çalışma, kaotik sistem tabanlı şifreleme yöntemini kullanarak görüntülerin güvenli bir şekilde saklanmasını sağlamaktadır. Bu yöntem görüntü şifrelemede geleneksel şifreleme yöntemlerinden daha güvenli ve sağlam bir yapı sağlamaktadır. Bu çalışmanın literatüre katkısı, blokzincir teknolojisi ve dağıtık sistemler konusunda bir uygulama sunmasıdır. Ayrıca, Shamir'in sır paylaşım yöntemi ve kaotik sistem tabanlı şifrelemeyöntemlerinin kullanımı literatürdeki çalışmaları zenginleştirmektedir.

İkinci çalışmanın literatüre katkısı, şeffaf bağış uygulamalarının blokzincir teknolojisi ile nasıl uygulanabileceği ve Ethereum tabanlı akıllı sözleşmelerin bu alanda nasıl kullanılabileceği konusunda örnek bir uygulama sunmasıdır. Ayrıca, aynı anda birden fazla kuruluşa bağış yapılabilmesine olanak sağlaması nedeniyle daha geniş bir bağışçı kitlesine hitap edebilmesi ve bağışların izlenebilirliğini artırması gibi avantajları vardır. Bu çalışma, bağış uygulamaları ve blokzincir teknolojisi arasındaki olası bir birlikteliği ortaya koyması açısından da literatüre önemli bir katkı sağlamaktadır.

1.4.Tezin Organizasyonu

Bu tez çalışması 6 bölümden oluşmaktadır. Her bir bölümde yer alan ana başlıklardan bahsedildiğinde;

Bölüm 1, tezde gerçekleştirilen uygulamaların depolama, veri doğrulama, güvenlik, gizlilik, veri bütünlüğü ve izlenebilirlik açısından çalışmalarda kullanılan yöntemlerden bahsetmektedir. Bu yöntemlerin kullanımına bağlı olarak çalışmalara sağladığı katkılar, kullanım amacı ve literatüre katkısı ile ilgili bilgiler verilmiştir.

Bölüm 2’de, literatürde yer alan ve genellikle akıllı sözleşme tabanlı blokzincir çalışmalarının odaklandığı sorunlar ve geliştirdiği uygulamalar incelenmiştir. Benzer çalışmalar tespit edilerek her birinde kullanılan teknolojiler ayrıntılı bir şekilde raporlanmıştır. Gizlilik ve güvenlik tabanlı blokzincir temelli çalışmalar genel bir taramayla incelenerek bu çalışmalarda yer alan mekanizmalar raporlanmıştır. Ayrıca HLF

tabanlı çalışma ile kıyaslanarak tablo haline getirilmiştir. En çok kullanılan fikir birliği algoritmalarına ait incelemeler yapılarak bu konu hakkında detaylı bilgi sahibi olunmuştur. Akıllı sözleşmelerin kullanım amaçları, güvenlik katkıları, dayanıklılıkları, kullanıldığı sektörler, gizlilik ve mahremiyet konuları açısından incelemeler yapılmıştır.

Bölüm 3'de blokzincirin temel yapısından, eşler arası iletimden, bir blokta bulunması gereken verilerden, fikir birliği algoritmalarından, kriptografik temellerden ve işlem süreleri gibi bilgilerden detaylıca bahsedilmektedir. Genel, özel ve konsorsiyum blokzincir sistemleri hakkında bilgi verilmiştir. Ayrıca birbirinden farklı protokollere sahip blokzincir platformlarının yapısında bulunan fikir birliği algoritmaları ve gizlilik, bütünlük, erişilebilirlik açısından birbirlerine göre durumları analiz edilmiştir. Fikir birliği algoritmaları en önemli parametreleri ele alınarak kıyaslamalı analiz yapılmıştır.

Bölüm 4'de anahtar dağıtımı, görüntü şifreleme, dijital imza oluşturma, akıllı sözleşme ile verileri blokzincire kaydetme ve sorgulama aşamalarından bahsedilmiştir. Ayrıca görüntü şifreleme performans analizleri gerçekleştirilmiştir. Hypeledger Fabric ağının güvenlik ve gizlilik açısından etkilerinden bahsedilmiştir.

Bölüm 5'de Ethereum ağında kullanılmak üzere geliştirilen şeffaf bağış çalışmasının şeması, akıllı sözleşmenin mimarisi ve akıllı sözleşmenin içerdiği parametrelerden bahsedilmiştir. Ethereum ağının şeffaf bağış uygulamasında tercih edilme nedenleri ve dezavantajları ele alınmıştır.

Bölüm 6'da HFSecImg (Hyperledger Fabric tabanlı) ve Ethtrace (Ethereum tabanlı) ile ilgili genel sonuçlara yer verilmiştir. Ayrıca bu iki çalışmanın birbirine göre avantajları, dezavantajları ve farkları detaylıca incelenmiştir.

2. LİTERATÜR TARAMASI

Bu bölümde, öncelikle Hyperledger Fabric ile gerçekleştirilen genellikle akıllı sözleşme tabanlı uygulamalar kapsamında olan çalışmalara yer verilmiştir. Takip eden alt başlıklarda genel literatür taraması yapılarak gerekli görülen yerlerde tablolar yardımıyla ilgili çalışmaların konuları, kullanılan blokzincir teknolojisi, kriptografik altyapısı vb. bilgiler verilerek kıyaslamalar yapılmıştır.

2.1.Veri Saklama ve Doğrulama

Elektronik sağlık kayıtlarının paylaşımı ve güvenliğinin sağlanması önemli çalışma alanlarından biridir. Tanwar ve arkadaşları, Hyperledger Fabric temelli elektronik sağlık kaydı paylaşımı için bir mimari önermiştir. Hastaların sisteme eklenmesi, tedavi uzmanı ve laboratuvara ait sorgular için chaincode (akıllı kontrakt) çağrılarak tüm işlemler onay mekanizmasından geçtikten sonra blokzincirdeki deftere dahil edilmektedir. Her bir organizasyona ait eşlerin belirli sayıdaki transaction için işlem süreleri, işlem gecikmeleri, sorguların gerçekleşme süreleri, kaynak tüketimi (CPU), ağ trafiği istatistikleri vb. performans analizleri ve değerlendirmelerini yapmaktadırlar. Blok boyutu, transaction modu, ağ büyüklüğü ve eşlere göre işlemler değişiklik göstermesine rağmen başarılı sonuçlar elde edildiği görülmektedir (Tanwar vd., 2020). Mimari yapısı (Tanwar vd., 2020)'ye benzer olmakla birlikte Singh ve arkadaşları (A. P. Singh vd., 2020), Hyperledger Fabric tabanlı sağlık yönetim sistemi uygulaması geliştirmiştir. Iqbal ve arkadaşları (Iqbal vd., 2021), veteriner kliniğinde kullanılmak üzere Hyperledger Fabric tabanlı blokzincire dayalı bir sistem geliştirmiştir. Geliştirilen bilgi yönetim sisteminde bulunan akıllı sözleşme sayesinde elde edilen veriler ile tahmine dayalı veri analizi yapmaktadırlar. Belirtilen sağlık uygulamalarının hepsinde amaç güvenlik, güvenilirlik ve veri doğrulama işlemlerinin garanti altına alınmasını sağlamaktır. Video kayıtlarının güvenliğini sağlamak amacıyla akıllı kontraktlar kullanılabilir. Khan ve arkadaşları (Khan vd., 2020), kaydedilen bir videoda değişiklik olup olmadığını doğrulamak ve kayıtların güvenilirliğini maksimum düzeye çıkarabilmek için blokzincirini kullanarak güvenli bir sistem oluşturmuşlardır. Sensör katmanında bulunan kamera ile alınan video kaydı, görev ve kamera bilgileri veri katmanında bulunan Hyperledger Fabric dağıtık defter platformunda doğrulanarak uygulama katmanında bulunan son kullanıcıya sunulur. Bu sayede değişmezlik sağlandığından güvenilirlik son derece yüksek olmaktadır. Chi ve arkadaşları, endüstriyel nesnelerin internetinde (IIoT) Hyperledger Fabric ile güvenliğin sağlandığı veri paylaşımı şeması önermiştir. Süreç kimlik doğrulama, imza ve doğrulama ve veri paylaşım fazlarından oluşmaktadır. Veri paylaşım

çalışma çatısı incelendiğinde istemcilerin kontrolü, imza doğrulaması, topluluk doğrulama işlemleri blokzincir ağı ile gerçekleştirildiği görülmektedir. Akıllı kontraktları çağırarak topluluk doğrulama bilgileri elde edilir ve spesifik kayıtları elde etmek için belirlenen ilgili kayda ait etiketler girilir. Bilgiler merkez düğüme iletilir ve topluluk algılama algoritması ile benzerlik hesaplanarak geriye istenilen sorguya ait sonuçlar döndürülür (Chi vd., 2020). Akıllı kontrakt hazırlarken dikkat edilmesi gereken bazı hususlar vardır. Global değişken tanımlama, rasgele sayı üretimi, sistemin zaman damgası, obje adreslerinin çevreye bağlı olması, hız problemlerini önlemek için programın eşzamanlılığı gibi faktörler deterministik olmayan programlama dili kaynaklı belirsizliklerdir. Web servisi ve sistem komutlarının eşler için farklı olması, harici dosya erişimi ve harici kütüphane çağırma riskleri, veritabanı sorgu riskleri, her işlemin her eşte aynı sonucu vermemesi durumu, giriş/çıkış argümanları kontrolü vb. durumlar blokzincirin yapısında problem oluşturabilecek temel sebeplerdir (Yamashita vd., 2019). Hyperledger Fabric geliştirme çatısında chaincode hazırlarken ve ağ yapısı oluşturulurken yukarıda belirtilen durumlara dikkat edilmelidir. Raman ve Varshney, blokzincirde veri bütünlüğünü özel anahtar şifreleme ile sağlar ve veri dağıtımını yapmak için Shamir'in sır paylaşım şemasını kullanır. Dağıtık depolama yapmak amacıyla özel anahtarın (K) ve önceki bloğun karma değerinin sır paylaşımı SSS kullanılarak sağlanır. Bu sayede özel anahtara ait herhangi bir parça (K_i) ile özel anahtara (K) ait herhangi bir bilgi elde edilemez. Anahtarın ve karma değerin parçaları eşlerde saklanır. Eş sayısı ve m küme boyutuna bağlı olarak veri kurtarma maliyeti tahmini yapılır (Raman & Varshney, 2018). Özel anahtarın tekrar elde edilmesi aşamasında herhangi bir kayıp olduğunda bozulan verinin onarılması neredeyse imkansızdır ve transaction verisi tamamen kaybedilir. Gezegenler arası dosyalama sistemi (IPFS), merkezi olmayan bir sistemde veri depolama ve paylaşım yapmayı sağlayan belirli protokollere dayalı eşler arası ağdır. Naz ve arkadaşları, IPFS ile güvenli dosya aktarımını sağlamak için Ethereum (akıllı kontrakt) blokzinciri tabanlı veri paylaşım platformunu önermiştir. IPFS sunucusuna yüklenen veri SSS yöntemi ile n adet parçaya bölünür. RSA imza şeması ile kullanıcı kimlik doğrulaması yapılır. Kullanıcının talep ettiği verilere ulaşabilmesi için depolanan verinin büyüklüğüne bağlı dijital içerik fiyatı belirlenmektedir (Naz vd., 2019). Feixiang ve arkadaşları, Henon-Zigzag haritasını ve kaotik kısıtlı Boltzman makinesini kullanarak asimetrik renkli görüntü şifreleme yapmışlardır (Feixiang vd., 2021). Henon haritası kullanılarak üretilen iki yeni rastgele sayı dizisini Zigzag haritası modüle eder. Permütasyon işleminin güvenliğini sağlamak amacıyla iki harita karıştırılmıştır. Ardından, 10 sunucuya kurulan ve her birinin bir düğüm olduğu bir ağ oluşturulmuştur. PoW (Proof of Work) konsensüsü kullanılarak yeni blok üretilir ve

güncellenir. Bloklar kayıt defterine eklendikten sonra P2P protokolü kullanılarak doğrulama ve senkronizasyon işlemleri gerçekleştirilir. Şifreli görüntü, göndericinin açık anahtarı, alıcının açık anahtarı ve şifrelemede kullanılan gizli anahtarın etiketini kullanarak bir dijital imza üretilir. Üretilen imza ve transaction şifreleyici tarafından blokzincir ağına gönderilir. Blokzincir ağında bulunan düğümlerde doğrulama işlemleri gerçekleşir. Ardından şifreli görüntü alıcıya gönderilir ve deşifreleme gerçekleştirilir. Yapılan iş bizim çalışmamızla benzerlik göstermesine rağmen asimetrik görüntü şifreleme, konsensüs protokolü, transaction bloğunda belirgin farklar vardır. Verilerin gizliliğini ve işlenebilirliğini sağlamak amacıyla açık anahtar kimlik doğrulamalı aranabilir şifreleme (PAEKS) mekanizması kullanılabilir. Bitcoin vb. geleneksel blokzinciri sistemlerinde kimlik gizliliği tam anlamıyla sağlanamamaktadır. Bu durum bazı sistemlerde dezavantaj olarak algılanabilir. Zhang ve arkadaşları, kimlik doğrulamalı aranabilir bir şifreleme olan blokzincire dayalı şemayı sunmaktadır. DASES ile görüntü verilerinin izlenebilirliğini sağlamayı ve kaybedilmesini engellemeyi amaçlamışlardır. Şifreli bilgiler bulut sunucusunda ve bu bilgilerle oluşturulan imzalar blokzincirde tutulmaktadır (Zhang vd., 2020). Byun ve arkadaşları, Endüstriyel Nesnelerin İnterneti'nde (IIoT) izinli özel blokzinciri kullanarak görüntü şifrelemede güvenliği sağlamayı amaçlamıştır. Şifreli görüntünün gizliliğini ve güvenliğini sağlamak amacıyla şifrelenen piksel değerleri blokzincirde tutulmaktadır. Geliştirdikleri uygulamada bir cihazın sensörleri aracılığıyla alınan görüntü işlenmesi için zincire gönderilir. Görüntünün özet değeri akıllı kontrakt kullanılarak blokzincire eklenir ve tüm düğümlere doğrulama işlemi için gönderilir. Bloğun açık anahtarı kullanılarak orijinal görüntü elde edilir (Khan & Byun, 2020). Kaotik görüntü şifreleme şemalarında parmak iziyle ilgili yöntemler de kullanılmaktadır. Dağıtıcıların parmak izlerinden etkilenen anahtar akışları oluşturulmaktadır. Li, parmak izi yöntemiyle blokzincir çerçevesi kullanılarak şifrelenmiş görüntünün distribütörden gönderilmesi sağlanmıştır. Güvenlik aşamasında parmak izi ve doğrulama kısmında blokzincir çerçevesi kullanılmaktadır. Açık kanal aracılığıyla şifreli görüntüler iletilir; doğrulama, şifreleme işlemleri kullanıcı katmanında bulunan farklı dağıtıcılar tarafından gerçekleştirilmektedir. Kullanıcılar şifreleme, şifre çözme, imzalama, doğrulama, izleme, parmak izi çakışması gibi işlemleri yapmaktadır (Li, 2021). Önemli uygulama alanlarından biri olan görüntü doğrulamada blokzincir kullanılabilmektedir. Abrar ve arkadaşlarının geliştirdiği uygulamada Advanced Encryption Standard ile şifrelenen filigran, orta frekans bölgelerinde üçüncü seviye ayrık dalgacık dönüşümüne gömülür. Doğrulama yapmak amacıyla, filigran gömülen gri seviye görüntü Secure Hash Algoritması 256 (SHA-256) kullanılarak Ethereum tabanlı blokzincirine kaydedilir. Bu sayede kimlik doğrulamanın

blokzincir vasıtasıyla gerçekleştirilmesi sağlanmaktadır (Abrar vd., 2021). Tıbbi Nesnelerin İnterneti (IoMT) alanında ve sağlık uygulamalarında merkezi yönetimi ortadan kaldırmak ve güvenli veri iletimini sağlamak amacıyla blokzincir mimarisi kullanılmıştır. Alqaralleh ve arkadaşları, önerdikleri modelde, güvenli görüntü aktarımını sağlamak için blokzincirini kullanmaktadır. Uygulama blokzincir doğrulamada kullanılmak amacıyla şifrelenmiş görüntülerin dijital olarak imzalanmış karma değerlerini tutar (Alqaralleh vd., 2021). Hassas olan tıbbi bir görüntüye ait özellikler blokzincirde saklanarak gizlilik ve mahremiyet endişeleri ortadan kaldırılabilir. Shen ve arkadaşları tıbbi bir görüntünün şifreleme özelliklerini ve şifreli görüntüye ait hash değerini blokzincire kaydeder ve zamanı geldiğinde doğrulama için yerel bir veritabanına indirir. Blokzincirdeki veri bütünlüğü sağlama görevini madenci üstlenir. Görüntünün hash değeri ve dijital imzalar üretildikten sonra bilgiler blokzincire madenci tarafından yüklenir (Shen vd., 2019). Li ve arkadaşları, bulut sunucusu ve blokzincir kullanarak şifrelenen görüntülerin güvenilirliğini ve arama sürecinin şeffaf olmasını sağlamaya çalışmıştır. Şifreli dizinler Ethereum tabanlı blokzincirde depolanmaktadır ve doğrulama sorgular sayesinde sağlanmaktadır. Diğer çalışmalara benzer şekilde şifreli indeksler blokzincirde depolanmaktadır. Akıllı kontraktlar kullanılarak transactionlar yapılır. Bu sayede transactionlar izlenebilir ve değiştirilemez. Akıllı kontraktlar güvenlik indekslerini saklama, çıktı arama ve güncelleme vb. işlevleri yapmak amacıyla fonksiyonlar bulundurmaktadır. Şifreli görüntüleri depolamak için bulut sunucusu kullanılmaktadır. Deşifreleme ve doğrulamanın yapılabilmesi için blokzincirden şifreli görüntüye ait bilgiler elde edilir ve bulut sunucusundan şifreli görüntü indirilir. Simetrik şifreleme ile şifrelenen görüntü özel anahtar ile deşifre edilir. Yazarlar nihai hedefleri olan şifreli görüntü arama sürecinde ortaya çıkan güven sorunlarını çözmek amacıyla blokzincirini kullanmışlardır (Li vd., 2021). Feixiang ve arkadaşları, Henon-zikzak haritasını ve kaotik kısıtlı Boltzmann makinesini (CRBM) kullanan bir renkli görüntü şifreleme algoritması önermiştir. Algoritma performans ve güvenlik bakımından görüntü şifrelemede güvenliği sağlamaktadır. Çalışmalarında, blokzincirinin ve önerilen algoritmanın birleşik kullanımına dayalı olarak, yeni bir görüntü şifreleme/şifre çözme sistemi önerilmiştir. Sistemin iki özelliği vardır: görüntülerin asimetrik olarak şifrelenmesi/şifresinin çözülmesi ve şifrelenmiş görüntülerin bütünlüğünün yetkili doğrulaması işlemlerinin yapılmasıdır (Feixiang vd., 2021). Khan ve Byun, IIoT bağlamında görüntüyü şifrelerken güvenliğini sağlamak için izin verilen özel blokzinciri tabanlı bir çözüm önermişlerdir. Bu şemada, bir görüntünün kriptografik piksel değerleri (hash), görüntü verilerinin gizliliğini ve güvenliğini sağlayarak blokzincirinde depolanır. Piksel değişim hızı (NPCR), birleşik ortalama değiştirilmiş yoğunluk (UACI) ve

bilgi entropi analizine dayanarak, farklı saldırılara göre önerilen görüntü şifreleme algoritmasının başarımı değerlendirilmiştir (Khan & Byun, 2020). Li, parmak iziyle ilgili yeni bir kaotik görüntü şifreleme şeması önermiştir. Anahtar akışlarının oluşturulması, görüntülerin düz metin ilişkili ziyade dağıtıcıların parmak izlerinden etkilendiğini savunmaktadır. Ayrıca, şifrelenmiş görüntünün distribütörden doğru bir şekilde gönderilmesini sağlamak için blokzincir çerçevesi kullanılmıştır (Li, 2021). Blokzincir tabanlı gerçekleştirilen çalışmaların tanımı ve içeriği, kullanılan geliştirme çerçevesi, ağ yapısı, bulut depolama kullanılması durumu, sır paylaşımı şeması içermesi durumu, kripto mekanizmaya sahip olması, dijital imza kullanılması, erişim kontrolünün olması, kimlik doğrulama yapılması ve gizlilik parametreleri gibi bilgiler Tablo 2.1'de verilmiştir.

Tablo 2.1. Blokzincir tabanlı yaklaşımlarda kullanılan güvenlik mekanizmaları

Referans	Tanım	Blokzincir	Geliştirme Çatısı	Akıllı sözleşme	Blokzincir ağı	Bulut depolama	Secret sharing	Kripto mekanizma	Dijital imza	Erişim kontrolü	Kimlik doğrulama	Gizlilik
(Tanwar vd., 2020)	Elektronik sağlık kaydı	✓	Hyperledger Fabric	✓	İzinli	Hayır	Hayır	Hayır	Hayır	✓	✓	✓
(A. P. Singh vd., 2020)	Elektronik sağlık kaydı	✓	Hyperledger Fabric	✓	İzinli	✓	Hayır	Hayır	Hayır	✓	✓	✓
(Iqbal vd., 2021)	Veteriner kliniği bilgi yönetimi	✓	Hyperledger Fabric	✓	İzinli	Hayır	Hayır	Hayır	Hayır	✓	✓	✓
(Khan vd., 2020)	Güvenlik kameraları için veri doğrulama sistemi	✓	Hyperledger Fabric	✓	İzinli	Hayır	Hayır	✓	✓	✓	✓	✓
(Chi vd., 2020)	Veri paylaşım şeması	✓	Hyperledger Fabric	✓	İzinli	Hayır	Hayır	✓	✓	✓	✓	✓
(Raman & Varshney, 2018)	Blokzincirde gizli paylaşım	✓	—	Hayır	—	✓	✓	✓	Hayır	✓	Hayır	✓
(Naz vd., 2019)	Blokzincir kullanan veri paylaşım platformu	✓	Ethereum	✓	İzinli	✓	✓	✓	✓	✓	✓	✓
(Feixiang vd., 2021)	Blokzincir ile görüntü şifreleme ve doğrulama	✓	—	Hayır	—	Hayır	Hayır	✓	✓	Hayır	Hayır	Hayır
(Zhang vd., 2020)	Blokzincir ile aranabilir şifreleme şeması	✓	—	Hayır	—	✓	Hayır	✓	✓	Hayır	Hayır	✓
(Khan & Byun, 2020)	Blokzincir tabanlı görüntü şifreleme şeması	✓	Hyperledger Fabric	✓	İzinli	Hayır	Hayır	✓	✓	✓	✓	✓

Tablo 2.1 Tablonun Devamı

(Li, 2021)	Blokzincire dayalı kaotik görüntü şifreleme şeması	✓	—	Hayır	—	Hayır	Hayır	✓	✓	Hayır	Hayır	✓
(Abrar vd., 2021)	Blokzincir kullanarak görüntü kimlik doğrulaması	✓	Ethereum	Hayır	İzinsiz	Hayır	Hayır	✓	Hayır	✓	✓	✓
(Alqaralleh vd., 2021)	Blokzincir destekli görüntü aktarımı	✓	—	Hayır	—	✓	Hayır	✓	Hayır	Hayır	Hayır	✓
(Shen vd., 2019)	Blokzincir ile gizliliği koruyan görüntü geri alma	✓	Ethereum	✓	İzinsiz	✓	Hayır	✓	✓	✓	✓	✓
(Li vd., 2021)	Blokzincir tabanlı şifreli görüntü geri alma şeması	✓	Ethereum	✓	İzinsiz	✓	Hayır	✓	Hayır	✓	✓	✓
Önerilen Hyperledger Fabric şeması	Şifrelenmiş görüntünün depolanması ve doğrulanması	✓	Hyperledger Fabric	✓	İzinli	✓	✓	✓	✓	✓	✓	✓

2.2.Gizlilik ve Güvenlik

Blokzincir alanında gerçekleştirilen güvenlik ve gizlilik tabanlı uygulamaların literatür araştırması yapılmıştır. Bu alanda yapılan çalışmalar:

Zhang ve arkadaşları, karar vermeyi merkezi olmayan ve güvenli bir şekilde kolaylaştırmak için, eşlerine mevcut herhangi bir güvenilir ya da üçüncü tarafa ihtiyaç duymadan mevcut blokzincir ağı üzerinden oy vermeleri için yerel bir blokzinciri oylama protokolü önermişlerdir. Protokol uçtan uca gizliliği korur ve aldatmaya karşı tespit edilebilirlik ve düzeltilebilirlik gibi özelliklere sahiptir. Ayrıca, protokol geçerliliğini ve pratik uygulanabilirliğini gösteren, HyperLedger Fabric üzerinde referans uygulamasını da yapmışlardır (Zhang vd., 2018). Eyal ve arkadaşları, ölçeklendirmek için tasarlanmış yeni bir blokzincir protokolü olan Bitcoin-NG'yi (Yeni Nesil) sunmaktadır. Eyal ve arkadaşlarına göre, Bitcoin-NG aşırı yayılmaya dayanıklı ve Bitcoin'le aynı güven modelini paylaşan Bizans hatalarına dayanıklı bir blokzinciri protokolüdür. Bitcoin-NG, Bitcoin'e benzeyen işlemleri seri hale getiren, ancak diğer özelliklerden ödün vermeden gecikme ve bant genişliğinde daha başarılı olan bir blokzincir protokolüdür (Eyal vd., 2016). Önemli çalışmalardan biri de giderek artan oranda bulut tabanlı bir ortamda güvenlik ve gizlilik sorunlarını ele almak için geleneksel kriptografik temellerin ve erişim kontrol modellerinin kullanılmasının sınırlamaları vardır. Esposito ve arkadaşları, bulutta barındırılan sağlık verilerini korumak için blokzincir teknolojisini kullanma potansiyelini araştırmışlardır. Araştırma sonucuna göre; blokzincir başlangıçta nispeten küçük ve doğrusal olan işlem verilerini kaydetmek için tasarlanmıştır. Başka bir deyişle, yalnızca mevcut işlemin orijinal “anlaşma” ile geriye doğru izlenip izlenemeyeceği ile ilgilenir. Bununla birlikte, görüntüleme ve tedavi planları gibi sağlık verileri, arama gerektiren büyük ve ilişkisel veriler olabilir. Blokzincir depolamanın her ikisi ile ne kadar iyi başa çıkabileceği ile ilgili gereksinimler şu anda belirsizdir. Blokzincir’de verilerin silinemez ve değiştirilemez olması da ayrıca bir sorun teşkil eder. Bu zorlukların üstesinden gelmek için, çoğu, verilerin geleneksel veya dağıtılmış bir veritabanında blokzincirinin dışında tutulduğu, verilerin zincir dışı depolanması kavramını önermektedir, ancak verilerin özetleri (hash) blokzincirinde saklanmaktadır. Aynı zamanda, sağlık hizmeti verilerinin değişmeyen özetleri, zincir dışı tıbbi kayıtların doğruluğunu kontrol etmek için zincir halinde depolanır (Esposito vd., 2018). Henry ve arkadaşları, Zcash ve Monero gibi ortaya çıkmakta olan gizlilik merkezli kripto para birimleri, sıfır bilgi ispatı olan özlü bilgi içermeyen bilgi etkileşimi argümanı (zk-SNARK), izlenebilir halka imzaları (ring signature), gizli işlemler, gizli adresler gibi Bitcoin işlemlerinden daha iyi gizlilik özellikleri

sunmak için şifreleme ilkeleri kullanıyor. Bunun dışında CoinJoin'de gönderici ve alıcı arasındaki işlemleri gizlemek amacıyla oluşturulmuş merkezi bir karıştırma servisi bulunmaktadır. Bu servis, kullanıcıların işlemlerini deftere kaydetmeden önce karıştırılmasını sağlar. Karıştırma servisi alıcı ve gönderici arasındaki ilişkiyi bildiğinden tam anlamıyla gizlilikten bahsedilememektedir (Henry vd., 2018). Kopp ve arkadaşları, depolama sağlayıcılarının katkıda bulunacakları finansal teşviklere sahip olan, gizliliği koruyan dağıtılmış bir depolama sistemi tasarlamışlardır. Blokzincir mimarisini, fonları ve depolama sözleşmelerini idare etmek için kullanmaktadırlar. Depolama alanı sağlayıcıları, geri alınabilirlik ispatlarını, yani gerçekten de depolamayı sağladıklarını gösteren şifreleme ispatlarını yayınlayarak, depolama sözleşmelerine uyumu kanıtlayabilmektedir. Bağlanabilir halka imzaları (linkable ring signature) sayesinde doğrulanabilirlik ve gizlilik sağlanmaktadır. Bundan dolayı, bu tasarım başarılı bir güvenlik ve gizlilik sağlamaktadır (Kopp vd., 2017). Halpin ve arkadaşları, mevcut blokzincir teknolojilerinde gizliliği korumak için iki yaklaşım olduğunu söylemektedir. Birincisi, Gizli Transferler (Confidential Transfers) gibi tekniklerle mevcut blokzincirlerinde anonimizasyon sağlamaktır. Diğer yöntem, Bitcoin ile uyumlu olmayan Zerocash gibi yeni blokzincirleri oluşturmaktır (Halpin & Piekarska, 2017). Zerocash, sıfır bilgi ispatını "bilgi edinmenin etkileşimli olmayan bağımsız argümanı" kullanmaktadır (SNARKS – Succint Non-interactive Argument of Knowledge) (Sasson vd., 2014). Rahulamathavan ve arkadaşları, blokzincir mimarisinde işlem verilerini korumak amacıyla nitelik tabanlı şifreleme (attribute-based encryption) tekniği kullanmışlardır. Nitelik tabanlı şifrelemenin basitliği ve hassas yapısı sayesinde işlem verileri rahatlıkla kontrol edilebilmektedir. Önerilen modelde güvenlik ve gizlilik analizi yapılmaktadır ve saldırıları azaltmak amacıyla stratejiler geliştirilmektedir. Sayısal analiz bölümüne bakıldığında, blokzincir ile çalışan IoT'nin gizlilik sağlama açısından minimum hesaplama yükü ile nitelik tabanlı şifrelemeden yararlanabileceğini göstermeye çalışmışlardır (Rahulamathavan vd., 2017). Li ve arkadaşları, kimlik doğrulama ve güvenlik önlemlerinde geleneksel IoT'nin eksikliklerine değinmişlerdir. Buna bağlı olarak blokzincir tabanlı bir model önermişlerdir. Önerilen sistemi doğrulamak için Hyperledger Fabric'i temel alan genel özellik ve sadeliğe sahip prototip oluşturmaktadırlar. Düşük uygulama maliyeti, IoT gibi hafif cihazlara dağıtım için uygun olmasını sağlar. Ayrıca, çoklu zincirin yapısı özelliğiyle farklı güven alanları arasında ek güvenlik sağladığından bahsetmişlerdir (Li vd., 2018). Biswas ve arkadaşları, ölçeklenebilirlik sorunları ele alınmadıkça IoT ve blokzincirin entegre olamayacağını belirtmektedir. IoT'deki büyük ölçekli ticari işlemlerin daha iyi ölçeklenebilmesine olanak sağlamayı amaçlar ve blokları depolamak için bellek gereksinimi sorununu ele alır. Bunun

için ölçeklenebilir bir yerel defter uygulayarak küresel blokzincire giren işlemlerin sayısını kısıtlamaktadırlar (Biswas vd., 2018). Sarıtekin ve arkadaşları, blokzincir teknolojisine dayanan bir iletişim uygulaması (Cryptouch) önermişlerdir. Blokzincirinde büyük miktarda veri depolanamadığından dolayı Gezegenlerarası Dosya Sistemi (IPFS) kullanılmıştır. IPFS sistemi verileri saklamak için halka açık bir veritabanı olarak tasarlandığından çalışmada blokzincir ile birlikte kullanılmıştır (Sarıtekin vd., 2018). Dai ve arkadaşları, blokzincirin kandırılamazlık (tamper-proofing), sistem kurtarma (disaster recovery) ve gizlilik koruması (privacy protection) gibi güvenlik avantajlarından bahsetmişlerdir. Ayrıca blokzincirin siber güvenlik alanında, merkezi olmayan dağıtılmış güvenli alan adı hizmeti, anahtarsız imza altyapısı, güvenli veri depolama gibi mevcut araştırma ve uygulamalarından bahsetmişlerdir. Blokzincirin teknik altyapısının sınırları, kriptografi uygulamalarındaki potansiyel riskler, açık kaynak kodlu blokzincir platformlarının yoğun saldırıya maruz kalması, güvenlik yönetimi gibi güvenlik meselelerine değinilmiştir (Dai vd., 2017). Kosba ve arkadaşları, merkezi olmayan bir akıllı sözleşme sistemi Hawk'ı sunmuşlardır. Hawk, finansal işlemleri blokzincirde açık bir şekilde saklamamasından dolayı işlem gizliliğini koruyan bir yapıya sahiptir. Bu sözleşmelerde sıfır bilgi ispatı (zero knowledge proof) gibi kriptografik ilkeller kullanılarak etkili bir şifreleme protokolü oluşturulmuştur. Çalışmalarında Hawk ile bir programın (akıllı sözleşmenin) oluşturulması ve derlenmesi için gerekli bilgiler verilmektedir. Gizlilik için sıfır bilgi şifrelemenin yeni bir biçimi olan zk-SNARK (Zero-Knowledge Succinct Non-Interactive Argument of Knowledge) protokolü kullanılmıştır (Kosba vd., 2016). Bitcoin, çoklu imza (multi-signature) adı verilen cüzdanlarının güvenliğini arttırma amacıyla, bölme kontrol tekniğini kullanarak güvenliği sıkılaştırmaya yönelik yerleşik bir işleve sahiptir. Çoklu imza işlemlerini kullanmak kullanıcının gizliliğini ve bilinemezliğini önemli ölçüde tehlikeye atmaktadır. Gennaro ve arkadaşları, verimli ve optimal eşik Dijital İmza Algoritması'nı (DSA) özel anahtarları güvenceye almak için önermişlerdir (Gennaro vd., 2016). Önerilen eşik (çoklu) imzalarının kullanımının arkasındaki temel neden, özel anahtarın paylara (parçalara) bölündüğü gizli paylaşımdan (secret sharing) kaynaklanmaktadır (Shamir, 1979). Cüzdanda belirlenen bir eşiğe eşit veya bu eşik değerinden daha büyük olan hisse senetlerinin herhangi bir alt grubu, özel anahtarı yeniden yapılandırabilir, ancak daha küçük olan herhangi bir alt set, anahtar hakkında bilgi sahibi olmamaktadır. Eşik imzalarında (Goldfeder vd., 2014), katılımcılar anahtarın asla açığa çıkmaması amacıyla, doğrudan bir imza oluştururlar. Conti ve arkadaşları, Bitcoin blokzincirinin yapısını ayrıntılı bir şekilde ortaya koymuşlardır. Güvenlik ve gizlilik anlamında e-ticaret endüstrisinde oluşabilecek sorunları tartışmışlardır. Ayrıca Bitcoin'in temel tasarım ilkeleri bozulmadan mahremiyetin

güçlendirilmesi ve gizliliğin geliştirilmesini amaçlayan çalışmaların olduğundan bahsetmiştir (Conti vd., 2018). Zerocash'in (Sasson vd., 2014) bu yapıyı koruyarak çalışan gizlilik tabanlı bir blokzincir yapısına sahip olduğundan bahsetmiştir. Gizlilik merkezli bir kripto para birimi olan Monero (Nisan 2014), Cryptonote protokolünü temel almaktadır. En büyük kripto para birimi olan Bitcoin'in işlem grafiğinde hangi paranın harcandığı açıkça görülebilmektedir (Miller vd., 2017). Fakat Cryptonote, işlem grafiğini gizlemek amacıyla karma (mixins) olarak adlandırılan belirsiz işlem girişlerini dahil etmektedir. Bitcoin, sistemin dağıtık yapısının esnek olmamasından dolayı tüm ağ kullanıcılarının istemcilerini güncelleyene kadar yeni özelliklerin uygulanmasını engeller. Cryptonote, bu eksiklikleri göz önüne alarak geliştirilen bir elektronik paradır (Van Saberhagen, 2013). Zheng ve arkadaşları (X. Zheng vd., 2018), sağlık verilerinin güvenli ve şeffaf bir şekilde paylaşılması için bulut depolaması ile blokzincir teknolojisini birlikte kullanmıştır. Temel hedefleri, kullanıcıların verilerini güvenli bir şekilde kontrol etmelerini, paylaşmalarını sağlayacak ve Genel Veri Koruma Yönetmeliği'ne (General Data Protection Regulation) uyumlu bir yapı oluşturmaktır. Bu yapının genel mimarisi ve akış diyagramı incelendiğinde kullanıcı, müşteri ve bulut platformları için gerekli sistem tasarımı hakkında bilgi sahibi olunabilmektedir. Ayrıca geliştiricilerin kendi para birimlerini oluşturabilmeleri, bir varlık veya sanal bir paylaşımın yapılabilmesini mümkün kılması sebebiyle uygulama platformu olarak Ethereum'u (Wood, 2014) seçmişlerdir. Monero kripto parasında bulunan Halka Gizli İşlem (RingCT) protokolünün altyapısında halka imzası (ring signature) ve taahhüt planları bulunmaktadır. RingCT, orijinal CryptoNote protokolüne kıyasla halka imzasının boyutunu %50 oranında kısaltsa da hala halkada bulunan ortak anahtarların sayısıyla doğrusal orandadır. Fakat bu protokoldeki bağlanabilir halka imzasının (linkable ring signature) büyüklüğünden dolayı işlemlerin büyüklüğü artmıştır. Halka imzası şeması, grup adına bir mesajı imzalamayı ve aynı zamanda imzalayanların kimliklerinin gizli kalmasını sağlamaktadır. Ayrıca aynı grup içerisinde herhangi bir üyenin iki defa imzaladığını kontrol etmek mümkün değildir. Herhangi bir grup yöneticisi yoktur. Halka imzası iyi derecede anonimlik sağlar. Bundan dolayı bazı durumlarda güçlü senaryolar ortaya çıkabilir. İsimsiz bir e-oylama sisteminde, bir kişinin birden fazla oy verip vermediğini tespit etmek gerektiğinde sorun olabilir. Bağlanabilir (linkable) halka imzası bağlantılı gizliliğin mükemmel bir örneğidir. Burada doğrulayıcı gruptaki kullanıcılardan biri dışında, imzalayan hakkında bilgiye sahip değildir. Yine de iki imza verildiğinde, doğrulayıcı gerçek imzalayanın kim olduğunu bilmese bile aynı imzalayan tarafından üretilip üretilmediğini bilir. Aynı kişinin imzaladığını bilmek e-oylama gibi sistemlerde ikinci kez aynı kişinin oylama yapıp yapmadığını kontrol etmek için daha

önemlidir (Sun vd., 2017). Monero blokzincir yapısında gizliliği sağlamak amacıyla halka imza (ring signature) kullanılmaktadır. Nitelik tabanlı şifrelemede özel anahtarlar, sistemdeki her kullanıcı için yayınladığı tüm özellikleri veya bilgileri doğrulayacak bir konumda olması gereken bir merkezi otorite tarafından verilmektedir. Bu sistemler, bir alan veya kuruluş içinde yayınlanan nitelikler üzerine bir politikaya göre bilgi paylaşmak için kullanılabilir, ancak birçok başvuruda bir taraf, farklı güven alanlarında ve kuruluşlarda verilen nitelikler veya kimlik bilgileri üzerine yazılmış bir politikaya göre verileri paylaşmak istemektedir. Örneğin, bir taraf tıbbi verileri yalnızca bir tıbbi kuruluş tarafından verilen "Doktor" ve bir klinik araştırmanın yöneticileri tarafından verilen "Araştırmacı" özelliğine sahip bir kullanıcıyla paylaşmak isteyebilir. Mevcut ABE sistemlerinin bu uygulamalar için kullanılması, farklı organizasyonlar arasındaki öznitelikleri doğrulayabilen ve sistemdeki her kullanıcıya özel anahtarlar verebilen tek bir otoriteye ihtiyaç duyduğundan sorunlu olabilir (Lewko & Waters, 2011). Bundan dolayı merkezi olmayan ABE blokzincir uygulamalarında kullanılabilir. *Zero Knowledge SNARK (Sıfır-Bilgi Özlü İnteraktif Olmayan Bilginin Argümanı):* zk-SNARK sıfır bilgi ispatına dayalı bir sistemdir. Sıfır bilgi ispatı, bir kişinin belli bir bilgiye sahip olduğu ve bu bilgiyi açığa çıkarmadan ispatlayabildiği yapılardır. Blokzincir yapılarında bu bilgi bir gizli anahtar olabilir. Şu anda, etkileşimli olmayan ve bir blokzincirine yayınlanacak kadar kısa sıfır bilgi ispatları üretmenin bilinen en etkili yolu, kanıtlayıcı (prover) ve doğrulayıcı (verifier) arasında paylaşılan ortak bir referans dizesi (string) oluşturan bir başlangıç kurulum aşamasına sahip olmaktır. Bu ortak referans dizesi sistemin genel parametreleridir (Banerjee vd., 2020). Shamir, D verisinin herhangi bir k parçadan kolayca yeniden oluşturulabileceği şekilde D verisinin nasıl n parçaya bölüneceğini göstermiştir, bunun yanında k-1 parçanın tam bilgisi bile D hakkında kesinlikle hiçbir bilgi ortaya çıkarmadığını göstermiştir. Bu teknik, kriptografik sistemler için, olumsuz bir olay parçaların yarısını yok ettiğinde ve güvenlik ihlalleri kalan parçaların biri hariç hepsini açığa çıkarsa bile güvenli ve güvenilir bir şekilde çalışabilen sağlam anahtar yönetim şemalarının oluşturulmasını sağlamaktadır (Shamir, 1979). Literatürde yer alan gizlilik ve güvenlikle ilgili çalışmaların konusu, blokzincir ağı, eğer varsa güvenlik protokolü, kimlik ve veri gizliliğinin sağlanıp sağlanmaması, veri bütünlüğü ve ölçeklenebilirlik açısından değerlendirilmesi Tablo 2.2'de özetlenmiştir. Bu sayede bu çalışmaların ağırlık verdiği parametreler ön plana çıkmaktadır.

Tablo 2.2. Gizlilik ve güvenlik tabanlı çalışmaların değerlendirilmesi

Referans	Tanım	Geliştirme Çatısı	Güvenlik / Gizlilik Protokolü	Harici Şifreleme Mekanizması	Kimlik Doğrulama	Kimlik Gizliliği	Veri Gizliliği	Ölçeklenebilirlik
(Zhang vd., 2018)	Blokzincir ağında yerel bir oylama protokolü	Hyperledger Fabric	Konsensüs mekanizması	Homomorfik Şifreleme	Evet	Evet	Evet	Hayır
(Eyal vd., 2016)	Ölçeklendirme için tasarlanan blokzincir protokolü önerisi	Bitcoin	Konsensüs mekanizması	Hayır	Hayır	Hayır	Evet	Yüksek
(Kopp vd., 2017)	Kullanıcı mahremiyeti tabanlı dosya depolama	Bitcoin	Halka imza ve tek seferlik adres	Evet	Evet	Evet	Evet	Orta
(Sasson vd., 2014)	Gizlilik tabanlı dijital para birimi: Zerocash	Zerocash	zk-SNARK	Eliptik Eğri Şifrelemesi	Evet	Evet	Evet	Orta
(Rahulamathavan vd., 2017)	Öznitelik tabanlı şifreleme ve IoT için gizlilik tabanlı blokzincir mimarisi	Bitcoin	ABE	ABE	-	Evet	Evet	Orta
(Li vd., 2018)	Blokzincir tabanlı kimlik doğrulama ve güvenlik mekanizması	Akıllı Sözleşme Tabanlı	-	Hayır	Evet	Hayır	Hayır	-
(Biswas vd., 2018)	IoT'de güvenli işlemler için ölçeklenebilir blokzincir çerçevesi	Hyperledger Fabric	Konsensüs mekanizması	Hayır	Evet	-	-	Yüksek
(Sarıtekin vd., 2018)	Blokzincir tabanlı güvenli iletişim uygulaması	Bitcoin	IPFS	-	Evet	Evet	Hayır	-

Tablo 2.2 Tablonun Devamı

(Kosba vd., 2016)	Gizlilik tabanlı bir akıllı sözleşme modeli: Hawk	Ethereum	zk-SNARK	Senc (Seçici Şifreleme)	Evet	Evet	Evet	-
(Gennaro vd., 2016)	Bitcoin cüzdan güvenliği uygulaması	Bitcoin	DSA, ECDSA	Homomorfik Şifreleme, Gizli Paylaşım, RSA	Evet	Evet	Evet	-
(Miller vd., 2017)	Monero blokzincirinde izlenebilirliğin değerlendirilmesi	Monero	Cryptonote protokolü	-	Evet	Evet	Evet	-
(Van Saberhagen, 2013)	Cryptonote kriptoparası önerisi	Cryptonote	Cryptonote protokolü, Halka imza	Diffie-Hellman	Evet	Evet	Evet	Yüksek
(X. Zheng vd., 2018)	Blokzincir tabanlı kişisel sağlık verisi paylaşım sistemi	Ethereum	Konsensüs mekanizması	AES, Shamir'in sır paylaşım şeması	Evet	Evet	Evet	-
(Sun vd., 2017)	Monero'da bulunan RingCT protokolünün güvenlik incelemesi	Monero	RingCT	-	Evet	Evet	Evet	Yüksek
(Banerjee vd., 2020)	zk-SNARK sıfır bilgi ispatının Zcash algoritmasındaki rolü	Zcash	zk-SNARK, ECDSA	ECDSA	Evet	Evet	Evet	-
Önerilen HLF şeması	Şifrelenmiş görüntünün depolanması ve doğrulanması	Hyperledger Fabric	Konsensüs mekanizması	SSS, RSA	Evet	Hayır	Evet	Yüksek

2.3.Fikir Birliği Algoritmaları

Ağda bulunan farklı eşlerin (düğümlerin) blokzincirde bir bloğu onaylamak amacıyla anlaşmaya varmasını sağlayan protokol ve matematiksel hesaplama yöntemlerine dayanan algoritmalara fikir birliği (konsensüs) algoritmaları adı verilir.

Bach ve arkadaşları güvenlik, enerji tüketimi, ölçeklenebilirlik gibi parametrelere bağlı olarak en çok kullanılan konsensüs algoritmalarının kıyaslamalı analizini gerçekleştirmişlerdir. Proof of Work (PoW) algoritmasına sahip bir sistemin saldırganlar tarafından kontrol edilebilmesi ve işlemlerin değiştirilmeye zorlanabilmesi için sistemin hesaplama gücünün en az %25'inin gerekli olduğunu belirtmektedirler. Proof of Stake (PoS) algoritmasında ise ağda bulunan hissenin %51'i gereklidir (Bach vd., 2018). Baliga, popüler blokzinciri konsensüs algoritmalarının detaylı analizini yapmıştır. Bu analizde kullanılan blokzincirin türü (izinli, izinsiz), işlem sonlandırma yapısı, işlem hızı, token ihtiyacı, katılım maliyeti, eşler arası ağın ölçeklenebilirliği, güven modeli ve saldırı toleransı gibi parametreler mevcuttur. Ayrıca bir konsensüs protokolünün uygulanabilirliği için üç temel özelliğe sahip olmasının gerektiğini vurgulamıştır. Bunlar güvenlik, canlılık ve hata toleransıdır (Baliga, 2017). Canlılık, fikir birliği protokolünün işlemlerin sonunda kesin olarak bir değer üretmesidir. Vukolic, ilk geliştirilen PoW konsensüs algoritmalarındaki ölçeklenebilirliğin artık günümüzde bir anlam ifade etmediğini belirtmektedir. Ethereum gibi modern kripto para birimlerinin rasgele dağıtılmış uygulamaları desteklediğinden dolayı çok daha iyi performansa ihtiyaç duyduklarını belirtmektedir. PoW ve Bizans Hata Toleransı (BFT) tabanlı blokzincirlerini düğüm ölçeklenebilirliği ve performans açısından karşılaştırmıştır (Vukolić, 2016). Mevcut blokzincirlerin en büyük problemleri çift harcama (double spending) ve bizans generalleri problemidir. Çalışmasında bu problemlere değinen Alsunaidi ve Alhaidari (Alsunaidi & Alhaidari, 2019), ispat tabanlı ve oylama tabanlı blokzincir konsensüs algoritmalarını düğüm kimliği yönetimi, veri modeli, enerji tasarrufu, işlem ücreti, blok ödülü, doğrulama hızı, saniye başına işlem sayısı, blok oluşturma hızı, ölçeklenebilirliği, genişletilebilirliği, çift harcama olasılığı, Bizans hata toleransı ve uygulama alanları gibi birçok yönden inceleyerek detaylı bir analiz yapmışlardır. Lei ve arkadaşları, pratik BFT (PBFT) algoritmasının özellikle konsorsiyum blokzincirinde kullanıldığından bahsetmişlerdir. Bu tür fikir birliği algoritmalarında bulunan zararlı düğümlerin gerçek zamanlı tespitinin oldukça zor olduğunu belirtmişlerdir. Bundan dolayı İtibar Tabanlı Bizans Hata Toleransı (RBFT) algoritmasını geliştirmişlerdir. Blokzincirde bulunan tüm düğümlerin işlemlerini değerlendirmek amacıyla itibar modeli içeren bir algoritma önermişlerdir. İtibar seviyesi,

hatalı düğümlerin blok üretme sürecinde söylem haklarının azalmasına neden olmaktadır. PBFT ile kıyaslandığında %15 daha verimli ve %10 daha az gecikmeye sahip olduğunu ileri sürmektedirler (Lei vd., 2018). Tendermint fikir birliği protokolünde, konsensüs sürecinde bloklar için oyları imzalayan doğrulayıcılar bulunmaktadır. Bu protokolde 3 tip oylama bulunmaktadır; bunlar ön oylama, ön-taahhüt ve taahhüt aşamalarıdır. Oylama ve taahhüt aşamasında doğrulayıcıların 2/3'ünden taahhüt alınması gerekmektedir. Eğer doğrulayıcıların 2/3 çoğunluğu bir bloğu imzalar ve eşler arası ağda yayınlarsa bu blok kabul edilmektedir (Kwon, 2014). Üç aşamalı bir yapıda olması bu algoritmayı güvenlik açısından güçlendirir fakat, işlem hızı bakımından yavaşlatabilir. Blokzincir çalışmalarında bulunan fikir birliği algoritmaları yukarıda belirtilenler ile sınırlı değildir. Fikir birliği algoritması blokzincirin mimari yapısı ile ilgili birçok bilgi verebilmektedir. Fakat bu tez çalışmasının temeli akıllı sözleşmeler olduğundan dolayı fikir birliği algoritmaları hakkında incelemeler dar kapsamlı tutulmuştur. Ayrıca 3. bölümde fikir birliği algoritmalarına yer verilmiştir.

2.4.Akıllı Sözleşmeler

Cachin, araştırmasında Hyperledger Fabric'in mimarisinden ve özelliklerinden bahsetmiştir. Mutabakat protokolü olarak PBFT'nin, yani Bizans hataya dayanıklı konsensüs uygulamasının kullanıldığı, güvenlik için TLS sertifikaları, kayıt sertifikaları ve işlem sertifikaları gibi Certificated Authority (CA) tarafından sağlanan bir yapı barındırdığını belirtmektedir. RocksDB tarafından desteklenen bir anahtar-değer saklama arayüzü bulunduğundan bahsetmiştir. Modüler mimariye sahip oldukça güçlü güvenlik ve kimlik özelliklerini barındıran bir blokzincir uygulaması olduğunu ifade etmektedir (Cachin, 2016). Blokzincir teknolojisinin benimsenmesi, tedarik zinciri boyunca veri toplayan ve üreticiden nihai müşteriye kadar her ürün grubunu izleyen dağıtılmış bir defterin oluşturulmasını amaçlamaktadır. Diğer bir hedef, gıda tedarik zincirini verimlilik ve hız açısından iyileştirmek, depo yönetimini desteklemek ve optimize etmek ve düzenlemelere uyulmasını sağlamaktır. Blokzincir teknolojisinin benimsenmesinin temel faydalarından bahsetmiştir (Perboli vd., 2018). Bunlar;

- İş gücü ve kapasite planlaması açısından gelen etkinliğin iyileştirilmesi,
- Güvenli olmayan stoklama koşullarından kaynaklanan son kullanma tarihi geçmiş ürünlerin veya atığın azaltılması,
- Sağlık düzenlemelerine uyumu sağlamak için bileşenlerin izlenmesinde doğruluğun iyileştirilmesi,

- Tedarik zincirinde blokzincirin benimsenmesinin faydaları, esas olarak sahteciliğin azaltılması ve müşteri güveninin artmasıyla satışlarda artışa neden olması

Manevich ve arkadaşlarına göre, Hyperledger Fabric'teki (HLF) işlem akışı Bitcoin veya Ethereum'a kıyasla daha karmaşıktır ve bu akışı koordine etme sorumluluğu müşteriye düşmektedir. Müşteri birden fazla varlıkla iletişim kurmalı ve bunlardan haberdar olmalıdır: Zincir kodu konumu, onay politikası, onaylayan eşler ve sipariş hizmeti. Tüm bu varlıklar, platformun yaşam döngüsü boyunca değişebilir: eşler ve kuruluşlar gelip gidebilir, zincir kodu ve onay politikaları güncellenebilir. HLF'nin izinli bir defter olması, istemcinin yapılandırmasının, çok teknik ve hataya açık olabilen kimlik doğrulama, yetkilendirme ve erişim kontrolünün karmaşıklıklarını içerdiği anlamına gelir (Manevich vd., 2019). Androulaki ve arkadaşları (IBM), blokzincirler için sipariş yürütme mimarisi, emir yürütme sıralamaları, mevcut mimariler ve Fabric'in avantajları'ndan (örn: değiştirilebilir konsensüs uygulamaları) bahsetmişlerdir. Ayrıca Fabric bileşenleri olan üyelik servisi, sipariş servisi, eşler arası iletişim (Peer Gossip), defter, chaincode çalıştırma ve konfigürasyon (özellikle kanal konfigürasyonu) ile ilgili detaylı bilgilere yer verilmiştir (Androulaki vd., 2018). Blokzincir, dijital bir işlem kaydının oluşturulmasına izin veren dağıtılmış bir veritabanıdır. Merkezi bir otoriteye ihtiyaç duymadan işlemlerin güvenli bir şekilde kayıt altına alınmasını sağlayan merkezi olmayan bir sistemdir. Kriptografik temellere dayanan etkin bir iletişim ortamı sağlar. Bu sayede birçok alanda etkin bir şekilde kullanılmaktadır. Blokzincir, sağlık hizmetleri (Hawashin vd., 2021), sigorta (Bader vd., 2018), tedarik zinciri (Helo & Hao, 2019), merkezi olmayan finans (Ethereum, 2015), kimlik takibi (Süzen & Duman, 2021), oylama (Yavuz vd., 2018), çevrimiçi oyunlar (Boroń & Kobusińska, 2021), eğlence (Chavan vd., 2019), yatırım (Gil-Alana vd., 2020), askeri (Krichen vd., 2022) dahil olmak üzere birçok alanda kullanılmaktadır. Ethereum akıllı sözleşmeleri kullanılarak çeşitli sektörlerde mevcut yöntemlerin iyileştirilmesi için çaba sarf edilmektedir. Ethereum akıllı sözleşmeleri, özellikle oyun, takas, kumar, finans, mülk, cüzdan vb. alanlarda kullanılmaktadır (Ethereum, 2015). Alassaf ve Yusoff, güven ve şeffaflığa dayalı bir bağış uygulaması geliştirmek için blokzincir kullanmıştır. Hayır kurumlarına güvenin sorgulanması ve şeffaflığın bir zorunluluk haline gelmesi nedeniyle blokzincir teknolojisi kullanımının dürüstlüğü artıracağını belirtmiştir. Şeffaf bağış çalışmaları için en uygun sistemin Ethereum platformu olduğunu savunmuşlardır. Şeffaf bağış için uygun blokzincir platformunun belirlenmesinde ağ türü, kripto para birimi, akıllı sözleşme ve konsensüs algoritması seçiminin önemli bir rol oynadığını belirtmiştir.

Karar destek mekanizmasına göre uygun platformun seçilebilmesi için blokzincir sisteminin ağ tipinin herkese açık olması, güvenlik açısından mutabakat algoritmasının Proof of Work veya Proof of Stake olması, akıllı sözleşmelere ve kripto para birimlerine dayalı olması ve Ethereum, yüksek işlem hızı gibi parametrelere bağlı olarak belirlenir. Ayrıca geliştirdikleri merkezi olmayan uygulama (DApp) ile hayırsever değerlendiriciler ve bağışçılar sisteme dahil edilerek dolandırıcılık durumlarının önüne geçilmektedir. Kampanya organizasyonu ve bağış fonları için iki ayrı akıllı sözleşme belirlenmiştir (Alassaf & Yusoff, 2021). Benzer bir çalışmada (Saleh vd., 2019), şeffaf bir bağış platformu modeli önerilmiştir. Yardım fonlarına yapılan bağışların nereye, ne zaman ve kime gittiğinin takibi ve izlenmesi amaçlanmıştır. Bağışçılar, banka hesaplarından veya mobil uygulamalardan yapılan bağışların istenilen kuruma ulaşıp ulaşmadığını bilmemektedir. Bu nedenle Ethereum ve Hyperledger Fabric gibi akıllı sözleşme tabanlı bir sistem önermişlerdir. Bağış platformu oluşturmak isteyenler için bir kriter tablosu oluşturmuşlardır. Tablo, blokzincir platformunun herkese açık mı yoksa özel mi olduğu, saniyede gerçekleştirebileceği işlem sayısı, blok boyutu ve mutabakat protokolü hakkında bilgiler içerir. Bu bilgilerin dışında platformun güvenilirliği, ölçeklenebilirliği, dinamik yapısı, işlem ücretleri, hedef kitle büyüklüğü ve geliştirildiği programlama dili önemlidir. Lee ve arkadaşları (Lee vd., 2018) şeffaf bağış konusuna değinerek mahremiyete dikkat çekmektedir. Bağışçılar veya yararlanıcılar bazı durumlarda bunun olmasını istemezler. Hem şeffaflık hem de gizliliğe dayalı bir sistemin gerekliliğini ifade eder. Önerdikleri çalışmada, kişisel bilgileri korumak için tek seferlik bir hesap adresi kullanan bir sistem tasarladılar. Adının açıklanmasını istemeyen bağışçının adresi, yetkili olarak belirtilen oy grubu tarafından farklı bir adrese değiştirilir ve bağış, yardımı alan kişiye aktarılır. Bu sayede kişi kendisine bağış yapan bağışçıyı tanımaz. Vericinin kimliğinin gizliliği, Diffie-Hellman anahtar değişim algoritmasına benzer bir yapı ile sağlanmaktadır. Doğrulama, dijital imza algoritmalarında da bulunan özel ve genel anahtarlar kullanılarak yapılır (Lee vd., 2018). (Wu & Zhu, 2020), hayır kurumlarının hizmet talepleri için fizibilite ve güvenilirliği ölçmek için blokzinciri tabanlı bir mimari önermektedir. (Saleh vd., 2019) ve (Lee vd., 2018) 'de yapılanlara benzer öneriler bulunsa da kullanılabilecek blokzincirin okuma-yazma erişimi, merkezilik düzeyi, kullanım alanları hakkında bilgiler bulunmaktadır. Bilinen, stabil ve iyi işleyen bir mekanizmaya sahip olduğu için Ethereum'un daha çok tercih edildiği kabul edilmektedir. Hayır kurumu bağış yönetimi için Ethereum tabanlı CharityCoin dijital para birimi (Farooq vd., 2020)'te kapsamlı bir şekilde sunulmuştur. CharityCoin için İlk Para Arzı (ICO), resmi para birimi ile CharityCoin satın alma, çerçevenin katmanları, bağış toplama ve akıllı sözleşmeye dağıtım aşamaları ayrıntılı olarak anlatılmaktadır. Çalışmaları diğerlerinden

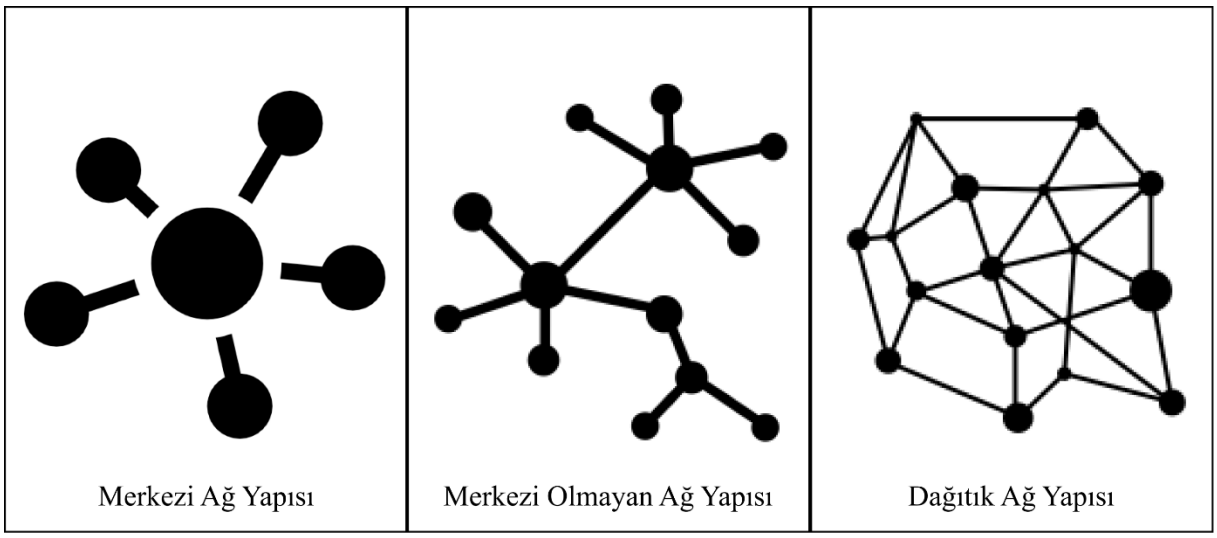
farklı kılan sebep, bir dijital para biriminin arzı gerçekleştirilmesi ve gerçek zamanlı bir sistem geliştirilmesidir. Sistemde bağış için gerekli olan iki temel akıllı sözleşme bulunmaktadır. Bu sayede bağışların geçerliliği ve kişisel mi yoksa kurumsal bağış mı olduğu kontrol edildikten sonra doğrulama ve güvenlik analizi yapılarak işlemler gerçekleştirilir (Farooq vd., 2020). Akıllı sözleşmelerin ayrıntılı bir incelemesinde (Oliva vd., 2020), bunların Ethereum platformuna ilişkin analizleri verilmektedir. Ethereum'da kullanılan bir sözleşmenin aktivite seviyesi, yaşı ve mevcut boşta kalma süresi analiz edildi. Şu anda teyitli kontratların %2,2 seviyesinde olduğu ve bu kontratların %72 civarında kullanıldığı belirtiliyor. Test amacıyla kullanılan birçok onaylanmamış sözleşme var gibi görünüyor. Araştırmaya göre doğrulanmış sözleşmelerin kullanım alanları sırasıyla oyun, takas, kumar, finans, mülk, cüzdan ve diğerleri olarak söylenebilir. Benzer şekilde (A. Singh vd., 2020), (Shaheen vd., 2021), (Mohite & Acharya, 2018) vb. çalışmalar yardım, bağış ve fon gibi faaliyetleri izlemek amacıyla yapılmıştır. Ayrıca kan bağışı (Sadri vd., 2021) ve özel olarak Ethereum tabanlı akıllı sözleşmelerin kullanılması için tasarlanmış bir çözüm (Hawashin vd., 2021)'de sunulmuştur. Kan bağışında farklılıklar olmakla birlikte mahremiyet ve güvenlik ön plandadır. Ayrıca bu çalışmada yer alan ancak üzerinde durulmayan depolama ve kimlik doğrulama sistemlerinde private blokzincirlerin kullanılması güvenliği arttıran bir diğer unsurdur. Tanrıverdi, araştırmasında blokzincir tabanlı özel bir kimlik doğrulama sistemi geliştirmiştir. Blokzincirindeki veriler üzerindeki tüm işlemlerin kayıt altına alınması, özel blokzincir sayesinde izinlerle sisteme dahil edilmesi, kamu-özel anahtarı ile gizliliğin sağlanması gibi özellikleri ile tercih nedeni olmuştur (Tanrıverdi, 2020). Başka bir çalışmada (Farooq vd., 2020), e-ticaret uygulamasında kullanılan kredi kartı bilgileri özel bir blokzincir kullanılarak saklanmaktadır. Gizliliğin sağlanması için her bir kredi kartı bilgisi ayrı bir blokta saklanmakta ve şifrelenmektedir.

3. BLOKZİNCİR VE ÖZELLİKLERİ

Bu bölümde, öncelikle blokzincirin genel yapısı üzerinde durulmuş olup devamında da blokzincir temel prensibleri kıyaslamalı olarak analiz edilmiştir. Ayrıca genel, özel ve konsorsiyum blokzincirlerin özelliklerinin kıyaslanması yapılmıştır. Son olarak fikir birliği algoritmaları hakkında bilgi verilmiş ve algoritmaların kıyaslamalı analizi yapılmıştır.

3.1.Blokzincirin Genel Yapısı

Blokzincir temel olarak para, kimlik, değerli kağıtlar, akıllı kontratlar gibi verilerin güvenli ve emin bir şekilde depolanması ve yönetilmesi için tasarlanmış bir teknoloji olarak tanımlanabilir. Günümüz yaklaşımlarında genellikle kapalı merkezi sistemler kullanılmaktadır. Şekil 3.1'de farklı ağ türlerine ait topoloji şekilleri mevcuttur. Blokzincirin amacı tüm bilgileri dağıtık/merkezi olmayan (distributed), katılımcılara açık bir ağ yapısı üzerindeki tüm düğümlerde eşlenik kopyalar halinde tutmaktır. Bu şekilde tekil bir ara kuruma ihtiyaç ortadan kalkmakta, bu durumun getirdiği maliyetler ve riskler ortadan kaldırılmaktadır.



Şekil 3.1. Ağ yapıları ve blokzincir dağıtık ağ yapısı.

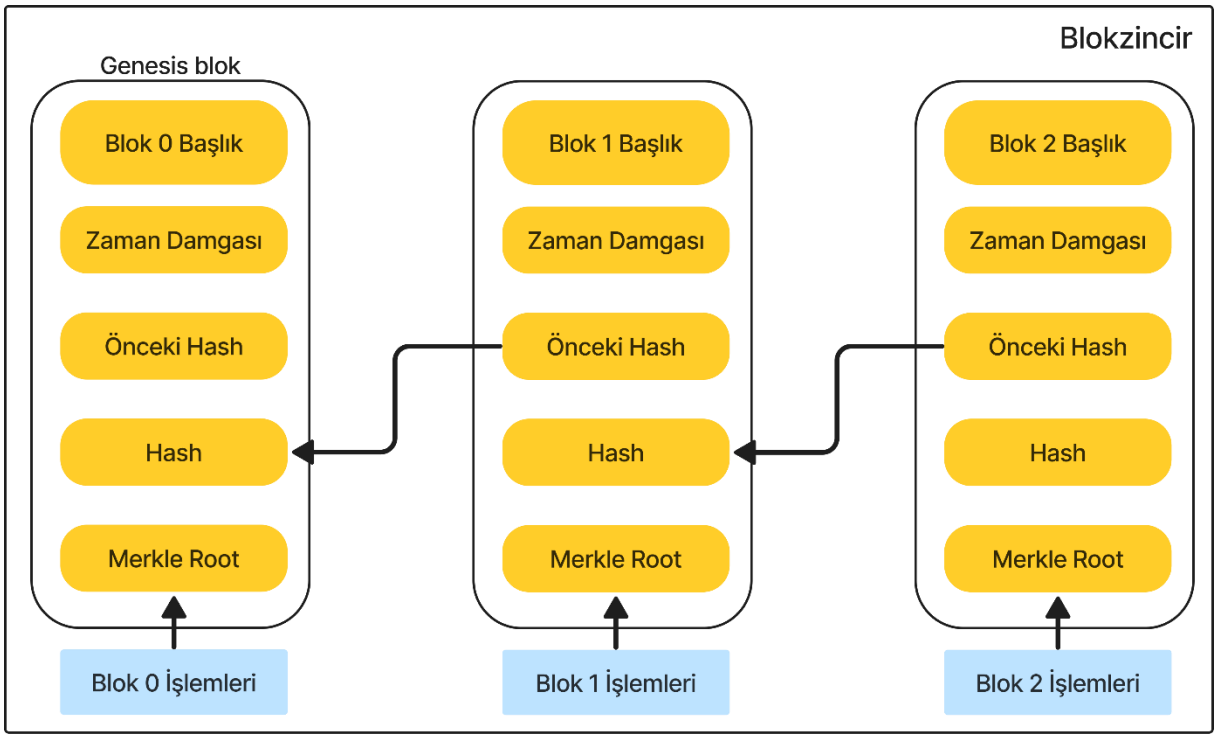
Blokzincirin temeli olan Bitcoin 2008 yılında Satoshi Nakamoto (Nakamoto, 2008) lakaplı gizli bir yazar tarafından ortaya atılmıştır. Blokzincir altyapısını kullanan Bitcoin dijital parası ile birlikte dünyada yeni bir uluslararası para biriminin varlığından bahsedilmeye başlanılmıştır. Bitcoin, başlangıçta sadece para olarak telakki edilirken, daha sonra Bitcoin'in altyapısında bulunan blokzincir altyapısının çeşitli alanlarda ve uygulamalarda kullanılabileceği görülmüştür. En genel ifadeyle, blokzincir, merkezi bir sunucunun veya

güvenilir bir otoritenin kaldırılmasına olanak sağlayarak, merkezi güvenin internet ortamında dağıtılmasına denir. Blokzincir teknolojisinin Bitcoin, Ethereum, Bytecoin, Monero, Zcash gibi sanal paraların altyapısını oluşturduğu bilinmektedir. Fakat bu teknoloji sağladığı olanaklar ve çeşitlendirilebilir uygulamaları ile çok daha geniş bir yelpazeye sahiptir. Tabi ki bu olanakların genişlemesi güvenlik ve gizlilik gibi önlemlerinin alınmasını gerektirmektedir.

Bitcoin bloğu başlığında temel olarak aşağıdaki bilgiler mevcuttur:

- Bir önceki bloğa ait özet değeri (Previous hash)
- Merkle kök değeri (Merkle root)
- Zaman damgası (Timestamp)
- Nonce (istenilen blok özet değerini SHA-256 özet algoritmasını kullanarak üretmek amacı ile kullanılan değiştirilebilen sayı değeri)

Bir blokzincir blok başlığı, zaman damgası önceki bloğun blok başlığının özeti, merkle root değeri ve işlemlerden (transactions) meydana gelmektedir. Bitcoin blokzincir yapısı Şekil 3.2'de görülmektedir.



Şekil 3.2. Bitcoin blokzincir yapısı.

3.2.Blokzincirin Temel Prensipleri

Blokzincir teknolojisinin temel prensiplerini oluşturan başlıklar aşağıda görülmektedir. Bu başlıkların bazı popüler blokzincirlerde ifade ettiği temel bileşenler analiz edilerek

açıklanmıştır.

Dağıtılmış veritabanı: Blokzinciri elinde bulunduran herkes tüm veritabanına ve geçmişe erişebilir.

Eşler arası iletim: İletişim bir merkezden kontrol edilmek yerine eşler arasında vasıtasız gerçekleşir.

Takma ad ile şeffaflık: Sistemin kurulu olduğu herkes her işlemi ve ilişkili olduğu değerleri görebilir.

Kayıtların geri dönüşümsüzlüğü: Her işlem veritabanına girildikten sonra tüm hesaplarda güncelleme yapılır, kayıt silinemez, çünkü tüm işlem kayıtları kendinden önceki kayıtlara bağlıdır.

Hesaplamalı mantık: Defterin dijital yapısı, blokzincir işlemlerinin hesaplamalı mantığa (logic) ve programlanan niteliğe bağlı olabileceği anlamına gelir (Iansiti & Lakhani, 2017).

Blok Yapısı: Blokzincirdeki veriler, bloklar halinde dağıtık defterde tutulur. Bloklarda yapılan işlemlerin ve önceki bloğun bir parçası olan bir kriptografik bağlantının (hash) birleşimi yer almaktadır. Blokların yapısını ve bütünlüğünü korumak için hash gereklidir.

İşlem Onayı: Blokzincirdeki işlemler, ağdaki düğümler tarafından matematiksel algoritmalar kullanılarak doğrulanır. İşlemlerin geçerliliği ve bütünlüğü için doğrulama süreci gereklidir. Blokzincir ağları, konsensüs mekanizmaları aracılığıyla işlem geçerliliğini ve blokların eklenmesini sağlar. Konsensüs mekanizmaları, farklı düğümler arasında fikir birliği sağlayarak güvenilir bir blokzincir ağı oluşturur. Blokzincir yapısı bloklar ve bu blokları oluşturan kayıtlardan meydana gelir. Bu kayıtlar kayıt defteri denilen bir yapıda tutulur.

İşlem Süresi: Blokzincirdeki işlemler, bloklara eklenmeden önce ağdaki düğümler tarafından doğrulanır. İşlem onayı belirli bir süreçten geçtiğinden dolayı blok eklemek için belirli bir süreye ihtiyaç duyulmaktadır.

Kriptografi: Blokzincirdeki veriler ve işlemler, güvenliği sağlamak için kriptografik algoritmalar kullanılarak şifrelenir. Kriptografi, verilerin gizliliğini, bütünlüğünü ve kimlik doğrulamasını sağlar. Bilgi güvenliği, bir bilginin sadece yetkili kişiler tarafından erişilebilmesi, kullanılabilmesini sağlamak ve yetkisiz kişilerin bilgiye erişimini, değiştirmesini ve zarar vermesini engellemek olarak tanımlanabilir. Kendi içerisinde veri,

uygulama, ağ, uç nokta güvenliği gibi alanlara ayrılmaktadır. Bilgi güvenliği üç temel yapıdan oluşur:

- *Gizlilik:* Bilginin yetkisiz kişilerin eline geçmesinin engellenmesi anlamına gelir. Yalnızca yetkilendirilmiş kullanıcıların erişebilmesi ve bunun dışında gizli kalması gerekir. Hassas ve kişisel bilgiler gizlilik açısından büyük önem taşımaktadır. Blokzincirde anonimlik ve veri gizliliğini esas alır.
- *Bütünlük:* Yetkisiz kişilerin bilgiyi değiştirmesinin engellenmesi anlamına gelir. Veri bütünlüğünün korunması ve müdahalelere karşı önlem alınması gerekmektedir. Ayrıca veri değişikliklerinin izlenmesi, kontrol ve doğrulama mekanizmaları sayesinde veri bütünlüğü sağlanmaktadır. Blokzincirde dağıtılmış doğrulama ve şeffaflığı temel alır.
- *Erişilebilirlik:* Yetkili kişilerin bilgiye ihtiyaç duyduğunda ulaşabilmesi ve kullanabilmesi anlamına gelmektedir. Blokzincirde dağıtılmış yapı, işlem hızı ve ölçeklenebilirliği temel alır.

Tablo 3.1. Blokzincir temel prensiplerine dayalı kıyaslamalı analiz

Blokzincir	Takma Ad ile Şeffaflık	Defterin Dijital Yapısı	İşlem Onayı	İşlem Süresi (işlem/saniye)	Blok Süresi	Kriptografi	Gizlilik
Bitcoin	Hayır	Açık	Proof of Work (PoW)	7 (PoW için)	10 dakika	SHA-256	Düşük
Ethereum	Evet	Açık	Proof of Stake (PoS)	3 (PoS için)	13-15 saniye	Ethash	Orta
Hyperledger Fabric	Evet	Özel	Konsorsiyum (PBFT)	Ölçeklenebilir	Ölçeklenebilir	AES	Yüksek
Cardano	Evet	Açık	Proof of Stake (PoS)	257	20 saniye	ECDSA, Blake2b	Orta
Stellar	Evet	Açık	Federated Byzantine Agreement (FBA)	1000	5 saniye	ECDSA	Orta
Ripple	Evet	Açık	Consensus Protocol	1500	3-5 saniye	ECDSA, ECIES	Orta
Zcash	Evet	Gizli	Proof of Work (PoW)	Ölçeklenebilir	2.5 dakika	zk-SNARK	Yüksek
Solana	Evet	Açık	Proof of History (PoH)	Ölçeklenebilir	0.4 saniye	ECDSA	Orta

Blokzincirin temel prensiplerine dayalı oluşturulan blokzincirlerin özellikleri ve bunların birbirlerine göre kıyaslamalı durumu Tablo 3.1'de görülmektedir. Tablo 3.1'e eklenmeyen maddeler, belirtilen blokzincirler için ortak olan özelliklerdir.

3.3.Blokzincir Türleri ve Özellikleri

Blokzincir sistemleri genel, özel ve konsorsiyum blokzincir olarak üç türe ayrılır. Genel blokzincir özellikleri:

1. Herhangi bir düğüm konsensüs sürecine katılabilir.
2. Herkese görünür işlemlerden oluşur.
3. Gerçek blokzinciri değiştirmek mümkün değildir.
4. İşlemler ve bloklar çok sayıda düğüm nedeniyle yavaşça yayılır. İşlemler sınırlıdır ve gecikme süresi yüksektir ve ayrıca ağ güvenliği de çok önemlidir.
5. Merkezi olmayan yapıya sahiptir.
6. İzinsizdir. Herkes dünyadaki uzlaşma sürecine katılabilir.

Özel blokzinciri özellikleri:

1. Yalnızca seçilen bir düğüm kümesi bloğunu doğrular.
2. Okuma izni özel bir blokzincire bağlıdır.
3. Blokzincir, çoğunluğa bağlı olarak değiştirilebilir veya tersine çevrilebilir.
4. Daha az sayıda doğrulayıcı kullanılarak daha verimli bir blokzinciri elde edilebilir.
5. Tamamen merkezi bir yapıya sahiptir.
6. İzinlidir. Bir düğüm, onaylanmadıkça fikir birliği sürecine katılamaz.

Konsorsiyum (organizasyon) blokzinciri özellikleri:

1. Oybirliği bir grup tarafından sağlanır.
2. Okuma izni bir konsorsiyum blokzincirine bağlıdır. (İstenirse konsorsiyum kısıtlamaları kaldırabilir.)
3. Blokzincir çoğunluğa bağlı olarak değiştirilebilir veya tersine çevrilebilir.
4. Daha az sayıda doğrulayıcı kullanılarak daha verimli bir blokzinciri elde edilebilir.
5. Kısmen merkezi bir yapıya sahiptir.

6. İzinlidir. Bir düğüm, onaylanmadıkça fikir birliği sürecine katılamaz.

Konsorsiyum blokzincirinde oy tabanlı konsensüs algoritmaları kullanmak daha uygundur. PoW gibi ispat tabanlı algoritmalarda bulunan düşük TPS, yüksek gecikme ve maliyet konsorsiyum blokzincir yapısına uygun değildir. Konsorsiyum blokzincirinde "müşterinizi tanıyın" (Know Your Customer - KYC) yapısı bulunmaktadır. Buna bağlı olarak düğümler fikir birliğine ulaşmak için birkaç tur karşılıklı oylama gerçekleştirmektedirler. Genel, özel ve konsorsiyum blokzincir yapıları ait yukarıda verilen özellikler Tablo 3.2'de görülmektedir.

Tablo 3.2. Genel, özel ve konsorsiyum blokzincirlerin özelliklerinin kıyaslanması.

Parametreler	**Genel Blokzincir**	**Özel Blokzincir**	**Konsorsiyum Blokzincir**
İşlem Hızı (TPS)	7-30	100 ve üzeri	100 ve üzeri
Ölçeklenebilirlik	Zorluğa Duyarlı	İyi	Orta
Güvenlik	Yüksek	Yüksek	Yüksek
Erişim	Herkese Açık	Sınırlı	Sınırlı
İşbirliği	Tam Merkezsiz	Sınırlı	Kısıtlı
Veri Gizliliği	Genellikle Açık	Yüksek	Orta-Yüksek
İşlem Maliyeti	Değişken	Daha Yüksek	Daha Düşük
İşlem Onay Süresi	10-60 dakika	1 saniye-10dakika	1 saniye – 10 dakika
Esneklik	Sınırlı	Yüksek	Yüksek
Uygulama Alanı	Geniş	Sınırlı	Kısıtlı
Güncelleme Süreci	Karmaşık	Daha Kolay	Daha Kolay
Konsensüs	PoW, PoS, dPOS	Özel Algoritmalar	Özel Algoritmalar
Okuma İzni	Herkese Açık	Sınırlı	Sınırlı
Değişmezlik	Yüksek	Yüksek	Yüksek
Verim	Değişken	Yüksek	Orta
Merkezilik	Tam Merkezsiz	Merkezi	Kısıtlı Merkeziyet
Konsensüs Süreci	Katılımcılar Arasında Oylama	Merkezi Yetkililer	Sınırlı Katılımcılar Arasında Oylama

Genel, özel ve konsorsiyum blokzincirlerinde konsensüs algoritmalarının seçimi önemlidir. Bu amaçla fikir birliği algoritmaları incelenerek avantajları ve dezavantajları

ortaya konulmuştur. (Z. Zheng vd., 2018)'de konsensüs algoritmalarının seçimi için gerekli kriterlerin bazıları belirtilmiştir.

3.4.Blokzincir Fikir Birliği Algoritmaları

Konsensüs algoritmaları, bilgisayar bilimlerinde dağıtılmış süreçler veya sistemler üzerinde onaylanarak belirli bir verinin tehlikeye girmesine izin veren bir uzlaşmadır. Uzlaşmada sistemlerinin güvenilir olup olmadığı önemli değildir. Bu nedenle, blokzincirn karşılıklı güven gerektirmediğinden emin olmak için uzlaşma algoritmaları kullanılır.

İyi bir fikir birliği algoritması güvenlik, verimlilik ve uygunluk içermelidir. Mevcut konsensüs algoritmalarının bazı avantajları ve dezavantajları vardır. Bu nedenle yeni fikir birliği algoritmaları geliştirilmektedir. Blokzincirde kullanılan bazı konsensüs algoritmaları;

- Proof of Work (PoW)
- Proof of Stake (PoS)
- Delegated Proof of Stake (DPoS)
- Pratik Bizans Hata Toleransı (PBFT)
- Proof of Authority (PoA)
- Proof of Elapsed Time (PoET)
- Proof of Capacity (PoC)
- Proof of History (PoH)

Konsensüs algoritmalarının etkilediği durumlar ve özellikleri aşağıdaki maddelerde belirtilmiştir. Bu maddelere bağlı olarak Tablo 3.3'de konsensüs algoritmalarının kıyaslamalı analizi ve her bir kıyaslamanın açıklamaları detaylı bir şekilde verilmiştir. İşlem kesinliği her biri için yüksek olduğundan tabloya eklenmemiştir.

- **Blokzincir türü:** Konsensüs modelinin kullanılabileceği izin verilen veya izinsiz olarak sunulan blokzinciri platformunun türünü belirtir.
- **İşlem kesinliği:** İşlemin nihai olarak kabul edilip edilmediğini belirtir. PoW ve PoET konsensüs modellerinde aynı anda birden fazla blok oluşma riski vardır. Burada, blokzincirinde geçici çatal üretilerek oluşan bloklar reddedilir. İşlemlerin onaylanması ve sonuçlandırılması için bekleme süresi artar.
- **İşlem oranı:** İşlemleri (transaction) doğrulama ve uzlaşma adımlarının hızlı gerçekleştiği platform işlem oranı açısından başarımı en yüksektir. Olasılıklı bir yaklaşım olan PoW, kriptografik bulmacayı çözmek için zorluğa bağlı olarak zaman harcamaktadır. BFT tabanlı konsensüs algoritmaları hızlı bir şekilde

doğrulama yapabildiğinden işlem oranı yüksektir.

- **Token ihtiyacı:** Token, Bitcoin gibi dijital bir varlık türüdür; fakat kripto para olmak zorunda değildir. PoW ve PoS modellerinde bir kriptografik token'a ihtiyaç vardır. Tasarım token'ın varlığına bağlı olarak geliştirilmiştir. Diğer modellerde, konsensüs algoritmalarının çalışması için token gerekmez.
- **Katılım maliyeti:** Konsensüs yapısına katılmak için maliyet gerekir. PoW, enerji tüketen bir yapıya sahip olduğundan belli özelliklere sahip bir cihaz gereklidir. PoS, başlangıç için bir ücret ödenmesini gerektirir.
- **Eşler arası ağın ölçeklenebilirliği:** Blokzincirdeki düğümler artarken konsensüs modelinin ölçeklendirilebilmesidir. BFT ve çeşitlerinde ölçeklenebilirlik yoktur. BFT'de ağda bulunan eşlerin sayısının yirmiyi geçmesi fazla miktarda yük meydana getirir.
- **Güven modeli:** Fikir birliği sağlamak amacıyla çalışan düğümlerin bilinmesi ve bu düğümlere güvenilmesinin/güvenilmemesinin gerekliliği konusunda bilgi sağlar. Ağda bulunan düğümlerin %50'den fazlası güvenilir olmalıdır. Konsensüs, du düğümlerin %33'ü tehlikeye girmediği sürece bozulmaz.
- **Hata payı toleransı:** Ağın bir bölümü fikir birliğinde herhangi bir etkiye yol açmadan tehlikeye girebilmektedir. Konsensüs algoritmaları bu bağlamda belirli bir eşik değerine sahiptir.

Mevcut fikir birliği (konsensüs) algoritmalarının bazı dezavantajlarından dolayı geliştirilmesi gerektiği görülmüştür. Bu konuda çalışmalar halen devam etmektedir ve yeni fikir birliği algoritmalarının ortaya çıkacağı açıkça görülmektedir. Bunun dışında blokzincirde güvenlik, gizlilik ve mahremiyet konularında da çalışmalar yapılmaktadır. Bitcoin blokzincirinde kişinin kimliği gizli tutulmaktadır; fakat yapılan tüm işlemler zincirde izlenebilir olduğundan %100 gizliliği sağlamadığı açıktır. Bu anlamda Monero ve Zcash gibi gizlilik tabanlı blokzincirler mevcuttur. Bu blokzincirlerinde kullanıcının kimliği, mahremiyeti ve yapılan işlemler belli protokoller aracılığıyla talep edilen nispette gizlenmektedir. Zcash blokzincir yapısında gizliliği (privacy) sağlamak amacıyla zk-SNARK sıfır bilgi ispatını kullanılmaktadır. Tablo 3.3'de konsensüs algoritmalarının kullanıldığı blokzincir platformu örnekleri verilmiştir. Tüm parametreler konsensüs algoritmaları için kıyaslanmış, fakat blokzincirin yapılandırmasına ve ağ yoğunluğuna bağlı değişen parametrelerden dolayı bazı kısımlar değişken ve konsensüslerin kendi arasında düşük veya yüksek gibi metrikler kullanılarak değerlendirilmiştir. İşlem oranları saniye başına işlem

sayısı bakımından değerlendirilmiştir. Konsensüs onayı için ihtiyaç duyulan token, işlemci gücü vb. tabloda ifade edilmiştir.

Tablo 3.3. Konsensüs algoritmalarının kıyaslamalı analizi.

Konsensüs Algoritması	Kullanıldığı Blokzincirler	İşlem Oranı (TPS)	Konsensüs ihtiyacı	Katılım Maliyeti	Ölçeklene-bilirlik	Güven Modeli	Hata Payı Toleransı
Proof of Work (PoW)	Bitcoin, Ethereum	7-10	İşlemci gücü, enerji	Yüksek	Düşük	İşlem gücü	%50
Proof of Stake (PoS)	Cardano, Ethereum 2.0	30-50	Token	Orta	Düşük	Token sahipliği	%50
Delegated PoS (DPoS)	EOS, Tron	300-1000	Token	Düşük	Orta	Güvenilir düğümler	Düşük (~%5)
Byzantine Fault Tolerance (BFT)	Ripple, Hyperledger Fabric	1000 ve üzeri	Oylama	Düşük	Yüksek	%66'lık konsensüs	%33
Practical Byzantine Fault Tolerance (PBFT)	Hyperledger Fabric	1000 ve üzeri	Oylama	Düşük	Yüksek	Güvenilir düğümler	Düşük
Federated Byzantine Agreement (FBA)	Stellar	Değişken veya 1000 ve üzeri	Token olabilir	Düşük	Yüksek	Güvenilir düğümler	Çok düşük (~%1)
Proof of Authority (PoA)	Ethereum Classic, VeChain	Değişken	Token olabilir	Düşük	Orta	Yetkili düğümler	Çok düşük
Proof of Elapsed Time (PoET)	Hyperledger Sawtooth	Değişken	Zaman	Düşük	Orta	Zaman damgası	Düşük
Proof of Capacity (PoC)	Burst, Chia	1000 ve üzeri	Depolama alanı	Düşük	Orta	Depolama kapasitesi	Değişken
Proof of History (PoH)	Solana	Değişken	Kriptografik fonksiyon	Düşük	Yüksek	Zaman damgası	Düşük

Katılım maliyeti ve ölçeklenebilirlik tüm konsensüslerin kendi aralarında kıyaslanmasıyla yüksek, düşük olarak nitelendirilmiştir. Blok onayı için gerekli güven modelinin belirlendiği araçlar tabloda görülmekedir. Hata payı toleransı tüm konsensüs mekanizmaları için kendi matematiksel altyapısına bağlı olarak hesaplanmaktadır. Teorik değerler tablonun ilgili sütununda görülmektedir. Konsensüs algoritmaları tüm parametreler ve platformun gerekliliklerine göre değişiklik göstermektedir. Bundan dolayı aralarında seçim yapmadan önce belirli bir analiz süreci gerekmektedir.

3.4.1.Proof of work (PoW) ve çalışma prensibi

Bitcoin, LiteCoin, Ethereum gibi elektronik para birimlerinin kullandığı SHA-256

algoritmasıdır. Blokzincirde bir önceki bloğun hash değeri ile nonce değeri kullanılarak bir SHA-256 değeri oluşturulmaya çalışılır. İşlemci rastgele denemeler yaparak nonce değerini bulmaya çalıştığından kaynak tüketimi oldukça fazladır. Bundan dolayı blok üretim hızı düşük ve enerji tüketimi nedeniyle çevre düşmanı olarak nitelendirilmektedir.

Bir blokzincirine bir blok eklemek için, bir düğümün bazı işler yapması gerekir. İş Kanıtı (PoW) olarak tanımlanır. Bir blok üretmek için zorluk seviyesinin belirlenmesi gerekir. Bu sabit veya dinamik bir seviye olabilir. Zorluk derecesine bağlı olarak, tüm düğümlerden bir karma değer istenir. Madenciler bu karma değeri bulmaya çalışırlar. Bu hash'ı bulan ilk düğüm bloğu onaylar ve onu zincire ekler. Böylece madenci ödülü kazanır (Baliga, 2017). Bu algoritmada en önemli detay işlem gücü ve madenci sayısıdır. Madencinin ödülü kazanması için elde etmesi gereken hash özel bir değerdir.

Blokzincirin bir bloğunda bütünlüğü sağlamak amacıyla bir blok başlığı yer alır. Blok başlığında versiyon, önceki blok özeti, merkle root, zorluk hedefi, nonce ve zaman damgası bilgileri yer alır. Ayrıca transfer kayıt bilgilerinin tutulduğu her işlemin gönderici, alıcı, transfer miktarı bilgileri işlem bilgileri kısmında tutulur.

Versiyon: Blokta kullanılan Bitcoin'in protokol versiyon numarasını belirtir.

Merkle Root: Merkle ağacı yapısına bağlı olarak oluşturulan kök özet değeridir. İşlemlerin bütünlüğünü sağlamak ve değişikliklerin önüne geçmek amacıyla kullanılır.

Zorluk Hedefi: Bloğun özet değerinin üretilmesinin belirli bir zorluk seviyesinde olmasını sağlar. Özet değerinin en anlamlı bitlerinin bir kısmının tamamen sıfır olmasını gerektirir. Madencilik gücüne göre ayarlanır. Bitcoin'de maksimum hedef olarak belirlenen ve en düşük zorluğa karşılık gelen değerin zorluk değerine bölünmesiyle elde edilir.

Zaman Damgası: Unix formatına göre bloğun oluşturulma zamanını gösterir.

Nonce: Madencilerin bloğu onaylamak amacıyla sürekli deneme yaptıkları bir değerdir. Bu deneme yanılma işlemleri belirli sayıda sıfırdan oluşan özet değeri hesaplanıncaya kadar devam eder.

Bazı durumlarda, birden fazla düğüm aynı anda geçerli bir karma üretebilir. Bu düğümlerin her biri kendi bloğunu doğrular ve eşler arası ağ üzerinden yayınlar. Bu gibi durumlarda geçici bir çatal oluşur. Birbirine yakın olan düğümleri zincire blok eklemeye devam edebilir. Dolayısıyla çatallardaki blok sayısı artmaya devam eder ve protokol,

maksimum PoW'lu dalın (en uzun dal) blokzincirine dahil edilmesine ve diğerlerini iptal etmesine izin verir. Bu, tüm düğümler arasında tutarlılığı sağlar (Baliga, 2017).

Şekil 3.3. Blokzincirin bir bloğu, veri boyutları ve örnek veriler

PoW kullanarak oluşturulan blokzincirleri DDoS saldırılarına karşı %100'e yakın koruma sağlar. Bunun nedeni, sistemdeki düğümlerin işlem gücüdür. Bir DDoS saldırısının başarılı olması için, blokzincir sisteminin toplam işlem gücünden daha fazla güce sahip olması gerekir. PoW kullanan blokzincirlerde %51 saldırı etkili olabilir. Bu saldırının asıl amacı, sistemdeki işlem gücünün büyük bir bölümünü alarak sonraki blokları ekleme işlemini almaktır. Büyük blokzincir yapıları (Bitcoin, Ethereum vb.) için bu zor olabilir, ancak PoW kullanarak küçük blokzincirlerinin manipülasyonu mümkündür.

Şekil 3.3'de blokzincire ait örnek bir blok, blokta yer alan verilerin boyutu ve örnek veriler bulunmaktadır. Bloğun hash değerleri nonce değerine ve konsensüs kuralına bağlı

olarak üretilen madencilik işleminden geçmektedir ve en anlamlı basamaklar sıfırlardan oluşmaktadır.

PoW'un çalışma prensibi: Blokzincir yapısında bulunan özet (hash) algoritması ile güven yapısını sağlamaktadır. PoW fikir birliği için, bir düğümün bulmaca çözer gibi 0 ile 2^{32} (4 milyar) arasında değişen bir nonce değeri üretmesini sağlayarak bloğu onaylaması ve ağa dağıtması sağlanır. Üretilen nonce değeri ile istenilen zorlukta (sınırda) bir hash değeri karşılığının elde edilmesi amaçlanmaktadır. Olası hash değerlerinin sayısı $2^{256-zorluk\ biti\ sayısı}$ formülü ile hesaplanmaktadır.

Üretilmesi planlanan hash değeri zorluk biti sayısı ile belirlenmektedir. İstenen hash değeri 256 bitlik bir yapıya sahiptir. Zorluk biti sayısına göre oluşması planlanan örnek hash ve örnek nonce değerleri Tablo 3.4'deki gibidir. Nonce ile hash arasında matematiksel olarak herhangi bir bağlantı yoktur.

Tablo 3.4. Zorluk biti sayısına göre elde edilmesi muhtemel hash ve nonce değerleri.

Zorluk biti sayısı	Hash (Üretilmesi istenilen aralık)	Nonce
4 bit	**0**a51c867fa6ab52d99e58f240b73d7a8ea8317acbc4d7987e019617234a31eae	1563
8 bit	**00**c3c867fa6ab52d99e58f240b73d7a8ea8317acbc4d7987e019617234a31eae	16634
16 bit	**0000**c867fa6ab52d99e58f240b73d7a8ea8317acbc4d7987e019617234a31eae	958
32 bit	**00000**867fa6ab52d99e58f240b73d7a8ea8317acbc4d7987e019617234a31eae	2672

Tablo 3.4 incelendiğinde, zorluk biti sayısı arttıkça üretilmesi istenen hash değerinin daha küçük olması gerektiği görülmektedir. Hash değerinin üretildiği alan sınırlandırıldığından dolayı blok onayı için geçen sürenin her 4 bitte 2^4 oranında arttığı söylenebilir. Cihan hash üretim gücü 500.000 hash/saniye ve zorluk biti sayısı 32 olduğu durumda;

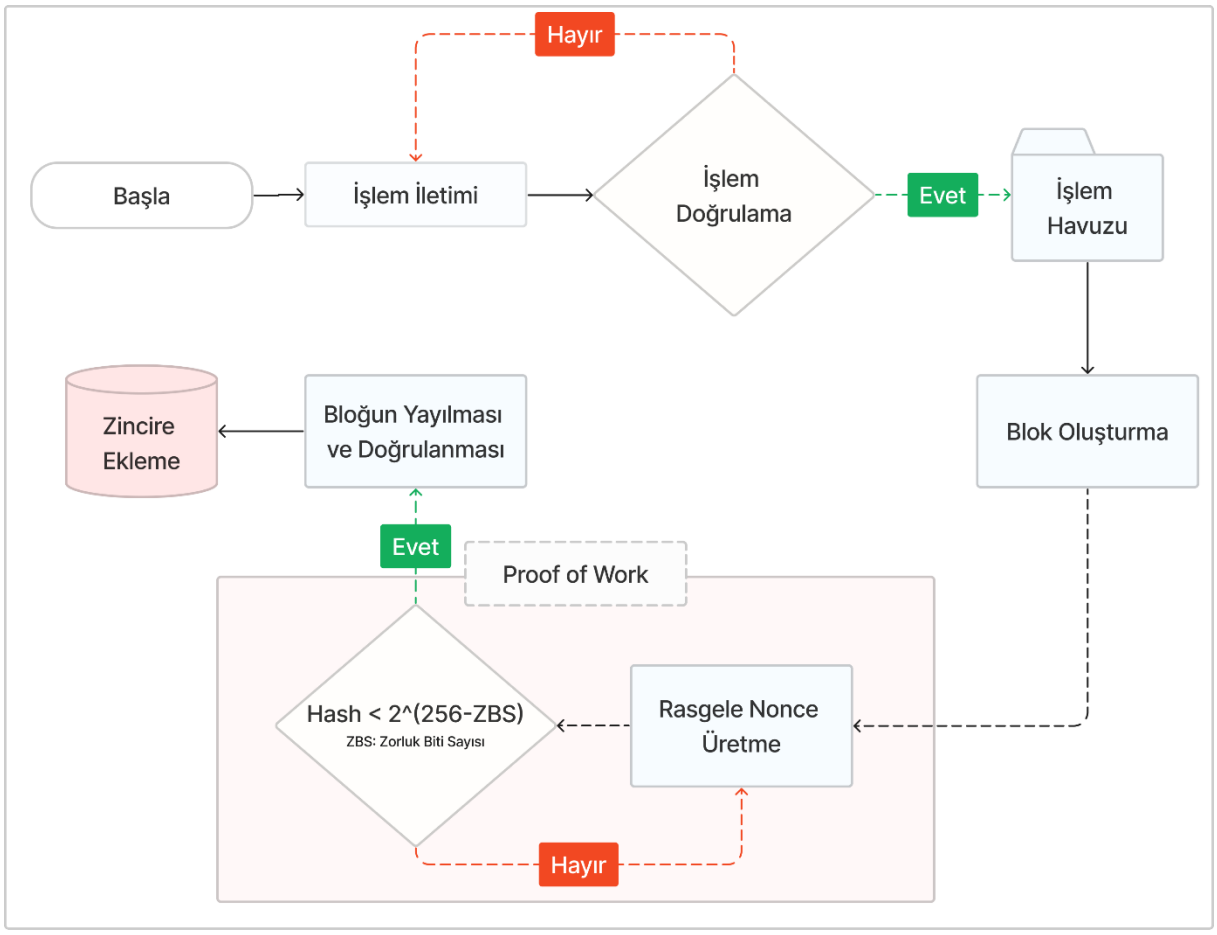
$$\text{Olası hash değerlerinin sayısı} = 2^{256-zorluk\ biti\ sayısı} = 2^{256-32} = 1{,}15e+77 \quad (3.1)$$

$$\text{İstenmeyen hash sayısı} = 2^{256} - 1{,}15e+77 = 4.294.967.296\ (yaklaşık\ 4\ milyar) \quad (3.2)$$

$$\text{Maks. Süre} = \text{İstenmeyen hash sayısı/Hash gücü} = \frac{4 \times 10^9}{500.000} = 133{,}33\ dk. \quad (3.3)$$

Burada zorluk biti sayısı 32 seçilen bir blokzincir PoW fikir birliğinde, bir cihazın saniyede 500.000 adet nonce değeri kullanarak hash ürettiği varsayıldığında bulunması istenen hash değerlerinin aralığı Eşitlik 3.1'de hesaplanan 1,15e+77 gibi çok büyük bir

değerdir. Eşitlik 3.2'ye göre kalan hash değerlerinin sayısı yaklaşık 4 milyardır. Bulunması beklenen hash değerleri aralığı çok büyük olsa da hash üretimi tamamen rasgele bir süreç olduğundan dolayı bu durum beklenildiği kadar kolay olmamaktadır. Saniyede 500.000 hash gücüne sahip bir cihazla yapılan deneyde, 32 bit zorluk ile bir hash üretmek için geçen süreler sırasıyla 183, 22, 300, 133, 1288 saniye gibi birbirinden oldukça farklı verilerdir. Eşitlik 3.3'de tüm durumlar denendiğinde oluşacak maksimum süre verilmiştir. Bu zorlukta maksimum blok doğrulama süresi teorik olarak yaklaşık 133 dakika sürebilmektedir.



Şekil 3.4. Hash üretimi için bir konsensüs süreci.

Bitcoin blokzincirinde, üretilen hash değerinin ilk 21 dijitinin 0 olması (16'lık sistemde) gerekmektedir. Yani zorluk biti sayısı $21 \times 4 = 84$ 'tür ve kişisel bir bilgisayar ile üretimi (saatlerce sürebilir) oldukça zordur. Hash, rasgele üretilen nonce değerine bağlı olduğundan istenilen aralıkta bir değer her defasında bulunamayabilmektedir. Böyle bir durumda blokta bulunan bazı işlemler değiştirilir ve tekrar hash üretilmeye başlanır. İstenilen aralıkta hash bulununcaya kadar bu süreç devam eder. Şekil 3.4'de bulunan diyagram bu sürecin bir gösterimidir.

3.4.2.Proof of stake (PoS)

Bu yaklaşımda blokzincir ağı üzerinde sahip olunan pay; blok üretim ve geçerlilik onayı ile doğru orantılıdır. Yani en fazla paya sahip olan düğüm zincirde en çok söz hakkına sahip olmaktadır. Bunu önlemek için en çok paya sahip olana en yüksek şansı vererek rastlantısal fonksiyona dayalı blok üretimi veya bir problem belirlenip bu problemin düğümler tarafından çözülmesi istenerek (PoW'a benzer) blok üretimi gerçekleştirilebilir (X. Zheng vd., 2018).

Temel olarak, madencinin elindeki token miktarının önemli olduğu bir algoritmadır. Madencilerin blok ekleme şansı, sahip oldukları token miktarıyla doğrudan orantılıdır.

Kripto para birimlerinin enerji tüketimine bağlı olduğu ve bu nedenle bu tür şebekelerin önemli maliyet gerektirdiği görülmektedir. Merkezi olmayan bir kripto para birimine sahip olmak için enerji tüketimi gerektirmeyen çözümler sunmaya çalışmanın önemi vurgulanmaktadır. Bir insan bu sistemde çok fazla şifreleme parazitine sahipse, blokzinciri ağına saldırma ihtimalinin daha düşük olduğu düşünülmektedir (Zheng vd., 2017).

Bu yaklaşımda, blokzinciri ağının payı doğrudan blok üretimi ve doğrulaması ile orantılıdır. Blokzincirine blok ekleme hakkı en fazla paya sahip olan düğümlere verilir. Bunu önlemek için, rasgele işlev veya bir soruna dayanan blok üretimi veya en yüksek paya sahip olanlara en yüksek şansı vererek veya bu sorunu düğümler (çözme işine benzer) ile çözmeyi isteyenlere en yüksek şansı vererek karar verilebilir (Garzik & Donnelly, 2018).

3.4.3.Delegated proof of stake (DPoS)

DPoS, fikir birliğini sağlamak için paydaşların onay gücüne dayalı bir sistemdir. Blok aralıkları ve işlem boyutu seçilen delege veya delegeler tarafından ayarlanabilir. Sistemdeki paydaşlar blok üretmek için herhangi bir sayıda tanık seçebilir. Tanıkların her biri bir blok ürettiğinde, kripto parası bir ödül olarak ödenir. Aldıkları ücretler menfaat sahipleri tarafından seçilen delegeler tarafından belirlenir. Bir tanık blok oluşturmadığında, ödül alamaz. Adil ve demokratik bir çözüm sağlamak için belli bir merkeziyet derecesini korur. Sistemdeki herkes bir temsilci temsilcisi olabileceğinden, seçim süreci güvenli bir ortamda gerçekleşir. Bu sistem, Daniel Larimer tarafından saniyede 100.000 işlem yapabilecek şekilde tasarlanmıştır. Daha az enerji tüketen ve kısa sürede daha fazla çalışmaya izin veren bir sistemdir. Ayrıca oylama sistemi ile daha güvenli bir yapı sağlamaktadır (Bitshares, 2019).

DPoS ünvan, popülerlik veya para gücüne bağlı olduğundan, yeni üyeler çok fazla gelir elde edemiyor. Bu sistemin tamamen güce bağımlı olmasını önlemek için kısmi

rastgelelik sağlayan yeni çalışmalar yürütülmektedir. DPoS'un avantajları ve dezavantajları:

- Enerji tasarrufu sağlar.
- İşlem hacimleri artar.
- Daha hızlı doğrulama süreleri sağlar.
- Merkezi değildir.
- Sürekli gelişmeyi destekler.
- Seçilen delegeler kötü amaçlı olabilir ve ağı yavaşlatabilir.
- Üyeler arasında gruplama olabilir.

3.4.4.Bizans hata toleransı (BFT)

Blokzincirde bulunan dağıtılmış yapı çok sayıda taraf arasında güven sağladığı için Bizans fikir birliği olarak kabul edilir. Bizans fikir birliği, blokzincirde kullanılan ve bu yapıya daha uyumlu hale getirilmesi amacıyla üzerinde çalışmalar yapılan bir algoritmadır. Bu algoritmanın blokzincirdeki yapısı ispat bazlı konsensüs ve oy bazlı konsensüs olarak ikiye ayrılabilir. İspat bazlı fikir birliğinde, yeterli kanıta sahip olan bir düğümün blokzincire yeni blok ekleme ve kripto para cinsinden ödül kazanmasını sağlamaktadır. Bitcoin'de, madencilerin zorlu bir bulmacayı çözmesi için rekabet etmesi gerekmektedir. Bu durum büyük miktarda küresel kaynak tüketimine sebep olmaktadır. PoW'un kaynak tüketimine kıyasla Bitcoin blokzincirinde saniyede 7 işlem (transaction per second-TPS) yapılabilmektedir. PoS ise hisse değerlerine bağlı bir zorluk belirlemektedir. PoS'un temel yapısı, en fazla kripto paraya sahip olan düğümlerin blok üretimi yapabilmek için daha fazla şansa sahip olmasını sağlamaktadır. Sahip olunan hisse, oylama kanıtı ile ilişkilendirilebilir (Lei vd., 2018).

3.4.5.Pratik bizans hata toleransı (PBFT)

HyperLedger Fabric tarafından kullanılan PBFT, Bizans hatalarını tolere eden bir algoritmadır. BFT'nin geliştirilmiş bir sürümü olan PBFT, Bizans generalleri problemi için yüksek hızlı işleme sağlar. Bu algoritmada, tüm düğümlerin 2/3'si önerilen bloğu onaylamayı kabul ederse, fikir birliğine varılır. Bu durumda, kötü niyetli düğümlerin oranı tüm düğümlerin 1/3'ini aşamaz. Bu nedenle, PBFT'deki her düğümün ağ tarafından bilinmesi gerekir (X. Zheng vd., 2018).

3.4.6.Federasyon Bizans Anlaşması (FBA)

Ripple (Schwartz vd., 2014) ve Stellar (Mazieres, 2015), Bizans hata tolerans

konsensüs modelinin varyasyonlarından oluşan finansal sistemlerde ve ödeme sistemlerinde kullanılan blokzinciri tabanlı platformlardır. Günümüz altyapısında sınır ötesi işlemler günler boyu sürebilmektedir. Bunun aksine Ripple ve Stellar gibi blokzinciri ödeme protokolleri kullanılarak bu işlemler birkaç saniyede gerçekleştirilebilmektedir.

Bu tür sistemlerde son kullanıcılar, finansal kurumlar (ağ geçidi görevi gören) ve pazar üreticileri olan kullanıcılar veya finansal kurumlardır. Son kullanıcılar, blokzincirin istemci yazılımını kullanarak kripto para ile ödeme işlemini gerçekleştirir. Bu sırada kullandığı ağ geçitlerine ödeme sırasında güvenmek zorundadır. Ağ geçitleri, sanal banka görevi gören yazılımlardır. Kullanıcıların fonlarını tutar ve paralarının Stellar veya Ripple ağında eşdeğer kripto para karşılığını oluşturur. Blokzincirde bulunan hesap bakiyelerine bakılarak gerçekleştirilen ödeme işlemleri sırayla tüm düğümler tarafından doğrulanır. Kullanıcının gönderdiği ve aldığı kripto paralar bir denge oluşturur. Her blok onayında denge ayarlaması yapılmaktadır.

3.5.Hyperledger Fabric Blokzincir Şeması

Hyperledger, kurumsal düzeyde blokzincir dağıtımları için bir dizi geliştirme çatısı, araçlar ve kütüphane geliştirmeye odaklanan, Linux Foundation tarafından oluşturulmuş açık kaynaklı bir topluluktur. Amaç küresel bir iş birliği sağlamaktır. Finans, bankacılık, nesnelerin interneti, tedarik zincirleri, üretim ve teknoloji liderlerini içerir. Tekniksel yönetişim ve açık iş birliği kapsamında oluşturulmuştur. Ayrıca hizmet ve çözüm sağlayıcıları, bireysel geliştiriciler, devlet kuruluşları, kurumsal üyeler ve son kullanıcılar hyperledger teknolojilerinin geliştirilmesine ve tanıtımına katkıda bulunabilmektedir. Linux Vakfı genellikle uyguladığı prensiplere benzer şekilde; projeleri barındırmak (hosting) için Hyperledger da modüler bir yaklaşıma sahiptir.

Temelde, Hyperledger bir kuruluş, bir kripto para birimi ağı veya bir blokzinciri sistemi değildir. Bitcoin gibi bir kripto para birimini desteklemez, ancak endüstriyel kullanım amacıyla çeşitli blokzincir tabanlı sistemler ve uygulamalar geliştirmek için gerekli altyapı ve standartları sağlayarak çalışmaktadır. Hyperledger, çeşitli bireysel blokzinciri tabanlı proje ve araçların şemsiyesi altında çalıştığı bir merkez olarak tanımlanabilir.

Ölçeklenebilir ve esnek olması sayesinde finans, tedarik zinciri, sağlık hizmetleri, endüstriyel hizmetler vb. birçok alanda kullanılabilmektedir. Hyperledger Fabric ile ilgili önemli terminoloji aşağıda verilmiştir. Hyperledger'da Chaincode, Channel, Ledger, World State, Identity gibi belli başlı bazı terimler mevcuttur (Steemit, 2016).

Chaincode: Bilgisayar kodu ile oluşturulan ve kullanıcıların blokzincir ile etkileşime girmesini sağlayan içerik, mülkiyet, paylaşım veya değerli şeylerin alışverişini kolaylaştırmayı sağlayan bir yapıdır.

Channel: Bir Hyperledger Fabric kanalı, özel ve gizli işlemlerin gerçekleştirilmesi amacıyla iki veya daha fazla belirli ağ üyesi arasındaki özel bir iletişim alt ağıdır (subnet). Bir kanal, üyeler (kuruluşlar), üye başına sunucu eşleri, paylaşılan defter, zincir kodu uygulamaları ve sipariş hizmeti düğümleri tarafından tanımlanır. Ağdaki her işlem, her tarafın kimliğinin doğrulanması ve o kanalda işlem yapmak için yetkilendirilmesi gereken bir kanalda yürütülür. Bir kanala katılan her eşin (peer), kendi kanalındaki eşlerine ve hizmetlerine kimliğini doğrulayan üyelik hizmetleri sağlayıcısı (MSP – Membership Service Provider) tarafından verilen kendi kimliği vardır.

Ledger: Kayıtların tutulduğu, sondan eklemeye izin veren ve değiştirilemez kayıt defteridir.

World State: Blokzincirdeki herhangi bir iş nesnesinin niteliklerinin mevcut değerini benzersiz (unique) bir defter durumu olarak tutar. Programlar genellikle bir nesnenin mevcut değerini gerektirdiği içi bu durum önemlidir; aksi takdirde bir nesnenin mevcut değerini hesaplamak için tüm blokzincirini geçmek ve hesap yaparak ilerlemek gerekmektedir.

Identity: Ağdaki tüm kişilerin eşsiz kimlikleri mevcuttur. İşlemlerin güvenliği, doğruluğu ve kimlik doğrulaması için dijit al sertifikalar, şifreleme ve dijital imza gibi teknolojiler kullanılması önemlidir.

3.5.1.Hyperledger Fabric ağ yapısı

Basit bir Hyperledger ağı örneği Şekil 3.5'te görülmektedir. Burada O1, O2 ve O3 organizasyonları ifade etmektedir. Üç kuruluş Konfigürasyon adlı bir yapılandırmaya sahiptir. Bu yapılandırma içerisinde kuruluşların tanımını ve kanalda oynayacağı rolleri belirleyen politikaları içermektedir. Bu organizasyonlar ağı ortak olarak kuracaktır.

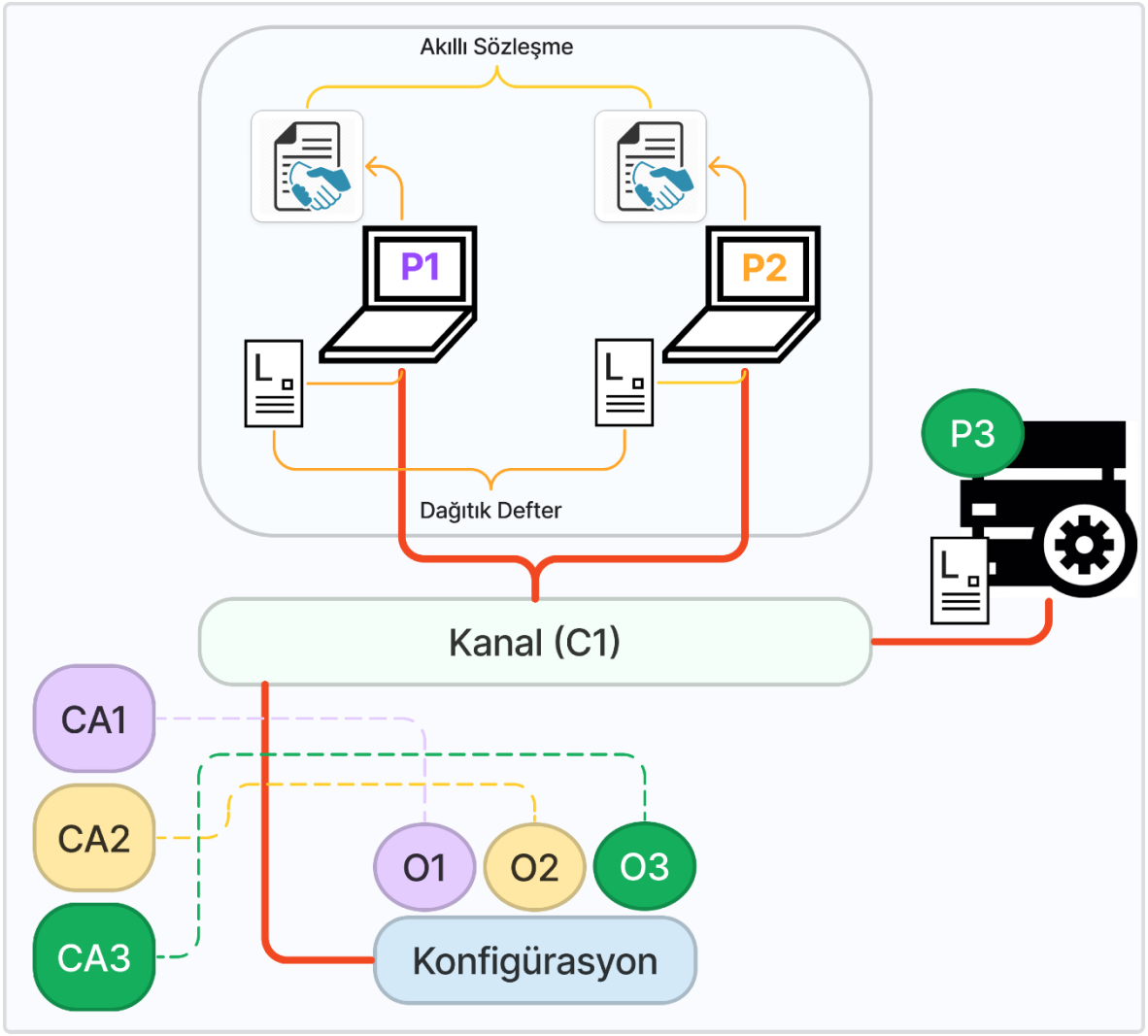
Hyperledger Fabric'te bir düğümün bir kanala katılmak için bazı adımları uygulaması gerekmektedir. Bu adımlar;

1. Konfigürasyon dosyası hazırlama
2. Kanal yapısını (yapılandırmasını) oluşturma
3. Genel anahtar ve sertifika işlemleri
4. Katılma isteği yollama ve onay alma

5. Kanala katılma bloğu alınması ve onaylanması

P1 ve P2 eşleri sırasıyla O1 ve O2'ye ait eşlerdir. P3 ise kanalın düzenleme, yönetme (ordering) servisidir. İşlem kayıtlarının tutulduğu defter olan L'nin kopyası P1, P2 ve P3 düğümlerinin tamamında bulunmaktadır. O1 ve O2 kanalla (C1) etkileşime girmek için bir uygulamaya ihtiyaç duymaktadır.

O1, O2 ve O3'ün sertifika yetkilisi (Certified Authority) sırasıyla CA1, CA2 ve CA3'tür. Sertifika yetkilisi organizasyon düğümleri, yönetici bilgileri, organizasyon tanımı ve uygulamalara gerekli sertifikaları oluşturur. P1, P2 ve P3 için x.509 sertifikaları CA'lar tarafından oluşturulur.



Şekil 3.5. Hyperledger Fabric ağ yapısı.

P3 düğümü yönetici (orderer) olarak işlemleri ve blokları düzenli bir şekilde sıralar. Ağdaki diğer düğümler arasında konsensüs sağlamak için merkezi otorite görevi yapar. İşlem siparişi, konsensüs sağlama, blok dağıtımı ve güvenlik sağlama işlemleri P3 düğümünün

görevidir. P1 ve P2 düğümlerinde aynı akıllı sözleşmenin kurulu olması gerekmektedir, fakat P3 düğümünde akıllı sözleşmenin kurulu olmasına gerek yoktur. Bunun sebebi P3'ün yönetici düğüm olmasıdır ve işlem önerisi yapmamasıdır. Akıllı sözleşme P1 ve P2 düğümlerine dağıtıldıktan sonra, istemci uygulamaları aracılığıyla Fabric Gateway hizmeti sayesinde akıllı sözleşmedeki işlemleri yürütebilir. Ek olarak, bir ağda birden fazla akıllı sözleşme veya birden fazla kanal yer alabilir. Konfigürasyonda hangi organizasyonlar birlikte yer alırsa oluşturulan kanalda bu organizasyonlara ait düğümler yer alır. Ayrıca bir düğüm birden fazla kanalda yer alabilir.

Hyperledger Fabric geliştirme çatısı ile Bitcoin ve Ethereum arasındaki farklar Tablo 3.5'de görülmektedir. Tabloya göre Bitcoin ve Ethereum kriptoparaya dayalı, anonim sistemler iken HLF değildir. Bitcoin ve Ethereum'da ağa katılım izinsiz, konsensüs protokolü sadece PoW ve farklı bir amaç için sistemin dizaynı söz konusu değildir. HLF'de ağa katılım izinli, modülerlik konusunda son derece başarılı ve konsensüs protokolleri çeşitlilik göstermektedir. Akıllı sözleşme açısından ise Ethereum'da "smart contract", HLF'de ise benzer şekilde "chaincode" yapısı mevcuttur (HyperledgerFabric, 2015).

Tablo 3.5. Bitcoin, Ethereum ve Hyperledger blokzincirler şemalarının özellikleri

	Bitcoin	**Ethereum**	**Hyperledger Fabric**
Kriptoparaya dayalı	Evet	Evet	Hayır
İzinli	Hayır	Hayır	Evet
Anonimlik	Evet	Evet	Hayır
Denetlenebilir	Evet	Evet	Evet
Değiştirilemez defter	Evet	Evet	Evet
Modülerlik	Hayır	Hayır	Evet
Akıllı sözleşme	Hayır	Evet	Evet
Konsensüs protokolü	PoW	PoW, PoS	RAFT, Kafka, Solo, …

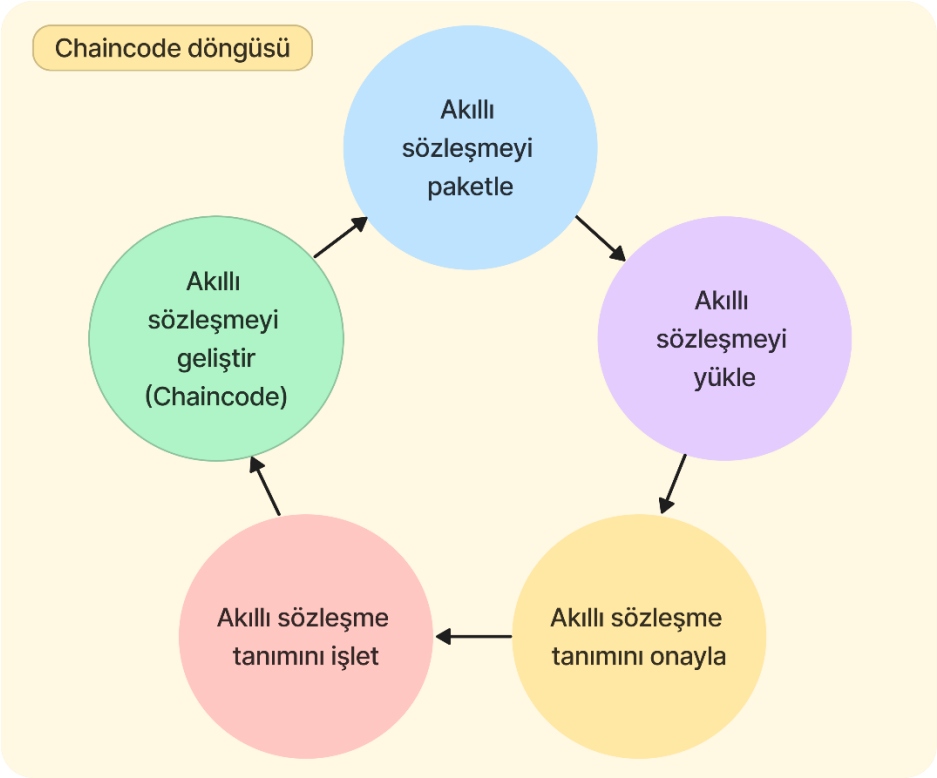
3.5.2.Chaincode yaşam döngüsü

Chaincode yaşam döngüsü ve bu döngüde bulunan aşamaların önemli bir kısmı ilerleyen bölümlerde açıklanmıştır. Ayrıca geliştirilen uygulamaya ait bilgiler de bu bölümde verilmiştir.

Şekil 3.6'da görülen chaincode yaşam döngüsü sırasıyla aşağıdaki adımlardan oluşmaktadır.

- Akılı sözleşme geliştirme

- Akıllı sözleşmeyi paketleme
- Akıllı sözleşmeyi eşlere yükleme
- Akıllı sözleşme tanımını onaylama
- Akıllı sözleşme tanımını kaydetme
- Akıllı sözleşme fonksiyonlarına erişim



Şekil 3.6. Chaincode yaşam döngüsü

3.5.2.1.Akıllı sözleşme geliştirme

Chaincode geliştirme aşamasında yapılması gereken birçok işlem adımı vardır. Geliştirilecek chaincode'un yapısı belirlenmeden geliştirmeye doğrudan başlanamaz. Sırasıyla;

- Mevcut olan gereksinimler ve ağ yapısı belirlenmelidir.
- Ağdaki kuruluş/kurum (organization) sayısı belirlenmelidir.
- Geliştiricinin dışında nasıl bir işlevsellik beklendiği belirlenmelidir.

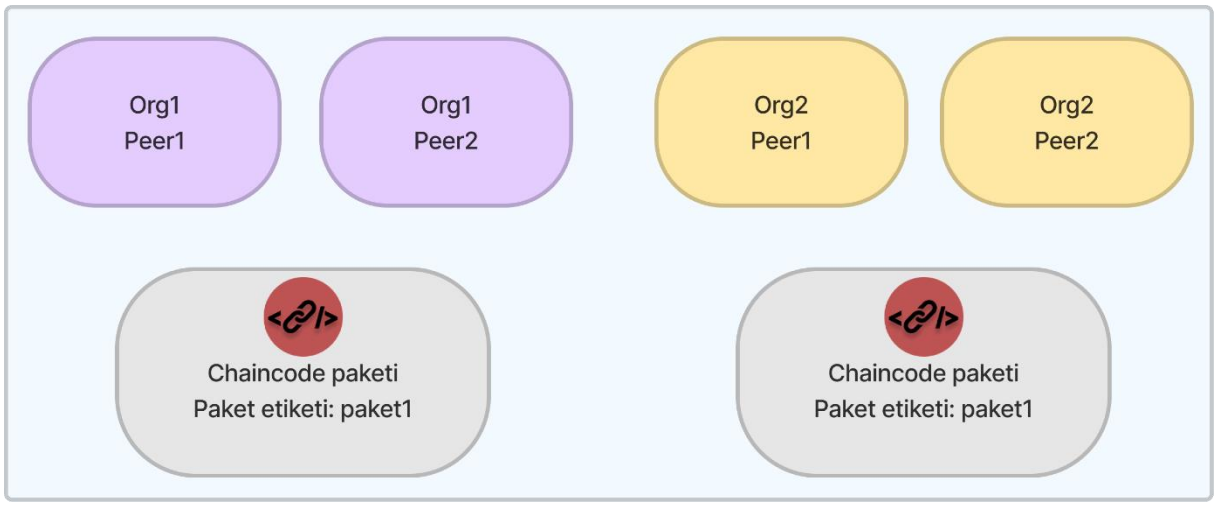
Gereksinimler listelenir ve geliştirici takım ile görüşme/tartışma yapılır. Geliştiriciden zincirin tasarımının bilgileri alınır. Gereksinimler ve tasarım belirlendikten sonra geliştirme aşamasına geçilir. Geliştirme işlemi için programlama dili seçilir. Chaincode popüler olan Go programlama dili ile geliştirilmekle birlikte Java ve Node.js kullanılarak da

geliştirebilmektedir.

3.5.2.2.Akıllı sözleşmeyi paketleme

Chaincode, ağda bulunan eşlere yüklenmeden önce bir tar dosyasında paketlenmelidir. Chaincode, Fabric eş ikili (binary) dosyaları, Node Fabric SDK'sı veya GNU tar gibi üçüncü taraf bir araç kullanarak paketlenebilmektedir. Bir chaincode paketi oluşturulduğunda, paketin kısa/öz ve insan tarafından okunabilir bir tanımını oluşturmak için bir chaincode paket etiketi de sağlamak gerekir (Steemit, 2016).

Chaincode'u paketlemek için üçüncü taraf bir araç kullanılırsa, ortaya çıkan dosyanın aşağıdaki biçimde olması gerekir. Fabric eş ikili dosyaları ve Fabric SDK'ları bu formatta otomatik olarak bir dosya oluşturur. Chaincode yazıldıktan sonra paketleme işlemleri yapılır, gerekli meta dosyaları, chaincode dili, kodun yolu ve paket etiketleri belirlenir.



Şekil 3.7. Chaincode paketleme yapısı

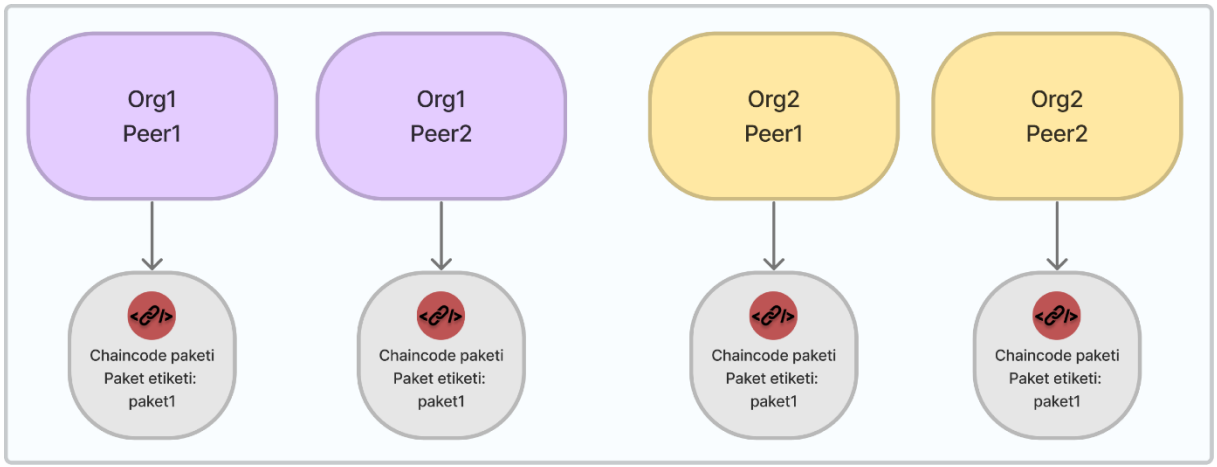
Şekil 3.7'de chaincode, Org1 ve Org2 tarafından ayrı ayrı paketlenir. Her iki kuruluş da adı ve sürümü kullanarak paketi tanımlamak için paket etiketi olarak paket1'i kullanır. Kuruluşların aynı ambalaj etiketini kullanması gerekli değildir.

3.5.2.3.Akıllı sözleşmeyi eşlere yükleme

Eşlere yükleme işleminde Command Line Interface kullanılmaktadır. CLI komut satırını kullanarak ağı (chaincode'u) dağıtmayı, toplamayı, ağ arşivi oluşturmayı veya Rest API oluşturmayı sağlar. İşlemleri yürütecek ve onaylayacak her eşe chaincode paketinin yüklenilmesi gerekmektedir. CLI veya başka bir SDK ile Peer Administrator (Eşlerin Yöneticisi) kullanarak bu adımı tamamlamak gerekir. Peer, chaincode yüklendikten sonra chaincode'u oluşturacak ve chaincode ile ilgili bir sorun varsa bir yapı hatası döndürecektir.

Kuruluşların bir chaincode'u yalnızca bir kez paketlemeleri ve ardından aynı paketi kuruluşlarına ait olan her eşe (peer) yüklemeleri önerilir. Bir kanal, her kuruluşun aynı chaincode'u çalıştırdığından emin olmak istiyorsa, bir kuruluş bir chaincode'u paketleyebilir ve bunu bant dışındaki diğer kanal üyelerine gönderebilir.

Başarılı bir yükleme komutu, geriye paketin özeti (hash) ile birleştirilmiş paket etiketi olan bir chaincode paket tanımlayıcısı döndürür. Bu paket tanımlayıcı, eşlere yüklenmiş olan bir zincir kodu paketini, kuruluş tarafından onaylanan bir zincir kodu tanımıyla ilişkilendirmek için kullanılır. Peer CLI kullanarak peer'a yüklenmiş paketleri sorgulayarak paket tanımlayıcısı elde edilebilir.



Şekil 3.8. Chaincode paketinin eşlere yüklenmesi.

Şekil 3.8'e göre, Org1 ve Org2'den bir eş yönetici, kanala katılan eşlere chaincode paketi paket1'i kurar. Chaincode paketinin yüklenmesi, zincir kodunu oluşturur ve paket1:hash (paket özeti) paket tanımlayıcısını oluşturur.

3.5.2.4.Akıllı sözleşmeyi onaylama

Chaincode, zincir kodu tanımıyla yönetilir. Kanal üyeleri bir zincir kodu tanımını onayladığında, onay, bir kuruluş tarafından kabul ettiği zincir kodu parametreleri üzerinde bir oylama işlevi görür. Bu onaylanmış organizasyon tanımları, kanal üyelerinin bir kanalda kullanılmadan önce bir zincir kodu üzerinde anlaşmalarına izin verir. Zincir kodu tanımı, kuruluşlar arasında tutarlı olması gereken aşağıdaki parametreleri içerir:

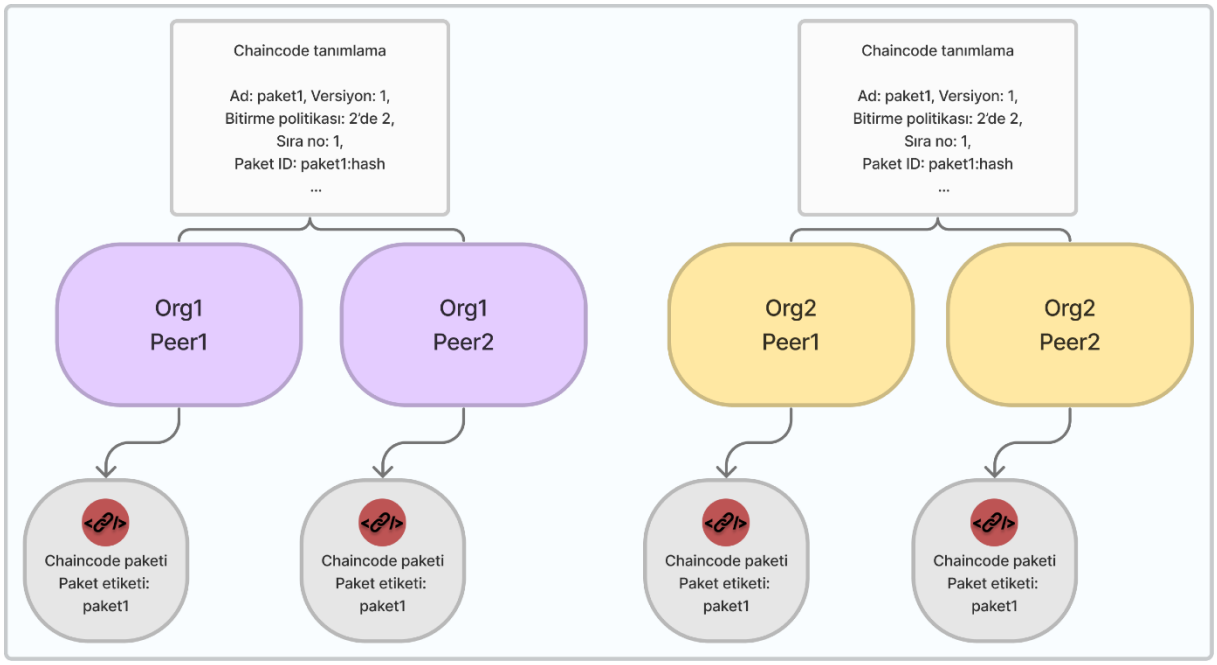
Ad: Chaincode çağrılırken uygulamaların kullanacağı addır.

Versiyon: Belirli bir zincir kod paketiyle ilişkili bir sürüm numarası veya değerdir.

Sıra (Sequence): Zincir kodunun tanımlanma sayısı. Bu değer bir tamsayıdır ve zincir kodu güncellemelerini takip etmek için kullanılır. Örneğin, bir zincir kodu tanımını ilk kurup onaylandığında, sıra numarası 1 olacaktır. Zincir kodu bir sonrakine yükseltildiğinde, sıra numarası 2'ye yükseltilecektir.

Onay Politikası (Endorsement Policy): Hangi kuruluşların işlem çıktısını yürütmesi ve doğrulaması gerektiğidir. Onay politikası, CLI'ye iletilen bir dize olarak ifade edilebilir veya kanal yapılandırmasında bir politikaya referans verebilir. Varsayılan olarak, onay politikası Kanal/Uygulama/Onay (Channel/Application/Endorsement) olarak ayarlanır ve bu varsayılan olarak kanaldaki kuruluşların çoğunun bir işlemi onaylamasını gerektirir.

Koleksiyon Yapılandırması (Collection Configuration): Chaincode ile ilişkilendirilmiş özel bir veri toplama tanım dosyasının yoludur.

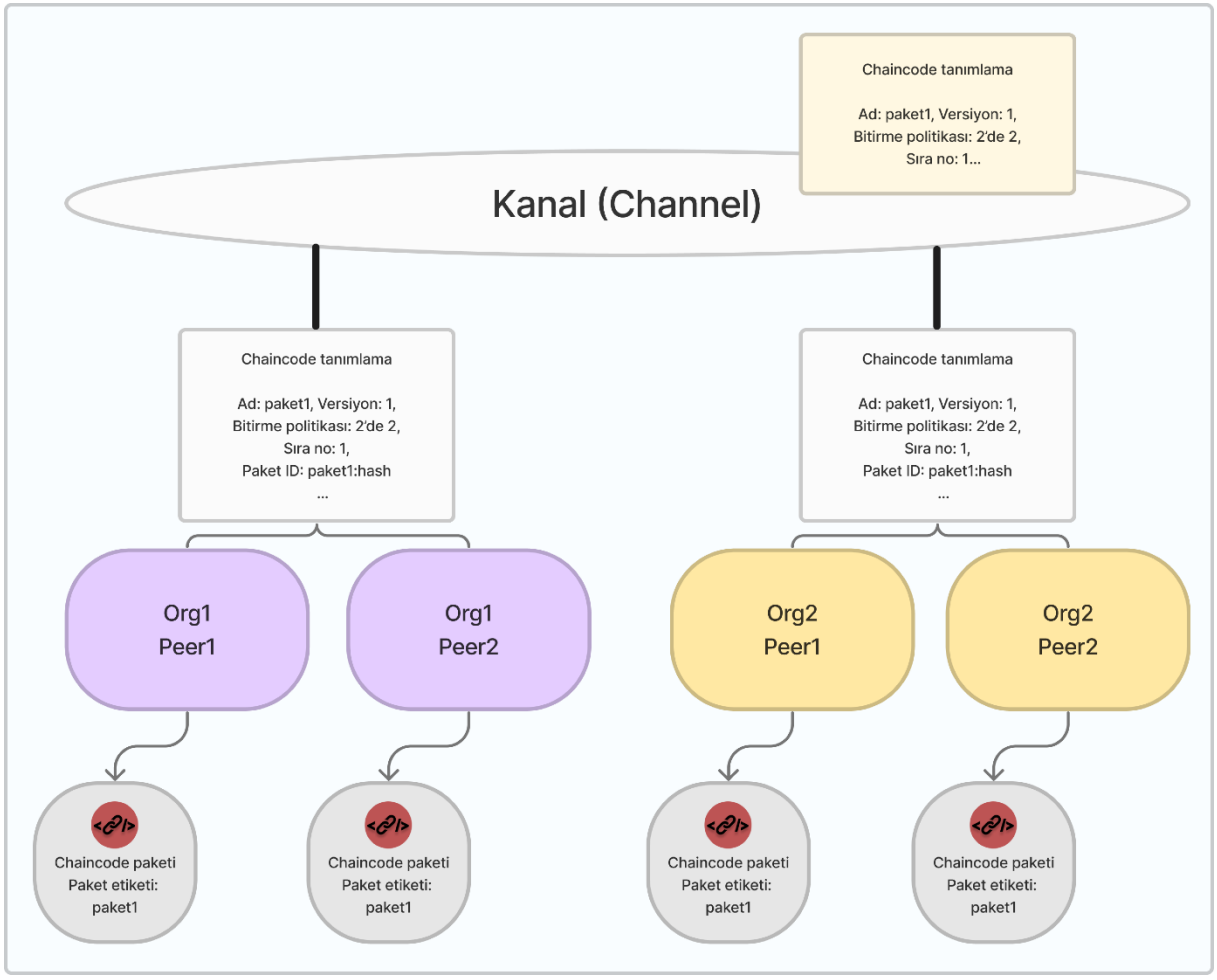


Şekil 3.9. Chaincode tanımını onaylama

ESCC/VSCC Eklentileri: İlgili zincir kodu tarafından kullanılacak özel bir onay veya doğrulama eklentisinin adıdır.

Başlatma (Initialization): Init işlevinin yürütülmesini istemek, zincir kodu başlatıldığında çalıştırılan mantığı uygulamaya izin verir. Örneğin bazı başlangıç durumlarını ayarlamak için buna ihtiyaç vardır. Bir zincir kodunun versiyonu her artırıldığında zincir kodunu başlatmak için, versiyonu artıran zincir kodu tanımının Init'in gerekli olduğunu varsayarak Init'i çağırması gerekecektir.

Şekil 3.9'da görüldüğü gibi, Org1 ve Org2'den bir organizasyon yöneticisi, organizasyonları için property'nin zincir kodu tanımını onaylar. Zincir kodu tanımı, diğer alanların yanı sıra zincir kodu adını, versiyonunu ve onay politikasını içerir. Her iki kuruluş da işlemleri desteklemek için zincir kodunu kullanacağından, her iki kuruluş için de onaylanan tanımların paket kimliğini içermesi gerekir.



Şekil 3.10. Chaincode tanımını kanala işleme

3.5.2.5.Akıllı sözleşmeyi kanala kaydetme

Yeterli sayıda kanal üyesi bir zincir kod tanımını onayladıktan sonra, bir organizasyon, tanımı kanala işleyebilir. Checkcommitreadiness komutu, eş CLI kullanarak kanala kaydetmeden önce hangi kanal üyelerinin bir tanımı onayladığına bağlı olarak zincir kodu tanımlamasının başarılı olup olmayacağını kontrol etmek için kullanılabilir. Taahhüt işlemi teklifi ilk olarak, kuruluşları için onaylanan chaincode tanımını sorgulayan ve kuruluşları onayladıysa tanımı onaylayan kanal üyelerinin eşlerine gönderilir. İşlem daha sonra sipariş hizmetine gönderilir ve daha sonra zincir kodu tanımını kanala işler. Onay (commit) tanımlama işleminin Organizasyon Yöneticisi olarak gönderilmesi gerekir.

Org1 veya Org2'den bir kuruluş yöneticisi, chaincode tanımını Şekil 3.10'daki gibi kanala işler. Kanaldaki tanım paket kimliğini içermez.

3.5.2.6.Akıllı sözleşmenin fonksiyonlarına erişim

Kanala chaincode tanımlandıktan sonra chaincode içerisindeki fonksiyonlara erişim CLI kullanılarak yapılmaktadır. Tüm bunlar yapılırken veya tekrarlanırken ağın açık olup olmadığı kontrol edilmelidir.

Bu aşamalardan sonra chaincode'u güncelleme işlemleri gelmektedir. Güncelleme işlemleri bazı değişiklikler olmakla birlikte yapılanların tekrarı niteliğindedir.

3.5.3.Blokzincir seçim kriterleri

Hyperledger Fabric ve Ethereum, merkezi olmayan uygulamalar ("dApps" olarak da bilinir) oluşturmak için kullanılabilen açık kaynaklı blokzincir platformlarıdır. Ancak, birkaç temel açıdan farklılık gösterirler. Genel olarak, Ethereum, merkezi olmayan bir platform ve akıllı sözleşmeler için destek gerektiren uygulamalar oluşturmak için daha uygundur; Hyperledger Fabric ise güvenlik, gizlilik ve performans için daha katı gereksinimlere sahip kurumsal düzeyde uygulamalar oluşturmak için daha uygundur. Ethereum veya Hyperledger Fabric platformlarından herhangi birini uygulama geliştirmek amacıyla tercih etmek için birçok parametre mevcuttur. Platform seçimi özel kullanım durumuna ve projenin gereksinimlerine bağlıdır. Hyperledger Fabric ve Ethereum arasında karar verirken dikkate alınması gereken faktörler aşağıda belirtilmiştir.

• *Amaç:* Ethereum, akıllı sözleşmeler yürütmek için merkezi olmayan bir platform olarak tasarlanırken Hyperledger Fabric, kurumsal düzeyde blokzincir uygulamaları oluşturmak için özel olarak geliştirilmiştir.

• *Konsensüs mekanizması:* Ethereum, madencilerin işlemleri doğrulamak ve bunları blokzincirine eklemek için karmaşık matematik problemlerini çözmesini içeren bir çalışma kanıtı olan PoW mutabakat mekanizması kullanır. Hyperledger Fabric, Kafka tabanlı mutabakat, solo mutabakat ve diğerleri gibi çeşitli algoritmaların kullanımına izin veren takılabilir bir mutabakat mekanizması kullanır.

• *Programlama dili:* Ethereum, akıllı sözleşmeler yazmak için Solidity dilini kullanırken Hyperledger Fabric, Go, Java ve JavaScript dahil olmak üzere çeşitli programlama dillerini desteklemektedir.

• *İzin Verme:* Hyperledger Fabric, ağ yöneticilerinin hangi kuruluşların ağa erişebileceğini ve hangi eylemleri gerçekleştirebileceğini kontrol etmesine olanak tanıyan daha ayrıntılı bir izin sistemine sahiptir. Ethereum ise herkesin katılabileceği tamamen açık bir ağdır.

• *Fikir birliği algoritması:* Hyperledger Fabric, takılabilir bir fikir birliği algoritması kullanır; bu, kullanım durumunuz için en uygun fikir birliği mekanizmasını seçebileceğiniz anlamına gelir. Öte yandan Ethereum, fikir birliği algoritması olarak iş ispatını kullanır.

• *İzinli ve izinsiz:* Hyperledger Fabric, izin verilen bir blokzincir platformudur, bu da ağa katılımın yetkili üyelerle sınırlı olduğu anlamına gelir. Ethereum ise izinsiz bir blokzincir platformudur, bu da herkesin ağa katılabileceği anlamına gelir.

• *Ölçeklenebilirlik:* Hyperledger Fabric, saniyede binlerce işlemi destekleyebilme özelliğiyle yüksek düzeyde ölçeklenebilir olacak şekilde tasarlanmıştır. Ethereum da oldukça ölçeklenebilir, ancak çalışma kanıtı konsensüs algoritması nedeniyle çok büyük ölçekli uygulamalar için uygun olmayabilir.

• *Kullanım durumları:* Hem Hyperledger Fabric hem de Ethereum, çok çeşitli kullanım durumları için uygundur, ancak belirli uygulama türleri için daha uygun olabilirler. Örneğin, Hyperledger Fabric genellikle kurumsal uygulamalar ve tedarik zinciri yönetimi için kullanılırken, Ethereum daha yaygın olarak merkezi olmayan finans ve tahmin pazarları için kullanılır.

• *Ağ mimarisi:* Hyperledger Fabric, mutabakat algoritması gibi farklı bileşenlerin takılmasına ve belirli uygulamaların ihtiyaçlarını karşılayacak şekilde özelleştirilmesine izin veren modüler bir ağ mimarisine sahiptir. Öte yandan Ethereum, tüm bileşenlerin sıkı bir şekilde entegre edildiği yekpare bir mimariye sahiptir.

• *Gizlilik:* Hyperledger Fabric, bir işlemin farklı bölümlerinin farklı taraflarca görülebilmesini sağlayan yerleşik mahremiyet ve mahremiyet desteğine sahiptir. Bu, farklı katılımcı gruplarının genel Fabric ağı içinde kendi özel alt ağlarına sahip olmalarına izin veren kanalların kullanılmasıyla elde edilir. Ethereum'un gizlilik ve mahremiyet için yerel desteği yoktur, ancak sıfır bilgi kanıtları veya diğer teknikler kullanarak gizliliği koruyan uygulamalar oluşturmak mümkündür.

• *Birlikte Çalışabilirlik:* Hyperledger Fabric, diğer sistem ve teknolojilerle kolayca entegre olmasını sağlayan modüler bir tasarıma sahiptir. Ayrıca, diğer dağıtılmış defter platformlarıyla birlikte çalışmasına izin veren takılabilir mutabakat algoritmalarını da

destekler. Öte yandan Ethereum, bağımsız bir platformdur ve diğer dağıtılmış defter platformlarıyla birlikte çalışabilirliği yerel olarak desteklemez.

Bu kriterler özet halinde Tablo 3.6'de görülmektedir.

Tablo 3.6. Hyperledger Fabric ve Ethereum arasındaki farklar

Özellikler	Hyperledger Fabric	Ethereum
Amaç	Kurumsal düzeyde blokzincir uygulamaları	Akıllı sözleşmeler yürütmek için merkezi olmayan platform
Konsensus mekanizması	Değiştirilebilir (Raft, Kafka, Solo)	PoW, PoS
Programlama Dili	Go, Java, JavaScript	Solidity
Saniye başına transaction (TPS)	Yapılandırmaya ve iş yüküne bağlıdır	Ethereum ana ağında 15'e kadar TPS
Blok boyutu	Yapılandırılabilir	Sınırsız (teoride, ancak gas limitiyle sınırlıdır)
Token	Herhangi bir token ile veya token olmadan kullanılabilir	Token olarak Ether'i (ETH) kullanır
İzin durumu	İzinli (kontrollü erişim)	İzinsiz (açık erişim)
Ölçeklenebilirlik	Yüksek (kanallar kullanılarak ulaşılabilir)	Sınırlı (PoW fikir birliği nedeniyle ölçeklendirme zorlukları)
Kullanım durumları	Tedarik zinciri, ticaret finansmanı, sağlık hizmetleri vb.	Merkezi olmayan finans (DeFi), tahmin piyasaları, oyun vb.
Ağ mimarisi	Özel (izinli) veya genel (izinsiz)	İzinsiz
Gizlilik	Özel veri toplamayı ve seçici onayı destekler	Yerleşik gizlilik özelliği yok (zk-SNARK'ları veya diğer çözümleri kullanabilir)
Birlikte çalışabilirlik	Konnektörler ve adaptörler aracılığıyla birlikte çalışabilirliği destekler	Sınırlı birlikte çalışabilirlik (özel köprüler veya protokoller gerektirir)
Akıllı sözleşme yürütme	Temel blokzincirinden ayrı	Temel blokzincir ile entegre
Dağıtım seçenekleri	Şirket içinde veya bulutta devreye alınabilir	Ethereum ana ağında veya bir test ağında konuşlandırılabilir

• *Akıllı sözleşme yürütme:* Hyperledger Fabric, akıllı sözleşmelerin bir işlem bağlamında yürütüldüğü bir çalışma zamanı modeli kullanır. Bu, bir akıllı sözleşmenin doğrudan çağrılamayacağı ve bir işlemin parçası olarak çağrılması gerektiği anlamına gelir. Öte yandan Ethereum, akıllı sözleşmelerin doğrudan çağrılabileceği ve özerk varlıklar olarak yürütülebileceği bir sanal makine modeli kullanır.

• *Dağıtım seçenekleri:* Hyperledger Fabric, şirket içi, bulut ve hibrit dağıtımlar dahil olmak üzere çeşitli dağıtım seçenekleri sunar. Öte yandan Ethereum, öncelikle bulutta veya bağımsız bir ağ olarak konuşlandırılır.

4.HYPERLEDGER AKILLI SÖZLEŞME UYGULAMASI (HFSecImg)

Hyperledger Projesi, kurumsal düzeyde açık kaynaklı dağıtılmış bir defter çalışma çatısı (ledger framework) ve kod tabanı oluşturmak için ortak bir girişimdir. Dağıtılmış defterler için sektörler arası açık standart bir platform tanımlayarak ve gerçekleştirerek, ticari işlemlerin küresel olarak yürütülme şeklini dönüştürebilen blokzinciri teknolojisini geliştirmeyi amaçlamaktadır. 2016 yılının başlarında Linux Vakfı'nın bir projesi olarak kurulan Hyperledger Projesi'nin şu anda 50'den fazla üyesi mevcuttur.

Blokzincirin yapısında barındırdığı dağıtık defter sayesinde değişmezlik ve veri doğrulama işlemlerinde oldukça güvenilir olduğu görülmektedir. Askeri, tıbbi görüntülerin veya gizliliği üst seviyede önemli olan görüntülerin/belgelerin saklanması bazı kurumlar açısından büyük önem arz etmektedir. Bu çalışmada veri güvenliğini ve gizliliğini sağlamak amacıyla secret sharing, kaotik sistem tabanlı görüntü şifreleme, dijital imza oluşturma ve özel blokzincir tabanlı doğrulamayı birlikte sağlayan bir şema önerilmektedir. Önerilen şema, orijinal görüntüye ulaşmak için sır paylaşım şeması ile paylaşılan anahtarı elde etmek amacıyla paydaşların tamamına veya belirli bir kısmına ihtiyaç duyan blokzincir tabanlı bir şemadır.

İlk aşamada görüntü şifreleme için kullanılacak olan gizli anahtar belirli sayıda kişiye dağıtılarak secret sharing ile gizliliği sağlanmaktadır. Görüntü şifreleme algoritmasının performans analizleri gizlilik ve güvenlik açısından incelenmektedir. Şifreli görüntü bulut sunucusunda saklanmaktadır. Şifreli görüntünün karma değeri gizli anahtarın paylaşıldığı paydaşlar tarafından RSA tabanlı dijital imza algoritması ile imzalanmaktadır. Doğrulama için gerekli olan imzalar ve karma değer Hyperledger Fabric tabanlı smart contract aracılığıyla dağıtık deftere kaydedilmektedir. Şifre çözme aşamasında smart contract vasıtasıyla sorgu yapılarak veri doğrulama işlemi gerçekleştirilmektedir.

Veri doğrulamada şifreli görüntünün karma değeri ve paydaşların imzaları kontrol edilmektedir. Doğrulama işlemi başarılı olduğunda paydaşların belirli bir kısmından elde edilen anahtarlar ile deşifreleme yapılarak orijinal görüntü elde edilmektedir. Gizli anahtarın ve orijinal görüntünün elde edilebilmesi için en az belirlenen eşik sayısı kadar paydaşın kendi anahtarını paylaşması gerekir. Bu sayede iki veya daha fazla paydaşın kendi aralarında oluşturdukları gizli bir bilginin saklanması ve izlenmesi amaçlanmaktadır. Sonuçlar ve incelemeler veri güvenliği ve gizliliği açısından önerilen çözümün önemli ölçüde başarılı olduğunu göstermektedir.

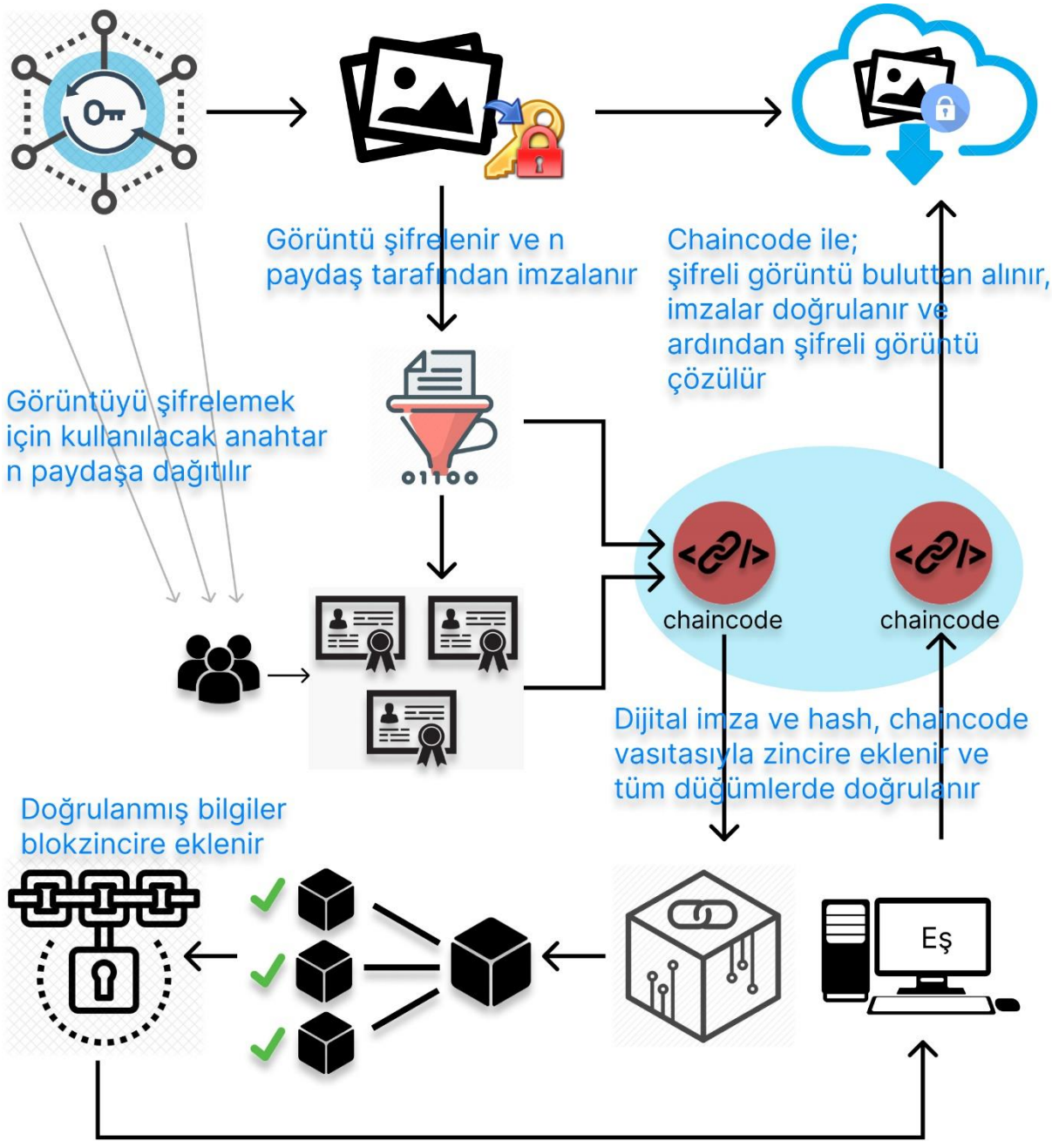
4.1.Şifreli Görüntünün Saklanması ve Doğrulanması Şeması

Bu tezde önerilen şemada anahtar belirleme, anahtar dağıtımı, görüntü şifreleme, dijital imza oluşturma, blokzincirde veri depolama, verileri doğrulama ve görüntüyü deşifre etme aşamalarını gerçekleştirerek şifreli görüntünün saklanması ve doğrulanması gerçekleştirilmiştir. Önerilen yöntemin blok şeması Şekil 4.1'de verilmiştir. Şekilden de takip edileceği üzere yöntemin işlem aşamaları adım adım aşağıda kısaca açıklanmıştır.

1. Görüntü şifreleme anahtarı (*secret*) belirle.
2. SSS kullanarak *secret*'i n parçaya ayır ve n adet paydaşa dağıt.
3. Kaotik sistem tabanlı görüntü şifrelemeyi anahtar (*secret*) ile yap.
4. Şifrelenen görüntünün SHA256 ile karma değerini hesapla.
5. Şifrelenen görüntünün karma değerini n adet paydaş ile ayrı ayrı imzala.
6. Doğrulama işlemlerinde kullanmak amacıyla şifrelenmiş görüntüleri açık/gizli bir bulut platformunda sakla.
7. Bilgileri doğrulamak amacıyla kullanılacak olan özel blokzinciri oluştur. Akıllı kontraktın kurallarını belirle.
8. Blokzincirde şifreli görüntü karma değeri, paydaş 1,2,…,n imzaları ve şifreleme zamanı bilgilerini sakla. (Ayrıca işlemi yapan birim, şahıs gibi ek bilgiler de eklenerek blokzincirde saklanır.)
9. Blokzincire eklenen bilgileri ağda bulunan tüm eşlere gönder ve onayla. (Konsensüs protokolü)
10. Deşifre edilmek istenen şifreli görüntüyü belirle ve blokzincir üzerinde sorgulama yaparak karma değeri bilgilerini ve paydaşların imza bilgilerini elde et.
11. Paydaşların dijital imzalarını doğrula ve hangi paydaşların görüntüyü imzaladığını tespit et.
12. Şifreleme anahtarına sahip olan paydaşların (n adet) içinden t adetinden secret'i oluşturmak için anahtar talep et ve secret'i elde et.
13. Elde edilen secret ile görüntüyü deşifre et.

Şekil 4.1'de görülen şemaya göre yapılan işlemler adım adım anlatılmıştır. Görüntü şifrelemede kullanılacak olan gizli anahtar secret sharing ile bir asal sayı olarak belirlenmektedir ve paydaşlara dağıtılmaktadır. Paydaşlardan hiçbiri orijinal anahtarla ilgili tahmin edilebilir bir bilgiye sahip olmamaktadır. Belirlenen anahtar kaotik sistem tabanlı görüntü şifrelemede başlangıç değerlerinde ve parametrelerde kullanılmaktadır. Şifreli

görüntünün boyutu fazla olduğundan blokzincirin dağıtık defterinde saklanması işlem hacmini ve yoğunluğunu oldukça artıracaktır. Bundan dolayı şifreli görüntü bulut sunucusunda saklanmaktadır. Şifreli görüntüye ait karma değer paydaşlar tarafından imzalanarak akıllı sözleşme aracılığıyla blokzincire eklenerek eşlere dağıtılmaktadır. Deşifre edilmek istenen görüntüye ait karma değeri blokzincire akıllı sözleşme vasıtasıyla yapılan bir sorgu ile gönderilerek paydaşların imzaları doğrulanır. İmzası doğrulanan ve başlangıçta belirlenen en az t sayıda paydaş arasından anahtar talep edilerek gizli anahtar elde edilir ve deşifreleme ile orijinal görüntü elde edilir.



Şekil 4.1. Önerilen şemanın blok yapısı

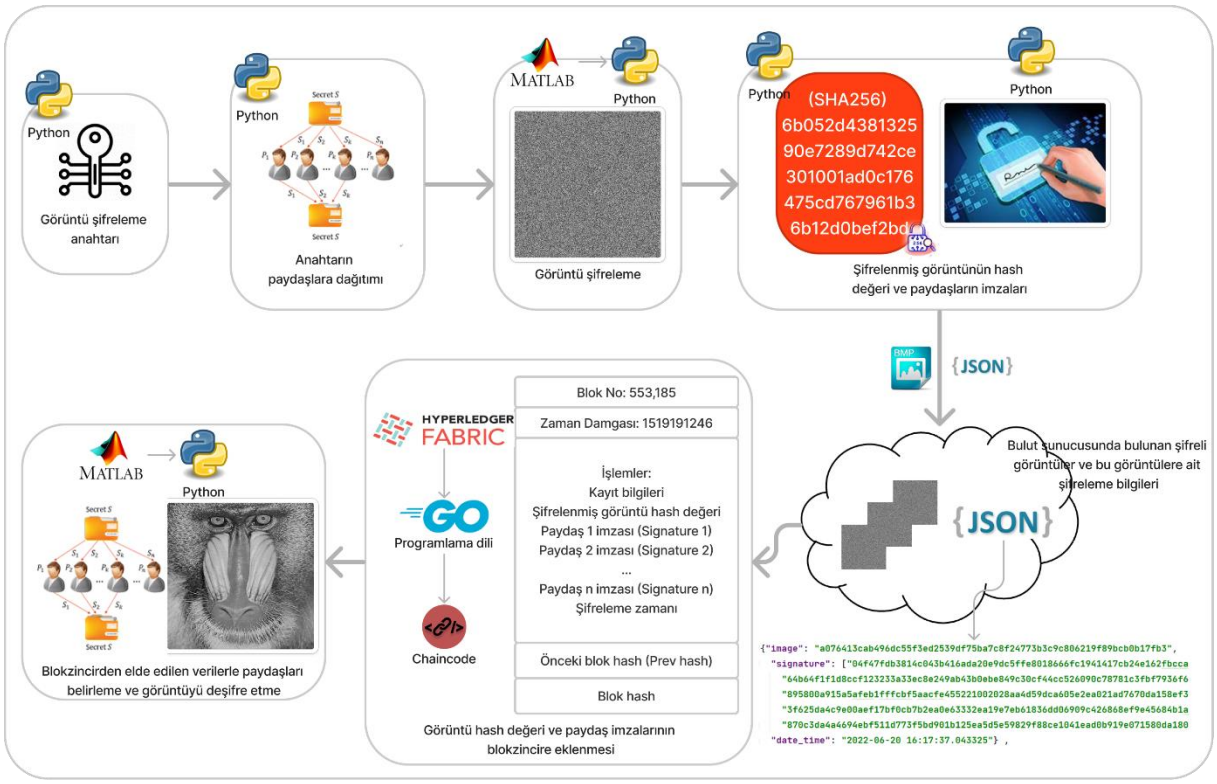
Uygulama geliştirilirken birçok ortam ve programlama dili entegre bir şekilde kullanılmıştır. Python programlama dili ile secret sharing, şifreli görüntünün karma değeri ve dijital imza uygulamaları gerçekleştirilmiştir. Matlab ortamında görüntü şifreleme gerçekleştirilmiştir. Python ile geliştirilen uygulamalar Matlab ortamında ilgili yerlerde çağrılarak şifrelenen görüntü bulut sunucusuna gönderilmektedir. Şifreli görüntüye ait bilgiler json formatına dönüştürülerek chaincode'un Go programlama dili ile yazılan ilgili fonksiyonuna gönderilmektedir. Chaincode ile transaction ve sorgular çalıştırılmaktadır. Hyperledger Fabric'te bulunan konsensüs algoritması ile onay mekanizmasından geçtikten sonra transaction eşlere dağıtılarak blokzincire eklenmektedir. Algoritma 1 anahtar paylaşımından transaction'un onayına kadar olan adımları göstermektedir. Algoritma 2 şifreli görüntünün bulut sunucusundan alınması, dijital imza doğrulama işlemlerinin yapılması ve görüntünün deşifre edilme adımlarını göstermektedir.

Algoritma 1 Gizli anahtar üretimi, görüntü şifreleme, dijital imza ve bulut depolama süreci

```
begin
  Secret_key = random(prime_number)
  [Deploy_to_participants] = Deploy_shares(Secret_key,t,n) // SSS
  Save_n_piece_of_key(Deploy_to_participants)
  Original_image = read(image)
  Cipher_image = CCM_encrypt(Original_image,Secret_key)
  Image_hash = SHA256(Cipher_image)
  for i=1 to n
    Digital_signatures[i] = RSA(Image_hash,Participant_private_key)
  end
  Save_to_cloud(Cipher_image)
  Send_transaction_value_to_chaincode(Digital_signatures,Image_hash)
  Execute_transaction_by_chaincode()
end
```

Algoritma 2 Veri doğrulama, anahtar yeniden üretimi ve görüntü deşifreleme süreci

```
begin
  Cipher_image = Import_image_from_cloud()
  Image_hash = SHA256(Cipher_image)
  // p[i] = digital signature of participant[i]
  [p] = Find_image_encryption_shareholders(Image_hash,Participant_public_key)
  if Chaincode_verification_query(Image_hash,[p]) is true
    for i = 1 to t (or any t<=n value)
      keys[i] = Piece_of_key(Image_hash,p[i])
    end
    Secret_key = Reconstruct_key(keys) // SSS
  end
  Decipher_image = CCM_decrypt(Cipher_image,Secret_key)
end
```



Şekil 4.2. Detaylı blok yapısı, programlama dilleri, platformlar ve veri tipleri

Şekil 4.1'de önerilen şemanın blok yapısına ek olarak Şekil 4.2'de kullanılan programlama dilleri, platformlar, veri saklama biçimleri detaylıca belirtilmiştir. Bazı bölümlerde Matlab ve Python hibrit olarak kullanılmıştır. Bunun yanı sıra Docker konteyner yapısı kullanılarak bilgiler Hyperledger Fabric blokzincirine Go programlama dili ile oluşturulan chaincode vasıtasıyla eklenmiştir.

Hyperledger Fabric geliştirme çatısında oluşturulan blokzincirin yapısı, depolanan bilgiler, eklenen bloklar vb. bilgileri görüntülemek amacıyla Hyperledger Explorer aracı konfigüre edilerek web ortamında Şekil 4.2'deki gibi bilgiler görüntülenmektedir. Ayrıca blok sayısı, işlem sayısı, düğüm sayısı, chaincode, ağda bulunan organizasyonlar, işlemlerin içeriği, hash bilgileri vb. bilgiler de simüle edilerek blokzincirin yapısı incelenebilmektedir. Görüntünün karma değeri, 5 adet katılımcının imzaları ve işlemin yapıldığı tarih bilgileri yer almaktadır.

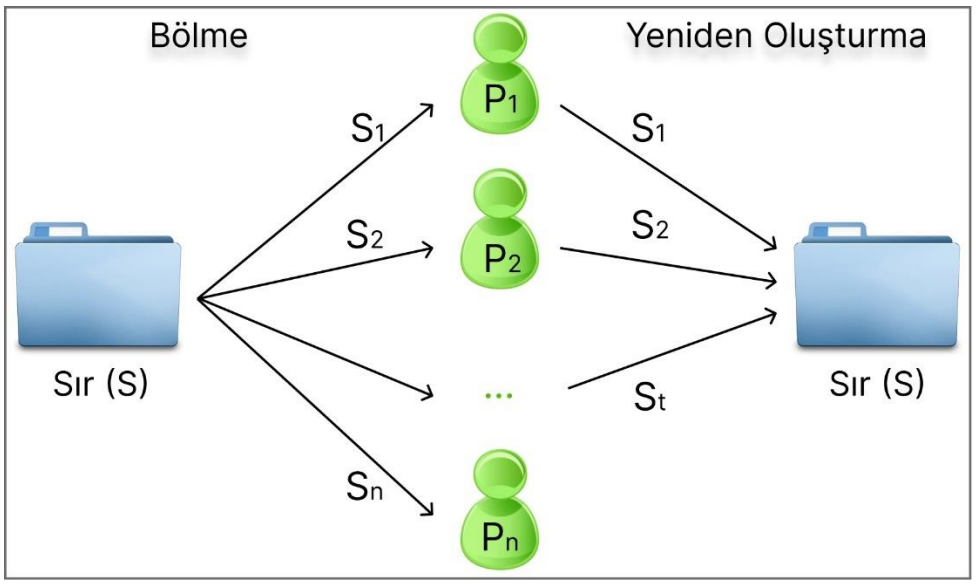
Şekil 4.3. Hyperledger Explorer ile işlem detayları gösterimi

Önerilen şemada kullanılan ve çalışmada üzerinde yoğunlaşılan teknolojilere ait teorik bilgiler ilerleyen bölümlerde anlatılmaktadır. Shamir'in eşik şeması, RSA dijital imza, Hyperledger Fabric ve akıllı kontrakt geliştirmek için kullanılan chaincode detaylı bir şekilde açıklanmaktadır.

4.2.Shamir'in (t,n) Eşik Şeması

Gizli paylaşım eşik şeması (t, n) bir gizli K'yı parçalara bölmek için kullanılır. n parçaya bölünen K paydaşlara (p1,p2,…,pn) dağıtılır, t veya daha fazla paydaş tarafından elde edilebilir. t'den az sayıda katılımcı sırrı elde edemez. Kullanılabilirlik ve güvenilirlik sorunlarını çözmek amacıyla sır paylaşımı yapılabilir. Şekil 4.4'de görülen Shamir'in sır

paylaşımına ait, sırrın dağıtım aşaması ve sırrın elde edilmesi ilerleyen aşamalarda anlatılmaktadır.



Şekil 4.4. Shamir'in sır paylaşım şeması

F herhangi bir alan, $(x_1, y_1),\ldots,(x_n, y_n)$ F^2'de x_i'ye karşılık n adet nokta olsun. Derecesi n-1'e eşit veya küçük olan bir P(x) polinomu için, i = 1,…,n durumunda $p(x_i) = y_i$ ise bu noktaların F üzerinde ara değerleri (enterpolasyonu) hesaplanabilir. Teorik sonuç aşağıdaki gibi hesaplanır.

Teorem: F herhangi bir alan ve $x_1,x_2,\ldots,x_n$ F'nin n farklı elemanı olsun. n noktayı enterpolasyon yapan derecesi n-1'e eşit veya küçük olan benzersiz tek (unique) polinom vardır.

Bu teoreme göre, Shamir'in (t, n) eşik şeması kabaca aşağıdaki gibidir. Bir F alanı seçilir. Sabit terimi K olan sır $K \in F$'dir. En fazla t – 1 dereceli bir P(x) polinomu belirlenir. Herhangi x_i'nin sıfır olmadığı $x_1,x_2, \ldots,x_n$ seçilir ve n adet katılımcının her birine i = 1, …,n olmak üzere $(x_i, P(x_i))$ dağıtılır. Bu sayede gizli K'yi elde etmek amacıyla n kişi içerisinden herhangi bir t kişi enterpolasyon ile P(x) polinomunu belirleyebilir. Buna ek olarak, sınırlı bir alan verilmediğinde, sonsuz bir alan verildiğinde ve kişi sayısı t'den az olduğunda, verilen noktaları enterpolasyon yapabilen sonsuz t dereceli polinomlar bulunur. Bu durumda doğru polinomu tespit etme olasılığı sıfırdır (Chum vd., 2018).

Chum ve arkadaşları, Shamir şemasının daha açık bir versiyonunu sunmak amacıyla büyük bir q asal sayısını seçerek Zq sonlu alanını kullanmaktadır. Düz metinler

ve şifreli metinler kümesine sonlu bir ölçü oluşturabilmek amacıyla, Shamir sonlu bir alan kullanır. t'den az kişinin bulunduğu durumlarda tüm sırlar eşit derecededir (Gil, 2020). Chum ve arkadaşlarının şeması aşağıdaki gibidir:

K sır olarak belirlenir. Zq sonlu alanında en fazla t – 1 dereceli P(x) polinomu (Eşitlik 3.1) üretilir. Dağıtıcı q > n olacak şekilde asal sayıyı seçerek polinomu oluşturur.

$$P(x) = a_0 + a_1 x_1 + \cdots + a_{t-1} x_{t-1} (\text{mod q}) \tag{4.1}$$

Burada sır a_0 = K'dır. Rastgele oluşturulan $a_1,\ldots,a_{t-1} \in Zq$ şeklinde belirlenir. Dağıtıcı bunların içinden n farklı değeri $x_i \in Zq - (0)$, i = 1,2,...,n herhangi bir kurala bağlı kalmaksızın istediği şekilde seçer. $x_1,x_2,\ldots,x_n$ değerleri açık bir alanda saklanır. Her i için x_i polinomda yerine koyularak $y_i = P(x_i)$ (mod q) hesaplanır. Dağıtıcı güvenli bir kanal kullanarak n tane katılımcı (p_i) ile sırasıyla y_i'yi paylaşır. Sır oluşturma ve dağıtım şeması bu şekildedir. Sırrı elde etme kısmı ilerleyen bölümde anlatılmaktadır.

t tane katılımcının (p_i) bir araya geldiği durumda i = 1, 2,…t için Eşitlik 4.2 ile sırrı elde etme işlemleri uygulanır.

$$y_1 = P(x_1) = a_0 + a_1 x_1 + \cdots + a_{t-1} x_1^{t-1} (mod\ q),$$

$$y_2 = P(x_2) = a_0 + a_1 x_2 + \cdots + a_{t-1} x_2^{t-1} (mod\ q),$$

$$\ldots,$$

$$y_t = P(x_t) = a_0 + a_1 x_t + \cdots + a_{t-1} x_t^{t-1} (\text{mod q}) \tag{4.2}$$

Matris ile ifade edilirse (Eşitlik 4.3);

$$\begin{bmatrix} 1 & x_1 & {x_1}^{t-1} \\ \ldots & \cdots & \ldots \quad \ldots \\ 1 & x_t & {x_t}^{t-1} \end{bmatrix} \begin{bmatrix} a_0 \\ \cdots \\ a_{t-1} \end{bmatrix} = \begin{bmatrix} y_1 \\ \cdots \\ y_t \end{bmatrix} (mod\ q) \tag{4.3}$$

Yukarıdaki matris Vandermonde matrisi olarak belirlenirse determinantı Eşitlik 4.4'deki gibi hesaplanır.

$$\det(M) = \prod_{1 \le j < k \le t}^{t} (x_k - x_j) (mod\ q) \tag{4.4}$$

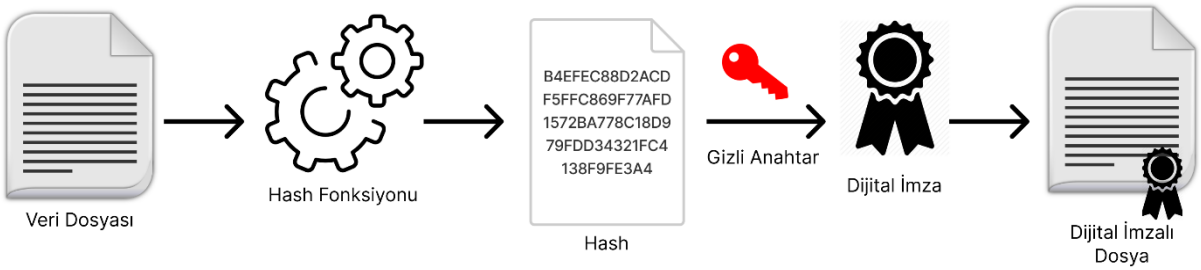
det(M) ≠ 0 ve farklı x_i'ler ve farklı noktalar seçildiğinden dolayı tek (unique) çözüm oluşmaktadır. Denklem sistemi çözülerek sır elde edilebilir. Bunun dışında Eşitlik 4.5'deki Lagrange enterpolasyon yöntemi ile de sır a_0 (Eşitlik 4.6) elde edilebilir.

$$P(x) = \sum_{i=1}^{t} y_i l_i(x), \qquad l_i(x) = \prod_{j=1, j\neq i}^{t} \frac{x - x_j}{x_i - x_j} (mod\ q) \tag{4.5}$$

$$a_0 = P(0) = \sum_{i=1}^{t} y_i \prod_{j=1, j\neq i}^{t} \frac{-x_j}{x_i - x_j} (mod\ q) \tag{4.6}$$

4.3.RSA Dijital İmza Şeması

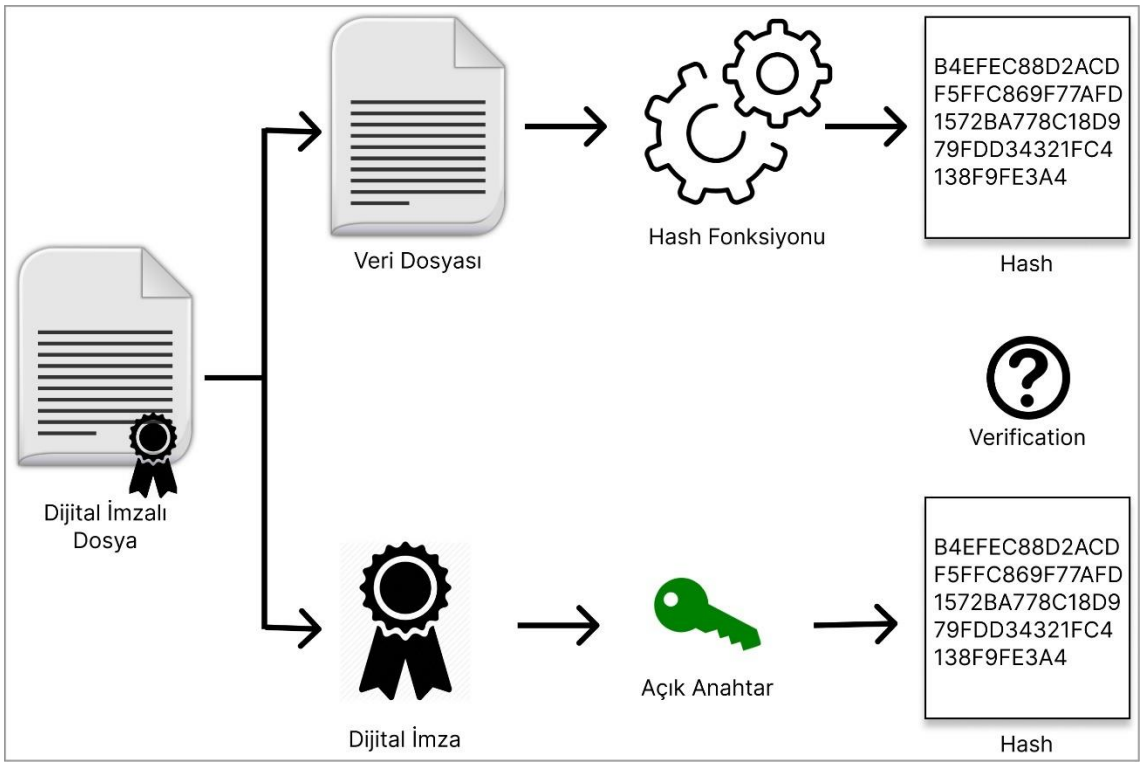
Dijital imza oluşturma ve doğrulama süreci sırasıyla gönderici ve alıcı taraflar tarafından sağlanır. Bir mesaj ile dijital imza oluşturulmadan önce mesajın sabit uzunluklu bir özeti oluşturulur. Özet oluşturmak için farklı hash fonksiyonları bulunmaktadır. Şekil 4.5'de görüldüğü gibi yalnızca mesajı gönderen tarafta bulunan özel bir anahtar ile mesaj özeti kullanılarak mesaja ait dijital imza oluşturulur.



Şekil 4.5. Dijital imza ile bir veri dosyasının imzalanması

Dijital imzanın açık anahtar ile doğrulama aşaması Şekil 4.6'da görülmektedir. Dijital imzayı doğrulamak için oluşturulan açık anahtar kullanılarak alıcı tarafından mesaj doğrulanır. Açık anahtar kriptografisi ile gizli anahtar ve açık anahtardan oluşan iki anahtar üretilir. Gizli anahtar sadece gönderici tarafta bulunur ve üçüncü şahıslar ile paylaşılmaz. Bunun nedeni sahte imzaların oluşturulmasını önlemektir. Gönderici tarafında üretilen mesaj özeti ve alıcı tarafında hesaplanan mesaj özeti aynı olmalıdır. Bu durum gerçekleştiğinde alıcı dijital imzanın geçerli olduğunu bilir. Açık ve gizli anahtar çifti üretilirken matematiksel yöntemlerle birbirini doğrulamak üzere tersinir olarak belirlenmektedir. Bu sayede alıcıya imzanın doğruluğuna ait güvence sağlanmaktadır.

Asimetrik açık anahtar şifreleme algoritmalarından biri olan RSA kullanılarak oluşturulan dijital imza şeması günümüzde yaygın olarak kullanılmaktadır. Dijital imza oluşturma sürecinde olduğu gibi, RSA algoritması kullanılarak açık ve gizli anahtar üretilerek dijital imza oluşturma ve doğrulama işlemleri yapılır (Gil, 2020).



Şekil 4.6. Dijital imzalı dosyanın açık anahtar ile doğrulanması

Hem RSA dijital imza algoritmasında hem de RSA açık anahtar şifrelemede anahtar oluşturma işlem adımları tamamen aynıdır. RSA'da anahtar oluşturma adımları aşağıdaki gibidir:

- 2 adet asal sayı seçilir. (p ve q)
- Açık ve gizli anahtar için mod değeri belirlenir. $n = p.q$
- $\vartheta(n) = (p-1)(q-1)$ totient hesaplanır.
- $1 < e < \vartheta(n)$ aralığında bir e tamsayısı üretilir. $\vartheta(n)$ ve e aralarında asal olmalıdır. e ve n açık anahtar olarak dağıtılır.
- $d.e = 1\ mod(\vartheta(n))$'e bağlı olarak d tamsayısı belirlenir. d, gizli anahtardır ve paylaşılmaz.
- Bir mesaj (m) belirlenir. Özet fonksiyonu kullanılarak hash değerine H=h(m) dönüştürülür.
- Özel anahtar d ve hash değeri (H) kullanılarak dijital imza $S = H^d mod(n)$ hesaplanır.
- $H = S^e\ mod(n)$ hesaplanarak doğrulama süreci gerçekleştirilir. Yani S'nin tersi mesajın özeti (H) olmaktadır.
- Gönderenin dijital imzasının geçerli olması için hesaplanan H ve mesajın özet

değerleri h(m) birbirine eşit olmalıdır.

4.4.Hyperledger Fabric ile Akıllı Sözleşme

Hyperledger, dağıtık defter teknolojisine (DLT) sahip izinli ve açık kaynaklı bir projedir. Konsorsiyum tarafından yönetilir ve ağın güvenilirliğini sağlamak amacıyla tüm katılımcılar birbirini tanır. Hyperledger Fabric izinli bir blokzincir platformudur. İzinli blokzincirde katılımcılar ilk olarak birbirlerini tanır ve güven mekanizması oluşturulur. İzinli ağda bir katılımcının veya hizmet sağlayıcıların kötü niyetli kod girmesi çok düşük bir olasılıktır. Başvuru işlemleri, ağ yapılandırması, akıllı sözleşmesi dağıtımı vb. eylemler onay politikasına bağlı olarak gerçekleştirilir. Anonim değildir ve herhangi bir sistem açığı kolayca tespit edilebilir ve yönetişim modeline göre incelenebilir.

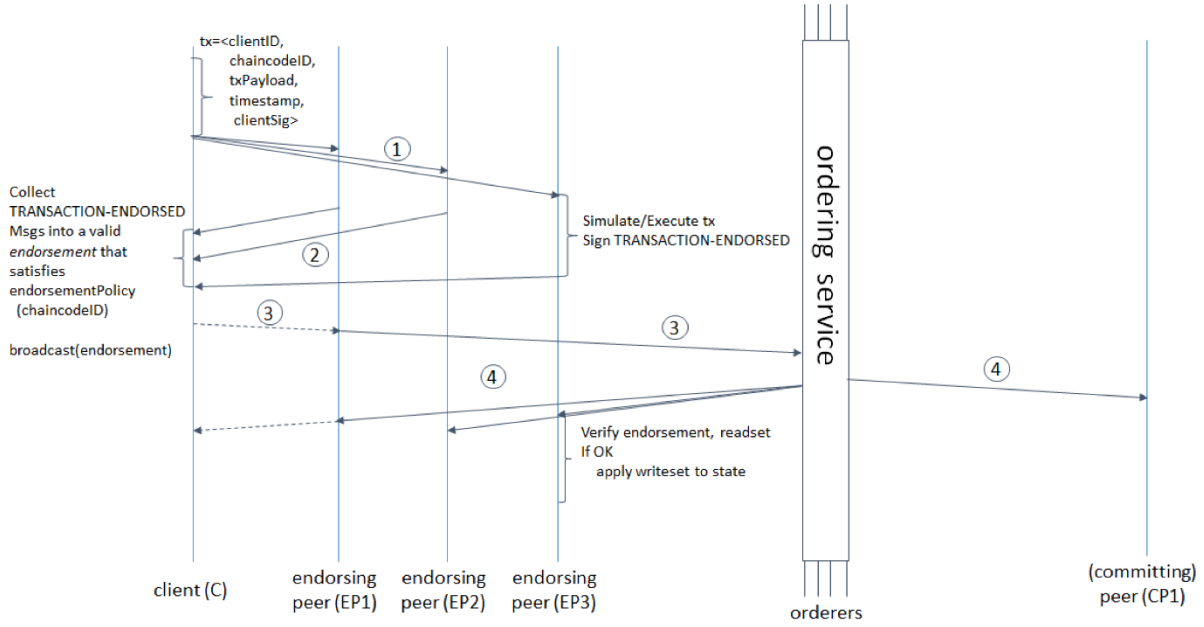
Hyperledger Fabric platformu kanal mimarisi ve izinli olması nedeniyle gizliliği esas almaktadır. Her bir kanalda, üyelerin sadece belirli işlem kümelerini yürütebilmesini sağlayan alt ağlar kurulur. Kanala tanımlı olmayan üyeler kanalın sahip olduğu sözleşmelere/kurallara tabi değildir. İzinli platformu kontrol eden otorite, üyeleri gerekli olan kanallara kaydeder ve bu üyeler kanalın yürüttüğü akıllı sözleşmeleri (chaincode), işlemlerin yapıldığı verileri görebilirler. Kanala katılan düğümler platform tarafından güvenilir olarak kabul edilir. Üyeler platformun gizlilik ve mahremiyetini korunmasından sorumludur. Ayrı bir kanal oluşturma performans ve bakım yükü açısından dezavantajlar oluşturabilir.

Ağdaki mantıksal eşler işlemleri yürütür ve defteri korur. Ordering servisi Hyperledger Fabric'te fikir birliği için işlemlerin düzenlenmesinden sorumludur. Konsensüs modüler bir mimariye sahip olduğu için ihtiyaç olduğunda güven mekanizmalarına göre dizayn edilebilir. Çökme hatasına dayanıklı (CFT) ve Bizans hatasına dayanıklı (BFT) mimariye uygun tasarlanan sistemlerin kullanımına izin verir. Fabric şu anda Raft protokolünün etcd kütüphanesine dayalı bir CFT sipariş hizmeti (ordering service) uygulaması sunmaktadır. Fabric ağında birden fazla sipariş hizmeti desteklenebilir. Bu yönetici servislerinde (ordering service) farklı uygulamalar ve uygulama gereksinimleri bulunabilir (Androulaki vd., 2018).

Bu çalışmada Go programlama dili ile yazılan chaincode; transaction ile defter durumunu başlatan ve yöneten bir programdır. Chaincode iş mantığını belirlediği için akıllı sözleşme olarak kabul edilmektedir. Bir defter üzerinde yapılan güncellemelere ve sorgulara sadece ilgili chaincode ile erişilebilir. Chaincode yaşam döngüsü dört aşamadan

oluşur.

- Chaincode'un paketlenmesi: Chaincode eşlere yüklenmeden önce paketlenir.
- Chaincode'un eşlere yüklenmesi: İşlemleri yapacak ve çalıştıracak olan tüm eşlere chaincode yüklenir.
- Bir chaincode'un organizasyon için onaylanması: Kanalda bulunan üyelerin chaincode için geçerli ad, versiyon, sıra, onay politikası, koleksiyon yapılandırması vb. tanımlamaları onaylamaları gerekir. Yeterli sayıda kanal üyesi tarafından onaylanırsa chaincode kanalda kullanılabilir.
- Chaincode tanımının kanala eklenmesi: Chaincode tanımının kanala taahhüt ettirilmesi gerekir. Taahhüt işlemi gerçekleştikten sonra onay politikasına bağlı olarak kanalda chaincode'un kullanılmasına izin verilir.



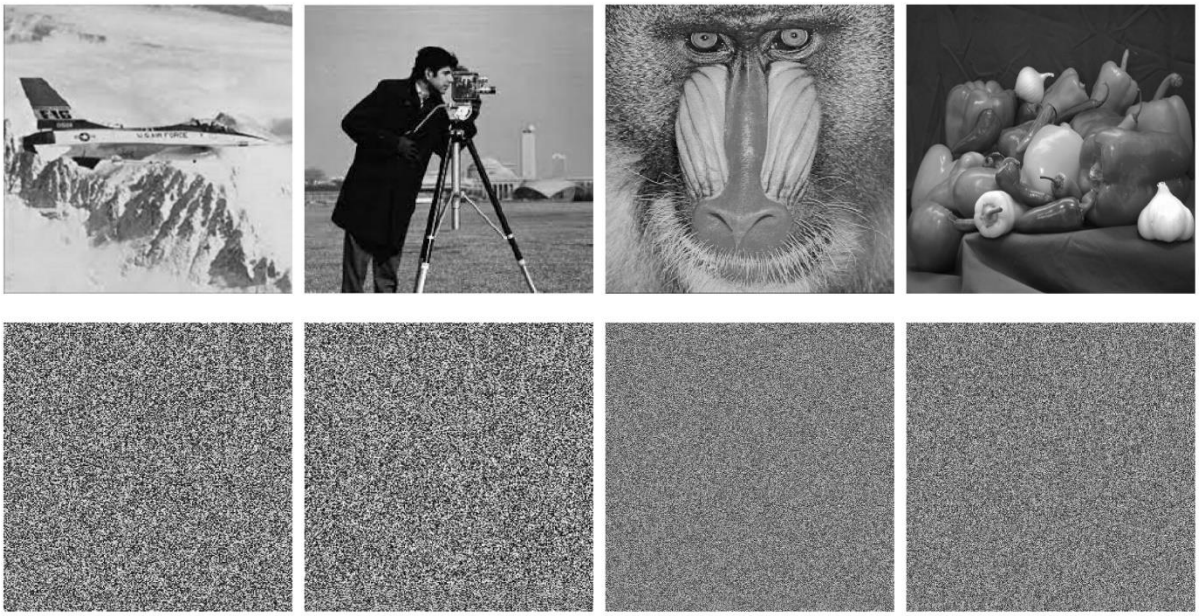
Şekil 4.7. Hyperledger Fabric'in işlem akış şeması.

Şekil 4.7'de Hyperledger Fabric'in (HyperledgerFabric, 2015)'den alınan transaction akışı görülmektedir. Herhangi bir client 1. evrede chaincode'u kullanarak bir işlem başlatır. İşlem teklifini yürütmek isteyen eşlerin kriptografik kimlik bilgileri endorsing peers tarafından doğrulanır ve işlem yürütülür. Transaction teklifine gelen yanıtlar incelenir. Yanıtlar endorsing peer'ların hepsinde aynı olmalıdır. Transaction teklifine gelen yanıt işlem mesajı olarak endorsing peer'a yayınlanır. Bir sonraki aşama işlemin doğrulanması ve taahhüt edilmesidir. Kanaldaki tüm eşlere teslim edilen transaction blokları ile defter durumundaki değişiklik doğrulanır. Transaction'un sonucu geçerli/geçerli değil şeklinde etiketlenir. Defter güncellenir ve eşler bloğu bulundukları

kanalın zincirine ekler. Son olarak veritabanında güncelleme yapılır.

4.5.Görüntü Şifreleme

CCM tabanlı görüntü şifreleme algoritması (Chen vd., 2004) ile şifrelenen gri seviye airplain ve cameraman (256x256), baboon ve peppers (512x512) görüntülerinin orijinal ve şifreli durumları Şekil 4.8'de görülmektedir. Anahtar uzunluğu (Chen vd., 2004)'de 128 bit iken bu çalışmada 256 bite çıkarılmıştır. Ayrıca 128 bit anahtar kullanılarak yapılan CCM tabanlı görüntü şifreleme ve analiz sonuçları (Tunçer & Karakuzu, 2022)'de incelenmiştir. Boyutu 256 bite çıkarılan anahtar, görüntü şifrelemede performans kriterlerini ve şifreleme süresini etkilememiştir. Orijinal ve şifreli görüntüler kullanılarak yapılan performans analizleri takip eden alt başlıklarda anlatılmaktadır.



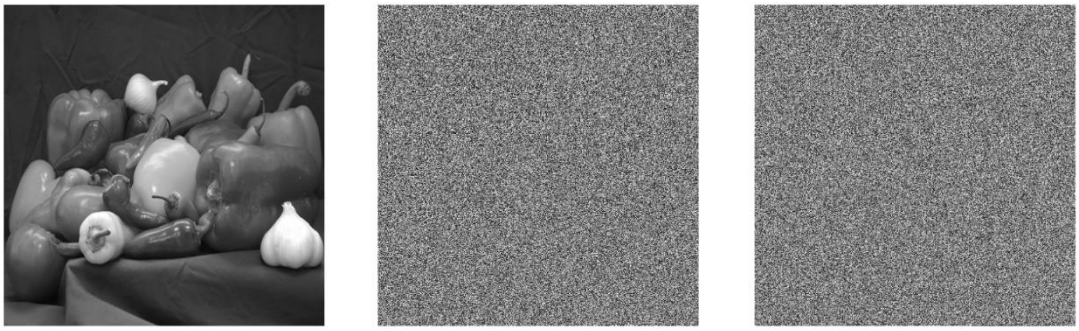
Şekil 4.8. Üsttekiler sırasıyla orijinal airplain, cameraman, baboon ve peppers görüntüleri, alttakiler sırasıyla şifreli airplain, cameraman, baboon ve peppers görüntüleri.

4.5.1.Anahtar boyutu ve anahtar hassaslığı güvenliği

Şifrelemede kaba kuvvet saldırılarına karşı dayanıklılığı sağlayan en büyük etkenlerden biri anahtar boyutudur. CCM tabanlı görüntü şifrelemede anahtar boyutu 256 bit olarak belirlenir. Anahtar uzayı 2^{256} olduğundan dolayı kaba kuvvet saldırılarına karşı daha güvenli hale gelmiştir. Kaotik sistemlerde kullanılan kontrol parametreleri anahtara dahil edilirse anahtar boyutu artmaktadır.

Kaotik sistemler, anahtar hassasiyetini önemli ölçüde artırır. Kaotik sistemin giriş parametreleri ve başlangıç değerleri değiştiğinde şifrelenmiş görüntüden orijinal

görüntüyü elde etmek imkânsız hale gelir. Başlangıç değerlerinin değişimi en az 10^{-14} hassasiyetinde veya şifrelemede kullanılan gizli anahtarın herhangi bir biti değiştirildiğinde şifreli görüntü tamamen değişir. Şekil 4.9'da gizli anahtarın bir biti değiştirildiğinde şifreli görüntüden orijinal görüntünün elde edilemediği gösterilmiştir. Algoritma gizli anahtara duyarlı ve saldırılara karşı dirençlidir.



Şekil 4.9. Peppers görüntüsünün sırasıyla orijinal, şifreli ve anahtarın bir biti değiştiğinde deşifre edilmiş durumu

4.5.2.Gizli anahtar analizi

Diferansiyel saldırılara karşı gizli anahtarın gücünü nicel olarak tahmin edebilmek için iki metrik vardır. Bunlar piksel sayısı değişim oranı (NPCR) ve birleşik ortalama değişen yoğunluk (UACI) metrikleridir (Wu vd., 2011). Sırasıyla Eşitlik 4.7 ve Eşitlik 4.8 formülleri ile hesaplanır.

$$NPCR = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} D\left(I_{i,j} - I'_{i,j}\right) \times 100\% \tag{4.7}$$

$$D(x) = \begin{cases} 0, & x = 0 \\ 1, & x \neq 0 \end{cases}$$

$$UACI = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{\left|I_{i,j} - I'_{i,j}\right|}{255} \times 100\% \tag{4.8}$$

Burada görüntü satır sayısı M, sütun sayısı N ile ifade edilir. $I_{i,j}$ ve $I'_{i,j}$ sırasıyla orijinal ve şifreli görüntünün i nci satır j nci sütun piksel değeridir. (Feixiang vd., 2021), (Khan & Byun, 2020) ve (Gao vd., 2022)'deki çalışmalar için ortalama NPCR sırasıyla 99.69, 99.60, 99.57 ve ortalama UACI sırasıyla 33.43, 33.41, 33.46 olarak hesaplanmıştır. Tablo 4.1'de elde edilen değerler (Feixiang vd., 2021), (Khan & Byun, 2020) ve (Gao vd., 2022)'de elde edilenlere göre kıyaslandığında kaotik cat map (CCM) tabanlı şifrelemenin diferansiyel saldırılara karşı direncinin yüksek olduğu söylenebilir.

Tablo 4.1. Şifreli görüntülerin NPCR ve UACI değerleri

Görüntü	NPCR (%)	UACI (%)
Airplain	99.58	33.50
Cameraman	99.60	33.48
Baboon	99.60	33.46
Peppers	99.61	33.60

4.5.3.Bilgi entropi analizi

Entropi, bir sistemin belirsizlik derecesi olarak adlandırılır. m mesajına ait bilgi entropisi Eşitlik 4.9'daki formül ile hesaplanır.

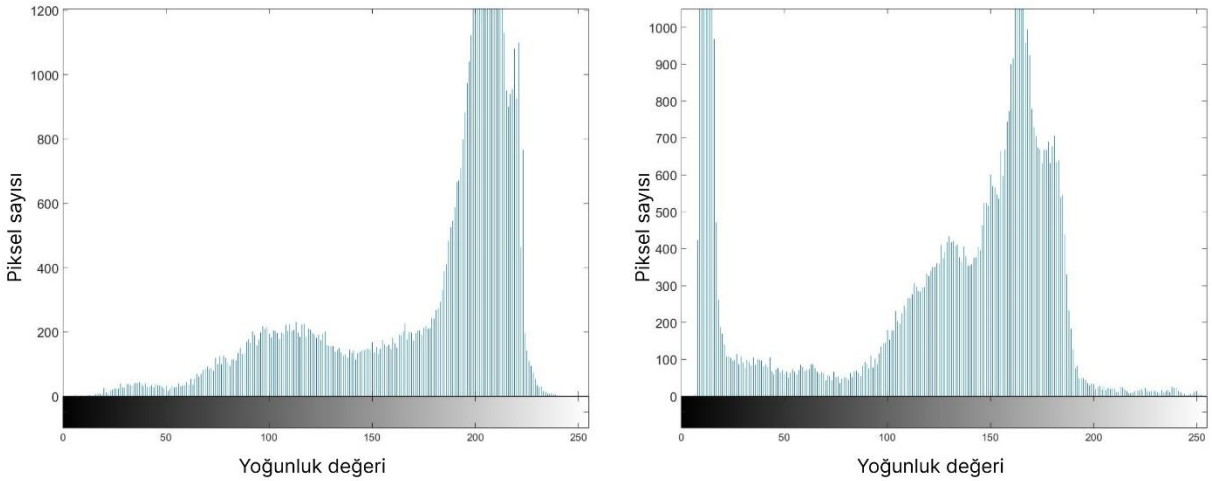
$$E(m) = -\sum_{j=0}^{2^n-1} p(m_j) log_2 \frac{1}{p(m_j)} \tag{4.9}$$

$p(m_j)$, m_j'nin meydana gelme olasılığıdır. $p(m_j) = 2^{-8}$, $n = 8$ ise her sembol eşit olasılığa sahiptir. Yani, entropideki dağılım $E(m) = 8$ olmaktadır. $E(m) = 8$ değerine yakın olması dağılımın görüntünün etkili bir şekilde bozulduğunu göstermektedir.

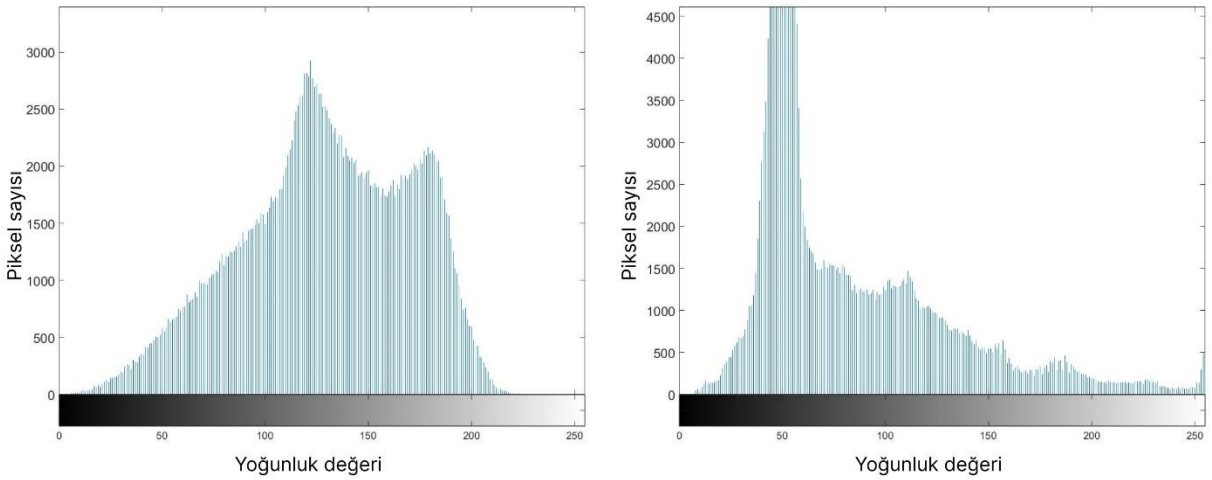
Sırasıyla Airplain, Cameraman, Baboon ve Peppers görüntülerinin entropisi 7.9967, 7.9970, 7.9971, 7.9965 elde edilmiştir. Ortalama entropi 7.9968'dır. CCM ve referans (Feixiang vd., 2021), (Khan & Byun, 2020), (Abdelfatah, 2020) ve (Gao vd., 2022)'deki ortalama entropiler sırasıyla 7.9968, 7.9937, 7,9972, 7.9993, 7.9969 olarak bulunmaktadır. Entropi açısından CCM tabanlı şifrelemenin başarılı olduğu görülmektedir.

4.5.4.Histogram analizi

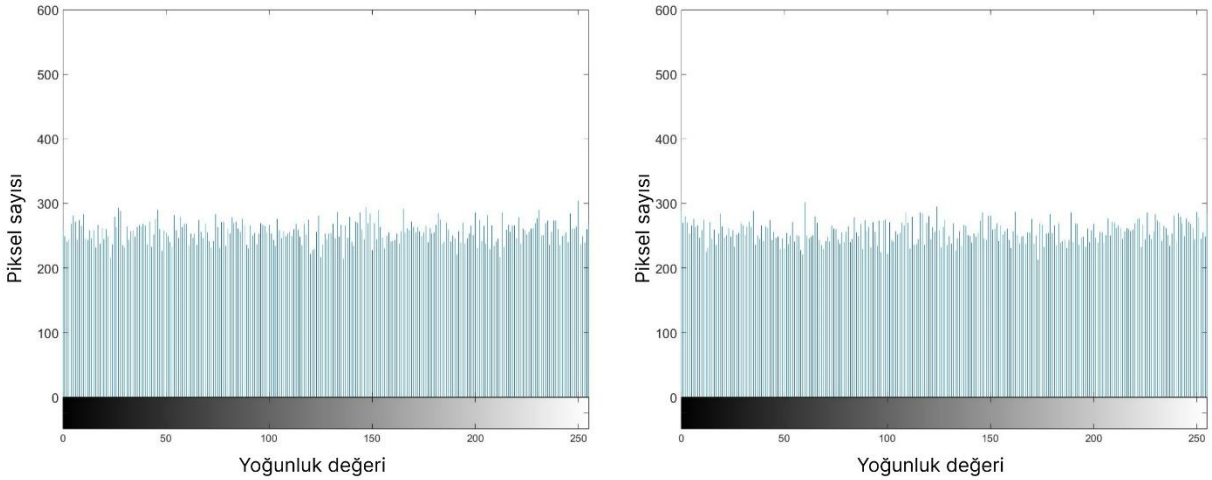
Bir görüntüdeki piksel değerleri [0, 255] aralığında bir tamsayıdır. Herhangi bir pikselin belirtilen aralıktaki rasgele bir değer olma olasılığı 1/256'dır. Şifreleme algoritmasının istatistiksel saldırılara karşı dayanıklı olması için şifreli görüntünün piksel değerlerinin homojen bir şekilde dağılması gerekir. Şekil 4.10'da 256x256 ve Şekil 4.11'de 512x512 boyutlarında gri seviye orijinal görüntülere ait histogramlar verilmiştir. Şekil 4.12 ve Şekil 4.13 sırasıyla şifreli görüntülere ait histogramlardır. Şifreli görüntülere ait histogram incelendiğinde oldukça düz olduğu ve piksel değerlerinin homojen dağıldığı görülmektedir. Şifreli görüntüden orijinal görüntüye ait herhangi bir bilgi elde edilemediği için istatistiksel saldırılara karşı direnç yüksektir.



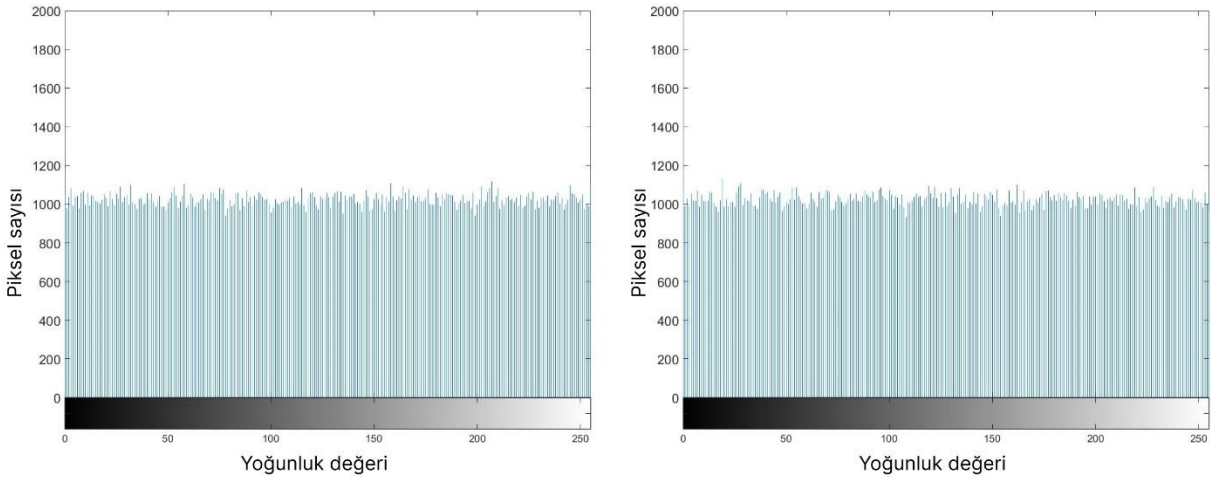
Şekil 4.10. Orijinal Airplain (sol) ve Cameraman (sağ) görüntülerinin histogramları



Şekil 4.11. Orijinal Baboon (sol) ve Peppers (sağ) görüntülerinin histogramları



Şekil 4.12. Şifreli Airplain (sol) ve Cameraman (sağ) görüntülerinin histogramları



Şekil 4.13. Şifreli Baboon (sol) ve Peppers (sağ) görüntülerinin histogramları

4.5.5.Korelasyon katsayı analizi

İki veya daha fazla verinin arasındaki ilişki birbirine göre azalma veya artış açısından incelenerek korelasyon hesaplanır. Herhangi bir görüntüde bitişik olan pikseller genellikle birbirine yakın değerlerden oluşur. Bundan dolayı orijinal görüntülerde korelasyon katsayısı genellikle yüksektir. Şifreli görüntülerde komşu piksellerin değerlerinin arasındaki benzerlik minimuma düşürülmelidir. Bu sayede korelasyon katsayısı küçük olur ve şifreleme algoritmasının etkili olduğu söylenebilir. Korelasyon katsayısı k_{xy} Eşitlik 4.10'daki formüller ile hesaplanmaktadır. x, y iki pikselin gri seviye değerleri, $e(x)$ ortalama değer, $d(x)$ varyans, $cov(x,y)$ oluşan değişikliği temsil etmektedir.

$$\begin{cases} e(x) = \dfrac{1}{N_l}\displaystyle\sum_{i=1}^{N_l} x_i \\ d(x) = \dfrac{1}{N_l}\displaystyle\sum_{i=1}^{N_l} (x_i - e(x))^2 \\ cov(x,y) = \dfrac{1}{N_l}\displaystyle\sum_{i=1}^{N_l} (x_i - e(x))(y_i - e(y)) \\ k_{xy} = \dfrac{cov(x,y)}{\sqrt{d(x)}\sqrt{d(y)}} \end{cases} \tag{4.10}$$

CCM ile şifrelenen dört farklı orijinal ve şifreli görüntüye ait yatay, dikey, diyagonal piksellerinin ortalama korelasyon değerleri Tablo 4.2'de görülmektedir. Görüntülerden rastgele 5000 bitişik piksel çiftleri seçilerek korelasyon analizi yapılmıştır. Şifreleme algoritmasının başarılı olduğunu söyleyebilmek için korelasyon değerlerinin

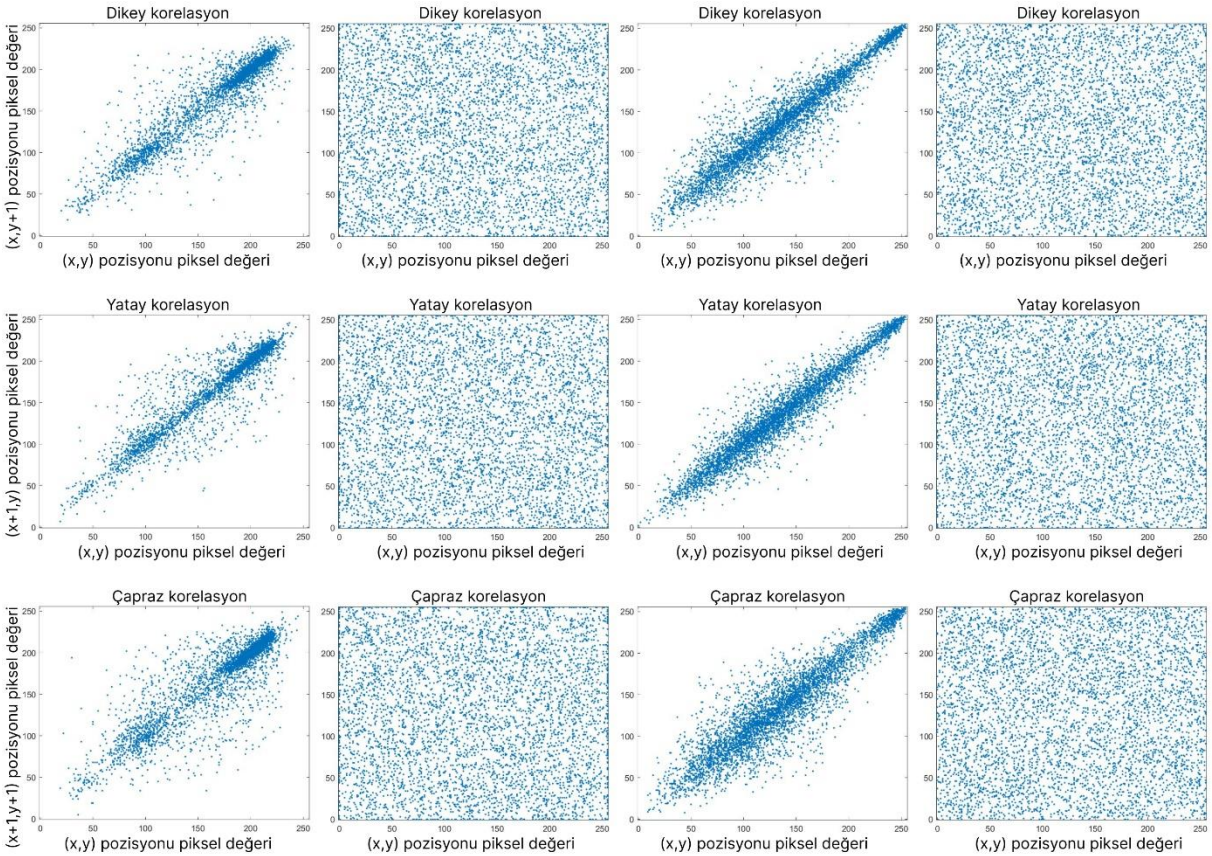
sıfıra yakın çıkması gerekmektedir. (Feixiang vd., 2021), (Khan & Byun, 2020), (Abdelfatah, 2020) ve (Gao vd., 2022)'de bu çalışmada kullanılmayan başka şifreli görüntülerin ortalama korelasyonu sırasıyla yaklaşık olarak 0.0074, 0.0457, 0.0025 ve 0.0023 civarında olduğu görülmektedir. Bu değerler dikey, yatay ve diyagonal korelasyon için değişiklik göstermektedir.

Tablo 4.2. Orijinal ve şifreli görüntülerin dikey, yatay ve çapraz korelasyonu

Görüntü	Dikey	Yatay	Çapraz
Airplain	0.9401 0.0016	0.9369 0.0246	0.8896 0.0167
Cameraman	0.9606 0.0021	0.9330 0.0097	0.9078 0.0182
Baboon	0.9432 0.0058	0.9621 0.0082	0.9156 0.0279
Peppers	0.9917 0.0351	0.9928 0.0049	0.9872 0.0036

Bu çalışmada denektaşı olarak literatürde de sıklıkla kullanılan görüntüler için dikey, yatay, çapraz korelasyon değerleri Tablo 4.2'de verilmiştir. Tablonun hücresinde verilen iki değerden üstteki orijinal görüntü için, alttaki ise şifreli görüntü için hesaplanan korelasyon değeridir. Tablo 4.2'den görüleceği üzere (Feixiang vd., 2021), (Zhang vd., 2020), (Abdelfatah, 2020) ve (Gao vd., 2022)' e bakılarak CCM'nin performansı açısından korelasyon katsayılarının uygun ve yeterli olduğu söylenebilir.

Tablo 4.2'de orijinal (satır 1) ve şifreli (satır 2) görüntüye ait korelasyon değerleri görülmektedir. Orijinal görüntülerde %90'ın üzerinde seyreden korelasyon, şifreli görüntülerde ihmal edilebilir derecede küçüktür. Şekil 4.14'de iki görüntünün dikey, yatay ve çapraz korelasyon grafikleri mevcuttur. Şifreli görüntülere ait grafikler homojen dağılım gösterdiğinden dolayı pikseller arasındaki ilişki zayıftır bu da kullanılan algoritmanın başarılı olduğunu göstermektedir.



Şekil 4.14. Sırasıyla Airplain (soldaki 2 sütun) ve Baboon (sağdaki 2 sütun) görüntülerine ait dikey, yatay ve çapraz korelasyon grafikleri

4.6.Sonuçlar

Gizliliğin ve özellikle güvenliğin ön planda olduğu görüntü şifreleme ve akıllı kontrakt tabanlı blokzincir doğrulama şeması önerilmiştir. Görüntü şifrelemede kullanılan gizli anahtar sır paylaşımı ile paydaşlara dağıtılmaktadır. Bu sayede paydaşlar arasında karşılıklı güvene dayalı bir şifreleme mekanizması oluşturulmaktadır. Bu mekanizma ile önemli görüntülere sadece bir paydaşın ulaşmasının engellenmesi sağlanmaktadır. Hyperledger Fabric tabanlı izinli blokzincir çatısı kullanılarak yazılan akıllı kontraktlar sayesinde paydaş imzaları ve görüntüye ait bilgiler dağıtık defterde saklanmaktadır. Akıllı kontrakt ile yapılan her sorgu, kayıt ve istek blokzincirdeki dağıtık deftere kaydedildiğinden dolayı güvenlik artmaktadır. Dağıtık defterin değişmezliği sayesinde gizli anahtarın dağıtıldığı paydaşlar ve şifrelenen görüntü rahatlıkla doğrulanabilmektedir. Şifreli görüntülerin açık/gizli bir bulut platformunda bulunması yapılan performans analizlerine bakıldığında güvenlik açısından zaafiyet oluşturmamaktadır. Bu çalışma ile gizlilik ve yüksek güvenlik sağlayan gerçek bir ağda uygulanabilir bir görüntü şifreleme ve doğrulama mekanizması oluşturulmuştur.

5. ETHEREUM AKILLI SÖZLEŞME UYGULAMASI (Ethtrace)

Akıllı sözleşme tabanlı blokzincir uygulamaları izlenebilirlik, değişmezlik ve doğrulama işlemlerinde güvenliği sağladığından dolayı tercih edilmektedir. Yapısında barındırdığı dağıtık defter ile gerçekleşen işlemler, para transferleri ve sorguların tamamı kayıt altına alınmaktadır. Blokzincirde işlemlerin ağda bulunan düğümler arasında onaylanmasını ve dağıtık deftere kaydedilmesini sağlayan fikir birliği algoritması kaba kuvvet (*brute force*) ataklarına karşı güvenliği sağlamaktadır. Bu çalışmada, merkezi bir bilgi işlem birimine ihtiyaç duymadan, bağışçı kuruluş ile bağış yapan kişi/kurum arasında blokzinciri üzerinde yayınlanan akıllı sözleşme ile kripto para transferinin sağlandığı şeffaf bir bağış sistemi geliştirilmiştir. Sistemde, akıllı sözleşme ile blokzincir üzerinde yapılan işlemler kayıt altına alınır. Önerilen çalışma bağışçının, bağışlanan miktarın ve bağış yapılan kurumların şeffaf bir şekilde takip edilebilmesini sağlayan merkezi olmayan bir uygulamadır. Bağışların akıllı sözleşme ile birden fazla kuruma aynı anda aktarılmasını sağlamak amaçlanmaktadır. Kurumların dürüst, güvenilir ve şeffaf olmalarını sağlayarak halktan alınan bağışları teşvik etmek amacıyla bu çalışma gerçekleştirilmiştir.

Bu çalışmada sivil toplum kuruluşları ve bağış alan kuruluşlar tarafından kullanılabilecek kripto para birimi ile bağış yapılabilen blokzincir tabanlı bir sistemin temellerinin atılması amaçlanmaktadır. Bu yaklaşım sayesinde üçüncü şahıslar ortadan kaldırılabilmekte ve para transferleri tamamen şeffaf bir şekilde herkes tarafından takip edilebilmektedir. Bağışların uygun yerlerde kullanılması blokzincir teknolojisi sayesinde daha kolay izlenebilmekte, doğrulanabilmekte ve son kullanıcı ile kurumlar arasında bir güven ortamı oluşturmaktadır. Bu sistem sayesinde son kullanıcının arka planda işlemleri göremediği bankalar, aracı servisler ve yazılımlar yerine tüm dünyada kripto para bağışlarının kolaylıkla yapılabileceği bir ortam oluşturulmaktadır. Merkezi olmayan bir uygulama olduğu için bağış ekosisteminde güven ortamının oluşmasında da hayati bir rol oynamaktadır.

Blokzincir üzerinden kripto para bağışlamak için bu çalışmada tercih edilen programlama dili dışında birçok seçenek bulunmaktadır. Solidity programlama dilini seçmedeki en önemli kriter, merkezi olmayan uygulama geliştirme sürecinin karmaşıklığıdır. Yerel blokzinciri olarak kullanılan Ganache üzerinde akıllı sözleşme yazma, yayınlama ve test etmede sunduğu birçok kolaylık nedeniyle Truffle çerçevesi seçildi. Merkezi olmayan bir uygulama geliştirme, bilinen programlama dilleri ile

merkezi bir uygulama geliştirmeye kıyasla kodlama, derleme, test etme, temel çalışma ilkeleri ve çalışma mantığı açısından birçok farklılığa sahiptir. Uygulama herhangi bir otoriteye bağlı olmadığı için hedeflenen bir hizmete veya çözüme ulaşma amacı dışında oluşabilecek hukuka aykırı bir durum, hizmetin aksaması, öngörülemeyen kodlama hataları gibi birçok konunun derinlemesine ele alınması gerekir. Ethereum'un güvenlik, ölçeklenebilirlik ve hız kriterlerinin yeterli olması, uygulamayı performans kriterleri açısından güvenilir kılmaktadır. Uygulamada kullanılan teknolojiler ve uygulamanın mimari yapısı ilerleyen alt bölümlerde açıklanmıştır. Ayrıca akıllı kontrat geliştirme için Ethereum'un tercih edilme sebepleri belirtilmiştir.

Bu çalışma, akıllı sözleşme tabanlı Ethereum blokzincirine dayalı şeffaf bir hayır kurumu bağış uygulaması sunmaktadır. Son yıllarda hayır kurumlarında güven ortamının azalmaya başlamasıyla birlikte şeffaflık ve dürüstlük en çok aranan kavramlar haline geldi. Mevcut sistemde istenmeyen durumlardan dolayı bankalar ve mobil uygulamalar üzerinden yapılan bağışlar azalmaktadır. Blokzincirindeki dağıtık defter sayesinde bağışların halka açık olarak takip edilebildiği ve değişmezliğin sağlandığı bir çalışma yapılması amaçlanıyor. Bu sayede güven ortamı oluşturulmakta ve hayır kurumları merkezi olmayan bir uygulama ile tek çatı altında toplanarak izlenebilmeleri ve incelenebilmeleri sağlanmaktadır. Bu çalışmayı literatürdeki diğer çalışmalardan ayıran en önemli özellik, tanınmış bağış kuruluşlarına eş zamanlı bağışların tek çatı altında yapılabilmesine imkân verecek şekilde tasarlanmış olmasıdır. Akıllı sözleşmeye dahil olan tüm kurumlara blokzincir tabanlı bir uygulama üzerinden bağış yapılabilmesi ve bağışların denetlenmesi için bu çalışma önerilmektedir. Bu sayede tüm bağışların şeffaf bir şekilde izlenmesi amaçlanmaktadır.

5.1.Ethereum Tabanlı Akıllı Sözleşme

Akıllı sözleşme uygulamasına ait genel bilgi ve içerik (Tunçer vd., 2022)‘de verilmiştir. Uygulamanın hayata geçirilmesi sırasında Ethereum tabanlı Solidity dili kullanılmıştır. Akıllı sözleşme yazılımının uygulanması ve test edilmesi sırasında Truffle çerçevesini barındıran web tabanlı bir uygulama, Truffle üzerinde yazılan akıllı sözleşmenin yayınlanması ve sözleşmenin web arayüzünde ve web3'te kullanılması için Ganache ile geliştirilmiştir. Son kullanıcıların ve kuruluş sahiplerinin kullanımına yönelik js kitaplığı. Truffle, Ethereum Sanal Makinesi'ni (EVM) kullanarak blokzincirlerde akıllı sözleşme uygulamaları oluşturmak, yayınlamak ve test etmek için bir yazılım çerçevesidir.

Ganache, Ethereum tabanlı akıllı sözleşme uygulama geliştirme, yayınlama ve test etme için kullanılan kişisel bir blokzincir geliştirme ortamıdır. Test aşaması, özellikle uygulamanın güvenli ve deterministik bir yapıya sahip olup olmadığını kontrol eder. Solidity, sözleşme tabanlı bir programlama dilidir. C++ ve Python dillerinden etkilenerek geliştirilmiştir ve EVM üzerinde çalışacak şekilde tasarlanmıştır. Bu çalışmada bağış için kullanılan akıllı sözleşme Solidity dili ve Truffle çerçevesi kullanılarak yazılmış ve test edilmiştir.

Akıllı sözleşmeler, gönderici ile alıcı arasındaki anlaşmanın kodudur. Koda dönüştürülen akıllı sözleşme ile yapılan anlaşmalar bir blokzincirinde saklanır. Kod yayınlandıktan sonra üçüncü şahıslar tarafından değiştirilemez ve üzerinde yapılan işlemleri kontrol eden, izleyen ve geri alınamaz hale getiren bir mekanizma oluşturur. Bir blokzinciri, bir veya daha fazla akıllı sözleşme içerebilir. Ağ içindeki kullanıcılar akıllı sözleşmeleri kullanabilir veya sadece istediklerini kullanabilir. Akıllı sözleşmelerin çalışma prensipleri basitçe "if/when-then" durumları ve bunları kod olarak düzenleyerek blokzincirinde kayıt altına almak olarak tanımlanabilir. Blokzincir ağına bağlı bilgisayarlar, sözleşmedeki şartların yerine getirilmesi ve doğrulamanın ardından işlemleri gerçekleştirir. Bu işlemler arasında belirli taraflara para transferi, araç kaydı, bildirim gönderme veya bilet düzenleme yer alabilir. Bir işlem yürütüldüğünde, dağıtılmış defter, o işlemle blokzincirine eklenerek güncellenir. Bu, işlemin değiştirilemeyeceği veya geri alınamayacağı anlamına gelir. Yalnızca yetkili kullanıcılar/kuruluşlar sonuçları görebilir.

EVM, blokzincirindeki tüm düğümleri birbirine bağlayan programdır. Ethereum yapısında barındırdığı EVM sayesinde merkezi olmayan bir yapıya sahiptir. Akıllı sözleşmelerin yürütülmesi ve bakımı EVM tarafından yapılır. Algoritmayı çalıştırdığı ve sözleşme yaptığı için, blokzincirde bulunan sorunlardan biri olan çifte harcamanın önüne geçer. İşlem gerçekleştikten sonra ağdaki düğümlere yayılarak ve kaydedilerek çifte harcama olmaması sağlanır. Metamask, kullanıcıların ether ve diğer kripto para birimlerini tutabilecekleri bir kripto cüzdan uygulamasıdır. Birincil amacı, Ethereum blokzinciri ile iletişim kurmaktır. Ayrıca, kullanıcıların merkezi olmayan web siteleriyle etkileşime girmesine olanak tanır. Bir tarayıcı uzantısı veya mobil uygulama kullanarak Ethereum cüzdanına erişim sağlar.

Bitcoin (Nakamoto, 2008), Ethereum (Ethereum, 2015), Hyperledger Fabric (Androulaki vd., 2018), Corda (Brown, 2018), EOS (Xu vd., 2018), Nxt (NxtCommunity,

2014), Tron (TronCommunity, 2018) vb. popüler blokzinciri uygulamalarıdır. Kullanım amaçlarına göre yapılarında farklı özellikler taşırlar. Tablo 5.1, şeffaf bağışta kullanılacak blokzinciri platformlarının seçimini etkileyen önemli kriterleri sunmaktadır. Ayrıca Tablo 5.1'de bu çalışmada seçilen blokzincir (Ethereum) ve özellikleri gösterilmektedir. Şeffaf bağış başvuruları akıllı sözleşmeler içermeli ve blokzincir ağı halka açık olmalıdır. Mutabakat protokolü, güvenlik ve hız açısından beklentileri karşılamalıdır.

Tablo 5.1. Blokzincir platformlarının değerlendirme kriterleri

Referans	Blokzincir	Akıllı Sözleşme	Blok-zincir Ağı	Konsensüs Protokolü	Token	Blok Boyutu	Saniye Başına İşlem Sayısı	Program-lama Dili
(Nakamoto, 2008)	Bitcoin	Hayır	Genel	PoW	Evet	1 MB	7	C++
(Ethereum, 2015)	Ethereum	Evet	Genel	PoW, PoS	Evet	80 KB	20000-100000	Solidity
(Androulaki vd., 2018)	Hyperledger Fabric	Evet	Özel	BFT, Raft, Kafka, Solo	Hayır	Değiş-ken	3500	Go
(Brown, 2018)	Corda	Evet	Özel	BFT	Evet	Değiş-ken	20000	Kotlin
(Xu vd., 2018)	EOS	Evet	Genel	DPoS	Evet	1 MB	1000000	C++
(NxtCommunity, 2014)	Nxt	Hayır	Genel	PoS	Evet	42 KB	255	Java
(TronCommunity, 2018)	Tron	Evet	Genel	DPoS	Evet	1.9 MB	2000	Solidity

Güvenlik açısından en üst seviyede olan Proof of Work (PoW) mutabakat protokolü, işlemlerin çok yavaş gerçekleşmesine neden olur. Ayrıca blok boyutu mutabakat protokolüne göre belirlenmezse işlem gecikmeleri oldukça artar. Bu nedenle Ethereum'a alternatif olan Proof of Stake (PoS) tercih edilmelidir. Solidity programlama dilinin deterministik olması, kullanılmasının en önemli nedenlerinden biridir. Tüm bu kriterlerin dışında platformun geniş bir kitle tarafından kabul görmesi, saldırılara karşı kendini kanıtlamış olması ve akıllı sözleşme geliştirme aşamasında sağladığı avantajlar tercih edilmesinde etkili oluyor.

5.2.Akıllı Sözleşme Mimarisi

Önerilen sistemde kullanıcı arayüzü ve akıllı sözleşme birbirinden bağımsız olarak tasarlanmış ve daha sonra entegre edilmiştir. İlk olarak Truffle çerçevesi kullanılarak Solidity programlama dilinde bir akıllı sözleşme geliştirilmiştir. Daha sonra akıllı sözleşmede yer alan parametrelere bağlı olarak web arayüzü tasarlanmıştır. Blokzincir üzerinde yayınlanan akıllı sözleşmenin tüm fonksiyonları web3.js kütüphanesi yardımıyla web arayüzünde kullanılabilmektedir.

Bir akıllı sözleşmede, bir görevi tamamlamak için yeterli koşullar belirtilebilir. Sözleşme yazımı sırasında kodun karmaşıklığını artıran koşullardan kaçınılmalıdır. Akıllı sözleşme kapsamında yapılan her işlem Gas adı verilen ücretle değerlendirilir. Akıllı sözleşme kodlarının karmaşıklığı ve blokzincir üzerinde büyüyecek işlem boyutları, Gas ücretini çok yüksek seviyelere çıkarabilir. Ayrıca yazılı sözleşmedeki kuralların fazlalığı hata olasılığını artırabilmektedir. Bu çalışmada web arayüzünde bağış yapılacak kurumlar ve adresleri belirlenmiştir. Sözleşme kapsamında yüksek Gas ücretlerinden kaçınmak için sadece seçilen adresin geçerliliği kontrol edilir.

Ganache bileşeni kullanılarak, ilgili tüm katmanların blokzincir üzerinde çalışması sağlanır. Şekil 5.1'de önerilen sistemdeki akıllı sözleşmedeki temel fonksiyonların çalışma prensipleri şu şekildedir;

• Metamask aracılığıyla web arayüzüne kripto cüzdan bağlantısı kurma

• Ad, tür ve cüzdan değerleri ile sözleşmeye yeni kuruluş kaydı

• İşlemin yapıldığı cüzdan ile önceki kayıtları kontrol etme

• Akıllı sözleşmeye kayıtlı cüzdanları sorgulama

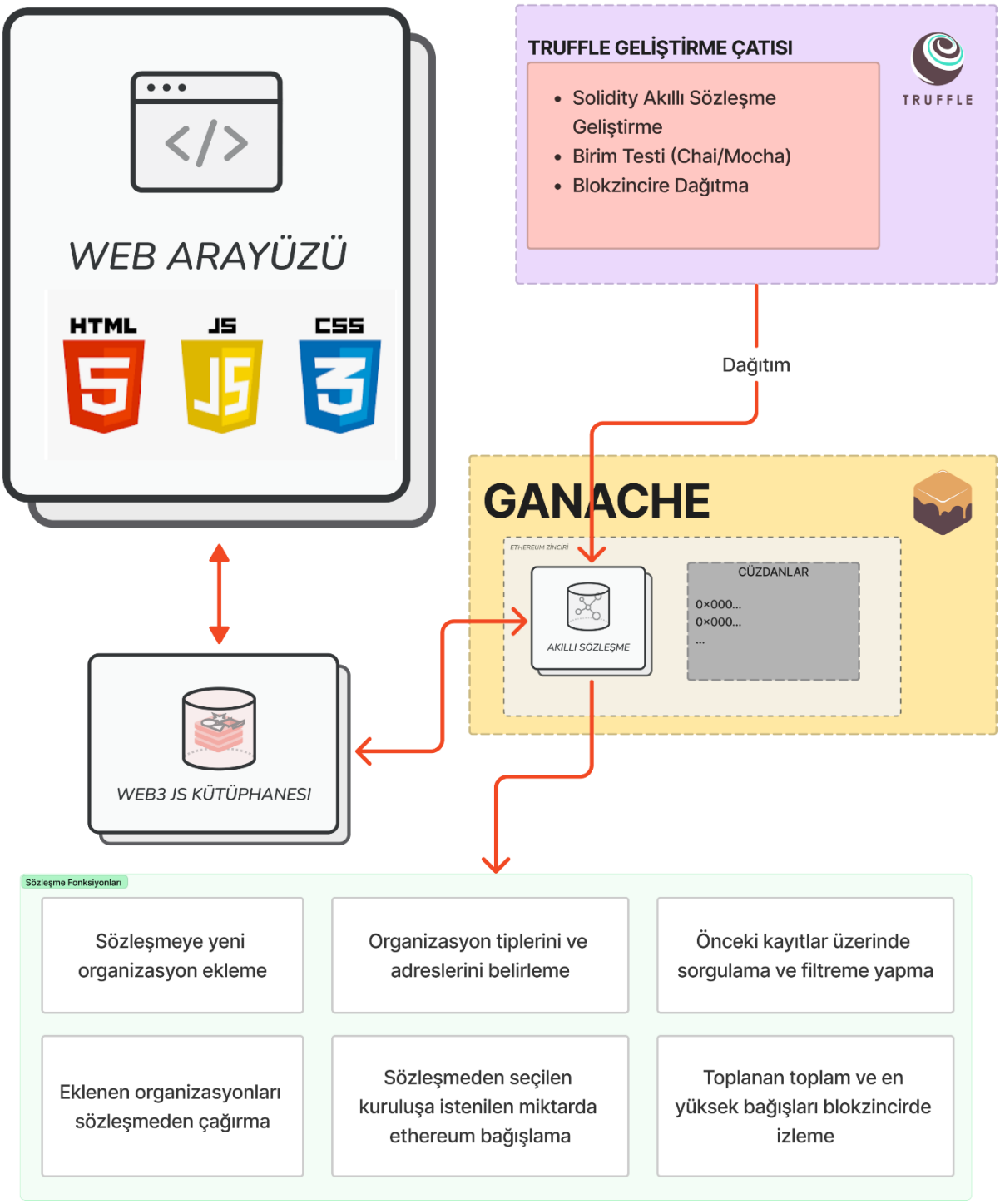
• Bireysel cüzdanın akıllı sözleşmede bir kuruluş olarak kayıtlı olup olmadığını kontrol etme

• Anonim cüzdanlardan yalnızca akıllı sözleşmeye kaydolan bir veya daha fazla kuruluşa doğrudan bağışlar.

- Verici ve alıcı adreslerinin farklı olup olmadığının kontrol edilmesi
- Bağışlanan miktar için cüzdandaki mevcut bakiyeyi kontrol etme
- Aynı anda birden fazla adrese yapılan bağışların yüzde dağılımı

• Yapılan tüm bağışların miktarını sorgulama ve filtreleme

• Blokzincirindeki işlem bloklarında akıllı sözleşme bağışlarının ve kayıtlı kuruluşların olay olarak kaydedilmesi

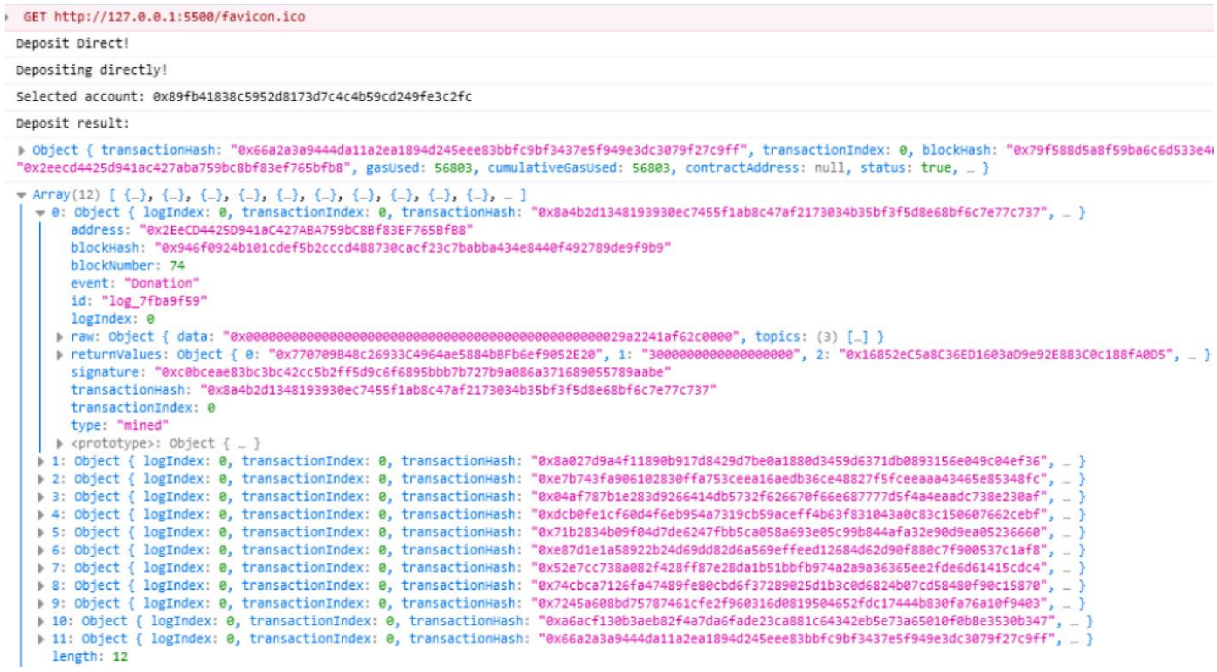


Şekil 5.1. Ethereum akıllı sözleşme tabanlı uygulama geliştirme şeması

Uygulamada kullanılan yöntemlerin ve mimarinin amacı, akıllı sözleşmenin doğrulayıcı ve aracı rolü, sözleşme kapsamındaki adreslerin doğrulanması, kayıtlı kuruluşlar dışında işlemlerin önlenmesi ve oransal dağılımın kontrolü gibi temel işlevler akıllı sözleşmeye dayalı bağışların oranı yukarıda açıklanmıştır. Uygulamada güvenlik açısından oluşabilecek tüm güvenlik açıkları akıllı sözleşme ile engellenmeye çalışılmıştır. Bu nedenle, kullanıcı arayüzündeki güvenlik kontrolleri en aza indirilir.

Geçerli veya geçersiz tüm işlemler, akıllı sözleşme aracılığıyla blokzincirine kaydedilir. Dağıtılmış defterdeki bu işlemler geri alınamaz ve silinemez olduğundan güvenlik üst seviyededir.

Blokzincirindeki dağıtılmış defterdeki bağış kayıtlarının bir örneği Şekil 5.2'de gösterilmektedir. Gerçekleşen olay bir bağış, para transferi veya sorgulama vb. bir işlem olabilir. Bir bağış olayı gerçekleştiğinde, blok numarası, işlem hash değeri, indeks değeri, bağışçının adresi, dijital imza kaydı, bağış yapan kurumun adresi, blok hash değeri, bir önceki bloğun hash değeri ve işlemin teyidi ağ (tip) dağıtılmış deftere kaydedilir. Ayrıca akıllı sözleşmeye göre hesaplanan Gas ücret bilgisi de Ethereum ağında saklanmaktadır. Dağıtılmış defterin blokzincirinde değişmezliğini sağlayan en önemli kısım, blok hash değeri ve bir önceki bloğun hash değeridir.



Şekil 5.2. Dağıtık defterdeki bağış kayıtları

Hayır amaçlı bağışlarla ilgili etkinlik oluşturma ve bağış dağıtımı için gerçekleştirilen algoritmanın genel yapısı Tablo 5.2'de gösterilmektedir. Akıllı sözleşmedeki fonksiyonların genel çalışma şeması verilmiş olup tüm kodları içermemektedir.

Tablo 5.2. Akıllı sözleşmede event oluşturma ve bağış dağıtımı

Initialize Owner
Initialize Enum Organization Type
Education, //0
Health, //1
Environment, //2
Religion, //3
CivilSociety, //4
International, //5
BigInternational, //6
GovernmentOrganized //7
Initialize Struct Organization
address payable organizationAddress;
string organizationName;
OrganizationType[] organizationTypes;
Initialize address payable[] charityAddresses;
Initialize uint256 totalDonationsAmount;
Initialize uint256 highestDonation;
Initialize address highestDonor;
Creating Events
event Donation(address indexed _donor, uint256 _value, address indexed _destinationAddress)
event OrganizationAdded(Organization indexed _organization)
Setting Up Mapping
mapping (address => bool) isManagerAddress
mapping (address => Organization) charityAddressInfos
Setting up functions
Func(addNewOrganization**)**
Input: string memory organizationName, OrganizationType[] memory organizationTypes
Require: charityAddresses[] not contains [msg.sender](address)
charityAddressInfos[[msg.sender]] = Organization({
organizationAddress: payable([msg.sender]),
organizationName: organizationName,
organizationTypes: organizationTypes,
isValidated: false
});
charityAddresses.push(payable([msg.sender]));
End Func
Func(getAddresses**)**
Return charityAddresses
End Func
Func(getAddressInfos**)**
Input: address payable charityAddress
Return ('Organization' struct values of input address)
End Func

Func(deposit**)**
Type: Payable
Input: address payable destinationAddress, address payable[] memory otherAddresses, uint256 mainPercentage
Require: [msg.sender] != destinationAddress
Require: transferAmount > 0
Require: charityAddresses[] contains destinationAddress
Require: charityAddresses[] contains otherAddresses
Require: check mainPercentage is between 0 to 100
uint256 donationAmount = ([msg.value] * mainPercentage) / 100 ;
uint256 actualDeposit = [msg.value] - donationAmount;
uint256 otherAddressAmount = actualDeposit / otherAddresses.length;
destinationAddress.transfer(donationAmount)
Charity Distribution
for(otherAddresses[i].transfer(otherAddressAmount))
totalDonationsAmount += donationAmount
if donationAmount > highestDonation
then
    highestDonation = donationAmount;
    highestDonor = [msg.sender];
End Func
Func(depositDirect**)**
Type: Payable
Input: address payable destinationAddress
Require: [msg.sender] != destinationAddress
Require: transferAmount > 0
Require: charityAddresses[] contains destinationAddress
uint256 donationAmount = [msg.value];
destinationAddress.transfer(donationAmount);
totalDonationsAmount += donationAmount;
if donationAmount > highestDonation
then
    highestDonation = donationAmount;
    highestDonor = [msg.sender];
End Func
Func(getTotalDonationsAmount**)**
Return totalDonationsAmount
End Func
Func(getHighestDonation**)**
Return highestDonation, highestDonor
End Func
Func(Destroy**)**
Require: Restrict To Owner Of Contract
Self destruct of contract
End Func
End Smart Contract

5.3.Sonuçlar ve Tartışma

Tablo 5.3'de akıllı sözleşmeye bağlı olarak bir donörün belirli sayıdaki adrese yaptığı bağışa karşı oluşan tahmini minimum Gas ücretleri görülmektedir. Tasarlanan sistemde bağış yapılan tutar Gas ücretini etkilememektedir. Ethereum ağında Gas ücretini etkileyen faktörlerden başka faktörler de bulunmaktadır. Bahşiş, taban ücret ve ücret artışı durumları göz ardı edilerek tahmini bir hesap yapılmıştır. Bu sayede çoklu bağış yapıldığında akıllı sözleşmedeki şartlara göre oluşan Gas ücreti birbirine göre kıyaslanmaktadır.

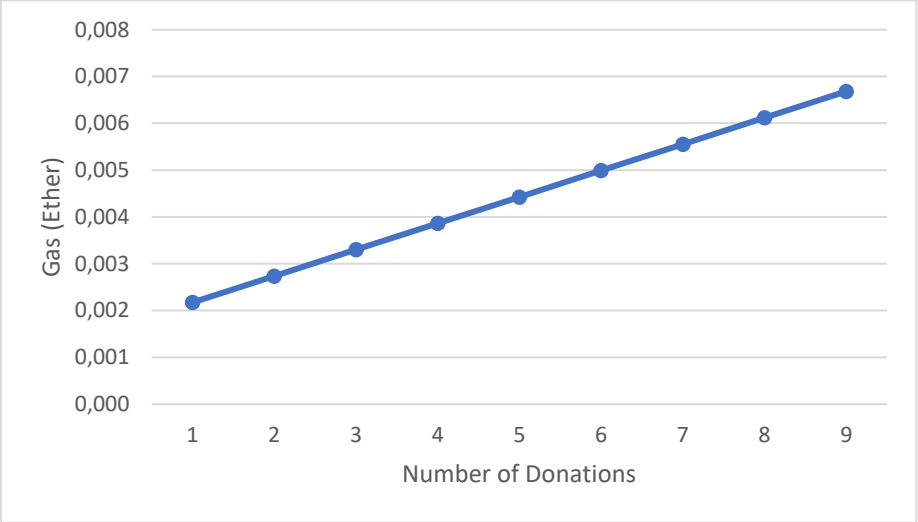
Tablo 5.3. Bağışlara göre oluşan Ethereum ücretleri

Bağış Yapan Adres	Bağış Yapılan Tutar (ether)	Bağış Yapılan Adres Sayısı	GAS ücreti (gwei)	GAS ücreti (ether)
0x770709b48c26933c4964ae5884bbfb6ef9052e20	0,05	1	2172880	0,00217288
	0,05	2	2736380	0,00273638
	0,05	3	3299880	0,00329988
	0,05	4	3863380	0,00386338
	0,05	5	4426880	0,00442688
	0,05	6	4990380	0,00499038
	0,05	7	5553880	0,00555388
	0,05	8	6117380	0,00611738
	0,05	9	6680880	0,00668088

Gas ücreti Etherum'un bir birimi olan gwei cinsinden belirtilir ve ether birimine dönüştürülebilir. Tablo 5.3'de her ikisi de mevcuttur. Şekil 5.3'de aynı anda bağış yapılan adres sayısına bağlı olarak değişen Gas ücretinin grafiği görülmektedir. Bağış yapılan adres sayısı artırıldıkça yaklaşık 0,0005635 ether miktarında doğrusal bir artış olmaktadır. Sistemin tasarımı akıllı sözleşme ile çoklu bağışa uygun tasarlandığı için donör bir kerede istediği birden fazla kuruma bağış yapabilmektedir. Tek tek bağış yapılan durumlarda minimum Gas ücreti her seferinde tedarik edileceğinden dolayı işlem ücreti artacaktır. Gas fiyatı ağın talebine, blok boyutuna, gas limitine ve işlem önceliğine (ağdaki yüksek ücretli işlem sayısına) bağlı olarak değişir. Tablo 5.3'de belirlenen ücretler sistemin belirli bir durumuna ait olup güncel ücretler değildir. Ayrıca uygulamada kullanılan konsensüs algoritması da Gas ücretini etkilemektedir.

Ethereum iyi bilinen, istikrarlı ve iyi işleyen bir mekanizmaya sahip olmasından dolayı benzer çalışmaların %90'ından fazlasında tercih sebebi olmaktadır. Hyperledger veya benzeri bir akıllı sözleşme tabanlı blokzincir mekanizmasıyla da bu çalışma

gerçekleştirilebilir. Fakat Hyperledger'in yapısında bir token bulunmaması, private olması, fikir birliği algoritmasının farklılığı gibi sebepler bu tarz bir sistemin altyapısını geliştirmeyi zorlaştırmaktadır. Ethereum'un kullanılmasındaki en büyük etken akıllı sözleşmeler, güvenlik, kolay entegrasyon ve birçok uygulamada kullanılarak kendini ispatlamış olmasıdır. Saniye başına gerçekleşen işlem sayısı ve işlem ücretleri Ethereum'un dezavantajı olarak söylenebilir.



Şekil 5.3. Bağış sayısına bağlı değişen Gas ücreti

Solidity kullanan şeffaf bağış sistemleri hem bağışçılara hem de alıcılara birtakım avantajlar sağlayabilir. Bağışçıların fonlarının kullanımını izlemelerine ve bağışlarının nasıl bir fark yarattığını görmelerine olanak tanıyan şeffaf bağış sistemleri, hayırseverlik sektöründe güven ve itimat oluşturabilir. Aynı zamanda, şeffaf bağış sistemleri, fonlarının nasıl kullanıldığını ve sahip oldukları etkiyi göstererek alıcıların sorumluluğu artırmasına ve bağışçılarla güven inşa etmesine yardımcı olabilir. Ancak, şeffaf bağış sistemlerinin zorluklarının da olduğunu unutmamak önemlidir. Ana zorluklardan biri, akıllı sözleşmeye dayalı bir sistemin uygulanmasının maliyeti ve karmaşıklığıdır. Bir akıllı sözleşme oluşturmak ve dağıtmak, özel bilgi ve kaynaklar gerektirir ve sistemin güvenliğini ve bütünlüğünü tehlikeye atabilecek hata veya güvenlik açıkları riski vardır. Diğer bir zorluk da şeffaf bağış sistemlerinin ölçeklenebilirliğidir. Giderek daha fazla insan sistemi kullanmaya başladıkça, işlem sayısı ve işlenen veriler önemli ölçüde artabilir ve bu da ölçeklenebilirlik sorunlarına yol açabilir. Bu, özellikle Ethereum gibi sınırlı kapasiteye sahip ve yüksek talep zamanlarında tıkanabilen halka açık blokzincir platformları için sorunlu olabilir.

Bu zorluklara rağmen, Solidity kullanan şeffaf bağış sistemleri, hayır amaçlı

bağışların yapılma ve takip edilme biçiminde önemli bir potansiyele sahiptir. Şeffaflığı ve hesap verebilirliği artırarak, hayırseverlik sektöründe güven ve güven oluşturmaya yardımcı olabilirler, bu da sonuçta önemli amaçlar için daha fazla destek ve finansmana yol açabilir.

6. SONUÇLAR

Bu çalışmada, iki farklı alan için blokzincir teknolojisi ve şifreleme yöntemlerinin kullanımına yönelik iki önemli çalışma gerçekleştirilmiştir. HFSecImg ve Ethtrace çalışmaları, güncel teknolojilerin uygulanması ve bu alanlarda yapılan araştırmaların önemini vurgulamaktadır.

HFSecImg, Shamir's sır paylaşım yöntemi ve kaotik sistem tabanlı şifreleme yöntemi kullanarak şifrelenen bir görüntünün özet bilgisinin ve paydaşların RSA şifrelemesi ile oluşturulan dijital imzalarının Hyperledger Fabric tabanlı blokzincirde saklanmasını sağlamıştır. Ayrıca, bu yapıda paydaşların imzalarının ve şifrelenen görüntünün veri doğrulaması chaincode vasıtasıyla gerçekleştirilmiştir. Bu çalışma, güvenli veri saklama ve paylaşımı konusunda önemli bir katkı sağlamaktadır.

Ethtrace, şeffaf bağış uygulamalarının blokzincir teknolojisi ile nasıl uygulanabileceğini ve Ethereum tabanlı akıllı sözleşmelerin bu alanda nasıl kullanılabileceğini gösteren örnek bir uygulama sunmaktadır. Bu çalışma, bağış uygulamaları ve blokzincir teknolojisi arasındaki birlikteliği ortaya koyması açısından literatüre önemli bir katkı sağlamaktadır. Ayrıca, aynı anda birden fazla kuruluşa bağış yapılabilmesine olanak sağlayarak daha geniş bir bağışçı kitlesine hitap etmekte ve bağışların izlenebilirliğini artırmaktadır.

Her iki çalışma da blokzincir teknolojisi ve şifreleme yöntemleri gibi güncel konulara odaklanarak bu alanlarda önemli birer katkı sağlamışlardır. Bu çalışmaların sonuçları ilgili alanlarda daha fazla araştırma yapılmasına ve bu teknolojilerin daha yaygın bir şekilde kullanılmasına katkı sağlayacaktır. Bu nedenle gelecekteki çalışmaların bu alanlarda yapılan araştırmaları ve uygulamaları daha da geliştirmesi ve genişletmesi düşünülmektedir.

HFSecImg ve Ethtrace arasındaki farkları ve benzerlikleri gösteren Tablo 6.1’de sonuçlar açık bir şekilde ortaya konulmuştur.

Tablo 6.1. HFSecImg ve Ethtrace'e ait sonuçların değerlendirmesi

Özellik	HFSecImg	Ethtrace
Ana Tema	Güvenli veri saklama ve paylaşımı	Şeffaf bağış uygulamaları
Şifreleme Yöntemi	Shamir'in sır paylaşım şeması ve kaotik sistem tabanlı şifreleme	Ethereum tabanlı akıllı sözleşmelerde bulunan mekanizmalar
Blokzincir Platformu	Hyperledger Fabric	Ethereum
Veri Saklama	Blokzincirde özet bilgi ve dijital imzalar, bulut sisteminde görüntü	Blokzincirde bağış bilgileri ve akıllı sözleşmeler
Veri Doğrulama	Chaincode vasıtasıyla	Akıllı sözleşmeler vasıtasıyla
Kullanıcı Katılımı	Paydaşlar	Bağışçılar ve kuruluşlar
Hedeflenen Fayda	Güvenli ve doğrulanabilir veri saklama ve paylaşımı	Şeffaf, izlenebilir ve birden fazla kuruluşa bağış yapılabilen bağış uygulamaları
Literatüre Katkı	Güvenli veri saklama ve paylaşımı alanında yeni bir yapı sunması	Bağış uygulamaları ve blokzincir teknolojisi arasındaki birlikteliği ortaya koyması
Uygulama Alanı	Görüntü şifreleme ve paylaşımı	Bağış ve yardım sektörü
Güvenlik Protokolleri	Shamir'in sır paylaşım şeması, kaotik sistem tabanlı şifreleme, RSA şifrelemesi	Ethereum tabanlı akıllı sözleşmeler
Performans ve Ölçeklenebilirlik	Hyperledger Fabric'in ölçeklenebilirliği ve performansı	Ethereum'un ölçeklenebilirliği ve performansı (gelişmelerle birlikte)
Kullanıcı Arayüzü	Paydaşlar için özet bilgi ve dijital imza erişimi	Bağışçılar için bağış bilgileri ve bağış izleme
Veri Gizliliği	Görüntü verilerinin şifrelenmesi ve paydaşların imzalarının saklanması	Bağışçıların ve bağış alıcı kuruluşların bilgilerinin gizliliği (isteğe bağlı)
İşlem Maliyeti	Hyperledger Fabric işlem maliyetleri	Ethereum işlem maliyetleri (Gas ücretleri)

KAYNAKLAR

Abdelfatah, R. I. (2020). A new fast double-chaotic based Image encryption scheme. *Multimedia Tools and Applications*, *79*(1-2), 1241-1259.

Abrar, A., Abdul, W., & Ghouzali, S. (2021). Secure Image Authentication Using Watermarking and Blockchain. *Intelligent Automation & Soft Computing*, *28*(2).

Alassaf, A. O. A. K., & Yusoff, F. H. (2021). Multi-point Fundraising and Distribution via Blockchain. *International Journal of Advanced Computer Science and Applications*, *12*(7).

Alqaralleh, B. A., Vaiyapuri, T., Parvathy, V. S., Gupta, D., Khanna, A., & Shankar, K. (2021). Blockchain-assisted secure image transmission and diagnosis model on Internet of Medical Things Environment. *Personal and ubiquitous computing*, 1-11.

Alsunaidi, S. J., & Alhaidari, F. A. (2019). A survey of consensus algorithms for blockchain technology. *2019 International Conference on Computer and Information Sciences (ICCIS)*.

Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., Enyeart, D., Ferris, C., Laventman, G., & Manevich, Y. (2018). Hyperledger fabric: a distributed operating system for permissioned blockchains. *Proceedings of the thirteenth EuroSys conference*.

Bach, L. M., Mihaljevic, B., & Zagar, M. (2018). Comparative analysis of blockchain consensus algorithms. *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*.

Bader, L., Bürger, J. C., Matzutt, R., & Wehrle, K. (2018). Smart contract-based car insurance policies. *2018 IEEE Globecom workshops (GC wkshps)*.

Baliga, A. (2017). Understanding blockchain consensus models. *Persistent*, *4*(1), 14.

Banerjee, A., Clear, M., & Tewari, H. (2020). Demystifying the Role of zk-SNARKs in Zcash. *2020 IEEE conference on application, information and network security (AINS)*.

Biswas, S., Sharif, K., Li, F., Nour, B., & Wang, Y. (2018). A scalable blockchain framework for secure transactions in IoT. *IEEE Internet of Things Journal*, *6*(3), 4650-4659.

Bitshares. (2019). *Bitshares Documentation.* [Erişim: 15.06.2019, https://how.bitshares.works/en/master/technology/dpos.html].

Boroń, M., & Kobusińska, A. (2021). Alternative authentication with smart contracts for online games. *2021 IEEE 46th Conference on Local Computer Networks (LCN)*.

Brown, R. G. (2018). The corda platform: An introduction. *Retrieved, 27*, 2018.

Cachin, C. (2016). Architecture of the hyperledger blockchain fabric. *Workshop on distributed cryptocurrencies and consensus ledgers*.

Chavan, S., Warke, P., Ghuge, S., & Deolekar, R. V. (2019). Music streaming application using blockchain. *2019 6th international conference on computing for sustainable global development (INDIACom)*.

Chen, G., Mao, Y., & Chui, C. K. (2004). A symmetric image encryption scheme based on 3D chaotic cat maps. *Chaos, Solitons & Fractals, 21*(3), 749-761.

Chi, J., Li, Y., Huang, J., Liu, J., Jin, Y., Chen, C., & Qiu, T. (2020). A secure and efficient data sharing scheme based on blockchain in industrial Internet of Things. *Journal of Network and Computer Applications, 167*, 102710.

Chum, C. S., Fine, B., & Zhang, X. (2018). A Survey: Shamir Threshold Scheme and Its Enhancements *Infinite Group Theory: From the Past to the Future* (pp. 19-41): World Scientific.

Conti, M., Kumar, E. S., Lal, C., & Ruj, S. (2018). A survey on security and privacy issues of bitcoin. *IEEE Communications Surveys & Tutorials, 20*(4), 3416-3452.

Dai, F., Shi, Y., Meng, N., Wei, L., & Ye, Z. (2017). From Bitcoin to cybersecurity: A comparative study of blockchain application and security issues. *2017 4th International Conference on Systems and Informatics (ICSAI)*.

Esposito, C., De Santis, A., Tortora, G., Chang, H., & Choo, K. K. R. (2018). Blockchain: A Panacea for Healthcare Cloud-Based Data Security and Privacy? *Ieee Cloud Computing, 5*(1), 31-37.

Ethereum. (2015). *Ethereum Whitepaper*. [Erişim: 21.06.2021, https://ethereum.org/en/whitepaper/].

Eyal, I., Gencer, A. E., Sirer, E. G., van Renesse, R., & Assoc, U. (2016, Mar 16-18). Bitcoin-NG: A Scalable Blockchain Protocol. *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, Santa Clara, CA.

Farooq, M. S., Khan, M., & Abid, A. (2020). A framework to make charity collection transparent and auditable using blockchain technology. *Computers & Electrical Engineering, 83*, 106588.

Feixiang, Z., Mingzhe, L., Kun, W., & Hong, Z. (2021). Color image encryption via Hénon-zigzag map and chaotic restricted Boltzmann machine over Blockchain. *Optics & Laser Technology, 135*, 106610.

Gao, X., Mou, J., Xiong, L., Sha, Y., Yan, H., & Cao, Y. (2022). A fast and efficient multiple images encryption based on single-channel encryption and chaotic system. *Nonlinear Dynamics, 108*(1), 613-636.

Garzik, J., & Donnelly, J. C. (2018). Blockchain 101: an introduction to the future *Handbook of Blockchain, Digital Finance, and Inclusion, Volume 2* (pp. 179-186): Elsevier.

Gennaro, R., Goldfeder, S., & Narayanan, A. (2016). Threshold-optimal DSA/ECDSA signatures and an application to bitcoin wallet security. *Applied Cryptography and Network Security: 14th International Conference, ACNS 2016, Guildford, UK, June 19-22, 2016. Proceedings 14*.

Gil-Alana, L. A., Abakah, E. J. A., & Rojo, M. F. R. (2020). Cryptocurrencies and stock market indices. Are they related? *Research in International Business and Finance, 51*, 101063.

Gil, S. K. (2020). Proposal for Analog Signature Scheme Based on RSA Digital Signature Algorithm and Phase-shifting Digital Holography. *Current Optics and Photonics, 4*(6), 483-499.

Goldfeder, S., Bonneau, J., Kroll, J., & Felten, E. (2014). Securing bitcoin wallets via threshold signatures.

Halpin, H., & Piekarska, M. (2017). Introduction to Security and Privacy on the Blockchain. *2017 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*.

Hawashin, D., Mahboobeh, D. A. J., Salah, K., Jayaraman, R., Yaqoob, I., Debe, M., & Ellahham, S. (2021). Blockchain-based management of blood donation. *Ieee Access, 9*, 163016-163032.

Helo, P., & Hao, Y. (2019). Blockchains in operations and supply chains: A model and reference implementation. *Computers & Industrial Engineering, 136*, 242-251.

Henry, R., Herzberg, A., & Kate, A. (2018). Blockchain Access Privacy: Challenges and Directions. *Ieee Security & Privacy, 16*(4), 38-45. doi:10.1109/msp.2018.3111245

HyperledgerFabric. (2015). *Hyperledger Fabric Tutorials.* [Erişim: 11.06.2022, https://hyperledger-fabric.readthedocs.io/en/latest/tutorials.html].

Iansiti, M., & Lakhani, K. (2017). The Truth about Blockchain. Harvard Business Review. Harvard University. *Retrieved, 27*(9).

Iqbal, N., Jamil, F., Ahmad, S., & Kim, D. (2021). A novel blockchain-based integrity and reliable veterinary clinic information management system using predictive analytics for provisioning of quality health services. *Ieee Access, 9*, 8069-8098.

Khan, P. W., Byun, Y.-C., & Park, N. (2020). A data verification system for CCTV surveillance cameras using blockchain technology in smart cities. *Electronics, 9*(3), 484.

Khan, P. W., & Byun, Y. (2020). A blockchain-based secure image encryption scheme for the industrial Internet of Things. *Entropy, 22*(2), 175.

Kopp, H., Mödinger, D., Hauck, F., Kargl, F., & Bösch, C. (2017). Design of a privacy-preserving decentralized file storage with financial incentives. *2017 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*.

Kosba, A., Miller, A., Shi, E., Wen, Z., & Papamanthou, C. (2016). Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. *2016 IEEE symposium on security and privacy (SP)*.

Krichen, M., Ammi, M., Mihoub, A., & Almutiq, M. (2022). Blockchain for modern applications: A survey. *Sensors, 22*(14), 5274.

Kwon, J. (2014). Tendermint: Consensus without mining. *Draft v. 0.6, fall, 1*(11).

Lee, J., Seo, A., Kim, Y., & Jeong, J. (2018). Blockchain-based one-off address system to guarantee transparency and privacy for a sustainable donation environment. *Sustainability, 10*(12), 4422.

Lei, K., Zhang, Q., Xu, L., & Qi, Z. (2018). Reputation-based byzantine fault-tolerance for consortium blockchain. *2018 IEEE 24th international conference on parallel and distributed systems (ICPADS)*.

Lewko, A., & Waters, B. (2011). Decentralizing attribute-based encryption. *Advances in Cryptology–EUROCRYPT 2011: 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings 30*.

Li, D., Peng, W., Deng, W., & Gai, F. (2018). A blockchain-based authentication and security mechanism for IoT. *2018 27th International Conference on Computer Communication and Networks (ICCCN)*.

Li, R. (2021). Fingerprint-related chaotic image encryption scheme based on blockchain framework. *Multimedia Tools and Applications, 80*, 30583-30603.

Li, X., Li, J., Yu, F., Fu, X., Yang, J., & Chen, Y. (2021). BEIR: A Blockchain-based Encrypted Image Retrieval Scheme. *2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*.

Manevich, Y., Barger, A., & Tock, Y. (2019). Endorsement in Hyperledger Fabric via service discovery. *IBM Journal of Research and Development, 63*(2/3), 2: 1-2: 9.

Mazieres, D. (2015). The stellar consensus protocol: A federated model for internet-level consensus. *Stellar Development Foundation, 32*, 1-45.

Miller, A. K., Möser, M., Lee, K., & Narayanan, A. (2017). An Empirical Analysis of Linkability in the Monero Blockchain. *ArXiv, abs/1704.04299*.

Mohite, A., & Acharya, A. (2018). Blockchain for government fund tracking using Hyperledger. *2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS)*.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. *Decentralized business review*, 21260.

Naz, M., Al-zahrani, F. A., Khalid, R., Javaid, N., Qamar, A. M., Afzal, M. K., & Shafiq, M. (2019). A secure data sharing platform using blockchain and interplanetary file system. *Sustainability, 11*(24), 7054.

NxtCommunity. (2014). *Nxt Whitepaper*. [Erişim: 21.06.2021, https://www.jelurida.com/].

Oliva, G. A., Hassan, A. E., & Jiang, Z. M. (2020). An exploratory study of smart contracts in the Ethereum blockchain platform. *Empirical Software Engineering, 25*, 1864-1904.

Perboli, G., Musso, S., & Rosano, M. (2018). Blockchain in logistics and supply chain: A lean approach for designing real-world use cases. *Ieee Access, 6*, 62018-62028.

Rahulamathavan, Y., Phan, R. C.-W., Rajarajan, M., Misra, S., & Kondoz, A. (2017). Privacy-preserving blockchain based IoT ecosystem using attribute-based encryption. *2017*

IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS).

Raman, R. K., & Varshney, L. R. (2018). Distributed storage meets secret sharing on the blockchain. *2018 information theory and applications workshop (ITA)*.

Sadri, S., Shahzad, A., & Zhang, K. (2021). Blockchain traceability in healthcare: Blood donation supply chain. *2021 23rd International Conference on Advanced Communication Technology (ICACT)*.

Saleh, H., Avdoshin, S., & Dzhonov, A. (2019). Platform for tracking donations of charitable foundations based on blockchain technology. *2019 Actual Problems of Systems and Software Engineering (APSSE)*.

Sarıtekin, R. A., Karabacak, E., Durgay, Z., & Karaarslan, E. (2018). Blockchain based secure communication application proposal: Cryptouch. *2018 6th International Symposium on Digital Forensic and Security (ISDFS)*.

Sasson, E. B., Chiesa, A., Garman, C., Green, M., Miers, I., Tromer, E., & Virza, M. (2014). Zerocash: Decentralized anonymous payments from bitcoin. *2014 IEEE symposium on security and privacy*.

Schwartz, D., Youngs, N., & Britto, A. (2014). The ripple protocol consensus algorithm. *Ripple Labs Inc White Paper, 5*(8), 151.

Shaheen, E., Hamed, M. A., Zaghloul, W., Al Mostafa, E., El Sharkawy, A., Mahmoud, A., Labeb, A., Al Enany, M. O., & Attiya, G. (2021). A track donation system using blockchain. *2021 international conference on electronic engineering (ICEEM)*.

Shamir, A. (1979). How to share a secret. *Communications of the ACM, 22*(11), 612-613.

Shen, M., Deng, Y., Zhu, L., Du, X., & Guizani, N. (2019). Privacy-preserving image retrieval for medical IoT systems: A blockchain-based approach. *IEEE Network, 33*(5), 27-33.

Singh, A., Rajak, R., Mistry, H., & Raut, P. (2020). Aid, charity and donation tracking system using blockchain. *2020 4th international conference on trends in electronics and informatics (ICOEI)(48184)*.

Singh, A. P., Pradhan, N. R., Luhach, A. K., Agnihotri, S., Jhanjhi, N. Z., Verma, S., Ghosh, U., & Roy, D. S. (2020). A novel patient-centric architectural framework for

blockchain-enabled healthcare applications. *IEEE Transactions on Industrial Informatics, 17*(8), 5779-5789.

Steemit. (2016). *Steemit Blog.* [Erişim: 13.06.2019, https://steemit.com/@steemitblog].

Sun, S.-F., Au, M. H., Liu, J. K., & Yuen, T. H. (2017). Ringct 2.0: A compact accumulator-based (linkable ring signature) protocol for blockchain cryptocurrency monero. *Computer Security–ESORICS 2017: 22nd European Symposium on Research in Computer Security, Oslo, Norway, September 11-15, 2017, Proceedings, Part II 22*.

Süzen, A. A., & Duman, B. (2021). Blockchain-Based Secure Credit Card Storage System for E-Commerce. *Sakarya University Journal of Computer And Information Sciences, 4*(2), 204-215.

Tanrıverdi, M. (2020). Design and Implementation of Blockchain Based Single Sign-On Authentication System for Web Applications. *Sakarya University Journal of Computer And Information Sciences, 3*(3), 343-354.

Tanwar, S., Parekh, K., & Evans, R. (2020). Blockchain-based electronic healthcare record system for healthcare 4.0 applications. *Journal of Information Security and Applications, 50*, 102407.

TronCommunity. (2018). *Tron Dao.* [Erişim: 21.06.2021, https://trondao.org/community/home/].

Tunçer, S., & Karakuzu, C. (2022). Performance Analysis of Chaotic Neural Network and Chaotic Cat Map Based Image Encryption. *Journal Of Computer And Information Sciences*, 37.

Tunçer, S., Özdede, A., & Karakuzu, C. (2022). Transparent Donation Management with Smart Contract-Based Blockchain. *3*(2), 8-15.

Van Saberhagen, N. (2013). CryptoNote v 2.0.

Vukolić, M. (2016). The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication. *Open Problems in Network Security: IFIP WG 11.4 International Workshop, iNetSec 2015, Zurich, Switzerland, October 29, 2015, Revised Selected Papers*.

Wood, G. (2014). Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper, 151*(2014), 1-32.

Wu, H., & Zhu, X. (2020). Developing a reliable service system of charity donation during the covid-19 outbreak. *Ieee Access, 8*, 154848-154860.

Wu, Y., Noonan, J. P., & Agaian, S. (2011). NPCR and UACI randomness tests for image encryption. *Cyber journals: multidisciplinary journals in science and technology, Journal of Selected Areas in Telecommunications (JSAT), 1*(2), 31-38.

Xu, B., Luthra, D., Cole, Z., & Blakely, N. (2018). EOS: An architectural, performance, and economic analysis. *Retrieved June, 11*, 2019.

Yamashita, K., Nomura, Y., Zhou, E., Pi, B., & Jun, S. (2019). Potential risks of hyperledger fabric smart contracts. *2019 IEEE International Workshop on Blockchain Oriented Software Engineering (IWBOSE)*.

Yavuz, E., Koç, A. K., Çabuk, U. C., & Dalkılıç, G. (2018). Towards secure e-voting using ethereum blockchain. *2018 6th International Symposium on Digital Forensic and Security (ISDFS)*.

Zhang, W. B., Yuan, Y., Hu, Y. Y., Huang, S. H., Cao, S. J., Chopra, A., Huang, S., & Ieee. (2018, Jul 02-07). A Privacy-Preserving Voting Protocol on Blockchain. *11th IEEE International Conference on Cloud Computing (CLOUD) Part of the IEEE World Congress on Services*, San Francisco, CA.

Zhang, Y.-l., Wen, L., Zhang, Y.-j., & Wang, C.-f. (2020). Deniably authenticated searchable encryption scheme based on Blockchain for medical image data sharing. *Multimedia Tools and Applications, 79*, 27075-27090.

Zheng, X., Mukkamala, R. R., Vatrapu, R., & Ordieres-Mere, J. (2018). Blockchain-based personal health data sharing system using cloud storage. *2018 IEEE 20th international conference on e-health networking, applications and services (Healthcom)*.

Zheng, Z., Xie, S., Dai, H.-N., Chen, X., & Wang, H. (2018). Blockchain challenges and opportunities: A survey. *International journal of web and grid services, 14*(4), 352-375.

Zheng, Z., Xie, S., Dai, H., Chen, X., & Wang, H. (2017). An overview of blockchain technology: Architecture, consensus, and future trends. *2017 IEEE international congress on big data (BigData congress)*.