T.C.
HARRAN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

BLOK ZİNCİRİ İLE AKILLI SÖZLEŞME SİMÜLASYONU

Ahmet TABANLIOĞLU

ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

ŞANLIURFA
2019

T.C.
HARRAN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

BLOK ZİNCİRİ İLE AKILLI SÖZLEŞME SİMÜLASYONU

Ahmet TABANLIOĞLU

ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

ŞANLIURFA
2019

Dr. Öğr. Üyesi Mehmet Emin TENEKECİ danışmanlığında, Ahmet TABANLIOĞLU'nun hazırladığı "**Blok Zinciri İle Akıllı Sözleşme Simülasyonu**" konulu bu çalışma 27/08/2019 tarihinde aşağıdaki jüri tarafından oy birliği ile Harran Üniversitesi Fen Bilimler Enstitüsü Elektrik Elektronik Mühendisliği Anabilim Dalı'nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

İmza

Danışman: Dr. Öğr. Üyesi Mehmet Emin TENEKECİ ……

Üye: Dr. Öğr. Üyesi Abdülkadir GÜMÜŞÇÜ ……

Üye: Dr. Öğr. Üyesi Faruk KÜRKER ……

Bu Tezin Elektrik Elektronik Mühendisliği Anabilim Dalında Yapıldığını ve Enstitümüz Kurallarına Göre Düzenlendiğini Onaylarım.

Doç. Dr. İsmail HİLALİ
Enstitü Müdürü

Not: Bu tezde kullanılan özgün ve başka kaynaktan yapılan bildirimlerin, çizelge, şekil ve fotoğrafların kaynak gösterilmeden kullanımı 5846 sayılı Fikir ve Sanat Eserleri Kanunundaki hükümlere tabidir.

İÇİNDEKİLER

Sayfa No

ÖZET i
ABSTRACT ii
TEŞEKKÜR iii
ŞEKİLLER DİZİNİ iv
ÇİZELGELER DİZİNİ v
SİMGELER VE KISALTMALAR DİZİNİ vi
1. GİRİŞ 1
2. ÖNCEKİ ÇALIŞMALAR 3
3. MATERYAL ve YÖNTEM 7
3.1. Blok Zinciri Teknolojisi Teknik Detayları 7
3.1.1. Defter-i kebir 7
3.1.2. Blok zinciri teknolojisi 10
3.1.3. Blok zinciri teknolojisi kullanım durumu 19
3.1.4. Kripto paralar ve madencilik 20
3.1.5. Ethereum ve ethereum sanal makinesi: 27
3.1.6. Akıllı kontratlar 28
3.2. Nesnelerin İnterneti 31
3.2.1. Nesnelerin interneti nedir? 31
3.2.2. Endüstriyel nesnelerin interneti nedir? (IIoT) 33
3.3. Blok Zincirinin Akıllı Kira Sözleşmesinde Kullanımı 36
3.4. Yapılan Çalışmalar 37
3.4.1. Web3.js 37
3.4.2. Truffle framework 39
3.4.3. Bootstrap framework 41
3.4.4. MetaMask 42
3.4.5. GETH 43
3.4.7. Teyit yazılımı 50
4. ARAŞTIRMA BULGULARI ve TARTIŞMA 59
5. SONUÇ ve ÖNERİLER 60
5.1. Sonuçlar 60
5.2. Öneriler 61
KAYNAKLAR 62
ÖZGEÇMİŞ 64

ÖZET

Yüksek Lisans Tezi

BLOK ZİNCİRİ İLE AKILLI SÖZLEŞME SİMÜLASYONU

Ahmet TABANLIOĞLU

Harran Üniversitesi
Fen Bilimleri Enstitüsü
Elektrik Elektronik Mühendisliği

Danışman: Dr. Öğr. Üyesi Mehmet Emin TENEKECİ
YIL: 2019, Sayfa: 64

Son yıllarda geleneksel veri tabanı yöntemine alternatif olarak önerilen blok zinciri bilgi teknolojisi ve iş dünyasının dikkatini çekmektedir. Özellikle blok zinciri ile geliştirilen Bitcoin'den insanların gelir elde etmesiyle farkına varılan blok zinciri teknoloji araştırmacıların ilgi odağı haline gelmiştir. Bu ilgi ile doğru orantılı olarak geleneksel veri tabanı ile problemlerini çözemeyen farklı disiplinler blok zincirine yönelmiştir. Temelde dağıtık veri tabanı (defter) teknolojisinin fikre bürünmüş yapısı olan blok zinciri, yapılan işlemleri tek merkezde değil işe dahil olan tüm cihazlarda sağlıyor olmalıdır. Şöyle ki; işlemi ya da işlemin yapıldığı nesnenin kaydını temsil eden veri yapının koruma altına alınarak bir hiyerarşiye göre dijital olarak saklanmasıdır. Bu tezde 2 temel bölüm bulunmaktadır. Bunlardan ilki blok zinciri, kripto paralar, madencilik, çalışma mantığını konu alacaktır. İkinci yani uygulama bölümünde ise lokalde üretilen bir blok zincirinde oluşturulan hesaplar, madencilik işlemi, kripto para transferi işlemleri incelenmesi ile birlikte gerçek kullanıcı cüzdanları ile Ropsten ağında para transferleri, transfer doğrulama işlemi neticesinde akıllı sözleşmeye konu olan parametrelerin kontrol edildiği ve şartların yine sözleşmede belirtildiği gibi nesnelerin interneti ile yerine getirilmesi sağlanmıştır.

ANAHTAR KELİMELER: Blok zinciri, dağıtılmış defter teknolojisi, kripto paralar

ABSTRACT

MSc Thesis

SMART CONTRACT SIMULATION WITH BLOCKCHAIN

Ahmet TABANLIOĞLU

Harran University
Graduate School of Natural and Applied Sciences
Department of Electrical and Electronic Engineering

Supervisor: Assist. Prof. Dr. Mehmet Emin TENEKECİ

Year: 2019, Page: 64

The blockchain proposed as an alternative to the traditional database method in recent years has attracted the attention of information technology and business world. Particularly, block chain technology, which has been recognized by people's income from Bitcoin developed with block chain, has become the focus of interest of technology researchers. In line with this interest, different disciplines that cannot solve their problems with the traditional database are directed to the block chain. The chain of blocks, which is basically the idea-crafted structure of distributed database (notebook) technology, should ensure that operations are performed not on a single center but on all devices involved. Namely; is the digital structure representing the operation or the registration of the object from which the operation is performed, and stored digitally according to a hierarchy. There are 2 basic sections in this thesis. The first of these will be about blockchain, cryptocurrencies, mining, working logic. In the second part of the application, the accounts created in a blockchain produced in the local area, mining operation, cryptocurrency transfer operations together with real user wallets and money transfers in the Ropsten network, as a result of the transfer verification process, the parameters that are subject to the smart contract are controlled and the conditions are fulfilled with the Internet of things

KEY WORDS: Blockchain, distributed ledger technologies, cryptocurrencies

TEŞEKKÜR

Bu tez çalışmasının başından sonuna kadar her aşamasında bana yardımcı olan, bilgi ve deneyimlerini titizlikle aktaran değerli danışman hocam Dr. Öğr. Üyesi Mehmet Emin TENEKECİ'ye, selef danışman hocam Dr. Öğr. Üyesi Mehmet Akif NACAR'a ve maddi manevi destekleri ile her zaman yanımda olan eşime ve aileme sonsuz teşekkürlerimi sunarım.

ŞEKİLLER DİZİNİ

Sayfa No

Şekil 3.1 Mimari modeller 8
Şekil 3.2 Mısır hiyerogliflerinden bir kesit 11
Şekil 3.3 Tekli yerine koyma metodu örneği 12
Şekil 3.4 Simetrik anahtarlı şifreleme şekli 12
Şekil 3.5 Simetrik anahtarlı şifreleme şekli 14
Şekil 3.6 Blok zinciri teknolojisi sizin için uygun mu? 19
Şekil 3.7 FGA mining kümesi 25
Şekil 3.8 ASIC cihaz görüntüsü 26
Şekil 3.9 Kiracı – Ev sahibi arasındaki işlemlerin gösterimi 36
Şekil 3.10 DAPP çalışma grafiği 37
Şekil 3.11 Web3 ile sunulan katman hizmeti şekli 38
Şekil 3.12 Truffle Freamwork Ganache test ortamı hesap cetveli 40
Şekil 3.13 Truffle Freamwork Ganache test ortamı transfer işlemleri dökümü 41
Şekil 3.14 MetaMask ekran görüntüsü 42
Şekil 3.15 Blok zincirinin temel klasörünün belirlenmesi 45
Şekil 3.16 Blok zincirinin türünün belirlenmesi 45
Şekil 3.17 Blok zincirinde hesap açma işlemi 46
Şekil 3.18 Blok zincirine var olan blok sayısı 46
Şekil 3.19 Blok zincirinde madencilik işlemi 47
Şekil 3.20 Hesap 1'de biriken kripto para 48
Şekil 3.21 Üretilen ethereum miktarı 48
Şekil 3.22 Transfer işlemi 49
Şekil 3.23 Transfer işlemi onayı için madencilik işleminin yapılması 49
Şekil 3.24 Madencilik sonrası hesap bakiyeleri 50
Şekil 3.25 Kiracı hesap bilgileri dökümü 51
Şekil 3.26 Ev sahibi hesap bilgileri dökümü 51
Şekil 3.27 Kiracı ve ev sahibi hesap bakiyeleri görüntüleme 53
Şekil 3.28 Kiracıdan ev sahibine transfer onay ekranı 53
Şekil 3.29 Kiracıdan ev sahibine transfer aşamaları görüntüsü 54
Şekil 3.30 Güncel hesap bakiyeleri 56
Şekil 3.31 Uygulama çıktısı 56
Şekil 3.32 Veri tabanı ara yüzü çıktısı 57

ÇİZELGELER DİZİNİ

Sayfa No

Çizelge 3.1 Özetleme algoritmaları karşılaştırma tablosu 17
Çizelge 3.2 Popüler kriptolaralar tablosu 21

SİMGELER VE KISALTMALAR DİZİNİ

DLT	Dağıtılmış defter teknolojisi
IoT	Nesnelerin interneti
M2M	Makineden makineye
P2P	Düğümden düğüme
IIoT	Endüstriyel nesnelerin interneti
BcoT	Nesneler için blok zinciri
ETH	Ethereum
XMR	Monero
XRP	Ripple
LTC	Litecoin
MIOTA	IOTA
CPU	Merkezi işlem birimi
GPU	Grafiksel işlem birimi
FBGA	Alanda Programlanabilir Kapı Dizileri
ASIC	Uygulamaya özgü tümleşik devre
DAPP	Merkezi olmayan bir uygulama
EVM	Ethereum sanal makinesi
PoW	İş kanıtı
GUI	Grafiksel kullanıcı ara yüzü

1. GİRİŞ

Bitcoin' in hayatımıza giriş yapan ve ekonomik olarak birçok kişinin ilgisini çekmesi ile blok zinciri kavramı araştırılmaya başlanmıştır. En genel tanımı ile blok zinciri, dağıtılmış bir defter teknolojisi denilebilir. Farklı bir deyişle değiştirilmeye veya dışarıdan müdahale edilmeye karşı korumalı dijital bir hesap defteridir. Oluşturmayı ve bu defteri anlık olarak paylaşmayı mümkün kılan verilerin güvenlikli bir şekilde birbirine bağlandığı bir veri yapısı modelidir. Bu veri yapısı kriptografik olarak birbiri ile bağlantılı veri-işlem bloklarından oluşur. İşlem gerçekleştikçe işlemleri kayıt altına almak için zincire yeni bir halka eklenir ve süreç böyle devam eder. Bu defter-i kebire kaydedilen verileri bloklarını değiştirmek veya çıkartmak son derece zordur ve bu işlem tüm katılımcıları bilgisi ile gerçekleştirilir. Blok zincirini ilgi odağı kılan da aslında bu zorluk derecesidir. Değiştirilemediği ve müdahale edilemediği için kullanıcılara sonsuz bir güven vermektedir.

Nesnelerin interneti ile insanların dışında cihazların da kişisel hayatlara ait verileri internet taşıması ile ortaya çıkan en büyük sorun olan güvenlik-mahremiyet ikilisi hep akıllarda soru işareti bırakmaktadır. Nesnelerin interneti, kullanıcılara hayatlarında yer verdikleri tüm donanımları uzaktan ve olayca özelleşmiş olarak kontrolünü sağlama imkânı verdiği için kolayca vaz geçilmez bir teknoloji olmuştur(He ve ark, 2014;Yang, 2014). Gartner'a göre IoT teknolojisinin katma değerli ürün ve ekonomik katkısı 2020 yılına kadar 263 Milyar dolara ulaşması beklenmektedir. Nesnelerin interneti ile çözüm pazarı büyüklüğü yine 2020'de 7 Trilyon dolara ulaşacaktır(MAcGillivray ve ark, 2013). Bu kadar hızlı büyüyen bir teknoloji için güvenlik, teknolojinin gelecekte var olma yahut yok olma anlamına gelmektedir.

Bu çalışmada blok zinciri teknolojisinin nesnelerin interneti uygulamalarında güvenlik ve mahremiyet problemine güçlü bir teknolojik alt yapı oluşturacağı savunulmaktadır. Nesnelerin interneti teknolojinin mimari yapısına daha yakından bakıldığında temel bir dezavantaj, merkezileşmiş bir bulut üzerinde güven duyulmamalarıdır. İnternete bağlı nesneler için blok zinciri metodu yüksek güvenlik

politikası sağlar. Hatta bu yaklaşımı desteklemek adına endüstriyel nesnelerin interneti (IIoT) için olmazsa olmaz nesneler için blok zinciri (BcoT) kavramı ortaya atılmıştır.

Blok zincirinin nesnelerin internetine dahil edilmesi sürecinde birçok güvenlik önlemi mekanizması desteklenmektedir. Nesnelerin internetine bağlı olan aygıtların ortak bir defter-i kebirde benzersiz kimliklerle tutmasını sağlamak için blok zinciri özelleştirilerek farklı bir kurgu oluşturulmuştur. Aslında bir akıllı cihaz zinciri oluşturmak da denilebilir. Bu zincire dahil olan cihazlar birbiri ile iletişim kurabilecek, akıllı sözleşmeler yürütebilecek ve sonuç verilerini zincire dahil edebilecektir. Blok zinciri tabanlı sunulacak bir mimari yöntem ile geleneksel yöntemlere dayanmayan daha verimli ve daha güvenli bir yapı oluşturulmak istenmektedir.

Bu tez çalışmasında blok zinciri teknolojisi kullanılarak gerçek hayatta bir aracı vasıtasıyla gerçekleşen ev sahibi – kiracı ilişkisini akıllı bir kontrat ile otonom bir şekilde uygulanması savunulmaktadır. Ev sahibi ve kiracı kendi aralarında belirledikleri sözleşme şartlarını esas alan teyit yazılımı, kiracı ve ev sahibi hesapları arasında yine sözleşmede yer alan miktarın transferinin doğrulanması sonrasında kiracıya evde bulunan tüm hizmetlerin kapısını açmaktadır. Bu öneride bir otorite veya aracıdan bağımsız olarak ilerleyen sistemde transfer maliyetleri minimuma indirilmiş olup bununla birlikte sonsuz bir güven ve şeffaf bir ticaret sağlanmıştır. Ayrıca bu işlem blok zincirinde yıllara sari olarak saklandığından güçlü bir loglama işlemi de sağlanmış olacaktır.

2. ÖNCEKİ ÇALIŞMALAR

Blok zincirinin mimari olarak temel mantığını oluşturan dağıtık sistemler çok uzun bir zamandır Peer-to-Peer (P2P) dosya paylaşımı olarak hayatımızda kullanılmaktadır. P2P ağları dağınık sistemlerin kişisel kullanımda olan en yaygın uygulamasıdır. Eşten eşe çalışan bu sistem de blok zinciriyle merkezi olmaması yani dağıtık mimari ile çalışması konusunda benzerlik göstermektedir. Sadece dağıtık değil, aynı zamanda merkezi olmayan, adaptasyon ve ölçeklenebilirlik gibi diğer teknik hususları da bünyesinde bulundurmaktadır(Rowstron ve Druschel, 2001). Gnutella kişisel kullanıma sunulan ilk P2P uygulamalarından biridir. Uygulamanın ana fikri müzik kaynaklarının kişiler arasında paylaşmasındaki telif haklarını korumaktır.(Adar ve Huberman, 2000; Ripeanu, 2001). Ancak P2P uygulamalar sadece müzik paylaşımı için kullanılmaz. P2P'in günümüzde kullanılan en popüler uygulaması ise Bittorrent'tir. Uygulama daha yüksek düzeyde sağlamlık ve kaynak kullanım sağlar(Cohen, 2003). Bu uygulamada, tüm yükleme maliyetleri sistemde bulunan tüm makinelere dağıtılmaktadır. Birden fazla makine aynı dosyayı aynı anda indirmeye başladığında, bunlar da birbirlerine dosya parçaları yükler. Bu mekanizma sınırsız sayıda indirme makinesine büyük ölçeklenebilirlik sağlar(Bharambe ve ark, 2005). IPTV hizmetleri için de geçerli olabileceğini gösteren çok sayıda araştırma var. Çok oyunculu çevrimiçi oyun, trafik yerelleştirme vb.(Chen ve ark, 2007; Douglas ve ark, 2005; Seedorf ve ark, 2009).

Blok zinciri teknolojisi, Bitcoin ile birlikte Satoshi Nakamoto tarafından tanıtıldı(Nakamoto, 2008). Bitcoin işlemleri blok zinciri adlı halka açık bir deftere kaydedilir. IBM ve Samsung, blok zinciri teknolojisinden yararlanarak merkezi olmayan IoT çözümleri oluşturmak için bir işbirliği yaptığını açıkladı(Brody ve Pureswaran, 2014). Gerçek dünyadaki fiziksel nesnelerin blok zinciri tarafından kontrol edilmesini sağlayan Slocks adlı bir akıllı kilit teknolojisi geliştirmiştir. Gerçek dünyadaki fiziksel nesnelerini (örneğin evler, arabalar veya bisikletler) kiralamak isteyen bir Slock sahibi, nesnelerin kullanımı için bir depozito tutarı ve bir fiyat belirledi. Kullanıcılar, cep telefonlarını kullanarak Slock'ları bulabilir. İşlemler Ethereum ağında doğrulandıktan sonra, kullanıcılar Slock'ları akıllı telefonlarıyla

açma veya kapatma iznine sahip olurlar. Akıllı bir sözleşme otomatik olarak uygulanır. Trans Active Grid, akıllı sözleşmeleri ve blok zincirini kullanarak kullanıcıların birbirlerinden güvenli ve otomatik olarak güneş enerjisi almalarını ve satmalarını sağlayan yazılım ve donanım teknolojilerinin bir kombinasyonunu geliştirmiştir. Blok zinciri protokolü ve Dağıtılmış Defter Teknolojisi (Distributed Ledger Technology-DLT) günümüzde popüler olmasına rağmen yeni icatlar değildir. Protokolün temelini oluşturan prensipler ve teknolojiler, modern bilgisayar teknolojilerinde on yıllardır kullanılıyordu. Bitcoin kurucusu tarafından yayınlanan bu uçtan uca sistem, yeni bir teknolojiyle değil, şu anda kullanılan teknolojilerin farklı ve mükemmel bir bakış açısıyla bir araya gelmesini sağladı. Bitcoin ana fikri tamamen dijital para transferi olduğundan, yalnızca elektronik para transferi problemleriyle sınırlı bir çözüm değil, aynı zamanda geniş bir alana kolayca uygulanabilir bir çözüm olduğunu fark etmek uzun zaman aldı. Bitcoin ile gerçekleştirilen ödemeler, takaslar ve takas işlemlerine uygulanması doğal bir etkisi oldu(Mills ve ark, 2016). DLT'ye yatırım yapan sadece finans şirketleri değil, aynı zamanda DLT araştırmalarına katılan ve IBM'in hyperledger ve hyperledger fabric gibi uygulamalarını ortaya koyan diğer endüstrilerdi (Cachin, 2016). Hiperger fabric "beyaz bülteninde "söylendiği gibi bir blok zinciri dağıtılmış bir protokol aracılığıyla güvenilir bir bilgisayar servisini taklit eder. Ürünleri, işletme sınıfı, açık kaynak kodlu bir dağıtım çerçevesi ve kod tabanıdır. Protokol sadece dağıtılmış defterler sağlamakla kalmaz, aynı zamanda yüksek güvenlik, ölçeklenebilirlik ve değiştirilemez işlem geçmişi tutmayı sağlar. Bu özellikler, sağlık, enerji ticareti, tedarik zinciri ve tarım, IoT gibi diğer endüstrilerin de blok zinciri teknolojisini kullanmak istemesinin ana sebebidir. Araştırmacılar tarafından devletlerin mevcudiyetinin istikrarlı bir blok zincir yapısına hâlâ ihtiyaç duyulup duyulmadığı sorusu sorulmuştur(Atzori, 2015; Davidson ve ark, 2016). Sadece bitcoin'i değil, aynı zamanda makalelerinde, üst düzey alan sansürüne, ademi merkeziyetçi oy sistemi, ademi merkeziyetçi özerk kuruluş / şirketler / toplumlar ve diğer blok zinciri işlevlerine dayanan ademi merkeziyetçi alan adı sisteminin oluşturulmasına dayanan diğer uygulamalara atıfta bulunuyorlar. Bu fikrin ipucu noktaları, blok zinciri protokolünün devlet hizmetlerinin yerini almaya başlayan özellikleridir. Protokol sağlık hizmetleri için de çok uygundur. 2016 yılında blok zinciri tabanlı sağlık bilgi alışverişi ağı hakkında bir makale yayınlandı(Peterson ve

ark, 2016). Hasta bilgilerini saklamak ve kurumlar arasında güvenli bir şekilde paylaşmak önemlidir. Çalışmanın ana fikri, hassas sağlık hizmetlerini etkin ve güvenli bir şekilde paylaşabilen bir yaklaşımı tanımlamaktır. Dağıtılmış muhasebe teknolojilerine yatırım yapan bir başka sektör de enerji ticareti pazarıdır. Bilindiği gibi, enerji ticareti piyasası para transferi ve ödemeler endüstrisi için en büyük ticaret platformlarından biridir. Ödemeler ve para transferleriyle aynı kısıtlamalar sayesinde, blok zinciri enerji ticareti piyasası için çok uygundur. Yeşil enerji ticareti yapmayı amaçlayan, WePower adında umut verici bir proje vardır(WePower, 2018). Proje ethereum blok zinciri üzerine kuruludur ve enerji birimini belirtmektedir. Sağlık hizmetlerine benzer şekilde, tedarik zinciri ve tarım endüstrileri de DLT' yi memnuniyetle karşılamaktadır. Son yıllarda dijital tedarik zinciri entegrasyonu giderek daha dinamik hale geliyor. Müşteri talebine ulaşmak, talepleri etkin bir şekilde paylaşmak, ürün ve hizmeti izlemek her geçen gün daha da önem kazanıyor. DLT çözümleri tedarik zinciri endüstrisinde başarılı, verimli ve ölçeklenebilir bir dijital tedarik zinciri yaratmada popüler hale geldiğinden, bu alandaki bazı çalışmalar DLT çözümleriyle özelleşmiş blok zinciri uygulamalarını geliştirmek için kullanılabilecek yöntemleri tartışmakta ve analiz etmektedir(Korpela ve ark, 2018). Blok zinciri teknolojisinin değişik alanlarda kullanımının yaygınlaşma ile tarımsal bilişim alanında da popüler hale getirmiştir. Uygulamalar erken uyarı sistemlerinden temel verileri ölçmeye ve planlamaya kadar değişiklik gösterir. Blok zinciri protokolü tek nokta başarısızlık sorunlarını ortadan kaldırdığından ve kritik bilgi paylaşımını daha güvenli hale getirdiğinden, e-tarım ve çiftlik sistemlerini daha iyi hale getirmek için de kullanılabilir(Lin ve ark, 2017). Sadece Bilgi ve İletişim Teknolojileri değil, IoT de son yıllarda popülerlik kazanıyor. Hayatımızın çeşitli yönlerini kapsamaya başladı.

Günümüz dünyasında hayatımızın her alanında kullanılan bilgi iletişim teknolojisinde, gizlilik ve güvenlik açıkları kullanımı noktasında tereddütler oluşturmaktadır. Bitcoin ile blok zinciri protokolü bu problemi üçüncü ve güvenilir bir taraf olmadan çözebileceğini kanıtlamıştır. Bununla birlikte, blok zinciri protokolü ayrıca hesaplama maliyeti, yüksek bant genişliği gerektiren ve gecikmeler gibi IoT uygulamalarında kullanılacak zayıf yönlere sahiptir. IoT topolojisinin daha ucuz ve daha hızlı çözümler gerektirdiği açıktır. 2016'da yayınlanan ve bu iki yaklaşımı

toplayan bir araştırma çalışması vardır(Dorri ve ark, 2016). Bazı diğer çalışmalar, DLT'yi akıllı şehirler, kimlik yönetimi, çevrimiçi kamu noterliği vb. Gibi diğer alanlara uygulamak için yeni fırsatlar ve olanaklar olduğunu göstermektedir (Sun ve ark, 2016; Jacobovitz, 2016; Arrendondo, 2018).

Nesnelerin İnterneti (IoT) insandan insana veya insandan bilgisayara etkileşimi gerektirmeden ağ üzerinden veri alışverişinde bulunabilecek insanlar, hayvanlar veya nesneler arasındaki bağlantıyı tanımlayabilir. IOT, makineden makineye iletişim (M2M) içinde çalışan ve uygulamalar, etki alanları ve protokollerle örtüşen aygıtlardan, sistemlerden ve hizmetlerden çeşitli bağlantılar sunar(Höller, 2009; Aggarwal ve ark, 2012). Nesnelerin İnterneti'nin [IoT] geliştirilmesi, temel olarak, içine gömüldükleri meta zincirleri aracılığıyla tüm nesneleri takip etme kabiliyetinin sağladığı öngörü ve öngörülebilirlikten büyük ölçüde faydalanmayı amaçlayan büyük şirketlerin ihtiyaçlarından kaynaklanmaktadır(Lianos and Douglas, 2000). Nesneleri kodlama ve takip etme yeteneği, şirketlerin daha verimli olmalarını, süreçleri hızlandırmalarını, hataları azaltmalarını, hırsızlığı önleme ve IoT aracılığıyla karmaşık ve esnek organizasyonel sistemleri kullanmalarını sağlamıştır(Ferguson, 2002). IoT, hemen hemen her şeyin bağlanabileceği ve daha önce hiç olmadığı gibi akıllı bir biçimde iletişim kurduğu bir dünyayı tanımlar. Nesnelerin İnterneti olarak adlandırılan şeyde fiziksel yollara yerleştirilmiş sensörler genellikle Internet'i bağlayan aynı İnternet IP'sini kullanarak kablolu ve kablosuz ağlar üzerinden bağlanır. Bu ağlar, analiz için bilgisayarlara akan büyük miktarda veri üretmektedir. Nesneler hem çevreyi algılayarak hem de iletişim kurabildiklerinde, verileri anlamlandırma ve ona hızlı bir şekilde yanıt verme araçları haline gelmektedir(Biddlecombe, 2009; Reinhardt, 2004)

3. MATERYAL ve YÖNTEM

3.1.Blok Zinciri Teknolojisi Teknik Detayları

3.1.1. Defter-i kebir

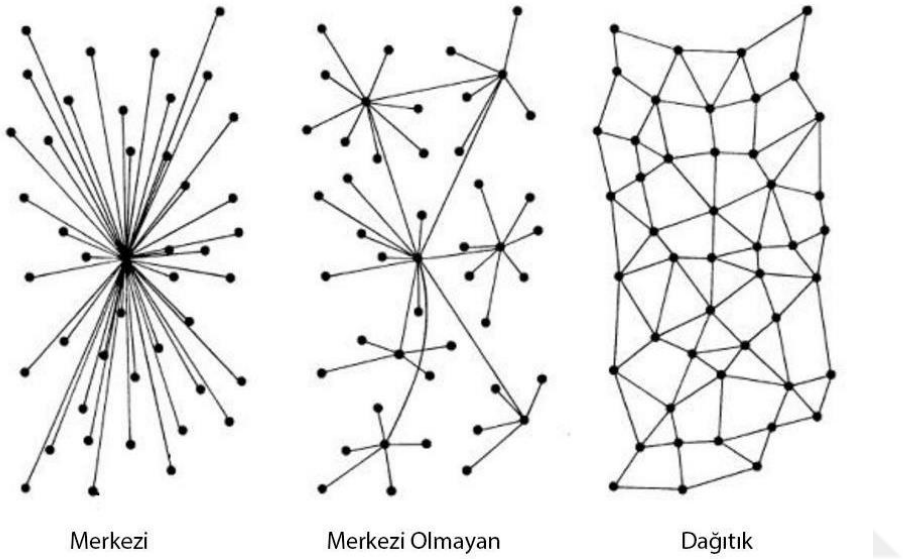
Gerçek hayatta bir alışveriş sonucunda alınan ürün veya hizmetlerin karşılığı olarak ödenen ücreti nakit veya borçlanma olabilir. Borçlanma yöntemiyle alışveriş neticesinde karşılıklı borcun bir kayıt altına alınması söz konusudur. Durum böyleyken hem ürün veya hizmeti alan hem de satan otoriteler borç konusunda mutabık kalmalıdır. Geçmişte bu işlemin veresiye başka bir deyişle borç defterine yazılmaktaydı.

Geçmişten günümüze bilgisayar teknolojilerinin çeşitlenmesinde kullanıma sunulan donanımlar etkili olmuştur. Kullanıcılar bir masaüstü bilgisayar, dizüstü bilgisayar ve mobil cihazlar ile etkileşimi sağlamaktadırlar. Bu durum da sunucu tarafında bir kalibrasyona ihtiyaç olduğunu ortaya çıkardı. Bu ihtiyacı karşılamak için ise geliştiriciler; istemci-sunucu mimarisi ile veri tabanı olgusunu ortaya attı. Temelde amaç gösterime sunulacak verilerin ve diğer materyallerin ayrı ayrı saklanması hatta işlenmesiydi.

Sunucuda tutulan veri tabanları donanım tarafındaki cihazlardan bağımsız olarak senkronize edilebilecek ve sürdürülebilir olarak güncel tutulacaktı. Bu noktadan sonra bulut bilişim insanların bulundukları konumdan ve cihazlardan bağımsız olarak tüm verilere ulaşabilmesi ile hayatlarında vazgeçilmez bir yer edindi. Temel sebebi hayat kolaylaştırma iken ek olarak ucuz maliyet de var. Bu ucuzluk ile insanların etkileşimli oldukları tüm cihazları internete bağlayarak yine cihaz ve yer bağımsız şekilde kullanmasının önünü açtı. Durum böyleyken müthiş derece yoğun bir istemci-sunucu yapısı ortaya çıkmasından dolayı güvenlik ve mahremiyet konuları da gündeme gelmiştir. Şimdiye kadar olan süreçte veri tabanı yapısı hep bir merkezde tutulmuştu. Yani tüm sistemin omurgasını merkezde bulunan bir sunucu oluşturmaktadır. Bu

yapının güvenlik parametreleri incelendiğinde belirlenen açıklar araştırmacıları farklı bir model arayışına itmiştir. Bu alanda çalışmanın bir zaruri ihtiyaç olması gündeme gelmiştir.

Kullanıcılardan soyutlanan veri; saklanması itibariyle 3 ayrı modele ayrılır. Bunlardan birincisi merkezi, ikincisi merkezi olmayan ve üçüncüsü ise dağıtık mimaridir.



Şekil 3.1 Mimari modeller

Bu üç mimari 5 ayrı başlık altında incelendiğinde

- Bakım: Merkezi modelde tek bir bakım noktası olduğu için kolaydır. Merkezi olmayan ise nokta fazla olduğu için daha zordur. Dağıtık mimari ise bakımı en zor olan modeldir.

- Kararlılık-Güvelik: Merkezi modelin hata toleransı tek bir noktaya bağlı olduğu için çok yüksektir. Merkezi olmayan ise daha kararlıdır. Olası bir hata telafi edilebilir. Dağıtık mimari en kararlı modeldir. Hatalar sistemin kesintisiz çalışmasını engelleyemez.

- Ölçeklenebilirlik: Merkezi model düşük seviyede bir ölçeklenebilirlik sunar. Bu durum merkezi olmayanda biraz daha fazladır. Dağıtılmış ise sonsuz bir büyüme söz konusudur ve sistemin performansı asla düşmez.

- Geliştirme Kolaylığı: Merkezi olan mimaride tasarım aşamasında bir sonu olacağı planlanacağından çok kolay değildir. Merkezi olmayanda da aynı durum söz konudur. Dağıtıkta ise sisteme eklenen her düğüm ile geliştirme kolayca sağlanabilir.

- Dönüşüm: Merkezi modelde tek bir metot kullanılmalıdır. Diğerlerinde ise çeşitlilik kolayca sağlanabilir.

Üç ayrı veri saklama modeli incelendikten sonra "Dağıtık Mimari"; kararlılık, güvenlik, ölçeklenebilirlik, geliştirme kolaylığı ve dönüşüm çeşitliliği bazında en yüksek seviyede fayda sağlamaktadır.

2008 yılında Satoshi Nakamoto tarafından "Bitcoin: Uçtan uca bir elektronik para sistemi" ile hayatımıza giren bu yaklaşım aslında bulut bilişim, nesnelerin interneti, sensor network, smart grid gibi Gartner eğrisinde yer alan teknolojilerin kilitlendiği noktayı çözüme kavuşturacaktı.

Dağıtılmış veri tabanı mimarisi, geliştirilmiş güvenliği, veri değişmezliği ve esnek yapısı ile önemli bir yenilik getirmektedir. Defter-i Kebir; saklanması gereken verileri farklı ve birden fazla düğümlerde sakladığı ve aynı anda senkronize ederek güncel tuttuğu, yapılan işlemlerin de kaydını tutabilen bir veri tabanıdır. Her bir düğüm, merkezi otorite onayı olmadan bir konsensüs temelinde işlem yapar. Temel olarak ağdaki her düğüm işlemleri sonuçlandırdıktan sonra uzlaşmak için sonuçları yayınlar. Mutabık kalındıktan sonra düğümlerdeki tüm defterler güncellenir ve her düğümde bulunan kopyalar birbirinin eşidir. Bu işlem yapıldıktan sonra herhangi bir düğüm geri dönük olarak verilere müdahalede bulunamayacağını göstermiş olur. Ancak bu durum için konsensüse ihtiyaç duyulmaktadır. Defter-i Kebir; blok zinciri ve akıllı sözleşmelerden oluşur.

Bir defter-i kebir olmazsa olmaz 4 temel bileşen vardır. Bunlar;

- ✓ Dağıtık bir defter-i kebir yapısı,
- ✓ Değiştirilmesi mümkün olmayan ve müdahaleye açık olmayan salt yapıdaki önceki veri blokları
- ✓ Ağ mimarisini yöneten konsensüs algoritmaları
- ✓ Şifreleme ve yetkilendirmeden kullanılan asimetrik anahtarlar.

3.1.2. Blok zinciri teknolojisi

Bilgisayarın icadından sonraki süreçte ve internetin yaygın hale gelmesi ile kullanıcılar arasında dijital bir iletişim ağı ortaya çıkmıştır. Bu ağın oluşmasıyla birlikte haberleşme için güvenlik önem kazanmakla birlikte iletişim sürekli ve bütünlük içerisinde olması da söz konusudur. Teknolojinin olağan gelişimi ile çeşitli güvenlik önlemleri denenip kullanılmıştır. İletişimde gönderici ve alıcı arasındaki haberleşmenin matematiksel ifadelere büründürülüp sağlanmasına kriptografi nedir. Kriptografinin tarihsel gelişimine bakılırsa aynı yerde olmayan iki birey veya topluluğun güvenli ve gizli haberleşmesi ile kullanılmıştır. Hiyeroglifler işaretlerle bir veya birden fazla harfi temsil edebildiğinden çok fazla sayıda karakter içerir. Tarihte Mısır Hiyeroglifleri literatürde bulunan en net işaretlerdir. Alfabelerinde 700'den fazla karakteri barındırır. Toplumların birbirleri ile etkileşiminde gelişme gösteren bu bilim, 2. Dünya Savaşı'na kadar ilk seviyede seyretmiştir. Haberleşmenin olmazsa olmazı haline gelmesinde ise savaş harekat planlarının iletilmesi, gizli bilgilerin saklanması gibi amaçlarla insanlık tarihinde yerini almıştır. Temelde kriptografi şifreleme bilimi olarak adlandırılır. Bu bilimde hem şifreleme hem de şifrenin çözülmesi işleri vardır. Bu 2 ayrı iş ise alt bilim dalı olarak düşünülebilir. Günümüzde ise bu bilim, teknolojinin her aşamasında insanların hayatına girmiştir. 21. Yy'da veri eşittir güç ilkesi doğrultusunda verinin de gizlilik ilkelerine dayanarak saklanması, bütünlük içermesi ve gönderici-alıcı doğrulaması neticesinde erişimde olması kriptografi ile mümkündür(ÖZDAĞ, 2009).



Şekil 3.2 Mısır hiyerogliflerinden bir kesit

Şifreleme teknikleri temelde 3 ayrı kısımda incelenebilir. Bunlar; klasik şifreleme, modern şifreleme ve kuantum şifreleme teknikleri olarak tanımlanabilir.

- Klasik Şifreleme: Geçmişten günümüze kadar ulaşmış en basit şifreleme metodudur. Yer değiştirme ve yerine koyma olarak kendi içerisinde ayrılır.

    - Yer Değiştirme: Güvenlik ve bütünlüğün sağlanması için iletiyi oluşturan karakter ve-veya karakterlerin yerlerini değiştirerek anagram yani düzenlenmiş bir halde iletilmesini içeren yöntemdir. İletinin uzunluğu ve kelime sayısına göre düşünüldüğünde varyasyon sayısı karakter sayısının faktöriyeli kadar olabilir. Bu da düşük seviye bir yöntem olduğunu kanıtlar. Tarihte ilk olarak beşinci yüzyılda Spartalılar tarafından kullanıldığı literatürde yer almaktadır(Singh ve Simon, 2004).

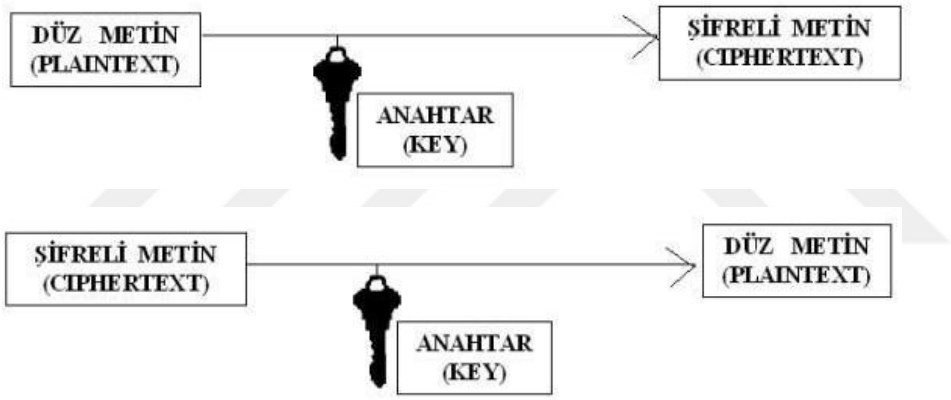
    - Yerine Koyma: Bu yöntemde iletiyi oluşturan karakterleri bir kuralara göre yeniden düzenlemeyi içerir. Eğer ileti ardışık karakterler olarak bir örüntü düzeninde değiştirilmesi yapılır. Bu yöntem de yerine konulan karakter(ler)in sayısına ve dolayısıyla zorluk derecesine göre tekli alfabetik yerine koyma ve çoklu alfabetik yerine koyma olarak ayrılır.

Düz metin	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Şifreli metin	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c

Şekil 3.3 Tekli yerine koyma metodu örneği

- Modern Şifreleme: Haberleşmenin en temel yapı taşı olan modern şifreleme teknikleri bir kural neticesinde oluşturulan karmaşık işlemleri içerir. Bu yöntemde kullanılan tüm algoritmalarda hem şifreleme hem de şifre çözme için bir anahtar kullanılır. Simetrik ve asimetrik olarak ayrılır.

    - Simetrik Anahtarlı Şifreleme: Gönderici ve alıcı tarafların bildiği anahtar yardımı ile sağlanan güvenli iletişimde gönderici iletiyi anahtar yardımıyla şifreleyip alıcıya gönderir. Alıcı ise aynı anahtar ile şifrelenmiş iletiyi çözerek içeriğe ulaşır.



Şekil 3.4 Simetrik anahtarlı şifreleme şekli

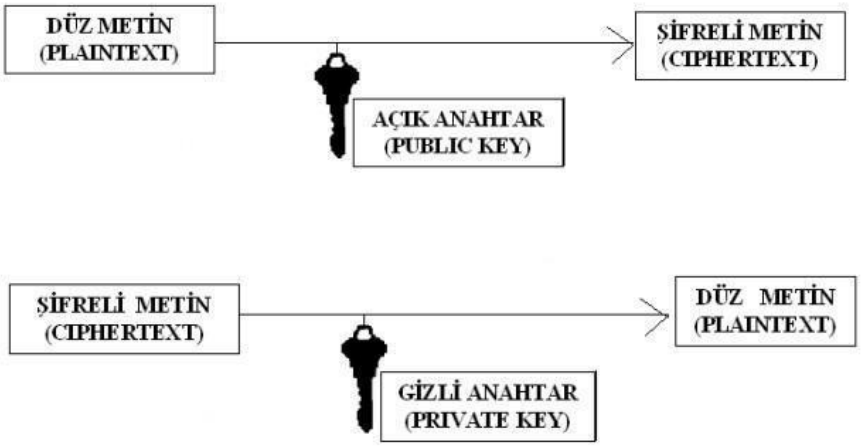
Simetrik anahtarlı şifreler akışlı şifreler ve blok şifreler olarak 2'ye ayrılır. Akış olan şifreleri kısaca tanımlamak gerekirse iletiyi oluşturan her karakter XOR lojik işlemine tabi tutularak şifrelenir. Bu sebeple şifreleme anahtarı her karakter için farklı olmaktadır(Courtois ve Nicolas, 2003).

İleti: i = i(1)i(2)i(3).....i(r)
Anahtar: a = a(1)a(2)a(3).....a(r)
Şifrelenmiş İleti: k = k(1)k(2)k(3).....k(r)

Blok şifrelerde ise ileti bloklara ayrılır, her blok aynı anahtarlar yardımı ile şifrelenerek iletilir. Her blok bir şifreleme işleminden geçirilerek şifrelenir. Bloklar birbirine organik olarak bağlı değildir. Yaygın olarak kullanılan bir yöntemdir. Sık kullanılan örnekleri; DES/3DES(Daemen;Rijmen:1996), DESX(Daemen;Rijmen:1996), AES (Rijndael) (Daemen,1999), MARS (Burwick,1999), RC6 (Rivest,1999), Twofish (Schneier,1998), Serpent (Anderson, 1998), Safer (Massey,1993), Blowfish (Schneier,1993)'dir.

Simetrik anahtarlı şifreleme tekniğinde yüksek oranda veri üretimi yapılır. Anahtar seçiminde esneklik sağlamaz. Temelde güçlü şifrelerin oluşturulması için kullanılabilir. Analizi kolay olmasına karşın tekniğin içerisinde zayıf olan dönüşümler daha güçlü şifreleri oluşturabilir. Anahtar odaklı olduğu için sistemin güvenliği anahtarın gizliliğine bağlıdır. Küçük ve geçici sistemler için kullanımı uygundur. En önemlisi ise rassal ve hash üretimlerinde kullanılan algoritmaların temelini oluşturur.

- Asimetrik Anahtarlı Şifreleme: Literatürde açık anahtarlı şifreleme olarak da yer alan bu yöntemde iki adet şifre bulunur. İlki herkes ile paylaşılan ve dağıtılan açık anahtar, diğeri ise gizli anahtardır. Açık anahtar şifrelemesinde iletinin göndericide açık anahtar ile şifrelenmesi, alıcı tarafta ise gizli anahtar ile çözülmesidir. Kavramsal olarak daha iyi anlatmak için yaygın örnek üzerinden gitmek doğru bir yöntem olacaktır. Bu yönteme en iyi örnek posta kutusudur. Alıcı kişinin adresini bilen herkes posta gönderebilir ama kutunun sahibi mevcut bulunan anahtar ile kutuyu açabilir. En yaygın açık anahtarlı sistemler; Diffie-Hellman, Merkle-Hellman, RSA, El Gamal, Rabin, McEliece, Knapsack, Dijital İmza'dır(Thomas, 2002).



Şekil 3.5 Simetrik anahtarlı şifreleme şekli

RSA Algoritması: 1978 yılında Ron Rivest, Adi Shamir ve Leonard Adleman kişileri tarafından geliştirilmiş olup soy isimlerinin ilk harflerinin yan yana gelmesiyle isimlendirilmiştir. Açık anahtar diğer bir isimle asimetrik anahtarlı şifreleme yöntemidir. Bu algoritma günlük hayatta dijital imzalama için kullanılır. Uygulamada anahtar oluşturma, şifreleme ve çözme olarak 3 adımdan oluşur.

Açık anahtar tekniklerinin tümünde olduğu gibi açık ve gizli olmak üzere iki anahtar vardır. Anahtar oluşturma işleminde iki adet asal sayı (p ve q) seçilir. Tanımlama işleminde bu iki değer birbiri ile çarpılır. (n=pq) Bu sayıların Totient fonksiyonuna tabii tutulur. (T(n)=(p-1)(q-1)) Daha sonra 1<e<T(n) olarak şekilde bir e sayısı üretilir. Burada e açık anahtar olarak kayıt edilir. de = 1mod(T(n)) olarak şekilde d sayısı üretilir. Bu işlemden ortaya çıkacak d ise gizli anahtar olarak kayıt edilir.

(n,e):Açık anahtar oluşumunda kullanılan değişkenler
(n,d): Gizli anahtar oluşumunda kullanılan değişkenler

Bu işlemler kümesinde p, q veT(n) değerlerinin gizli kalması neticesinde iletiler güvende olur. Şifreleme yukarıda anlatıldığı yapılırken şifre çözme işleminde ise işlemler tersine ilerler.

Açık anahtarlı sistemlerde sadece gizli anahtar ve gizli anahtar üretiminde kullanılan parametreler gizli tutulmalıdır. Kullanım durumuna göre gizli ve açık anahtar çifti değiştirmeksizin uzun bir süre kullanılır. Büyük ve global ağlarda bu yöntem sağlıklı olarak kullanılabilir. Avantajlarının yanı sıra dezavantajları da bulunmaktadır. Simetrik anahtarlı sistemlere göre daha yavaştır. Üretilen anahtar uzunlukları simetrik anahtarlara göre çok daha uzun olmaktadır(Menezes ve ark, 1996).

3.1.2.1.Özetleme ve Hash Üretimi:

Bir verinin boyutundan ve dolayısıyla uzunluğundan bağımsız olarak sabit uzunlukta ve alınan veriye has şekilde yeniden bir çıktı oluşturulması olarak ifade edilebilir. İdeal bir hash üretme fonksiyonu şu özelliklere sahip olmalıdır.

- Herhangi bir mesaj için özet hesaplamak kolay olmalıdır.
- Bir özete karşılık gelecek mesajı oluşturmak zor olmalıdır.
- Özeti değişmeyecek şekilde mesajı değiştirmek zor olmalıdır.
- Aynı özete sahip iki farklı mesaj bulmak zor olmalıdır.

Günlük hayatımızda teknoloji odaklı baktığımızda sıkça hash işlemini kullanmaktayız. En basit şekilde ifade etmek gerekirse bir web sitesine kayıt olma esnasında metin kutusu aracılığıyla girdiğimiz şifreler, ilgili sunucunun veri tabanında hash işlemine tabii tutularak saklanır. Böylece web sitesi yönetici dahi bizim şifremizi doğrudan göremez. Özetleme algoritmalarının tarihsel sıralamalarına göre baktığımızda MD5, SHA-0, SHA-1, SHA-2, SHA-3 ve SHA256 olarak devam etmektedir.

MD5 özetleme algoritması, iletimi sağlanacak verilerin bütünlüğünü kontrol etmek için kullanılmıştır. Ron Rivest tarafından 1992 yılında geliştirilen bu özet algoritması iletimi sağlanacak mesaj boyutundan bağımsız olarak 128 bitlik özet veriler üretir. Bu özetler tamamen benzersiz olması matematiksel olarak mümkün olmamasına rağmen günümüz teknolojilerinde az önem arz eden verilerin saklanmasın kullanılmaktadır.

İletilecek Mesaj: Ahmet Tabanlıoğlu
MD5 İle Üretilen Özet: b3d8e6f3f86a47c5ac61b835ce031de1
İletilecek Mesaj: “Ahmet Tabanlıoğlu”
MD5 İle Üretilen Özet: c10dc10e44bf2013a6714c8e12c89925

SHA 1 özetleme algoritması, Amerika Ulusal Güvenlik Kurumu tarafından geliştirilmiş olup uzunluğu en fazla 264 bit mesajları girdi olarak alabilir. Ve bunun

neticesinde 160 bitlik bir mesaj özeti üretir. Bloklara ayrılarak özetlenen veri MD5'te olduğu gibi matematiksel yöntemlerle iteratif olarak sonucu ortaya koyar.

İletilecek Mesaj: Ahmet Tabanlıoğlu
MD5 İle Üretilen Özet: bcf5b534ebb21a3b4e793baa588ebae6214db24a
İletilecek Mesaj: "Ahmet Tabanlıoğlu"
MD5 İle Üretilen Özet: 9023e091fa0615e702d389f2130cfbcdd2864be6

SHA2 özetleme algoritmaları; SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224 ve SHA-512/256 olmak üzere 6 alt gruba ayrılmaktadır. Bu ayrımı ortaya çıkan temel sebep ise üretilen özetin çıkış boyutudur.

Çizelge 3.1 Özetleme algoritmaları karşılaştırma tablosu

Algoritma Versiyonları		**Çıkış Boyutu (bits)**	**iç Yapı Boyutu (bits)**	**Blok Boyutu (bits)**	**Max Mesaj Boyutu (bits)**	**Uzunluğu (bits)**	**Rounds**
MD5		128	128	512	$2^{64}-1$	32	64
SHA-0		160	160	512	$2^{64}-1$	32	80
SHA-1		160	160	512	$2^{64}-1$	32	80
SHA-2	*SHA-224*	224	256	512	$2^{64}-1$	32	64
	SHA-256	256					
	SHA-384	384	512	1024	$2^{128}-1$	64	80
	SHA-512	512					
	SHA-512/224	224					
	SHA-512/256	256					
SHA-3		224/256/384/512	1600 (5×5 array of 64-bit words)			64	120 (varsayılan)

En yaygın olarak kullanılan SHA-256 özetleme algoritması blok zincirinde de kullanılmaktadır. Bu özetleme fonksiyonu ile üretilen veriler sabit uzunlukta oldukları için farklı veri kümelerinde çakışması mümkün değildir. SHA-256 algoritması ile üretilebilecek varyasyonların sayısı 2^256 olarak hesaplanır.

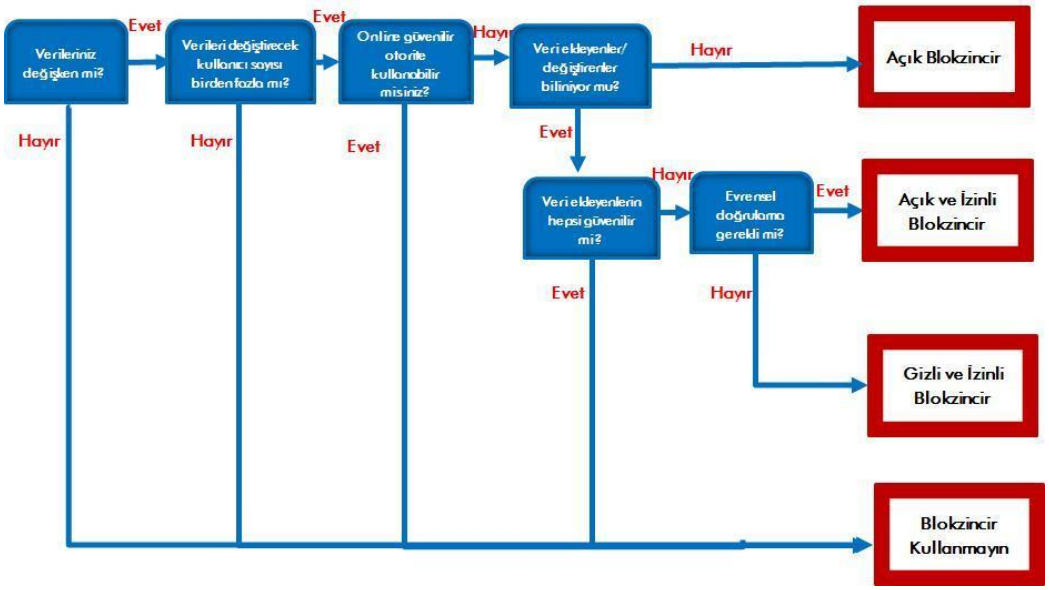
İletilecek Mesaj: Ahmet Tabanlıoğlu
MD5 İle Üretilen Özet:
555F1E25ABC193AD9204F1C69DFC6C488910545189DDBB19A5131BF0E7FB9250

İletilecek Mesaj: "Ahmet Tabanlıoğlu"
MD5 İle Üretilen Özet:
77B330C15A89B23B16D01D1C5B325CD27D2668E13A2A176EEAB455CCF4F3877B

İletilecek Mesaj: "Ahmet Tabanlıoğlu, bir bilgisayar mühendisidir."
MD5 İle Üretilen Özet:
6751B77E63709AEECA882D30DA38C979A6AEC6390F3426A4C432E3A49007E9D7

3.1.3. Blok zinciri teknolojisi kullanım durumu:

Blok zincir tüm iş modelleri için uygun olmayabilir. Blok zinciri metodunun uygun olup olmadığı hususuna proje başında dikkat edilmelidir. Tüm parametreler göz önünde bulundurularak kullanılacak teknolojiye karar verilmelidir. Yapılacak gereksinim analizi ile ihtiyaca cevap verip vermediği kontrol edilmelidir. Dikkat edilmesi gereken parametreler; saklanacak verinin birden fazla istemci tarafından üretilip üretilmeyeceği, birden fazla ise tarafların birbirine duydukları güven ve sadakat gibi durumlar seçici parametrelerdir. Yine önemli olan diğer bir husus veri tabanının genele açık olup olmaması durumudur. Ve en temel parametre ise verinin büyüklüğüdür. Bu parametrelerden sonra blok zincir teknolojisi kullanılacak olursa, sisteme sağlanan avantajlar ise; güvene dayalı olmayan bağlılık, gündelik matematiğin üzerinde kurgulanmış temel çalışma prensibi ile gelen mahremiyet, ve otorite olmaksızın sağlanan işlem güvenliği.



Şekil 3.6 Blok zinciri teknolojisi sizin için uygun mu?

3.1.4. Kripto paralar ve madencilik

Günümüzde en popüler kripto para birimi olarak tanımlanan Bitcoin; gerçekte blok zinciri metodunun açık kaynak olarak geliştirilen bir ürünüdür. Bu sistemin kullanımını için doğrudan elle dokunulan gözle görülen bir paraya ihtiyaç yoktur. Fiziki olarak değil dijital olarak ekonomide yer edinmiştir. Bu yöntemin organik bir yapıda olması ve bir merkez tarafından yönetilmiyor olması ilgiyi doğrusal olarak artırmaktadır.

Açık kaynak olarak 2009 yılında geliştirilen ve sunulan Bitcoin, genel olarak ilk dağıtılmış kripto para birimi olarak kabul edilir (Sagona-Stophel, n.d.). Bu tarihten günümüze kadar 4000 adet altcoin piyasaya sunulmuştur. Bunların tamamı Bitcoin'in çalışma mantığını referans almış blok zinciri metodunu farklı varyansları kullanarak üretilmiştir.

Akademik literatürde bir kripto para için;

- Merkezi olamaz. Dağıtılmış olmalı ve ortak bir konsensüs ile oluşturulmalı.

- Üretimi veya arzı için bir mekanizması olmalı. Eğer üretim varsa metodu belirlenip duyurulmalı.

- Mülkiyet sadece dijital cüzdanda görüntülenebilir. Fiziki bir görüntüleme söz konusu değildir.

- İşlemler, mimariye göre teyide muhtaçtır. Ancak doğrulama yapıldığında geçerli olur.

Bitcoin'den sonra üretilen tüm kripto paralar altcoin olarak adlandırılmıştır. Altcoinlerden bazıları ETHER, RIPPLE, ZCASH, Monero ve DASH vb... Altcoinler Bitcoin'in gerek mimarisi gerek ise piyasa değerine göre farklılık gösterebilirler. Bazıları, bir ülkenin tüm vatandaşlarına verilen altcoinler gibi farklı bir ekonomik modele veya farklı bir madeni para dağıtım yöntemine sahiptir. Bazıları, özel

madencilik donanımına karşı koymak için farklı iş kanıtı algoritmaları kullanmaktadır. Bazıları ise uygulamaları oluşturmak için daha çok yönlü bir programlama dili sunabilir. Bu farklılığı oluşturan temel sebep ise blok zinciri teknolojisinin amaca göre yorumlanması ve modelin bu ihtiyaçlara göre belirlenmesidir. Bu altcoinlerden popüler olanlarını kısaca tanımlamak gerekirse;

Çizelge 3.2 Popüler kriptolaralar tablosu

	ETHEREUM (ETH): Kendine ait bir makinenin diğer programlanabilir bilgisayarların yapabileceği hesaplamaları yapabilme kabiliyetine sahip altcoin türüdür. Her bir düğüm tarafından üretilen ve yürütülen komut dosyalarını veya sözleşmeleri kaydeder. Zincire katılan bir istemciye yerel şifreleme para birimiyle ödemeler yapar. Bu birimin adı 'eter'dir. Resmen 2015 yılında başlatılan Ethereum, birçok firmanın ilgisini çekmiştir.
	DASH: 2014 yılında ortaya çıkan DASH, kullanıcı gizliliği, mahremiyet ve anonimliği sağlamayı hedeflemiştir. Nakamoto'nun kurgusuna ek olarak ilk defa kriptoparalarda gizlilik ilkesi bu altcoinle gündeme geldi. Burada gelirler madenciler ve masternode arasında eşit olarak paylaşılıyor.
	MONERO (XMR): En eski altcoinler arasında yer alır. Havuda madencilik yapmayı mümkün kılar. Tekil yaklaşım yoktur. Madencilik havuzunda ortak olarak hash güçlerini birleştirip bloğun çözümüne katkı sağlayanlara katkı ile orantılı olarak ödül verir. Hem CPU hem de GPU eşit madencilik imkanı sunar. Bu altocinde de gizlilik ön plandadır.

Çizelge 3.3 Devam

	RIPPLE (XRP): Günümüzde yaygın olarak borsası bulunan kripto para cinsidir. Diğerlerinden ayıran en büyük özelliği ise madenciliğinin olmamasıdır. Bir şirket gibi çalışan ripple, geliştiricilerin zincir üzerinden ürettiği belirli sayıda XRP token'i ile çalışır.
	LITECOIN (LTC): 2011 yılında bitcoin çatallanmasıyla ayrıldı. Kripto paralar arasında Litecoin "gümüş" olarak tanımlanır. Satoshi Nakamoto'nun oluşturduğu kripto paranın hafif ve hızlı hali. Bitcoin işlemlerine göre 4 kat daha az zamanda işlemi tamamlanabilir.
	IOTA (MIOTA): 2015'te David Sønstebø, Sergey Ivancheglo, Dominik Schiener ve Dr. Serguei Popov tarafından kurulmuştur. Nesnelerin internetine odaklanmış bir kripto paradır. Blok zinciri yerine Tangle adı verilen kimliksiz ve merkezsiz bir dağıtım modelini kullanan IOTA, bu kullanımdaki amacını ise veri transferinin de kullanması nedeniyle açıklıyor.

3.1.4.1.Bitcoin Madenciliği

Bitcoin madenciliği, Bitcoin'in eski işlem defterine işlem kayıtları ekleme işlemidir. Geçmiş işlemlerin bu defterine, blok zinciri olduğu için blok zinciri adı verilir. Blok zinciri, ağın geri kalanına gerçekleşen işlemleri onaylamaya hizmet eder. Bitcoin düğümleri, yasal Bitcoin işlemlerini zaten başka bir yerde kullanılmış olan paraları yeniden kullanma girişimlerinden ayırmak için blok zinciri kullanır.

Bitcoin madenciliği kasıtlı olarak kaynak yoğun ve zor olacak şekilde tasarlanmıştır, böylece madenciler tarafından her gün bulunan blokların sayısı sabit kalır. Bireysel bloklar, geçerli sayılabilecek bir çalışma kanıtı içermelidir. Bu çalışma

kanıtı, her blok aldıklarında diğer Bitcoin düğümleri tarafından doğrulanır. Bitcoin hashcash iş kanıtı işlevini kullanır.

Madenciliğin temel amacı, Bitcoin düğümlerinin güvenli ve kurcalamaya karşı dayanıklı bir fikir birliğine ulaşmasını sağlamaktır. Madencilik aynı zamanda Bitcoin'leri sisteme dahil etmek için kullanılan bir mekanizmadır: Madencilere, yeni yaratılan paraların "sübvansiyonu" kadar herhangi bir işlem ücreti ödenir. Bu, her ikisini de yeni madeni paraları ademi merkeziyetçi bir şekilde yayma ve insanları sisteme güvenlik sağlama konusunda motive etme amacına hizmet ediyor.

Bitcoin madenciliğine, diğer malların madenciliğine benzemesi nedeniyle denir: çaba gerektirir ve yavaş yavaş yeni para birimlerini kullanılabilir hale getirir. Altın gibi malların zeminden madencilik oranına benzeyen bir oranda kullanılabilir.

Madencilik sırasında, Bitcoin madencilik donanımı blok başlığı olarak adlandırılan şey üzerinde bir şifreleme sağlama fonksiyonu (iki tur SHA256 turu) çalıştırır. Denenen her yeni karma için, madencilik yazılımı blok başlığının rasgele öğesi olarak farklı bir sayı kullanacaktır, bu sayıya "nonce" denir. Nonce'ye ve blokta başka ne olduğuna bağlı olarak, karma işlevi şöyle görünen bir karmaşıklık verecektir: 93ef6f358fbb998c60802496863052290d4c63735b7fe5bdaac821de96a53a9a

Bu karışıma gerçekten uzun bir sayı olarak bakabiliriz. (A-F harfleri 10-15 rakamları anlamına gelen onaltılık bir sayıdır.). Geçerli bir bloğu oluşturmak için madenci zorluk hedefinin altında bir karma bulmak zorundadır.

Örneğin zorluk hedefi =
“100000000000000000000000000000000000000000000000000000000000000000”
olsun.

Sıfır ile başlayan herhangi bir sayı hedefin altında olur;
“0787a6fd6e0782f7f8058fbef45f5c17fe89086ad4e78a1520d06505acb4522f”
sayısı hedefimizin altında kalmış olur. Hedefimizi düşürürsek yeni hedefimiz =

“0100000000000000000000000000000000000000000000000000000000000000” olsun.

Şimdi hedefin altında olmak için iki sıfıra ihtiyacımız var:
00db27957bd0ba06a5af9e6c81226d74312a7028cf9a08fa125e49f15cae4979

Hedef tonlarca rakamı olan rastgele bir sayı olduğundan, kullanıcılar mevcut hedefi ifade etmek için genellikle daha basit bir sayı kullanır. Bu sayıya madencilik zorluğu denir. Madencilik zorluğu, mevcut bloğun ilk bloğa kıyasla ne kadar zor oluşturduğunu ifade eder. Bu yüzden 70000'lik bir zorluk, Satoshi Nakamoto'nun ilk bloğu oluşturmak zorunda kaldığından 70000 kat daha fazla iş yapmanız gereken mevcut bloğu oluşturmak anlamına geliyor o zamanlar madencilik donanımı ve algoritmaları çok daha yavaş ve optimize olmadığından ilk bloğu oluşturmak çok uzun zaman aldığı tahmin edilmektedir.

Her 10 dakikada bir blokların kabaca çıkmasını sağlamak için, zorluk her 2016 blokta paylaşılan bir formül kullanılarak ayarlanır. Ağ, şu anki küresel ağ işleme gücündeki 2016 bloklarının yaklaşık 14 gün süreceği şekilde değiştirmeye çalışıyor. Bu yüzden ağ gücü arttığında zorluk da artar.

3.1.4.2.Bitcoin Madencilik Donanımı

Başlangıçta CPU ile bitcoin madenciliğinin tek yoluydu ve orijinal Satoshi müşterileri tarafından yapılırdı. Ağı daha da güvenli hale getirme ve daha fazla bitcoin kazanma arayışı içinde, birçok yerden yeni madenciler türedi ve yıllarca zaman harcandı. CPU madenciliği nispeten boşuna olmuştur çünkü günlük bilgisayarların CPU'su ile tek bir bitcoin elde etmek onlarca yıl sürmektedir.

Ağ başladıktan yaklaşık bir buçuk yıl sonra, bitcoin madenciliğinde yüksek seviye grafik kartlarının çok daha verimli olduğu ve ekosistemin değiştiği keşfedildi. CPU bitcoin madenciliği GPU'ya yol açtı (Grafik İşlem Birimi). Bazı GPU'ların büyük ölçüde paralel niteliği, iş birimi başına çok daha az güç kullanırken bitcoin madenciliği gücünde 50x ila 100x artış sağladı.

CPU'dan GPU'ya geçiş ile olduğu gibi, bitcoin madenciliği dünyası teknoloji zincirini FPGA kadar ilerletti. Butterfly Labs FPGA Single'ın başarılı bir şekilde piyasaya sürülmesiyle, bitcoin madenciliği donanım ekosistemi, madencilik bitcoinlerine adanmış özel olarak üretilmiş donanıma yol açtı.



Şekil 3.7 FGA mining kümesi

FPGA'lar, CPU'lardan GPU'lara geçişte görüldüğü gibi madencilik hızında 50x - 100x'lik bir artış olmamıştır, güç verimliliği ve kullanım kolaylığı ile fayda sağlamıştır. Tipik bir 600 MH / s grafik kartı 400 w'lık güç tüketirken, tipik bir FPGA madenciliği cihazı 80w güçte 826 MH / s'lik bir hashrat sağlayacaktır. Bu 5 kat iyileştirme, ilk büyük bitcoin maden çiftliğinin işletme karında inşa edilmesine izin verdi. Bitcoin madencilik endüstrisi doğdu.

Bitcoin madenciliği dünyası artık Uygulamaya Özel Entegre Devre (ASIC) dönemindedir. Bir ASIC, özellikle bir şeyi ve bir şeyi yapmak için özel olarak tasarlanmış bir çiptir. FPGA'ların aksine, bir ASIC diğer görevleri yerine getiremez. Bitcoin madeni yapmak için tasarlanan bir ASIC sadece bitcoin madenciliğinde kullanılır. Bir ASIC'in esnekliği, önceki tüm teknolojilere kıyasla güç tüketimini azaltmıştır.



Şekil 3.8 ASIC cihaz görüntüsü

Bitcoin Madenciliğinin 3 temel yolu vardır; Kendi başınıza madencilik yapmak, bir Bircoin maden havuzunun parçası olmak veya Bitcoin bulut madenciliği sözleşmeleri ile Bitcoin madenciliği yapılabilir. Bunlar arasında en popüler madencilik bir Bitcoin maden havuzuna madencilik yapmaktır çünkü madencilik sırasında şansı arttırmaktadır.

3.1.5. Ethereum ve ethereum sanal makinesi:

Ethereum, geliştiricilerin karmaşık işlemlerini geliştirmek istedikleri şekilde tanımlamalarına izin veren genel bir blok zinciri platformudur. Kripto para birimleri dağıtık mimari kullanılırlar, ancak bununla sınırlı değildir. Bitcoin'in aksine, finansal olmayan blok zinciri dağıtık uygulaması kullanılabilir.

Ethereum, dağıtık uygulamalar platformunu tanımlayan protokollere işaret eder. Bunun merkezinde keyfi algoritmik karmaşıklık kodunu çalıştırabilen Ethereum Sanal Makinesi vardır. Ethereum'un bir diğer amacı "Turing'i tamamlamak" dır. Geliştiriciler, Python ve JavaScript gibi mevcut dillerde modellenen Solidty adlı yerleşik bir dilde Dapps oluşturabilirler.

Ethereum, başka herhangi bir blok zinciri gibi bir P2P ağ protokolü içerir. Blok zinciri veri tabanı ağa bağlı birçok düğüm tarafından korunur ve güncellenir. EVM, ağın her düğümünde çalışır. Aynı şekilde, talimatlar da tüm düğümlerde çalışır.

Tüm Ethereum ağı genelinde hesaplamanın bu muazzam paralelliği, hesaplamayı daha verimli yapmaz. Aslında, bu paralelleştirme Ethereum'u geleneksel bir bilgisayardan çok daha pahalı ve çok daha yavaş yapar. Ancak her Ethereum düğümü, ağda fikir birliğini sağlamak için EVM'yi çalıştırır. Dağıtılmış görüş birliği, tek bir erişim noktasını kaldırarak ve blok zinciri sansürüne dirençli ve değiştirilemez hale getirilmiş verileri depolayarak sıfır aksama süresi sağlayan Ethereum mutlak hata tolerans seviyeleri sağlar.

Ethereum platformu, özelliksiz veya değer agnostik olacak şekilde tasarlanmıştır. Programlama dillerine benzer şekilde hangi amaç için kullanılması gerektiğine karar vermek girişimciler ve geliştiricilere kalmıştır. Ethereum, meslektaşlar arasındaki doğrudan etkileşimi otomatikleştirmek veya bir ağdaki koordineli grup hareketini kolaylaştırmak isteyen bir uygulama için özellikle uygundur. Örneğin P2P pazar yeri koordinasyonu veya karmaşık finansal sözleşmeler otomasyonu için uygulamalar. Bitcoin, finansal kuruluşlar, bankalar veya devletler

gibi herhangi bir aracıya dahil olmadan bireylerin nakit alışverişinde bulunmalarına izin verir.

Bitcoin, Ethereum etkisi sadece finansal borsa otomasyonundan çok daha fazlasıdır. Güvenlik, güven ve kalıcılığın oylama, varlık-sicil, yönetim ve IoT gibi önemli olduğu her türlü finansal olmayan uygulamaya uygulanabilir.

3.1.6. Akıllı kontratlar

Bir kripto sözleşmesi olarak da bilinen akıllı sözleşme, belirli koşullar altında taraflar arasında dijital para birimlerinin veya varlıkların transferini doğrudan kontrol eden bir bilgisayar programıdır. Akıllı bir sözleşme, sadece bir sözleşmeyle ilgili kuralları ve cezaları geleneksel bir sözleşmedeki gibi tanımlamaz, bunun yanında bu yükümlülükleri otomatik olarak da uygulayabilir.

Sözleşme imzalanması ile sözleşmede belirtilen kurallara uygun olarak girişe bir değer atar ve bu sözleşmeye bağlı hükümlerin gerektirdiği eylemleri gerçekleştirir. Örneğin, bir varlığın bir kişiye mi gitmesi gerektiğini belirlemek veya koşulların sağlanmadığı durumlarda varlığın kişiye geri dönmesini sağlamak sözleşmenin içerisinde kodlanmıştır. Bu sözleşmeler, bitcoin ve diğer kripto para birimlerinin de temelini oluşturan merkezi olmayan bir defter olan blok zinciri teknolojisi üzerinde depolanmaktadır.

Akıllı sözleşmeler karmaşık bir yapıya sahiptir ve potansiyelleri varlıkların basit transferinin ötesine geçer. Yasal işlemlerden sigorta primlerinden kitle fonlaması sözleşmelerine ve finansal türevlere kadar geniş bir alanda işlem yapabilirler ve işlemleri otomatikleştirebilirler. Ek olarak, akıllı sözleşmelerin sadece süreçleri otomatikleştirmekle kalmayıp aynı zamanda davranışları kontrol etme ve gerçek zamanlı denetim ve risk değerlendirmeleri ile potansiyellerini de yerine getirme kabiliyetine sahiptir.

Akıllı sözleşmeler ifadesi ilk olarak 1994 yılında Nick Szabo tarafından önerildi. Szabo, dijital para birimi için zemin hazırlığı yaptığı bilinen yasal bir alim ve kriptograftır. O zamanlar akıllı sözleşmelere çok az ilgi bulunmamakta idi. Çünkü destekleyecek dijital bir platform ya da dağıtılmış defter ve gerekli hesaplama gücü yoktu. 2008 yılında, ortaya çıkan bitcoin teknolojisi, sözleşmenin tüm şartlarını blok zincirine girmek için kullanılan akıllı sözleşme kodunun geliştirilmesini sağlamıştır.

Artık birçok platform Ethereum ve bitcoin dahil akıllı sözleşmelerin kullanılmasına izin veriyor. Bugün, bitcoinin artan bir şekilde benimsenmesi ve blok zincir teknolojilerinin desteğiyle, akıllı sözleşmeler, genellikle ödemeleri tetiklemek için dijital para birimlerinin üzerine inşa edilen popülerlikle artıyor.

Blok zinciri, teknolojinin güvenliği ve değişmezliği nedeniyle akıllı sözleşmeleri saklamak için idealdir. Akıllı sözleşme verileri paylaşılan bir defterde şifrelenerek bloklarda depolanan bilgilerin kaybolmasını imkansız hale getirir. Blok zinciri teknolojisinin akıllı sözleşmelere dahil edilmesinin bir diğer avantajı esnekliktir. Geliştiriciler bir blok zinciri içinde hemen hemen her tür veriyi depolayabilir ve akıllı sözleşme dağıtımı sırasında aralarından seçim yapabileceğiniz çok çeşitli işlem seçeneklerine sahiptir. Blok zinciri tabanlı akıllı sözleşmeler, yapılacak işlerin ve diğer işlemlerin daha güvenli, verimli ve uygun maliyetli olmasına yardımcı oluyor.

Örneğin, 2016'da Cook County, Ill., Mülk başlıklarını aktarmak ve izlemek amacıyla bir veri tabanı oluşturmak için blok zinciri kullandı. Bu işlemler gerçekleştiğinde, geleneksel senet işlemine ek olarak, alıcı mülkiyet kanıtı olarak dijital bir jeton alır.

Çeşitli sektörler, blok zinciri bazlı akıllı sözleşmelerin tedarik zincirinin bir parçası olarak kullanılmasından faydalanabilir. Sağlık hizmeti ödeme işlemlerini akıllı sözleşmeler kullanarak otomatikleştirmek, aşırı ödemeyi azaltabilir ve sahtekarlığı önleyebilir. Müzik endüstrisi, müzik zincirinin içindeki müziğin sahipliğini kaydedebilir ve müzik ticari amaçlar için kullanıldığında telif ücretlerinin ödenmesini sağlamak için akıllı bir sözleşme uygulayabilir. Akıllı sözleşmeler ve blok zinciri, araç

bakımı, kaza ve mülkiyet geçmişi hakkında hazır bilgileri kolayca depolayarak otomobil endüstrisine fayda sağlayabilir.

Akıllı sözleşmeleri kullanırken bazı potansiyel iş avantajları vardır.

Maliyet: Akıllı sözleşmeler, operasyonel giderlerini ve gelişmelerin izlenmesi için gereken personel ihtiyacını ortadan kaldırır.

İşleme hızı: Akıllı sözleşmeler otomatik süreçler üzerinden yürütülür ve çoğu durumda, insan katılımını ortadan kaldırarak sözleşmede öngörülen ticari işlemlerin hızını artırabilir.

Özerklik: Akıllı sözleşmeler ağ tarafından otomatik olarak gerçekleştirilir ve böylece üçüncü tarafın akıllı sözleşme dağıtımına dahil olma gereksinimi ve buna bağlı riskleri ortadan kaldırır.

Güvenilirlik: Blok zincirine girilen veriler değiştirilemez veya silinemez. Bir tarafın yükümlülüklerini yerine getirmemesi durumunda diğer taraf akıllı sözleşmenin koşullarıyla korunacaktır. Otomatik işlemler ayrıca insan hatası olasılığını ortadan kaldırır ve sözleşmeleri yaparken doğruluk sağlar.

Akıllı sözleşmelerde de birçok potansiyel dezavantaj vardır. Blok zinciri, kripto para birimleri ve akıllı sözleşmelere odaklanan uluslararası düzenlemelerin eksikliği, bu teknolojilerin küresel ekonomide izlenmesini zorlaştırmaktadır. Akıllı sözleşmelerin uygulanması da karmaşıktır. Bu güvenlikle ilgili bir avantaj olarak kabul edilirken, taraflar akıllı sözleşme sözleşmesinde herhangi bir değişiklik yapamaz veya yeni bir sözleşme geliştirmeden yeni detaylar ekleyemez.

3.2. Nesnelerin İnterneti

3.2.1. Nesnelerin interneti nedir?

Nesnelerin İnterneti veya IoT, internete bağlı olan, veri toplayan ve paylaşan dünyadaki milyarlarca fiziksel cihazı ifade eder. Ucuz işlemciler ve kablosuz ağlar sayesinde, herhangi bir şeyi haptan uçağa, otomatik sürüş arabasına ve IoT'nin bir parçası haline getirmek mümkündür. Bu, aksi takdirde düşünme yetisi olmayan cihazlara dijital bir zeka katıyor ve bu sayede insanla uğraşmadan gerçek zamanlı veri iletişimini sağlıyor, dijital ve fiziksel dünyaları etkin bir şekilde birleştiriyorlar.

İnternete bağlanabilir ve kontrol edilebilirse, hemen hemen her fiziksel nesne bir IoT cihazına dönüştürülebilir. Bir akıllı telefon uygulaması kullanılarak açılabilen bir ampul, ofisinizde veya bağlı bir sokak lambasında bir hareket sensörü veya akıllı bir termostat gibi, bir IoT cihazıdır. Bir IoT cihazı, bir çocuk oyuncağı kadar basit ya da sürücüsüz bir araba kadar kompleks olabilir ya da verimli bir şekilde çalıştığından emin olmak için verileri toplayan ve ileten binlerce sensörle dolu bir jet motoru kadar karmaşık olabilir. Daha da büyük ölçekte, akıllı şehirler projeleri çevreyi anlamamıza ve kontrol etmemize yardımcı olmak için tüm bölgeleri sensörler ile doldurabilir. IoT terimi genel olarak internet bağlantısına sahip olması genellikle beklenmeyen ve ağ ile insan eyleminden bağımsız olarak iletişim kurabilen cihazlar için kullanılır. Bu nedenle, bir PC genel olarak IoT cihazı olarak kabul edilmez bunun yanında hiçbir akıllı telefon da IoT sınıfına girmez. Bununla birlikte, bir akıllı saat veya spor bandı veya başka bir giyilebilir cihaz, IoT cihazı olarak sayılabilir.

Temel nesnelere algılayıcı ve internet ekleme fikri 1980'lerde ve 1990'larda tartışıldı, ancak bazı projeler haricinde ilerleme yavaştı çünkü teknoloji hazır değildi. Kullanılabilir ancak tek kullanımlık olacak kadar ucuz ve güçlü olan işlemciler milyarlarca cihaza bağlanmak için uygun maliyetli hale gelmeden önce teknolojinin hazır hale gelmesi gerekiyordu. Kablosuz iletişim kurabilen düşük güçlü yongalar olan RFID etiketlerinin benimsenmesi, bu sorunların bir kısmını çözdü ve geniş bantlı

internet ve hücresel ve kablosuz ağların kullanılabilirliğini arttırdı. IPv6'nın kabul edilmesi IoT'nin ölçeklendirilmesi ve gelişmesi için gerekli en temel adım oldu. Kevin Ashton, teknolojinin vizyonu yakalaması için en az on yıl sürmesine rağmen, 1999'da 'Nesnelerin İnterneti' ifadesini kullandı. Mevcut konumlarını izlemeye yardımcı olmak için pahalı ekipman parçalarına RFID etiketleri eklemek, ilk IoT uygulamalarından biriydi. IoT başlangıçta, makineden makineye (M2M) olarak iş ve imalat sektörleri için tasarlanmıştı, ancak şimdi evlerimizi ve ofislerimizi akıllı cihazlarla doldurmaya devam ediyoruz.

IoT'nin iş dünyasına sağladığı faydalarından biri işletmelerin kendi ürünleri ve kendi iç sistemleri hakkında daha fazla veriye erişebilmeleri ve bunun sonucunda değişiklik yapma konusunda daha büyük imkanlara sahip olmalarıdır.

Üreticiler ürünlerindeki bileşenlere sensörler ekleyebiliyor, böylece performanslarını nasıl yansıttığını öğrenebiliyorlar. Bu, bir bileşenin arızalanması ve zarar vermeden önce onu değiştirmesi muhtemel olduğunda şirketler bunu fark edebiliyorlar ve müdahale edebiliyor. Şirketler ayrıca, bu sensörlerin ürettiği verileri, sistemlerini ve tedarik zincirlerini daha verimli hale getirmek için kullanıyorlar, çünkü gerçekte neler olup bittiğiyle ilgili çok daha doğru veriler olacaktır.

IoT'nin kurumsal kullanımı iki bölüme ayrılabilir: üretim tesisindeki sensörler veya sağlık bakımı için gerçek zamanlı konum cihazları gibi sektörler ve akıllı klima veya güvenlik sistemleri gibi tüm endüstrilerde kullanılabilen IoT cihazları.

3.2.2. Endüstriyel nesnelerin interneti nedir? (IIoT)

Endüstriyel Nesnelerin İnterneti (IIoT) veya dördüncü sanayi devrimi veya Endüstri 4.0, IoT teknolojisinin bir işletme ortamında kullanılması için verilen adlardır. Konsept IoT tüketici için aynıdır; endüstriyel süreçleri ölçmek ve optimize etmek için bir sensör, kablosuz ağ, büyük veri ve analitik kombinasyonunun birleşimiyle oluşur.

Sadece tek tek şirketler yerine tüm tedarik zinciri boyunca tanıtılır, tam zamanında teslimat ve üretimin baştan sona yönetimi ile bu etki daha da büyük hale getirilir. İşgücü verimliliğini veya maliyet tasarrufunu artırmak iki önemli amaçtır.

IoT ile güvenlik en büyük sorunlardan biri. Sensörler birçok durumda son derece hassas veriler topluyor - örneğin kendi evinizde söylediklerimiz kelimeler ve yaptığınız şeyler. Bu güvende tutmak, tüketici güveni için hayati önem taşıyor, ancak şimdiye kadar IoT'nin güvenlik kayıtlarının kaydı oldukça zayıf bir durumdaydı. Çok fazla IoT cihazı, aktarım halindeyken ve dururken verileri şifrelemek gibi güvenlik temelleri kullanmıyor.

IoT, dijital dünya ile fiziksel dünya arasındaki köprüyü oluşturtuluyor, bu da cihazlara saldırmanın tehlikeli gerçek dünya sonuçları doğurabileceği anlamına geliyor. Bir güç istasyonundaki sıcaklığı kontrol eden sensörlere müdahale etmek operatörleri yıkıcı bir karar alma konusunda kandırabilir; Sürücüsüz bir arabanın kontrolünü ele geçirmek de felaketle sonuçlanabilir.

Bir IoT ekosistemi, ortamlarından elde ettiği verileri toplamak, göndermek ve üzerinde işlem yapmak için yerleşik işlemcileri, sensörleri ve iletişim donanımlarını kullanan web özellikli akıllı cihazlardan oluşur. IoT cihazları, topladıkları sensör verilerini bir IoT ağ geçidine veya verilerin yerel olarak analiz edilmek üzere buluta gönderildiği veya yerel olarak analiz edildiği buluta gönderildiği diğer bir uç cihaza bağlayarak paylaşır. Bazen, bu cihazlar diğer ilgili cihazlarla iletişim kurar ve

birbirlerinden aldıkları bilgilere göre hareket eder. Cihazlar işlerin çoğunu insan müdahalesi olmadan yapıyor, ancak insanlar cihazlarla etkileşime girebiliyor.

Eksiksiz bir IoT sistemi dört ayrı bileşeni birleştirir: sensörler / cihazlar, bağlantı, veri işleme ve bir kullanıcı Interface'den oluşur.

- Sensörler / Cihazlar

İlk olarak, sensörler veya cihazlar çevrelerinden veri toplar. Bu sıcaklık okuması kadar basit veya tam bir video beslemesi kadar karmaşık olabilir.

"Sensörler / cihazlar" olarak ikiye ayrılabilir, çünkü çoklu sensörler bir araya toplanabilir veya sensörler bir şeyleri algılamaktan daha fazlasını yapan bir cihazın parçası olabilir. Örneğin, telefonunuz birden fazla sensöre sahip bir cihazdır (kamera, ivmeölçer, GPS, vb.), Ancak telefonunuz yalnızca bir sensör değildir.

- Bağlantı

Sensörlerden/Cihazlardan alının veriler buluta gönderilir, verilerin buluta ulaşması bir bağlantıya ihtiyacı vardır.

Sensörler / cihazlar buluta hücresel, uydu, WiFi, Bluetooth, düşük güçlü geniş alan ağları (LPWAN) veya doğrudan internete ethernet aracılığıyla bağlanma gibi çeşitli yöntemlerle bağlanabilir. Her seçeneğin güç tüketimi, menzil ve bant genişliği arasında farklar vardır. Hangi bağlantı seçeneğinin en iyi olduğunu belirli IoT uygulamasına bağlıdır.

- Veri İşleme

Veri buluta ulaştığında, yazılımda veri üzerinde işlemler gerçekleşir. Bu, sıcaklık okumasının kabul edilebilir bir aralıkta olup olmadığını kontrol etmek gibi çok basit olabilir.

- Kullanıcı arayüzü

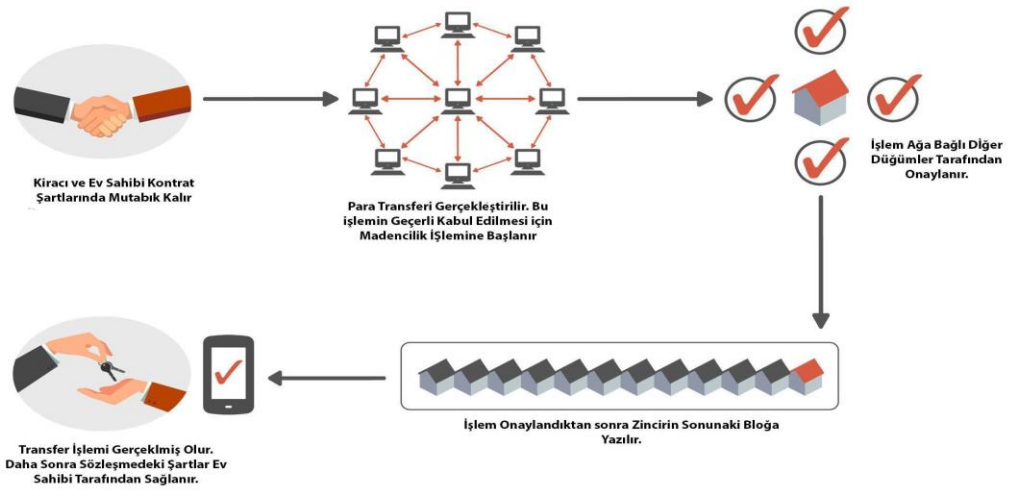
Bulut üzerinde işlenen bilgiler son kullanıcı için bir şekilde faydalı hale getirilir. Bu kullanıcı için bir uyarı yoluyla olabilir (e-posta, metin, bildirim vb.). Örneğin, şirketin soğuk deposunda sıcaklık çok yüksek olduğunda bir metin uyarısı.

Ayrıca, bir kullanıcı proaktif olarak sistemi kontrol etmelerini sağlayan bir ara yüze sahip olabilir. Örneğin, bir kullanıcı, video beslemelerini evlerinde bir telefon uygulaması veya web tarayıcısı üzerinden kontrol etmek isteyebilir.

Her zaman tek yönlü bir bağlantı olmayabilir. IoT uygulamasına bağlı olarak, kullanıcı ayrıca bir eylem gerçekleştirebilir ve sistemi etkileyebilir. Örneğin, kullanıcı, telefonundaki bir uygulama aracılığıyla soğuk hava deposundaki sıcaklığı uzaktan ayarlayabilir ve bazı işlemler otomatik olarak gerçekleştirilir. Sıcaklığı ayarlamanızı beklemek yerine, sistem önceden tanımlanmış kurallarla otomatik olarak yapabilir. Ayrıca, davetsiz misafirleri sizi uyarması için sizi aramak yerine, IoT sistemi aynı zamanda ilgili makamları otomatik olarak haberdar edebilir.

3.3. Blok Zincirinin Akıllı Kira Sözleşmesinde Kullanımı

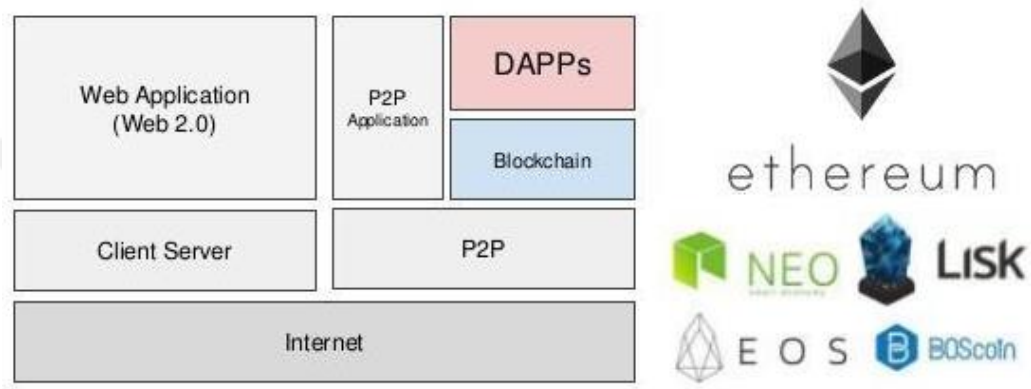
Bu teze konu olan uygulamada 2 önemli bileşen bulunmaktadır. Bunlardan ilki lokal bir fiziksel cihazda parametreler girilerek oluşturulan bir blok zinciri üzerinden hesap oluşturma, hesaplar arasında wei ve ethereum cinsinden para transferinin yapılması ve blok zincirinin temelini oluşturan madencilik işlemi gösterilmiştir. Diğer bileşen ise global bir ağda yani çevrimiçi olan tüm cihazlardan ethereum cinsten para transferi, transfer işleminin teyit uygulaması ile doğrulanması ile birlikte ev sahibi ve kiracı arasında mutabık kalınan sözleşme şartlarının sağlanması için çıktı ifade edilmiştir. Bu uygulama PoW yani yapılabilirliğin ispatı kavramı üzerinden geliştirilmiş olup geliştirilen teyit yazılımı çıktısı sözleşmede bahsedilen evin kullanım izni olarak değerlendirilmiştir.



Şekil 3.9 Kiracı – Ev sahibi arasındaki işlemlerin gösterimi

3.4.Yapılan Çalışmalar

Ethereum Dapps; kullanıcı dostu GUI ile blok zinciri etkileşimi sağlayan uygulamalardır. Dapps'in temel işi sunduğu akıllı sözleşmeler ile istemciler arasında bağlantıyı kusursuz olarak sağlamaktır. Bunu yaparken de HTML ve Javascript kullanır. Sözleşmenin kontrolü süreci pahalı ve zor olduğundan Dapps uygulamaları blok zincirinde bu hesaplamaları yapmak için tasarlanmıştır.



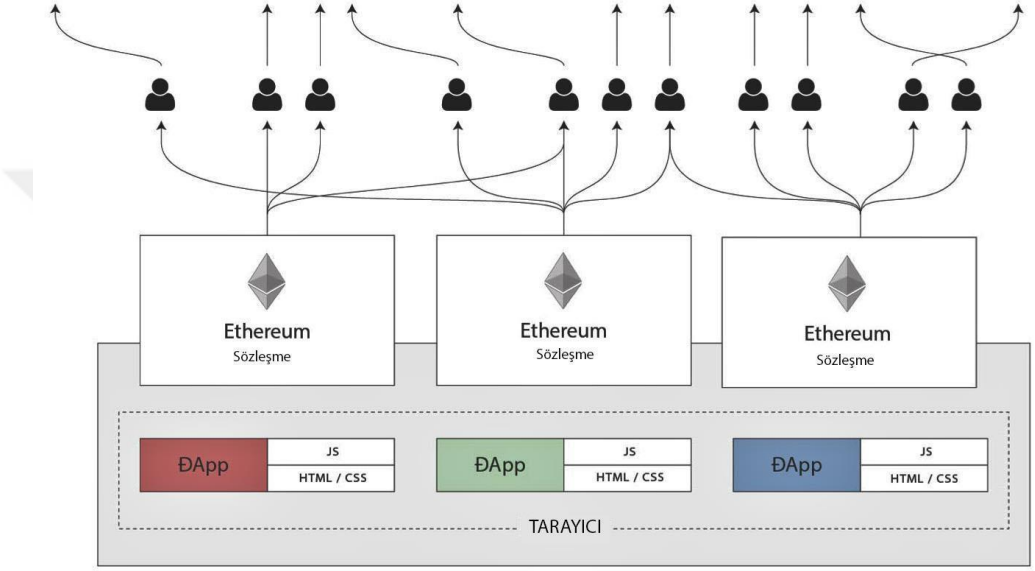
Şekil 3.10 DAPP çalışma grafiği

3.4.1. Web3.js

Ethereum düğümü RPC ara yüzü sağlar. Ara yüz, blok zinciri ile etkileşime geçmek ve blok zincir tarafından sağlanan işlevleri kullanmak için Dapps'a erişim sağlar. Ancak, JSON-RPC ara yüzünü kullanarak iletişim, özellikle Uygulama İkili Arabirimi (ABI) ile çalışırken çok açıktır. Web3.js bir soyutlama katmanı sağlayan bir javascript kütüphanesidir.

Bu durumda Ethereum RPC üstünde çalışır. Kullanıcı dostu ve daha az hata eğilimli olduğundan tercih edilmiştir. Depolama, uygulama-uygulama iletişimi gibi bir web uygulamasının başka bir yönünü merkezden dağıtma çabası olan blok zincirine ek olarak yeni gelişmeler de devam etmektedir. Bu, web uygulamalarının her bileşeninin blok zinciri gibi merkezsizleştirileceği ve bu hizmetlerin çoğuna, gösterildiği gibi web3 API'sinden erişilebilir olacak şekilde dağıtılmış ağa yönelik bir çabadır.

Web3'teki yeni servis desteğinin çoğu, şu anda resmi ifadeye göre piyasaya sürülmemiş olan 1.0 sürümünde mevcut olacak, şu anda Swarm desteği gibi bazı özellikler eksik gözükmektedir. Swarm, dosya içeriğini saklamak için kullanılacak p2p depolama paylaşım ağıdır. Sürü dosyaları içinde içeriklerinin karma tarafından ele alınır. Aynı anda birden fazla düğümden veri almak mümkündür. Bir sistem verileri barındırdığı sürece, tüm ağ üzerinden erişilebilir olacaktır.



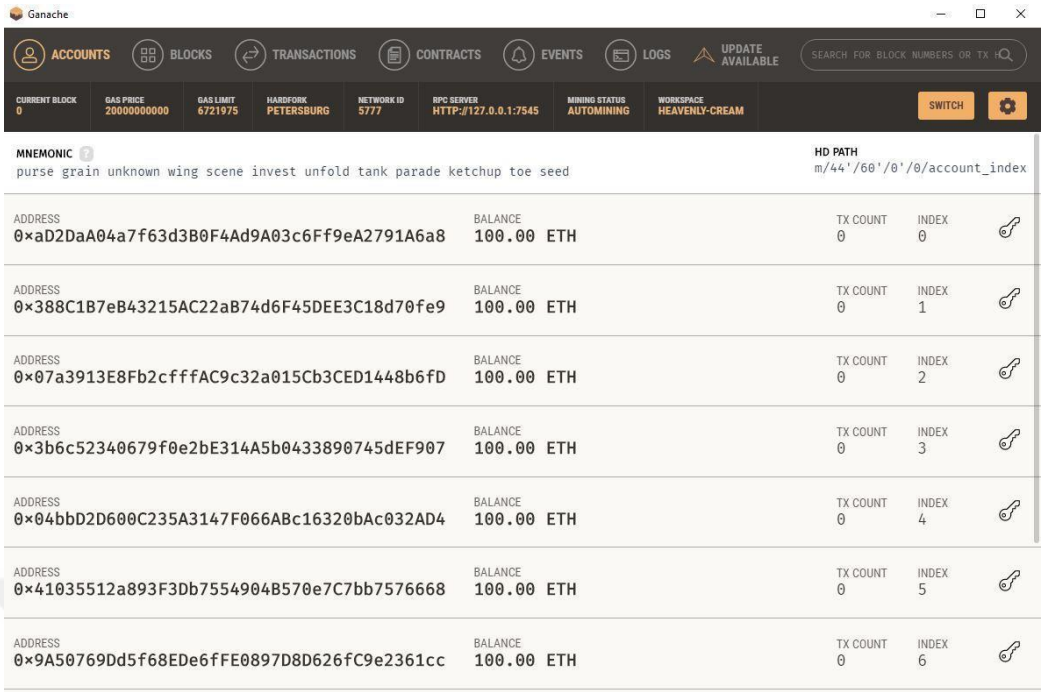
Şekil 3.11 Web3 ile sunulan katman hizmeti şekli

3.4.2. Truffle framework

Truffle Frameworkü; Ethereum geliştiricisi olarak hayatı kolaylaştırmaya çalışan, çerçeve ve varlık hattını test eden bir geliştirme ortamıdır. Bu çalışmada, Truffle akıllı sözleşme derleme ve dağıtım süreci geliştirme çerçevesi olarak kullanılır. Bu framework şu hizmetleri sunar;

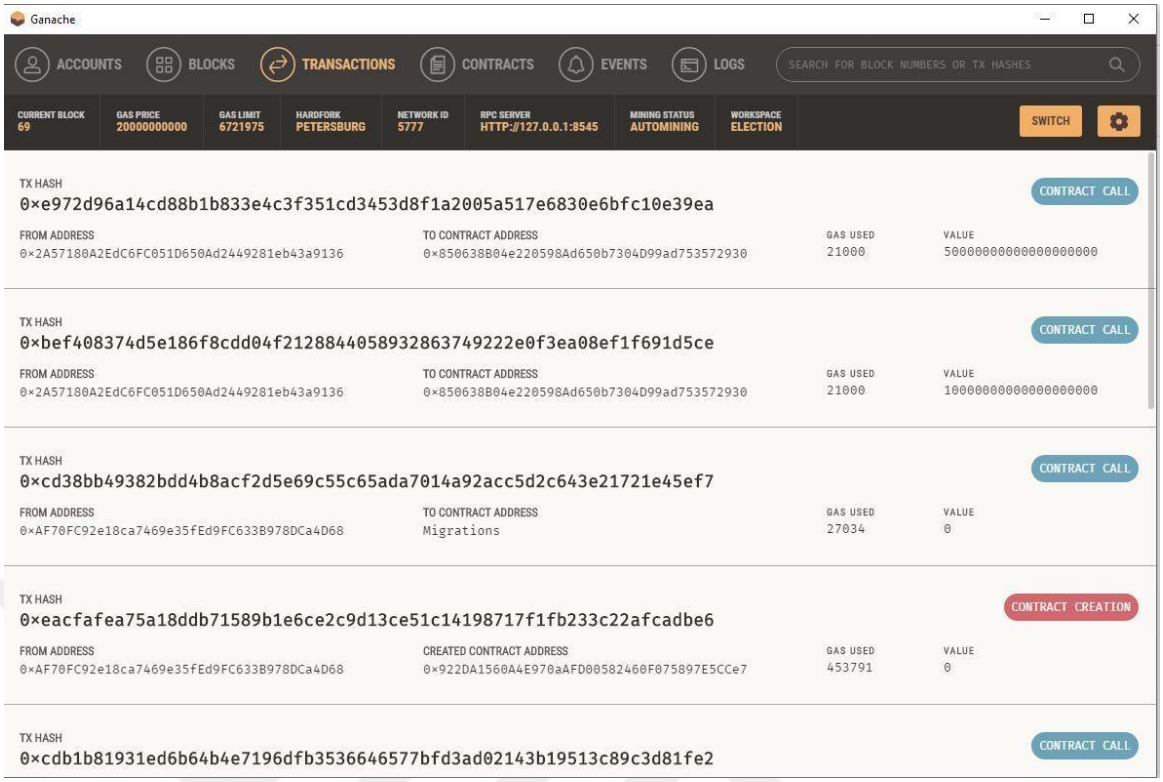
- Entegre akıllı sözleşme derleme, bağlama, dağıtım ve bu süreçlerin binary olarak yönetilmesi,
- Çevik geliştirme için otomatik sözleşme testi,
- Hızlı düzenlenebilirlik ve genişletilebilir dağıtım,
- Herhangi bir veya birden fazla özel ve genel ağlarda çalışma imkanı,
- Doğrudan sözleşme iletişimi için interaktif konsol,
- Sıkı entegrasyon desteği ile yapılandırılabilir yapı iletişim hattı,
- Truffle ortamında komut dosyalarını yürüten harici komut dosyası çalıştırıcısı.

Sunduğu test ortamı ile geliştiricilere blok zincirinde olması zorunlu fonksiyonları doğrudan tanılamakla birlikte hesap işlemlerinin testi için 10 ayrı hesap ile birlikte 100 ethereum para da yüklemektedir.



Şekil 3.12 Truffle Freamwork Ganache test ortamı hesap cetveli

Şekil 3.12'de görüldüğü üzere bu ethereumların mali bir değeri olmamakla birlikte test ortamında kullanılmak üzere tanımlanır.



Şekil 3.13 Truffle Freamwork Ganache test ortamı transfer işlemleri dökümü

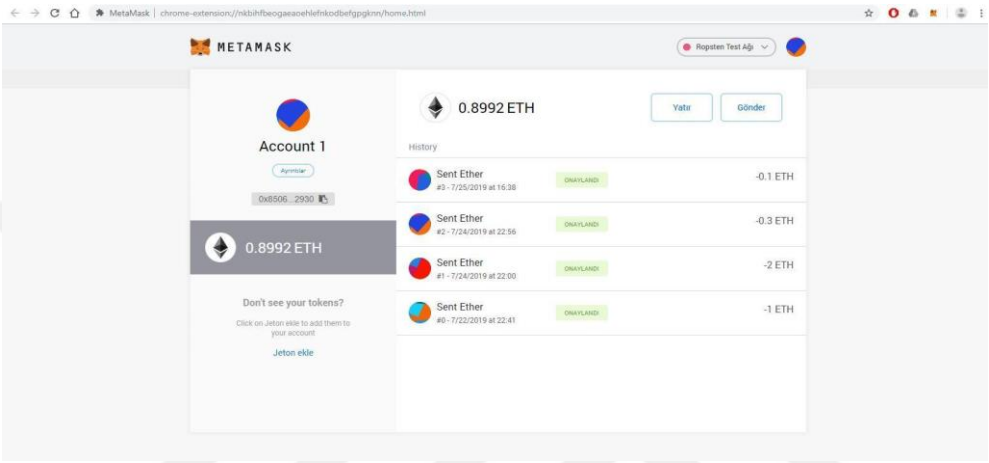
Ayrıca bu uygulama, hesaplar arası yapılan transferlerin gösterilmesi, blok zincirinde üretilen bloklar ve zincirde yapılan tüm işlemlerin incelenmesi imkanı da sunmaktadır.

3.4.3. Bootstrap framework

Bootstrap açık kaynak kodlu bir Front-End web uygulama geliştirme çerçevesidir. Tipografi, düğmeler, formlar, gezinme ve diğer bazı bileşenler için HTML ve CSS tabanlı tasarım şablonları içerir. Bootstrap bir ızgara sistemi veya duyarlı düzen kullanır. Standart 1170 piksel geniş ızgara düzeni ile birlikte gelir. Alternatif olarak, geliştirici değişken genişlikli bir düzen de kullanabilir. Her iki durumda da, araç seti, cep telefonları, portre ve manzara, tabletler ve düşük ve yüksek çözünürlüklü PC'ler gibi çeşitli çözünürlükleri ve evices sınıflarını kullanmak için dört varyasyona sahiptir(Mark Otto,2017).

3.4.4. MetaMask

Bir tarayıcı eklentisi olan MetaMask, kullanıcılara basit bir ara yüzle kripto para transfer işlemini yapmalarını sağlayan bir Javascript tabanlı bir uygulamadır. Bu uygulama doğrudan Ethereum çekirdek ekibi tarafından kullanıcılara güvenli ve esnek işlem kolaylığı sağlamayı amaçlamaktadır.



Şekil 3.14 MetaMask ekran görüntüsü

Yukarıdaki ekran görüntüsünde de görüldüğü üzere bu eklenti ile tanımlanan hesaplar arası gönder-al işlemi kolaylıkla yapılabilmektedir.

3.4.5. GETH

Go diliyle hazırlanmış kullanıcılara blok zinciri için en temel özellikleri sağlayan bir uygulamadır. Son derece basit kullanımı olan bu uygulama ile son kullanıcılar kendi lokal cihazlarında blok zincir kullanarak ethereum ağını oluşturabilir ve akıllı kontrat uygulamalarını geliştirebilirler.

Windows işletim sistemine sahip cihazlarda komut satırı ile çalışan bu uygulamada organize edilen bir genesis blok ile mining yapılıp wei cinsinden maden üretilip farklı kullanıcılara transferi yapılabilmektedir. Bunu bir örnekler anlatmak gerekirse;

İlk adım olarak genesis.json dosyası oluşturulup aşağıdaki ayar betiği dosya içerisine yazılıp kayıt edilir.

```
{
  "config": {
    "chainId": 12345,
    "homesteadBlock": 0
  },
  "alloc": {},
  "coinbase": "0x0000000000000000000000000000000000000000",
  "difficulty": "0x20000",
  "extraData": "",
  "gasLimit": "0x2fefd8",
  "nonce": "0x0000000000000042",
  "mixhash":
"0x0000000000000000000000000000000000000000000000000000000000000000",
  "parentHash":
"0x0000000000000000000000000000000000000000000000000000000000000000",
  "timestamp": "0x00"
}
```

Terimlerin anlaması ise;

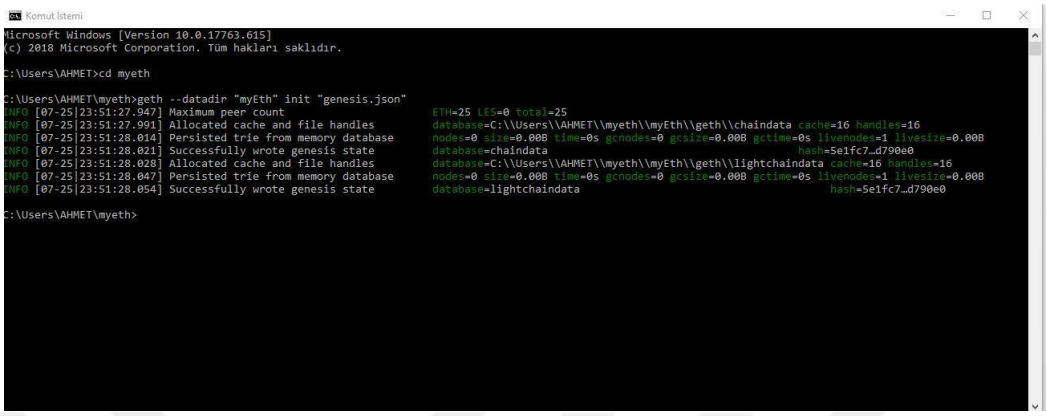
- chainID: Blok zincirini tanımlayan parametredir. Ana ethereum zincirinin kimliği vardır ancak kendi blok zincirimiz için bir değer tanımlamamız gerekir.
- homesteadBlock: Ethereum platformunun üretim yapılabilen ilk sürümüdür. Bu parametre 0 olarak seçildiğinde çalışır.

- Coinbase: Hashleme için başlangıç değeri parametresidir.
- Difficuly: Madencilikte zorluk parametresidir. Hash üretiminde üretim yapacak istemcilerin aşması gereken zorluk değerini ifade eder.
- gasLimit: Blok başına gaz maliyetini tanımlar.
- Nonce: Şifreleme metotlarında kullanılan "Number Used Once"yi ifade eder. Bir sefer kullanılan sayıyı tanımlar.

Windows konsol uygulamasına aşağıdaki kodlar yazılarak blok zincirini oluşturup temel işlemler yapılabilir.

>geth --datadir "myEth" init "genesis.json"

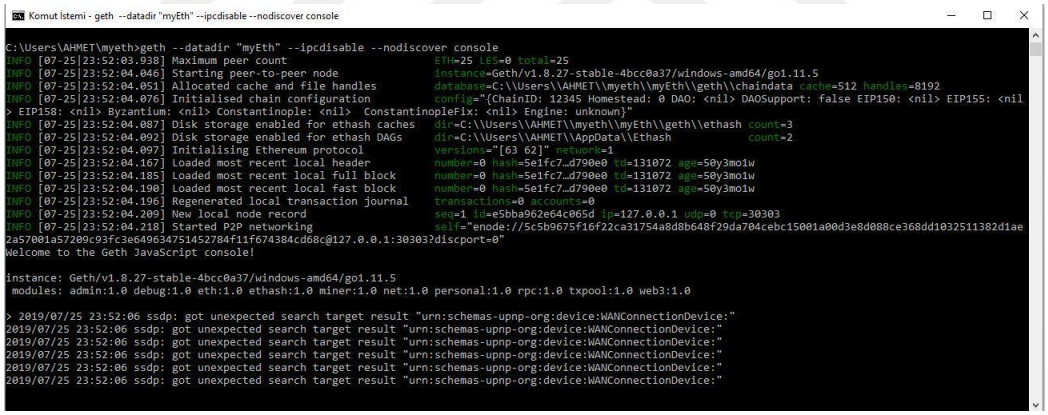
//myEth klasörü içerisinde bulunan genesis.json ile init parametresi girilir. Zincirin temel klasörü işaretlenir.



Şekil 3.15 Blok zincirinin temel klasörünün belirlenmesi

>geth --datadir "myEth" --ipcdisable --nodiscover console

//nodiscover ile özel bir zincir olduğu tanımlanır. İpcdisable ile de IPC-RCP devre dışı bırakılır.



Şekil 3.16 Blok zincirinin türünün belirlenmesi

Bu işlemden sonra geth platformu javascript ile konsoluna geçiş yapılır. Bu konsolda zinir içerisindeki tüm işlemlerin yapılması sağlanır.

>personal.newAccount()

Passphrase:

Repeat Passphrase:

"0xf9b0489c9474edd8a4dde0dcfccec0c803dcbc6f"

//İşlemler için 2 adet hesap tanımlanır.

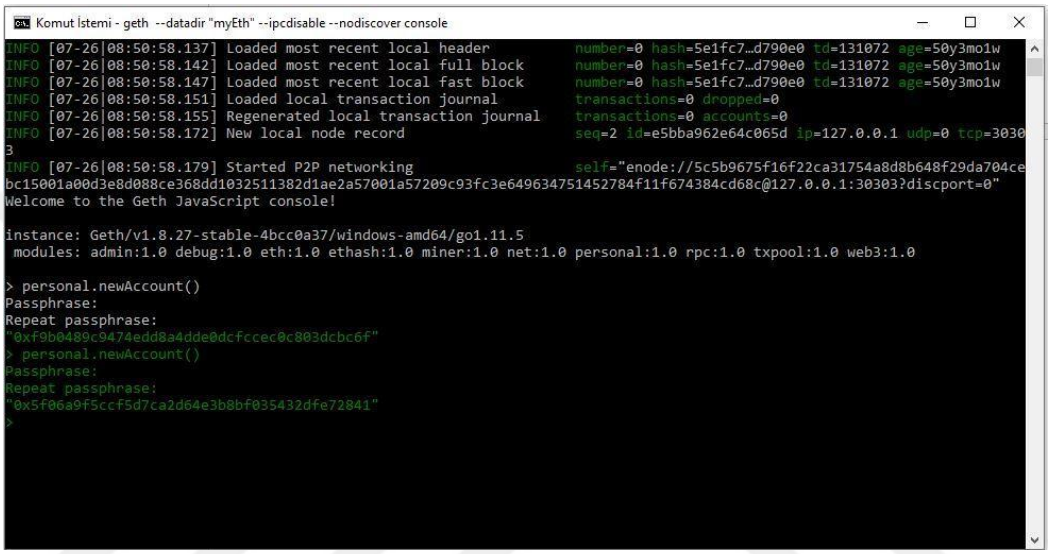
//Hesap 1: 0xf9b0489c9474edd8a4dde0dcfccec0c803dcbc6f

//Hesap 2: 0x5f06a9f5ccf5d7ca2d64e3b8bf035432dfe72841

> personal.listAccounts

["0x1ffd11dc1f0ba911a34eff2193f8ecb7bb0a51e7"]

["0x0214fec507a5af6f34758069203b37ec85888075"]



Şekil 3.17 Blok zincirinde hesap açma işlemi

>eth.blockNumber

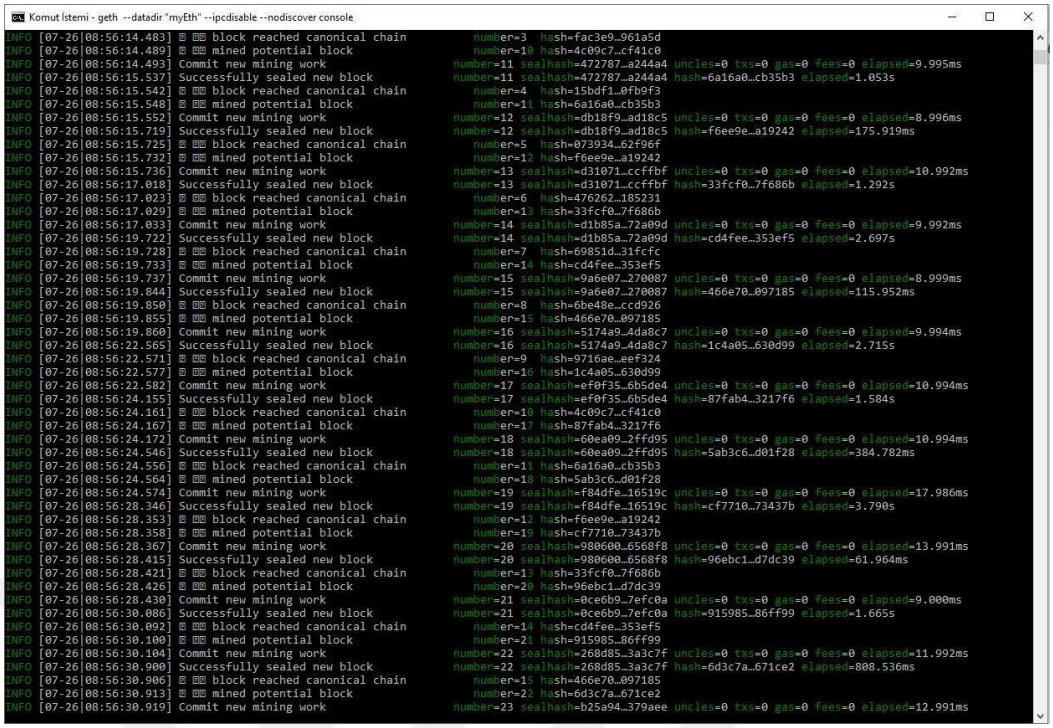
//İşlem yapılan blok numarası



Şekil 3.18 Blok zincirine var olan blok sayısı

>miner.start(1)

//Bu kod ile kazı işlemi başlar. İşlem süresi ayar parametresinde belirtilen zorluk değerine göre zaman alır. DAG prosesi tanımlanır ve bloklara işlemler yazılmaya ve ihtiyaç halinde yeni blokların zincire dahil edilmesi işlemi yapılır.

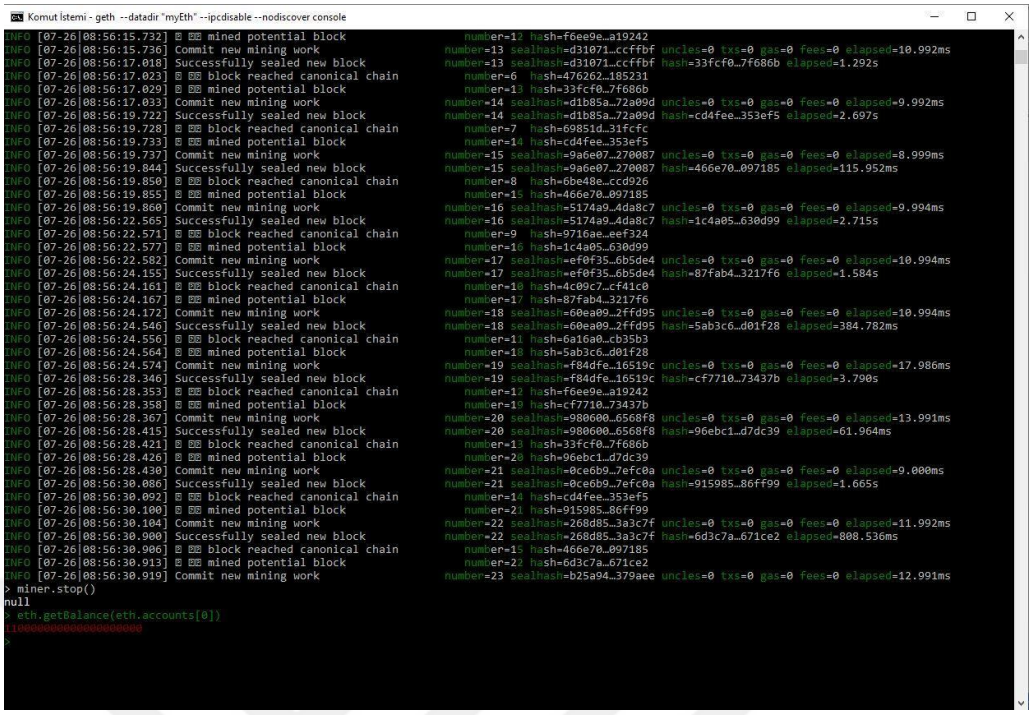


Şekil 3.19 Blok zincirinde madencilik işlemi

Şekil 3.19'da görüldüğü gibi blok zincirinde hash üretim işlemi yapılmıştır Yeni bloklar üretilip madencilere ödül olarak wei cinsinden kripto para verilmiştir.

>miner.stop()
//Kazı işlemi durdurulur.

>eth.getBalance(eth.accounts[0])



Şekil 3.20 Hesap 1'de biriken kripto para

Madencilik işlemi sonrası Hesap 1'de 110000000000000000000 Wei yani 110 ETH üretilmiştir.

//Hesap 1'de biriken madenin wei cinsinden değeri öğrenilir.

>web3.fromWei(eth.getBalance(eth.accounts[0]), "ether")



Şekil 3.21 Üretilen ethereum miktarı

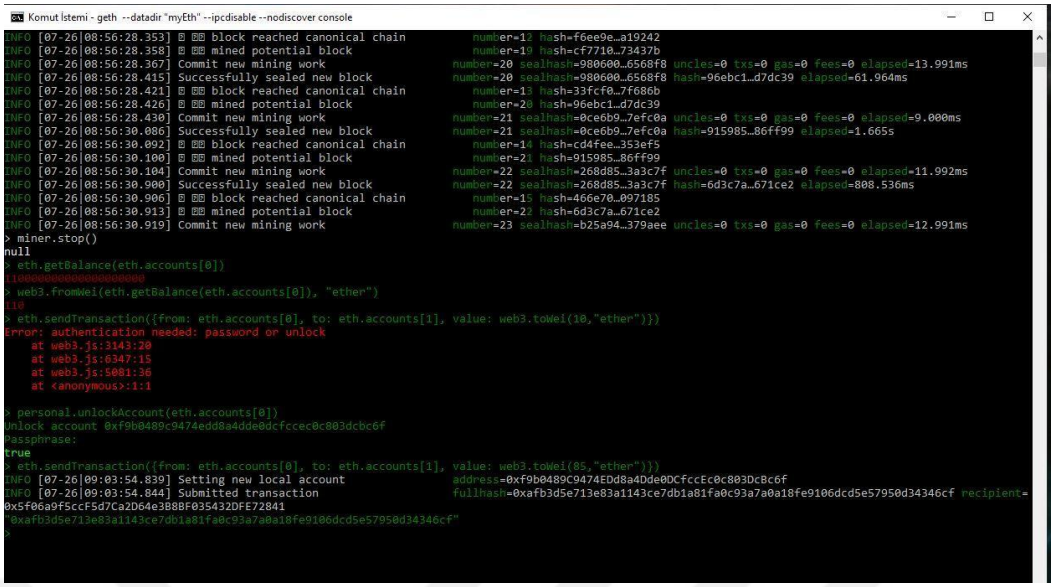
//Hesap 1'de üretilen maden ether cinsine çevrilir.

>personal.unlockAccount(eth.accounts[0])

//Transer etmek için gönderici hesaptaki kilit çözülmelidir.

>eth.sendTransaction({from: eth.accounts[0], to: eth.accounts[1], value: web3.toWei(85,"ether")})

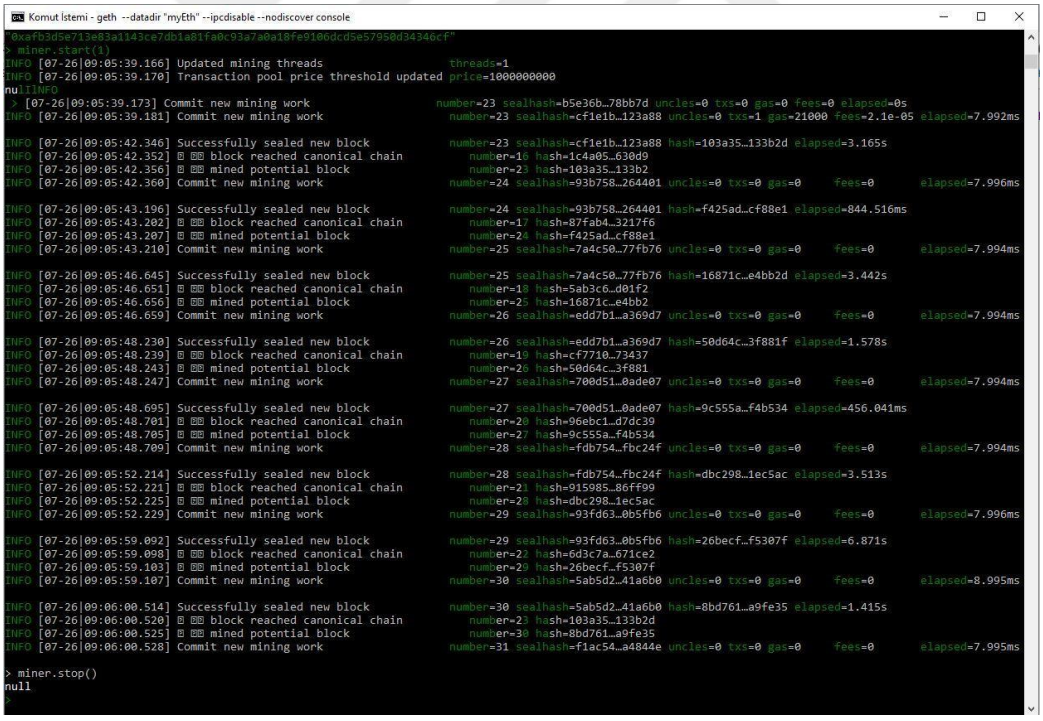
//Bu kod parçacığı ile de Hesap 1'den Hesap 2'ye ether cinsinden 85 adet maden transfer edilir.



Şekil 3.22 Transfer işlemi

Transfer Numarası:

"0xafb3d5e713e83a1143ce7db1a81fa0c93a7a0a18fe9106dcd5e57950d34346cf"



Şekil 3.23 Transfer işlemi onayı için madencilik işleminin yapılması

Burada dikkat edilmesi gereken husus bu işlem ortak konsensüs ile mutabık kalındıktan sonra Defter-i Kebir'e kayıt edilmelidir. Bu işlem için de kazı işleminin devam etmesi gerekir. Eğer etmezse transfer işlem yapılan bloğa kayıt edilmediğinden

onaylanmaz. Yani transfer işlemi gerçekleşmez. Onay için tekrar kazı işlemi yapılmalıdır.



Şekil 3.24 Madencilik sonrası hesap bakiyeleri

Transfer sonrası hesap bakiyelerine bakıldığında transfer işleminin gerçekleştiği gözlemlenmektedir.

Hesap 1: 65 ETH

Hesap 2: 85 ETH

Burada toplam rakamın 110 ETH olması beklenirken 2. Madencilik işleminde onay dışında kazı da yapıldığı için 40 ETH Hesap 1'e aktarılmıştır.

3.4.7. Teyit yazılımı

Bu uygulamada kiracı ve ev sahibi arasındaki sözleşmenin yerine getirilip getirilmediği kontrol edilmektedir. Mutabık kalınan sözleşmede ki tüm şartlar uygulamaya giriş parametresi olarak işlenir. Bu parametreler;

- Kiracı cüzdan numarası
- Ev sahibi hesap numarası
- Birim ücret (günlük)'tir.

Bu parametreleri tanımlamak gerekirse;

API: *2E8P16ITTB4X4MJZ5QST2QUCGI7F33GWHG*

Kiracı Cüzdan Numarası: *0x850638B04e220598Ad650b7304D99ad753572930*

Ev Sahibi Cüzdan Numarası: *0x2A57180A2EdC6FC051D650Ad2449281eb43a9136*

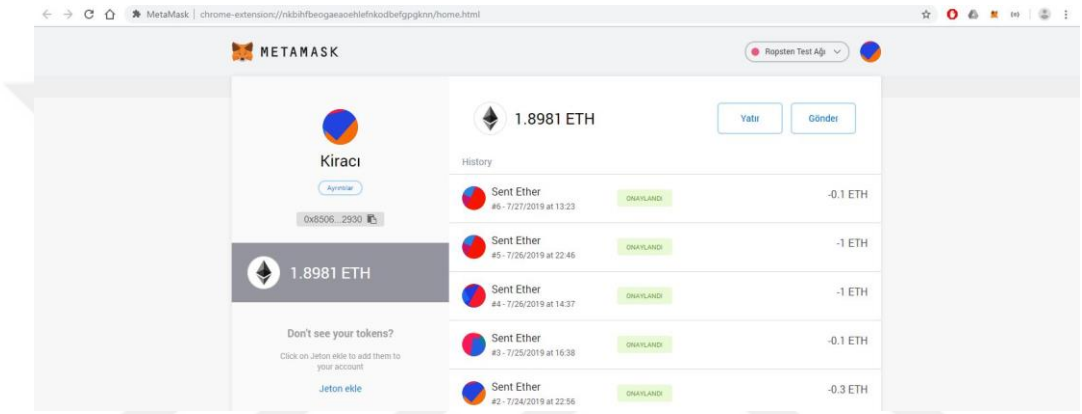
Birim Fiyat: *0,01 ETH / Gün*

Uygulamada transfer işlemini sorgulamak için EtherScan platformundan API erişim yetkisi alınmıştır. Bu işlem WEB API mimarisinde şu işlem bloklarında sorgulara müsaade etmektedir.

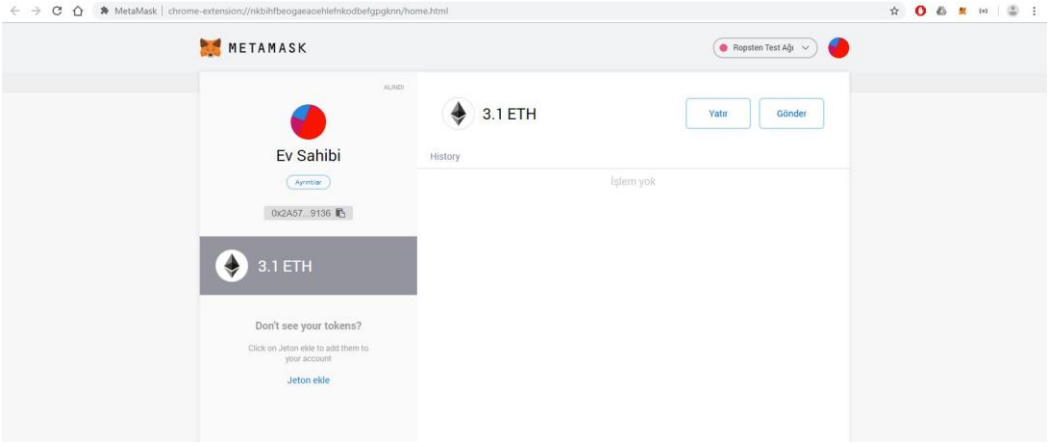
- Hesap işlemleri,
- Sözleşme mutabakat işlemleri,

- Transfer işlemleri,
- Blok işlemleri,
- GETH ortamı işlemleri.

EtherScan platformunda oluşturulan API bilgisi yukarıda ifade edildiği gibi hesap işlemlerinin yapılabilmesi için ise MetaMask eklentisi kurulmuştur. Bu eklenti ile 2 ayrı hesap oluşturulmuş olup bunlar da uygulamaya parametre olarak girilmiştir. Hesapların detayları Şekil 3.25 ve Şekil 3.26'da gösterilmiştir.



Şekil 3.25 Kiracı hesap bilgileri dökümü



Şekil 3.26 Ev sahibi hesap bilgileri dökümü

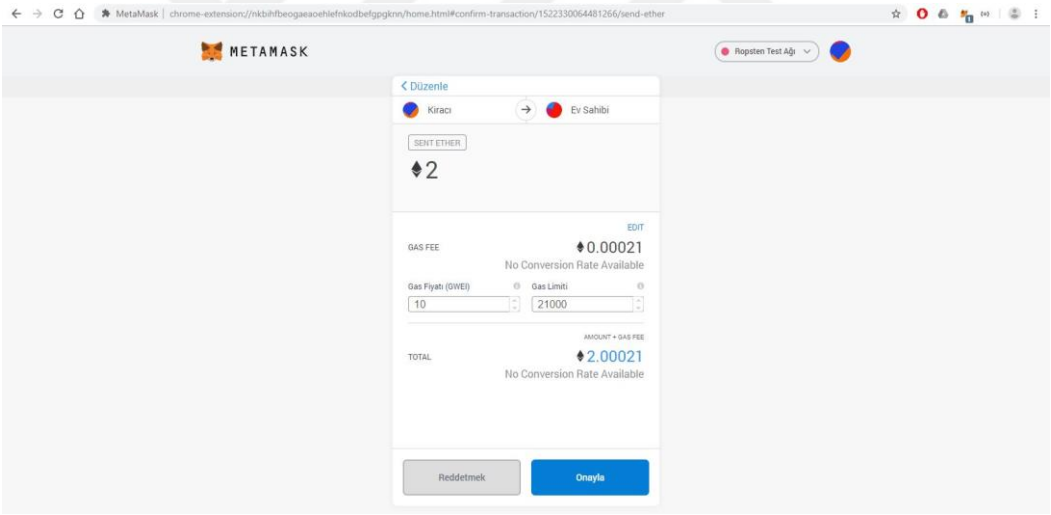
Bu durumda sağlanan API erişimi ile uygulamada transferin gerçekleşip gerçekleşmediği kontrolü ile birlikte Ev Sahibi hesaba yapılan tüm transferler listelenecek olup Kiracı hesap numarasından bir ödeme olup olmadığı kontrol edilmektedir. Daha sonra transfer edilen değer ETH cinsine çevrilecek olup yapılan

transfer ile oranlanacaktır. Daha sonra da bu kapsamda Kiracı'ya mülkün giriş izni ödeme yapılan gün kadar uygulamada bir çıktı ile verilmektedir.

Transfer işlemini başlatmadan önce güncel hesap bakiyeleri Şekil 3.27'de gösterilmiştir. Daha sonra MetaMask uygulaması üzerinden Kiracı hesabından Ev Sahibi hesabına 2 ETH para aktarılır. (Şekil 3.28)

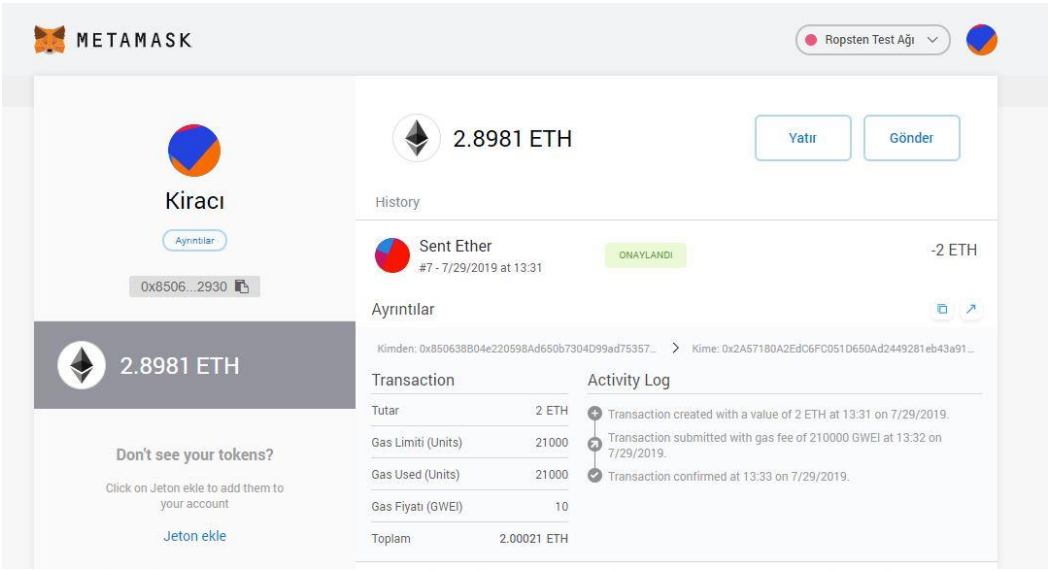
```
{
  "status": "1",
  "message": "OK",
  "result": [
    {
      "account": "0x850638B04e220598Ad650b7304D99ad753572930",
      "balance": "2898110000000000000"
    },
    {
      "account": "0x2A57180A2EdC6FC051D650Ad2449281eb43a9136",
      "balance": "3100000000000000000"
    }
  ]
}
```


Şekil 3.27 Kiracı ve ev sahibi hesap bakiyeleri görüntüleme



Şekil 3.28 Kiracıdan ev sahibine transfer onay ekranı

İşlemin onaylanmasından sonra transfer süreci başlar. İşlem Ropsten test ağında bulunan düğümler tarafından onaylandığında ara yüz ekranına başarılı olduğu bildirimi düşmektedir. (Şekil 3.29)



Şekil 3.29 Kiracıdan ev sahibine transfer aşamaları görüntüsü

Bu işlemden sonra son olarak hesap bakiyelerini kontrol edildiğinde aktarım işleminin başarıyla tamamlandığı gözlemlenebilir. (Şekil 3.30) Bu işlemin kontörlünü sağlayan teyit yazılımının tanımlama kodları şöyledir;

```
cnx = mysql.connector.connect(user='root', password='', host='localhost',
database='py')

import requests
from decimal import Decimal

kiraci = "0x850638B04e220598Ad650b7304D99ad753572930";
evsahibi = "0x2A57180A2EdC6FC051D650Ad2449281eb43a9136";
gunlukbirimucret = 1000000000000000000;
sozlesmebedeli = 2000000000000000000;
tahsissuresi = sozlesmebedeli / gunlukbirimucret;
an = datetime.now()
bitistarihi = datetime.now() + timedelta(days=tahsissuresi)


url = "http://api-ropsten.etherscan.io/api?module=account&action=txlist&address="
+ evsahibi +
"&startblock=0&endblock=99999999&sort=asc&apikey=2E8P16ITTB4X4MJZ5QS
T2QUCGI7F33GWHG";
response = requests.get(url)

address_content = response.json();
result = address_content.get("result")
```

```
for n, transaction in enumerate(result):
    hash = transaction.get("hash")
    tx_from = transaction.get("from")
    tx_to = transaction.get("to")
    value = transaction.get("value")
    confirmations = transaction.get("confirmations")


    print('Transaction ID:')
    print(n)
    print('Hash Degeri:')
    print(hash)
    print('Kimden:')
    print(tx_from)
    print('Kime:')
    print(tx_to)
    print('Deger (wei):')
    print(value)
    eth_value = float (value) / 1000000000000000000
    print("Deger (eth)")
    print(eth_value)
    print('Dogrulama:')
    print(confirmations)
    if int(confirmations) >=16:
        print("Dogrulandi")
    else:
        print("Dogrulanmadi")

    print("\n")
```

```
{
  "status": "1",
  "message": "OK",
  "result": [
    {
      "account": "0x850638B04e220598Ad650b7304D99ad753572930",
      "balance": "897900000000000000"
    },
    {
      "account": "0x2A57180A2EdC6FC051D650Ad2449281eb43a9136",
      "balance": "5100000000000000000"
    }
  ]
}
```


Şekil 3.30 Güncel hesap bakiyeleri

İşlemin başarıyla gerçekleştirildiği kontrol edildikten sonra uygulamada bu işlemin test ekranında ara yüze ve veri tabanına yansımaları Şekil 3.31, Şekil 3.32'de gösterilmiştir.

```
Transaction ID:
6
Hash Degeri:
0xfdf28c4119e6c51e0189d161e450afd5c3144416b9252758ca95a664a759f3e7
Kimden:
0x850638b04e220598ad650b7304d99ad753572930
Kime:
0x2a57180a2edc6fc051d650ad2449281eb43a9136
Deger (wei):
1000000000000000000
Deger (eth)
1.0
Dogrulama:
65
Dogrulandı

Para geldi 2.0 gun kadar ev tahsis edildi. Dosyaya yazildi.
vt yazma basarili

Process finished with exit code 0
```

Şekil 3.31 Uygulama çıktısı

+ Seçenekler

kimden	kime	hash	deger	sure	aktarimtarihi	bitistarihi
1	1	1	1	1	0	0
0	0	0	0	0	0	0
tx_to	0	0	0	0	0	0
0x2a57180a2edc6fc051d650ad2449281eb43a9136	0x850638b04e220598ad650b7304d99ad753572930	0xe0ee00827659ebc68cc3d6e607066a396dff232cfb7ceaf3...	0	0	0	0
0x2a57180a2edc6fc051d650ad2449281eb43a9136	0x850638b04e220598ad650b7304d99ad753572930	0xe0ee00827659ebc68cc3d6e607066a396dff232cfb7ceaf3...	0	2	2019-08-02 00:27:00.132052	0
0x2a57180a2edc6fc051d650ad2449281eb43a9136	0x850638b04e220598ad650b7304d99ad753572930	0xe0ee00827659ebc68cc3d6e607066a396dff232cfb7ceaf3...	0.1	2	2019-08-02 00:29:18.570633	0
0x2a57180a2edc6fc051d650ad2449281eb43a9136	0x850638b04e220598ad650b7304d99ad753572930	0xe0ee00827659ebc68cc3d6e607066a396dff232cfb7ceaf3...	0.1	2	2019-08-02 00:31:58.986526	0
0x2a57180a2edc6fc051d650ad2449281eb43a9136	0x850638b04e220598ad650b7304d99ad753572930	0xfc0a96e0518ae5ded0edd7af11ce3437c3fb170bd53e4ae4...	1	2	2019-08-25 12:06:50.509795	0
0x2a57180a2edc6fc051d650ad2449281eb43a9136	0x850638b04e220598ad650b7304d99ad753572930	0xfc0a96e0518ae5ded0edd7af11ce3437c3fb170bd53e4ae4...	1	2	2019-8-25	0
0x2a57180a2edc6fc051d650ad2449281eb43a9136	0x850638b04e220598ad650b7304d99ad753572930	0xfdf28c4119e6c51e0189d161e450afd5c3144416b9252758...	1	2	2019-08-25 13:02:57.128845	0
0x2a57180a2edc6fc051d650ad2449281eb43a9136	0x850638b04e220598ad650b7304d99ad753572930	0xfdf28c4119e6c51e0189d161e450afd5c3144416b9252758...	1	2	2019-08-25 13:03:28.706705	2019-08-26 13:03:28.706705
0x2a57180a2edc6fc051d650ad2449281eb43a9136	0x850638b04e220598ad650b7304d99ad753572930	0xfdf28c4119e6c51e0189d161e450afd5c3144416b9252758...	1	2	2019-08-25 13:04:07.828762	2019-08-27 13:04:07.828762

Şekil 3.32 Veri tabanı ara yüzü çıktısı

Teyit yazılımında işlemin gerçekleşip gerçekleşmediğini kontrol eden kod kısmı ise aşağıda belirtilmiştir.

```
if tx_to == "0x2a57180a2edc6fc051d650ad2449281eb43a9136":
   print("Para geldi " + str (tahsissuresi) + " gun kadar ev tahsis edildi. Dosyaya
yazildi.")
   file = open("sonuc.txt", "w")
   file.write("Para geldi " + str (tahsissuresi) + " gun kadar ev tahsis edildi")
   file.close()

   cursor = cnx.cursor()
   sql = """INSERT INTO
tablo1(kimden,kime,hash,deger,sure,aktarimtarihi,bitistarihi)
          VALUES (%s, %s, %s, %s, %s, %s, %s)"""
   val = (tx_to,tx_from,hash,eth_value,tahsissuresi,an,bitistarihi)
   try:
      cursor.execute(sql,val)
      cnx.commit()
      print("vt yazma basarili")
   except:
      cnx.rollback()
      print("vt yazma basarili degil")
   cnx.close
```

4. ARAŞTIRMA BULGULARI ve TARTIŞMA

Bu çalışmada blok zinciri metodunun çalışma prensibi araştırılmıştır. Bununla birlikte yerel bir cihazda amaca yönelik bir blok zinciri oluşturulmuş, bu blok zincirinde kripto para madenciliği yapılmış, üretilen paralar hesaplar arası transfer edilmiştir. Nesnelerin interneti teknolojisinde güvenlik sorununun çözümüne yönelik olarak Ropsten test ağında da hesaplar arası transferin kontrolünü yapan bir teyit yazılımı üretilmiştir.

Geliştirilen uygulamada amaç, bir otoriteye ihtiyaç duymadan hesaplar arası para transferinin gerçekleşmesi sürecini teyit etmek ve mutabık kalınan sözleşmeye tarafların uymasını sağlamaktır. Durum böyleyken test ortamında geliştirilen uygulama çıktısı bir veri tabanına kayıt edilerek diğer senaryoların ad kullanımında mümkün olacaktır.

Yerel olarak geliştirilen blok zincirinde GETH yazılımı kullanılmıştır. Ropsten test ağında yapılan işlemlerin teyidini sağlayan yazılım ise Python diliyle yazılmış ve bununla birlikte program çıktılarını MySQL veri tabanına kayıt etmektedir.

Bu çalışmada günlük hayatta kullanılan metodların dezavantajı olan bir otorite varlığı ve bu otoriteden doğan maliyetlerin ortadan kaldırılmasını mümkün kılan bir konsept ortaya konulmuş ve gerçekte taraflar arasındaki sözleşme şartlarına istisnasız uyulmasını sağlayan bir teyit yazılımı üretilmiştir.

5. SONUÇ ve ÖNERİLER

5.1. Sonuçlar

Bu çalışmada bir blok zincirinin mimarisi ve çalışma mantığı araştırılıp incelenmiştir. Lokalde çalışan bir blok zinciri kurgulanıp gerçeklenmiş ve hesap işlemleri test edilmiştir. Bununla birlikte gerçek hatatta yer edinmesi adına tez senaryosuna konu olan kiracı ve ev sahibi için birer ethereum hesabı tanımlanmış ve bu hesaplar üzerinden test ethereumları transfer edilip teyit yazılımı ile doğrulanmıştır.

Nesnelerin interneti teknolojisinin en büyük sorunu olan güvenlik ve mahremiyet kavramlarına en düşük maliyetle blok zinciri teknolojisi kullanılarak geliştirilen metotta ev sahibi ve kiracı yani alışveriş yapan 2 tarafın işlemlerin tamamını çevrimiçi yapmaları sağlanmıştır.

Alışverişi tanımlayan sözleşmeye konu olan tüm şartları parametre girdisi olarak alan teyit yazılımı ile hata payı, blok zinciri teknolojisinin hata payına eşitlenmiştir ki geleneksel ve güncel yöntemlerle en az hata payına sahip yöntemdir.

Kiracı ve ev sahibi arasındaki alışveriş senaryosunu konu alan bu çalışmada nesnelerin interneti tarafı üzerine yoğunlaşılmamış olmasına karşın teyit yazılımı çıktısı aygın kullanılan IoT teknolojilerine girdi olarak sağlanabilmektedir.

Yine sözleşmeye konu olan maddi miktar minimum masrafla değiştirilemez bir zinciri üzerinden transfer edildiği için loglama yani yıllara sari kayıt altına alınması ve işlemin müdahaleye açık olmaması bu çalışmayı uygulanabilir kılmaktadır.

Yapılan çalışma ile geleneksel yöntemlerle ciddi anlamda sorun teşkil eden yöntem çevrimiçi bir sistem ile şeffaf, güvenlikli ve kullanıcı dostu olarak gerçeklenmiş olup literatüre kazandırılmıştır.

5.2.Öneriler

Günümüzde para transferine dayanan tüm işler için kullanılabilecek olan blok zinciri teknolojisinin yaygınlaşması için kamu otoriteleri tarafından desteklenmesi kırılma noktası olacaktır. Ülkemizde bu teknolojinin yaygınlaşması adına hayati bir adım atılmıştır. Cumhurbaşkanlığı tarafından onaylanan 11. Kalkınma Planı 249-5 maddesinde "Blokzincir tabanlı dijital merkez bankası parası uygulamaya konulacaktır." ifadesine yer verilmiş olması bu alana gerekli önemin verileceğinin işareti niteliğindedir.

Sadece finans piyasalarında bu teknoloji kullanmak yaygınlaştırmak adına ilk adım olabilir. Bu teze konu olan kiracı-ev sahibi alışverişi uygulaması günlük hayatta kullanılabilecek alanlardan tek bir tanesidir.

Teknik olarak ise blok zincirlerinin özelleşerek ihtisaslaştırılması ve hatta iş odaklı blok zinciri oluşturulması yaygınlaşma için önemli bir etken olacaktır.

KAYNAKLAR

HE W, YAN G and DA XU L, 2014. Developing vehicular data cloud services in the IoT environment. Ind Inform IEEE Trans, 10(2):1587–1595.

YANG G, 2014. A health-IoT platform based on the integration of intelligent packaging, unobtrusive bio-sensor, and intelligent medicine box. Ind Inform IEEE Trans, 10(4):2180–2191.

MACGİLLİVRAY C, TURNER V and LUND D, 2013. Worldwide internet of things (iot) 2013 -2020 forecast: billions of things, trillions of dollars. IDC. Doc, 243661(3).

A. ROWSTRON and P. DRUSCHEL, 2001. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. IFIP/ACM International Conference on Distributed Systems Platforms and Open Distributed Processing, 1:329-350.

E. ADAR and B. A. HUBERMAN, 2000. Free riding on Gnutella,. First monday, 5:1-10.

M. RİPEANU, 2001. Peer-to-peer architecture case study: Gnutella network. Proceedings. First International Conference, 2001, pp. 99-100.

B. COHEN, 2003. Incentives Build Robustness in BitTorrent. Workshop on Economics of Peer-to-Peer systems, 68-72.

A. R. BHARAMBE, C. HERLEY and V. N. PADMANABHAN, 2005. Analyzing and improving bittorrent performance. Microsoft Research, Microsoft Corporation One Microsoft Way Redmond WA, 98052:100-116.

Y.-F. CHEN, Y. HUANG, R. JANA, H. JİANG, M. RABİNOVİCH, B. WEİ and Z. XİAO, 2017. When is P2P technology beneficial for IPTV services. Proceedings of the 17th International Workshop on Network and Operating System Support for Digital Audio and Video, 1:04-05.

S. DOUGLAS, E. TANİN, A. HARWOOD and S. KARUNASEKERA, 2005. Enabling massively multi- player online gaming applications on a p2p architecture. Proceedings of the IEEE international conference on information and automation, 5:7-12.

J. SEEDORF, S. KİESEL and M. STİEMERLİNG, 2009. Traffic localization for P2P-applications: The ALTO approach. IEEE Ninth International Conference on, 1:171-177.

NAKAMOTO, S., 2008. Bitcoin: A Peer-to-Peer Electronic Cash System.

BRODY P. and PURESWARAN, V., 2014. Device Democracy: Saving the Future of the Internet of Things. IBM.

MILLS, D. C. A. WANG, K. A. MALONE, B. A. RAVI, A. A. MARQUARDT, J. C. A. BADEV, A. I. A. BREZINSKI, T. A. FAHY, L. A. LIAO, K. A. KARGENIAN and V. A. OTHERS, 2016. Distributed ledger technology in payments, clearing, and settlement.

CACHİN, 2016. Architecture of the Hyperledger blockchain fabric, IBM, 2016.

S. DAVIDSON, P. D. FILIPPI and J. POTTS, 2016. Disrupting Governance: The New Institutional Economics of Distributed Ledger Technology.

K. PETERSON, R. DEEDUVANU, P. KANJAMALA and K. BOLES, 2016. A blockchain-based approach to health information exchange networks. Proc NIST Workshop Blockchain Healthcare, 1: 1-10.

K. KORPELA, J. HALLIKAS and T. DAHLBERG, 2017. Digital supply chain transformation toward blockchain integration. 50th Hawaii international conference on system sciences.

Y.-P. LIN, J. R. PETWAY, J. ANTHONY, H. MUKHTAR, S.-W. LIAO, C.-F. CHOU and Y.-F. HO, 2017. Blockchain: The Evolutionary Next Step. ICT E-Agriculture, Environments, 4:50.

A. DORRI, S. S. KANHERE and R. JURDAK, 2016. Blockchain in internet of things: challenges and solutions.

J. SUN, J. YAN and M. A. Z. K. ZHANG, 2016. Blockchain-based sharing services: What blockchain technology can contribute to smart cities. Financial Innovation, 2: 26.

O. JACOBOVITZ, 2016. Blockchain for identity management. The Lynne and William Frankel for Computer Science Department of Computer Science, Ben-Gurion University, Beer Sheva.

A. ARRENDONDO, 2018. Arredondo, Abelardo Blockchain and certificate authority cryptography for an asynchronous on-line public notary system. The University of Texas.

AGGARWAL, R. and LAL DAS, M., 2012. RFID Security in the Context of "Internet of Things". First International Conference on Security of Internet of Things, 1:51-56.

LİANOS, M. and DOUGLAS, M., 2000. Dangerization and the End of Deviance: The Institutional Environment. British Journal of Criminology, 40: 261-278.

FERGUSON, T., 2002. Have Your Objects Call My Object. Harvard Business Review, June: 1-7.

BİDDLECOMBE, E.,2009. UN Predicts "Internet of Things". Retrieved July 6.

COURTOİS, NİCOLAS T., 2003. Fast algebraic attacks on stream ciphers with linear feedback. Annual International Cryptology Conference. Springer, Berlin, Heidelberg, 1-5.

THOMAS, H. BAR;, 2002. Invitation to Cryptology. Upper Saddle River, Prentice Hall, s243-314

MENEZES A. J.; OORSCHOT P. C. VAN and VANSTONE S. A., 1996. Handbook of Applied Cryptography. CRC Press, 31-32.

KIRAZ MEHMET SABIR, "Blokzincir Teknolojisinde Güvenlik ve Mahremiyet".

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Ahmet TABANLIOĞLU
Uyruğu : T.C.
Doğum Yeri ve Tarihi : Tokat-Zile – 01/09/1992
E-mail : ahmet.tabanlioglu@hotmail.com
Tel : 0542 597 0664

EĞİTİM

Derece	Adı, İlçe, İl	Bitirme Yılı
Lise	: Zile Dinçerler Lisesi, Tokat	2010
Üniversite	: Harran Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü, Şanlıurfa	2015
Yüksek Lisans	: Harran Üniversitesi Fen Bilimleri Enstitüsü Elektrik-Elektronik Mühendisliği Anabilim Dalı, Şanlıurfa	2019

İŞ DENEYİMLERİ

Yıl	Kurum	Görevi
2014-2015	Yaba Ajans Reklamcılık Org.Tan.Tur.İnş.San.Tic.Ltd.Şti	Bilgi-İşlem Sorumlusu
2015-2017	Vidora R&D	Genel Müdür / Ar-ge Mühendisi
2015-Devam Ediyor	GAP Bölge Kalkınma İdaresi	IT Uzmanı

YAYINLAR

TENEKECİ M. E., GÜMÜŞÇÜ A., BEŞLİ N., İLKHAN M. A., ABAMOR S., and TABANLIOĞLU A., 2018. Energy Efficiency in Residences via Internet of Things (IoT). GAPYENEV2018, 1:232-233.

TABANLIOGLU A., NACAR M. A., TENEKECİ M. E., GUMUSCU A. And KARADAG K., 2018. Energy Consumption and Sustainability on Bitcoin Mining. , GAPYENEV2018, 1:270-271.

TABANLIOGLU A., TENEKECI M. E., NACAR M. A., 2019. Blok Zinciri İle Sensor Network'teki Nesnelerin Mahremiyet Problemine Bir Çözüm Yaklaşımı, AB2019, 1:20-22.