



**SAĞLIK VERİLERİNİN GÜVENLİ YÖNETİMİ İÇİN BLOKZİNCİR
TABANLI İOT-BULUT MİMARİSİ**

Rukiye Nur ÇAYAN

**YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANA BİLİM DALI**

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

HAZİRAN 2025

ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmada yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Rukiye Nur ÇAYAN

30/06/2025

SAĞLIK VERİLERİNİN GÜVENLİ YÖNETİMİ İÇİN BLOKZİNCİR TABANLI IOT- BULUT MİMARİSİ

(Yüksek Lisans Tezi)

Rukiye Nur ÇAYAN

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Haziran 2025

ÖZET

Nesnelerin İnterneti (Internet of Things/IoT), hızla yayılmakta ve günlük hayatımıza sorunsuz bir şekilde entegre olmaktadır. Ayrıca internete bağlanan nesnelerin sayısı giderek artmaktadır. IoT uygulamaları; akıllı evler, akıllı şehirler, ulaşım ve sağlık hizmetleri gibi çeşitli alanlara yayılmaktadır. Bu alanlar arasında, uzaktan sağlık takibini mümkün kılması nedeniyle akıllı sağlık sistemleri özellikle önemli bir yer tutmaktadır. Akıllı sağlık sistemleri aracılığıyla hasta verileri ağ sistemleri üzerinden iletilerek uzaktan sağlık izleme gerçekleştirilmektedir. Ancak bu sistemlerde, hasta sağlık verilerinin gizliliğinin ve bütünlüğünün korunması konusunda önemli sorunlar devam etmektedir. Bu çalışma, söz konusu sorunları ele almak üzere SağlıkZinciri adlı yeni bir model önermektedir. Bu modelde, veri toplamak için IoT cihazları ve verileri analiz etmek için kenar bilişim kullanılmaktadır. Ayrıca veri bütünlüğünü sağlamak için blokzinciri ile entegre akıllı sözleşmeler ve verileri güvenli bir şekilde saklamak için blokzinciri teknolojisi kullanılmaktadır. Bu kapsamda, bu çalışmada önerilen model; gizlilik, bütünlük ve erişilebilirlik gibi temel bilgi güvenliği ilkeleri çerçevesinde değerlendirilmiş ve modelin çeşitli saldırılara karşı dayanıklılığı tartışılmıştır. Ayrıca önerilen model üzerinde, Blowfish-CBC algoritması HMAC-SHA256 ile birlikte, AES-GCM ve ChaCha20-Poly1305 olmak üzere üç farklı şifreleme ve bütünlük algoritması uygulanmış ve uygulama sonucunda ChaCha20-Poly1305 algoritmasının diğerlerine kıyasla daha iyi performans gösterdiği görülmüştür. Önerilen model, kullanıcı sayısı arttıkça veri iletimi ve şifreleme süreçlerini güçlendirmekte, işlem hacmini artırmakta ve gecikmeyi azaltmaktadır. Ayrıca çalışma kapsamında akıllı sözleşme mekanizması da değerlendirilmiş ve bu mekanizmanın veri bütünlüğünü yönetmede güvenilir olduğu tespit edilmiştir. Sonuç olarak, SağlıkZinciri modeli, sağlık verilerinin ağ üzerinden güvenli bir şekilde ve veri bütünlüğü bozulmadan iletilmesini sağlamaktadır. Ayrıca bu model, verinin bütünlüğünü, gizliliğini ve güvenliğini koruyarak, geleneksel yaklaşımlara kıyasla sağlık hizmetlerinin kalitesini ve güvenilirliğini artırmaktadır.

Bilim Kodu : 92435

Anahtar Kelimeler : Nesnelerin interneti, blokzinciri, kenar bilişim, sağlık verisi, gizlilik

Sayfa Adedi : 74

Danışman : Dr. Öğr. Üyesi Feyza YILDIRIM OKAY

BLOCKCHAIN-BASED IOT-CLOUD ARCHITECTURE FOR SECURE
MANAGEMENT OF HEALTH DATA

(M. Sc. Thesis)

Rukiye Nur ÇAYAN

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

June 2025

ABSTRACT

The Internet of Things (IoT) is rapidly expanding and seamlessly integrating into daily life, with the number of connected devices increasing steadily. IoT applications are spreading across various fields, including smart homes, smart cities, transportation, and healthcare services. Among these areas, smart healthcare systems hold particular significance due to their ability to enable remote health monitoring. Through such systems, patient data is transmitted via network infrastructures, facilitating remote health surveillance. However, significant challenges remain in ensuring the confidentiality and integrity of patient health data in these systems. To address these issues, this study proposes a new model called SağlıkZinciri. In this model, IoT devices are used for data collection and edge computing is employed for data analysis. Additionally, blockchain-integrated smart contracts are utilized to ensure data integrity, and blockchain technology is used to securely store the data. Within this scope, the proposed model has been evaluated in terms of core information security principles such as confidentiality, integrity, and availability, and its resilience against various attacks has been discussed. Moreover, three different encryption and integrity algorithms were implemented within the proposed model: Blowfish-CBC combined with HMAC-SHA256, AES-GCM, and ChaCha20-Poly1305. As a result of the implementation, the ChaCha20-Poly1305 algorithm demonstrated better performance compared to the others. The proposed model enhances data transmission and encryption processes, increases transaction throughput, and reduces latency as the number of users increases. Additionally, the smart contract mechanism was also evaluated in the study and was found to be reliable in managing data integrity. In conclusion, the SağlıkZinciri model ensures the secure transmission of health data over the network without compromising data integrity. Furthermore, by protecting the integrity, confidentiality, and security of the data, this model enhances the quality and reliability of healthcare services compared to traditional approaches.

Science Code : 92435

Key Words : IoT, blockchain, edge, health data, privacy

Page Number : 74

Supervisor : Assist. Prof. Dr. Feyza YILDIRIM OKAY

TEŞEKKÜR

Bu çalışmada değerli fikirleri ve rehberliği ile yol gösteren sayın danışmanım Dr. Öğr. Üyesi Feyza YILDIRIM OKAY'a, bana her koşulda inanan ve daima yanımda olan kıymetli anneme ve kız kardeşlerime içten şükranlarımı sunarım. Ayrıca, Bilim İnsanı Destek Programları Başkanlığı (BİDEB) eğitim ve burs destekleri kapsamında beni destekleyen ve tezimin gerçekleştirilmesine katkıda bulunan Türkiye Bilimsel ve Teknolojik Araştırma Kurumu (TÜBİTAK)'na minnettarlığımı sunarım.



İÇİNDEKİLER

	Sayfa
ÖZET	iv
ABSTRACT.....	v
TEŞEKKÜR.....	vi
İÇİNDEKİLER	vii
ÇİZELGELERİN LİSTESİ.....	ix
ŞEKİLLERİN LİSTESİ.....	x
SİMGELER VE KISALTMALAR.....	xi
1. GİRİŞ.....	1
2. İLGİLİ ÇALIŞMALAR.....	5
3. TEMEL BİLGİLER.....	15
3.1. Nesnelerin İnterneti (Internet of Things/ IoT)	15
3.2. Nesnelerin İnterneti ve Bilişim Teknolojileri: Bulut ve Kenar Bilişim.....	20
3.2.1.Bulut bilişim.....	20
3.2.2.Kenar bilişim.....	21
3.3. Blokzinciri	23
3.4. Kullanılan Kriptografik Yöntemler.....	27
3.4.1. Blowfish- CBC ve HMAC-SHA256 şifrelem ve bütünlük sağlama algoritması.....	27
3.4.2. AES-GCM şifreleme ve bütünlük sağlama algoritması.....	29
3.4.3. ChaCha20-Poly1305 şifreleme ve bütünlük sağlama algoritması	30
3.4.4. Diffie Hellman anahtar değişim algoritması	31
4. ÖNERİLEN MODEL.....	33
4.1. Aşama 1: IoT Cihazları ve Kenar Bilişim Yapısı Arasında Güvenli Veri Toplama ve Şifreli Aktarım Süreci	34

	Sayfa
4.1.1. Diffie-Hellman anahtar deęiřimi ile ChaCha20 anahtarının üretilmesi....	35
4.1.2. Salt üretimi	35
4.1.3. Anahtar türetme.....	36
4.1.4. Veri řifreleme.....	36
4.2. Ařama 2: Kenar Biliřim Yapısından Akıllı Sözleşmeye –Verilerin Deřifrenmesi, Sınıflandırılması ve Otomatik Karar Süreci	38
4.3. Ařama 3: Saęlık Verilerinin Güvenli Kaydı İçin Akıllı Sözleşme ve Blokzinciri Etkileřimi	40
4.4. Ařama 4: Blokzincir Üzerindeki Anormal Verilerin Zaman Damgalı Grafiklerle Görselleřtirilmesi ve Yetkili Arayüzü Üzerinden Gerçek Zamanlı Geri Bildirim Saęlanması	40
4.5. Geri Bildirim Mekanizması.....	43
5. ÖNERİLEN MODELİN ANALİZİ.....	47
5.1. Güvenlik Analizi	47
5.1.1. Önerilen modelde gizlilik analizi	47
5.1.2. Önerilen modelde kimlik doęrulama ve bütünlük analizi.....	55
5.1.3. Önerilen modelde erişilebilirlik analizi.....	58
5.2. Performans Analizi.....	59
5.2.1. Kullanılan teknolojiler, varsayımlar ve kısıtlar.....	59
5.2.2. Önerilen modelde kullanılan řifreleme ve bütünlük algoritmalarının performans analizi	60
5.2.3. Postman kullanılarak akıllı sözleşme fonksiyonlarının işlevsellik ve güvenilirlik testi.....	62
6. SONUÇ ve ÖNERİLER.	65
KAYNAKLAR	67
ÖZGEÇMİř	74

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 2.1. Sağlık verileri için IoT tabanlı blokzinciri modellerine yönelik güncel çalışmaların özeti	12
Çizelge 3.1. Bulut bilişim ve kenar bilişimin temel farkları	23
Çizelge 4.1. Kan şekeri düzeylerine göre kategoriler ve açıklamaları.....	39
Çizelge 4.2. Akıllı sözleşme içerisinde yer alan metotlar	40
Çizelge 5.1. Blowfish-CBC algoritmasının kullanıcı sayısına göre performans ölçümleri	61
Çizelge 5.2. AES-GCM algoritmasının kullanıcı sayısına göre performans ölçümleri	61
Çizelge 5.3. ChaCha20-Poly1305 algoritmasının kullanıcı sayısına göre performans ölçümleri	61

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 1.1. Dünya genelinde yıllara göre IoT bağlantılı cihazların sayısı	1
Şekil 3.1. IoT'nin karakteristik özellikleri	16
Şekil 3.2. IoT mimarisinin katmanları	17
Şekil 3.3. Glikoz seviyesini ölçmek amacıyla geliştirilen esnek ve şeffaf akıllı kontakt lens	18
Şekil 3.4. Blok yapısı	25
Şekil 4.1. IoT, kenar bilişim, blokzinciri ve akıllı sözleşmeler kullanarak kan şekeri seviyelerinin güvenli izlenmesi ve yönetimi için SağlıkZinciri modeli	34
Şekil 4.2. Kayıt olma arayüzü	41
Şekil 4.3. Giriş yapma arayüzü	41
Şekil 4.4. Kullanıcının başarılı giriş sonrası karşılaştığı ana arayüz	42
Şekil 4.5. Veri görüntüleme ve geri bildirim arayüzüne ait zaman damgalı grafiksel gösterim	43
Şekil 4.6. Durum ekleme arayüzü	44
Şekil 4.7. Zaman damgalarına göre organize edilmiş hasta durum görüntüleme arayüzü	45
Şekil 4.8. Acil durum bildirimi için hastaya iletilen e-posta örneği	45
Şekil 5.1. TLS 1.3. el sıkışma aşamaları	49
Şekil 5.2. Wireshark ile elde edilen şifresiz ağ trafiği çıktısı	50
Şekil 5.3. Wireshark ile elde edilen şifreli ağ trafiği çıktısı	51

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Kısaltmalar	Açıklamalar
AES	Advanced Encryption Standard
DPoS	Delegated Proof of Stake
GCM	Galois/Counter Mode
HKDF	HMAC Extract-and-Expand Key Derivation Function
HMAC	Hash-based Message Authentication Code
IaaS	Infrastructure as a Service
IPFS	InterPlanetary File System
IoT	Internet of Things
LISP	Locator ID Separation Protocol
MAC	Message Authentication Code
MITM	Man in The Middle
NIST	National Institute of Standards and Technology
NFT	Non-Fungible Token
P2P	Peer-to-Peer
PBFT	Practical Byzantine Fault Tolerance
PoA	Proof of Authority
PoC	Proof of Capacity
PoET	Proof of Elapsed Time
PoS	Proof of Stake
PoSpace	Proof of Space
PoT	Proof-of-Trust
PoW	Proof of Work

Kısaltmalar**Açıklamalar****PaaS**

Platform as a Service

RFID

Radio Frequency Identification

SaaS

Software as a Service

TPM

Trusted Platform Module

TPS

Transactions Per Second

WAN

Wide Area Network

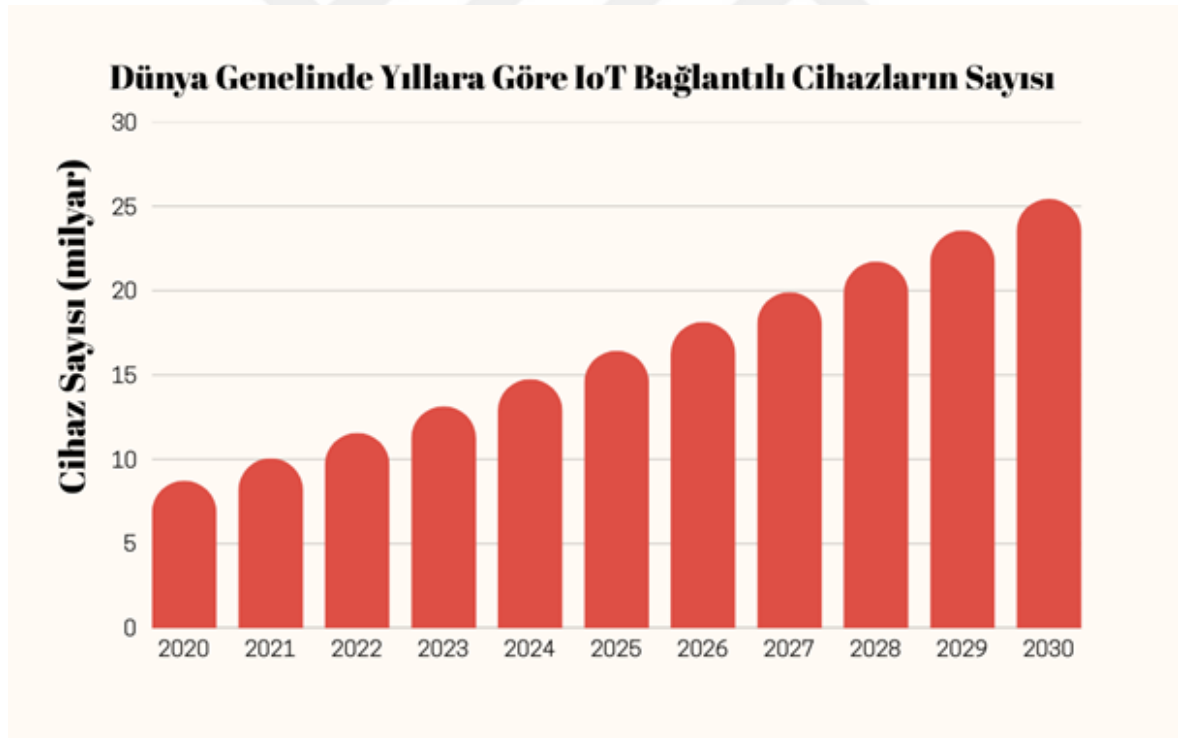
WSN

Wireless Sensor Network



1. GİRİŞ

Akıllı cihazların kullanımının artmasıyla birlikte internet aracılığıyla birbirine bağlı cihazların sayısında da önemli bir artış gözlemlenmektedir. Bu gelişme, literatüre 'Nesnelerin İnterneti' (Internet of Things/IoT) kavramını kazandırmıştır. IoT kavramıyla birlikte nesneler, herhangi bir zamanda, herhangi bir yerden, herhangi bir ağ üzerinden, herhangi bir kişiye bilgi veya hizmet sağlamak amacıyla internet ile iletişim kurabilmektedir [1]. IoT'nin gelişimi beklenenden daha hızlı gerçekleşmiş ve birçok alanda etkili olmaya başlamıştır. IoT bağlantılı cihaz sayısı son on yılda katlanarak artmış ve internete bağlanan cihaz sayısındaki bu artışın, önümüzdeki yıllarda da üstel bir şekilde devam etmesi beklenmektedir. Şekil 1.1'de, 2020 ile 2030 yılları arasındaki IoT bağlantılı cihazların sayısını göstermektedir. Ayrıca, bu dönemde IoT cihaz sayısının yaklaşık 2,5 kat artacağı tahmin edilmektedir [2].



Şekil 1.1. Dünya genelinde yıllara göre IoT bağlantılı cihazların sayısı [2]

IoT bağlantılı cihazların sayısındaki bu artış, IoT tabanlı teknolojilere olan taleple açıklanabilmektedir. Bu talepler birçok alanda olmakla birlikte, akıllı evler, kamu yönetimi, ticaret, ulaşım, altyapı, ekonomi, eğitim, akıllı şehirler ve sağlık hizmetleri gibi çeşitli sektörlerde kendini göstermektedir [3]. IoT'nin birçok alanda önemli fonksiyonları

bulunmakla birlikte, bu alanlardan en kritik olanlarından biri sağlık hizmetleridir. IoT teknolojilerinin sağlık hizmetlerinde kullanımı sayesinde hastaların uzaktan uzmanlar tarafından izlenebilmesi, tanı konulabilmesi ve tedavi edilebilmesi mümkün hale gelmiştir [4]. Bu gelişmeler, IoT tabanlı sağlık teknolojilerindeki yeniliklerin bireylerin sağlık durumlarının sürekli ve etkin bir biçimde izlenmesine olan gereksinimi artırdığını ortaya koymaktadır [5].

IoT ile sağlık hizmetlerine erişim daha etkili hâle gelirken, kişisel sağlık verilerinin toplanması ve analizine verilen önem de artmaktadır. Günümüzde akıllı saatler ve akıllı telefonlar gibi giyilebilir cihazlar sayesinde bireyler sağlık verilerini sürekli olarak izleyebilmektedir. Ayrıca, EKG gibi kalp atış hızını ölçebilen ve kan şekeri seviyelerini izleyebilen özel sensörler de mevcuttur [6]. Bu teknolojiler, bireylerin günlük yaşamlarında sağlık durumlarını kendi kendilerine takip etmelerine ve gerektiğinde sağlık hizmetlerine anında erişim sağlamalarına olanak tanımaktadır. Bununla birlikte, bu verilerin sürekli izlenmesi, sağlık durumlarının erken tespit edilmesini ve dolayısıyla bireyin sağlığını proaktif bir şekilde yönetmeye yönelik müdahalelerin yapılmasını mümkün kılmaktadır [7].

Son dönemde yapılan çalışmalar, sağlık verilerini dört kategoriye ayırmaktadır: Birinci kategori; elektronik sağlık kayıtları, reçeteler, laboratuvar verileri, patoloji görüntüleri, radyografik görüntüler ve sigorta talepleri gibi tıbbi sistemler tarafından üretilen verileri kapsamaktadır. İkinci kategori; giyilebilir fitness cihazları, insülin pompaları ve kalp pilleri gibi medikal cihazlardan, sağlık izleme uygulamalarından, hasta tarafından bildirilen sonuç anketlerinden ve sağlık hizmetlerine doğrudan bağlı olmayan DNA analizleri gibi doğrudan tüketiciye yönelik test ve tedavi türlerinden elde edilen bireysel sağlık ve iyilik hâli verilerini içermektedir. Üçüncü kategori, sosyal medya paylaşımları, arama geçmişleri ve konum verileri gibi tüketici faaliyetlerinden kaynaklanan ve dijital artık olarak adlandırılan verileri kapsamaktadır. Dördüncü kategori ise yaş, gelir, istihdam durumu ve eğitim düzeyi gibi sağlıkla doğrudan ilgili olmayan demografik ve sosyoekonomik verileri içermektedir. Bu veriler sağlık ve iyilik hâline ilişkin daha geniş sosyal belirleyicilerin tanımlanmasına yardımcı olmaktadır [8].

İlk iki kategori, bireylerin kolayca tanımlanmasını sağlayabilecek kişisel veriler içerdiğinden oldukça hassas veri sınıfına girmektedir. IoT kavramı ile bu tür hassas

verilerin ağlar üzerinden toplanması ve paylaşılması kolaylaşmıştır. Ancak, bu kişisel ve hassas bilgiler hem iletim hem de depolama aşamasında siber saldırılara karşı dayanıklı değildir ve bu bilgilerin bu saldırılara karşı güvence altına alınması gerekmektedir. Ayrıca, bu verilerin ikincil kullanımı ile ilgili olarak gizlilik ve veri bütünlüğü sorunları ortaya çıkmaktadır. Bu durum, IoT tabanlı sağlık sistemlerinin güvenliği, bireysel mahremiyetin korunması ve etik kullanımı gibi çok boyutlu problemleri beraberinde getirmektedir [9,10].

Uzaktan tedavi, teşhis ve izleme süreçlerinde bu veri kümelerinin bütünlüğü ve doğruluğu oldukça önemlidir. Bu nedenle veri güvenliği, etkili sağlık hizmeti yönetiminin hayati bir hedefi hâline gelmektedir [9,10]. Bu tür hassas verilerin korunmasında birçok farklı şifreleme algoritması ve güvenlik protokolü kullanılmaktadır. Ayrıca güncel araştırmalar, hassas verilerin güvenliğini sağlamak için birden fazla teknolojik çerçevenin kullanımını ortaya koymaktadır. Özellikle, IoT'nin blokzinciri ve bulut bilişim ile birleşimi, veri güvenliği için yenilikçi yaklaşımlar oluşturmaktadır [11,12].

Sağlık verilerinin artan karmaşıklığı ve hacmi, güvenli ve verimli yönetim, depolama ve paylaşım mekanizmalarına duyulan ihtiyacı arttırmaktadır. Bu gereksinimi karşılamak amacıyla, bu çalışmada hassas sağlık verilerinin gizliliğini, güvenliğini ve bütünlüğünü sağlamak üzere IoT, kenar bilişim ve blokzinciri teknolojilerini entegre eden yenilikçi bir model olan SağlıkZinciri önerilmektedir. Önerilen model ile giyilebilir cihazlar ve özel sensörlerle veriler neredeyse gerçek zamanlı olarak toplanabilmekte ve sağlık göstergeleri sürekli olarak izlenebilirken, veri toplama sürecinde gizlilik de sağlanmaktadır. Ayrıca, kenar bilişim verilerin yerel kaynakta işlenmesine olanak tanıyarak merkezi depolamaya göre gecikmeleri ve veri ihlali riskini en aza indirir. Blokzinciri teknolojisinin merkeziyetsiz ve değiştirilemez yapısı, verilerin güvenli şekilde tutulmasını, bozulmadan kalmasını ve her zaman kaynağına kadar izlenebilmesini sağlayarak kenar bilişim avantajlarıyla mükemmel bir uyum sergiler. Tüm bu teknolojilerin birleşimi, kişisel sağlık verileri için hem gizlilik hem de bütünlük endişelerini ele alan sağlam, güvenli ve verimli bir model sunar ve böylece sağlık hizmetlerinde güvenli bir ekosisteme katkı sağlar.

SağlıkZinciri modeli, sağlık verilerinin güvenliğini sağlamaya yönelik kapsamlı ve bütünlük bir yaklaşım sunarak gizlilik ve veri bütünlüğüne ilişkin kritik sorunları ele alması açısından mevcut literatüre katkı sunmaktadır. Bu modelin ilk aşamasında, veriler toplandıkları anda şifrelenmekte ve kenar bilişim yapısına güvenli bir şekilde aktarılmakta,

böylece ağ düzeyinde olası saldırılara karşı korunmakta ve veri bütünlüğü iletimin tüm yaşam döngüsü boyunca sağlanmaktadır. İkinci aşamada, gerçek zamanlı veri analizine dayalı karar alma süreçlerini otomatikleştiren akıllı sözleşmeler devreye girmekte ve sağlık hizmetlerini daha duyarlı hale getirmektedir. Üçüncü aşamada ise, anormal sağlık verileri değiştirilemez bir deftere kaydedilmekte ve bu sayede blokzinciri teknolojisiyle tam hesap verebilirlik ve şeffaflık sağlanmaktadır. Ayrıca model, verinin ağlar üzerinden iletimi süresince sağlık verilerinin bütünlüğünü de korumaktadır. Son aşamada ise sistem arayüzü, yetkilendirilmiş kişilerin geri bildirimde bulunmasına olanak tanıyarak hastalar ve sağlık hizmeti sağlayıcıları arasında iletişim akışını sağlamaktadır. Böylece önerilen model, yalnızca veri güvenliğini güçlendirmekle kalmamakta, aynı zamanda gerçek zamanlı sağlık müdahaleleri için ölçeklenebilir bir çözüm de sunmaktadır. Genel olarak, bu model sağlık durumlarına daha proaktif ve zamanında yanıt verilmesini mümkün kılarak sağlık ekosisteminin gelişmesine katkı sağlamaktadır.

Yukarıda verilen bilgiler doğrultusunda çalışma bölümlere ayrılmıştır. Bu çalışmanın bölümleri şu şekilde düzenlenmiştir: Bölüm II ilgili çalışmaları incelemektedir. Bölüm III arka plan bilgilerini sunmaktadır. Bölüm IV önerilen modele dair önemli ayrıntıları vermekte, Bölüm V ise önerilen modelin güvenlik ve performans analizini içermektedir. Son olarak, Bölüm VI sonuçların tartışılmasıyla çalışmayı özetlemektedir.

2. İLGİLİ ÇALIŞMALAR

Son yıllarda, sağlık verilerini yönetmek için yenilikçi teknolojilerin, özellikle sağlık bilgilerinin gizliliğini, güvenliğini ve bütünlüğünü korumak amacıyla blokzinciri teknolojisinin kullanımı üzerine büyük bir ilgi olmuştur. Birçok araştırma, sağlık verilerinin daha iyi izlenmesi, saklanması ve paylaşılması için blokzincirinin diğer teknolojilerle, özellikle bulut bilişim ve IoT ile birleştirilmesinin mümkün olup olmadığını incelemiştir. Bu öncü çalışmalar, sağlık sistemlerinde güvenlik ve verimlilik için çerçeveler oluşturmuştur. Bu bölüm, sağlık verisi yönetimi, veri güvenliği ve uzaktan hasta izleme konularına vurgu yaparak, blokzinciri çözümlerine yönelik önemli çalışmaları kronolojik olarak incelemektedir.

İlk çalışmalardan biri olan 'HealthSense' [13], blokzinciri tabanlı teknolojik uygulamaların uzaktan hasta izleme ve hassas sağlık bilgilerini güvence altına alma kullanımını araştırmıştır—bu veriler, bir hastanın yatağında yer alan sensörlerden alınan hastanın hayati verilerini içermektedir. Bu veriler, bir blokzinciri sisteminde akıllı sözleşme aracılığıyla saklanmıştır. Bu uygulama, REST tabanlı bir API (Uygulama Programlama Arayüzü) üzerinden blokzincirine veri girme, okuma, güncelleme ve yazma işlemleri yapmayı sağlamaktadır. Ayrıca, hastaların hastane faturalarını hesaplamak için akıllı sözleşme yapısı kullanılmaktadır. Nispeten basit olan bu yapıda, blokzinciri mimarisi belirtilmemiş ve bir bulut yapısı kullanılmamıştır.

'BlockCloud' [14] çalışması, heterojen sağlık verilerinin saklanması ve işlenmesine odaklanmaktadır. Çalışmaya göre, kalp ve kan şekeri seviyeleri gibi homojen sağlık verilerinin yanı sıra, EKG (Elektrokardiyogram) ve röntgen gibi heterojen sağlık verileri de mevcuttur. Bu tür çok çeşitli verilerin saklanması başlı başına oldukça zorluk oluşturmakta olup, sağlık verilerinin hassas doğası göz önüne alındığında daha fazla dikkat gerektirmektedir. Önerilen modelde, hem blokzinciri hem de bulut mimarisi entegre edilerek kullanılmıştır. Bulut yapısı, heterojen verileri işlemekte ve analiz etmekte, ardından işlenmiş veriler blokzincirinde saklanmaktadır. Bu çalışmadan da görüldüğü üzere bulut yapısı ile hassas verilerin güvenli bir şekilde saklanması mümkün olmamıştır. Bu nedenle, verilerin güvenliğini sağlamak için blokzinciri yapısı önerilen modele entegre edilmiştir.

2019 yılından sonra blokzinciri teknolojisinin sağlık verilerine entegrasyonu ile sağlık verilerinin güvenliğini, gizliliğini ve bütünlüğünü korumak için yapılan araştırmalar önemli ölçüde artmıştır. Chakraborty ve arkadaşları [15], giyilebilir cihazlardan toplanan sağlık verileri için blokzinciri ve bulut tabanlı bir model önererek, makine öğrenimi algoritmaları kullanmışlardır. Makine öğrenimi algoritmaları, verilerdeki anormallikleri tespit etmiş ve sağlık verilerinin bütünlüğünü, gizliliğini ve güvenliğini sağlamıştır. Diğer bir çalışmada, Dilawar ve arkadaşları [16], doktorların uzaktan hastaları takip ettiği ve blokzinciri üzerinde hastalara rapor ekleyebildiği bir model önermiştir. Bu raporlar, hastanın geçmiş verilerine dahil edilerek hastaya ait bütüncül bir sağlık geçmişi oluşturulmuştur. Çalışma ayrıca, uzaktan sağlık sistemleri aracılığıyla toplanan verilerin blokzincirinde saklanması yüksek maliyetli olabileceği konusunu vurgulamıştır. Çözüm olarak, veriler yerine yalnızca verilerin özet (hash) değerlerinin blokzincirinde saklanması önerilmiş ve bu şekilde blokzinciri maliyetlerinin düşürülmesi sağlanmıştır. Bhawiyuga ve arkadaşları [17] tarafından yapılan bir çalışmada ise özelleştirilmiş bir IoT sensörüyle toplanan sağlık verileri, bir API ağ geçidi aracılığıyla işlenmiş ve bir akıllı sözleşme tarafından blokzinciri işlem için tetiklenmiştir. Ağdaki düğümler, işlem önerisini simüle etmiş ve işlemin blokzincirine kaydedilmeden önce doğruluğunu garanti altına almıştır. Düğümler işlem önerisini onayladıktan sonra onaylarını API ağ geçidine geri göndermiş ve ağ geçidi bu onayları topladıktan sonra sıralayıcıya (orderer) iletmıştır. Sıralayıcı, onayları aldıktan sonra işlemi içeren bloğu tüm düğümlere yayınlamış ve düğümler işlemle ilgili son doğrulamayı yapmıştır. Bu süreç, sağlık verilerinin gizliliğini, güvenliğini ve bütünlüğünü garanti altına almıştır. Önerilen bu modelde, herhangi bir bulut yapısı kullanılmamış, blokzinciri yapısı olarak da Hyperledger Fabric blokzinciri çerçevesi kullanıldığı belirtilmiştir. Modelde kimlik doğrulama, JavaScript ile yazılmış bir akıllı sözleşme aracılığıyla gerçekleştirilmiştir. Çalışmanın API analizi, çalışmanın sonunda JMeter kullanılarak yapılmıştır. Dwivedi ve arkadaşları [18] tarafından önerilen diğer bir model, IoT, blokzinciri ve bulut mimarisinin sağlık verilerinin gizliliğini ve güvenliğini sağlamak için entegre edilerek kullanılmasını sağlamaktadır. Önerilen modelde geleneksel blokzincirlerinin madencilik mekanizmaları yoğun hesaplama gücü gerektirmesi nedeniyle, blokzinciri yapısı yerine blokzincirinin davranışını taklit eden bir üst ağ (overlay network) oluşturulmuştur. Veriler doğrudan bu üst ağa kaydedilmemiş, bunun yerine bulut depolama sunucularında saklanmış ve her bir bit bloğu, bir Merkle Ağacı kullanılarak özetlenmiştir. Bu veriler, blokların özet değerlerini doğrulayan ve zincirleyen üst ağdaki düğümlere gönderilmiştir. Hastalar, sağlık hizmeti sunucuları ve diğer yetkili

kurumlar, ağ üzerinde sağlık verilerini izleyebilme yetkisine sahiptir. Verilerde anomali tespiti akıllı sözleşmeler tarafından yapılmakta olup, verilerde anomali durumunun olması durumunda sağlık hizmeti sağlayıcıları otomatik olarak bilgilendirilmektedir. Bu modelde, hastanın sağlık verilerine erişim veri sahibinin kontrolündedir; hasta, tedavi sonrasında isterse bu verilerin paylaşımını sonlandırabilmektedir. Shahnaz ve arkadaşları [19] tarafından önerilen üç katmanlı modelde, elektronik sağlık verileri ilk olarak bir akıllı sözleşmenin tetiklendiği ilk katmanda işlenmektedir. Söz konusu çalışmada akıllı sözleşme, Solidity programlama dili ve Open Zeppelin aracı kullanılarak yazılmıştır; sistemde kimlik doğrulama, akıllı sözleşme aracılığıyla kontrol edilmektedir. Kimlik doğrulama işleminden sonra, ikinci katman olan kullanıcı katmanında gerçekleşen güncelleme, ekleme, görüntüleme ve silme işlemleri Ethereum blokzincirine kaydedilmektedir. Elektronik sağlık verileri, bir bireyin kimlik numarası, adı, kan grubu gibi temel parametrelerini ve hastanın tedavisi için tıbbi personel tarafından gereken profesyonel raporları içermektedir. Bu uzman raporları, Dağıtık İçerik Adresleme Sistemi (InterPlanetary File System/IPFS) olarak bilinen dağıtık dosya sistemi protokolü kullanılarak saklanmıştır. Çalışmanın sonunda, performans testi Apache JMeter aracı kullanılarak yapılmıştır. Sistem, 100-500 kullanıcıyla test edilmiş ve kullanıcı sayısı arttıkça sistemin verimliliğinin de arttığı gözlemlenmiştir. Attia ve arkadaşları [20] tarafından önerilen bir diğer çalışmada ise bu model için iki farklı blokzinciri yapısı kullanılmıştır. Söz konusu modelde, birinci blokzincirinin işlevi giyilebilir sensörler aracılığıyla toplanan sağlık verilerinin saklanması iken; ikinci blokzinciri, hastanın tüm sağlık geçmişinin tutulmasından sorumludur. Dolayısıyla, modele göre, hem hastalar hem de sağlık profesyonelleri ilk blokzincirindeki bilgileri görebilirken, ikinci zincirdeki bilgilere yalnızca sağlık profesyoneli erişebilmektedir. Her iki zincir de Hyperledger Fabric blokzinciri çerçevesine dayanmaktadır. Çalışmada acil durum bildirimlerini etkinleştirmek için, Go programlama dilinde yazılmış bir akıllı sözleşme yapısı kullanılmıştır. Modelde bulut yapısı yer almamaktadır.

Satamraju [21] tarafından yapılan daha sonraki bir çalışmada, sağlık verilerinin bütünlüğünün korunması ve yetkisiz erişimin önlenmesi çalışmanın temel amacı olarak belirtilmiştir. Bu çalışmada üç katmanlı bir model önerilmiştir. İlk katman olan uygulama katmanı, IoT cihazları ile blokzinciri arasında bir arayüz işlevi görmektedir. İkinci katman olan iş katmanı; veri ve hastaya ait reçetelerin görüntülenmesi veya yeni verilerin eklenmesi gibi tüm işlevsellikleri içermektedir. Son katman olan depolama katmanında ise

tüm veriler saklanmaktadır. Blokzinciri üzerinde veri ve işlemlerin saklanması yüksek maliyetli olması nedeniyle, yalnızca işlem verileri blokzincirine aktarılmış; ilgili şifrelenmiş veriler ise şifreli bir veritabanında saklanmıştır. Modelde blokzinciri yapısı olarak Ethereum kullanılmıştır. Veriler bir kullanıcı arayüzü üzerinden görselleştirilmiş; doktorlar, eczacılar ve sigorta şirketleri gibi yetkilendirilmiş kişiler bu arayüze erişebilmiştir. Benzer bir çalışmada, Khatoon [22] tarafından önerilen model, sağlık verilerine çok paydaşlı erişim imkânı sağlamıştır. Veritabanı, sağlık verilerinin şifrelenmiş ve özetlenmiş gerçek değerlerini içermekte, ayrıca bu özet değerlerin kayıtları blokzincirinde tutulmaktadır. Bu sayede, veri bütünlüğü, özet değerlerin birbirleriyle karşılaştırılması yoluyla doğrulanabilmektedir. Kullanılan blokzinciri yapısı Ethereum olup, veri erişim yetkilendirmesi Solidity dili ile yazılmış bir akıllı sözleşme aracılığıyla gerçekleştirilmiştir. Ayrıca veriler üzerinde gerçekleştirilen tüm işlemler blokzincirine kaydedilmiştir. Ismail ve arkadaşları [23] tarafından önerilen ‘BlockHR’ adlı model, mevcut istemci-sunucu mimarilerine kıyasla daha hızlı ve güvenli bir hasta izleme olanağı sağlamaktadır. Önerilen model, mevcut istemci-sunucu sistemlere göre 20 kat daha hızlı performans sergilemektedir. Mevcut çalışmaların çoğunda olduğu gibi, hastalar ağ üzerinden kolaylıkla blokzincirine veri ekleyebilmiş veya verileri blokzincirinden sorgulayabilmektedir. BlockHR modelinin avantajı, hasta verilerinden öğrenerek bu bilgiler üzerinden teşhis koyan tahmin aracına dayanmasıdır. Veriler yalnızca yetkili kişiler tarafından kontrol edilebilmiş ve değiştirilebilmiş; her işlem kayıt altına alınarak şeffaflık ve hesap verebilirlik sağlanmıştır. Jamil ve arkadaşları [24] tarafından yedi farklı sensörden toplanan sağlık verilerinin görselleştirilmesi ve işlenmesine yönelik bir model önerilmiştir. Önerilen modelde toplanan verilere yalnızca kimlik doğrulaması yapılmış kişiler erişebilmiştir. Modelin mimarisinin temelini Hyperledger Fabric blokzinciri yapısı oluşturmuştur. Mimari dört katmandan oluşmaktadır. İlk katman olan sensör katmanı, verilerin toplanmasından sorumludur. İkinci katman olan ağ katmanı, toplanan verilerin blokzincirine iletilmesini sağlamaktadır. Üçüncü katman olan blokzinciri katmanında, verinin eşik seviyesinin altında olup olmadığı akıllı sözleşme aracılığıyla kontrol edilmiş ve kimlik doğrulama da yine aynı akıllı sözleşme ile gerçekleştirilmiştir. Son olarak, uygulama katmanı kullanıcıların sistemle etkileşime geçmesini ve veri girişi yapmasını sağlamaktadır. Bu blokzinciri modelinin performansı, Caliper aracı ile 300, 500 ve 1000 kullanıcı ile test edilmiş ve kullanıcı sayısı arttıkça performansta iyileşme gözlemlenmiştir. Banotra ve arkadaşları [25], klasik istemci-sunucu mimarisine dayalı uzaktan sağlık hizmeti modellerinin merkezi bir yapıda olmaları nedeniyle güvenli olmadığını

belirtmiştir. Bu nedenle, uzaktan sağlık hizmetleri için güvenli bir platform sağlamak amacıyla dört katmanlı bir model önermişlerdir. İlk katman, verilerin toplandığı ve hastaların bulunduğu fiziksel katmandır. İkinci katman, bu verileri toplayan sensör cihazlarına aittir. Üçüncü katman, verilerin buluta gönderilmesi için bir API kullanan iletişim katmanıdır; en üst katman ise iletilen verilerin normalleştirilip analiz edildiği katmandır. Son katman, verilerin saklanması sağlayan dağıtık defter sistemi olarak adlandırılmış olup, deftere erişim web uygulaması veya mobil uygulama aracılığıyla mümkün kılınmıştır. Ngabo ve arkadaşları [26] tarafından önerilen bir diğer model, hem bulut altyapısını hem de blokzinciri teknolojisini entegre etmiştir. Bu modelde, sensörler aracılığıyla toplanan sağlık verileri eliptik eğri algoritmaları kullanılarak şifrelenmiş ve kenar bilişim ortamına iletilmiştir. Kenar bilişim kullanımı, veri iletimi sırasında gecikmenin azaltılmasına yardımcı olmuştur. Veriler bulut bilişim ortamında saklanırken, veriler üzerinde gerçekleştirilen tüm işlemler blokzincirine kaydedilmiş ve bu işlemlerin yürütülmesi için kimlik doğrulama mekanizması zorunlu kılınmıştır. Model ayrıca, blokzinciri üzerinde gerçekleştirilen işlem kayıtlarının kopyalarının saklandığı bir bulut bilişim yapısı da içermektedir. Literatürde önerilen bir sonraki model ise FogChain modelidir [27]. Bu modelin temel amacı sağlık verilerinin güvenliğinin sağlanmasıdır. Önerilen model, çalışmamızın yapısıyla benzerlik göstermekte olup, blokzinciri yapısı olarak Hyperledger Fabric kullanılmıştır. Ayrıca kenar bilişim mimarisi de modele dahil edilmiştir. Bu modelde, hastalardan toplanan sağlık verileri sis bilişim katmanına iletilmiştir. Veri değerleri, belirlenen eşik değerlerin dışına çıktığında, sis bilişim katmanı verileri işleyerek blokzinciri üzerinde bir akıllı sözleşmeyi tetiklemiş ve söz konusu akıllı sözleşme, eşik değer aralığında olmayan verilerin kaydedilmesini sağlamıştır. Sis bilişimin kullanımı, iletim gecikmesini en aza indirmiş ve böylece modelin verimliliğini arttırmıştır.

Bataineh ve arkadaşları [28] tarafından önerilen ERTCA modeli, sağlık verilerinin güvenliğini, zengin ve ince istemcilerin yanı sıra blokzinciri teknolojisinin entegrasyonu ile ele almıştır. Bu modelde Ethereum, blokzinciri yapısı olarak kullanılmış ve İş Kanıtı (Proof of Work/PoW) konsensüs mekanizması tercih edilmiştir. Model, sağlık verilerini toplayan ve API aracılığıyla blokzincirine gönderen bir IoT cihazı olarak işlev gören ince istemciyi temsil etmektedir. Daha büyük bir belleğe sahip zengin istemci ise blokzincirine erişim sağlamaktadır ve madencilik faaliyetlerine katılmaktadır. Bu modelde herhangi bir bulut yapısı kullanılmamıştır.

Elvas ve arkadaşları [29] tarafından yapılan çalışmaya göre, model sadece hastaları ve uzmanları değil, aynı zamanda eczaneleri de çalışmaya dahil etmektedir. Çalışmanın temel amacı, veri sahibi olan hastaya, verilerine kimlerin erişebileceğini belirleme yetkisi vermek ve böylece kimlik doğrulama sürecini daha güvenli hale getirmektir. Ayrıca, verinin güvenliğini sağlamak da amaçlanmıştır. Modelin işleyişine göre, hasta sisteme kaydolduktan ve hastanede muayene olduktan sonra, uzman, blokzinciri üzerinde bir akıllı sözleşme oluşturmakta, hastanın verilerini ve gerekli ilaç listesini şifrelemekte ve ardından bunları blokzinciri üzerinde saklamaktadır. Muayene sonucunda, hasta eczaneye gitmeye gerek duymamaktadır çünkü gerekli ilaçlar blokzinciri aracılığıyla reçetelenmektedir. Sistemde kimlik doğrulaması yapıldıktan sonra, eczane hastanın ilaçlarını hazır hale getirebilmektedir. Bu yaklaşım, tedavi sürecini hızlandırmanın yanı sıra verinin korunması için daha güvenli bir yöntem sunmuştur. Model herhangi bir bulut yapısı kullanmamıştır. Cheikhrouhou ve arkadaşları [4] tarafından önerilen yeni bir çalışma, Jamil ve arkadaşları [24] tarafından yapılan önceki araştırmalarla paralel olarak, verinin ağ katmanında meydana gelebilecek saldırılara karşı hassasiyetini vurgulamıştır. Bu nedenle, Cheikhrouhou ve arkadaşları [4], veri güvenliğini artırmak ve sistem verimliliğini iyileştirmek için blokzinciri ve kenar bilişim destekli güvenli uzak hasta izleme sistemi önermiştir. Yeni yaklaşımda, sensörler ile toplanan veriler ilk olarak yerel bir blokzincirinde saklanmakta ve ardından eşik değerlerine karşı testler yapılmak üzere bulut ortamına aktarılmaktadır. Kabul koşulu sağlandığında, verilerin global blokzincirine aktarılması başlatılmaktadır. Sadece yetkilendirilmiş kişiler her iki blokzincirine erişebilirken, bulut yapısı veriler ve işlemleri hakkında bilgileri güvenli bir şekilde tutmaktadır. Model, iki bulut yapısı kullanmış ve bu yapılarla yanıt sürelerinde %40'lık bir iyileşme sağlanmıştır. Hyperledger Fabric, blokzinciri çerçevesi olarak seçilmiştir. Çalışma ayrıca, önerilen modeli anahtar saldırıları, tekrar saldırıları ve taklit saldırıları gibi çeşitli saldırı senaryolarına tabi tutmuş ve modelin bu tehditleri başarılı bir şekilde engellediğini belirtmiştir.

Oladele ve arkadaşları [30] tarafından geliştirilen BEHeDaS, sağlık verilerinin güvenli ve şeffaf bir şekilde saklanmasını sağlamak amacıyla tasarlanmış özel bir blokzinciri sistemidir. Sağlık personelleri ve hastalar, kendilerine ait kullanıcı adı ve şifrelerle kimlik doğrulaması yapmakta, yetkilendirme işlemleri ise blokzinciri üzerinden yönetilmektedir. Sağlık çalışanları, blokzinciri üzerinden hastaların tıbbi geçmişine erişebilmekte, tanı koyabilmekte, testler yapabilmekte ve tıbbi sonuçları yükleyebilmektedir. Bu veriler

şifrelenmekte ve blokzinciri ağı içerisinde mutabakata tabi tutulmaktadır. Hastalar da tıbbi sonuçlarını görüntüleyebilmekte ve kişisel verilerinde değişiklik talebinde bulunabilmektedir; ancak bu talepler ancak ağ katılımcılarının onayıyla işleme alınmaktadır. Sistem, güvenliği artırmak amacıyla ceza temelli bir uygulama modeli içeren Güven Kanıtı (Proof-of-Trust/PoT) konsensüs mekanizması ile çalışmakta; kimlik doğrulama ve yetkilendirme işlemleri ise akıllı sözleşmeler aracılığıyla yürütülmektedir. Çalışmada kullanılan blokzinciri yapısının Eşler Arası (Peer-to-Peer/P2P) tabanlı özel bir blokzinciri olması nedeniyle sistem, güvenlik ve veri bütünlüğüne odaklanmakta, ancak işlem hızı (Transactions Per Second/TPS) halka açık blokzincirlerine kıyasla daha düşüktür. Said ve arkadaşları [31] tarafından önerilen bir diğer modelde ise, Hyperledger Fabric ile değiştirilemez varlıklar (Non-Fungible Token/NFT) entegre edilmiştir. Süreç, sağlık kuruluşlarının (klinikler, hastaneler, laboratuvarlar ve eczaneler) blokzinciri ağına kaydedilmesiyle başlamaktadır. Hastalara ilişkin işlemler, laboratuvar sonuçları, reçeteler ve danışmanlık raporları da dahil olmak üzere hastaya ait tüm belgeler, blokzincirinde güvenli bir şekilde saklanmakta ve ilgili hastayla doğrudan ilişkilendirilmektedir. Hastanın kimliği, kimlik numarasının blokzinciri kayıtları ile karşılaştırılması yoluyla doğrulandıktan sonra, dijital olarak onaylanmış bir NFT üretilmektedir. Bu NFT, hastanın doğrulanmış tıbbi kayıtlarının kriptografik bir temsili olup, ağ içindeki sağlık kuruluşlarının kimlik doğrulaması yapmasına ve tıbbi verilere erişmesine imkân tanımaktadır. Ancak blokzincirine erişim süresi sınırlıdır ve oturum önceden tanımlanmış bir sürenin ardından otomatik olarak sona ermektedir. Erişim kontrol politikaları, Go programlama diliyle geliştirilen bir akıllı sözleşme aracılığıyla uygulanmakta olup, bu durum hastaların sağlık verileri üzerindeki erişim haklarını doğrudan belirlemelerine imkân tanımaktadır.

Yukarıda bahsi geçen tüm çalışmalar Çizelge 2.1’de özetlenmektedir. İlgili sütundaki ✓ sembolü, çalışmanın o teknoloji ile entegre olduğunu belirtmektedir; buna karşılık, sütunda yer alan - sembolü ise böyle bir entegrasyonun bulunmadığını ifade etmektedir. B (Belirtilmemiş) ifadesi de, çalışmanın söz konusu teknoloji entegrasyonuna ilişkin herhangi bir bilgi sunmadığı anlamına gelmektedir.

Çizelge 2.1. Sağlık verileri için IoT tabanlı blokzinciri modellerine yönelik güncel çalışmaların özeti

	Blokzinciri Türü	Bulut Entegrasyonu	Akıllı Sözleşme Kullanımı	Akıllı Sözleşme Dili	Amacı
[13]	B	-	✓	B	Güvenlik
[14]	B	✓	-	-	Farklı veri türlerinin güvenliği ve depolanması
[15]	B	✓	-	-	Bütünlük, gizlilik, güvenlik
[16]	B	-	-	-	Güvenlik
[17]	Hyperledger Fabric	-	✓	Node.js	Gizlilik, güvenlik ve bütünlük
[18]	B	✓	✓	B	Gizlilik ve güvenlik
[19]	Ethereum	-	✓	Solidity	Güvenlik, bütünlük
[20]	Hyperledger Fabric	-	✓	Go	Güvenli erişim
[21]	Ethereum	-	✓	Solidity	Güvenlik, bütünlük
[22]	Ethereum	-	✓	Solidity	Gizlilik, güvenlik, erişilebilirlik
[23]	İzinli blokzinciri	-	✓	B	Gizlilik, bütünlük, erişilebilirlik
[24]	Hyperledger Fabric	-	✓	Node.js	Gizlilik, güvenlik, ölçeklenebilirlik
[25]	B	✓	✓	B	Gizlilik, güvenlik
[26]	B	✓	✓	B	Gizlilik, güvenlik, mahremiyet

Çizelge 2.1. (devam) Sağlık verileri için IoT tabanlı blokzinciri modellerine yönelik güncel çalışmaların özeti

[27]	Hyperledger Fabric	✓	✓	Node.js	Bütünlük, güvenlik, gizlilik
[28]	Ethereum	-	✓	Solidity	Güvenlik
[29]	Ethereum	-	✓	B	Farklı veri türlerinin güvenliği ve depolanması
[4]	Hyperledger Fabric	✓	✓	B	Gizlilik, güvenlik, mahremiyet
[30]	Hyperledger Fabric	-	✓	Go	Güvenlik, birlikte çalışabilirlik, gizlilik
[31]	Hyperledger Fabric	-	✓	Go	Gizlilik, güvenlik, ölçeklenebilirlik, birlikte çalışabilirlik, izlenebilirlik
Bu Çalışma	Hyperledger Fabric	✓	✓	Go	Bütünlük, gizlilik, güvenlik



3. TEMEL BİLGİLER

Bu başlık altında, önerilen modelin ayrıntılı bir şekilde ele alınmasından önce, modelin kavramsal çerçevesinin daha iyi anlaşılabilmesi amacıyla genel bir bilgi sunumu yapılmaktadır. İlk olarak, önerilen modelde kullanılan teknolojiler alt başlıklar halinde incelenmekte, ardından bu teknolojiler arasındaki veri akışında kullanılan şifreleme algoritmaları ayrı bir alt başlık altında ele alınmaktadır.

3.1. Nesnelerin İnterneti (Internet of Things/ IoT)

IoT kavramı, her şeyin internet ile bağlantılı hale getirilmesi anlamına gelmektedir. Yani çamaşır makineleri, akıllı telefonlar, kahve makineleri ve giyilebilir cihazlar gibi araçların internet aracılığıyla birbirine bağlanması olarak tanımlanabilir [32]. IoT; insanlar ile insanlar, insanlar ile nesneler ya da nesneler ile nesneler arasında bağlantı kurulmasını sağlayarak bir ağı oluşmasını sağlamaktadır. Bu iletişim ağı sayesinde cihazlar arasında sürekli ve dinamik bir şekilde veri aktarımı gerçekleştirilebilmektedir [1].

Şekil 3.1'de görüldüğü gibi, IoT farklı bağlamlar, cihazlar ve hizmetlerle etkileşim içinde çalışabilen esnek bir sistemdir. IoT'nin sahip olduğu özelliklere bakıldığında, IoT herhangi bir zamanda, herhangi bir cihazla ve her yerde çalışabilmektedir. IoT, her yerden ve her koşulda veri toplama ve paylaşma yeteneğine sahiptir. Ayrıca, IoT herhangi bir ağı kullanarak veri iletimi yapabilmektedir [1,3].



Şekil 3.1. IoT'nin karakteristik özellikleri [1]

Yukarıdaki temel özelliklere sahip olan IoT; verileri toplarken, cihazlar arasında toplanan verileri aktarırken katmanlı bir yapı kullanmaktadır. IoT mimarisi, farklı katmanlardan oluşur ve her bir katman, verilerin toplanması, iletilmesi, işlenmesi ve analiz edilmesi süreçlerinde belirli bir rol üstlenir. IoT için standartlaşmış bir mimari yoktur. Ancak, birçok çalışmaya göre IoT dört katmandan oluşur: Algılama Katmanı, Ağ Katmanı, Destek Katmanı ve Uygulama Katmanı. Algılama katmanı, fiziksel dünya ile bağlantı kurmak ve çevreden veri toplamakla sorumludur. Radyo Frekansı ile Tanımlama (Radio Frequency Identification /RFID) etiketleri, kameralar, sensörler ve kablosuz sensör ağları (Wireless Sensor Network/WSN) gibi çeşitli teknolojileri kullanarak çevreden veri toplanan katmandır. Toplanan veriler arasında sıcaklık, pH seviyesi, ışık yoğunluğu, konum ve hareket gibi veriler yer alır. Diğer bir katman olan ağ katmanı, algılama katmanında toplanan verileri üst katman olan ara katmanına iletmekten sorumludur. Verilerin kablolu veya kablosuz iletim ortamları üzerinden iletilmesini sağlar. Ağ katmanı, veri iletimini sağlamak için çeşitli protokoller ve teknolojiler kullanır. Bu protokoller ve teknolojilere örnek olarak 4G/5G, Wi-Fi, ZigBee, Bluetooth, 6LoWPAN ve WiMAX verilebilir. Ağ katmanı, verilerin algılama katmanından ara katmanına etkin ve güvenli bir şekilde iletilmesini sağlamak için kritik bir rol oynar. Ara katman, ağ katmanından veri alır, sistemi bulut ve veritabanına bağlar ve veri işleme ve depolama işlemleri gerçekleştirir. Aynı zamanda, uygulama katmanının gereksinimlerini karşılamak için API'ler sunar. Ara

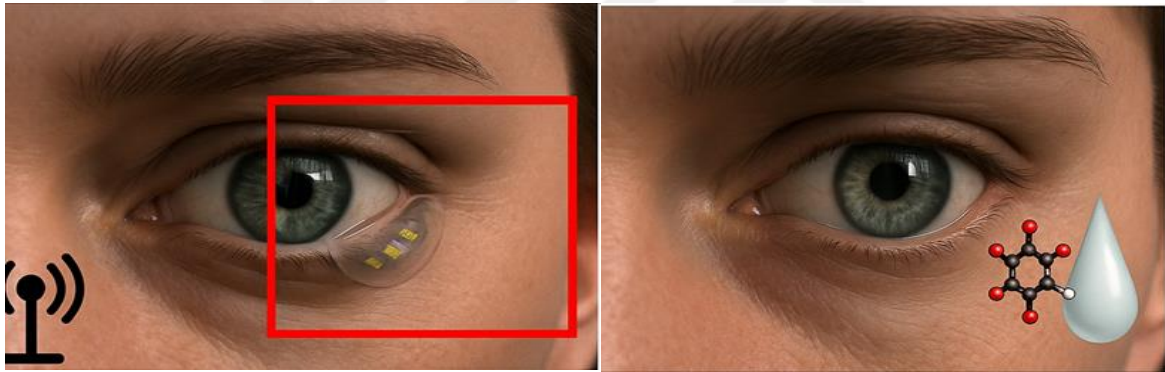
katman içinde, veritabanı güvenliği ve bulut güvenliği gibi temel konular yer alır ve bunlar uygulama katmanındaki hizmet kalitesini etkiler. Son katman olan uygulama katmanı ise ara katmandan alınan verileri işleyip manipüle etmek ve son kullanıcılara yüksek kaliteli hizmet sunmak amacıyla çalışır. Yukarıda açıklanan IoT mimarisinin katmanları, her bir katmanın fonksiyonlarını ve bileşenlerini daha iyi anlamak için Şekil 3.2'de görsel olarak özetlenmektedir [33-36].



Şekil 3.2. IoT mimarisinin katmanları [37]

Uygulama katmanında çeşitli hizmet ve uygulamalar sunan IoT; bu hizmet ve uygulamaları çok çeşitli alanlarda sunmaktadır. Bu kullanım alanlarına; askeri alanlar, akıllı evler, akıllı şehirler, katı atık yönetimi, akıllı sayaçlama, çevre gözetimi, tarım ve sağlık hizmetleri gibi birçok örnek verilebilmektedir. Bu alanlara örnek olarak IoT'nin bazı sektörlerdeki kullanımlarını şu şekilde özetlemek mümkündür: Askerî alanlarda askeri personel, IoT sensörleri aracılığıyla uzaktan izlenebilmektedir. Personele ait kasklara yerleştirilen sensörler sayesinde, askeri personellerin konumları ve hayati değerleri uzaktan takip edilebilmektedir. Bu uzaktan takip genellikle gerçek zamanlı olarak gerçekleşmektedir. Önemli bir diğer alan olan akıllı evlerde de bir evin içinde internete bağlı çeşitli akıllı cihazlar, ev sahibi tarafından uzaktan kontrol edilebilmektedir. İnternete

bağlanan bu cihazlar arasında klima, güvenlik sistemleri, fırın, buzdolabı ve aydınlatma sistemleri yer alabilir. Farklı sistemlerde ev sahipleri, söz konusu cihazların çalışma durumlarını uzaktan izleyebilmekte ve ilgili arayüzler aracılığıyla cihazları uzaktan kontrol edebilmektedir. Diğer bir kullanım alanı olan sağlık sektöründe, özellikle yaşlı, hasta veya engelli bireylerin hayati değerlerinin uzaktan izlenmesi ön plandadır. Kişinin vücuduna takılan IoT sensörleri aracılığıyla toplanan çeşitli veriler sayesinde, bireyler uzmanlar tarafından uzaktan takip edilebilmektedir. Bu bireyler; diyabet, kalp hastalığı gibi hayati belirtilerinin gözlemlenmesi gereken kişiler olabilir. Hastaların uzaktan takibinde kullanılan IoT sensörlerinden örnek vermek gerekirse; nabız, kan şekeri, vücut sıcaklığı ve kan basıncı gibi hayati parametreleri ölçen sensörler yaygın olarak kullanılmakta ve bu sayede bireylerin sağlık durumları gerçek zamanlı olarak izlenebilmektedir [38]. Bu sensörlerden biri, Şekil 3.3'te gösterildiği üzere, gözyaşındaki glikoz seviyesini ölçmek amacıyla geliştirilen esnek ve şeffaf akıllı kontakt lensdir [39].



Şekil 3.3. Glikoz seviyesini ölçmek amacıyla geliştirilen esnek ve şeffaf akıllı kontakt lens [39]

Yukarıda bahsedilen tüm alanlarda IoT'nin kullanımı büyük önem arz etmekle birlikte, bu teknolojinin sağlık alanındaki uygulamaları hem bireysel sağlık takibi hem de sağlık hizmetlerinin etkinliği açısından ayrı bir öneme sahiptir. Sağlık alanındaki nesnelerin interneti; gerçek zamanlı veri iletimi, hastanın fizyolojik durumunun farklı zaman aralıklarında izlenebilmesi gibi birçok avantaj sunmaktadır. Tansiyon ölçerler, glikoz ölçüm cihazları, kalp izleme implantları, Elektroensefalografi (EEG), Elektrokardiyogram ve Elektromiyografi (EMG) gibi giyilebilir cihazlar sayesinde sağlık hizmeti sağlayıcıları, hastaya ait sağlık verilerini yerel olarak toplayabilir ve hasta sağlık verilerine dayanarak gerçek zamanlı kararlar alabilmektedirler. Ancak, IoT'nin sağlık alanındaki kullanımının artmasıyla birlikte, verilerin güvenliği ve bütünlüğü büyük bir endişe kaynağı haline

gelmektedir. IoT cihazları, genellikle internet üzerinden veri ilettiği için, bu verilerin yetkisiz erişimlere ve siber saldırılara karşı savunmasız olma riski taşımaktadır. Ayrıca, cihazların ve ağların güvenliği sağlanmadığı takdirde, hasta verilerinin gizliliği ve doğruluğu tehlikeye girebilir. Bu sebeple, IoT tabanlı sağlık uygulamalarında güçlü şifreleme yöntemleri, kimlik doğrulama mekanizmaları ve güvenli iletişim protokollerinin kullanılması kritik bir öneme sahiptir [40]. Ancak, yukarıda bahsedilen IoT'nin sunduğu avantajlar kadar, bu teknolojinin kısıtlılıkları da söz konusudur. Bu kısıtlılıklar aşağıdaki başlıklar altında incelenebilir:

- **Heterojenlik:** IoT sistemlerinin farklı yapıları, iletişim protokolleri ve yapılandırılmış, yarı yapılandırılmış ve yapılandırılmamış çeşitlilikteki veri türlerini kullanması sistemlerin heterojen olmasına sebep olmaktadır. Sistemlerin heterojen olması gizlilik ve güvenlik gibi sorunların temel kaynağıdır.
- **Ağların karmaşıklığı:** IoT sistemlerinin çok sayıda iletişim ve ağ protokolünü bir arada kullanması ağların karmaşık hale gelmesine yol açmaktadır. Bu iletişim ve ağ protokollerine örnek olarak Bluetooth, 6LoWPAN, LoRa gibi teknolojiler verilebilir.
- **Zayıf birlikte çalışabilirlik (Interoperability):** IoT sistemleri hem yazılımları hem de donanımları sebebiyle birbiriyle veri alışverişi yapmaları, birlikte çalışmaları ve iş birliği kurmaları oldukça zordur.
- **Kaynak kısıtlamaları:** IoT cihazları sınırlı işlem gücüne, depolama alanına ve enerji kapasitesine sahiptir. Bu kaynak kısıtlamaları cihazları kötü amaçlı saldırılara karşı savunmasız hale getirir.
- **Gizlilik açıkları:** IoT sistemlerinin merkezi olmaması, karmaşıklığı ve heterojenliği sebebiyle sistemlerdeki verilerin gizliliğini korumak zorlaşmaktadır.
- **Güvenlik açıkları:** IoT sistemlerinin heterojen ve merkezi olmayan yapısı nedeniyle kimlik doğrulama, iletişim şifreleme gibi geleneksel şifreleme algoritmalarını IoT sistemlerine uygulamak oldukça zordur. Bu durum IoT sistemlerini kötü amaçlı saldırılara açık hale getirmektedir. [41]

IoT cihazlarının yukarıda bahsedilen kısıtları sebebiyle, güçlü şifreleme yöntemleri, kimlik doğrulama mekanizmaları ve güvenli iletişim protokollerinin IoT cihazlarında kullanılması genellikle zordur. Bu da, sağlık verilerinin güvenliği ve bütünlüğü açısından ek riskler yaratmaktadır [42]. Bu kısıtların üstesinden gelebilmek için IoT sistemleri çalışmalarda birçok yapıyla birlikte kullanılmıştır. Bu yapılara örnek olarak bulut ve kenar bilişim sistemleri gösterilebilir.

3.2. Nesnelerin İnterneti ve Bilişim Teknolojileri: Bulut ve Kenar Bilişim

Bu başlık altında öncelikle bulut bilişim ve kenar bilişimin temel özellikleri ele alınmaktadır. Ardından bu iki bilişim teknolojisinin IoT sistemleriyle kullanımına dair avantaj ve dezavantajlar tartışılmaktadır.

3.2.1. Bulut bilişim

Müşterinin değişen ihtiyaçlarına göre hizmet sağlayabilen sanal makineler grubundan oluşan dağıtık yapıya bulut bilişim denmektedir [43]. Ulusal Standartlar ve Teknoloji Enstitüsü (NIST), Bulut'u; ağ, depolama, donanım, uygulamalar vb. yapılandırılabilir bilgi işlem kaynaklarının paylaşılan bir havuzuna kolay erişim sağlayan, bu kaynakların hızlı bir şekilde tahsis edilebildiği, ölçeklenebildiği ve en az yönetim çabası veya hizmet sağlayıcısı müdahalesiyle serbest bırakılabildiği bir model olarak tanımlamaktadır. Ayrıca kullanıcı, bulut bilişim yapısını kullanırken herhangi bir donanım satın almaz, güncelleme yapmaz veya donanımına herhangi bir şey kurmaz; bunların tamamı bulut bilişim tarafından gerçekleştirilir. Bulut bilişim, kullanıcılara çeşitli hizmetler sunmaktadır. Bu hizmetler, NIST tarafından üç kategoriye ayrılmıştır. Bu hizmetlerin ilki, Hizmet Olarak Yazılım (Software as a Service/SaaS) modelidir. Bu hizmet modelinde, tüketiciler istedikleri uygulamayı bulut altyapısı aracılığıyla internet üzerinden kullanabilirler. SaaS bağlamında, kullanıcılar kurulum maliyetleri, altyapı maliyetleri veya bakım maliyetleri için ödeme yapmazlar. Güncelleme işlemleri dahi bulut tarafından gerçekleştirilir. İkinci hizmet modeli, Hizmet Olarak Platform (Platform as a Service/PaaS) olarak adlandırılmaktadır ve uygulamaların geliştirilmesi, yürütülmesi ve yönetilmesini kolaylaştıran entegre bir bulut tabanlı bilgi işlem ortamını ifade eder. Bu model, bir bilgi işlem platformunun internet üzerinden sunulması anlamına gelmektedir. Son hizmet modeli ise Hizmet Olarak Altyapı (Infrastructure as a Service/IaaS) olup, altyapı, donanım, ağ ve depolama hizmetleri gibi

bilgi işlem kaynaklarının sanal olarak sağlanmasını ifade eder. Bu üç hizmet arasında, IaaS kullanıcıya bulut altyapısı üzerinde en fazla kontrol imkânı sunarken; en az kontrol Hizmet Olarak Yazılım (SaaS) modelinde sağlanmaktadır. Kullanıcılar, bu hizmetler aracılığıyla, kullandıkça öde modeliyle paylaşılan kaynak havuzundan taleplerini karşılayabilirler [44]. NIST'e göre, bahsedilen üç hizmet modelinin dağıtımı dört farklı şekilde gerçekleşmektedir. Bunlardan ilki, genel bulut (public cloud) modelidir. Genel bulutta, hizmetler genellikle genel kullanıcılara veya büyük işletmelere sunulmaktadır. Genel bulut yapısı; kullanıcılarına ölçeklenebilirlik ve güvenilirlik gibi avantajlar sağlasa da, güvenlik açısından bazı sorunlara sebebiyet verebilmektedir. Çünkü bu tür bir bulutta, kullanıcılar verilerinin nerede ve nasıl saklandığını bilmemektedir; bu durum çok kiracılık (multi-tenancy) sorununu ortaya çıkarmaktadır. Bu soruna çözüm olarak, ikinci dağıtım modeli olan özel bulut (private cloud) yapısı kullanılabilir. Bu dağıtım modelinde, hizmetler yalnızca tek bir kuruluşa özel olarak sağlanmaktadır. Hassas verileri olan kuruluşlar için faydalı olsa da, bu yaklaşım aynı zamanda depolama ve kapasite yönetimiyle ilgili zorlukları da beraberinde getirmektedir. Bir diğer dağıtım modeli ise topluluk bulutu (community cloud) modelidir. Bu modelde, ortak çıkarları olan kuruluş topluluklarına paylaşımlı hizmet sağlanmaktadır. Genel bulut kadar olmasa da, bu modelde de verilerin topluluğa açık olması nedeniyle güvenlik sorunları söz konusudur. Son model ise hibrit bulut (hybrid cloud) dağıtım modelidir. Bu model, genel bulutlardaki maliyet ve ölçeklenebilirlik avantajlarını sağlarken, aynı zamanda özel bulutların sunduğu güvenlik ve kontrolü de temin etmektedir. Esasen, her iki bulut dağıtım modelinin birleşik bir biçimidir [43].

3.2.2. Kenar bilişim

Kenar bilişim, kaynakların, başka bir deyişle sunucuların, mobil cihazlara, sensörlere, son kullanıcılara ve esasen IoT cihazlarına yakın konumlandırılmasıdır. Geleneksel bulut yapısından temel farkı, kenar bilişimin veriyi üreten cihazlara yakın, merkezi yapıdan ziyade dağıtık bir şekilde konumlandırılmış olmasıdır [45]. Kenar bilişim de, bulut bilişim gibi depolama ve işlem gerçekleştirme işlevlerini yerine getirmektedir. Ayrıca, bulut bilişim tarafından sunulan SaaS, IaaS ve PaaS hizmetlerini de sağlamaktadır. Kenar bilişimin bu hizmetleri sağlayabilmek için karakteristik özellikleri bulunmaktadır [46]:

- Yakınlık: Kenar bilişim hizmet kaynakları, kullanıcı deneyimini iyileştirmek ve kaynak tahsisi açısından avantaj sağlamak amacıyla oldukça yakın bir konumda yer alabilir. Bu yapının çeşitli avantajları bulunmaktadır. Örneğin, ağ yöneticileri, tüm Geniş Alan Ağını (Wide Area Network/WAN) geçmek zorunda kalmadan konuma dayalı hareketlilik hizmetlerini kolaylıkla sağlayabilmektedir. Yakınlık; daha hızlı ve daha doğru büyük veri analitiğine olanak tanır. Bu durum gerçek zamanlı uygulamalar için oldukça avantajlıdır.
- Hareketliliği destekleme: Kenar bilişim, Konum ve Kimlik Ayırma Protokolü (Locator ID Separation Protocol/LISP) kullanarak hareketliliği desteklemektedir. Bu protokol, bir cihazın ana kimliği ile konum kimliğini birbirinden ayırır. Bu sayede kenar yapı, mobil cihazlarla doğrudan iletişim kurabilmektedir. Böylece, mobil cihazlar bir ağdan başka bir ağa geçtiğinde konum bilgisi korunmakta ve kesintisiz bağlantı sağlanmaktadır.
- Konum farkındalığı: Bu özellik, mobil kullanıcıların en yakın kenar bilişim hizmetlerine erişmesini sağlamaktadır.
- Düşük gecikme: Kenar bilişim kaynaklarının kullanıcılara yakın olması, kaynakların ve hizmetlerin gecikme olmadan kullanılmasına olanak tanımaktadır. Bu durum kaynakların tıkanıklığını azaltmakta ve kullanıcı deneyimini arttırmaktadır.
- Bağlam farkındalığı: Bağlam bilgisi, mobil cihazların konum bilgisinden farklı olarak sahip olduğu bir özelliktir. Kenar bilişim, mobil cihazların bağlam bilgilerini kullanarak bulut yük boşaltma (cloud offloading) kararları alabilmekte, hizmetleri cihazlardan buluta aktarabilmekte ve kullanıcı deneyimini artırmaya yönelik hizmetleri tanımlayabilmektedir.

Yukarıda bahsedilen karakteristik özelliklere sahip olan kenar bilişim birçok çalışmaya göre temelde üç türe ayrılmaktadır: 1) Sis Bilişim (Fog Computing), 2) Bulutçuk (Cloudlet) ve 3) Mobil Kenar Bilişim (Mobile Edge Computing). Cisco tarafından geliştirilen Sis Bilişim, bulut bilişimde olduğu gibi depolama, hesaplama ve ağ hizmetleri sunmakta; ancak kullanıcıya daha yakın konumlandırılması ile diğer kenar bilişim türlerinden ayrılmaktadır. Öte yandan, Bulutçuk ve Mobil Kenar Bilişim, yalnızca yerel kaynakları kullanan ve özellikle mobil kullanıcılara hizmet veren kenar bilişim türleridir [46].

Kenar bilişim, bulut bilişimin yerini almış ve onu geri plana itmiş gibi görünse de, çoğu çalışmada bu iki yapının birbirini tamamlayıcı nitelikte olduğu belirtilmektedir. Bunun nedeni, mevcut teknoloji ortamında verilerin hâlen kalıcı olarak bulut bilişimde saklanması ve kenar bilişimin üstesinden gelemediği işlem görevlerinin bulut bilişim mimarisi içerisinde yönetilmesidir. Ancak, bu iki kavram aynı değildir ve aralarında bazı farklılıklar bulunmaktadır. Temelde, ilk fark bulut bilişim merkezi yapıda iken, kenar bilişim dağıtık bir yapıdadır ve kullanıcılara daha yakın çalışmaktadır. Bulut bilişim özellikle gerçek zamanlı olmayan veri işleme alanlarında derinlemesine analiz gerçekleştirme konusunda öne çıkarken, kenar bilişim gerçek zamanlı ihtiyaçları karşılamada çok daha başarılıdır ve yerel olarak çalışmaktadır. Bulut bilişimin merkeziyetçi bir yapıda olması sebebiyle verilerin merkezi yapısı iletilmesi için belirli ölçüde zaman ve bant genişliği ihtiyacını doğurmaktadır. Kenar bilişimin kullanıcıya daha yakın ve dağıtık şekilde konumlandırılmış olması sebebiyle bulut bilişimde gerçekleşen gecikmeyi ve bant genişliği ihtiyacını minimize etmektedir. Bu nedenle gerçek zamanlı uygulamalarda kenar bilişimin kullanımı daha faydalı olmaktadır. Ayrıca, kenar bilişimin küçük ölçekli verilerin analizinde daha iyi bir seçenek olduğu; buna karşılık, bulut bilişimin büyük verilerin analizinde ve depolanmasında daha uygun olduğu da söylenebilmektedir [46,47]. Bulut bilişim ve kenar bilişimin temel farkları Çizelge 3.1’de ifade edilmektedir:

Çizelge 3.1. Bulut bilişim ve kenar bilişimin temel farkları

	Uygulanabilir Durum	Ağ Bant Genişliği Baskısı	Gerçek Zamanlılık	Hesaplama Modu
Bulut bilişim	Küresel	Fazla	Yüksek	Büyük ölçekli merkezi işlem
Kenar bilişim	Yerel	Az	Düşük	Küçük ölçekli akıllı analiz

3.3. Blokzinciri

Blokzinciri, birbirine bağlı bloklardan oluşan değiştirilemez bilgiler içeren dağıtık deftere verilen isimdir. Bu teknolojinin temel özelliği, herkesin erişebildiği bir defter olmasıdır; bu defterin bir sahibi yoktur ve ağdaki her düğüm bu defterin kopyasına sahiptir [48].

Blokszinciri teknolojisinin derinlemesine anlaşılabilmesi için bazı temel kavramların açık bir şekilde kavranması gerekmektedir. Bu kavramların başında düğüm (node) kavramı gelmektedir. Düğüm, blokszinciri mimarisinde yer alan ve genellikle ağ oluşturan bilgisayarları ifade eder. Her düğüm, dağıtık defter(ledger) in bir kopyasını saklamakta, işlemlere katılmakta ve gerçekleştirilen işlemlerin geçerliliğini doğrulamaktadır [49].

Blokszinciri yapısının bir diğer temel unsuru bloktur. Blok, işlem detaylarını içeren, zaman damgası (timestamp) ve önceki bloğa referans sağlayan bağlantı gibi bilgileri barındıran bir veri yapısıdır. Bloklar, zincir yapısının sürekliliğini sağlamak amacıyla kriptografik özet algoritmalarıyla birbirine bağlanır. Bu bağlama sürecinde bloklarda yer alan işlemlerin bütünlüğü ve güvenliği, Merkle ağacı (Merkle tree) adı verilen özel bir veri yapısı ile sağlanır. Merkle ağacında yaprak düğümler, bireysel işlemlerin özet değerlerinden oluşurken; iç düğümler, alt düğümlerin özetlerinin birleşiminden oluşur. Böylece işlem bütünlüğü hızlı ve güvenilir bir biçimde doğrulanabilir [49-51].

Bir blok, temel olarak blok başlığı ve blok gövdesi olmak üzere iki ana bileşenden oluşur ve bu yapı Şekil 3.4'te gösterilmektedir. Blokların bu yapısı ve ardışık şekilde birbirine bağlanması, blokszinciri mimarisinde zincir yapısını meydana getirir. Blokların zincire eklenebilmesi için ise ağdaki düğümlerin belirli kurallar çerçevesinde fikir birliğine varması gerekir; bu sürece konsensüs (consensus) adı verilir. Birçok farklı konsensüs mekanizması mevcut olmakla birlikte, literatürde en sık karşılaşılanlar şunlardır: İş Kanıtı (Proof of Work/PoW), Pay Kanıtı (Proof of Stake/PoS), Temsilci Pay Kanıtı (Delegated Proof of Stake/DPoS), Pratik Bizans Hata Toleransı (Practical Byzantine Fault Tolerance/PBFT), Otorite Kanıtı (Proof of Authority/PoA), Geçen Zaman Kanıtı (Proof of Elapsed Time/PoET), Alan Kanıtı (Proof of Space/PoSpace) ve Kapasite Kanıtı (Proof of Capacity/PoC) [51].

Sürüm
Önceki Bloğun Özeti
Merkle Kökü
Zaman Damgası
Zorluk Hedefi
Tek Seferlik Sayı

Şekil 3.4. Blok yapısı [49]

Blokszinciri yapısının bir diğer temel unsuru işlem (transaction) kavramıdır. İşlem, ağ üzerinde gerçekleşen veri aktarımlarını temsil eden ve bloklara kaydedilen en küçük bilgi birimidir. Her blok, ardışık ve zaman damgalı işlemlerden oluşarak sistemin bütünlüğünü ve kronolojik yapısını garanti altına alır [49,50].

Son olarak, blokszinciri mimarisinde önemli bir diğer yapı taşı da akıllı sözleşme"(smart contract, chaincode) kavramıdır. Akıllı sözleşme; belirli koşullar sağlandığında üzerinde tanımlanan görevleri otomatik olarak yerine getiren uygulamalardır. Ethereum, Hyperledger Fabric gibi blokszinciri yapıları akıllı sözleşme yapısını desteklerken, Bitcoin gibi bazı blokszinciri yapıları da akıllı sözleşme yapısını desteklememektedir [49].

Blokszinciri çeşitleri, temel yapıları bakımından benzerlik gösterse de, blokszinciri teknolojisi farklı gereksinimlere göre çeşitlenebilen bir yapıya sahip olup, her bir türün kendine özgü kullanım alanları ve avantajları bulunmaktadır. Bu bağlamda, blokszinciri platformları da farklı güvenlik, erişim ve yönetim gereksinimlerine göre sınıflandırılabilir. Buna göre blokszinciri güvenlik, erişim ve yönetim gereksinimlerine göre üç çeşittir. Bunlardan ilki açık blokszinciridir. Açık blokszincirinde herhangi bir düğüm ağa katılabilir, işlemlere katılabilir ve blok oluşturabilir. Bitcoin, Ethereum, Dash ve Litecoin açık blokszincirine örnek olarak verilebilmektedir. Diğerleri özel blokszinciridir. Özel blokszincirinde hem okuma-yazma izinleri hem de ağa katılım sınırlandırılmıştır. Hydrachain, Ripple, Monax ve Multichain özel blockchain platformlarına örnektir. Üçüncü blokszinciri türü ise açık ve özel blokszinciri arasında bir ara çözüm olan konsorsiyum veya

izne dayalı (permissioned) blokzinciridir. Bu tür blokzinciri ağları, belirli bir düğüm grubu tarafından kontrol edilir. Mevcut izne dayalı blokzinciri platformlarına Hyperledger Fabric ve Corda örnek verilebilir [20,51].

Güvenlik, erişim ve yönetim gereksinimlerine göre birçok çeşidi olan blokzincirinin temel özellikleri, bu teknolojinin birçok alanda tercih edilebilir olmasını sağlamaktadır. Bu temel özellikler aşağıdaki gibidir [41]:

- Blokzinciri ağına eklenen veriler, blokzinciri defterinde yer almakta olup, sonradan değiştirilememekte veya silinememektedir.
- Bloklar, blokzinciri ağında birbirine özet algoritmalarıyla bağlıdır; bu yapı sebebiyle blokzinciri defterindeki verilerin değiştirilmesi imkânsız hâle gelmektedir.
- Blokzinciri defterine eklenen geçmiş işlemler, ağda güvenli bir şekilde saklanmaktadır.
- Blokzinciri ağındaki veriler, ağdaki tüm düğümler arasında dağıtılarak saklanmakta ve her düğüm blokzinciri defterinin bir kopyasına sahiptir.
- Blokzinciri sistemi, merkezi bir otoriteye ihtiyaç duymadan çalışmaktadır.
- Blokzinciri ağındaki veri değişiklikleri, yalnızca ilgili özel anahtara sahip kişiler tarafından yapılabilmekte, özel anahtarı olmayan kişiler veriler üzerinde herhangi bir değişiklik gerçekleştirememektedir.
- Blokzinciri ağı üzerinde saklanan verilerin gizliliği ve bütünlüğü, kriptografik algoritmalar tarafından korunmaktadır.

Bu temel özelliklere sahip olan blokzincirinin birçok alanda kullanıldığı söylenebilmektedir. Bu alanlara örnek olarak lojistik ve tedarik zinciri, gıda ve tarım, sağlık ve ilaç sektörü, enerji, banka ve finans, ulaşım, gayrimenkul ve eğitim verilebilmektedir [52]. Bu kullanım alanlarında tek bir blokzinciri yapısının kullanıldığını söylemek mümkün değildir. Yani performans, güvenlik ve ölçeklenebilirlik gibi çeşitli yönler açısından farklı blokzinciri platformlarını değerlendirmek ve analiz etmek için net bir yöntem bulunmamaktadır [53]. Çalışmalarda sistemlerin gereksinimlerine göre uygun blokzinciri yapısı seçilmektedir. Veri güvenliği, gizliliği ve bütünlüğünün amaçlandığı çalışmalarda açık blokzinciri türleri (örneğin Bitcoin, Ethereum, Dash) uygun bir yapı sunmamaktadır. Bunlar gibi açık blokzinciri yapıları yerine, veri güvenliğinin amaçlandığı çalışmalarda genellikle Hyperledger Fabric yapısı kullanılmaktadır [4,17,20,24,27,30,31]. Linux Foundation tarafından sunulan Hyperledger Fabric, özellikle IoT sistemlerinin

güvenliğinin sağlanmasında giderek daha fazla önem kazanmakta olup, işlem bilgilerini gizli tutmak amacıyla yalnızca belirli düğümlerin işlem verilerine erişebildiği kanal adı verilen bir yapı kullanılmaktadır. Hyperledger Fabric, mevcut verileri tutarak güncel bilgilere doğrudan erişim sağlayan dünya durumu (world state) ile bu verilerdeki geçmiş değişiklikleri kaydederek işlem geçmişini şeffaf ve denetlenebilir hâle getiren blokzinciri olmak üzere iki ana veri yapısından oluşur. İşlemleri gerçekleştirmek için Fabric ağında üç ana düğüm türü vardır: Onaylayıcı, sıralayıcı ve kaydedici düğüm. Onaylayıcı düğüm, işlemleri doğrulamaktan sorumludur. Sıralayıcı düğüm, işlemleri düzenler; kaydedici düğüm ise işlem kayıtlarını blokzincirine kalıcı olarak yazmaktadır [53,54]. Ayrıca, akıllı sözleşme yapısını destekleyen Hyperledger Fabric; Go, Java ve Node.js dillerinde zincir kodu (akıllı sözleşme) geliştirme imkânı sunmaktadır. Yapılan son çalışmalar, Go dilinin zincir kodu geliştirmek için en uygun dil olduğunu belirtmektedir [54].

3.4. Kullanılan Kriptografik Yöntemler

Sağlık verisi terimi bireylerden ve diğer kurumlardan aktif ve pasif yollarla toplanan çok büyük miktarda olan veriye denmektedir. Bu verilerin hassas veriler olması, veri sahibi tarafından birçok bilgiyi içermesi sebebiyle bu verilere yalnızca yetkili ve onaylanmış kişilerin erişmesi gerekmektedir [11]. Bu kapsamda bu başlık altında, çeşitli kriptografik algoritmalar performans, güvenlik ve uygulanabilirlik açısından değerlendirilmekte; ayrıca simetrik şifreleme algoritmalarıyla sıkça birlikte kullanılan Diffie-Hellman anahtar değişim algoritması da bu çerçevede ele alınmaktadır.

3.4.1. Blowfish- CBC ve HMAC-SHA256 şifreleme ve bütünlük sağlama algoritması

Blowfish algoritması, hızlı ve güvenilir bir blok şifreleme yöntemidir. Simetrik anahtar algoritmalarından biri olmakla birlikte, sabit uzunluktaki blokları (64 bit) şifrelemek için değişken uzunlukta anahtarlar (32 ila 448 bit) kullanabilmektedir. Literatürde sıkça kullanılan kısaltmasıyla CBC (Cipher Block Chaining), Türkçe ifadesiyle Şifre Blok Zincirleme algoritması, şifreleme sırasında her bir blok veriyi bir önceki şifrelenmiş blokla ilişkilendirerek çalışır ve bu sayede veri desenlerinin tekrar etmesini engelleyerek veri güvenliğini artırır. Blowfish-CBC birleşimine ek olarak temel bütünlük kontrolü için HMAC (Hash-based Message Authentication Code/Anahtarlı Karma Mesaj Doğrulama Kodu) yapısı kullanılmaktadır. Blowfish-CBC ve HMAC-SHA256 algoritmalarının

önerilen modelde şifreleme ve deşifreleme işlemlerine ilişkin sözde kodları aşağıda sunulmaktadır:

Blowfish-CBC ve HMAC-SHA256 Şifreleme Metotu Sözde Kodu

for Metot BlowfishSifreleme(salt, blowfishAnahtarı, iv, hmacAnahtarı) çalıştırıldığında
do

2: Salt verisi için HMAC hesapla (HMAC-SHA256 kullanarak)

3: Salt verisi ile hesaplanan HMAC değerini birleştir

4: Blowfish algoritması, CBC modu ve PKCS5Padding kullanarak iv ile birlikte veriyi şifrele

5: Şifrelenmiş veriyi döndür

6: end for

Blowfish-CBC-HMAC Deşifreleme Metotu Sözde Kodu

for Metot BlowfishDesifreleme (degerwithHmac, blowfishAnahtarı, iv, hmacAnahtarı)
çalıştırıldığında do

2: Blowfish algoritması, CBC modu ve PKCS5Padding kullanarak iv ile birlikte veriyi çöz

3: Çözülen veriyi veri kısmı ve HMAC kısmı olarak ayır

4: Veri için beklenen HMAC değerini hesapla (HMAC-SHA256 kullanarak)

5: Hesaplanan HMAC ile çözülen HMAC'i karşılaştır

- Eğer eşleşiyorsa "HMAC doğrulandı" mesajı yazdır

- Eşleşmiyorsa hata fırlat ("HMAC doğrulaması başarısız! Veri bozulmuş olabilir.")

6: Deşifrelenmiş veriyi döndür

7: end for

3.4.2. AES-GCM şifreleme ve bütünlük sağlama algoritması

AES (Advanced Encryption Standard/Gelişmiş Şifreleme Standardı), simetrik şifreleme algoritmalarından biridir ve 128, 192 veya 256 bit anahtar uzunluklarıyla şifreleme yapmaya olanak tanır. Şifreleme sürecine ek olarak, GCM (Galois/Counter Mode) modu ise veri bloğunun bütünlüğünü sağlamak için bir mesaj doğrulama kodu (MAC/Message Authentication Code) üretir. MAC değeri, verinin şifrelenmiş hâliyle birlikte iletilir ve bu MAC değeriyle alıcı, deşifreleme esnasında verinin bütünlüğünü kontrol edebilir. GCM modu paralel işlemeyi destekler ve bu durum şifreleme işlemlerini hızlandırır. Bu özellikleriyle AES-GCM, veri iletim güvenliği için yaygın olarak tercih edilmektedir. AES-GCM algoritmalarının önerilen modelde şifreleme ve deşifreleme işlemlerine ilişkin sözde kodları aşağıda sunulmaktadır:

AES-GCM Şifreleme Metotu Sözde Kodu

for Metot AESSifreleme(hamVeri, aesAnahtari) çalıştırıldığında do

2: 12 baytlık rastgele nonce oluştur

3: AES-GCM algoritması için anahtarı 128-bit olarak dönüştür

4: AES-GCM modunu başlat (şifreleme modu, anahtar ve nonce kullanarak)

5: Veriyi şifrele

6: Nonce ve şifrelenmiş veriyi birleştir

7: Şifrelenmiş veriyi (nonce ile birlikte) döndür

8: end for

AES-GCM Deşifreleme Metodu Sözde Kodu

for Metot AESDesifreleme (şifrelenmişVeri, aesAnahtarı) çalıştırıldığında do

2: Şifrelenmiş verinin ilk 12 baytını nonce (IV) olarak ayır

3: Geriye kalan veriyi şifreli metin olarak ayır

4: AES-GCM algoritması için anahtarı 128-bit olarak dönüştür

5: AES-GCM modunu başlat (şifre çözme modu, anahtar ve nonce kullanarak)

6: Şifreli veriyi orijinal değerine dönüştür

7: end for

3.4.3. ChaCha20-Poly1305 şifreleme ve bütünlük sağlama algoritması

ChaCha20, bir akış şifreleme algoritmasıdır. Sabit uzunluktaki bir anahtar ve tek seferlik sayı (nonce) değerlerini kullanarak veriyi simetrik şekilde şifrelemektedir. Poly1305 algoritması ise, şifrelenmiş verinin bütünlüğünü sağlamak amacıyla kullanılan bir MAC algoritmasıdır. Bu ikili algoritmanın birlikte kullanımı modern, hızlı ve güvenlidir ve bu iki algoritma birlikte çalışarak hem verinin gizliliğini hem de verinin yetkisiz şekilde değiştirilmediğini garanti eder. Özellikle mobil ve düşük güçlü cihazlarda (örneğin, IoT cihazları) yüksek performans sunması sebebiyle yaygın şekilde tercih edilmektedir.

ChaCha20-Poly1305 Şifreleme Metodu Sözde Kodu

for Metot ChaCha20Sifreleme (hamVeri, gizliAnahtar, nonce) çalıştırıldığında do

2: Şifreleme algoritması "ChaCha20-Poly1305" olarak seç

3: Gizli anahtar nesnesi oluştur

4: GCM şeması için 128-bit doğrulama etiketi ve nonce tanımla

- 5: Şifreleme işlemi başlat (gizliAnahtar + nonce ile)
- 6: Ham veri (plaintext) şifrelenir → "şifreli veri + etiket" elde et
- 7: Nonce, şifreli veri ve etiketi birleştirerek tek bir çıktı oluştur
- 8: Şifreli veriyi (nonce + ciphertext + tag) döndür

6: end for

ChaCha20-Poly1305 Deşifreleme Metotu Sözde Kodu

for Metot ChaCha20Desifreleme(şifreliVeriVeEtiket, gizliAnahtar, nonce)
çalıştırıldığında do

2: ChaCha20-Poly1305 algoritması ve BouncyCastle sağlayıcısı ile Cipher nesnesini oluştur

3: Verilen anahtarı ChaCha20 algoritması için anahtar nesnesine dönüştür

4: Verilen nonce değerini Initialization Vector (IV) olarak ayarla

5: Cipher'ı şifre çözme modunda, anahtar ve nonce ile başlat

6: Cipher ile verilen şifreli veriyi çöz

8: end for

3.4.4. Diffie Hellman anahtar değişim algoritması

Diffie-Hellman anahtar değişim algoritması, iki tarafın güvenli bir iletişim kanalı üzerinden ortak bir gizli anahtarı paylaşımlarını sağlayan bir kriptografik protokoldür. Algoritma asal sayılar ve modüler aritmetik kullanarak üçüncü bir kişinin anahtarı elde etmesini zorlaştırmayı amaçlamaktadır. Bu algoritma ile iki taraf daha önce hiç iletişim kurmamış olmamış olsa bile ortak bir anahtar üretebilmektedir. Diffie Hellman algoritması

genellikle simetrik şifreleme algoritmalarıyla kullanılmaktadır. Önerilen modelde anahtar değişim algoritması ağ geçidi ve kenar bilişim arasında gerçekleşecektir. Bu doğrultuda bu iki yapı arasında uygulanan Diffie Hellman anahtar değişim algoritmasının matematiksel gösterimi aşağıdaki gibidir:

1. Ortak bir asal sayı p ve ilkel kök g seçilir. Bu değerler her iki taraf tarafından bilinir.

2. Her taraf kendine özel gizli anahtar belirler:

Ağ geçidi: a

Kenar bilişim: b

3. Açık anahtarlar hesaplanır ve karşı tarafa gönderilir:

Ağ geçidi: $A = g^a \bmod p$

Kenar bilişim: $B = g^b \bmod p$

4. Ortak anahtar her iki taraf tarafından hesaplanır:

Ağ geçidi: $K = B^a \bmod p$

Kenar bilişim: $K = A^b \bmod p$

5. Sonuçta, iki taraf aynı ortak anahtara ulaşır:

Ortak anahtar: $K = g^{ab} \bmod p$

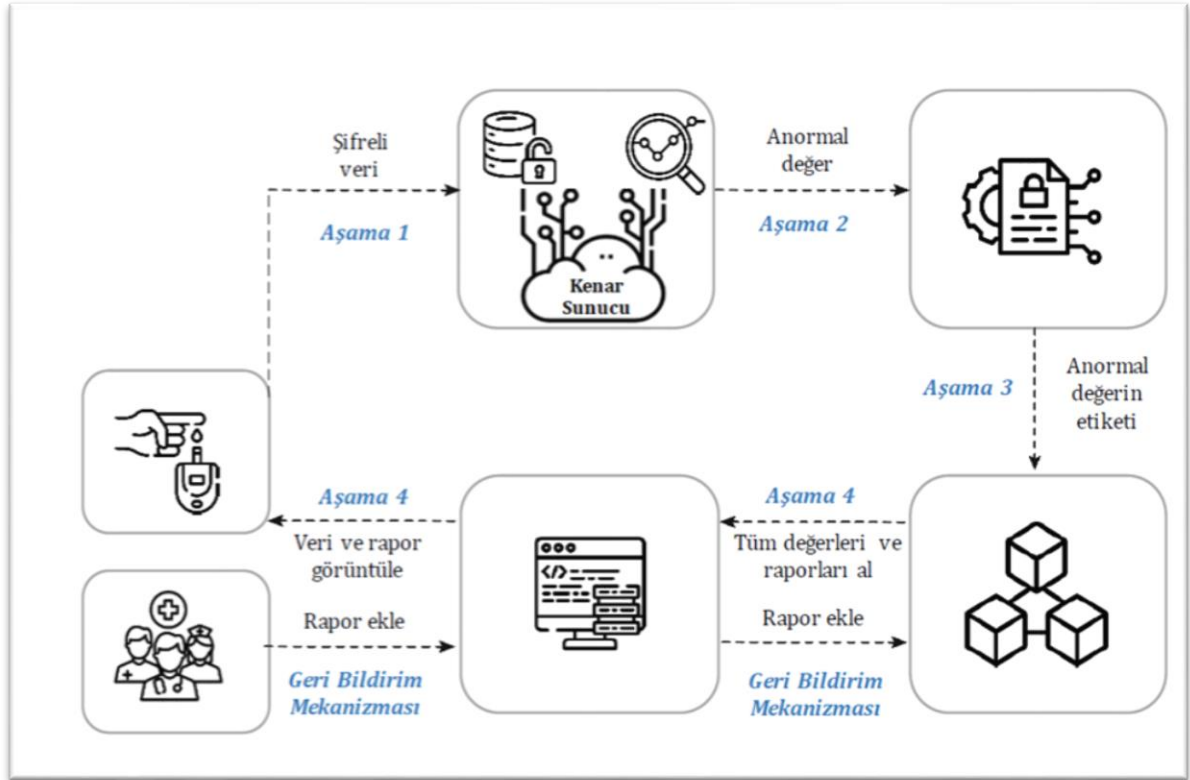
Genellikle simetrik şifreleme algoritmalarıyla kullanılan Diffie Hellman algoritması sonucunda oluşturulan ortak anahtar bu şekilde güvenli şekilde iki yapı arasında paylaşılmış olur.

4. ÖNERİLEN MODEL

Uzaktan sağlık teknolojisinin gelişimi, tıbbi durumların izlenme ve tedavi edilme biçimini dönüştürmüş, böylece hastalar için daha fazla esneklik ve erişilebilirlik sağlamıştır [10]. Özellikle kalp atış hızı, kan basıncı ve kan şekeri seviyeleri gibi sağlık parametrelerinin sürekli izlenmesi, diyabet gibi kronik hastalıkların etkin yönetimi için giderek daha önemli hale gelmektedir [20]. Bu sağlık durumları, hasta tedavisini optimize etmek ve komplikasyonları önlemek amacıyla sürekli ve hassas izleme gerektirmektedir. Bu parametreler arasında, kan şekeri seviyelerinin güvenli takibi ve yönetimi özellikle önemlidir; çünkü bu durum, hastanın kendi rahatsızlığını kontrol etme yeteneğini doğrudan etkiler ve kronik sağlık komplikasyonları riskini azaltır [7].

Hassas sağlık verilerinin uzaktan izlenmesi, özellikle veri güvenliği ve gizliliği açısından çeşitli zorluklar barındırmaktadır. Hastaların sağlık sistemlerine duyduğu güvenin sürdürülebilmesi için, bu verilerin yetkisiz erişimden korunması ve izleme sürecinde bütünlüğünün sağlanması büyük önem taşımaktadır. Bu sorunlara çözüm olarak geliştirilen SağlıkZinciri modeli; IoT, kenar bilişim, blokzinciri ve akıllı sözleşme teknolojilerini entegre ederek, kan şekeri seviyelerinin şifrelenmiş biçimde iletimini ve değiştirilemez şekilde güvenli bir ortamda saklanmasını amaçlamaktadır.

Önerilen modelin aşamaları, her bir aşamanın hasta verilerinin güvenliği ve bütünlüğünün korunmasına nasıl katkı sağladığını açıklayan sonraki bölümlerde detaylandırılmıştır. Şekil 4.1'de önerilen modelin mimarisi, IoT, kenar bilişim, blokzinciri ve akıllı sözleşmeler kullanılarak kan şekeri seviyelerinin güvenli izlenmesi ve yönetilmesinde yer alan temel aşamalar genel hatlarıyla sunulmaktadır.



Şekil 4.1. IoT, kenar bilişim, blokzinciri ve akıllı sözleşmeler kullanarak kan şekeri seviyelerinin güvenli izlenmesi ve yönetimi için SağlıkZinciri modeli

4.1. Aşama 1: IoT Cihazları ve Kenar Bilişim Yapısı Arasında Güvenli Veri Toplama ve Şifreli Aktarım Süreci

İlk aşamada, hasta verilerinin toplanması, IoT cihazları aracılığıyla sensörler tarafından gerçekleştirilir. Sağlık verileri yüksek derecede hassas olduğundan, ağ tabanlı tehditlere karşı savunmasıdır. Bu nedenle, verilerin IoT cihazları tarafından toplandıktan sonra doğrulanması ve iletim sırasında güvenliğinin sağlanması için şifreleme kritik bir rol oynar. Önerilen modelde, verilerin güvenli şekilde şifrelenmesi için ChaCha20 algoritması ve bütünlüğünün kontrolü için Poly305 bütünlük kontrolü algoritması kullanılmaktadır. Şifreleme işlemi, veri toplama ve iletim süreçleri boyunca bilgilerin korunmasını garanti eder. Ayrıca Poly1305 algoritması ile de verinin iletilmesi sırasında bozulmadığı garanti edilmiş olmaktadır. Modelde kullanılan şifreleme mekanizmasının matematiksel formülasyonu aşağıdaki gibidir:

4.1.1. Diffie-Hellman anahtar deęiřimi ile ChaCha20 anahtarının üretilmesi

İlk adımda, saęlık verilerini (kan řekeri deęerlerini) toplayan IoT sensörlerinden veriyi alarak řifreleyen aę geęidi yapısı (A) ile bu verileri işlemekle sorumlu kenar biliřim yapısı (B), ortak bir modülüs p ve taban g deęerlerini paylařır. Her taraf, kendi gizli anahtarını baęımsız olarak seęer ve ortak gizli anahtarı oluřturmak üzere açık anahtarlarını deęiř tokuř eder. Aę geęidi yapısı (A) ve kenar biliřim yapısı (B) ařaęıdaki denklemde gösterildięi gibi özel anahtarlarını kullanarak açık anahtarlar üretir:

$$A_{pub} = g^a \bmod p, B_{pub} = g^b \bmod p \quad (4.1)$$

Burada, A_{pub} ve B_{pub} sırasıyla aę geęidi yapısı ve kenar biliřim yapısının açık anahtarlarını, a ve b ise tarafların özel anahtarlarını temsil eder.

Her iki taraf daha sonra Diffie-Hellman anahtar deęiřim algoritması kullanarak ortak gizli anahtar K_{ortak} 'ı hesaplar:

$$K_{pub} = B_{pub}^a \bmod p = A_{pub}^b \bmod p \quad (4.2)$$

Elde edilen bu ortak anahtar K_{ortak} ; aę geęidi yapısı ve kenar biliřim yapısı tarafından bilinen ortak gizli anahtar olur ve güvenli veri řifrelemesi için ChaCha20 anahtarının türetilmesinde kullanılır.

$$K_{ChaCha20} = H(K_{ortak}) \quad (4.3)$$

Ortak anahtar K_{ortak} , bir özet fonksiyonu H aracılıęıyla 256-bit uzunluęunda ChaCha20($K_{ChaCha20}$) anahtarına dönüřtürülür:

4.1.2. Salt üretimi

İkinci adımda, anahtar türetme için rastgele bir tuz (salt) S_{salt} oluřturulur. Eř. 4.4'te matematiksel olarak ifade edilen bu ek güvenlik önlemi, aynı ChaCha20 anahtarının birden fazla veri noktası için tekrar kullanılmasının yol açabileceęi saldırı risklerini azaltır.

$$S_{salt} = \text{GenerateSalt}() \quad (4.4)$$

Burada salt değeri (S_{salt}) aynı verinin (örneğin, iki aynı kan şekeri ölçümü) farklı şifreli sonuçlar üretmesini sağlayarak verinin gizliliğini korur.

4.1.3. Anahtar türetme

Her yeni veri noktası (örneğin, yeni bir kan şekeri ölçümü) için, ChaCha20 anahtarı önceki ChaCha20 anahtarı $K_{ChaCha20}$ ve oluşturulan salt S_{salt} kullanılarak Anahtar Türetme Fonksiyonu (Key Derivation Function, KDF) ile yeniden türetilir. Bu fonksiyon, hesaplama karmaşıklığını artırarak kaba kuvvet saldırılarına karşı direnç sağlar. Türetilen anahtarın ($K_{türetilmiş}$) matematiksel gösterimi Eş. 4.5'teki gibidir:

$$K_{türetilmiş} = KDF(K_{ChaCha20}, S_{salt}, \text{iterasyonlar}) \quad (4.5)$$

Burada iterasyon sayısı güvenlik ve performans arasında denge sağlar.

4.1.4. Veri şifreleme

Yeniden türetilen ChaCha20 anahtarı $K_{türetilmiş}$, hastanın kan şekeri verisi gibi verileri şifrelemek için kullanılır. Eş. 4.6'da ifade edilen işlemle gerçekleşen şifreleme işlemi, ağ geçidi yapısında gerçekleştirilerek verinin kenar bilişim yapısına güvenli şekilde iletilmesini sağlar.

$$E_{ChaCha20}(m) = ChaCha20(m, K_{türetilmiş}) \quad (4.6)$$

Burada m , hastanın kan şekeri değerini ifade etmektedir. Türetilen anahtar, aynı temel ChaCha20 anahtarı kullanılsa bile her veri noktası için benzersiz bir şifreleme sağlar.

Bu çok aşamalı süreç sayesinde önerilen model; Diffie Hellman anahtar değişim algoritmasıyla ChaCha20 anahtarının güvenliğini sağlamakla kalmayıp, kan şekeri seviyeleri ile ilgili hasta sağlık verilerinin gizliliğini de koruyarak, anahtar üretimi ve şifreleme işlemlerinde performans ve ölçeklenebilirliği artırmaktadır. Yukarıda 6 eşitlik ile matematiksel gösterimi yapılan şifreleme sürecinin sözde kodu aşağıdaki Algoritma 1 ile ifade edilmektedir:

Algoritma 1: Ağ Geçidi Yapısı (Anahtar Oluşturma, Anahtar Değişimi, Şifreleme ve Veri İletimi)

```

1: for Her önceden belirlenmiş zaman aralığında do
2:   IoT sensöründen toplanan çevresel veriyi al (örneğin kan şekeri seviyesi)
3:   Mevcut zamanı al  $\rightarrow$  currentTime = LocalTime.now()
4:   if  $12:00:00 \leq \text{currentTime} \leq 12:00:10$  ise
5:     İletim isteğini reddet
6:   else
7:     Gizli anahtarı bellek önbelleginden al
8:     Salt değerini bellek önbelleginden al
9:     Anahtar türetme fonksiyonu (örneğin PBKDF2) kullanarak gizli anahtar ve salt ile
       ChaCha20 anahtarını türet
10:    Türetilen ChaCha20 anahtarı ile veriyi (örneğin kan şekeri seviyesi) şifrele
11:    Şifrelenmiş veriyi kenar bilişim yapısına gönder
12:  end if
13: end for
14: for Her gece yarısı (00:00) do
15:   Kenar bilişim yapısı ile anahtar değişimi için Diffie-Hellman anahtarlarını oluştur
16:   Açık anahtarı şifrele
17:   Kenar bilişim yapısının açık anahtarını bekle
18:   Kenar bilişim yapısının açık anahtarını aldıktan sonra Diffie-Hellman ile paylaşılan
       gizli anahtarı oluştur
19:   Paylaşılan gizli anahtardan ChaCha20 gizli anahtarını türet
20:   Anahtar türetme fonksiyonu (örneğin PBKDF2) kullanarak bir salt oluştur
21:   ChaCha20 gizli anahtarı ve salt kullanarak şifreleme anahtarını türet
22:   Salt değerini ChaCha20 ile şifrele
23:   Türetilen gizli anahtar ve salt değerini sonraki işlemlerde kullanmak üzere bellek
       önbelleginde sakla
24: end for

```

4.2. Aşama 2: Kenar Bilişim Yapısından Akıllı Sözleşmeye – Verilerin Deşifrelenmesi, Sınıflandırılması ve Otomatik Karar Süreci

Bu aşamada şifrelenmiş veriler kenar bilişim yapısına iletilmekte ve güvenli bir şekilde veri deşifreleme işlemi gerçekleştirilmektedir. Veri deşifreleme tamamlandığı anda, verilerin değeri belirlenmek üzere detaylı bir analiz yapılmaktadır. Bu analiz sonucuna dayanarak veriler, veri değerlerinin veri sahibi üzerindeki beklenen etkilerine göre kategorilere ayrılmaktadır. Aralıklar tanımlanmadan önce, gerçek dünyaya ait sağlık verileri analiz edilerek mevcut fizyolojik koşulları doğru şekilde yansıtan değerler göz önüne alınmaktadır ve analiz de bu değerlere göre yapılmaktadır [68]. Analiz için kategoriler oluşturulmuştur. Beş adet kategori bulunmaktadır: ‘normal’, ‘dikkat’, ‘yüksek’, ‘çok yüksek’ veya ‘acil’. 70 ile 139 arasındaki veriler ‘normal’ olarak kabul edilmektedir; bu, önemli bir sağlık riski olmadığını ve değerlerin kabul edilebilir aralıkta olduğunu göstermektedir. 140-179 aralığındaki veriler ‘dikkat’ kategorisindedir; bu durum, sağlık açısından potansiyel bir risk olabileceği ve yakın takip ve/veya ek inceleme gerektirebileceği anlamına gelmektedir. 180-199 aralığındaki ‘yüksek’ kategorisi, verilerin önemli bir sağlık sorunu, örneğin diyabet başlangıcı gibi, gösterdiğini ve müdahale gerektiğini belirtmektedir. ‘Çok yüksek’ kategorisine giren veriler 200 ile 249 arasındadır ve bu değerler, acil müdahalenin gerekli olduğunu, çünkü sağlık göstergesinin çok yüksek seviyelerde olduğunu göstermektedir. 250’nin üzerindeki değerler ‘acil’ sınıflandırmasına dahildir ve hayati bir tehdit, örneğin komaya veya ölüme yol açabilecek bir durum olduğunu göstermekte ve derhal tıbbi müdahale gerektirmektedir. Normal aralık dışındaki herhangi bir veri anormal olarak sınıflandırılır ve bu durum, akıllı sözleşmenin önceden tanımlanmış işlemleri, örneğin sağlık profesyonellerine uyarı gönderme veya potansiyel sağlık risklerini azaltmaya yönelik diğer önlemleri uygulama gibi, yürütmesini tetikler. Bu sınıflandırmaların detaylı aralıkları ve karşılık geldikleri kategoriler, aşağıda sunulan Çizelge 4.1’de gösterilmektedir.

Çizelge 4.1. Kan şekeri düzeylerine göre kategoriler ve açıklamaları

Değer Aralığı (mg/dL)	Kategori	Açıklama
70 – 139	Normal	Kabul edilebilir aralık, önemli bir sağlık riski bulunmamaktadır.
140 – 179	Dikkat	Potansiyel risk taşır, yakın takip ve/veya ek inceleme gerekebilir.
180 – 199	Yüksek	Önemli bir sağlık sorununun işareti olabilir, müdahale gerekebilir.
200 – 249	Çok Yüksek	Acil müdahale gerektirebilir, sağlık göstergesi ciddi derecede yüksektir.
250 ve üzeri	Acil	Hayati tehdit oluşturabilir, derhal tıbbi müdahale gerektirir.

Belirtilen aralıklar doğrultusunda işleyen sistem, sağlık verilerinin doğru şekilde yönetilmesini sağlamakta, zamanında müdahalelere olanak tanımakta ve genel sağlık sonuçlarının iyileştirilmesine katkıda bulunmaktadır. Kenar bilişim yapısında şifrelenmiş verinin deşifreleme süreci Algoritma 2’de sunulmaktadır:

Algoritma 2: Kenar Bilişim Yapısı (Anahtar Üretimi, Anahtar Değişimi, Şifre Çözme)

- 1: for Her gece yarısı (12:00 AM) için döngüyü başlat do
- 2: Ağ geçidi yapısına ait açık anahtarı al
- 3: Ağ geçidi yapısının açık anahtarı alındıktan sonra Diffie-Hellman algoritmasını kullanarak ortak bir gizli anahtar oluştur
- 4: ChaCha20 anahtarını kullanarak salt değerini deşifre et
- 5: Türetilmiş gizli anahtarı ve salt değerini sonraki işlemlerde kullanmak üzere bellek önbelleğinde sakla
- 6: end for
- 7: for Ağ geçidi yapısından veri alındığında döngüyü başlat do
- 8: Gizli anahtarı bellek önbelleğinden al (anahtar = keyCache.get("dailySecretKey"))
- 9: Salt değerini bellek önbelleğinden al (salt = keyCacheSalt.get("dailySalt"))
- 10: Gizli anahtar ve salt değeri kullanılarak, bir anahtar türetme fonksiyonu (örneğin

PBKDF2) ile ChaCha20 anahtarı türet

11: Veriyi (örneğin kan şekeri düzeyi) türetilen ChaCha20 anahtarıyla çöz

12: end for

4.3. Aşama 3: Sağlık Verilerinin Güvenli Kaydı İçin Akıllı Sözleşme ve Blokzinciri Etkileşimi

Üçüncü aşamada, akıllı sözleşme blokzinciri ile etkileşime geçmektedir. Anormal olarak etiketlenmiş veriler, bütünlüğünün korunması amacıyla blokzincirine güvenli bir şekilde kaydedilmektedir. Blokzinciri, tüm işlemleri, veri etiketlerini ve zaman damgalarını kalıcı biçimde kaydeden açık, merkeziyetsiz ve değiştirilemez bir defter yapısına sahiptir. Bu özellikleri sayesinde, verilerde yapılan herhangi bir değişiklik kaynağına kadar izlenebilir hâle gelmekte; böylece veri üzerinde yetkisiz değişiklik yapılması engellenmektedir. Akıllı sözleşme içerisinde yer alan metotlar Çizelge 4.2’de ayrıntılı olarak sunulmaktadır:

Çizelge 4.2. Akıllı sözleşme içerisinde yer alan metotlar

Fonksiyon Adı	Açıklama
<i>DefteriBaslat()</i>	Defteri, başlangıç kayıtlarını oluşturarak başlatır.
<i>DegerEkle()</i>	Deftere yeni bir etiket (değer) ekler.
<i>RaporEkle()</i>	Kullanıcıların deftere yeni bir rapor eklemesine olanak tanır.
<i>TumDegerleriGetir()</i>	Defterde saklanan tüm etiketleri (değerleri) getirir.
<i>TumRaporlariGetir()</i>	Defterde saklanan tüm raporları getirir.

4.4. Aşama 4: Blokzincir Üzerindeki Anormal Verilerin Zaman Damgalı Grafiklerle Görselleştirilmesi ve Yetkili Arayüzü Üzerinden Gerçek Zamanlı Geri Bildirim Sağlanması

Dördüncü aşamada, anormal veriler hastalar ve yetkili kullanıcılar tarafından erişilebilen bir arayüz aracılığıyla izlenmektedir. Bu arayüze erişim sağlamak isteyen hastalar, öncelikle Şekil 4.2’de gösterilen ekrandan sisteme kayıt olmaları gerekmektedir. Bu

ekranda hasta kayıt olurken ‘Doktor Seçin’ seçeneğini kullanarak verilerine erişecek sağlık uzmanını seçmektedir.

Hasta Takibi

Kayıt Ol

Kullanıcı Adı:

E-mail:

Şifre:

Doktor Seçin: Bir doktor seçin ▾

Kayıt Ol

Şekil 4.2. Kayıt olma arayüzü

Kayıt olma işleminden sonra kullanıcılar (veri sahibi veya sağlık uzmanı) Şekil 4.3'te gösterilen ekrandan kullanıcı adı, parola ve e-mail adreslerini girerek kimlik doğrulama işlemini gerçekleştirmektedir. Başarılı bir şekilde giriş yapan veri sahibi kullanıcılar yalnızca kendi verilerine erişebilirken, uzman kullanıcılar ise yalnızca kendileriyle veri paylaşımı yapılmış olan hastaların verilerine erişim hakkına sahiptir. Kullanıcının sisteme başarılı bir şekilde giriş yapmasının ardından karşılaştığı arayüz Şekil 4.4'te sunulmaktadır.

Hasta Takibi

Giriş Yap

Kullanıcı Adı:

Şifre:

E-posta Adresi:

Giriş Yap

Hesabınız yok mu? Kayıt Olun

Şekil 4.3. Giriş yapma arayüzü

Hasta Takibi

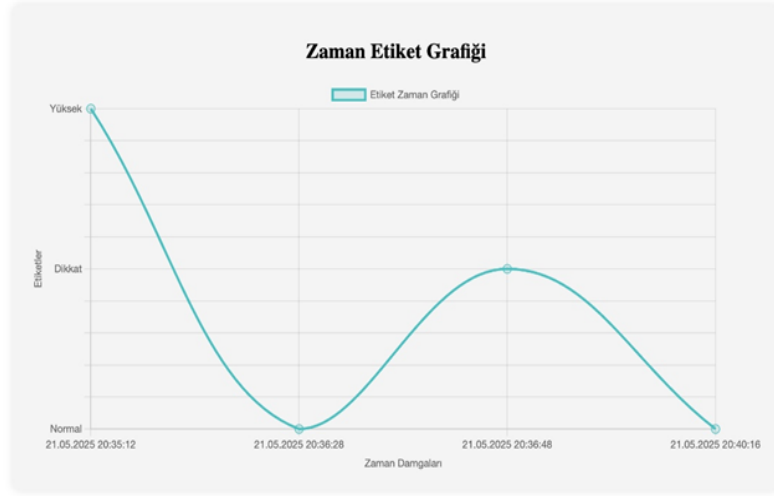


Şekil 4.4. Kullanıcının başarılı giriş sonrası karşılaştığı ana arayüz

Açılan bu arayüz üzerinden, kullanıcı 'Acil Durum Ekle', 'Verileri Görüntüle', 'Durumları Görüntüle' ve 'Çıkış Yap' gibi işlevsel seçeneklere erişim sağlayabilmektedir. 'Acil Durum Ekle' seçeneğine yalnızca sağlık uzmanları erişim sağlayabilmektedir. Diğer seçeneklere tüm kullanıcılar erişim sağlayabilmektedir.

'Verileri Görüntüle' seçeneği aracılığıyla erişilen arayüzde, veri etiketleri zaman damgalı bir çizgi grafik üzerinde gösterilmektedir. Bu grafik verinin blokzincire ne zaman eklendiğine dair zaman bilgisini de içermektedir. Söz konusu grafik yapısına ilişkin örnek bir gösterim, Şekil 4.5'te sunulmaktadır. Ayrıca, yetkili kişiler bu arayüz aracılığıyla hastaya geri bildirimde bulunabilmektedirler. Böylece, arayüz ile gerçek zamanlı geri bildirim ve müdahale imkânı sağlanmaktadır. Bu yapıya ilişkin daha ayrıntılı bilgiler "Geri Bildirim Mekanizması" başlığı altında sunulacaktır.

Hasta Takibi



Geri

Şekil 4.5. Veri görüntüleme ve geri bildirim arayüzüne ait zaman damgalı grafiksel gösterim

4.5. Geri Bildirim Mekanizması

Geri bildirim mekanizması, yetkili kişilerin blokzinciri ile etkileşim kurarak hastaya gerçek zamanlı geri bildirim sağlamalarına olanak tanır. Veriler ve etiketler, zaman etiketli bir çizgi grafik üzerinde görselleştirilir. Yetkili bir kişi, grafik üzerinde belirli bir zaman noktasına tıkladığında, durum ekleme arayüzü açılır. Bu kişi kendi değerlendirmesine bağlı olarak şu tür durumlar ekleyebilir: “iyi,” “izlenmeli,” “stabil,” “doktorla görüş,” “acil” veya “kritik.” Şekil 4.6’da geri bildirim ekleme arayüzü görülmektedir. Bu durumlar sistemde saklanır ve hastanın arayüzünde zaman damgalarına göre organize edilerek görüntülenir. İlgili arayüzün kullanıcıya sunum biçimini yansıtan örnek bir görsel, Şekil 4.7’de ayrıntılı olarak gösterilmektedir.

Hasta Takibi



Ger

Hasta Takibi

Durum Ekle

- İyi:** Kullanıcı, hastanın sağlığı stabil olduğunu belirtir.
- Kritik:** Durum gerçekten acil ve ciddi, hemen müdahale gerekebilir.
- Acil:** Acil müdahale gerekebilir, ancak hayatı tehdit edici değil.
- Doktora Gitmeli:** Durum ciddi, ancak acil müdahale gerekmiyor.
- Stabil:** Durum kontrol altında, ancak takip edilmesi gerekiyor.
- Takip Edilmeli:** Durum şüpheli ve izlenmeli.

Gönder

Ger

- ☒
- ☐
- ☐
- ☐
- ☐
- ☐

Şekil 4.6. Durum ekleme arayüzü

Hasta Takibi

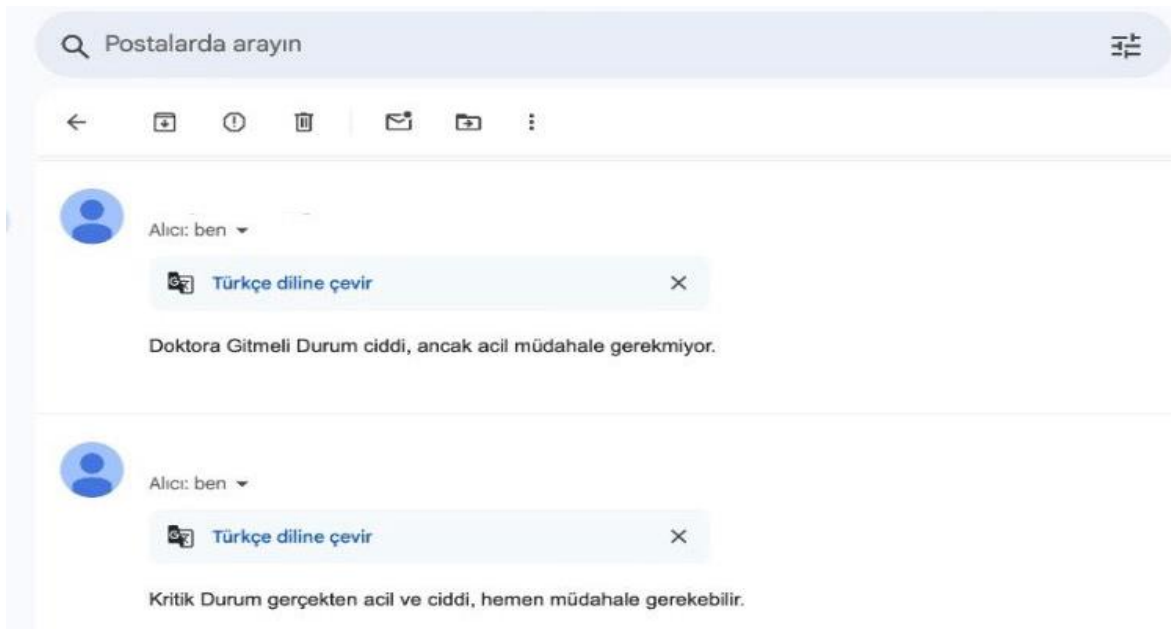
Durumlar

Zaman **Durum**
21.05.2025 20:41:27 İyi Kullanıcı, hastanın sağlığı stabil olduğunu belirtir.

Geri

Şekil 4.7. Zaman damgalarına göre organize edilmiş hasta durum görüntüleme arayüzü

Yukarıda bahsedilen arayüzlere ek olarak acil bir durumda, başka bir geri bildirim mekanizması devreye girer. Eğer yetkili kişi durumu kritik olarak değerlendirirse, “acil” veya “kritik” durumlarını belirtebilir. Bu tür bir bildirim, hastaya e-posta yoluyla iletilir. Bu mekanizma, sağlık hizmeti sağlayıcılarının hastaya anlık rehberlik sunmasını sağlayarak, hastanın durumunun proaktif bir şekilde yönetilmesini temin eder. Acil durumlara ilişkin geri bildirimlerin sağlandığı arayüz, Şekil 4.6’da gösterilen ‘Durum Ekle’ arayüzü ile aynı yapıyı paylaşmaktadır. Bu arayüz aracılığıyla veri sahibine gönderilen örnek e-posta içeriği ise Şekil 4.8’de sunulmaktadır.



Şekil 4.8. Acil durum bildirimi için hastaya iletilen e-posta örneği



5. ÖNERİLEN MODELİN ANALİZİ

Bu bölümde önerilen model, güvenlik analizi ve performans analizi olmak üzere iki ana başlık altında incelenecektir. Analiz sırasında sıkça bahsedilen düğüm yapısının daha iyi kavranabilmesi için modelde dört temel düğüm bulunduğu belirtilmelidir. Bunlardan ilki, verilerin toplandığı IoT sensörlerinden oluşan uç düğümdür. İkinci düğüm, verilerin şifrelenip güvenli bir şekilde taşındığı ağ geçidi düğümüdür. Üçüncü düğüm ise verilerin çözüldüğü, analiz edildiği ve blokzincirine iletildiği ara katman, yani kenar bilişim düğümüdür. Son olarak, verilerin kalıcı ve güvenli şekilde saklandığı blokzinciri düğümü yer almaktadır. Bu dört düğüm arasındaki etkileşimler, modelin güvenlik ve performans özelliklerinin temelini oluşturmaktadır.

5.1. Güvenlik Analizi

Güvenlik analizi kapsamında, model; bilgi güvenliğinin temel ilkeleri olan gizlilik, kimlik doğrulama-bütünlük ve erişilebilirlik ölçütleri doğrultusunda değerlendirilecektir.

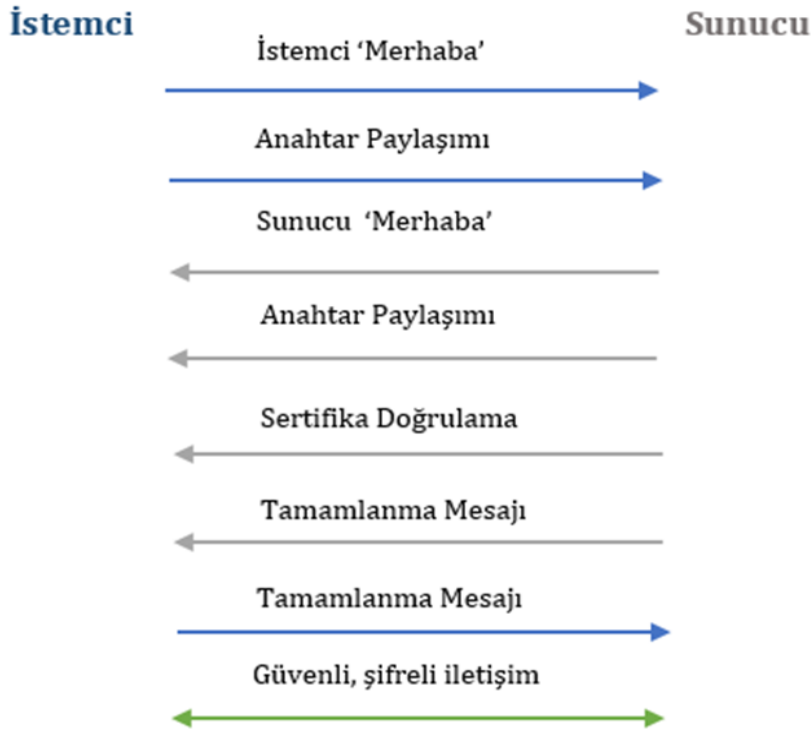
5.1.1. Önerilen modelde gizlilik analizi

Önerilen modelin gizlilik analizi yapılırken, model için sıkça karşılaşılabilecek üç tür saldırı ele alınacaktır. Bunlardan ilki, saldırganın iki taraf arasındaki iletişimi gizlice dinlediği, değiştirdiği hatta yönlendirdiği Araya Girme Saldırısı (Man in The Middle/MITM) dır. Araya girme saldırısı, şifreli iletişimin zayıf olduğu veya güvenli olmayan ağlarda yaygındır [55]. Bu tür saldırılarda genellikle aşağıdaki saldırı desenleri meydana gelmektedir [56]:

- Kesinti: A kişisinden B kişisine gönderilen verinin B kişisine ulaşmamasıdır.
- Dinleme: Saldırgan C'nin, A'nın B'ye gönderdiği veriyi ele geçirmesidir.
- Değiştirme: Saldırgan C'nin, A'nın B'ye gönderdiği veriyi ele geçirmesi ve içeriğini değiştirmesidir.
- Sahtecilik: Saldırgan C'nin, A'nın kimliğini kullanarak B'ye başarılı bir şekilde veri göndermesidir.

Araya girme saldırısında A ile B'nin iletişimi sağladığı gerçek kanal saldırgan tarafından ortadan kaldırılır ve bunun yerine sahte bir kanal oluşturur. Salırgan, bu sahte kanal üzerindeki verileri dinleyebilir, deęiştirebilir ve hatta A ya da B gibi davranarak kimlik hırsızlığı gerçekleştirebilir. Salırganın izlediğı yönteme göre araya girme saldırısı; pasif veya aktif olarak sınıflandırılmaktadır. Pasif saldırılarda saldırgan veriyi ele geçirir ve içeriğini deęiştirmez ancak aktif saldırılarda verinin içeriğı de deęiştirilir. Sonuç olarak, araya girme saldırılarında yukarıda bahsedilen tüm saldırı desenleri meydana gelebilmektedir. Bu nedenle, böyle bir saldırıya karşı ağı ve hassas verileri korumak büyük önem taşımaktadır [55].

Araya girme saldırılarına karşı alınabilecek en temel ve etkili önlemlerden biri, ağ iletişiminde TLS (Aktarım Katmanı Güvenliğı) protokolünün kullanılmasıdır. Günümüzde sıkça kullanılan ve TLS 0, TLS 1 ile TLS 1.2 sürümlerine kıyasla oldukça güvenli olan TLS 1.3 sürümü, Diffie-Hellman algoritmasını zorunlu tutmakta ve bu algoritma için türetilen anahtarlar da HKDF (HMAC-based Extract-and-Expand Key Derivation Function/HMAC Tabanlı Ayıklama ve Geniřletme Anahtar Türetme Fonksiyonu) fonksiyonunu kullanmaktadır. Bu fonksiyon ile anahtarların bütünlüğü sağlanmaktadır [57]. TLS 1.3 sürümü iki aşamadan oluşmaktadır. Bu sürüm; istemci ve sunucunun güvenli olmayan bir kanal üzerinden kimlik doęrulaması yaptığı ve simetrik anahtarların hesaplandığı el sıkışma (handshake) aşaması ile bu anahtarların kimlik doęrulamalı şifreleme amacıyla kullanıldığı güvenli iletişim aşamasından meydana gelmektedir [58]. Bu aşamalar, aşağıdaki Şekil 5.1'de görsel olarak sunulmaktadır:

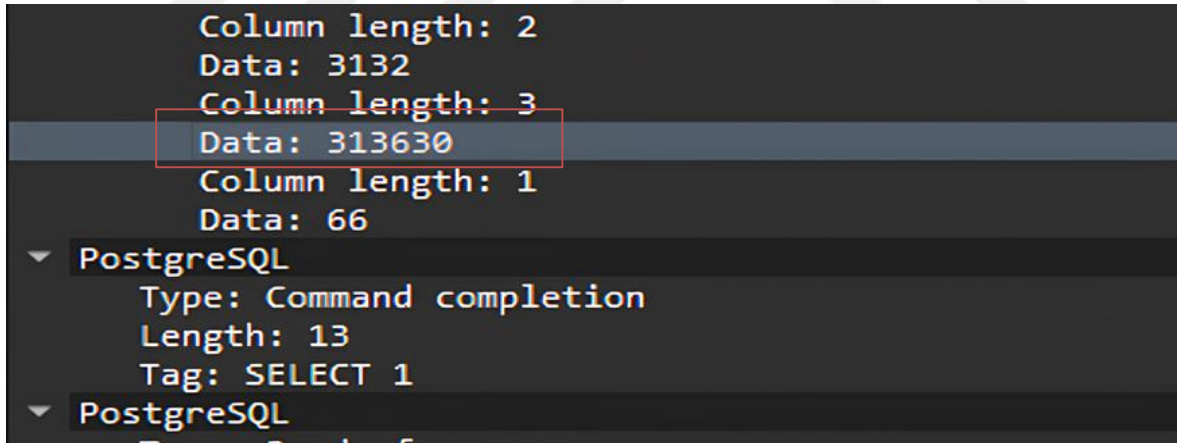


Şekil 5.1. TLS 1.3. el sıkışma aşamaları

TLS 1.3'te istemci ve sunucu, iletişim sırasında tam bir el sıkışma gerçekleştirebilmekte veya önceki bir oturumu “oturum devam ettirme” (session resumption) yoluyla sürdürebilmektedir. İlk yöntem ile Diffie-Hellman algoritması kullanılarak mükemmel ileri gizliliğin (perfect forward secrecy) sağlanması mümkün olmaktadır. Mükemmel ileri gizlilik sayesinde, oturumun şifreleme anahtarları ele geçirilse bile bu anahtarlarla geçmiş oturumların içeriği çözülemez [59]. Ancak “oturum devam ettirme” yöntemiyle bir önceki oturum devam ettirilirse, bu durumda mükemmel ileri gizlilik sağlanamayabilmektedir. Çünkü bir istemci, bir TLS sunucusuna bağlandığında, Diffie-Hellman algoritması gereği rastgele bir ön anahtar oluşturmakta ve bunu sunucunun açık anahtarıyla şifreleyip sunucuya göndermektedir. Ön anahtar, her iki tarafın ortak bir oturum anahtarı üretmek için kullanacağı gizli bir anahtardır. Ön anahtarı yalnızca sunucunun özel anahtarına sahip biri çözebilmektedir. Ancak sunucunun özel anahtarı her zaman düşünüldüğü kadar güvenli şekilde saklanamayabilir. Sunucunun özel anahtarı, kötü yapılandırmalar, yetersiz erişim kontrolleri, donanım açıkları, yazılım zafiyetleri veya kötü niyetli saldırılar nedeniyle ele geçirilebilir ve bu da mükemmel ileri gizliliğin sağlanmasını tehlikeye atmaktadır. Bu nedenle, “oturum devam ettirme” yönteminde mükemmel ileri gizlilik sağlanamayabilmektedir [58].

Bu bağlamda, TLS'in Diffie-Hellman gibi güçlü bir anahtar değişim algoritmasını kullanmasının yanı sıra; protokolün zayıf şekilde yapılandırılması, sertifika yönetimi hataları gibi durumlarda ve bazı senaryolarda mükemmel ileri gizliliği sağlayamaması nedeniyle, hassas veriler üzerinde tam güvenlik sağlanamayabilmektedir. Ayrıca vurgulamak gerekir ki, veri TLS sonlandırmasının yapıldığı noktada açık hale gelmektedir. Yani, TLS yalnızca verinin taşındığı alanı güvenli hale getirir; verinin kendisini şifrelemez. Zayıf yapılandırılmış bir ağ ortamında verilerin şifreleme mekanizmaları kullanılmadan iletilmesi, araya girme saldırılarına karşı ciddi bir güvenlik açığı oluşturmaktadır. Bu tür bir durumda, iletilen verilerin kötü niyetli kişilerce ele geçirilmesi oldukça olasıdır. Yapılan uygulamalı bir çalışmada, yerel ağ üzerinde gerçekleştirilen trafik analizi sonucunda, şifrelenmeden iletilen verilerin üçüncü taraflarca kolaylıkla görüntülenebildiği tespit edilmiştir.

Bu kapsamda, Wireshark adlı ağ analiz aracı kullanılarak gerçekleştirilen dinleme işlemi sonucunda, hedef sistemden şifresiz olarak iletilen veriler açık biçimde gözlemlenmiştir. İlgili duruma ilişkin ağ trafiği çıktısı aşağıdaki Şekil 5.2'de sunulmaktadır:

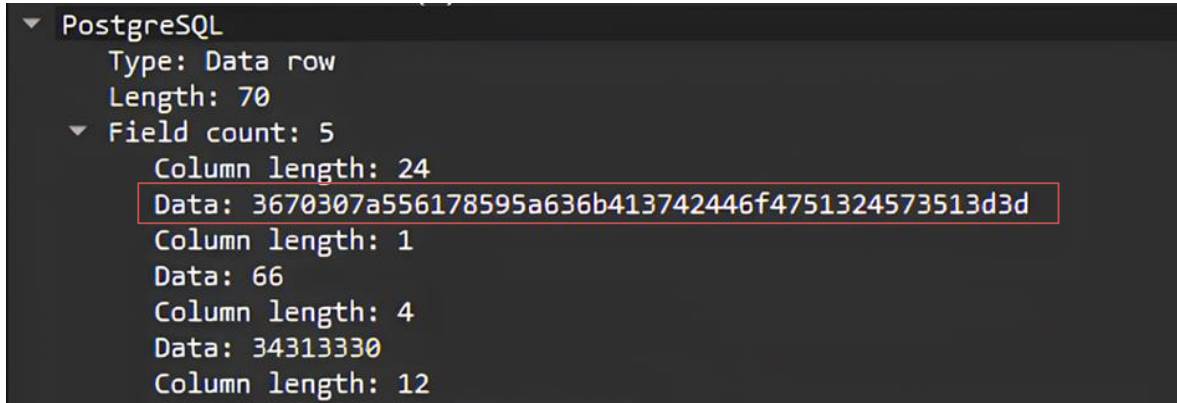


Şekil 5.2. Wireshark ile elde edilen şifresiz ağ trafiği çıktısı

Şekildeki ifadelerle bakıldığında, 'Data' alanının hastaya ait kan şekeri değerini temsil ettiği görülmektedir. Onaltılık sistemdeki 313630 ifadesi ASCII karakter karşılığına çevrildiğinde sırasıyla 31 → '1', 36 → '6' ve 30 → '0' değerlerini almaktadır. Bu dönüşüm sonucunda, 313630 hexadecimal ifadesi ASCII karakterlerle "160" değerini ifade etmektedir. Dolayısıyla, hasta verisinin 160 olarak elde edildiği anlaşılmaktadır.

Aynı ağda veri şifrelenerek iletildiğinde ise Şekil 5.3'te deki gibi verinin şifrelenmiş ve

çözülemez bir onaltılık değerde olduğu görülmektedir:



Şekil 5.3. Wireshark ile elde edilen şifreli ağ trafiği çıktısı

Günümüzde birçok iletişim kanalında kullanılan uçtan uca şifrelemenin sağlanabilmesi için hem verinin hem de verinin taşındığı alanın şifreli olması gerekmektedir [60]. Önerilen modelde ağ geçidi yapısı ve kenar bilişim yapısı arasında olan iletişimde hem TLS 1.3 kullanılmakta hem de taşınan hassas veri güçlü bir şekilde şifrelenmektedir. Şifreleme ve bütünlük algoritması olarak 5.2. Performans Analizi başlığı altında değerlendirilen ve modelde en iyi performans gösteren ChaCha20-Poly1305 ikilisi kullanılmakta ayrıca anahtar değişimi için de Ephemeral Diffie-Hellman (EDH) veya daha yaygın ismiyle Ephemeral Elliptic Curve Diffie-Hellman (ECDHE) anahtar değişim algoritması kullanılmaktadır. ChaCha20-Poly1305 şifreleme ve bütünlük algoritmasıyla şifre güvenli ve bütünlüğü bozulmadan ağda taşınmaktadır. ECDHE algoritması ise her veri için geçici anahtar üretimini sağlamaktadır. Her veri için geçici anahtarın üretimi mükemmel ileri gizliliği sağlamaktadır.

Bu başlık altında ele alınabilecek diğer saldırı türü düğüm ele geçirme saldırısıdır. Düğüm ele geçirme saldırısı; pasif, aktif ve fiziksel saldırı olmak üzere üç yöntemle gerçekleştirilebilmektedir. Saldırganın saldırıyı başlatabilmesi için, kablosuz sensör ağını dinlemesi gerekmektedir. Pasif dinleme yoluyla ağ üzerindeki mesaj alışverişlerini takip edebilir. Mesaj içerikleri şifreli olsa dahi, saldırgan ağın işleyişi ve durumu hakkında çıkarımlarda bulunabilir. Pasif dinlemeye ek olarak, aktif yöntemle saldırgan ağ protokollerine katılarak ağdan bilgi sızdırabilir veya ağa kötü niyetli veri enjekte edebilir. Aktif ve pasif yöntemlerle ağın durumu ve işleyişi hakkında yeterli bilgiye ulaşan saldırgan, fiziksel müdahale yoluyla ağdaki düğümleri ele geçirebilir. Bu süreçte elde

edilen bilgiler sayesinde, saldırgan hangi düğümleri ele geçireceği konusunda daha bilinçli kararlar verebilir [61].

Önerilen modelde, söz konusu saldırı açısından değerlendirilmesi gereken dört temel düğüm bulunmaktadır. Bu düğümler; verinin toplandığı IoT sensörlerinden oluşan uç düğüm, verinin taşındığı ve şifrelendiği ağ geçidi düğümü, verinin çözülerek analiz edildiği ve blokzincirine iletilindiği ara katman (kenar bilişim) düğümü ile verinin kalıcı ve güvenli bir biçimde saklandığı blokzinciri düğümüdür. Bu yapı, saldırılara karşı her bir düğümün ayrı ayrı güvenlik önlemleriyle değerlendirilmesini gerekli kılmaktadır.

Modelin fiziksel saldırılara karşı gizliliğinin korunması

Düğümlerin fiziksel olarak ele geçirilmesi açısından bir değerlendirme yapıldığında, öncelikle verinin toplandığı IoT sensörlerinden oluşan uç düğüm ele alınmalıdır. Uç düğüm, güç, batarya ve hafıza gibi kaynaklar açısından kısıtlı olduğundan, bu düğümlerde daha hafif ve gömülü güvenlik çözümlerinin tercih edilmesi gerekmektedir. Bu nedenle, donanım güvenlik modülleri doğrudan bu düğümlere uygulanamaz [62]. Önerilen modelde, uç düğüm için Güvenilir Platform Modülü (Trusted Platform Module/TPM) kullanılmaktadır. Güvenilir Platform Modülü, bilgisayarların güvenliğini donanım düzeyinde sağlamak amacıyla kullanılan güvenli bir yongadır [63]. Diğer üç düğüm açısından bir değerlendirme yapıldığında, bu düğümlerin fiziksel olarak ele geçirilmesi, düğümlerde depolanan anahtarların da ele geçirilebilmesi anlamına gelmektedir. Bu durum, hassas verilerin açığa çıkmasına neden olabilmektedir. Söz konusu düğümleri fiziksel saldırılardan korumak için ortak olarak alınabilecek bir önlem bulunmaktadır. Bu önlem, Donanım Güvenlik Modülleri'nin (Hardware Security Module - HSM) kullanımıdır. Bu modüllerin kullanımıyla yazılım tabanlı güvenlik önlemlerinin zayıflıkları telafi edilebilmektedir. Ayrıca yazılım tabanlı bir önlem olarak, bu modüller önerilen model için kritik öneme sahip anahtarların kötü niyetli yazılımlardan korunaklı bir şekilde üretilmesini, depolanmasını ve saklanmasını sağlar. Modülün kullanımı sayesinde, donanımların kurcalanmasına yönelik saldırıların önlenmesi amacıyla etkili fiziksel güvenlik önlemleri de alınmaktadır [64].

Modelin aktif ve pasif dinleme saldırılarına karşı gizliliğinin korunması

Düğümelerin fiziksel olarak korunmasının yanı sıra, aktif ve pasif dinlemelere karşı korunması da büyük önem taşımaktadır. Dinleme sonucunda şifreleme için kullanılan anahtarlar ele geçirilebilir. Bu nedenle, anahtarların güvenli bir biçimde yönetilmesi oldukça önemlidir. Önerilen modelde, anahtarların değişimi için Geçici Diffie-Hellman (Ephemeral Diffie-Hellman – EDH) ya da daha yaygın adıyla Geçici Eliptik Eğri Diffie-Hellman (Ephemeral Elliptic Curve Diffie-Hellman – ECDHE) anahtar değişim algoritması kullanılmaktadır. Her veri için yeni bir anahtar kullanılması sayesinde, mükemmel ileri gizlilik sağlanmakta ve anahtar yönetimi başarıyla gerçekleştirilmektedir. Ayrıca, ağın TLS 1.3 protokolü ile korunması ve verilerin şifreleme algoritması kullanılarak şifrenmesi ve bütünlüğünün sağlanması da bu saldırıya karşı alınan önlemlerden biridir. Böylece saldırgan, aktif yöntemle ağdan veri sızdıramaz; sızdırsa dahi veri şifreli olduğu için içeriği görüntüleyemez. Öte yandan, önerilen modelde ağ geçidi düğümü üzerinde gerçekleştirilen anomali tespiti ve izleme mekanizması, aktif yöntemle ağa kötü niyetli veri enjekte edilmesinin önüne geçilmesi amacıyla alınan bir diğer önlemdir.

Önerilen modelde, düğüm ele geçirme saldırısında en kritik hedeflerden ve tek başına değerlendirilmesi gereken düğümlerden biri blokzinciri düğümüdür. Bunun temel nedeni, bu düğümlerin ağdaki veri bütünlüğü, konsensüs süreci ve işlem güvenliği açısından merkezi bir rol oynamasıdır. Ele geçirilen bir blokzinciri düğümü, sahte veriler yayabilmekte, konsensüs mekanizmasını manipüle edebilmekte ve sistemin işlem güvenliğini zedeleyebilmektedir. Ayrıca bu tür düğümler, diğer düğümler için referans noktası olarak işlev gördüğünden, ağın geneline zarar verme potansiyeli taşımaktadır. Bu nedenle, blokzinciri düğümlerinin güvenliğinin sağlanması, dağıtık sistemlerde siber güvenlik açısından öncelikli olarak ele alınmalıdır.

Önerilen modelde bu tür saldırılara karşı Hyperledger Fabric ağı üzerinde Raft konsensüs mekanizması kullanılmaktadır. Raft konsensüs mekanizmasını izleyen düğümler; lider, takipçi ya da aday rolünde olabilmektedirler. Bu doğrultuda başlangıçta bir seçim yapılmakta ve lider düğüm belirlenmektedir. Diğer tüm düğümler takipçi konumundadır. Ancak, bir takipçi belirli bir süre boyunca liderden yanıt almazsa aday konumuna geçmektedir. Aday, ardından diğer düğümlerden oy istemekte ve diğer düğümler bu oy

talebine yanıt vermektedir. Eğer aday, düğümlerin çoğunluğundan oy alırsa lider olmaktadır. Bu işleme "lider seçimi" adı verilmektedir. Raft algoritmasının normal işleyişinde sistemde yalnızca bir lider bulunmakta ve diğer tüm düğümler takipçi rolünde kalmaktadır. Lider, tüm takipçilere periyodik olarak kalp atışları (heartbeat) göndererek otoritesini sürdürmekte ve tüm işlemler lider aracılığıyla gerçekleştirilmektedir. Her işlem, düğümün defterine bir kayıt olarak eklenmektedir. Özellikle, lider aldığı yeni işlemi önce takipçilere çoğaltmaktadır. Bu noktada kayıt henüz blokzincirine eklenmemektedir. Lider, kaydın çoğunluk tarafından onaylandığına dair geri bildirim aldıktan sonra, takipçilerine bu girişin onaylandığını bildirmektedir. Bu işleme "defter çoğaltma" adı verilmektedir. Raft algoritması, sistemdeki düğümlerin yarıdan fazlası sağlıklı kaldığı sürece konsensüs mekanizmasını sürdürmeye devam etmektedir. Bu düğümlerden lider olan saldırıya uğrarsa ve bunun sonucunda belirli bir süre yanıt vermezse, yeni bir lider seçilmektedir. Takipçi olan düğümler ele geçirilirse, bu düğümlerin tek başına karar yetkisi olmadığından saldırgan, ağa yeni blok ekleyemez ya da ağa zarar veremez ancak, takipçi düğümlerin yarısından fazlası ele geçirilirse blokzinciri işlem alımını durdurmakta ve blokzinciri düğümü ulaşılamaz hale gelmektedir. Ulaşılamaz blokzinciri üzerinde saldırgan herhangi bir işlem gerçekleştirememektedir. Bu bilgiler doğrultusunda, Raft konsensüs mekanizmasının çökme (hata) durumlarına karşı dayanıklı olduğu söylenebilmektedir [65].

Bu başlık altında ele alınması gereken son saldırı türü ise izinsiz dinleme (eavesdropping) saldırısıdır. Bu saldırı aktif olabileceği gibi, izinsiz dinleme saldırısında genellikle saldırgan pasiftir ve yalnızca ağı gizlice dinlemektedir [66]. Bu durumda, ağdaki verinin şifrelenmemiş olması, verileri bu saldırıya daha açık hâle getirir. Önerilen modelde ise ağda veri düz metin olarak iletilmemektedir. ChaCha20-Poly1305 şifreleme ve bütünlük algoritmalarıyla şifrelenen veri, kenar bilişim yapısında analiz edildikten sonra etiketlenir ve bu analiz sonucunda veriye ait bir etiket oluşturulmaktadır. Sadece bu etiket, blokzinciri ağına kalıcı olarak kaydedilmektedir. Böylece, blokzinciri üzerinde ham verinin kendisi yer almadığı için, olası bir izinsiz erişim durumunda dahi hassas verilere ulaşılamamaktadır. Blokzinciri ağı, yalnızca yetkilendirilmiş kullanıcıların kimlik doğrulaması yaparak erişebileceği bir web arayüzü üzerinden görüntülenmektedir. Ayrıca, sistemde kullanılan akıllı sözleşmeler, kullanıcıların veriye yalnızca okuma (görüntüleme) yetkisine sahip olmasını sağlamaktadır. Bu yapı sayesinde, kullanıcı yetkili olsa bile veri etiketini değiştirmesi mümkün değildir; bu durum, sistemin güvenilirliğini ve veri bütünlüğünü önemli ölçüde artırmaktadır.

5.1.2. Önerilen modelde kimlik doğrulama ve bütünlük analizi

Bu başlık altında, önerilen modelin Değiştirme Saldırısı (Modification Attack), Sahtecilik Saldırısı (Fabrication Attack), Yeniden Oynatma Saldırısı (Replay Attack), Yetkisiz Erişim (Unauthorized Access) ve Çoklu Sahte Kimlik Saldırıları (Sybil Attack)'na karşı güvenilirliği analiz edilecektir.

Değiştirme saldırısı

Değiştirme saldırısı; bir saldırganın, iki taraf arasında iletilen veriyi değiştirerek veya manipüle ederek sistemi yanıltmaya çalıştığı bir siber saldırı türüdür. Bu saldırı türü genellikle şifrelenmemiş veriler üzerinde gerçekleştirilir. Ayrıca, verinin iletildiği ağ güvenli değilse, bu saldırının gerçekleşme ihtimali artar. Bu saldırının temel amacı, verinin bütünlüğünü bozmaktır [67]. Önerilen modelde verinin hassas sağlık verisi olması, verinin bütünlüğünün korunması açısından önem arz etmektedir. Bu doğrultuda, önerilen modelin bu saldırıya karşı dayanıklı olması beklenmektedir. Önerilen modelde uç düğüm, ara geçit düğümü ve kenar bilişim düğümleri arasında ağda veri; ChaCha20-Poly1305 şifreleme ve bütünlük algoritmaları kullanılarak şifrelenmekte ve bütünlüğü korunmaktadır. Ayrıca, ağda TLS 1.3 protokolünün kullanılması da ağın bu tür saldırılara karşı korunmasını sağlamaktadır. Blokzinciri düğümünde de verinin bütünlüğü oldukça büyük önem taşımaktadır. Hyperledger Fabric ağı üzerinde, metotları özelleştirilerek Go diliyle yazılmış olan akıllı sözleşme metotları; yalnızca sistemde okuma ve ekleme işlemlerine izin verecek şekilde özelleştirilmiştir. Bu sebeple akıllı sözleşme silme ve güncelleme metotları içermediğinden veri etiketi üzerinde güncellemeye ve silmeye izin verilmemektedir. Bu durumda, veri etiketi blokzinciri üzerinde değiştirilemez hale gelmektedir. Ayrıca, veri etiketi blokzincirine kaydedilirken; veri etiketinin sahibine ait bilgiler ve verinin kaydedildiği zamana ilişkin zaman damgası ile birlikte kaydedilmektedir. Blokzincirinin değiştirilemez doğası gereği kaydedilen veriler geçmişe dönük olarak değiştirilememektedir [69]. Bu özellik, zaman damgası ve veri sahibi bilgilerinin de değiştirilememesini garanti etmektedir. Bunun yanında, blokzinciri üzerindeki verilere yalnızca veri sahibi ve veri sahibinin izin verdiği uzmanlar, bir web arayüzü üzerinden başarılı bir kimlik doğrulama ile erişebilmektedir.

Sahtecilik saldırısı

Ele alınması gereken saldırılardan birisi de sahtecilik saldırısıdır. Sahtecilik saldırısı, saldırganın yetkisiz bir şekilde ağa veri eklemesi veya mevcut veri üzerinde değişiklik yapmasıdır [70]. Önerilen modelde, sahte bir verinin ağa enjekte edilmesi ya da gerçek veriler üzerinde değişiklik yapılması veri sahibi hasta için hayati sonuçlara yol açabilmektedir [71]. Bu nedenle, modelin bu tür saldırılara karşı dayanıklı olması gerekmektedir. Önerilen modelde, sahtecilik saldırılarına karşı alınan önlemlerden ilki, ağ geçidi düğümünde gerçekleştirilen anomali tespittir. Anomali tespiti sayesinde, sisteme gönderilen verilerin doğruluğu ve tutarlılığı sürekli olarak izlenmekte, şüpheli veriler ise blokzinciri ağına iletilmeden önce filtrelenmektedir. Ayrıca, modelde kullanılan ChaCha20 ile veriler şifrelenmekte, mesaj bütünlüğünü sağlamak amacıyla kullanılan Poly1305 ile de veri bütünlüğü koruma altına alınmaktadır. Blokzinciri üzerinde etkin olan akıllı sözleşmeler, yalnızca yetkili kullanıcıların verilere erişmesine izin verecek şekilde tasarlanmıştır. Bu yetkili kullanıcılar verilere erişebilseler bile, veriler üzerinde herhangi bir güncelleme, değiştirme veya silme işlemi gerçekleştiremezler. Böylece, yalnızca okuma yetkisi tanınarak verilerin bütünlüğü korunmaktadır. Veriler üzerinde yapılan her işlem, blokzincirinde zaman damgalarıyla birlikte kaydedilmekte olup bu durum, blokzincirin inkâr edilemezlik ve değiştirilemezlik ilkeleri doğrultusunda gerçekleştirilmektedir [72].

Yeniden oynatma saldırısı

Yeniden oynatma saldırısı, yetkisiz bir kullanıcının ağ trafiğini yakalayarak, kendisini orijinal gönderici gibi göstererek alıcıyla iletişim kurması durumunda gerçekleşmektedir. Saldırgan, sistemin güvenliğini kazanmak amacıyla genellikle alıcı tarafından daha önce alınmış olan bir paketi yeniden göndererek bu saldırıyı gerçekleştirmektedir [73]. Önerilen modelde bu saldırıya karşı alınan önlemlerden ilki, her veri için farklı anahtarlar kullanılmasıdır. Böylelikle her veri ile ilişkilendirilen anahtarlar farklıdır ve sadece bir defa kullanılır. Alınan bir diğer önlem ise verilere ağ geçidi düğümünde şifrelendikten sonra bir tek kullanımlık sayı (nonce) değeri eklenmesidir. Bu tek kullanımlık sayı değeri her veri için farklılaşır ve aynı tek kullanımlık sayı değerine sahip veriler tespit edilirse iletilen veri alıcı tarafından engellenir. Önerilen modelde alınan diğer önlem, blokzincirine verinin kaydedilirken zaman damgasıyla birlikte kaydedilmesidir. Bu, saldırganın eski

verilere dayanarak yeniden oynatma yapmasını engellenmesine olanak sağlamaktadır [74].

Yetkisiz erişim

Yetkisiz erişim; bir ağa izinsiz erişim sağlanması ya da mevcut yetkinin amacı dışında kullanılması anlamına gelmektedir. Bu tanım doğrultusunda, hem yetkisiz kişilerin hassas verilere erişiminin engellenmesi hem de yetkili kişilerin yalnızca verilere erişim ve analiz sınırları içinde kalmaları gerekmektedir [75]. Önerilen modelde, yetkisiz erişimin engellenmesi ve mevcut yetkinin amacı dışında kullanılmaması amacıyla akıllı sözleşme yapısı kullanılmaktadır. Akıllı sözleşme üzerinde tanımlı fonksiyonlar aracılığıyla yalnızca yetkili kişiler, bir web arayüzü yardımıyla verilerin etiket değerlerine erişebilmekte ve bu değerleri çizgi grafik üzerinden görüntüleyebilmektedir. Veri sahibi kullanıcılar yalnızca etiketleri görüntüleyebilirken, veri sahiplerinin yetkilendirdiği uzmanlar bu verileri analiz ederek değerlendirme mesajı gönderebilmekte veya acil durumlarda e-posta yoluyla bildirimde bulunabilmektedir. Verilerin bütünlüğünü ve gizliliğini korumak amacıyla bu işlemler dışında herhangi bir veri etiketleri üzerinde güncelleme ya da silme fonksiyonu tanımlanmamıştır. Ayrıca yetkisiz erişimin önlenmesi açısından ağ güvenliği de kritik öneme sahiptir. Bu kapsamda, ağ TLS 1.3 protokolünü kullanmakta; ChaCha20-Poly1305 şifreleme ve bütünlük algoritmalarıyla güvenlik seviyesi artırılmaktadır.

Çoklu sahte kimlik saldırısı

Çoklu sahte kimlik saldırısı, bir saldırganın bir ağ bileşeninin (örneğin bir düğüm, cihaz, kullanıcı ya da IP adresi) kimliğini taklit ederek sistemi manipüle etmeye çalıştığı bir saldırı türüdür. Bu tür saldırılar sistemin işleyişini bozmak, veri bütünlüğünü zayıflatmak veya karar alma süreçlerini manipüle etmek amacıyla gerçekleştirilmektedir [76]. Önerilen modelde kullanılan veri, hem hassas hem de bütünlüğünün bozulması durumunda hayati sonuçlara yol açabilecek nitelikte sağlık verileridir [77]. Bu nedenle, bu verilerin söz konusu saldırılara karşı korunması büyük önem taşımaktadır. Modelde yer alan tüm düğümler ve cihazlar, dijital sertifikalar aracılığıyla tanımlanmaktadır. Dijital sertifika, bir nesnenin (örneğin bir sensör, aktüatör veya kullanıcı) kimliğini, güvenilir bir üçüncü tarafın dijital imzasıyla bir açık anahtara bağlayan elektronik bir belgedir. Bu yapı sayesinde, dışarıdan sahte bir düğüm eklenmesinin önüne geçilmektedir. Ayrıca, ağ geçidi düğümünde kimlik doğrulama mekanizması bulunduğundan, bu noktaya da sahte bir

düğüm eklenmesi mümkün değildir. Uç düğüm ile kenar bilişim düğümü arasındaki iletişimde TLS 1.3 protokolü kullanıldığından, bu ağ segmentine sahte bir düğüm yerleştirilmesi engellenmektedir [78]. Son olarak, saldırgan kenar bilişim katmanında fiziksel olarak sahte bir düğüm eklese bile, bu düğüm özel anahtara sahip olmadığı için şifreli veriyi çözüp içeriğine erişemez. Çoklu sahte kimlik saldırısına karşı, blokzinciri düğümünde Raft konsensüs mekanizması kullanılmaktadır. Raft konsensüs mekanizmasında, ağdaki her düğüm önceden tanımlıdır; yani yeni bir düğüm ağa kendiliğinden katılamaz [65]. Hyperledger Fabric mimarisine özgü olarak, her bileşen (sıralayıcı, kullanıcı vs.) X.509 formatında dijital sertifikalarla tanımlanmakta [79] ve bu sertifikalar, Üyelik Hizmeti Sağlayıcısı (MSP) aracılığıyla doğrulanmaktadır. MSP ile yalnızca yetkili düğümler işlem yapabilmektedir [80]. Bunlarla birlikte blokzinciri düğümü de bu saldırıya karşı dayanıklı hale gelmektedir.

5.1.3. Önerilen modelde erişilebilirlik analizi

Bilgiye zamanında ve güvenilir şekilde erişim ve kullanım anlamına gelen erişilebilirlik [81] kavramı, önerilen model için büyük bir önem taşımaktadır; çünkü modelde işlenen sağlık verilerinin doğru zamanda ve güvenli bir şekilde ilgili sağlık personeline ulaştırılması, hastaların yaşamı açısından kritik bir rol oynamaktadır [77]. Önerilen modeldeki düğümlerin genellikle dağıtık bir yapıda olması, yalnızca tek bir bileşenin arızalanmasının tüm sistemi işlevsiz hâle getirmesine neden olan tek hata noktası (single point of failure) riskini ortadan kaldırmayı hedeflemektedir [65]. Ele alınması gereken düğümlerden biri olan ağ geçidi düğümünde herhangi bir ağ kesintisi durumunda, uç düğümden gelen veri şifrelenerek tampona alınmaktadır. Ağ bağlantısı yeniden sağlandığında ise bu veri, kenar bilişim düğümüne iletilerek modelin işleyişi devam ettirilmektedir. Ayrıca düğüm üzerinde yapılan anomali tespitiyle, sistemi kesintiye uğratabilecek DDoS saldırılarına karşı sistem güvence altına alınmaktadır. Ele alınması gereken diğer düğüm ise kenar bilişim düğümüdür. Merkeziyetçi bir bulut yapısı yerine, daha az bant genişliği gereksinimi olan ve verilerin daha yakın bir noktada işlenmesini sağlayan kenar bilişim düğümüyle, gecikmeler azaltılmakta ve dağıtık yapısıyla tek hata noktası riski ortadan kaldırılmaktadır [44]. Son olarak, blokzinciri düğümünün merkeziyetçi yapıda olmaması sebebiyle, tek hata noktası riski ortadan kaldırılmaktadır [82]. Çünkü blokzinciri ağına bağlı olan tüm düğümler, erişim kontrol listesini içeren akıllı sözleşme verilerini depolamaktadır. Ayrıca sağlık verilerinin saklandığı defter (ledger)

yapısının kopyası tüm düğümlerde bulunmaktadır [49]. Blokzinciri ağında bir düğümün erişilebilir olmaması durumunda, ağdaki diğer düğümler işlemleri gerçekleştirebilmektedir. Bu, sistemin güvenliğini, sürekliliğini arttırmakta ve tek hata noktası olasılığını düşürmektedir [83].

5.2. Performans Analizi

Performans analizi kapsamında, öncelikle modelin uygulanmasında kullanılan teknolojiler, kabul edilen teknik varsayımlar ve karşılaşılan uygulama kısıtları ele alınmıştır. Devamında, modelde kullanılacak en uygun şifreleme algoritmasını belirlemek amacıyla üç farklı algoritma, Apache JMeter aracılığıyla performans testlerine tabi tutulmuştur. Her algoritma; işlem süresi, tepki süresi gibi ölçütler açısından ayrı ayrı değerlendirilmiş ve yapılan karşılaştırmalar sonucunda, sistemin güvenlik gereksinimlerini karşılayacak şekilde en hızlı ve etkili çalışan algoritma seçilmiştir. Böylece geliştirilen model, hem güvenlik hem de performans açısından dengeli ve verimli bir yapıda tasarlanmıştır.

5.2.1. Kullanılan teknolojiler, varsayımlar ve kısıtlar

Önerilen model, RESTful prensipleri temel alınarak geliştirilmiştir. Uç düğüm ve kenar bilişim düğümü, IntelliJ IDEA ortamında Spring Boot çerçevesi kullanılarak deneysel olarak yapılandırılmıştır. Modelde, uç düğüm verileri toplamakta, ağ geçidi düğümü veriyi şifrelemekte, kenar bilişim düğümü ise bu verileri deşifreleyerek analiz ederek analiz sonuçlarına bağlı olarak akıllı sözleşmeleri tetiklemektedir. Şifreleme işlemleri için Blowfish-CBC, AES-GCM ve ChaCha20-Poly1305 algoritmaları; anahtar değişimi için ise Diffie-Hellman algoritması kullanılmıştır. Her bir şifreleme algoritması, 100, 200 ve 500 kullanıcı senaryolarıyla performans testlerine tabi tutulmuştur ve testler 10 defa tekrarlanarak ortalama değerler alınmıştır.

Blokzinciri yapısı, Hyperledger Fabric mimarisi üzerine inşa edilmiş ve Fabric Samples açık kaynak kodlu test ağı kullanılarak yapılandırılmıştır. Ağ, Docker konteynerleri ile oluşturulmuş; yönetimi ise Docker PS komutları aracılığıyla gerçekleştirilmiştir. Akıllı sözleşme, Go programlama diliyle yazılarak ağa dağıtılmıştır. Web uygulaması bileşenlerinde ise React ve Spring Boot birlikte kullanılmıştır. Gerçek glikoz değerleri yerine, 70 ile 250 arasında rastgele üretilen sensör verileri simülasyon amaçlı olarak

Postman aracılığıyla sisteme gönderilmiştir.

Modelin uygulaması gerçekleştirilmiş olup, model çeşitli teknik varsayımlar doğrultusunda yapılandırılmıştır. Teknik varsayımlar arasında, modeldeki tüm cihaz ve yapıların kimlik doğrulama süreçlerinde dijital sertifikalar kullandığı öngörülmektedir. Bu sayede yalnızca güvenilir bileşenlerin sisteme katılımı sağlanmakta ve yetkisiz erişimler engellenmektedir. Ayrıca, IoT sensörlerinden kenar bilişim katmanına veri iletiminin TLS 1.3 protokolüyle sağlandığı varsayılmış; böylece veri iletimi çift katmanlı bir güvenlik mekanizmasıyla korunmuştur. Donanım tabanlı güvenliği temin etmek adına, sistemde güvenli anahtar saklama ve kriptografik işlem gerçekleştirme işlevlerine sahip Donanım Güvenlik Modülleri'nin (HSM) kullanıldığı kabul edilmiştir. Bu teknik varsayımlar doğrultusunda modelin güvenlik ve bütünlük hedeflerine ulaşması sağlanmıştır.

Ayrıca önerilen model bazı sınırlamalara da sahiptir. Modelin uygulaması sınırlı bir test senaryosu ve belirli donanım konfigürasyonları çerçevesinde gerçekleştirilmiş; geniş ölçekli sistemlerde performans değerlendirmeleri ve stres testleri yapılmamıştır. Geliştirme ve değerlendirme süreçlerinde, hastalara ait gerçek kan şekeri verileri yerine, bu verilerin yerine geçebilecek şekilde simüle edilmiş veri setleri kullanılmıştır. Ayrıca, blokzinciri altyapısı Hyperledger Fabric test ağı üzerinde kurgulanmış ve akıllı sözleşmeler bu ortamda geliştirilmiştir. Bu durum, modelin üretim ortamındaki davranışını sınırlı ölçüde yansıtmaktadır. Tüm bu sınırlamalara rağmen, SağlıkZinciri modeli, sağlık verilerinin güvenli bir biçimde işlenmesi ve bütünlüğünün korunması konusunda gelecek çalışmalara sağlam bir temel teşkil etmektedir.

5.2.2. Önerilen modelde kullanılan şifreleme ve bütünlük algoritmalarının performans analizi

Blowfish-CBC, AES-GCM ve ChaCha20-Poly1305 şifreleme ve bütünlük algoritmaları; 100, 200 ve 500 kullanıcı için sırasıyla 100, 200 ve 500 aşamalı yükleme süresi (ramp-up) değerlendirilmiştir; ilgili sonuçlar aşağıdaki çizelgelerde sunulmaktadır:

Çizelge 5.1. Blowfish-CBC algoritmasının kullanıcı sayısına göre performans ölçümleri

Test Edilen Algoritma	Kullanıcı Sayısı	Aşamalı Yükleme Süresi (sn)	Minimum Yanıt Süresi(ms)	Maksimum Yanıt Süresi(ms)	Ortalama Yanıt Süresi(ms)
Blowfish-CBC	100	100	3	132	9
	200	200	2	15	5
	500	500	3	12	2

Çizelge 5.2. AES-GCM algoritmasının kullanıcı sayısına göre performans ölçümleri

Test Edilen Algoritma	Kullanıcı Sayısı	Aşamalı Yükleme Süresi (sn)	Minimum Yanıt Süresi(ms)	Maksimum Yanıt Süresi(ms)	Ortalama Yanıt Süresi(ms)
AES-GCM	100	100	3	187	9
	200	200	2	17	5
	500	500	2	18	5

Çizelge 5.3. ChaCha20-Poly1305 algoritmasının kullanıcı sayısına göre performans ölçümleri

Test Edilen Algoritma	Kullanıcı Sayısı	Aşamalı Yükleme Süresi (sn)	Minimum Yanıt Süresi(ms)	Maksimum Yanıt Süresi(ms)	Ortalama Yanıt Süresi(ms)
ChaCha20-Poly1305	100	100	3	148	8
	200	200	2	15	5
	500	500	2	12	3

Yapılan performans testlerine göre üç algoritmanın sayısal sonuçları birbirine çok yakın olmakla beraber ChaCha20-Poly1305 algoritmasının yüksek kullanıcı sayıları altında daha hızlı ve istikrarlı bir performans sergilediği görülmektedir. ChaCha20-Poly1305 algoritması özellikle maksimum yanıt süresi açısından daha düşük değerler göstererek yüksek yük altında dahi etkili bir işlem gerçekleştirmektedir. ChaCha20-Poly1305 algoritmasından sonra AES-GCM algoritması Blowfish-CBC'ye kıyasla daha istikrarlı bir performans sergilemektedir. Blowfish-CBC algoritması eski bir şifreleme algoritması olmasına rağmen düşük kullanıcı sayılarında ortalama bir performans sergilemektedir ancak yüksek yük altında gecikmeler yaşanmakta bu durumda sistemin verimliliğini olumsuz etkilemektedir. Literatüre bakıldığında nümerik bir sağlık verisine göre daha karmaşık olan ancak aynı kriptografik temellere dayanan görüntü şifrelemede bu

algoritmalar test edilmiş ve ChaCha20 şifreleme algoritmasının Blowfish ve AES şifreleme algoritmalarına göre daha hızlı olduğu belirtilmiştir [84]. Ayrıca diğer bir çalışmada ChaCha20-Poly1305 algoritmasının hızından dolayı genellikle mobil telefon donanımlarında ve Apple'ın M1 işlemcisinde tercih edilen bir seçenek olduğundan bahsedilmiştir [85]. Literatüre ve test sonuçlarına göre önerilen model için en performanslı ve uygun seçeneğin ChaCha20-Poly1305 algoritması görülmektedir. Bu nedenle önerilen modelde sağlık verilerini şifrelemek ve veri bütünlüğünü sağlamak için ChaCha20-Poly1305 algoritması kullanılmaktadır.

5.2.3. Postman kullanılarak akıllı sözleşme fonksiyonlarının işlevsellik ve güvenilirlik testi

Uygulanan akıllı sözleşme fonksiyonlarının işlevselliği ve güvenilirliği, Spring Boot çerçevesi kullanılarak geliştirilen RESTful API'lere yönelik HTTP istekleri gönderilerek kapsamlı bir şekilde test edilmiştir. Bu test sürecinde, Postman aracı etkin bir şekilde kullanılarak farklı senaryolarda API çağrıları gerçekleştirilmiş ve sistemin gerçek zamanlı yanıtları dikkatle incelenmiştir. Böylece, akıllı sözleşmenin temel fonksiyonlarının beklenen şekilde çalışıp çalışmadığı, hata durumlarına karşı verdiği tepkiler ve genel performansı değerlendirilmiştir.

Akıllı sözleşmede dört ana fonksiyon yer almakta olup, bu fonksiyonlar blokzinciri üzerindeki veri yönetiminin temel yapı taşlarını oluşturmaktadır. *DegerEkle()* fonksiyonu, deftere yeni etiketler eklemek için kullanılırken, *RaporEkle()* fonksiyonu blokzincirine yeni raporların eklenmesini sağlamaktadır. Bu fonksiyonlar, sisteme dinamik veri girişine olanak tanımakta ve verilerin blokzinciri üzerinde kalıcı, değiştirilemez bir şekilde kayıt altına alınmasını garanti etmektedir. Öte yandan, *TumDegerleriGetir()* fonksiyonu, sistemde daha önce eklenmiş olan tüm etiketleri topluca döndürürken; *TumRaporlariGetir()* fonksiyonu da blokzincirindeki kayıtlı tüm raporların listesini sağlamaktadır. Bu sayede kullanıcılar ve sistem bileşenleri, mevcut veri setine hızlı ve güvenilir bir şekilde erişebilmektedir.

Fonksiyonların işleyişiyle ilgili yapılan testlerde, *DegerEkle()* ve *RaporEkle()* fonksiyonlarına geçerli ve doğru biçimlendirilmiş veriler gönderildiğinde sistemin başarılı bir işlem gerçekleştirdiğine dair HTTP 200 OK yanıtı verdiği gözlemlenmiştir. Bu, işlemin sorunsuz tamamlandığını ve verilerin blokzincire başarıyla kaydedildiğini göstermektedir. Buna karşılık, eksik veya hatalı veri içeren isteklerde sistem HTTP 400 Bad Request yanıtı dönmekte, böylece giriş doğrulama mekanizmalarının etkin bir biçimde devrede olduğu ve hatalı verilerin sisteme zarar vermesinin önlendiği anlaşılmaktadır. Bu durum, uygulamanın güvenilirliği ve veri bütünlüğünün korunması açısından kritik bir öneme sahiptir. Son olarak, *TumDegerleriGetir()* ve *TumRaporlariGetir()* fonksiyonları kullanılarak yapılan çağrılar sonucunda, blokzincirinde saklanan etiketler ve raporların eksiksiz ve tutarlı biçimde elde edilmesi sağlanmıştır. Bu, akıllı sözleşme fonksiyonlarının yalnızca veri girişinde değil, veri sorgulamada da doğru ve güvenilir çalıştığını kanıtlamaktadır. Tüm bu test ve doğrulamalar ile geliştirilen akıllı sözleşme fonksiyonlarının blokzinciri ağı içerisinde güvenilir, tutarlı ve hatasız bir şekilde işlev gösterdiği ve sistemin genel stabilitesine katkı sağladığı sonucuna ulaşılmıştır.



6. SONUÇ VE ÖNERİLER

Cihaz sayısındaki hızlı artışla birlikte, özellikle sağlık alanında, sağlık bilgileri gibi hassas verilerin miktarı da önemli ölçüde artmıştır. Bu verilerin iletimi ve depolanması sırasında güvenliğinin, gizliliğinin ve bütünlüğünün sağlanması büyük önem taşımaktadır. Mevcut IoT modelleri çoğunlukla yüksek maliyetlere ve tek bir hata noktasına yol açabilen merkezi bir sunucuya bağımlıdır. Bu tür iletişimler, güvenli veri iletiminden yoksun olup işlemler açısından değiştirilebilir yapıdadır ve bu da onları saldırılara karşı savunmasız hâle getirmektedir.

Bu çalışmada, geleneksel IoT modellerine alternatif olarak sağlık cihazları işlemlerini ve iletişimini kolaylaştırmak amacıyla, SağlıkZinciri adlı blokzinciri ve kenar bilişim tabanlı bir IoT modeli önerilmektedir. Önerilen model, verilerin güvenliği, gizliliği ve bütünlüğünü sağlamak üzere tasarlanmış hibrit bir IoT ve kenar bilişim blokzinciri çerçevesidir. Bu modele blokzincirinin entegrasyonu ile işlemler değiştirilemez ve güvenli hâle getirilmiş, cihazlar arasındaki iletişim ise şifrelenmiştir. Modelin merkeziyetsiz yapısı sayesinde, IoT cihazlarının işlem gücü optimize edilmiştir. Kan şekeri düzeyi verileriyle gerçekleştirilen deneysel testler, modelin sağlık verilerinin güvenliği, gizliliği ve bütünlüğü açısından etkinliğini ortaya koymaktadır. Bu çalışmada, geliştirilen model öncelikli olarak gizlilik, kimlik doğrulama-bütünlük ve erişilebilirlik kriterleri çerçevesinde analiz edilmiştir. Her bir başlık altında, modele yönelik olası saldırı türlerine karşı ne tür güvenlik önlemleri içerdiği detaylı biçimde incelenmiş ve yapılan değerlendirmeler sonucunda modelin bu saldırılara karşı dayanıklı olduğu ortaya konmuştur. Ayrıca bu çalışmada, modelde en yüksek performansı sağlayan şifreleme algoritmasının belirlenebilmesi amacıyla Blowfish-CBC, AES-GCM ve ChaCha20-Poly1305 algoritmaları Apache JMeter aracılığıyla performans testlerine tabi tutulmuştur. Gerçekleştirilen testler sonucunda elde edilen bulgular ve bazı çalışmalar ışığında, model için en uygun ve en yüksek performanslı algoritma olarak ChaCha20-Poly1305 belirlenmiştir. Bu model, sağlık verilerinin güvenliği ve bütünlüğüne yönelik çözüm sunabilecek potansiyele sahiptir. Sonuç olarak, önerilen SağlıkZinciri modeli, sağlık alanındaki güvenli veri yönetimi sorunlarına yönelik umut verici çözümler sunmakta ve sağlık teknolojilerinde önemli bir boşluğu doldurmaktadır.

Gelecek alıřmalarda modelin sınırlılıkları ele alınabilir, daha doęru ve dinamik veri toplama amacıyla gerek dnya IoT sensrleri ve aę geitleri ile uygulanması geniřletilebilir; ayrıca performansın artırılması iin geliřmiř aę yapılandırılmaları entegre edilebilir. Buna ek olarak, leklenebilirlik ve gvenilirlik daha da geliřtirilebilir; yapay zekâ ve makine ęrenmesi gibi yeni teknolojiler, kestirimci analizler ve karar verme srelerinin iyileřtirilmesi amacıyla modele dâhil edilebilir. Saęlık sektrnn tesinde, bu model; mahremiyet, gvenlik ve veri btnlę gerektiren finans, kiřisel kimlik sistemleri ve akıllı řehirler gibi dięer alanlarda da uygulanabilir. Bu sektrler giderek daha gl ve gvenli veri ynetimi zmlerine ihtiya duymakta olup, modelin daha geniř uygulama alanları iin uygun bir seenek olmasını saęlamaktadır.



KAYNAKLAR

1. Talwana, J. C., Hua, H. J. (2016, December). *Smart world of Internet of Things (IoT) and its security concerns*. 2016 IEEE international conference on internet of things (iThings) and IEEE green computing and communications (GreenCom) and IEEE cyber, physical and social computing (CPSCom) and IEEE smart data (SmartData), IEEE, 240-245.
2. Singh, P., Elmi, Z., Meriga, V. K., Pasha, J., Dulebenets, M. A. (2022). Internet of Things for sustainable railway transportation: Past, present, and future. *Cleaner Logistics and Supply Chain*, 4, 100065.
3. Sarker, I. H., Khan, A. I., Abushark, Y. B., Alsolami, F. (2023). Internet of things (iot) security intelligence: a comprehensive overview, machine learning solutions and research directions. *Mobile Networks and Applications*, 28(1), 296-312.
4. Cheikhrouhou, O., Mershad, K., Jamil, F., Mahmud, R., Koubaa, A., Moosavi, S. R. (2023). A lightweight blockchain and fog-enabled secure remote patient monitoring system. *Internet of Things*, 22, 100691.
5. Baker, S. B., Xiang, W., Atkinson, I. (2017). Internet of things for smart healthcare: Technologies, challenges, and opportunities. *Ieee Access*, 5, 26521-26544.
6. Al-Kahtani, M. S., Khan, F., Taekeun, W. (2022). Application of internet of things and sensors in healthcare. *Sensors*, 22(15), 5738.
7. Ranjan, R., Ch, B. (2024). A comprehensive roadmap for transforming healthcare from hospital-centric to patient-centric through healthcare internet of things (IoT). *Engineered Science*, 30, 1175.
8. McGraw, D., Mandl, K. D. (2021). Privacy protections to encourage use of health-relevant digital data in a learning health system. *NPJ digital medicine*, 4(1), 2.
9. Rayan, R. A., Tsagkaris, C., Iryna, R. B. (2021). The Internet of things for healthcare: applications, selected cases and challenges. *Iot in Healthcare and Ambient Assisted Living*, 1-15.
10. Boopathi, S. (2023). Internet of Things-integrated remote patient monitoring system: healthcare application. *İçinde Dynamics of Swarm Intelligence Health analysis for the next generation*, IGI Global, 137–161.
11. Xiang, D., Cai, W. (2021). Privacy protection and secondary use of health data: strategies and methods. *BioMed Research International*, 2021(1), 6967166.
12. Nowrozy, R., Ahmed, K., Kayes, A. S. M., Wang, H., McIntosh, T. R. (2024). Privacy preservation of electronic health records in the modern era: A systematic survey. *ACM Computing Surveys*, 56(8), 1-37.
13. Dey, T., Jaiswal, S., Sunderkrishnan, S., Katre, N. (2017, December). *HealthSense: A medical use case of Internet of Things and blockchain*. 2017 International Conference on Intelligent Sustainable Systems (ICISS), IEEE, 486-491.

14. Kaur, H., Alam, M. A., Jameel, R., Mourya, A. K., Chang, V. (2018). A proposed solution and future direction for blockchain-based heterogeneous medicare data in cloud environment. *Journal of Medical Systems*, 42, 1-11.
15. Chakraborty, S., Aich, S., Kim, H. C. (2019, February). *A secure healthcare system design framework using blockchain technology*. 2019 21st International Conference on Advanced Communication Technology (ICACT), IEEE, 260-264.
16. Dilawar, N., Rizwan, M., Ahmad, F., Akram, S. (2019). Blockchain: Securing Internet of Medical Things (IoMT). *International Journal of Advanced Computer Science and Applications*, 10(1), 82-89.
17. Bhawiyuga, A., Wardhana, A., Amron, K., Kirana, A. P. (2019, December). *Platform for integrating Internet of Things based smart healthcare system and blockchain network*. 2019 6th NAFOSTED Conference on Information and Computer Science (NICS), IEEE, 55-60.
18. Dwivedi, A. D., Srivastava, G., Dhar, S., Singh, R. (2019). A decentralized privacy-preserving healthcare blockchain for IoT. *Sensors*, 19(2), 326.
19. Shahnaz, A., Qamar, U., Khalid, A. (2019). Using blockchain for electronic health records. *IEEE Access*, 7, 147782-147795.
20. Attia, O., Khoufi, I., Laouiti, A., Adjih, C. (2019, June). *An IoT-blockchain architecture based on Hyperledger framework for healthcare monitoring application*. NTMS 2019-10th IFIP International Conference on New Technologies, Mobility and Security, IEEE Computer Society, 1-5.
21. Satamraju, K. P. (2020). Proof of concept of scalable integration of Internet of Things and blockchain in healthcare. *Sensors*, 20(5), 1389.
22. Khatoon, A. (2020). A blockchain-based smart contract system for healthcare management. *Electronics*, 9(1), 94.
23. Ismail, L., Materwala, H., Sharaf, Y. (2020, October). *Blockhr—a blockchain-based healthcare records management framework: Performance evaluation and comparison with client/server architecture*. 2020 International Symposium on Networks, Computers and Communications (ISNCC), IEEE, 1-8.
24. Jamil, F., Ahmad, S., Iqbal, N., Kim, D. H. (2020). Towards a remote monitoring of patient vital signs based on IoT-based blockchain integrity management platforms in smart hospitals. *Sensors*, 20(8), 2195.
25. Banotra, A., Sharma, J. S., Gupta, S., Gupta, S. K., Rashid, M. (2021). Use of Blockchain and Internet of Things for securing data in healthcare systems. İçinde K. J. Giri, S. A. Parah, R. Bashir, K. Muhammad (Eds.), *Multimedia Security: Algorithms for Intelligent Systems*, Singapur: Springer.
26. Ngabo, D., Wang, D., Iwendi, C., Anajemba, J. H., Ajao, L. A., Biamba, C. (2021). Blockchain-based security mechanism for the medical data at fog computing architecture of Internet of Things. *Electronics*, 10(17), 2110.

27. Mayer, A. H., Rodrigues, V. F., da Costa, C. A., da Rosa Righi, R., Roehrs, A., Antunes, R. S. (2021). Fogchain: A fog computing architecture integrating blockchain and Internet of Things for personal health records. *IEEE Access*, 9, 122723–122737.
28. Bataineh, M. R., Mardini, W., Khamayseh, Y. M., Yassein, M. M. B. (2022). Novel and secure blockchain framework for health applications in IoT. *IEEE Access*, 10, 14914-14926.
29. Elvas, L. B., Serrão, C., Ferreira, J. C. (2023). Sharing health information using a blockchain. *Healthcare*, 11(2), 170.
30. Oladele, J. K., Ojugo, A. A., Odiakaose, C. C., Emordi, F. U., Abere, R. A., Nwozor, B., Ejeh, P. O., Geteloma, V. O. (2024). BEHedas: A blockchain electronic health data system for secure medical records exchange. *Journal of Computing Theories and Applications*, 1(3), 231–242.
31. Said, H. E., Al Barghuthi, N. B., Badi, S. M., Hashim, F., Girija, S. (2024, August). *Developing a decentralized Blockchain framework with Hyperledger and NFTs for secure and transparent patient health records*. International Conference on Innovations in Computing Research. Cham: Springer Nature Switzerland, 478-489.
32. Torğul, B., Şağbanşua, L., Balo, F. B. (2016). Internet of Things: A survey. *International Journal of Applied Mathematics Electronics and Computers, Special Issue-1*, 104-110.
33. Thilakarathne, N. N. (2020). Security and privacy issues in iot environment. *International Journal of Engineering and Management Research (IJEMR)*, 10(1), 26-29.
34. Yang, Y., Wu, L., Yin, G., Li, L., Zhao, H. (2017). A survey on security and privacy issues in Internet-of-Things. *IEEE Internet of things Journal*, 4(5), 1250-1258.
35. Vashi, S., Ram, J., Modi, J., Verma, S., Prakash, C. (2017, February). *Internet of Things (IoT): A vision, architectural elements, and security issues*. 2017 international conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC). IEEE. 492-496.
36. Ali, I., Sabir, S., Ullah, Z. (2019). Internet of things security, device authentication and access control: A review. *arXiv preprint arXiv:1901.07309*.
37. Abou-Nassar, E. M., Iliyasu, A. M., El-Kafrawy, P. M., Song, O. Y., Bashir, A. K., Abd El-Latif, A. A. (2020). DITrust chain: towards blockchain-based trust models for sustainable healthcare IoT systems. *IEEE access*, 8, 111223-111238.
38. Ramson, S. J., Vishnu, S., Shanmugam, M. (2020, March). *Applications of internet of things (iot)–an overview*. 2020 5th international conference on devices, circuits and systems (ICDCS), IEEE, 92-95.
39. Kang, M., Park, E., Cho, B. H., Lee, K. S. (2018). Recent patient health monitoring platforms incorporating internet of things-enabled smart devices. *International Neurourology Journal*, 22(Suppl 2), S76.

40. Shukla, S., Thakur, S., Hussain, S., Breslin, J. G., Jameel, S. M. (2021). Identification and authentication in healthcare internet-of-things using integrated fog computing based blockchain model. *Internet of Things*, 15, 100422.
41. Dai, H. N., Zheng, Z., Zhang, Y. (2019). Blockchain for Internet of Things: A survey. *IEEE internet of things journal*, 6(5), 8076-8094.
42. Dwivedi, A. D., Malina, L., Dzurenda, P., Srivastava, G. (2019, July). *Optimized blockchain model for internet of things based healthcare applications*. 2019 42nd international conference on telecommunications and signal processing (TSP). IEEE. 135-139.
43. Basu, S., Bardhan, A., Gupta, K., Saha, P., Pal, M., Bose, M., Sarkar, P. (2018, January). *Cloud computing security challenges & solutions-A survey*. 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC). IEEE. 347-356.
44. Hamdan, S., Ayyash, M., Almajali, S. (2020). Edge-computing architectures for internet of things applications: A survey. *Sensors*, 20(22), 6441.
45. Shi, W., Pallis, G., Xu, Z. (2019). Edge computing [scanning the issue]. *Proceedings of the IEEE*, 107(8), 1474-1481.
46. Khan, W. Z., Ahmed, E., Hakak, S., Yaqoob, I., Ahmed, A. (2019). Edge computing: A survey. *Future Generation Computer Systems*, 97, 219-235.
47. Cao, K., Liu, Y., Meng, G., Sun, Q. (2020). An overview on edge computing research. *IEEE access*, 8, 85714-85728.
48. Foschini, L., Gavagna, A., Martuscelli, G., Montanari, R. (2020, June). *Hyperledger fabric blockchain: Chaincode performance analysis*. ICC 2020-2020 IEEE International Conference on Communications (ICC), IEEE, 1-6.
49. Nehra, V., Sharma, A. K., Tripathi, R. K. (2020). Blockchain implementation for Internet of Things applications. *Handbook of Research on Blockchain Technology*. Academic Press.
50. Golosova, J., Romanovs, A. (2018, November). *The advantages and disadvantages of the blockchain technology*. 2018 IEEE 6th workshop on advances in information, electronic and electrical engineering (AIEEE), IEEE, 1-6.
51. Almagrabi, A. O., Ali, R., Alghazzawi, D., AlBarakati, A., Khurshaid, T. (2021). Blockchain-as-a-Utility for next-generation healthcare Internet of Things. *Computers, Materials & Continua*, 68(1).
52. Tripathi, G., Ahad, M. A., Casalino, G. (2023). A comprehensive review of blockchain technology: Underlying principles and historical background with future challenges. *Decision Analytics Journal*, 9, 100344.
53. Nasir, Q., Qasse, I. A., Abu Talib, M., Nassif, A. B. (2018). Performance analysis of hyperledger fabric platforms. *Security and Communication Networks*, 2018(1), 3976093.

54. Foschini, L., Gavagna, A., Martuscelli, G., Montanari, R. (2020, June). *Hyperledger fabric blockchain: Chaincode performance analysis*. ICC 2020-2020 IEEE International Conference on Communications (ICC), IEEE, 1-6.
55. Ylli, E., Fejzaj, J. (2021, May). Man in the Middle: Attack and protection. *RTA-CSIT*, 198-204.
56. Rahim, R. (2017). Man-in-the-middle-attack prevention using interlock protocol method. *ARPN J. Eng. Appl. Sci*, 12(22), 6483-6487.
57. Knecht, P., Sieber, M. (2020). *TLS 1.3 for StrongSwan: A Client-Side Prototype*, Doctoral dissertation, HSR Hochschule für Technik Rapperswil.
58. Arfaoui, G., Bultel, X., Fouque, P. A., Nedelcu, A., Onete, C. (2019). The privacy of the TLS 1.3 protocol. *Cryptology ePrint Archive*.
59. Huang, L. S., Adhikarla, S., Boneh, D., Jackson, C. (2014). An experimental study of TLS forward secrecy deployments. *IEEE Internet Computing*, 18(6), 43-51.
60. İnternet: Raza, M., End-to-End Encryption: What It Is and How It Works. Splunk. URL: https://www.splunk.com/en_us/blog/learn/end-to-end-encryption.html, Son Erişim Tarihi: 21.02.2025.
61. Tague, P., Poovendran, R. (2008, September). *Modeling node capture attacks in wireless sensor networks*. 2008 46th annual allerton conference on communication, control, and computing, IEEE, 1221-1224.
62. Al-Shatari, M., Hussin, F. A., Aziz, A. A., Eisa, T. A. E., Tran, X. T. (2023). IoT edge device security: An efficient lightweight authenticated encryption scheme based on LED and PHOTON. *Applied Sciences*, 13(18), 10345.
63. Osborn, J. D., Challener, D. C. (2013). Trusted platform module evolution. *Johns Hopkins APL Technical Digest (Applied Physics Laboratory)*, 32(2), 536-543.
64. Wolf, M., Gendrullis, T. (2012). Design, implementation, and evaluation of a vehicular hardware security module. *Information Security and Cryptology-ICISC 2011: 14th International Conference, Seoul, Korea, November 30-December 2, 2011. Revised Selected Papers 14*, Springer Berlin Heidelberg, 301-318.
65. Li, Y., Fan, Y., Zhang, L., Crowcroft, J. (2023). RAFT consensus reliability in wireless networks: Probabilistic analysis. *IEEE Internet of Things Journal*, 10(14), 12839-12853.
66. Xu, Q., Ren, P., Song, H., Du, Q. (2016). Security enhancement for IoT communications exposed to eavesdroppers with uncertain locations. *IEEE Access*, 4, 2840-2853.
67. Lin, C., He, D., Huang, X., Choo, K. K. R., Vasilakos, A. V. (2018). BSeIn: A blockchain-based secure mutual authentication with fine-grained access control system for industry 4.0. *Journal of network and computer applications*, 116, 42-52.

68. İnternet: Memorial. Tokluk kan şekeri kaç olmalı? *Memorial*. URL: <https://www.memorial.com.tr/saglik-rehberi/tokluk-kan-sekeri-kac-olmali>, Son Erişim Tarihi: 21.01.2025
69. Bigini, G., Freschi, V., Lattanzi, E. (2020). A review on blockchain for the internet of medical things: Definitions, challenges, applications, and vision. *Future Internet*, 12(12), 208.
70. Bhardwaj, I., Kumar, A., Bansal, M. (2017, September). *A review on lightweight cryptography algorithms for data security and authentication in IoTs*. 2017 4th International Conference on Signal Processing, Computing and Control (ISPCC), IEEE, 504-509.
71. Dai, H. N., Imran, M., Haider, N. (2020). Blockchain-enabled internet of medical things to combat COVID-19. *IEEE Internet of Things Magazine*, 3(3), 52-57.
72. Maitra, S., Yanambaka, V. P., Puthal, D., Abdelgawad, A., Yelamarthi, K. (2021). Integration of Internet of Things and blockchain toward portability and low-energy consumption. *Transactions on Emerging Telecommunications Technologies*, 32(6), e4103.
73. Rughoobur, P., Nagowah, L. (2017, December). *A lightweight replay attack detection framework for battery depended IoT devices designed for healthcare*. 2017 international conference on Infocom technologies and unmanned systems (trends and future directions)(ICTUS), IEEE, 811-817.
74. Makhdoom, I., Abolhasan, M., Abbas, H., Ni, W. (2019). Blockchain's adoption in IoT: The challenges, and a way forward. *Journal of Network and Computer Applications*, 125, 251-279.
75. Bellia, P. L. (2016). A code-based approach to unauthorized access under the Computer Fraud and Abuse Act. *George Washington Law Review*, 84, 1442.
76. Puthal, D., Nepal, S., Ranjan, R., Chen, J. (2016). Threats to networking cloud and edge datacenters in the Internet of Things. *IEEE Cloud Computing*, 3(3), 64-71.
77. Ahmed, A., Adewale, A. L., James, A., Olaniyi, O. M., Umar, B. U., Emmanuel, S. (2018). Human vital physiological parameters monitoring: A wireless body area technology based Internet of Things. *Jurnal Teknologi dan Sistem Komputer*, 6(3), 115-121.
78. Paul, S., Kuzovkova, Y., Lahr, N., Niederhagen, R. (2022, May). *Mixed certificate chains for the transition to post-quantum authentication in TLS 1.3*. Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security, 727-740.
79. İnternet: Hyperledger Fabric. Certificate authority and certificate management — Hyperledger Fabric documentation. URL: https://hyperledger-fabric.readthedocs.io/en/latest/certs_management.html, Son Erişim Tarihi: 01.05.2025
80. Krstić, M. S., Krstić, L. J. (2020). Hyperledger frameworks with a special focus on Hyperledger Fabric. *Vojnotehnicki glasnik/Military Technical Courier*, 68(3), 639-663.

81. Samonas, S., Coss, D. (2014). The CIA strikes back: Redefining confidentiality, integrity and availability in security. *Journal of Information System Security*, 10(3), 21–45.
82. Florea, B. C. (2018, June). *Blockchain and Internet of Things data provider for smart applications*. 2018 7th Mediterranean conference on embedded computing (MECO), IEEE, 1-4.
83. Huang, D., Ma, X., Zhang, S. (2019). Performance analysis of the raft consensus algorithm for private blockchains. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 50(1), 172-181.
84. Muhammed, R. K., Aziz, R. R., Hassan, A. A., Aladdin, A. M., Saydah, S. J., Rashid, T. A., Hassan, B. A. (2024). Comparative analysis of aes, blowfish, twofish, salsa20, and chacha20 for image encryption. *arXiv preprint arXiv:2407.16274*.
85. Degabriele, J. P., Govinden, J., Günther, F., Paterson, K. G. (2021, November). *The security of ChaCha20-Poly1305 in the multi-user setting*. Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, 1981-2003.



Gazili olmak ayrıcalıktır