BLOCKCHAIN BASED ACCESS MANAGEMENT FRAMEWORK FOR INTERNET OF THINGS (IoT)

A thesis submitted to

DELHI TECHNOLOGICAL UNIVERSITY

For the award of degree of

DOCTOR OF PHILOSOPHY

in

Department of Computer Science and Engineering

By

Rajiv Kumar Mishra

2K18/PHDCO/501

Under the Supervision of

Dr. Rajesh Kumar Yadav

Associate Professor
Department of Computer Science
and Engineering
Delhi Technological University

Dr. Prem Nath

Associate Professor
Department of Computer Science
and Engineering
H.N.B. Garhwal University

Department of Computer Science and Engineering

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042, India

DECEMBER 2023

DELHI TECHNOLOGICAL UNIVERSITY

(Govt. of National Capital Territory of Delhi)
BAWANA ROAD, DELHI – 110042

DECLARATION

I, Rajiv Kumar Mishra (2K18/PHDCO/501), hereby affirm that the research work presented in this thesis titled "**BLOCKCHAIN BASED ACCESS MANAGEMENT FRAMEWORK FOR INTERNET OF THINGS (IoT)**" is an original contribution conducted under the guidance of Dr. Rajesh Kumar Yadav, Department of Computer Science and Engineering, Delhi Technological University, and Dr. Prem Nath, Department of Computer Science and Engineering, H.N.B. Garhwal University. This thesis has not been previously submitted to any other academic institution for the purpose of obtaining a degree or diploma. Throughout the writing process, I have adhered tothe prescribed Ph.D. rules and regulations set forth by the Institute.

I confirm that the thesis does not contain any classified information. Whenever I have utilized external sources, proper acknowledgement has been provided by citing them within the text and including them in the reference list. Direct quotations from external sources have been explicitly identified by quotation marks and have been appropriately referenced both in the text and the reference list.

Date:

Rajiv Kumar Mishra
2K18/PHDCO/501

DELHI TECHNOLOGICAL UNIVERSITY

(Govt. of National Capital Territory of Delhi)
BAWANA ROAD, DELHI – 110042

CERTIFICATE

This is to certify that the research work presented in this thesis titled **"BLOCKCHAIN BASED ACCESS MANAGEMENT FRAMEWORK FOR INTERNET OF THINGS (IoT)"** by Rajiv Kumar Mishra (2K18/PHDCO/501) is an original contribution conducted under the guidance of Dr. Rajesh Kumar Yadav, Department of Computer Science and Engineering, Delhi Technological University and Dr. Prem Nath, Department of Computer Science and Engineering, H.N.B. Garhwal University.

This thesis has not been previously submitted for any other degree or diploma. Rajiv Kumar Mishra has followed the prescribed Ph.D. rules and regulations throughout the writing process.

The thesis does not contain any classified information. Proper acknowledgment has been given for external sources through citations within the text and inclusion in the reference list. Direct quotations have been appropriately identified and referenced.

Date:

Dr. Rajesh Kumar Yadav
Associate Professor (Supervisor)
Department of Computer Science and Engineering
Delhi Technological University

Dr. Prem Nath
Associate Professor (Joint Supervisor)
Department of Computer Science and Engineering
H.N.B. Garhwal University

ACKNOWLEDGEMENT

I want to express my sincere gratitude to **Dr. Rajesh Kumar Yadav**, Associate Professor, Department of Computer Science and Engineering, Delhi Technological University, Delhi, and **Dr. Prem Nath**, Associate Professor, Department of Computer Science & Engineering, H.N.B. Garhwal University, for providing valuable guidance and constant encouragement throughout the work. Their immense knowledge, motivation, expertise, and insightful commentshave helped me immensely at every stage of the preparation of this research plan.

I would also like to extend my heartfelt thanks to the DRC Chairman Prof. Dr. Rahul Katarya, and Prof. Dr. Vinod Kumar for their valuable insights, suggestions, and critical evaluation of my research work. Their expertise and scholarly advicehave played a crucial role in improving the quality of this thesis.

I am thankful to the members of my review committee Prof. Dr D. K. Lobiyal and Dr. S.K. Dhurendher for their valuable suggestions during my presentations.They were always being very supportive and generous throughout the discussions.

I would like to sincerely thank the esteemed members of the Department of Computer Science & Engineering at Delhi Technological University. Their unwavering support and provision of invaluable research resources have been instrumental in the successful completion of this work. Their guidance and assistance have been a constant source of motivation throughout my academic journey.

Moreover, I am profoundly grateful to my parents (Late Mr. Narendra Mishra, Mrs. Shrimati Mishra), my In-laws (Mr. Surendra Chaturvedi, Mrs. Kalindi Chaturvedi), my wife (Mrs. Pragya Mishra), my daughters (Mudra & Udichi),my sisters (Mrs. Ranjana Tripathi, Mrs. Archana Tiwari, Mrs. Anshul Mishra), my Brother-In-laws (Dr. A.P. Tripathi, Mr. Sarvesh Tiwari, Mr. Abhishek Mishra,Mr. Shailendra Chaturvedi, Mr. Sarvendra Chaturvedi, Mr. Nripendra Chaturvedi), whose unconditional support and blessings have been a cornerstone of my achievements. My friends and lab mates (Aastha, Rahul, and Raju) are all appreciated for being such a wonderful group of people, both in and out of the lab. Their unwavering faith in my abilities, endless lessons, moral support, and

care have played an integral role in shaping my academic pursuits. I hold the utmost respect and affection for my elders, whose wisdom and encouragement have guided every step of my journey.

I recognize that this accomplishment would not have been possible without the collective efforts and contributions of all those mentioned above. Their presence in my life has been truly invaluable, and I am forever indebted to them for their unwavering support and belief in my abilities.

Rajiv Kumar Mishra
2K18/PHDCO/501

CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Description
IoT	Internet of Things
BC	Blockchain
DAM	Data Access Management
AC	Access Control
DoS	Denial of Service
QoS	Quality of Service
XACML	Extensible Access Control Markup Language
OAuth	Open Authorization
UMA	User-Managed Access
AS	Authorization Server
PoP	Proof of Possession
JSON	Java Script Object Notation
JWT	JSON Web Token
PEP	Policy Enforcement Point
PDP	Policy Decision Point
CBOR	Concise Binary Object Representation
CWT	CBOR Web Token
AdRBAC	Adaptive Risk-Based Access Control
RBAC	Role-Based Access Control
OBAC	Organization Based Access Control
ABAC	Attribute-Based Access Control
UCON	Usage Control Based Access Control
CBAC	Capability-Based Access Control
TBAC	Trust-Based Access Control
SmartOBAC	Smart Organization Based Access Control
PBAC	Pervasive Based Access Control
TrBAC	Transaction-Based Access Control
SCBAC	Smart Contract-Based Access Control
IPFS	Interplanetary File System
HTTP	Hypertext Transfer Protocol
CoAP	Constrained Application Protocol

DHT	Distributed Hash Table
CA	Certificate Authority
MSP	Membership Service Provider
ABE	Attribute-based Encryption
HABE	Hierarchical Attribute-based Encryption
DO	Data Owner
DR	Data Requestor
CoAP	Constrained Application Protocol
ARC	Access Rights Contract
EC	Entity Contract
TC	Trust Contract
PoW	Proof of Work
PoS	Proof of Stack
PBFT	Practical Byzantine Fault Tolerant
IBFT	Istanbul Byzantine Fault Tolerant
ACC	Access Control Contract
RC	Register Contract
JC	Judge Contract
DC	Device Contract
PC	Policy Contract
RC	Resource Consume
RO	Resource Owner
R	Resource
TS	Trust Score
TS_{local}	Local Trust Score
TS_{global}	Global Trust Score
$W^{(n-i)}$	Aging Parameter
NoRR	Number of Recent Request
SCBAC	Smart Contract Based Access Control

ABSTRACT

Blockchain-Based Access Management Framework for Internet of Things (IoT)

1. Introduction

The Internet of Things (IoT) has revolutionized the landscape of data generation and exchange, where the vast data produced by IoT devices necessitate stringent security measures. Securing IoT devices poses challenges due to resource constraints, limited processing power, and the lack of standardized security protocols. The substantial data volume generated raises privacy concerns, emphasizing the need for ensuring confidentiality and integrity while sharing information across devices and networks. As IoT applications expand across industries like healthcare, manufacturing, and smart cities, the criticality of robust security protocols becomes even more pronounced. Addressing these challenges demands innovative access control frameworks capable of adapting to the dynamic and diverse nature of IoT environments while ensuring data protection and access integrity.

2. Challenges and Motivation

From the literature analysis, various research gaps have been identified which are given below:

- Implementation of challenges arises when applying existing access control standards to smart objects due to their limited capabilities.
- The incorporation of a robust, trusted third party for access control might compromise user privacy.
- Assumptions regarding the constant connectivity of IoT devices to the internet may not always hold true.
- There exists a demand for high data availability within IoT environments.
- The need to dynamically adjust policies to align with the evolving organizational needs remains a crucial consideration.

3. Research Objective

We have focused on the following research objectives:

i. Study and development of authorization algorithm for the data sharing.
ii. Study and design of an architecture for the IoT data management.
iii. Study and development of a data access control model for the Internet of Things.

iv. To perform a comparative analysis of proposed access control techniques with the state-of-the-arts techniques.

4. Research Proposal

We have carried out detailed study of existing literature and identified research gaps which are mentioned above. We have proposed following research proposals to address the identified research gaps:

- **Authorization Algorithm for Data Sharing**: One of the pivotal components of this framework is its implementation of a two-tier authorization system, encompassing both static and dynamic authorization policies. Static policies define access rights based on predefined rules, while dynamic policies dynamically adapt based on real-time interactions, leveraging historical access patterns to assess trustworthiness. The fusion of these authorization mechanisms augments the framework's adaptability, catering to the dynamic nature of IoT environments.
- **Secure IoT Data Management and Sharing Architecture**: The integration of Blockchain and IPFS in this framework ensures secure and decentralized data storage, addressing concerns surrounding data integrity and accessibility. Access policies, encrypted record hashes, and dynamic authorization configurations are securely stored on the Blockchain, while IPFS (Inter Planetary File System) serves as a distributed storage mechanism for actual IoT-generated data. This combination enhances data security, integrity, and availability, while circumventing single points of failure and vulnerabilities inherent in centralized systems.
- **Blockchain based Access Control Model for IoT Environment**: This research presents a pioneering approach to data access management in IoT ecosystems, leveraging Blockchain and IPFS to fortify security, enable efficient data sharing, and establish trust among entities. Adopting a trust-based access control model and implementing dynamic authorization policies offer a resilient and adaptive solution to the evolving challenges of securing IoT environments.

Performance evaluations and simulations underscore the efficacy and scalability of the research proposals, emphasizing its ability to handle substantial IoT data volumes efficiently. The evaluations provide insights into the running costs of smart contracts, validating the framework's feasibility for real-world deployment.

CHAPTER 1

INTRODUCTION TO SECURE DATA SHARING IN IoT ECOSYSTEM

The Internet of Things (IoT) is a transformative paradigm that refers to the interconnectivity of various physical objects, devices, and systems through the internet [1]. These objects, which can range from everyday appliances to complex machinery are embedded with sensors, actuators, and communication technology, which enable them to collect and exchange data with each other and with the broader digital environment. This networked ecosystem enables devices to communicate, share information, and collaborate autonomously, leading to a wide array of applications across diverse industries [2]. The essence of IoT lies in its ability to enhance efficiency, automation and decision-making in both personal and industrial contexts. Through continuous data collection and analysis, IoT enables real-time monitoring, remote control and intelligent automation of processes. This has far-reaching implications from optimizing energy usage in smart homes to enabling predictive maintenance in industrial settings and even revolutionizing healthcare through wearable devices that monitor vital signs [3-5]. IoT's potential impact extends to smart cities, agriculture, transportation, healthcare, manufacturing and more. It holds the promise of creating more sustainable and interconnected systems that can adapt and respond to changing conditions. However, it also brings forth challenges such as security, privacy and the need for robust infrastructure to support the massive influx of data [6, 7].

1.1 Introduction to IoT

In essence, the Internet of Things represents a new era in the digital landscape, where the physical and digital worlds are seamlessly integrated, offering unprecedented opportunities for innovation and efficiency across a wide spectrum of industries and everyday life. As IoT continues to evolve, it is poised to shape the way we live, work, and interact with the world around us [8, 9]. Amid this paradigm shift, IoT introduces security challenges, emphasizing risks in data sharing,

intricate data access controls, and authorization complexities stemming from diverse devices. Ensuring data integrity and confidentiality in the face of large data volumes necessitates agile authorization models and resilient data management strategies.

1.2 Components of IoT

The Internet of Things (IoT) encompasses a diverse range of components, each playing a crucial role in creating a connected ecosystem [10-14]. Here are the key components of IoT:

i. **Sensors and Actuators:** Sensors are devices that detect changes in the physical environment and convert them into electrical signals. These changes could be anything from temperature variations to motion or light. Actuators, on the other hand, are responsible for performing actions based on the instructions they receive. For example, a thermostat (actuator) might adjust the temperature based on input from a temperature sensor.

ii. **Connectivity:** This component enables devices to communicate with each other and with central systems or servers. It includes various communication protocols such as Wi-Fi, Bluetooth, Zigbee, LoRa, and cellular networks. These technologies facilitate the transmission of data between IoT devices and the broader network or cloud.

iii. **Microcontrollers and Processors:** Microcontrollers are embedded chips that serveas the "brains" of IoT devices. They handle data processing, decision-making and control functions. These components are essential for managing the operation of IoT devices, running software and interfacing with sensors and actuators.

iv. **IoT Gateways:** Gateways act as intermediaries between IoT devices and the central data processing systems or the cloud. They aggregate data from multiple devices, perform initial processing or filtering and then transmit the relevant information to the cloud. This helps reduce the amount of data sent over the network and can provide local processing capabilities.

v. **Cloud and Edge Computing:** The cloud serves as a centralized platform for storing, processing, and analyzing the vast amounts of data generated by IoT devices. Edge computing, on the other hand, involves processing data locally on devices or in nearby edge servers. This reduces latency and bandwidth usage, enabling faster decision-making in real-time applications.

vi. **IoT Platforms:** IoT platforms provide the infrastructure and tools necessary to manage and orchestrate the various components of an IoT ecosystem. They often include features for data management, device management, security, analytics and application development. These platforms are essential for creating, deploying, and managing IoT applications and solutions.

vii. **Security and Encryption:** Security measures are crucial to protect IoT devices and the data they transmit. This includes encryption protocols to secure communication channels, authentication mechanisms to verify the identity of devices and users, andother security features to safeguard against unauthorized access and cyber threats.

viii. **User Interface and Applications:** This component involves the interfaces that allow users to interact with IoT devices and systems. This can include mobile applications, web interfaces, or even voice-activated assistants. These interfaces provide users with the ability to monitor, control and receive information from IoTdevices.

ix. **Data Storage and Analytics:** Data storage solutions are necessary to store and manage the large volumes of data generated by IoT devices. Analytics tools enable the extraction of valuable insights from this data. This can include techniques like data mining, machine learning and predictive analytics.

x. **Power Management and Energy Efficiency:** This component focuses on optimizing power consumption to extend the lifespan of IoT devices and reduce the need for frequent battery replacement or recharging. Energy-efficient design considerations are crucial for IoT devices that operate on limited power sources.

xi. **Scalability and Interoperability:** IoT solutions need to be scalable to accommodate a growing number of devices and users. Interoperability ensures

thatdifferent devices and systems can work together seamlessly, even if they come from different manufacturers or use different communication protocols.

These components collectively form the foundation of the Internet of Things, working together to create connected ecosystems that enhance automation, efficiency, and decision-making across a wide range of industries and applications [15, 16].

1.3 Security Challenges in IoT

The Internet of Things (IoT) is a transformative technology paradigm that has rapidly evolved in recent years, connecting a vast array of devices and systems to the internet. IoT encompasses everything from everyday consumer devices like smart thermostats and wearable to industrial machinery and critical infrastructure components. This interconnected landscape has brought about numerous benefits, including enhanced efficiency, convenience and data-driven insights. However, it has also introduced a host of security challenges and vulnerabilities that must be addressed to realize its full potential [17-19].

Following are some key security challenges encountered within IoT environment:

i. **Interoperability and Standards:** IoT encompasses a wide array of devices, each with its own set of communication protocols and security mechanisms. This heterogeneity can lead to security gaps and vulnerabilities as devices struggle to communicate and collaborate effectively [20]. Addressing this challenge requires a collaborative effort to establish and adhere to comprehensive security standards that are universally applicable across diverse IoT devices and platforms.

ii. **Device Proliferation and Heterogeneity:** In the realm of IoT, one of the fore most challenges lies in the sheer proliferation of devices, coupled with their inherent heterogeneity. IoT ecosystems are characterized by an extensive array of interconnected devices, each exhibiting distinct functionalities, communication protocols and security capabilities [21]. This diversity spans from high-end, well-secured devices to resource-constrained sensors and actuators that may lack robust security features.

iii. **Data Privacy and Encryption:** The nature of IoT deployments often involves the collection of sensitive data from a multitude of connected devices, necessitating the highest levels of data privacy and security. To safeguard this sensitive information, robust encryption and data protection mechanisms are paramount. Encryption ensures that data remains confidential and tamper-resistant, even in the event of unauthorized access or interception [22]. However, the challenge arises from the sheer scale and diversity of IoT ecosystems. Implementing end-to-end encryption and guaranteeing data privacy across this expansive network of interconnected devices can be a formidable task.

iv. **Authentication and Authorization:** The foundation of IoT security lies in ensuring that access to IoT resources is limited to authorized devices and users. Robust authentication and authorization mechanisms are pivotal in achieving this objective. Authentication verifies the identity of entities attempting to access IoT resources while authorization governs the privileges and permissions granted to those entities. Implementing authentication and authorization mechanisms that are both secure and suitable for resource-constrained devices can be a complex endeavor [23].

v. **Data Management:** IoT generates vast amounts of data and managing these data efficiently are a major challenge. It involves issues like data storage, real-time processing, data integration and ensuring data accuracy and consistency [24].

vi. **Network Security:** The foundation of the Internet of Things (IoT) lies in its ability to connect and communicate seamlessly. The challenge in IoT network security stems from the complexity and diversity of these networks which can include traditional wired networks, wireless technologies and various communication protocols [25]. Vulnerabilities in network protocols, coupled with inadequate network security measures, can create significant points of vulnerability within IoT systems. Securing IoT networks requires a comprehensive approach that encompasses encryption, intrusion detection systems, access controls and ongoing monitoring to detect and respond to

potential threats promptly.

vii. **Denial of Service (DoS) Attacks:** In the landscape of the Internet of Things (IoT),susceptibility to Denial of Service (DoS) attacks is a notable concern. These attacks aim to disrupt IoT systems by overwhelming them with a flood of traffic or requests, rendering devices unresponsive or unavailable. The challenge arises fromthe resource limitations of many IoT devices [26]. These devices often operate withconstrained computational power and memory, making them particularly vulnerable to resource-intensive DoS attacks. Mitigating such attacks in IoT environments is a complex endeavor.

viii. **Scalability:** As the number of IoT devices continues to grow rapidly, managing and scaling IoT infrastructure becomes increasingly complex. Scalability issues canlead to network congestion, reduced performance and difficulties in device management [27].

ix. **Reliability and Quality of Service (QoS):** IoT devices often operate in challengingenvironments therefore ensuring their reliability and QoS can be difficult. Factors such as network latency, device failures and connectivity issues can impact the performance of IoT systems [28].

x. **Energy Efficiency:** Many IoT devices are battery-powered or have limited energy sources. Optimizing energy consumption to extend device's lifespan is crucial. Energy-efficient communication protocols and low-power hardware design are essential for IoT success [29].

1.4 Data Access Management in IoT

Data Access Management (DAM) in connection with IoT is a critical framework and set of practices that govern how data is accessed, utilized and protected within IoT ecosystems [30]. IoT represents a vast and interconnected network of devices, sensors and actuators that generate and exchange a staggering volume of data. These data includes a wide range of information from environmental and operational data to sensitive personal and industrial data. Data access management is essential to ensure that this wealth of information is accessed and handled securely, efficiently, and in compliance with regulations [31].

Data access management in IoT is essential for ensuring the security and integrity of the data generated and exchanged within an IoT ecosystem. It involves controlling and regulating who can access what data and under what circumstances. Authentication protocols are used to verify the identity of users, ensuring that only authorized entities can access the information. Authorization mechanisms determine what actions a user or device is permitted to perform, such as read, write, or modify data [32]. Encryption is employed to safeguard data during transit and storage, protecting it from unauthorized access or tampering. Access policies, often based on models like Role-Based Access Control (RBAC) or Attribute-Based Access Control (ABAC), help define and enforce these permissions. Additionally, compliance with relevant data protection and privacy regulations is crucial, ensuring that data access management practices align with legal requirements [33-35]. Overall, an effective data access management strategy in IoT is a multi-layered approach that combines technical measures with well-defined policies and procedures, adapted toevolving security threats and technologies.

Both data access control and data management as depicted in Figure 1.1 are integral components of a comprehensive access management strategy in IoT. Together, they help safeguard the confidentiality, integrity, and availability of IoT data, ensuring that it is used and accessed appropriately while minimizing security risks. In the subsequent section, we have further elaborated on these two aspects of data managementin detail.

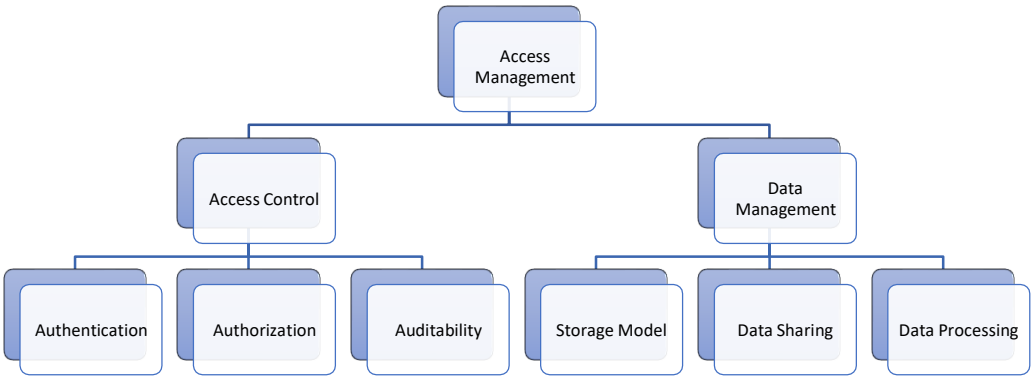

Figure 1.1: Access Management Process

Access Control: Access control in the context of Data Access Management (DAM) plays a pivotal role in determining who can access data within an IoT ecosystem. It is the process of regulating and restricting access to data resources to ensure that only authorized entities, including users and devices, can interact with the information [36]. Access control encompasses defining access policies, authentication, authorization, and enforcement mechanisms. In DAM, effective access control mechanisms help prevent unauthorized access, mitigate security risks, and maintain data privacy and integrity.

Data Management: Data management within DAM encompasses the practices and processes involved in organizing, storing, and maintaining data in an IoT environment. It addresses data collection, storage, indexing, and retrieval. Proper data management ensures data is well-organized, easily accessible, and efficient to handle [37]. It also includes data categorization, data lifecycle management, and measures to maintain data integrity. In the context of DAM, robust data management supports effective access control by categorizing and structuring data, making it easier to apply access policies, and optimizing data sharing for IoT applications while adhering to compliance and privacy regulations.

1.5 Data Access Control

Data access control refers to the mechanisms and policies put in place to manage who has permission to access specific resources or data within an IoT ecosystem. It essentially governs how users and devices interact with the IoT network and each other. This includes both human users and automated systems ordevices.

In the IoT context, access control involves authenticating and authorizing users based on their identity, roles, and permissions [38]. It aims to prevent unauthorized access and restrict actions to only those that are permitted. For example, in a smart home environment, access control might dictate that only authorized users can control the thermostat settings or access the video feed from asecurity camera. Access control solutions are mainly either framework-based or model-based. In the subsequent section, we will explore a few prominent frameworks and models which are applied to achieve access control.

The framework-based solutions are categorized into policy-based, token-based, and hybrid solutions. From the model-based perspective, the access control scheme is broadly classified into three major categories: Centralized Model, Decentralized Model, and Hybrid Model, while from the framework-based perspective access control is bifurcated into the centralized and decentralized model. These classifications are further categorized and elaborated below as in Figure 1.2.

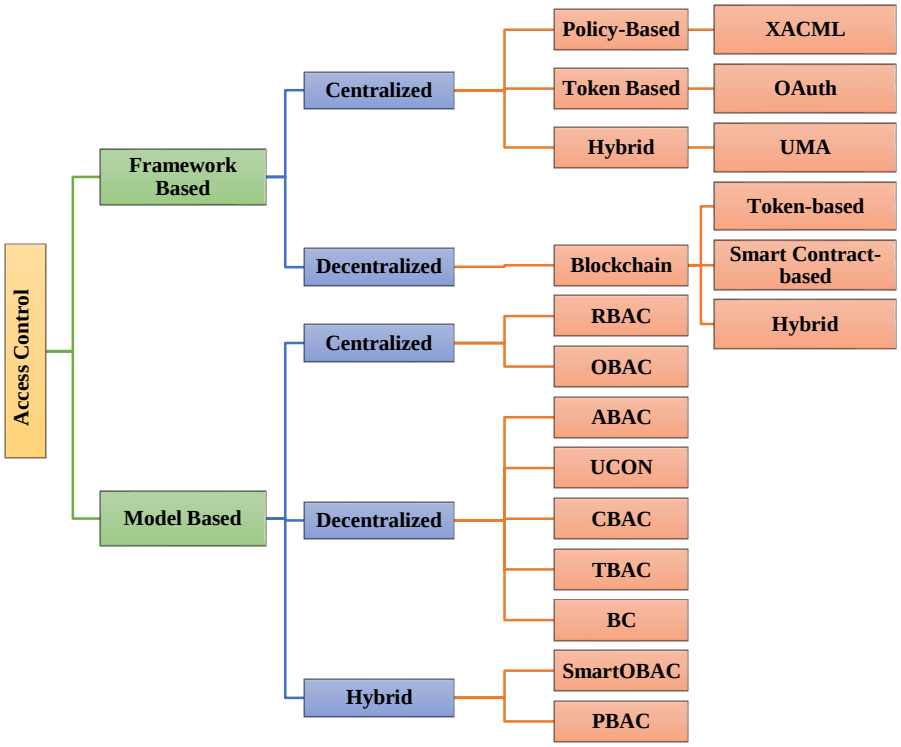

Figure 1.2: Access Control Solutions

1) Frameworks for Access Control

The framework-based classification consists of two categories: Centralized and Decentralized. Centralized frameworks are further classified into policy-based (XACML), token-based (OAuth), and hybrid (UMA). The decentralized approach is based on Blockchain technology.

i. **XACML:** Extensible Access Control Markup Language (XACML) is an extensively used standard that gives the capability to express distributed AC policies and to describe response decisions for the same [39]. Among the various components of XACML, two major components are PEP and PDP. Policy Enforcement Point (PEP) is accountable for the enforcement of the

application of the underlying policies. To do the same, it translates the request into corresponding business logic and forwards it to the PDP. Only after receiving an evaluation from the PDP, a decisionis made as to whether to allow or deny access to the requested resource. A Policy Decision Point (PDP) interacts with a pre-existing repository where authorization policies are stored and based on the supplied attributes and authorization policies, the PDP performs an evaluation and passes it to the PEP. Apart from these two components, XACML consists of two additional components: Policy Administration Point (PAP) and Policy Information Point (PIP). PIP is a source of information required for policy evaluation. PAP serves as a policy repository and tender mechanism to manage policy.

ii. **OAuth:** Open Authorization (OAuth) is an open protocol that offers a token-based secure authorization framework. This framework lets third-party applications access restricted resources on behalf of the user without imposing the need for the user's credentials [40]. The architecture encompasses four key components:

- Client: Third-party application that wishes to gain access to controlled resources.
- Resource Owner: Authorize to allow or deny access to the requested resource.
- Resource Server: Entity entitled to exposing restricted resources.
- Authorization Server: Entity that controls and manages the authorization mechanism.

iii. **UMA:** User-managed Access (UMA) is an OAuth-based specification that seems a very attractive mechanism for access control. UMA focuses on extending the very basic use of authorization policy to a broader and heterogeneous perspective. UMA follows the core concepts of OAuth 2.0 with many improvisations. For token representation, UMA exploits Concise Binary Object Representation (CBOR) WebToken (CWT) and as a replacement for TLS (Transport Layer Security), it utilizes Datagram Transport Layer Security (DTLS) [41]. For the application layer protocol, UMA implements

CoAP/MQTT.

iv. **Blockchain:** In general, Blockchain (BC) can be described as a technology that facilitates the integrity and immutability of the information where multiple distributed interlinked nodes in a P2P network are utilized to maintain records of transactions [42]. The distinct properties of BC make it a promising solution to address the fast-growing nature of IoT networks in a decentralized environment. BC technology carries out a self-directed verification before granting the transaction which in turn plays a vital part in offering security.

2) Authorization Model-Based Access Control

This section covers several models to enforce access control. These models are broadly categorized into three groups: centralized, decentralized, and hybrid. The subsequent sub-sections will describe various access control models.

i. **Role-Based Access Control (RBAC):** Role-Based Access Control (RBAC) is a widely used access control model that defines permissions and access rights based on users' roles within an organization. It provides a structured and efficient way to manage and control access to resources, ensuring that users have the appropriate level of access based on their roles and responsibilities [43].

ii. **Organization-Based Access Control (OBAC):** An Organization-Based Access Control (OBAC) model is a type of access control system that revolves around the structure and hierarchy of an organization. Unlike Role-Based Access Control (RBAC), which focuses on user roles and their associated permissions, OBAC places emphasis on the organizational structure and relationships among different entities within an organization [44]. In an OBAC model, access decisions are often made based on the position, department, or other organizational attributes of users.

iii. **Attribute-Based Access Control (ABAC):** Attribute-Based Access Control (ABAC) is a dynamic and flexible access control model that operates based on attributes associated with users, resources, and contextual conditions. In ABAC, entities are characterized by attributes such as roles, job titles, or

location, providing a comprehensive representation [45]. Access control policies are expressed using attribute-value pairs, allowing for fine-grained and context-aware rules. Following a Subject-Object-Action (SOA) model, ABAC evaluates policies dynamically, adapting to changes in attributes or context. This model supports real-time adjustments to access permissions and considers attributes like time, location, and device information.

iv. **Usage Control Based Access Control (UCON):** Usage Control (UCON) is an access control model that focuses on governing access based on dynamic changes in the conditions of use. Unlike traditional access control models, UCON extends beyond static permissions and incorporates continuous evaluation of attributes, context, and the ongoing behavior of users during the interaction with resources [46]. It emphasizes defining policies that dictate how access should be granted or denied based on the evolving usage context. This model provides a more nuanced approach to access control, allowing organizations to express complex rules that respond to changing circumstances. UCON is particularly beneficial in scenarios where access requirements are contingent on contextual factors, making it suitable for dynamic and evolving environments.

v. **Trust-Based Access Control (TBAC):** Trust-Based Access Control (TBAC) is a security model that leverages the concept of trust relationships to govern access to resources. In TBAC, access decisions are influenced by the level of trust established between entities within a system. Trust is a subjective measure that reflects the confidence or reliability attributed to an entity based on its past behavior, credentials, or other relevant factors [47]. In this model, entities can be users, devices, or any components interacting within a network or system. The level of trust associated with each entity is dynamically evaluated and updated over time. Access privileges are then determined by considering not only the permissions assigned but also the trustworthiness of the requesting entity.

vi. **Capability-Based Access Control (CBAC):** Capability-Based Access Control (CBAC) is a security model that focuses on granting permissions

based on the possession of specific capabilities, or "tokens," rather than relying solely on user identities or roles [48]. In CBAC, entities are given tokens that represent specific permissions, and access is granted based on the possession of the appropriate token.

vii. **Smart Organization Based Access Control (SmartOBAC):** SmartOBAC, an extension of the Organization-Based Access Control (OBAC) model, is a sophisticated access control framework designed to address the unique challenges of the Internet of Things (IoT) environment. OBAC, as a foundational access control model, focuses on organizing access policies around an organization's structure, roles, and responsibilities [49]. SmartOBAC considers the constraints of IoT devices, addresses organizational hierarchies, and facilitates secure collaboration, making it well-suited for the dynamic and interconnected nature of IoT ecosystems.

viii. **Pervasive Based Access Control Model (PBAC):** The Pervasive-Based Access Control (PBAC) model addresses the intricacies of pervasive computing environments, where computing capabilities are seamlessly integrated into various facets of daily life. This model is designed to be adaptive and context-aware, considering dynamic factors such as location, time, and user activities in access control decisions [50]. PBAC's user-centric design ensures personalized access control, incorporating individual preferences and profiles. Recognizing the diverse array of devices in pervasive computing, PBAC accommodates their varied characteristics and capabilities.

1.6 Data Management

Data management in IoT involves the processes and practices related to handling, storing, processing, and securing the data generated by IoT devices. It encompasses various aspects such as data collection, storage, retrieval, processing, analysis, and disposal. Effective data management in IoT is crucial for maximizing the value of IoT deployments. It enables organizations to derive actionable insights, improve operational efficiency, and ensure data security and compliance [51]. Additionally, it helps in optimizing resource utilization and minimizing the risks associated with

managing large volumes of IoT-generated data. Following are the key aspects of data management in IoT as represented in Figure 1.3 and elaborated subsequently:

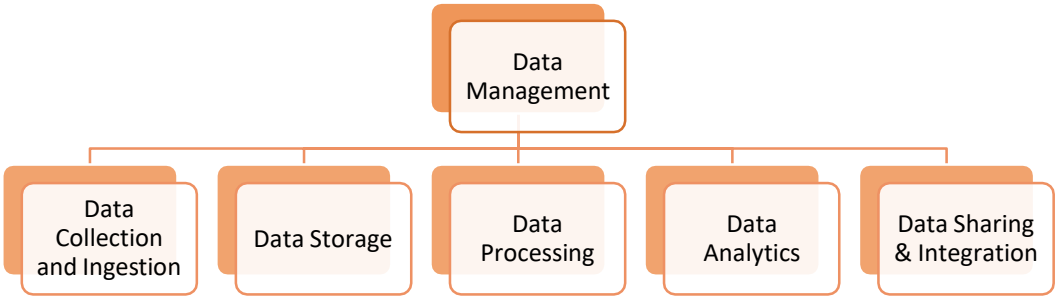

Figure 1.3: Data Management Elements

i. **Data Collection and Ingestion**: Data collection and ingestion in IoT involves gathering information from various sources, such as sensors, devices, and applications. These sources may communicate using different protocols and send data in various formats. Establishing standardized protocols and data formats is crucial to ensure reliable transmission and interpretation of data [52]. Additionally,implementing redundancy and error-checking mechanisms ensures that data is collected consistently, even in the event of network disruptions or device failures. This phase is fundamental as it forms the foundation for subsequent data processingand analysis.

ii. **Data Storage**: Data storage in IoT encompasses determining where and how the collected data will be stored. This can include cloud platforms, on-premises servers,edge computing devices, or hybrid environments. Each storage option has its own considerations; for instance, cloud platforms offer scalability and accessibility, while edge devices provide low-latency processing [53].

iii. **Data Processing**: Data processing and transformation involve converting raw, often unstructured data into a usable format for analysis. Pre-processing steps are applied to clean, aggregate, filter, and enrich the data, making it suitable for further analysis [54].

iv. **Data Analytics**: Data analytics in IoT involves applying various algorithms,

statistical methods, and machine learning techniques to gain meaningful insights from the processed data. These analyses can range from basic statistical measures to sophisticated machine learning models. The goal is to identify patterns, anomalies, trends, and correlations within the data [55].

v. **Data Sharing and Integration**: Data sharing and integration involve facilitating the exchange of information between different devices, systems, or platforms within the IoT ecosystem [56]. Seamless data sharing and integration are critical for achieving a cohesive and efficient IoT ecosystem.

By addressing these aspects of data management in IoT, organizations can effectively handle the influx of data from IoT devices, extract meaningful insights, and ensure the security and privacy of sensitive information. Among above-discussed data Management aspects, our work emphasizes features such as data storage and data sharing & integration.

1.7 Blockchain and its Integration with IoT

Blockchain is a decentralized, distributed ledger technology that enables secure and transparent record-keeping of transactions across a network of computers. It operates on a peer-to-peer network, where each participant, or node has a copy of the entire ledger. This technology is known for its robust security features, immutability of records, and its ability to establish trust among participants without the need for a central authority.

When integrated with IoT, Blockchain can enhance the security and privacy of data transactions within the IoT ecosystem. One significant application of this integration is in implementing access control. This involves leveraging Blockchain's unique attributes to manage and regulate the authorization and authentication processes for IoT devices. By utilizing Blockchain for access control in IoT, organizations can establish a tamper-proof and transparent system. Access permissions and authentication information are recorded on the Blockchain, providing an immutable ledger of device interactions [57]. This ensures that only authorized devices and users have access to specific resources or data, and any attempts at unauthorized access can be easily identified and prevented.

Furthermore, Blockchain's decentralized nature eliminates the need for a central authority or intermediary to oversee access control, reducing the potential for single points of failure or security breaches. Instead, access rights are managed through smart contracts, which are self-executing contracts with predefined rules encoded on the Blockchain. These smart contracts automatically enforce access policies, ensuring that only authenticated and authorized entities can interact with IoT devices. Incorporating Blockchain into IoT access control also addresses the issue of trust among participants [58]. The transparency and immutability of Blockchain records still have confidence in the authenticity of transactions, making it difficult for malicious actors to manipulate access privileges.

Overall, the integration of Blockchain technology with IoT for access control offers a powerful solution to enhance the security and integrity of IoT ecosystems. It provides a robust framework for managing access permissions, ensuring that only trusted and authorized entities can interact with IoT devices, thereby safeguarding sensitive data and critical resources.

1.8 Major Challenges in IoT Environment and Motivation

The motivation for focusing on access control and data management models in the context of the Internet of Things (IoT) is driven by a complex interplay of crucial factors. Security is a paramount concern, given the diversity of IoT devices and the potentialvulnerabilities they face. Robust access control models are imperative to safeguardagainst unauthorized access and potential threats to the IoT infrastructure. Privacy preservation is equally critical, particularly with the collection of sensitive personal and organizational data in IoT applications. Access control and data management models play a pivotal role in ensuring the confidentiality and security of this information. Moreover, the efficient utilization of the vast amount of data generated by IoT is essential for enhancing operational efficiency. These models help organize and categorize data, enabling the right information to be accessed by the right entities at the right time, supporting real-time decision-making and, optimizing IoT systems. Lastly, these efforts contribute to building a resilient and scalable foundation for IoT technology, ensuring its future relevance and security as it continues to expand into diverse industries and applications.

Here are a few key challenges present within the IoT environment:

i. **Security and Vulnerabilities:** The IoT landscape encompasses a wide array of devices, some of which may have limited security features. This diversity introduces security challenges, making it crucial to develop robust access control and data management models. These models help prevent unauthorized access, mitigating potential threats and vulnerabilities within the IoT ecosystem.

ii. **Data Privacy:** IoT applications often involve the collection of sensitive and personal data, such as healthcare information or smart home data. Protecting the privacy of individuals and organizations is both a moral imperative and a legal requirement. Effective access control and data management models ensure the confidentiality and security of this sensitive information.

iii. **Operational Efficiency:** IoT generates massive volumes of data, and optimizing its use is essential for improving operational efficiency. Access control and data management models facilitate the organization and categorization of data, ensuring that the right data is accessible by the right entities at the right time. This streamlines data sharing, supports real-time decision-making and maximizes the utility of IoT systems.

iv. **Secure Data Sharing:** Secure data sharing ensures that data is exchanged among authorized entities while maintaining its confidentiality and integrity. This aspect is particularly significant in IoT, where data often needs to be shared across various devices and applications to enable real-time decision-making and collaborative processes.

In summary, the motivation behind working on access control and data management models in the context of IoT is rooted in the need to address security challenges, protect data privacy, enhance operational efficiency, achieve compliance, adapt to dynamic environments, and future-proof IoT technology as it continues to evolve and expand.

1.9 Problem Statement and Research Proposal

Given the above limitations observed in the literature, we have formulated the following problem statement in the context of significant concerns regarding the security and management of access to IoT-generated data.

- Existing access management systems often struggle to address the unique challenges posed by IoT,including scalability, data availability, and the need for decentralized access control.
- The problem at hand is to design, develop, and evaluate a robust and scalable Blockchain-based access management framework tailored specifically for IoT environments.

In connection with the above problem statement, we have formulated the following research proposal:

- Develop a framework ensuring secure, efficient, and decentralized access control for IoT resources, emphasizing confidentiality, integrity, and availability.
- Address challenges linked to data availability, privacy, and compliance with emerging IoT data protection regulations.
- Bridge the gap in existing IoT access management solutions by leveraging Blockchain technology to offer a comprehensive and future-proof solution for IoT access control challenges.

1.10 Organization of Thesis

This thesis presents a comprehensive data access management framework tailored for the Internet of Things (IoT) environment. The study is structured into distinct chapters, each offering unique contributions to the overarching thesis. The following outlines the contributions of each chapter."

Chapter 1: Introduction

In this chapter, we provided an introduction to the Internet of Things(IoT), offering insight into its fundamental concepts and principles. Subsequently, we have delved into the motivation that pushed this study and presented a comprehensive problem statement, articulating the specific challenges and objectives we aim to address.

Additionally, we proposed a brief overview of the succeeding chapters of this thesis.

Chapter 2: Literature Survey

A comprehensive and in-depth review of the existing literature is provided, encompassing a wide range of approaches and techniques employed in the context of data access management within the IoT environment.

Chapter 3: Authorization Techniques

In this chapter, a comprehensive exposition of the proposed authorization algorithms is offered, accompanied by a detailed analysis of the results, including comparisons with alternative methods.

Chapter 4: Data Management

This chapter presented details of the data management model that facilitated secure data sharing.

Chapter 5: Access Control Model

This chapter explained the method of access control for the IoT environment along with its results.

Chapter 6: Conclusion and Future Scope of Work

This chapter encapsulated the findings and outcomes stemming from the methods and algorithms put forth in this study. It also delved into an extensive discourse on prospective avenues for future research and development.

1.11 Simulation Platform

The development environment setup involves installing specific prerequisites on a system configured with an Intel Core i5 CPU running at 2.25 GHz, 8 GB of primary memory, and operating on Ubuntu 20.04. The required prerequisites include Git client, Docker & Docker Compose, Go programming language, and Node.js & NPM. Upon the successful installation of these prerequisites, Hyperledger Fabric, a permissioned Blockchain platform version 2.2 LTS, is deployed. Additionally,

Kafka is implemented to facilitate consensus among the nodes within the Blockchain network. The smart contract (also known as chaincode) is developed using the Go programming language. Within the Hyperledger Fabric framework, client nodes utilize chain code to propose transactions, while peer nodes execute the chain code and collaborate for consensus among the network nodes. The management hub that acts as an interface between IoT smart devices and Blockchain nodes is a JavaScript interface. To interact with Blockchain nodes, the interface employs web3 JavaScript, and to communicate with smart devices it uses the CoAP JavaScript library. The simulations of the proposed model were performed on Kosarak [59], a real data set representing the number of clicks of a news portal. The number of news pages accessed within a specific day by a user is recorded by the data set. The experimental work was performed on the 10 MB of data while the Kosarak data set size was slightly larger, so a fraction of records were removed from the data set. Additionally, the caliper-benchmark testtool is utilized to simulate and conduct experiments for evaluating the system's performance and functionality.

CHAPTER 2

LITERATURE SURVEY

A literature survey on data access management in the Internet of Things (IoT) reveals the evolving landscape of research and development in this field. Access control in IoT is a critical aspect, as it ensures the security and privacy of IoT devices and the data they generate.

In this chapter, we have presented a comprehensive examination of state-of-the-art methods and techniques employed in the domain of data access management within the Internet of Things (IoT). The survey is systematically structured into two primary sections. The initial section explores access control techniques applied at the application layer within IoT environments, while the subsequent section delves into the realm of data management, presenting secure data-sharing methodologies.

2.1 Data Access Control for IoT Environment

Data access control is a critical aspect of IoT security and privacy, and understanding the approaches used at the architecture layer is essential for safeguarding IoT ecosystems [60]. In this section, we have discussed many prominent framework-based access control solutions for IoT environments.

2.1.1 Framework-based Access Control

Atlam et al. [61] proposed an access control mechanism “AdRBAC” - Adaptive Risk-Based Access Control. The presented model consists of four inputs: user context, resource sensitivity, action severity, and risk history. These elements are utilized to determine the risk linked with every access request. Eventually, the risk policy gets evaluated against the value of risk determined in the previous step to make access decisions. The risk policy of the AdRBAC model is specified through the XACML standard. The flow of the decision process for access control begins with the user sending an access request. Once the access request is received by the system, the risk value is estimated by considering various factors like user context, resource sensitivity, action severity, and risk history. Finally, based on the risk policy and estimated risk value, access is either granted or denied.

Sciancalepore et al. [62] proposed an open standards-based access control framework OAuth-IoT. Moreover, OAuth 2.0 is based on the assumption that the secured resource is always equipped with an internet connection, and with this connectivity, resources are capable of interacting with AS to verify the scope and validity of tokens supplied by clients. In a constrained environment, this assumption/requirement is not possible.

Cirani et al. [63] offered an OAuth-based authorization service framework for the IoT environment "IoT-OAS". This architecture provides HTTP/CoAP service providers with an authorization delegation scheme in the IoT scenario even without bothering to implement OAuth logic. This framework encompasses five elements to achieve delegation function-based authorization. However, the big size of packets at the application level requires fragmentation which in turn contributes to higher radio transmission. Consequently, higher energy consumption is incurred.
The author in [64] presented a mechanism to offer access control for web-based services in IoT by incorporating IoT devices with an access control scheme. Working on this architecture starts with the resource owner (RO) registering the resource server (RS) on the authorization server (AS). The client on behalf of the RP seeks access to protected resources through the Resource Server. Once the Client getsequipped with token and authorization data, it is capable of accessing the resource on RP's behalf.

2.1.2 Authorization Model-Based Access Control

This section covers several models to enforce access control. These models are broadly categorized into three groups: centralized, decentralized, and hybrid. The subsequent sub-sections will describe various access control models.

Role-Based Access Control (RBAC) [65] and Organization-Based Access Control (OBAC) [66] are two major centralized-based schemes for access control. Since both of these models, RBAC and OBAC are based on a centralized architecture, they are easy to implement and manage but also confronted with few inbuilt limitations. The major limitations of these schemes are that they are not suitable for IoT devices as implementation is too complex to implement without any lightweight tool or mechanism. Additionally, single-point failure, large-scale implementation, and flexibility are other concern that prevails. The distributed architecture comprises several models: Attribute-Based Access Control (ABAC) [67, 68], Usage Control-Based Access Control (UCON) [69, 70], Trust-Based Access Control (TBAC) [71], and Capability-Based Access Control (CBAC) [72]. Attributes being the core concepts in ABAC models provide more scalable and fine-grained means to gain access to resources. However, it also exhibits a few limitations and the most crucial one is its complex deployment, apart from that sensor data and attribute values are required to map together.

UCON encompasses a collection of new perceptions in contrast to prevalent conventionalmodels but it's not enough to take the context of IoT into account for several reasons: broad elucidation of the access method is missing and the availability of only conceptual model as of now.

TBAC introduced some dynamic elements in the decision process of the access scheme in terms of trust value which is associated with every constrained device. But so far this model is only implemented for the cloud environment and it is not fit for a constrained environment.

CBAC is based on the notion of capability which is nothing but a privilege and entities possessingthe privilege are granted to access the specified resource. Despite providing better flexibility and distribution than the previous mechanisms, this model has to cope with various limitations. One of the major concerns of this model is its usability on mobile devices and not considering the context during the evaluation of the accesspermission process. Capability propagation and revocation are also an issue that needs to be tackled.

Hybrid architecture-based models are Smart Organization Based Access Control (SmartOBAC) [73] and Pervasive Based Access Control (PBAC) [74]. Although these hybrid approaches tender better flexibility and scalability but are susceptible to DoS attack in certain scenarios (if overflow at node surpasses threshold) and security strategy descriptions are complicated.

A summary of comparative analysis of model-based access control solutions is illustrated in Table2.1

Table 2.1: Comparative analysis of data access control models

Reference	**Model**	**Advantages**	**Limitations**
[65]	Role Based AccessControl (RBAC)	Deals with the distribution problem of competencies where time and location change	-Centralized architecture -Unable to provide scalable and flexible access control in IoT environment.
[66]	Organization Based Access Control (OBAC)	Simplicity coming from the abstraction of entities	-Centralized architecture -Large scale implementation. -Too complex to implement in IoT environment
[67,68]	Attribute Based Access Control (ABAC)	-More scalable and fine grained control over existing model. -Deals with the dynamic propagation problems.	-Do not consider specification of IoT environment. -How to effectively link sensor data to attribute values.
[69,70]	Usage Control Based Access Control (UCON)	-Offers attribute mutability -Authorization is handled in dynamic way.	-Does not have a precise definition of its rigorous definition -Only a conceptual model

[71]	Trust Based Access Control (TBAC)	Access request can be evaluated within reasonable and acceptable processing time.	-Appropriate for cloud and not IoT oriented.
[72]	Capability Based Access Control (CBAC)	Offers more flexibility and distribution than the earlier models	-Context is not considered. -Usability on mobile devices.
[73]	Smart Organization Based Access Control (SmartOBAC)	More flexible, scalable and fine-grained capacity.	-Complexity of definition of its security policy
[74]	Pervasive Based Access Control (PBAC)	-Large scale adaptation -Dynamic and Pro-active.	-Working model for decentralized architecture yet to be implement.

2.1.3 Blockchain-based Access Control Solutions

Because of various striking features, Blockchain technology has been explored to propose decentralized data access control models in a trust-less network environment. Access Control policy based on Blockchain technology is used to publish access policy and permit distributed transfer access rights among the users. This mechanism permits distributed auditability and thus stops any third party from unethically repudiating rights allowed by the policy. Blockchain-based solutions have multiple inbuilt advantages like decentralization, immutability, resilience, and data integrity. Consequently, many researchers have presented an array of access control solutions by incorporating existing models with Blockchain (mainly smart contracts). Solutions based on Blockchain are mainly classified into Token-Based Access Control, Smart Contract-Based Access Control (SCBAC), and Hybrid-based Access Control.

Following are the different categorization of Blockchain-based data access control model:

a) Token-Based Access Control

Maesa et al. [75] introduced a Blockchain-based solution for generating and managing access tokens, representing the access privileges of a subject for a

specific resource. The access model involves two types of transactions. Firstly, the resource owner can initiate the first transactions to create and transfer an access token, under the condition that access policies are satisfied by the subject attributes. Access policies and user attributes play a role in the evaluation process of an access request, and both are stored in an external authorization system. The interaction between this external authorization system and the resource owner forms the basis for every access decision. The second type of transaction occurs when the transfer of an access token from one subject to another subject takes place. However, there is notable room for improvement in embedding access control within the framework of Blockchain technology. In this model, access control policies are not self-enforced, and the evaluation of policies is not automatic.

Ding et al. [76] outlined an access control mechanism, specifically attribute-based, leveraging Blockchain concepts to enhance access management in the IoT environment. The model introduces a novel transaction type designed to record attribute authorizations. It emphasizes the independence of IoT devices from the consensus process of Blockchain technology. However, the application of Blockchain is limited to distributing attributes to prevent data tampering and avoid single-point failure. Notably, there is a significant computational overhead, directly proportional to the number of attributes in the system. Additionally, the storage overhead of session keys scales with the number of participants in the network, raising concerns about effective scalability.

Ouaddah et al. [77] presented an access control solution for an IoT environment named Fair Access. The owners of the underlying resource have the authority to state access strategies and generate an access token if these policies are fulfilled by the user. To perform this process an explicit transaction is utilized termed a “Grant Access Transaction”. Whenever a user wishes to interact with a resource, he needs to perform a “Get Access Transaction”. Besides, by transferring the token to a new owner, the current owner (having a token) can delegate access to someone else. In this model, Consensus confirmation may lead to a considerably long wait time. To achieve granular access control scripting language is not appropriate and can be better replaced with smart contracts.

Xue et al. [78] proposed a private Blockchain-based access control mechanism for smart homes, in which the admin is responsible for visitor authentication and regulates all smart devices by defining access policies for them. On every access request by the visitor, the admin verifies the identity and access rights of the visitor. An access token and a key are produced if verification is successful. To prevent data tampering, all the access policies are recorded on the Blockchain. However, despite offering better security to smart devices, it misses self-enforcement of access rules which is a key feature in the context of Blockchain.

BlendCAC [79] is a Blockchain-enabled capability-based access management technique for the IoT and highlights three key things: managing the capability, approving access rights, and delegating the rights. After receiving a service query from the subject (user), the service provider retrieves a capability token from the smart contract, and based on the local access rights it determines whether to approve the service or not. The authorization of access rights incorporates smart devices which makes the entire system more scalable. However, in this scheme, smart devices are always presumed to be connected which is not feasible in the context of IoT.

Fotiou et al. [80] presented a Blockchain-based access control framework, in which clients are not supposed to connect with gateways or smart devices directly but through the Blockchain. The access rights of the clients depend on how many tokens they possess, more tokens means more access rights. Moreover, to access any device a client is required to have at least one such token. The smart contract verifies if the client has the required number of tokens or not, meanwhile, the gateway checks the role assigned to the user and the location of the resource. Based on thesefactors access to the requested resource is granted by the proposed framework.

Patil et al. [81] presented a Blockchain-based lightweight framework to secure a smart agriculture farm. To avoid a single point of failure, each smart node collectively elects the head of the cluster which is part of an overlay Blockchain network. In this framework, there is a local Blockchain implemented through a private Blockchain platform that manages the interaction among smart greenhouse nodes. All the devices in the smart greenhouse record their data over the central

cloud storage. However, this proposal has no implementation or simulation yet. Additionally, this framework does not highlight the synchronization process for the transaction between the two types of Blockchain networks involved.

Dorri et al. [82] presented "Smart Home" a Blockchain-based solution to secure IoT. This framework encompasses a central storage scheme, an overlay (Blockchain-based) network, and a smart home. The access rules are recorded on a local Blockchain network implemented through a private Blockchain. In this scheme, a centralized manager termed as "block manager" is elected as head of thecluster among all the devices in the smart home. The block manager distributes shared keys to the devices in the smart home, regulates all transaction queries, andmanages the devices in the smart home. So, IoT devices in the smart home are controlled by a centralized node and thus this framework does not provision decentralization.

b) Smart Contract-Based Access Control (SCBAC)

Novo et al. [83] have presented a smart contract-based solution. IoT devices are excluded from the Blockchain because of their constrained nature. Alternatively, a new component is incorporated into the architecture named the management hub whose function is to request access control information on behalf of IoT devices. Moreover, the architecture consists of dual Blockchain terminals: the managers and the agent. Manager's node is accountable for managing access control permission of IoT devices and they are not supposed to participate in the mining process. A smart contract encompasses all the capabilities of access management and its deployment is done by an agent node. On receiving access requests for a resource, the IoT device leads these requests to the management hub which in turn is associated with the miner node. Miner node makes interaction with the smart contract and verifies whether the requester has appropriate permission on the specified resource or not. Although, the feasibility of this framework has not been demonstrated until now in real-world IoT scenarios.

Hwang et al. [84] expanded the work presented in [78] and offered a distributed access control mechanism in the context of the dynamic environment of IoT where managers are equipped with the functionality to create access policies at run time

for the legitimate devices. In the proposed model, IoT devices are classified into three categories: the first category signifies the unlisted devices with limited access, the second category represents a set of listed devices with no access policy, and the third category shows listed devices having access policies. Management hub and dynamic policy generator are the two key components, where the former has the same functionality as in [78] and the latter is accountable for creating access policies dynamically. On receiving access queries from the category 2 devices, the management hub communicates with the manager through the smart contract for producing an access policy and eventually lists it on the Blockchain. As soon as access policies are listed, the management hub proceeds with the data sharing between the devices.

To offer dynamic access control [85] propose a technique that combines the concept of machine learning and smart contract by evaluating the subject's behavior. The proposed architecture comprises multiple access control contracts (ACC), one judge contract (JC), and one register contract (RC). RC contract is dedicated to managing both the ACC contract and the JC contract. One access control routine of each subject and resource pair is defined by the ACC, a function meant to update the access policy is also implemented by ACC. ACC also manages a misbehavior list that describes every misbehavior done by the subject to the specific resource. Whenever a subject seeks access, ACC is executed for detecting misbehavior and report to the JC contract if any misbehavior is detected. The JC contract states the penalty is based on a predefined judging method for misbehavior. Finally, the access request is granted only in the absence of misbehavior. Until now this framework is not deployed in a real-world IoT environment and thus overhead test has not been conducted extensively to demonstrate the feasibility of this framework.

Liu et al. [86] presented multiple smart contract-based models in connection with the pre-existing ABAC core concept. Their model comprises three types of smart contracts: Device Contract (DC), Policy Contract (PC), and Access Contract (AC). Each of these contracts serves distinct functionality to offer decentralized, dynamic, and fine-grained access management for a constrained environment. DC offers a

means to store the URL of the resource data produced by the device and a relevant way to query the data. PC includes a method to manage ABAC policies and is meant for the admin user. AC is the central part and is used to employ access control processes for common users. The distributed performance of the model is yet to be demonstrated. There is a scope to improve the scalability of this model. The throughput and reliability of the system were tested on very limited physical devices.

In the work presented by Pinno et al. [87], a framework named "ControlChain" was introduced for access control, leveraging Blockchain technology. The ControlChain framework encompasses four distinct Blockchain within its database: Context Blockchain, Relationship Blockchain, Accountability Blockchain, and Rule Blockchain. However, a notable drawback in this scheme arises from the inclusion of smart devices as integral components of the Blockchain, leading to several adverse effects on the system. The major challenge is scalability, as the involvement of smart devices in the Blockchain introduces challenges in managing the increasing scale of the system. Additionally, constrained devices with limited resources face difficulties in maintaining updates, posing an additional constraint on the overall effectiveness of the framework.

Paillissue et al. [88] extended the idea of group-based policy to provision a multi-administrative sphere through a permissioned Blockchain (Hyperledger Fabric). The secure policy distribution and preserving the individuality of the associated organization are the main focus of the proposed model. It lets users from diverse organizations to access and share their resources among themselves irrespective of the organization they belong to. The framework of this model encompasses three layers: the policy interface, the Blockchain, and the network layer. The policy layer lets the administrators specify users and is accountable for making/removing access policies. The Blockchain safeguards the integrity and correctness of recorded data. Pal et al. [89] proposed an architecture for access right delegation for the IoT by exploiting the concept of Blockchain. The usage of Blockchain offers enough safety to the IoT data. In this model, the main objective is to exploit the attributes to authenticate the identity of the entities rather than counting on their distinct identity.

This architecture is based on the notion of dual Blockchain, one private Blockchain is used to record the user's attribute to offer privacy to the data while another public Blockchain is exploited for the additional computations as required. Apart from Blockchain the other key components of the model are brokers, buyers, attribute providers, user devices, resources, and resource managers. Moreover, the authors presented two implementations: the first offers better security but incurs more cost for an access request, while the second decreases the cost from the user's perspective but increases the computational work at the attribute provider side.

c) Hybrid-Based Access Control

Alphand et al. [90] introduced an access management model named "IoT Chain," utilizing Blockchain technology for enhanced security in IoT environments. In this framework, resources requiring protection are stored in encrypted form on a resource server, owned by the resource owner. When a third-party client seeks access to a protected resource, it requests a key from the key server, where its authorization is authenticated through the Blockchain. Upon successful validation, a key is provided, allowing the client to connect with the resource server. The client then downloads the resource in encrypted form and subsequently decrypts it using the key obtained from the key server. However, the effectiveness and resilience of this architecture have yet to be evaluated across various applications implemented on the IoT Chain.

In the study by Siris et al. [91], a decentralized system for the Internet of Things (IoT) environment was proposed, utilizing Blockchain and interledger technologies. The Blockchain component of the model also functions as a database where policy definitions and cryptographic hashes of authorization information are stored. The architecture includes multiple authorization servers (AS), and a scheme for selecting a subset of m authorization servers from a pool of n such servers is implemented to achieve distributed authorization. Two methods are employed for the selection of m AS from the available n AS: one based on performance parameters and the other based on the responses of the first m AS. It is noteworthy that the cost of the entire authorization procedure increases with the higher number of authorization servers involved in the system.

Outchakoucht et al. [92] introduced an architecture for access control in IoT which employs machine learning and Blockchain technology. In the proposed scheme, the resource holder is supposed to define access rules for their resources within the smart contract. On each access query, smart contracts execute automatically and produce an authorization token if the access request is found valid. Additionally, machine learning algorithms are utilized to optimize the predefined access rules. However, this architecture is still to be implemented, and its feasibility in the context of security is not demonstrated yet.

Shafagh et al [93] presented an IoT data-sharing scheme facilitated by the distributed access control system. The proposed scheme is decomposed into a data plane and a control plane. The data are recorded at the network through a Blockchain-controlled locality-aware decentralized storage model. The data is stored on fixed-size successive chunks which are uploaded on nodes closer to the owner whereas metadata of chunks is recorded in the distributed hash table (DHT). Additionally, some notable works in [94,95] offer access control solutions for IoT environments but their solution either does not encompass dynamic authorization policies or considers IoT nodes as part of the Blockchain network. Both assumptions or not suitable in the context of IoT.

2.2 Data Management Solutions

The emergence and adoption of Blockchain technology as an influential tool for offering tamper-proof data in the domain of data security has significant dominance as well as development space. To avail the benefits associated with Blockchain technology, several researchers have already started incorporating it. Data recording, data sharing, payment, and privacy safeguarding, are the few applications where Blockchain has been significantly effective. Some of the relevant existing work are highlighted here:

Balamurugan et al. [96] offered token-based access control and data-sharing mechanisms for the IoT environment. The proposed platform facilitates resource sharing among smart devices having a valid token, which is approved by the data providers.

However, in this scheme, in addition, to a large number of data consumers, the load at data providers increases significantly and lowers the performance of the model. Additionally, this model does not employ compact tokens and on top of this, there is no support for either dual authentication or accesscontrol scalability.

Sun et al. [97] presented a data-sharing platform for the IoT environment. The proposed framework enjoys greater execution efficiency asresource sharing among smart devices is enabled through a third-party service organization. However, the creation and maintenance of data sharing incur more costs and are susceptible to multiple attacks by unauthorized entities. This platform does not address data tampering or data leakage issues.

Ge et al. [98] presented a Blockchain-based lightweight and secure data-sharing platform for the IoT environment. However, on deploying this model in a particular industry, it is noted that this framework is neither capable of handling concurrency nor offer privacy protection to the shared data significantly.

Xue et al. [99] proposed a Blockchain-powered medical data-sharing mechanism with an enhanced consensus protocol. The refined consensus protocol addresses the problem of examining, recording, and syncing healthcare data among various healthcare organizations.

Liang et al. [100] introduced Hyperledger fabric-based information-sharing scheme applicable to power networks. The authentication and network consensus are done by the power node, added dynamically through a different kind of transaction. Multiple trading centers exploit data consensus mechanisms for data storage resulting in ease of supervision and incurring a low cost of management. However, as of now, it applies to small-scale applications.

Xu et al. [101] presented ABE(Attribute-based Encryption) technology-based hierarchical attribute-based encryption (HABE) algorithm by introducing many hierarchical authorization centers. This scheme also provides a smart contract-enabled access control and data management mechanism for systems with huge data and enormous users. The two smart contracts employed to minimize

decryption costs and offer secure access to shared data are the decryption contract andvalidation contract respectively. The Decryption contract aims to enhance the user's decryption performance by carrying out fractional encryption on the cipher text. The Validation contract analyses users ' access rights and ensures only valid users with appropriate attributes have the privilege to access shared information.

Al Breiki et al. [102] presented a Blockchain-based decentralized andscalable solution for access control and data management in the IoT environment. The author utilizes the features of multiple oracles that act as an interface betweenIoT, Blockchain, and users to offer interoperability among the participating entities.The proposed mechanism employs multiple smart contracts: access control contract, reputation contract, and aggregator contract. The access contract forwardsthe access query to the aggregator contract that passes this query to a set of oracles.The trusted oracles fetch required data and based on their reputation score aggregator collects the desired data. The reputation score is computed by the reputation contract and it also creates an access token for the users. However, this mechanism retrieves data that is not used completely at a time. Moreover, multipleoracles lead to higher costs and higher latency.

Battah et al. [103] proposed a Blockchain-based mechanism to access IPFS-encrypted data. In this work, the data owner (DO) employs a symmetric key method to encrypt its data and send it to the IPFS with another encrypted key. Eventually, do use the private key and public key of the data requestor (DR) to generate a re-encryption key. On receiving an access request from DR, the smart contract verifies the request and issues an access token if validation is successful. Additionally, for the validation of access query multi-party authorization is employed. After receiving encrypted IPFS data, DR first decrypts the key with its private key and then decrypts the desired data with recently decrypted eventually. However, several cycles of encryption and decryption incur a higher cost of data sharing.

Sun et al. [104] proposed a Blockchain and IPFS-based storage and access control scheme for insurance data. The client on registering themselves with an insurance agency receives a pair of public and private keys and their insurance is recorded on

the IPFS and the corresponding cryptographic hash is uploaded on the Blockchain. While claiming for the insurance, this model verifies the claim as well as the client and returns a token encompassing the cryptographic hash. Subsequently, the insurance record is downloaded by the client through hash value forwarded towards the encrypted insurance record to the fog node for decryption.

Marangappanavar et al. [105] envisaged a Blockchain-powered and IPFS-based four-layer framework for decentralized storage and access control namely: user layer, query layer, data control layer, and storage layer. This framework was proposed for the healthcare domain. The user-layer deals with the storage and access of data and comprises patients, doctors, and medical-claim agencies. The query layer acts as an interface for the participating entities and is responsible for retrieving, recording, sharing, and answering queries. Once a user has registered him/her into the system, the query layer delivers a private key and address to the user. The data control layer performs some computation and keeps track of actions performed on the data. Additionally, it ensures that data is not accessed without the consent of the corresponding patient. The storage layer is accountable for recording the data on the IPFS and its hash on the Blockchain. The access policies are enforced through the smart contract. However, the key limitation of this framework is scalability as it is meant for a single hospital.

Shuaib et. al. [106] presented a Blockchain-based framework for sharing medical records on decentralized storage independent of third-party intermediaries. The medical records are primarily in the form of medical images which are first encrypted and subsequently, their cryptographic hash is recorded on the decentralized storage media, making the entire storage model more secure and free from central storage issues such as single point of failure and censorship of data.

Zaabar et. al. [107] presented a Blockchain-based solution to address security issues, especially in dealing with cyber threats. The proposed system is developed on top of Hyperledger Fabric and IPFS enables remote patient monitoring and offers off-chain storage ofencrypted health data over IPFS. The simulation and testing were done using Hyperledger Caliper to evaluate its usefulness for throughput and

latency.

Azbeg et al. [108] presented a Blockchain and IPFS-based healthcare system (BlockMedCare) for managing chronic diseases such as diabetes. In this proposal, patients are equipped with IoT-enabled wearable devices through which the system collects and shares their data with concerned medical teams. All the related entities such as doctors, hospitals, and diagnostic centers connect with patients through a Blockchain network to access patients' health data which is recorded on IPFS in encrypted form.

Oktian et al. [109] proposed "BorderChain" a Blockchain-based mechanism to achieve access control. In this, only authorized nodes are allowed to communicate with IoT gateways. However, it is also a token-based approach and is not compatible with real-time use cases.

Rizzardi et al. [110] presented the integrationof permissioned Blockchain along with IoT middleware considering fog computing perspective. This mechanism utilizes the consensus feature of Blockchain to preventaltering predefined access rules within the IoT network. However, this work was restricted to very few data sources and did not simulate malicious behaviors in theIoT context.

Han, Dezhi, et al. [111] proposed an attribute-based access control modelfor the Internet of Things. The proposed solution utilized Hyperledger fabric to protect against unauthorized access to sensitive data. However, the scalability of this solution is a big concern, as there is a big mismatch in the speed of IoT data generation and Blockchain block creation and validation. Additionally, this model has not been implemented or tested in real-world scenarios.

Shi et al. [112] proposed aprivate Ethereum-based access control mechanism for distributed IoT systems. This model ensures a single identity applicable to all domains within a distributed IoT network and thus simplifies the complexity of the identity management process. However, the proposed model is not suitable for small IoT networks, and even the conventional access control approach performs better in thisscenario. Additionally, this approach is not appropriate for privacy

protection as the algorithm works quite slowly.

Sisi Zuhu et al. [113] proposed a Blockchain-based solution for energy-aware mobile crowd sensing in the Internet of Things (IoT). However, the model was designed to be energy-aware, but the exact energy consumption of the system was not evaluated.

Kamal et al. [114] presented a confidentiality-preserving architecture for the distributed cloud storage systems. The proposed architecture even beats popular techniques such as Advanced Encryption Standard (AES) in terms of memory consumption and time taken to perform encryption and decryption. However, it is restricted to genetic algorithms only while can have a better scope if extended for deep learning or fuzzy logic.

In addition, some notable work done by [115, 116] by incorporates Blockchain technology to offer a secure data-sharing scheme with a reduced ledger size at each peer. Following are the summary of some of the prominent work in the domain of data access control and data-sharing techniques in Table 2.2.

Table 2.2: Comparative analysis of access control and data sharing solutions

Ref	Technology	Advantages	Limitations
[63]	OAuth 2.0, Cloud	• Very efficient if implemented for specific organization	• IoT devices are need to be always connected. • Big size of packets requires fragmentation which in turn contributes to higher radio transmission and higher energy consumption.
[74]	XACML	• offers computational offloading and collaboration between several organizations	• Simultaneous access incurs a moderate overflow at the node implementing the corresponding code. • Prone to DoS attack.

[76]	Hyperledger Fabric	• This method avoided data tampering and simplifed the access control protocol to meet the computing power and energy constraints of the IoT devices.	• Computational overhead is significant as it is proportional to the number of attributes in the underlying system. • Incurs high storage overhead.
[77]	Bitcoin	• Implement more granular access control policies	• Necessity of contact with the owner of the resource for each new access or each token expiration • Support to token-based authorizations only
[79]	Private Ethereum	• Involving smart objects in access right authorization process allows device-to-device communication, which implies better scalability	• It is essentially still a centralized AC scheme • The potential depth of the delegation tree makes it hard for administrators to follow and manage authorizations.
[83]	Ethereum	• Query permission operation does not incur any fee or significant delay. • Fetching access control information of specific device are executed immediately.	• Devices are not able to verify their registration under a manager that makes the system substantially insecure if a malicious manager. • If management hub fails, devices connected to it disappear.
[85]	Ethereum	• Prevents DoS and DDoS attacks	• One contract is required per subject-object pair. The policies are therefore static and hyper-specific. • The addition of a user requires the deployment of a number of smart contracts which takes time, and space.
[86]	Hyperledger Fabric	• High throughput in large scale request environment. • Reach consensus efficiently.	• Reliability and throughput of the system need to test on more physical devices. • Performance of the system is carried on two PC's only.

[87]	Ethereum	• Fully decentralized, more scalable and fault-tolerant	• For better compatibility this framework becomes more complicated and inapplicable on resource constraineddevices.
[91]	Ethereum, OAuth 2.0	• Reduced transactions costs and delay • Reduce the amount of data that needs to be sent to IoT devices.	• Implements token-based authorization. • Multiple authorization servers incur higher operating cost.
[94]	Ethereum	• Better data reliability and availability as compared with traditional cloud storage. • Fine-grained access control is achieved.	• Data owner have to distribute secret key for users and encrypt hisdata. • Access policy update and user attribute revocation is not possible.
[95]	Hyperledger Fabric	• Improved transaction processing rate of all peers due to load distribution.	• High computation cost
[101]	Ethereum, IPFS	• It reduces the reliance on centralized data storage and potentially increasing data availability	• Does not address data transmission from cloud. • The transaction fees associated with using Ethereum can become substantial as the volume of transactions and data storage increases. • Scalability limitations can be a hindrance when managing a large number of files.
[104]	Hyperledger Fabric, IPFS	• Scalability • Low storage space	• Double chain cooperation • Higher operational cost
[115]	Ethereum, IPFS	• Multiparty authorization schemes offer robust control over data access	• Medical device data control is not considered. • The implementation cost is relatively high. • The need for multiple parties potentially slow down key exchanges.

[116]	Ethereum, Cloud	• Mitigates various types of attacks, including Denial of Service, mining attacks, and storage breaches	• High bandwidth • High computation cost. • Scalability issue

2.3 Summary

Given the above-observed limitations in existing literature, our proposition introduced an architecture leveraging Blockchain and IPFS. This framework facilitated the storage of IoT-generated data across distributed storage systems while storing their corresponding hashes on the Blockchain. Notably, the model incorporated a trust-based access control mechanism comprising both static and dynamic components. Our proposed system adopted a two-phase authorization scheme tailored to accommodate the dynamic nature inherent in IoT systems.

CHAPTER 3

AUTHORIZATION ALGORITHM FOR DATA SHARING

The core concept behind the inception of the Internet of Things (IoT) is to provide convenient, smart services through interconnected devices, shaping our daily life. These devices observe and gather data from the IoT environment, distributing them to authorized users. The management of IoT data involves various stakeholders, including resource owners, consumers, administrators, data repository hosts, and providers. However, this interconnected IoT landscape is susceptible to unauthorized access, potentially exposing sensitive data to malicious entities.

3.1 Introduction

In this chapter, we have introduced a decentralized authorization model for IoT based on Blockchain technology. This model leverages smart contracts, offering a decentralized data-sharing mechanism free from a single point of failure. Notably, in our proposed model, IoT devices operate independently from the Blockchain network, enhancing scalability.

Security and privacy remain significant concerns within the dynamic and varied IoT environment. Figure 3.1 illustrates prominent threat categories prevalent in IoT, highlighting the need for a robust authorization technique to mitigate these threats. Access management mechanisms, including authentication, authorization, and accountability, play a pivotal role in securing communication among IoT devices and preventing unauthorized resource access. The primary aim of an effective access management system is to ensure the security of IoT devices and resources, meeting crucial security requirements such as availability, integrity, and confidentiality.

Preventing unauthorized access to data resources is paramount, especially in IoT environments, where numerous interconnected devices exchange sensitive information.

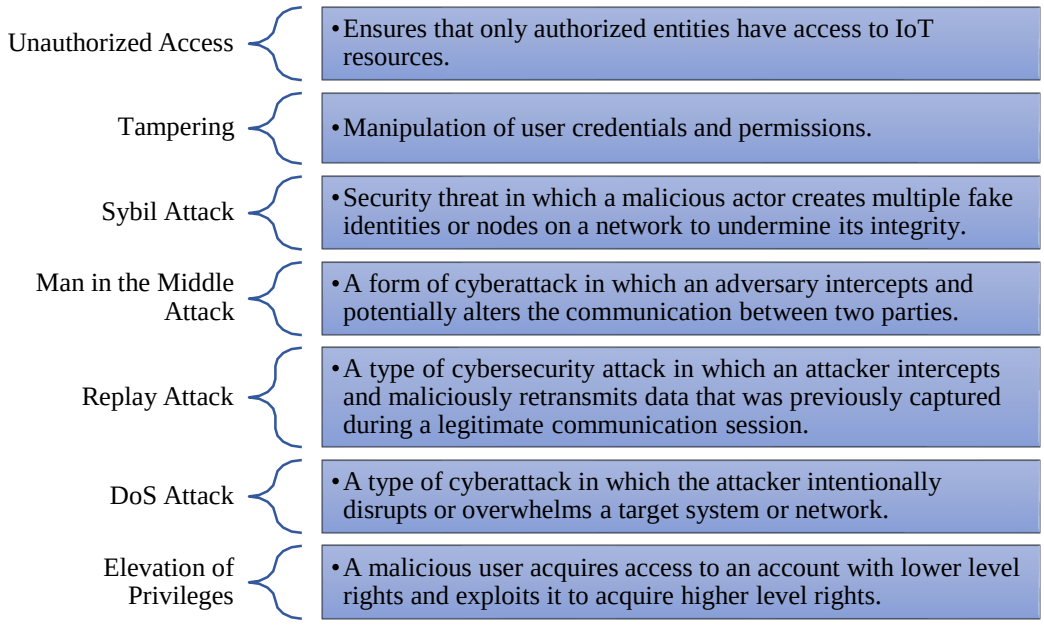

Figure 3.1: Security attacks

Authorization techniques play a pivotal role in ensuring secure data access and preventing a wide array of potential attacks in information systems, including those within the context of the Internet of Things (IoT). These techniques act as a safeguard by defining and enforcing the specific permissions and privileges granted to users or entities, determining what they can and cannot do with the data. By meticulously regulating access to data resources, authorization techniques not only protect against unauthorized access but also shield against a multitude of potential attacks.

Firstly, authorization ensures that only authorized users or devices can interact with sensitive data. This eliminates the risk of data breaches resulting from unauthorized entry, a common target for malicious actors seeking to gain access to valuable information. By clearly delineating who can access the data and under what conditions, authorization techniques effectively thwart this initial layer of attack.

Secondly, they prevent data tampering and unauthorized modifications. By defining permissions for actions like editing, deleting, or modifying data, authorization techniques serve as a safeguard against data manipulation or unauthorized alterations. This is especially critical in IoT applications, where data

integrity is paramount for accurate decision-making.

Furthermore, authorization mechanisms enable the implementation of the principle of least privilege. This means that users or entities are granted only the minimum level of access necessary to perform their tasks. It limits their scope of interaction with the data, reducing the attack surface and minimizing the potential impact of security breaches. In the event of an attack, the restricted privileges can help contain and mitigate the damage.

Lately, Blockchain technology has emerged as a crucial component in overseeing, managing, and securing smart devices. Nevertheless, both Blockchain and IoT come with their limitations that require resolution before their integration. Current research on merging these technologies is still in its early stages, with significant ongoing efforts aimed at enabling the integration of Blockchain networks within IoT ecosystems.

This chapter presented a robust authorization algorithm for IoT access control involving careful consideration of the unique characteristics and challenges of the IoT ecosystem while adhering to security best practices and compliance requirements. Designing an authorization algorithm for access control in IoT involves creating a framework that allows or denies access to IoT resources based on predefined policies, permissions, and some dynamic information (trust score) associated with each requesting entity within the IoT ecosystem. Our algorithm mainly comprises two elements: static and dynamic aspects. The static parts consist of pre-defined access policies corresponding to a given resource and its consumer while the dynamic aspects take the trust score of each requesting entity into consideration. The trust score of every entity (owner and consumer) keeps updating depending on their past access interaction. Additionally, the most recent interaction has a high influence on the trust score while the oldest access interaction has the least weightage. On every access request raised by a consumer, both static and dynamic authorization are evaluated and if both are evaluated as true then only the request isapproved.

3.2 Motivation and Contribution

The motivation behind designing decentralized authorization algorithms stems from the inherent limitations and vulnerabilities present in centralized authorization systems, particularly within the expanding landscape of the Internet of Things (IoT). Traditional authorization frameworks often encounter scalability challenges, single points of failure, and susceptibility to security breaches. The design of decentralized authorization algorithms aims to overcome these hurdles by fostering a more resilient, scalable, and secure approach. By distributing authorization logic across the network and leveraging the principles of decentralization, these algorithms not only enhance system reliability but also bolster user privacy and data security. This chapter contributed a Blockchain-based authorization algorithm that aligned with the evolving landscape of decentralized technologies, ensuring robust access control in dynamic and interconnected environments.

3.3 Proposed Work

The proposed authorization techniques consist of the following components as illustrated in Figure 3.2, that form a comprehensive IoT access control system that ensures secure, transparent, and controlled access to IoT resources while leveraging Blockchain technology for trust and auditability.

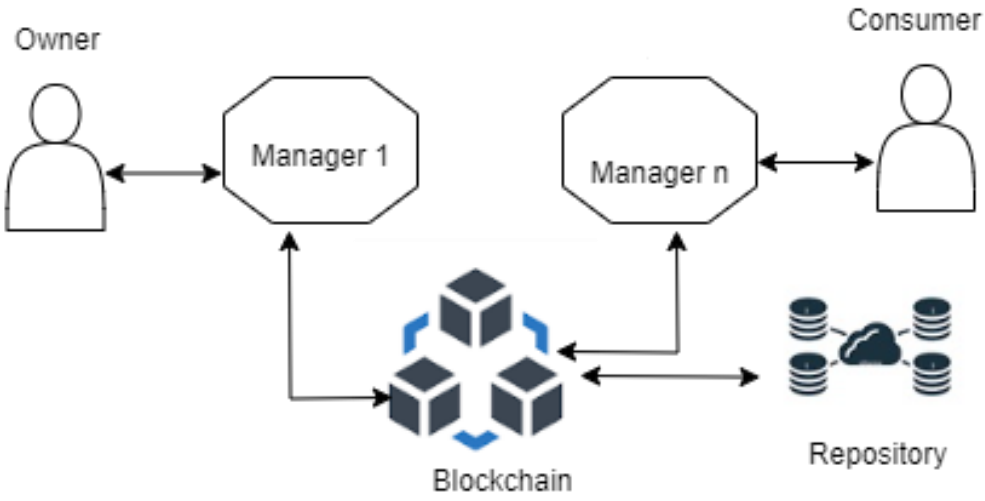

Figure 3.2: System Components

Consumer: An IoT device functioning as a data client that seeks access to resources within the IoT ecosystem. Consumers initiate requests to access IoT resources, such as data or services, and must go through an authorization process to gain access.

Owner: An IoT device that possesses and controls a specific set of resources within the IoT ecosystem. Owners have ownership rights over IoT resources, including data, and determine access policies for Consumers. They grantor deny access based on predefined rules.

Manager: An interface connecting IoT devices to the Blockchain network, responsible for various tasks, including IoT device management, access policy recommendations, and the registration of Consumers and Owners in the Blockchain. Managers act as intermediaries, facilitating communication between IoT devices and the Blockchain. They offer administrative functions to manage IoT devices efficiently and assist in access control policy management.

Blockchain: A decentralized ledger that records transactions occurring within the peer-to-peer (P2P) IoT network. All participating nodes in the network maintain identical copies of this ledger. The Blockchain serves as a secure and immutable record of access control transactions, ensuring transparency, trust, and accountability in the access management process.

Repository: A storage system where IoT Owners upload data following successful authentication. Authorized Consumers can retrieve data from this repository after completing the authorization process. The Data Repository interfaces with both IoT devices and the Blockchain network. The DataRepository securely stores IoT data and facilitates controlled access to authorized Consumers. It is integrated with IoT devices for data storage and retrieval and maintains links to the Blockchain for recording access events.

The proposed work introduced a system model comprising five key operations performed over the Blockchain-based system as depicted in Figure 3.3, are elaborated below:

System Setup: The manager initiates the system setup by generating a public-private key pair for each constrained device registered under its purview. These keys are linked to the Blockchain - the public key is stored in a public database while the encrypted private key is incorporated into the transaction.

Registration: Before registering devices on the Blockchain, the manager first registers constrained devices. Device consent is mandatory before registration with the manager. Although devices retain the option to de-register from the manager, they are required to remain registered with at least one manager at all times.

Policy Definition: The manager, in collaboration with the resource owner, defines access policies governing the system.

Token Generation: Upon successful authorization, authorized consumers receive an access token integrated with a timestamp. This token is generated and transmitted to both the manager and the repository.

Token Verification: The access token generated by the Blockchain contains the consumer's signature and a timestamp, which undergoes verification at the repository to ensure validity.

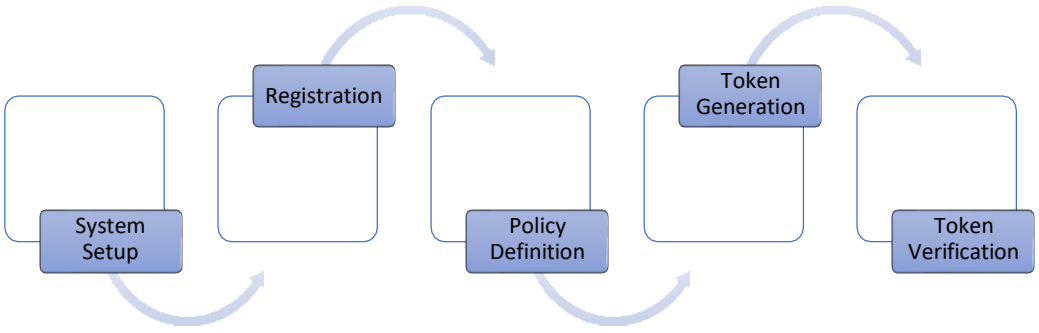

Figure 3.3: System Operations

The authorization model is shown in Figure 3.4. It is evident that the consumer's access request is submitted to the enforcement point, which further interacts with the authorization management smart contract. In this smart contract, first, the decision point performs static and dynamic authorization, and if both the

authorization returns true, then only the request is granted. Static authorization deals with predefined access policies while dynamic authorization, determines the trust score of the consumer through the past interaction score. There are two more smart contracts proposed with the model: user management smart contract and data management smart contract. The user management smart contract deals with the registration of participating entities such as owners and consumers. The data management smart contract covers post-access activities and critical for evaluating access behavior of the consumers.

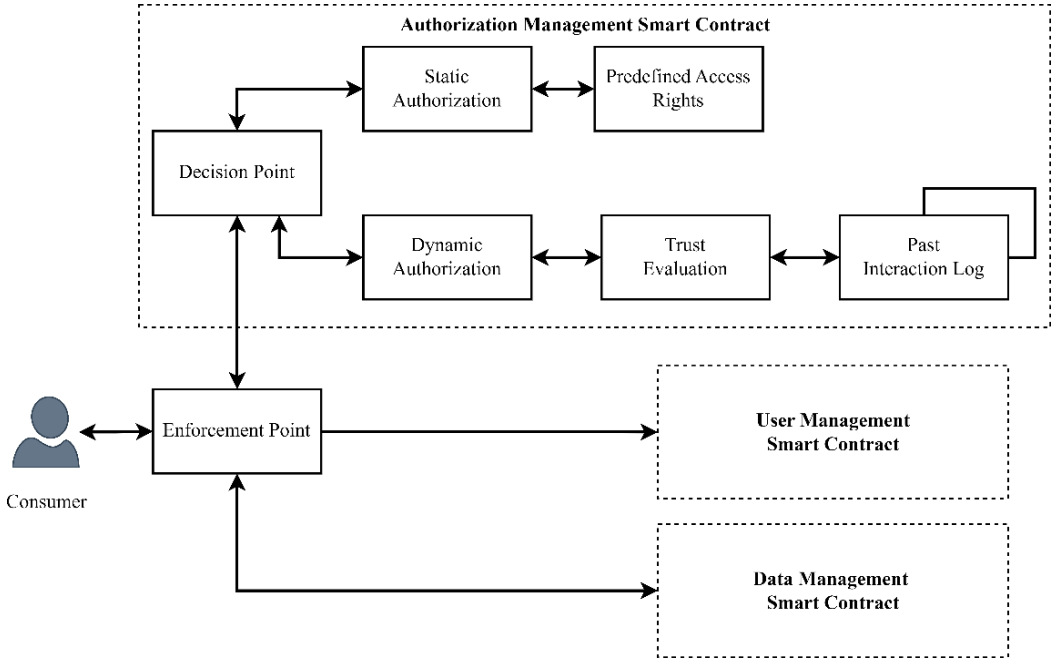

Figure 3.4: Proposed Authorization Model

The flowchart of the authorization process is depicted in Figure 3.5. Initially,the access query is submitted by the resource consumer, and following it, access policy verification is performed by the decision point method encompassed by authorization management smart contract. If policy verification is failed, then access is denied otherwise token generation process commences. Once the token is generated, access to the resource is initiatedalong with the token. Subsequently, on the successful verification of the token request is approved. However, if the token cannot successfully verified then the access to the desired resource is denied.

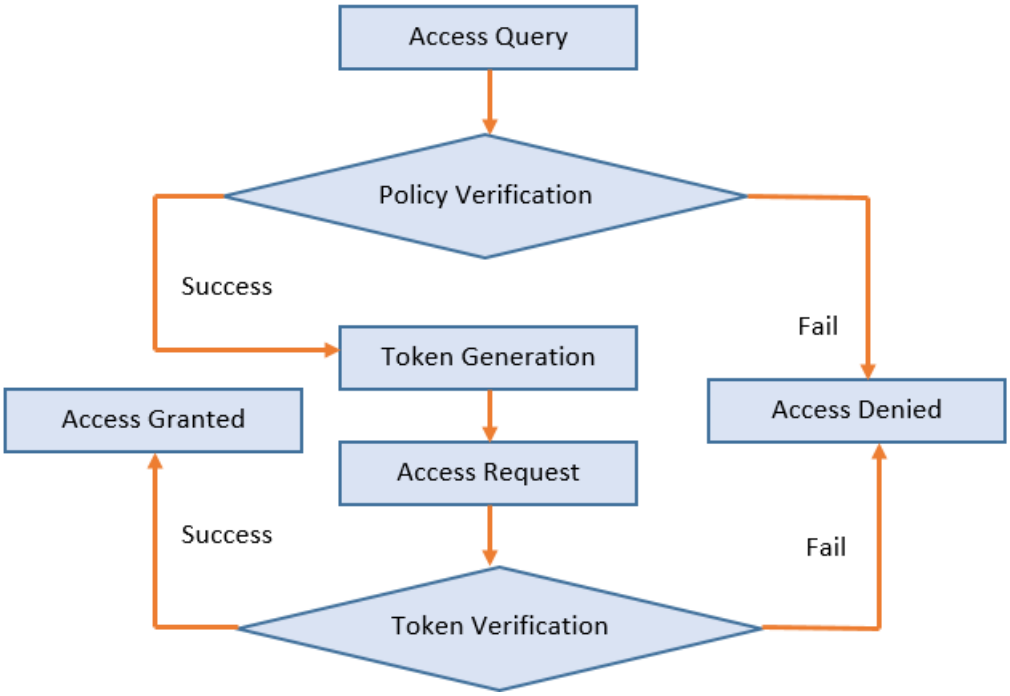

Figure 3.5: Authorization Flowchart

3.4 Algorithm and Implementation

This section presents pivotal algorithms designed to enable secure data sharing within the IoT ecosystem. The proposed system revolves around three core authorization algorithms detailed below. Furthermore, an authentication algorithm for user/device verification and a storage management algorithm is incorporated to simulate and enforce the proposed technique, ensuring robust authorization for secure data sharing among IoT smart devices and users.

This Consumer Authorization algorithm assesses whether a specific consumer is authorized for requested action on the given resource or not. It first checks if the action matches the action associated with the resource object (OBJ). If there is no match or if the resource object doesn't exist, it invokes a Review Authorization function. If either of these conditions results in a positive authorization, the algorithm allows the action; otherwise, it denies it. This algorithm represents a simplified access control process and is often part of a more comprehensive access control and security system in an IoT environment.

Algorithm 3.1: *Consumer Authorization*

Input: *ConsumerID, OwnerID, ResourceID, Action*

Output: *"Allow" or "Deny"*

```
1   check1=check2=FALSE
2   Create object OBJ corresponding to the given ConsumerID, OwnerID,
    and resourceID
3   if (OBJ is not NULL)
4   |  then if (OBJ [action] ==Action)
5   |  |  then check1= TRUE
6   else if (ReviewAuthorization ( ConsumerID, OwnerID, ResourceID,
    Action))
7   |  |  check2 =TRUE
8   |  endif
9   if (check1==TRUE OR check2 ==TRUE)
10  |  return "allow"
11  else
12  |  return "deny"
13  |  endif
```

Algorithm 3.2 "Review Authorization" is a process that checks whether a specific action is authorized for a consumer to perform on a resource. It involves compatibility checks, the creation of new access policies, and obtaining consent from the resource owner. If all conditions are met, the algorithm returns TRUE, indicating authorization; otherwise, it returns FALSE, indicating denial of access. This algorithm plays a key role in a broader access control system within an IoT environment.

Algorithm 3.2: *Review Authorization*

Input: *ConsumerID, OwnerID, ResourceID, Action*

Output: *TRUE or FALSE*

```
1   Review← look into the action entry corresponding to the ConsumerID
2   if Review is found compatible with the ResourceID
3       then define a new access policy and seek consent from the OwnerID
4       if OwnerID  approves the access policy then
5           if Action matches with the action of the newly defined policy
6               then return TRUE
7           endif
8       endif
9   else
10      return FALSE
11  endif
```

Algorithm 3.3, plays a pivotal role in ensuring secure and authorized access to resources within an IoT environment, combining authentication and authorization checks to grant or deny access based on established policies. The process unfolds methodically, starting with the verification of the consumer's identity and authentication, signifying that the consumer must provide valid credentials or proof of identity to gain access to the system. Once the consumer is successfully authenticated, the algorithm proceeds to assess the authorization of the requested action. This is done by invoking a function, "ConsumerAuthorization," which likely evaluates authorization policies based on the provided ConsumerID, OwnerID, ResourceID, and Action. If the authorization check returns a positive result, allowing the action, the algorithm generates a unique identifier (UID) through hashing the relevant data elements. Subsequently, a new token is created, combining the UID, ConsumerID, and ResourceID.

This token is used to represent the consumer's authorized access. In this scenario, the algorithm concludes by returning the new token as the output.

Algorithm 3.3: *Access_Control*

Input: *ConsumerID, OwnerID, ResourceID, Action*

Output: *newTOKEN or ERROR*

```
1   Verify authentication of ConsumerID
2   if ConsumerID is authenticated then
3       if (ConsumerAuthorization (ConsumerID, OwnerID, ResourceID,
        Action) =="allow") then
4           UID←SHA256(ConsumerID, ResourceID, block. timestamp)
5           newTOKEN ← (UID,ConsumerID, ResourceID)
6           return newTOKEN
7       endif
8   else
9       return ERROR
10  endif
```

3.5 Simulation and Result Analysis

An analysis was conducted on the throughput of the smart contract methods within this architecture. The relationship between the average response time, policy count, and the count of requests is examined and visualized in Figure 3.6. The three sets of request counts considered for the experiments were: 1, 25, and 50, shown through orange, yellow, and green colors respectively. For request count 1, the average response time becomes saturated beyond 150 policy count. However, for the request count 25 and 50, the curve grows almost linearly, as the response time is linearly proportion to the number of request received. Figure 3.7 illustrates the evaluation of authorization, addPolicy, and deletePolicy processes within the chaincode (smart contract) concurrently handling access requests from N= 100, 200, 400, 600, and 800 simulated clients. Since to add a new policy, resource owner consent is needed and the transaction is required to be executed, thus it takes more time as compared to deleting policy or authorization operation.

Figure 3.6: Average response time against policy count

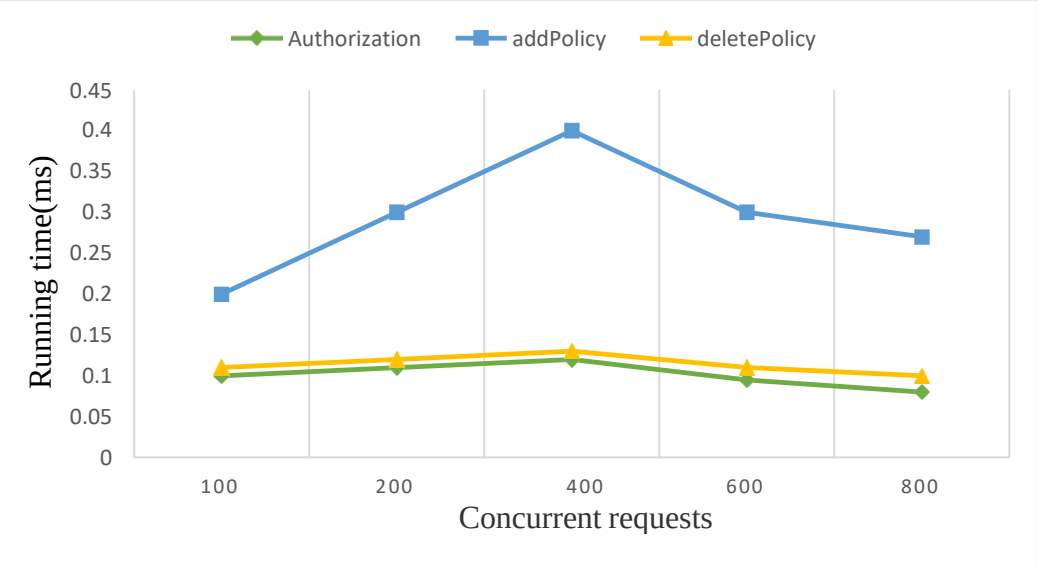

Figure 3.7: Execution time on simultaneous request

In the Figure 3.7, maximum running time can be observed for 400 simultaneous access request while for 600 simultaneous access requests and onwards, system achieves optimal performance.

3.6 Summary

In this chapter, we delved into a sophisticated two-level authorization model that harnesses the potential of Blockchain technology to combat the security

complexities within the IoT domain. Our envisioned solution heavily relies on smart contracts, a fundamental feature of Blockchain, to furnish a sturdy and expansible resolution, granting users the capability to both deploy and retrieve data from remote locations. Our emphasis lay in the development of authorization algorithms specifically tailored to facilitate the sharing of IoT data. The core of this initiative entailed addressing the limitations imposed by traditional centralized authorization systems through the advocacy of decentralized methodologies, all supported by the transformative capabilities of Blockchain technology. Furthermore, our methodology led the way in integrating adaptable authorization policies tailored to accommodate the inherently dynamic nature of IoT environments.

CHAPTER 4

SECURE IoT DATA MANAGEMENT AND SHARING ARCHITECTURE

Smart devices connected through the internet generate vast amounts of data that power numerous applications, significantly enhancing our daily routines. However, due to the sensitive nature of this data, secure sharing among these devices is imperative. The IoT environment, owing to its appeal to cyber criminals and a history of successful cyber-attacks, demands protection against unauthorized access. Blockchain technology emerges as a promising solution to address these security challenges effectively. Yet, storing the sheer volume of data generated by smart devices proves inefficient within the Blockchain due to the rapid pace of data collection and the transaction confirmation speed in the Blockchain network.

Technological advancements such as Artificial Intelligence (AI), Big Data, and cloud-based computing have enabled the generation of vast data through conventional or mobile devices. However, many IoT systems still rely on centralized storage, posing inefficiencies and limitations. Centralized systems are susceptible to data censorship, tampering, and hacking, compromising their reliability and trustworthiness in IoT contexts. Blockchain offers a secure, distributed, and tamper-resistant approach to data management. However, in the IoT context, challenges arise due to the rapid pace of data generation and transaction validation speed within Blockchain networks. The extensive real-time data generated when numerous smart devices connect to the IoT network presents scalability challenges. The existing consensus protocols and transaction validation rates in Blockchain networks struggle to match the speed of data creation in the IoT environment.

4.1 Introduction

This chapter focuses on integrating Blockchain and the Inter-Planetary File System (IPFS) to enable data storage in a distributed manner and implement stringent access controls, allowing authorized entities exclusive access to recorded data.

Access policy definitions for secure data sharing and cryptographic hash content are stored over the Blockchain network to ensure data integrity and security.

The Interplanetary File System (IPFS) functions as a peer-to-peer distributed file system that relies on content-based retrieval of records. It utilizes cryptographic hash concepts similar to Uniform Resource Locators (URLs) on the web. IPFS is often referred to as a version-controlled structure since it maintains past versions of files that have changed over time. Existing storage models and sharing systems encounter challenges related to safety, scalability, and reliability, all of which IPFS addresses. In IPFS, identical files possess identical hashes, ensuring originality and eliminating redundancy. To maintain consistency of recorded data across peer nodes in the distributed storage system, cryptographic hashes are distributed among all peers. Additionally, the content-based addressing scheme employed by IPFS results in high throughput.

This chapter proposed a solution aimed at mitigating the limitations of both centralized storage systems and Blockchain storage systems. The integration of Blockchain and IPFS ensures data reliability and availability, overcoming these limitations. In this work, we establish a connection between Blockchain and the IPFS to facilitate data storage in a distributed manner and enable querying of this recordeddata. Initially, we introduced an architecture for data storage that leverages Blockchain and IPFS. This architecture facilitates the automatic encapsulation and parsing of sensor-generated data, including images, text, and video. Subsequently, the cryptographic hash representing this data is stored on the Blockchain, while the actual data is uploaded onto IPFS.

4.2 Motivation and Contributions

In traditional central data storage models, a major concern arises when data is stored on a central server, leading to potential loss of ownership and privacy breaches for data owners. Moreover, these centralized systems are susceptible to being a single point of failure. To counter these issues, this paper explores the integration of

Blockchain technology with the IPFS and proposes an architecture tailored for managing and sharing healthcare data.

Blockchain-based applications rely on smart contracts to achieve consensus on transaction execution. These applications encompass various nodes executingsmart contracts, storing Blockchain data, and handling transactions. However, operational challenges arise when dealing with extensive data files. Key concerns include data replication across nodes, resource-intensive mining processes, and theinefficiency of storing large files directly on the Blockchain.

The limitations posed by the size restrictions of each block in the Blockchain necessitate breaking down files and storing them off the Blockchain. Additionally, information about the arrangement of these files also demands space. While smart contracts can manage and access this additional data, transmitting and recording substantial files through smart contracts triggers increased execution costs at each node, resulting in higher gas costs.

Storing large files on the Blockchain incurs additional costs associated with working with miner nodes. Large files require substantial data transmission, processing, and storage by nodes, demanding high bandwidth and increased Blockchain storage. These observations highlight that Blockchain isn't the optimal platform for sharing or storing extensive data files.

Fortunately, file-sharing frameworks can leverage Blockchain alongside IPFS, capitalizing on Blockchain technology while keeping file sizes manageable for efficient processing and scalability. This integration allows the benefits of Blockchain to be harnessed without the constraints of handling large data files directly within the Blockchain.

Similar to public Blockchain, data stored on IPFS can be queried and accessed by any entity connected to the IPFS network. This presents a significant challenge for Blockchain-based applications dealing with extensive files containing sensitive or private information. Consequently, there is a necessity for a framework based on Blockchain and IPFS that can restrict access to authorized entities only. This

framework must encompass the capability to define and deploy access policies on the Blockchain. Additionally, all interactions with IPFS for data storage or access requests should undergo evaluation to ensure compliance with established access policies. IPFS then collaborates with smart contracts to enforce these policies. Depending on the outcome of the policy evaluation, IPFS either grants or denies the requested service. Such a framework should enable users to register new files on IPFS and manage predefined access policies by initiating and transmitting transactions through smart contracts. The interaction between IPFS nodes and smart contracts enables IPFS to rely on the latter for policy evaluation and enforcement, ensuring that access is only granted under the predefinedpolicies. The significant contributions of this article are as follows:

- A Blockchain-powered, IPFS-based resource access management framework is presented for secure resource sharing in the healthcare domain.
- IPFS-based data storage model offers a high availability of uncensored data.
- This scheme facilitates two-phase authorization for data access- static authorization (based on predefined access policies) and dynamic authorization (based on resource access behavior)

4.3 Proposed Work

This section comprises several subsections, starting with addressing different facets of the system model and design followed by an exploration of the data management model. Subsequent sections cover an examination of the policy model within the proposed architecture following the details of the architectural design and system interaction and workflow. Finally, a comprehensive discussion unfolds, elucidating the integration and interplay of the three smart contract types within the smart contract layer.

4.3.1 System Model and Design

This section encompasses several sub-sections, each delving into specific facets of the system model and design. It commences with a discussion on the data management model, elucidating how data is handled within the proposed

framework. Following this, the policy model of the architecture is explored, detailing the principles and guidelines governing the system's operation. Subsequently, attention is directed to the architecture design, providing insights into the structural elements and organization of the proposed system. The section then proceeds to elucidate the system's interaction and workflow, outlining the processes and sequences involved in its operation. Finally, a comprehensive exploration of the smart contract layer is presented, covering all three types of smart contracts and delineating their interconnections within the system.

4.3.2 Data Management Model

This section provides a comprehensive explanation of how IoT-generated data is categorized, segmented, and managed before being accessed by authorized entities in the IoT ecosystem. The entirety of IoT-generated data undergoes management through a hierarchical structure resembling a tree, with distinct subcategories for various classes of data resources, such as images, videos, and text files. Among these classes, some data files are notably smaller in size, necessitating aggregation into data blocks. These aggregated data blocks collected over a specific time frame (for instance, a day), are then assembled into data packages. These packages are subsequently directed for storage on the IPFS.

The proposed framework for data storage and management relies on a model based on Blockchain and IPFS. A visual depiction illustrating the storage structure of this proposed architecture is presented in Figure 4.1.

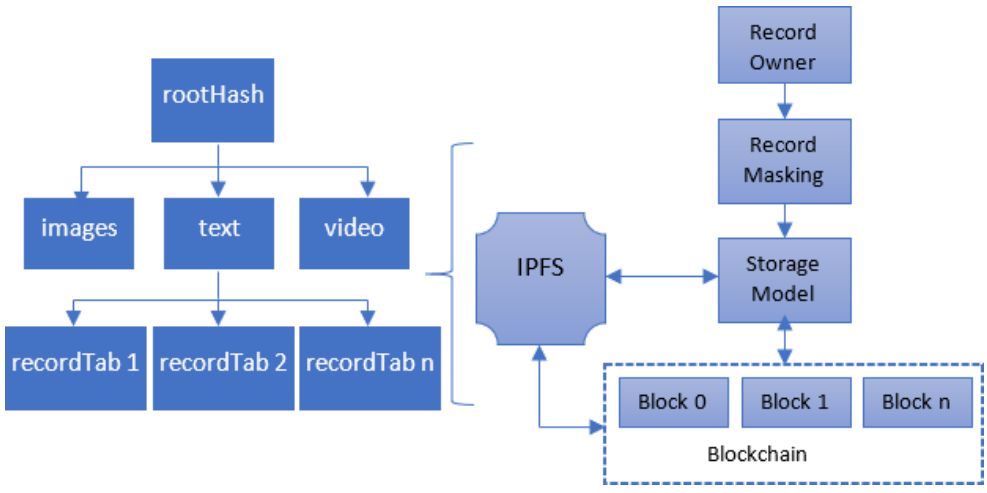

Figure 4.1: Storage structure of proposed architecture

4.3.3 Policy Model

As technology advances, so do the vulnerabilities associated with systems and data. Security measures, particularly access policies, serve as critical safeguards, shielding systems from potential threats and ensuring that only legitimate entities are granted appropriate access rights. The proposed architecture designed for secure data sharing within an IoT environment institutes a two-phase authorization process before granting access to protected resources.

The initial phase of authorization is static, verifying pre-defined access rights. The subsequent phase encompasses dynamic elements, leveraging the trust score of the requesting entity. This trust score is an accumulation of the entity's historical access interactions, prioritizing recent interactions with higher weightage while assigning lesser weightage to older interactions. Upon successful verification of the entity, the Blockchain system responds by providing an encrypted recordHash alongside the requester's public key, a timestamp, and a signed session key. This cryptographic exchange ensures secure communication between entities within the system.

4.3.4 Architecture Design

In this section, we introduce the Blockchain-powered IPFS-based Resource Access Management (BI-RAM) framework, outlining its constituent components:Patient, Hospital, Insurance Agency, Management Hub, Device Manager, and Blockchain network. Within this framework, the patient, hospital, and insurance agency are identified as smart entities. The management hub operates as an intermediary, facilitating communication between these smart entities and the Blockchain network.

Each smart entity within this system is mandated to register itself under at least one device manager, which is an integral part of the Blockchain network. These device managers are responsible for defining access privileges for the enrolled smart entities, ensuring that access rights are established with their explicit consent. All data generated by these smart entities are stored on the IPFS, with their data hash

(commonly known as content hash) being uploaded onto the Blockchain. A visual representation delineating the comprehensive architecture of this proposed system is illustrated in Figure 4.2.

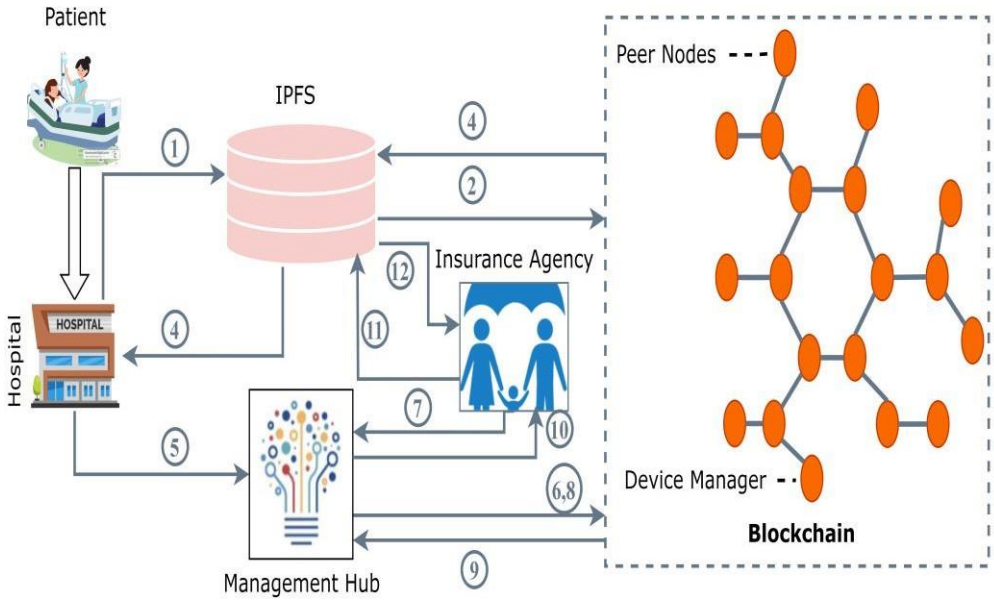

Figure 4.2: Proposed architecture

4.3.5 System Interaction and Workflow

The operational sequence of the proposed system is depicted in the sequence diagram showcased in Figure 4.3. The series of interactions unfolds as follows:

- The patient initiates an appointment for treatment.
- The hospital submits a query to upload health records onto the IPFS, detailing patient diagnosis and treatment.
- IPFS verifies the legitimacy of the hospital's identity; upon validation, records are uploaded, and the corresponding hash record is provided. If the identity verification fails, the request is denied, with Blockchain conducting the verification process.
- To address privacy concerns, the hospital encrypts the hash record and forwards it to the management hub.
- The management hub relays the query to the Blockchain network.
- Blockchain conducts identity verification and proceeds with record uploading.

- Simultaneously, the patient seeks health insurance for their medical expenses.
- The insurance agency queries related data through the management hub.
- The management hub converts the request into a Blockchain action and forwards it to the Blockchain network.
- Blockchain executes the action, verifying the authorization of the insurance agency. Upon successful verification, an encrypted hash record with a timestamp is returned; otherwise, access is denied.
- The management hub delivers the message to the insurance agency in CoAP format.
- The insurance agency decrypts the message to obtain the hash record. A subsequent query is dispatched to IPFS for the actual record.
- IPFS verifies the validity of the timestamp and hash data upon receiving the query. Successful validation results in the retrieval of the relevant record; otherwise, access is denied.
- After obtaining the requested record, the insurance agency cross-verifies its hash value with the hash record obtained from Blockchain for validity. It proceeds with claim settlement and submits an interaction score to Blockchain via the management hub.
- Subsequently, upon claim settlement, the patient returns an agency score to Blockchain.
- Blockchain records both trust scores in the state database for future interactions and data-sharing processes.

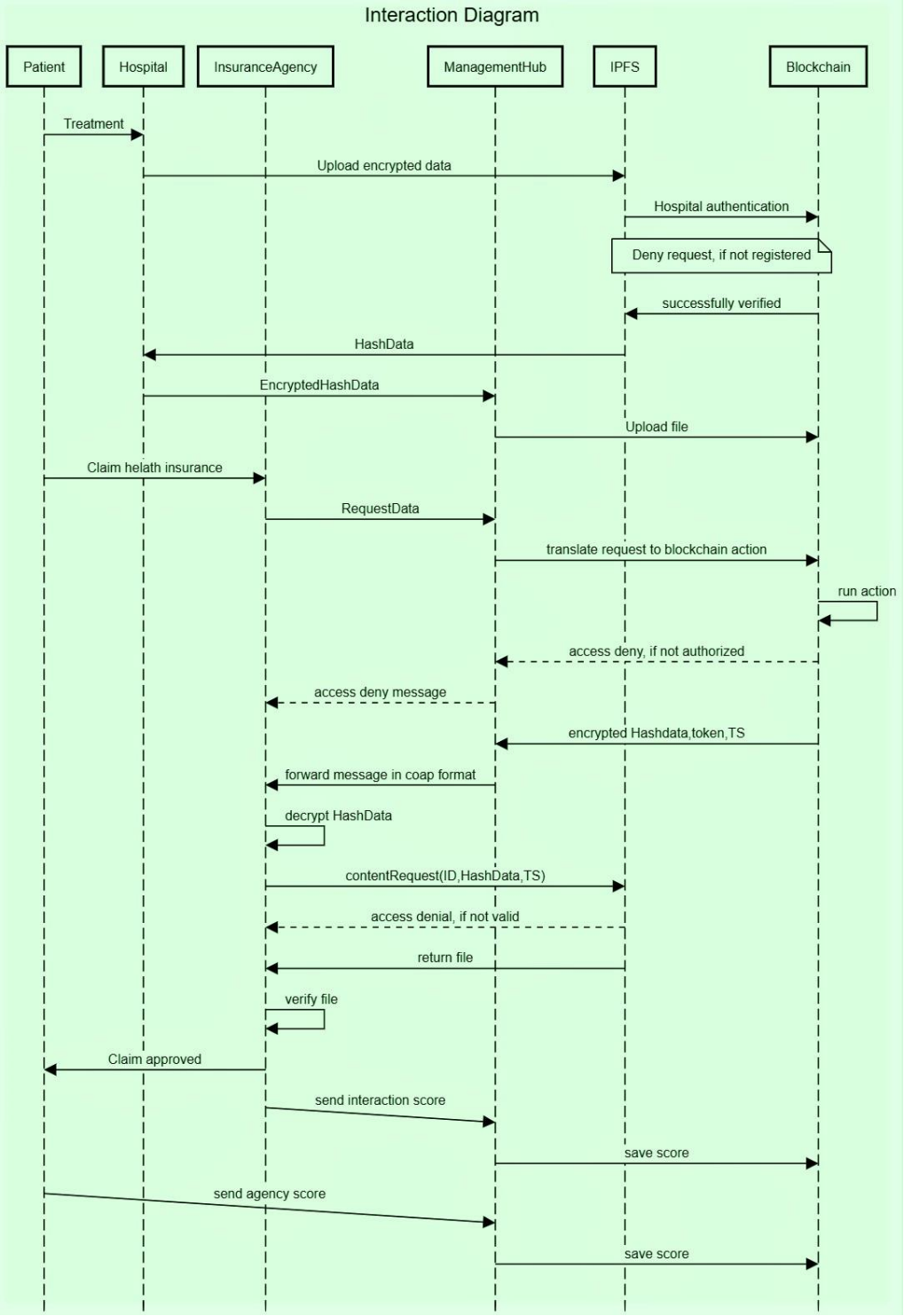

Figure 4.3: Data sharing sequence

The interconnection of vital components of the proposed solution is depicted in the Figure 4.4

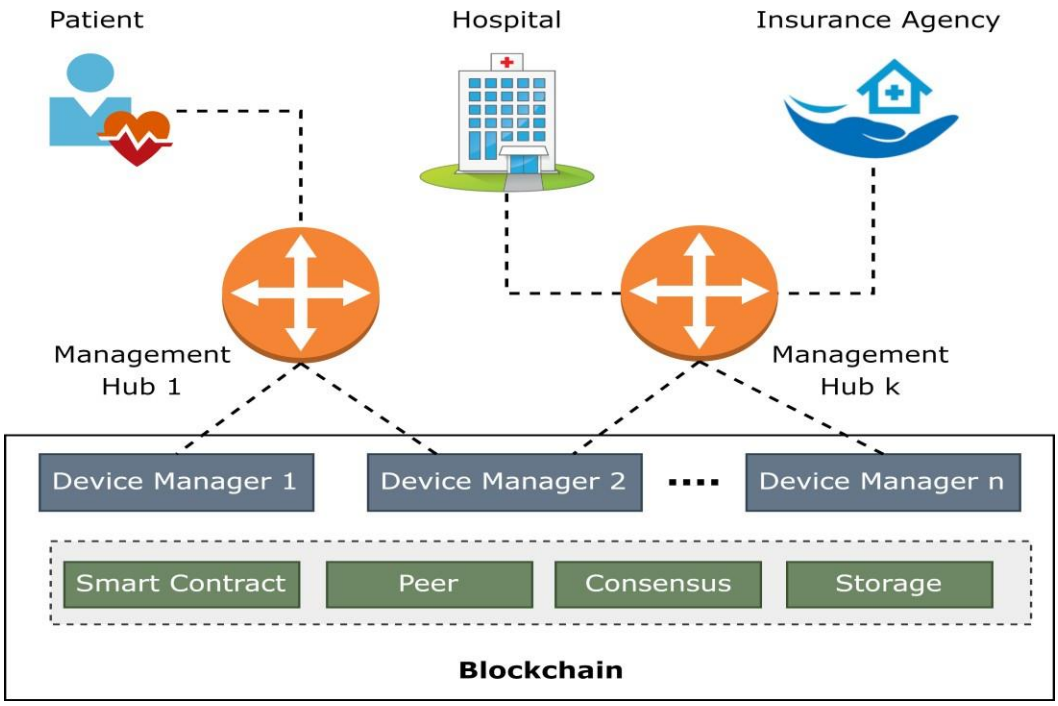

Figure 4.4: Interconnection of Key Components

4.3.6 Smart Contract Layer

The proposed framework relies on three key smart contracts to facilitate the sharing of records among the smart entities: Access Rights Contract (ARC), Entity Contract (EC), and Trust Contract (TC). Each contract serves a distinct purpose within the system.

- ARC takes charge of implementing access management policies. It interacts with EC to validate the registration of smart entities on the Blockchain and provides methods to manage the upload and download of records from the IPFS.
- EC, complementing ARC, handles the registration of smart entities onto the Blockchain. It facilitates the upload and download of records from the IPFS and provides essential methods for identity verification upon access queries.
- TC plays a crucial role in assigning trust scores to participating entities based on their historical behavior concerning data sharing. These trust scores are essential components used in determining access rights.

Whenever an access query is initiated, ARC calls upon the relevant method within EC to authenticate the identity of the requested entity. Subsequently, TC is consulted to retrieve the trust score associated with that entity. Based on this evaluation, ARC determines the appropriate access rights for the requested entity and furnishes the corresponding outcome.

A detailed representation of the employed smart contracts within this proposed framework is depicted in Figure 4.5, accompanied by associated algorithms elucidating their functionalities.

a) Entity Contract (EC)

EC is tasked with managing all entity-related information crucial for the entity identification process. It encapsulates several essential methods:

- **registerEntity:** This method is responsible for capturing and storing the entity's information at the time of its registration under a device manager within the system.
- **getEntity:** Upon query by the Blockchain node, this method retrieves and provides the relevant information pertaining to a specific entity.
- **identifyEntity:** Used for the authentication process, this method ensures the verification of the requested entity, confirming its legitimacy within the system. Additionally, EC handles the incorporation of hash records onto the Blockchain and retrieves hash records from the Blockchain through the following functions:
- **addHash**: This function is responsible for including hash records onto the Blockchain, ensuring their proper storage and verification within the system.
- **fetchHash:** Designed for retrieval purposes, this function retrieves specific hash records from the Blockchain when required.

b) Trust Contract (TC)

The main objective of TC revolves around assessing the historical behavior of entities involved in data-sharing activities. It executes four distinct functions to fulfill this purpose:

- **setScore:** This function calculates the trust score of participating entities by

analyzing their patterns of access within the system. It determines the trustworthiness of entities based on their historical interactions.

- **setPenalty:** In case of undesirable access patterns observed from an entity, this function imposes a fine or penalty as a consequence of their behavior within the system.
- **getPenalty:** getPenalty is utilized to fetch the imposed penalty or fine on a specific entity.
- **getScore:** getScore retrieves the trust score assigned to an entity, enabling examination and evaluation of their historical behavior within the system.

c) Access Rights Contract (ARC)

ARC serves as the central contract responsible for enforcing access control measures among smart entities, ensuring that only authorized entities can share and access records across the network. Upon receiving an access query, ARC interacts with EC to gather identity information and retrieve the trust score from TC. It then evaluates predefined access rights alongside the cumulative trust score of the requesting entity. Based on this assessment, it generates either “allow” or “deny” result. ARC encompasses multiple methods designed to accomplish these objectives:

- **newPolicy:** This function is responsible for introducing and incorporating new access rights into the system, ensuring the inclusion of additional authorization protocols.
- **removePolicy:** Used for administrative purposes, this function facilitates the removal or deletion of existing access rights within the system.
- **modifyPolicy:** To adapt to changing requirements, this function enables the modification or adjustment of existing access rights, allowing for flexibility in the authorization process.
- **validateAccess:** This function plays a crucial role in verifying requested access rights, ensuring that the permissions requested align with the predefined policies and the entity's trust score, ultimately determining whether access should be granted or denied.

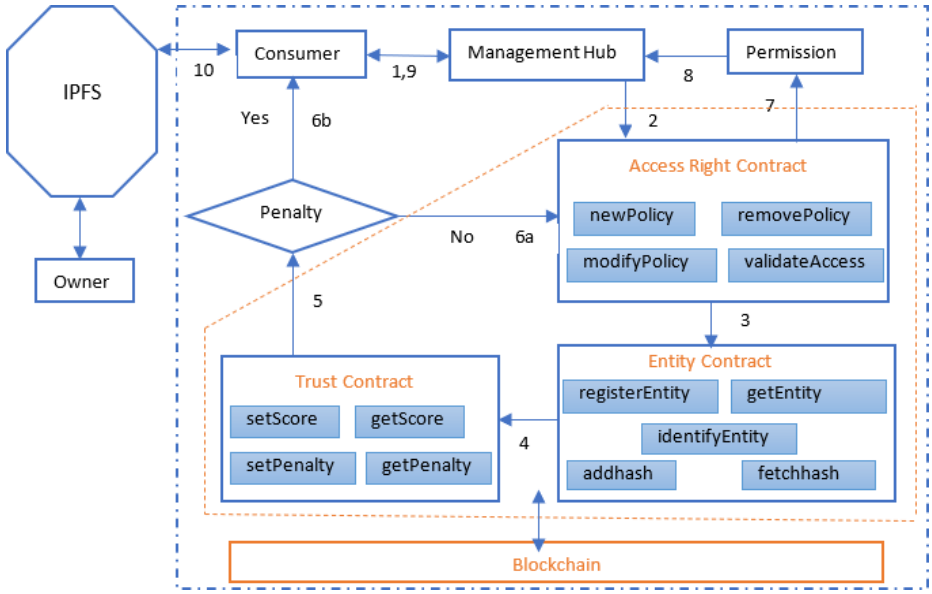

Figure 4.5: Smart contract of the proposed architecture

The pseudo code of "Uploading record to IPFS", "Uploading recordHash to Blockchain", "Retrieving recordHash from Blockchain", and "Download the record from IPFS" are discussed below.

Algorithm 4.1: *Uploading record to IPFS*

Input: *$Hospital_{ID}$, $Patient_{ID}$, record, Context, protocol*

Output: *success, fail*

```
1   ipfs←IpfsAPI ('ipfs', HospitalID, protocol) //connect to ipfs
2   result← verify (HospitalID,PatientID)
3   if result = = FALSE then
4      return 'fail'
5   else
6      buff← Buffer (record)
7      recordType←ipfs. getType(record)
8      ipfs.dag.put (buff, recordType, err)
9      if (err) then
10        return 'fail'
11     endif
12     return 'success'
```

13	***endif***

Algorithm 4.1 begins by establishing a connection to the IPFS network using the specified HospitalID and protocol. It then verifies the authenticity and authorization between the HospitalID and PatientID. Upon successful verification, it converts the provided record into a buffer and identifies its type before uploading it to the IPFS network as a Directed Acyclic Graph (DAG) node. The algorithm provides feedback as 'success' if the record upload completes without errors. If any errors occur during the upload process or if the verification fails, it returns 'fail,' indicating an unsuccessful attempt to upload the record onto the IPFS network.

Algorithm 4.2: *Uploading recordHash to Blockchain*

Input: *$Hospital_{ID}$, $Patient_{ID}$, Context, recordHash, SK*

Output: *result (success or fail)*

1	*EncryptedHash ← SHA256(recordHash, SK)*
2	*Message ← Encapsulate (EncryptedHash, $Patient_{ID}$, $Hospital_{ID}$)*
3	*err ← uploadQuery (Message, $Hospital_{ID}$)*
4	***if*** *err = = NIL* ***then***
5	***return*** *success*
6	***else***
7	***return*** *fail*
8	***endif***

Algorithm 4.2 generated an encrypted hash of the provided recordHash using the SHA256 hashing algorithm and a secret key (SK). Next, it encapsulated this encrypted hash along with the PatientID and HospitalID into a message. Subsequently, the algorithm attempted to upload this constructed message to the Blockchain via the uploadQuery function, specifically linked to the HospitalID. If the upload process encounters no errors (NIL), the algorithm returns 'success,' indicating the successful inclusion of the record hash onto the Blockchain. Conversely, if any errors arise during the upload process, the algorithm returns 'fail,' signifying an unsuccessful attempt to upload the record hash onto the Blockchain.

Algorithm 4.3: *Retrieving recordHash from Blockchain*

Input: *$Owner_{ID}$, Context*

Output: *recordHash (success or fail)*

```
1  str, err ←queryRecord (Consumer_ID, Patient_ID, Context)
2  if err = NIL then
3      TransactionID←get transaction_ID (Patient_ID, Context)
4      RecordHash←fetch (Transaction_ID)
5      return recordHash. Success
6  else
7      return fail
8  endif
```

Algorithm 4.3 retrieved a record hash from the Blockchain based on specified input parameters. Initially, it initiated a query to retrieve a string ('str') related to the ConsumerID, PatientID, and Context using the queryRecord function. If this query process encounters no errors (NIL), the algorithm proceeds by acquiring the TransactionID through the get transaction_ID function, utilizing the PatientID and Context parameters. Subsequently, it fetched the record hash associated with the obtained TransactionID. If these retrieval processes succeed without errors, the algorithm returns the retrieved recordHash and signifies 'success.' However, if any errors occur during the query or retrieval operations, the algorithm returns 'fail.'

Algorithm 4.4: *Download the record from IPFS*

Input: *recordHash, token*

Output: *record, NULL*

```
1  ID_c, TS_d ← Dec(token)
2  α ← verify (ID_c, TS_d)
3  if α = = FALSE then
4      return NULL
5  else if TS >current_time then
6      return NULL
7  else
```

```
8        rootHash ← fetch (recordHash)
9        objectHash ← fetch (rootHash)
10       data ← get (objectHash)
11       record ←E_pk (data)
12       return record
13     endif
14   endif
```

Algorithm 4.4 is designed to retrieve a record from the Inter Planetary File System (IPFS) based on provided inputs. Initially, it extracts the IDc and TS_d from the token by employing a decryption operation. Following this, the algorithm verifies the authenticity of the extracted IDc and TS_d. If this verification fails, the algorithm returns NULL, indicating an unsuccessful attempt to retrieve the record. Subsequently, it evaluated the timestamp (TS) against the current time, returning NULL if the TS is greater than the current time, signifying an expired token. If these conditions are not met, the algorithm proceeds by fetching the rootHash associated with the recordHash, followed by retrieving the objectHash linked to the rootHash. After obtaining the data corresponding to the objectHash, the algorithm decrypts it using E_{pk}, resulting in the retrieved record. Finally, it returns the retrieved record if all previous checks and operations are successful and NULL otherwise.

4.4 Simulation and Result Analysis

This section explored the analysis of the obtained results, discussing and interpreting the outcomes within the context of established solutions within the domain. By comparing and contrasting with notable existing techniques,the section aims to highlight the advantages and benefits of the proposed solution.

4.4.1 Results

The depicted Figure 4.6, showcases the data dispersion time comparison between the original p2p model and the proposed scheme. The bandwidth and average distribution time (in seconds) are represented on the x-axis and y-axis respectively. The proposed scheme and original p2p approach are illustrated through orange and

blue colors respectively. While the proposed scheme exhibits slightly longer dispersion times compared to the original p2p approach, this is due to the fact that our scheme performed encryption before distributing data over the storage system. Therefore, our work guarantees enhanced record security via its decentralized access control mechanism. Notably, experiments conducted on larger file sizes, up to 1GB, yielded similar outcomes, highlighting that the encoding and decoding costs have a negligible impact on the overall running costs of the model. This underscores the suitability of the proposed scheme within the IoT environment. The integration of a permissioned Blockchain network with a distributed file system (IPFS storage system) significantly augments the system's performance across multiple parameters. IPFS operates as a peer-to-peer hypermedia protocol and distributed file system, interlinking various computing systems. It aims to store versioned records on decentralized media. In contrast to conventional storage schemes utilizing location-based addressing, IPFS employs content-based addressing. Each file on IPFS possesses a cryptographic hash that facilitates locating the actual content within the network.

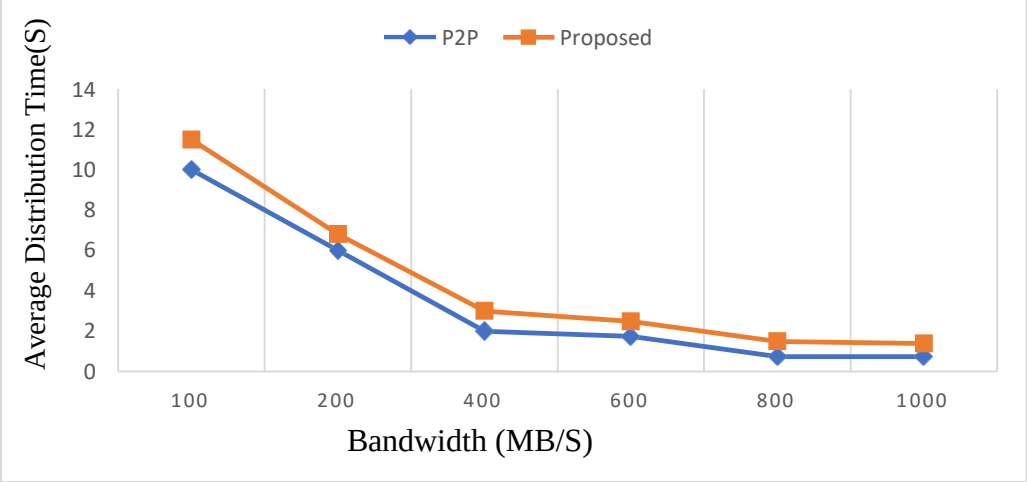

Figure 4.6: 10MB data distribution

The graphical representation in Figure 4.7 demonstrates the relationship between IoT data uploaded per minute and IoT data processed per minute. The orange line corresponds to the framework incorporating Blockchain alone, while the blue line represents the framework integrating both Blockchain and IPFS.

It is evident fromthe graph that the framework employing both Blockchain and IPFS excels in processing substantial volumes of IoT data, while the framework utilizing only Blockchain reaches a saturation point when handling such extensive IoT data. When the IoT generated data are in small size there is not much difference in the two results. However, this difference is visible when data generation rate of the IoT system is quite high, because in the proposed scheme only the hash value of the actual data (smaller in size) is uploaded on the Blockchain.

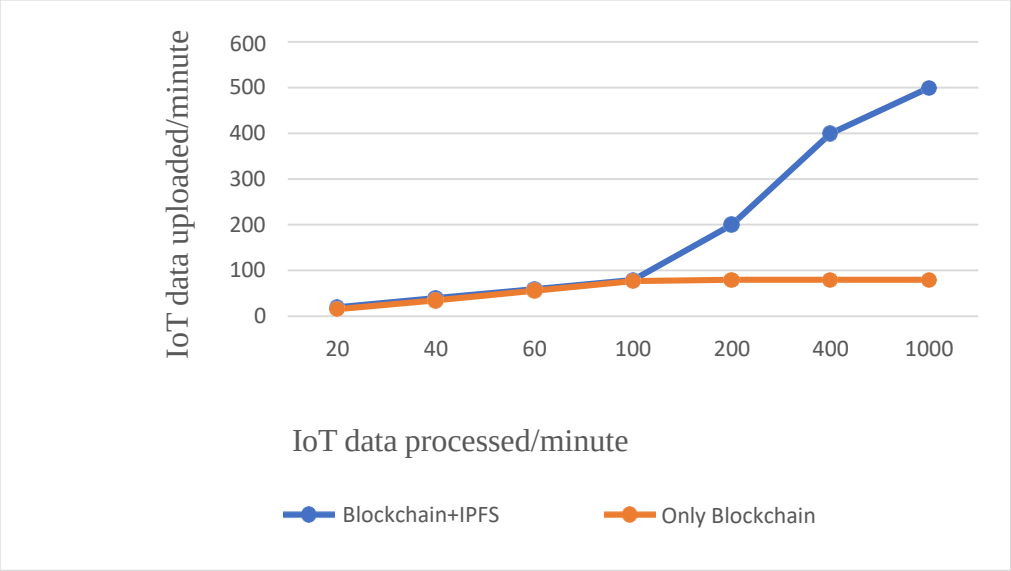

Figure 4.7: Storage rate comparison

We have evaluated the running time and average running time of EC and APC methods against multiple queries respectively in Figure 4.8, Figure 4.9, Figure 4.10, and Figure 4.11, where consumers are set to 50, 100, 200, 400, and 800. The addHash method and fetchHash method are depicted through the blue line and orange line respectively in Figure 4.8 and Figure 4.9. It is evident that the time taken to add a new hash is significantly higher than the time taken to fetch the existing hash values. There is a minimum 142% to a maximum of 200% rise in running time for addHash method against the fetchHash method as shown in Figure 4.8.

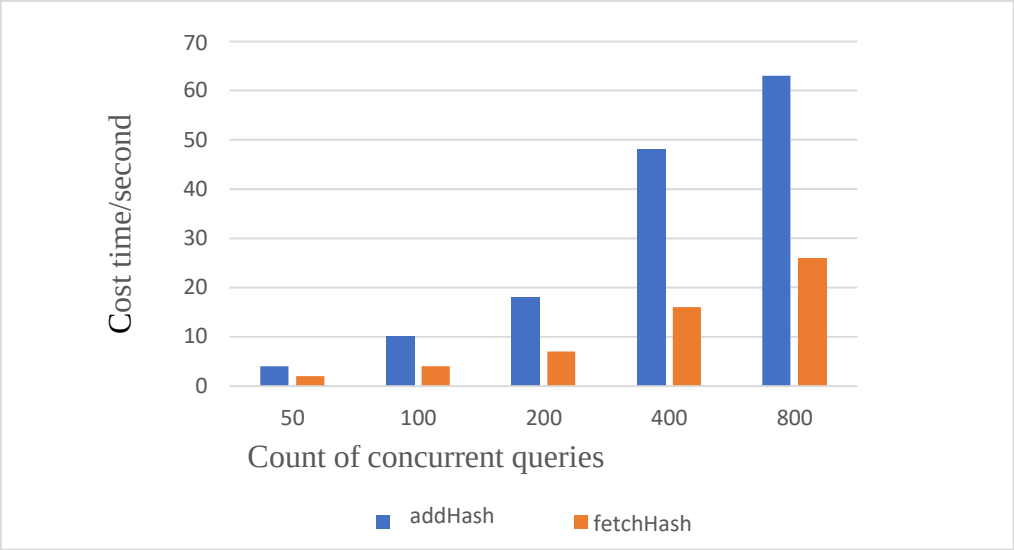

Figure 4.8: Running time of EC's methods

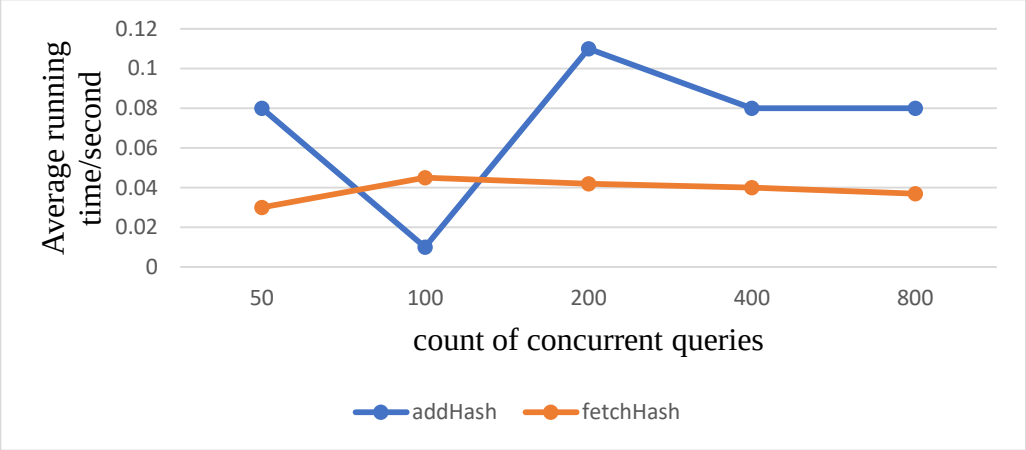

Figure 4.9: Average running time of EC's method

Writing new hash and new policy or modifying existing policy incur more time as compared to reading hash or policies. Retrieving any sort of information (fetching hash value or policy) from the Blockchain is faster as no transaction is involved in the process and therefore almost a saturated curve can be seen in Figure 4.9, depicting fetchHash method. Moreover, transaction within the Blockchain network takes time to get approved by the network thus adding new policies or new hash incur more time as comparedto fetching information from the Blockchain network. Similarly, various methods of APC smart contract are evaluated with respect to concurrent queries and running time and average running time as shown in Figure

4.10 and Figure 4.11 respectively. The methods newPolicy, modifyPolicy, removePolicy, and validateAccess are shown through blue, orange, gray, and yellow color as illustrated in Figure 4.10 and Figure 4.11.

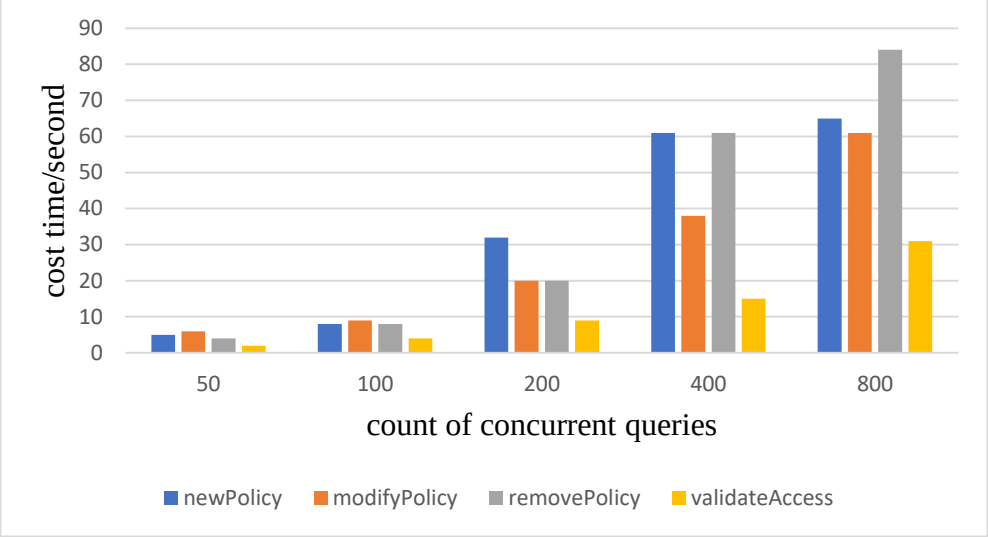

Figure 4.10: Running time of APC's methods

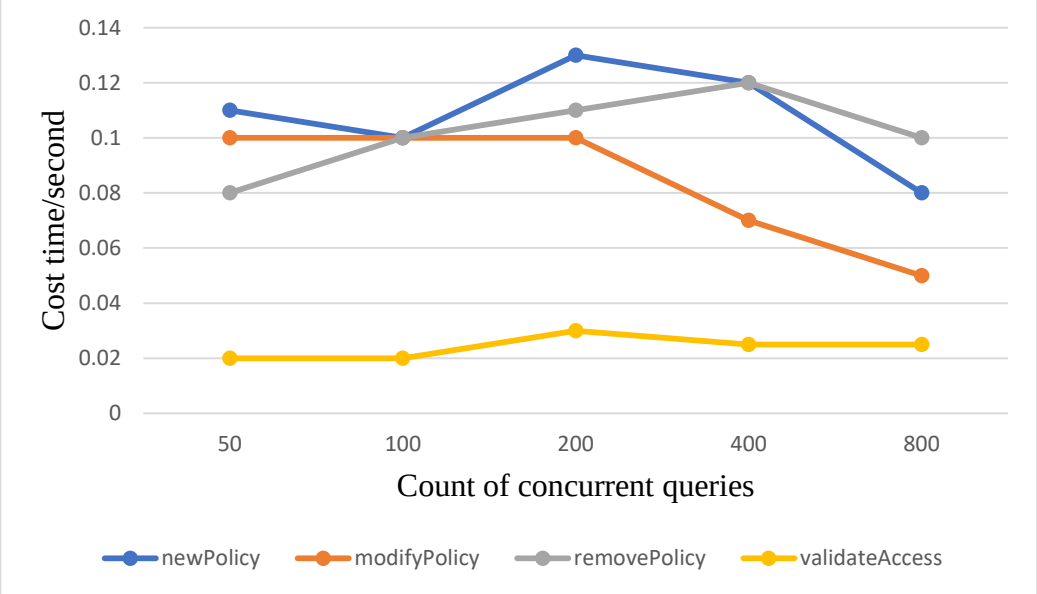

Figure 4.11: Average running time of APC's methods

From these two figures, it is evident that the validateAccess method runs quite faster than the other methods. This is achieved from the fact that, to run validateAccess method neither smart contract is deployed nor any transaction is executed. Additionally, after 200 simultaneous access requests, a fall in running time is witnessed

for newPolicy and modifyPolicy methods, while the same is observed for removePolicy method after 400 access requests.

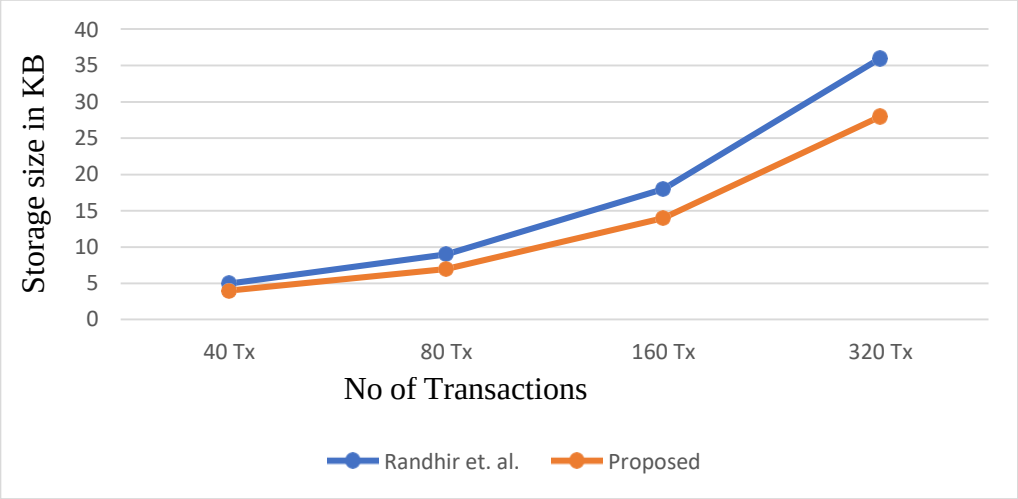

Figure 4.12: Off-chain storage size

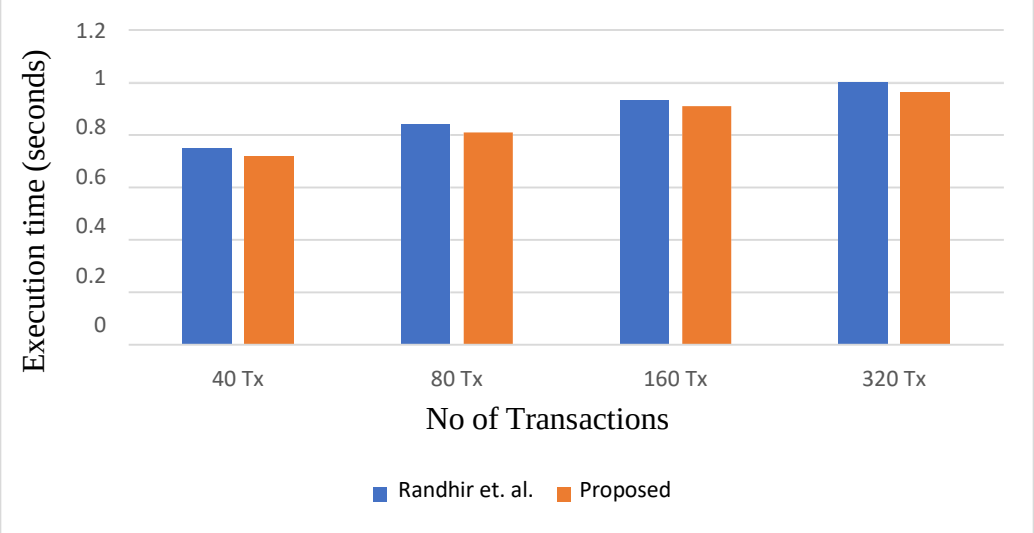

Figure 4.13: Execution time for transactions upload

The storage size needed for off-chain storage on IPFS with different transaction volumes (40, 80, 160, and 320 transactions) in both, our proposed method and the approach presented by Randhir et. al. [114] are demonstrated in Figure 4.12. This visualization vividly showcases the superior performance of our proposed method compared to the solution presented in [114]. Additionally, for the selected set of transactions, our approach took almost 25% to 28% less storage size (in KB) as compared to [114], and this is due to the selection of different encryption algorithm in both the approaches. Furthermore, the

upload execution time for multiple transactions is illustrated in Figure 4.13, showcasing the performance contrast between our approach and the one presented by Randhir et. al. [114]. Once again, the slightly better running time is achieved by our approach is due the selection of encryption algorithm applied on the IoT generated data.

4.4.2 Security Analysis

We conducted a comprehensive security assessment of the proposed system, rigorously examining its adherence to core security principles, including Confidentiality, Integrity, Availability, Authorization, and non-repudiation (CIAAN) model. The integration of Blockchain (BC) technology within the proposed architecture enabled us to effectively fulfill the stipulated security requisites outlined in Table 4.1 based on the CIAAN model.

Table 4.1: Security parameters and description

Security Principle	Description
Confidentiality	Transport Layer Security (TLS) is employed to ensure secure communication between the user/IoT device, the Device Manager, and the Blockchain. TLS encrypts the data transmitted over the network, providing confidentiality and integrity, thereby enhancing the overall security of the communication process.
Integrity	To guarantee the integrity of both the platform and data during transit, we have employed the SHA-256 hash function. By using SHA-256, we can verify the integrity of the transmitted data, ensuring that it remains unchanged and untampered with throughout its journey.
Availability	By incorporating decentralized system for enforcing access policy and data storage model, censorship on data is eliminated and thus ensured a better data availability for authorized users.

Authorization	Authorization is implemented by the methods of access right contract and trust contract in the smart contract layer.
Non-repudiation	To attain non-repudiation, digital signature mechanism is employed. Each user is required to cryptographically sign their respective transactions. This ensured that the origin of the transaction can be verified, andonce signed, the user cannot deny their involvement or the authenticity of the transaction.

4.5 Summary

In this study, we have proposed an access management system that integrates Blockchain and IPFS, resolving numerous limitations in existing IoT data-sharing solutions. The IPFS-based storage addressed issues like single point of failure and data censorship, providing high data availability, reduced storage costs, and improved throughput. Notably, our framework separated IoT devices from the Blockchain network, significantly reducing communication and computational overhead. The management hub enhances scalability by enabling multiple IoT devices to connect with the Blockchain. To realize this, we have deployed three smart contracts: ARC, EC, and TC, each serving a specific role. ARC governs data access, EC manages entities registration and data handling, and TC evaluates entities' historical access behavior and imposes penalties for improper access. Our simulations validated the system's proficiency in large-scale data processing. We have also evaluated the running costs of smart contracts and their methods.

Although, our current focus was on a single Blockchain platform, limiting the replication of real-world scenarios involving multiple platforms. Recognizing this limitation underscores the importance of addressing interoperability challenges among diverse Blockchain platforms. Future works involves resolving these issues for seamless communication and data exchange across varied platforms, aligning our work with real-world demands.

CHAPTER 5

BLOCKCHAIN-BASED ACCESS CONTROL MODEL FOR IoT ENVIRONMENT

Over the past few years, both Internet of Things (IoT) and Blockchain (BC) technologies have dominated their respective research domains. The fusion of IoT and Blockchain has paved the way for numerous efficient services, leveraging inherent qualities such as scalability, flexibility, resilience, availability, and integrity. However, due to the inherent limitations of IoT devices, the implementation of BC peers on these devices presents significant challenges. The rapid production of transactions by a multitude of constrained devices also poses hurdles to the effective utilization of Blockchain.

5.1 Introduction

To address these challenges, our proposed solution introduces the utilization of the Interplanetary File System (IPFS) for managing resources generated by IoT devices. This system is built upon the Hyperledger Fabric BC framework, housing smart contracts responsible for policy definition, enforcement, user identity management, and data retrieval. Our experimental results indicate that the execution time of smart contract methods in our proposed solution is notably lower compared to existing prominent works in the same domain. Performance evaluations showcase the efficacy of our model in achieving Confidentiality, Availability, and Integrity, and in thwarting DoS and DDoS attacks. The remarkable expansion in computer hardware and internet capabilities has immensely facilitatedthe interconnection of a vast array of devices through wireless networks, leading to an exponential proliferation of the Internet of Things (IoT). This interconnectedness and transformation of devices into smarter entities have significantly enhanced the convenience in our daily lives. However, this convenience has also resulted in increased intrusion into our privacy due to continuous monitoring by IoT devices. The data generated by these devices often comprises sensitive information, posing a serious threat if accessed without

authorization.

Numerous companies are engaged in various operations involving storage, processing, sharing, and analysis of data generated by smart devices to offer innovative services to society. Yet, data security and privacy in the context of IoT remain pressing concerns that demand more attention. Consequently, the primary challenge faced by IoT revolves around ensuring Privacy and Security. To tackle this, robust access control techniques are imperative to safeguard resources. Access control mechanisms, commonly employed in various systems, play a pivotal role in securing these resources while considering the constrained storage and processing capabilities of IoT devices. Without adequate protection measures integrated into IoT systems, the sensitive data they generate becomes vulnerable to security breaches. Therefore, aligning IoT strategies with access policies becomes crucial. Our proposed resource access control scheme emphasizes the significance of context in designing policies, during resource access, and post-access. This mechanism empowers resource owners to dictate and monitor access, determining who accesses their resources, which resources are accessed, and the timing of access events.

An effective access control system comprises three core elements: authentication, authorization, and auditing. Authentication ensures the validation of the requester's identity. Authorization determines whether the requester has the appropriate permissions to access specific resources or carry out certain operations.Auditing enables retrospective analysis of activities within the system.

Existing literature often addresses access control challenges using centralized schemes, where a central entity manages authorization mechanisms. However, these traditional methods fall short in the IoT environment, lacking scalability, flexibility, and resilience. A distributed scheme, where multiple entities participate in authorization decisions without relying on a central entity, can resolve these issues. The security and privacy concerns prevalent in the IoT era require access control mechanisms to meet additional non-functional requirements like scalability, flexibility, resilience, and lightweight characteristics, beyond the fundamental

aspects of integrity, confidentiality, and availability. Given the limitations of centralized and decentralized access control models, Blockchain-based solutions emerge as promising options in the IoT landscape. Blockchain technology ensures data immutability and integrity through a peer-to-peer network with distributed nodes maintaining transaction records. Traditional data-sharing methods often involve recording IoT data in third-party agencies, potentially compromising sensitive information. To enhance data availability and privacy, it's essential to transition from centralized to decentralized storage systems. Distributed storage offers advantages such as increased data throughput, cost-effectiveness, and enhanced resilience. This proposed framework introduces an effective access control system for data stored in a distributed environment. It leverages a distributed, peer-to-peer storage system known as IPFS to achieve these objectives.

5.2 Motivation and Contributions

The motivation for undertaking this work stems from the critical need to address the evolving challenges in securing and managing access to IoT generated data. As the IoT ecosystem continues to expand, there is a pressing requirement for robust data access control mechanisms that can provide secure, decentralized, and scalable solutions. Blockchain technology, with its decentralized and tamper-resistant nature, presents an intriguing avenue for enhancing the security and integrity of access control in IoT. Additionally, the integration of the IPFS into this model holds the promise of efficient and distributed data storage, further contributing to data availability and resilience. The motivation lies in the potential to develop a trust-based data access control model that not only addresses the inherent challenges in securing IoT environments but also aligns with the principles of decentralization. Such a model could significantly advance the security paradigm for IoT, ensuring confidentiality, integrity, and availability of data in a highly interconnected and dynamic ecosystem.

This chapter proposed a novel framework for decentralized resource access control by integrating Blockchain with IPFS. Since, public Blockchain has its challenges such as scalability issues and higher cost of transaction, in the proposed solution, access control schemes are imposed through Hyperledger fabric.

The key contributions of this chapter are as follows:

- A Blockchain-based, access behavior-driven access control mechanism is proposed for efficient resource sharing among IoT devices.
- IPFS-based decentralized data storage scheme resulting in high availability of data.
- Multiple permission levels are defined to offer permissioned access privileges to resource consumers.
- Either positive or negative value is assigned to IoT devices depending on their access behavior which eventually facilitates a dynamic resource access scheme.
- Provides two-phase authorization for resource access- static authorization (based on predefined access policies) and dynamic authorization (based on resource access behavior).

5.3 Proposed Work

In this section, we introduce a Blockchain-driven access control system tailored for the IoT landscape. We delve into secure data sharing, breaking down the process into several key sub-sections: Policy Model, Storage Model, System Architecture, System Interaction and Workflow, Smart Contract, and Algorithm and Implementation. Each sub-section provides a detailed elaboration of its respective aspects concerning the secure management and sharing of data within this framework.

5.3.1 Policy Model

The data resources produced by constrained devices are predominantly unstructured. In various real-world IoT scenarios, devices like smart cameras capture real-world images, generating pictures or video resources, while microphones receive sounds, producing audio resources. Sensors detect physical signals such as humidity, temperature, pressure, and light, translating them into digital signal resources. Due to the unstructured nature of this data, storing it directly in a relational database is impractical. Additionally, these real-time data resources need to be promptly distributed to authorized entities. As a solution, the resource data from constrained devices are disseminated across IPFS, and in return,

cryptographic hashes (hyperlinks) are generated. These resource hyperlinks are then uploaded to the Blockchain via an application gateway.

The proposed system's flowchart, depicted in Figure 5.1, initiates with a query for access initiated by the resource consumer. This query undergoes evaluation by a static authorization mechanism that examines predefined access policies. If the corresponding access policy is established, it returns a "true" verdict; otherwise, it yields a "false" outcome. After the static authorization phase, the process advances to dynamic authorization, which involves accessing historical interactiondata to assess the requesting entity's trust score. If this trust score exceeds a predefined threshold value, the outcome is deemed "true." Access authorization is granted only when both static and dynamic authorization phases yield a "true" result; otherwise, access is denied.

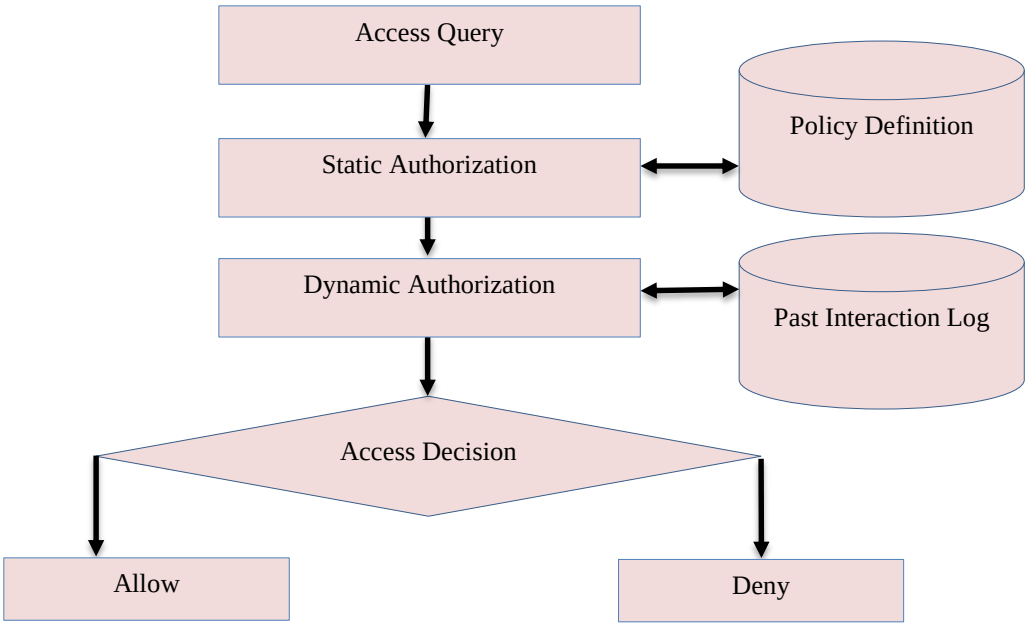

Figure 5.1: Proposed policy model

5.3.2 Storage Model

In this study, we have devised an innovative data storage model built upon the fusion of IPFS and Blockchain technologies. Our proposed storage model encompasses a decentralized storage approach using IPFS, where the original data generated by IoT devices is securely recorded in an encrypted format. Simultaneously, the cryptographic hash corresponding to this data is uploaded onto

the Blockchain network. This cryptographic hash represents a fixed-size data structure that demands notably less space, making it highly compatible and ideal for the integration of IoT and Blockchain technologies.

The storage scheme illustrated in Figure 5.2 outlines the process: all IoT-generated data (referred to as RO data) is stored in off-chain storage in its encrypted form, while the corresponding fixed-size hash is documented on the Blockchain. Moreover, whenever an authorized entity (referred to as RC) intends to access specific data, it initially retrieves the hash data from the Blockchain and subsequently obtains the actual data from the off-chain storage. This retrieval mechanism ensures a secure and efficient method for accessing IoT-generated data while maintaining integrity and confidentiality through the utilization of Blockchain and IPFS.

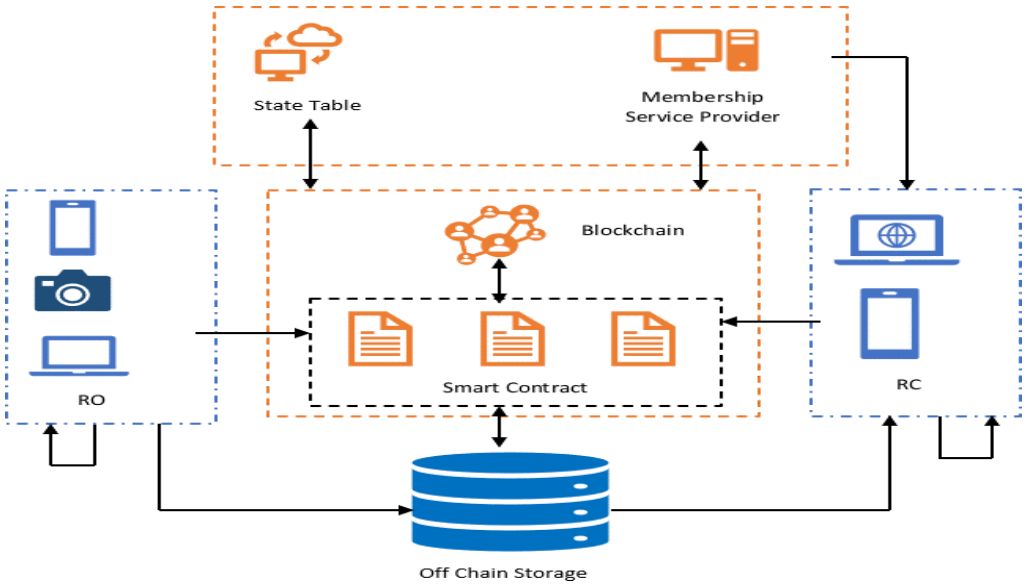

Figure 5. 2: Storage Model

5.3.3 System Architecture

The proposed IoT Blockchain platform encompasses a huge number of IoT devices (RC & RO), distributed data storage (IPFS), user devices (RC), servers (RO), and Gateways that are coupled together with a Blockchain network. Both RC & RO who seek access to the resource and hold the requested resource respectively are linked to the Blockchain through a Gateway. The IPFS is connected to IoT devices for data storage & data retrieval. All the RC's and RO's are required to register

themselves under at least one Device Manager (DM) who subsequently registers them within the Blockchain network. The complete structure is represented in Figure 5.3.

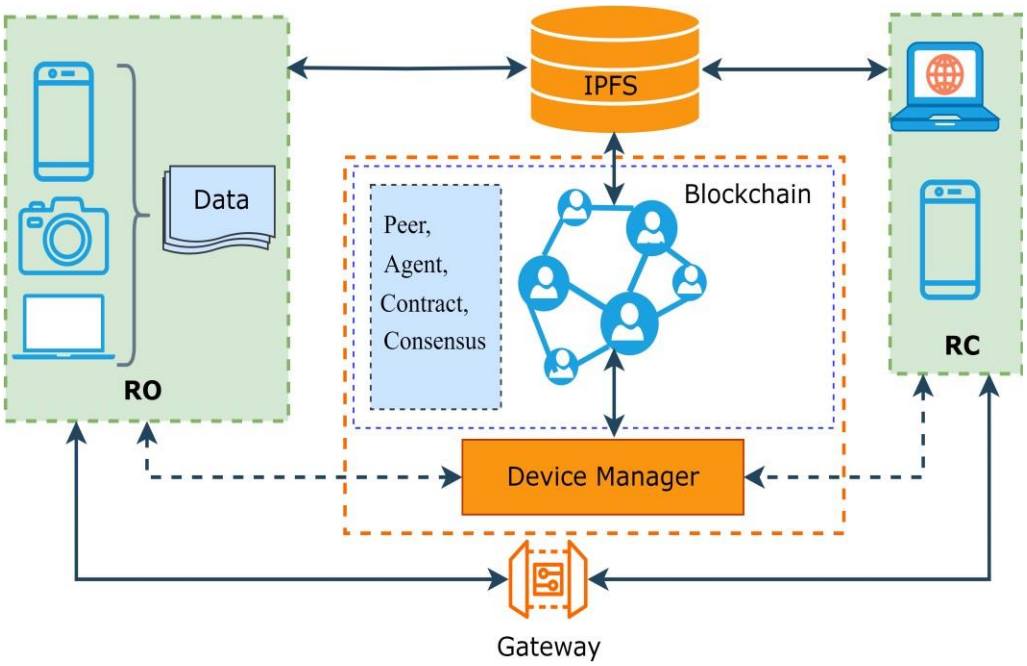

Figure 5.3: Proposed system architecture

5.3.4 System Interaction and Workflow

The sequence diagram depicted in figure 5.4 captures the sequence of interactions among various components of the proposed scheme which are explained as follows:

- Network setup
- Deployment of chaincode by Endorsing peer
- Device Managers register themselves onto the Blockchain network.
- Device Manager registers IoT devices (RO's and RC's) under them. (However, an IoT device can de-register itself from any Device Manager)
- Resource access policies are defined by the device manager and forwarded to the corresponding RO seeking its approval.
- The RO approves the received access policy and notifies it to the nearest Gateway.
- Access policy is recorded on the Blockchain through a transaction.
- The RO sends resource (data) upload requests to the IPFS along with its

authentication credentials.

- After successful authentication, the IPFS uploads the resource and returns the content hash of the uploaded data to the RO.
- The RO forwards the signed content hash along with its retrieval context to the Gateway.
- The content hash and the context are recorded onto the Blockchain through a transaction.
- RC sends a request for a resource to a Gateway.
- The Gateway translates the request to a Blockchain action.
- The Blockchain action is run by peers.
- If RC is not an authorized entity access request is denied and RC is notified through the Gateway.
- If RC is an authorized entity then encrypted (by the RC public key) content hash is returned to the Gateway.
- The Gateway forwards the message into the CoAP format.
- Upon receiving the message, RC decrypts the message by its private key and requests to the IPFS by sending a signed content hash.
- The IPFS returns the requested resource.
- Upon receiving resources along with its content hash, The RO verifies the content hash with the one it received from the Blockchain.
- Meanwhile, The IPFS report to the corresponding RO about its resource access by this RC.
- The RO sends an interaction score (trust score) to the Gateway.
- The interaction score is recorded on the Blockchain.

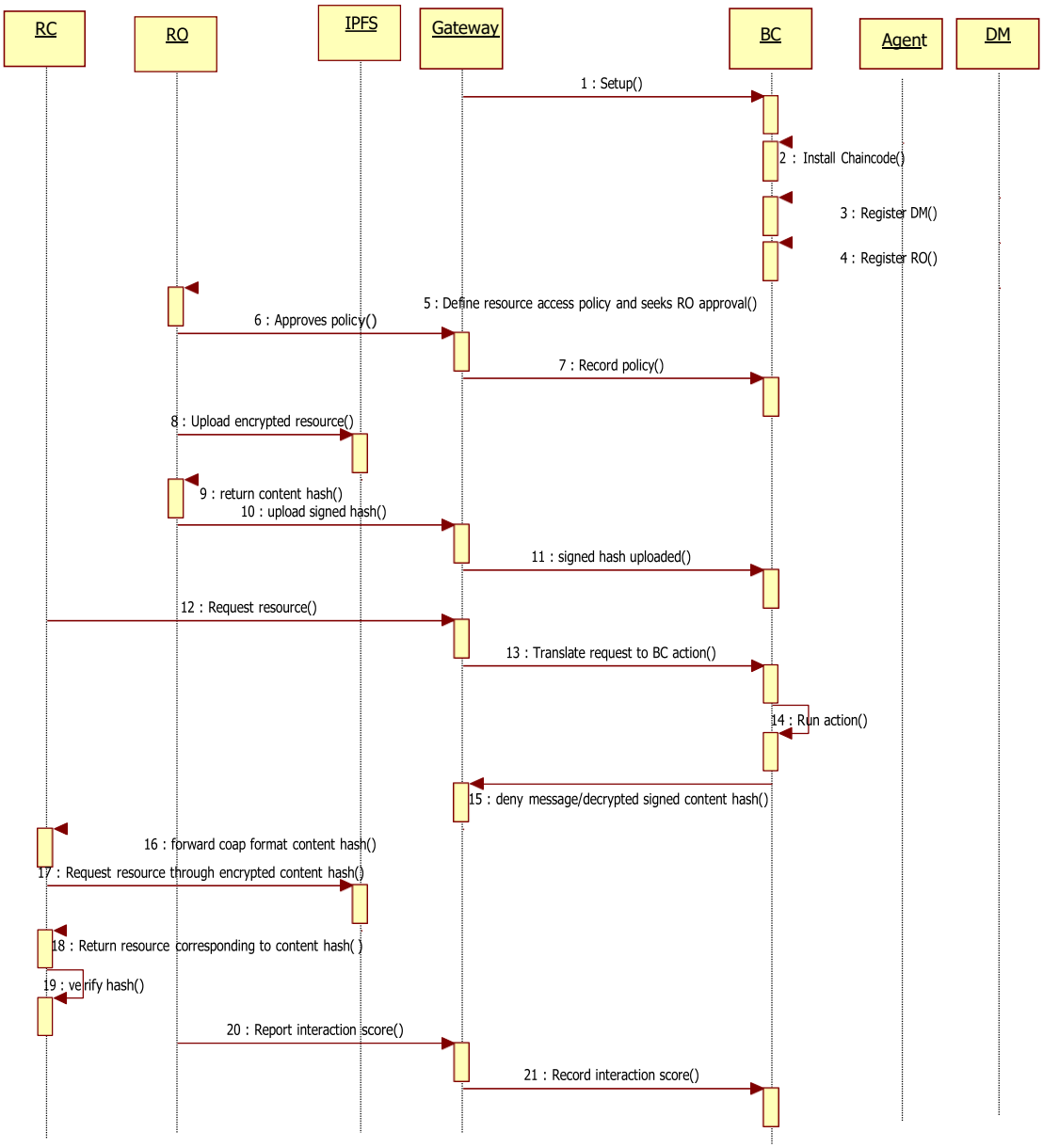

Figure 5.4: Control message flow of the proposed model

5.3.5 Smart Contract

The proposed system is structured around three smart contracts, also known as chaincode, designed to govern resource sharing among IoT devices. These contracts are the Access Policy Contract (APC), Device Contract (DC), and Trust Contract (TC). APC serves as the cornerstone of the model, responsible for implementing an access management scheme. It defines the rules and protocols governing access to resources.

DC manages the process of uploading the URL (content hash) of data generated by IoT devices and provides a querying mechanism for this data. It also contains crucial information regarding IoT devices for their identification and authentication.

TC encompasses a method to assign a trust value to each IoT device based on its access behavior and a method to retrieve this value. It evaluates the historical behavior of devices to ascertain trust levels. When a resource consumer (RC) initiates a request for resource access, the system evaluates both the permission level requested and the past access behavior of the RC. Access is permitted only if a positive assessment is derived from this evaluation process. For a comprehensive understanding of how APC, DC, and TC function, a detailed breakdown of their functionalities and operations is provided below. These contracts collaboratively manage and regulate resource access within the IoT ecosystem, ensuring security and efficient utilization of resources.

a) Device Contract (DC)

This smart contract incorporates functionalities to upload the content hash of data produced by IoT devices and a process to retrieve this information. Within this contract, a Device Information Table (DIT) is managed as shown in Table 5.1, housing pertinent device details essential for device identification and authentication during both device registration, managed under Device Manager, and subsequent resource access requests. It contains essential methods such as registerThing, getThing, and getAuthenticity, which collectively handle the registration of devices, retrieval of device information, and verification of authenticity respectively. The DIT includes the following information:

- RC: the entity that sends access requests.
- RO: the entity that holds (owner) requested resources.
- Resource(R): specific resource (data or file) requested by RC.
- Action: read (r), write (w), or execute(x) operation.
- LRtime: last request time for a resource by the RC.
- SLRtime: second last request time for a resource by the RC.

- TrustScore(TS): final trust score of the RC according to its access behavior.

Table 5.1: Device Information Table (DIT)

RC	RO	R	Action	LRtime	SLRtime	TS
RC A	RO X	File 1	R	2022/01/09/15:21:18	2022/01/08/10:20:33	0.19
RC B	RO Y	File 2	R, W	2022/01/01/03:47:00	2022/01/01/03:44:00	-0.51
RC C	RO Z	Program 3	W	2022/01/05/18:42:12	2022/01/03/14:11:48	0.39

b) Access Policy Contract (APC)

As the central smart contract in the proposed model, this contract oversees access control among IoT devices. When an RC seeks access to an RO's resource, it forwards an access request to the system via the gateway. The Access Policy Contract (APC) is then triggered and handles the access management for the requesting RC. This smart contract encompasses several methods aimed at fulfilling this purpose: addPolicy, deletePolicy, updatePolicy, verifyPolicy, and verifyAccess. These methods are utilized for distinct functions such as introducing new access policies, removing existing policies, modifying access rules, confirming newly established policies, and validating authorization for the current access request, respectively. This contract ensures the systematic administration of access rights within the IoT ecosystem.

c) Trust Contract (TC)

This smart contract is essentially dedicated to evaluate the access conduct of registered IoT devices (both RC and RO) through the implementation of three key methods: setTrust, getTrust, and setFine. These methods operate by assigning either a positive or negative value (referred to as a fine) to the registered devices based on their access behavior. The setTrust method manages the assignment of trust values, while getTrust retrieves these assigned values. Additionally, the setFine method deals with associating penalties or negative values depending on the observed

access behavior of the devices. Multiple reasons cause negative fine assignments which are as follows:

- RC sends access requests before the allowed interval (too frequent access requests)
- Multiple access requests by the RC within a fixed period.
- Access request of a resource having a higher authorization level
- Not accessing resources (canceled request) after approval.

If none of these situations occurs then a positive value is assigned to the corresponding device. TC also maintains a penalty table as shown in Table 5.2 to record access behavior and the corresponding penalty of the IoT devices.

Table 5.2: Device Penalty Table

RC	RO	Access Behavior	Penalty
RC A	RO X	Request cancelled after approval	Request blocked for 15 minutes
RC B	RO Y	Too frequent access request	Request blocked for 1 hour
RC C	RO Z	Multiple requests within a fixed period	Request blocked for 2 hour
RC D	RO W	Requesting higher authorization level resource	Request blocked for 20 minutes

5.3.6 Trust-based Authorization Algorithm

With the trust-based authorization scheme, both participating entities (RC & RO) are assigned a Trust Score (TS) depending on their past access interactions. There are two types of TS values:

- Local: - Corresponding to RC for the requested resource.
- Global: - Overall trust score for RC.

$$\mathbf{TS = TS_{local} + TS_{global}} \quad (1)$$

TS= trust.setTrust(r, fine) (2)

Local TS is computed corresponding to the current access request, and it depends on multiple access violations. On each such violation, a predefined fine is imposed on the requested entity.

$$\mathbf{TS_{local}} = \left(\sum_{i=0}^{2} \boldsymbol{fine}[i]^{i+1}\right) / \mathbf{3} \quad (3)$$

The second argument of the above method "fine" is a vector that comprises multiple components representing different access violation scenarios.

fine[0] = X (RC tries to access a resource that requires a higher authorization level)

fine[1] = Y (RC tries to access resource before allowed time interval)

fine[2] = Z (RC made frequent access request)

$$\mathbf{TS_{global}} = \left(\sum_{i=1}^{n} \boldsymbol{W^{n-i}} * \mathbf{TS_{locali}}\right) / \mathbf{n} \quad (4)$$

Where Wi is a weight, higher weight is assigned to most recent interactions and lower weight for past interactions.

$W^{(n-i)}$ is an aging parameter.

Now, assigning expression for TS_{local} & TS_{global} from equation 3 & 4 into equation 1.

$$\mathbf{TS} = \left(\sum_{i=0}^{2} \boldsymbol{fine}[i]^{i+1}\right) / \mathbf{3} + \left(\sum_{i=1}^{n} \boldsymbol{W^{n-i}} * \mathbf{TS_{locali}}\right) / \mathbf{n} \quad (5)$$

The above proposed concept is well defined in algorithm 1 and all the related terminologies used within the algorithm is described in Table 5.3.

Table 5.3: Terminologies used in Algorithm.

Term	**Description**
RC	Resource Consumer
RO	Resource Owner
R	Resource
Action	read/write/execute.
UnblockTime	time until which request is blocked
StaticCheck	predefined access policies
DynamicCheck	regulates consumers behavior dynamically
LRtime	Last Request time
allowedInterval	the minimum allowable time between successive requests.
NoRR	Number of Recent Requests (request made in a fixed time interval)
fine]X]	Request cancelled after approval
fine]Y]	Too frequent access request
fine]Z]	Multiple requests within a fixed period

The "accessControl()" algorithm orchestrates access control within a system, governed by a series of input parameters and predefined policies. It begins by initializing internal variables and retrieving a policy based on specific inputs. It subsequently performs time-based evaluations, checking if the current time aligns with designated UnblockTime criteria and conducting static permission checks based on predefined policies. Additionally, it tracks dynamic interactions, imposing fines or penalties based on time intervals and predefined thresholds. Trust levels are then evaluated using the fines accumulated during the process. Depending on these trust levels and certain conditions, the algorithm may set future UnblockTime parameters. Ultimately, it assesses both static and dynamic checks to determine if they meet established criteria. Successful evaluation triggers the "getApproval()" function, while failure prompts notification of a penalty.

Algorithm 5.1: *accessControl ()*

Input: *RC, RO, R, time, Action*
Output: *result, penalty*
Requirement: *StaticCheck←false, DynamicCheck←true, penalty ← 0*

```
1   r ← Policy [(RC, RO, R)] [Action]
2   if time >=UnblockTime then
3   |  r. UnblockTime ← 0
4   |  if (r. permission) then
5   |  |  StaticCheck ← true
6   |  else if r. permissionLevel! = Action
7   |  |  fine [0]←X
8   |  |  endif
9   |  endif
10  |  if time - r. LRtime < r. allowedInterval then
11  |  |  fine [1]← Y
12  |  |  NoRR← time - r. SLRtime
13  |  |  if NoRR >= r. threshold  then
14  |  |  |  fine [2]← Z
15  |  |  |  Add implicit behavior IB to the behavior list of TC
16  |  |  endif
17  |  DynamicCheck← false
18  |  endif
19  endif
20  r. SLRtime ← r. LRtime
21  r.LRtime←time
22  TS←trust.setTrust(r, fine)
23  if (fine [2]) then
```

```
24 |   |   UnblockTime←time + mod(TS)*multiplier
25 | endif
26 | Check ← StaticCheck AND DynamicCheck
27 | if (Check = = TRUE) AND (TS>=TS_th) then
28 |   | Trigger getApproval()
29 | else
30 |   | Notify penalty
31 | endif
```

The "Uploading data to IPFS" algorithm manages the process of uploading data to the InterPlanetary File System (IPFS) while ensuring the integrity and structure of the data resources. It takes as input the data resource and the hash of the old version of the resource tree. Initially, it retrieves the previous version of the ResourceTree object using the provided hash. Depending on whether the data resource is text-based or not, the algorithm follows distinct paths. For text data, it organizes the data into a DataPackage based on the data type property. If no existing DataPackage is found, a new one is created and the ResourceData is inserted. If a DataPackage exists, the DataResource is appended to it. If the DataPackage reaches its storage limit, it's stored in IPFS, and its content hash is obtained. Subsequently, the DataResource is attached to a DataBlock to create a new resource tree hash. Alternatively, for non-text data, the algorithm stores the dataResource directly in IPFS to obtain its content hash. This content hash is then combined with the old resource tree hash to create a new resource tree object, generating a new hash for the updated resource tree.

Algorithm 5.2: *Uploading data to IPFS*

```
Input: data resource, Old resourceTree hash (ORT-hash)
Output: content hash of data resource, new ResourceTree hash (NRT-hash)
1  | ORT_object← get the old version of ResourceTree object by ORT_hash
2  | if data resource is text data, then
3  |   | DataPackage← get data package according to type of data resource
4  |   | if DataPackage is Null then
5  |   |   | Create new DataPackage
6  |   |   | Insert ResourceData to the DataPackage
7  |   | else
8  |   |   | Append DataResource to DataPackage
9  |   | endif
10 |   | if DataPackage reached storage limit then
11 |   |   | Store DataPackage to IPFS and get content hash
```

```
12      |   |   NRT_hash← attach DataResource to DataBlock and obtain new
            |   resource tree hash
13      |   else
14      |   |   DataPackage is temporarily stored
15      |   endif
16      else
17      |   Content_hash← store dataResource to IPFS and obtain the content hash
18      |   NRT_hash← attach content hash with ORT hash
19      |   Create a new resource tree object and get NRT_hash
20      endif
```

5.4 Simulation and Result Analysis

This section encompasses three subsections where result pattern, security aspects and limitations are discussed respectively.

5.4.1 Result and Discussion

We conducted an evaluation of the execution times for the DC, APC, and TC methods across various query scenarios involving concurrent requests set at 50, 100, 200, 500, and 1000. The time taken for each request was meticulously recorded for in-depth analysis, and these findings are graphically presented in Figures 5.5 through 5.11. Figure 5.5 demonstrates that registering new consumers entails a longer processing time compared to retrieving information about registered consumers. This is due to the transactional nature of registering a new consumer, contrasting with the simpler process of fetching information about an already registered consumer, which doesn't require transaction execution.

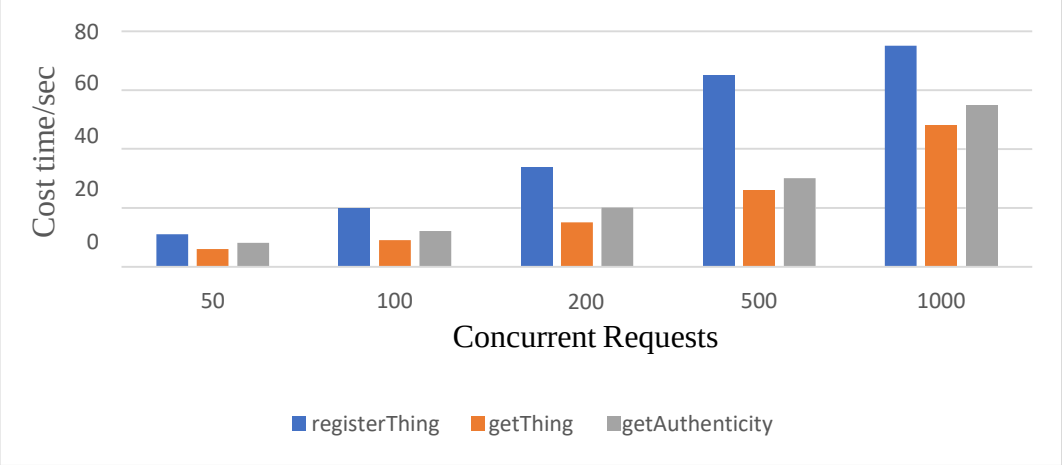

Figure 5.5: Running time of DC's methods

In Figure 5.6, we observed that computing trust scores and imposing fines on undesired consumers take more time than retrieving trust scores and imposed fines. This disparity arises from the complexities involved in evaluating trust scores and imposing fines. Furthermore, we compared the performance of our solution with that of references [85] and [116] concerning the execution time of access control contract methods. Figures 5.7 through 5.10 depict the running costs of addPolicy, updatePolicy, deletePolicy, and verifyAccess methods across these models. Our approach aligns closely with [116] in terms of adding and updating policies, while [85] exhibits higher time costs. However, our approach takes more time in verifying access policies than [116]. Addressing the disparity between the high rate of IoT data generation and therelatively slower data validation and storage on the Blockchain, we adopted astrategy where IoT data is not directly uploaded onto the Blockchain. Instead, it isuploaded onto IPFS, and only a fixed-size hash corresponding to the data isrecorded on the Blockchain.

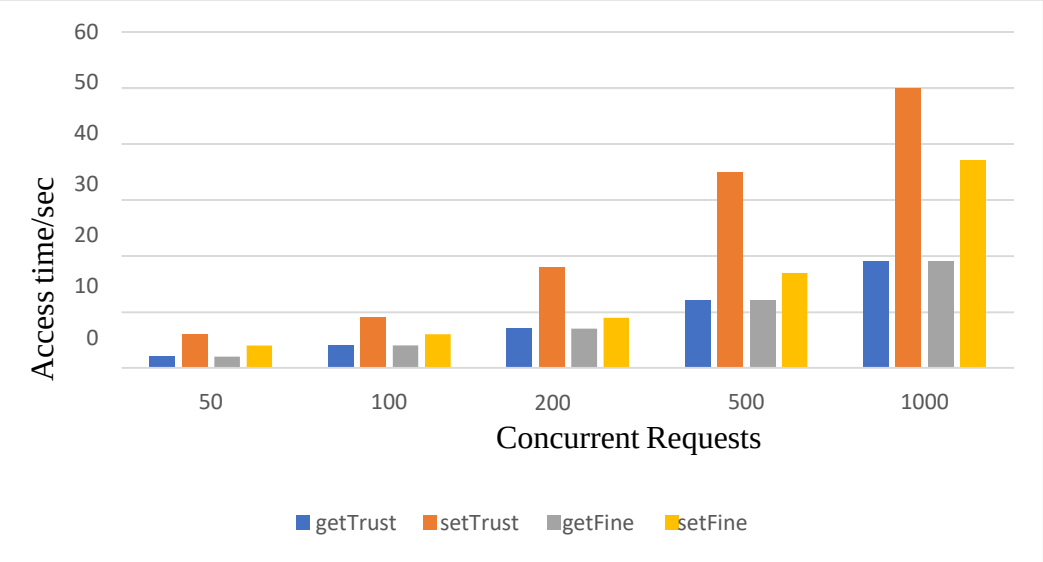

Figure 5.6: Running time of TC's methods

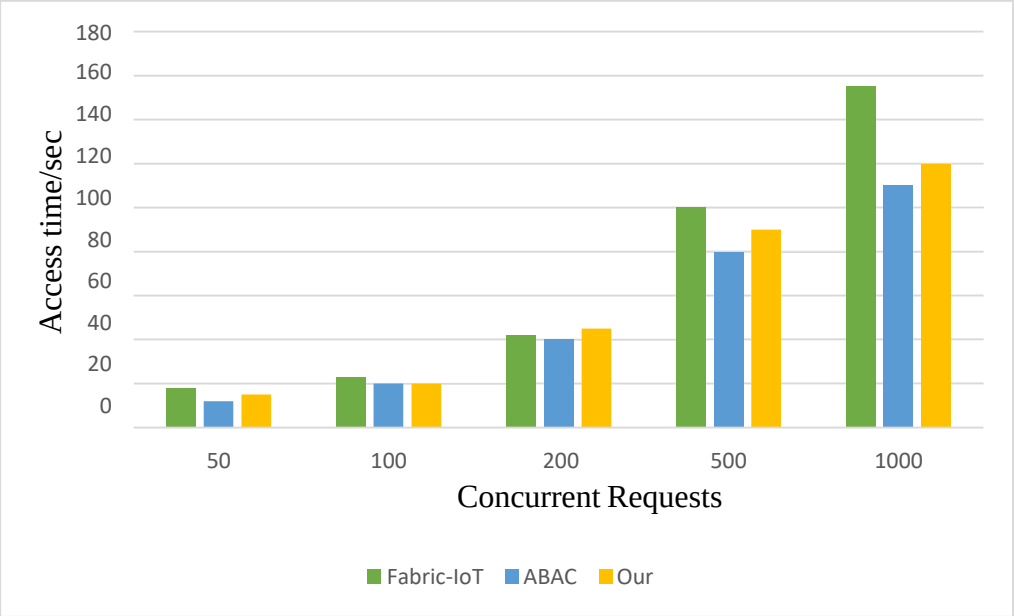

Figure 5.7: Running time of addPolicy method

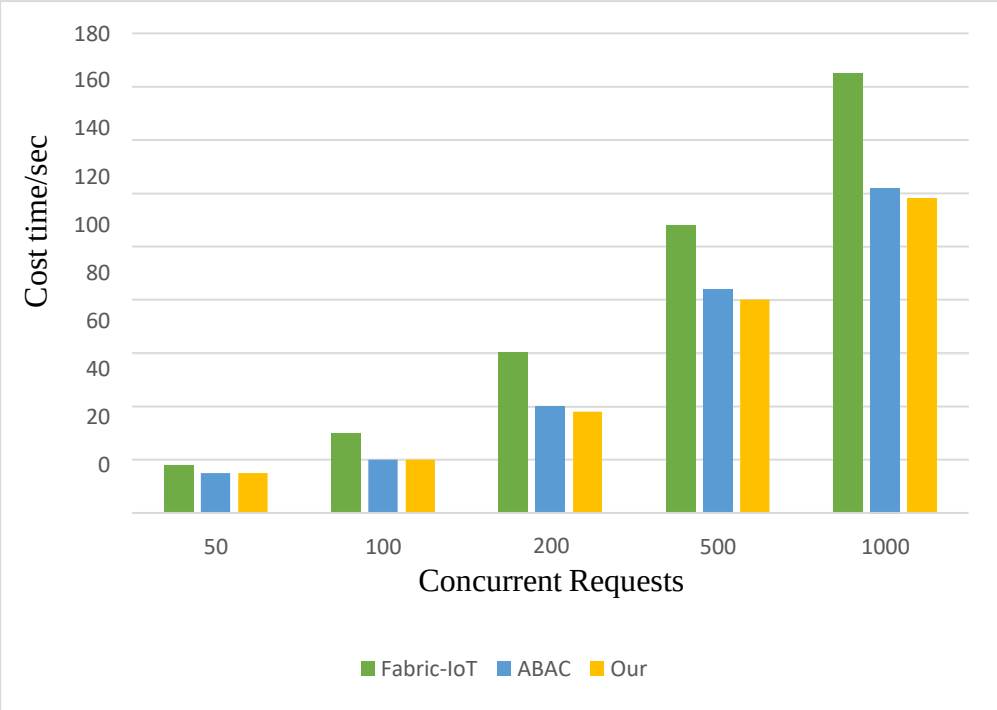

Figure 5.8: Running time of updatePolicy method

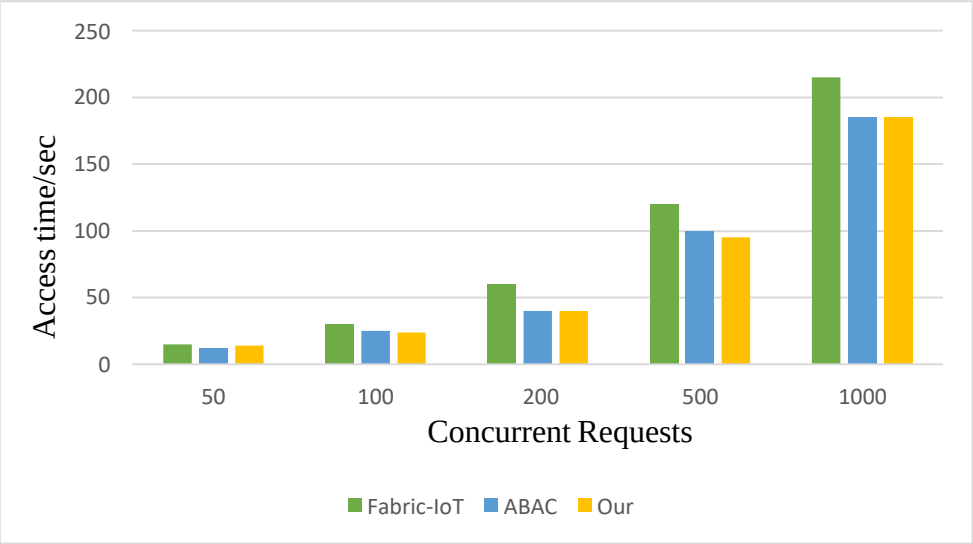

Figure 5.9: Running time of deletePolicy method

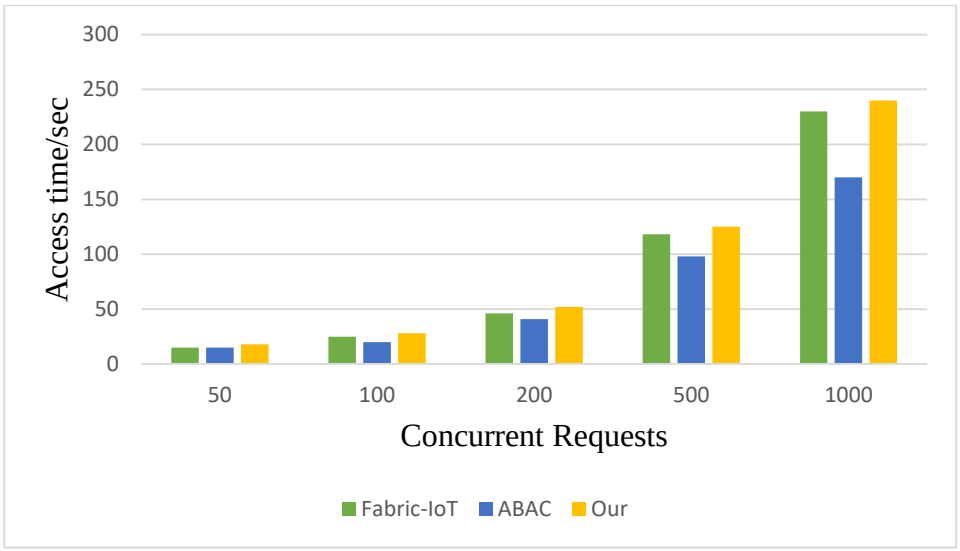

Figure 5.10: Running time of verifyAccess method

5.4.2 Security Analysis

The proposed architecture places significant emphasis on key security parameters, specifically Availability, Confidentiality, and Integrity.

Availability is ensured by implementing measures to thwart DoS and DDoS attacks while transitioning from centralized storage to a distributed storage system. This

transition mitigates the risk of data censorship and eliminates the vulnerability of a single point of failure. To counter DoS and DDoS attacks, the architecture mandates that each communicating entity registers within the blockchain network via at least one manager. Additionally, dynamic authorization within the proposed architecture scrutinizes the past interaction patterns of resource consumers (RC). It restricts frequent access requests from RCs exhibiting poor trust scores, bolstering availability by maintaining system stability.

Integrity is preserved by preventing the forgery of IoT-generated data. This is accomplished by uploading cryptographic hash content, along with the signature of the resource owner (RO), rather than the original data. Consequently, any resource consumer (RC) can utilize the hash and signature to easily verify the authenticity and integrity of the data, ensuring data reliability.

Confidentiality is upheld through the implementation of static and dynamic authorization mechanisms and fine-grained access control within the proposed architecture. Both on-chain and off-chain storage exclusively handle encrypted data, ensuring confidentiality. This strategy ensures that sensitive information remains secure and inaccessible to unauthorized entities, thus achieving confidentiality objectives seamlessly.

5.5 Summary

In this chapter, we successfully integrated IoT and Blockchain technologies, establishing a secure IoT data-sharing framework that prioritizes scalability, flexibility, resilience, and the integrity and availability of IoT data. This Blockchain-based and IPFS-enabled scheme operates through two distinct phases of authorization: static authorization and dynamic authorization. Static authorization focuses on validating predefined access policies, while dynamic authorization computes a trust score for each participating entity (including IoT and user devices) and compares it against a predefined threshold value. Access to requested resources is granted only upon successful authorization from both phases.

The integration of IPFS serves to record actual IoT data, significantly enhancing

the availability of IoT-generated data compared to centralized storage. Furthermore, the dynamic authorization component bolsters the prevention of DoS and DDoS attacks, while the incorporation of IPFS supports the realization of integrity and confidentiality.

Our experimental results highlight the efficiency gains achieved by uploading IoT data to IPFS versus directly onto the Blockchain. Additionally, the smart contract methods in our proposed solution demonstrate notably reduced running times compared to leading works in the same domain. However, our current architecture operates solely on a single Blockchain platform, limiting its compatibility within a genuine IoT ecosystem. Future enhancements aim to adapt this approach for hybrid Blockchain networks. Furthermore, our future endeavors involve extensive testing, including simulations of malicious behaviors.

CHAPTER-6

CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

In this comprehensive exploration, our developed Blockchain and IPFS-based access management framework represents a significant step toward fortifying security and enhancing accessibility in the Internet of Things (IoT) ecosystem. By integrating these cutting-edge technologies and implementing both static and dynamic authorization policies, we've established a sophisticated and robust model that reshapes how IoT data is managed, shared, and secured.

Chapter 3 focused on crafting specialized authorization algorithms tailored for facilitating IoT data sharing. This endeavor primarily aimed to overcome constraints inherent in traditional centralized authorization systems by advocating decentralized approaches, leveraging the transformative potential of Blockchain technology. Notably, our methodology pioneered the integration of adaptable authorization policies, specifically designed to suit the dynamic nature inherently present within IoT environments. This holistic approach aimed to bolster data sharing capabilities while addressing the limitations of conventional authorization systems, paving the way for a more resilient and scalable framework for IoT data management and access control.

In chapter 4, we integrated Blockchain and IPFS technologies that addresses critical limitations that have long hindered conventional data-sharing methodologies. Our framework's utilization of IPFS-based storage eradicates vulnerabilities associated with single points of failure, considerably mitigates risks of data censorship, and notably amplifies data availability. Simultaneously, it achieves these advancements while significantly reducing storage overheads, presenting a compelling case for decentralized data management within IoT environments.

In chapter 5, we proposed a two-phase authorization model involving static and dynamic authorization. Static authorization validates predefined access policies, while dynamic authorization computes trust scores for IoT and user devices,

comparing them to predefined thresholds. Successful authorization from both phases grants access to requested resources. This comprehensive approach not only bolsters the system against unauthorized access attempts but also proactively mitigates potential threats posed by DoS and DDoS attacks, ensuring a more resilient and secure IoT environment. Additionally, IPFS is employed to store IoT data, enhancing its availability compared to centralized storage. Our experiments indicate that uploading IoT data to IPFS is notably faster than direct Blockchain uploads, and the execution time of our proposed smart contract methods is considerably shorter than existing solutions in the same domain.

In essence, our Blockchain and IPFS-based access management framework, underpinned by dynamic and static authorization policies, signifies a pivotal advancement in securing and efficiently managing IoT data. The forthcoming trajectory aims to refine adaptability, scalability, and versatility across varied IoT environments, solidifying data security, integrity, and access while advancing the broader landscape of Blockchain-powered solutions within the IoT paradigm.

6.2 Future Scope

The current framework, mark a significant leap in Blockchain-integrated IoT access management, yet there exist avenues for future advancements. Rigorous testing, particularly scalability analysis involving a more extensive device pool, is imperative to validate real-world usability. Additionally, deeper dives into security measures, real-world use cases in various industries, and optimizing consensus protocols stands as promising areas for further research. These future pursuits are essential for refining the framework's capabilities, enhancing its adaptability across industries, and cementing its position as a transformative force in IoT data management and security.

Research Publications

Paper Published in International Journals:

1. Mishra, Rajiv K., Rajesh K. Yadav, and Prem Nath. "Blockchain Driven Access Control Architecture for the Internet of Things." Multimedia Tools and Applications (2023): 1-25. https://doi.org/10.1007/s11042-023-14881-5 (SCIE- 3.0)
2. Mishra, Rajiv Kumar, Yadav, Rajesh Kumar, and Nath, Prem. 'Secure IoT Data Management and Sharing Architecture for Information Security Using Cryptographic Technique'. 1 Jan. 2023 : 10951 – 10966. https://doi.org/10.3233/JIFS-232483 (SCIE-2.0)
3. Mishra, R.K., Yadav, R.K. & Nath, P. Integration of Blockchain and IPFS: healthcare data management & sharing for IoT Environment. Multimed Tools Appl (2024). https://doi.org/10.1007/s11042-024-20092-3 (SCIE- 3.0)
4. Mishra, R.K., Yadav, R.K. & Nath, P. Access Control Models and Frameworks for the IoT Environment: Review, Challenges, and Future Direction. Wireless Pers Commun (2024). https://doi.org/10.1007/s11277-024-11568-4 (SCIE-1.9)

Paper Published in International Conferences:

5. Mishra, Rajiv, and Rajesh Yadav. "Access control in IoT networks: analysis and open challenges." Proceedings of the International Conference on Innovative Computing & Communications (ICICC). 2020.
6. Mishra, Rajiv K., Rajesh K. Yadav, and Prem Nath. "Blockchain-Based Decentralized Authorization Technique for Data Sharing in the Internet of Things." 2021 5th International Conference on Information Systems and Computer Networks (ISCON). IEEE, 2021.
7. R. K. Mishra, R. K. Yadav and P. Nath, "Blockchain Powered IoT Access Control Model for Secure Data Sharing and Management: Performance Analysis," 2023 Second International Conference on Informatics (ICI), Noida, India, 2023, pp. 1-6, doi: 10.1109/ICI60088.2023.10420910.

References:

[1] Da Xu, L., He, W., & Li, S. (2014). Internet of things in industries: A survey. IEEE Transactions on industrial informatics, 10(4), 2233-2243.

[2] Gungor, V. C., & Hancke, G. P. (2009). Industrial wireless sensor networks: Challenges, design principles, and technical approaches. IEEE Transactions on industrial electronics, 56(10), 4258-4265.

[3] Čolaković, A., & Hadžialić, M. (2018). Internet of Things (IoT): A review of enabling technologies, challenges, and open research issues. Computer networks, 144, 17-39.

[4] Lin, J., Yu, W., Zhang, N., Yang, X., Zhang, H., & Zhao, W. (2017). A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications. IEEE internet of things journal, 4(5), 1125-1142.

[5] Sadhu, P. K., Yanambaka, V. P., & Abdelgawad, A. (2022). Internet of things: Security and solutions survey. Sensors, 22(19), 7433.

[6] Shafique, K., Khawaja, B. A., Sabir, F., Qazi, S., & Mustaqim, M. (2020). Internet of things (IoT) for next-generation smart systems: A review of current challenges, future trends and prospects for emerging 5G-IoT scenarios. Ieee Access, 8, 23022-23040.

[7] Patel, C., & Doshi, N. (2019). Security challenges in IoT cyber world. Security in smart cities: models, applications, and challenges, 171-191.

[8] Rao, T. A., & Haq, E. U. (2018). Security challenges facing IoT layers and its protective measures. International Journal of Computer Applications, 179(27), 31-35.

[9] Sisinni, E., Saifullah, A., Han, S., Jennehag, U., & Gidlund, M. (2018). Industrial internet of things: Challenges, opportunities, and directions. IEEE transactions on industrial informatics, 14(11), 4724-4734.

[10] Kumar, N. M., & Mallick, P. K. (2018). The Internet of Things: Insights into the building blocks, component interactions, and architecture layers. Procedia computer science, 132, 109-117.

[11] Lobanchykova, N. M., Pilkevych, I. A., & Korchenko, O. (2021). Analysis of attacks on components of IoT systems and cybersecurity technologies. In CEUR Workshop Proceedings (2021, in press) (pp. 83-96).

[12] Liao, Z., Nazir, S., Khan, H. U., & Shafiq, M. (2021). Assessing security of software components for Internet of Things: a systematic review and future directions. Security and Communication Networks, 2021, 1-22.

[13] Keramidas, G., Voros, N., & Hübner, M. (2016). Components and services for IoT platforms. Cham: Springer International Pu.

[14] Otalvaro, C. M. M., Andrade, J. C. B., Jaramillo, C. M. Z., & RiosPatiño, J. I. (2022). IoT Best Practices and their components: A Systematic Literature Review. IEEE Latin America Transactions, 20(10), 2217-2228.

[15] Molaei, F., Rahimi, E., Siavoshi, H., Afrouz, S. G., & Tenorio, V. (2020). A comprehensive review on internet of things (IoT) and its implications in the mining industry. American Journal of Engineering and Applied Sciences, 13(3), 499-515.

[16] Bansal, S., & Kumar, D. (2020). IoT ecosystem: A survey on devices, gateways, operating systems, middleware and communication. International Journal of Wireless Information Networks, 27, 340-364.

[17] Patel, C., & Doshi, N. (2019). Security challenges in IoT cyber world. Security in smart cities: models, applications, and challenges, 171-191.

[18] Litoussi, M., Kannouf, N., El Makkaoui, K., Ezzati, A., & Fartitchou, M. (2020). IoT security: challenges and countermeasures. Procedia Computer Science, 177, 503-508.

[19] Mohanty, J., Mishra, S., Patra, S., Pati, B., & Panigrahi, C. R. (2021). IoT security, challenges, and solutions: a review. Progress in Advanced Computing and Intelligent Engineering: Proceedings of ICACIE 2019, Volume 2, 493-504.

[20] Lee, E., Seo, Y. D., Oh, S. R., & Kim, Y. G. (2021). A Survey on Standards for Interoperability and Security in the Internet of Things. IEEE Communications Surveys & Tutorials, 23(2), 1020-1047.

[21] Wang, D., Bai, B., Lei, K., Zhao, W., Yang, Y., & Han, Z. (2019). Enhancing information security via physical layer approaches in heterogeneous IoT with

multiple access mobile edge computing in smart city. IEEE Access, 7, 54508-54521.

[22] Singh, S., Sharma, P. K., Moon, S. Y., & Park, J. H. (2017). Advanced lightweight encryption algorithms for IoT devices: survey, challenges and solutions. Journal of Ambient Intelligence and Humanized Computing, 1-18.

[23] Tahir, M., Sardaraz, M., Muhammad, S., & Saud Khan, M. (2020). A lightweight authentication and authorization framework for blockchain-enabled IoT network in health-informatics. Sustainability, 12(17), 6960.

[24] Krishnamurthi, R., Kumar, A., Gopinathan, D., Nayyar, A., & Qureshi, B. (2020). An overview of IoT sensor data processing, fusion, and analysis techniques. Sensors, 20(21), 6076.

[25] Wheelus, C., & Zhu, X. (2020). IoT network security: Threats, risks, and a data-driven defense framework. IoT, 1(2), 259-285.

[26] Baig, Z. A., Sanguanpong, S., Firdous, S. N., Nguyen, T. G., & So-In, C. (2020). Averaged dependence estimators for DoS attack detection in IoT networks. Future Generation Computer Systems, 102, 198-209.

[27] Sung, Y., Lee, S., & Lee, M. (2018). A multi-hop clustering mechanism for scalable IoT networks. Sensors, 18(4), 961.

[28] White, G., Nallur, V., & Clarke, S. (2017). Quality of service approaches in IoT: A systematic mapping. Journal of Systems and Software, 132, 186-203.

[29] Farhan, L., Hameed, R. S., Ahmed, A. S., Fadel, A. H., Gheth, W., Alzubaidi, L., ... & Al-Amidie, M. (2021). Energy efficiency for green internet of things (IoT) networks: A survey. Network, 1(3), 279-314.

[30] Singh, I., & Singh, B. (2023). Access management of IoT devices using access control mechanism and decentralized authentication: A review. Measurement: Sensors, 25, 100591.

[31] Mishra, R. K., Yadav, R. K., & Nath, P. Secure IoT data management and sharing architecture for information security using cryptographic technique. Journal of Intelligent & Fuzzy Systems, 1-16.

[32] Kim, H., & Lee, E. A. (2017). Authentication and Authorization for the Internet of Things. IT Professional, 19(5), 27-33.

[33] Echenim, K., Elluri, L., & Joshi, K. (2023). Ensuring privacy policy compliance of wearables with iot regulations. UMBC Center for Accelerated Real Time Analysis.

[34] Perez, A. J., Zeadally, S., & Cochran, J. (2018). A review and an empirical analysis of privacy policy and notices for consumer Internet of things. Security and Privacy, 1(3), e15.

[35] Morgner, P., & Benenson, Z. (2018). Exploring security economics in IoT standardization efforts. arXiv preprint arXiv:1810.12035.

[36] Mishra, R., & Yadav, R. (2020, March). Access control in IoT networks: analysis and open challenges. In Proceedings of the International Conference on Innovative Computing & Communications (ICICC).

[37] Jiang, Y., Wang, C., Wang, Y., & Gao, L. (2019). A cross-chain solution to integrating multiple blockchains for IoT data management. Sensors, 19(9), 2042.

[38] Ravidas, S., Lekidis, A., Paci, F., & Zannone, N. (2019). Access control in Internet-of-Things: A survey. Journal of Network and Computer Applications, 144, 79-101.

[39] Riad, K., & Cheng, J. (2021). Adaptive XACML access policies for heterogeneous distributed IoT environments. Information Sciences, 548, 135-152.

[40] Oh, S. R., Kim, Y. G., & Cho, S. (2019). An interoperable access control framework for diverse IoT platforms based on oauth and role. Sensors, 19(8), 1884.

[41] Lin, C. A., & Liao, C. F. (2020, December). User-managed access delegation for blockchain-driven IoT services. In 2020 International Computer Symposium (ICS) (pp. 462-467). IEEE.

[42] Mishra, R. K., Yadav, R. K., & Nath, P. (2023). Blockchain Driven Access control architecture for the internet of things. Multimedia Tools and Applications, 1-25.

[43] Moyer, M. J., & Abamad, M. (2001, April). Generalized role-based access control. In Proceedings 21st International Conference on Distributed Computing Systems (pp. 391-398). IEEE.

[44] Ravidas, S., Lekidis, A., Paci, F., & Zannone, N. (2019). Access control in Internet-of-Things: A survey. Journal of Network and Computer Applications, 144, 79-101.

[45] Hu, V. C., Kuhn, D. R., Ferraiolo, D. F., & Voas, J. (2015). Attribute-based access control. Computer, 48(2), 85-88.

[46] Park, J., & Sandhu, R. (2004). The UCONABC usage control model. ACM transactions on information and system security (TISSEC), 7(1), 128-174.

[47] Singh, A., & Chatterjee, K. (2019). Trust based access control model for securing electronic healthcare system. Journal of Ambient Intelligence and Humanized Computing, 10, 4547-4565.

[48] Rasifard, H., Gopinath, R., Backes, M., & Nemati, H. (2023, May). SEAL: capability-based access control for data-analytic scenarios. In Proceedings of the 28th ACM Symposium on Access Control Models and Technologies (pp. 67-78).

[49] Ahmed, A. H., Omar, N. M., & Ibrahim, H. M. (2019, December). Secured framework for IoT using blockchain. In 2019 Ninth International Conference on Intelligent Computing and Information Systems (ICICIS) (pp. 270-277). IEEE.

[50] Elsayed, W., Gaber, T., Zhang, N., & Ibrahim Moussa, M. (2016). Access control models for pervasive environments: A survey. In The 1st International Conference on Advanced Intelligent System and Informatics (AISI2015), November 28-30, 2015, Beni Suef, Egypt (pp. 511-522). Springer International Publishing.

[51] Jiang, Y., Wang, C., Wang, Y., & Gao, L. (2019). A cross-chain solution to integrating multiple blockchains for IoT data management. Sensors, 19(9), 2042.

[52] Arman, A., Bellini, P., Bologna, D., Nesi, P., Pantaleo, G., & Paolucci, M. (2021). Automating IoT data ingestion enabling visual representation. Sensors, 21(24), 8429.

[53] Srivastava, P., & Garg, N. (2015, May). Secure and optimized data storage for IoT through cloud framework. In International Conference on Computing, Communication & Automation (pp. 720-723). IEEE.

[54] Krishnamurthi, R., Kumar, A., Gopinathan, D., Nayyar, A., & Qureshi, B. (2020). An overview of IoT sensor data processing, fusion, and analysis techniques. Sensors, 20(21), 6076.

[55] Elijah, O., Rahman, T. A., Orikumhi, I., Leow, C. Y., & Hindia, M. N. (2018). An overview of Internet of Things (IoT) and data analytics in agriculture: Benefits and challenges. IEEE Internet of things Journal, 5(5), 3758-3773.

[56] Sultana, T., Almogren, A., Akbar, M., Zuair, M., Ullah, I., & Javaid, N. (2020). Data sharing system integrating access control mechanism using blockchain-based smart contracts for IoT devices. Applied Sciences, 10(2), 488.

[57] Al Sadawi, A., Hassan, M. S., & Ndiaye, M. (2021). A survey on the integration of blockchain with IoT to enhance performance and eliminate challenges. IEEE Access, 9, 54478-54497.

[58] Atlam, H. F., Azad, M. A., Alzahrani, A. G., & Wills, G. (2020). A Review of Blockchain in Internet of Things and AI. Big Data and Cognitive Computing, 4(4), 28.

[59] https://www.philippe-fournier-viger.com/spmf/datasets/Kosarak10k.txt

[60] Qiu, J., Tian, Z., Du, C., Zuo, Q., Su, S., & Fang, B. (2020). A survey on access control in the age of internet of things. IEEE Internet of Things Journal, 7(6), 4682-4696.

[61] Atlam, H. F., Alassafi, M. O., Alenezi, A., Walters, R. J., & Wills, G. B. (2018, March). XACML for Building Access Control Policies in Internet of Things. In IoTBDS (pp. 253-260).

[62] Sciancalepore, S., Piro, G., Caldarola, D., Boggia, G., & Bianchi, G. (2017, July). OAuth-IoT: An access control framework for the Internet of Things based on open standards. In 2017 IEEE symposium on computers and communications (ISCC) (pp. 676-681). IEEE.

[63] Cirani, S., Picone, M., Gonizzi, P., Veltri, L., & Ferrari, G. (2014). Iot-oas: An oauth-based authorization service architecture for secure services in iot scenarios. IEEE sensors journal, 15(2), 1224-1234.

[64] Cruz-Piris L, Rivera D, Marsa-Maestre I, De La Hoz E, Velasco JR. Access control mechanism for IoT environments based on modeling communication procedures as resources. Sensors. 2018 Mar;18(3):917.

[65] Sandhu, R.S., 1998. Role-based access control. In Advances in computers Elsevier, Vol. 46, pp. 237-286.

[66] Kalam, A. A. E., Baida, R. E., Balbiani, P., Benferhat, S., Cuppens, F., Deswarte, Y., ... & Trouessin, G. (2003, June). Organization based access control. In Proceedings POLICY 2003. IEEE 4th International Workshop on Policies for Distributed Systems and Networks (pp. 120-131). IEEE.

[67] Ye, N., Zhu, Y., Wang, R. C., Malekian, R., & Lin, Q. M. (2014). An efficient authentication and access control scheme for perception layer of internet of things.

[68] Kaiwen, S., & Lihua, Y. (2014). Attribute-role-based hybrid access control in the internet of things. In Web Technologies and Applications: APWeb 2014 Workshops, SNA, NIS, and IoTS, Changsha, China, September 5, 2014. Proceedings 16 (pp. 333-343). Springer International Publishing.

[69] Zhang, X., Parisi-Presicce, F., Sandhu, R., & Park, J. (2005). Formal model and policy specification of usage control. ACM Transactions on Information and System Security (TISSEC), 8(4), 351-387.

[70] Park, J., & Sandhu, R. (2002, June). Towards usage control models: beyond traditional access control. In Proceedings of the seventh ACM symposium on Access control models and technologies (pp. 57-64).

[71] Riad, K., & Yan, Z. (2017). Multi-factor synthesis decision-making for trust-based access control on cloud. International Journal of Cooperative Information Systems, 26(04), 1750003.

[72] Gusmeroli, S., Piccione, S., & Rotondi, D. (2013). A capability-based security approach to manage access control in the internet of things. Mathematical and Computer Modelling, 58(5-6), 1189-1205.

[73] Bouij-Pasquier, I., Ouahman, A. A., Abou El Kalam, A., & de Montfort, M. O. (2015, November). SmartOrBAC security and privacy in the Internet of Things. In 2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA) (pp. 1-8). IEEE.

[74] El Bouanani, S., El Kiram, M. A., Achbarou, O., & Outchakoucht, A. (2019). Pervasive-based access control model for IoT environments. IEEE Access, 7, 54575-54585.

[75] Maesa, D.D.F., Mori, P. and Ricci, L., 2017, June. Blockchain based access control. In IFIP international conference on distributed applications and interoperable systems, Springer, Cham. pp. 206-220.

[76] Ding, S., Cao, J., Li, C., Fan, K., & Li, H. (2019). A novel attribute-based access control scheme using blockchain for IoT. IEEE Access, 7, 38431-38441.

[77] Ouaddah, A., Abou Elkalam, A., & Ait Ouahman, A. (2016). FairAccess: a new Blockchain-based access control framework for the Internet of Things. Security and communication networks, 9(18), 5943-5964.

[78] Xue, J., Xu, C., & Zhang, Y. (2018). Private Blockchain-Based Secure Access Control for Smart Home Systems. KSII Transactions on Internet & Information Systems, 12(12).

[79] Xu, R., Chen, Y., Blasch, E., & Chen, G. (2018). Blendcac: A smart contract enabled decentralized capability-based access control mechanism for the iot. Computers, 7(3), 39.

[80] Fotiou, N., Pittaras, I., Siris, V. A., Voulgaris, S., & Polyzos, G. C. (2019, June). Secure IoT access at scale using blockchains and smart contracts. In 2019 IEEE 20th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM) (pp. 1-6). IEEE.

[81] Patil, A. S., Tama, B. A., Park, Y., & Rhee, K. H. (2018). A framework for blockchain based secure smart green house farming. In Advances in Computer Science and Ubiquitous Computing: CSA-CUTE 17 (pp. 1162-1167). Springer Singapore.

[82] Dorri, A., Kanhere, S. S., Jurdak, R., & Gauravaram, P. (2019). LSB: A Lightweight Scalable Blockchain for IoT security and anonymity. Journal of Parallel and Distributed Computing, 134, 180-197.

[83] Novo, O. (2018). Blockchain meets IoT: An architecture for scalable access management in IoT. IEEE internet of things journal, 5(2), 1184-1195.

[84] Hwang, D., Choi, J., & Kim, K. H. (2018, October). Dynamic access control scheme for iot devices using blockchain. In 2018 international conference on information and communication technology convergence (ICTC) (pp. 713-715). IEEE.

[85] Zhang, Y., Kasahara, S., Shen, Y., Jiang, X., & Wan, J. (2018). Smart contract-based access control for the internet of things. IEEE Internet of Things Journal, 6(2), 1594-1605.

[86] Liu, H., Han, D., & Li, D. (2020). Fabric-IoT: A blockchain-based access control system in IoT. IEEE Access, 8, 18207-18218.

[87] Pinno, O. J. A., Gregio, A. R. A., & De Bona, L. C. (2017, December). Controlchain: Blockchain as a central enabler for access control authorizations in the iot. In GLOBECOM 2017-2017 IEEE Global Communications Conference (pp. 1-6). IEEE.

[88] Paillisse, J., Subira, J., Lopez, A., Rodriguez-Natal, A., Ermagan, V., Maino, F., & Cabellos, A. (2019, May). Distributed access control with blockchain. In ICC 2019-2019 IEEE International Conference on Communications (ICC) (pp. 1-6). IEEE.

[89] Pal, S., Rabehaja, T., Hill, A., Hitchens, M., & Varadharajan, V. (2019). On the integration of blockchain to the internet of things for enabling access right delegation. IEEE Internet of Things Journal, 7(4), 2630-2639.

[90] Alphand, O., Amoretti, M., Claeys, T., Dall'Asta, S., Duda, A., Ferrari, G., ... & Zanichelli, F. (2018, April). IoTChain: A blockchain security architecture for the Internet of Things. In 2018 IEEE wireless communications and networking conference (WCNC) (pp. 1-6). IEEE.

[91] Siris, V. A., Dimopoulos, D., Fotiou, N., Voulgaris, S., & Polyzos, G. C. (2020). Decentralized authorization in constrained IoT environments exploiting interledger mechanisms. Computer Communications, 152, 243-251.

[92] Outchakoucht, A., Hamza, E. S., & Leroy, J. P. (2017). Dynamic access control policy based on blockchain and machine learning for the internet of things. International Journal of Advanced Computer Science and Applications, 8(7).

[93] Shafagh, H., Burkhalter, L., Hithnawi, A., & Duquennoy, S. (2017, November). Towards blockchain-based auditable storage and sharing of IoT data. In Proceedings of the 2017 on cloud computing security workshop (pp. 45-50).

[94] Wang, S., Zhang, Y., & Zhang, Y. (2018). A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems. Ieee Access, 6, 38437-38450.

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

[96] Balamurugan, B., Krishna, P. V., Devi, M. N., Meenakshi, R., & Abinaya, V. (2014, March). Enhanced framework for verifying user authorization and data correctness using token management system in the cloud. In 2014 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2014] (pp. 1443-1447). IEEE.

[97] A. B. Sun and T. K. Ji,(2016)``Big data open sharing platform and industrial ecological construction for smart cities," Big Data, vol. 2, no. 4, pp. 69_82.

[98] GE, L., JI, X., JIANG, T., & JIANG, Y. (2019). Security mechanism for internet of things information sharing based on blockchain technology. Journal of Computer Applications, 39(2), 458.

[99] Xue, T. F., Fu, Q. C., Wang, C., & Wang, X. (2017). A medical data sharing model via blockchain. Acta Automatica Sinica, 43(9), 1555-1562.

[100] Liang, W., Tang, M., Long, J., Peng, X., Xu, J., & Li, K. C. (2019). A secure fabric blockchain-based data transmission technique for industrial Internet-of-Things. IEEE Transactions on Industrial Informatics, 15(6), 3582-3592.

[101] Xu, H., He, Q., Li, X., Jiang, B., & Qin, K. (2020). BDSS-FA: A blockchain-based data security sharing platform with fine-grained access control. IEEE Access, 8, 87552-87561.

[102] Al Breiki, H., Al Qassem, L., Salah, K., Rehman, M. H. U., & Sevtinovic, D. (2019, November). Decentralized access control for IoT data using blockchain and trusted oracles. In 2019 IEEE International Conference on Industrial Internet (ICII) (pp. 248-257). IEEE.

[103] Battah, A. A., Madine, M. M., Alzaabi, H., Yaqoob, I., Salah, K., & Jayaraman, R. (2020). Blockchain-based multi-party authorization for accessing IPFS encrypted data. IEEE Access, 8, 196813-196825.

[104] Sun, J., Yao, X., Wang, S., & Wu, Y. (2020). Non-repudiation storage and access control scheme of insurance data based on Blockchain in IPFS. IEEE Access, 8, 155145-155155.

[105] Marangappanavar, R. K., & Kiran, M. (2020, February). Inter-planetary file system enabled blockchain solution for securing healthcare records. In 2020 third ISEA conference on security and privacy (ISEA-ISAP) (pp. 171-178). IEEE.

[106] Shuaib, K., Abdella, J., Sallabi, F., & Serhani, M. A. (2022). Secure decentralized electronic health records sharing system based on blockchains. Journal of King Saud University-Computer and Information Sciences, 34(8), 5045-5058.

[107] Zaabar, B., Cheikhrouhou, O., Jamil, F., Ammi, M., & Abid, M. (2021). HealthBlock: A secure blockchain-based healthcare data managementsystem. Computer Networks, 200, 108500.

[108] Azbeg, K., Ouchetto, O., & Andaloussi, S. J. (2022). BlockMedCare: A healthcare system based on IoT, Blockchain and IPFS for data management security. Egyptian Informatics Journal, 23(2), 329-343.

[109] Oktian, Y. E., & Lee, S. G. (2020). Borderchain: Blockchain-based access control framework for the internet of things endpoint. IEEE Access, 9, 3592-3615.

[110] Rizzardi, A., Sicari, S., Miorandi, D., & Coen-Porisini, A. (2022). Securing the access control policies to the Internet of Things resources through permissioned blockchain. Concurrency and Computation: Practice and Experience, 34(15), e6934.

[111] Han, D., Zhu, Y., Li, D., Liang, W., Souri, A., & Li, K. C. (2021). A blockchain-based auditable access control system for private data in service-centric IoT environments. IEEE Transactions on Industrial Informatics, 18(5), 3530-3540.

[112] Shi, N., Tan, L., Yang, C., He, C., Xu, J., Lu, Y., & Xu, H. (2021). BacS: A blockchain-based access control scheme in distributed internet of things. Peer-to-peer networking and applications, 14, 2585-2599.

[113] Sisi, Z., & Souri, A. (2021). Blockchain technology for energy-aware mobile crowd sensing approaches in Internet of Things. Transactions on Emerging Telecommunications Technologies, e4217.

[114] Kamal, M., Amin, S., Ferooz, F., Awan, M. J., Mohammed, M. A., Al-Boridi, O., & Abdulkareem, K. H. (2022). Privacy-aware genetic algorithm based data security framework for distributed cloud storage. Microprocessors and Microsystems, 94, 104673.

[115] Kumar, R., & Tripathi, R. (2021). Towards design and implementation of security and privacy framework for internet of medical things (iomt) by leveraging blockchain and ipfs technology. The Journal of Supercomputing, 1-40.

[116] Dwivedi, A. D., Malina, L., Dzurenda, P., & Srivastava, G. (2019, July). Optimized blockchain model for internet of things based healthcare applications. In 2019 42nd international conference on telecommunications and signal processing (TSP) (pp. 135-139). IEEE.

[117] Shammar, E. A., Zahary, A. T., & Al-Shargabi, A. A. (2022). An attribute-based access control model for Internet of things using hyperledger fabric blockchain. Wireless Communications and Mobile Computing, 2022.