IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 27 August 2023, accepted 8 September 2023, date of publication 22 September 2023, date of current version 4 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3318487

RESEARCH ARTICLE

Fabrication of Flexible Role-Based Access Control Based on Blockchain for Internet of Things Use Cases

TANZEEL ZAIDI[1], MUHAMMAD USMAN[1], MUHAMMAD UMAR AFTAB[1], (Member, IEEE), HANAN ALJUAID[2], AND YAZEED YASIN GHADI[3]

[1]Department of Computer Science, National University of Computer and Emerging Sciences, Chiniot–Faisalabad Campus, Chiniot, Islamabad 35400, Pakistan
[2]Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University (PNU), Riyadh 11671, Saudi Arabia
[3]Department of Computer Science and Software Engineering, Al Ain University, United Arab Emirates

Corresponding author: Muhammad Umar Aftab (ms.umaraftab@yahoo.com)

This work supported by the Princess Nourah Bint Abdulrahman University Researchers Supporting Project, Princess Nourah Bint Abdulrahman University, Riyadh, Saudi Arabia, under Project PNURSP2023R54.

ABSTRACT The Internet of Things (IoT) connects many objects and allows continuous communication and data sharing has emerged as a revolutionary technology. However, expanding IoT devices has raised concerns regarding data security and access control. Traditional access control mechanisms face challenges in managing access rights, particularly in scenarios where multiple users with the same roles try to access several resources which may lead to conflicting roles. Additionally, there is also an overhead of system performance using traditional approaches. In existing studies, the main problem of conflict roles is not addressed or not even identified appropriately. This paper proposes a framework to address these challenges using blockchain technology and role-based access control with a smart contract implementation on the hyperledger fabric framework. The proposed methodology introduces a role management system that resolves conflicts based on predefined rules and user preferences. It employs a consensus mechanism to determine access permissions, ensuring fairness and accountability. The findings demonstrate that applying the suggested framework eliminates conflicting problems, improves system security and also provides better results in response times for concurrent user requests.

INDEX TERMS Role-based access control (RBAC), smart contract, hyperledger fabric, blockchain based IoT.

I. INTRODUCTION

The Internet of Things (IoT) has quietly and steadily encroached on our lives due to the explosive rise of smart devices and high-speed networks. To achieve the needed functionality and communicate information, it is necessary to remotely operate an enormous number of IoT devices. However, security and privacy concerns cost customers money and impede the growth of the IoT [1].

Access Control (AC) is therefore viewed as one of the crucial mechanisms to ensure IoT security and privacy. Traditional access control methods such as Mandatory Access Control (MAC), Discretionary Access Control (DAC), Role-Based Access Control (RBAC) and Attribute-Based Access Control (ABAC) are used earlier in IoT applications to provide security. Many IoT access control techniques have been developed; however, these approaches rely on a single-server model. The study [2], comprehensively describes the issues for identification of user roles as its first step. Access is then given to the user through authorization after user IDs have been authenticated and verified. The accountability

The associate editor coordinating the review of this manuscript and approving it for publication was Mohamad Afendee Mohamed.

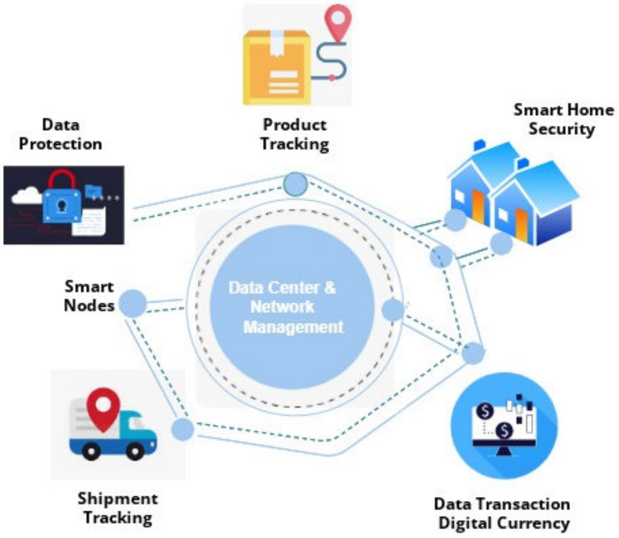


FIGURE 1. Modern day uses cases of Blockchain-based IoT applications.

process tracks activities related to persons in order to identify the person who updates data or a system. Access control systems are intended to prevent unauthorized individuals from misusing resources and to provide information on IoT research subjects. The four key elements of access control systems are discussed in the study [3]. However, the variety and decentralization of the IoT ecosystem make data access an administratively challenging task. The centralized systems have limitations, of single point of failure [4]. Furthermore, it is simple for a malicious or hacked server to change access policies and approve unauthorized access requests. Therefore IoT applications require decentralized, scalable, robust access control systems to provide better solutions for modern-day requirements.

The integration of Blockchain and IoT access control systems has become more and more popular to establish efficient and adaptable distributed IoT access control [5]. The applications of IoT in various areas are used to provide a secure platform to perform several tasks. The use case of IoT is shown in Figure 1. Blockchain employs smart contracts to make distributed access options or permission validation of users by probing the access policies. The integration of blockchain with IoT is promising as it ensures trust and reduces system overhead for IoT systems. It enables the development of a decentralized, reliable and openly verifiable database that will enable distributed trust among billions of connected devices. The trust among connected devices in the blockchain is achieved using consensus protocols. There are several different consensus protocols, including PoW (Proof-of-Work), Practical Byzantine Fault Tolerance and PoS (Proof-of-Stake) used in the studies [6], [7].

In existing studies, the researchers considered an IoT access control system based on the blockchain using access control lists (ACLs). Making ACLs or giving everyone in the system responsibilities is no longer required. In this study, we proposed a blockchain-based access control system for IoT networks utilizing Hyperledger Fabric and built-in features of the blockchain platform. In the proposed system, each device can be represented by a set of preset system characteristics that are distributed by attribute authorities following the device's identification or capability. No one is given access unless it satisfies the requirements of the access policy by sufficient confidence. The proposed system provides a granular level of roles based on AC and defines a three-level access control hierarchy, which includes a root Certificate Authority, intermediate CAs and edge devices as used in the study [8].

The primary goal of this research is to increase trust between entities while getting rid of the requirement for a single reliable authority to manage the trust levels for each entity. This is done by utilizing a decentralized blockchain architecture. A node with the fewest possible permissions is given access right delegation to offer a lightweight solution. The main contributions of this study are as follows:

- For reliable data exchange and authorized access control among users of IoT devices, we proposed a blockchain-based approach.
- RBAC smart contract is used to manage overall access control based on the user's roles to provide a secure gateway for transactions and data access.
- We provide flexible RBAC mechanisms to resolve the conflicting roles problem in the IoT network.
- Authentication and authorization of users is also maintained by our proposed contracts, certificates authority and signature policies.
- System provides flexibility in managing and scaling roles using blockchain technology integrated with IoT.

The organization of the paper is as follows: First, we discuss the concepts of IoT,hyperledger fabric, access control and blockchain in the background section. The literature review thoroughly examines modern access control methods using blockchain technology, including current trends and unresolved challenges. After that, we discuss the RBAC smart contract employed in this study in the proposed solution section followed by the environment setup section. After that, we discuss the results and performance measurements in the Results Section. The conclusion and future work are provided at the end.

II. BACKGROUND

A. INTERNET OF THINGS

The IoT means a network of actual physical things like vehicles, home appliances and other items with Internet connections and data-exchange capabilities. IoT devices are essentially "smart" objects that have the ability to link devices and people remotely as well as manage and control them. However, as IoT devices proliferate, there is a growing demand for efficient access control. The technique

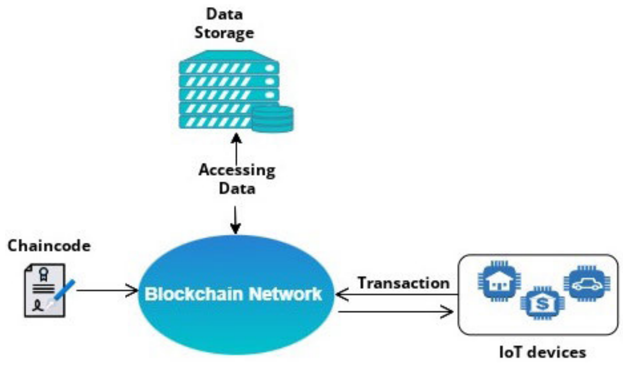


FIGURE 2. Combining IoT, Blockchain network and Chaoncode into a unified system.

of controlling who has access to a system or network and what activities they are permitted to carry out is known as access control. Access control is essential for upholding security and privacy in the IoT context.

The devices that are connected through the internet and provide access to multiple resources lead to security problems. It might be challenging to manage access and authentication because these devices might spread out across several physical locations [11]. To manage access and enforce security policies for IoT networks blockchain technology is used. It offers a safe gateway for carrying out various operations utilizing smart contract capability.

The implementation of efficient access control in IoT systems is fraught with difficulties. The massive volume and variety of the involved devices present one significant problem and lead to delays in time as discussed in the study [12]. Sensors, actuators, gateways, and other devices with various sizes, shapes and functions may be a part of IoT networks. The fact that these devices could be situated in many physical locations makes it challenging to control access and authentication. The integration of IoT, smart contracts and blockchain is shown in Figure 2.

The usage of many communication protocols and data formats by IoT devices can also make it more difficult to manage access and enforce security policies. Numerous access control mechanisms for IoT systems have been suggested as solutions to these problems. These mechanisms include both conventional means of access control, such as passwords and tokens, as well as more contemporary ones, including biometric identification and digital certificates. Several access control models are used to provide more security such as RBAC, ABAC, MAC and DAC are discussed in the study [14]. However many IoT platforms and protocols integrate security features like encryption, secure boot, and secure firmware updates to guarantee that devices and data are shielded from unauthorized access.

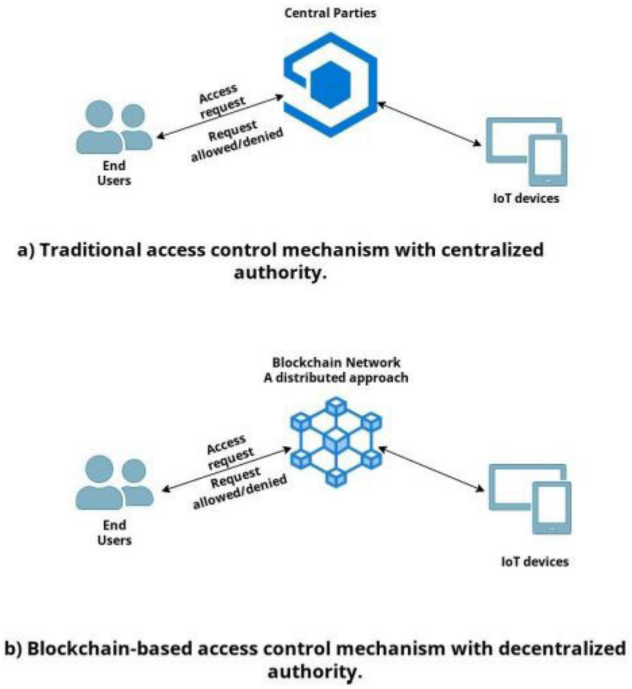


FIGURE 3. Access control with centralized and decentralized authorities.

B. ACCESS CONTROL AND BLOCKCHAIN

Access Control (AC) is a fundamental security measure that limits as well as allows resources to be available based on predefined policies. In the context of IoT, access control is a crucial component for guaranteeing the privacy and security of IoT devices and data [15]. However, due to the dynamic and decentralized nature of IoT systems with traditional access control mechanisms which are limited. These limitations are removed using blockchain technology which offers a decentralized and impenetrable platform for controlling access control rules and permissions. AC in the IoT can strengthen in terms of security and dependability by utilizing the immutability and transparency of the blockchain.

Blockchain is a decentralized digital ledger that securely and openly logs transactions. It consists of several blocks, each of which is composed of a number or collection of verified transactions. Once a block is added to a chain, it cannot be altered or withdrawn easily. The blockchain-based and traditional access control model is shown in Figure 3.

The blocks in blockchain technology have a header and body, and the block header contains the previous hash, the nonce, the Merkle root and the timestamp. The cryptographic hash function is used to link the blocks together. Blockchain comes in a variety; among them are "public", "private", and "consortium." Whenever the blocks have been coupled together and the validated transactions have been added by the miner nodes.

The blockchain can be simply classified as public and private, permissionless and permissioned blockchain. In a public

TABLE 1. A brief overview of studies on blockchain-based access control for IoT.

Studies	Approach	Blockchain Platform	Access Control Mechanism	Application Domain
Sharma and Park,[4]	Hybrid network architecture	Private blockchain	ABAC	Smart city
Ding et al.,[5]	ABAC	Private blockchain	ABAC	IoT
Islam and Madria,[6]	Permissioned blockchain AC	Hyperledger Fabric	RBAC	IoT
Saha et al.,[7]	Access control protocol design	Private blockchain	ABAC	Healthcare
Iftekhar et al.,[8]	Hyperledger Fabric using AC system	Hyperledger Fabric	RBAC	IoT
Algarni et al.,[17]	Secured access control	Private blockchain	RBAC	IoT
Sun et al.,[18]	AC system design	Public blockchain	RBAC	IoT
Nakamura et al.,[19]	Capability-based access control	Ethereum	CBAC	IoT
Liu et al.,[20]	Fabric-IoT AC system	Hyperledger Fabric	RBAC	IoT

blockchain, anyone can join a blockchain network and see the data stored there. For instance in Bitcoin and Ethereum while the Private blockchain, restricts access to blockchain data to a small number of chosen companies like MultiChain and Hyperledger Fabric, the detailed survey on the blockchain is conducted in the study [16]. On the other side, permissioned blockchain networks segregate users based on their access levels, whereas permissionless blockchains allow anybody to join and participate in a consensus method. A brief overview of studies on Blockchain-based Access Control for IoT is given in TABLE 1.

The blockchain is the perfect solution for safe and tamper-proof record keeping and data management because of its immutability and transparency nature. Blockchain uses sophisticated algorithms and cryptographic methods to guarantee the security and integrity of the data stored in it. Bitcoin and other digital currencies are built on blockchain technology which is also used in the financial, healthcare and supply chain management sectors.

The study [17], proposed a blockchain-based secured access control system for IoT. The past studies used several approaches to tackle the security issue in access control systems. The studies presented outline various methods for creating blockchain-based IoT access control systems.

The study [18] presented a blockchain-based IoT access control system that uses a capability-based access control model. This study provides access to resources and improves security across the system. The study [19], proposed an Ethereum blockchain-based capability-based access control scheme for IoT. The proposed Fabric-IoT is a blockchain-based access control system for IoT that uses Hyperledger Fabric and smart contracts to enforce access policies in a study [20]. In our proposed model, we used a role-based access control model to deal with the above issues and used blockchain technology with the smart contract.

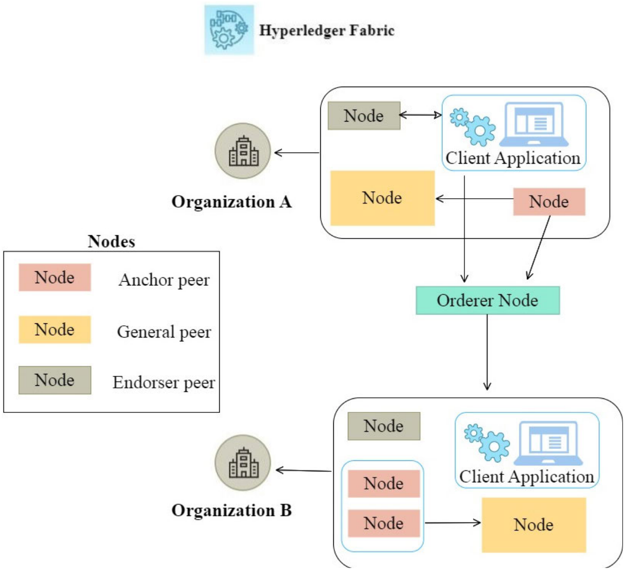


FIGURE 4. Generic workflow of hyperledger fabric between two organizations.

C. HYPERLEDGER FABRIC

The enterprise-grade distributed ledger platform called Hyperledger Fabric enables businesses to create and manage permissioned blockchain networks. It is a flexible blockchain platform that can be tailored to meet an organization's unique requirements. The network is made adaptable, scalable, and secure thanks to Fabric's modular architecture. The fabric supports SC which is written in several programming languages, including Java, Go, and JavaScript is one of Hyperledger Fabric's core features. This facilitates the development and deployment of smart contracts for use on the fabric network.

However, the fabric offers a strong access control system that enables businesses to manage who has access to their blockchain data. It is created to interact with current enterprise systems and databases, making it simpler for organisations to adopt blockchain technology [21]. This is built to interact with current enterprise databases and systems, making it simpler for businesses to adopt blockchain technology into their current operations and log into their current work processes auditable. The auditable access control system powered by blockchain is used for sensitive data in the study [22], To overcome the privacy issues, low transaction throughput, consistency issues and wastage of resources fabric tool is formed. The generic workflow of hyperledger fabric is shown in Figure 4.

D. COMPONENTS OF HYPERLEDGER FABRIC

The peer nodes, clients, ordering service, membership, and Chain code are some of the main parts of the hyperledger fabric. This component serves a specific function for various

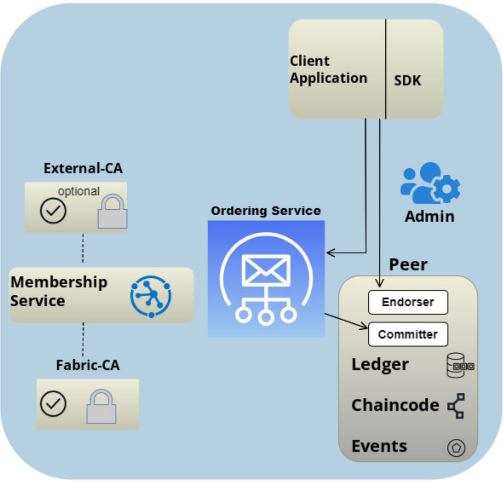


FIGURE 5. Working of hyperledger fabric architecture.

goals. The four primary phases of the transaction flow are endorsement, ordering, validation, and committing. There are two types of blockchain: public and private or permissionless and permissioned blockchain respectively [6]. One of the systems for permissioned blockchains is called Hyperledger Fabric, and it was created by the open-source Linux foundation community. Peer nodes, clients, an ordering service, membership, and Chaincode are some of the main parts of Hyperledger Fabric. Each component serves a specific function for various goals discussed in a study [7]. The four primary phases of the transaction flow are endorsement, ordering, validation, and committing. The majority of users in the peers-only Hyperledger Fabric network are businesses. The graphical representation of hyperledger fabric is shown in Figure 5.

Organization: The simple Hyperledger Fabric network is made up of multiple organizations called Org1 and Org 2. Organizations are entities that participate in the blockchain network, allowing for flexible governance, trust establishment and control over data and business processes.

Peer: Peers serve as the fundamental building blocks of a network host ledgers and smart contracts.

Channel: Channels are private communication channels between two or more users. It allows organizations to create separate environments for confidential collaboration while being part of a larger blockchain network.

Orderer: The Orderer (O) is in charge of putting transactions into Blocks and distributing them to Anchor Peers around the network.

Ledger: The Validation Ledger (L) holds vital data about business objects. The ledger represents the distributed database that stores and maintains a complete history of all transactions and data shared across the network. It serves as a reliable source of truth for all participants within the network, ensuring transparency, immutability and consistency. Each member maintains a copy of the ledger, and all updates to the ledger are performed through a consensus mechanism agreed upon by the network participants.

The global state and the transaction log are the two fundamental parts of the ledger in Hyperledger Fabric [8]. While the transaction log keeps a chronological record of all transactions, the world state depicts the current condition of all resources or data within the network.

Smart Contract or Chaincode: The languages such as Go, Node.JS, or Java programs called a "Smart Contract" or "ChainCode" CC executes a predetermined interface.

Certificate Authority: The term "Certificate Authority" (CA) refers to a CA that offers identity registration services or links to LDAP as the user registry and issues Enrollment Certificates (ECerts) as well as certificates for certificate renewal and revocation.

Client Application: The smart contract is activated by the client, the final user of an application, by sending a request on the channel. The created read-write sets are delivered to the client. The study [9], introduces several business applications and explains how IoT technology interacts with these applications.

III. LITERATURE REVIEW

The exponential growth of IoT necessitates an elevated level of distributed access control. Blockchain technology offers several advantages in this regard, including decentralization, data encryption, scalability and immutability. Smart contracts, which form the foundation of the blockchain, provide it with additional robust features and create a secure and dependable working environment for apps. Therefore, several researchers offered a range of IoT AC approaches by merging them with blockchain and smart contracts, building on existing access control methods. To improve security, scalability and privacy in applications for smart cities the authors of the study [4], propose a hybrid network design that integrates both public and private blockchains. To safely and effectively manage access to IoT devices and data the authors propose an ABAC approach in a study [5] and resolve the security issues.

The study [6] provides an IoT device and data access control system built on a permissioned blockchain and utilises smart contracts to provide more reliability and security. The issue of lack of confidence was resolved and the system is strengthened by this decentralized and scalable access control mechanism. Permissioned blockchain, controlled by private groups, poses a high risk of collision and overriding consensus. The authors in a study [7] developed a new access control system using private blockchain technology to exchange private and sensitive data securely and get rid of private groups. The ECC-based signature method recommended by the scheme was employed by the authors. In the study [8], the authors addressed the advantages of IoT device solutions provided by Hyperledger Fabric blockchain technology. Its capacity to employ these algorithms on a broad scale is a drawback. To overcome the IoT problems the authors in the study [9] give the potential of IoT and

TABLE 2. Overview of problems and approaches in previous studies.

Studies	Problems	Approach
R. Xu et al.,[2] and Sharma et al.,[4]	High latency, bandwidth, bottlenecks, security and privacy, and scalability.	To prove the integrity of data uses. Hash and digital signature.
Ding et al.,[5]	Security of data and information.	Authorization of users.
Islam and Madria [6]	Access control decision-making process.	Access Control List (ACLs) policies.
Sun et al.,[18],Ok-Chol R et al.,[30]	Roles Authorization and Authentication	Separation of duties Key pair authentication
Nakamura et al.,[19]	Conflicting transaction problem	CATP-Fabric operated and processed in three modules: 1) The key-based grouping module. 2) Transaction preprocessing module. 3) Conflicting transaction resolution module.

the challenges to addressing business problems related to them. To identify the problems faced in IoT applications the authors in the study [10], present a survey of access control mechanisms for researchers to resolve traditional problems faced in IoT systems and in a study [11] presents a hybrid deep learning method for resolving the problems that are identified authentication is one major issue is placed in that survey. The findings in the study [12] help us comprehend how IoT may optimise data processing and resource utilization. They also give a general outline of the difficulties involved in managing larger data transfers over a network. The authors discuss cloud access control mechanisms, decentralized Internet of Things access control systems and the possibility of blockchain for several use cases discussed in the study [13] and [14]. The smart contract enforces the policies within the network to restrict a user in the study [15] for the healthcare system. The summary of previous studies taken to explore problems and approaches is shown in TABLE 2.

In a study [16], the authors present a blockchain-based access control model that can address the potential of blockchain technology while lowering the dangers related to centralized access control systems. The study [17] proposed an architecture based on a multi-agent system and employing a distributed private blockchain for managing the delivery of a secure, lightweight and IoT access control. The primary goal of the suggested method is to protect the whole IoT architecture, including cloud computing, fog nodes, and communication between IoT devices. The Comparison of past studies based on their methodology and limitations of blockchain is shown in TABLE 3.

The authors of the study [18] proposed a native blockchain ledger that is lightweight for each IoT domain. To avoid utilizing delegate nodes to carry out the access control logic, they developed a lightweight blockchain that enables the majority of IoT domains to deploy their blockchain network on their IoT devices. But its drawback is, that as the number of edge devices rises, more malicious users assault the network.

The study [19], presents a CapabilityBased Access Control System (CapBAC) that uses Ethereum smart contracts. To achieve more fine-grained access control and more flexible token management than the current blockchain-enabled decentralised CBAC system. IoT access control utilizing blockchain technology should be planned and implemented using the fabric-IoT access control system in the study [20], which is built on Hyperledger Fabric by using the ABAC mechanism. The Internet of Things fabric can track records, provide dynamic access control management and address AC issues. The Distributed nature makes it easier to conduct secure transactions. A trust-based access control paradigm is proposed by a study [21] for Internet of Things networks, integrating a blockchain-based ABAC mechanism with an additional TRS for making dynamic access choices which makes the system more robust. An auditable ABAC paradigm is presented in the study [22] for managing access controls for small IoT application data using a private blockchain. This approach ensures transparency and accountability through records that are preserved.

The authors discuss the difficulties of adopting blockchain-based IoT solutions as well as the capacity of blockchain technology to improve IoT system security and privacy in the study [23]. To manage the assignments of underlying operational authorizations in an IoT context for smart homes, a decentralised publish-subscribe architecture based on ledgers was developed in the study [24]. A Proof of Concept implementation supporting the proposed architecture uses smart contracts to maintain the integrity of the administration and adds the inherent benefits of blockchain's distributed and transparent nature. In the study [25], through the use of the blockchain with smart contracts enabled, the authors offered a distributed and reliable access control solution for the Internet of Things. RBAC paradigm is suggested in the study [26], for controlling user-role rights within an enterprise using a blockchain-based smart contract. The drawback of the challenge-response procedure is every time a user uses a service that requires a digital signature there is an overhead in this study which is addressed by the author in a study [27]. They suggest flexible role allocation by presenting a DFRBAC model to acquire access to the activity report security inspection function, which can ensure the architecture's overall security. A comparison of recent studies in terms of security constraints, framework and conflicting roles is shown in TABLE 4. which demonstrates that while recent studies include a variety of access control and security constraint frameworks, none of them specifically address the issue of competing roles within those contexts.

The conditions for authorization or the standards for duty assignment are not discussed. CATP-Fabric, a permission blockchain technology that allows conflicting transaction

TABLE 3. Comparison of past studies based on their methodology, description and limitations in terms of blockchain.

References	Methodology	Description	Limitations
[4]	Hybrid network architecture	They suggest a distributed hybrid architecture in this study for a network for legitimate smart cities.	Their suggested strategy still has several drawbacks, such as ineffective edge node deployment and disabling of caching at the edge nodes.
[5]	Consortium nodes	Enable IoT devices independent of the consensus of the blockchain network. Reduces the total computational and communication overhead.	Their proposed method is only effective when IoT systems have a limited amount of energy and computational power.
[6]	IoT Testbeds	IoT devices exchange resources (data) with outside parties and offer granular access control.	When more attributes are needed to satisfy ABAC policy, reducing latency remains a difficult issue.
[7]	Elliptic Curve Cryptography (ECC) based signature	To be protected from numerous well-known assaults.	The Security of the proposed scheme is dependent on solving the ECDLP becomes intractable.
[8]	Model 4 of the Raspberry Pi using the ARM64 architecture.	Explored the IoT device solutions that Hyperledger Fabric's blockchain technology offers.	Its weakness is the capability to use these algorithms on a large scale.
[18]	Multi-agent Systems. Blockchain Managers	To handle the supply of lightweight, decentralized safe access management for an Internet of Things system.	Big header problem.
[19]	Policy decision point IoT network.	Make a small, local blockchain ledger for each IoT domain.	As the number of edge devices rises, more malicious users will assault the network.
[20]	Ethereum smart contracts	This method provides capability tokens in units of action to allow more flexible token management.	Access control may become unreliable if opponents can alter the policies without altering the URL links.
[21]	Combines blockchain technology with the ABAC model.	Employs a distributed architecture that can offer the physical network.	Future studies might help the integration of more IoT applications and boost the fabric scalability of IoT.
[22]	Trust and Reputation System	Provides dynamic and flexible access control by quantifying each network node's trust and reputation ratings.	Bootstrapping and attribute registration processing issues.
[23]	Auditable access control model	Support actual application situations in IoT settings while maintaining high throughput and assuring private data security.	Due to the features and restrictions of the Fabric platform, it is ineffective in defending against malicious assaults.
[24]	Proof-of-concept	Publish-subscribe based architecture is used to oversee the distribution of underlying operational authorizations.	Multi-user and multidevice environment doesn't provide monitoring even after being granted, the access occasionally prevents quick change.
[25]	Access Control Models	Focus on partially decentralized and fully decentralized.	The blockchain's bandwidth overhead problem can cause latency to grow.

TABLE 4. Comparison of studies in terms of security constraints, framework and conflicting roles.

References	Year	Security Constaint	Framework	Conflicting Roles
[31]	2022	Attribute-Based Access Control (ABAC)	Hyperledger Fabric blockchain	No
[32]	2022	Delegation of Authorization Policies	Blockchain	No
[35]	2022	Control of access between domains and systems	RBAC in Multi-Domain	No
[36]	2022	Effectiveness of Policy Designers	Hybrid Approach RBAC and ABAC	No
[37]	2022	Identification of Authorization Policy Conflicts	Multi-Cloud Environment	No

execution in IoT networks, has been suggested in the study [28]. The proposed scheme only worked for the read-only transaction. The study [29], an approach for analyzing the effects of blockchain on corporate performance and governance in freshly established firms is suggested as a theoretical contribution. Due to the use of uncertain values, such as fuzzy or grey methods, it is advisable to evaluate the model variables with uncertain techniques to generate more realistic results.

The authors in the study [30], offered a model that integrates the recommended model and a method for determining a user's ownership of a role to control access to data stored in cloud storage. The proposed approach only implements the separation of duties restriction for a single-session scenario. In the study [31], ABAC using the Hyper Ledger Fabric blockchain model ABAC-HLFBC has been proposed to provide a more secure platform on an attributes basis. The authors in the study [32], provide an approach that helps advance authorization capabilities, including contextual delegation, to be easily accessible and decouples authorization logic from a smart contract's fundamental functionalities. The suggested method uses more gas to process delegation requests as their users increase.

IV. PROPOSED METHODOLOGY

The Internet of Things is growing rapidly and requires the standardization of distributed networks. Existing research demonstrates that traditional access-control methods give the IoT system security and manageability [33]. However, these systems rely on central authority administration, which raises problems including a single point of failure, poor scalability and a lack of privacy. To address these problems, many researchers have proposed employing blockchain technology to provide decentralised access control study [34] presents the benefits of using blockchain-based access control. However, these models still struggle with issues like low scalability and excessive computational complexity. To make sure the safety and user authentication access control can be employed as the primary safety measure.

The traditional access control models face issues with single-point failure and provide less security. The study [35], provides blockchain-based authentication for smart city applications. The issue of verifying the accuracy of the stored policies or identifying potential inconsistencies between recorded policies has not yet been solved and they considered it for future work. The information needed for access control must be stored by a single central authority, which also verifies access permissions so it's necessary to check policies and enforce them to each user. The applications for smart cities involve a wide range of stakeholders, including the government, service providers and citizens. The roles and levels of access that each stakeholder has within the system may vary. When access control policies for various stakeholders are mismatched or inconsistent, conflicting roles may result. The communication between users and resources is shown in Figure 6. To address the issue of conflicting roles in such scenarios, we propose a solution in the form of an access control system for the IoT that utilizes blockchain technology. Our proposed solution leverages the hyperledger fabric platform for the development and execution of this system, ensuring robust and secure management of access rights. By utilizing blockchain technology, our aim is to provide a decentralized and immutable framework that effectively resolves conflicts and enhances the overall security and efficiency of IoT access control. Authorization of the user could be done with a secure signature-based authentication key exchange access control mechanism using smart contracts.

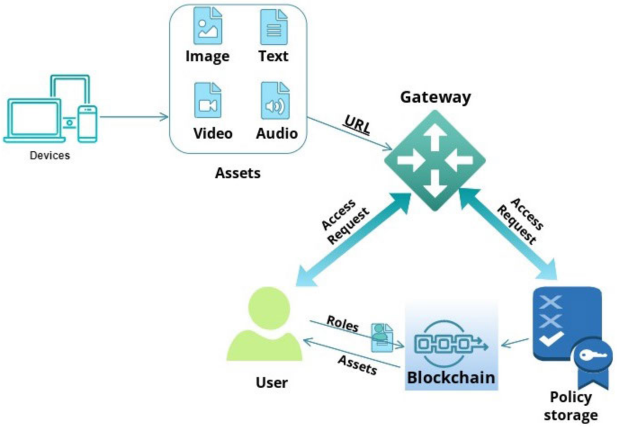


FIGURE 6. Communication between user and resources.

A. SYSTEM DESIGN AND ARCHITECTURE

The four elements make up an IoT access control system based on blockchain: smart contracts, a gateway, users and devices which are shown in Figure 7. In this study, we look into the best ways to implement an RBAC framework across organisations and protect the user-role assignments of those organizations. To attain these objectives the RBAC smart contract authentication mechanism [36]. The user role assignments are generated by a smart contract (SC) and then broadcast on the blockchain. For the quick, simple, and secure production of user-role tasks, the SC provides a variety of options. The usage of the blockchain allows for transparency in the roles generated while protecting user privacy. It also serves as a synchronization point for service-providing organizations to validate the asserted roles. There are two iterations of the framework. We initially created the Fabric Development Network to facilitate the creation of smart contracts. Second, to verify the RBAC smart contract logic we develop a Fabric Sample Network. The authors of the study [37] highlight the challenges of managing authorization policies across multiple cloud environments and propose a formal model to detect policy conflicts.

The study [38], evaluates the proposed approach on a prototype smart home system and demonstrates its effectiveness in preventing unauthorized access and ensuring data privacy. The study [20] shows the integration of blockchain and ABAC for IoT access control, introducing a device authority model and fabric-IoT. In contrast, our proposed approach emphasises RBAC in a Hyperledger Fabric blockchain, addressing conflicting roles by defining roles and hierarchies,

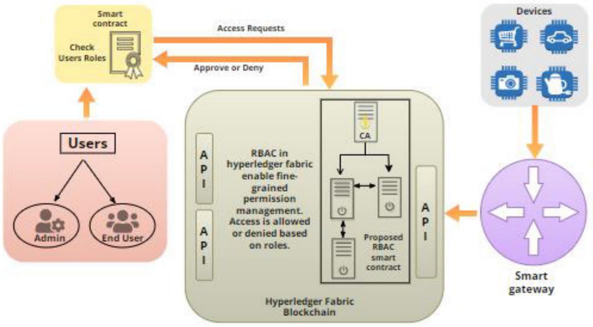

Smart contract
Check Users Roles
Access Requests
Approve or Deny
Devices
Users
Admin
End User
A P I
A P I
RBAC in hyperledger fabric enable fine-grained permission management. Access is allowed or denied based on roles.
CA
Proposed RBAC smart contract
A P I
Hyperledger Fabric Blockchain
Smart gateway

FIGURE 7. **Proposed RBAC framework for IoT networks using blockchain architecture.**

improving security and ensuring correlated duties. The workflow primarily consists of four phases.

PHASE 1: Administrators and ordinary users are the two types of users in this system. The admin is in charge of overseeing the blockchain system and operating the code for the smart gateway. The administrator requires a certificate to access the blockchain system.

PHASE 2: The gateway is a link between devices and the blockchain system that accepts messages from devices and adds the addresses they contain to the blockchain without putting excessive load on the system.

PHASE 3: The Smart Contract enforces the policies to grant access against user requests. The role-based access smart contract uses signature type and in our methodology represents as a (Sign_RBAC).

PHASE 4: The Hyperledger Fabric platform, which is used to build the blockchain, leverages smart contracts to enable access control. Consumers can use a smart gateway to access the blockchain system's API.

IoT devices are crucial to buyer-seller transactions in a Hyperledger Fabric-based blockchain technology control over access system. These devices enable real-time monitoring, smart contract interaction, data collection and transmission [39]. They also provide identity verification, automated payments, secure communication and data transmission. IoT devices enable automatic payments based on predetermined criteria and execute smart contract logic. It collects real-time data about goods or assets and verifies the identities of participants briefly shown in the study [40]. IoT devices improve the efficiency, transparency and dependability of buyer-seller transactions by using the security and transparency of blockchain technology, thereby expediting the whole process and reducing latency.

B. BLOCKCHAIN WITH SMART CONTRACT

Blockchain technology provides a secure, transparent access control framework, enhancing response time and security based on roles. Smart contracts encode access control rules, ensuring users can only perform actions aligned with their roles. Blockchain's transparency ensures an immutable record of transactions, accountability, and deterrence of unauthorized access attempts. This distributed nature expedites access decisions, resulting in improved response time. The decentralized and transparent mechanism for executing and enforcing agreements is made possible by blockchain technology and smart contracts. A distributed, unchangeable platform called the blockchain is used to carry out smart contracts and secure their results [41]. The immutability of blockchain ensures that once a smart contract is executed and recorded, it cannot be altered, providing a high level of security and eliminating the risk of fraud. Smart contracts on the blockchain eliminate the need for intermediaries, fostering trust between parties as the execution of the contract is automated and based on predefined rules. The automation of contract execution reduces the need for manual intervention, streamlining processes and reducing costs.

The authors of the study [42], give an in-depth overview of the different ways that blockchain technology might be used to enhance the construction process. They talk about the advantages of implementing blockchain in fields like asset management, project management, payment systems and supply chain management. The article also discusses the difficulties related to applying blockchain technology in the construction sector.

The author provides a comprehensive analysis of the decentralized financial ecosystem, discussing various applications such as lending, trading and asset management in a study [43]. The authors highlight the benefits of using smart contracts, including improved efficiency, automation and transparency. The authors also explore the difficulties of smart contracts, including the need for formal verification, regulatory concerns and issues with scalability [44].

C. HANDLING CONFLICTING ROLES

In the study [20], the authors use an attribute-based approach which may lead to complexity in a trans-organizational environment and within organizations where job changes occur frequently, implementing ABAC can be complex and unsuitable. ABAC enforces access control based on subject and context attributes. However, this challenge can be addressed by employing RBAC. RBAC introduces the concept of roles to effectively and flexibly manage the assignment between subjects and permissions. RBAC offers significant advantages in terms of flexibility and ease of management, making it widely utilized in industrial and commercial application systems. The first step in resolving role conflicts is to separate roles. This can be accomplished through RBAC testing, which comprises evaluating and analysing the current roles, permissions, users, and resources in the organisation. RBAC testing can be used to identify and report any gaps or redundancies in the access and authorization structure as well as any potential or role conflicts. Role analytics, role mining and role engineering software are examples of technologies that can be used to do RBAC testing manually or automatically. Some authors also work to resolve conflicting roles and conflicting permissions problems. They automate their schemes so that the conflict of interest should be handled through the system itself rather than the administrator manually handling it [45], [46].

In our proposed methodology, the solution is based on Hyperledger Fabric as an underlying blockchain for storing control mechanisms or policies and provides functions to automate effective roles from RBAC smart contracts to permission management at different levels for distinct events. Roles encompass permissions linked to user IDs, granting users access to specific data based on their roles. This method effectively addresses role conflict problems and ensures authorized data access aligned with individual responsibilities. We formulate the RBAC model with roles and permissions scenario from the supply chain management example. We provide proof that demonstrates that conflicts of role do not occur based on the defined permissions. We denote the roles as R and the set of permissions as P. Seller which is represented as Se, Distributor as Di and Buyer as$B$. Each role r is associated with a set of permissions P_r.

$$R = \{Se, Di, B\}$$

P = { CreateProduct, UpdateProduct,
TrackInventory, InitiateShipment,
ViewAvailability, PlaceOrder }

We define the permissions associated with each role as follows:

$$P_{\text{Se}} = \{\text{CreateProduct}, \text{UpdateProduct}\}$$
$$P_{\text{Di}} = \{\text{TrackInventory}, \text{InitiateShipment}\}$$
$$P_{\text{B}} = \{\text{ViewAvailability}, \text{PlaceOrder}\}$$

Now, it's proven that no conflicts of roles occur based on the defined permissions.

In the supply chain RBAC scenario with the defined roles and permissions, no user has conflicting roles. Suppose there exists a user u who has conflicting roles. This implies that u is associated with two roles r_1 and r_2, where $r_1 \neq r_2$, and both r_1 and r_2 have permissions that could lead to a conflict. Without loss of generality, assume r_1 = Seller and r_2 = Distributor. This would mean u can both create/update products and track inventory/initiate shipments. However, based on the permissions assigned to the roles, the actions associated with these permissions are disjoint. That is, there is no action that requires both creating/updating products and tracking inventory/initiating shipments. Therefore, by the definition of the permissions associated with each role, there is no conflict of roles for any user u.

D. RBAC MODEL WORKFLOW

The RBAC system verifies the authorization by examining the permissions for the access request associated with the user's assigned role. The access is permitted or denied based on the checking roles. The RBAC system enforces the allowed access if access is given by enabling users to carry out the specified actions or access the resources. The RBAC approach also permits modifications to roles and permissions as necessary. RBAC ensures adequate access control and improves system security.

The RBAC used in the study [30], uses a cloud environment which causes limited privacy and potentially affects user experience in a cloud environment where quick response times are crucial. The RBAC simplifies the process by assigning permissions to roles and reducing complexity in dynamic environments with frequent job additions and deletions.

However, the permissions combine objects (obj) and operations (ops). Let $p$represent the set of all permissions. R represents the set of all roles. U represents the set of all users. Each permission $p \in P$ is associated with a unique identifier or name. Permissions Set: $P = \{p_1, p_2, \ldots, p_n\}$ Roles Set: $R = \{r_1, r_2, \ldots, r_m\}$ Users Set: $U = \{u_1, u_2, \ldots, u_k\}$ Permissions for Role r: $Permissions(r) = \{p \mid p \text{ is associated with role } r\}$ Roles for User u: $Roles(u) = \{r \mid r \text{ is associated with user } u\}$. RBAC employs access control rules that govern how permissions are assigned to roles and how roles are associated with users. A basic access control rule can be represented as:

$$u \text{ is granted if } r \text{ is a role of } u \text{ and } p \in Permissions(r)$$

When a user u requests access to a resource protected by permission p, the RBAC system checks if any of the roles associated with u have permission p. If user u is granted permission p, it means that at least one of their associated roles has permission p. This can be expressed as: $p \in \bigcup_{r \in Roles(u)} Permissions(r)$. So in this way, the permissions can be obtained with the availability of The variety Objects O and Operations Op → permission $= 2^{O \times Op}$. In addition, the permissions are available for multiple roles. Similarly, the roles can have multiple permissions. Permissions are given to roles in RBAC, and roles are linked to users. A many-to-many relationship between roles and permissions is implied by the predefined permissions. Additionally, to accurately portray authorized users α U, it's crucial to confirm the many-to-many relationships between them. ASINGP means Assign Permissions and ASINGU represents Assigns Users.

$$\text{ASINGP} \subseteq P \times R$$
$$\text{ASINGU} \subseteq \alpha U \times R$$

The RBAC model provides a systematic approach to managing user access within a system. The workflow diagram of the role-based access control model is shown in Figure 8. The workflow of RBAC involves several steps. First, roles are identified based on responsibilities and permissions. Users are then assigned to specific roles and permissions are associated with each role. The hybrid RBAC model was proposed in the study [47], which automatically handles the permission creation and assignment processes. In that scheme, the RBAC model dynamically performed its operations and the scheme was designed for IoT-based traffic systems.

A number of procedures must be completed to construct a secure and controlled network. First, as stated in a relevant

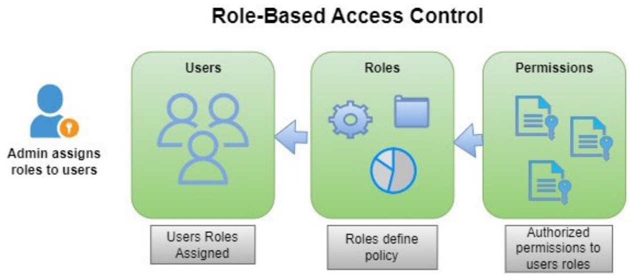

Role-Based Access Control
Admin assigns roles to users
Users
Roles
Permissions
Users Roles Assigned
Roles define policy
Authorized permissions to users roles

FIGURE 8. Workflow of Role-based access control model.

study [48], roles such as Admins (A) and Users (U) and their related rights are specified. Smart contracts are created to implement access control policies using the specified roles and permissions. RBAC model in which nodes are mapped into channels according to their individual roles, is supported by the network structure. To build, manage, authenticate and authorize identities based on RBAC principles. The Hyperledger Fabric integrated identity management solution, which includes Fabric CA and Fabric SDK is used shows in a study [49].

The initial static RBAC policy is enhanced dynamically while ensuring that internal users abide by the system security regulations off-policy learning. To provide an end-to-end proof of concept for their architecture in the study [50], they deploy their implementation within the smart grid space and more especially within a Distributed Energy Resources (DER) ecosystem. The existing studies show that without blockchain technology several models fail to maintain their accountability, scalability and performance. The blockchain resolves all the issues and provides more effectiveness and security for these models. With the help of blockchain technology, many organizations, businesses and IoT application users provide their services in a healthy environment.

In the study [51], the perspectives of the characteristics, elements, features and support for access control policy models of the blockchain systems are specified in depth. General information about blockchain access control systems is provided in this article. The characterization of the access control model for the blockchain technology environment is shown in TABLE 5. The table used low, medium and high terms to show how several models perform well or moderately in different criteria. The term yes and no refers to the reliability, adaptability and accessibility of multiple constraints. Due to their complexity and dynamic nature, traditional access control methods are not ideal for IoT environments. The importance of safeguarding sensitive essential data in IoT systems is the main emphasis of the study [52]. The context-Aware IoT Rule Based Access Control Algorithm is a suggested algorithm that is presented as a solution to this problem. A novel use of machine learning for hybrid access control that significantly improves the security of SCADA-enabled IIoT networks is presented in a recent study [53]. Their efforts have the potential to address changing security issues in industrial systems.

TABLE 5. Characterization of access control models parameters for blockchain technology environment.

Criteria	Access Matrix	ABAC	RBAC
Context Sensitive	No	Low	Medium
Authorization Based	No	Yes	Yes
Dynamic Control	No	Medium	Yes
Policy Specification	Low	Low	Yes
Policy Permission Representation	Subject, Access Mode	Attribute, Object, Access Mode	Roles,Access Mode, Object
Fine-Grained	No	Yes	Yes
Policy Enforcement	Low	Low	High
Ease of Use	Medium	Medium	High
Understandability	Simple	Simple	Simple
Applicability	Medium	High	High
Complexity	Low	Medium	Medium

TABLE 6. Software environment setup.

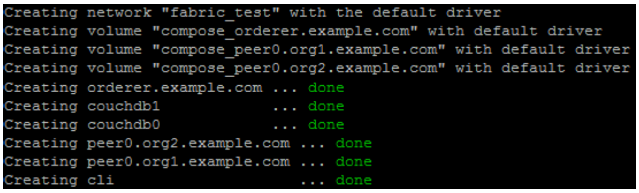
Software	Versions
docker	v20.04.2
docker-compose	v1.25.0
node	v10.19.0
go	v1.18.3
hyperledger fabric	v1.4.7

E. ENVIRONMENT SETUP

The environment setup of this paper is carried out on the Linux operating system Ubuntu 64-bit. A Hyperledger Fabric Linux foundation tool is used to deploy the blockchain test network. The development and deployment of blockchain networks on Hyperledger require some prerequisites as shown in TABLE 6 the software environment setup. Firstly, we install prerequisite software such as Docker and Go programming language. Docker allows for containerization, ensuring consistent deployment across different environments. We make sure the network is set up and thoroughly test and package the chaincode. Install and instantiate it with the appropriate endorsement policies and access controls before deploying a smart contract in Hyperledger Fabric. To ensure the effective deployment and operation of our blockchain network. A chaincode needs to be created. Golang is the language used in our chaincode source code.

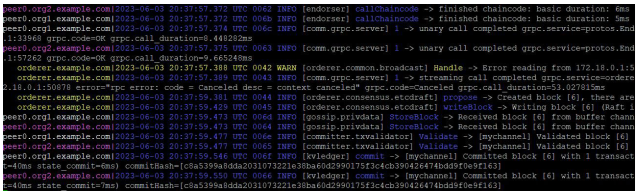
$$\text{Code(Function}\ldots) \rightarrow \text{Chaincode}$$

The Hyperledger Fabric test network is set up after installing the prerequisite we create a channel which includes peer nodes, ordering service nodes and certificate authorities. The outcome of the fabric test network's development is shown in Figure 9. These nodes play crucial roles in maintaining the blockchain network. In our environment setup, we configure the network by defining the cryptographic material, channel configuration and consensus mechanisms to provide the fundamental building blocks for privacy,



```
Creating network "fabric_test" with the default driver
Creating volume "compose_orderer.example.com" with default driver
Creating volume "compose_peer0.org1.example.com" with default driver
Creating volume "compose_peer0.org2.example.com" with default driver
Creating orderer.example.com ... done
Creating couchdb1            ... done
Creating couchdb0            ... done
Creating peer0.org2.example.com ... done
Creating peer0.org1.example.com ... done
Creating cli                    ... done
```

FIGURE 9. Result of deploying fabric test network.



```
peer0.org2.example.com|2023-06-03 20:37:57.372 UTC 0062 INFO [endorser] callChaincode -> finished chaincode: basic duration: 6ms
peer0.org1.example.com|2023-06-03 20:37:57.372 UTC 006b INFO [endorser] callChaincode -> finished chaincode: basic duration: 5ms
peer0.org1.example.com|2023-06-03 20:37:57.374 UTC 006c INFO [comm.grpc.server] 1 -> unary call completed grpc.service=protos.End
.1:33968 grpc.code=OK grpc.call_duration=8.448282ms
peer0.org2.example.com|2023-06-03 20:37:57.375 UTC 0063 INFO [comm.grpc.server] 1 -> unary call completed grpc.service=protos.End
.1:57262 grpc.code=OK grpc.call_duration=9.665248ms
   orderer.example.com|2023-06-03 20:37:57.388 UTC 0042 WARN [orderer.common.broadcast] Handle -> Error reading from 172.18.0.1:5
   orderer.example.com|2023-06-03 20:37:57.389 UTC 0043 INFO [comm.grpc.server] 1 -> streaming call completed grpc.service=ordere
2.18.0.1:50878 error="rpc error: code = Canceled desc = context canceled" grpc.code=Canceled grpc.call_duration=53.027815ms
   orderer.example.com|2023-06-03 20:37:59.381 UTC 0044 INFO [orderer.consensus.etcdraft] propose -> Created block [6], there are
   orderer.example.com|2023-06-03 20:37:59.429 UTC 0045 INFO [orderer.consensus.etcdraft] writeBlock -> Writing block [6] (Raft i
peer0.org1.example.com|2023-06-03 20:37:59.473 UTC 006d INFO [gossip.privdata] StoreBlock -> Received block [6] from buffer chann
peer0.org2.example.com|2023-06-03 20:37:59.473 UTC 0064 INFO [gossip.privdata] StoreBlock -> Received block [6] from buffer chann
peer0.org1.example.com|2023-06-03 20:37:59.477 UTC 006e INFO [committer.txvalidator] Validate -> [mychannel] Validated block [6]
peer0.org2.example.com|2023-06-03 20:37:59.477 UTC 0065 INFO [committer.txvalidator] Validate -> [mychannel] Validated block [6]
peer0.org1.example.com|2023-06-03 20:37:59.546 UTC 006f INFO [kvledger] commit -> [mychannel] Committed block [6] with 1 transact
t=40ms state_commit=6ms) commitHash=[c8a5399a8dda2031073221e38ba60d2990175f3c4cb390426474bdd9f0e9f163]
peer0.org2.example.com|2023-06-03 20:37:59.550 UTC 0066 INFO [kvledger] commit -> [mychannel] Committed block [6] with 1 transact
t=40ms state_commit=7ms) commitHash=[c8a5399a8dda2031073221e38ba60d2990175f3c4cb390426474bdd9f0e9f163]
```

FIGURE 10. Result of InvokePolicy in chaincode.

consensus and secure communication in Hyperledger Fabric networks, ensuring trust and dependability in blockchain-based applications. We begin by establishing the channel. Each channel joined a separate ledger and blockchain. In our scenario, there is only one channel created to check the performance of the system for transactions. Each channel joined a separate ledger and blockchain.

$$(\text{blockchain, ledger}) \rightarrow \text{Channel}$$

F. SYSTEM ARCHITECTURE

There are three sections to the experiment. The first part describes the architecture of a blockchain-based access control network and initiates setup and configuration procedures. The installation of the chaincode is covered in the second section. The third section explains how to use a smart contract to build the RBAC-based IoT resource access control mechanism. To record the action in the ledger, the value of the policy is kept in CouchDB.

$$\text{Contract(RBAC)} \rightarrow \text{Ledger, CouchDB}$$

1) INITIALIZATION OF TEST NETWORK

The process of using the Hyperledger cryptographic tool to create root certificates and private key pairs for nodes like peers and orderers. Within the network, these cryptographic tools guarantee safe communication and identity verification. The study [49] also discusses the architecture of the hyperledger fabric tools for their experiment. The steps involved in the initialization of the network are as follows.

- The created certificates and keys are placed in a designated directory that will be mounted by the Certificate Authority's (CA) docker image. The Other nodes can use their signatures to authenticate their identities to the CA while the CA container is functioning.
- A genesis block is created using the configtxgen tool. This block collects transactions that describe how nodes and channels are configured. The genesis block is added to the blockchain when the Fabric network first starts up, protecting each node's tamper-proof identifying information.
- Based on the initialization configuration specified in the docker-compose file, the remaining nodes are then started. Once every container has completed its operation, the peer nodes can be connected to a channel to enable additional network communications.

2) INSTALLATION AND UPGRADING OF CHAINCODE

We begin installing chaincode. The hyperledger commands are used to install the chaincode. The steps are listed below:

- Mount a directory on the client node and copy the chaincode source code into it.
- Issue the command to a peer node via a channel to package the chaincode.
- Distribute and instantiate chaincode that has been compiled to additional peer nodes. Chaincode is endorsed with a copy that is saved to a different container.

The Hyperledger Fabric SDK and client are used by the administrator to install Chaincode. The installation of all chaincode is on peer nodes. The chaincode gets initialized after installation. Chaincode is initialized using the invoke function. Every chaincode that is instantiated contains an endorsement saved in the container.

$$(\text{Install Chaincode}) \rightarrow \text{Peer}$$
$$(\text{Invoke init}) \rightarrow \text{Peer}$$

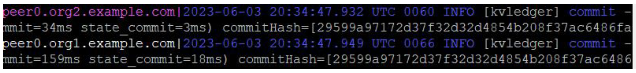
Installing software is equivalent to upgrading it. The moment a transaction is generated will the other peer nodes receive an upgrade, and only the node that initially installed the chaincode will receive an immediate upgrade to the new version.

The blockchain network integrity is improved by the Invoke Policy, which guarantees consensus and trust in the validity of transactions. The result of InvokePolicy in Chaincode is shown in Figure 10.

3) RBAC IMPLEMENTATION

RBAC implementation is a technique for controlling system access permissions. It entails giving users roles and setting the permissions attached to those roles. Users are given access depending on the responsibilities they have been given, simplifying access management and making sure they only have the privileges required for their respective roles. The study [50], shows how access models are implemented for IoT devices to secure the data and resources of the device. The steps are as follows that are involved in RBAC implementation.

- The secret key pairs that the CA node generates for a client are kept in the user's wallet.
- An administrator runs a client to connect to the peer node to submit or evaluate (related write and read operations) in a transaction.



```
peer0.org2.example.com|2023-06-03 20:34:47.932 UTC 0060 INFO [kvledger] commit -
mmit=34ms state_commit=3ms) commitHash=[29599a97172d37f32d32d4854b208f37ac6486fa
peer0.org1.example.com|2023-06-03 20:34:47.949 UTC 0066 INFO [kvledger] commit -
mmit=159ms state_commit=18ms) commitHash=[29599a97172d37f32d32d4854b208f37ac6486
```

FIGURE 11. Result of ApprovalPolicy of chaincode on a channel.

- The agreement is reached between peer nodes operating under the direction of the orderer node, the peer node queries or changes the CouchDB.

The RBAC smart contract is developed to enforce access control rules, validate transactions and resolve conflicts using the decentralized consensus mechanism provided by Hyperledger Fabric. This strategy reduces the difficulties brought on by competing roles and offers an effective, distributed and secure access control solution for Internet of Things environments. The hyperledger system's foundational processes are the set-up of the chaincode and the startup of the blockchain network.

To check the approval of the policy chaincode on the channel we invoke the policy command. The result of the approval policy of chaincode on the channel is shown in Figure 11. Both organizations seller and buyer are approved for the transaction. The state-commit time for Organization1 is 18 ms and for Organization2 is 3 ms.

The private data transmission smart contract encourages ownership by distinctive identities connected to the network. In our scenario, the person who owns the asset is a seller and member of Organization 1, whereas the buyer(B) is a member of Organization 2. The buyer identity registration process for one organization to join a network is demonstrated by Algorithm 1. The buyer ID, Name and Role are established throughout the registration procedure. I represent the set of identities of buyer I_B, seller I_Se and assets ownerI_O.

$$B = \text{Bid, Bname}$$

$$\text{Se} = \text{Seid, Sename}$$

$$I = \{I_B, I_{Se}, I_O\}$$

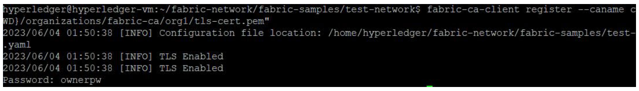
$$N = \{N_B, N_{Se}\}$$

$$R = \{R_B, R_{Se}\}$$

Algorithm 1 Buyer Registration

```
Input: $I_B$, $N_B$, $R_B$
Output: Register the Buyer as for contributor.
1. $I_B \leftarrow$ B;
2. $N_B \leftarrow$ name;
3. $R_B \leftarrow$ role;
4. $I_B \leftarrow$ Request for enrollment to the system;
5. If (Buyer information match) then
6. Return Registered;
7. else
8. Return "Error";
9. end if.
```

The user is added to the specified organization after the assets of the user have been confirmed. Access is allowed by an agreement that makes use of a policy that calls for a signature.



```
hyperledger@hyperledger-vm:~/fabric-network/fabric-samples/test-network$ fabric-ca-client register --caname c
WD}/organizations/fabric-ca/org1/tls-cert.pem"
2023/06/04 01:50:38 [INFO] Configuration file location: /home/hyperledger/fabric-network/fabric-samples/test-
.yaml
2023/06/04 01:50:38 [INFO] TLS Enabled
2023/06/04 01:50:38 [INFO] TLS Enabled
Password: ownerpw
```

FIGURE 12. Result of identity registration.

The seller identity registration process collects the same data as input for the seller's organization to join a network. Algorithm 2 shows how seller ID, Name and Role are taken as input to be established throughout the registration procedure. The user is added to the specified organization after the assets of the user have been confirmed.

Algorithm 2 Seller Registration

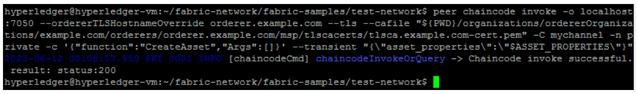
```
Input: $I_Se$, $N_Se$, $N_Se$
Output: Register the Seller as for contributor
1. $I_Se \leftarrow$ se;
2. $N_Se \leftarrow$ name;
3. $R_Se \leftarrow$ role;
4. $I_Se \leftarrow$ Request for the enrollment to the system;
5. If (Seller information match) then
6. Return Registered;
7. else
8. Return "Error";
9. end if.
```

After committing the chaincode a car contract for the buyer and seller transaction is considered in our example. The registration process of users is taken. To check whether the user's IDs are registered to their designated organizations the Register identity is invoked. Org1 is for the seller users and Org2 is for the buyer users. The result of identity registration is shown in Figure 12. The results show that the user is successfully registered.

Checking access ensures that the user's request for access is by the RBAC policy. Similar to Policy Contract, access control signs the request data using the user's private key, and then verifies the signature and the user's identity using the user's public key. CheckAccess is the essential process to manage access control, as depicted in Algorithm 3.

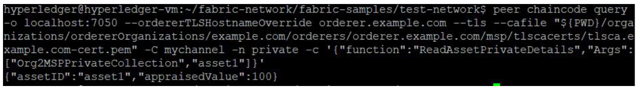
Algorithm 3 RBAC Contract Check Users Access

```
Input: $I_B$, Asset $I_O$
Output: Access the Data
1. $I_B \leftarrow$ Get ID from the participant registry;
2. Asset $I_O \leftarrow$ Asset ID from the participant archive;
3. Get Data Access $\leftarrow$ Buyer request to access data;
4. Start time $\leftarrow$ Get the accurate time;
5. If (Buyer request = true) then
6. Result $\leftarrow$ check the Time of Access and Grant access.;
7. else
8. Return "Access Denied";
9. end if.
```



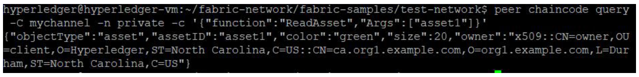
```
hyperledger@hyperledger-vm:~/fabric-network/fabric-samples/test-network$ peer chaincode invoke -o localhost
:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile "${PWD}/organizations/ordererOrganiza
tions/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem" -C mychannel -n p
rivate -c '{"function":"CreateAsset","Args":[]}' --transient "{\"asset_properties\":\"$ASSET_PROPERTIES\"}"
2023-06-12 00:08:57.950 PKT 0001 INFO [chaincodeCmd] chaincodeInvokeOrQuery -> Chaincode invoke successful.
 result: status:200
hyperledger@hyperledger-vm:~/fabric-network/fabric-samples/test-network$
```

FIGURE 13. Result of invoking AddAssets.



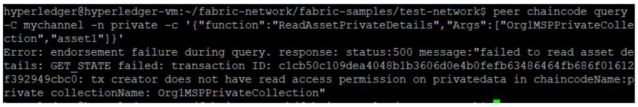
```
hyperledger@hyperledger-vm:~/fabric-network/fabric-samples/test-network$ peer chaincode query
-o localhost:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile "${PWD}/orga
nizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.e
xample.com-cert.pem" -C mychannel -n private -c '{"function":"ReadAssetPrivateDetails","Args":
["Org2MSPPrivateCollection","asset1"]}'
{"assetID":"asset1","appraisedValue":100}
```

FIGURE 14. Result of invoking GetAssets.



```
hyperledger@hyperledger-vm:~/fabric-network/fabric-samples/test-network$ peer chaincode query
 -C mychannel -n private -c '{"function":"ReadAsset","Args":["asset1"]}'
{"objectType":"asset","assetID":"asset1","color":"green","size":20,"owner":"x509::CN=owner,OU
=client,O=Hyperledger,ST=North Carolina,C=US::CN=ca.org1.example.com,O=org1.example.com,L=Dur
ham,ST=North Carolina,C=US"}
```

FIGURE 15. Result of invoking QueryPolicy as authorized user.



```
hyperledger@hyperledger-vm:~/fabric-network/fabric-samples/test-network$ peer chaincode query
-C mychannel -n private -c '{"function":"ReadAssetPrivateDetails","Args":["Org1MSPPrivateColle
ction","asset1"]}'
Error: endorsement failure during query. response: status:500 message:"failed to read asset de
tails: GET_STATE failed: transaction ID: c1cb50c109dea4048b1b3606d0e4b0fefb63486464fb686f01612
f392949cbc0: tx creator does not have read access permission on privatedata in chaincodeName:p
rivate collectionName: Org1MSPPrivateCollection"
```

FIGURE 16. Result of invoking QueryPolicy as unauthorized user.

The proposed technique attempts to simplify data access based on the IDs of the asset owner and the buyer. The algorithm first extracts the buyer ID from the participant registration and the ID of the asset owner from the registration data. The algorithm then gets a request to access the data from the buyer. The algorithm moves on to the following stage if the buyer's request is verified to be true. Here, it determines if the start time determined earlier and the access time constraint are in sync. Access to the data is permitted if the prerequisite is satisfied. If the requirement is not met, on the other hand, the algorithm returns the message "Access Denied." The algorithm then completes its operation. Access Decision (AD) binary matrix indicating whether user u is granted permission pi at ith permissions. kth users and jth roles at max users roles against permissions are presented below:

$$AD = \begin{cases} 1, & \text{if user } u_k \text{ is granted permission } p_i \\ 0, & \text{otherwise} \end{cases}$$

This is based on the user having a role with the required permission:

$$AD = \max_{j} \left(UR_{ku} \cdot RP_{ij} \right)$$

After the identity registration the assets are created on which users' requests are granted with checking their roles and denied for unauthorized users. First, it invokes the AddAssets() function to store assets in the ledger as shown in Figure 13. As the chaincode approach is being used for an asset to be transferred, both the asset owner and the buyer must accept the same appraised value. The Org2 membership service provider details collection on the Org2 peer will have the agreed-upon value. In our case, we considered the buyer and seller organizations which made transactions successfully without any leakage of their private data. Both organizations used the peer, orderer and endorser nodes that were involved in a car contract transaction based on smart contract rules. A very well-known Fabcar example of a tool is used in this paper. To verify the smart contract policy and check if this smart contract results as per our needs. To prevent policy contract enforcement the corresponding RBAC policy is used according to user roles. Secondly, it gets the assets by invoking GetAssets() function as shown in Figure 14.

The transaction takes place between two organizations within the channel so when the member of the organization seller who is the owner of data or as we say owner of a car generates assets which contain the information of the car. The assets now are on the blockchain network. IoT devices can monitor the whole scenario in conjunction with smart contracts and are involved in the shipment. If any member of the buyer organization wants to buy a car. It generates a bid on car prices or a request to get more details. That's how a complete transaction process is held. And data is secured in a network. The private data of any organization member is secured from unauthorized users.

V. RESULTS AND DISCUSSION

We examine the system's functionality in terms of response time, data accessibility and data privacy. The RBAC smart contract validates the policy that all peers of Org1 and Org2 are permitted by our collection definition to store the asset ID, colour, size, and owner private data in their side databases, but only peers in Org1 are permitted to store Org1's appraisal of their appraised Value private data. We perform both sets of private data queries in Org1 as an authorized peer. The result of invoking chaincode leads to enforcing the RBAC smart contract policy using signature type for the secured transaction. The result of invoking QueryPolicy as an authorized user shows the details of the asset for the user with role reader as shown in Figure 15.

The set of rules that are defined in a smart contract forces each user in an organization to follow the rule strictly. By doing this user is restricted from doing any operation. The signature policy is used to sign the agreement and provides security against each transaction. Both organizations sign the contract to protect the transaction. Once a transaction is recorded in a blockchain ledger, it cannot be changed or altered. The results show that the chaincode is successfully committed to the channel and enforces policies in each organization. The result of invoking QueryPolicy as an unauthorized user shows an error message that the user has no permission to access data as shown in Figure 16. The needed level of endorsement for a query transaction is specified by the QueryPolicy. It specifies the circumstances under which a query result is regarded as legitimate. This policy can be altered to meet the requirements of the application and is normally defined during the chaincode instantiating.

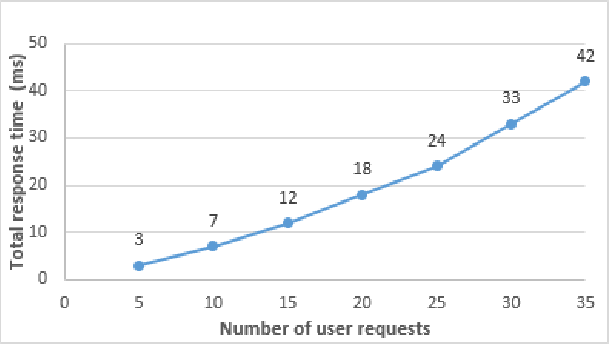


FIGURE 17. Total response time as the number of user requests increases.

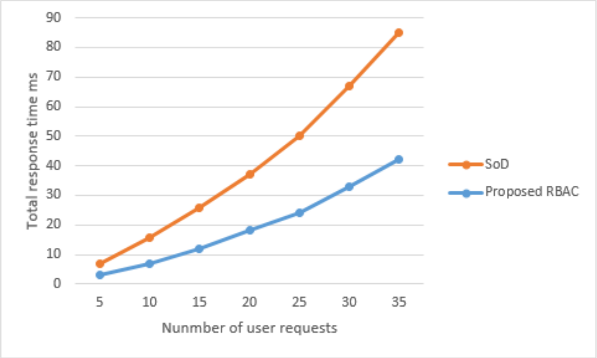


FIGURE 18. Comparison of response time between SoD and our proposed model.

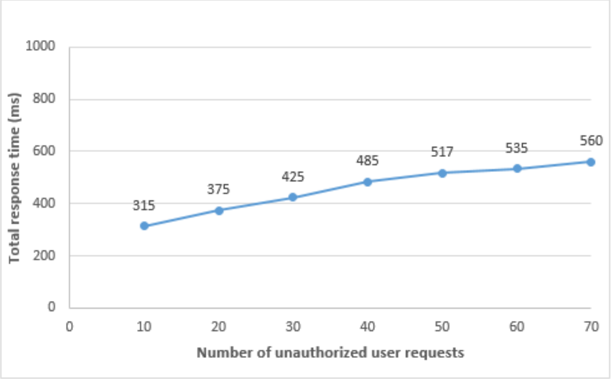


FIGURE 19. Total response time for Unauthorized User Requests.

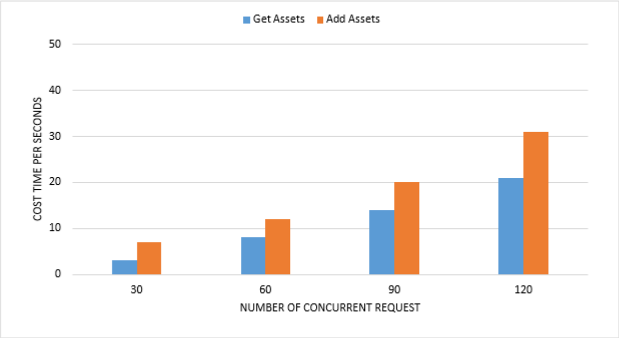


FIGURE 20. Cost time of RBAC contract at several numbers of concurrent requests.

The two sets of comparable tests are taken by replicating concurrent access to the system with multi-threaded users to examine the effectiveness of blockchain-based access control solutions for the Internet of Things. In evaluating the performance of the first set of tests, we took into account two model parameters. The first parameter is taken into account as the overall system response time to requests from different user counts. The total response time changes with the number of user requests shown in Figure 17.

The average response time for each user is 1.4 ms which shows better performance for the small blockchain IoT devices environment. The second one is the total response time for the enforcement of the RBAC policy constraint which shows the relationship between unauthorized user requests and total response time. The comparison of response time with base study SoD is also computed to check the performance of our strategy we take the average time of their studies by setting our user number of requests and the results of our model comparatively less than their studies as shown in Figure 18.

The proposed framework for the implementation of the conflicting role constraint has a stable performance for different numbers of unauthorized users, with an average response time of 32.12 milliseconds for each unauthorized user request. The analysis of response time in terms of the number of unauthorized user requests is shown in Figure 19.

In the second experiment, we measure the cost time per second while processing a variety of concurrent requests. There are 30, 60, 90, and 120 requests running simultaneously. The write operations (such as ''add'' and ''update'') take more time than the read operations (such as ''get'' and ''query''). The result of the contract's cost time under several concurrent requests is shown in Figure 20. The cost of creating and adding assets is higher than the cost of getting assets. When we are talking about assets. In Hyperledger Fabric, assets are defined and maintained via chaincode, stored on a distributed ledger, controlled by smart contracts and go through a thorough lifecycle management process. While consensus algorithms check transactions for a consistent view across the network. It ensures that only authorized participants can interact with assets and access control techniques make sure that transparency and data integrity are promoted.

The comparison of parameters between the fabric-iot study and the proposed RBAC model is shown in Figure 21. The user who wants to read an asset only reads if it has permission. After a user request to read assets the proposed RBAC model first check the request and then grant the permission to the user otherwise the permission is denied and the error generated fails to read data. The comparative analysis of studies based on numerous factors is shown in TABLE 7.

TABLE 7. Comparative analysis based on model approach, Avg.Concurrent requests and Avg. Response time (Seconds).

Study	Model Approch	Avg. Concurrent Requests	Avg. Response Time (Seconds)
Fabric-iot [20]	ABAC	500	55
SoD constraint [30]	RBAC with SoD constraint	120	22
RBAC SC(Proposed model)	RBAC with Role constraint	35	12

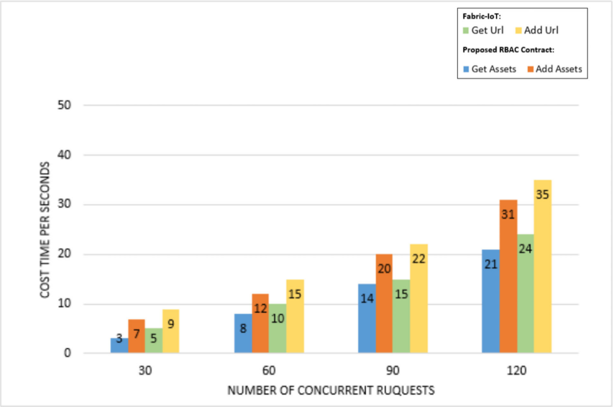


FIGURE 21. Comparison of Fabric-IoT study and our proposed RBAC model.

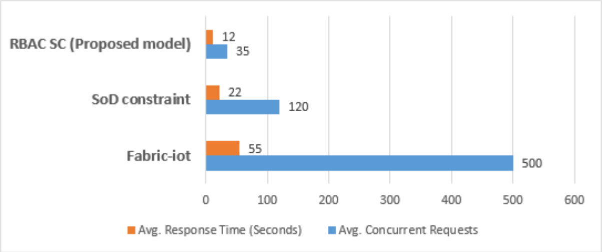


FIGURE 22. Comparison of existing studies and our RBAC proposed model.

The study fabric-iot stored data on the drive or clouds and generated their URLs to access this data in hyperledger fabric. The term used in their model is get URL and add URL while in our proposed study we used assets which are represented as get Asset and add Asset. The assets in Hyperledger Fabric are represented as a collection of key-value pairs, with state changes being recorded as transactions on a Channel ledger. Binary or JSON representations of assets are both possible.

The data provided in a table represents a study model approach that compares different access control mechanisms based on average concurrent requests and average response time in seconds. The conflicting roles issue is addressed and the response time rate is reduced as shown in the results by using the proposed RBAC approach in a hyperledger fabric blockchain environment in our scenario. In the existing studies as mentioned in the table implemented Attribute-Based Access Control (ABAC) with an average of 500 concurrent requests and an average response time of 55 seconds.

The SoD constraint study employed role-based access control handling 120 concurrent requests with an average response time of 22 seconds. RBAC Smart Contract (RBAC SC) utilized with Role constraint, achieving an average of 35 concurrent requests with a response time of 12 seconds.

In Hyperledger Fabric, the cost time in seconds against concurrent user requests can vary based on a multitude of factors, such as the hardware resources available and the complexity of the transactions being completed. The system may encounter longer latency and response times as the concurrent user requests increase in number. This is because the network requires resource allocation and synchronization. The performance and scalability of the Hyperledger Fabric network also have a significant impact on the cost time. To avoid the above issues we keep the architecture simple and the cost time taken in transaction use case to show how response time is improved.

The above results show how our proposed methodology performs better for the cost factor against user concurrent requests in a blockchain-based access control environment. Our proposed scheme provides a more robust approach to handling the problem of role conflict with the minimum response time as well. The comparison of existing studies and our RBAC proposed model shows an average response time and average concurrent requests of users in Figure 22. The fabric-iot study makes use of multiple respondents. It may take longer to respond since the user's device URL for resource access address first verifies the URL before checking the user's attributes to give access. They divided the users into two classes and then set a concurrent users request for the device contract of 50, 100, 500, and 1000.

A device, policy and three different forms of smart contracts were also utilized. The user only needs to register and get authorized with their roles and permissions in our suggested technique. We map the roles, and permissions of users into a single smart contract called RBAC smart contract to improve system efficiency and user response time.

The results show that our proposed RBAC smart contract model takes minimum response time at an average number of user requests instead of existing studies as shown in Figure 22. The number of users will be increased in future to check the behaviour of our system in terms of reliability, security and accessibility.

A. DISCUSSION AND IMPLICATION

We use the Hyperledger Fabric framework to test the performance of the network in terms of cost time, response time and accessibility. We considered a small car contract example in our scenario based on roles and used a limited number of users to keep the environment simple. The main goal of our study is to address conflicting roles and provide higher security in an IoT environment. In previous studies, traditional access control methods were employed, and testing revealed that such systems needed a lot of time to grant access to data collection, did not address the issue of competing roles, and had security issues because they relied on a single trusted authority. The Fabric-IoT ABAC model was introduced in a study [20]. According to the study, the amount of time it took to respond to user requests increased when specific attributes were given to each user in the access policy. Furthermore, there was a higher risk of decreased accessibility and compromised security when several attributes were taken into account. On the other hand in the study [30], the Blockchain-based RBAC model with SoD constraints in a cloud environment has drawbacks such as scalability, latency, energy consumption and lack of privacy. As the cloud environment is involved it also affects the user's response time.

In the context of IoT devices, role-based access control (RBAC) provides an efficient method for addressing competing roles. According to the findings, users only have access to the features and information required for their specific tasks, reducing conflicts and boosting security. RBAC offers a scalable and effective method for controlling access permissions, supporting the efficient use of IoT devices in a variety of situations. Our proposed blockchain-based access control system for the Internet of Things provides lower response time and higher security in terms of accessibility based on RBAC smart contracts. Cryptographic techniques are used to ensure confidentiality and transparency.

VI. CONCLUSION AND FUTURE WORK

This paper proposed an innovative approach that combines blockchain technology with the RBAC strategy to deal with the limitations of traditional centralized access control methods in IoT. By leveraging blockchain's decentralized, tamper-proof, and traceable nature, our aim is to fulfil the Internet of Things access control needs more effectively. The proposed approach consists of several key components. We develop a data privacy model based on actual production data from IoT devices. The RBAC paradigm is used to construct a blockchain-based access control system management that ensures secure access to device resources and avoids role-based conflicts through smart contracts. Furthermore, a distributed architecture is employed in the development of a free and open-source access control system known as Hyperledger Fabric. This approach provides the physical network with delicate and dynamic access control management. Additionally, we provide comprehensive guidance on constructing a blockchain network, encompassing chaincode installation and smart contract invocation. We present compelling results that substantiate the effectiveness of our approach through experimentation.

The access control scheme we propose verifies a user's ownership of a role through a policy-generated signature type and handles access requests using our model. Subsequently, we conduct a security analysis of the proposed method and an assessment of its effectiveness. To assess the effectiveness of our approach, we simulate a Fabcar example for a car transaction scenario in the Hyperledger Fabric test network. By adjusting the total number of system users and unauthorized requests to access, we measure the performance of the model. Notably, unauthorized user requests exhibit an average response time of 32.12 milliseconds, demonstrating the superiority of our approach over traditional methods. The experiment conducted in this paper utilizes the Fabcar example on a Hyperledger Fabric test network. This comprehensive implementation of the RBAC environment in Hyperledger Fabric allows for a thorough evaluation of the system's distributed performance and response time.

Future studies should consider the following factors:

1) Integrate these techniques with decentralized applications (dApps) to test the system's reliability and performance under real-time conditions.
2) Future research concentrates on increasing network scalability to enable more IoT applications integration, increasing the system's overall efficacy and usefulness.
3) Improve the network's ability to handle a larger volume of IoT-related functionalities to enhance the system's dependability and performance in real-time scenarios. By setting up the whole environment with RBAC dependencies in Hyperledger Fabric.

REFERENCES

[1] P. P. Ray, "A survey on Internet of Things architectures," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 30, no. 3, pp. 291–319, 2018.

[2] R. Xu, Y. Chen, E. Blasch, and G. Chen, "Exploration of blockchain-enabled decentralized capability-based access control strategy for space situation awareness," *Opt. Eng.*, vol. 58, no. 4, 2019, Art. no. 041609.

[3] F. Cai, N. Zhu, J. He, P. Mu, W. Li, and Y. Yu, "Survey of access control models and technologies for cloud computing," *Cluster Comput.*, vol. 22, pp. 6111–6122, Jan. 2019.

[4] P. K. Sharma and J. H. Park, "Blockchain based hybrid network architecture for the smart city," *Future Gener. Comput. Syst.*, vol. 86, pp. 650–655, Sep. 2018.

[5] S. Ding, J. Cao, C. Li, K. Fan, and H. Li, "A novel attribute-based access control scheme using blockchain for IoT," *IEEE Access*, vol. 7, pp. 38431–38441, 2019.

[6] Md. A. Islam and S. Madria, "A permissioned blockchain based access control system for IoT," in *Proc. IEEE Int. Conf. Blockchain (Blockchain)*, Jul. 2019, pp. 469–476.

[7] S. Saha, A. K. Sutrala, A. K. Das, N. Kumar, and J. J. P. C. Rodrigues, "On the design of blockchain-based access control protocol for IoT-enabled healthcare applications," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–6.

[8] A. Iftekhar, X. Cui, Q. Tao, and C. Zheng, "Hyperledger fabric access control system for Internet of Things layer in blockchain-based applications," *Entropy*, vol. 23, no. 8, p. 1054, Aug. 2021.

[9] I. A. Magomedov, A. M. Bagov, and A. L. Zolkin, "Internet of Things: Future business," in *Proc. Eur. Proc. Social Behavioural Sci.*, Oct. 2020, pp. 553–558.

[10] J. Qiu, Z. Tian, C. Du, Q. Zuo, S. Su, and B. Fang, "A survey on access control in the age of Internet of Things," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 4682–4696, Jun. 2020.

[11] J. I. Zong Chen and K.-L. Lai, "Internet of Things (IoT) authentication and access control by hybrid deep learning method—A study," *J. Soft Comput. Paradigm*, vol. 2, no. 4, pp. 236–245, Jan. 2021.

[12] T. Zaidi, S. Aziz, M. Usman, A. Azam, A. A. Cheema, and S. Ajmal, "Edge computing and computational task offloading analysis—A review study," in *Proc. Int. Conf. Emerg. Trends Electr., Control, Telecommun. Eng. (ETECTE)*, Dec. 2022, pp. 1–7.

[13] G. D. Putra, V. Dedeoglu, S. S. Kanhere, and R. Jurdak, "Trust management in decentralized IoT access control system," in *Proc. IEEE Int. Conf. Blockchain Cryptocurrency (ICBC)*, May 2020, pp. 1–9.

[14] R. E. Sibai, N. Gemayel, J. B. Abdo, and J. Demerjian, "A survey on access control mechanisms for cloud computing," *Trans. Emerg. Telecommun. Technol.*, vol. 31, no. 2, pp. 1–11, Feb. 2020.

[15] M. Sookhak, M. R. Jabbarpour, N. S. Safa, and F. R. Yu, "Blockchain and smart contract for access control in healthcare: A survey, issues and challenges, and open issues," *J. Netw. Comput. Appl.*, vol. 178, Mar. 2021, Art. no. 102950.

[16] I. Riabi, H. K. B. Ayed, and L. A. Saidane, "A survey on blockchain based access control for Internet of Things," in *Proc. 15th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jun. 2019, pp. 502–507.

[17] S. Algarni, F. Eassa, K. Almarhabi, A. Almalaise, E. Albassam, K. Alsubhi, and M. Yamin, "Blockchain-based secured access control in an IoT system," *Appl. Sci.*, vol. 11, no. 4, p. 1772, Feb. 2021.

[18] S. Sun, R. Du, S. Chen, and W. Li, "Blockchain-based IoT access control system: Towards security, lightweight, and cross-domain," *IEEE Access*, vol. 9, pp. 36868–36878, 2021.

[19] Y. Nakamura, Y. Zhang, M. Sasabe, and S. Kasahara, "Capability-based access control for the Internet of Things: An Ethereum blockchain-based scheme," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–6.

[20] H. Liu, D. Han, and D. Li, "Fabric-IoT: A blockchain-based access control system in IoT," *IEEE Access*, vol. 8, pp. 18207–18218, 2020.

[21] G. D. Putra, V. Dedeoglu, S. S. Kanhere, R. Jurdak, and A. Ignjatovic, "Trust-based blockchain authorization for IoT," *IEEE Trans. Netw. Service Manage.*, vol. 18, no. 2, pp. 1646–1658, Jun. 2021.

[22] D. Han, Y. Zhu, D. Li, W. Liang, A. Souri, and K.-C. Li, "A blockchain-based auditable access control system for private data in service-centric IoT environments," *IEEE Trans. Ind. Informat.*, vol. 18, no. 5, pp. 3530–3540, May 2022.

[23] P. Patil, M. Sangeetha, and V. Bhaskar, "Blockchain for IoT access control, security and privacy: A review," *Wireless Pers. Commun.*, vol. 117, no. 3, pp. 1815–1834, Apr. 2021.

[24] M. Shakarami, J. Benson, and R. Sandhu, "Blockchain-based administration of access in smart home IoT," in *Proc. ACM Workshop Secure Trustworthy Cyber-Phys. Syst.*, Apr. 2022, pp. 57–66.

[25] S. Namane and I. Ben Dhaou, "Blockchain-based access control techniques for IoT applications," *Electronics*, vol. 11, no. 14, p. 2225, Jul. 2022.

[26] P. Kamboj, S. Khare, and S. Pal, "User authentication using blockchain based smart contract in role-based access control," *Peer-Peer Netw. Appl.*, vol. 14, no. 5, pp. 2961–2976, Sep. 2021.

[27] D. Liu, A. Dong, B. Yan, and J. Yu, "DF-RBAC: Dynamic and fine-grained role-based access control scheme with smart contract," *Proc. Comput. Sci.*, vol. 187, pp. 359–364, Jan. 2021.

[28] X. Xu, X. Wang, Z. Li, H. Yu, G. Sun, S. Maharjan, and Y. Zhang, "Mitigating conflicting transactions in hyperledger fabric-permissioned blockchain for delay-sensitive IoT applications," *IEEE Internet Things J.*, vol. 8, no. 13, pp. 10596–10607, Mar. 2021.

[29] M. H. Ronaghi, "Contextualizing the impact of blockchain technology on the performance of new firms: The role of corporate governance as an intermediate outcome," *J. High Technol. Manage. Res.*, vol. 33, no. 2, Nov. 2022, Art. no. 100438.

[30] O.-C. Ri, Y.-J. Kim, and Y.-J. Jong, "Blockchain-based RBAC model with separation of duties constraint in cloud environment," 2022, *arXiv:2203.00351*.

[31] E. A. Shammar, A. T. Zahary, and A. A. Al-Shargabi, "An attribute-based access control model for Internet of Things using hyperledger fabric blockchain," *Wireless Commun. Mobile Comput.*, vol. 2022, pp. 1–25, Jul. 2022.

[32] S. N. A. Sherazi, E. Zahoor, S. Akhtar, and O. Perrin, "A blockchain based approach for the authorization policies delegation in emergency situations," *Trans. Emerg. Telecommun. Technol.*, vol. 33, no. 5, p. e4461, May 2022.

[33] B. B. Sinha and R. Dhanalakshmi, "Recent advancements and challenges of Internet of Things in smart agriculture: A survey," *Future Gener. Comput. Syst.*, vol. 126, pp. 169–184, Jan. 2022.

[34] R. Huo, S. Zeng, Z. Wang, J. Shang, W. Chen, T. Huang, S. Wang, F. R. Yu, and Y. Liu, "A comprehensive survey on blockchain in industrial Internet of Things: Motivations, research progresses, and future challenges," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 1, pp. 88–122, 1st Quart., 2022.

[35] C. Esposito, M. Ficco, and B. B. Gupta, "Blockchain-based authentication and authorization for smart city applications," *Inf. Process. Manage.*, vol. 58, no. 2, Mar. 2021, Art. no. 102468.

[36] Y. Li, Z. Du, Y. Fu, and L. Liu, "Role-based access control model for inter-system cross-domain in multi-domain environment," *Appl. Sci.*, vol. 12, no. 24, p. 13036, Dec. 2022.

[37] E. Zahoor, A. Ikram, S. Akhtar, and O. Perrin, "A formal approach for the identification of authorization policy conflicts within multi-cloud environments," *J. Grid Comput.*, vol. 20, no. 2, pp. 1–22, Jun. 2022.

[38] S. Ameer, J. Benson, and R. Sandhu, "Hybrid approaches (ABAC and RBAC) toward secure access control in smart home IoT," *IEEE Trans. Dependable Secure Comput.*, vol. 25, no. 5, pp. 4032–4051, Sep. 2022.

[39] R. Kumar and R. Sharma, "Leveraging blockchain for ensuring trust in IoT: A survey," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 10, pp. 8599–8622, Nov. 2022.

[40] S. Ahamad, P. Gupta, P. B. Acharjee, K. P. Kiran, Z. Khan, and M. F. Hasan, "The role of block chain technology and Internet of Things (IoT) to protect financial transactions in crypto currency market," *Mater. Today, Proc.*, vol. 56, pp. 2070–2074, 2022.

[41] M. Arunmozhi, V. G. Venkatesh, S. Arisian, Y. Shi, and V. R. Sreedharan, "Application of blockchain and smart contracts in autonomous vehicle supply chains: An experimental design," *Transp. Res. E, Logistics Transp. Rev.*, vol. 165, Sep. 2022, Art. no. 102864.

[42] J. Li and M. Kassem, "Applications of distributed ledger technology (DLT) and blockchain-enabled smart contracts in construction," *Autom. Construct.*, vol. 132, Dec. 2021, Art. no. 103955.

[43] F. Schär, "Decentralized finance: On blockchain- and smart contract-based financial markets," *Review*, vol. 103, no. 2, pp. 1–10, 2021.

[44] T. Hewa, M. Ylianttila, and M. Liyanage, "Survey on blockchain based smart contracts: Applications, opportunities and challenges," *J. Netw. Comput. Appl.*, vol. 177, Mar. 2021, Art. no. 102857.

[45] M. U. Aftab, Z. Qin, N. W. Hundera, O. Ariyo, N. T. Son, and T. V. Dinh, "Permission-based separation of duty in dynamic role-based access control model," *Symmetry*, vol. 11, no. 5, p. 669, May 2019.

[46] M. U. Aftab, Y. Munir, A. Oluwasanmi, Z. Qin, M. H. Aziz, N. T. Son, and V. D. Tran, "A hybrid access control model with dynamic COI for secure localization of satellite and IoT-based vehicles," *IEEE Access*, vol. 8, pp. 24196–24208, 2020.

[47] M. U. Aftab, A. Oluwasanmi, A. Alharbi, O. Sohaib, X. Nie, Z. Qin, and S. T. Ngo, "Secure and dynamic access control for the Internet of Things (IoT) based traffic system," *PeerJ Comput. Sci.*, vol. 7, p. e471, May 2021.

[48] S. Parkinson and S. Khan, "A survey on empirical security analysis of access-control systems: A real-world perspective," *ACM Comput. Surv.*, vol. 55, no. 6, pp. 1–28, Jul. 2023.

[49] H. Honar Pajooh, M. Rashid, F. Alam, and S. Demidenko, "Hyperledger fabric blockchain for securing the edge Internet of Things," *Sensors*, vol. 21, no. 2, p. 359, Jan. 2021.

[50] G. Fragkos, J. Johnson, and E. E. Tsiropoulou, "Dynamic role-based access control policy for smart grid applications: An offline deep reinforcement learning approach," *IEEE Trans. Hum.-Mach. Syst.*, vol. 52, no. 4, pp. 761–773, Aug. 2022.

[51] V. C. Hu, "Blockchain for access control systems," NIST Comput. Secur. Resource Center, Gaithersburg, MD, USA, Tech. Rep. NIST IR 8403, 2022.

[52] A. Kul, "Blockchain based context aware access control structure implementation for security of Internet of Things system," Ph.D. dissertation, Izmir Inst. of Technol., Urla, Türkiye, 2022.

[53] M. Usman, M. S. Sarfraz, U. Habib, M. U. Aftab, and S. Javed, "Automatic hybrid access control in SCADA-enabled IIoT networks using machine learning," *Sensors*, vol. 23, no. 8, p. 3931, Apr. 2023.

TANZEEL ZAIDI received the B.Sc. degree from the University of the Punjab, Lahore, Pakistan, in 2018, the M.Sc. degree in computer science from the University of Agriculture Faisalabad, in 2020, and the master's degree in computer science from the Department of Computer Science, FAST National University of Computer and Emerging Sciences, Chiniot-Faisalabad Campus, in 2023. Currently, she is a Visiting Lecturer with the Department of Computer Science, Superior College Chiniot. She was a web developer of private institution. Her research interests include blockchain, access control (ABAC and RBAC), and the Internet of Things.

MUHAMMAD USMAN is currently pursuing the Ph.D. degree with the FAST National University of Computer and Emerging Sciences, Chiniot–Faisalabad Campus. He is also a Lecturer with the Department of Computer Science, FAST National University of Computer and Emerging Sciences, Pakistan. His research interests are predictive analytics and machine learning applications for cross domain applications.

MUHAMMAD UMAR AFTAB (Member, IEEE) received the B.S. degree from Government College University Faisalabad, in 2011, the master's degree in computer science from the National Textile University, Pakistan, in 2014, and the Ph.D. degree from the School of Information and Software Engineering, University of Electronic Science and Technology of China (UESTC), in 2020. Currently, he is an Assistant Professor with the Department of Computer Science, FAST National University of Computer and Emerging Sciences. His research interests include information/network security, authorization/access control (RBAC and ABAC), cryptography, and network technologies. He is serving as a reviewer and a guest editor for various renowned journals.

HANAN ALJUAID received the B.S. degree from KAU University. Furthermore, Her M.S. and Ph.D. degrees in computer science from UTM University. She is currently with the Computer Sciences Department, College of Computer and Information Sciences, Princess Nourah Bint Abdul Rahman University (PNU), Saudi Arabia. She has published numerous articles in pattern recognition, the IoT, and data science. Much of her work has been on improving the understanding, design, and performance of pattern recognition, mainly through the application of data mining and machine learning. She has given numerous invited talks and tutorials. Her research interests include computer vision and NLP.

YAZEED YASIN GHADI received the Ph.D. degree in electrical and computer engineering from Queensland University. His dissertation on developing novel hybrid plasmonic photonic on-chip biochemical sensors. He was a Postdoctoral Researcher with Queensland University. He is currently an Assistant Professor of software engineering with Al Ain University. He has published more than 80 peer-reviewed journals and conference papers and holds three pending patents. His current research interests include developing novel electro-acoustic-optic neural interfaces for large-scale high-resolution electrophysiology and distributed optogenetic stimulation. He was a recipient of several awards. He received the Sigma Xi Best Ph.D. Thesis Award for his Ph.D. degree.