sustainability

MDPI

Article

An Attribute-Based Access Control for IoT Using Blockchain and Smart Contracts

Syed Yawar Abbas Zaidi [1], Munam Ali Shah [1], Hasan Ali Khattak [2,*], Carsten Maple [3], Hafiz Tayyab Rauf [4], Ahmed M. El-Sherbeeny [5] and Mohammed A. El-Meligy [5]

[1] Department of Computer Science, COMSATS University Islamabad, Islamabad 44500, Pakistan; yawar.abbas3636@gmail.com (S.Y.A.Z.); mshah@comsats.edu.pk (M.A.S.)
[2] School of Electrical Engineering and Computer Science, National University of Sciences and Technology (NUST), Islamabad 44500, Pakistan
[3] Secure Cyber Systems Research Group (SCSRG), University of Warwick, Coventry CV4 7AL, UK; cm@warwick.ac.uk
[4] Department of Computer Science, Faculty of Engineering & Informatics, University of Bradford, Bradford BD7 1DP, UK; h.rauf4@bradford.ac.uk
[5] Industrial Engineering Department, College of Engineering, King Saud University, P.O. Box 800, Riyadh 11421, Saudi Arabia; aelsherbeeny@ksu.edu.sa (A.M.E.-S.); melmeligy@ksu.edu.sa (M.A.E.-M.)
* Correspondence: hasan.alikhattak@seecs.edu.pk

Abstract: With opportunities brought by the Internet of Things (IoT), it is quite a challenge to maintain concurrency and privacy when a huge number of resource-constrained distributed devices are involved. Blockchain have become popular for its benefits, including decentralization, persistence, immutability, auditability, and consensus. Great attention has been received by the IoT based on the construction of distributed file systems worldwide. A new generation of IoT-based distributed file systems has been proposed with the integration of Blockchain technology, such as the Swarm and Interplanetary File System. By using IoT, new technical challenges, such as Credibility, Harmonization, large-volume data, heterogeneity, and constrained resources are arising. To ensure data security in IoT, centralized access control technologies do not provide credibility. In this work, we propose an attribute-based access control model for the IoT. The access control lists are not required for each device by the system. It enhances access management in terms of effectiveness. Moreover, we use blockchain technology for recording the attribute, avoiding data tempering, and eliminating a single point of failure at edge computing devices. IoT devices control the user's environment as well as his or her private data collection; therefore, the exposure of the user's personal data to non-trusted private and public servers may result in privacy leakage. To automate the system, smart contracts are used for data accessing, whereas Proof of Authority is used for enhancing the system's performance and optimizing gas consumption. Through smart contracts, ciphertext can be stored on a blockchain by the data owner. Data can only be decrypted in a valid access period, whereas in blockchains, the trace function is achieved by the storage of invocation and the creation of smart contracts. Scalability issues can also be resolved by using the multichain blockchain. Eventually, it is concluded from the simulation results that the proposed system is efficient for IoT.

Keywords: IoT; multichain; smart contract; interplanetary file system; access control

check for updates

Citation: Zaidi, S.Y.A.; Shah, M.A.; Khattak, H.A.; Maple, C.; Rauf, H.T.; El-Sherbeeny, A.M.; El-Meligy, M.A. An Attribute-Based Access Control for IoT Using Blockchain and Smart Contracts. *Sustainability* **2021**, *13*, 10556. https://doi.org/10.3390/su131910556

Academic Editor: Fadi Al-Turjman

Received: 14 August 2021
Accepted: 16 September 2021
Published: 23 September 2021

1. Introduction

IoT has become the most promising technology in industry and academia. Some of the aims of IoT are enabling, sharing, and collecting data anonymously from home appliances, vehicles, and physical and intelligent devices. In 2017, more than 8.4 billion devices joined this worldwide network, which shows the increased limit of 31% from 2016 [1]. On the contrary, Gartner [2] forecasts that it will reach 25 billion by 2021, and by 2023, the buying and selling of IoT data will become an essential part of many IoT systems. With a large number of devices involved, storage-related challenges also arise, and along with that, data protection and large-scale efficient data storage are significant issues [3].

New challenges and security risks keep increasing due to the increasing amount of connected devices,as shown in Figure 1. Security devices are becoming vulnerable to privacy-threatening attacks launched by malicious users, and because of these attacks, it is difficult to completely control the widely distributed IoT devices. For controlling the data leakage from IoT devices, an authorized access mechanism is needed to protect sensitive and valuable information [4].

There is rapid technological advancement for user's data sharing between the enterprises. By using data sharing applications, user experiences are improving in terms of functionality. Approaches based on standard security techniques while sharing user data without using any trusted authority have been addressed by Sherstha et al. [5]. The questions regarding what type of data and when or whom has been discussed by Meadows et al. [6], in which the data sharing with increasing incentive is a matter of intense research. For personal data storing, certain privacy and security issues, such as data theft and breaches, are present. When using the centralized authority, the deletion of user data and not delivering user's data are major problems [7].

Various technologies for the collection of data and sharing user data have been deployed using cloud computing, Federated learning [8], and RFID (Radio Frequency Identification). In strong privacy legislation, e.g., GDPR, the data owner's consent needs to be asked. The consent of data sharing and its use needs to be renewed, which provides meaningful incentives [9].

To provide effective unauthorized control, one of the most important and useful technologies is an access control system. Discretionary access control (DAC), which is known as traditional access control, and identity-based access control (IBAC) both fail to provide an appropriate result for the implementation of access control in IoT systems since the access control list of each unknown identity in the IoT system is almost impossible to make. Mandatory access control (MAC) is another technique that suffers from a single point failure due to the central administrator's imposition [10].

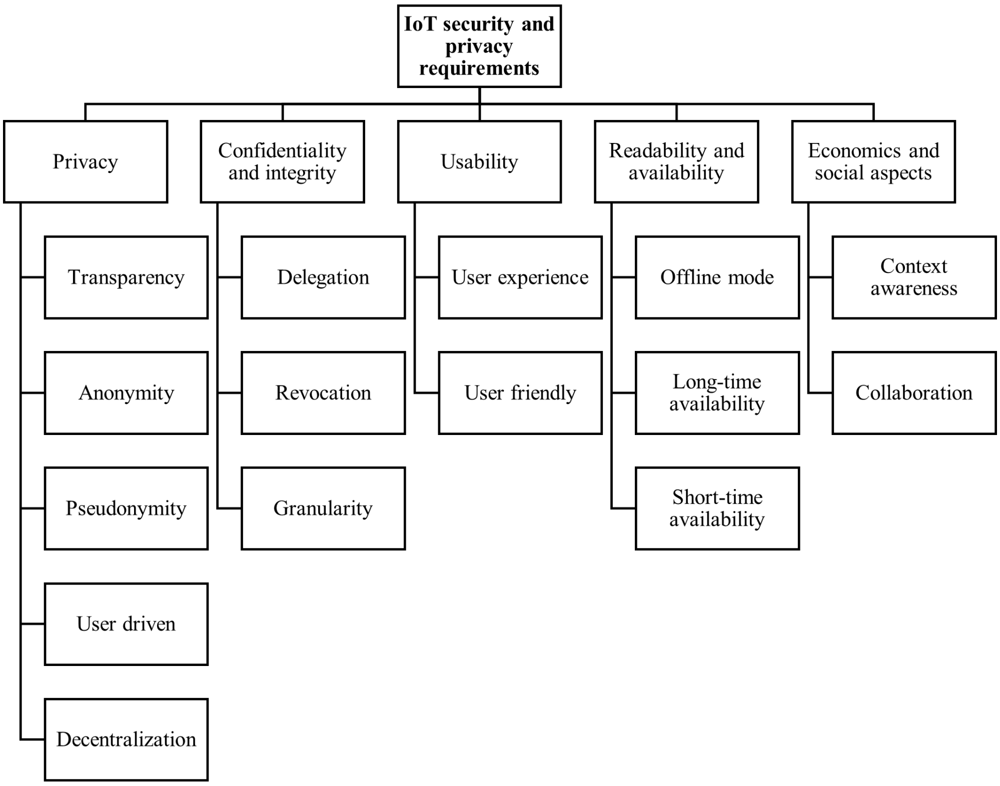


Figure 1. IoT security and privacy requirements.

1.1. Attribute-Based Access Control (ABAC)

A new type of dynamic, fine-grained, and flexible access control has been provided by attribute-based access control (ABAC), in which the attribute authorities issue the identities or roles to a set of attributes; therefore, making separate access control lists for every entity present in the system is not required. It effectively simplifies access management due to the smaller number of attributes compared to the number of users in the system [11].

The costs associated with the storage devices have been decreasing due to the advancement of storage technology. As compared to blockchains, the cost of cloud storage services based on a centralized system are gradually increasing. From this point of view, the future requires a decentralized storage system, which is independent of third-party interference, that honestly stores and transmits the user's data. After the advent of Bitcoin, its underlying blockchain technology provides a kind of decentralized storage facility [12,13]. The implementation of distributed file systems is expected to become a promising research field because of the peer-to-peer study, such as Napster [14], Morpheus [15], Gnutella [16], and Kazaa [17]. On the contrary, Bitcoin [18] is one of the most popular P2P network systems and supports up to 100 million users. Blockchain is a hot topic for the business community and technology giants [19]. In the network, system clients and storage resources are dispersed to form a distributed file system, where every user is a consumer and creator of stored data.

The expectation of ensuring trust and reducing overhead for IoT systems [20,21] has led the combination of Blockchain technology with IoT to become a promising trend, through which a publicly verifiable, decentralized, and credible database can be established, and a distributed trust of billions of connected things can also be achieved. In our daily lives, the involvement of electronic devices are increasing day by day. For example, an automatically repairing order by the coffee machine, the identification of parking lot usage, and the detection of rubbish bin fullness are all electronic devices used daily [22].

1.2. Paper Contributions

In our proposed work, we propose a blockchain-based architecture similar to the one proposed in [23] for enhancing the IoT security and privacy and to overcome the authentication and access control issues present in existing IoT systems. Moreover, the main contributions are as follows:

- We propose a blockchain-based network for reliable data sharing between resource-constrained IoTs.
- Storing the huge data generated by IoTs, a distributed file system, i.e., IPFS or swarm, is used.
- Proof of Authority (PoA) is used instead of Proof of Work (PoW), which increases throughput and reduces the system latency.
- A smart-contract-based access control mechanism is implemented to securely share data.
- Through smart contracts, the data ciphertext can be stored in the blockchain by the data owner.
- Data can only be decrypted in a valid access period given by the data owner.
- In blockchains, the trace function is achieved by the storage of invocation and the creation of smart contracts.
- Validating the effectiveness of cpabe and the access model, extensive simulations are performed in pylab, and the performance parameters are the total cost consumption and cpu utilization.
- To resolve the scalability issues, different kinds of blockchains have been used for data storing and data sharing.
- The simulation results show that our proposed scheme significantly reduces the execution and transaction cost as well as the verification time of the transaction in a blockchain.

The rest of the paper is organized as follows: Background information and the motivation behind the study are provided in Section 2. Preliminaries are discussed in Section 3.

Section 4 shows the literature review, whereas the system model and proposed methodology are demonstrated in Sections 5. Section 6 gives a description of our policy model. The attacker model, security assumptions, and security features of the proposed model are to be considered in Section 7. In Section 8, implementations related to the performance evaluation have been provided, and finally, future work and conclusions are provided in Section 9.

2. Background and Motivation

Over the past few years, the efforts and interest of using sensors and devices in our daily life have been increasing. The smart and socially skilled objects' development is also increasing, which revolutionizes IoT [24] aspects, such as social interaction modeling research and human management investigations. To address these aspects, many architectures have been proposed by researchers. The latest three architectures are the social IoT (SIoT [25]), multiple IoT [26], and multiple IoT environment [27]. With the evolution of these architectures, severe privacy and security issues have been caused. To address these issues, in the last decade, different solutions have been proposed in terms of access control [27,28], intrusion detection [29,30], and privacy [31].

IoT's privacy and security with interconnected internet cause particular challenges in areas of the computing network. It means that at every moment, from everywhere, an attack can be created on the internet resources. As a result, numerous threats, such as denial of service, fabrication of identity, physical threats, communication channel targeting, and many more, have emerged. The biggest challenge in this research field is power resource consumption and computational overheads on IoT devices. Many solutions have been proposed by researchers, where strategies based on blockchain, homomorphic encryption with data collecting objects, and attribute-based encryption for achieving integrity, are provided [32].

IoT devices play a huge role in different aspects of life, e.g., security, energy, safety, healthcare, smart grid, vanets, industry, entertainment, and can directly impact the quality of life. However, in terms of battery power, network protocol, high-level computation, and their infrequent connectivity, they have fundamentally constrained resources. Due to these constraints, sustaining user privacy impacts the applicability of using advanced technology. The huge risk of interconnected devices on the internet without having any standard security scheme implementation is also present, from which security concerns, such as data misuse, arise [33].

IoT devices collect personal information of users, such as their identity, contact number, energy consumption, and location, which is more dangerous than simple security threats. These devices reveal users' information about their daily activities (e.g., watching movies, playing, home activities, and gatherings).

Recently, the interest and efforts in IoT security have been growing. IoT can offer a variety of services, whether they are of safety or non-safety applications. The most important objective of enhanced safety in IoTs is to enhance the user′s security by providing location privacy in a comfortable environment. From a non-safety perspective, many applications and services, such as internet access, geo-location information, the weather forecast for the comfort of user′s convenience as well as infotainment, are considered non-safety services [34,35]. However, in terms of power consumption, network connectivity, high-level computation, and their infrequent connectivity, they are have fundamentally constrained resources [36,37].

Due to these constraints, sustaining user privacy may impact the applicability of using advanced technologies [38]. The huge risk of interconnected devices on the internet without having any standard security scheme is data misuse [39,40]. The challenging task for the researchers in this research domain is power resource consumption and computational overheads of IoT devices [41,42]. Many solutions to the mentioned challenges have been proposed by researchers [43,44]. However, the solutions that are based on blockchains, homomorphic encryption with data collecting objects, attribute-based encryption for achieving integrity are dominant. We address the user transparency, security, privacy, and data sharing incentive issues by proposing a new smart-contract-based technique that relies on data sharing and user control privacy policies [32].

2.1. Existing Access Control IoT Architectures and Related Challenges

In a constrained environment, the application of lightweight security mechanisms is required by the integration of physical objects. However, solutions designed with the current access control and security standards are not meeting the requirement of nascent ecosystems. Lightness, interoperability, end-to-end security, and scalability issues have recently attracted researchers' attention. Existing IoT architectures are outlined below.

2.1.1. Centralized Architecture

This approach consists of a trusted third party's involvement for providing outsource access control operations. The devices are managed by a gateway or back-end server known as the Policy Decision Point (PDP). In stored access policies, the access requests are analyzed by the server, as shown in Figure 2.

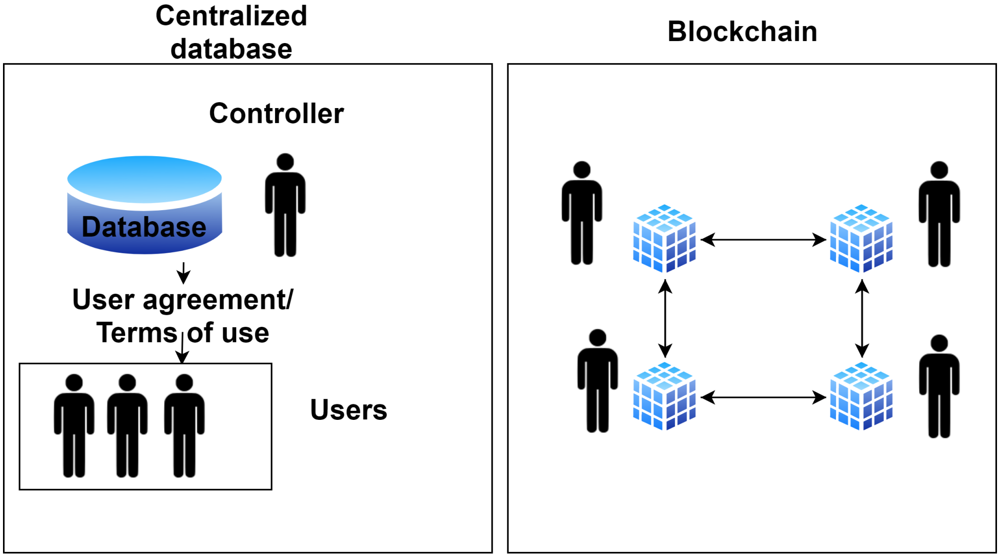


Figure 2. Central vs. blockchain architectures.

To access the end device's data, the requesters should ask to pass by those trusted third parties. This architecture relieves the processing burden of constrained IoT devices (actuators, sensors, etc.). However, major disadvantages are seen in the context of IoT architecture. By the use of a trusted third party, its end-to-end security drops. In the decision-making process, the IoT devices role is strictly limited. The authorization requests of users and resource owner (RO) access control policies are revealed by the trusted third party. The privacy of the resource requester or owner is corrupted due to these conditions.

2.1.2. Trust Entity with Decentralized Architecture

The partial participation of IoT devices in access control decisions are present. From the surrounding environment, the contextual information was sent to a trusted third party that was gathered by IoT devices (e.g., power level, location, etc). The decision made by the trusted third party was based on the access control requests with pre-defined policies and the smart objects' contextual information collection, as shown in Figure 3. To transfer the information in a secure communication channel between the end devices and the trusted third party, the additional security measures are required. In real-time scenarios, such as healthcare, it is not suitable because of the nature of the contextual information transfer; thus, it will not help in real-time access decisions. The requester and data owner's privacy is also not considered.

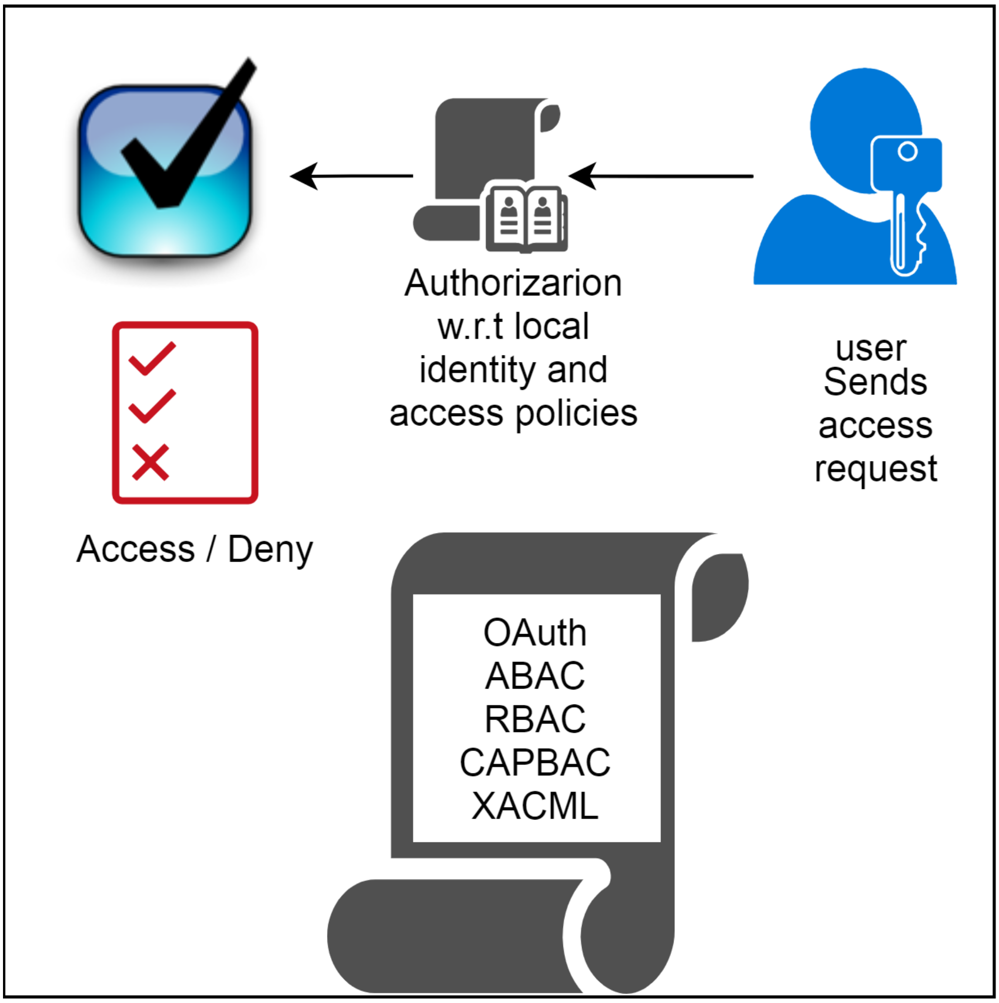


Figure 3. Existing access control architectures.

2.1.3. Distributed Architecture

In the device side, the processing of access control decisions is done in a distributed manner. Due to the absence of a trusted third party, it shows impressive advantages regarding the requester and resource owner privacy. The end users obtain more power in defining their own policies and access control decisions with its edge intelligence principle. Real-time smart access control decisions are also possible. The generated data of IoT devices are less expensive in terms of cost management because the cloud back-end for each device is not provided. The devices only have the authority to transmit information in necessary conditions, and the achievement of end-to-end security makes it more secure than the previous approaches.

2.2. Issues Faced by the Present Architectures

As shown in Figure 4, cloud-based servers, which have large storage capacities and processing power, are connected with trusted entities that can have either decentralized or centralized approaches. IoT devices' authenticated and identification techniques are discussed in [45], which are useful for small-scale IoT networks. However, it is not useful for large IoT networks for the following reasons [46,47].

- Cost: Due to two main reasons, the IoT solutions are expensive:
    - Infrastructure cost: There are billions of connected IoT devices that generate and store a huge amount of data, while the servers are required for their interconnected communication costs and the analytical processing.

  - High maintenance: Updating the software in the millions of IoT devices that have a centralized cloud architecture and huge network equipment requires a high maintenance cost.
- Scalability: The huge amount of IoT devices' data generation and processing (big data) causes a bottleneck to scaling the centralized IoT architectures. Data acquisition, transmission, and storage can be handled by these application platforms.
- Single point of failure: In critical healthcare systems, it is very important to collect the data timely. However, in cloud servers, a single point of failure may cause the whole network to shut down.
- Lack of Transparency: Transparent security architecture needs to be developed because of service providers' irrefutable lack of trust for data collection by the millions of IoT devices in centralized models.
- Insufficient security: A huge amount of connected insecure devices on the internet is a major challenge in IoT privacy and security due to recent DoS attacks [48].

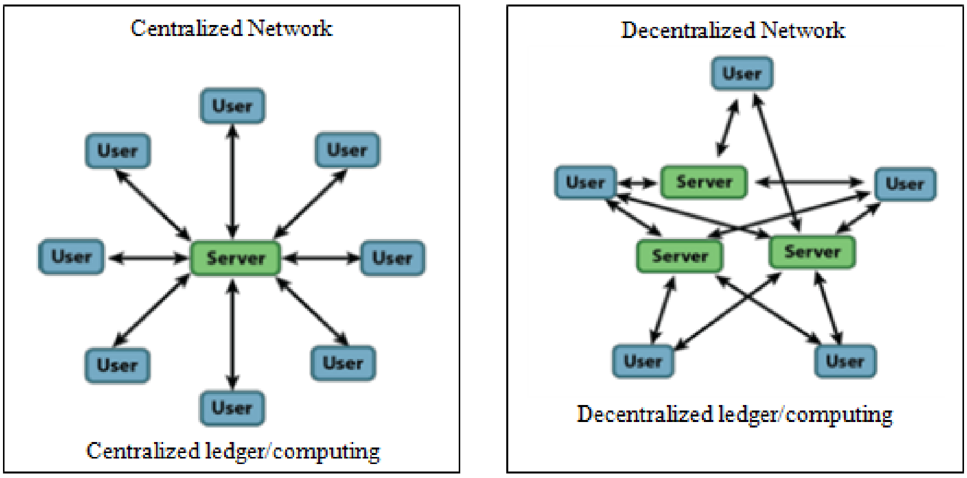


Figure 4. Centralized vs. decentralized networks.

3. Preliminaries

3.1. Blockchain

In the simple form, a blockchain is a distributed and decentralized ledger. Blockchain is a technology based on a distributed ledger initially developed for crypto-currencies, such as Bitcoin. In 2008, Satoshi Nakamoto introduced blockchain technology, which gained attention over the years for its decentralized nature of data sharing and distributed network of computing [49].

Blockchain consists of three main components: nodes, miners, and blocks, as shown in Figure 5. Each block contains the nonce, hash, and data, but it does not have fixed block limits. To secure the blockchain transactions, the nonce is joined with the data for the collection of hash. The block is added after the mining process, in which a complex mathematical problem is solved by the miners to find the nonce. To hack the blockchain, high computational power is required, which is difficult for hackers. Due to its distributed nature, as the number of blocks increases, it becomes more and more secure. The genesis block is the first block of every blockchain. With the consensus mechanism, the addition of blocks to a blockchain network with the majority of nodes' approval is done.

3.2. Multichain

Multichain is a platform for the deployment and creation of a private blockchain between organizations. It aims to overcome the control and privacy obstacles present in the deployment of blockchain structures. For easy integration with existing systems, it can easily work with windows and UNIX servers with the addition of a simple command line and simple API interface.

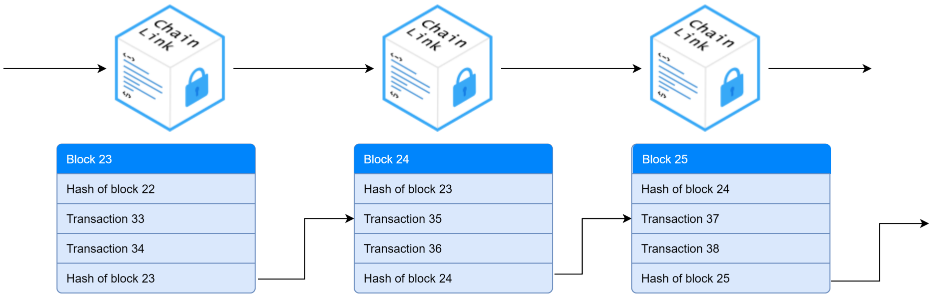


Figure 5. The structure of a blockchain.

A multichain's three main objectives to solve the problems of openness via the integrated management of user permissions, privacy, and mining are:

- To permit the selected transactions only;
- To permit the selected participants to see the blockchain's activities;
- To conduct mining securely and without the associated costs of proof of work.

To resolve the scalability issues, multichain allows the users to set all the parameters and the maximum block size of the blockchain in a configuration file [50]. Because the blockchain contains the participant's selected transactions that are of interest, it contains hash up to 1 GB of off-chain data with auto delivery in the peer-to-peer network. The genesis block's miner can automatically receive administrative privileges, including the management of other users and their accessing permissions.

3.3. Smart Contracts

Computer programs and codes that can work anonymously are known as smart contracts. In a public blockchain network, all participating nodes have the privilege of deploying the smart contract without any specific requirements. For this functionality, the network participants pay a certain fee and agree on explicit conditions. In Ethereum, solidity language is used for creating the contracts, while Metamask [51] is used for Id creation. Finally, Remix IDE [52] is used for its online demonstration and application results. Banking, supply chain, IoT, and insurance industries are deploying permissioned smart contracts. A smart contract is also considered an agreement or consensus between the two parties. Users cannot alter or delete the smart contract once it is published on the blockchain network. No central authority involvement is needed for the validation of tasks. The results computed by the vehicles and nodes do not have any interference from outside the network. Through smart contracts, mobility services and smart transportation are implemented and defined in IPFS by J Benet et al. [53], in which an infrastructure based on distributed ledger technology (DLT) with distributed data management technologies has been used for data sharing and smart services. In IPFS, an Ethereum smart contract and an IOTA-based architecture for authenticity have been proposed by Zichichi et al. [54], in which the entities' coordination, access authorization, and users' privacy have been achieved. Zero-knowledge proof was used for the privacy offer, and a proof of location guarantee was used. The rules stored by a smart contract include the following.

- The negotiation of terms;
- Automatic verification;
- Agreed terms execution.

Different kinds of functions that a smart contract consists of might be extracted from other smart contracts or outside the blockchain. The reliance between transaction parties on a central system can be removed due to the combination of smart contract and blockchain technology. All the parties present in the blockchain network have a copy of the stored smart contracts. The execution of agreed terms present in the smart contract are triggered by an authorized event. Every transaction's audit trail of events is stored. All the parties present in the network can detect the changes in the transaction or contract. Therefore, it creates a large secure system without having a centralized model's trust, costs, and risks issues.

To write the smart contracts, solidity programming language has been used due to its lightweight coding condition. For the representation of each operation in the contracts, Ethereum Virtual Machine code is used. The message data with the amount of Wei is sent in the transaction as output, and a byte array is returned. A truffle framework is used for testing and the deployment of Ethereum-based smart contracts.

4. Related Work

With the significant growth in the number of IoT devices, it has become a challenge to store IoT data and an even bigger challenge to protect that data from unauthorized access and harm. Another issue is trust; centralized servers are not always honest. These issues are addressed in [55]. In order to remove these central servers from the system, the authors have used blockchain and certificateless cryptography for storing and protecting the data. Edge computing has been used for data storage management, whereas an un-validated IoT framework has been presented [22].

4.1. Ethereum-Based Existing Access Control Schemes

In [56], a scheme is proposed for data storage and sharing using an encryption based on Ethereum blockchains and attributes. A keyword search utility is provided using a smart contract. An attribute-based access control mechanism is designed in [57] for IoTs to simplify access management. To avoid a single point of failure and the loss of integrity, a blockchain is deployed. The access control mechanism is deployed for low-cost computations across IoT systems.

A scheme for providing availability and a keyword search is proposed in [57] using blockchains. This keyword search function is different from that of [56] since the permission for the keyword search is granted by the data owner in this scheme.

An attribute-based encryption scheme for encryption, keygen, and decryption with verified outsourcing is proposed by Wang et al. The ciphertext complexity and size was increased with the number of attributes in the access policy. It successfully reduces the execution time but suffers from a high communication cost because computationally expensive operations are performed by the encryption proxy server [58].

Many access control solutions that have a centralized model have been designed for IoTs [59–61]. As a result of adopting a centralized system, there have been a lot of issues, such as low scalability, no transparency for users information, and built-in interoperability is also not provided. Access to a distant centralized server mostly requires connectivity, and the access control decisions were moved away from the edge nodes. Many of these issues are resolved by using the decentralized approaches presented in Table 1. In the recent proposals presented in Table 2, the decentralized-based access control systems in IoTs by using blockchain technology have been listed.

Table 1. Existing Blockchain Techniques.

Ref	Technology Used	Contributions	Addressed Problems
[15]	IoT and blockchain	Blockchain-based simple mechanism for database	IoTs applications Database
[62]	IoT, smart contract, and blockchain	A blockchain, smart contract, and IoT combination is used for identifying solutions	Complex processes automation
[63]	Blockchain edge/fog computing	Edge/fog working relationship with blockchain	Blockchain-enabled fog applications
[64]	IoT and blockchain	In IIOT, traceability and revocability with a blockchain-based access control system	Malicious users tracking and revocation
[65]	IoT, smart contract, and blockchain	Web interface for controlling entities information with smart contracts	Identity, interoperability, and security of IoT

Table 2. Overview of existing literature.

Ref	Author	Description of Research	Techniques	Contributions	Evaluation Criteria	Limitations
[55]	Li et al. 2018	Blockchain for large-scale IoT data storage and protection	Distributed Hash Tables (DHTs) and edge computing	Security, accountability, and traceability	Transaction verificatio and distributed data storage	User authentication not provided; Will not work on a complicated access control scheme
[56]	Wang et al. 2018	Blockchain-based fine-grained decentralized storage scheme	IPFS, Ethereum, and attribute-based encryption (ABE) technologies	Secure access control policies achieved; keyword search function; wrong results in the traditional cloud storage is solved	IoTs authentication and attribute-based AC	Attribute revocation is not considered
[57]	Ding et al. 2019	Blockchain-based access control scheme	Elliptic curve digital signature algorithm and AKA protocol	Scalability, robustness, IoTs consensus independence, low computation, and communication overhead	Security authentication in the decentralized AC	No real-time scenario is considered
[66]	Do et al. 2017	Blockchain-based private keyword searching scheme	Proof of storage and distributed encrypted data storage	Data integrity and enforcing proof-of-retrievability	Anonymous access control and off-chain	Outsourcing data storage; does not support credential revocation; Boolean keywords
[67]	Zhang et al. 2018	Blockchain/cloud-based data storage scheme	Cloud and hyper ledger fabric	identity management, fine-grained access control, scalability, and distant access	Data chain and behavior chain permission levels	Single-system restriction and authentication
[68]	Steichen et al. 2018	Blockchain-based decentralized access control for IPFS	Smart contract, IPFS, and Ethereum	Sharing of large sensitive files	Fixed gas amount	Authentication of nodes; more time-consuming
[69]	Sifah et al. 2018	Chain-based big data access control infrastructure	ECDSA and PoC	Off-chain sovereign blockchain	security and data mismanagement and execution time	Inefficient in industries
[70]	Zhang et al. 2018	Smart-contract-based access control for IoT	Ethereum smart contract platform	distributed and trustworthy access control for IoT systems	Gas price and timing	Overhead and capital cost

4.2. Cipher-Text-Policy-Based Attribute-Based Encryption (CP-ABE) Schemes

A CP-ABE-based outsource ABE scheme was proposed by Nguyen et al. [71]. In this scheme, the users only specify the access policy before passing it to the delegatee (DG), and the key generation center is responsible for the delegation key generation. Encryption of data with an access policy is done by the delegatee.

By storing and pre-computing tuples, the authors in [72] speed up the encryption process. The number of attributes in an access policy is directly proportional to the number of tuples created during pre-computation. This requires extra memory to re-run pre-computation after modifying the access policy. The size of the cipher text increases with the number of attributes.

The access policy hiding in CP-ABE is an active research area. It supports many kinds of access policies, such as Tree-based [73], threshold-based [74], AND-based [75], and the linear secret-sharing systems matrix (LSSS) [76]. The access policy hiding in CP-ABE was first introduced by Nishade et al. [75]. Multiple values of AND gates have been used that have a limited range of expression. To reduce the cipher text size and hide the access policy, schemes based on AND gates have been proposed in [77,78].

Sarhan and Carr proposed a distributed cryptographic agent-based secure multiparty computation (ADB-SMC) access control, in which secure multiparty computation and active data bundles can be combined with ABE. Instead of using the blockchain infrastructure, distributed hash tables have been used, which affect the infrastructure costs but do not reduce the communication and computation overheads.

Cipher text policy-based Attribute-Based Encryption (CP-ABE) enforces the policies in an encrypted format that is useful for sensitive information. Most of the existing CP-ABE schemes generate large-sized cipher text and secret keys. The cipher text and key size is linear with the involved attributes, and the number of bilinear mapping pairs is directly proportional to the attribute size. Because bilinear pairings are used in ABE, its use is challenging for IoT devices due to the heavy computation for their small storage and computation capacities. The use of CP-ABE with timely and minimal bilinear pairings affects the access control computation in our work. Therefore, a comparison chart with other access control schemes has been presented in Table 3, in which the features of our model are illustrated.

Table 3. Comparison with other models

Ref No	Blockchain	Scalability	Adaptability	Cost-Effective	Privacy Efficiency	Access Period
[79]	*	x	*	*	*	x
[70]	*	x	*	*	*	x
[71]	*	*	*	x	x	*
[55]	*	x	x	*	x	x
our model	*	*	*	*	*	*

5. System Model and Proposed Methodology

In our solution, we propose an attribute-based access control mechanism for IoT devices. By using Blockchain technology with cipher-text-policy-based attribute-based encryption (CP-ABE), we avoid data tempering and eliminate a single point failure. For lightweight authentication and meeting the high efficiency in IoT, we optimize the access control process by creating smart contracts. We use two kinds of blockchain networks: a public blockchain for authentication purposes of the IoT devices, attribute servers, and storing the user-defined policies, as shown in Figure 6. Conversely, in the consortium blockchain, the hashes of transactions have been stored after the validation of user and devices.

A typical IoT scenario is depicted in Figure 6. In an IoT system, three entities are evolved—IoT devices, attribute servers, and the gateway. Devices, such as mobile phones, computers, and smartwatches, can easily access the direct wire or WiFi connection. Conversely, the dedicated gateway is required by certain lightweight devices. The registration server is responsible for the collection and authorization of IoT devices and users. After the server authentication, numerous data access requests and exchanges can be performed by such entities.

In our blockchain network, each node has its own account through which the trade transactions are performed. A pair of public and private keys are assigned by the registration server for the signing and addressing of transactions, which proves the identity of the user and cannot be altered by any entity. Smart contracts and transactions are recorded on unique addresses by the distributed blockchain. Therefore, every interaction of the user will be considered as a transaction and recorded in the blockchain, which provides transparent user access and traceability. To resolve scalability issues, multiple blockchains are used for the generation of transactions and the deployment of smart contracts.

By using smart contracts, the access control management and authorization are provided. Every device requires its own credentials, and a user owns one or more IoT devices. Therefore, each device would be individually authenticated by the user. This would create an authentication overhead; however, by using smart contracts, the users with their devices can be registered in a public Ethereum blockchain. By using a wallet address, the user and its devices can be verified by attribute authority, and then the transactions are performed. By using access control smart contracts, a single authorization server can be replaced by a distributed authorization server.

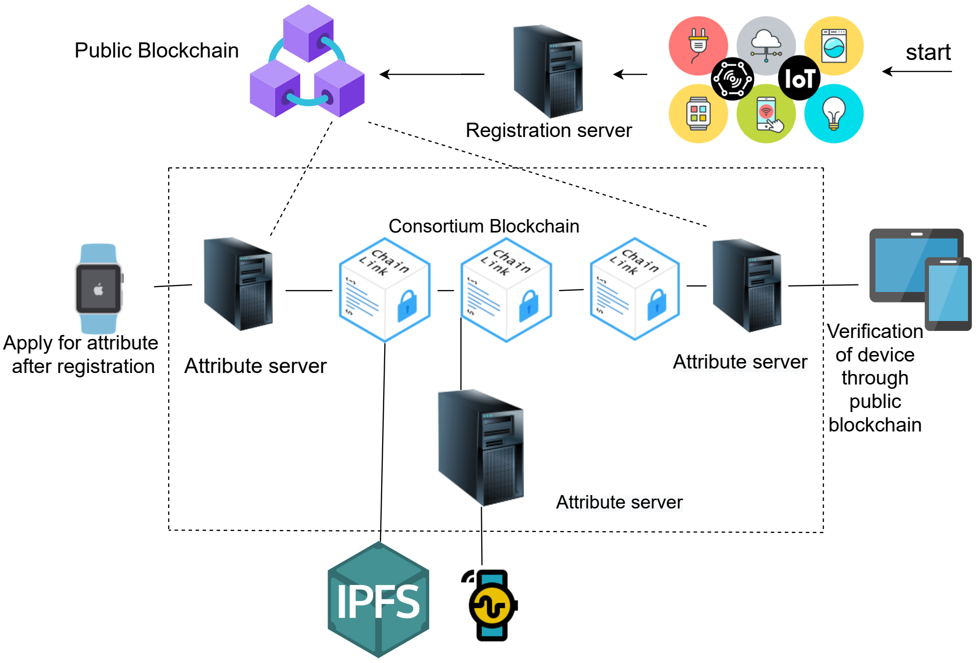


Figure 6. The system model of our proposed architecture.

5.1. Smart Contract System

The mechanism consists of four smart contracts, as shown in Figure 7, that are implemented on the Ethereum blockchain. The access control contract (ACC) consists of the object attribute management contract (OAMC) and the subject attribute management contract (SAMC), whereas the policy management contract (PMC) holds the policies of each subject and object with their specified actions. The addition and deletion of attributes is handled by the ACC and PMC.

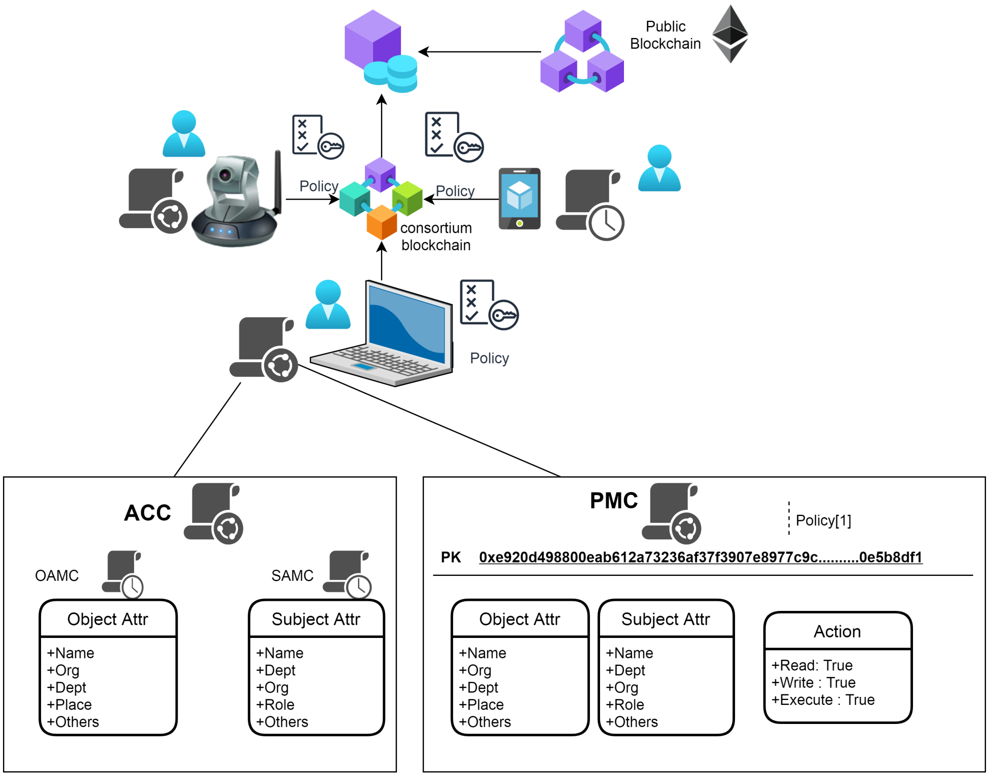


Figure 7. Building blocks of the access control mechanism.

5.2. Access Control Contract (ACC)

In IoT systems, the requests from subjects to objects can be controlled by the ACC. Subjects can execute the ACC by sending the required request information of transactions. After successful authentication is provided by the PMC, OAMC, and SAMC, the ACC can retrieve the subject and object attributes with the concerned policy information and verify the results.

5.3. Subject Attribute Management Contract (SAMC)

SAMC is deployed for the management and storage of IoT system attributes. Subject administrators only have the authority to execute smart contracts. For example, the administrators are owners in the case of IoT, whereas in the case of citizens, the city office acts as an administrator. Each subject can be represented by a unique identifier in the system. In our paper, we use an Ethereum account as an ID of a subject. Multiple attributes are associated with each subject ID, as shown in Figure 7. In addition, deleting and updating subject attributes can also be handled by the SAMC.

5.4. Object Attribute Management Contract (OAMC)

Object administrators manage and store the attributes of an object with the execution of the OAMC. Multiple attributes are associated with uniquely identified Ethereum accounts. Table 4 shows the attributes involved in our model.

In addition, deleting and updating object attributes can also be handled by the OAMC through the application binary interfaces (ABIs) of objectdelete() and objectadd().

5.5. Policy Management Contract (PMC)

Attribute-based access control policies can be managed and executed by the policy management contract (PMC). Only the policy administrators have the authority to execute the policies. A policy is a combination of subject and objects attributes with their specified actions, as shown in Table 5. For example, subject attributes are Department=B:Organization=C, and object attributes are Department=B:Organization=C . Then, Policy=Read only states that the user can only have read access.

Table 4. Subject and Object Attributes.

Subject Attributes	Object Attributes
Name:	Name:
Dept:	Dept:
Org:	Org:
Role:	Place:
Others:	Others:

Table 5. Subject and Object Attributes with Actions.

Subject Attributes	Object Attributes	Actions
Name:	Name:	Read:True
Dept=IS:	Dept=IS:	Write:False
Org:COMSATS:	Org:COMSATS	Execute:False
Role:	Place:	
Others:	Others:	

5.6. Data Sharing Model

In this section, as shown in the Figure 8, a user can upload the data after verification from the public blockchain through the attribute server by implementing attribute-related policies. Once completed, a user can extract the information if he/she satisfies the predefined conditions. The contract also provides policy updating, revocation of a policy, and an ownership transfer. The contract for managing the attribute-based access control system is written in Solidity and compiled using compiler version 0.4.20. For this purpose, we use multichain to resolve the scalability issues present in the blockchain technology. In the policy model, the detailed terminology has been defined. On the request of the data user, encryption based on the cipher text policy has been done in a timely manner. Multichain allows the users to set all the parameters and the maximum block size of a blockchain in a configuration file [50]. A blockchain with the participant's selected transactions contains hash up to 1 GB of off-chain data with auto delivery in the peer-to-peer network. The administrative privileges can be automatically received by the genesis block's miner, including the management of other users and their accessing permissions.

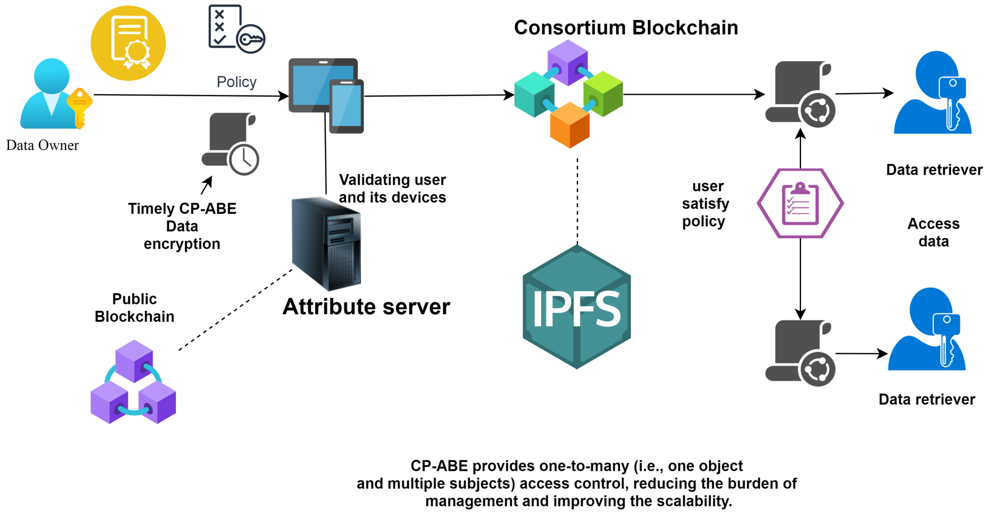


Figure 8. The data sharing model.

6. Policy Model

In our scheme, encrypted files can be stored using smart contracts. By running encryption and decryption algorithms, the data owners can store and retrieve their data through the implementation of smart contracts. On the blockchain, every contract call has been recorded. Therefore, the information between the data owner and user is non-tempered and non-repudiated. In our model, four entities have been evolved: the data owner, the data retriever, IPFS, and the Ethereum blockchain.

1. **Data owner:** Upload encrypted data with assigning attributes sets and access control policies and is responsible for the creation and deployment of smart contracts.
2. **Data user:** Access the encrypted data stored on IPFS. After satisfying access control polices and attribute sets, the secret key is obtained, which decrypts the encrypted data.
3. **IPFS:** Used for the storage of encrypted data that can be stored by the data owners.
4. **Ethereum:** To store and retrieve the data, smart contracts have been deployed on the Ethereum blockchain.

The process in Figure 8 is as follows:

- After the device and user registration process using blockchain technology [65], the data owner uploads the encrypted data with access control policies in smart contracts.
- The returned contract address with the encrypted data hash would be stored on IPFS.
- The path of data stored in the IPFS location can be returned to the data owner.
- In Ethereum, the encrypted data key has been stored in ciphertext format.
- When the data retriever sends the access request using the timely CP-ABE, the data owner adds the policies under the effective period, encrypts the secret key, and stores it in a smart contract.
- The data retriever that satisfies the access policies in an effective period of time downloads the data and obtains the secret key from the contract.

6.1. Attribute-Based Encryption

We have implemented the cipher-text-policy-based attribute-based encryption (CP-ABE). Ciphertexts are attached to access policies, and attribute sets are associated with secret keys. The secret key is used for recovering the cipher text if attribute sets satisfy the access policy. The encryption of data in attribute-based encryption can be handled under an access policy with certain attributes. During data encryption, the cipher text contains a part of the access policy in the CP-ABE. Data encryption in classic public key cryptography can be done for a specific individual entity using its private key. In this case, the sender must know about the receiver and his public key. During the continuous changes in such constructions, the addition and removal of the collaborator is done with every encrypted dataset. Therefore, the encryption has to be done for every legitimate identity. For such cases, the hybrid schemes have been proposed, but these schemes contain the limitation of handling increasing participants.

CP-ABE allows a user to encrypt the data using attribute-based encryption instead of knowing the respective individuals of those attributes. Through the cryptographic mechanism, traditional access control systems' trust issues can be solved, which is a silent feature of attribute-based encryption. In that case, only legitimate users can decrypt and access the data stored publicly. Individually generated private keys and attributes assignment has to be done by the key management authority. However, the absolute trust needed by a key server to issue a private key to only legitimate users and to revoke a user's key is a major drawback in existing schemes. Access rights transparency has also not been provided. We address these issues in this paper. An example of encryption has been presented in which the user who satisfies both notations can decrypt the data.

6.2. Access Policy

- Access policy P is a rule in ABE that returns either 0 or 1.
- Attributes set is A (A1, A2, . . . , Am).
- If P answers 1 on A, only then can we say A satisfies P.
- Usually, to represent the fact that A satisfies P, the notation A = P is used.
- The case that A does not satisfy R is denoted as A!= P.
- We consider the AND gate policy in our construction.
- If Ai = Pi or Pi=* for all 1 <= i <= m, we say A = P; otherwise, A!= P.
- It is noted that the wildcard * in P plays the role of a "do not care" value.

For example:

access policy P = (Clinic : 1; physician ; * ; Pakistan);
a attributes set A1 = (Clinic :1 ; physician; male; Pakistan);
A2 = (Clinic :1 ; Nurse; male; Pakistan);

Then
A1 = P , A2 != P.

With the combination of the ABAC model and the data generated by IoT devices, the flowchart of our models's access control policy is defined in Figure 9, where Policy (P) = (AS, AO, AP, AE):

- Attribute Subject = (userId,role, group);
- Attribute Object = (deviceId, MAC);
- Attribute Permission = (1, allow 0, deny);
- Attribute Environment = (createTime, endTime, allowed).

The access control with data storage has been composed based on the following algorithms:

1. **Setup (PK, SK):**

Data owners execute the algorithm with the inputs, universal attributes set A, and security parameter P, resulting in a public and secret key pair. Afterwards, the data can encrypt with the AES encryption algorithm and hash using the SHA 256 algorithm as H(data). Along with these attributes, the encrypted data can be uploaded, and in return, the address or path of those data can be returned by the IPFS server.

2. **Encrypt (PK, T, sek)-> CT**

The public key, symmetric encryption key, and access tree structure can be used as inputs, and the generated cipher text will be stored in a smart contract.

3. **KeyGen(sk, A) -> PrK**

The data owner executes the key generation algorithm after the collection of access requests by the data retriever. The data owner assigns the data retriever a set of attributes with the effective period of time. The algorithm outputs the private key Prk in return for entering the secret key sk and set A attributes and stores it in the smart contract.

4. **Decrypt (PK,sk,CT) -> sek**

The data retriever executes the decryption algorithm after obtaining the effective access period from the smart contract. It can only be performed within the valid access period. By obtaining the cipher text CT and secret key from the smart contract and entering them into the decryption algorithm with its public key PK, the data retriever can only get the symmetric encryption key sek when it satisfies the access policy T. Afterwards, the data can be decrypted with this key; otherwise, the data owner would change the policy, and no one can access the information.

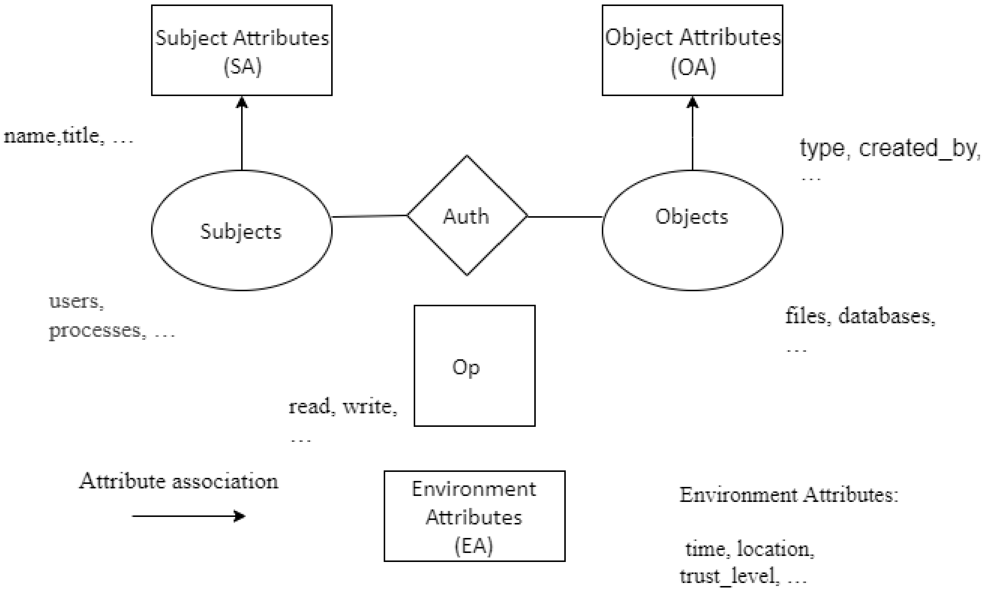


Figure 9. Flowchart of the access policy.

7. Security Assumptions and Attacker Model

In data accessing and sharing among IoTs, privacy and security are the main issues in the existing models. The prevention of privacy leakage and potential security threats are the main concern of our model. For our proposed model, the following attack and security assumptions are considered.

- **Consistency of blockchain over nodes and timing:** Blockchain transactions are accepted by the nodes present in the network.
- **Growth of the blockchain:** The eventual integration of valid transactions into the blockchain.
- **Reliable and trustworthy gateway:** The trustworthy and accessible gateway is assumed.
- **Trusted entities:** Attribute servers and certification authority are trusted.
- **Security of keys in the blockchain:** Keys are secure and cannot be lost or stolen.
- **Strong cryptographic measures:** Cryptographic primitives, hashes, and signatures are not broken.

The attacker model for our research is given below.

- **Privilege Elevation:** The attacker convinces the device by declaring himself an authenticated attribute entity and promoting a fake attribute-issuing entity. He also replays a valid transaction previously performed by an attribute entity.
- **Identity Revealing Attack:** To reveal the real identity of authorized devices and personal data collection, the malicious entity tries to target the devices.

- **Man-in-the-Middle Attack:** The interception of shared data and data tempering by the malicious node between the IoT nodes and attribute servers.
- **Forgery Attack:** The malicious attribute server has fake keys and signatures of an authentic user and transfers it to other entities to affect the network.

Security features of our proposed model are given below.

- **Privacy preservation:** Crypto ID has been used for the communication between the entities present in our model. To enhance the privacy, we are not using the device's real identification number as its identity. All the transactions are done in an encrypted format that preserves the identities of the users and devices.
- **Data Confidentiality**: Using symmetric key encryption, the communication between the IoT devices are encrypted, which enhances the security and prevents tempering of communication data.
- **Data Integrity:** Data generated by IoT devices are encrypted using symmetric key encryption and stored in IPFS (an example of distributed file system). To provide data integrity, we encrypt the data under a certain cipher-text-policy-based encryption, and its hashes are stored in the blockchain so that data tempering is not possible.
- **Single Point of Failure:** A distributed file system and multiple attribute servers have been used in our model, which eliminate the single point of failure. The attribute servers only interact with the devices of their associated identities, which enhances the system security.

8. Performance Evaluation

To analyze the performance and feasibility of our model, the Ubuntu 16.04 system with 4GB RAM, Intel core i3 has been used for the implementation of the prototype. For smart contracts, solidity language and C++ has been used. The simulation of smart contracts are performed using Remix IDE. Ganache [80] is used for providing virtual accounts and for executing smart contracts, and Metamask, the extension of the chrome browser, is used for Remix and Ganache connectivity. The PBC library is used for computing parings. For testing, we use Truffle for smart contract testing at the development level and use Testnets, e.g., Ganache(local blockchain) and Ropsten (online), for free smart contract deployment. To validate the analysis of the ABE program, we implement the cipher text policy in attribute-based encryption by using a cpabe toolkit. The algebraic operations are done with the PBC library. For the implementation of crypto operations, libbswabe is used, and for user interface and high level functions, cpabe is used.

8.1. An Attribute-Based Access Control Model for IoTs

User and device registration, storing data on distributed file systems, such as IPFS, information and the management of data under specified policies, and its results are shown in Figure 10.

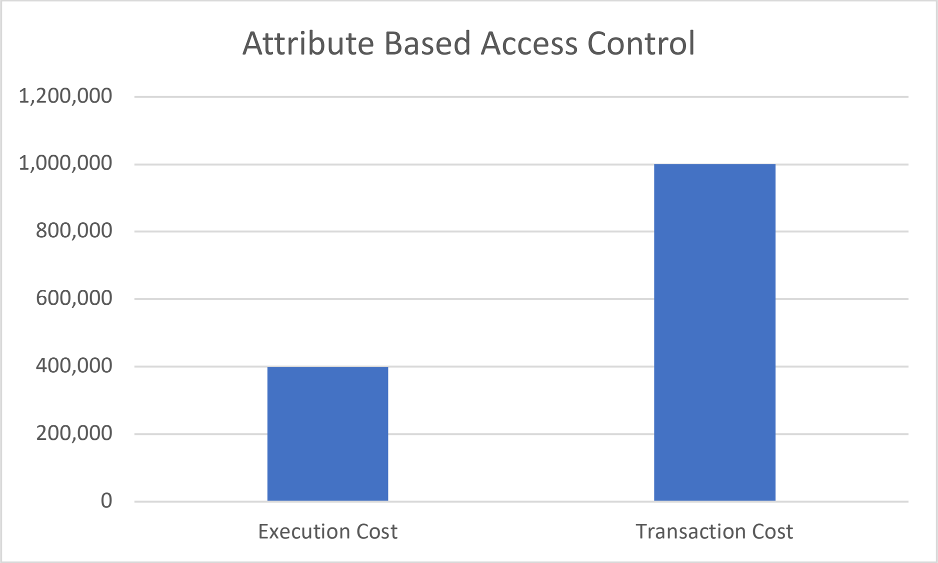


Figure 10. Attribute-Based Access control Model details.

In Ethereum, gas is a small unit of cryptocurrency. The unit is deducted from the users' accounts when performing a transaction in the Ethereum. Figure 11 shows the gas consumption of a smart contract's operation. There are four different functions in the smart contract that are used in the proposed work, which are: (a) grant, (b) init, (c) policy, and (d) request. The gas consumption depends on the complexity of the smart contract. The deployment of the smart contract is an expensive operation in Ethereum.

However, the transaction cost is incurred when sending the smart contract to the Ethereum. The execution cost depends on the operations that are executed as a result of the transaction. Therefore, the execution cost is included in the transaction cost.

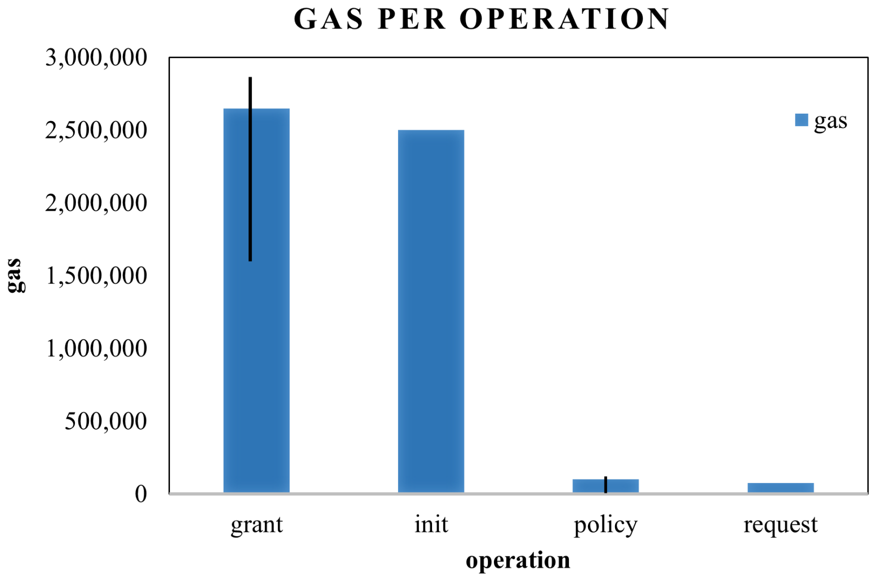


Figure 11. Gas per operation.

Figure 12 shows the processing time for encryption and decryption operations of our scheme. In our scheme, each user's private key is associated with a group of attributes that represent their capabilities. A decryption can only be done when satisfying a certain policy requirement, which is why it took less time than encryption. As the number of attributes increases, the processing time also increases.

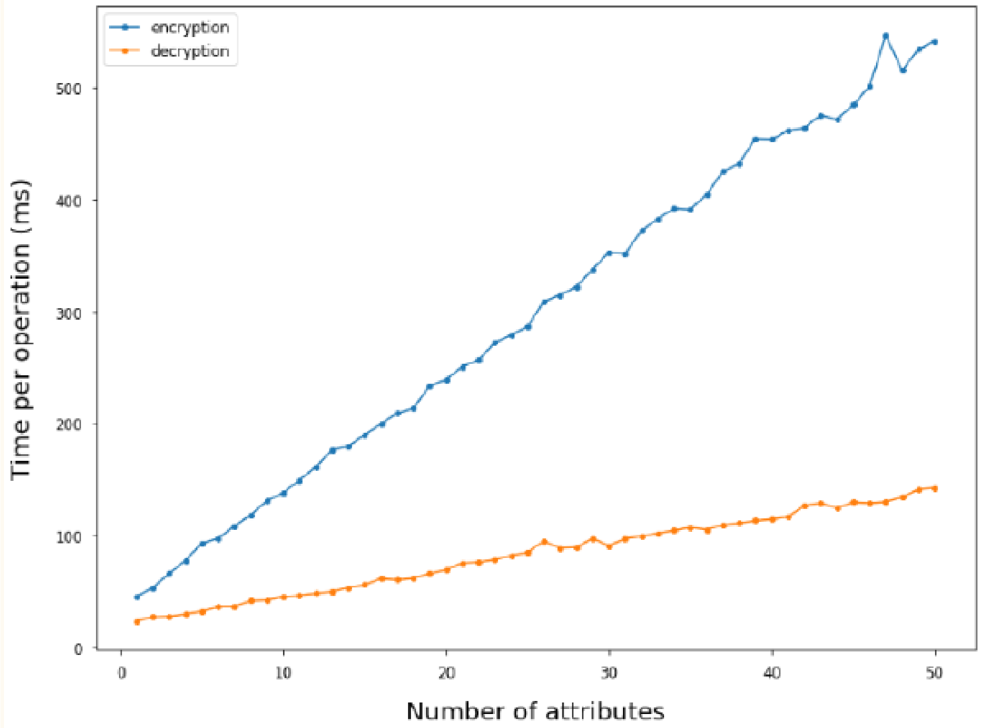
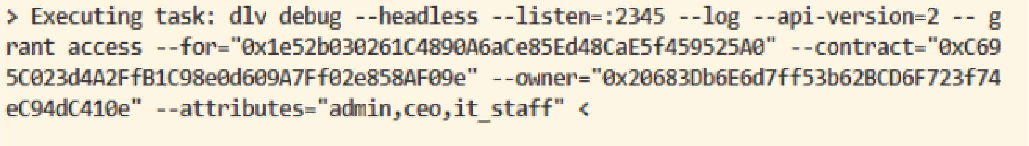


Figure 12. Operation time with the number of attributes.

We used three attributes in the simulation and used the AND-gate-based access structure for ensuring each attribute. The execution details of our system are shown in Figure 13, and the details of sharing data with the owner are also provided. In the registration setup, the web3j library has been used for access control. If the port combination that contained the device name and service name under certain policies successfully verified the transaction, the receiver can access the data.



```
> Executing task: dlv debug --headless --listen=:2345 --log --api-version=2 -- g
rant access --for="0x1e52b030261C4890A6aCe85Ed48CaE5f459525A0" --contract="0xC69
5C023d4A2FfB1C98e0d609A7Ff02e858AF09e" --owner="0x20683Db6E6d7ff53b62BCD6F723f74
eC94dC410e" --attributes="admin,ceo,it_staff" <
```

Figure 13. Access verification.

8.2. Cost Evaluation and Comparison

For the deployment of smart contracts on the blockchain, the execution fee is required from the users for the execution of contracts' ABIs. To perform the tasks on Ethereum, a gas unit will be used to measure the operations amount. More gas will be consumed for more complex tasks. With the passage of time, the gas prices of Ethereum change. The total cost for performing a task depend on the gas price and the amount of consumed gas. We set the gas price to 5 Gwei, where 1 ETH = 1 × 10^9 (1,000,000,000) gwei. For example, if we have a transaction of 20,000 gas, then its cost will be 20,000 × 5 = 100,000 gwei (0.000100 ETH). 1 ether = 226.6946 gas = USD 357.839639 (as we accessed on September 2020), but now, Ethereum is sold as the world's most expensive non-fungible token (NFT). For evaluation, we compare our proposed model with [55,70]. The comparison charts are given in Figures 14 and 15.

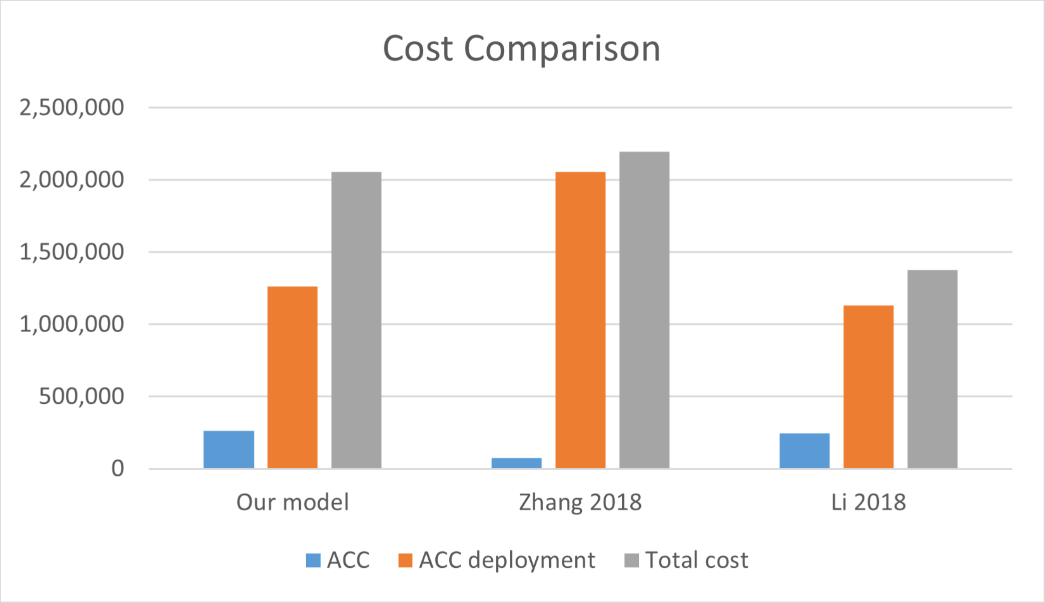


Figure 14. Cost comparison with [55,70].

Instead of using RC and JC in [70] and DR and VT in [55], we are calculating the deployment cost of ACC, PMC, OAMC, and SAMC. The actual access cost of the proposed scheme is 262,531 gas, which is almost USD 4.54325. The chart in Figure 14 shows that our model consumes more cost than [55,70], but there is a monetary gap in the US dollars. In one access control of [55,70], only one-to-one pairing has been done; thus, as the number of subjects and objects increases, the monetary cost of the system also increases. However, many-to-many subjects and objects pairing in the access control are achieved in our model. In the case of [55,70], when subject and object pairs increase, the gas consumption also increases, which costs more than that of our model.

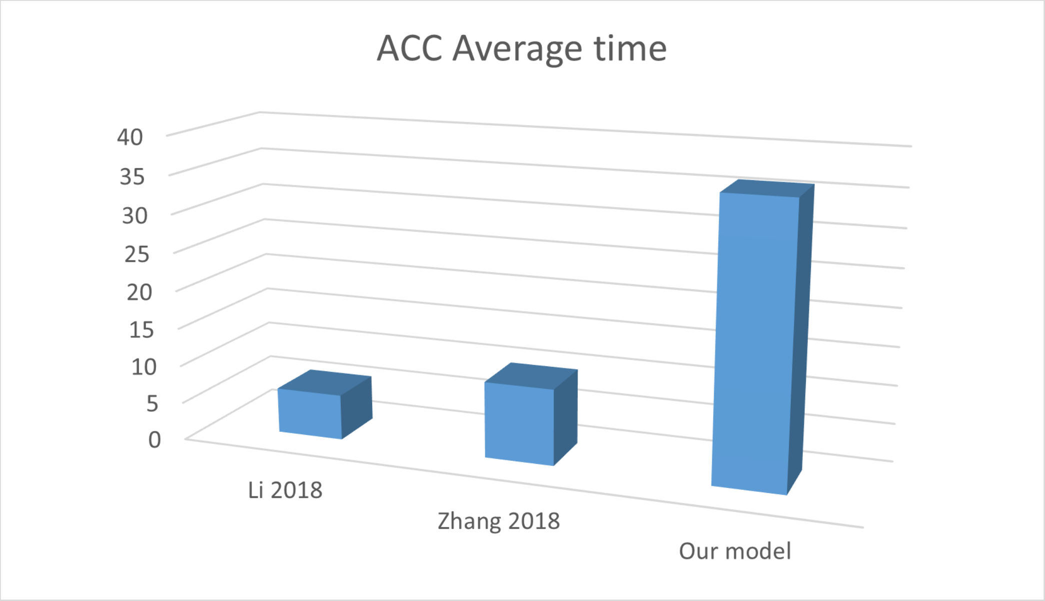


Figure 15. ACC time comparison with [55,70].

Due to the attribute-based encryption and the complex interaction between the access control and other contracts on Ethereum, it takes more time than other schemes, as shown in Figure 15. It also depends on various factors, such as the computational power of system. Additionally, the computational time in Ethereum may also vary time to time, so the time of mining also affects the results. The network architecture also affects the system performance. To evaluate the performance of our proposal, we compare the simple access control with the cipher text policy-based attribute-based encryption and its implementation with blockchain.

Verification costs of access control with respect to the number of attributes used in the policy are shown in Figure 16. We used three architectures to evaluate the results: one is for centralized verification of the access control with timely cpabe; a decentralized access control with timely cpabe and blockchain; and the last one is a timely access control list using blockchain technology. The results show that an additional cost has been adopted by the decentralized architecture. The timely access control list is less efficient than timely cpabe and increases the verification cost. Rather than not providing access verification by the access control list, cpabe provides decentralized access management in a more efficient way.

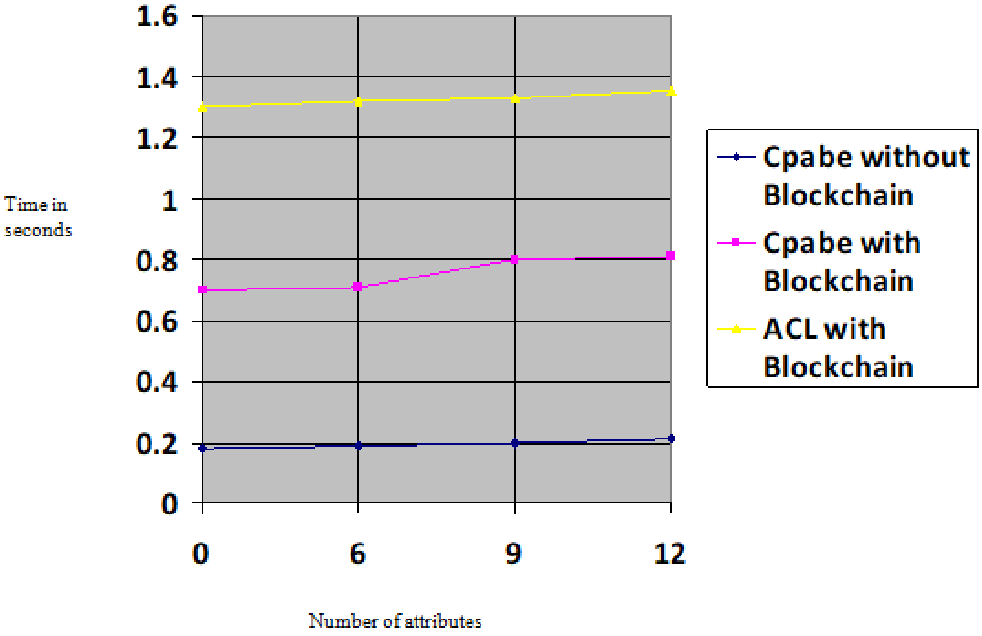


Figure 16. Cpabe performance comparison.

9. Conclusions

We propose an attribute-based access control mechanism for IoTs that provides local access, authorization of clients, privacy, and interoperability by using smart contract data sharing and user-controlled encoded policies. The user can own their data and have authority to share it with other users. No scheme fulfills the requirements of our proposed model. We used the ABAC model for its high compatibility and expressiveness.

We overcome the issues presented in [55,70], which are high computational time and overhead from deploying the number of smart contracts for every additional user with a single point failure and un-authentication of present users, using blockchain for authentication and smart contracts for the data access process in our mechanism. To overcome the data-transfer-related communication assumptions, a secure mechanism of data storage has been introduced. We also made an ownership contract of each user with its own devices to enhance the privacy of our model. It is not feasible for actual user data to be exposed by any entity in our blockchain architecture. The off-chain data are stored in an encrypted format, which makes data tempering impossible. Only a consumer who meets the specific policies can access the data after the invocation of smart contracts. In the future, we will work on the security and privacy of IoT data from unauthenticated edge nodes. Although the blockchain is providing reliability and decentralization, it has a few drawbacks: scalability and monetary cost issues. We will be considering scalability and reliability aspects using IOTA.

Author Contributions: Conceptualization, M.A.S. and H.A.K.; Formal analysis, A.M.E.-S. and M.A.E.-M.; Funding acquisition, A.M.E.-S. and M.A.E.-M.; Investigation, M.A.S.; Methodology, H.A.K. and H.T.R.; Resources, C.M.; Supervision, H.A.K.; Validation, H.T.R.; Writing—original draft, S.Y.A.Z. and H.A.K. All authors have read and agreed to the published version of the manuscript.

Funding: The authors extend their appreciation to King Saud University for funding this work through the researchers supporting project number (RSP-2021/133), King Saud University, Riyadh, Saudi Arabia.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors extend their appreciation to King Saud University for funding this work through the researchers supporting project number (RSP-2021/133), King Saud University, Riyadh, Saudi Arabia.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tung, L. *IoT Devices Will Outnumber the World's Population This Year for the First Time*; ZDNet, A RED VENTURES COMPANY; Volume 1. Available online: https://www.zdnet.com/article/iot-devices-will-outnumber-the-worlds-population-this-year-for-the-first-time/ (accessed on 7 February 2017).
2. Top, G.I. Strategic IoT Technologies and Trends, Gartner. Available online: https://www.gartner.com/en/newsroom/press-releases/2018-11-07-gartner-identifies-top-10-strategic-iot-technologies-and-trends (accessed on 4 September 2019).
3. Ekbatanifard, G. An Energy Efficient Data Dissemination Scheme for Distributed Storage in the Internet of Things. *Comput. Knowl. Eng.* **2018**, *1*, 1–8.
4. Ahmad, I.; Shah, M.A.; Khattak, H.A.; Ameer, Z.; Khan, M.; Han, K. FIViz: Forensics Investigation through Visualization for Malware in Internet of Things. *Sustainability* **2020**, *12*, 7262. [CrossRef]
5. Shrestha, A.; Vassileva, J. Towards decentralized data storage in general cloud platform for meta-products. In Proceedings of the International Conference on Big Data and Advanced Wireless Technologies, Blagoevgrad, Bulgaria, 10 Novermber 2016; pp. 1–7.
6. Meadows, A. To Share or Not to Share? That Is the (Research Data) Question. Available online: https://scholarlykitchen.sspnet.org/2014/11/11/to-share-or-not-to-share-that-is-the-research-data-question (accessed on 9 September 2021).
7. Kiran, S.; Khattak, H.A.; Butt, H.I.; Ahmed, A. Towards Efficient Energy Monitoring Using IoT. In Proceedings of the 2018 IEEE 21st International Multi-Topic Conference (INMIC), Karachi, Pakistan, 1–2 November 2018; pp. 1–4.
8. McMahan, B.; Ramage, D. Federated Learning: Collaborative Machine Learning without Centralized Training Data. Google AI Blog. Available online: https://ai.googleblog.com/2017/04/federated-learning-collaborative.html (accessed on 20 September 2021).
9. Asghar, A.; Abbas, A.; Khattak, H.A.; Khan, S.U. Fog Based Architecture and Load Balancing Methodology for Health Monitoring Systems. *IEEE Access* **2021**, *9*, 96189–96200. [CrossRef]
10. Andaloussi, Y.; Ouadghiri, M.; Maurel, Y.; Bonnin, J.; Chaoui, H. Access control in IoT environments: Feasible scenarios. *Procedia Comput. Sci.* **2018**, *130*, 1031–1036. [CrossRef]
11. Deebak, B.D.; Al-Turjman, F.M. Privacy-preserving in smart contracts using blockchain and artificial intelligence for cyber risk measurements. *J. Inf. Secur. Appl.* **2021**, *58*, 102749.
12. Gray, C. Storj vs. Dropbox: Why Decentralized Storage Is the Future. 2014. Available online: https://bitcoinmagazine.com/articles/storjvs-Dropboxdecentralized (accessed on 15 September 2021).
13. Šarac, M.; Pavlović, N.; Bacanin, N.; Al-Turjman, F.; Adamović, S. Increasing privacy and security by integrating a Blockchain Secure Interface into an IoT Device Security Gateway Architecture. *Energy Rep.* **2021**, *78*, 1–8.
14. Ripeanu, M. Peer-to-peer architecture case study: Gnutella network. In Proceedings of the First International Conference on Peer-to-Peer Computing, Linkoping, Sweden, 27–29 August 2001; pp. 99–100.
15. Tseng, H.; Zhao, Q.; Zhou, Y.; Gahagan, M.; Swanson, S. Morpheus: Creating application objects efficiently for heterogeneous computing. *ACM SIGARCH Comput. Archit. News* **2016**, *44*, 53–65. [CrossRef]
16. Giesler, M.; Pohlmann, M. *The Anthropology of File Sharing: Consuming Napster as a Gift*; Association for Consumer Research, University of Minnesota Duluth: Duluth, MN, USA, 2003; Volume 3.
17. Good, N.; Krekelberg, A. Usability and privacy: A study of Kazaa P2P file-sharing. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Ft. Lauderdale, FL, USA, 5 April 2003; pp. 137–144.
18. Pouwelse, J.; Garbacki, P.; Epema, D.; Sips, H. The bittorrent p2p file-sharing system: Measurements and analysis. In *International Workshop on Peer-to-Peer Systems*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 205–216.
19. Queiroz, M.; Telles, R.; Bonilla, S. Blockchain and supply chain management integration: A systematic review of the literature. *Supply Chain Manag Int. J.* **2019**. *25*, 241–254. [CrossRef]

20. Rehiman, K.; Veni, S. A trust management model for sensor enabled mobile devices in iot. In Proceedings of the 2017 International Conference on I-SMAC (IoT in Social; Analytics and Cloud), Palladam, India, 10–11 February 2017; pp. 807–810.
21. Yuan, J.; Li, X. A reliable and lightweight trust computing mechanism for IoT edge devices based on multi-source feedback information fusion. *IEEE Access* **2018**, *6*, 23626–23638. [CrossRef]
22. Shahid, H.; Shah, M.A.; Almogren, A.; Khattak, H.A.; Din, I.U.; Kumar, N.; Maple, C. Machine Learning-based Mist Computing Enabled Internet of Battlefield Things. *ACM Trans. Internet Technol. (TOIT)* **2021**, *21*, 1–26. [CrossRef]
23. Ouaddah, A. *A Blockchain Based Access Control Framework for the Security and Privacy of IoT with Strong Anonymity Unlinkability and Intractability Guarantees*, 1st ed.; Elsevier Inc.: London, UK, 2018; Volume 115.
24. Zhang, C.; Zhu, L.; Xu, C.; Sharif, K.; Du, X.; Guizani, M. LPTD: Achieving lightweight and privacy-preserving truth discovery in CIoT. *Futur. Gener. Comput. Syst.* **2019**, *90*, 175–184. [CrossRef]
25. Atzori, L.; Iera, A.; Morabito, G. Siot: Giving a social structure to the internet of things. *IEEE Commun. Lett.* **2011**, *15*, 1193–1195. [CrossRef]
26. Baldassarre, G.; Giudice, P.; Musarella, L.; Ursino, D. The MIoT paradigm: Main features and an ad-hoc crawler. *Futur. Gener. Comput. Syst.* **2019**, *92*, 29–42. [CrossRef]
27. Baldassarre, G.; Giudice, P.; Musarella, L.; Ursino, D. A paradigm for the cooperation of objects belonging to different IoTs. In Proceedings of the 22nd International Database Engineering & Applications Symposium, Villa San Giovanni, Italy, 18 June 2018; pp. 157–164.
28. Liu, J.; Xiao, Y.; Chen, C. Authentication and access control in the internet of things. In Proceedings of the 2012 32nd International Conference on Distributed Computing Systems Workshops, Macau, China, 18–21 June 2012; pp. 588–592.
29. Aloqaily, M.; Otoum, S.; Ridhawi, I.; Jararweh, Y. An intrusion detection system for connected vehicles in smart cities. *Ad Hoc Netw.* **2019**, *90*, 101842. [CrossRef]
30. Otoum, S.; Kantarci, B.; Mouftah, H. On the feasibility of deep learning in sensor network intrusion detection. *IEEE Netw. Lett.* **2019**, *1*, 68–71. [CrossRef]
31. Al-Turjman, F.; Zahmatkesh, H.; Shahroze, R. An overview of security and privacy in smart cities' IoT communications. *Trans. Emerg. Telecommun. Technol.* **2019**, *1*, e3677. [CrossRef]
32. Nawaz, A.; Ahmed, S.; Khattak, H.A.; Akre, V.; Rajan, A.; Khan, Z.A. Latest Advances in Interent Of Things and Big Data with Requirments and Taxonomy. In Proceedings of the 2020 Seventh International Conference on Information Technology Trends (ITT), Abu Dhabi, United Arab Emirates, 25–26 November 2020; pp. 13–19.
33. Henna, S.; Davy, A.; Khattak, H.A.; Minhas, A.A. An Internet of Things (IoT)-Based Coverage Monitoring for Mission Critical Regions. In Proceedings of the 2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS), Canary Islands, Spain, 24–26 June 2019; pp. 1–5.
34. Tewari, A.; Gupta, B. Security, privacy and trust of different layers in Internet-of-Things (IoTs) framework. *Futur. Gener. Comput. Syst.* **2020**, *108*, 909–920. [CrossRef]
35. Rault, T.; Bouabdallah, A.; Challal, Y. Energy efficiency in wireless sensor networks: A top-down survey. *Comput. Netw.* **2014**, *67*, 104–122. [CrossRef]
36. Islam, S.; Kwak, D.; Kabir, M.; Hossain, M.; Kwak, K. The internet of things for health care: A comprehensive survey. *IEEE Access* **2015**, *3*, 678–708. [CrossRef]
37. Lee, I.; Lee, K. The Internet of Things (IoT): Applications, investments, and challenges for enterprises. *Bus. Horiz.* **2015**, *58*, 431–440. [CrossRef]
38. Zhang, Z.K.; Cho, M.; Wang, C.W.; Hsu, C.W.; Chen, C.K.; Shieh, S. IoT security: Ongoing challenges and research opportunities. In Proceedings of the 2014 IEEE 7th International Conference on Service-Oriented Computing and Applications, Matsue, Japan, 17–19 November 2014; pp. 230–234.
39. Baccelli, E.; Hahm, O.; Günes, M.; Wählisch, M.; Schmidt, T. RIOT OS: Towards an OS for the Internet of Things. In Proceedings of the 2013 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Turin, Italy, 14–19 April 2013; pp. 79–80.
40. Dunkels, A.; Gronvall, B.; Voigt, T.; IEEE.Abomhara, M.; Køien, G. Contiki-a lightweight and flexible operating system for tiny networked sensors. In Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks, Tampa, FL, USA, 16–18 November 2004; pp. 455–462.
41. Alromaihi, S.; Elmedany, W.; Balakrishna, C. Cyber security challenges of deploying IoT in smart cities for healthcare applications. In Proceedings of the 2018 6th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW), Barcelona, Spain, 6–8 August 2018; pp. 140–145.
42. Alsaadi, E.; Tubaishat, A. Internet of things: Features, challenges, and vulnerabilities. *Int. J. Adv. Comput. Sci. Inf. Technol.* **2015**, *4*, 1–13.
43. Abomhara, M.; Køien, G.M. Security and privacy in the Internet of Things: Current status and open issues. In Proceedings of the 2014 International Conference on Privacy and Security in Mobile Systems (PRISMS), Aalborg, Denmark, 11–14 May 2014; pp. 1–8.
44. Al-Turjman, F.; Baali, I. Machine learning for wearable IoT-based applications: A survey. In *Transactions on Emerging Telecommunications Technologies*; Bernabe, J., Hernández, J., Moreno, M., Gomez, A., Eds.; Willey: Hoboken, NJ, USA, 2019.
45. Roman, R.; Zhou, J.; Lopez, J. On the features and challenges of security and privacy in distributed internet of things. *Comput. Netw.* **2013**, *57*, 2266–2279. [CrossRef]

46. Cirani, S. A scalable and self-configuring architecture for service discovery in the internet of things. *IEEE Internet Things J.* **2014**, *1*, 508–521. [CrossRef]
47. Tsai, C.W.; Lai, C.F.; Vasilakos, A. Future internet of things: Open issues and challenges. *Wirel. Netw.* **2014**, *20*, 2201–2217. [CrossRef]
48. Antonakakis, M.; April, T.; Bailey, M.; Bernhard, M.; Bursztein, E.; Cochran, J.; Durumeric, Z.; Halderman, J.A.; Invernizzi, L.; Kallitsis, M.; et al. Understanding the mirai botnet. In Proceedings of the 26th {USENIX} Security Symposium ({USENIX} Security 17), Vancouver, BC, Canada, 16–18 August 2017; pp. 1093–1110.
49. Khattak, H.A.; Tehreem, K.; Almogren, A.; Ameer, Z.; Din, I.U.; Adnan, M. Dynamic pricing in industrial internet of things: Blockchain application for energy management in smart cities. *J. Inf. Secur. Appl.* **2020**, *55*, 102615. [CrossRef]
50. Kan, L.; Wei, Y.; Muhammad, A.; Siyuan, W.; Linchao, G.; Kai, H. A multiple blockchains architecture on inter-blockchain communication. In Proceedings of the 2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C), Lisbon, Portugal, 16–20 July 2018; pp. 139–145.
51. Lee, W.M. Using the metamask chrome extension. In *Beginning Ethereum Smart Contracts Programming*; Apress: Berkeley, CA, USA, 2019; pp. 93–126.
52. Taş, R.; Tanrıöver, Ö.Ö. Building a decentralized application on the Ethereum blockchain. In Proceedings of the 2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), Ankara, Turkey, 11–13 October 2019; pp. 1–4.
53. Benet, J. Ipfs-content addressed, versioned, p2p file system. *arXiv* **2014**, arXiv:1407.3561.
54. Zichichi, M.; Ferretti, S.; D'Angelo, G. A distributed ledger based infrastructure for smart transportation system and social good. In Proceedings of the 2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 10–13 January 2020; pp. 1–6.
55. Li, R.; Song, T.; Mei, B.; Li, H.; Cheng, X.; Sun, L. Blockchain for Large-Scale Internet of Things Data Storage and Protection. *IEEE Trans. Serv. Comput.* **2019**, *12*, 762–771. [CrossRef]
56. Wang, S.; Zhang, Y.; Zhang, Y. A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems. *IEEE Access* **2018**, *6*, 38437–38450. [CrossRef]
57. Ding, S.; Cao, J.; Li, C.; Fan, K.; Li, H. A Novel Attribute-Based Access Control Scheme Using Blockchain for IoT. *IEEE Access* **2019**, *7*, 38431–38441. [CrossRef]
58. Wang, H.; He, D.; Shen, J.; Zheng, Z.; Zhao, C.; Zhao, M. Verifiable outsourced ciphertext-policy attribute-based encryption in cloud computing. *Soft Comput.* **2017**, *21*, 7325–7335. [CrossRef]
59. Fernández, F.; Alonso, A.; Marco, L.; Salvachúa, J. A model to enable application-scoped access control as a service for IoT using OAuth 2.0. In Proceedings of the 2017 20th Conference on Innovations in Clouds, Internet and Networks (ICIN), Paris, France, 7–9 March 2017; pp. 322–324.
60. Hummen, R.; Shafagh, H.; Raza, S.; Voig, T.; Wehrle, K. Delegation-based Authentication and Authorization for the IP-based Internet of Things. In Proceedings of the 2014 Eleventh Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), Singapore, 30 June–3 July 2014; pp. 284–292.
61. Gusmeroli, S.; Piccione, S.; Rotondi, D. A capability-based security approach to manage access control in the internet of things. *Math. Comput. Model* **2013**, *58*, 1189–1205. [CrossRef]
62. Biswas, K.; Muthukkumarasamy, V. Securing smart cities using blockchain technology. In Proceedings of the 2016 IEEE 18th international conference on high performance computing and communications; IEEE 14th international conference on smart city; IEEE 2nd international conference on data science and systems (HPCC/SmartCity/DSS, Sydney, NSW, Australia, 12–14 December 2016; pp. 1392–1393.
63. Rehan, M.; Rehmani, M. *Blockchain-Enabled Fog and Edge Computing: Concepts, Architectures and Applications: Concepts, Architectures and Applications*; CRC Press: Boca Raton, FL, USA, 2020.
64. Yu, K.P.; Tan, L.; Aloqaily, M.; Yang, H.; Jararweh, Y. Blockchain-enhanced data sharing with traceable and direct revocation in IIoT. *IEEE Trans. Ind. Inform.* **2021**, *17*, 7669–7678. [CrossRef]
65. Šimunić, S. *Upotreba Blockchain Tehnologije za Registraciju i Upravljanje IoT Uređajima*; Department of Computer, Faculty of Engineering, University of Rijeka: Rijeka, Croatia, 2018.
66. Do, H.; Ng, W. Blockchain-Based System for Secure Data Storage with Private Keyword Search. In Proceedings of the 2017 IEEE World Congress on Services (SERVICES), Honolulu, HI, USA, 25–30 June 2017.
67. Zhang, G.; Li, T.; Li, Y.; Hui, P.; Jin, D. Blockchain-Based Data Sharing System for AI-Powered Network Operations. *J. Commun. Inf. Netw.* **2018**, *3*, 1–8. [CrossRef]
68. Steichen, M.; Fiz, B.; Norvill, R.; Shbair, W.; State, R. Blockchain-Based, Decentralized Access Control for IPFS. In Proceedings of the 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Halifax, NS, Canada, 30 July–3 August 2018; pp. 1499–1506.
69. Sifah, E. Chain-based big data access control infrastructure. *J. Supercomput.* **2018**, *74*, 4945–4964. [CrossRef]
70. Zhang, Y.; Kasahara, S.; Shen, Y.; Jiang, X.; Wan, J. Smart contract-based access control for the internet of things. *IEEE Internet Things J.* **2019**, *6*, 1594–1605. [CrossRef]

71. Nguyen, K.; Oualha, N.; Laurent, M. Securely outsourcing the ciphertext-policy attribute-based encryption. *World Wide Web* **2018**, *21*, 169–183. [CrossRef]
72. Oualha, N.; Nguyen, K. Lightweight attribute-based encryption for the internet of things. In Proceedings of the 2016 25th International Conference on Computer Communication and Networks (ICCCN), Waikoloa HI, USA, 1–4 August 2016; pp. 1–6.
73. Hur, J.; Kang, K. Secure data retrieval for decentralized disruption-tolerant military networks. *IEEE/ACM Trans. Netw.* **2012**, *22*, 16–26. [CrossRef]
74. Bethencourt, J.; Sahai, A.; Waters, B. Ciphertext-policy attribute-based encryption. In Proceedings of the 2007 IEEE Symposium on Security and Privacy (SP'07), Berkeley, CA, USA, 20–23 May 2007.
75. Nishide, T.; Yoneyama, K.; Ohta, K. Attribute-based encryption with partially hidden encryptor-specified access structures. In Proceedings of the International Conference on Applied Cryptography and Network Security, New York, NY, USA, 3–6 June 2008; pp. 111–129.
76. Khan, F.; Li, H.; Zhang, L.; Shen, J. An expressive hidden access policy CP-ABE. In Proceedings of the 2017 IEEE Second International Conference on Data Science in Cyberspace (DSC), Shenzhen, China, 26–29 June 2017; pp. 178–186.
77. Zhou, Z.; Huang, D.; Wang, Z. Efficient privacy-preserving ciphertext-policy attribute based-encryption and broadcast encryption. *IEEE Trans. Comput.* **2013**, *64*, 126–138. [CrossRef]
78. Phuong, T.; Yang, G.; Susilo, W. Hidden ciphertext policy attribute-based encryption under standard assumptions. *IEEE Trans. Inf. Forensics Secur.* **2015**, *11*, 35–45. [CrossRef]
79. Hammi, M.; Bellot, P.; Serhrouchni, A. BCTrust: A decentralized authentication blockchain-based mechanism. In Proceedings of the 2018 IEEE Wireless Communications and Networking Conference (WCNC), Barcelona, Spain, 15–18 April 2018; pp. 1–6.
80. Ganache. Trufflesuite. Ganache ONE CLICK BLOCKCHAIN SOLUTION. Available online: https://www.trufflesuite.com/ganache (accessed on 15 September 2021).