Computer Systems
Science and Engineering

Tech Science Press

DOI: 10.32604/csse.2023.037099
Article

A Lightweight Approach (BL-DAC) to Secure Storage Sharing in Cloud-IoT Environments

Zakariae Dlimi*, Abdellah Ezzati and Said Ben Alla

Hassan First University of Settat, Faculté Sciences et Technique, LAVETE, Settat, 26000, Morocco
*Corresponding Author: Zakariae Dlimi. Email: zakariae.dlimi@gmail.com
Received: 23 October 2022; Accepted: 02 February 2023; Published: 26 May 2023

Abstract: The growing advent of the Internet of Things (IoT) users is driving the adoption of cloud computing technologies. The integration of IoT in the cloud enables storage and computational capabilities for IoT users. However, security has been one of the main concerns of cloud-integrated IoT. Existing work attempts to address the security concerns of cloud-integrated IoT through authentication, access control, and blockchain-based methods. However, existing frameworks are somewhat limited by scalability, privacy, and centralized structures. To mitigate the existing problems, we propose a blockchain-based distributed access control method for secure storage in the IoT cloud (BL-DAC). Initially, the BL-DAC performs decentralized authentication using the Quantum Neural Network Cryptography (QNNC) algorithm. IoT users and edge nodes are authenticated in the blockchain deployed by distributed Trusted Authorities (TAs) using multiple credentials. The user data is classified into sensitive and non-sensitive categories using the Enhanced Seagull Optimization (ESO) algorithm. Also, the authentication to access this data is performed by a decentralized access control method using smart contract policy. Sensitive user data is encrypted using the QNNC algorithm and stored in the private cloud. In contrast, non-sensitive data is stored in the public cloud, and IPFS is used to store data in a decentralized manner with high reliability. In addition, data security is improved by using a hierarchical blockchain which improves scalability by managing the multiple blockchains hierarchically and is lightweight using Proof of Authentication Consensus (PoAH). The BL-DAC is simulated and validated using the Network Simulator-3.26 simulation tool and validated. This work shows better results than the compared ones in terms of validation metrics such as throughput (26%), encryption time (19%), decryption time (16%), response time (15%), block validation time (31%), attack detection rate (16%), access control precision (13%), and scalability (28%).

Keywords: Internet of Things; blockchain; access control; secure storage

1 Introduction

The Internet of Things (IoT) is an emerging technology that collects and shares data from various applications, such as home appliances and smart devices [1]. However, resource limitation for IoT devices is driving the adoption of cloud computing technology [2]. Cloud storage is a solution to store digital information using multiple servers at multiple locations, which provides reliable data management services by third-party service providers with minimal user intervention [3]. IoT and cloud integration are mainly used to store IoT data in cloud storage. However, integration faces many challenges regarding heterogeneity, security, and privacy [4]. The access control process is applied in several works to allow access only to authorized users to perform services in a centralized manner using roles, attributes, and policies [5,6]. On the other hand, centralization of access control leads to problems such as One Point Failure and difficult scalability. Decentralized access control is implemented based on user attributes to overcome these problems. However, attribute-based access control is not sufficient to provide effective security, and this is done by limiting only user-specific attributes and leveraging user roles and services [7]. User legitimacy validation allows the admission of authorized users and ignores manipulated devices based on several parameters, such as username and password. In several works, a single trusted authority is implemented to validate the authenticity of users. However, authentication based on a single authority leads to high complexity and authentication time [8]. A huge amount of data is extracted from IoT devices and stored in cloud storage, but this increases privacy and security issues due to the sensitivity of the data, which is mainly correlated to users. Hence, the existing work proposed the introduction of blockchain-based lightweight authentication to address security and privacy issues [9,10]. Some data may include user's personal information such as health records, medical records, and bank details. The leakage of such sensitive data negatively impacts the privacy of an individual. Secure data storage is performed to improve data integrity to overcome these problems. Multiple IoT devices share the cloud server to store and protect their data in a centralized storage system [11]. Adopting a centralized storage system leads to scalability and Single Point of Failure issues. Various approaches are made in several works for secure storage sharing. Blockchain is used to protect the storage sharing system by performing easy access and secure storage in a decentralized manner with its unbreakable and immutable nature [12,13]. In particular, it increases security by detecting several attacks, such as DDoS and man-in-the-middle attacks. However, it increases scalability and time complexity issues due to its linear structure [14]. Centralized cloud storage increases security threats and is overcome by implementing the Interplanetary File System (IPFS) to store and share data in a decentralized (i.e., distributed) manner to increase the security of storage sharing [15]. To overcome issues such as overhead storage and scalability in blockchain IoT environment, the authors in [16] propose a Multi-Zone-Wise Blockchain that is also used in attack detection use case, and the authors in [17] design a data security storage referring to Fabric project pattern. In [18], authors designed a method that secures the transmission of IoT data in the cloud, using a secured channel and an optimized deep machine learning model.

1.1 Research Contribution

The scalable and privacy-preserving method is proposed in this research to secure the cloud-unified IoT environment. Some of the major contributions of this research are provided below:

- The user's legitimacy is ensured by providing authenticity to him; in which BL-DAC adopts Quantum Neural Network Cryptography (QNNC) by converting the information into quantum states, which leads to resistance against eavesdropping attacks.

- BL-DAC categorizes user data into sensitive and non-sensitive using Enhanced Seagull Optimization (ESO) algorithm to ease the access control method. This categorization improves access control precision by reducing control overhead.
- Adopting a hierarchical blockchain for storing the transactions ensures the system's soundness and scalability by managing the multiple blockchains at the hierarchy levels. Further, the Proof of Authentication consensus (PoAH) is utilized to reduce the block validation time and enhance security.

The simulation results of BL-DAC are given in terms of validation metrics such as throughput (15.2 Mbps), encryption time (9.75 s), decryption time (10.25 s), response time (16.6 ms), block validation time (36.8 s), attack detection rate (22.75), access control precision (76.5%), and scalability (57.8%).

1.2 Paper Organization

This manuscript is ordered as follows. Section 2 provides a literature survey of the existing works. Section 3 provides the problem statement. Section 4 explains BL-DAC in detail with suitable diagrams, equations, and pseudocodes. Section 5 illustrates the experimental results in which simulation setup, comparative analysis, security analysis, and research summary are elaborated. Section 6 gives the conclusion and future direction of BL-DAC.

2 Literature Survey

2.1 Data Sharing and Secure Storage Systems

The authors [19] proposed a new secure storage and sharing method using blockchain. This model comprises four processes: registration, storage, secure sharing, and processing. The original records are encrypted and stored outside the blockchain, and the encrypted data is stored in the blockchain to improve security. The sharing process consists of four sub-processes: initialization, verification, transmission, and secure storage. This work has the limitations of low scalability, and a Single Point of Failure as the registration and the query management are centralized. In [20], users' logistical data were stored securely using the blockchain. Users register their identities through a trusted authority, and their data is encrypted by the AES algorithm. Then the authenticated IoT devices upload their data to the data store. Here, the original records are stored in the interplanetary file systems, and the hash values are stored in the data string. After completing the secure storage process, this work performs update and maintenance. The authors of [21] propose a secure data-sharing method using a decentralized blockchain in IoT. The proposed work includes three processes: authentication, data encryption, and recovery. Initially, all users register their information for authentication. After authentication is completed, data encryption is performed by the SALSA20 algorithm. Finally, data recovery is performed using the DenFT-based index algorithm, and then the secret key is exchanged using the ReDH algorithm. The limitation of this work was that, as it uses the SALSA 20 algorithm, the encryption takes a long time, resulting in high latency. In addition, it is more difficult to implement, which leads to high complexity. PUF-based authentication using blockchain in IoT was proposed by the authors [22]. Initially, all devices register their information with the registry center, which provides the secret key using the ECC algorithm. Then, the smart contract is executed between the registration center and the IoT node mapping correlation using the blockchain. During authentication, the IoT nodes are verified based on the registered entities, and here the mutual authentication is performed based on the session key. If the IoT node is compromised, the proposed work initiates revocation and re-registration by generating a new public key. The ECC algorithm is used for secret key generation in

this job, which takes a long time to generate the secret key, results in high latency during authentication, and reduces system performance which was the main limitation of this work. The authors in [23] have done secure storage and sharing of cloud servers. This work includes two entities such as users and cloud service providers. Here, authentication and access control are performed by the cloud service providers. In authentication, the ECDSA algorithm is proposed for key generation. Here, access control is performed based on two operations: read and write. The writing process includes two sub-processes: file sharing and file storage. The reading process is only responsible for data sharing. The authors [24] proposed an efficient and secure data transfer to the cloud server. This work consists of three steps: three-factor authentication, user data compression, and data transfer by encryption. The three-factor authentication step contains three sub-steps: registration, login, and verification. Only authenticated user data is compressed in the user data compression step using the Huffman compliments algorithm. Finally, the compressed data is encrypted in the data transfer stage by encryption using a modified elliptic curve cryptography algorithm. This work is limited to overload and single-point-of-failure attacks because it has a centralized controller for authentication.

2.2 Access Control Schemes

The authors [25] proposed a role-based access control using the concept of authorization workflow. The proposed work includes three entities: conceptual view, policy, and proof of concept. The access control includes task instance constraints to confirm the responsibility and governance of access. Here, XACML-based policies are used for dynamic access control, and access control rules are used to decide the access control. The proposed work is evaluated based on six access control scenarios to prove that the proposed work has high security. The limitation of this work was poor security, as the legitimacy of IoT nodes was not considered. In [26], the authors proposed that the security and scalability of IoT should be improved by using access control methods. This work includes the configuration phase, the initialization and registration phase of the smart device, and the authentication phase of the key establishment. In the configuration phase, public and private keys are calculated by smart device gateways using elliptic curve cryptography. In the initialization and registration phase, smart devices are registered with the gateway by their attributes. To join the network and communicate with it, the registered smart devices are authenticated by a gateway to mitigate various attacks. The access control was achieved by the logical BAN protocol. The authors [27] proposed an access control method for the cloud environment by adopting the blockchain method. This work comprises blockchain, data holder, data user and storage server. First, the data holder calls on the storage server regarding the data, and the storage server acquires and encrypts the data. Meanwhile, the storage server stores the data in the blockchain network via a smart contract. Finally, the data user tries to access the data using the blockchain network. The blockchain network provides the data to the user only when the user has specified access rights otherwise revoked from the network. Also, the authors in [28] proposed a new Multi-layer secure architecture based on a fog model using an elliptical curve algorithm. This model follows three steps to upload files: deduplicating data, dividing information into parts and finally encoding and saving parts in the three allowed zone, which are local, fog server, and cloud server. User privacy was ensured by providing access control to the distributed IoT device based on blockchain technology [29]. Here, the Domain Management Server publishes all the meta information and permission in the blockchain. Four processes were designed for authorization, revocation, access control, and audit. The main idea in this paper is that all users in the network maintain a copy of the whole blockchain ledger, which lets them verify transaction data at any time they need. The limitation of this work is that the blockchain structure used is not scalable for the IoT environment. In order to protect data sharing in the cloud environment, the study in [30]

suggests an enhanced version of ciphertext-policy attribute-based encryption. The suggested method is extremely dependable and secure, as evidenced by the comparison with the current scheme.

3 Problem Statement

3.1 Specific Existing Problems

This section discusses in detail some of the problems encountered by existing works.

The authors' work [31] proposed data access and storage sharing with an effective authentication method for cyber-physical systems (IA-DAS). The entities involved in the proposed work were cloud servers, data handlers, and data holders who perform secure processes such as mutual authentication and encryption method to secure their data. The problems posed by this work were:

- Here, authentication was performed between the entities and the authentication system, which was effective. However, the authentication system was a single entity that could easily be compromised by attackers, leading to a single point of failure and security threats.
- This work provides centralized access control to data handlers with authenticated metrics. However, the trust of the handler has yet to be considered, which can lead to high-security threats.

To solve the above-mentioned problems, the author [32] enables the attribute-based access control method using smart contracts in the blockchain for IoT (ABC-BLS). This work consists of entities such as IoT nodes, a public blockchain, and a consortium blockchain. Access control to authenticated users is provided by smart contracts in which four contracts are involved, namely access control contract, service control contract, user control contract, and policy control contract. Similar smart contract-based access control methods for secure storage have been carried out by the authors [33,34], in which only authenticated entities were allowed to access the downloaded data. Indeed, access control for the above-mentioned works was provided through smart contracts.

The main limitations of this existing works are outlined below in Table 1.

Table 1: Summary of related works

Paper	Contribution	Limitation
[31]	Securing cyber-physical systems in terms of data access and storage sharing by an efficient authentication method	• The authenticated system was a single entity that could easily be compromised by attackers that leading to a single point of failure and security threats. • The manipulator trust was not considered to lead to high-security threats.
[32]	Designing effective access control methods based on attributes by using smart contracts in blockchain for IoT	• The registration entity was insecure, which led to the tampering of data. • The public blockchain is not suitable for real-time scenarios which face low scalability, high energy consumption, and time complexity issues. • The IoT nodes' legitimacy is not evaluated during access control.

(Continued)

Table 1: Continued

Paper	Contribution	Limitation
[33]	Introducing secure data sharing by blockchain-based access control methods to ensure the secrecy of the private data	• The secrecy in the secure sharing of data was affected while not considering the IoT manipulator's legitimacy. • The time taken for encryption and decryption of data was high, leading to time complexity issues. • The distance between the data holder and the blockchain network leads to latency, thereby affecting communication reliability.
[34]	Designing blockchain-based methodology to secure sensitive medical records	• Privacy is affected by allowing third-party administrators to access sensitive patient medical records, leading to spoofing attacks. • This method faces increased security threats as the legitimacy of the entity has not been ensured, resulting in the leakage of sensitive data.

3.2 Research Aim

Secure storage sharing in the Cloud IoT environment has encountered many challenges regarding authenticity, access control, and privacy. The state-of-the-artwork focuses on the significant problems of secure storage sharing, but the complete solution has yet to be given. The main limitations encountered are summarized in the following:

- Centralized access control
- Inefficient blockchain structure
- Lack of confidentiality

The main objective of this research is to increase security and privacy when sharing storage using blockchain in the IoT environment. The other objectives of this research are listed below:

- Increase the security of the IoT environment
- Minimizing security threats
- Improve security when sharing storage
- Reduce energy consumption and block validation time

4 Proposed Blockchain-Based Distributed Access Control Method (BL-DAC)

4.1 System Model and Design Goals

This research focuses on realizing secure storage sharing to provide high security in the IoT environment. This proposed method consists of several layers: the physical layer consists of several types of IoT devices (i.e., IoT users), the edge layer consists of edge nodes, and the cloud layer consists of cloud storage. Furthermore, hierarchical blockchain and IPFS are implemented in this research to

reduce link failures and increase security, robustness, and data privacy. Fig. 1 represents the system model of BL-DAC in detail.

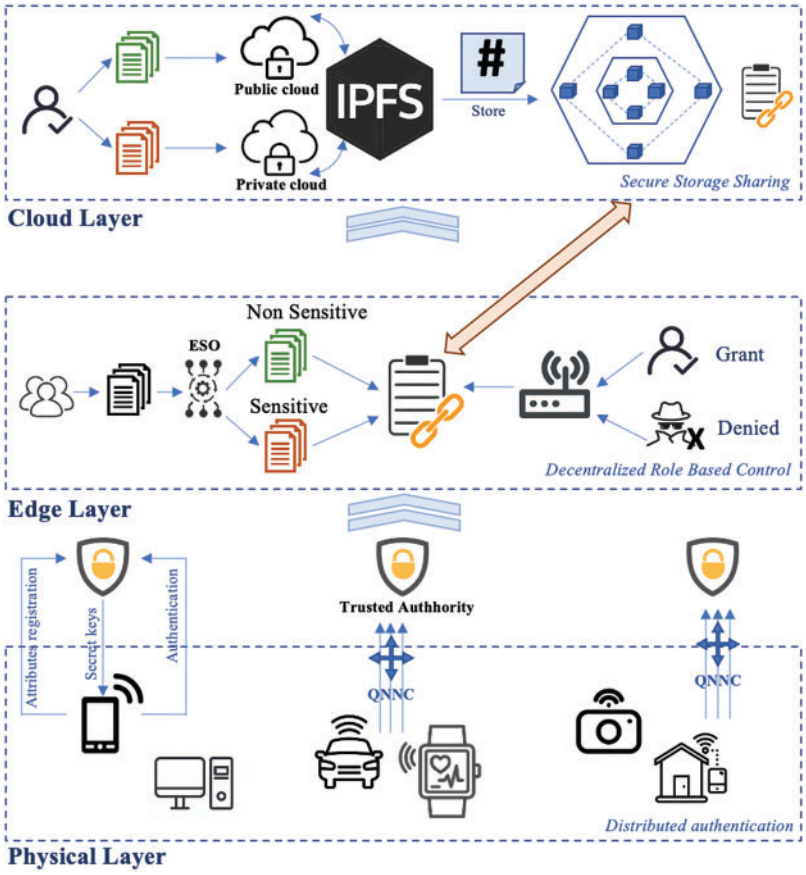

Figure 1: Proposed BL-DAC system model

The entities of BL-DAC are described below:

- **Physical layer**: The physical layer consists of IoT users who intend to store data in the cloud in a secure manner. IoT devices can be used in any location, which can be mobile phones, computers, or cameras.
- **Trusted Authority (TA)**: TAs are deployed by the blockchain whose responsibility is to provide authenticity to IoT users and edge nodes by acquiring their credentials and providing them with secret keys.
- **Edge layer**: The edge layer is composed of edge nodes that are responsible for dividing data into sensitive and non-sensitive data and also have the responsibility of providing access control to IoT devices.
- **Cloud layer**: The cloud layer consists of a public and private cloud to store sensitive and non-sensitive data, respectively. In addition, the cloud layer is juxtaposed with the hierarchical blockchain and IPFS to reduce security issues in the network.
- **Hierarchical blockchain**: Hierarchical blockchains are responsible for storing the transaction securely, in which multiple blockchains manage their transaction data hierarchically without affecting the semantics of the transactions.

In this research, our main objective is to protect the secure data of cloud users through access control and blockchain methods. The objectives of our design to achieve this goal are illustrated below:

- **Adoption of massive IoT users:** for the cloud environment, many IoT devices benefit from using the cloud in terms of storage and computing infrastructure. BL-DAC needs to be adopted to support the massive number of IoT devices in terms of QoS.
- **Security of user data:** As a large number of devices are connected to the network, security is one of the major concerns. User data can be classified into sensitive and non-sensitive data. Some users hold non-sensitive data where security is not required, while some users hold sensitive data which is considered their digital assets. Therefore, the user's sensitive data must be secured.
- **High efficiency:** user-acquired data must be managed as reliably as possible in terms of timely storage, link quality, robust logic, and high scalability.

4.2 Distributed Authentication

Initially, all IoT devices and edge nodes are registered with the distributed TAs by providing their attributes in which the TAs are deployed in a distributed manner to reduce the single point of failure. For IoT devices, the TAs acquire device attributes such as biometrics, ID, and password and edge attributes such as ID, PUF, and location. Once registration is complete, the TAs generate a secret key based on these attributes using quantum neural network cryptography (QNNC), which optimizes the neural network parameters based on the weights and stores the attributes based on the quantum states to improve security.

Then, these attributes are stored in a lightweight blockchain in a hashed manner to increase attribute security by mitigating DDoS attacks. During authentication, TAs verify the authenticity of IoT devices and edge nodes by comparing the registered credentials with the current timestamp credentials. The processes involved in registration and authentication are illustrated below:

(i) Registration phase

Step 1—The IoT devices ($\forall$) and the edge nodes ($\exists$) submit their credentials to TA for registration, which can be formulated as follows:

$$\forall(Bio, ID, PUF, role, ser) \rightarrow TA \tag{1}$$

$$\exists(ID, PUF, Loc) \rightarrow TA \tag{2}$$

where *Bio* indicates the biometric, *ID* indicates the identity of the device, *PUF*, *role*, and *ser* indicates the non-cloneable physical function, and *Loc* indicates the location.

Step 2—After obtaining the credentials of the $\forall$ *and* $\exists$, TA performs encryption and generates secret keys. The acquired credentials (C) for both $\forall$ *and* $\exists$ are encrypted according to the isomorphism $\langle Z(C)|0\rangle \leftrightarrow \hat{a}$, in which the entries are mapped to $\hat{a}$: $\left\{\hat{a}_{(1)}, \hat{a}_{(2)}, \ldots, \hat{a}_{(c)}\right\}$. From this mapping, the amount of encryption required to enter the vector c can be defined as $\left\lceil \frac{Le(C)}{c} \right\rceil$, and Le represents the credential length. The simplified encryption process can be formulated as follows:

$$\hat{b}_{(k)} = \left\{\otimes_{k=1}^{c'} \epsilon\left(\vartheta_k\right) \otimes_{k=1}^{c'} Ds\left(\mathscr{J}_k\right) \otimes_{k=1}^{c'} \propto_2 (\Delta_k) \otimes_{k=1}^{c'} q(s_k) \otimes_{k=1}^{c'} \propto_1 (\Delta_k)\right\}^n \hat{a}_{(k)} \tag{3}$$

where $\epsilon\ (\vartheta_k)$ denotes the non-gaussian gates, $Ds\left(\mathscr{J}_k\right)$ denotes the local movement, $\propto_1\ (\Delta_k)\ and\ \propto_2 (\Delta_k)$ denote the 1st and 2nd rotation, respectively, and $q\ (s_k)$ denotes the squeeze gates. From the above encryption, the neural network output $\left\langle \hat{b} \right\rangle$ can be formulated as:

$$\left\langle \hat{b} \right\rangle = \left(\epsilon\ (a)\ , \hat{b}_{(k)}\epsilon\ (a)\right) = \left\langle \epsilon\ (a)\ |\hat{b}|\epsilon\ (a) \right\rangle \tag{4}$$

The error function $(er_{(k)})$ from the above equation can be formulated as,

$$er_{(k)} = \hat{a}_{(k)} - \left\langle \epsilon\left(\hat{a}_{(k)}\right)|\hat{b}_{(k)}|\epsilon\left(\hat{a}_{(k)}\right)\right\rangle \tag{5}$$

The process of encryption can be simplified by computing the above equations in which the input C is given as quantum modes to QNN $\hat{a}_{(k)}$, which are processed to give the output of $er_{(k)}$. Eventually, the hidden layer $\widehat{hi}_{(k)}$ acts as an authentication code for the given messages (Au_c) and also gives the output, which is combined to produce the cipher text as $EnC(\widehat{hi}_{(k)}, er_{(k)})$. The Au_c is stored in TA and also provided as a secret key to the $\forall\ and\ \exists$, which can be formulated as,

$$TA\ sK(Au_c) \rightarrow \{\forall, \exists\} \tag{6}$$

(ii) Authentication phase

Step 1—After successful registration, $\forall\ and\ \exists$ are authenticated with their secret keys, which can be represented as:

$$\forall(sK_{t+1}(Au_c)) \rightarrow TA \tag{7}$$

$$\exists(sK_{t+1}(Au_c)) \rightarrow TA \tag{8}$$

Step 2—After obtaining the authentication response, the TA retrieves the already stored credentials from the blockchain and then verifies the current timestamp credentials of the $\forall\ and\ \exists$ by decrypting them to obtain the plain text, which can be presented as:

$$TA \rightarrow DeC(sK_{t+1}(Au_c)) \tag{9}$$

Here the $sK_{t+1}(Au_c)$ is composed of the credentials such as $Bio, ID, PUF, andLoc$ of both entities.

Step 3—When decrypting $sK_{t+1}(Au_c)$, the TA compares it with previously stored identification information, which can be formulated as follows:

$$TA \rightarrow sK\ (Au_c) = sK_{t+1}(Au_c) \tag{10}$$

Step 4—If the two credentials match, then authenticity is assured, or else revoked from the network. This can be formulated as:

$$TA = \begin{cases} sK\ (Au_c) = sK_{t+1}\ (Au_c), "Y" \\ sK\ (Au_c) \neq sK_{t+1}\ (Au_c), "N" \end{cases} \tag{11}$$

where "Y" and "N" denote yes and no, respectively, in which yes represents the authenticity and no represents the revocation. Fig. 2 shows the sequencing of the registration and authentication flows of BL-DAC.

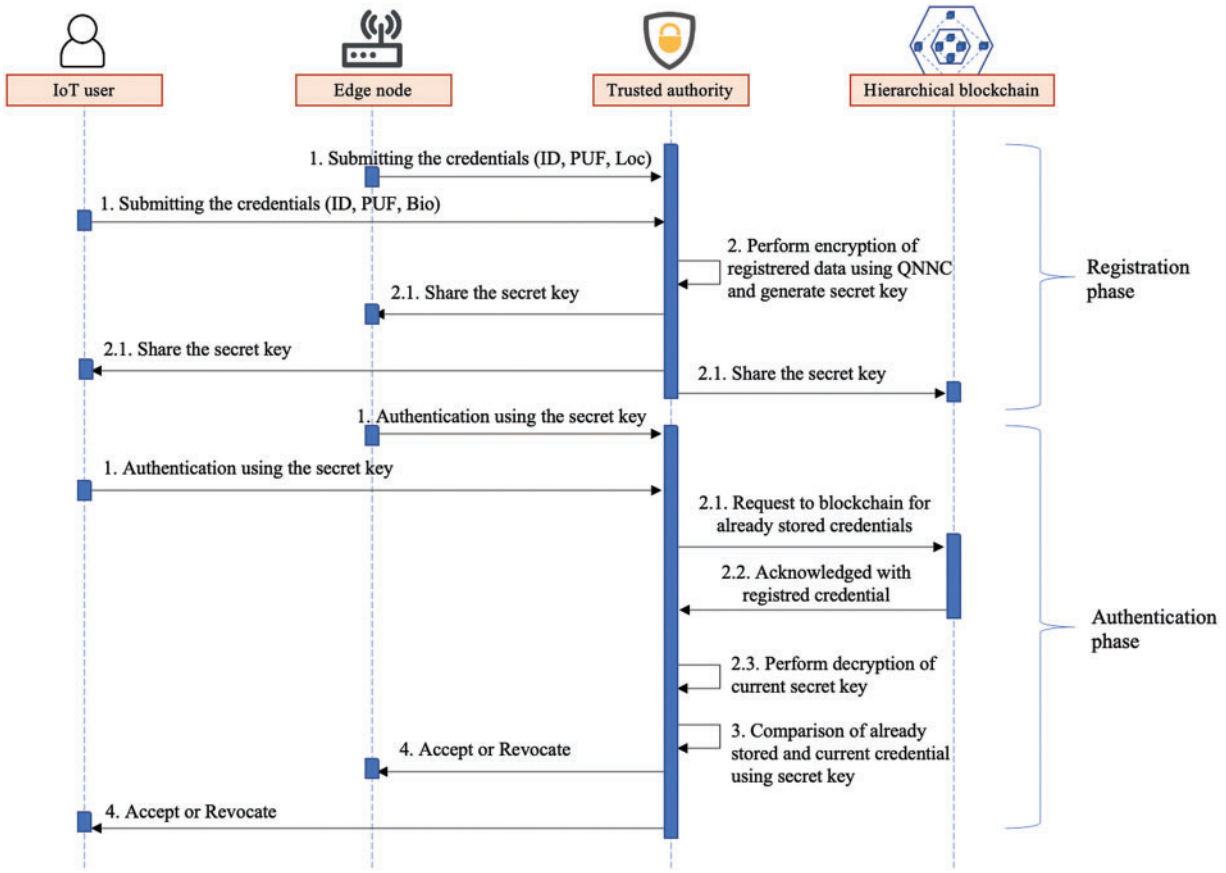

Figure 2: BL-DAC authentication flow

4.3 Decentralized Role-Based Access Control

After successful authentication, the corresponding data of the particular device is classified into two main categories $\exists$ such as sensitive data and non-sensitive data, using the Enhanced Seagull Optimization (ESO) algorithm, which achieves high convergence through its adaptive mechanism. The reason for dividing the data into sensitive and non-sensitive data facilitates access control by improving access control accuracy and reducing the difficulty during access control.

Inspired by the behavior of seagulls, this work adopts ESO for classifying data into sensitive and non-sensitive as an optimization problem. Here enhanced seagull represents by incorporating intelligent behavior to avoid local minima trap. Here, the $\exists$ acts as a seagull and $\forall$ are their prey. Suppose that, $PoP_{\exists}$ denotes the population of edge nodes, and the dimension of the problem space as Θ with the position of $\exists$ can be presented as $\exists_i = \left(\exists_i^1, \exists_i^2, \ldots, \exists_i^{\Theta}\right)$ in which $i = 1, 2, \ldots, PoP_{\exists}$. After defining the initial values, the fitness value of each $\exists$ can be calculated and compared to each other $\exists$ to find the global optimal values. Once the fitness value is calculated, the $\exists$ updates its best position to avoid data interference by traffic $\forall$.

A recall value R is adopted to avoid data interference, which can be formulated as follows:

$$M\,(ite) = R \times ip\,(ite) \tag{12}$$

where $ip(ite)$ represents the initial position ∃, R is the support variable that ensures the interference of data from ∃ in the search space, and $M\ (ite)$ represents the updated position of ∃ after performing the data inference avoidance. The R can be formulated as,

$$R = F_q - \left(ite \times \left(F_q / maxiite\right)\right) \tag{13}$$

From the above equation, R will be tuned linearly, in which the value F_q decreases linearly to zero and *maxiite* denotes the maximum iteration.

The new best position of the ∃ without data interference can be formulated as follows,

$$Bs\,(ite) = |M\,(ite) + P\,(ite)| \tag{14}$$

In the above equation, $P\,(ite)$ is the previous updated position, and $Bs\,(ite)$ represents the best position without obtaining the data inference.

Once the positions of the ∃ are updated, user data classification takes place according to their sensitivity and non-sensitivity information. Sensitive information includes users' personal information, such as medical or health data, genetic data, and biometric data, and non-sensitive information includes public data, such as users' entertainment content. On this basis, the ∃ performs a spiral action in the search space using factors such as $T, W, and\ O$, respectively.

The equations below provide the behavior of the classification:

$$T = g \times \cos(\theta) \tag{15}$$

$$W = g \times \sin(\theta) \tag{16}$$

$$O = g \times \theta \tag{17}$$

$$g = \gamma \times e^{\theta v} \tag{18}$$

where g is the radius of each turn of the spiral corresponding to the classification of ∃, $v\ and\ \gamma$ denote the definition constants of the shape of the balance spring, and θ denotes the angle ranging from 0 to 2π.

The optimal classification of ∃ without trapping in local minima in an intelligent way can be formulated as follows:

$$ip(ite) = Bs(ite) \times T \times W \times O + best(ite) + (best(ite) - Bs(ite) \times \varphi) \tag{19}$$

From the above equation, the sensitive and non-sensitive data can be calculated as follows:

$$\exists\,(best\,(ite)) = \begin{cases} \forall_{sensitive} \\ \forall_{nonsensitive} \end{cases} \tag{20}$$

The following pseudocode shows the ESO classification process in detail

Pseudocode: ESO-Based Packet Classification
Input: Population of ∃ ($PoP_{\exists}$)
Output: best position ∃ ($best\,(ite)$)
Start
Set $PoP_{\exists}, \Theta, \exists_i, R$

(Continued)

Pseudocode: Continued

Set $T, W, O, \gamma, e^{\theta v}$
While $(ite < maxiite)$ **do**
Calculate the fitness value
Traffic interference avoidance using (13)
Update the position $Bs\,(ite)$ using (14)
Compute the classification behavior using (15)–(18)
Perform intelligent classification using (19)
Get $\exists\,(best\,(ite))$ using (20)
End
End

At the same time, the hierarchical blockchain (*HBL*) provides the smart contracts to the ∃ blockchain by defining specific policies based on the role, service types, and trust values that the TA acquired upon registration, which can be designated as follows:

$$HBL(SMC[\circledR, \check{\mathrm{T}}, tr]) \rightarrow \exists \tag{21}$$

where *SMC* denotes the smart contract that points to the specified set of predefined legitimate contracts in which ®, ť, *and tr* represent the role, service type, and trust values, respectively.

Here, the user's trust values are analyzed based on threshold generation. The trust values are obtained as the direct and indirect trust of the same users and corresponding neighbors, respectively. The direct trust is given by the same user itself, which is represented by (dir_{tr}), and the indirect trust is collected from the neighbors based on their behavior which can be represented by $(idir_{tr})$. The threshold value (*Thr*) is defined as :

$$\forall = \begin{cases} Accessed, (dir_{tr}, idir_{tr}) \leq Thr \\ Denied, (dir_{tr}, idir_{tr}) \geq Thr \end{cases} \tag{22}$$

In addition to the smart contract policy and categorized data, the edge node provides distributed access control to devices in the network. The authorization or denial of access to cloud storage is made based on the digital signature provided by the smart contract, user trust values, data sensitivity, and the edge of the edge layer. Fig. 3 represents the decentralized access control in detail:

4.4 Secure Storage Sharing

After providing access to users, the data is encrypted using the quantum cryptography algorithm and stored in the interplanetary file system (IPFS) of hybrid cloud storage at the cloud layer. IPFS is mainly used to store and share data in a decentralized manner, increasing overall communication security. Both sensitive and non-sensitive user content is encrypted. Adopting a hybrid cloud aims to improve scalability, workload optimization, and security. Sensitive data is stored in the private cloud, and non-sensitive data is stored in the public cloud. Hash values of encrypted data are stored in the blockchain to prevent data tampering and increase data integrity, thus mitigating man-in-the-middle attacks. A hierarchical blockchain is implemented in this research to store transactions in a hierarchical and lightweight structure.

The proposed hierarchical blockchain consists of the features that are presented below:

- There are *N* hierarchical levels in BL-DAC in which, for each level, several blockchains are managed
- Each blockchain at each level maintains local transactions in its ledger

- The top level of the hierarchical blockchain provides a limited view (i.e., it retains some of its transactional records) of the lower-level blockchain

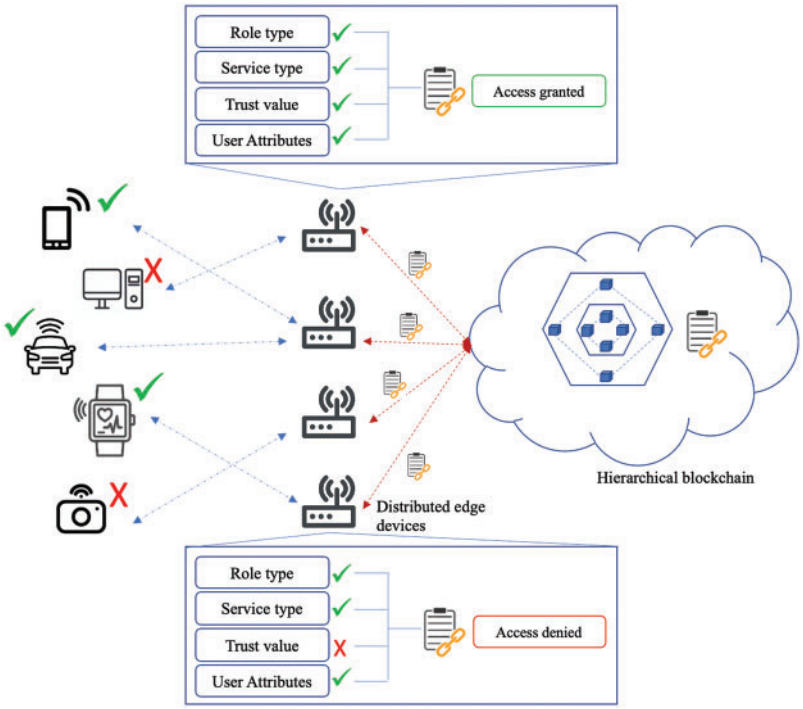

Figure 3: Edge-based decentralized access control

In addition, the abstract interpretation method is adopted to improve the system's robustness from semantic and scalability perspectives. BL-DAC applies abstract interpretations for each blockchain level at consistent time intervals.

Formation of hierarchical blockchain and validation of transactions

The BL_u^{Hi} is the blockchain network that is managed by the u^{th} peer networks at the first ranking level of Hi, in which $u = 1, 2, \ldots, p_{Hi}$. The peer network at the first hierarchical level is formed as $P2P^1 = \left\{P2P_1^1, P2P_2^1, \ldots, P2P_{p_1}^1\right\}$. For each hierarchical level, a level leader is selected who is responsible for forming the next peer network $\left(\text{i.e., } P2P^2 = \left\{P2P_1^2, P2P_2^2, \ldots, P2P_{p_2}^2\right\}\right)$. Similarly, multiple levels are formed in this manner.

The level heads in the multi-level blockchain are responsible for the trade-off between the regularity and the soundness of the data. Finally, the block transactions are validated to ensure security.

For this, BL-DAC adopts a secure consensus mechanism named Proof of Authentication (PoAh). This consensus is used in the proposed blockchain model to reduce the time consumption for transactions and block validation which makes the blockchain lightweight and also reduces the energy consumption of the IoT network.

Pseudocode: Pseudocode: PoAh consensus
Input: all participating blocks, secret keys
Output: Block validation
(i) The transactions are grouped
(ii) Participants → Tac
(iii) Signed with its own secret key (QNNC)
(iv) Broadcaster → Signed block distributed
(v) Trusted Elected node → signed block
[Secret Key] [New Block] → check MAC
(vi) **If** new block == Auth, **Then**
Add [Block||PoAH] → Broadcast
Hash (new block) → Add to the existing chain
(vii) **Else**
Reject the block
(ix) **Go to** the first step until the iteration is completed

The above pseudocode is explained by these steps:

- Initially, the components of the IoT network are responsible for generating the transactions (*Tac*). The candidates then distribute the generated blocks for evaluation.
- Applicants are responsible for generating keys to secure their blocks. The QNNC algorithm is adopted for the key generation that generates a secret key and signs the block.
- The preferred trusted node, which has a considerable trust value in the network, is selected to evaluate the block to validate new transactions. For each successful validation, the trust value of the nodes increases.
- The evaluation of the new block validated by the privileged node is done by validating the secret key of the new node for the first round and the message authentication code for the second round. Once the evaluation is complete, the new block is broadcast and added as a block to the blockchain. The above pseudocode provides the PoAH workflow. In this way, transactions are stored in the blockchain securely and reliably.

5 Experimental Results

5.1 Simulation Setup

BL-DAC (Blockchain Decentralized Access Control) is simulated using the Network Simulator-3.26 (NS-3.26). The use of the NS-3.26 simulator provides a reliable network environment perfectly suited for the proposed work. The system configuration must be respected to simulate BL-DAC. The hardware configuration of random-access memory (RAM) and 4 GB, the hard disk is 500 GB capacity, and the processor is Intel (R) Core (TM) i5-4590S CPU @ 3.00 GHz. The software specification in which the operating system (OS) is used is Ubuntu 14.04 LTS, and the simulation tool used is NS-3.26. The parameters initialized and used in BL-DAC are provided in Table 2 below.

Table 2: Simulation parameters

Parameter	Value
Number of IoT devices	70
Number of edge nodes	5
Number of trusted authority TA	1
Number of cloud servers	1
Simulation area	1000 ×1000
Package size	512 bytes
Simulation time	250 s
Data transmission rate	2 Mb
Type of traffic	Transmission Control Protocol (TCP)
Media Access Control (MAC) protocol	IEEE 802.11p
Packet transmission rate	3 packages/s

5.2 Comparative Analysis

In this subsection, the performance of the BL-DAC is compared with the existing works, which are IA-DAS [31] and ABC-BLS [32], in terms of validation metrics such as throughput, encryption time, decryption time, response time, block validation time, transaction time and attack detection rate.

5.2.1 Comparison of Throughput

Throughput (Tr_p) is defined as the number of packets successfully transmitted by the sender in a given time which can be formulated as follows:

$$\left(Tr_p\right) = \frac{Suc_p}{T} \tag{23}$$

where Suc_p indicates the successfully transmitted packet and T indicates the time taken for transmission. Fig. 4 represents the throughput comparison of the proposed BL-DAC and the existing works IA-DAS and ABC-BLS. It shows that as the number of users increases, the throughput also increases. In this, BL-DAC achieves high throughput among the existing works due to the increase in the transmission rate of successful packets because it adopts a distributed authentication method. Here, TAs are distributed in the network and provide authenticity to entities. The proposed authentication method reduces security threats by validating users with their biometrics, IDs, roles, services, and PUFs, and edge nodes with their IDs, PUFs, and locations using the QNNC algorithm.

For high throughput, reducing security threats implies a considerable reduction in lost packets. On the other hand, existing work IA-DAS and ABC-DAC also perform authentication. Still, the authentication metrics needed to be improved, and at the same time, the authenticator needed to be more secure, which led to a high packet loss rate and reduced throughput.

The numerical results in Fig. 4 show that when the number of users increases to 70, the throughput of BL-DAC also increases to 24 MB. On the contrary, the throughput decreased for the existing jobs ABC-BLS for 20 MB and IA-DAS for 17 MB for the same number of users. Therefore, the numerical results show that BL-DAC reaches 3–7 MB more than the existing jobs.

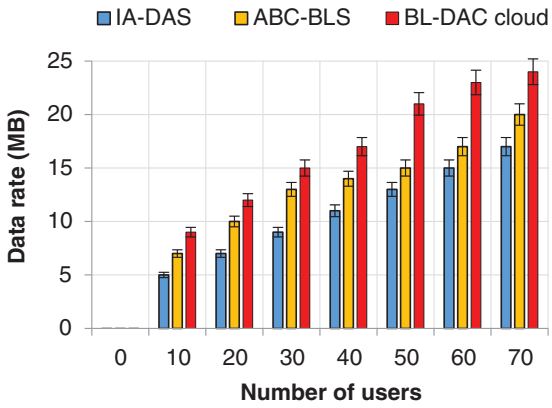

Figure 4: Number of users *vs*. throughput

5.2.2 Comparison of Encryption and Decryption Time

The encryption time (en_T) is defined as the time needed to encrypt the user's data into plain text. It can be formulated as follows:

$$en_T = \frac{P_{enT}}{tot_T} \tag{24}$$

where P_{enT} represents the actual encryption time taken for the packet while tot_T represents the total encryption time. Similarly, the decryption time (dec_T) is defined as the time needed to decrypt the user's ciphertext into plain text, which can be formulated as follows:

$$dec_T = \frac{P_{decT}}{tot_T} \tag{25}$$

Here P_{decT} indicates the actual packet decryption time for the encrypted packet.

From Fig. 5, the encryption time increases significantly when the number of users in the environment increases. From there, BL-DAC achieves less encryption time. Similarly, in Fig. 6, the decryption time also decreases significantly, and BL-DAC achieves less decryption time than the existing work. The reason why BL-DAC performs less encryption and decryption time is due to the adoption of the QNNC algorithm. The QNNC algorithm converts the user's credentials into quantum nodes and encrypts them using a neural network. Using the neural network reduces the time required to generate random numbers for the raw text and improves the scalar multiplication speed, which reduces the encryption time. Similarly, for the decryption time, the neural network easily decrypts the cipher text by its intelligent behavior in less decryption time. On the contrary, existing works IA-DAS and BL-DAC are limited by adopting conventional cryptographic algorithms, which result in high time for encryption and decryption (i.e., it takes more time for random number generation for the raw text).

From the numerical results of the encryption time, when the number of users increases to 70, BL-DAC achieves a lower encryption time of 19 s, while in the case of the existing works ABC-DAS and IA-DAS, the time required for encryption is high, estimated to be 22 and 28 s respectively. BL-DAC achieves (3–9) s less than the existing work.

When the number of users increases to 70, BL-DAC achieves a lower decryption time of 17 s, while in the case of the existing jobs ABC-DAS and IA-DAS, the decryption time is high at the 20 and 25 s, respectively. BL-DAC achieves (3–8) s than the existing work.

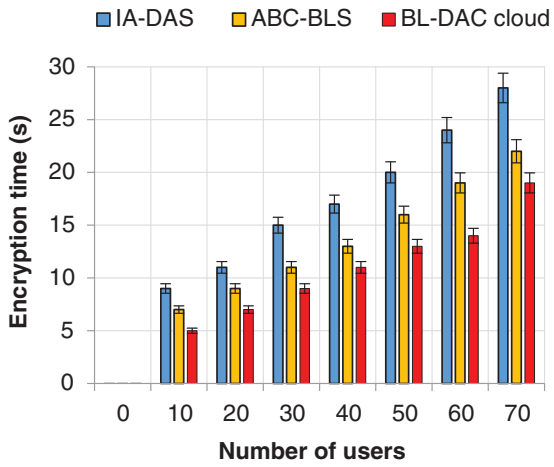

Figure 5: Number of users *vs*. encryption time

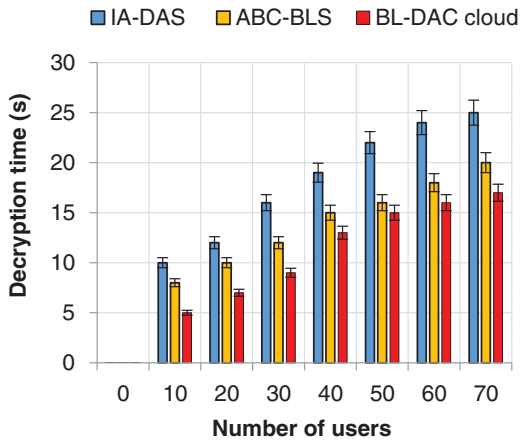

Figure 6: Number of users *vs*. decryption time

5.2.3 Comparison of Response Time

The calculation of the time taken between the beginning and the end of the process is called response time (Rs_T). The response time can be formulated as follows:

$$Rs_T = \frac{T_{Rs}}{Tot_{Rs}} \tag{26}$$

where T_{Rs} indicates the time taken to answer and Tot_{Rs} indicates the total number of responses.

The graphical plot in Fig. 7 shows that when the number of users in the network increases, the response time also increases. In this finding, the response time of BL-DAC is lower than the existing works, even with the increase in the number of users. The main reason for such a decrease is that BL-DAC uses distributed authentication and role-based distributed access control, respectively. In distributed authentication, the ATs are deployed in a distributed manner by the blockchain, which results in a high packet success rate and also avoids a Single Point of Failure, which reduces the response time. In the same case, for distributed role-based access control, the edge nodes provide access to the user, which also reduces the response time. On the other hand, for the case of existing IA-DAS and

ABC-BLS work, the entities are deployed centrally, which results in a high response time for a user's data to complete the process and also leads to a Single Point of Failure.

From the numerical results, it is shown that BL-DAC achieves a lower response time of 30 ms when the number of users increases to 70, while the existing works ABC-BLS and IA-DAS achieve a high response time of 34 and 37 ms, respectively for the same number of users. Finally, BL-DAC achieves 4 to 7 ms less than the existing work.

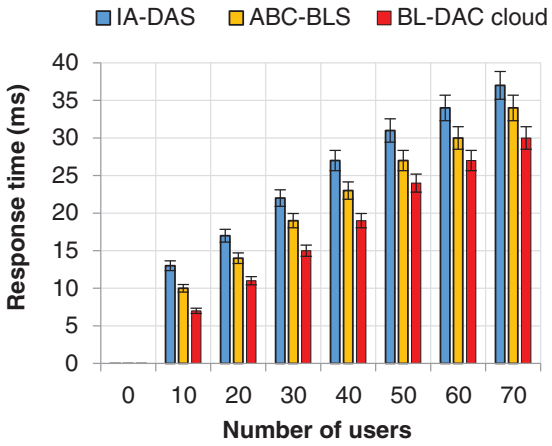

Figure 7: Number of users *vs*. response time

5.2.4 Comparison of Block Validation Time

Block validation time is defined as the time taken by miners to validate the block in order to add it as a new block in the blockchain. As a general rule, a good blockchain system should have less block validation time.

Fig. 8 shows that as the number of user attributes increases, the block validation time also increases. From there, BL-DAC achieves less block validation time. The reason for acquiring less block validation time is due to the adoption of a hierarchical blockchain with PoAH consensus. Since there are multiple hierarchical levels in the blockchain, user attributes are easily managed. Finally, PoAH consensus ensures the security constraints of miners and also has less block validation time than existing work-based models. In comparison, the existing ABC-BLS adopts the public blockchain with conventional consensus, which leads to scalability issues and less block validation time, respectively.

The numerical results show that when the number of user attributes is increased to 40, BL-DAC achieves a lower block validation time of the 70 s, while the existing works ABC-BLS and IA-DAS achieve high block validation of 80 and 95 s, respectively. The overall numerical results show that BL-DAC achieves a lower block validation time of (10–25) s than the existing works.

5.2.5 Comparison of Attack Detection Rate

The attack detection rate (Ad_{rt}) is defined as the ratio of true positives to the sum of the true positive and false negative rates. The attack detection rate can be formulated as follows:

$$Ad_{rt} = \frac{TruP}{TruP + FalN} \tag{27}$$

where $TruP$ is the true positive rate, and $FalP$ is the false positive rate.

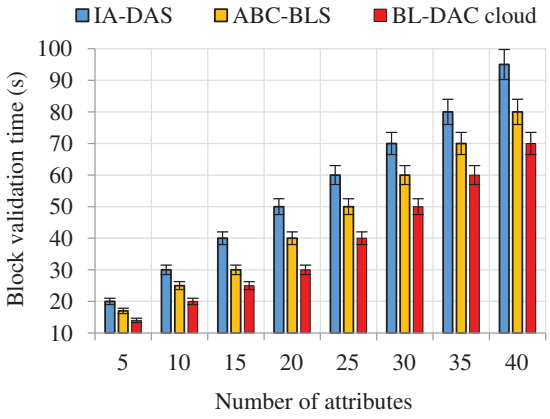

Figure 8: Number of attributes *vs*. block validation time

From Fig. 9, it is shown that as the number of transactions increases, the attack detection rate also increases. From this graphical plot, it is deduced that BL-DAC achieves a high attack detection rate even if the number of transactions increases. The reason for such an increase is the proposed methods, such as distributed authentication and distributed role-based access control.

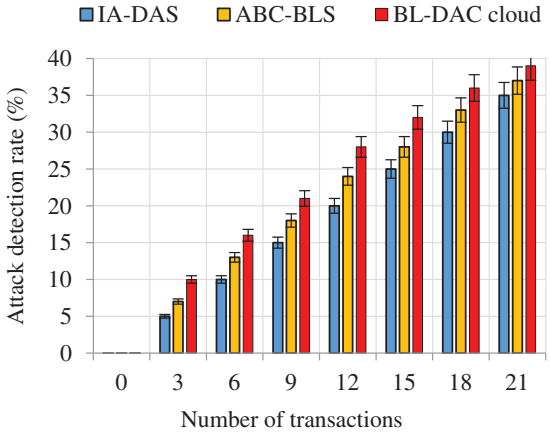

Figure 9: Number of users *vs*. attack detection rate

Distributed authentication was performed by distributed ATs using the QNNC algorithm that detects malicious users that mimic normal users. On the other hand, edge-based distributed access control using smart contracts based on user roles, trust, and service type reduces the conspiracy and in-house attacks, respectively. In addition, user data is stored in an encrypted manner, which reduces tampering attacks in the network. On the other hand, the existing work IA-DAS and ABC-BLS focus only on in-house attacks by providing centralized authentication and access control, respectively. Centralization leads to a Single Point of Failure, which results in poor user data management, and reduces the detection rate of attacks.

The numerical results show that BL-DAC achieves an increased attack detection rate of 39 when the number of transactions increases to 21. On the contrary, the existing works IA-DAS and ABC-BLS achieve a low attack detection rate of 35 and 37, respectively. The final results show that BL-DAC achieves (2–5) more than the existing work.

5.2.6 Comparison of the Access Control Method

The accuracy of access control (Ac_{pr}) is defined as the accuracy with which the user accesses the network. The formulation of access control accuracy is calculated as follows:

$$Ac_{pr} = \frac{TruP}{TruP + FalP} \tag{28}$$

where $TruP$ is the true positive rate, and $FalN$ is the false negative rate.

The graph in Fig. 10 compares the proposed access control method with the existing ones. It deduced that the access control accuracy increases when the number of users increases. BL-DAC achieves higher access control accuracy due to the user task categorization and the distributed role-based access control method, respectively. First, the user data is categorized into sensitive and non-sensitive using the ESO optimization algorithm. Then, with these categorized data, access control is provided in a distributed manner by the edge nodes using a smart contract policy. The smart contract policy includes user trust values, roles, and permissions. Categorizing user data and adopting a distributed access control method reduces complexity and control overhead, thus improving access control accuracy. In contrast, the existing ABC-BLS method also performs smart contract-based access control. However, it only relies on user attributes, and IA-DAS provides access to the user based on authenticity centrally, resulting in high control overhead, thus reducing the accuracy of access control for existing jobs.

The numerical results in the graph show that as the number of users increases to a maximum of 50, the access control accuracy is 98%. In comparison, the existing ABC-BLS and IA-DAS works achieve 89% and 73% control rates, respectively, for the same number of users. Overall, BL-DAC achieves (9–25)% more than the existing works.

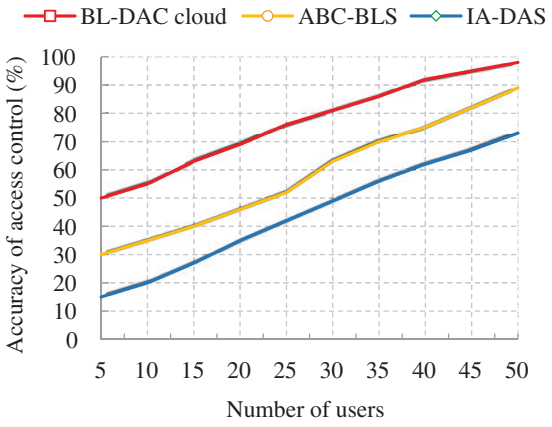

Figure 10: Number of users *vs*. scalability

5.2.7 Comparison of the Scalability of the Blockchain

Scalability is the amount of data the blockchain system can tolerate to process the data. Scalability is directly proportional to system performance. As scalability increases, the system's performance increases, and vice versa. Scalability is calculated in terms of resource utilization, throughput, and cost.

The graph in Fig. 11 shows the scalability comparison of the proposed blockchain model's with existing ones. From this result, it is inferred that the proposed working blockchain model achieves high scalability compared to the existing working blockchain model. The reason why BL-DAC achieves so

much scalability is the adoption of a lightweight hierarchical blockchain. In the proposed hierarchical blockchain model, multiple blockchains are managed hierarchically. Since our environment is large-scale, managing and processing the user's transaction at multiple levels increases the scalability of the proposed blockchain. Moreover, the adoption of PoAH consensus makes the blockchain lightweight, which reduces the cost constraints. On the other hand, the existing BL-DAC method adopts the conventional blockchain structure, which limits its performance in a large-scale environment in terms of data management, leading to poor scalability.

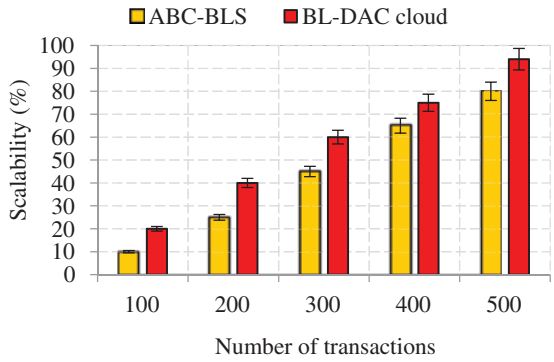

Figure 11: Number of users *vs*. scalability

The numerical results show that even if the number of transactions increases to 500, the proposed blockchain model achieves high scalability of 94%. In comparison, the existing BL-DAC blockchain model achieves low scalability of 80% for the same number of transactions. The final results show that BL-DAC achieves 14% higher scalability than the existing work.

5.3 Security Analysis

This subsection explains the security analysis of BL-DAC. BL-DAC ensures user and data security by providing a secure and reliable framework. According to the security analysis below, BL-DAC is resistant to several attacks in the cloud-unified IoT environment.

Internal malicious attacks: The illegitimate user who gains successful access to the network imposes several threats to the network by modifying legitimate access and false injection. In our work, the internal malicious users are considered IoT users. BL-DAC mitigates this attack by providing decentralized role-based access for the user and also performs encryption in the cloud storage that is also resistant to eavesdropping. Therefore, the internal malicious person trying to gain access will not be able to modify the data.

Impersonation attacks: The illegitimate user manipulates the legitimate user's credentials and removes useful information from the network. The proposed framework considers IoT users and edge nodes as impersonators. BL-DAC mitigates this risk by ensuring the authenticity of both entities based on their credentials via TA. Here, TAs are deployed through the blockchain, increasing the security level.

Conspiracy attacks: In this type of attack, one or more malicious users simulate legitimate users and suppress their access. Here, the conspirators are considered as the users who try to deceive normal users with unprivileged techniques. BL-DAC performs decentralized role-based access control using smart contract policies to get rid of them. The blockchain provides the proposed smart contracts, including user trust.

51% attacks: This type of attack has been observed in the blockchain in which malicious blockchain miners who take over 50% of the role of network validators will exploit the blockchain. Here, the user is adopted as a miner, which can lead to the exploitation of the blockchain. BL-DAC resists such attacks by verifying the authenticity of the miner using PoAH consensus. Any new block mined by the miners must be convinced that the consensus will be allowed as a block in the blockchain network.

5.4 Research Summary

In this subsection, the numerical performance of the proposed BL-DAC work is summarized. The experimental results show that BL-DAC performs better in metrics such as throughput, encryption and decryption time, block validation time, response time, attack detection rate, access control accuracy, and blockchain scalability. Table 3 describes the overall comparison of BL-DAC and the existing works in terms of average results.

Table 3: Average numerical results of BL-DAC and the existing works

Performance metrics		BL-DAS cloud	ABC-BLS	IA-DAS
Number of users	Throughput	15.12	12	9.6
	Encryption time	9.75	12.12	15.5
	Decryption time	10.25	12.3	16
	Response time	16.6	19.65	22.62
	Access control precision	76.5	58.2	44.6
Number of attributes	Block validation time	38.62	46.2	55.5
Number of transactions	Attack detection rate	22.75	20	17.5
	Scalability	57.8	45	–

The principles of the research are presented below:

- For improving security, distributed authentication is performed for both users and edge servers using quantum neural network cryptography (QNNC) by storing attributes in the blockchain, which increases security.
- For improving privacy and security, this paper classified legitimate data into two classes such as sensitive and non-sensitive, using the optimization algorithm (ESO) based on the data classification edge server providing access control by executing a smart contract.
- For providing security when sharing storage, this paper encrypts and store data in IPFS on a hybrid cloud service that provides storage and sharing in a decentralized manner.
- For providing scalability and reducing transaction time, this paper proposed a lightweight hierarchical blockchain in which multiple blockchains are managed, and block validation is performed by PoAh consensus, which reduces validation time.

6 Conclusion

The problems in cloud-integrated IoT storage sharing, such as centralized authentication, access control, and poor scalability, are solved by proposing the BL-DAC method. For data privacy, BL-DAC uses a hybrid cloud structure with IPFS in which sensitive and non-sensitive data are stored in public and private IPFS clouds, respectively, in which sensitive data is encrypted using the QNNC algorithm.

For providing data security, data is stored as transactions in the hierarchical blockchain using PoAH consensus, which improves scalability and reduces block validation time. The simulation of BL-DAC is carried out using the NS-3 simulation tool, and the results are validated with performance metrics such as throughput, encryption and decryption time, response time, block validation time, attack detection rate, access control accuracy, and blockchain scalability. The simulation result reveals that BL-DAC performs better than the existing work.

The future works of this article are to adopt, in the first step, a federated learning approach in the IoT cloud for secure storage sharing that preserves user and data privacy without revealing the identity and mitigates intrusions into the environment, and as a second step is to optimize the communication model and distribute the transactions between the nodes of the blockchain to reduce further network traffic.

Acknowledgement: We show our gratitude to anonymous reviewers for their useful work.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: All authors contributed to the design and implementation of the research, to the analysis of the results, and to the writing of the manuscript.

Conflicts of Interest: The authors declare they have no conflicts of interest to report regarding the present study.

References

[1] Z. Dlimi, A. Ezzati and S. Ben Alla, "A lightweight blockchain for IoT in smart city (IoT-smartchain)," *Computers, Materials & Continua*, vol. 69, no. 2, pp. 2687–2703, 2021.

[2] I. Mohiuddin and A. Almogren, "Security challenges and strategies for the IoT in cloud computing," in *Proc. 11th ICICS*, Irbid, Jordan, pp. 367–372, 2020.

[3] T. Alam, "Cloud-based IoT applications and their roles in smart cities," *Smart Cities*, vol. 4, no. 3, pp. 1196–1219, 2021.

[4] M. Mohammed Sadeeq, N. M. Abdulkareem, S. R. M. Zeebaree, D. Mikaeel Ahmed, A. Saifullah Sami *et al.,* "IoT and cloud computing issues, challenges and opportunities: A review," *Qubahan Academic Journal*, vol. 1, no. 2, pp. 1–7, 2021.

[5] D. Gupta, S. Bhatt, M. Gupta, O. Kayode and A. S. Tosun, "Access control model for google cloud IoT," in *Proc. IEEE 6th ICBDSC*, Baltimore, MD, USA, pp. 198–208, 2020.

[6] S. Ravidas, A. Lekidis, F. Paci and N. Zannone, "Access control in Internet-of-Things: A survey," *Journal of Network and Computer Applications*, vol. 144, no. 4, pp. 79–101, 2019. https://dx.doi.org/10.1016/j.jnca.2019.06.017

[7] Q. Yang, M. Zhang, Y. Zhou, T. Wang, Z. Xia *et al.,* "A non-interactive attribute-based access control scheme by blockchain for IoT," *Electronics*, vol. 10, no. 15, pp. 1855–1866, 2021.

[8] J. Koo, S. -R. Oh, S. H. Lee and Y. -G. Kim, "Security architecture for cloud-based command and control system in IoT environment," *Applied Sciences*, vol. 10, no. 3, pp. 1035–1052, 2020.

[9] A. Y. F. Alsahlani and A. Popa, "LMAAS-IoT: Lightweight multi-factor authentication and authorization scheme for real-time data access in IoT cloud-based environment," *Journal of Network and Computer Applications*, vol. 192, no. 1871, pp. 103177, 2021. https://dx.doi.org/10.1016/j.jnca.2021.103177

[10] M. Tahir, M. Sardaraz, S. Muhammad and M. Saud Khan, "A lightweight authentication and authorization framework for blockchain-enabled IoT network in health-informatics," *Sustainability*, vol. 12, no. 17, pp. 6960, 2020.

[11] M. K. I. Nirjhor, M. A. Yousuf and M. S. Mhaboob, "Electronic medical record data sharing through authentication and integrity management," in *Proc. 2nd ICREST*, Dhaka, Bangladesh, pp. 308–313, 2021.

[12] M. Shah, M. Shaikh, V. Mishra and G. Tuscano, "Decentralized cloud storage using blockchain," in *Proc. ICOEI*, Tirunelveli, India, pp. 384–389, 2020.

[13] V. -H. Hoang, E. Lehtihet and Y. Ghamri-Doudane, "Privacy-preserving blockchain-based data sharing platform for decentralized storage systems," in *Proc. IFIP Networking Conf.*, Paris, France, pp. 280–288, 2020.

[14] H. Gao, S. Luo, Z. Ma, X. Yan and Y. Xu, "BFR-SE: A blockchain-based fair and reliable searchable encryption scheme for IoT with fine-grained access control in cloud environment," *Wireless Communications and Mobile Computing*, vol. 2021, no. 4, pp. 1–21, 2021. https://dx.doi.org/10.1155/2021/5340116

[15] R. Kumar, R. Tripathi, N. Marchang, G. Srivastava, T. R. Gadekallu *et al.,* "A secured distributed detection system based on IPFS and blockchain for industrial image and video data security," *Journal of Parallel and Distributed Computing*, vol. 152, no. 2, pp. 128–143, 2021. https://dx.doi.org/10.1016/j.jpdc.2021.02.022

[16] S. Kably, T. Benbarrad, N. Alaoui and M. Arioua, "Multi-zone-wise blockchain based intrusion detection and prevention system for IoT environment," *Computers, Materials & Continua*, vol. 74, no. 1, pp. 253–278, 2022.

[17] P. Wang and W. Susilo, "Data security storage model of the Internet of Things based on blockchain," *Computer Systems Science and Engineering*, vol. 36, no. 1, pp. 213–224, 2021.

[18] J. A. Alzubi, R. Manikandan, O. A. Alzubi, I. Qiqieh, R. Rahim *et al.,* "Hashed needham schroeder industrial IoT based cost optimized deep secured data transmission in cloud," *Measurement*, vol. 150, no. 1, pp. 107077, 2020. https://dx.doi.org/10.1016/j.measurement.2019.107077

[19] Q. Qin, B. Jin and Y. Liu, "A secure storage and sharing scheme of stroke electronic medical records based on consortium blockchain," *BioMed Research International*, vol. 2021, no. 5, pp. 14, 2021. https://dx.doi.org/10.1155/2021/6676171

[20] H. Li, D. Han and M. Tang, "Logisticschain: A blockchain-based secure storage scheme for logistics data," *Mobile Information Systems*, vol. 2021, no. 1, pp. 1–15, 2021. https://dx.doi.org/10.1155/2021/8840399

[21] U. Narayanan, V. Paul and S. Joseph, "Decentralized blockchain based authentication for secure data sharing in cloud-IoT," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, no. 2, pp. 769–787, 2022. https://dx.doi.org/10.1007/s12652-021-02929-z

[22] Y. Zhang, B. Li, B. Liu, Y. Hu and H. Zheng, "A privacy-aware PUFs-based multiserver authentication protocol in cloud-edge IoT systems using blockchain," *IEEE Internet of Things Journal*, vol. 8, no. 18, pp. 13958–13974, 2021.

[23] A. Niknia, M. Correia and J. Karimpour, "Secure cloud-of-clouds storage with space-efficient secret sharing," *Journal of Information Security and Applications*, vol. 59, no. 4, pp. 102826, 2021. https://dx.doi.org/10.1016/j.jisa.2021.102826

[24] D. V. K. Vengala, D. Kavitha and A. P. S. Kumar, "Three factor authentication system with modified ECC based secured data transfer: Untrusted cloud environment," *Complex & Intelligent Systems*, vol. 1, no. 1, pp. 7, 2021. https://dx.doi.org/10.1007/s40747-021-00305-0

[25] M. Uddin, S. Islam and A. Al-Nemrat, "A dynamic access control model using authorising workflow and task-role-based access control," *IEEE Access*, vol. 7, pp. 166676–166689, 2019. https://dx.doi.org/10.1109/ACCESS.2019.2947377

[26] U. Iqbal, A. Tandon, S. Gupta, A. R. Yadav, R. Neware *et al.,* "A novel secure authentication protocol for IoT and cloud servers," *Wireless Communications and Mobile Computing*, vol. 2022, no. 1, pp. 1–17, 2022. https://dx.doi.org/10.1155/2022/7707543

[27] N. Sohrabi, X. Yi, Z. Tari and I. Khalil, "BACC: Blockchain-based access control for cloud data," in *Proc. ACSW*, Melbourne, Victoria, Australia, pp. 1–10, 2020.

[28] N. Niyaz Ahamed and N. Duraipandian, "Secured data storage using deduplication in cloud computing based on elliptic curve cryptography," *Computer Systems Science and Engineering*, vol. 41, no. 1, pp. 83–94, 2022.

[29] N. Shi, L. Tan, C. Yang, C. He, J. Xu *et al.,* "BacS: A blockchain-based access control scheme in distributed Internet of Things," *Peer-to-Peer Networking and Applications*, vol. 14, no. 5, pp. 2585–2599, 2021. https://dx.doi.org/10.1007/s12083-020-00930-5

[30] S. Xue and C. Ren, "Security protection of system sharing data with improved CP-ABE encryption algorithm under cloud computing environment," *Automatic Control and Computer Sciences*, vol. 53, no. 4, pp. 342–350, 2019.

[31] Z. Ghaffar, S. Ahmed, K. Mahmood, S. H. Islam, M. M. Hassan *et al.,* "An improved authentication scheme for remote data access and sharing over cloud storage in cyber-physical-social-systems," *IEEE Access*, vol. 8, pp. 47144–47160, 2020. https://dx.doi.org/10.1109/ACCESS.2020.2977264

[32] S. Y. A. Zaidi, M. A. Shah, H. A. Khattak, C. Maple, H. T. Rauf *et al.,* "An attribute-based access control for IoT using blockchain and smart contracts," *Sustainability*, vol. 13, no. 19, pp. 10556, 2021.

[33] H. Gao, Z. Ma, S. Luo, Y. Xu and Z. Wu, "BSSPD: A blockchain-based security sharing scheme for personal data with fine-grained access control," *Wireless Communications and Mobile Computing*, vol. 2021, no. 1, pp. 1–20, 2021. https://dx.doi.org/10.1155/2021/6658920

[34] J. Kaur, R. Rani and N. Kalra, "Blockchain-based framework for secured storage, sharing, and querying of electronic healthcare records," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 20, pp. 1, 2021. https://dx.doi.org/10.1002/cpe.6369