applied sciences

MDPI

Article

AAC-IoT: Attribute Access Control Scheme for IoT Using Lightweight Cryptography and Hyperledger Fabric Blockchain

Suhair Alshehri * and Omaimah Bamasag

Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia; obamasek@kau.edu.sa

* Correspondence: sdalshehri@kau.edu.sa

Abstract: The Internet of Things (IoT) is an integrated environment as it merges physical smart objects to the Internet via wireless technologies to share data. The global connectivity of IoT devices brings the needs to ensure security and privacy for data owners and data users. In this paper, an attribute-based access control scheme for IoT (AAC-IoT) using Hyperledger Fabric (HLF) blockchain is proposed to address the security challenges. In the AAC-IoT scheme, data owners are registered and authenticated using identities, certificates and signatures. Data users, however, are registered with identities, certificates, signatures and physical unclonable function (PUF); then a credence score is computed for users to predict the originality during authentication. For access control, attribute-based access control (ABAC) is used, and the number of attributes is selected based on the sensitivity of the data. In accordance with the attributes count, the access control policies are generated. The novel concept of attribute count is determined from a fuzzy logic method using data type and preference. Hyperledger Fabric (HLB) blockchain is presented to manage meta-data and security credentials from data owners and data users, respectively, using a lightweight hashing algorithm. The AAC-IoT model using HLF blockchain is developed with Java programming language and iFogSim simulator. The performance metrics are measured based on latency, throughput and storage overhead, and the results show better outcome than the previous research work.

Keywords: attribute based access control; authentication; Hyperledger Fabric; blockchain; Internet of Things

Citation: Alshehri, S.; Bamasag, O. AAC-IoT: Attribute Access Control Scheme for IoT Using Lightweight Cryptography and Hyperledger Fabric Blockchain. *Appl. Sci.* **2022**, *12*, 8111. https://doi.org/10.3390/app12168111

Academic Editor: Dimitris Mourtzis

Received: 30 June 2022
Accepted: 10 August 2022
Published: 12 August 2022

1. Introduction

The Internet of Things (IoT) grows with the utilization of emerging technologies. The IoT devices as sensors, actuators, CCTV cameras, etc., capture data from different environments and pass it to the backend server. This enables IoT to associate with a variety of smart applications including smart home, smart city and smart industry [1–4]. The IoT data owners will collect data from the deployed IoT devices and grant access to users based on the permission of the owner. The data from devices are collected at different sensitivity levels; hence the security provisioning is a challenging process in IoT [5–7]. In general, the devices operating in IoT are low-power constrained; thus, they are able to perform only limited computations such as monitoring, transferring data and listening to channels.

The requirement for security is significant to assure privacy of data as well as user credentials. Security is needed for the following reasons: (1) an illegitimate user enters the system using credentials of legitimate users; (2) different types of attacker devices are involved with the intention of spying and extracting sensitive data or modifying the emergency data into normal data; (3) intruders the specialize in interrupting the process of data transfer based on stealing the resources of the system; and (4) compromised users in the system exhibit normal behavior while they operate selfishly for some benefits received from the attackers.

According to the major issues in IoT, the solutions for security are cryptographic techniques, access control methods and hashing. However, there are massive numbers devices in IoT , so a distributed security environment using blockchain is presented for ensuring security [8–12].

Blockchain technology yields efficient results in privacy and security. In general, the blockchain is designed as a major combination of users, regulators, ledgers, and smart contracts. The potential of blockchain is that no one can modify the contents in each block since the values are in hashes [13–15]. Artificial Intelligence (AI) methods are also incorporated into blockchain for data protection [16]. Blockchain is able to support the processes of authenticating devices, storing data, and validating access control policies [17,18]. Access control (AC) in IoT prefers to use lightweight methods due to the low computation ability of IoT devices [19–22]. AC performs authentication and authorization based on previously defined policies. Attribute-based access control (ABAC) is one of the popular AC mechanisms followed to ensure security in the designated system. The attributes in ABAC depend on the device, its characteristics and the data it collects. For instance, the attributes of a device can be identity, name, etc., In this way, values of attributes differ from device to a device.

To improve security by ABAC, multiple attributes were taken into account, and blockchain is introduced for this process. A signature is one of the significant attributes that is unique for users, and it is verified in blockchain. In ABAC, the attributes are protected using hashing or cryptography. Since the leakage of attributes leads to access for illegitimate users, the attributes are hidden. On the other hand, the data from an owner is stored encrypted based on the data attributes. Security and privacy play a vital role in ensuring that the user's data is protected in the storage. In this paper, the provisioning of ABAC for IoT devices by using Hyperledger Fabric (HLF) blockchain is proposed. The consideration of attributes in accordance with the device and the data it produces will improve security.

1.1. Contributions of This Paper

The major contributions of the proposed attribute-based access control scheme for IoT (AAC-IoT) system model are listed below.

- Develop a data owner and data user authentication system using unique security credentials as well as transfer the element with the assurance that the credentials are not leaked.
- In an ABAC scheme, the number of attributes for each owner is defined independently based on the data types and preferences. For this selection, a fuzzy logic method is used and then attributes are taken into account for the generation of access policies.
- HLF blockchain is presented to manage meta-data and security credentials from data owners and data users, respectively. In addition, the authentication of multiple incoming data users takes place by a neural network algorithm which can handle many users at a time.
- The data user is validated with identity, certificate, signature and Physical Unclonable Function (PUF); then a credence score is computed for each user device for identifying whether the device is legitimate or illegitimate.

1.2. Organization of This Paper

This paper is organized into the following sections. Section 2 details the background of this access control and the use of Hyperledger Fabric blockchain. Section 3 discusses the previous research work in attribute-based access control methods and their limitations and gives the problem defined in the access control of IoT. Section 4 illustrates the proposed access control scheme in the IoT environment. Section 5 demonstrates the experiments and its results. Section 6 illustrates the conclusion of the access control scheme.

2. Background

In this section, the work nature, elements, and importance of access control and HLF blockchain is discussed in two sub-sections. This provides an understanding of the background of this research.

2.1. ABAC IN IoT

ABAC is a process of extracting a set of attributes and defining a policy from it. These policies were used to allow access for particular devices. If the device fails to pass the generated access control policy, it will be restricted from the system. The attributes are defined based on the specifications of the entities and their data [23–26]. Hence, attributes differ for each data owner device.

The attributes are defined based on the subject, object, environment and permission. Actually, an attribute consists of a name and its corresponding value. The main categories of attributes are elaborated below.

- Subject Attributes: The subject attributes define the type of data captured by the IoT device.They include the characteristics of the device. For instance, in an IoT farm field, sensors will capture temperature, pressure or humidity for the specified time interval, but not all the devices can capture only a single datum; hence the subject attribute dependd on the application used. Some of the subject attributes are time, location coordinates, position, etc.
- Object Attributes: Object attributes are illustrated as the attributes of the device individuality, i.e., personal constraints belonging only to the particular type of device. These attributes are defined by the manufacturer of the device which are unique for each device. A few object attributes are identity, registration number, model type, model name, and so on.
- Environment Attributes: The environment attributes are the application in which the device is used, counts of data, etc.
- Permission Attributes: This type of attributes defines the permission provided to a particular user that can be read, written, edited, so on.

The preference of attributes for the access control policy is commonly performed by taking into account all the attributes. At present, IoT devices are popular and used globally for sharing data. The major challenges that exist in ABAC is scalability, resource utilization and heterogeneity. Due to the increase in the number of devices in the IoT, the challenges are critical. Designing an ABAC method, needs to concentrate on these challenges.

2.2. Hyperledger Fabric Blockchain

Blockchain technology is defined as a system that consists of recorded information in terms of blocks. Each block stores information in hashes so that the specific information is unable to be altered or hacked. The structure of HLF blockchain is shown in Figure 1. The couch database (DB) is used to store binary values, whereas the level database (DB) is used for storing key-value states that are involved in the peer process. The DB maintains key, key range and key queries.

HLF blockchain is a private framework constructed for the purpose of security [27–29]. This HLF is constructed using the major elements as follows.

- Peer: In blockchain technology, peers are the nodes that have records of transactions. This is one of the fundamental elements, and it takes the roles as endorsing peer, leader peer, anchor peer and committing peer. These peer nodes are responsible to host the ledger as well as the smart contract.
- Ledger: This is a collection of historical records in the blockchain maintained for all states of transactions. It is composed of a sequence of transactions. The ledger in this HLF is classified into two types: world state and level database (DB). One transaction in the block is a chaincode in HLF that is a key-value pair.

- Chaincode: The chaincode is defined as a program which is executed in HLF. It is responsible for managing and modifying the instructions to write the transactions. Whenever a transaction is initiated from a user, the chaincode is analyzed for decision making, and then it is applied to the ledger.
- Membership Services (MSP): The MSP in a blockchain consists of a set of identities that are permissioned. This identity is created based on public and private keys. The private key is kept secret, while the identity is validated without outsourcing the private key. The certificate authority plays a significant role in generating identity, and MSP occurs as local MSP and channel MSP.
- Channel: The channel in HLF is a private "subnet" which is required to create communication from one member to another. The transactions are considered to be confidential; hence the channel is used for it. Each peer that needs to perform a transaction joins a channel using its own identity provided by MSP.

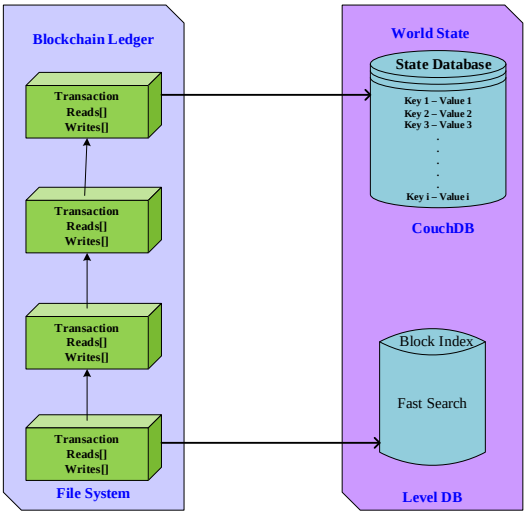

Figure 1. The Ledger in HLF blockchains.

3. Related Work

Prior research works in security provisioning of the IoT environment that perform authentication and ABAC schemes using blockchain are investigated.

In [30], the authors have developed a lightweight and secure architecture that uses a one-way hash function, an elliptic curve cryptography (ECC) and an XOR operation. The unique identity and password were converted into hash at the registration phase and submitted to a trusted server. Then the trusted server computes the challenges and stores it in a database. During the authentication phase, the challenge was computed and delivered to a server. Then the server checks the first challenge; then it computes the second challenge. After verifying the challenge, a session key was generated, and then a communication channel was established between the server and the device. The security credentials were shared using the ECC algorithm. The issue in this work was lack of sufficient security due to the consideration of simpler credentials for authentication which can be guessed easily. In addition, the selection of a weak curve in the ECC tends to produce poor generation of keys for encryption and decryption. The incorporated ECC algorithm was composed of multiple complex computations, and it is not sufficient in providing confidentiality.

A blockchain-based architecture was designed to perform authentication using the position and identity of the user in accordance with the determined trust level [31]. The trust level of each user depends on interactions. This work was called in short the Secure Path. The users create multi-signatures and check one after the other; then it connects to the blockchain for validation of the trust value which was dynamically determined. A decentralized security management system was proposed in this work that uses blockchain technology [32]. The conventional design of a blockchain using a Merkle hash tree was

involved for authentication. The gateway was responsible to check an IoT device, and the entities used for authentication were signature, metadata and Merkle tree proof. The smart contracts in the blockchain were presented for registration. According to the registered devices, the blockchain authenticates each device. In this work, multi-signatures are used; thus, there is a need to verify them one after the other which consumes time.

In [33], a blockchain-based authentication was proposed that uses a token mechanism. Initially, the registration was handled by the identity and address information as the user credentials. The signature generation was based on the elliptic curve digital signature function and hashing by SHA 256. According to this work, a token was generated with the hash identity and address. If the credentials are true, the token is granted, otherwise, it is denied. SHA 256 is a well performing hashing algorithm; however, it is comparatively poor with lightweight hashing. user credentials were commonly not secured during the registration process. To increase the efficiency of authentication, a decentralized blockchain–based authentication (DBA) protocol was proposed that takes into account multiple factors [34,35]. The multiple factors in DBA were identity, password and location . For the purpose of security, the lightweight SALSA algorithm was used. The issue of this work is the absence of security provisioning during registration of the user credentials.

A decentralized secure access control for the IoT was proposed in this paper that incorporates blockchain [36]. A hierarchical structure was developed that is composed of four blockchain managers as cloud, core fog, fog and local. The signature plays a vital role for verification of a user in blockchain. An authentication agent shares the secret key in a secure way with the Diffie–Hellman key exchange algorithm. After verification of identity by the authentication agent, the policy was checked by the authorization agent; later, the data was hashed and updated in the blockchain. The key exchange algorithm was not applicable to sign signatures; thus, it is vulnerable to a man-in-middle attack. In [37], the authors proposed an ABAC with authorized search scheme (EACAS); then an attribute-based encryption is used to ensure secure data. The major entities that are involved in this system model are data owner, data user and cloud server. The EACAS operates consecutively as system setup, key generation, encryption, trapdoor generation, data search and reverse of encryption. The attributes were hidden based partially on the conversion of the cipher text. Then the access trees were constructed using AND and OR gates during which user access was provided. The increase in policies will also increase the length of the tree.

A device access control scheme was proposed for the IoT, and it was known in short as DACS-IoT [38]. A one-way cryptographic hash function, i.e., SHA-1 and ECC algorithm, was proposed in this work. Initially, the user identity was verified followed by a certificate check and then a signature verification check. This verification was carried out between two smart devices, before data transmission. The ECC algorithm was weak in the selection of the curve. An ABAC scheme was carried out by the following steps of system initialization, registration, address creation, attribute request and access control [39]. The device has a unique identity and an address. This address was created by hashing the public key and identity using a concatenation operator. Then the hashed address was signed in attribute authority which will be updated in a Merkle tree that is present in the blockchain. During registration, identity-based cryptography is used to verify identities; however, this is not sufficiently secure since identities are basic credentials that can be forged.

Two concepts of ABAC and identity-based signature (IBS) were presented in [40]. The designed architecture performs four steps as system setup, request control, point decision point (PDP) and point enforcement point (PEP) execution. The PDP consists of an identity device, a request number, a decision and a signature. On the other hand, PEP is composed of the identity, decision and signed identity and signature. Based on the access request, the access decision was evaluated for each user. However, the identity was not a secure constraint; hence it can be eavesdropped on easily by attackers. A blockchain-aided searchable attribute-based encryption (BC-SABE) that was decentralized using a blockchain was proposed [41]. The designed coalition blockchain enables creation of partial tokens

for users. This system model is composed of four main entities such as data owner, data user, blockchain and cloud. The sequential processes are system initialization, registration, revocation, key generation, encryption, token generation, search and decryption. The user attribute set was validated with respect to the access structure, and then the search results are returned. The access validation in blockchain was performed one after the other which consumes time for processing.

A blockchain-enabled decentralized capability access control, i.e., BlendCAC was discussed in this paper [42]. The BlendCAC architecture is comprised ofthe data user, the data owner and the blockchain network. Initially, in registration, the public and private keys were generated, and the identity was verified for authentication. A virtual identity (VID) was also developed. Then, according to the process of access control, a capability token was generated, and the policy decision was performed to give access to the user. An attribute-based framework was proposed to provide access control and communication control (ABAC-CC) [43]. Communication was established between the devices to the gateway, gateway to gateway, gateway to cloud and cloud to cloud. The authentication of the devices was performed using the cryptographic keys and then followed by the access control based on the attributes. The development of virtual identities creates conflict between the original identities and the virtual identities.

The major limitations in the existing research work for the process of authentication and attribute-based access control with the assistance of blockchain is illustrated and solved in this proposed work.

Problem Statement

The problems identified from previous research works are expanded in this section. An access control scheme and a user credibility incentive mechanism was proposed in [44] called LBAC. The designed Hyperledger Fabric operates in three parallel phases, and the phases are execute, order and validate. The certificate authority (CA) initializes at first; then the attribute authority (AA) submits the attribute to the blockchain. A further attribute token was generated using the set of attributes of the user which is then allowed for storage or access into the Cloud Service Provider (CSP). In this work, meta-data were stored in the blockchain, whereas the whole data were stored in the CSP. The meta-data from the DO consists of a symmetric key in cipher text and a hash value of data in cipher text. Hence, it performs proxy re-encryption, i.e., pre-decryption in the blockchain and final decryption for the CSP data.

In [45], Device Contract (DC), Policy Contract (PC) and Access Contract (AC) were proposed. These three contracts define the development of an attribute-based access control (ABAC) model, an ABAC policy and resource management. The defined policy can be updated and deleted in the blockchain. The major problems stated in these works are illustrated in the following,

- An IoT environment deals with a massive number of devices; hence data arrive in a short time period. The decryption requires IoT devices, which are actually computation-constrained devices, to utilize scarce resources.
- User data are not protected during storage that causes alteration of data, since the data can be sensitive and data privacy is essential.
- In this work, the admin has rights to modify, add or delete the policy contract, but the admin was not authenticated.If the admin is compromised, the contracts will be altered in favor of the illegitimate user's requirements.
- The blockchain technology uses the SHA-256 algorithm for hashing which is also used in this work. It is not a lightweight hashing algorithm; hence it is difficult with massive numbers of IoT devices.

In [46], an HLF Blockchain was proposed using the performance of local authentication of client devices. A multi-layer architecture network is designed, and the layers are IoT devices, cluster heads, base station and data storage (server). A client, i.e., an IoT device and server communication, was established with a hello packet. It then exchanges a key

using the Diffie–Hellman algorithm. The authentication takes place by exchanging the challenge–response that was generated by the client itself, and during authentication, the response was validated at the server side. The basic research issues in this proposed work are,

- The performance of authentication using a challenge–response is generated by the client; in case the client is not legitimate, it can interrupt the system. This is the only validation that is performed to guarantee a client. The response computation at the server side is simpler using a single predefined mathematical operator.
- The local authentication between the client and server fails to consider the unique security credential of the client.

In [47], an access tree was constructed and managed by the authority node (AN). The access tree was created from the chaincode that defines the credentials of requesters. The ECC algorithm was applied for key generation which is used as the identity of the device. The device's credentials such as identity, public key, access policy, IP address, group identity, access history and attribute library are used for validation of device accounts. Then the device creates an access policy which is further converted into a collaboration access tree using the access for each device provided. The problems identified from this work are illustrated below.

- In this work, if the request is not satisfied by the access tree in the authority node, then the access tree is reconstructed. This increases time consumption due to re-construction of trees.
- The used ECC algorithm returns a large size of the encrypted data and also a complex algorithm that requires a large amount of resources.

The overall problem in this research on the IoT environment for the authentication and access control with a blockchain is the use of complex cryptography, hashing algorithms and, on the other hand, the consideration of attributes for the construction of access policies. The problems stated in this section are focused and solved by our proposed model.

4. The Proposed AAC-IoT System Model

In this section, the proposed AAC-IoT system model is detailed along with the defined solutions. This section is divided into three parts: system model, registration and authentication and ABAC scheme.

4.1. System Model

The proposed architecture is designed to assure secure data storage and attribute-based access control using the assistance of a Hyperledger Fabric blockchain. This system model consists of a data owner (DO), a data user (DU),Sity (CA), HLF blockchain and cloud. The proposed AAC-IoT system model is depicted in Figure 2.

- Data Owner: The data owner first registers with the certificate authority and then uploads meta-data (cipher text of asymmetric key and hash value of identity and number of attributes) into the fabric blockchain followed by an upload of the remaining data into the cloud storage. The data owner has to be authenticated each time to upload the data in the cloud. Let N be the total number of DOs represented as $\mathcal{D}_N = D_1, D_2, \ldots\ldots, D_N$. Each owner has an individual identity that is received from a certified authority.
- Data User: The data user also registers with the certificate authority and then submits an access request to the Hyperledger Fabric blockchain. The DUs are n in total number, and they are denoted as $\mathcal{U}_n = u_1, u_2, \ldots\ldots, u_n$. The DUs are the mobile devices that submit the query and receive the required response from the cloud. The data user's credentials are validate,d and the access policy is validated.
- Certificate Authority: This entity is the trusted one in the system, and it is responsible for managing certificates of the DOs and the DUs. On the other hand, this entity

computes the number of attributes to be considered later for identifying a device account in the blockchain.

- Hyperledger Fabric Blockchain: This type of blockchain is open access, and it is used for authentication of the DO and the DU. In addition, it verifies the access control policy generated based on attribute and allows or denies access.
- Cloud Server: This is a storage system that allows large size data to be stored according to the permission provided. To protect the data, the original data is encrypted and stored in the cloud server.

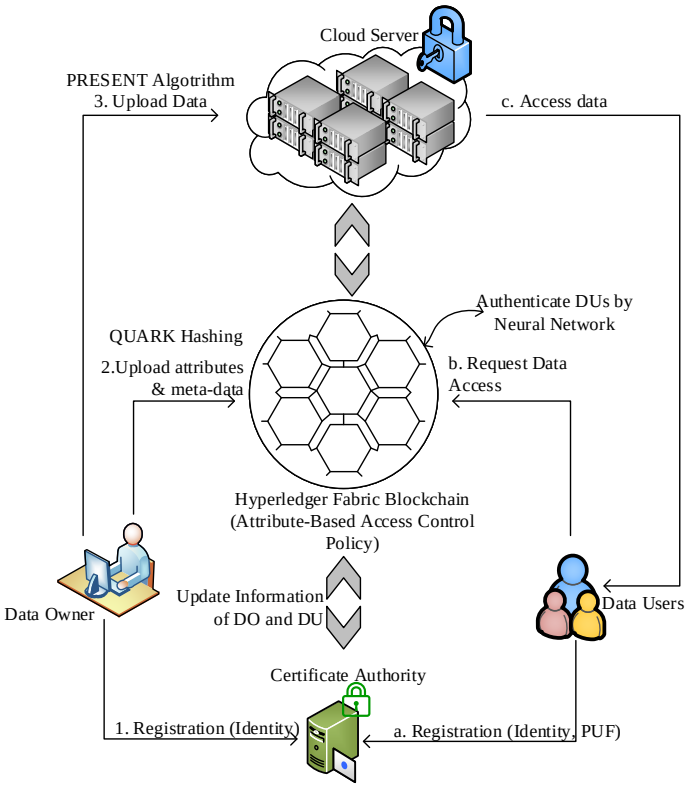

Figure 2. The Proposed AAC-IoT system model.

4.2. Registration and Authentication

The entities in the network system begin with the process of registration. The DO and the DU use different types of credentials for registration and authentication since the DO submits only the data, whereas the DU uploads a request query and then receives data. The registration procedure is illustrated in sequential steps.

4.2.1. DO Registration

Step 1: Let $\mathcal{D}_1$ be a DO. It initially sends a request to the CA with the unique identity and timestamp. The registration of the DO is $R_1 \rightarrow \{Id_1, T_0\}$, where Id_1 is the owner's identity, R_1 denotes the first request from the DO, and T_0 is the initial timestamp.
Step 2: The CA receives R_1, and if $T_0 < \Delta T$, where ΔT is the timestamp for receiving the request message, the identity is considered for registration. If the timestamp is not validated, then the expired communication link of the owner and the CA will also be taken in account for registration.

Step 2.1: Then the certificate is generated for each DO and transferred securely in the form of encryption. For encryption, a lightweight PRESENT algorithm is used which consumes limited computations [48].

Step 2.2: Using addRoundkey generation, the certificate is encrypted into cipher text. The key lengths are 80-bit and 128-bit, and the block length of 64-bits is used. This

block cipher algorithm is handled in 31 rounds. The key for each round is K_i, where $1 \leq i \leq 32$ and it uses an XOR operation. The sBoxLayer and pLayer state; hence it uses an S/P network.

Step 2.3: Then, the CA sends $R_2 \rightarrow E\{Id_1, E(Ct_1), T_1\}$ to $\mathcal{D}_1$ with the encrypted certificate $E(Ct_1)$ and timestamp T_1.

Step 3: The $\mathcal{D}_1$ receives R_2 and checks timestamp $(T_1 - T_0) < \Delta T$. If valid, the certificate is decrypted and then sends signature SO_{xx} generation requests to CA. The $R_3 \rightarrow \{Id_1, SO_{xx}, T_2\}$ where R_3 is the request from owner, SO_{xx} denotes the unknown signature, and T_2 is the timestamp.

Step 4: CA validates $(T_2 - T_1) < \Delta T$, then generates a signature for the $\mathcal{D}_1$ and sends it to the owner. The security credentials generated for all $\mathcal{D}_N$ are updated in the HLF blockchain of the system.

The designed DO registration workflow is depicted in Figure 3 which demonstrates all the three entities involved during the registration using multiple security credentials.

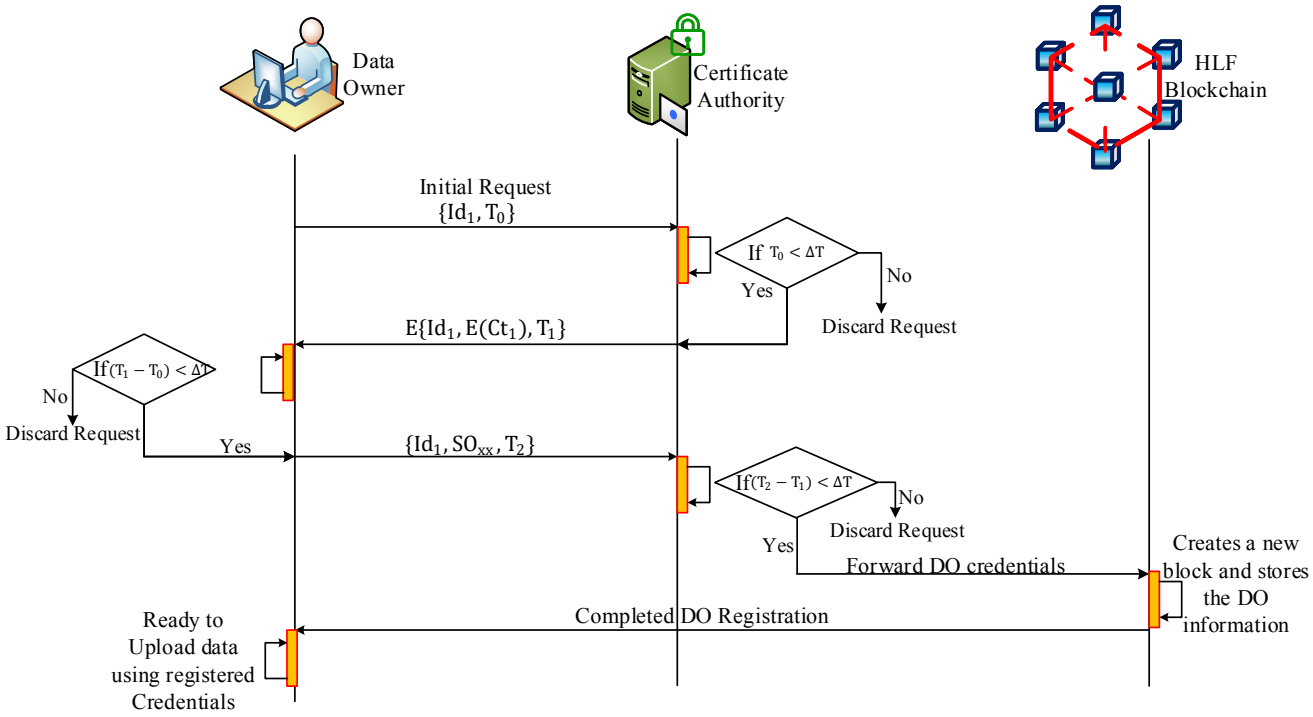

Figure 3. Workflow of the DO registration.

4.2.2. DU Registration

Step 1: Consider a DU, u_1 from $\mathcal{U}_n$ that registers the same as a DO. We perform step 1 to step 4 as in the DO registration.

Step 2: CA checks $(T_2 - T_1) < \Delta T$ and submits a request for the PUF of the corresponding, and the message is $R_3 \rightarrow \{Id_1, SU_1, T_3\}$.

Step 3: The u_1 checks the timestamp validity as $(T_3 - T_2) < \Delta T$, then gives a PUF that consists of challenges $(C_1, C_2, ..., C_k)$ and the corresponding responses $(Re_1, Re_2, ..., Re_k)$, where k denotes the total number of challenges and the responses in the PUF of the u_1. The u_1 sends $R_4 \rightarrow \{Id_1, SU_1, (C_k, Re_k), T_4\}$ to CA.

Step 4: The CA receiving R_4 from u_1, extracts the message and checks timestamp $(T_4 - T_3) < \Delta T$, and forwards the (C_k, Re_k) pairs to the HLF blockchain.

Step 5: After submission of all the credentials from u_1 to CA, the process of registration is successfully completed, and the user can access data from the cloud server based on the access policy. The complete workflow of this registration is shown in Figure 4.

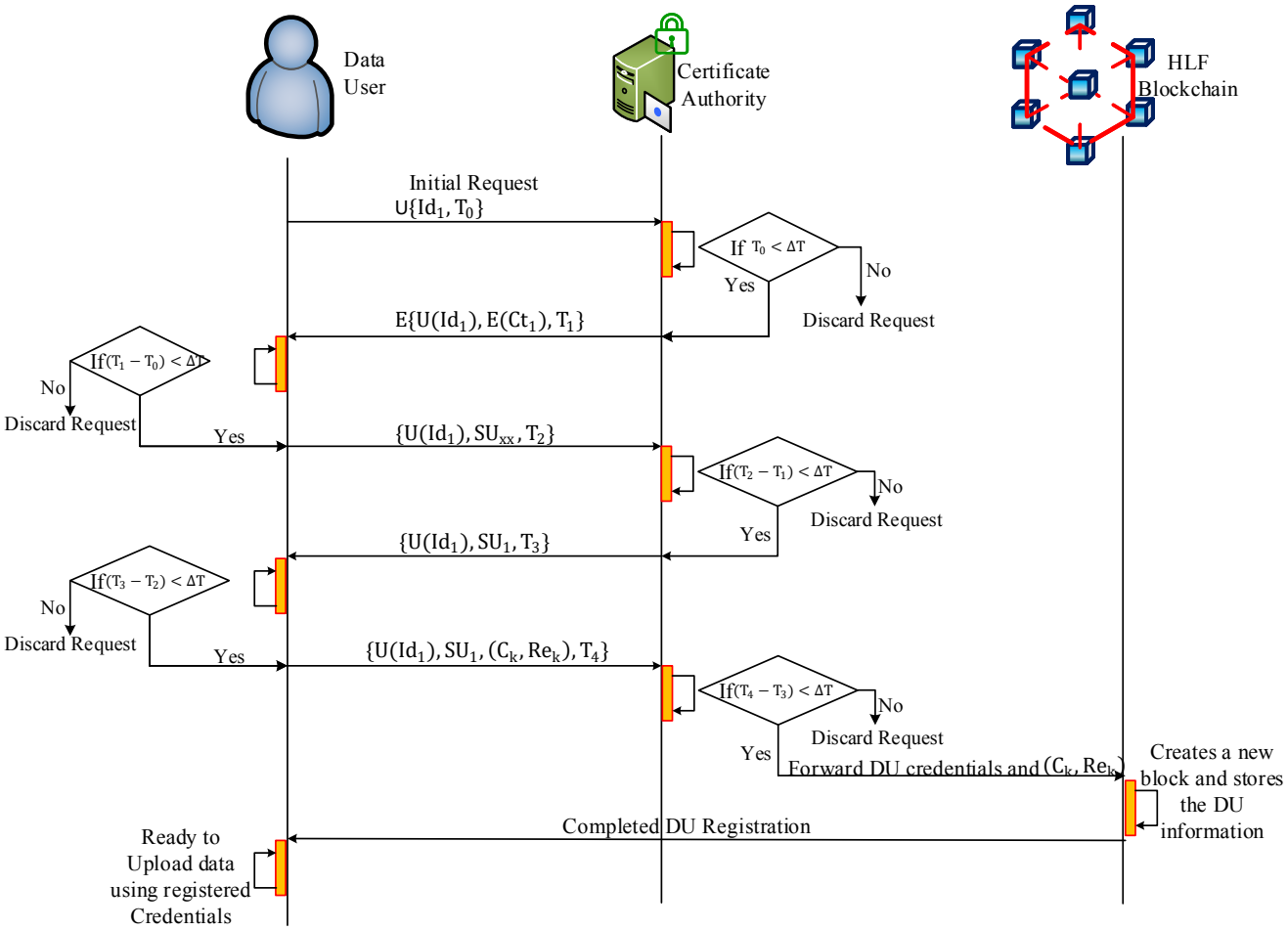

Figure 4. Workflow of the DU registration.

4.2.3. DO Authentication

The authentication of the DO is based on the credentials identity, certificate and signature. The sequential process followed in the DO authentication is represented in Figure 5. The DO is authenticated whenever a data upload is needed. In this work, the DO will upload meta-data into the HLF including the attributes for generating access control policies. On the other hand, the main-data of the DO are stored in a cloud server using a lightweight PRESENT block cipher algorithm. The steps followed for the DO authentication are illustrated as follows.

Step 1: The DO $\mathcal{D}_1$ submits the authentication request consisting of $R_1 \rightarrow \{H(Id_1), T_0\}$, i.e., hashed identity with timestamp, to the CA. The hashing is performed using a QUARK algorithm which is lightweight [49].

Step 1.1: Construction of sponge by padding at initialization until the specified length is reached.

Step 1.2: An XOR operation is performed in the absorbing phase and returns permutation from this phase.

Step 1.3: Next, in the squeezing phase, the bits to be hashed are extracted, and the process of extraction is completed.

Step 1.4: The hashed security credentials in the HLF are verified from the given DO credentials.

Step 2: The CA checks timestamp, $T_0 < \Delta T$ and then the hash identity in the HLF blockchain is extracted from the block. and if valid, the signature and certificate are requested for the DO as $R_2 \rightarrow \{H(SO_1 \oplus h(Ct_1), T_1)\}$ to HLF blockchain via the CA.

Step 3: The CA checks timestamp $(T_1 - T_0) < \Delta T$. If valid it forwards the $DO \rightarrow \{H(SO_1 \bigoplus h(Ct_1)\}$ to the HLF blockchain for validation. On completion of this authentication, the DO can submit meta-data to the HLF. The HLF blockchain sends $R_3 \rightarrow \{\mathbb{S}, T_2\}$ to the DO, where $\mathbb{S}$ denotes successful authentication.
Step 4: The DO verifies the timestamp $(T_2 - T_1) < \Delta T$, and then the meta-data and attributes for the data $\{a_1, a_2, a_3, \ldots.\}$ are uploaded to the HLF blockchain and the complete data are uploaded into a cloud server using the $\mathbb{S}$ which assures the DO is legitimate.

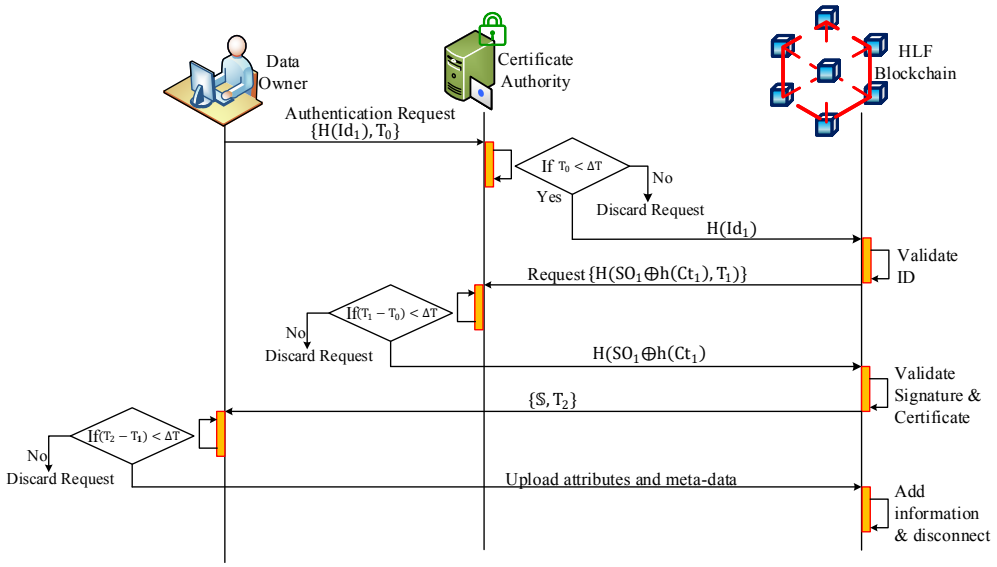

Figure 5. Workflow of DO authentication.

4.2.4. DU Authentication

The authentication process of the DU is based on the neural network that is able to tolerate all the arrived user devices. The neural network is enabled to consider multiple user requests received at a time. The security credentials are transferred in hash format for assuring security. Let u_Q represent the total number of DU authentication requests arrived at time t. The neural network consists of three layers: an input layer, a hidden layer and an output layer. Initially, the requests are received at the input layer, and then security credentials are extracted individually for each DU.

A set of nodes participates in each layer of the network. The layers and their work process is illustrated below. The construction of an artificial neural network (ANN) for DU authentication is depicted in Figure 6. The nodes in each layer of the ANN connect to the nodes in the next layer. The operational functions that are used in the neural network are the linear function or the activation function.

- Layer 1: Input Layer
  This is the first layer of the ANN that receives the DU requests along with the identity, signature, certificate, successful access to policy and rating value. The entity's identity, signature and certificate are in hash values. The input security credentials are independent variables; hence they are to be validated individually.
- Layer 2: Hidden Layers
  The hidden layer can be constructed in multiple numbers. In this work, four layers are created in a hidden layer. Each hidden layer extracts the particular credential and validates the hash values. In the fourth hidden layer, the credence score for the DU is computed, and then the decision is taken. The credence score is estimated from the total number of counts of the successful access policy (S_P) and the behavior of the DU

that gives a higher rating value or a low rating value (R_v). The credence score CS is given as,

$$CS = S_P + R_v \quad (1)$$

If the $CS > CS(T)$, then the DU is legitimate; if the condition fails, then the DU access will be denied as it is identified as illegitimate.

- Layer 3: Output Layer
  The validated security credentials are involved in decision making in this layer. If all the credentials pass, then the result is positive, and the CA sends a challenge from the PUF.

The DU receiving a C_k will identify the corresponding Re_k, and finally the authentication process of a DU has been successfully completed. After this, based on the given access policies, the meta-data are extracted from the HLF blockchain and original data from cloud in an encrypted format.

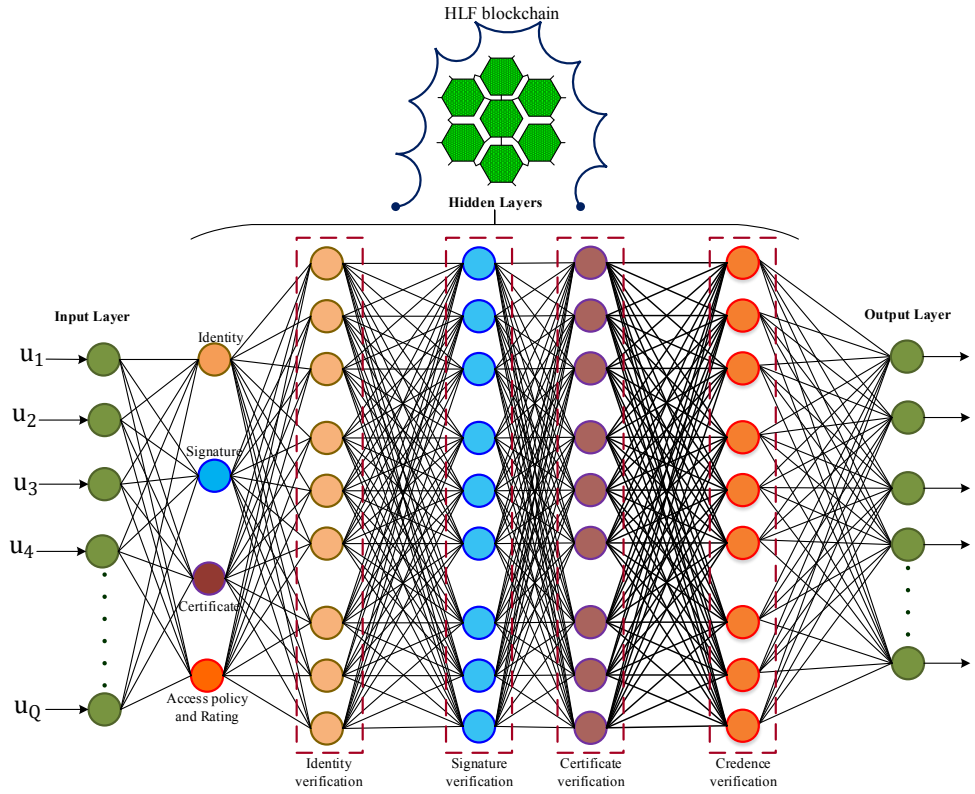

Figure 6. Neural Network for DU Authentication.

4.3. Attribute-Based Access Control Scheme

The ABAC scheme is proposed to allow the DUs to access the DO's uploaded data. In general, a set of attributes are predefined, and they are used for the construction of each access control policy. In this proposed ABAC scheme, the number of attributes is selected in accordance with the importance of data, and then the access policy is created. For instance, industrial data can be submitted with lesser attributes due to its non-sensitivity. In the same way, healthcare data are required to be provided with a high number of attributes due to the sensitivity of the data. The number of attributes for the access control policy is determined from a fuzzy logic method by taking into account data type and data preference.

- Data Type (D_T): The input data type is defined as the type of the data that can be either sensitive or non-sensitive. The sensitive data represent the importance of the data, i.e., if data are confidential, then they are sensitive; otherwise, they are non-sensitive data.

- Data Preference(D_P): The preference of data is defined as the priority provided from the DO. The DOs give a priority based on their requirement for the selection of the number of attributes. If the DO decides to be more protected, then the preference is higher, otherwise it is less.

The degree of truth is the key concept of fuzzy logic which is enabled to make a decision according to the given input. The fuzzy logic method works on the basis of three processes as follows,

- Fuzzification: The two inputs D_T and D_P are in the form of crisp values which are converted into a fuzzy set in the fuzzifier; then it moves to the next block after conversion.
- Interference Engine: This engine is responsible for decision making, and the process takes place based on the generated IF–THEN rules. A knowledge base is connected to an interference engine which consists of a set of defined rules based on the considered set of inputs.
  Table 1 depicts the input and its corresponding output. The fuzzy sets are operated using the membership function that is represented as μ_F: $X \rightarrow [0, 1]$. The X is the input that maps towards the membership function, and F is the fuzzy set. The term $N(a)$ represents the number of selected attributes, and $n(T)$ is the total number of attributes. The input follows the result as per the rule and gives the output to the next block.
- De-Fuzzification: The fuzzy set output from the engine is converted into crisp values in this defuzzifier. According to the obtained results, the number of attributes is used for generation of policy.

Table 1. Fuzzy input and output.

D_T	D_P	Output	Attributes
0	0	0	$N(a) < n(T)$
0	1	0	$N(a) < n(T)$
1	0	1	$N(a) = n(T)$
1	1	1	$N(a) = n(T)$
Condition		**THEN (Action)**	
IF	**AND**		
$D_T = L$	$D_P = L$	Use few attributes	
$D_T = L$	$D_P = H$	Use few attributes	
$D_T = H$	$D_P = L$	Use all attributes	
$D_T = H$	$D_P = H$	Use all attributes	

A policy is defined based on the attributes that are based on subject attributes(A_s), object attributes (A_O), environment attributes (A_E) and permission attributes (A_P). According to the four type of attributes, the access policy is generated, and then the DU that passes the policy is allowed to access the DO's data.

The fuzzy logic method in the selection of attributes is illustrated in Figure 7. While the result is to consider a lesser number of attributes, then A_P and A_s will be taken into account for the construction of the access policy. The fuzzy membership function is defined initiall,y and it can be updated as per increases in input. This fuzzy logic method is operated for each upload request from the DO. The attribute-based access control policies are generated and updated into the HLF blockchain and used whenever the DU submits a request for a data.

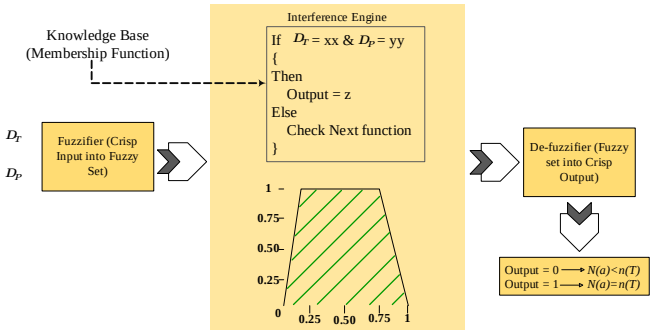

Figure 7. Fuzzy logic method in ABAC.

5. Experiment and Comparison

The proposed AAC-IoT system is developed, and the results are evaluated for comparison. This section is organized in two parts: implementation environment and comparative analysis.

5.1. Implementation Environment

The proposed AAC-IoT system is implemented using the Java Development Kit, and the working model of this is designed by the ifogsim simulator. The tool and the other specifications that are used in the development are illustrated in Table 2.

The topological structure that is designed for result evaluation is depicted in Figure 8. The same configuration and parameters are used to evaluate the outcome of previous LBAC. The lightweight algorithms PRESENT and QUARK are included; authentication is performed using an ANN, and the selection of number of attributes are based on a fuzzy logic method. The performance measurements are measured and analyzed.

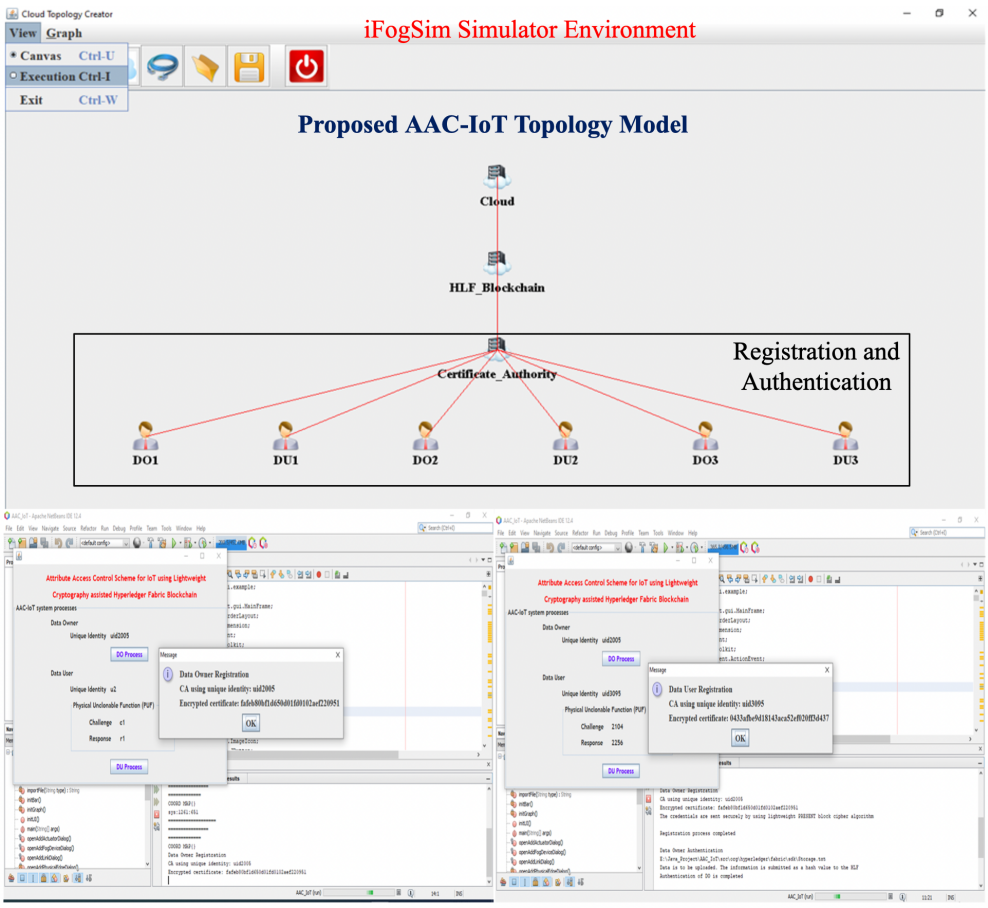

Figure 8. The AAC-IoT implementation model.

Table 2. Tools and parameters.

System Parameter	**Specification**
Number of DOs	3
Number of DUs	3
Number of CA	1
Hardware / Software Used	**Version**
Java Development Kit	1.8
NetBeans IDE	8.2
Windows	10 (64-bit)
RAM	2 GB (Min)
Processor	Dual core & above

5.2. Comparative Analysis

In this comparative analysis, the proposed AA-IoT model is compared with the previous LBAC in [44]. The previous work uses the same set of entities, and the HLF blockchain is involved. However, in this work, the hashing was based on SHA-256, and the storage of data was carried out twice. The DUs are IoT devices that have limited ability to perform computations. On the other hand, the arrival of multiple requests from each DU is required to be handled by the system. The processing of arrived requests one by one consumes more time than the parallel processing of it. The efficiency of this attribute-based access control scheme is determined by evaluating parameters such as latency, throughput and storage overhead. Each parameter plays a vital role in predicting the performance of the proposed access control scheme. The use of a blockchain assures successful security. On the other hand, an authentication process restricts to allow illegitimate users into the system.

5.2.1. Efficiency of Latency

Latency is defined as the delay in processing the data after transmission. The increase in latency represents poor construction and performance of the system. The latency is measured and plotted with respect to the increase in transaction number and transaction rate.

The proposed AAC-IoT incorporates authentication of the DO and the DU; then the ABAC policy scheme is used. These processes take place with the assistance of the HLF blockchain that enables the management of security credentials and is stronger against multiple attacks. The comparison of latency is demonstrated in Figures 9 and 10, which show the increase in transaction count and rate, respectively. From the results, the AAC-IoT achieves less latency than the previous LBAC method. The decrease in latency is due to the use of an ANN for authentication that is able to manage multiple requests based on parallel processing. This aspect is not present in the LBAC method. On average, the latency is 1.8 S and 2.0 S in the number of transactions; then it is 2.0 S and 2.5 S in the transaction rate for AAC-IoT and LBAC, respectively. However, the difference is minor. It is large when the transactions of users increases massively. The proposed work minimizes the latency compared to the previously implemented LBAC algorithm by approximately 10%. The reasons for the minimizing latency in the proposed AAC-IoT compared to the existing LBAC are given below:

- Ability of the algorithm to validate the credentials in parallel that gives results for all the users arrived at the time t.
- The authentication of users also reflects on the minimization of latency, in the aspect of restricted unnecessary requests from malicious or illegitimate users.

- The use of an HLF blockchain is decentralized which allows users to access from any location using the registered credentials.

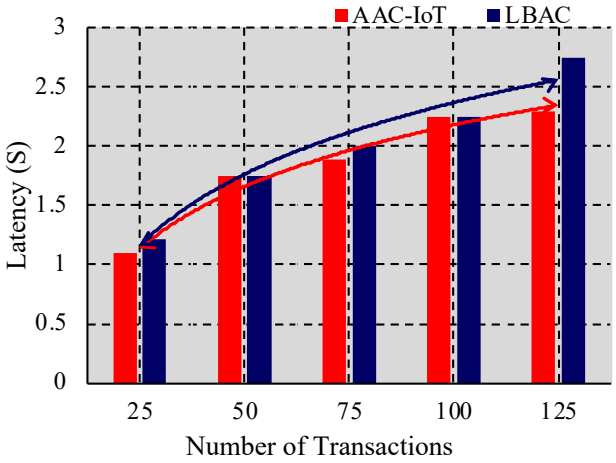

Figure 9. Performance of latency plot against transaction number.

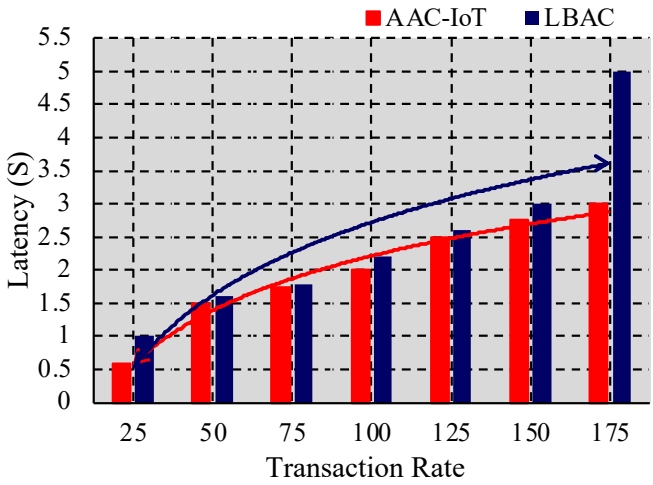

Figure 10. Performance of latency plot against rate of transaction.

5.2.2. Efficiency of Throughput

The parameter throughput defines successful transmission of data appropriately. The increase in throughput illustrates that the efficiency of the system is better. Hereby, the measured throughput is marked across the transaction number, and the results are depicted in Figure 11.

According to the increase in transaction count, the throughput also increases in the AAC-IoT , where it is higher than the existing LBAC. The throughput increases because of the parallel authentication process that allows users to submit a request and receive a response. In contras,t the LBAC uses the HLF blockchain for access control policy, whereas the authentication process is not efficient; hence it allows illegitimate user requests that tend to minimize the throughput with the increase in transaction counts. In an average estimation, the throughput is about 82 Mbps and 189 Mbps for LBAC and AAC-IoT, respectively. There is a large difference between the proposed and existing works, and this is due to the performance of the authentication and access control mechanism. The proposed AAC-IoT achieves higher throughput compared to the previous LBAC method. On average, 50% of the throughput has been increased in the proposed work. The increase in throughput will improve with the increase in transaction number on a large scale.

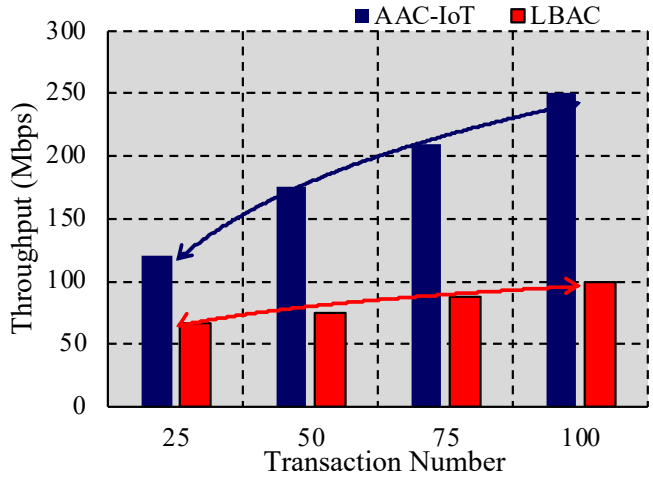

Figure 11. Performance of throughput plot against number of transactions.

5.2.3. Efficiency of Storage Overhead

The storage overhead is the excessive usage of the capacity. As per this ABAC scheme, the number of attributes used for the construction of policies for each data is predicted as storage overhead. As per the increase in the number of attributes, the storage overhead also increases; hence to solve this issue, the number of attributes are computed for the generation of access control policy in the AAC-IoT. This selection of number of attributes is not performed in LBAC, and hereby, the difference in the performance of storage overhead is shown in Figure 12.

From the result outcome, storage overhead is evaluated, and the LBAC attains higher storage overhead than the proposed AAC-IoT. This is because of the consideration of all the attributes in LBAC for the generation of the access policy, whereas it is counted in AAC-IoT using a fuzzy logic method. The designed AAC-IoT system with fuzzy logic takes into account the data type and preference for the computation of the use of the number of attributes to create an access control policy. From Figure 12, the storage overhead in the proposed AAC-IoT is minimized by approximately 20% from the previously implemented LBAC algorithm. This improvement of storage overhead enables the AAC-IoT to enhance the arrival of data for storage.

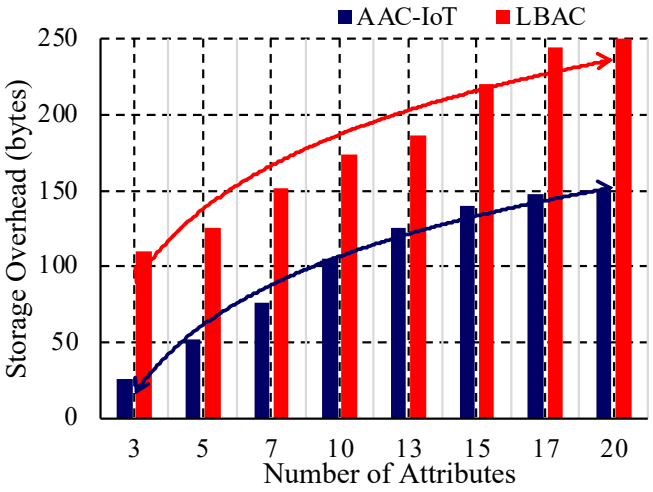

Figure 12. Performance of storage overhead plot against number of attributes.

Overall, the proposed AAC-IoT system model achieves better performances than the existing LBAC. The increase in throughput and the reduction of latency and storage overhead is efficient; hence this model can be used for sensitive applications.

6. Conclusions and Future Work

The proposed AAC-IoT system incorporates the HLF blockchain for the processing of an authentication and attribute-based access control scheme. The authentication process is constructed for the DO as well as the DU who uploads and accesses the data, respectively. The security credentials are secured using the lightweight PRESENT block cipher algorithm. The meta-data are uploaded to the HLF blockchain and the main data are encrypted and stored on a cloud server. The HLF is responsible for the authentication of the credentials that are in the form of hash values generated using a lightweight QUARK algorithm. The authentication credentials include certificate, signature and PUF. In this work, the number of attributes to be used in an access policy is determined using a fuzzy logic method. According to the importance of the data, the access policy is defined to minimize the matching access policy as it does not consider all the attributes for all types of data. On the other hand, the DOs and the DUs are authenticated on each request. To make it faster for multiple requests, an ANN is used for authentication of the DU. As an outcome, this proposed AAC-IoT achieves better in terms of latency, throughput and storage overhead than the previous method. In future, the designed AAC-IoT system model is planned to be evaluated using two different datasets from healthcare and industry, and the testing will be conducted in a large-scale environment.

Author Contributions: Conceptualization, S.A. and O.B.; funding acquisition, S.A.; investigation, S.A. and O.B.; methodology, S.A. and O.B.; project administration, S.A.; validation, S.A.; writing—original draft, S.A.; writing—review and editing, S.A. and O.B. All authors have read and agreed to the published version of the manuscript.

Funding: This project was funded by the Deanship of Scientific Research (DSR) at King Abdulaziz University, Jeddah, under grant No. J:71-612-1442. The authors, therefore, acknowledge with thanks DSR for technical and financial support.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

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

References

1. Zikria, Y.B.; Kim, S.W.; Hahm, O.; Afzal, M.K.; Aalsalem, M.Y. Internet of Things (IoT) Operating Systems Management: Opportunities, Challenges, and Solution. *Sensors* **2019**, *19*, 1793. [CrossRef] [PubMed]
2. Bhatti, F.; Shah, M.A.; Maple, C.; Islam, S.U. A Novel Internet of Things-Enabled Accident Detection and Reporting System for Smart City Environments. *Sensors* **2019**, *19*, 2071. [CrossRef] [PubMed]
3. Pradhan, B.; Bhattacharyya, S.; Pal, K. IoT-Based Applications in Healthcare Devices. *J. Healthc. Eng.* **2021**, *2021*, 6632599. [CrossRef] [PubMed]
4. Jahmunah, V.; Sudarshan, V.K.; Oh, S.L.; Gururajan, R.; Gururajan, R.; Zhou, X.; Tao, X.; Faust, O.; Ciaccio, E.J.; Ng, K.H.; et al. Future IoT tools for COVID-19 contact tracing and prediction: A review of the state-of-the-science. *Int. J. Imaging Sys. Technol.* **2021**, *31*, 455–471. [CrossRef]
5. Abiodun, O.I.; Abiodun, E.O.; Alawida, M.; Alkhawaldeh, R.S.; Arshad, H. A Review and the Security of the Internet of Things: Challenges and Solutions. *Wirel. Pers. Commun.* **2021**, *119*, 2603–2637. [CrossRef]
6. Mousavi, S.K.; Ghaffari, A.; Besharat, S.; Afshari, H. Security of Internet of things based on cryptographic algorithms: A survey. *Wirel. Netw.* **2021**, *27*, 1515–1555. [CrossRef]
7. Bhatt, S.; Ragiri, P.R. Security trends in Internet of Thing: A survey. *Appl. Sci.* **2021**, *3*, 121.
8. Mohammed, M.H.S. A Hybrid Framework for Securing Data Transmission in Internet of Things (IoTs) Environment using Blockchain Approach. In Proceedings of the IEEE International IoT, Electronics and Mechatronics Conference (IEMTRONICS), Toronto, ON, Canada, 21–24 April 2021.
9. Giannoutakis, K.M.; Spathoulas, G.; Filelis-Papadopoulos, C.K.; Collen, A.; Anagnostopoulos, M.; Votis, K.; Nijdam, N.A. A Blockchain Solution for Enhancing Cybersecurity Defence of IoT. In Proceedings of the IEEE International Conference on Blockchain, Rhodes, Greece, 2–6 November 2020.

10. Atlam, H.F.; Azad, M.A.; Alzahrani, A.G.; Wills, G. A Review of Blockchain in Internet of Things and AI. *Big Data Cogn. Comput.* **2020**, *4*, 28. [CrossRef]
11. Picone, M.; Cirani, S.; Veltri, L. Blockchain Security and Privacy for the Internet of Things. *Sensors* **2021**, *21*, 892. [CrossRef]
12. Uddin, M.A.; Stranieri, A.; Gondal, I.; Balasubramanian, V. A Survey on the Adoption of Blockchain in IoT: Challenges and Solutions. *Blockchain Res. Appl.* **2021**, *2*, 100006. [CrossRef]
13. Shala, B.; Trick, U.; Lehmann, A.; Ghita, B.; Shiaeles, S. Blockchain and Trust for Secure, End-User-Based and Decentralized IoT Service Provision. *IEEE Access* **2020**, *8*, 119961–119979. [CrossRef]
14. Christidis, K.; Devetsikiotis, M. Blockchains and Smart Contracts for the Internet of Things. *IEEE Access* **2016**, *4*, 2292–2303. [CrossRef]
15. Singh, S.; Hosen, A.S.; Yoon, B. Blockchain Security Attacks, Challenges, and Solutions for the Future Distributed IoT Network. *IEEE Access* **2021**, *9*, 13938–13959. [CrossRef]
16. Kim, J.; Park, N. Blockchain-Based Data-Preserving AI Learning Environment Model for AI Cybersecurity Systems in IoT Service Environments. *Appl. Sci.* **2020**, *10*, 4718. [CrossRef]
17. Sultana, T.; Almogren, A.; Akbar, M.; Zuair, M.; Ullah, I.; Javaid, N. Data Sharing System Integrating Access Control Mechanism using Blockchain-Based Smart Contracts for IoT Devices. *Appl. Sci.* **2020**, *10*, 488. [CrossRef]
18. Vivekanandan, M.; Sastry, V. N.; Srinivasulu Reddy, U. BIDAPSCA5G: Blockchain based Internet of Things (IoT) device to device authentication protocol for smart city applications using 5G technology. *Peer-Peer Netw. Appl.* **2021**, *14*, 403–419. [CrossRef]
19. Ouaddah, A.; Elkalam, A.; Ouahman, A. FairAccess: A new Blockchain-based access control framework for the Internet of Things. *Secur. Commun. Netw.* **2017**, *9*, 5943–5964. [CrossRef]
20. Dorri, A.; Kanhere, S.S.; Jurdak, R.; Gauravaram, P. LSB: A Lightweight Scalable Blockchain for IoT security and anonymity. *J. Parallel Distrib. Comput.* **2019**, *134*, 180–197. [CrossRef]
21. Liu, Y.; Lu, Q.; Chen, S.; Qu, Q.; O'Connor, H.; Choo, K.K.R.; Zhang, H. Capability-based IoT access control using blockchain. *Digit. Commun. Netw.* **2020**, *7*, 463–469. [CrossRef]
22. Song, L.; Li, M.; Zhu, Z.; Yuan, P.; He, Y. Attribute-Based Access Control Using Smart Contracts for the Internet of Things. *Procedia Comput. Sci.* **2020**, *174*, 231–242. [CrossRef]
23. Bezawada, B.; Haefner, K.; Ray, I. Securing Home IoT Environments with Attributes-Based Access Control. In Proceedings of the Third ACM Workshop on Attribute-Based Access Control, Tempe, AZ, USA, 19–21 March 2018; pp. 43–53.
24. Aliane, L.; Adda, M. HoBAC: Toward a Higher-order Attribute-Based Access Control Model. *Procedia Comput. Sci.* **2019**, *155*, 303–310. [CrossRef]
25. Wang, J.; Wang, H.; Zhang, H.; Cao, N. Trust and Attribute-Based Dynamic Access Control Model for Internet of Things. In Proceedings of the International Conference on Cyber-Enables Distributed Computing and Knowledge Discovery (CyberC), Zhengzhou, China, 18–20 October 2018.
26. Chmiel, M.; Korona, M.; Kozioł, F.; Szczypiorski, K.; Rawski, M. Discussion on IoT Security Recommendations against the State-of-the-Art Solutions. *Electronics* **2021**, *10*, 1814. [CrossRef]
27. Antwi, M.; Adnane, A.; Ahmad, F.; Hussain, R.; Rehman, M.H.; Kerrache, C.A. The case of HyperLedger Fabrc as a blockchain solution for healthcare applications. *Blockchain Res. Appl.* **2021**, *2*, 100012. [CrossRef]
28. Bhutta, M.N.M.; Khwaja, A.A.; Nadeem, A.; Ahmad, H.F.; Khan, M.K.; Hanif, M.A.; Song, H.; Alshamari, M.; Cao, Y. A Survey on Blockchain Technology: Evolution, Architecture and Security. *IEEE Access* **2021**, *9*, 61048–61073. [CrossRef]
29. Foschini, L.; Gavagna, A.; Martuscelli, G.; Montanari, R. Hyperledger Fabric Blockchain: Chaincode Performance Analysis. In Proceedings of the IEEE International Conference on Communications (ICC), Dublin, Ireland, 7–11 June 2022.
30. Thakare, A.; Kim, Y.G. Secure and Efficient Authentication Scheme in IoT Environments. *Appl. Sci.* **2021**, *11*, 1260. [CrossRef]
31. Bracciale, L.; Loreti, P.; Pisa, C.; Shahidi, A. Secure Path: Block-Chaining IoT Information for Continuous Authentication in Smart Spaces. *IoT* **2021**, *2*, 326–340. [CrossRef]
32. Ferreira, C.M.S.; Garrocho, C.T.B.; Oliveira, R.A.R.; Silva, J.S.; Cavalcanti, C.F.M.D.C. IoT Registration and Authentication in Smart City Applications with Blockchain. *Sensors* **2021**, *21*, 1323. [CrossRef]
33. Hameed, K.; Garg, S.; Amin, M.B.; Kang, B. A formally verified blockchain-based decentralized authentication scheme for the Internet of things. *J. Supercomput.* **2021**, *77*, 14461–14501. [CrossRef]
34. Narayanan, U.; Paul, V.; Joseph, S. Decentralized blockchain based authentication for secure data sharing in Cloud-IoT. *J. Ambient. Intell. Humaniz. Comput.* **2021**, *13*, 769–787. [CrossRef]
35. Khalid, U.; Asim, M.; Baker, T.; Hung, P.C.; Tariq, M.A.; Rafferty, L. A decentralized lightweight blockchain-based authentication mechanism for IoT systems. *Clust. Comput.* **2020**, *23*, 2067–2087. [CrossRef]
36. Algarni, S.; Eassa, F.; Almarhabi, K.; Almalaise A.; Albassam, E; Alsubhi, K.; Yamin, M. Blockchain-Based Secured Access Control in an IoT System. *Appl. Sci.* **2021**, *17*, 1772. [CrossRef]
37. Hao, J.; Liu, J.; Wang, H.; Liu, L.; Xian, M.; Shen, X. Efficient Attribute-Based Access Control With Authorized Search in Cloud Storage. *IEEE Access Secur. Priv. Cloud IoT* **2019**, *7*, 182772–182783. [CrossRef]
38. Malani, S.; Srinivas, J.; Das, A.K.; Srinathan, K.; Jo, M. Certificate-Based Anonymous Device Access Control Scheme for IoT Environment. *IEEE Internet Things J.* **2019**, *6*, 9762–9773. [CrossRef]
39. Ding, S.; Cao, J.; Li, C.; Fan, K.; Li, H. A Novel Attribute-Based Access Control Scheme Using Blockchain for IoT. *IEEE Access Secur. Priv. Cloud IoT* **2019**, *7*, 38431–38441. [CrossRef]

40. Sun, S.; Du, R.; Chen, S.; Li, W. Blockchain-Based IoT Access Control System: Towards Security, Lightweight, and Cross-Domain. *IEEE Acces* **2021**, *9*, 36868–36878. [CrossRef]
41. Liu, S.; Yu, J.; Xiao, Y.; Wan, Z.; Wang, S.; Yan, B. BC-SABE: Blockchain-aided Searchable Attribute-based Encryption for Cloud-IoT. *IEEE Internet Things J.* **2020**, *7*, 7851–7867. [CrossRef]
42. Xu, R.; Chen, Y.; Blasch, E. Decentralized Access Control for IoT Based on Blockchain and Smart Contract. In *Modeling and Design of Secure Internet of Things*; Wiley: Hoboken, NJ, USA, 2020; pp. 505–528.
43. Bhatt, S.; Sandhu, R. ABAC-CC: Attribute-Based Access Control and Communication Control for Internet of Things. In Proceedings of the 25th ACM Symposium on Access Control Models and Technologies, Barcelona, Spain, 10–12 June 2020; pp. 203–212.
44. Qin, X.; Huang, Y.; Yang, Z.; Li, X. LBAC: A lightweight blockchain-based access control scheme for the Internet of things. *Inf. Sci.* **2021**, *554*, 222–235. [CrossRef]
45. Liu, H.; Han, D.; Li, D. Fabric-iot: A Blockchain-Based Access Control System in IoT. *IEEE Access-Blockchain-Enabled Trust. Syst.* **2020**, *8*, 18207–18218. [CrossRef]
46. Pajooh, H.; Rashid, M.; Alam, F.; Demidenko, S. Hyperledger Fabric Blockchain for Securing the Edge Internet of Things. *Sensors* **2021**, *21*, 359. [CrossRef]
47. Zhang, Y.; Li, B.; Liu, B.; Wu, J.; Wang, Y.; Yang, X. An Attribute-Based Collaborative Access Control Scheme Using blockchain for IoT Devices. *Electronics* **2020**, *9*, 285. [CrossRef]
48. Bogdanov, A.; Knudsen, L.R.; Leander, G.; Paar, C.; Poschmann, A.; Robshaw, M.J.; Seurin, Y.; Vikkelsoe, C. PRESENT: An ultra-lightweight block cipher. In I*nternational Workshop on Cryptographic Hardware and Embedded Systems*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 450–466.
49. Aumasson, J.P.; Henzen, L.; Meier, W.; Naya-Plasencia, M. QUARK: A Lightweight Hash. *J. Cryptogr.* **2013**, *26*, 313–339. [CrossRef]