Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 8621230, 12 pages
https://doi.org/10.1155/2021/8621230

Research Article

Unified Authentication and Access Control for Future Mobile Communication-Based Lightweight IoT Systems Using Blockchain

Shubham Joshi,[1] Shalini Stalin,[2] Prashant Kumar Shukla,[3] Piyush Kumar Shukla,[4] Ruby Bhatt,[5] Rajan Singh Bhadoria,[6] and Basant Tiwari[7]

[1]*Department of Computer Engineering, SVKM'S NMIMS MPSTME Shirpur, Maharashtra, India 425405*
[2]*Department of Information Technology, Indian Institute of Information Technology (IIIT), Maulana Azad National Institute of Technology (MANIT) Campus, Bhopal, Madhya Pradesh 462003, India*
[3]*Department of CSE, Koneru Lakshmaiah Education Foundation (KLEF) Vaddeswaram, KL University Vijayawada, Andhra Pradesh-520002, India*
[4]*Computer Science & Engineering Department, University Institute of Technology, Rajiv Gandhi Proudyogiki Vishwavidyalaya, (Technological University of Madhya Pradesh), Bhopal 462033, India*
[5]*Department of Computer Science, Medicaps University, Indore, M.P., India*
[6]*Department of Computer Application, Technocrats Institute of Technology, RGPV, Bhopal, MP, India*
[7]*Hawassa University, Awasa, Ethiopia*

Correspondence should be addressed to Basant Tiwari; basanttiw@hu.edu.et

Received 5 October 2021; Accepted 22 November 2021; Published 17 December 2021

Academic Editor: Deepak Kumar Jain

The Internet of Things (IoT) is a new revolution defined by heterogeneous devices made up of intelligent, omnipresent items that are all hooked up to The internet. These devices are frequently implemented in different areas to offer innovative programs in various industrial applications, including intelligent urban, medicine, and societies. Such Internet of Things (IoT) equipment generates a large volume of private and safety information. Because IoT systems are resource-constrained in terms of operation, memory, and communication capability, safeguarding accessibility to them is a difficult task. In the blockchain concept, the majority, or even all network nodes, check the validity and accuracy of exchanged data before accepting and recording it, whether this data is related to financial transactions, measurements of a sensor, or an authentication message. In evaluating the validity of exchanged data, nodes must reach a consensus in order to perform a special action, in which case the opportunity to enter and record transactions and unreliable interactions with the system is significantly reduced. Recently, in order to share and access management of IoT devices' information with a distributed attitude, a new authentication protocol based on blockchain has been proposed, and it is claimed that this protocol satisfies user privacy while preserving security. Today's identification and authentication techniques have substantial shortcomings due to rapidly growing prevalence and implementation. As a result, the protection of such gadgets is critical to guarantee the program's efficacy and safety. A decentralized authentication and access control method for lightweight IoT systems are proposed in this work and a blockchain-based system that enables identification and secures messaging with IoT nodes. The technique is built on fog information systems and the idea of a blockchain system; when contrasted to something like a blockchain-based verification system, the testing findings show that the suggested mechanism outperforms it. The authentication and verification system undergoes using the blockchain technique. Our method takes advantage of blockchain's inherent advantages while also associated with development authentication systems. Our suggested blockchain-based approach, structure, and layout, in particular, provide for transparency, consistency, and provenance while also providing tamper-proof records. The article describes the general systems architectural style and the analysis and execution of a real scenario as just a prototype system. The authentication included give as protected prototype that can transmit data with secured protocol and achieves minimum error rate.

1. Introduction

IoT in many industries and formats may be installed everywhere. Such gadgets can talk with each other, acquire, share, and process data to provide a service [1]. Analysts at CISCO, Ericsson, as well as other organizations predict that by 2020, there will be more than 45 billion devices that are interconnected [2]. IoT is being used in a variety of industries, including household appliances, hospital instruments, and personal accessories. These gadgets must have particular qualities to enable such capability. They ought to be able to operate on a moderate energy basis and communicate with some other heterogeneous systems. They must also be capable of maintaining a steady connection with the back-end, if only one exists, and obtain updates as needed. The authentication method is a critical idea for managing system resources and connections in a secure manner. These categories should be reinterpreted in the IoT context compared to the previously indicated features. The issue of restricted resources must be considered in identification systems and authorization rules. According to a previous report issued by Gartner, the volume of interconnected devices could reach Twenty billion in 2019 [3]. IoT is already present in practically every aspect of life (for example, health and transportation), and numerous IoT programs simplify everyday tasks (i.e., home automation). Integrated recyclers, transportation infrastructure, smart grid, smart transportation, environmental sensing, traffic control, and a variety of other technologies are examples of these systems [4]. IoT systems produce a large amount of information, some of which could be sensitive. In intelligent medical systems, for instance, patient-attached devices create confidential information such as personal medical status [5, 6]. This information is then relayed to the clinic, where it is regularly checked to activate sirens in the case of emergencies. As a result, the privacy of this equipment, as well as the sensed data, is critical to ensuring the IoT program's natural behavior, since all of the IoT program's major decisions are dependent upon the collected data [7]. If a rogue device gains access to a Network infrastructure, it can destabilize the program's regular function, resulting in severe consequences. Data processing [8], secrecy [9], authenticity [10], accessibility [11], and nonrepudiation [12] all seem to be aspects of IoT security. The authentication method, on the other hand, will be the first line of protection, limiting information exchange to those with appropriate rights. To preserve data integrity and security, secured IoT applications require an authentication process among IoT devices as well as other platforms. If not, these platforms would be subject to a range of security issues, including unauthorized access, theft of data, and information modification [13]. Blockchain technology may be divided into two categories, namely, permission less and permissioned (see below). A permissionless blockchain, often known as a public blockchain, is accessible to anybody who wants to use it. Despite the fact that it has enormous potential, similar to Bitcoin, it may not be suited for company owners that want to maintain control over the transaction processing system. Business processes may have special criteria and complicated procedures that necessitate the use of customizable solutions that limit the participation of outsiders in such processes. Aside from these issues, permissionless blockchain has other difficulties, including as scalability, regulatory institutions, and control over evolution. This has offered corporations the opportunity to investigate other possibilities, such as permissioned blockchain, which may be managed privately and can limit membership in the blockchain network to only those who are known and trusted. A permissioned blockchain, sometimes known as a private blockchain, is a kind of blockchain that has been granted permissions. This will completely change the way transactions are carried out in the future.

Bitcoin's fundamental technique is known as blockchain technology [14]. A developing network of information could be characterized as it. The blockchain inherited effective properties by construction, such as decentralized, tamper-proof blocks containing information that may be viewed by every node equitably. This notion can be applied to any application that necessitates the validation of information or activities by a trustworthy 3rd person. The blockchain enabled all confidence to be transferred network by replacing the trusted third party with such an accessible, unmanipulated block of information that is accessible in a distributed form. The smart contract is indeed an efficient solution that makes use of blockchain technology. An auto or self-executing software had first been described as just a smart contract in 1996. The Ethereum blockchain has functionality such as activities and records. A response (returned value) first from the smart card towards the user interface that interacts with this is called an event. The basic purpose of using events and records is to make it easier for contracts and the programmed that interact with them just to communicate.

Blockchains features, such as increased dependability, the integrity of the information, and flexibility, make it an excellent solution for identification issues. Smart contracts, which provide fine-grained network access over IoT systems, can also be integrated with blockchain. Furthermore, fog computing as well as blockchain-based provides solid foundations for developing and managing decentralized confidence and safety solutions for time-critical fog-enabled IoT networks [15]. Likewise, the researchers of [16] demonstrated a successful and powerful collaborative fog-based IoT network platform. We suggest a delay-sensitive blockchain-enabled security authentication methodology for IoT networks, based on the properties of fog computing as well as the decentralized nature of blockchain. The following are indeed the article's main factors that contribute:

(1) A revolutionary decentralized technique that enables identification and security systems over IoT applications, allowing them to operate in a safe and trustworthy atmosphere

(2) A proof-of-concept of the suggested technique demonstrates its capability to handle IoT security goals

(3) Evaluation of the suggested mechanism's effectiveness against a state-of-the-art IoT biometric identification

2. Related Works

The Ethereum blockchain has functionality such as events and logs. A response (returned data) from the contract to a user experience that interacts with it is called an event. The basic purpose of using events and logs is to make it easier for contracts as well as the programmed that interact with them to communicate. Figure 1 depicts an example Ethereum Smart Contract Application environment. The customer first asks the contract for accessibility to a certain commodity or commodity. Secondly, the software system determines whether the asset is available for use and then collects the payment first from the customer. In this case, the customer was paying with Ethereum, a digital currency. Finally, the contract keeps the resources reserved for the present client. Fourth, the customer makes appropriate use of the service. Afterward, assuming all of the contractual terms were followed, the smart contract would bill the client like promised. It is vital to remember that perhaps the contract is completely self-contained, and the proprietor is not participating in any of the 4 stages [17].

(1) The customer demands that the smart contract provide him access to a certain resource or item

(2) The smart contract checks to see whether the asset is available and then records the fee received from the customer. In this particular example, the customer is making a payment using the cryptocurrency Ethereum

(3) The smart contract ensures that the resource is only available to the present client

(4) The customer makes use of the resource in accordance with its approval

(5) If everything went according to the contract conditions, the smart contract will charge the customer the amount agreed upon by the parties

2.1. Blockchain. A distributed database, or blockchain, is a historical reminder of all activities processed and reviewed in the community. The blockchain is indeed a peer-to-peer technology that is decentralized in character. Every network device where blockchain has been used keeps a complete transaction record. Such blockchains refresh in real-time as each transaction is validated [18]. The blockchain is created to use as a bank transaction technology, but it was first implemented in Bitcoin. This unforgeability, decentralized structure, and fault-tolerant, on the other hand, make it suited for the cyber defense environment. There are now various security methods [19] that leverage blockchain-based that can provide fundamental security criteria to safeguard a service, like access controls. The number of blocks linked along with a hashing technique is recorded in the logbook. Every block is split into 2 parts, one of which presents the number of completed and verified operations. A medical chart, a money activity, or a communication systems signal are all examples of transactions. Various data formats are used to organize such systems. The reversed hashing technique is employed, for instance, in Merkel's tree structure, with the center core hashing kept as the block identifier [20]. The data block is indeed the second element of the blocks, and it stores packet headers such as the transactional date stamp, block hashes, and previous block hash. As just a result, a hash-supported network is formed from a collection of existing blocks. The chain grows increasingly resistant to falsification because it grows longer. Furthermore, because all following blocks are connected via hashing, if a malevolent user wants to change or modify the operations of a block, then she must perform the same modifications throughout all blocks. Figure 2 depicts the blockchain's general premise as it is deployed on the internet. In the blockchain, there are primarily types of nodes. The first sort of node is referred to as an inactive station or verifying node; so, it is responsible for holding and receiving block data, but it cannot develop new blocks or initiate transactions. The mining network is the second sort of network, and it may also build blocks and verify transactions. A multitude of consensus techniques was employed to verify new blocks and tie them towards the original network.

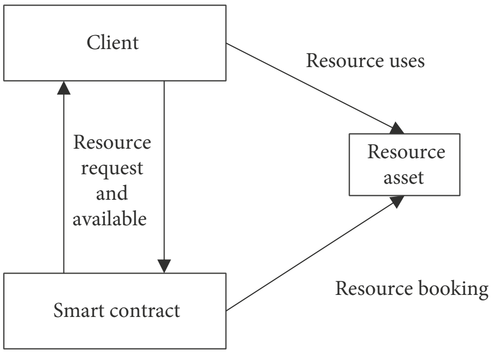


Figure 1: A sample Ethereum smart contract scenario.

The consensus method allows nodes inside the blockchain system to agree upon the addition of a new block to a chain. Proof-of-work (PoW) is among the Bitcoin network's consensus mechanisms. A mathematics riddle is introduced to the PoW method, that must be answered by mining nodes for a block to be validated. The puzzle's complexity could be adjusted based on the mining node's computational resources as well as the time required to verify new blocks. The PoW technique is utilized in contexts when computing power is not a constraint [21]. One of the key motivations for nodes to join as minors is that miners were compensated for contributing new transactions. Other consensual methods include the Proof-of-stack (PoS) method, Byzantine fault tolerance method [22], and the ripple method, in addition to PoW. The PoS methodology is suggested to resolve the constraints of the PoW technique. There are nodes termed forgers in the PoS process that validate new blocks. The forgers are chosen based on their current account and the amount of money they are willing to put in a stack. A staked node has a better likelihood of validating a block and adding this to the genesis blockchain. Ripple uses the XRP Ledger Consensus Protocol to reach networks consensus. Each node inside the ripple network maintains a Unique Node Listing (UNL), which is used in

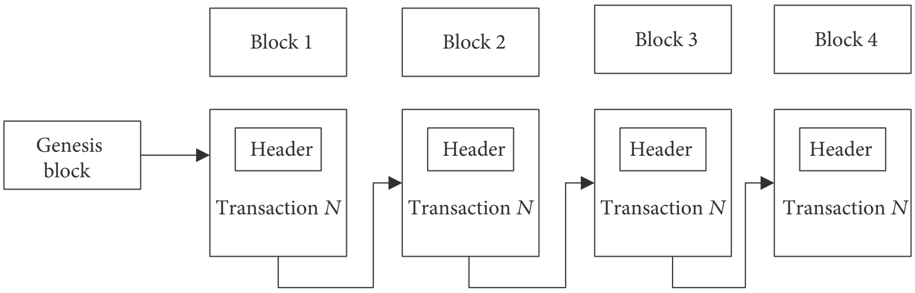


Figure 2: Genesis block flow.

this technique. A data model called Candidates Set is used to batch all operations. To be included in the ledger, every transaction in the candidate set must receive 80 percent of the votes [23]. The protocol is low in energy consumption. It is, however, widely regarded as centralized because it involves a vote from the subset of nodes.

2.2. IoT Authentication Traditional Models. A model for establishing trust in the identity of IoT machines [24] and devices in order to protect data and control access when information travels over an unsecured network, such as the Internet, is called Internet of Things (IoT) authentication. In order for linked IoT devices and equipment to be trusted to defend against control orders from unauthorized people or devices, strong IoT authentication is required. Strong IoT authentication is required. A second benefit of authentication is that it prevents attackers from posing as IoT devices in the goal of gaining access to data stored on servers such as recorded conversations, photos, and other potentially sensitive information.

A basic way would be to have a login and password to log into every device. Since each registered person's responsibilities and privileges are established and saved on the devices mostly by the administrator, this approach provides sufficient security controls (owner). This solution, unfortunately, generates significant expense but does not scalable because the client must identify every system separately [25]. Traditional IoT devices, such as IP cameras and Internet-connected home utilities, use this technology. A much more sophisticated alternative is to use single-sign-on technologies for authentication. Whenever OAuth2 is used as an authentication process, for example, individuals attempt to access devices by logging in to a trustworthy OAuth2 supplier. Google, Facebook, and other reputable third parties can be used [26]. The trustworthy entity allows access if they authenticate properly and then have the proper authorization. By identifying the trusted source, all devices administered by the same person can be accessible. Secondly, the client authorizes the application to interact with the authentication server, also known as the OAuth2 providers shown in Figure 3.

Because the user can access numerous entities by identifying a single entity, this method saves time and effort. In addition, the OAuth2 supplier is usually a reputable third party, making integration of such a system easier. At the same time, relying on a centralized organization increases the risk of one-time failures, which jeopardizes the reliability of the existing approach. Furthermore, if a user account or a central entity is hacked, then the whole system is hacked. Phishing, which has a high rate of success, is an important attack method that could cause this approach to fail [27]. Furthermore, spear-phishing efforts are becoming more intense, accurate, and intelligent in recent years, with the potential to fool even the most informed individuals.

3. Problem Definition

3.1. Authentication. The foundation of security in the Internet of Things is guaranteeing the authenticity of a device's identification when it connects to the network [28]. Authentication is a network method for determining if a user has access to specific resources. There are three types of authentications: Knowledge, Ownership, and Rights.

3.2. Safety. Maintaining the stability of IoT devices ensures that the IoT is secure. It is still vulnerable to attacks from hostile users while performing the work due to software or system faults. To prepare for further penetration, the attacker will usually change the network entity to escape from a back door into the device and edit the device key configuration file [29], which interferes with the entire network. We monitor changes to vital data regularly to detect potential violations as soon as possible.

3.3. Presumptions

3.3.1. Verification of Registration. Use permit channels where the network administrator manages the authorization permissions. Anything that wants to connect to the network must first register in the blockchain. The permissions channel access control layer only allows devices with legal identifiers to store information in the blockchain [30].

3.3.2. Safe Route of Communication. Consider a safe connection to avoid intermediate attacks. As a result, no one else can intercept or change the messages. The accuracy of the data is the primary purpose of such a secure communication route [31]. The nodes can communicate with each other and verify the details precisely.

With the fast growth of voice control technology, improving the accuracy of speech recognition in many Internet of Things sectors has proven to be an intractable challenge to tackle. Because there are many different

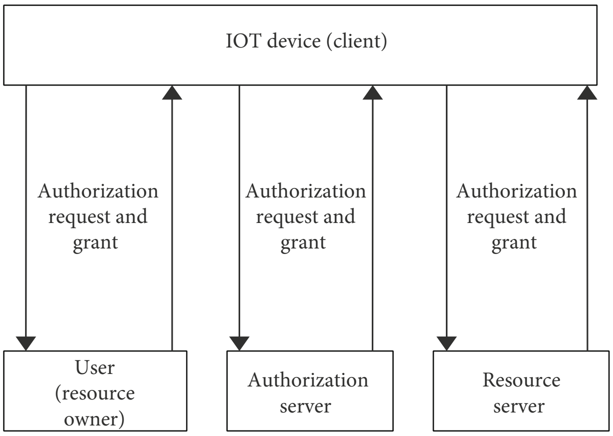

IOT device (client)
Authorization request and grant
Authorization request and grant
Authorization request and grant
User (resource owner)
Authorization server
Resource server

Figure 3: An overview of the proposed approach movement.

conversation situations, determining the context of a dialogue scene is a critical challenge for voice control systems to address. The fact, on the other hand, is that the amount of training data available for dialogue systems is always inadequate. In this study, we primarily address the issue of data scarcity in dialogue systems via the use of data augmentation techniques [32].

The example is carried out using collection of datasets from the Los Alamos National Laboratory enterprise network was used to gather network and computer (host) events. The Unified Host and Network Dataset was created from this collection of network and computer (host) events. The sample networks and attacks are collected from the dataset of unified Host and Network dataset [33].

3.3.3. Organization within a Short Timeframe. When a function is inserted in the blockchain, simply request the ledger of a less quantity of knots instead of a more quantity of knots. As the timing nodes are randomly selected, it can be assumed that they are reliable.

4. Proposed System Architecture and Design

This paper has presented a blockchain solution with a unique system design. This fills the gaps in current solutions. It should also be portable, unlike the block stack, and work-over any network with minimum dependencies. It is designed for IoT devices with low computing capacity. It also proposes the idea to implement OAuth with a smart contract that allows users to connect once and control all approved devices without having to check-in for each IoT device separately. Smart contracts can also be executed by IoT equipment, allowing them to become self-sufficient. It will go through a series of testing on a working prototype, as well as the outcomes of those tests. The testing will include performance tests as well as attacks targeted against them. The use of Ethereum as the basis for this solution has several advantages. Ethereum has a strong development structure in place, as well as a built-in incentive for minors to help with hashing problems. Besides, the Ethereum lite client protocol can be used on IoT devices with little processing power and memory, which is necessary for the proposed solution. The process of verification is followed by authentication phase, verification phase, and security phase. When the user sends the request to the authentication, the request undergoes various processing under smart contract, and the security is checked at the security phase. Later, the verification is successful the person enable to access the data.

4.1. Assumptions. Here is a list of steps to implement such a solution.

(i) One or more IoT devices are available to the user

(ii) The user's private key to the Ethereum key store has not been hacked

(iii) The user belongs to the Ethereum network

(iv) Ethereum blockchain links the user to the IoT device

(v) User to implement their smart contract

The system's total capabilities allow it to change the ultimate hypothesis. Building centralized smart contracts that authenticate users on individual IoT devices is conceivable. One of the goals of this section, however, is to avoid relying on a single source of data. Users should be encouraged to create their intelligent contracts so that they can exercise full control over their systems. Use an authorization route in which the permission privileges are managed by the network manager.

4.2. Architecture of the System. The phases of the authentication procedure are described in the message sequencing diagram in Figure 4.

(1) The user authenticates his Ethereum wallet address to the smart contract. The input samples are taken in real time from the client side during the access of data

(2) If the client is genuine, the smart contract provides the receiver an access token as well as the shipper's Ethereum address. The intelligent contract data is accompanied by the client and the IoT tools

(3) The user assembles a package that comprises information such as the user's IP address, Ethereum public key, token access, and duration. The Ethereum private key is used to sign this package and is then sent with the relevant public key. If required, the package will be encrypted, and then there is no need for the protocol to work. Then it is the integrity of the message that matters; so, it is signed

(4) When the delivery is received, the IoT gadget controls its contents. If successful, the device allows the user to access from the IP address of the sender for the period provided. Otherwise, the request is refused if one of these tests fails

4.3. Security in Authentication Phase. This phase is categorized by the Server Processing Unit (SPU). The input data r_m from the m-th input user is selected who needs the

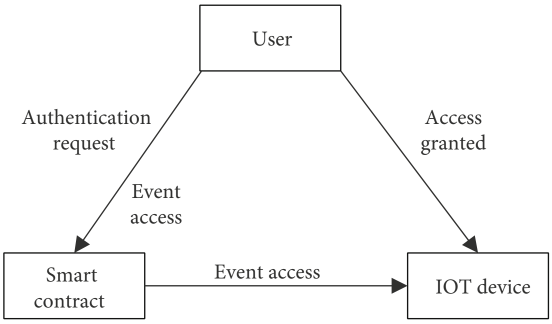


Figure 4: The second solution authentication scenario.

```
A sends connection request (ID_A, M_signed_by_A) to B:
//B query Key_A and verify the identity of A
If (Key_A_exist_in_local)
Verify the identity of A;
Else
If (Key_A_exists_in_consensus_nodes)
Verify the identity of A;
Else
Reject the connection request of A;
```

Algorithm 1: Process for authenticating P2P identity.

permission to access. The particular id of the user tends to be IDr$_m$ and belongs to the group of users with the id of GID$_m$.

The input Processing unit sends the signal to the Iot device using the verification code with the details of (IDr$_{m,}$ GID$_{m\ ,}$ r_{m}) to verify the authentication using smart contract and give the authentication access to the verified user. The 48 connections between the reader and the SPU take place via a secure channel and are accessible from a single location.

The computation of input is carried out as represented in equation (1)

$$M_h = (\mathrm{IDrm} \oplus \mathrm{GIDm} + \mathrm{rm}). \tag{1}$$

The input broadcasted through the processing unit. The condition that authenticate the input user to the IoT devices is represented as the following equation (2):

$$\sum T_j \, 1 < j < m, \tag{2}$$

where equations (3) and (4)represent the tagging smart contract input to verify the authentication process using blockchain technology.

$$\mathrm{NTj} = (\mathrm{Mh} - (\mathrm{GIDm} \oplus \mathrm{IDrm}))\mathrm{VrTj}, \tag{3}$$

$$QT_j = e_i \oplus (\mathrm{Sg\,VrTj}), \tag{4}$$

where e_i represents the sequence number of Ti, and Sg represents the group secret. Rm receives (NTj, QTj, rTj) from the tag T_j, thanks to the reader Rm. The reader can only extract ei if and only if NTj is legitimate, in which case $e_i = \mathrm{QTj}\,(\mathrm{Sg\,rTj})$.

Similarly, it accumulates the information from all of the tags associated with the group until the timer expires or the group is deleted. The reader then delivers the information (IDr$_{m,}$ GID$_{m,}r_m$, e1, e2,..., ej) to the SPU through a secure communication channel. This is done by comparing the received information and responses (IDr$_{m,}$ GID$_{m,}r_m$, Re1, Re2, Re3), where re1, Re2, and Re3 are pseudorandom numbers that are created for each tag individually. The reader maintains a record of critical information required for authentication in its memory.

In order to connect with the initial tag, the reader establishes a Temp ID for each tag in the group. The reader Rm computes $r'\mathrm{T1} = \mathrm{rT1\,re1}$ and $\mathrm{TempIDT1} = (\mathrm{GIDj\,e1})$ $(r'$ T1 IDRm) using the formulas $r'\mathrm{T1} = \mathrm{rT1\,re1}$ and TempIDT1. Group g receives (QT1, TempIDT1, First, re1) from the reader and is assigned to the first tag in the group. In addition, the tag computes TempIDT1 and compares it to the one that was received. $\mathrm{TempIDT1} = (\mathrm{re1} + \mathrm{ST1})$ and $\mathrm{CT1} = \Big(((\mathrm{e1}\,r'\,\mathrm{T1})\,\mathrm{NT1})$ are computed by the tag and sent to the reader in the form of (TempIDT1, CT1, ZGIDg) to the reader. As soon as CT1 is determined to be genuine, the reader transmits the following tags to the next tag in the group: (QT2, TempoIDT2, ZGIDg, re2). Reader has private secret knowledge about all of the active tags after obtaining the specifics of all of the tags in the group or after the time out.

4.4. Verification Phase of the System. In order to activate tags, the reader provides the following information to BPS: (IDr$_m$, GID$_m$, r_m, e1,...eq), where e1,...eq are the sequence numbers of the active tags. If certain tags are successfully validated, the presence of the object is confirmed. With this information, BPS is notified of any tags that are not verified successfully.

4.5. Input Attack. Algorithm 1 explains how to extract the group secret Sg from the given parameters and how to do it effectively. Assume that the attacker has obtained a copy of all of the communications sent between the tags and a reading device. Let r_m and rTj be the bit vectors of length l for the reader and the I-th tag, respectively, where $1jn$ is the number of readers and tags. The group secret Sg has a length of l characters, and $\mathrm{NTj} = \mathrm{Sg\,rRm\,rTj}$.

The attack rate may happen based on the input criteria shown in equation (5):

$$\frac{Tj \lhd * C_1}{T \lhd C_0}. \tag{5}$$

Using the recognizability rule R_1 in conjunction with the assumption T_j,

$$\frac{T\mathrm{j} \equiv \phi(r_1), T э\, \mathrm{D1ID2}}{T\mathrm{j} \equiv \phi(h(D1ID2 \| r_1))} \tag{6}$$

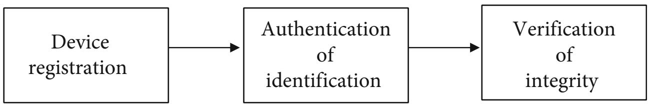


Figure 5: System workflow.

Freshness rule is explained in equation (7):

$$\frac{Tj| = \#(r_{\mathrm{m}})}{Tj|\equiv\#(h(D1ID2\|r_1))}. \tag{7}$$

As long as T thinks that r_0 is new, tag has every right to believe that any formula in which r_0 is a component is likewise new. As a result, G_1 is accomplished.

The aim G_2 is shown to be effective in ensuring that the reader feels that the tag communicated M_1 and is current. Using the assumptions AR5 and the possession rule P_1, this may be shown in a manner similar to that of the objective G_1 . As a bonus, this assures that the message M_1 did not come from the reader. In order to demonstrate G_3, we must first apply P_2.

$$\frac{R1 \ni C_5, R1 \ni RIDm, R϶\, r_1}{R϶\,\mathrm{OID}}. \tag{8}$$

Using the parser output M_3 and the previous step result, we apply P_1 and obtain

$$\frac{R1 \lhd M_4}{R1_{>} T_{\mathrm{idm}}} \tag{9}$$

Applying R_2, we get

$$\frac{CS|\equiv\phi(OID1 \oplus S_{Rm}), CS_{϶} S_{Rm}}{CS1|\equiv\phi(OID1)}. \tag{10}$$

The cloud server authenticates the reader and retrieves the OID, which is subsequently used to get more information from the database.

5. Model of the System

Blockchain can accommodate complex and evolving conditions as an accessible, secure, and decentralized consensus system for transactions. The stability of the application was not impaired by the breakdown of some units. Malware nodes cannot infiltrate the network through distributed authentication. The registry will not be modified even if a few nodes are hacked.

Whenever a new device is added to a multinode network, the device credentials must be stored in the blockchain. For each device, the blockchain ledger contains the user's IP address, Ethereum public key, token access, duration, and other data. There are three steps for system operation. All equipment must follow the blockchain registration process before it can be authenticated.

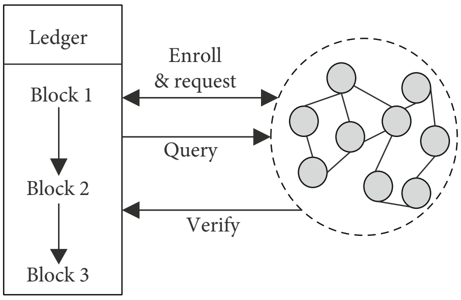


Figure 6: System model.

Table 1: Node events and careers.

Careers	Activities
Nodes of agreement	Blocks creation, blocks checking, consensus process participation
Nodes of nonconsensus	Data transfer

If a device needs to join the network, the information recorded in the blockchain will be used to authenticate it. Following authentication, the integrity of a device's vital information hash will be checked to detect possible intrusion activities. Figure 5 shows how our system works, while Figure 6 shows the system model.

5.1. Node Functions. Based on the different goals of the permissions chain, nodes can be classified as consensual or nonconsensual. Consensus nodes take part in the consensus process, generate blocks, and send them to nodes that disagree. Table 1 shows the responsibilities of the two types of nodes.

5.2. Registration of Appliances. All nodes in the Internet of Things must be recorded using the blockchain. Each device generates a pair of keys based on its security key module. The private information is stored on the device and encryption, while the public key is stored on a blockchain ledger. The consensus nodes treat data information as a recording occurrence, resulting in the formation of blocks. During the registration system, it must preserve the hash value of vital data in the blockchain, such as the local configuration file and firmware, to prepare for eventual data security certification.

Blockchain-based solutions may offer tamper-proof records as well as decentralization, which can be used to augment existing methods of recording. This paper-based smart contract-based approach may be utilized for the authentication and access control of Internet of Things devices. As a result of being created and executed in real-world circumstances utilizing readily accessible devices and technology, the solution has the advantage of being readily deployable as required. The technique was effective in allowing genuine users to access their IOT devices once they were validated. Aside from that, it was impervious to well-constructed

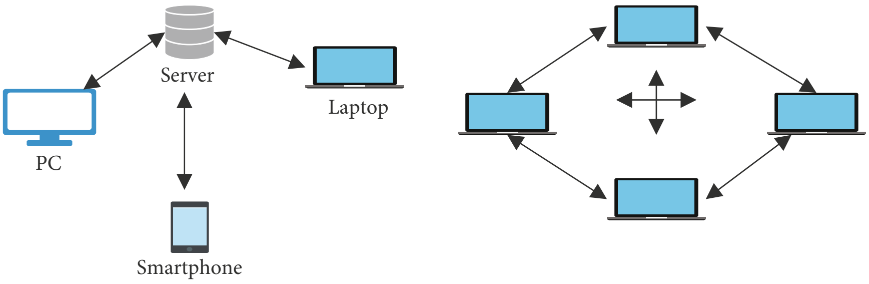


Figure 7: The proposed architecture of blockchain system structure.

assaults that attempted to hijack genuine sessions and brute force credentials.

5.3. A Technique of Peer-to-Peer ID Verification. Assuming that *A* sends a request to connect to *B* on a network, which includes the *M* message, the authentication of the identifier is as follows.

When *B* receives the message, it uses *A*'s ID to look for *A* 's public key in the local blockchain ledger. If *A*'s public key is discovered in a local blockchain ledger, it can be used to verify *A*'s identity. *B*, on the other hand, looks for consensus nodes for *A*'s public key. If *A*'s public key is obtained, *B* certifies *A*'s identity. Otherwise, *A* is not a member of the blockchain system, and *B* will decline *A*'s connection request. The P2P authentication technique is shown in Algorithm 1.

The hash of critical information from every equipment device was preserved in the blockchain during the logging procedure. While performing a task, IoT nodes transmit a critical data integrity check request to the nearest unit. If the verification fails, a critical configuration file has been changed, and a warning has been given. After hashing the data, the information obtained and log records made by the tools during the process of the work can be accompanied to blockchain for protection and security auditing. Aside from firmware and IoT device configuration, the files are important.

6. Implementation

6.1. Environmental Deployment. We decided to use the IoT application script for authorization. Using the open-source Hyperledger Fabric program, we created a blockchain network using the Raspberry Pi. Every Raspberry Pi joins the blockchain network as a node, with units joining in a random sequence.

Depending on the multichannel and route technologies, blockchain may be separated into various subnets, and IoT units can create a variety of subnets based on business needs. When a subnet interacts with another subnet, there are no distractions. The topology of a blockchain system is seen in Figure 7.

The blockchain is a distributed ledger that keeps all transactions across the whole blockchain network. The data arrangement of blocks is the same as that of Bitcoin. Transactions, on the other hand, are events like equipment registration, authorization, and identity checks. The data arrangement of the block is shown in Figure 8.

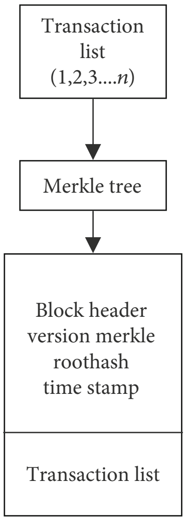

Transaction list (1,2,3....n)
Merkle tree
Block header version merkle roothash time stamp
Transaction list

Figure 8: Block data layout.

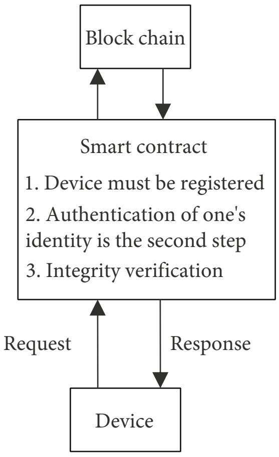

Block chain
Smart contract
1. Device must be registered
2. Authentication of one's identity is the second step
3. Integrity verification
Request
Response
Device

Figure 9: The interaction between devices and blockchain.

6.2. The Chain's Transactions. Transactional operations serve as the link between peripherals and blockchain. 3 varieties of transactions were identified by smart and intelligent contracts. Intelligent contracts take requests from devices and respond by doing various blockchain actions, like reading and writing, in response to those needs. The relationship inbetween the equipment and the blockchain is depicted in Figure 9.

6.3. Key Generation Technique Based on CRG. Each IoT device consists of a pair of keys: a private key and a public key. The two keys are used to identify the device. The public key can be created using the elliptical multiplying curve and a private key is a random number.

Finding an entropy generator that is secure and reliable is the most essential stage in generating the private key. A cryptographically pseudointellectual number generator is commonly used as a randomized resource to guarantee that the production of selected random data is unpredictable or nonrepetitive. CRG constructs cryptographic hash functions using a pseudorandom number generation. CRG constructs secure cryptography pseudorandom numbers including additional pseudorandom capabilities, as opposed to statistics and smaller pseudorandom numbers generators. We collect various IoT device information, such as storage utilization status, disc space available, I/O, the amount of operation, and CPU speed, to generate an estimated random seed in our system.

The public key may be calculated using the private key thanks to the elliptical curve steps. The equation $K = k \times G$ is irreversible. Here, k is indeed the private key, G seems to be the constant point of the generating point, and K is the public key of the generating point. If the public key K is established, determining the correct key by vigorously testing including all possible values to obtain the private key k is exceedingly tough. Figure 10 shows how to create the keys. The information's security is verified by the blockchain.

To sign a file, blockchain uses distributed ledger technology rather than traditional file signatures. Data falsification and theft are nearly impossible. The file storage system within the blockchain is depicted in Figure 11. Files are first saved as hash values, which are then added together to form the hash. This procedure is carried out until a root hash value is discovered. Merkle Tree is the name given to the generated database table (hash tree). To sign a file, the hashed route from every file towards the root hashes is needed.

It is only necessary to overwrite the hash record in the signature route when checking the integrity of a file. After then, compare the new hash value to the original hash.

6.4. Performance Analysis. Throughput and latency are system performance indicators that are mostly dependent on the Hyper ledger Fabric blockchain platform; hence, these details will not be detailed here.

The PBFT (Practical Byzantine fault tolerance) provides for the detection of anomalous behavior and the synchronization of data in the ledger to achieve blockchain network coherence. The strength of PBFT is critical to the safety of our system. The number of miners (offending nodes) in a Byzantine fault tolerance system with n nodes is t, as long as $n > 3t$. The arrangement will expire in a certain amount of time, and the loyalty party (honest nodes) will finally reach an agreement shown in Figure 12.

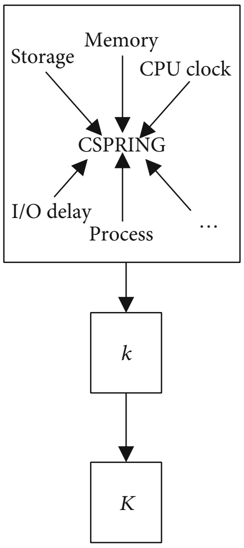


Figure 10: Key generation process.

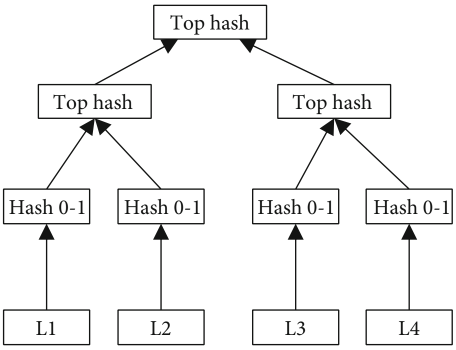


Figure 11: File storage structure in the blockchain.

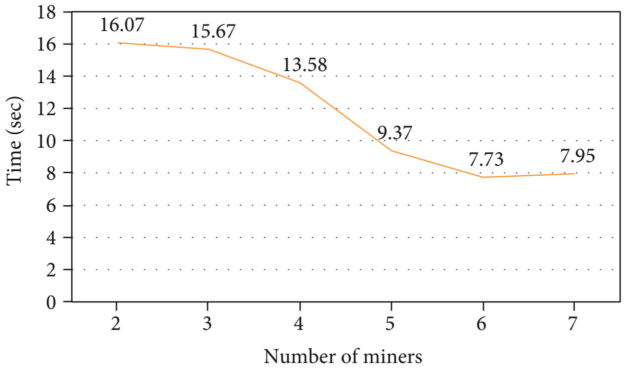
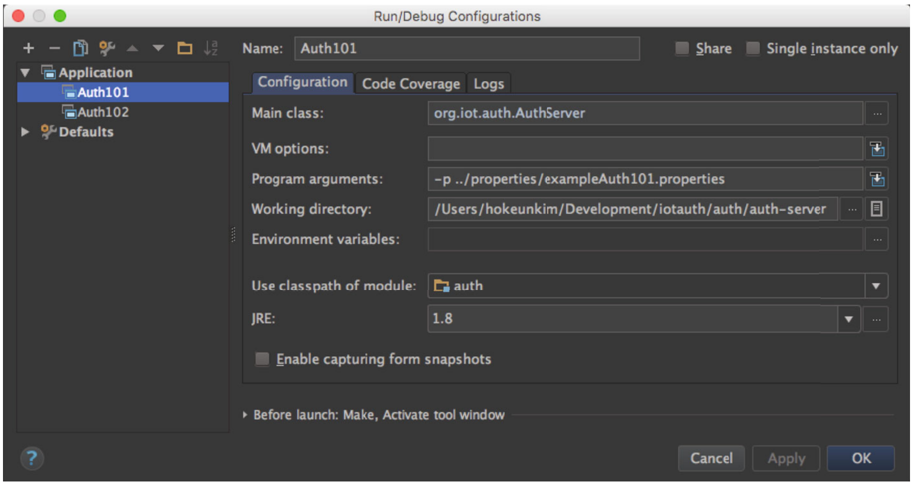


Figure 12: Prevent the malicious code depends on time.

The ideal situation was evaluated initially, once the solution prototype had been successfully tested. With the help of his or her MIST Ethereum client, an authorized user invokes the smart contract function: login admin. The smart contract delivers the authentication token as well as the user's Ethereum address to both the user and the IOT device at the same time, saving time and effort. In the test, the first step was finished in less than 4 seconds on a private blockchain, according to the results. The user then establishes a connection with the IOT device by submitting the authentication package.



Run/Debug Configurations

- Application
  - Auth101
  - Auth102
- Defaults

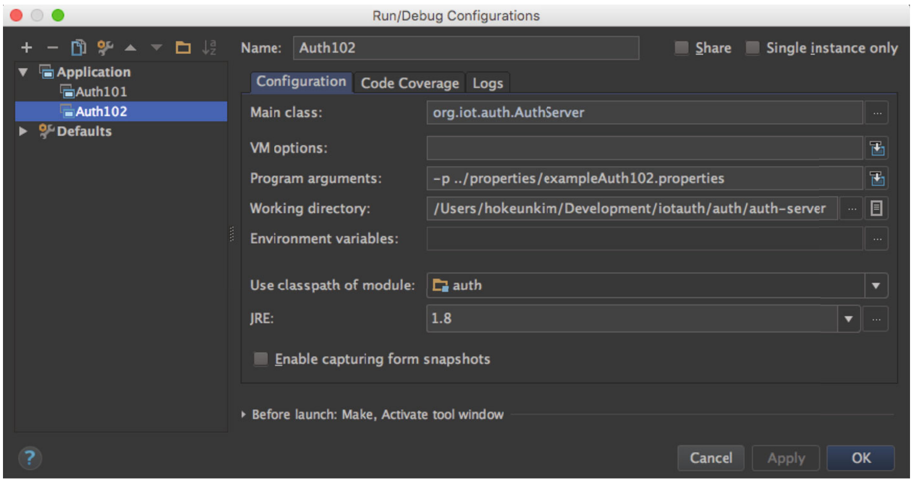
Name: Auth101
□ Share □ Single instance only

Configuration | Code Coverage | Logs

Main class: org.iot.auth.AuthServer
VM options:
Program arguments: -p ../properties/exampleAuth101.properties
Working directory: /Users/hokeunkim/Development/iotauth/auth/auth-server
Environment variables:
Use classpath of module: auth
JRE: 1.8
□ Enable capturing form snapshots

Before launch: Make, Activate tool window

Figure 13: Input front end authentication.



Run/Debug Configurations

- Application
  - Auth101
  - Auth102
- Defaults

Name: Auth102
□ Share □ Single instance only

Configuration | Code Coverage | Logs

Main class: org.iot.auth.AuthServer
VM options:
Program arguments: -p ../properties/exampleAuth102.properties
Working directory: /Users/hokeunkim/Development/iotauth/auth/auth-server
Environment variables:
Use classpath of module: auth
JRE: 1.8
□ Enable capturing form snapshots

Before launch: Make, Activate tool window

Figure 14: Attack from authentication 2.

Figure 13 represents the authentication system inputs. By running a few malicious attacks on the IOT authentication script, it was discovered that it was possible to bypass the verification procedures. The following malicious attacks were tested:

Because the attacker's source IP must match the source IP specified in the signed authentication message, a replay attack was unsuccessful, attempting to modify the signed authentication message failed because the script checks the signature of the message.

Due to the fact that the public key should point to the Ethereum address of a genuine user, injecting the attacker's own authentication package resulted in an error.

Figure 14 represents the authentication system in the proposed blockchain technology. A man-in-the-middle may be able to smell outgoing authentication packets if he gets close enough.

However, the integrity of the signed authentication message is safeguarded since he or she is unable to change it.

In terms of security, our technology offers the following advantages.

(1) Prevent malicious nodes from gaining access. We use a string of permissions to perform peer-to-peer authentication, and the access control layer effectively prevents dangerous nodes from entering

(2) The ability to withstand DDoS attacks. The suggested system is decentralized, as all nodes share a register. Even if certain nodes are under DDoS attack, the system will continue to function

(3) Close the back door in the firmware. We store important data hash values such as firmware and configuration files on the blockchain because of its inviolable nature. Any node on a network can check the integrity of a piece of data and find the firmware back door instantly

```
Inputs Data from the user, =
Qutput: Private key from the user
Start
    fork = 0to length (private key)
    $q(j + 1) = q(j) + L^* \cos((j))$
    $(j + 1) = (j) + q(j + 1)$
    end for
    smart contract=int 0
    smart contract=smart contract % length (private key)
    Private key=check private key
    fori = 0to length (private key)
    idx1 = i
    idx2 =key value (i)
    swap(smart contract (idy1), smart contract (idy2))
    end for
    End
```

Pseudocode 1: Pseudocode of Authentication.

The next stage is to compare the suggested solution to prior solutions that have been offered in order to ensure that it is of high quality. The assessment measure is based on whether or not the supplied authentication method was able to resolve issues that had arisen with the previous authentication mechanisms that had been suggested for IOT devices. The evaluation metrics are more explicitly established in this comparison than in the previous one. Availability is defined as the removal of the bottleneck and the capacity to operate without a single point of failure. The term "scalability" is used to describe the additional overhead that occurs as new devices are added to the application's use. Decentralization refers to the capacity of an authentication application to operate without relying on a central entity that, if disrupted, may cause the system to malfunction. Tamper proofing is the guarantee that stored data and transactions will not be tampered with after they have been recorded in the system's log files.

7. Conclusions and Future Work

We looked at the cons of classic IoT for identification and security services in this article. We also presented a blockchain paradigm for IoT security and authentication. The system's implementation was also described in length. In addition, to test the proposed system, we are developing a prototype system based on Hyperledger Fabric. In comparison to other research, ours has the advantages of being generic and simple. It is appropriate for deployment on lightweight tools such as the Internet of Things due to its minimal implementation cost. Furthermore, the multichain structure adds an extra layer of security between distinct regions of trust. The focus of future work will be on integrating vast amounts of IoT data with traditional blockchain-based financial transaction data.

Data Availability

The data used to support the study are included within the article.

Conflicts of Interest

The authors of this manuscript declared that they do not have any conflict of interest.

References

[1] M. A. Bouras, B. Xia, A. O. Abuassba, H. Ning, and Q. Lu, "IoT-CCAC: a blockchain-based consortium capability access control approach for IoT," *PeerJ Computer Science*, vol. 7, article e455, 2021.

[2] A. J. Dadhania and H. B. Patel, "Access control mechanism in Internet of Things using blockchain technology: a review," in *2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)*, pp. 45–50, Thoothukudi, India, December 2020.

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

[4] R. Xu, Y. Chen, and E. Blasch, "Decentralized access control for IoT based on blockchain and smart contract," in *Modeling and Design of Secure Internet of Things*, C. A. Kamhoua, L. L. Njilla, A. Kott, and S. Shetty, Eds., pp. 505–528, John Wiley & Sons, Inc., 2020.

[5] P. Zhai, L. Zhang, and J. He, "A review of Blockchain-based access control for the industrial IoT," *Converter*, vol. 2021, no. 3, pp. 308–316, 2021.

[6] R. Fotohi and F. Shams Aliee, "Securing communication between things using blockchain technology based on authentication and SHA-256 to improving scalability in large-scale IoT," *Computer Networks*, vol. 197, article 108331, 2021.

[7] S. Algarni, F. Eassa, K. Almarhabi et al., "Blockchain-based secured access control in an IoT system," *Applied Sciences*, vol. 11, no. 4, p. 1772, 2021.

[8] A. I. Abdi, F. E. Eassa, K. Jambi, K. Almarhabi, and A. S. A. L. M. al-Ghamdi, "Blockchain platforms and access control classification for IoT systems," *Symmetry*, vol. 12, no. 10, p. 1663, 2020.

[9] R. Sekaran, R. Patan, A. Raveendran, F. al-Turjman, M. Ramachandran, and L. Mostarda, "Survival study on blockchain based 6G-enabled mobile edge computation for IoT automation," *IEEE Access*, vol. 8, pp. 143453–143463, 2020.

[10] M. Zhang, L. Lin, and Z. Chen, "Lightweight security scheme for data management in E-commerce platform using dynamic data management using blockchain model," *Cluster Computing*, vol. 24, no. 2, pp. 1–15, 2021.

[11] T. Hewa, G. Gür, A. Kalla, M. Ylianttila, A. Bracken, and M. Liyanage, "The role of blockchain in 6G: challenges, opportunities and research directions," in *2020 2nd 6G Wireless Summit (6G SUMMIT)*, pp. 1–5, Levi, Finland, March 2020.

[12] Y. Zhang, B. Li, B. Liu, J. Wu, Y. Wang, and X. Yang, "An attribute-based collaborative access control scheme using blockchain for IoT devices," *Electronics*, vol. 9, no. 2, p. 285, 2020.

[13] G. Nyame, Z. Qin, K. O. B. Obour Agyekum, and E. B. Sifah, "An ECDSA approach to access control in knowledge management systems using blockchain," *Information*, vol. 11, no. 2, p. 111, 2020.

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

[15] B. Arunkumar and G. Kousalya, "Blockchain-based decentralizedsecure lightweight E-health system for electronic health records," in *Intelligent Systems, Technologies and Applications*, pp. 273–289, Springer, New York, 2020.

[16] T. Sultana, A. Almogren, M. Akbar, M. Zuair, I. Ullah, and N. Javaid, "Data sharing system integrating access control mechanism using blockchain-based smart contracts for IoT devices," *Applied Sciences*, vol. 10, no. 2, p. 488, 2020.

[17] A. H. Lone and R. Naaz, "Applicability of Blockchain smart contracts in securing internet and IoT: a systematic literature review," *Computer Science Review*, vol. 39, article 100360, 2021.

[18] K. Sekaran, R. Rajakumar, K. Dinesh et al., "An energy-efficient cluster head selection in wireless sensor network using grey wolf optimization algorithm," *Telkomnika*, vol. 18, no. 6, pp. 2822–2833, 2020.

[19] T. P. Latchoumi and L. Parthiban, "Quasi oppositional dragonfly algorithm for load balancing in cloud computing environment," *Wireless Personal Communications*, 2021.

[20] W. Serrano, "The blockchain random neural network for cybersecure IoT and 5G infrastructure in smart cities," *Journal of Network and Computer Applications*, vol. 175, article 102909, 2021.

[21] R. Kumar and R. Tripathi, "Scalable and secure access control policy for healthcare system using blockchain and enhanced bell–LaPadula model," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 2, pp. 2321–2338, 2021.

[22] N. R. Sivakumar, "Investigation study on secured data communication with blockchain and IOT in green cloud computing," in *Information and Communication Technology for Competitive Strategies (ICTCS 2020)*, pp. 675–686, Springer, Singapore, 2022.

[23] N. Shi, L. Tan, C. Yang 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.

[24] N. Hussain, P. Maheshwary, P. K. Shukla, and A. Singh, "Detection of black hole attack in GPCR VANET on road network," in *International Conference on Advanced Computing Networking and Informatics*, R. Kamal, M. Henshaw, and P. Nair, Eds., vol. 870 of Advances in Intelligent Systems and Computing, Springer, Singapore, 2019.

[25] A. Bansal, M. K. Ahirwar, and P. K. Shukla, "A survey on classification algorithms used in healthcare environment of the Internet of Things," *International Journal of Computer Sciences and Engineering*, vol. 6, no. 7, pp. 883–887, 2018.

[26] H. Dehariya, P. Kumar Shukla, and M. Ahirwar, "A survey on detection and prevention techniques for SQL injection attacks," *International Journal of Wireless and Microwave Technologies*, vol. 6, no. 6, pp. 72–79, 2016.

[27] R. Singh, P. Rawat, and P. Shukla, "Robust medical image authentication using 2-D stationary wavelet transform and edge detection," in *2nd IET International Conference on Biomedical Image and Signal Processing (ICBISP 2017)*, pp. 1–8, Wuhan, China, 2017.

[28] N. Hussain, A. Singh, and P. K. Shukla, "In depth analysis of attacks & countermeasures in vehicular ad hoc network," *International Journal of Software Engineering and Its Applications*, vol. 10, no. 12, pp. 329–368, 2016.

[29] P. KumarShukla, S. Singh Bhadauria, and S. Silakari, "ARA MAC - a qualifying approach for improving attack resiliency and adaptivity for medium access control protocol in WLAN 802. 11," *International Journal of Computer Applications*, vol. 49, no. 19, pp. 1–11, 2012.

[30] A. K. Saxena, S. Sinha, and P. Shukla, "A review on intrusion detection system in mobile ad-hoc network," in *2017 International Conference on Recent Innovations in Signal processing and Embedded Systems (RISE)*, pp. 549–554, Bhopal, India, October 2017.

[31] N. Hussain, P. Maheshwary, P. K. Shukla, and A. Singh, "Mobility-aware GPCR-MA for vehicular ad hoc routing protocol for highways scenario," *International Journal of Organizational and Collective Intelligence*, vol. 8, no. 4, pp. 47–65, 2018.

[32] E. K. Wang, J. Chen, Y. Peng, and L. Zhang, "Editorial: physical layer security and wireless access control (QSHINE 2017)," *Mobile Networks and Applications*, vol. 25, no. 1, pp. 1–3, 2020.

[33] M. Turcotte, A. Kent, and C. Hash, "Unified host and network data set," *Data Science for Cyber-Security*, pp. 1–22, World Scientific Publishing, 2018.