Egyptian Informatics Journal 23 (2022) 345–361

Contents lists available at ScienceDirect

Egyptian Informatics Journal

journal homepage: www.sciencedirect.com

Full length article

Hierarchical blockchain structure for node authentication in IoT networks

Mahmoud Tayseer Al Ahmed [a,b], Fazirulhisyam Hashim [a,*], Shaiful Jahari Hashim [a], Azizol Abdullah [c]

[a] Department of Computer and Communication Systems Engineering, and Wireless and Photonics Networks Research Centre (WiPNET), Faculty of Engineering, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia
[b] Department of Communication Engineering and Technology, Faculty of Engineering and Technology, Palestine Technical University – Kadoorie, Palestine
[c] Department of Communication Technology and Network, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia

ARTICLE INFO

Article history:
Received 23 September 2021
Revised 25 December 2021
Accepted 11 February 2022
Available online 19 February 2022

Keywords:
Blockchain
IoT
Security
Authentication
Clustering
Hierarchy

ABSTRACT

Internet of Things (IoT) networks are large peer-to-peer networks of small devices that require a competent security system that is scalable and adaptable to the limited resources of the IoT devices. Node authentication is a crucial part of IoT security. The current authentication solutions require a centralized trusted party for authentication, which presents a single point of failure. Blockchain as a peer-to-peer network with decentralized authentication can provide a decentralized solution for node authentication. In existing literature, most blockchain applications in IoT are connected to existing blockchain networks by more computationally capable devices, thereby limiting their adaptability for IoT networks and presenting single point of failure problem. Considering the issues, this paper proposes a blockchain-based decentralized structure for authentication by arranging the IoT devices into clusters based on their computational capability, energy reserve and their location. The devices in each cluster are authenticated by a hierarchical structure of interconnected blockchains. To reduce the processing load we introduced a consensus protocol based on verifying identity-based encryption key signature of the device and its related cluster. The proposed structure simulation has shown a reduction of the processor and memory load of IoT devices. Further testing using Docker container network and Raspberry Pi devices network has shown that the proposed blockchain structure and consensus algorithm have reduced computational load. The analysis of the structure security and performance shows it offers comprehensive security protection while being lightweight and scalable.

1. Introduction

Internet of Things (IoT) is a smart environment containing smart devices that have sensory and communication capability to autonomously generate data and transmit it using the Internet [1]. With the growing number of applications in different industrial and commercial fields, the IoT ecosystem is required to be an open environment in which all devices are interconnected making these devices vulnerable to malicious security attacks.

The IoT environment needs to have reliable and lightweight security and privacy [2–5]. One important aspect of IoT security is the need to prevent advisories from assuming the identity of IoT devices to gain access to the network or the data. Hence, Robust IoT device authentication is required to ensure devices connected to the IoT network can be trusted to be what they allege to be. Multiple solutions for IoT security were introduced based on cryptography depending on key distribution centers. Several solutions based on the existing cryptographic application were introduced such as zero-knowledge proof authentication [6], and anonymous access authentication scheme for wireless sensor networks [7]. However, these applications require the existence of key

* Corresponding author.
E-mail addresses: mahmoud.sawalha@ptuk.edu.ps (M.T. Al Ahmed), fazirul@upm.edu.my (F. Hashim), sjh@upm.edu.my (S. Jahari Hashim), azizol@upm.edu.my (A. Abdullah).

Peer review under responsibility of Faculty of Computers and Information, Cairo University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.eij.2022.02.005

management servers and as a result, there exists the problem of having a single point of failure in the IoT security system which can compromise the whole security of the network. This suggests the need for a decentralized security solution for IoT networks

On the other hand, blockchain presents a network of nodes that operate with no central controlling entity [8,9]. It is a decentralized ledger that depends on the collective trust between its contributors by presenting a mechanism of proofing their authenticity by sacrificing some valuable resources to join and interact with the ledger. The distributed nature of blockchain that proved to be secure and immutable, can provide the solution for IoT security requirements. Specifically, the distributed nature of the IoT network. Multiple solutions for blockchain use in IoT where reviewed in [10–12]. These solutions can be categorized into the following, first using smart contract for authentication of IoT devices or data. These smart contracts are managed and published in blockchain network by introducing an authenticating node or server. Second category is using blockchain network for key distribution and security management of the IoT network. The final category introduces or adapts new form of blockchain to provide the security requirements of IoT networks. In general, the existing blockchain applications in IoT networks introduce some limitations including the large size of the blockchain ledger that would increase the memory load and the data bandwidth used by the IoT network [13,14]. Besides, using blockchain applications such as Bitcoin and Ethereum, will limit the availability of these networks for use with the increasing numbers of IoT devices because the IoT devices will have to compete with other devices that use the Ethereum and Bitcoin platforms for financial profit. Moreover, using a token system to authenticate the IoT device where a controlling device that mines for tokens used for authentication would also present a single point of failure. Finally, the use of complicated proof of work algorithms increases the load on the IoT devices [15,16].

The motivation of this paper is to address the IoT need for a decentralized authentication process that is scalable, dependable, and secure. The introduction of blockchain technology for IoT security can help to satisfy these authentication requirements. However, the implementation of the existing blockchain applications inherits multiple limitations, especially the dependency on existing financial applications and the dependency on entities that facilitate interaction with the blockchain network. In this paper we propose a new method for authentication for IoT networks inspired by the decentralized ledger used in transaction authentication in the blockchain.

We started by creating a network model that can help the IoT devices to share the load of the authentication process by letting the more capable devices to be the main verifier to the less capable devices within the group. This led us to introduce hierarchical arrangement of clusters where every level of nodes share a level of the blockchain. In addition, the requirement of having each node related to the adjacent nodes adds an increased level of security because an advisory cannot randomly attack a part of the network without going through the process required to select the cluster head of that part of the network. At the same time the block addition process depends on the cluster head providing the seeding hash to the blockchain in the cluster. Finally, a consensus algorithm based on identity-based encryption (IBE) is introduced, this algorithm uses a single transaction in each block where the keys of the node and the related cluster head are verified to create a new block. We will show that this algorithm is fast and resource efficient. To address the needs of the distributed nature of the IoT network the nodes that were added to the blockchain are added to an authentication table that can be distributed to the IoT devices in the network. This table is arranged in a way that can provide ease of access by the nodes and meets the storage limitations in IoT devices. The new structure is simulated and analyzed. The results show that it would satisfy the security requirements of the IoT ecosystem for a lightweight and scalable decentralized security. The contributions of this paper are as follows:

1. A network model based on grouping the devices in multiple levels of clusters where each cluster head verifies the devices within the cluster.
2. A multi-level blockchain structure where each cluster incorporates its own blockchain. Each blockchain is connected to other cluster blockchains. This approach would reduce the memory and computational loads on the devices within each cluster.
3. A consensus scheme derived from the proof of authentication protocol is applied. Where the consensus is achieved by a signed request from each device. The node requests authentication, and when the block is created and approved, the authenticated nodes are added to the authentication table.
4. An authentication scheme is introduced based on the multilevel blockchain structure, where devices are verified by the device identification and the block hash that contains the device credentials from the authentication table and the multilevel blockchain.
5. Several experiments are conducted to evaluate the performance of the proposed framework.

The remainder of this paper is organized as follows. Section 2 presents a review of existing and related works of IoT node authentication using blockchain. Section 3 discusses the proposed framework in detail. Finally, the simulation process and discussion are presented at the end of this paper (Section 4), followed by concluding remarks.

2. Literature review

Recently, numerous researchers have been interested in the integration of blockchain technology into IoT ecosystems to benefit from blockchain properties of distribution, immutability, and decentralization. However, very few were interested in how blockchains can help to meet IoT security requirements. Most researchers adopt the blockchain to IoT networks by adding gateways or edge devices to manage and authenticate the IoT devices.

Huh et al. [17] proposed an approach to integrate blockchains into IoT. Their approach relies on the idea of configuring each object by a dedicated smart contract that defines its actions. However, their work is still in a very primary state and no details about the considered use cases were provided. Finally, they consider the full anonymity of the used objects, which allows any user, even malicious, to make use of the system.

Dorri et al. [18,19] proposed a blockchain-based architecture for IoT. Their approach relies on three interconnected blockchains: a local private blockchain for each use case, a shared private blockchain, and a public blockchain that interconnects and incorporates the private blockchain networks. While the solution resolves the problem of identification, it has multiple shortcomings where each operation requires at least 8 network communications which can quickly flood the whole communication medium in case of high activity of nodes. *Hardjono et al.* [20] proposed Chain Anchor, a privacy-preserving method for commissioning an IoT device into a cloud ecosystem. Chain Anchor supports device-owners being remunerated for selling their device sensor-data to service providers and allows device-owners and service providers to share sensor-data in a privacy-preserving manner. However, its goal is

the full anonymity of the participating devices and is not adapted to numerous IoT use cases where identification is needed.

"Fair Access", a blockchain-based access control framework in IoT was proposed in [21]. Fair Access works by storing the policies in a private blockchain. Thus, the authentication and the updates of policies are always guaranteed. However, it can handle only policy-based compatible systems and use cases, which cannot be applied to numerous IoT contexts. In [22] *Hammi et al.* introduced a decentralized Blockchain-based authentication system for IoT. The authors suggest a virtual secure area of IoT devices. Where a center node issues permission based on Ethereum. These permissions or tokens are used by the users to access the data provided by IoT devices. However, this approach uses a central node that is assigned to the system and offers no solution to this node's unviability situation.

Yiming Jiang et al. [23] proposed a cross-chain framework to integrate multiple blockchains for efficient and secure IoT data management. They propose a solution that builds an interactive decentralized access model that employs a consortium blockchain as the control station. Other blockchain platforms customized for specific IoT scenarios run as the backbone of all IoT devices. It is equivalent to opening the off-chain channels on the consortium blockchain. However, their work uses existing Ethereum and IOTA blockchain structures which makes the network dependent on the availability of these services.

Meanwhile, the authors of [24] proposed a blockchain model based on hypergraphs. This model aims to reduce storage consumption and to solve additional security issues. In the model, the hyperedge is used as the organizer of storage nodes and converts the entire networked data storage into part network storage. The author presents only two levels for the chain and the application is set towards data storage security and does not address the device authentication.

Also, in [25], a novel consensus algorithm called Proof-of-Authentication (PoAh) to replace Proof-of-Work and introduce authentication in such environments to make the blockchain application-specific. Where the authors use pre-distributed asymmetric key pairs as a method of authentication in the blockchain to replace the use of the conventional mining processes. The approach presents an improvement to the use of node authentication of IoT networks. However, their approach requires the presence of authenticating nodes within the network which causes an increased traffic towards these nodes and increased bandwidth requirements for these nodes.

The authors of [26] introduced a hybrid blockchain structure that uses local blockchain to authenticate the WSN nodes and a public blockchain to connect multiple clusters of nodes. However, the proposed work assumes the presence of access point devices in each cluster which is a single point of failure in each cluster, also the authentication, in general, depends on Ethereum smart contracts to authenticate for authenticating the access points in the public network. In [27], the authors presented a capability-based access control for IoT devices. Their approach depends on creating a register of participating nodes ID's that are managed by a smart contract in consortium blockchain.

Chenhao Xu et al. presented in [28] a bidirectionally linked blockchain and consensus based on selected members in IoT networks. The authors presented a way to link the hash values of the blocks in the blockchains by adding a random value to the hash. Their work presents a payment method to prevent double spending attacks. However, their work requires additional complexity to the consensus process by requiring additional steps of verification for selected number of nodes.

k. Agyekum et al. [29] proposed a data sharing approach in IoT networks based on identity based encryption and information-centric networking that are stored and managed by a blockchain. However, the proposed work uses existing blockchain technology to store and share public key values. In [30], a blockchain based trust system is proposed to deter malicious data reporting parties in the network. The proposed application uses the blockchain for only data storage. Similarly, authors of [31] presented a module to detect and isolate malicious devices in IoT network. The authors used the network capable devices as blockchain clients that determine whether a device is complaint to the security parameters or not. Blockchain is used as a database to store the status of the IoT devices.

The authors of [32] proposed a blockchain based access control module for medical IoT applications where the medical data is shared and managed by smart contract in blockchain environment. In a similar way the authors of [33] used smart contract in blockchain to authenticate and manage the resource allocations within the IoT network. Both works used the blockchain as a secure decentralized data base that can be accessed and managed by single points of access.

For the benefits of readers, we summarize the previously highlighted related works in Table 1. It can be observed that most of the works utilized the existing bitcoin or Ethereum platform as the basis for authentication mechanism (i.e., either for economical profit by monetizing the network access authentication or by using a token issued to allow access to the network). Nevertheless, only a small number of researchers have used the blockchain structure for authentication. Most of the reviewed works use existing blockchain technology such as Bitcoin and Ethereum, which are designed for financial applications and not specifically designed for IoT authentication. Moreover, while some are operated on cluster-based algorithms, the cluster head selection process is predetermined. Unlike the existing works, our proposed framework focuses on the dynamic assignment of cluster heads among the nodes. In addition, we are presenting a new lightweight method of consensus that considers the resource constrained nature of IoT devices.

3. Proposed blockchain-based decentralization framework

3.1. An overview of blockchain

The first proposed application of blockchain is Bitcoin [34], which was the first form of cryptocurrency without a centralized controlling body for issuing currency and controlling financial transactions. Based on its security applications [35], blockchain has two main categories permissioned and permissionless. Permissioned blockchain requires access to be allowed. Permissioned blockchain has three types: private blockchains, consortium blockchains, and public blockchains. In the other hand permissionless blockchain requires no authorization to access the network and any entity can join the blockchain.

The structure of the blockchain is shown in Fig. 1. Each block in the blockchain consists of a field for the previous block hash value using SHA256 standard, a field for the time stamp, and a field for a nonce value [36]. There is also a field to store a Merkle tree of transaction hash. The Merkle tree root is combined with the other fields to calculate the nonce value. This value must result in a hash value for the block that satisfies the blockchain consensus requirements.

Blockchain principle operations can be described as follows [37]; a node records new data values and broadcasts them to the network. The receiving nodes verify the data and store them in a block. All nodes in the network perform a mathematical operation depending on the requirements of the network. This process is called mining. This mathematical algorithm requires consensus from the nodes in the network. If the consensus is achieved the

Table 1
Summary of related works.

Reference	Contribution	Blockchain application	Challenges
[17]	Using smart contracts to control IoT devices	Ethereum	Delay times caused by Ethereum consensus requirements and the need for proxy device to access Ethereum
[18,19]	Uses multiple levels of private and public blockchain for IoT security and presents a smart home use case.	PoW blockchain	High number of communications for each device. Each sensor network depends on a single access point to access the blockchain
[20]	Provides full anonymity of the devices by providing multiple keys for each device that are managed and stored in a blockchain	Semi-permissioned blockchain where certain devices can write to the blockchain, and any device can read from the blockchain	Requires the device manufacturer to register the devices in the blockchain. Has multiple single points of failure in the system. Device identification is not allowed
[21]	Decentralized access management using smart contracts	Bitcoin based PoW blockchain	Has large overhead time caused by bitcoin consensus requirements. The access points must store large values of data.
[22]	Creates safe zones that are authenticated by using smart contracts	Ethereum	Limits device communications to the safe zone, uses an access point to extend the communication range which presents a point of failure
[23]	Uses a side chain based on IOTA for device security management that is connected to Ethereum blockchain	IOTA and Ethereum	Complicated implementation because the research uses two different blockchain technologies. uses dedicated chains for accessing the Ethereum blockchain
[24]	Randomly groups the nodes. the groups of nodes are shared in the same transaction. this allows for decentralized transaction storage	POW and linear independence matrix	Randomly selects the nodes without consideration to their capabilities. Each group uses a mining device for blockchain access and storage which presents a point of failure.
[25]	Presents new consensus algorithm based on proof of authentication	Proof of authentication	Uses validating nodes which increases the bandwidth load and present a point of failure
[26]	Uses Hybrid blockchain structure consisting of local blockchain that are connected to public blockchain.	PoW based blockchain	Local blockchain is stored in cluster head. the access to the public blockchain is managed by the cluster head and an access point.
[27]	Capability-based IoT access control using blockchain where a set of defined credintial determines the the device access to the network	Proof-of-Authority	Device registration requires multiple steps with large communication overhead.
[28]	The authors presented a way to link the hash values of the blocks in the blockchains by adding a random value to the hash	Committee Members Auction consensus	Consensus is reached within selected devices, the calculation for the linked hash increases the computational requirements.
[29]	Presents data secure sharing scheme for IoT network based on proxy re-encryption where the keys are distributed on a blockchain	PBFT	the blockchain is used as a key storage and distribution authority. The author does not address the complexity of blockchain application in IoT network.
[30]	Proposes a scheme where UAV's monitor the IoT network and detects malicious data reporting	Not specified	Blockchain is used for data storage, the use of UAV's adds complexity to the scheme. Also using UAV to monitor the IoT network presents privacy concerns.
[31]	Proposes module isolate malicious devices in IoT network	PBFT	Has scalability issues and devices may be isolated because of possible transmission error.
[32]	Uses smart contracts to secure patients' data generated by medical IoT devices	Ethereum	In this application blockchain is used for data storage in smart contract transactions. the application depends on the consensus parameters and requirements of the Ethereum network.
[33]	Uses blockchain enabled IoT where each base station is equipped with an edge server for computing offloading. The resource allocation is managed by smart contracts	Ethereum	Resource allocation is managed by dedicated servers that can present potential point of failure. Also, the process of buying the computational resources would be costly when the number of nodes increases.

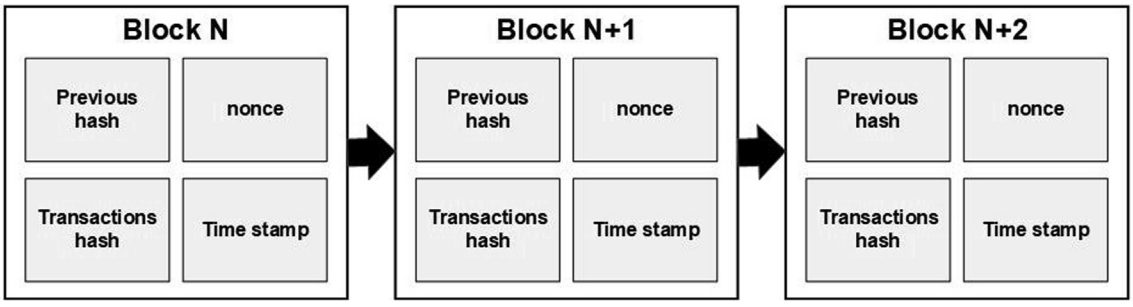

Fig. 1. Blockchain structure.

block is added to the blockchain and the blockchain is updated to all nodes in the network.

The mining process aims to make any change to the blockchain as complicated and demanding as possible. The most used consensus protocols that can be applied in blockchain include [22,38]:

- Proof of Work (PoW): Where the nodes are required to calculate a nonce value to meet certain value requirements of the block hash.
- Proof of Stake (PoS): The consensus is based on the stake of the node in the network. The node with a higher stake can validate new blocks in the chain. This approach is introduced to reduce the high calculation requirements of the PoW approach.
- Practical Byzantine Fault Tolerance (PBFT): Is commonly used in private systems where the network assumes a number of faulty nodes. Based on this assumption, the node requires the consensus of the remaining nodes to generate new blocks in the chain.
- Proof of Elapsed Time (PoET): This consensus is used in private networks. It requires the nodes to wait for a random time period before achieving the consensus on a new block.

The main security threats for blockchain include the majority attack where an attacker or group of attackers has more than 51% of the processing power in the network [34]. Another major concern is the forking problem; when the rules for the difficulty of PoW are changed the nodes tend to generate blocks that follow the easiest rules which will cause the chain to have more than one branch with different authentication rules. Another issue with blockchain is the scale of the blockchain, as the blockchain grows, the data size increases. These issues were addressed [39] in different ways to provide a solution for the demanding requirements of the consensus algorithms and the security threats to the widely adopted consensus protocols.

In this work, we are adapting proof of authentication protocol presented in [25] and [40]. Proof of authentication consensus algorithm uses an identity-based cryptography (IBE) to create a key pair for each node. The protocol assigns some trusted nodes as validators that validate the node's transactions that contain a signature. Nodes increase their trust level by validating more nodes.

In the proposed work we are presenting a more simplified version of the proof of authentication by considering the nodes in the network that are assigned as cluster heads as the initial validators. This would reduce the consensus overhead and would require shorter time for validation and block creation. The details will be discussed further in the following sections.

3.2. Proposed framework

The main goal of the proposed framework is to provide an authentication method inspired by the immutable ledger from blockchain technology while minimizing the processing loads required by the blockchain mining process.

This proposed authentication framework is envisaged to satisfy the following goals:

- **Decentralization:** With an increasing number of IoT devices in a network, the use of centralized authentication may not be feasible. Moreover, such a centralized approach exposes the network to the single point of failure problem. Owing to these issues, our proposed framework presents a decentralized model for authentication.
- **Efficiency of resources requirements:** Given the limited resources of the IoT devices, the proposed framework presents a more efficient method of blockchain use in IoT networks. This can be achieved by introducing clustering or hierarchical structure of blockchain in the network.
- **Anonymity:** This goal is of utmost importance in any network, mainly to preserve the privacy of the users. The proposed framework is required to provide a high degree of anonymity to the users.
- **Scalability:** The proposed framework is designed to be flexible and scalable.
- **Immunity to security attacks:** The proposed framework strengthens the security of IoT devices and networks notably the authentication process. Moreover, it is also immune to most security attacks that are related to IoT networks and blockchain.

The first hurdle in applying blockchain technology in IoT networks is the high calculation and bandwidth demand required by blockchain consensus algorithms. To overcome this, we are using a simplified form of consensus where the nodes are validated by a group of adjacent nodes by verifying a transaction that contains digitally signed information. This method is inspired by the work presented in [25].

The proposed blockchain-based decentralized authentication framework is organized in a multi-level or hierarchical structure. The framework incorporates two important stages, namely, clustering process and blockchain-based authentication process. In the former process, the nodes are grouped into clusters based on predefined criteria. Meanwhile during the second process, the nodes are authenticated by the suggested blockchain structure, and then added to the authenticated nodes table.

3.2.1. Clustering

As described previously, IoT networks consist of a large number of devices. To improve and strengthen the security management of our proposed framework, these devices are grouped into clusters. In particular, the clusters are created to reduce the size of blockchain data stored in each node. Moreover, creating smaller groups of nodes reduces the size of data shared between the nodes in the blockchain. The cluster structure ensures adaptability with existing networks that have access points or existing cluster heads that can perform the required tasks based on the proposed framework. The rules for cluster head selection are simplified to reduce the overhead required by the cluster's initialization stage.

First the devices are divided into three general categories based on their functionality and capability this arrangement is related to the devices functionality and operation within the IoT network where we have sensor and actuators with limited processing and connectivity capabilities, hub devices that have system on chip with connectivity that manage and collect data from the sensors and control the actuators. Also, we have communication devices with higher processing powers and larger bandwidths that collect and store the data and can perform much more complicated computing and finally larger cloud servers which are used for data storage and analysis. As can be seen in Fig. 2 the lowest category is the connected devices such as sensors and actuators. In general, these devices have limited connectivity and limited processing powers. The second category is the local processing devices that have higher processing power and memory. The final category is devices connected to the cloud that have larger processing power, memory, and higher bandwidth connections. Secondly, the devices are arranged into clusters based on three criteria that include the device processing power, the device energy reserve, and the device location relative to other devices in the network.

The nodes are grouped into clusters based on the algorithm described in Fig. 3 (to be discussed). The cluster heads are working as the initial validators of the nodes joining the blockchain, also they are the connecting point between the different blockchains, and they must store the authentication table related to all the nodes in the cluster as we will discuss later. Considering this, the

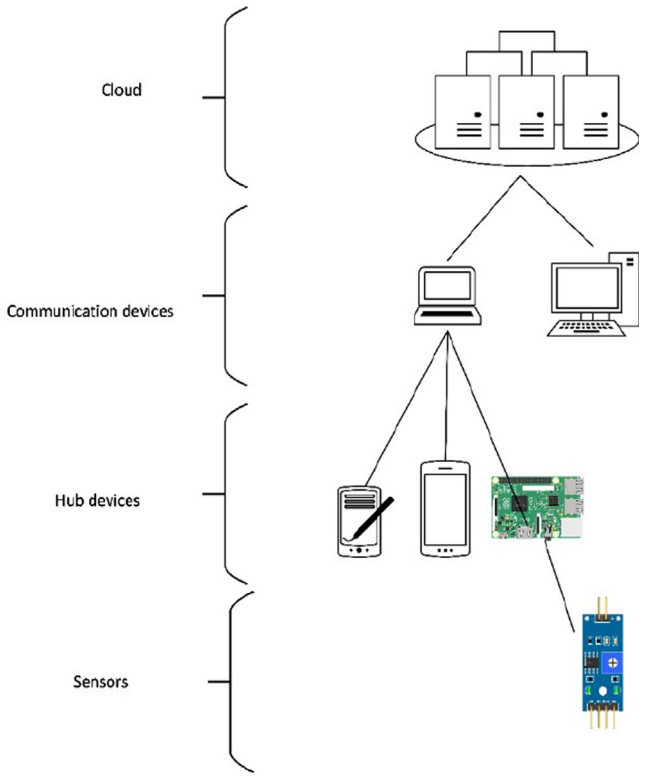

Fig. 2. Devices in IoT network.

nodes with higher energy residual are preferred to be selected as cluster heads.

Fig. 3 illustrates the procedures for the formation of clusters and selection of cluster head in our proposed framework. In particular, the sequence of cluster head selection for level 1 (CHL1) and level 2 (CHL2) is presented in the figure. As shown in step 1 of Fig. 3, each node will broadcast information about its processing power, its location, and its energy level E_{Node}, to the neighboring nodes N within a predefined range. Based on Algorithm 1, the nodes select the most suitable node to be CHL2. Each node has an indicator for its processing power index (i) based on the categories described earlier in this section, this indicator has the values $i = \{1, 2, 3\}$ to match the processing power levels. The higher value assigned to the more capable device. The nodes compare the received values of the index (i) and assign the nodes eligible to be CHL2.

The nodes with similar processing power index (i) values calculate the average energy level $E_{av} = \frac{\sum E_{node}}{N}$, and if it has an energy level larger than the average value such that $E_{node} > E_{av}$, it nominates itself as a CHL1. The node then broadcasts its nomination to the neighboring nodes (step 2). The nodes that receive the nomination messages have the choice to join the cluster based on the strength of the received signal and based on their energy levels as per step 3 of Fig. 3. If the node decides to join the cluster, it announces its selection by transmitting a cluster head selection message (step 4). This process is listed in detail by Algorithm 1. Eventually, the nodes are arranged into multiple clusters with each cluster having a level cluster head (*CHL1*).

Fig. 4 illustrates the outcome of the previously mentioned process, notably the nodes and clusters arrangement of the proposed

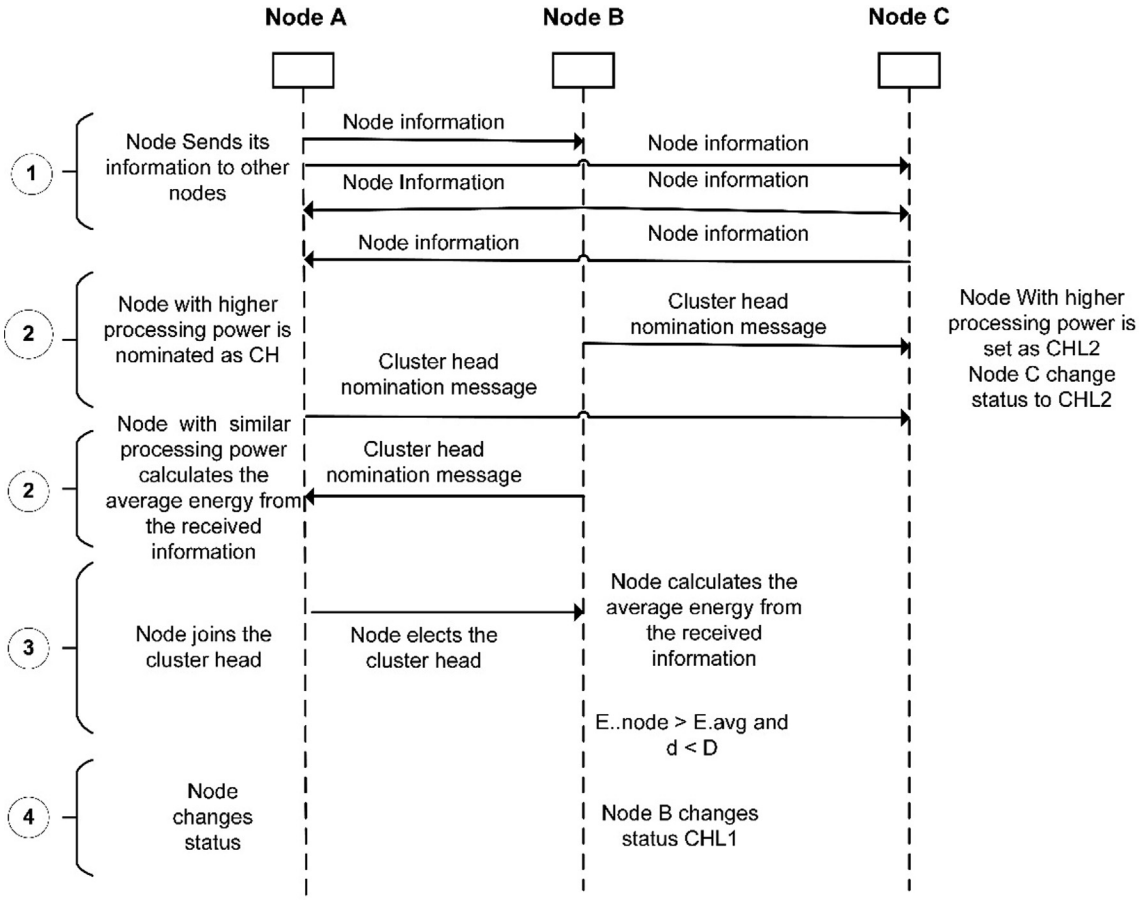

Fig. 3. Cluster formation and cluster head selection sequence for level 1 and 2.

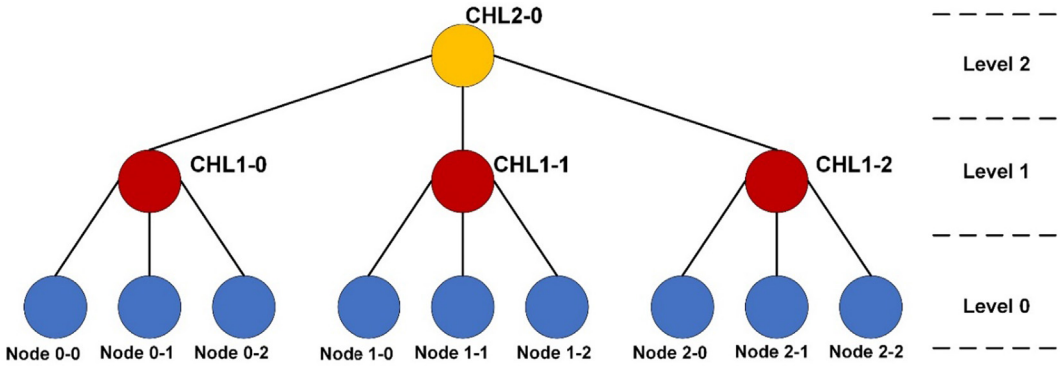

Fig. 4. Node arrangement in cluster.

framework. As can be observed in this figure, the resulting structure consists of IoT devices that are assigned as nodes (level 0 nodes). Some devices that are assigned as cluster heads *CHL1* (level 1 nodes) within the same category of processing devices and finally some devices that are assigned as a higher-level cluster head *CHL2* (level 2 nodes). This process can be repeated if the number of nodes is very large to have more than three levels of cluster arrangement. For brevity and demonstration purposes, three levels of blockchain networks are used as example throughout this paper.

Algorithm 1: Cluster Head Nomination.

Inputs: set of adjacent nodes (adjacentNodes),
number of adjacent nodes N,
node processing power indicator $Pi \in \{1,2,3\}$,
nodes energy level E_{Node},
average energy level $E_{av} = \frac{\sum E_{node}}{N}$,
predefined distance D_{preset} ,
distance between node and nominated CH d
outputs: set of level 2 cluster heads (CHL2)
set of level 1 cluster heads (CHL1)
set of nodes (nodes)
Function selectCHL2:
For node in adjacentNodes :
Add nodes with $Pi = 3$to set (CHL2)
Send CHL2 nomination message.
Function selectCHL1.
For node in adjacentNodes :
If ($E_{node} > E_{av}$) **and** node is **not** CHL2
Add nodes with $Pi = 2$ to set (CHL1)
Send CHL1 nomination message.
Else
Wait for CH nomination message
Function joinCluster.
If $d \leq D_{preset}$ **then**
Node selects the CH to join its cluster
Node broadcasts CH selection message
Else
Node is set as CH

This clustering process and algorithm will be triggered to assign a hierarchical order to the nodes as previously illustrated in Fig. 3. Moreover, this clustering process does not affect the message routing process because this process is only required for the node authentication, to be discussed in the next section.

3.2.2. Node Authentication:

As a prerequisite for the authentication process each node generates a public key pair using RSA algorithm. The key pair containing the public key PuK_{node} and private key PrK_{node} is generated by applying RSA key generation algorithm on the nodes IP address $IPAdrr_{nod}$.

$$(PuK_{node}, PrK_{node}) = RSA(IPAdrr_{nod})$$

The public key PuK_{node} is used as the node identifier in the blockchain *(NodeID)*. This eliminates the need for a centralized key generating service and at the same time provides anonymity to the nodes in the network. It is worthwhile to highlight that based on the clustering process each node has two parameters namely, the public key of the node (*NodeID*), and the public key of the cluster head node (*CHID*).

The authentication process of the proposed framework which is based on blockchain technology is illustrated in Fig. 5 and Fig. 6. As shown in Fig. 5, each group of level 0 nodes is connected in a separate blockchain. This branch-chain starts with a genesis block provided by the cluster head CHL1 node of the group as illustrated by Fig. 6. This indicates that the nodes of level 0 cannot create any blocks unless the cluster head has provided the genesis hash. This implies that any new node connecting to this group must be approved by the cluster head and by the nodes in the group. This prevents any attempts of eclipsing attacks because the blockchain cannot be forked without the original genesis block from the cluster head.

Algorithm 2 describes the proposed authentication process of the multi-level blockchain. For a node to be authenticated, it sends an authentication request (*AuthReq*) that contains the node ID, the CH ID, and the signed value of them using the node's private key to all the nodes in the cluster. The nodes validate the signed value of the authentication request and when all the nodes verify the request a block is created. The newly created block is transmitted to the cluster head and the nodes in the cluster and if the block is validated and consensus is achieved, the block is added to the blockchain. The consensus is reached by verifying the signature in the authentication request and validating the block hashes in the blockchain. The node has validated authentication requested that has been added to the blockchain is set as an authenticated node in the cluster.

The authenticated node is added to the Authenticated Nodes Table (*AuthTable*) that stores the authenticated nodes parameters. This list contains the public key of the node and the hash of the created block that granted it the authenticated status. The process is repeated for the remaining nodes of level 0 in the cluster. The values stored in the authentication table are shown in Fig. 7.

As illustrated in Fig. 6, nodes of level 1 are also connected in a blockchain with genesis block that is generated using the hash provided by the level 2 node. The node CHL1 sends an authentication request containing its ID, the CHL2 ID, and the digital signature of the request. The CHL1 nodes receive and validate the request by

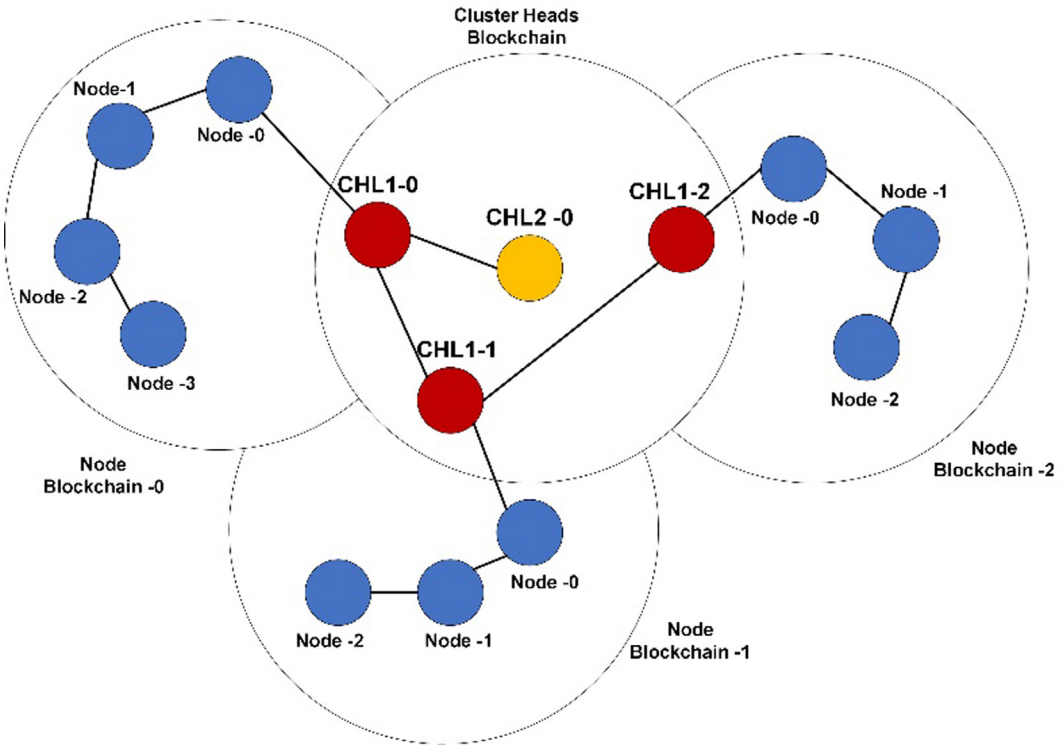

Fig. 5. Multi-level blockchain connection for different node levels.

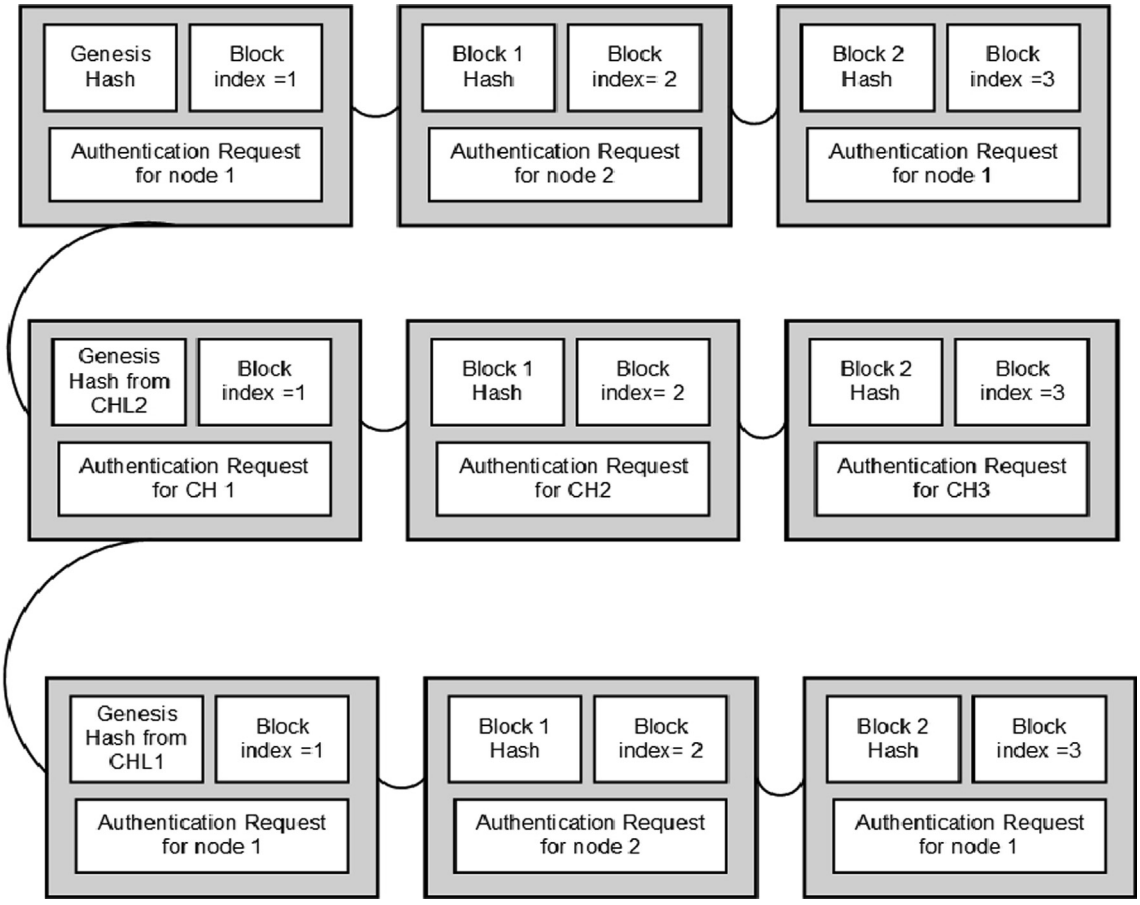

Fig. 6. Genesis block structure in the multi-level blockchain.

Device Id	Cluster Head Id	Block Hash of Authentcation	Block index

Fig. 7. Values stored in the authentication table.

verifying the digital signature, if all the nodes and the CHL2 verify the signature a block containing the CHL1 request is created and if the block is validated by CHL2 and other CHL1 nodes the block is added to the level 1 blockchain, the level 1 node and its branch chain is therefore approved and authenticated. Subsequently, this node and its branch are added to the authenticated nodes table in level 2.

Algorithm 2: Authentication based on Multi-level Blockchain.

Inputs: public key of the node (*NodeID*), public key of the cluster head node (*CHID*),
If node is not *CHID* **AND** block index = 0 **then**
Node gets hash from CH
Else if block index ≠ 0
Node sends *Authreq*= (*NodeID*, *CHID*, signature)
If *AuthReq* signature is verified
Block is created
If block validated
Add block to cluster blockchain.
Add node to *AuthTable*

hashes contained in the authenticated nodes table. This illustrates the advantage of having a cluster-based or hierarchical structure of blockchain in IoT networks.

Algorithm 3: Node Authentication based on Authentication Table.

Inputs: *NodeID*, *CHID*.
To send Data message
Node sends (*NodeID*, *CHID*)
If (*NodeID*, *CHID*) ∈ *AuthTable* **then**
Function match_Block_from Blockchain
If true, **then**
Node authenticated
Send Data message
Else
Node is not valid
Reject Data message
Else
Node is not valid
Reject Data message

As illustrated in Fig. 7 the authenticated-nodes-table contains the node parameters (*NodeID*, *CHID*) in addition to the block index and block hash of the block containing the authentication request of the node. To minimize the memory needed to store the authentication table the values are arranged in ascending levels with relation to the blockchain level that provided the authentication hash for the node as presented in Fig. 8. The authenticated nodes table is used as a reference to authenticate the node in the network by comparing the stored values of *NodeID* and *CHID* of the nodes to the values stored in the authentication table.

The node authentication process based on the Authentication Table is listed in Algorithm 3. When a node requires access to the network, the node sends its parameters (*NodeID* and *CHID*). Then, the nodes parameters are compared to the stored parameters in the authenticated nodes table (*AuthTable*). Also, the block that contains the node's authentication request (*AuthReq*) is hashed and compared to the stored hash value in the *AuthTable*. Finally, if the parameters are matched, the node is allowed to send its data to other nodes in the network. On the other hand, if the parameters are not matched the node is rejected.

The multiple-level blockchain helps to reduce the processing load and memory requirements in each node also this helps to minimize the risk of attacks in the cluster. If there was a successful attack, the compromised nodes are contained and restricted within a single cluster, and therefore may not affect the entire blockchain networks. Moreover, the attacker is required to recalculate all the

3.3. Operation Scenario

To fully explain the proposed structure, we are presenting a scenario to demonstrate how the structure is used for node authentication. First, we would like to emphasize that the proposed structure is not a direct application of blockchain, meaning that we are not looking at a continues blocks connected in a linearly increasing way. We are aiming at using the decentralized nature of the blockchain structure to benefit from its immutability and at the same time reduce its complexity to be used in IoT networks.

As explained earlier the IoT devices can be categorized into edge devices, fog devices and sensing devices. For applying our proposed authentication scheme these are categorized into three groups described in previous sections. In this scenario we are describing the steps required for initializing the authentication process and how these devices interact with each other.

First step is the clustering process, as described in section 3.1 the process arranges the devices based on their processing power, residual energy, and location into multiple levels. As can be seen in Fig. 9 we are assuming to have one CHL2 node, three CHL1 nodes and nine nodes. Every three nodes are joined with a CHL1 node in a cluster and the three CHL1 nodes are joined with CHL2 in another cluster.

For the authentication process initially as shown in Figs. 10 and 11, respectively, the nodes arranged in a single cluster. As described above each device creates a key pair based on its address.

Authentication table in level 3											
Device Id L3	Cluster Head Id	Block Hash of Authentcation	Block index	Authentication table in level 2							
				Device Id L2	Cluster Head Id	Block Hash L2	Block index	Authentication table in level 1			
								Device Id L1	Cluster Head Id	Block Hash L1	Block index

Fig. 8. Authentication table structure for multiple levels of blockchains.

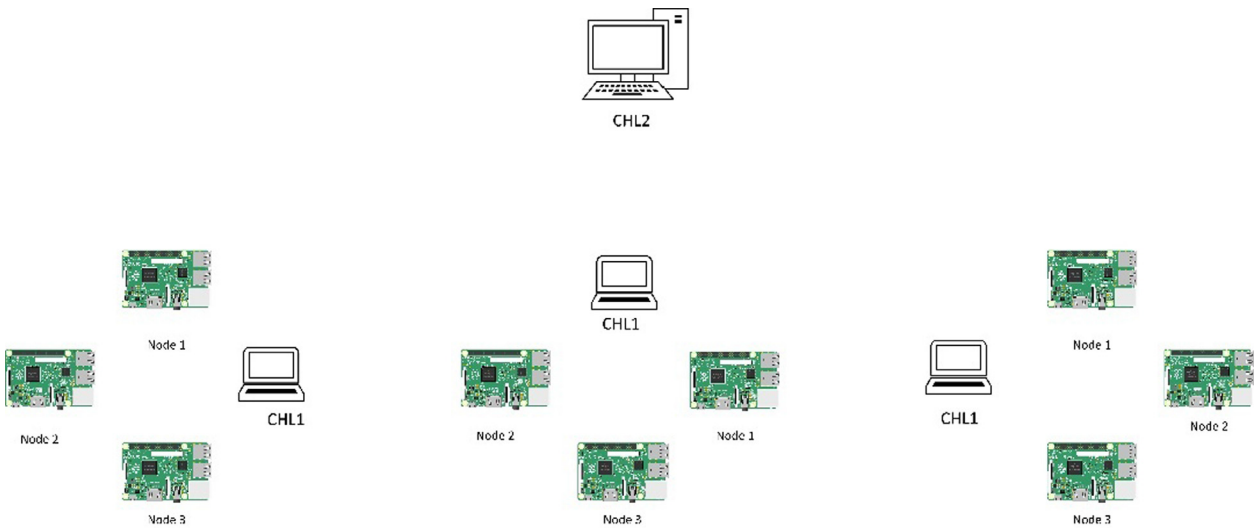

Fig. 9. Node arrangement for the operation scenario.

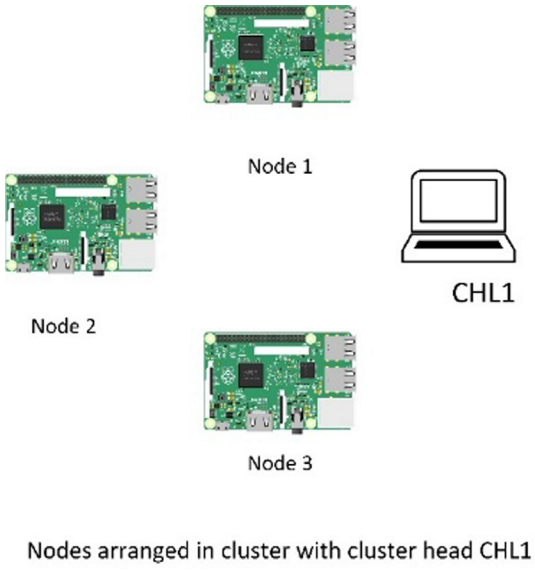

Fig. 10. The nodes in a single cluster.

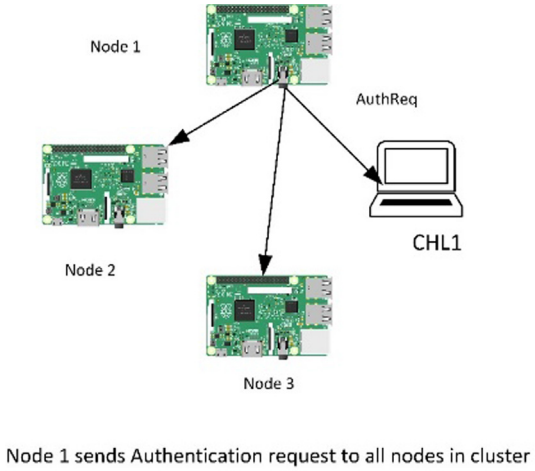

Fig. 11. Authentication request transmission in one cluster.

The nodes would send an authentication request containing their NodeID, CHID and a digital signature of the request signed using the node's private key. Here the cluster head has two functions, it works as the initial validator for the authentication request in the cluster and works as the connection point to other CHL1 nodes and CHL2 nodes in the network.

The CHL1 node receives authentication requests from the nodes in the cluster. It provides the genesis hash value for creating the first block in the cluster blockchain. As presented in Fig. 12, the node's authentication request is tested to match the node id and the cluster head ID and the signature is verified. If the values are not matched or if the signature is not verifiable, the request is rejected. If the node's IDs are matched and the signature is verified, a block is created and broadcasted to all nodes in the cluster. The new block is validated and added to the cluster blockchain.

When a node is verified, and a block is created containing its authentication values, the node is then added to the authentication table as described in Fig. 13. The authentication is shared in the network with each device storing sections of the authentication table as described in Fig. 8.

If a device requires to send data over the network, it has to be authenticated by comparing the nodeID and CHID with the values stored in the authentication table and comparing the hash value of the block containing its authentication request to the block hash from the cluster blockchain with the matching block index as

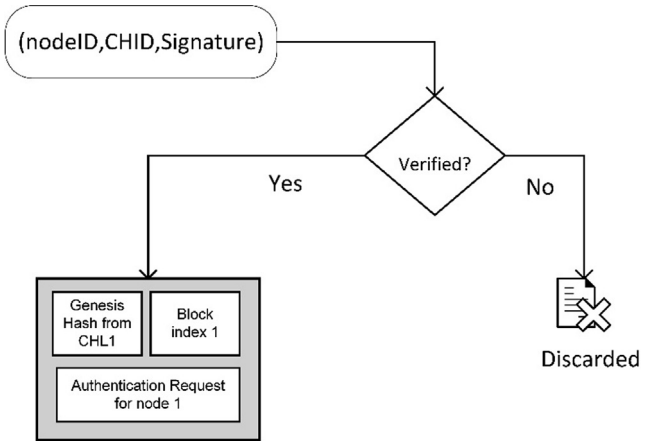

Fig. 12. The validation process for creating a block based on proof of authentication.

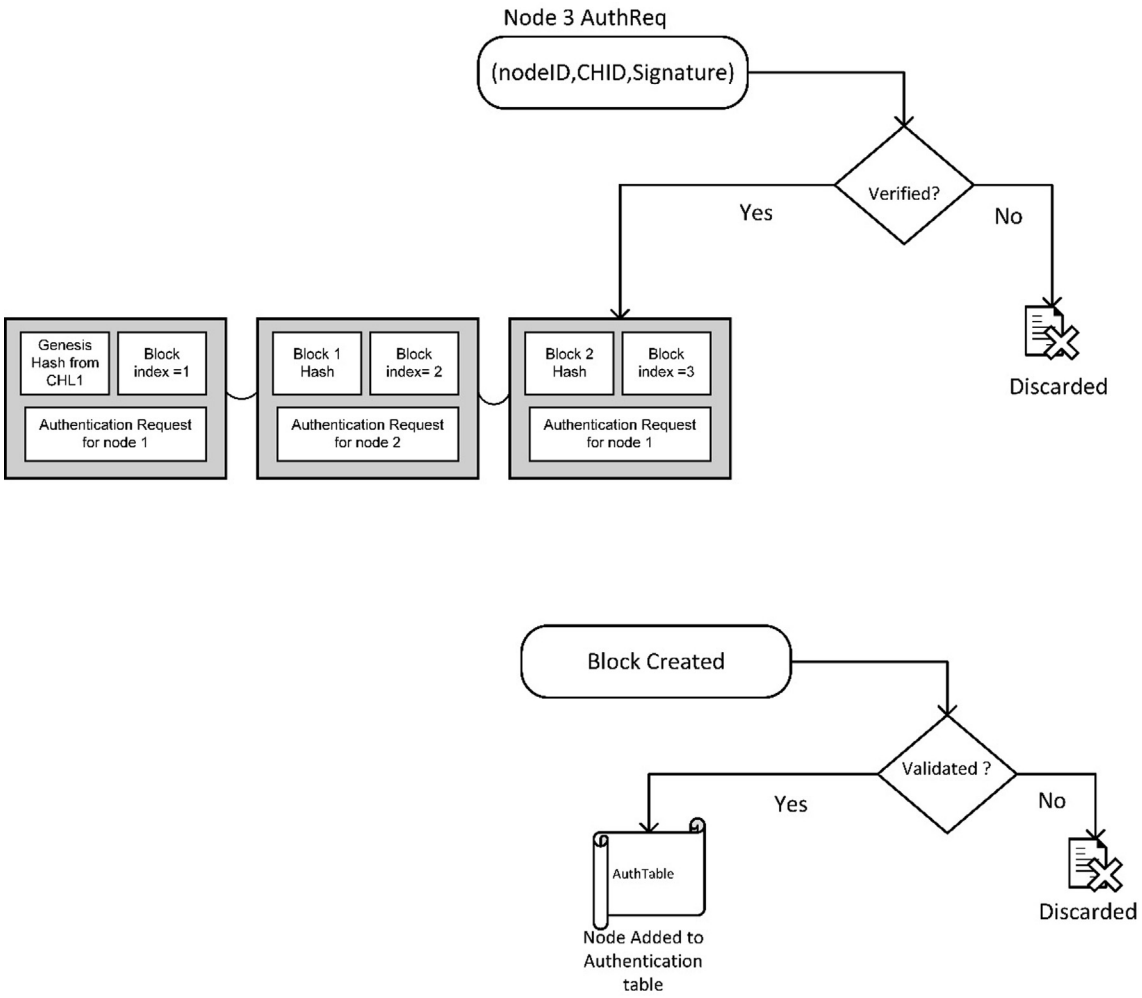

Fig. 13. Steps for node validation and verification for blockchain and authentication table.

described in Algorithm 3. The process is repeated for other nodes in other clusters in the network. Moreover, the same procedure is repeated for the CHL1 and CHL2 nodes in the networks to create a number of blockchains that authenticate multiple levels of the network.

In the case of mobile nodes, the node that changes its position can be authenticated by comparing its identification values with the values stored in the authentication table which is distributed in the network. The distribution of the authentication table in multiple levels provides a safety measure against possible failure of any cluster head.

4. Performance evaluation

This section discusses the details of our simulation settings and process. Then, it highlights and discusses the results that we obtained from our simulation.

4.1. Simulation settings and process

As described previously, the proposed framework consists of two main parts. The clustering mechanism to group the IoT devices in clusters, and the blockchain structure used for the authentication of the IoT devices in the network. The simulation and testing processes are conducted in two steps.

Firstly, the clustering processes were simulated using Omnet++, which is a C++ based discrete event simulator commonly used for network simulation [41]. It uses NED language for network environment configuration and C++ for module programming. The IoT nodes are presented as modules with wireless communication channels. The IoT devices are randomly positioned in the network and assigned a random power value at the beginning of the simulation process. The clustering algorithm, as described previously, calculates the average energy available in the nodes. The nodes with the larger power reserve are announced and nominated as cluster heads. If a node is within a predefined distance from the nominated cluster head the node selects this cluster head and joins the cluster. Then the same process is repeated for the cluster heads from level 2.

Secondly, for the blockchain authentication process, the network is simulated using docker containers and docker network. Docker provides a solution for simulating the nodes in the network in a single machine while at the same time provides enough configuration flexibility for testing the framework and for monitoring the performance of the nodes [42,43]. The simulation was conducted on a computer with Intel I7- 10510u processor and 16 Giga Bytes of RAM. The testing experiment was conducted using Docker containers running Ubuntu 20.04 and a network containing seven nodes, as presented in Fig. 14. The framework application was coded using Python 3.9 and Flask [44]. For brevity, the nodes are pre-determined to be a level 0 node, a level 1 cluster head (CHL1) or a level 2 cluster head (CHL2). Each cluster contains two nodes that are connected to a level 1 cluster head and the level 1 cluster heads (CHL1) are connected to a single level 2 cluster

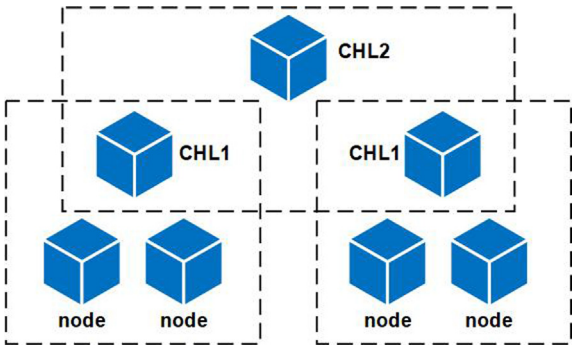

Fig. 14. Node arrangement for multi-level blockchain simulation.

head (CHL2). Each node transmits an authentication request using Flask API to the corresponding cluster head and the cluster head responds with the cluster genesis block. If a node creates an authentication block, it sends the new block to the cluster head and other nodes for validation. If the block is validated, the new block is added to the local chain and the hash of the block, and the block public key is added to the cluster head table of authenticated nodes. Each device is required to authenticate any communication request by providing the hash of the authenticated block and the public key of the device. For performance comparison a network with 7 nodes is also simulated using a single level blockchain authentication (i.e., as per the conventional blockchain approach), as shown in Fig. 15.

Finally, for further validation of the results. A network is created using Raspberry Pi devices. The network consists of two raspberry pi zero w with ARM 11 processor running at 1 GHz CPU and 512 MB RAM and two raspberry pi 3B + with Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1.4 GHz 1Gbyte of RAM. All raspberry Pi devices running Raspberry Pi OS. In addition, we are using a computer with a 1.8 GHz Intel dual-core CPU with 2 Giga Byte of RAM running Ubuntu 18.04 LTS in addition to A laptop with Intel i7-10510U CPU @ 1.80 GHz up to 2.3 GHz and 16GByte of RAM running Windows 11.

Fig. 16 illustrates the Raspberry Pi network used in our work. The devices are organized to form two clusters where each cluster consists of one Raspberry Pi zero and one Raspberry Pi B+, and one PC working as the cluster head. To investigate the performance of the framework, the time required to create and add a block to every blockchain is measured.

Our simulation process also measures two important parameters, namely the maximum message size transmitted during the authentication process to measure the bandwidth load used within the network and the maximum CPU load percentage required in each node for the authentication process to measure the computational load required for adding a block to the blockchain. The maximum message size is measured from within the node by adding a function that records the response size for each authentication response. This ensures that the actual data used by the authentication process is measured without the headers added by the API for transmission over the IP network. The maximum CPU percentage presents a measure of the load required by the application to authenticate a node, and it is monitored by the Docker daemon.

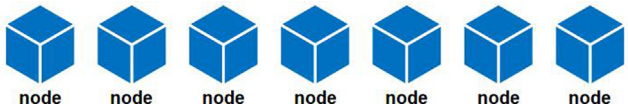

Fig. 15. Node arrangement for one level blockchain simulation.

4.2. Performance analysis

This section presents the simulation results and analyzes the performance of the proposed framework. We will present results and the performance analyses for the clustering process, the hierarchical blockchain performance and finally we analyze the security performance of the framework.

4.2.1. Clustering algorithm

The clustering algorithm was presented in Algorithm 1 where it uses the information available for each node to classify the nodes to three types based on their processing power, node energy reserve and the node's location. The nodes are classified into CHL2, CHL1 and ordinary node.

The processing power index in this example has 3 values {1,2,3} with the higher value indicating more capable device. In Function "selecCHL2" if Pi = 3, the node is added to the set (CHL2), there is no possibility for other nodes to be added to the set (CHL2). The node sends a message indicating that its statues is set to CHL2.

In Function "selectCHL1", there are two conditions, the first that the node energy is larger than the average energy reserve for the adjacent nodes to avoid selecting nodes with depleted energy reserve, the second is that the node is not selected as CHL2 to avoid duplicate values in the sets.

The nodes meeting the previous two conditions and has Pi = 2 are added to the set (CHL1). Then, the node sends a message indicating its status is changed to CHL1. Nodes that do not meet the defined conditions would wait for CH nomination message. The Function "joinCluster" calculates the distance between the node and the cluster heads adjacent to the node. The node would select the nearest cluster head. if there are no cluster heads within the nodes predefined area, the node is set as CHL1 cluster head.

The time complexity for algorithm 1 consists of the time complexity of the functions "selecCHL2", "selectCHL1" and "joinCluster" and depends on the number of nodes in the network N where the function "selectCH2" has two nested for loop and has complexity of $O(N^2)$. In addition, the function "selectCHL1" has two nested for loops and complexity $O(N^2)$. Finally, the function "joinCluster" has one for loop and complexity $O(N)$. The total complexity of the algorithm is $O(N^2)$.

The algorithm requires each node to store the data related to the adjacent nodes. The data size is estimated to be 535 bytes for each node. The total data size depends on the number of nodes N and is equal to 535 N Bytes.

For the clustering performance measure, the most relevant measurement is how efficiently the nodes are grouped to create the hierarchy of clusters explained in previous sections. The simulation is performed to test the efficiency of the clustering algorithm to reduce the blockchain size. By using Omnet++, the simulation is performed with the nodes randomly distributed in an area of $150m \times 150m$ with variable device density and each device is assigned a random value of energy reserve. For blockchain using the proposed consensus algorithm that is based on proof of authentication. Using multiple levels of blockchains has reduced the blockchain size required to be stored and handled by each device in the network. Fig. 17 shows that the blockchain size for each cluster is substantially reduced compared to using a single blockchain. It can be noted that the number of nodes in each blockchain is reduced the largest number of nodes are in level 0 blockchain and assuming a worst-case scenario of all the nodes joining a single cluster. The number of level 0 nodes is 3 for a network of 10 devices and rises to 59 for 135 devices network. Thus, the reduced number of the blockchain node ranges from 30% to a about 44%.

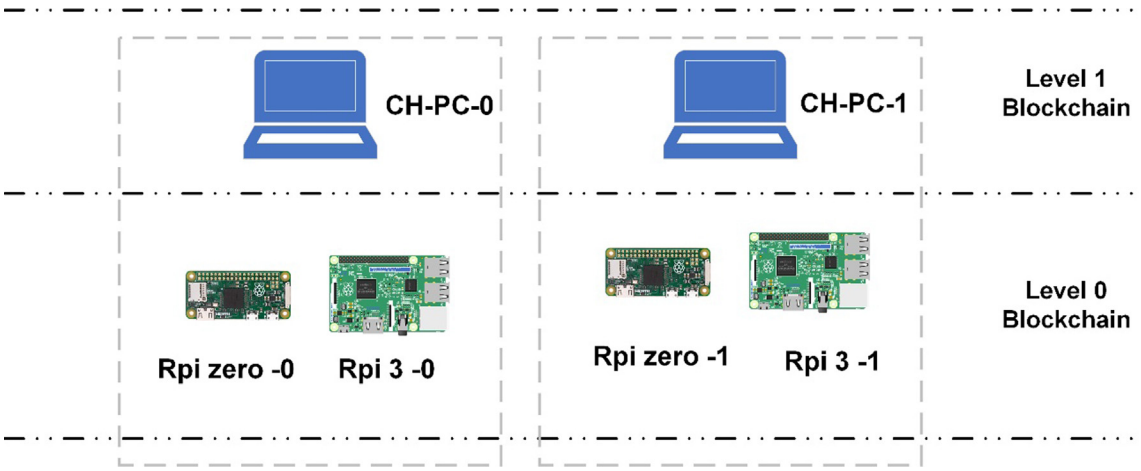

Fig. 16. Raspberry Pi network.

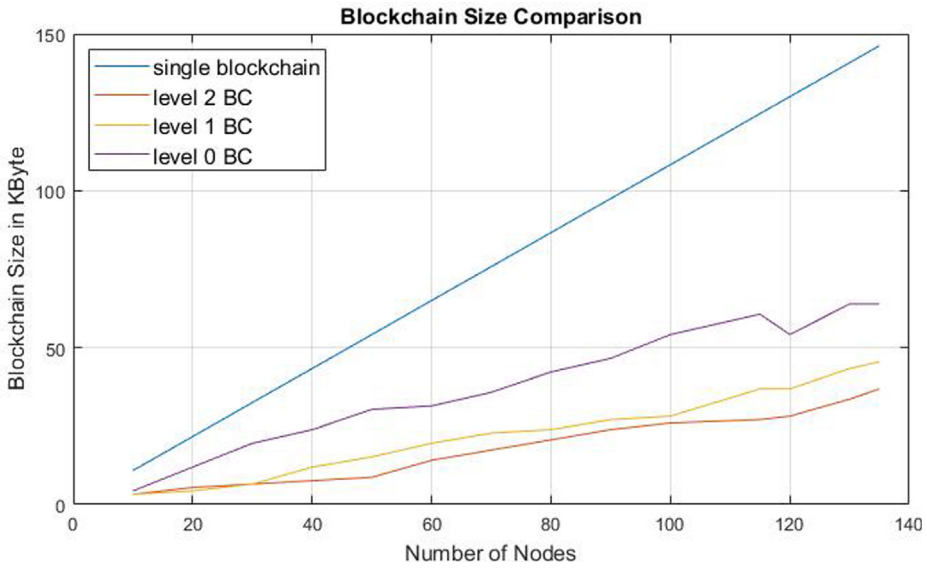

Fig. 17. The size of blockchains created for one level blockchain and multiple-level blockchain.

4.2.2. Blockchain performance

Referring to Algorithm 2, it is used to create the blockchain in each cluster and create a block containing the validated authentication request for each node in the cluster. If the cluster nodes did not create a blockchain, then the first block must be created by obtaining a hash value from the cluster head. If the node is not assigned as a cluster head and the initial value for the block index equals to zero, then the node requests the hash value from the cluster head.

If the block index of the last block is not zero, then there is a blockchain created by the nodes in the cluster. The new node would send a request to the other nodes in the cluster contain the signed values of the node id and the cluster head id. The simplified consensus algorithm only requires that the authentication request value is verified to create a new block.

When the new block is created and sent to the nodes in the cluster, the nodes validate the new block by validating that the previous hash value in the new block matches the hash values in their copy of the blockchain. Once validated, it will be added to the authentication table.

The time complexity of the algorithm consists of the hash time T_{hash}, signature time T_{sign}, and signature verification time T_{verf}. The time required to create a single block equals time to verify the signature T_{verf} and the time to hash the block contents T_{hash}, thus $T_{block} = T_{verf} + T_{hash}$.

The total time T_{total} is the sum of time for signing the authentication request T_{sign} , the time to create the block hash T_{hash} and the time to validate a number n of blocks in the cluster blockchain $n(T_{hahs})$. Thus, the total time $T_{total} = T_{sign} + T_{block} + n(T_{hash}) = T_{sign} + T_{verf} + (1+n)(T_{hash})$.

By adopting the values from [45] and [46], the time complexity of our proposed method can be approximated as presented in Table 2.

The data size of Algorithm 2 is dependent of the blockchain size and the authentication table size. Each block consists of the previous block hash with size $D_{prev-hash}$ bytes, the block index with size $D_{block-index}$ and the authentication request having a size $D_{authreq}$ bytes which contains the node id with size $D_{node-id}$, the cluster head id with size D_{ch_id} and the RSA signature with size D_{sign}. The total size of the authentication request is $D_{authreq} = D_{node-id} + D_{ch\_id} + D_{sign}$. This results in a block size equals to $D_{block} = D_{prev\_hash} + D_{authreq} + D_{block-index}$. The total blockchain size for a number of nodes n is $D_{total} = nD_{block}$. The storage requirements for n devices in a cluster are shown in Table 3.

Table 2
Time complexity values.

Time variable	Time in seconds	Source
T_{sign}	0.16	[45]
T_{verf}	0.02	[45]
T_{hash}	0.016	[46]
$T_{block} = T_{verf} + T_{hash}$	0.036	[45,46]
Total time T_{total}	(0.196 + n*0.016)	

Table 3
Data storage requirement for n devices in a cluster blockchain for the proposed framework.

Data size variable	Data size value
D_{hash}	32 bytes
$D_{node-id}$	128 bytes
D_{ch_id}	128 bytes
D_{sign}	128 bytes
$D_{authreq}$	128 + 128 + 128 = 384 bytes
$D_{block-index}$	1 byte
D_{block}	417 bytes
D_{total}	417n bytes

Table 4
Data storage requirement for the authentication table of n devices in the IoT network for the proposed framework.

Data size variable	Data size value
D_{hash}	32 bytes
$D_{node-id}$	128 bytes
D_{ch_id}	128 bytes
$D_{block-index}$	1 byte
D_{total} for first level nodes	289 n_1 bytes
D_{total} for second level nodes	289 $(n_1 + n_2)$ bytes
D_{total} for third level nodes	289 $(n_1 + n_2 + n_3)$ bytes

The authentication table fields have a size of $D_{auth-table}$ bytes and it includes device id, cluster head id, block index and block hash. The data stored in each field of the authentication table is equal to $D_{auth\_table} = D_{device-id} + D_{CH-id} + D_{block-hash} + D_{block\_index}$.

The authentication table has variable size depending on the level of the node in the hierarchical blackchin structure. For a network containing N devices that are arranged in three levels (nodes, CHL1 and CHL2) with number n_1, n_2, and n_3 respectively. Table 4 presents the authentication table data size for each level. The data size values for SHA256 and RSA keys and signature are from [45].

The proposed frame work has small storage requirement of 417 bytes for a single block compared to existing blockchain networks for example the bitcoin block size [47] is 1 mega Byte.

Fig. 18 shows that the time to create the multiple levels of blockchains is also reduced substantially because the blocks in each cluster are created simultaneously if the condition that the first block from the blockchain is validated from the upper-level cluster head in the network. Also, it can be noted that the average time required to create a single block based on the proposed algorithm is 0.031 s in docker containers, 0.267 for intel I7 PC, 0.3 s for intel Celeron PC and 0.302 s and 0.699 s for RPi 3 and RPi zero respectively. The block creation times when compared to other blockchain protocols are very short. The time values required to create a single block are presented in Table 5 for Bitcoin, Ethereum and proof of authentication protocol and the proposed protocol one devices with different processing powers.

It can be noted that the docker container running in docker network results are very low compared to other readings. This is expected because the docker network is virtual and the transmission delays are minimal. However, the times required to create a block using devices running in Local 2.4 GHz wireless network are also very small.

The performance of the framework is compared to a one level blockchain to analyze the efficiency of the framework. The maximum size of the messages sent by the nodes to achieve consensus is monitored by measuring the response size of the data sent by the

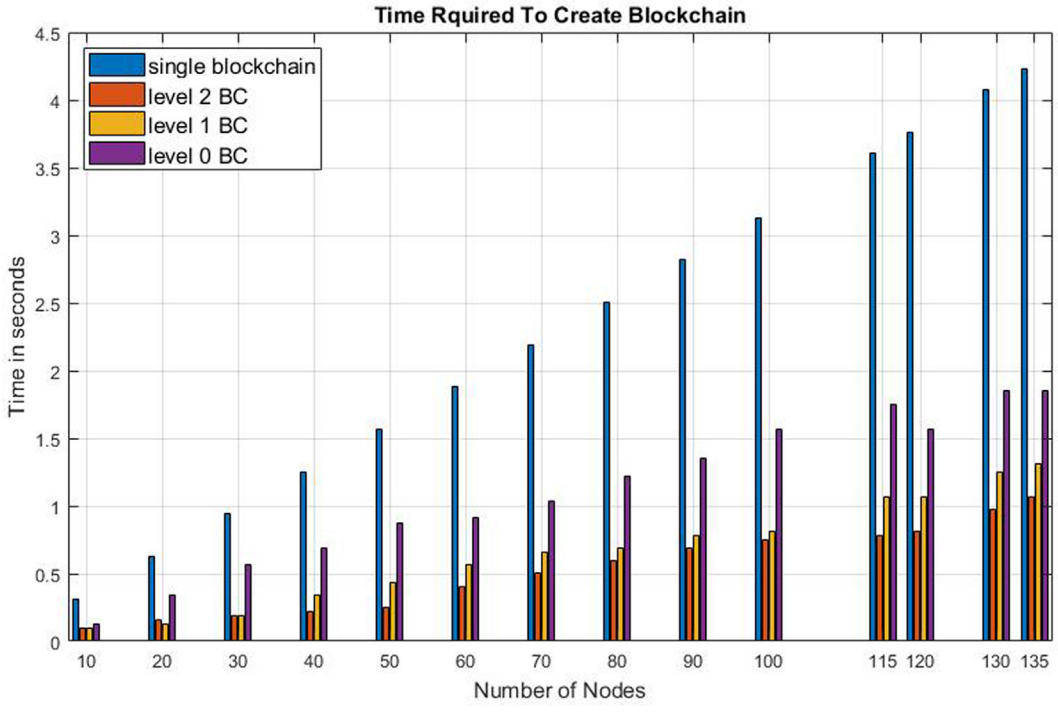

Fig. 18. Time required to create blockchain.

Table 5
Time required for creating a block.

Blockchain type	Time	Source
Bitcoin	10 min	[48]
Ethereum	10–20 s	[49]
Proof of Authentication blockchain	3.34 s	[25]
Proposed blockchain in Docker container	0.031 s	Measured
Proposed blockchain in Raspberry Pi 3B+	0.302543 s	Measured
Proposed blockchain in Raspberry Pi zero	0.699685 s	Measured
Proposed blockchain in Windows PC	0.267375 s	Measured
Proposed blockchain in Ubuntu PC	0.300687 s	Measured

consensus function in the node's API. The measured results of Fig. 19 show that the proposed framework in case of applying the hierarchical blockchain arrangement has maximum message size equals to 3.9 Kbyte while using the same consensus algorithm in a single level blockchain would produce maximum message size equals to 8.66 Kbyte. This means that the hierarchical blockchain has reduced the message size required for consensus within the cluster by approximately 45%.

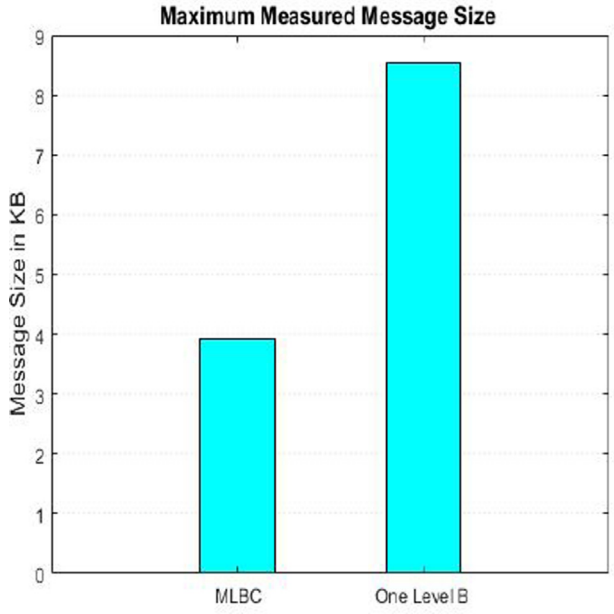

Fig. 19. Maximum message size measured for framework simulation.

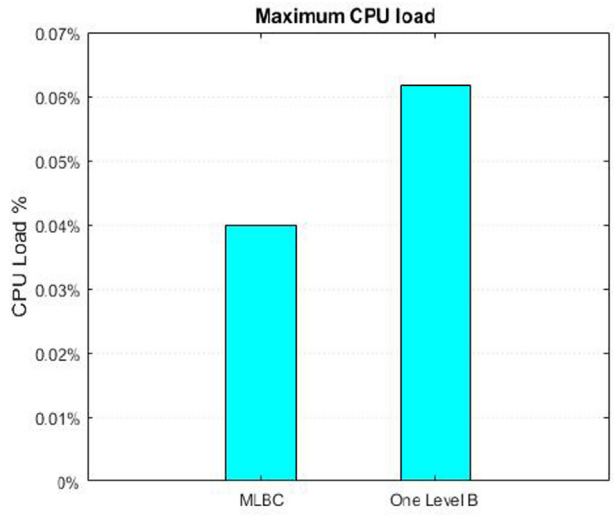

Fig. 20. Maximum CPU load measured for framework simulation.

The other performance measurement is maximum CPU load required to calculate the consensus in the cluster. Where the consensus for single blockchain required the use of 0.0616% of CPU load for intel I7 processor and the multilevel blockchain required 0.0398% of CPU load on the same processor. Fig. 20 shows that the proposed framework has 45.8% lower CPU load compared to one level blockchain.

4.2.3. Security analysis

To ensure the efficiency of the proposed framework, it is important to analyze its performance against the design requirements or goals stated in the earlier part of this paper (Section 3).

Decentralization: The proposed framework uses a distributed blockchain branch where each branch is independent from the other. The authentication decision is based on the consensus of each branch and is distributed via the authentication list. This ensures a decentralized authentication management of the nodes in the network.

Efficiency of resources requirements: The blockchain size for each node is reduced by about 30% compared to the blockchain size required by one level blockchain. It's also worth noting that the number of messages required for consensus for each block is reduced because the consensus is only between the nodes in each cluster and the remaining nodes in the network are not required to be part of the consensus. As shown by the simulation results, the message size for the authentication process is reduced by 45%, which indicates more efficient use of the network resources. Furthermore, our simulation also proved that the CPU load of the proposed framework decreases by 45.8%. This means that the proposed framework has lower demands in processing power. In addition, the time required to create the blockchain is reduced.

Anonymity: By using a combination of the block hash and the asymmetric key pair, the proposed framework ensures that each node is anonymous. The node is addressed by the public key and the block hash with no indication of its identity.

Scalability: The multi-level or hierarchical distribution of the nodes that creates separate and smaller blockchains ensures that the system is scalable. Moreover, it provides the scalability of nodes in different levels without compromising the security performance of the structure.

Security: The system security performance is analyzed against several well-known attacks affecting IoT networks and blockchain as follows:

Majority Attack: The suggested framework ensures a better level of security against the majority attack. Owing to its hierarchical structure, the probability of a successful majority attack on any level of the blockchain and on every blockchain in each cluster is independent of the probability of a successful majority attack in any other cluster. In order to demonstrate its efficiency, we denote the probability of a successful majority attack after z number of the blocks by [34]:

$$P_{successfulattack} = 1 - \sum_{k=0}^{z} \frac{\lambda^k e^{-\lambda}}{k!}\left(1 - \left(\frac{q}{p}\right)^{(z-k)}\right)$$

where

p is the probability that an honest node finds the next block.

q is the probability that the attacker finds the next block.

$\lambda = z\frac{q}{p}$ is the expected value of the Poisson distribution of the attacker's potential progress.

z is the number of blocks already added to the blockchain.

Since in our proposed framework, the blockchain of each cluster level is independent, the probability of successful majority attack for three-level clusters is given by:

$$P_{successfulattack} = P_{attackonlevel0} * P_{attackonlevel1} * P_{attackonlevel2}$$

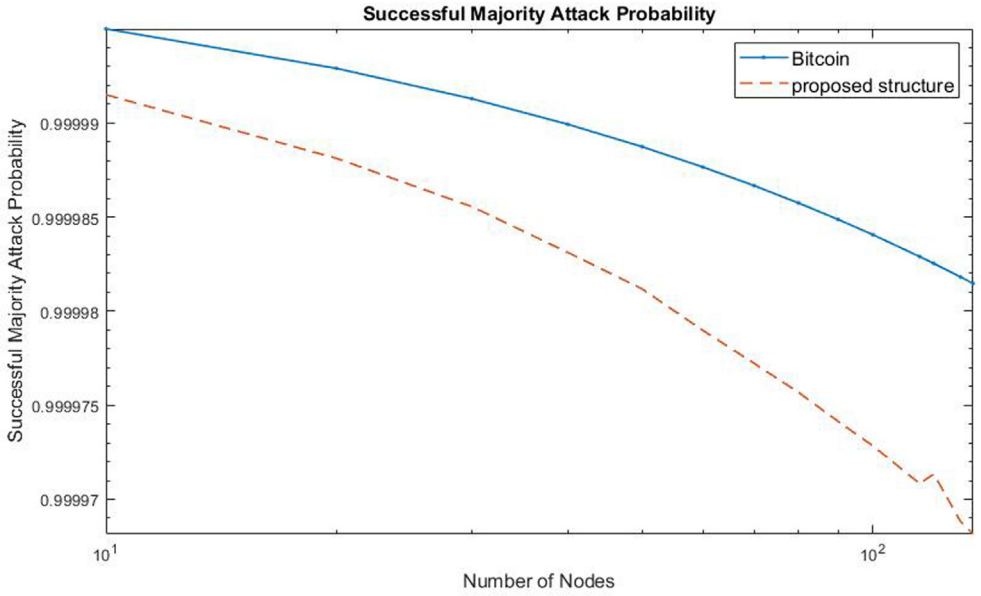

Fig. 21. Successful majority attack probability for a network containing 10 up to 135 nodes.

Fig. 21 illustrates the probability of successful majority attack for two blockchain networks notably Bitcoin (i.e., to represent conventional blockchain structure) and our proposed hierarchical blockchain structure. In our simulation, both networks contained 10 to 135 nodes. From the figure, it can be observed that the proposed framework offers a lower probability of successful attack than that of conventional blockchain.

Sybil Attack: Each node in the proposed framework has its unique identity which is the public key of the key pair generated from the node address. In addition, the node is authenticated based on the hash of the block that contains the authentication request to join the cluster. This ensures that faking node identity by an attacker is impossible.

Spoofing Attack: This attack happens when an attacker disguised as a trusted device to access the network. Nevertheless, such an attack is very difficult to be launched on the proposed framework where the nodes are defined by two values notably the node ID and the cluster-head ID. Besides, every node is authenticated and registered in the cluster blockchain with a unique hash value that can't be duplicated by the attacker.

Man in the Middle Attack (MITM): To initiate MITM, the attacker needs to achieve majority attack on two blockchains, calculate the authenticated hash and gain access to the private key of the node, which is very difficult to do in the proposed scheme.

Denial of Service (DoS): In the proposed framework, only authentication requests from a verified source are allowed in the network. This ensures that the nodes cannot be flooded with requests, furthermore the cluster structure ensures that the attacker must attempt DoS on multiple clusters at the same time which is very resource intensive.

From the above discussions, it can be concluded that the proposed framework offers comprehensive security protection to IoT networks. The framework also meets all the design objectives to efficiently ensure the security of the IoT network.

5. Conclusions

Blockchain technology can provide a decentralized solution for the IoT networks security requirements. The conventional blockchain application has proved to be resource demanding for resource constrained devices like the IoT devices. In existing literature, there are multiple suggested solutions for this constraint, but most of these solutions depend on existing blockchain platforms which introduce potential points of failure and inherit the constraints of the existing blockchain.

This paper introduced a novel blockchain based authentication framework for authenticating the IoT devices. The framework presented a method to reduce the computational load required to perform transactions in the blockchain by grouping IoT devices into "clusters" that are arranged in a hierarchical or multi-level structure, and each cluster creates a small blockchain to authenticate its members. Each cluster is connected to the larger blockchain by assigning a hash for starting the blockchain from the upper level blockchain.

Also, a lightweight and fast consensus algorithm where the nodes are validated based on their public key value and their related cluster head public key value is presented. This new consensus algorithm depends on the direct verification of the devices and the validation of the new blocks within each cluster. The proposed method had shown to be fast and lightweight without comprising the immutability characteristics of the blockchain.

The design has been simulated and tested using docker network and using Raspberry Pi network. The simulation results have shown that the proposed framework is lightweight and has lower computational and storage requirements compared to existing blockchain protocols. The analysis of the proposed structure shows that it is decentralized, efficient and immune to majority attack.

Currently, we are working on improving the framework adaptability to moving IoT devices by integrating the authentication values from the authentication table into smart contracts in Ethereum blockchain. Finally, a more comprehensive simulation with larger number of devices that addresses stationary and moving devices is being prepared to further validate the proposed work.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors extend their appreciation to the Universiti Putra Malaysia for partly supporting this research work through the project number (UPM.RMC.800-3/3/1/GP-GPB/2021/9701500).

References

[1] Chernyshev M, Baig Z, Bello O, Zeadally S. Internet of things (IoT): research, simulators, and testbeds. IEEE Internet Things J. 2018;5(3):1637–47. doi: https://doi.org/10.1109/JIOT.2017.2786639.
[2] Yang Y, Wu L, Yin G, Li L, Zhao H. A Survey on Security and Privacy Issues in Internet-of-Things. IEEE Internet Things J. 2017;4(5):1250–8. doi: https://doi.org/10.1109/JIOT.2017.2694844.
[3] Hassija V, Chamola V, Saxena V, Jain D, Goyal P, Sikdar B. A survey on IoT security: application areas, security threats, and solution architectures. IEEE Access 2019;7:82721–43. doi: https://doi.org/10.1109/ACCESS.2019.2924045.
[4] Yao X, Farha F, Li R, Psychoula I, Chen L, Ning H. Security and privacy issues of physical objects in the IoT: challenges and opportunities. Digit. Commun. Networks 2019;2020. doi: https://doi.org/10.1016/j.dcan.2020.09.001.
[5] Mukherjee S, Biswas GP. Networking for IoT and applications using existing communication technology. Egypt. Informatics J. 2018;19(2):107–27. doi: https://doi.org/10.1016/j.eij.2017.11.002.
[6] Soewito B, Marcellinus Y. IoT security system with modified Zero Knowledge Proof algorithm for authentication. Egypt. Informatics J. 2020;22(3):269–76. doi: https://doi.org/10.1016/j.eij.2020.10.001.
[7] Nashwan S. AAA-WSN: Anonymous access authentication scheme for wireless sensor networks in big data environment. Egypt. Informatics J. 2021;22(1):15–26. doi: https://doi.org/10.1016/j.eij.2020.02.005.
[8] Zhang Y, Kasahara S, Shen Y, Jiang X, Wan J. Smart contract-based access control for the internet of things. IEEE Internet Things J. 2019;6(2):1594–605. doi: https://doi.org/10.1109/JIOT.2018.2847705.
[9] Fernández-Caramés TM, Fraga-Lamas P. A review on the use of blockchain for the internet of things. IEEE Access 2018;6(May):32979–3001. doi: https://doi.org/10.1109/ACCESS.2018.2842685.
[10] Gadekallu TR et al. Blockchain for edge of things: applications, opportunities, and challenges. IEEE Internet Things J. 2021;4662(c):1–25. doi: https://doi.org/10.1109/JIOT.2021.3119639.
[11] J. Sengupta, S. Ruj, S. Das Bit, "A Comprehensive Survey on Attacks, Security Issues and Blockchain Solutions for IoT and IIoT," J. Netw. Comput. Appl., vol. 149, no. September 2019, p. 102481, 2020, doi: 10.1016/j.jnca.2019.102481.
[12] Lo SK et al. Analysis of blockchain solutions for IoT: a systematic literature review. IEEE Access 2019;7:58822–35. doi: https://doi.org/10.1109/ACCESS.2019.2914675.
[13] Ali MS, Vecchio M, Pincheira M, Dolui K, Antonelli F, Rehmani MH. Applications of blockchains in the internet of things: a comprehensive survey. IEEE Commun. Surv. Tutorials 2019;21(2):1676–717. doi: https://doi.org/10.1109/COMST.2018.2886932.
[14] Ferrag MA, Derdour M, Mukherjee M, Derhab A, Maglaras L, Janicke H. Blockchain technologies for the internet of things: research issues and challenges. IEEE Internet Things J. 2019;6(2):2188–204. doi: https://doi.org/10.1109/JIOT.2018.2882794.
[15] O. Abdulkader, A. M. Bamhdi, V. Thayananthan, F. Elbouraey, B. Al-Ghamdi, "A Lightweight Blockchain Based Cybersecurity for IoT environments," Proc. - 6th IEEE Int. Conf. Cyber Secur. Cloud Comput. CSCloud 2019 5th IEEE Int. Conf. Edge Comput. Scalable Cloud, EdgeCom 2019, pp. 139–144, 2019, 10.1109/CSCloud/EdgeCom.2019.000-5.
[16] Biswas S, Sharif K, Li F, Nour B, Wang Y. A scalable blockchain framework for secure transactions in IoT. IEEE Internet Things J. 2019;6(3):4650–9. doi: https://doi.org/10.1109/JIOT.2018.2874095.
[17] S. Huh, S. Cho, and S. Kim, "Managing IoT devices using blockchain platform," Int. Conf. Adv. Commun. Technol. ICACT, pp. 464–467, 2017, 10.23919/ICACT.2017.7890132.
[18] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "LSB: A Lightweight Scalable BlockChain for IoT Security and Privacy," pp. 1–17, 2017, [Online]. Available: http://arxiv.org/abs/1712.02969.
[19] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for IoT Security and Privacy : The Case Study of a Smart Home," no. March, 2017, 10.1109/PERCOMW.2017.7917634.
[20] T. Hardjono, N. Smith, "Cloud-Based Commissioning of Constrained Devices using Permissioned Blockchains," Proc. 2nd ACM Int. Work. IoT Privacy, Trust. Secur. - IoTPTS '16, no. May, pp. 29–36, 2016, 10.1145/2899007.2899012.
[21] Ouaddah A, Abou Elkalam A, Ait Ouahman A. FairAccess: a new Blockchain-based access control framework for the Internet of Things. Secur. Commun. Networks 2016;9(18):5943–64. doi: https://doi.org/10.1002/sec.1748.
[22] Hammi MT, Hammi B, Bellot P, Serhrouchni A. Bubbles of trust: a decentralized blockchain-based authentication system for IoT. Comput. Secur. 2018;78(June):126–42. doi: https://doi.org/10.1016/j.cose.2018.06.004.
[23] Jiang Y, Wang C, Wang Y, Gao L. A cross-chain solution to integrating multiple blockchains for IoT data management. Sensors (Switzerland) 2019;19(9):1–18. doi: https://doi.org/10.3390/s19092042.
[24] C. Qu, M. Tao, and R. Yuan, "A hypergraph-based blockchain model and application in internet of things-enabled smart homes," Sensors (Switzerland), vol. 18, no. 9, 2018, 10.3390/s18092784.
[25] D. Puthal, S. P. Mohanty, P. Nanda, E. Kougianos, and G. Das, "Proof-of-Authentication for Scalable Blockchain in Resource-Constrained Distributed Systems," 2019 IEEE Int. Conf. Consum. Electron. ICCE 2019, no. January, 2019, 10.1109/ICCE.2019.8662009.
[26] Cui Z et al. A hybrid blockchain-based identity authentication scheme for multi-WSN. IEEE Trans. Serv. Comput. 2020;13(2):241–51. doi: https://doi.org/10.1109/TSC.2020.2964537.
[27] Liu Y et al. Capability-based IoT access control using blockchain. Digit. Commun. Networks 2020;October. doi: https://doi.org/10.1016/j.dcan.2020.10.004.
[28] Xu C, Qu Y, Luan TH, Eklund PW, Xiang Y, Gao L. A light-weight and attack-proof bidirectional blockchain paradigm for internet of things. IEEE Internet Things J. 2021;4662(c):1. doi: https://doi.org/10.1109/jiot.2021.3103275.
[29] Agyekum KOBO, Xia Q, Sifah EB, Cobblah CNA, Xia H, Gao J. A proxy re-encryption approach to secure data sharing in the internet of things based on blockchain. IEEE Syst. J. 2021:1–12. doi: https://doi.org/10.1109/JSYST.2021.3076759.
[30] T. Li et al., BTS: A Blockchain-based Trust System to Deter Malicious Data Reporting in Intelligent Internet of Things, IEEE Internet Things J., vol. 4662, no. c, 2021, 10.1109/JIOT.2021.3085004.
[31] Seshadri SS et al. IoTCop: a blockchain-based monitoring framework for detection and isolation of malicious devices in internet-of-things systems. IEEE Internet Things J. 2021;8(5):3346–59. doi: https://doi.org/10.1109/JIOT.2020.3022033.
[32] Egala BS, Pradhan AK, Badarla V, Mohanty SP. Fortified-chain: a blockchain-based framework for security and privacy-assured internet of medical things with effective access control. IEEE Internet Things J. 2021;8(14):11717–31. doi: https://doi.org/10.1109/JIOT.2021.3058946.
[33] H. Cheng, Q. Hu, X. Zhang, Z. Yu, Y. Yang, N. Xiong, "Trusted Resource Allocation Based on Smart Contracts for Blockchain-enabled Internet of Things," IEEE Internet Things J., vol. 4662, no. c, 2021, 10.1109/JIOT.2021.3114438.
[34] Nakamoto S. Bitcoin: a peer-to-peer electronic cash system. Consulted. J. Gen. Philos. Sci. 2008;39(1):53–67. doi: https://doi.org/10.1007/s10838-008-9062-0.
[35] Ferrag MA, Shu L. The performance evaluation of blockchain-based security and privacy systems for the internet of things: a tutorial. IEEE Internet Things J. 2021;4662(c):1–25. doi: https://doi.org/10.1109/JIOT.2021.3078072.
[36] Wu M, Wang K, Cai X, Guo S, Guo M, Rong C. A comprehensive survey of blockchain: from theory to IoT applications and beyond. IEEE Internet Things J. 2019;6(5):8114–54. doi: https://doi.org/10.1109/JIOT.2019.2922538.
[37] Antonopoulos AM. A survey of blockchain security issues and challenges. Int. J. Netw. Secur. 2017;1919(55):653–9. doi: https://doi.org/10.6633/IJNS.201709.19(5).01.
[38] Cao B, Zhang Z, Feng D, Zhang S, Zhang L, Peng M, et al. Performance analysis and comparison of PoW, PoS and DAG based blockchains. Digit. Commun. Networks 2020;6(4):480–5. doi: https://doi.org/10.1016/j.dcan.2019.12.001.
[39] Oyinloye DP, Sen Teh J, Jamil N, Alawida M. Blockchain consensus: an overview of alternative protocols. Symmetry (Basel) 2021;13(8):1–35. doi: https://doi.org/10.3390/sym13081363.
[40] D. Puthal, S. P. Mohanty, V. P. Yanambaka, E. Kougianos, "PoAh: A Novel Consensus Algorithm for Fast Scalable Private Blockchain for Large-scale IoT Frameworks," pp. 1–26, 2020, [Online]. Available: http://arxiv.org/abs/2001.07297.
[41] "OMNeT++ - Simulation Manual." https://doc.omnetpp.org/omnetpp/manual/ (accessed Mar. 13, 2020).
[42] N. Prabhu, "ScholarWorks @ UMass Amherst Network Virtualization and Emulation using Docker , OpenvSwitch and Mininet-based Link Emulation," no. December, 2020.
[43] I. Miell, A.H. Sayers, Docker in Practice. 2009.
[44] "Welcome to Flask — Flask Documentation (1.1.x)." https://flask.palletsprojects.com/en/1.1.x/ (accessed Mar. 13, 2020).
[45] B. Schneier, Applied Cryptography, Second Edition: Protocols, Algorithms, and Source Code in C. 1996.
[46] Rachmawati D, Tarigan JT, Ginting ABC. A comparative study of Message Digest 5(MD5) and SHA256 algorithm. J. Phys.: Conf. Ser. 2018;978:012116. doi: https://doi.org/10.1088/1742-6596/978/1/012116.
[47] D. Khan, L. T. Jung, M. A. Hashmani, "Systematic literature review of challenges in blockchain scalability," Applied Sciences (Switzerland), vol. 11, no. 20. 2021, 10.3390/app11209372.
[48] G. O. Karame, E. Androulaki, S. Čapkun, "Double-spending fast payments in Bitcoin," Proc. ACM Conf. Comput. Commun. Secur., no. December 2016, pp. 906–917, 2012, 10.1145/2382196.2382292.
[49] A. H. Mohammed, A. A. Abdulateef, and I. A. Abdulateef, "Hyperledger, Ethereum and Blockchain Technology: A Short Overview," HORA 2021 – 3rd Int. Congr. Human-Computer Interact. Optim. Robot. Appl. Proc., 2021, 10.1109/HORA52670.2021.9461294.