Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2022) 30: 2355 – 2372
© TÜBİTAK
doi:10.55730/1300-0632.3943

Presenting a method to detect intrusion in IoT through private blockchain

Rezvan MAHMOUDIE[1], **Saeed PARSA**[2,*], **Amir Masoud RAHMANI**[1]
[1]Department of Computer Engineering, Branch of Science and Research, Islamic Azad University, Tehran, Iran
[2]Department of Computer Engineering, Iran University of Science and Technology, Tehran, Iran

Received: 04.05.2022 • **Accepted/Published Online:** 10.08.2022 • **Final Version:** 28.09.2022

Abstract: Blockchain (BC) has been used as a new solution to overcome security and privacy challenges in the Internet of Things (IoT). However, recent studies have indicated that the BC has a limited scalability and is computationally costly. Also, it has significant overhead and delay in the network, which is not suitable to the nature of IoT. This article aims at implementing BC in the IoT context for smart home management, as the integration of these two technologies ensures the IoT's security and privacy. Therefore, we proposed an overlay network in private BC to optimize its compatibility with IoT by increasing scalability and reducing network overhead and response delay. Devices with high-level resources (Computer, Mobile, etc.), named block managers, establish a two-dimensional overlay network that helps block managers to communicate with their neighbors' block managers. Their communication aims at managing the BC according to the trust and voting mechanism. Trust, encouragement, and penalties help the blocks to ensure accurate transactions. Our proposed private BC model provides the first solution for managing IoT transactions in overlay networks. Our experimental results showed that the proposed mechanism reduces packet overhead and delay in service delivery while increasing the BC's scalability in comparison to the state-of-the-art BC models. By limiting the number of effective block managers in voting, we could significantly reduce the average approval time of the blocks.

Key words: Blockchain, internet of things, overlay network, privacy, security, block manager

1. Introduction

The term 'IoT' was first introduced by Kevin Ashton in 1999. IoT was the idea of connecting the new RFID technology with the controversial issue of the Internet in the supply chain of the Procter & Gamble firm. After that, the MIT Auto-IDCentre presented its outlook of IoT in 2001, and then the International Telecommunication Union (ITU) officially introduced the technology in 2005 [1]. IoT is a novel approach to Information Technology (IT) encompassing all technical, social, and economic concepts. IoT projects have created an important effect on the Internet and the economy; predictions show that by 2025, we will have approximately 100 billion objects connected to IoT, which will affect over 11 trillion dollars in the global economy. Moreover, according to Gartner, the cost of IoT security around the world reached 1.5 billion dollars in 2018. This increased cost is due to corporate efforts to protect against IoT-based security threats. We will soon be observing the demand for tools and services that carry out penetration tests, software and hardware security assessments. Organizations are also interested in learning more about the impact of external communication on the network. The major factors lead to security costs reaching $3.1 billion by 2021 [2]. Many of the available advanced security frameworks are overly centralized and are unsuitable for the IoT. This paper argues that

*Correspondence: parsa@iust.ac.ir

BC is a vital technology for addressing the issues of security and privacy which arise from linking millions of daily tools and equipments to the Internet. Because of the centralization and resource consumption problems, traditional protection and privacy approaches are ineffective in IoT.

However, in the last decade, the centralized model has been fully functional.When the number of network nodes and transactions reaches millions and billions, respectively, the computational needs and costs will increase exponentially. IoT networks are susceptible to DoS/DDoS attacks. In such cases, the servers are the target of attacks, and since the traffic is overflowing, the target devices fail. This problem can have a devastating effect on the IoT ecosystem, especially if it has delicate tasks. Researches show that the IoT seeks to maintain security and privacy at a scalable and distributed scale. BC is an immutable timestamp ledger of blocks to store and share data in distributed manners [3]. BC technology, which is the basis of Bitcoin digital currency, can be effective due to the nature of distribution to address IoT's security and privacy challenges. BC technology creates secure mesh networks. In this case, IoT devices can be reliably connected and protected against threats such as disruption and forgery. With each node registered legally in the BC, the device can easily identify other devices without the need for central management and authorization. Millions of devices will be able to connect to the network without requiring additional resources [4]. To extract a block, several special nodes called miners try to solve a proof of work (PoW) puzzle, and the node that solves the puzzle, adds the block to the chain. Proof of activity (PoAc), proof of burn (PoB), proof of space (PoS) are other types of security checks. Moreover, the deterministic methods of practical Byzantine fault tolerance (PBFT) and federated Byzantine agreement (FBA) are used for the consensus mechanism [5]. If a hacker succeeds in interfering with a BC, they must change all the blocks in most nodes, rerun the proof of work procedure, which takes a long time, and gain control of more than half of the P2P network to accept a new block. This issue is virtually impossible, and it guarantees the cybersecurity of data [6]. Nevertheless, the presence of the BC on IoT is not accessible and the following critical challenges need to be addressed [7]:

1. BC consensus algorithms, including POW and POS, require significant computational resources that go far beyond the capabilities of most IoT devices.

2. The use of basic BC protocols creates high traffic and computational overhead, which is unsuitable for IoT devices with limited bandwidth.

3. Extraction of a block takes a long time, depending on the algorithms used in the BC, while in most IoT applications long delay in responding is not acceptable.

4. In a basic BC implementation, all blocks are broadcasted among all miners to be verified and controlled. This activity leads to issues like scalability. Broadcasting transactions across the network increase processing costs and become uncontrollable as the number of network nodes increases.

Therefore, our goal is to overcome the challenges of integrating BC technologies and the IoT in smart home management. To address this issue, we represent BC and voting techniques from direct neighbors. We have divided the communication of IoT devices into two layers. The first layer includes all low-level resource devices, and the second layer is an overlay network with high-level resources called block managers. In the second layer, each block manager and the neighboring block managers, four adjacent block managers mapped in the row and column of the block manager in a two-dimensional schema, participate in voting. For this purpose, a block manager is considered the manager of several devices in the first layer. They would help us enhance the speed of discovering attacks and decrease overhead, cost, and network delay. We use Omnet++ and Cooja

to run comprehensive simulations to test key performance parameters, such as processing time, latency, and cyber-attack resilience.
The following are the paper's main contributions:

1. Implementing BC in the IoT for smart home management, essentially integrating two technologies, BC and IoT, will ensure security and privacy.

2. We offer an overlay layer in private BC to optimize its compatibility with IoT by improving scalability and reducing network overhead and delay.

3. In our design, devices with high-level resources (Computer, Mobile, etc.) establish a two-dimensional overlay network on devices with low-level resources. Overlay network members manage the BC according to the trust and voting techniques from adjacent neighbors of blocks to reduce network traffic and overhead costs.

4. To address the problem of consensus algorithms in BC, we employ a time-based consensus algorithm.

The literature review is discussed in Section 2. Our proposed method is presented in Section 3. Detailed security analysis and assessment of the proposed algorithm performance are given in Section 4. Finally, Section 5 brings the paper to a close and outlines the future research.

2. Literature review

IoT is made up of a network of heterogeneous devices with hidden sensors [5].The majority of these devices are low-power, have minimal memory, and limited processing capacity; therefore, the most critical challenges of the IoT include: 1. limitations of resources, 2. mobility, 3. heterogeneity [5].

2.1. BC solutions for IoT security

The industry considers BC technology and the research community a superior technology that is critical for managing, controlling, and preserving the security of IoT devices. On the other hand, BC is widely used to provide reliability, authorized identity registration, ownership tracing, and product tracking. Approaches such as TrustChain [8] have been proposed to create reliable transactions using BC while preserving the accuracy and integrity of transactions in a distributed environment is challenging. The authors in [9] have categorized 18 uses of BC, four of which are in IoT. These four categories include: an immutable log of events, management of access control to data, trading of collected IoT data, and symmetric and asymmetric key management for IoT devices.

The authors in [10] propose a BC-based framework for industrial IoT (IIoT). IIoT devices can connect with both the cloud and BC networks using this architecture. Each IIoT device has a single-board computer system (SBC) with control and communication interface capacities for both cloud and Ethereum BC. BC smart contract applications for use in IoT have been examined by Chritidis et al. [11]. The authors explain how BC smart contracts can help and enable the sharing of IoT devices. Furthermore, the authors discuss how the IoT might benefit from BC networks in billing, transportation, supply chain management, and e-commerce. In their paper [12], the authors developed a multilayered BC architecture for receiving data from IoT devices and sharing it with organizations and individuals. The proposed architecture has three primary components: a data management protocol, a data storage system, and a messaging service. As the system is user-centered, there is no implicit or automatic reliance on either participant. While maintaining scalability, this control is complete and controls access, distribution, and system security, and user privacy. The main problem here is the high bandwidth and high memory required to store data in the BC. Intel has recently developed proof of elapsed time (PoET), a new consensus BC algorithm that combines with Hyperledger [13]. PoET is a consensus algorithm that runs on Intel processors in a trusted execution environment (TEE). A node must wait as long as a random time from a trusted and truly random range before storing a block in a BC. A time checker function confirms random time selections. After this period, the node can be added to the BC. In [14], the authors presented a new cryptographic base known as IoTA. By eliminating the block and exploration concepts, IoTA ensures that transactions are free and validation is quick. The "tangle," which is basically a directed acyclic graph (DAG), is the main innovation behind IoTA. A user must check two random transactions chosen by other users before sending a transaction. IoTA is a transaction encryption method in an IoT environment. Unlike Bitcoin, which has heavy and complex calculations, this method is light and tolerable. In [15], the authors use an overlay network to solve the problem of block unscalability.The clustering algorithm clusters the nodes. Each group has a cluster head in this method, and only the cluster heads are responsible for maintaining the BC. This algorithm reduces network traffic. As the number of nodes increases, the number of transactions increases; however, the number of verified transactions also increases. The lightweight scalable BC (LSB) uses a BC dissimilar to DAG used by IOTA. Thus, the LSB enjoys the inherent benefits of BC, including reliability and immutability. Moreover, the paper, as mentioned earlier, uses an algorithm called the distributed trust algorithm. In this algorithm, each cluster headfirst confirms all the transactions received from the other cluster heads. Still, over time, concerning the number of correct and verified transactions, the trust in the cluster heads increases. Hence, later, the number of transactions that need verification will be reduced. In [16], BC is used to create the IoT system. This article demonstrates how to use BC to control and configure IoT devices. Furthermore, the researcher has improved IoT security by utilizing public key infrastructure (PKI). Like public keys in Ethereum, the keys are stored and controlled using RSA public key encryption, and private keys are stored on individual devices. Because of its smart contract, Ethereum has been particularly chosen as the BC platform.

Authors in [18] have proposed a method for protecting the users' privacy in the smart home. Three different modules were used to implement the design. The data collector module gathers data from users in the smart home and transfers it to the data receiver module, which stores the data in two different data sets. To save privacy, the result-provider module monitors end-user's access to the data. This method makes sure that the actual user can only access the data. A general architecture and synchronization protocols are presented in [19], which allow synchronization of IoT endpoints to the BC with varying communication costs and reliable security levels. The author identifies modeling and traffic analysis generated by synchronization protocols and examines energy consumption and synchronization using numerical simulations.

The authors in [20] proposed an efficient certificate revocation scheme in VCS. The blockchain concept is introduced to simplify the network structure and distributed maintenance of the certificate revocation list (CRL). The proposed scheme embeds part of the certificate revocation functions within the security and privacy applications, aiming to reduce the communication overhead and shorten the processing time cost. Extensive simulations and analysis show the effectiveness and efficiency of the proposed scheme, in which the BC structure costs fewer network resources and gives a more economic solution against further cybercrime attacks [20]. The BC's powerful advantages increase IoT security and provide decentralized access to IoT data. (e.g., IBM Watson IoT Platform). Immutability also increases its power for detecting malicious actions. Smart contracts (self-executing scripts) are a cost-effective way to manage the exponentially growing number of smart things [21]. A more security-focused review of the BC technology in the context of IoT can be found in [21]. In [22], BC is used to prove transaction completeness and to enable the development of a decentralized IoT using private ledger to secure transactions between devices. To manage smart meter data, smart contracts have also been used in [23]. Similarly, [24] suggests using a BC-based system to manage firmware updates of IoT devices. [25] employs BC as a data storage system in a multitier IoT architecture to store access control data. In [26], employs BC and smart contracts to secure authorization requests to IoT resources. The main problem in the reviewed articles above is the bandwidth and high memory required to store data in the blockchain. On the other hand, in some articles, the classic blockchain has been used, which reduces the throughput of the blockchain in integrating with the IoT. Therefore, as an innovation, in this article, we have tried to reduce the number of devices participating in the approval of blocks by using a private blockchain and a hierarchical scheme in the IoT, so the bandwidth required to update the distributed hypeledger is reduced, and because block managers use a certain number of neighbors to vote in approving and recording transactions, thus reducing the amount of memory needed to store information at the network level. Another innovation of this paper is to increase the scalability of the blockchain in IOT using the voting technique. Moreover, the findings of this paper reducing the need for high bandwidth compared to public blockchain due to the use of private blockchain and reducing the number of block managers at the coverage network level.

3. The proposed method

The activity diagram of our proposed method for detecting intrusions and attacks on systems is illustrated in Figure 1. As shown, a service provider or a requester raises a request to a random block manager with the overlay network. A block manager or node is an entity in charge of controlling the BC. The communications are encrypted based on public and private keys [27]. Then, both the requester and the requestee (receiver) have public and private keys. A string request includes data, a requester signature, a requester public key, and a requestee public key. The requester signature is the encrypted hash of data. The hash is computed by LOCHA algorithm [28], and afterwards the hash is encrypted using the requester's private key. The public keys are used as identifiers. A block manager has a list of IoT devices' public keys. It answers a request when the requestee public key is in its own public keys list; otherwise, it shares the request with the neighboring block managers. Neighboring block managers are four adjacent block managers connected to a block manager on the overlay network.

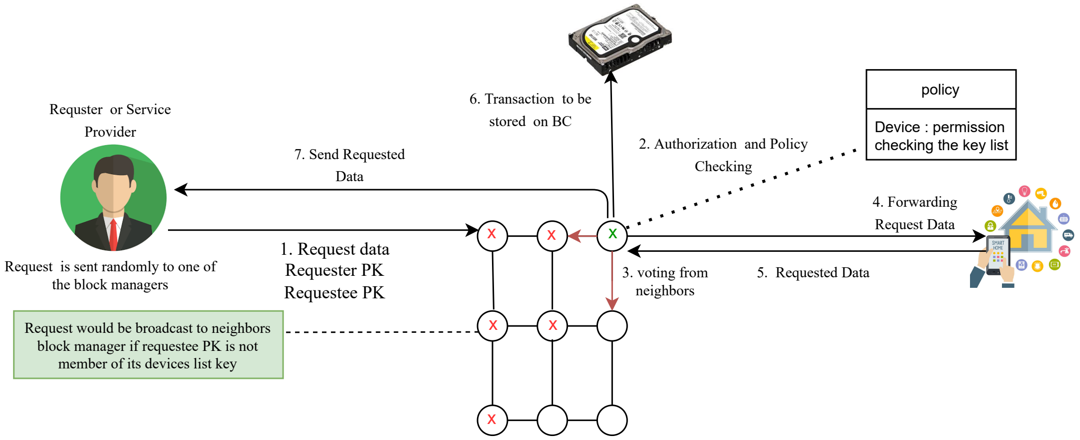

6. Transaction to be stored on BC
policy
Device : permission checking the key list
Requster or Service Provider
7. Send Requested Data
2. Authorization and Policy Checking
4. Forwarding Request Data
1. Request data
Requester PK
Requestee PK
Request is sent randomly to one of the block managers
3. voting from neighbors
5. Requested Data
Request would be broadcast to neighbors block manager if requestee PK is not member of its devices list key

Figure 1. The activity diagram of our proposed method.

When the request is delivered to the valid block managers, it is authenticated, and policy checks the request by voting mechanism. Further, the result of a verified request is stored as a transaction on BC and forwarded to the requester. The following section goes in-depth to describe the 1) structure, 2) voting mechanism, 3) block manager properties, 4) their controlling mechanism, 5) authentication, 6) valid transaction definitions, 7) transaction encryption method for IoT tools, 8) block manager authentication, and 9) detection of infiltrating and attack.

3.1. Overlay network structure

We map the overlay network as a 2D mesh (see Figure 2), block managers and edges clarify their communication channels on the overlay network. According to the row and column of mesh, four adjacent block managers connected to a block manager (according to the row and column of mesh) are defined as neighboring blocks of the block manager. These neighboring blocks participate in voting to detect intrusion signs of block managers. Limiting the number of voters increases the speed of detecting attacks and reduces the overhead and network delay. Block managers are devices with high-level resources to ensure scalability and are responsible for BC management because they must keep the policy and control mechanism of each IoT tool. Therefore, the IoT device should be defined as a member of a single block manager. Also, an IoT device only answers to the requests given from the defined block manager. A block manager handles the transactions of its members or IoT devices. However, a member can change its block manager while a period does not achieve any requests.

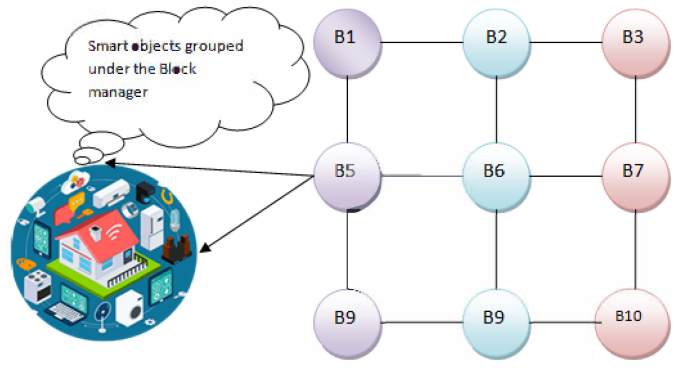


Figure 2. The arrangement of block managers in the overlay network.

3.2. Block manager structure

A block manager, also known as a node, is an entity in charge of managing the BC. This management involves production, verification, and the storage of transactions in the BC. A node chosen as a block manager should be able to stay online for a lengthy period of time and have enough resources to process transactions. The control fields of a block manager include:

- A set of network operations validated for the block manager to respond to a request (policy setting);
- The average exchange rate of the information;
- Public keys of the members;
- The amount of energy consumed;

- The weight value of neighboring block managers and itself (used for voting).

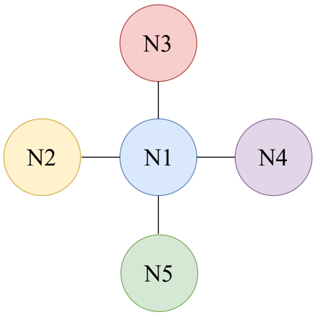
The block manager must confirm the following policy:

- The duties of block manager members clarify the type of operations. Sometimes, a specific operation may be assigned to a block manager.
- It would be better for each block manager to perform a limited number of operation types.
- The number of members of each block manager should be at least one and at maximum the half of the total number of network members.

Transactions generated by a block manager are encrypted and protected with asymmetric encryption, digital signatures, and hashing functions.

3.3. Arrangement of block manager neighbors in the overlay network

If all block managers are aware of each other's positions, there will be many delays and computational overheads. On the other hand, if we have a centralized control center, then we will have a centralized method that will not be suitable due to the existence of a single point of failure. To solve these problems, each block manager can be adjacent to N neighboring block managers. As shown in Figure 3, the proposed network structure includes two-dimensional BC managers. In this case, for each block manager, a maximum of four direct neighbors are considered. The BC applied in the overlay network is of exclusive type, so the BCs do not need to be identical. Using this technique also reduces synchronization costs.



$$\begin{bmatrix} 2 & 4 & 3 & 5 \\ -1 & 1 & -1 & -1 \\ -1 & -1 & -1 & 1 \\ 1 & -1 & -1 & -1 \\ -1 & -1 & 1 & -1 \end{bmatrix}$$

Figure 3. Arrangement of block manager neighbors and the related matrix.

3.4. Neighborhood matrix of block managers

As shown in Figure 4, each block manager, within the two-dimensional model of the overlay network, has a maximum of four neighbors. Therefore, a neighborhood matrix is an N*4 matrix that clarifies the neighbors of the block manager. N is the total number of block managers. Row ith clarifies the neighbors of ith block manager. We assumed that values of columns 1, 2, 3, 4 explain the neighbor block manager number of the block manager's left, right, top and downsides. In the case of unavailability, the value is –1. As an example, the following matrix states the neighborhood matrix of block managers shown in Figure 3.

3.5. Transaction structure

A transaction has header and data sections. The data section keeps the requested data and its answers. A header section has a set of control fields whose task is to control the performance of the transaction. The fields are mentioned below: 1. A transaction identifier;

2. A pointer to the previous transaction of the service provider;
3. Requester public key;
4. Requester signature (encrypted hash of requested data);
5. Requestee public key;
6. Requestee signature (encrypted hash of the answer of requested data) action.

Chains of service provider requests are created by the pointer to the previous transaction of the service provider. Thus, a hash, achieved from a transaction by LOCHA [28], is defined as a block in the BC structure.

3.6. Recording the block managers' authorized transactions

A block manager must perform its member's authorized transactions. A block manager has an authentication table. An example of this table is shown in Table . This table is shared with neighboring managers.

Table . A sample of authentication table inside a block manager.

Requester	Requestee	Transaction action	Operation
A	1	Access	Deny
B	2	Monitor	Deny
C	3	Exchange	Allow
D	4	Allow	Store

The requester, requestee, transaction action, and authentication operation are the table columns.

3.7. Encryption of block managers' authorized transactions

Transactions created by each block manager are encrypted using a light-weight one-way cryptographic hash algorithm (LOCHA). LOCHA generates a hash with a fixed and relatively small length. LOCHA satisfies all the properties of one-way hash (basic properties like collision resistance and preimage resistance) [28].

3.8. Weighting block managers for voting (distributed trust algorithm)

Most proposed solutions for IoT assume the presence of a trusted environment. But in worldwide, it is not right to ignore trust. Also, trust management is the fundamental requirement for network security solutions [8, 15]. As shown in Figure 4, in our proposed method, each block manager maintains the trust values of its neighbors. Trust is defined between a block manager, m, and a neighbor block manager, n. Figure 4 shows the activity diagram of an overlay network presented in Figure 3. As shown, the initial value of the trust for a block manager is configured to a default value arranged by the expert admin of the overlay network at the start or when a new block manager is entered.

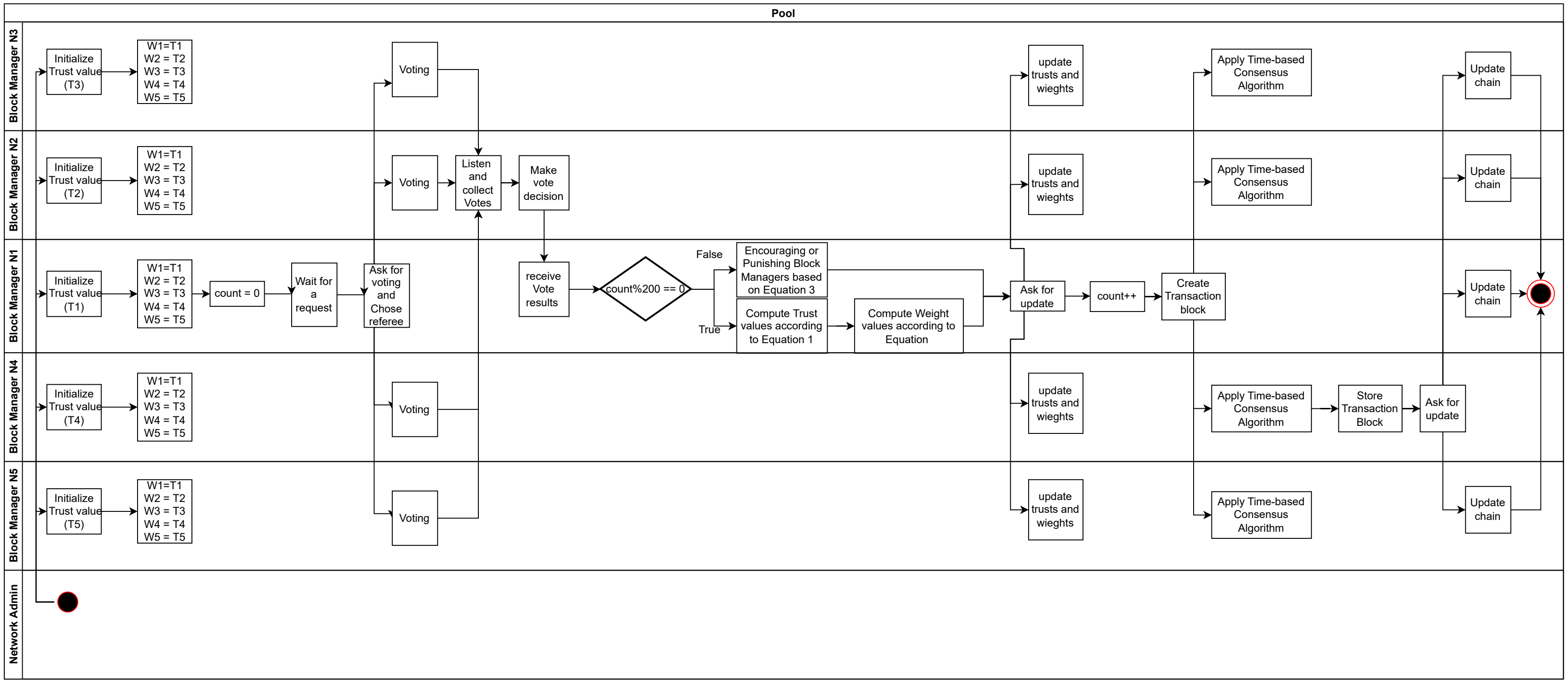

Pool
Block Manager N3
Block Manager N2
Block Manager N1
Block Manager N4
Block Manager N5
Network Admin
Initialize Trust value (T3)
Initialize Trust value (T2)
Initialize Trust value (T1)
Initialize Trust value (T4)
Initialize Trust value (T5)
W1=T1 W2 = T2 W3 = T3 W4 = T4 W5 = T5
count = 0
Wait for a request
Ask for voting and Chose referee
Voting
Listen and collect Votes
Make vote decision
receive Vote results
count%200 == 0
False
True
Encouraging or Punishing Block Managers based on Equation 3
Compute Trust values according to Equation 1
Compute Weight values according to Equation
Ask for update
count++
Create Transaction block
update trusts and wieghts
Apply Time-based Consensus Algorithm
Store Transaction Block
Ask for update
Update chain

Figure 4. The activity diagram of our voting mechanism for a request. Assumed N2 is selected as referee and N4 is computed for storing the obtained transaction.

If the number of block managers whose initial trust values are zero or one is greater than the half number of block managers, we recommend the initial trust values be set to 0.5. We are counting the number of requests. For every 200 requests, the block manager calculates trust and weight values according to Equations 1 and 2, respectively. Otherwise, the weights are updated based on Equation 3. Weight values are used within the voting process. Equation 1 computes the trust value while a new block manager is added in the overlay network, and their value is in the range [0,1]. The trust value of m with n is a division of ω with ς, where ω is the number of correct votes, queried by m, answered by the neighbor manager n, and ς is the total number of requests received by m.

$$\text{Trust}[\boldsymbol{m},\boldsymbol{n}] = \frac{\omega[m,n]}{\varsigma[m]} \tag{1}$$

Since block managers only keep the trust values of their neighbors, a small memory is required. Also, updating the trust value depends on the requester's satisfaction with the received service. Section 3.9.1 clarifies the updating progress. The weight is used in voting and is dependent on the trust value of block manager, m, and a neighbor block manager, n. The weight may change peer request and is calculated based on Equation 2, a fraction of the $Trust[m,n]$ by summation of all neighbor block managers' trust values of m, shown as $Sum[m]$.

$$\text{Weight}[\boldsymbol{n},\boldsymbol{m}] = \frac{\text{Trust}[m,n]}{\text{Sum}[m]} \tag{2}$$

3.9. The voting process either to determine the validity of a request or detect intrusion

While a block manager receives a request, its neighboring block managers vote to authenticate the request. The authentication process is described in subsection 3.6. Voters use the authentication process to vote whether a received request is valid or not. They encrypt the vote and send it to one of the neighboring block managers that the block manager randomly defined as a referee of this vote. The block manager sends the referee information to its neighboring block managers while sending a vote request. The vote is encrypted by homomorphic encryption [29, 30]. According to the aggregated homomorphic principle, the encrypted sum of information is equal to the sum of the encrypted information [29]. Therefore, this feature helps the referee to calculate the sum of votes with the voter's weights. If the sum result is greater than zero, then the referee states that the request is valid. Otherwise, the referee mentions that the request is not reasonable. The referee announces the result to the block managers and their neighbors. For a valid request, the block manager responds to the requester by forwarding the requestee's response. Also, the invalid request is draped, and its requester would have an extra negative point. A threshold on the negative points of a requester helps us to detect intrusions. If the threshold is triggered, the requester would be denied by the block manager. Also, the block manager alerts its neighbors to ban the requester.

3.9.1. Encouraging or punishing block managers

The neighbors are encouraged or punished when they act truly or unjustly, respectively. A neighbor performs truly when the referee selects its vote as the voting result. In contrast, a neighbor performs unjustly when its vote is not the result of the voting, that is, its vote is not a majority in the poll.

As shown in Equation 3, P percent of the weight of unjust neighbor, that indexed j, is subtracted for punishing. P is a value in a range $(0,1)$ randomly generated by the block manager indexed as i. A P value is conducted for all unjust neighbors and we name the summation of P percent of the weight of unjust neighbors as μ.

$$\mathrm{Weight[i,j]} = \mathrm{Weight[i,j]} - \mathrm{Weight[i,j]} * P \tag{3}$$

Equation 4 is used by the block manager, indexed i, to encourage each honest neighbor indexed k. Also, honest neighbors are the member of set l.

$$\mathrm{Weight[i,k]} = \mathrm{Weight[i,k]} + \frac{\mathrm{Weight[i,k]}}{\sum_{l} \mathrm{Weight[i,1]}} * \mu \tag{4}$$

In case where all neighbors are honest, μ would be equal to zero. Then the weight would be not changed.

3.10. Block chain

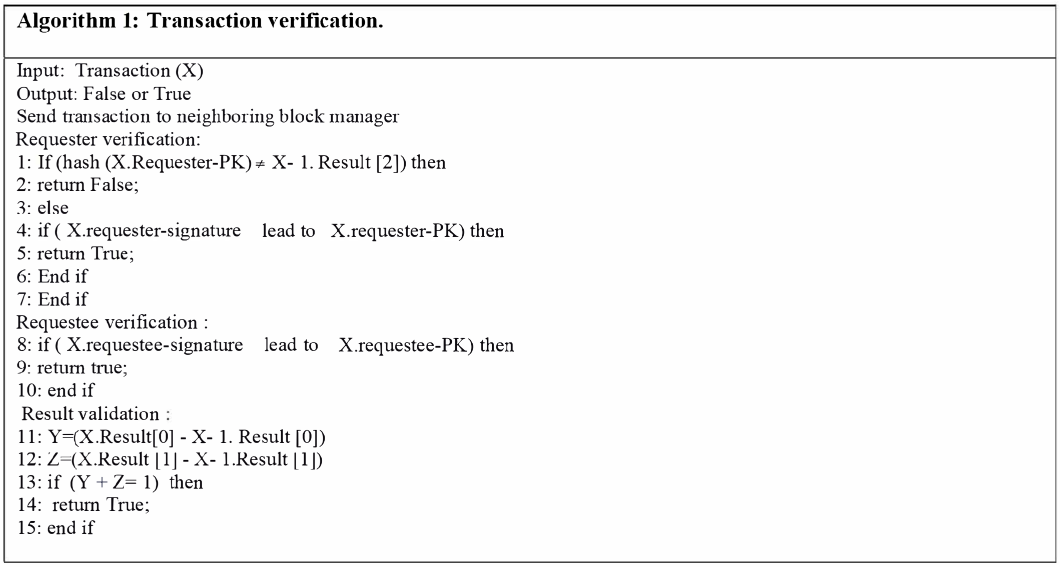
Transactions are stored as a block with the BC. Each block manager has a chain that keeps the blocks related to its neighbors' block managers. Also, the hash of blocks is computed by LOCHA [28].

3.11. Block validation

A neighbor block manager validates new blocks received from a block manager before adding the block into the BC. The public key of block managers is kept by themselves. Then, a block is identified as a valid block if, at first, the block manager is determined by its public key, and second, the transaction is confirmed based on the transaction fields and its hash. In the suggested algorithm, we confirm the transaction using the following form. The suggested transaction structure uses multiple control fields for validation in addition to the required data. The first element is a transaction identifier, while the second field is a pointer to the transaction that the same requester node was a part of before. As a result, a requester's transactions are linked together. The PK and signature of the requester and requestee are then included. When the requestee receives the transaction, the latter signature is added. The transaction output is the seventh field, which the requester sets. The following 3 entries are found in the result field: (1) the total number of requests made by the requester that the recipient has approved; (2) the total number of requests the recipient has denied; and (3) the hash of the PK that the recipient will use for the recipient's subsequent transaction. Algorithm 1 below illustrates the abovementioned procedures.

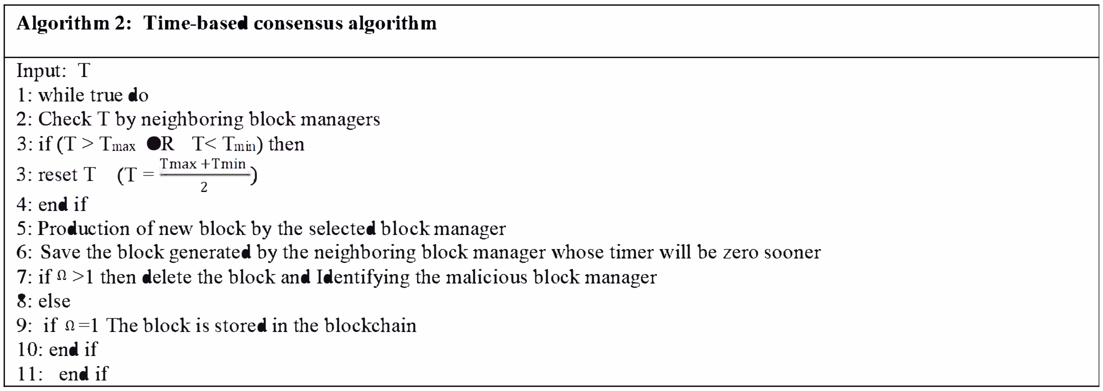
3.12. Time-based consensus algorithm

We present a time-based consensus algorithm to randomly choose a neighbor block manager that inserts the generated block into the BC to increase block security. This algorithm helps us to reduce the number of duplicate blocks that may be created concurrently. Furthermore, in the overlay network, the maximum waiting time (Tmax) is limited to twice the maximum point-to-point delay and the minimum waiting time (Tmin) is limited to the maximum point-to-point delay. When a valid request is identified, neighboring block managers wait a random amount of time. This operation is named a waiting time (T). Then, the neighboring block manager whose timer becomes zero earlier, stores the generated block. Also, it alerts other neighbors to update their chains according to the newly inserted block. If a block manager maliciously adds two or more blocks in a consensus period, other neighbors detect the block manager as a malicious block manager. Therefore, we limited the number of transaction blocks, which a neighbor block manager can store, to one for each transaction ($\Omega = 1$). Each block manager keeps track of how frequently other block managers create new blocks. Blocks that are in compliance are removed, and the trust attributed to the accountable block manager is reduced. The neighbouring block managers keep track of how frequently a block management generates new blocks at the beginning of the waiting time to stop them from constantly claiming to have a low waiting time. The block managers remove the block produced by their neighbour if the number of these blocks reaches a cutoff set depending on application by the BC designers. Algorithm 2 below illustrates the abovementioned procedures.



Algorithm 1: Transaction verification.

```
Input: Transaction (X)
Output: False or True
Send transaction to neighboring block manager
Requester verification:
If (hash (X.Requester-PK) $\neq$ X- 1. Result [2]) then
    return False;
else
    if ( X.requester-signature lead to X.requester-PK) then
        return True;
    End if
End if
Requestee verification :
if ( X.requestee-signature lead to X.requestee-PK) then
    return true;
end if
Result validation :
Y=(X.Result[0] - X- 1. Result [0])
Z=(X.Result [1] - X- 1.Result [1])
if (Y + Z= 1) then
    return True;
end if
```



Algorithm 2: Time-based consensus algorithm

```
Input: T
while true do
    Check T by neighboring block managers
    if (T > $T_{max}$ OR T< $T_{min}$) then
        reset T (T = $\frac{Tmax + Tmin}{2}$)
    end if
    Production of new block by the selected block manager
    Save the block generated by the neighboring block manager whose timer will be zero sooner
    if $\Omega$ >1 then delete the block and Identifying the malicious block manager
    else
        if $\Omega$=1 The block is stored in the blockchain
        end if
    end if
```

3.13. Malicious block manager detection

The following list clarifies the malicious block manager detection conditions:

1. If its public key is missed in the block validation process (see subsection 3.11).
2. If a block manager adds two or more transaction blocks in a consensus period (see subsection 3.12).
3. If a block manager does not react or respond within a time, T (In this case, neighbors randomly send requests that know their answer to verify the accuracy of the operations. If the correct answer is not received from the block manager at a particular time, this node will be removed from the network as a malicious node by voting from the neighbors).

3.14. Changing the malicious block manager

Suppose for a specified period, T, none of the members within a set of objects receive a transaction from the respective block manager. In that case, they can replace a new manager using the block manager selection algorithm. This algorithm uses four factors to select the block manager among the present managers in the network.

The selected manager has the least number of members (negative factor). The chosen manager has the greatest number of CPU resources (positive factor). The selected manager has the largest available memory (positive factor). The chosen manager has the highest amount of weight (degree of trust in the overlay network) (positive factor).

First, the decision matrix $(M * 4)$ is created and normalized to replace the malicious block manager. M is the number of available block managers. Each row of the matrix shows the values of the four factors of volunteer block managers. In the next step, columns are normalized based on Rumina [31]. Then, the average value of the normalized factors peer each volunteer block manager clarifies its score. Therefore, the member changes its block manager with another block manager that has the maximum score.

We have used the Rumina normalization method because the value of memory, CPU, and weight of the block manager are positive factors while the number of members is negative factors. The formula is shown in Equation 5. BM^+ is the maximum value is column of matrix. Also, BM^- is the minimum value within the column of matrix.

$$\mathrm{n}_{ij} = \begin{cases} \frac{BM_{ij}}{BM^+}, & \text{Positive factors} \\ \frac{BM^-}{BM_{ij}}, & \text{Negative factors} \end{cases} \tag{5}$$

For instance, suppose a member wants to change its block manager, and there are three block managers. Also, the member calculates the following decision matrix.

	m	c	D	W
1	4	2	10	0.7
2	2	8	10	0.6
3	8	8	40	0.4

Table 1. Matrix before normalization.

	m	c	d	W
1	0.5	0.25	1	1
2	0.25	1	0.5	0.85
3	1	1	0.25	0.56

Table 2. Matrix after normalization.

Therefore, the average values of these three block managers are 0.68, 0.55, and 0.7. Then, the third block manager is selected.

4. Assessment of the proposed algorithms

This section evaluates the quality and performance of the proposed algorithm against security threats. It is assumed that a malicious node can be a device in the smart home or a block manager in the overlay network. A malicious node can disrupt network communications, delete transactions, create fake transactions/blocks, and modify/delete stored data.

4.1. Evaluating the proposed algorithm using common IoT attacks

Based on common IoT attacks, we will examine the following attacks to evaluate the proposed method. We analyze the flexibility of the method presented in this study against any attack and the probability of an attack occurring according to the risk analysis criteria of the European Standards Institute (ETSI).

4.2. Assessment

The BC-based structure ensures security and privacy on IoT devices. Cooja [1] and Omnet++ [2] simulators are used to simulate the proposed method. Cooja is suitable for evaluating low-resource devices and can implement various IoT protocols. We can select, configure the desired type of sensor and examine the performance of sensors and actuators in the node. We use this method called ”basic-method-without-BC” in the implementation. We will use the IPv6 protocol and WLAN (6LoWPAN) as the primary communication protocol in the simulation. The number of nodes during the simulation varies from 5 to 60 nodes. The simulation is repeated several times, and results are presented on average during this time. The assessment is based on the following criteria:

Packed overhead: This factor relates to the length of packets transmitted after network encryption operations have taken place. This factor is calculated by examining access and store operations. Using encryption and hash increases the load volume of packets.

Time overhead: This factor refers to the processing time of each transaction within the block managers and is calculated from the time the block managers receive the transaction until the appropriate response is sent to the requester.

Memory overhead: This factor refers to the amount of memory used to store the BC. Given that not all block managers are required to keep the entire BC and only store the BCs related to the transactions of their direct neighbors, the amount of memory used is significantly reduced.

Processing overhead: This factor refers to the amount of time block managers take to confirm new transactions. This criterion is measured using a 60-node overlay network. Since the proposed structure confirms the block, it is unnecessary to broadcast all the blocks in the whole network. With the confirmation of four direct neighbors, this block will be confirmed so that the processing overhead will be reduced compared to the base model.

4.2.1. Evaluating the time overhead in the proposed algorithm

The proposed algorithm is compared with two other different methods. We simulate another method that the first algorithm is a basic algorithm without using a BC which is generally presented in the smart home structure on the market. In this model, the requester is communicating directly with the smart home manager without having to process the transaction. The second algorithm is the basic algorithm using BC, where the Bitcoin BC structure is utilized. This method broadcasts all blocks across the network. For simulating, 20 nodes are used in the overlay network. Figure 5 shows results for time overheads. The proposed algorithm takes a longer time to process packets than the basic algorithm without using a BC attributed to extra encryption and hashing operations. However, compared to the basic algorithm using BC, the proposed algorithm has a better function because it prevents broadcasting from taking advantage of the structure provided in the overlay network.

4.2.2. Evaluation of end-to-end delay in the proposed algorithm with the growing number of overlay network nodes

This section evaluates the end-to-end delay of BC managers in the overlay network for accessing or monitoring devices on the IoT. Delay is measured from the moment the request is generated to the moment the answer is received. We use the Omnet++ simulator with default settings composed of 15, 20, 25, 35, and 45 overlay nodes as block managers to perform the simulation. We have compared the proposed algorithm with a primary BC-

[1] http://anrg.usc.edu/contiki/index.php/CoojaSimulator.

[2] https://omnetpp.org/ [Online; accessed July-2019].

based method. Initially, the delay in the proposed algorithm was higher than the primary method. The higher delay can be attributed to the fact that the transactions need to be confirmed by the neighboring block managers. On the other hand, as the number of block managers increases, the proposed algorithm's performance enhances. This improvement pertains to the structure of the overlay network in which the broadcast of transactions is hindered. Figure 6 displays the simulation results.

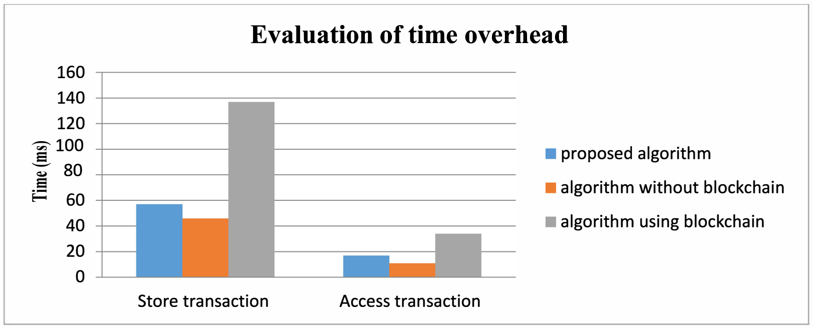


Figure 5. Evaluating the time overhead in the proposed algorithm.

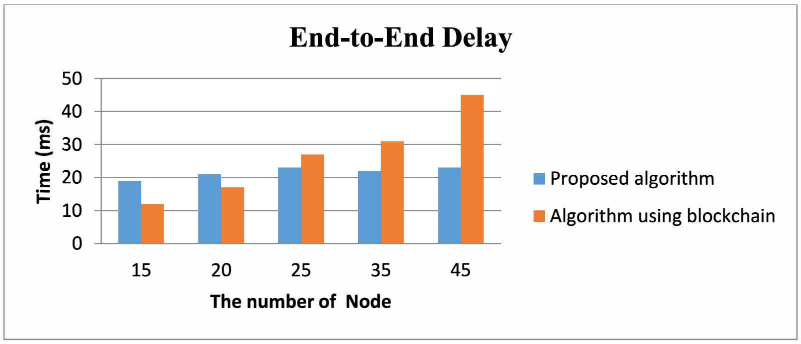


Figure 6. Evaluating end-to-end delay in the proposed algorithm.

4.2.3. Evaluation of the packed overhead

The simulation lasts 120 s since the volume of data and transactions is variable. Also, the results shown in the diagram are the results of an average of 10 runs. Figure 7 shows the results for packed overhead. Because in the basic algorithm using BC, the broadcast takes place at the level of the miners, so with the increase of the miners, the packed overhead explosively increases. However, the proposed algorithm keeps the packed overhead almost constant by controlling the nodes participating in voting.

4.2.4. Evaluation of average processing times in block managers to confirm the new transaction

According to the proposed algorithm, although the number of devices in the IoT increases, the average time to approve the block is reduced and remains almost constant since only the neighboring managers are involved in the approval and registration of blocks. When the network runs for a significant period, the proposed algorithm becomes stable. The processing time is reduced by more than 64 percent, compared with the basic algorithm using BC. Figure 8 shows the average processing time.

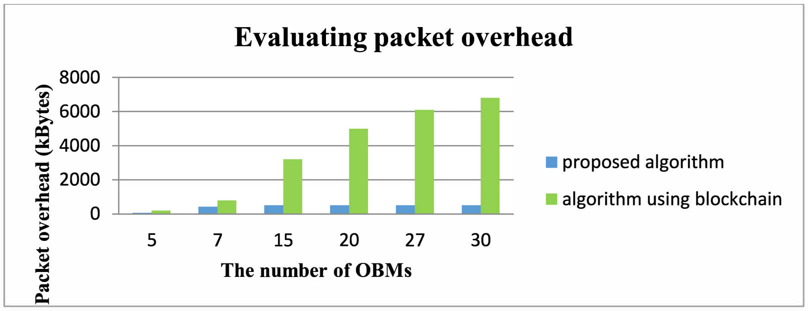


Figure 7. Evaluating packet overhead in the proposed algorithm.

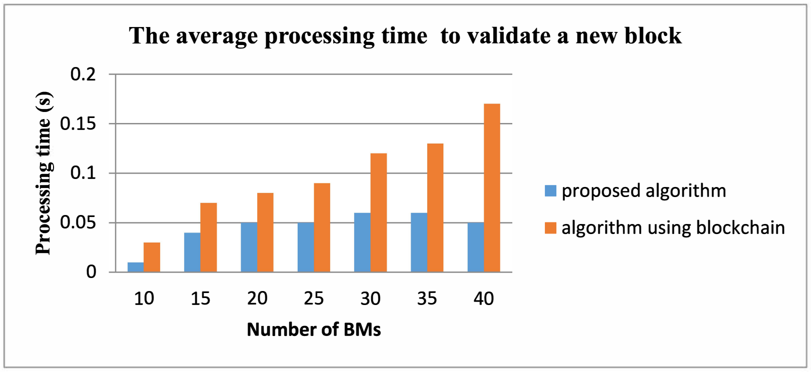


Figure 8. Evaluating the average processing time to validate a new block.

5. Conclusion

The major obstacles to integrating the BC with the IoT are bandwidth, the complexity of consensus algorithms, scalability, and packet overhead. Our proposed private BC removes these barriers and significantly increases IoT security. IoT requires fast network communications with lots of accessibility rules. To address the problem, we propose an overlay network on high-level resources, which may act as a block manager, to communicate with their neighboring block managers about accessibility rules by performing voting techniques. We offer a voting-based trust method to validate requests and block manager activities by computing the trust value of each neighboring block manager. Our simulation model experiments, implemented based on Omnet++ and Cooja, show that the proposed algorithm is highly secure against a wide range of attacks and reduces the response delay with lower network communication. Also, the bandwidth usage, processing time, memory, and power consumption are compared to the classical BC, and a significant reduction in their value is observed.

References

[1] Madakam S, Ramaswamy R, Tripathi S. Internet of Things (IoT): A literature review. Journal of Computer and Communications. 2015; 3 (5):164. doi: 10.4236/jcc.2015.35021

[2] Gubbi J, Buyya R, Marusic S, Palaniswami M. Internet of Things (IoT): A vision, architectural elements, and future directions. Future generation computer systems. 2013;29 (7):1645-60. doi: 10.1016/j.future.2013.01.010

[3] Kosba A, Miller A, Shi E, Wen Z, Papamanthou C. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In2016 IEEE symposium on security and privacy (SP) 2016; 22: pp. 839-858. IEEE. doi: 10.1109/SP.2016.55

[4] Tschorsch F, Scheuermann B. Bitcoin and beyond: A technical survey on decentralized digital currencies. IEEE Communications Surveys & Tutorials. 2016;18 (3):2084-123. doi: 10.1109/COMST.2016.2535718

[5] Kouicem DE, Bouabdallah A, Lakhlef H. Internet of things security: A top-down survey. Computer Networks. 2018; 141:199-221. doi:10.1016/j.comnet.2018.03.012

[6] Moradi J, Shahinzadeh H, Nafisi H, Gharehpetian GB, Shaneh M. Blockchain, a sustainable solution for cybersecurity using cryptocurrency for financial transactions in smart grids. In2019 24th Electrical Power Distribution Conference (EPDC) 2019;19 : pp. 47-53. IEEE. doi: 10.1109/EPDC.2019.8903713

[7] Salah K, Rehman MH, Nizamuddin N, Al-Fuqaha A. Blockchain for AI: Review and open research challenges. IEEE Access. 2019; 7:10127-49. doi: 10.1109/ACCESS.2018.2890507

[8] Otte P, Devos M, Pouwelse J. TrustChain: A Sybil-resistant scalable BC, Future Generation Computer Systems. (2017).

[9] Conoscenti M, Vetro A, De Martin JC. Blockchain for the Internet of Things: A systematic literature review. In2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA) 2016; 29 : pp. 1-6. IEEE. doi: 10.1109/AICCSA.2016.7945805

[10] Bahga A, Madisetti VK. Blockchain platform for industrial internet of things. Journal of Software Engineering and Applications. 2016 ;9 (10):533-46. doi: 10.4236/jsea.2016.910036

[11] Christidis K, Devetsikiotis M. Blockchains and smart contracts for the internet of things. Ieee Access. 2016 ; 4:2292-303. doi: 10.1109/ACCESS.2016.2566339

[12] Hashemi SH, Faghri F, Rausch P, Campbell RH. World of empowered IoT users. In2016 IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI) 2016; pp. 13-24. IEEE. doi: 10.1109/IoTDI.2015.39

[13] Chen L, Xu L, Shah N, Gao Z, Lu Y et al. On security analysis of proof-of-elapsed-time (poet). InInternational Symposium on Stabilization, Safety, and Security of Distributed Systems 2017 ;pp. 282-297. Springer, Cham.

[14] Popov S. The tangle. White paper. 2018;1(3) :30

[15] Dorri A, Kanhere SS, Jurdak R, Gauravaram P. LSB: A Lightweight Scalable Blockchain for IoT security and anonymity. Journal of Parallel and Distributed Computing. 2019 ;134: 180-97. doi: 10.1016/j.jpdc.2019.08.005

[16] Axon L. Privacy-awareness in BC-based PKI. 2015. [Online]. Available: http://goo.gl/3Nv2oK

[17] Panikkar S, Nair S, Brody P, Pureswaran V. ADEPT: An IoT Practitioner Perspective, Draft Copy Foradvance Review. IBM. 2015.

[18] Chakravorty A, Wlodarczyk T, Rong C. Privacy preserving data analytics for smart homes. In2013 IEEE Security and Privacy Workshops 2013 ; pp. 23-27. IEEE. doi: 10.1109/SPW.2013.22

[19] Danzi P, Angjelichinoski M, Stefanović Č, Popovski P. Distributed proportional-fairness control in microgrids via blockchain smart contracts. In2017 IEEE International Conference on Smart Grid Communications (SmartGridComm) 2017 Oct 23 (pp. 45-51). IEEE. doi: 10.1109/SmartGridComm.2017.8340713

[20] Lei A, Cao Y, Bao S, Li D, Asuquo P et al. A blockchain based certificate revocation scheme for vehicular communication systems. Future General Computer Systems 2020; 110: 892–903.

[21] Minoli D, Occhiogrosso B. Blockchain mechanisms for IoT security. Internet of Things, 2018: 1–13.

[22] Miller D. Blockchain and the internet of things in the industrial sector. IT professional. 2018 ;20 (3):15-8. doi: 10.1109/MITP.2018.032501742

[23] Huh S, Cho S, Kim S. Managing IoT devices using blockchain platform. In2017 19th international conference on advanced communication technology (ICACT) 2017; pp. 464-467. IEEE. doi: 10.23919/ICACT.2017.7890132

[24] Lee B, Lee JH. Blockchain-based secure firmware update for embedded devices in an Internet of Things environment. The Journal of Supercomputing. 2017;73 (3):1152-67.

[25] Hashemi SH, Faghri F, Rausch P, Campbell RH. World of empowered IoT users. In2016 IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI) 2016; pp. 13-24. IEEE. doi: 10.1109/IoTDI.2015.39

[26] Alphand O, Amoretti M, Claeys T, Dall'Asta S, Duda A et al. IoTChain: A blockchain security architecture for the Internet of Things. In2018 IEEE wireless communications and networking conference (WCNC) 2018; pp. 1-6. IEEE. doi: 10.1109/WCNC.2018.8377385

[27] Lu K, Qian Y, Guizani M, Chen HH. A framework for a distributed key management scheme in heterogeneous wireless sensor networks. IEEE transactions on wireless communications. 2008;7 (2):639-47. doi: 10.1109/TWC.2008.060603

[28] Chowdhury AR, Chatterjee T, DasBit S. LOCHA: a light-weight one-way cryptographic hash algorithm for wireless sensor network. Procedia Computer Science. 2014 ;32: 497-504. doi: 10.1016/j.procs.2014.05.453

[29] Huszti A. A homomorphic encryption-based secure electronic voting scheme. Publ. Math. Debrecen. 2011; 79 (3-4):479-96.

[30] Cramer R, Gennaro R, Schoenmakers B. A secure and optimally efficient multi-authority election scheme. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 103–118. Springer, Heidelberg (1997)

[31] Fakher H, Panahi M, Emami K, Peykarjou K, Zeraatkish S. New Insights into Development of an Environmental – Economic Model Based on a Composite Environmental Quality Index: A Comparative Analysis of Economic Growth and Environmental Quality Trend. Environmental Energy and Economic Research, 2021; 5 (3): 1-24. doi: 10.22097/eeer.2021.280746.1192