IJECBE (2024), **2**, **1**, 115–127
Received (5 June 2023) / Revised (1 December 2023)
Accepted (14 December 2023) / Published (30 March 2024)
https://doi.org/10.62146/ijecbe.v2i1.12
https://ijecbe.ui.ac.id
ISSN 3026-5258

International Journal of Electrical, Computer and Biomedical Engineering

RESEARCH ARTICLE

Evaluation of Smart Home Platform Based on Blockchain

I Gde Dharma Nugraha* and Hosea Yoarana

Department of Electrical Engineering, Faculty of Engineering, Universitas Indonesia, Depok, Indonesia
*Corresponding author. Email: i.gde@ui.ac.id

Abstract

A smart home consists of various sensors and actuators that can communicate with each other. In addition, various home appliances have similar capabilities. Therefore, the smart home became one of the Internet of Things (IoT) applications. This application utilizes multiple devices provided by various vendors. Hence, there are different challenges for the Smart Home Applications. One of the challenges is security. The difficulty of updating device firmware and the presence of illegal devices inserted into the network can cause security problems in IoT. Blockchain has been proposed as one solution to address these challenges. In this paper, we investigate and evaluate the performance of the blockchain-based solution for smart home IoT. Our experiment results show that the Blockchain can secure the IoT-based smart home, but it took 47% longer for packet delivery.

Keywords: Internet of Things, IoT, Centralized, Blockchain, Smart Home

1. Introduction

Smart Home is one popular application for the Internet of Things (IoT). The smart home is equipped with various sensors and actuators that can communicate with each other. The sensors are responsible for capturing the data inside and outside of the house. Meanwhile, the actuators are accountable for responding accordingly. The sensors and actuators use communication technology to exchange data so each device can understand the house's situation.

In addition, various home appliances inside the home are also equipped with sensors, actuators, and communication technology. Then, the ideal situation is that these devices can communicate and exchange data with the smart home to provide the services that fit the user's needs. This ideal situation becomes the challenge in the

smart home scenario, where it incorporates various devices from various vendors and has multiple standards and communication formats.

Some challenges of the IoT-based smart home are security and data validation [1, 2]. Security challenges arise because of communication technology as a medium for exchanging data between devices. The wrong people will likely steal information from the IoT device network. Also, the data may be tampered with or broken in the way, so it becomes not valid. The invalid information may result in the home's intelligent system making the wrong decision [3].

Therefore, various solutions have been proposed to improve the security of the IoT-based smart home. The proposed solution utilizes the network protocol's security standards and enhances the Gateway's security. The Gateway is the door for IoT devices to communicate through the internet. However, the usage of the secure network protocol is complicated. The usage of the Gateway also becomes a problem when the Gateway fails. Hence, the proper solution is needed to solve the security problem of the IoT-based smart home.

This study proposed a solution based on the Blockchain (BC) to address the security challenge. With the blockchain approach, we use a hierarchical structure. This hierarchical structure consists of three levels: IoT devices, network overlay, and cloud storage. In the network overlay, more than one Gateway is incorporated to protect and verify the data. Each Gateway acts as the hub manager for the set of IoT devices. This Gateway then becomes the BC network member, mining the valid data into the distributed ledger. This Gateway is located in the cloud storage to manage the storage dynamically. Then, we evaluate the performance of the proposed solution using the STRIDE model and examine the network performance.

The remainder of this paper is organized as follows. Section II describes the main component of the design. Section III describes the implementation and the setup used. Section IV presents the results of the measurements. Section V analyzes the speed and security of the system, and Section VI concludes this paper.

2. Main components of the Proposed Solution

2.1 Internet of Things (IoT)

Internet of Things (IoT) refers to the interconnection of smart devices to collect data and make intelligent decisions [4]. The Internet of Things, or IoT, refers to the billions of physical devices worldwide now connected to the Internet, collecting and sharing data. Thanks to the arrival of cheap computer chips and wireless networks, everywhere is becoming part of the IoT. Connecting all these different objects and adding sensors to them will add a level of digital intelligence to devices that are usually called "stupid" [5], enabling them to communicate real-time data without involving humans. The Internet of Things makes the world order more intelligent and responsive by combining the digital and physical worlds.

The idea of adding sensors and intelligence to basic objects was discussed throughout the 1980s and 1990s [6]. IoT allows users to manage and optimize electronic and electrical equipment using the Internet. It is estimated that shortly, most communication will occur between computers and other electronic equipment linked to each other, and information will be exchanged between them, thereby reducing human in-

teraction. One of the main challenges in IoT is bridging the gap between the physical world and the information world [6]. Sensors perform this process in general [4]. They serve as an interface between the user and the equipment. Sensors collect raw physical data from real-time scenarios and convert them into a machine-understandable format to be easily exchanged between various "Things" [6].

The Internet of Things (IoT) has many applications, including smart grids, smart cities, and health management [7]. However, the increasingly invisible, dense, and widespread collection, processing, and dissemination of data during private life pose severe security and privacy concerns. Some of the intrinsic features of IoT amplify its security and privacy challenges, including lack of central control, heterogeneity in device resources, multiple attack surfaces, context-specific risks, and scale [8].

2.2 Blockchain (BC)

Blockchain is a distributed ledger or digital ledger of cryptographically signed transactions grouped into blocks [9]. Blockchain is a type of database [9]. BCs collect information together in clusters, also known as blocks, which hold collections of information. Blocks have a specific storage capacity and, when filled, are chained to previously filled blocks, forming a data chain known as a "Blockchain" [9]. All the new information that follows the newly added block is compiled into a freshly formed block, which will also be added to the chain once filled.

In the Blockchain, each node has a complete record that has been stored on the Blockchain since its inception [8]. For Bitcoin, data is the entire history of all transactions. If one node has an error in its data, it can use thousands of other nodes as a reference point to correct itself. In this way, no single node in the network can change the information stored on it. Because of this, the transaction history in each block that makes up the Blockchain cannot be changed.

Blockchain networks can be categorized based on their permission model, determining who can issue blocks [9]. There are two categorizations in the Blockchain network:

1. Permissionless: is a decentralized ledger platform open to anyone who issues blocks without requiring permission from any authority [9]. In permissionless, users can create private addresses and then interact with the network by helping the network validate transactions or simply sending transactions to other users. The first type of permissionless Blockchain is Bitcoin [9]. Users can interact with the network by participating in the mining process. It is the process of solving complex mathematical equations and then using them to validate transactions.
2. Permissioned: refers to a network where users who publish blocks must be authorized by a specific authority (either centralized or decentralized) [9]. So, there is one difference from permissionless: users will interact with the network. Permissioned systems are also known to have limits on consensus, leaving allowed networks to be configured in such a way and fully controlled by the owner. A permissioned blockchain network can be used with open-source or closed-source software.

There are many benefits of permissioned, which make it the most preferable to use compared to permissionless. First, its efficient performance, when compared to

permissionless blockchains, its performance and performance is better. The main reason behind this is the limited number of nodes on the platform [7]. Removing unnecessary computations required to reach consensus on the network improves overall performance [9]. In addition, it allowed networks to have predefined nodes to validate transactions. Second, the proper governance structure, the permitted network does have the appropriate governance structure. So, it is usually configured as decentralized [9]. Permitted networks also use Blockchain properly, including leveraging its decentralized nature for data storage. Therefore cost-effective.

2.3 Cryptographic Hash

Hashing is one of the most important features of blockchain technology. Hashing enforces security measures on nodes that verify transactions and add blocks to the Blockchain via hash functions. The hash function is an essential mathematical operation used in blockchain platforms [10]. For example, it is required for proof of work, which is an essential component in forming consensus among miners on the Blockchain. It accepts inputs of various sizes and converts them into outputs of a specific size [10]. Useful for data authenticity and integrity, showing any changes to the input data. Therefore, manipulating the original data will cause a mismatch between the hash value of the manipulated data and the original data.

The cryptographic hash function often used in many blockchain implementations is the Secure Hash Algorithm (SHA), with an output size of 256 bits (SHA-256) widely supported by hardware. It makes SHA-256 fast to compute. SHA-256 has an output of 32 bytes (1 byte = 8 bits, 32 bytes = 256 bits), generally displayed as a 64-character hexadecimal string.

2.4 Addressing and Derivation

Some blockchain networks use addresses, which are short alphanumeric character strings derived from the public key of a blockchain network user using a cryptographic hash function, with some additional data (e.g., version number, checksum) [9]. Users connected to the same network know each other's addresses before they initiate any transfer. In a transaction, the "to" and "from" components are certainly known by both parties. With the asymmetric key method, we can use the public key to generate addresses, apply cryptographic hashes, and convert the hashes to text.

With some blockchain networks (especially permissionless ones), users must securely manage and store their private keys [9]. Network users often use software to keep themselves safe. This software can be referred to as a digital wallet. In addition to storing and managing keys, wallets are often used to calculate the number of digital assets owned. If the user loses the private key, then any owned digital assets associated with the private key will be lost, and the user will lose eligibility for the digital asset because it is not computationally feasible to generate the same and similar private keys. Figure 1 shows the general structure of the Blockchain.

2.5 Ethereum

Ethereum is open access to digital money and data-friendly service for everyone [11]. Ethereum is the community-built technology behind the cryptocurrency ether

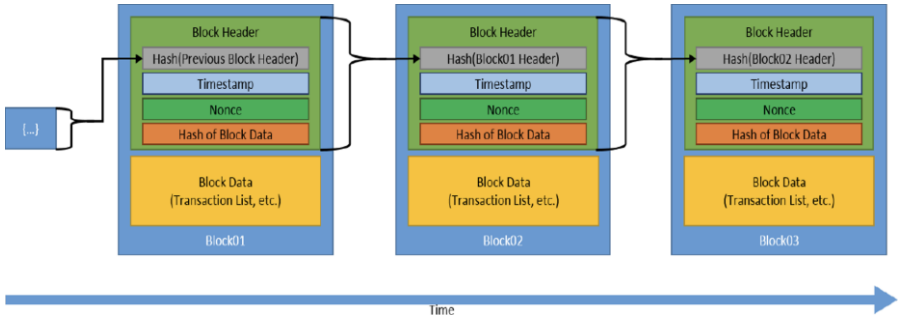

Figure 1. Generic Blockchain Structure

(ETH) and the thousands of applications today. Ethereum was first introduced in 2012 by Vitalik Buterin, a programmer from Toronto. Ethereum borrows heavily from the Bitcoin protocol and the underlying blockchain technology. The structure of the Ethereum blockchain is very similar to that of Bitcoin in that the two share a comprehensive transaction history. Every node in the network has a copy of the transaction. The main difference with Ethereum is that network nodes also log the current status of each transaction in addition to ether transactions. The network must track all Ethereum applications' status, user balances, intelligent contract codes, and storage locations. On the other hand, Bitcoin uses an unused transaction output protocol to track who owns how many bitcoins.

Ethereum is used in this research because it provides an open-source and easy-to-use framework. Ethereum is the second largest blockchain network, so many communities are involved in its development. Hence, we utilize ethereum in this research to evaluate its performance in the IoT use case.

2.6 *Smart Contracts (SC)*

Smart Contracts are self-executed contracts with the terms of the agreement between the buyer and seller written straight into lines of code. The code and agreements therein exist across a distributed and decentralized blockchain network. Code controls execution, and transactions are traceable and immutable. Nodes run SC in the blockchain network; all nodes running SC must get the same execution result, which is recorded on the Blockchain.

In Ethereum, SC is an Ethereum account type. SCs have balanced and sent transactions over the network (Community guides and resources | ethereum.org, n.d.). However, they are not controlled by the user. Instead, they are propagated to the network and executed as programmed. The user account can then interact with the SC by sending transactions that perform the specified functions on the SC. SC can define rules like regular contracts and automatically enforce them via code. SC can be used to create various Decentralized Applications (DApps), including games, digital collections, online voting systems, financial products, and many others [11].

SC is usually written in a high-level language, such as Solidity. However, they must be compiled to low-level bytecode that runs on EVM [11]. Once compiled, SC

is implemented on the Ethereum platform using custom contract creation transactions identified by being sent to a specific contract creation address. Each contract is identified by an Ethereum address derived from the contract creation transaction as a function of the originating account and the nonce. Ethereum contract addresses can be used in transactions as a recipient, sending funds to a contract, or calling one of the contract functions. On SC, the contract builder does not get any privileges at the protocol level (although it can be programmed explicitly in SC).

2.7 Azure Ethereum Cloud Nodes

The Azure cloud platform is a public cloud computing platform with services such as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) that can be used for services such as analytics, virtual computing, storage, networking, and much more [12]. Azure can be used as a replacement or addition to a local server [13]. One of the services from Azure is Ethereum Blockchain called Ethereum Blockchain as a Service (EBaaS) [14]. So that individuals and developers can have a cloud-based scalable Blockchain environment, it contains two tools that enable Smart Contract and BlockApps-based applications.

2.8 Ganache

Ganache makes it possible to create a private Ethereum blockchain to run tests, execute commands, and check statuses while simultaneously controlling how the chain operates [15]. Many developers use this to test smart contracts. Ganache provides easy-to-use tools such as advanced mining controls and a built-in block explorer.

3. Implementation

The initial step taken from this research is to determine the design of system requirements in software and hardware. The basis for determining specifications for testing the system used is ideally using a topology, as shown in Figure 2.

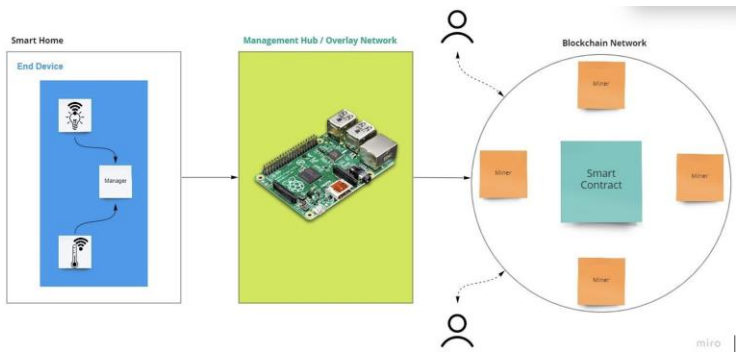

Figure 2. Overview of the Smart Home: Describes a decentralized access management system that uses BC technology to store and distribute access control information

All entities except IoT devices and Management Hub/Overlay Network nodes will be part of the blockchain technology. The nodes in the blockchain network must

include a copy of the BC. BC can be extensive and will continue to increase over time. Most IoT devices cannot store BC information due to its limited nature. As a result, BC is not included in the IoT device. Alternatively, it defines a new node called a management hub that requests access control information from the Blockchain on behalf of the IoT device. A manager interacts with the SC to define system access control policies.

BC Network in architecture is a private BC for the sake of simplicity. Private BC was used because it gives more definite results when evaluating the system. However, in real scenarios, public BC should be used to facilitate development. A private BC is a BC that anyone can read but only written by private nodes. Miners help keep the network safe and stable by approving transactions and keeping a copy of the Blockchain. Nodes can store and access specific device access control policies globally using the BC interface. This information is entirely decentralized and cannot be tampered with.

An edge gateway device is a management hub/overlay network that will function as an intelligent contract module, Hub and Agent, and edge node before entering the Azure Ethereum cloud node network as a blockchain network.

There are 6 phases for the transaction flow, as shown in Fig. 3:

1. Job Request: In the initial stage, when the leaf device sends data with a transaction, the leaf device will send a job request to the manager for further processing. This process will be handled by EdgeHub, which will act as an intermediary to create smart contracts and control the intelligent contracts against the IoT Hub.
2. Smart Contract Request: This stage logs and requests to create a smart contract to the Ethereum edge node.
3. Header Chain and other requests: Only transactions will be carried out at the edge gateway device. At this stage, a header chain will be given; if there are other requests, it will be given after the header. Mining and hashing are done in the next phase.
4. Hashing: The Ethereum cloud node will receive the header, which will be hashed and returned to the edge node. Inside the Ethereum cloud node, an Ethereum cluster has already been installed.
5. Smart Contract Acknowledge: After the digest value (the result of proof of work) is obtained, the smart contract will be approved. If the digest value given is different, it will be rejected.
6. Connect to IoT Hub: The last stage is when the smart contract has been approved and connected to the IoT Hub.

This implementation fulfills several prerequisites, including IoT Hub, IoT Edge Device, and End Device on Azure Portal. Then, before all three can be connected to the Raspberry Pi 4 device, a registry must be created that aims to be a third party between the Azure platform and the device. This registry can be created with Azure Container. After all the prerequisites mentioned have been made, it can be deployed into the Raspberry Pi 4.

After deploying the Raspberry Pi 4 to the Azure IoT Hub network, Smart contracts can be built using ganache-cli. Writing the following command at the command prompt will test ganache-cli to generate smart contracts.

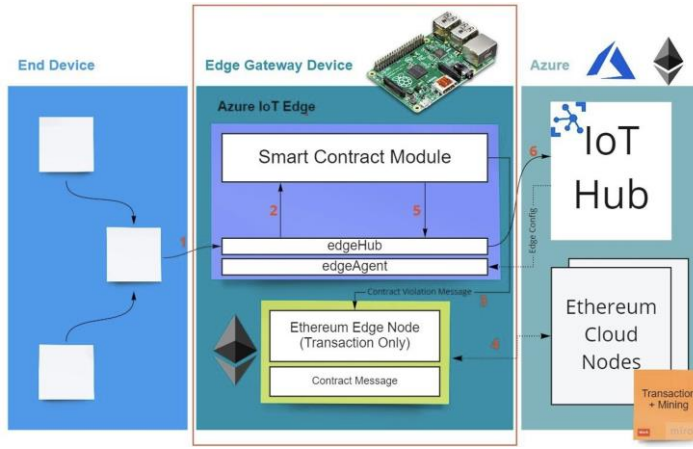

Figure 3. Smart Contract Module for Blockchain

Account opening needs to be performed before ganache-cli can be used. The user must register his device with ganache and then associate the device with a user. When the smart contract test succeeds, it can be connected to Azure.

4. Evaluation

The purpose of using Blockchain in IoT is to maintain high security without significantly reducing the speed of sending messages. The test scenario will analyze the device's performance using the created model. The Blockchain Network node can be checked using Go Ethereum (Get) on Azure Bash. Geth's function is to run Ethereum nodes implemented in Go Language. Geth will connect to an already implemented live BC. Geth also functions as a console to enter commands and perform certain functions, such as function eth.getTransactionReceipt("Hash[, callback]") will return the transaction receipt with the transaction hash. Then, the "from" data can be re-tracked on the logs that Azure IoT Hub has created.

4.1 Security Evaluation

There have been various studies on IoT and Smart Home security and privacy. Author [15] points out that IoT devices lack essential security protection by hacking into various Smart Home devices such as intelligent lights, smoke alarms, and CCTV. The author [16] argues that Smart Home is vulnerable to attacks by the user's smartphone even if the home gateway controls the exchange of packets to and from home.

The authors [17] propose three modules to protect user privacy in Smart Homes. The data collection module collects user data from the Smart Home and sends it to the receiving module, storing it in two different data sets. The results module controls user access to data to protect privacy. This method ensures that only the actual user can access the data. In addition, using two data sets can be guaranteed to link different user data with each other. However, this method does not provide data privacy when

Table 1. STRIDE Threat Categorization

	Threat	Property Violated	Threat Definition
S	Spoofing identify	Authentication	Pretending to be something or someone other than yourself
T	Tampering with data	Integrity	Modifying something on disk, network, memory, or elsewhere
R	Repudiation	Non-repudiation	Claiming that you didn't do something or were not responsible; can be honest or false
I	Information disclosure	Confidentiality	Providing information to someone not authorized to access it
D	Denial of service	Availability	Exhausting resources needed to provide service
E	Elevation of privilege	Authorization	Allowing someone to do something they are not authorized to do

the user needs to show his data to the service provider.

The method that will be used for testing is the STRIDE model. STRIDE evaluates the detailed design of the system. STRIDE is used to identify system entities, events, and system constraints. STRIDE applies a standard set of known threats by name, mnemonics, as shown in Table 1.

Security is critical in any management system. Although the system design aims to facilitate resource access control in a limited scenario, the solution must provide a satisfactory level of security. STRIDE classifies threats into six categories under abbreviations to identify threats in the system, as shown in Table 1. Although blockchain technology provides a certain level of security, such as data integrity and reliability, IoT devices are not part of the BC network and must rely on their access control decisions to the management hub node. The following forms of testing are carried out on the system:

1. Spoof: A malicious Management Hub can be deceptive (impersonating a management hub). Management Hub is essential because it becomes a liaison between End Device and BC Network. In addition, Management Hub serves as a provider of Smart Contracts for End Devices. Testing is done by experimenting with changing the identity of the IP address.
2. Tamper: Modify access control information sent to IoT devices directly from Smart Contracts.
3. Repudiation: Management Hub claims not to transmit data. Data is added, but the Hash process in brute force stops, so the digest value created does not match BC.
4. Denial of Service: DoS with unauthorized information from IoT devices. DoS is performed with a Distributed Denial of Service attack using Low Orbit Ion Cannon (LOIC) with High-Rate DDoS mode where infected IoT devices (Gateway and End devices) flood the BC network nodes.

4.2 Message Speed Evaluation

The test is carried out using the temperature sensor data transmission obtained from the dataset, which will then be tested for the speed of sending the message. Then, test the fee of each message. The test will be based on data obtained from the Edge Gateway Device. Testing is done by using a dataset. The data in the dataset is real-time temperature data that will be sent every 5 seconds and carried out within 1 hour. The data taken is the average message processing time in milliseconds and the packet overhead required to send the message (Blockchain-implemented devices). Packet

overhead is the length of the packet to be transmitted.

Table 2. Time Measurement Results Without Blockchain

No	Operation	Packet Overhead	Time Overhead
1	Transaction Period Client - Gateway	5 Bytes	20,139 ms
2	Transaction Period Gateway - Server	5 Bytes	50,356 ms
3	Access Transaction	5 Bytes	10,471 ms
	Average	5 Bytes	26,989 ms

As shown in Table 2, the average time for a device without the Blockchain is 26,989 ms. Time is measured in 3 parts:

1. Transaction Period Client – Gateway: The time it takes between the Client or end device to the Gateway. In this setting, the device periodically sends its data to the Gateway. This method is typical for various innovative home products today, such as thermostats.
2. Transaction Period Gateway – Server: Time required between Gateway to Server to confirm Smart contracts to Server Ethereum Cloud Nodes (Transaction + Mining).
3. Access Transaction: Access is granted when the Smart contract has been verified and can connect to the Azure IoT Hub.

The data obtained is measured in units of milliseconds. In IoT, every millisecond matters, especially for use cases like intelligent home monitoring. In Table 3, it is shown that the time required for the device with Blockchain is 39,606 ms—an average increase of 46,75% from the previous.

Table 3. Time Measurement Results of IoT Devices with Blockchain

No	Operation	Packet Overhead	Time Overhead
1	Transaction Period Client - Gateway	16 Bytes	30,256 ms
2	Transaction Period Gateway - Server	36 Bytes	69,912 ms
3	Access Transaction	16 Bytes	18,651 ms
	Average	23 Bytes	39,606 ms

5. Analysis

5.1 Security Analysis

Signed certificates can solve Spoof issues. Management hub nodes can get signed certificates from certificate authorities, and IoT devices can verify the authenticity of management hub nodes. Once an IoT device registers with another IP Address to the system for the first time, a malicious manager at BC can claim control of that device. However, IoT devices must verify registration under the manager before operations are accepted in BC. Otherwise, any manager can register any device under his control.

Queries from IoT devices to Management hub nodes are not transactions in BC for live performance, and BC loses the ability to verify which Management hub nodes correctly enforce access control rules. The information can be stored locally in any management domain where the IoT device resides. Since an IoT device can be part of different management domains over its lifetime, it will be spread across multiple nodes, making it difficult to audit and track it. BC can force the Management hub node to use transactions instead of queries for critical access control systems. This solution causes a reduction in performance but can increase security in the system.

The created design has a hierarchical defense against this attack. The first level of defense can be attributed to the fact that attackers are unlikely to install malware directly on Smart Home devices because these devices are not directly accessible. Miners must check all transactions. Suppose the attacker can still enter and infect the device. In that case, the second level of defense is that the miner must authorize all outgoing traffic by checking the access control policy headers. Since DDoS attack traffic requests will not be authorized, they will be blocked from leaving the Smart Home.

5.2 Message speed Analysis

Tables 2 and 3 compare the time it takes to send data from the end device to the Gateway and forward it to the Azure IoT Hub. The most significant time required by the device is the Transaction Period Gateway – Server of 50,356 ms and 69,912 ms for devices without and with Blockchain, respectively. It takes more time to connect to the IoT Hub because of the author's internet influence, and Ethereum Cloud Nodes also carry out the influence of mining and transaction calculations. To register a Gateway Device, the device must have an address from the mining pool provided by ganache-cli. The transaction can be carried out after obtaining the address from the aggregation. Access will be denied when the address requesting access does not match when re-searched through the returned address history or aggregation.

In both tables, the shortest time can be seen: the Access Transaction of 10,471 ms on devices without Blockchain and 18,651 ms on devices with Blockchain installed. Access Transactions can be granted when edgeHub has access to the IoT Hub. Data will go in and out of and to the Azure IoT Hub when this happens. The time required will be smaller because it is only intended to access and receive data from and to the Azure IoT Hub.

Packet overhead also affects the system's transaction speed. If we compare the packet overhead of devices with Blockchain and devices with Blockchain, it can be seen that there are more packets on devices with Blockchain.

Figure 4 shows the test results. The average of each operation gets a difference of approximately 47% longer. The method applied by the author causes this average to require more time to perform calculations–especially the mining process from Ethereum on the Azure IoT platform (Ethereum Cloud Nodes).

When compared in real-time, as in Figure 5, it can be seen that the time difference between devices without BC and devices with BC takes 18 – 69 ms when compared without BC, which takes 10 – 50 ms, so the difference between the two systems is only 8 – 20 ms apart. As a reference to the author of the paper (Zhou et al., 2018),

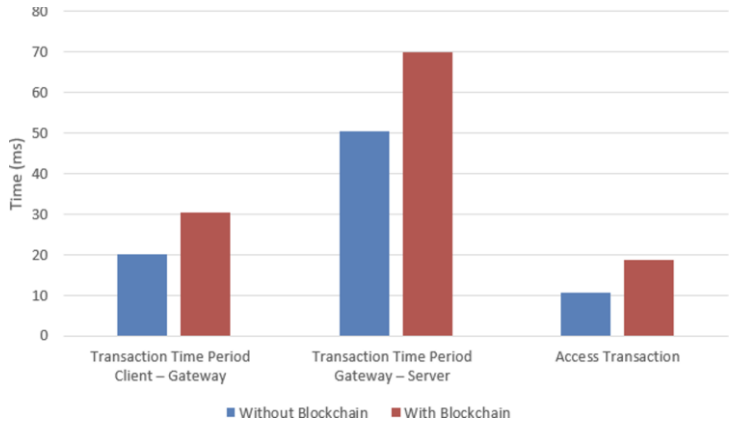

Figure 4. Device Comparison with and without Blockchain

it takes 27 – 38 ms to process the three operations with a difference of 9 – 31 ms. The time difference between devices with and without BC, as applied by the author, shows a number that is not very different. This time difference can also be affected by the internet connection used by the author.

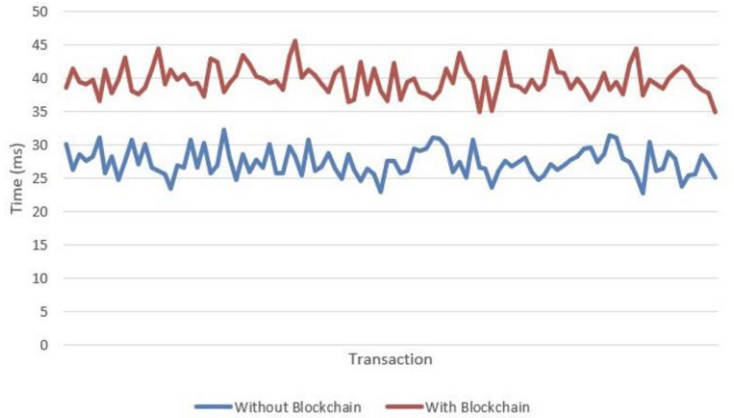

Figure 5. Comparison of 100 times Device transactions with and without Blockchain in real-time

6. Conclusion

This study carried out the Blockchain application with Ethereum Cloud Nodes as the Blockchain Network and Ganache-Cli as the Ethereum environment. Ethereum Cloud Nodes allow IoT devices to implement Blockchain with small resources. Ganache-Cli allows for the application of Ethereum as the basis for Blockchain.

Our experiment shows the effect of packet overhead on the time required to make a transaction. However, implementing blockchain technology can support data

transaction systems between IoT devices in a decentralized manner by implementing smart contracts. The result shows a decrease in speed of 38 – 40 ms in total for all operations in transaction processing. However, the decline is insignificant and worth considering the security and privacy benefits it offers.

References

[1] Meryem Ammi, Shatha Alarabi, and Elhadj Benkhelifa. "Customized blockchain-based architecture for secure smart home for lightweight IoT". In: *Information Processing & Management* 58.3 (2021), p. 102482. DOI: https://doi.org/10.1016/J.IPM.2020.102482.

[2] Samrah Arif et al. "Investigating smart home security: Is blockchain the answer?" In: *IEEE Access* 8 (2020), pp. 117802–117816. DOI: https://doi.org/10.1109/ACCESS.2020.3004662.

[3] Md Ashraf Uddin et al. "A survey on the adoption of blockchain in iot: Challenges and solutions". In: *Blockchain: Research and Applications* 2.2 (2021), p. 100006. DOI: https://doi.org/10.1016/J.BCRA.2021.100006.

[4] Alfonso Panarello et al. "Blockchain and iot integration: A systematic survey". In: *Sensors* 18.8 (2018), p. 2575. DOI: https://doi.org/10.3390/S18082575.

[5] Lijing Zhou et al. "Beekeeper: A blockchain-based iot system with secure storage and homomorphic computation". In: *IEEE Access* 6 (2018), pp. 43472–43488. DOI: https://doi.org/10.1109/ACCESS.2018.2847632.

[6] Priya Suresh et al. "A state of the art review on the Internet of Things (IoT) history, technology and fields of deployment". In: *2014 International conference on science engineering and management research (ICSEMR)*. IEEE. 2014, pp. 1–8.

[7] Ali Dorri, Salil S Kanhere, and Raja Jurdak. "Towards an optimized blockchain for IoT". In: *Proceedings of the second international conference on Internet-of-Things design and implementation*. 2017, pp. 173–178. DOI: https://doi.org/10.1145/3054977.3055003.

[8] Ali Dorri et al. "Blockchain for IoT security and privacy: The case study of a smart home". In: *2017 IEEE international conference on pervasive computing and communications workshops (PerCom workshops)*. IEEE. 2017, pp. 618–623. DOI: https://doi.org/10.1109/PERCOMW.2017.7917634.

[9] Dylan Yaga et al. In: *Blockchain Technology Overview* (Oct. 2018). DOI: https://doi.org/10.6028/nist.ir.8202.

[10] Fuqin Wang et al. "An experimental investigation into the hash functions used in blockchains". In: *IEEE Transactions on Engineering Management* 67.4 (2019), pp. 1404–1424. DOI: https://doi.org/10.1109/TEM.2019.2932202.

[11] Ethereum.org. *Intro to ethereum*. Accessed: (September 7, 2021). URL: https://ethereum.org/en/developers/docs/intro-to-ethereum/.

[12] Kgremban. *Azure Iot Hub Documentation*. Accessed: (September 7, 2021). URL: https://learn.microsoft.com/en-us/azure/iot-hub/.

[13] *What is Azure-Microsoft Cloud Services | Microsoft Azure*. Accessed: (September 7, 2021). URL: https://azure.microsoft.com/en-us/overview/what-is-azure/.

[14] Marley Gray. *Ethereum blockchain as a service now on Azure: Azure blog: Microsoft Azure*. Accessed: (September 7, 2021). May 2023. URL: https://azure.microsoft.com/es-es/blog/ethereum-blockchain-as-a-service-now-on-azure/.

[15] *Ganache*. Accessed: (September 7, 2021). URL: https://www.trufflesuite.com/docs/ganache/overview.

[16] Joshua E Siegel, Sumeet Kumar, and Sanjay E Sarma. "The future internet of things: Secure, efficient, and model-based". In: *IEEE Internet of Things Journal* 5.4 (2017), pp. 2386–2398.

[17] Ying-Tsung Lee et al. "Privacy-preserving data analytics in cloud-based smart home with community hierarchy". In: *IEEE Transactions on Consumer Electronics* 63.2 (2017), pp. 200–207.