Industrial Paper

Proposal of Tamper-Proof IoT System Using Blockchain

Tetsuo Furuichi [*], Tomochika Ozaki [**] , and Hiroshi Mineno [***]

[*]e-Cloud Computing&Co. / Graduate School of Informatics, Shizuoka University, Japan
[**] Hitachi, Ltd.
[***] Faculty of Informatics, Shizuoka University, Japan
furuichi.tetsuo.15@shizuoka.ac.jp, tomochika.ozaki.wr@hitachi.com, mineno@inf.shizuoka.ac.jp

Abstract - In recent years, IoT which connect things to everything has become more widespread. Many practical systems are actually in operation, and are of great use in our lives. On the other hand, many information security incidents are reported, and the demands for countermeasures against them have been further increased. While the measures against malicious third parties have been mainstream until now, there has been an increasing demand for data falsification detection and blocking by operators and parties. Therefore, we focused on blockchains used in virtual currency as data tampering prevention technology. By applying the blockchain to the IoT system, we built an IoT system with a tamper-proof function for sensor data. Specifically, an IoT Gateway, which had been directly implemented in hardware, was realized with a smart contract, and devices which enable to use the blockchain efficiently were implemented in the configuration module. The blockchain with the tamper-proof function has a penalty of data propagation time, so we evaluated the data propagation performance in the implemented system and examined practical application examples of this system.

Keywords: blockchain, smart contract, IoT, tamper-proof

1 INTRODUCTION

IoT (Internet of Things) realized by connecting things to things and people are put to practical use, and create new value and bring great economic opportunities [1]. In this paper, we propose an IoT system with a powerful tamper-proof function by using a smart contract of a blockchain specialized for IoT. The evaluation results show that the developed prototype system is effective as an IoT system for logging system of IoT data. In this chapter, the outline of the proposed system and the situation of the logging IoT system are explained. In Chapter 2, we introduce conventional research that applies IoT, blockchains, and blockchain to IoT. In Chapter 3, we describe the issues, configurations, functions and usage examples to be solved in the IoT system using the smart contract of a blockchain. In Chapter 4, we explain the evaluation environment of the system implemented as a prototype and shows the results of the evaluation. Finally, in Chapter 5, we describe the conclusions of this survey, analysis, development, and evaluation.

1.1 Outline of the Proposed System

The IoT technology has already been used in factories and industrial products, and the merits of use for agriculture have also been reported.

When IoT devices collect more data for various purposes, those data will become more important. Then, since information security attacks targeting those data are beginning to occur, demands for information security solution are increasing. There are three major elements of information security, confidentiality, integrity, and availability. The general IT (Information Technology) system which is already put into practice has a function for protecting these software elements, but in IoT devices these countermeasures are delayed. Many damages are caused by an attack by a third party whose identity is unknown. One of the technology to prevent data tampering is blockchains that support virtual currency. The blockchain treats transaction information as a decentralized managed ledger and has a mechanism that cannot change the transaction using encryption technology in a fixed mining cycle. Therefore, anyone can view the transaction information but no one can tamper with it.

We expected that the demand to prevent IoT data tampering will increase in the future, and examined and developed an IoT system using blockchains. Since this system directly handles IoT data acquired from sensors as blockchain transaction information, it can take advantage of the blockchain tampering prevention feature as it is. And we implemented the IoT gateway using a smart contract, which is a function to process blockchain transaction information.

Latency time is one of the important performances in IoT systems. The IoT system with low latency time has a wide range of use. However, low latency systems may increase costs and power consumption. Since blockchains have mining cycles, there is a penalty in latency time. Therefore, in this research, we aimed to make it possible to use in various usages by shortening the latency time as much as possible.

1.2 Outline of the IoT System

As devices for personal use, a fitness tracker or wearable device acquires information on activity and exercise from our body and transmits the information as data. As devices for home use, there are smart home devices that control air conditioning and monitor electricity usage, and the like. There

ISSN1883-4566 © 2019 - Informatics Society and the authors. All rights reserved.

are also smart security devices that are useful for home security. In the retail environment, proper stock managements and self-checkout functions are realized with IoT. In addition, in offices, security and energy managements have been implemented to improve the efficiency of the operation of buildings and raised the productivity of employees. Advance sale analysis, usage-based design, and condition-based maintenance are effective for vehicles. By using IoT devices for industry, safety and productivity can be improved. Utilization in cities is used for resource managements, environmental monitoring, smart meters and adaptive traffic managements, and it is desired in cities to put autonomous vehicles into practical use by using smart IoT devices [2].

And, in the IoT system, the latency time is an important requirement for measuring its performance. However, in order to shorten the latency time, a lot of resources are required. Therefore, it is necessary to set the latency time according to the application.

The information obtained by the IoT devices has often been affecting the interests of operators and users. Typical examples are log information of public transportations, cars, equipment, and healthcare products, and the like.

(1) Traffic probe data

Traffic probe data obtained from taxis and buses is one specific example of log information. These data are used as a congestion degree and snow removal information each time by statistically analyzing them collectively after a lapse of time. When this information concerns someone's interests, the importance of the data increases and accuracy is required. When the data amount is large, statistically it is possible to eliminate erroneous information including noise and alteration. However, since the number of sampled probe data acquired in rustic areas is small, filtering is difficult. For that reason, prevention of tampering is an important requirement for that information.

In the case of traffic probe data, it can be used for real-time navigation and warning systems for drivers when information can be acquired or processed with low latency time. Even when the probe data cannot be processed with a low latency time, it is possible to analyze road conditions by season and time zone by batch processing.

(2) Log of vehicle and machine

Working log of a vehicle, driver's driving log, machine working log, and an operator's operation log can be used for various purposes. The authenticity of the data is also important for this information.

In the case of log information of cars, drivers and machines, if information can be acquired and processed with low latency, it will be possible to construct a warning system and accident avoidance system for drivers. In addition, if the latency time is in minutes, it can be used for machine failure prediction and doze prediction. Even with even larger latency times, if any accident happens in future automated driving, it can be used to analyze the cause of the accident and clarify the location of responsibility.

(3) Measuring equipment and inspecting apparatus

As for the information on the measuring equipment and the inspecting apparatus, not only the measurement information and the inspection information of the object but also the accompanying information such as the identification information of the measurer and the measuring time are important. For example, in order to determine the shipment of products, their measurement information may be used. In many cases, the product shipping yield depends on that measurement information. The determination of the shipment by a specific measurer or measured value may affect the final product shipment number and profit. Prevention of tampering with that information is also important. If a third party tamper with that information, it will cause confusion in product shipment. In addition, producers who stick to the number of deliveries may alter the measurement data and give priority to securing their profits.

In the case of information on measuring instruments and testing equipment, information on low latency time, especially in shipping determination, is important, because it affects throughput. In addition, when investigating the influence of yield and production number, low latency time is unnecessary, because it only refers to past measurement information as batch processing.

(4) Sensor information on healthcare

Sensor information on healthcare may change the amount of insurance subscribed to by that party. Tampering with the sensor owner may possibly change the insurance premium or the amount of insurance.

In the sensor information of healthcare, if sensor information can be handled with low latency time, it can be used for abnormality detection and notification of vital data. Also, in the case of latency time in minutes, it may be used for abnormal announcement of vital data. In the case where the latency time is long, it is possible to diagnose the health condition of the parties by batch processing.

In the current IoT devices, the operator or the user often has administrative authority. In the case where these operator and user have to take some responsibility and compensation, if they have an administrator authority for tampering data, the authenticity of acquired data may be suspected.

For these measures, human measures, technical measures and physical measures can be considered. Human measures include moral education of information security and operation management education. Technical measures include encryption with multiple keys, obfuscation, signatures, timestamps and log management at multiple sites. Physical measures include entry and exit managements and locking managements are typical. These countermeasures are time-consuming and require a lot of large-scale measures, so it is a difficulty to incur development and operation costs.

2 PREVIOUS RESEARCH AND TASK

In this chapter, we will introduce IoT technology, blockchain technology, and previous research applying blockchains to IoT.

2.1 IoT

Different types of IoT modules are made according to their purposes and conditions. The sensor type IoT module communicates the sensor data with the cloud server via the Internet using some communication method. In the actuator type IoT module, control from the cloud server reaches the actuator via the Internet or proprietary communication.

As an index of the performance of the IoT module, there are a latency time and a data transmission band. These indicators are selected depending on the purpose, with a balance between cost and performance.

Figure 1 (a) shows the simplest connection between the IoT module and the server. In this example, the controller to which the sensor is connected is able to access directly to the server via the network. The controller sends the data of the sensor to the transmission line, the server which received the data records it in the storage, and the data in the storage becomes the reference data of the Web Application.

When the communication on the sensor module side has the purpose of power consumption reduction and security consideration, a protocol conversion module called a Gateway may be interposed in the communication line. Figure 1 (b) shows the IoT system via the Gateway. The controller connected to the sensor passes the data to the Gateway once, and the Gateway sends the data to the server via the Internet.

The IoT module connected to the Internet may be hacked by a malicious third party via the Internet. The most vulnerable points are wireless and Internet communication lines. In Fig 1 (c), in order to encrypt a communication line, a system using SSL or HTTPS of MQTT(Message Queuing Telemetry Transport) is shown. In this figure, MQTT Publisher is placed on the sensor side, the gateway is set as MQTT Broker, and the server is set as MQTT Subscriber.

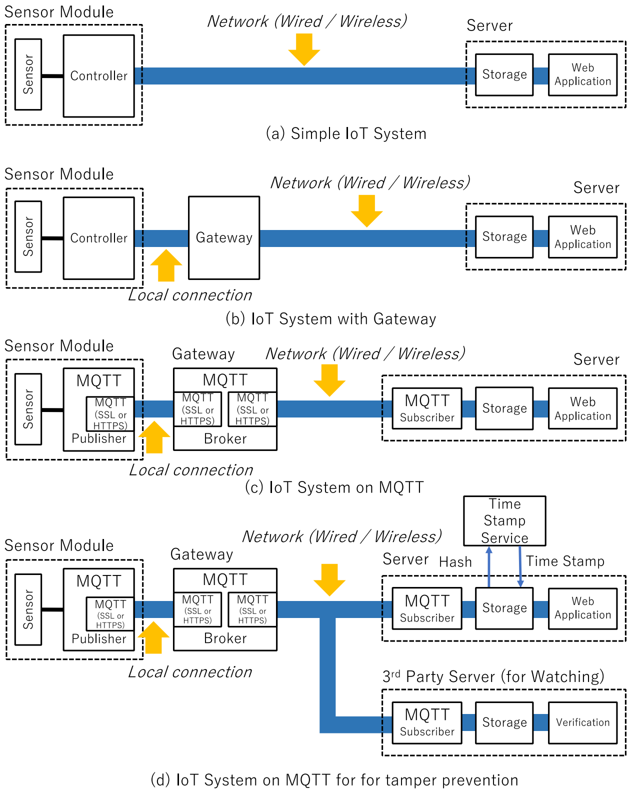
Now that many types of IoT modules have been put into practical use, there are many IoT modules that are operated as they are at shipment due to a misconfiguration of these devices. In 2016, MIRAI Botnet scans these default passwords for devices connected to the Internet, performs DDoS attacks using the compromised devices, and caused enormous damage. There are many variants of that MIRAI now.

Many IoT modules are vulnerable to various types of security threats. As a reason, there are many cases where the user is not near the IoT module, and since they are not supervised for a long time, in many cases it is not noticed that the user is under attack for a long time. In the case of wireless communication, eavesdropping is extremely easy.

In addition, components of the IoT module often have low performances in terms of their power consumption limitations and computing power. There seem to be some IoT modules on the market, commercialized without implementing sufficient security functions due to demands for low cost and a short deadline.

A powerful encryption algorithm is necessary as a technical measure against the information security threat of the IoT module. To that end, it is necessary to have a powerful processor capability that can calculate cryptographic algorithms in real-time [3].

As already mentioned, we need measures to prevent tampering by malicious third parties and parties in its operation.



(a) Simple IoT System

(b) IoT System with Gateway

(c) IoT System on MQTT

(d) IoT System on MQTT for for tamper prevention

Figure 1: Various IoT System configuration diagrams

As a method for preventing tampering, there is a method of taking an electronic signature and a method of taking an electronic signature by a timestamp server. In either case, since the administrator has all privilege, it is possible to perform operations related to tampering at any time. Therefore, there is a need for multiplexing data with a third party so that tampering history can be detected later. Figure 1 (d) shows the configuration diagram of the MQTT, the timestamp server, and the system multiplexing data with a third party. In the case of this configuration, it is assumed that the system after Subscribe of MQTT is multiplexed, and the third party's server is managed under a different administrator. When there is tampering on the original administrator side, it causes a difference from the multiplexed data, and the tampering can be detected. However, the configuration described the above makes the system complicated, and it is difficult to put into practical use in terms of labor and cost.

2.2 Blockchain

A blockchain is a system capable of tracing all transaction histories by a distributed consensus-building mechanism with network participants. So far, many kinds of virtual currency using a blockchain are distributed. Here we introduce Bitcoin which is the most famous blockchain and Ethereum which is a blockchain system that can execute application programs.

Bitcoin was developed based on the blockchain technology posted by a person named Satoshi Nakamoto in 2008. It started operation in 2009 and is a famous blockchain for virtual currency [4]. Bitcoin is a system composed of a blockchain node called Bitcoin client and a Bitcoin network. The transaction information issued by the Bitcoin client is sent as a transaction to the Bitcoin network, and minor, which is a kind of Bitcoin client, miners, so that the block is generated. And that block is approved from multiple nodes of the Bitcoin network [5].

Ethereum, proposed by Viralik Buterin's white paper in November 2013, is a blockchain and makes it possible to build applications by smart contract [6]. While Bitcoin is specialized in moving ownership of cryptographic currency, Ethereum is characterized by being able to create and execute distributed applications called smart contracts as well as moving cryptographic currencies [5][7].

A blockchain can be said a distributed database that realizes a "distributed ledger" that distributes and manages transactions as exchange information on a distributed network. We manage and operate a list of sequential data called "blocks" that summarizes those transactions on multiple nodes. Moreover, the validity of the block is secured by the mining processing using the distributed consensus algorithm. PoW (Proof of Work) is mainstream in the current distributed consensus algorithm. Under the agreement of the configuration node of the network, difficulty values are set and have a mechanism to adjust the mining time. In addition to PoW, PoS (Proof of Stake) and PoI (Proof of Importance) have been proposed as distributed consensus algorithms. Those methods have the effect of not consuming processor resources and power. In recent years, blockchains with different distributed consensus algorithms have also been released.

EOS which is one of the new blockchains uses the distributed consensus algorithm of DPoS (Delegated Proof of Stake) [8]. Generally, in PoS, block generators are determined by the amount of currency held, but in DPoS, block generators are determined by voting by other nodes in the blockchain network. Also, since the weight of the vote is determined by the amount of currency held on the blockchain, it cannot be determined by the block generator itself. However, if multiple voters agree on each other, it is possible to fix the block generator and also to perform the centralized operation. Therefore, new blockchains have been proposed that overcome these concerns [9]. These arguments are particularly active in the public blockchain, which aims at virtual currency functions. There is no need to limit the use and if the node is a member of a whitelist, it is not necessary to adhere to the expensive public blockchain, and it is more convenient to use it in a private blockchain or a consortium blockchain [10].

(1) Blockchain operation

A blockchain is composed of various elements such as nodes, P2P(Peer to Peer) networks, transactions, blocks, distribution ledgers, and mining. A blockchain is a virtual network configured on a physical ordinary network. The connection unit is called a node, and the nodes are physically connected by P2P. Each node has its own unique asymmetric key. The transaction information issued by the node is signed with the secret key of the node itself, and after another node's approval, it is spread to the blockchain network via P2P. The spread transactions are grouped together in a preset time and are blocked by mining. The block information is handled as information of the distributed ledger after undergoing multiple approval processes.

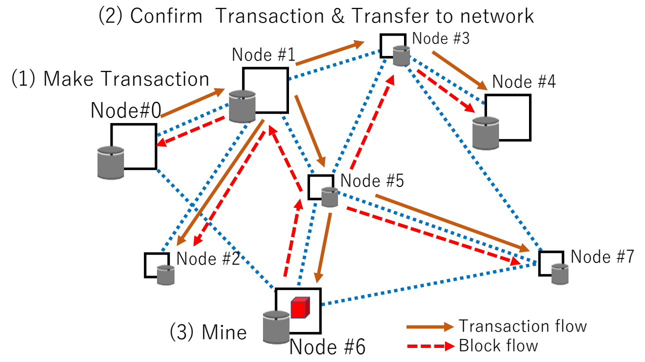


Figure 2: Process flow among blockchain nodes

Figure 2 shows an example of a flow of processing among nodes of the blockchain. First, if Node #0 issued a transaction of transaction, the information is passed to Node #1, approved, and then diffused with Node #2, #3, #5 -> #4, #6, #7. After that, Node #6 having a mining function performs Mining by using a distributed consensus algorithm, and then performs blocking. The information of the block is spread information of block information via Node # 5, approved in the entire blockchain, and is handled as a valid distributed ledger in each node.

(2) Smart contract

Whereas Bitcoin has a mechanism specialized for virtual currency trading, there is a blockchain that can handle smart contracts, which is a type of program shared on the blockchain, as well as virtual currency transactions. Ethereum is one of them, and each node can access virtual currency transaction, mainly to execute the virtual program. The creation and execution of the smart contract are treated as transactions, so their generation and execution records are stored in the blockchain and cannot be tampered with. Therefore, the reliability of the execution result of the smart contract becomes very high.

2.3 IoT + Blockchain

Focusing on the convenience of distributed management of a blockchain and the characteristics of the virtual currency, the degree of expectation for adaptation to IoT is increasing. In the field of electric power systems, there are cases where blockchains efficiently perform IoT updates on an ongoing basis. Blockchains have "a public blockchain" used in virtual currency and "a private blockchain" mainly used experimentally. This report recommends the use of a private blockchain to ensure security. Also, the number of Mining Nodes is recommended to minimize implementation considering security. Furthermore, since we disclose information by using blockchains, they recommend securing the confidentiality of data in a different way from the blockchain [10].

To cope with IoT, a mechanism is developed to cover a blockchain client program with a wrapper, and by using a network different from the blockchain, a weak data transfer of the blockchain is handled (Fig. 3) [11]. According to research to use IoT in Smart Home, the merit of information security is larger than the overhead of the processing blockchain [12].

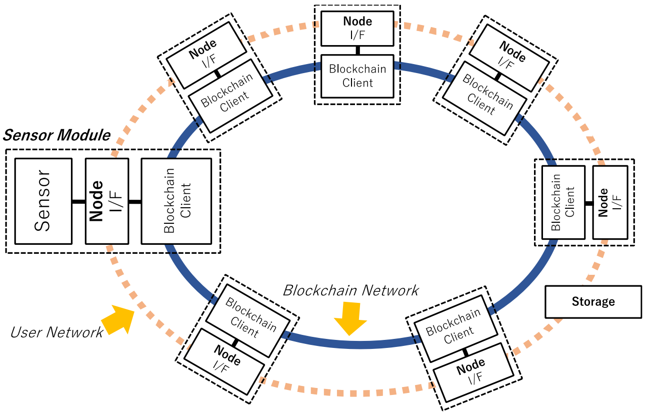


Figure 3: IoT system on blockchain with user network

In order to easily manage the configuration of the IoT module as research concretely using a smart contract, an IoT system that has the mechanism of RSA key management in Ethereum's smart contract has been reported [13].

3 IOT SYSTEM ON BLOCKCHAIN

We explained that there are new requests for information tampering prevention in the IoT system and that blockchain can be used as a means for preventing tampering. IoT systems with security requirements will have the ability to handle cryptographic algorithms at a reasonable speed. We focused on using that powerful resource for blockchain operation. However, we recognize that the penalty for latency time due to the use of the blockchain and the high cost of the huge volume of data for the blockchain. In this chapter, we will introduce the configuration of a new IoT system using a blockchain-based on the hypothesis that the bandwidth of the network including wireless will further expand, and the demand for information security will further increase.

3.1 Blockchain for Tamper Prevention

We have already explained that it is necessary for the supervisor to construct and operate a data-sharing server when realizing tamper-evident measures with the practical system configuration (Fig. 1 (d)). Implementing a new server and implementing data multiplexing in building a system takes time and labor for development and increases development cost. Furthermore, maintenance troubles and expenses including information security measures have increased, and it is clear that these configurations are not practical systems.

As techniques for tampering prevention, digital signatures and time stamps are known. These technologies can make it possible to prevent tampering by a malicious third party, but it is inevitable to prevent tampering by administrative users or users. As a method of avoiding these concerns, as already mentioned, there is a multiplex recording of data after the digital signature. That is, it requires maintenances of multiple recordings by a supervisor other than the administrator.

In the blockchain, transactions to be sent to that chain are digitally signed by the issued node and broadcasted. The node received the broadcast verifies the transaction. If the transaction is regarded as illegal, it is discarded. The approved transaction group is blocked by mining so that the data is confirmed and the record is held at each node. The advantage of the blockchain is that these series of actions have already been implemented as functions and already proven. However, even in the case of using a blockchain, it is clear that if parties such as operators and users have all nodes, it is impossible to prevent tampering from the parties. Therefore, as with the configuration in the practical system, it is necessary for the supervisor side other than the administrator to have Full Node.

As data multiplexing in practical systems has a heavy burden on system development and operation and maintenance, the use of a blockchain has a merit that a load of new system development is light. Furthermore, if major OSS (Open Source Software) blockchains are used, updates of information security, high reliabilities of the systems and lower operating costs are expected.

3.2 Block Diagram of the IoT System

Figure 4 shows a block diagram of the IoT system using a blockchain. All hardware are logically connected in the blockchain as client nodes of the blockchain. Also, the IoT Gateway composed of the smart contract has already been registered on the blockchain and can communicate with all client nodes. The configuration hardware are nodes composed of a Sensor Module, a Storage Module, a Network Gateway Module, and blockchain clients including those modules. In order to function as a blockchain, one of the client nodes has a function as a minor. in order to realize the tamper-proof function, each module communicates data as a transaction in which recording remains as a blockchain. Furthermore, in order to detect falsification of administrators and users, supervisors other than administrators and users should have one or more equivalent blockchain clients.

Generally, since the exchange information of the blockchain is handled as broadcast, the size of data that can be sent to the network is limited. This time, according to the specifications of Ethereum to be implemented, it is decided to pass data at about 1K-Bytes / sec. If more data is sent or received, it is possible to negotiate a data exchange method separately and deal with bypassing the hash value of the chunk of data to the blockchain. As an implementation method of the IoT function, an IoT Gateway is provided so that transmission and reception of IoT data can be controlled. This IoT Gateway has functions of approval of IoT module, buffering of IoT data, and distribution of IoT data. Also, by installing the IoT Gateway not by hardware implementation but by the smart contract, it is possible to avoid problems due to specific hardware or network malfunction, and realize the availability of the IoT system. Also, since this smart contract can be changed independently of hardware, scalability as an IoT system can also be secured. This implementation method is different from the conventional implementation method, and becomes a characteristic point.

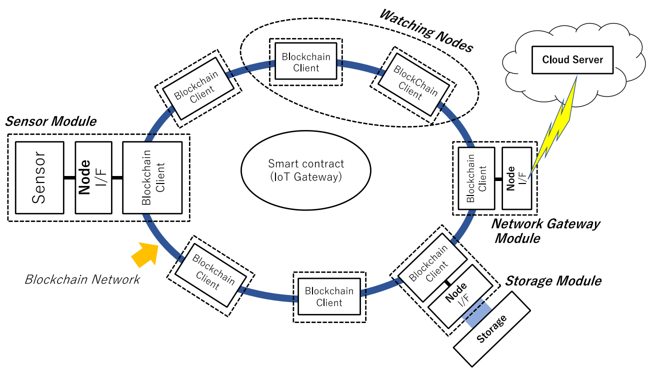


Figure 4: IoT system on a blockchain

3.3 Functions of the IoT System

This IoT system has four functions. Figure 5 shows the connection between these functions.

(1) IoT Gateway (smart contract)

The IoT Gateway has the central control function of the IoT system. In this implementation, we implemented the IoT Gateway into a smart contract of Ethereum. Individual functions include sensor module registration, activation and data buffering.

(2) Sensor Module Node

The sensor module has a function of acquiring sensor data from the sensor and sending it to the smart contract which is the IoT Gateway on the blockchain. Initially, the sensor module performs its own activation at the IoT Gateway. After approval, this sensor module will be able to send sensor data to the IoT Gateway.

(3) Network Gateway Module Node

The network Gateway module monitors events of the IoT Gateway. When an event occurs, this module receives data from the IoT Gateway and sends this data to the server on the Cloud Computer System. We will also implement a function of sending a data transmission request to the IoT Gateway according to an instruction from the cloud server.

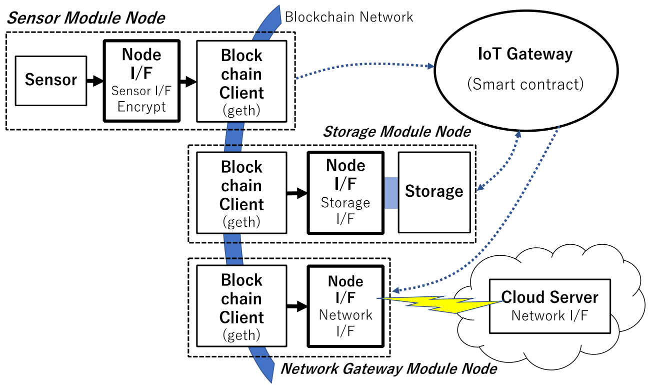

Sensor Module Node
Blockchain Network
Sensor
Node I/F Sensor I/F Encrypt
Block chain Client (geth)
IoT Gateway (Smart contract)
Storage Module Node
Block chain Client (geth)
Node I/F Storage I/F
Storage
Block chain Client (geth)
Node I/F Network I/F
Cloud Server Network I/F
Network Gateway Module Node

Figure 5: Various functions of IoT nodes

(4) Storage Module Node

The storage module receives the data from the IoT Gateway and stores the data in the specified channel. In addition, we will implement a function of reading data according to instructions from the IoT Gateway.

Since the function of IoT Gateway operates with the smart contract of the blockchain, when the smart contract function is called, it is executed at the timing of mining. Also, the node that has started needs to pay the execution cost of the virtual currency to the smart contract, and its function is effective for blocking a malicious third party who attacks unscrupulously.

3.4 Implementation Examples

In this section, we will examine how this IoT + blockchain system in some of the examples of systems that need tampering prevention introduced in Section 1.3.

(1) Traffic probe data

In this case, the place where the data is involved is the car that collects the information and the base to compile the data. Therefore, it is possible to create a system that prevents tampering by placing the clients of the blockchain at the automobile, the regional aggregate, the final summary and the audit site of the data. Specifically, we make the car a "Sensor Module Node" and the local area aggregate station a "Storage Module Node". Generally, the last summary office connects to the Cloud Server as a "Network Gateway Module Node". Each node exchanges data by communicating with "IoT Gateway (smart contract)" specialized for traffic probe data.

(2) Log information of car

There are two possibilities for car log information, such as the inside of a car and the system for collecting data from a car. Since there is a possibility of a mismatch between functional safety requirements and the blockchain specifications inside the car, we will examine a system to compile data from the car. It can be managed and operated with the same system as the above-described traffic alteration prevention system for traffic probe data.

(3) Information on measuring equipment

Regarding the log information of the measuring instrument, the measuring instrument is a "Sensor Module Node", and the management system of the base connecting the plurality of measuring instruments implements the "Storage Module Node". Also, the gateway system connecting the sites implements the "Network Gateway Module Node". Each node exchanges data by communicating with "IoT Gateway (smart contract)" specialized for measuring equipment.

(4) Healthcare sensor information

Sensor information on healthcare is more special in terms of cost and composition than the previous examples. In many cases, since the battery of the sensor is not large for miniaturization, chances of being connected to the network at all times may be small. A device that acquires data from the sensor is a "Sensor Module Node", and both a "Storage Module Node" and a "Network Gateway Module Node" are installed in a smartphone or a PC. As in previous systems, each node exchanges data via the "IoT Gateway (smart contract)" specialized for healthcare sensor processing.

3.5 Latency Time Model

One of the guidelines for measuring the performance of the IoT system is latency time. If the latency time is short, it becomes possible to use the IoT system for abnormality detection and machine control. In order to shorten the latency time, it is necessary to prepare a circuit having abundant processing capability and a high-speed communication line. These conditions increase development difficulty of the IoT system, and increase development and implementation cost. Therefore, it is desirable to set the latency time condition according to the purposes and conditions of the IoT system.

The target IoT system using the blockchain treats propagation of IoT data as a transaction of a blockchain. Therefore, it is necessary to consider the latency time along the data flow of the blockchain.

Figure 6 shows the flow of the delay model of the latency time of the IoT system on the blockchain. Sensor data is obtained at the sensor module and is sent as a transaction to the blockchain network through the blockchain client (Sensing Delay). A transaction flowed to the blockchain propagate to each node as broadcast, and reach the node where mining is executed. This delay is thought to depend on the total number of nodes and the logical connection configuration of Nodes (Broadcast Delay 1). In the mining execution node, mining is executed after the interval of mining timing, and the smart contract is executed (Mining Interval, Mining Delay, Execute smart contract). Events issued by the smart contract are broadcasted in the blockchain as a transaction, reaching the receiving node (Broadcast Delay 2). Receive processing is performed at the receiving node (Receiving Delay).

In this manner, the latency time of the data propagation of the transaction in the blockchain is constituted by many routes, the route distribution algorithm due to the blockchain, and the discontinuous traffic adjustment function for blocking, it is difficult to predict with high accuracy. Therefore, an evaluation should be performed in advance in an environment similar to the system to be realized.

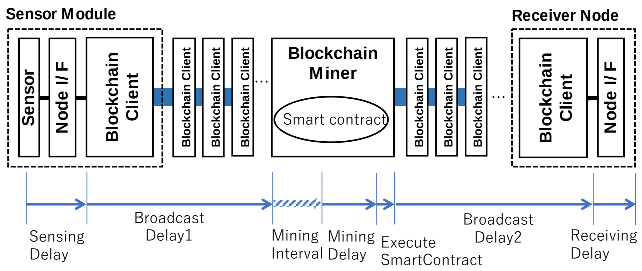


Figure 6: Latency time mode for IoT system

4 PROTOTYPE IMPLEMENTATION AND EVALUATION

In the IoT system using the blockchain designed in Chapter 3, we measured the latency time, which is one of important performance, under two conditions and evaluated the performance as an IoT system.

In the IoT system assumed this time, the function of the virtual currency of the blockchain is not important, and the main purpose is to use the information security function of the blockchain for the falsification deterrence function of IoT data. Therefore, it is not necessary to be a public blockchain with some security risk, and the prototype system and evaluation environment were implemented on a private blockchain. Also, the blockchain client program used this time is Ethereum geth v.1.8.6.

4.1 Evaluation Environment

Figure 7 shows the configuration of the environment used for this evaluation. Ethereum Private Net was constructed by implementing Ethereum 's client program (geth) on the server' s container, Note PC and IoT module. Furthermore, IoT Gateway(smart contract), which is an IoT API, is implemented in the blockchain net and it was created in advance as a transaction in the blockchain. Three types of Nodes were prepared for the evaluation. The first one is a general Node which is a contract Owner. The second one is a “Sensor Module Node” that uploads sensor data to the blockchain. The third one is an “Internet Gateway Module Node” that takes sensor data from the blockchain. Each node assigns an account address of the blockchain.

Figure 8 shows the hardware configuration of the evaluation environment. Since the evaluation at this time required a lot of nodes, we implemented the client program (geth) on the containers of server computer.

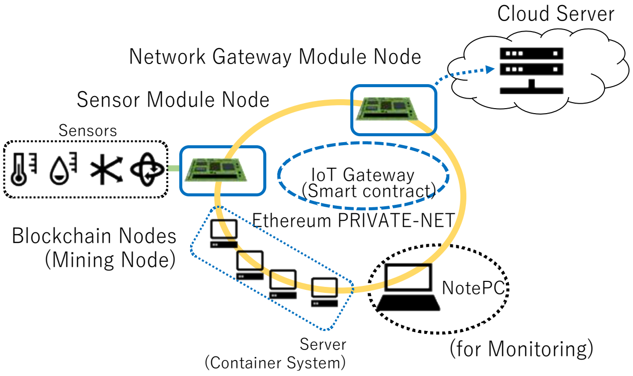


Figure 7: Overview of Logical nodes connection

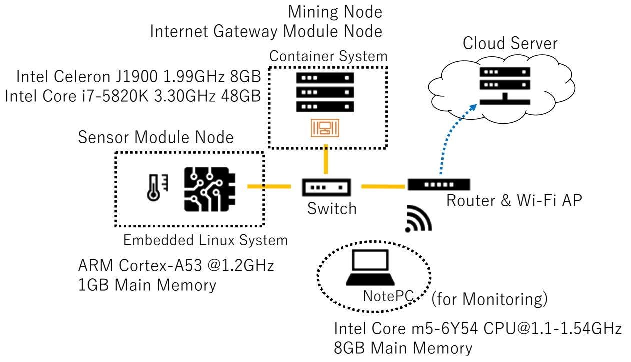


Figure 8: Physical Connection of Evaluation Environment

4.2 Evaluation Method

For evaluation, we prepared an IoT Gateway implemented on the smart contract we developed this time, Ethereum client program (geth) and JavaScript evaluation script. Each Node participates in the blockchain network by running the client program (geth). Referring to the logical function of Fig. 5, the evaluation script sends test data from the "Sensor Module Node" to the "IoT Gateway (smart contract)". The "IoT Gateway" then returns an event and the "Network Gateway Module Node" receives the event and reads the sent data. In order to accurately measure the latency time of data, this evaluation is to make the "Sensor Module Node" have the function of the "Network Gateway Module Node" and to measure the latency time of data within one node. These processes are executed for the specified number of times. Mining Node for blocking transactions in the blockchain is a Node running on a container connected to the network. Also, the execution instruction of the Mining process was manually performed.

4.3 Evaluation and Results

In this research, two kinds of data transfer latency times were evaluated.

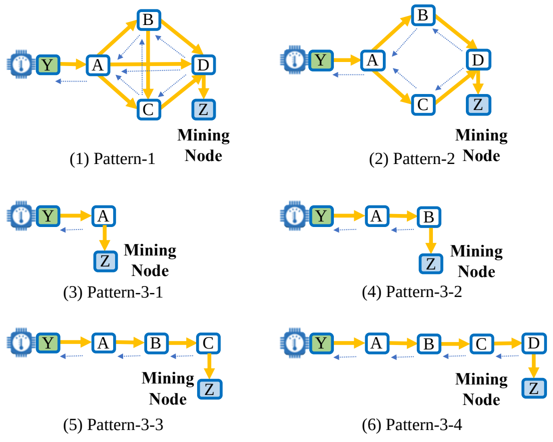


Figure 9: Various connection patterns with 1-miner

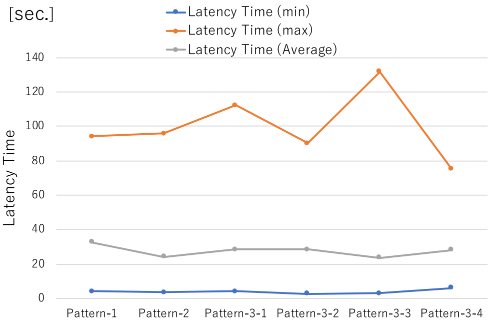


Figure 10: Latency time of each pattern with 1-miner

(1) Data transfer latency time by physical connection

This evaluation measures the data transfer latency time in "Node - smart contract - Node" by changing the physical connection form of Node in the blockchain. We made 100 data accesses in each physical connection form. In this evaluation, we used Intel Celeron J1900 1.99 GHz 8GB Main memory as a server and used a container environment. The number of blockchain nodes was evaluated at 3 to 6. Six types of physical connection were prepared. Figure 9 shows these connections. 'Y' in the green box is a Node with sensor data. 'Z' in the blue box is Mining Node. Pattern-1 indicates that the constituent Nodes A to D are mutually connected. In Pattern-2, configuration nodes A to D are connected in an annular shape. Pattern-3-1 to Pattern-3-4 are connection embodiments in which the number of Mining Nodes is changed from Node having sensor data. The yellow arrows indicate the expected direction of propagation of the issued transaction. The blue dotted arrows indicate the flow of the transaction after Mining.

Figure 10 shows the latency time results for each connection pattern. The minimum value of the latency time was 3.86 seconds on average and the average value of the pattern was 27.59 seconds, and there was no big difference in any physical connection pattern. However, the average of the maximum values in each connection pattern is 75.09 to 131.74 seconds, which widens the value range, and the dispersion increases as the physical distance increases.

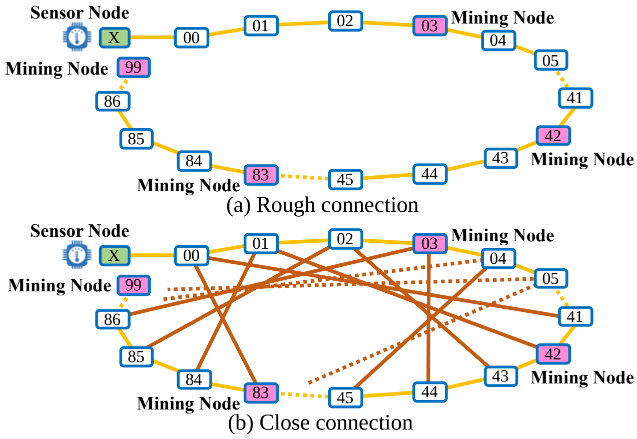
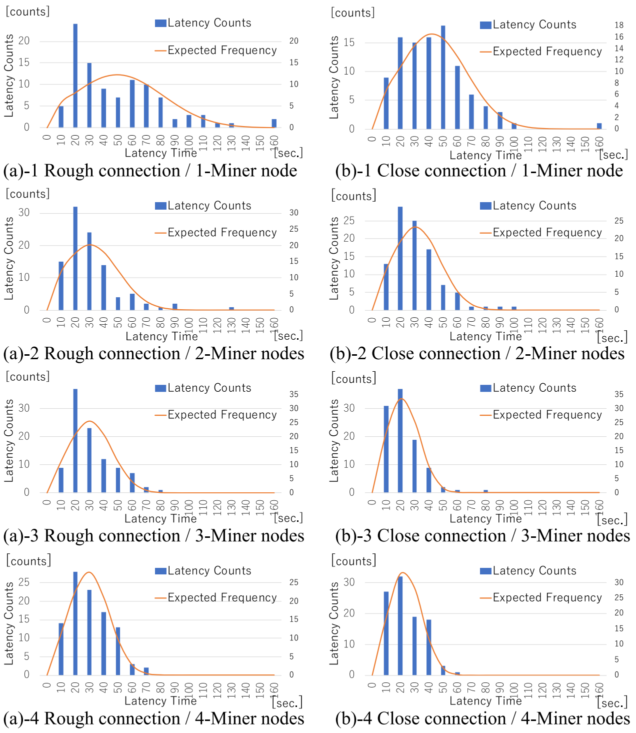


Figure 11: 100-Nodes blockchain network



(a)-1 Rough connection / 1-Miner node

(b)-1 Close connection / 1-Miner node

(a)-2 Rough connection / 2-Miner nodes

(b)-2 Close connection / 2-Miner nodes

(a)-3 Rough connection / 3-Miner nodes

(b)-3 Close connection / 3-Miner nodes

(a)-4 Rough connection / 4-Miner nodes

(b)-4 Close connection / 4-Miner nodes

Figure 12: Latency time under various conditions

(2) Data transfer latency time by mining number

This evaluation used a container environment with Intel Core i7-5820K 3.30 GHz 48GB Main memory as a server. Since the actual usage environment was assumed, a blockchain model was prepared with the number of nodes increased to 100. Under the environment of this model, 16 sensors were assumed, and the sensor data size was 32-byte, and a high load evaluation system was prepared that could apply 128 times the sensor data load compared to the evaluation environment of the previous section. The data is sent from 16 sensors without sending delay time. The mining cycle has a cycle of about every 12 seconds because it uses the original functions of Ethereum.

In addition, two types of inter-node connection models were prepared. The first is a model (Fig. 11 (a)) in which all nodes are connected in a straight line in order to express non-uniformity of connection between nodes. The second assumes a normal blockchain connection, and prepares a model (Fig. 11 (b)) in which each node interconnects six nodes. The green box "X" in Fig. 11 (a) and 11 (b) is a sensor node and sends sensor data to the blockchain. And 16 sensors send data to one sensor node for high load. In addition, the pink box is set as one to four mining nodes assumed this time.

Figure 12 (a) shows the distribution of latency times by the number of mining nodes in a model in which 100 nodes are arranged in a straight line as a rough connection. When there is one mining node, the average latency time is 45.4 seconds, the standard deviation is 32.3, and the variation is large. Even in this model, if there are two or more mining nodes, the variations become smaller.

Figure 12 (b) shows the distribution of latency times by the number of mining nodes in a model in which 100 nodes are closely interconnected. As a result, when the number of mining nodes is one, the average latency time is 38.2 seconds, the standard deviation is 23.9, the latency time is relatively large, and the variation is also large. When the number of mining nodes was three, the average latency time was shortened to 17.1 seconds, and the standard deviation was 11.5, and stable latency time could be kept. However, with 3 and 4 mining nodes, there was no significant difference in the variation of latency time.

5 CONCLUSION

In this paper, we developed a system that uses practical blockchains to suppress falsification of IoT data, and evaluated it specifically for latency, which is one of the required performance of IoT systems. As a result, we were able to construct a system that does not require short latency time such as real-time warning and notice, which is suitable for logging application of IoT data. As a characteristic implementation method of this time, in order to use IoT data directly in the blockchain, the IoT Gateway function that controls authentication and delivery of IoT data is realized by the smart contract of the practical blockchain.

In this evaluation, we prepared an environment that applies a high load to a relatively large blockchain network. The processing performance depends on the number of connections between nodes (the number of peers) and the number of mining nodes, but it was found that IoT data could be acquired with an average latency of 17 to 46 seconds. From these results, IoT systems using this blockchain are suitable for use in IoT data logging systems without falsification of data, rather than a real-time warning or prediction system that requires short latency. These applications include traffic probe data, car log information, measuring device information, and healthcare sensor information acquisition, introduced in Section 3.4.

Moreover, in the system using the practical blockchain, the basic function and the performance have a margin, and the average latency time as IoT framework has almost no burden of data propagation due to the number of configuration nodes and the like. It was found that the impact conditions are the number of issued transactions, which is equivalent to the frequency of occurrence of sensor data, and the processing performance of mining (TPS: Transaction Per Second). Furthermore, it was found that when the density of connection between nodes is coarse or there are few mining nodes, the dispersion of latency time becomes large.

Especially in the case of the rough connection between nodes, if node connection is disconnected for some reason, it may take time until data synchronization again. Therefore, the number of peers of each node connection in the blockchain IoT system should be 3 or more.

In this experiment, we implemented the IoT system using the smart contract of Ethereum. Nowadays, many other blockchains also have smart contracts with more useful functions and capabilities. As future developments, we will study on a framework that can handle many blockchains and make it a more sophisticated IoT.

REFERENCES

[1] D. Evans, “The Internet of Everything How More Relevant and Valuable Connections Will Change the World, ” Cisco IBSG (2012).

[2] R. Patterson, “How safe is your data with the IoT and smart devices?,” Information Security, https://www.comparitech.com/blog/information-security/iot-data-safety-privacy-hackers/ (2017).

[3] A. F. Mohammed, “Security Issues in IoT,” IJSRSET Volume 3 Issue 8, http://ijsrset.com/paper/3369.pdf (2017).

[4] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” https://bitcoin.org/bitcoin.pdf, (2008).

[5] “Ethereum home page, “ https://www.ethereum.org/ (2019).

[6] V. Buterin, “A Next Generation Smart Contract & Decentralized Application Platform,” http://blockchain-lab.com/pdf/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf, whitepaper (2014).

[7] “Ethereum Homestead Documentation,” http://www.ethdocs.org/en/latest/ (2019).

[8] L. M. Bach, B. Mihaljevic, M. Zagar, “Comparative analysis of blockchain consensus algorithms,” Electronics and Microelectronics (MIPRO), Opatija, Croatia, pp. 1545–1550 (2018).

[9] “IOST WHITEPAPER,” https://iost.io/iost-whitepaper/ (2018).

[10] K. Christidis and M. Devetsikiotis, “Blockchains and smart contracts for the internet of things,” IEEE Access, 4:2292--2303 (2016).

[11] M. A. Walker, A. Dubey, A. Laszka, and D. C. Schmidt, “PlaTIBART: a platform for transactive IoT blockchain applications with repeatable testing,” in 4th Workshop on Middleware and Applications for the IoT (M4IoT) (2017).

[12] A. Dorri, S. S. Kanhere, R. Jurdak, P. Gauravaram, “Blockchain for IoT security and privacy: The case study of a smart home,” In Pervasive Computing and Communications Workshops (PerCom Workshops), IEEE International Conference, 618–623 (2017).

[13] S. Huh, S. Cho, S. Kim, “Managing IoT devices using blockchain platform,” 19th InternationalConference on Advanced Communication Technology (ICACT), pp. 464-467 (2017).

(Received January 7, 2019)
(Revised May 6, 2019)

Tetsuo Furuichi received his B.E. degree in Electronic Engineering from Himeji Institute of Technology in 1985. He currently works for e-Cloud Computing&Co. and he is currently a Doctor-course student at Shizuoka University. He is currently interested in the embedded system, IoT, information security, and blockchain, and so on. He is currently the Registered Information Security Specialist [RISS]. He is a member of IEEE.

Tomochika Ozaki received the B.E. degree from the Nagoya University in 1988, the M.E. degree from the Nagoya University in 1990 and received the Ph.D. degree in Informatics from Shizuoka University, Japan, in 2018. In 1990, he joined Hitachi Ltd. His research interests include embedded systems, energy management systems and human machine interface. He is a member of Information Processing Society of Japan.

Hiroshi Mineno received his B.E. and M.E. degrees from Shizuoka University, Japan in 1997 and 1999, respectively. In 2006, he received his Ph.D. degree in information science and electrical engineering from Kyushu University, Japan. Between 1999 and 2002, he was a researcher in the NTT Service Integration Laboratories. In 2002, he joined the Department of Computer Science of Shizuoka University as an Assistant Professor. He is currently a Professor. His research interests include Intelligent IoT system as well as heterogeneous network convergence. He is also a member of ACM, IEICE, IPSJ, and the Informatics Society.