Regular paper

Journal of the KIECS. pp. 157-164, vol. 18, no. 1, Feb. 28. 2023, t. 117, pISSN 1975-8170 | eISSN 2288-2189
http://dx.doi.org/10.13067/JKIECS.2023.18.1.157

지연시간 및 보안을 위한 블록체인 기반 스마트홈 시스템 설계

아창위[*] · 김강철[**]

Blockchain-Based Smart Home System for Access Latency and Security

Chang-Yu Ao[*] · Kang-Chul Kim[**]

요 약

현대 사회에서 스마트홈은 사람들의 일상생활의 한 부분이 되고 있다. 전통적인 스마트홈 시스템은 보안, 데이터 집중화, 위변조 같은 문제들을 내포하고 있으며, 이러한 문제들을 해결하는 기술로서 블록체인이 각광받고 있다. 본 논문은 홈과 블록체인 네트워크 부분으로 구성된 블록체인 기반 스마트홈 시스템을 제안한다. 8개의 노드로 구성된 블록체인 네트워크는 도커 환경에서 하이퍼레저 패브릭 플랫폼에서 구현된다. 데이터 전송 보안을 위하여 ECC 암호화 기술이 사용되고, RBAC가 네트워크 회원의 인증을 관리한다. Raft 의사 결정 알고리즘은 분산처리 시스템의 모든 노드에서 데이터 일관성을 유지하고, 블록 발생 시간을 줄인다. 노드들이 스마트홈 데이터를 안전하고 효율적으로 접근하도록 스마트 컨트랙트가 쿼리와 데이터 전송을 제어한다. 실험 결과는 많은 동시 접근 하에서 안전한 평균 쿼리와 서브밋 시간이 84.5 [ms]와 93.67 [ms]로 유지되고, 모의 패킷캡쳐 공격에서 전송 데이터가 안전하다는 것을 보여준다.

ABSTRACT

In modern society, smart home has become a part of people's daily life. But traditional smart home systems often have problems such as security, data centralization and easy tampering, so a blockchain is an emerging technology that solves the problems. This paper proposes a blockchain-based smart home system which consists in a home and a blockchain network part. The blockchain network with 8 nodes is implemented by HyperLeger Fabric platform on Docker. ECC(Elliptic Curve Cryptography) technology is used for data transmission security and RBAC(role-based access control) manages the certificates of network members. Raft consensus algorithm maintains data consistency across all nodes in a distributed system and reduces block generation time. The query and data submission are controlled by the smart contract which allows nodes to safely and efficiently access smart home data. The experimental results show that the proposed system maintains a stable average query and submit time of 84.5 [ms] and 93.67 [ms] under high concurrent accesses, respectively and the transmission data is secured through simulated packet capture attacks.

키워드

Smart Home, Blockchain, Decentralization, Smart Contract, Access Latency
스마트 홈, 블록체인, 분산화, 스마트 계약, 접근 지연

[*] 전남대학교 대학원 컴퓨터공학과(aochangyu@163.com)
[**] 교신저자 : 전남대학교 공학대학 전기컴퓨터공학부

· 접 수 일 : 2022. 11. 23
· 수정완료일 : 2023. 01. 02
· 게재확정일 : 2023. 02. 17

· Received : Nov. 23, 2022, Revised : Jan. 02, 2023, Accepted : Feb. 17, 2023

· Corresponding Author : Kang-Chul Kim
Chonnam National University
Email : kkc@chonnam.ac.kr

Ⅰ. Introduction

In recent years, the Internet of Things (IoT)[1-2] has attracted more and more attention as an open and comprehensive intelligent network that automatically processes data and shares information. As the hardware technology increases, there is more room for development in the IoT sector[3]. According to the GSMA's Mobile Economy 2020 report, the Internet of Things reaches 24.6 billion connections by 2025 with a compound annual growth rate of 13%[1]. The development of IoT significantly improves the intelligence and informatization of the whole society[4].

Smart home is one of the main applications of IoT. With the continuous development of Information and Communication Technology(ICT) and IoT, the function and role of the smart home are also developing[5]. According to Stratecast, the global smart home market is estimated to grow by more than $7 billion by 2025[6]. However, traditional smart home systems suffer from a lack of centralized trustworthiness, excessive device authorization, and complex control frameworks[7]. User privacy leakage and security problems also bring new risks and challenges[8] to the IoT.

Blockchain[9] aims to be a safely distributed database architecture for all digital asset transactions performed in the storage platform. It ensures the data integrity of IoT devices[10] and provides a secure and scalable framework for user privacy leakage. Blockchain not only ensures information security but also solves the shortcomings of traditional smart home centralization. At present, there are very many smart home system solutions.

Madani[11] proposed a child-centric smart home solution that shares information collected about the activities of children with external parties. In this system, the data is processed locally and an attacker may access the sensor nodes illegally, threatening the confidentiality of the information.

Namdeo[12] proposed a smart home system for power management that relies on the collaboration between individual components to conserve energy. All nodes in this system communicate with a single Web Server. If this Web Server crashes, the entire network cannot work.

Lin[13] proposed traceability and privacy protection of access policies, using group signatures and message authentication codes for identity authentication. The system was built on a public chain which resulted in high communication latency. Although the communication latency was optimized in this paper, it was still much longer than the traditional smart home system.

Ma[14] built the prototype system with a Raspberry Pi(RPi), a DHT11 temperature and humidity sensor, and the Blynk App. The authors used Ethernet to create a decentralized application that simulates a smart home application. In this system, Ethernet has a blocking time of about 15 seconds, and there will be a large delay in data submission. Also, the Blynk App was stored in a smart contract, and each call incurred a transaction fee, which increased the cost of running the system.

Among many blockchain platforms, Hyperledger Fabric(Fabric)[15] has modular architecture management compared to other blockchain platforms. Docker containerization technology[16] as a runtime environment for Fabric is another manifestation of virtualization technology, which is an open-source lightweight virtualization container engine application written in Golang.

This paper proposes a blockchain-based smart home system with the distributed data storage on Hyperledger Fabric and docker for stable and high throughput. The smart contracts are written to make a user access the blockchain network more

[1] https://www.idc.com/getdoc.jsp?containerId=US47678515 (accessed on 30 July 2022).

secure and faster, and the asymmetric encryption algorithm is used to secure the system data transmission.

This paper is organized as follows: Section II presents the deployment of the proposed system layers, the structure, and the workflow. Section III shows the results and comparison. In Section IV, we conclude the paper.

II. System Design

2.1 System Layers

The smart home system has four deployment layers in our system: device, gateway, blockchain, and application as shown in Fig. 1.

At the device layer, the sensors collect information and communicate with the controller that stores the data in a local database. and is responsible for communicating with the router.

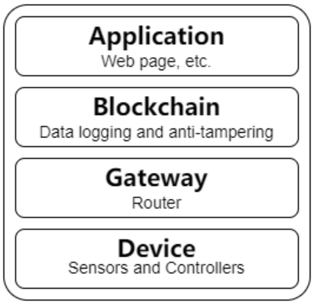


그림 1. 시스템 계층
Fig. 1 System Layers

The gateway layer has a router that exchanges data on the blockchain network. Data is sent through the router to the blockchain nodes.

The blockchain layer records the operation logs and data information of each node in the network and defines the rules of interaction in the network.

The application layer is a web page and a Node.js based client. The user accesses all the data within the permissions through the web page and performs conditional queries through regular expressions. The client adds, deletes, and checks data in the database.

2.2 Proposed System Structure

The proposed system architecture consists of a home part and a blockchain network part as depicted in Fig. 2.

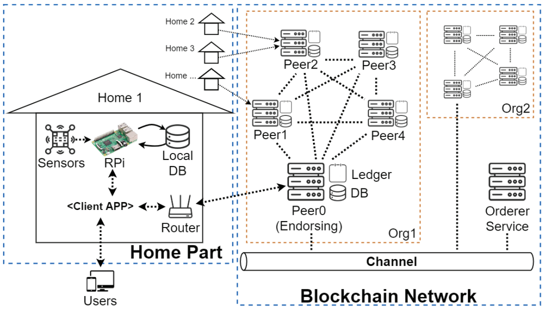


그림 2. 제안된 시스템의 다이어그램
Fig. 2 Diagram of the proposed system

The home part includes some sensors, RPi, routers, etc. and the blockchain network consists of several nodes. Each home in different regions can access the nearest 'peer' node. A node in Fabric belongs to its' corresponding organizations(Org), and the Org independently manages the nodes, users, and certificates. In each Org, a 'peer' node is selected as the endorsing peer, which is responsible for checking the transactions of requests and giving credibility to the transactions. The Raft consensus algorithm[17] selects the endorsing peer. The endorsing peer continuously sends to other peers heartbeats that are messages to confirm whether the peer is still working. When the endorsing peer is crashed, the heartbeat stops. Then the remaining peers initiate a vote to select a new endorsing peer. The network also includes an ordering service that is responsible for ordering and generating blocks of transactions. The ordering service keeps the database consistent for each peer. Channel is a bridge between different Orgs, and

Orgs share data only in the same channel. The channel keeps all the protocols and information required to communicate and the transactions between Orgs are recorded in the ledger as a form of blocks. Orgs that join the same channel have the same ledger. The policies of consensus and signature rights are recorded in the channel. CouchDB[18] is used for the database on the 'peer' node and a cache size of 64M is set to reduce the latency of accessing the database. Fig. 3 shows the complete structure of the Org.

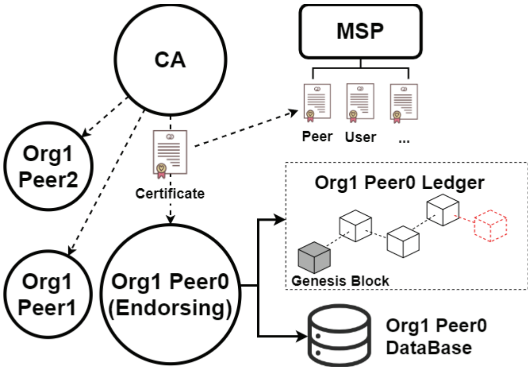


그림 3. 제안된 조직 구조
Fig. 3 Proposed structure of the Org

Each Org has a Certificate Authority(CA) and Member Service Provider(MSP). The CA is responsible for issuing certificate files to each member(Peer, users, devices) of the network. This system uses ECC algorithm to generate the public and private keys and uses the Elliptic Curve Digital Signature Algorithm(ECDSA)[19] for signing. The MSP stores all certificates and public keys of the current Org, so the MSP is used to manage all members of the organization and authenticate all identities. The CA and MSP of each Org are independent and each Org manages its users independently.

Each node in the Org has a ledger and a database. The ordering service makes all ledgers and databases always consistent. The ledger records all access requests and forms a tamper-proof chain through a hashing algorithm. The database stores the data of each home which is queried or added by access requests. The workflow for identity issuance and transaction requests is shown in Fig. 4.

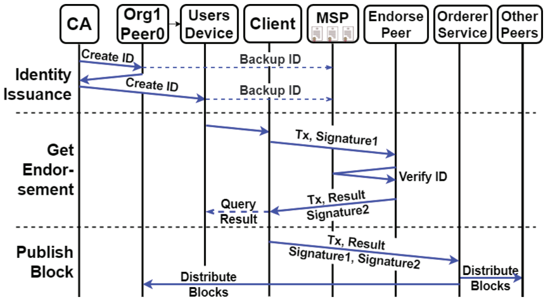


그림 4. 신분증 발급 및 거래 요청 워크플로
Fig. 4 Workflow of Identity Issuance and Transaction Request

When making a network, the network administrator first makes the Orderer Service and CA nodes and generates a certificate file for the Orderer Service. Then, the Orderer Service generates certificates and secret keys for all peers in the Org through the CA. A peer also generates certificates and key pairs for users and devices within the node through CA. All generated certificates and public keys are backed up in the MSP, and role-based access control(RBAC) is used to manage the permissions of these certificates. In this system, the certificate of the Orderer Service node is given the highest privileges. The highest privileges include adding network nodes, querying, and submitting data. The privileges of the 'Peer' node include querying and submitting data. User privileges include only querying data, and device privileges include only submitting data. RBAC makes it easier to preserve the relationship between roles and permissions. The user and RPi send signed transaction(Tx) requests to the endorsing peer through the client, and Txs are encrypted by the public key of the endorsing peer. The endorsement peer then verifies the legitimacy of Tx initiator through the MSP. If the identity is legitimate, the endorsing peer simulates running the request within Tx, but does not record the result in the ledger. The endorsing peer then signs on Tx and the result. After encryption, the result is returned to the client. If Tx is a query request, the user gets the data in the returned result. If Tx is a submission request, the client packs Tx, the result, and the signature together as a packet which is sent to the ordering service. The ordering service sorts Txs and creates blocks. Whenever a new block is created, the ordering service broadcasts it in the blockchain network using the Gossip[20] protocol. The 'peer' nodes in the network then update their ledgers and databases upon receiving the blocks.

III. Results and Analysis

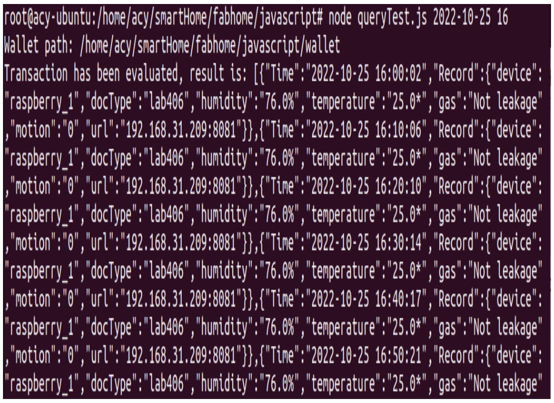
In this paper, one PC and two RPi's were used for the experiments and the hardware and software configurations are shown in Table 1.

표 1. 하드웨어 및 소프트웨어 구성
Table 1. Hardware and software configuration

Hardware	
CPU	i7-8750H
Memory	8G * 2
Hard Disk	512G
Software	
OS	Ubuntu 22.04 LTS
Docker	v20.10.17
Docker-Compose	v1.29.2
Golang	v1.18.3
Node.js	v12.22.9
Hyperledger Fabric	v2.2.0

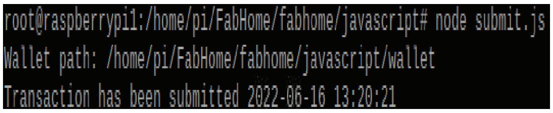
The system uses Docker on the PC to simulate all nodes of the blockchain network. DHT11, MQ-2[21], RPi camera module 2 NoIR(No Infrared filter)[22], and HC-SR501[23] are used in the home. The local database of RPi uses SQLite[24].

In this paper, smart contracts are written in JavaScript and contain functions for querying and submitting data. This smart contract provides APIs for client-side calls. On the Node.js-based client, 'queryTest.js' is used for the user to query data, and 'submit.js' is used for the RPi to submit data. Fig. 5 shows the results of queryTest.js and Fig. 6 shows the results of submit.js.



```
root@acy-ubuntu:/home/acy/smartHome/fabhome/javascript# node queryTest.js 2022-10-25 16
Wallet path: /home/acy/smartHome/fabhome/javascript/wallet
Transaction has been evaluated, result is: [{"Time":"2022-10-25 16:00:02","Record":{"device":
"raspberry_1","docType":"lab406","humidity":"76.0%","temperature":"25.0*","gas":"Not leakage"
,"motion":"0","url":"192.168.31.209:8081"}},{"Time":"2022-10-25 16:10:06","Record":{"device":
"raspberry_1","docType":"lab406","humidity":"76.0%","temperature":"25.0*","gas":"Not leakage"
,"motion":"0","url":"192.168.31.209:8081"}},{"Time":"2022-10-25 16:20:10","Record":{"device":
"raspberry_1","docType":"lab406","humidity":"76.0%","temperature":"25.0*","gas":"Not leakage"
,"motion":"0","url":"192.168.31.209:8081"}},{"Time":"2022-10-25 16:30:14","Record":{"device":
"raspberry_1","docType":"lab406","humidity":"76.0%","temperature":"25.0*","gas":"Not leakage"
,"motion":"0","url":"192.168.31.209:8081"}},{"Time":"2022-10-25 16:40:17","Record":{"device":
"raspberry_1","docType":"lab406","humidity":"76.0%","temperature":"25.0*","gas":"Not leakage"
,"motion":"0","url":"192.168.31.209:8081"}},{"Time":"2022-10-25 16:50:21","Record":{"device":
"raspberry_1","docType":"lab406","humidity":"76.0%","temperature":"25.0*","gas":"Not leakage"
```

그림 5. queryTest.js의 결과
Fig. 5 Results of queryTest.js

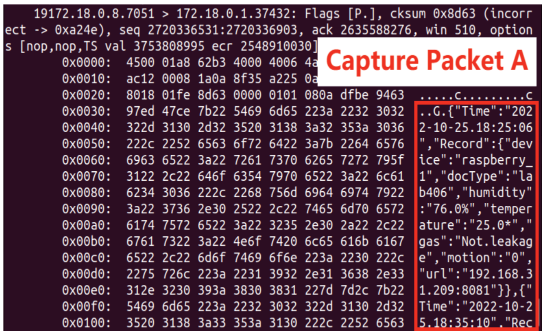


```
root@raspberrypi1:/home/pi/FabHome/fabhome/javascript# node submit.js
Wallet path: /home/pi/FabHome/fabhome/javascript/wallet
Transaction has been submitted 2022-06-16 13:20:21
```

그림 6. submit.js의 결과
Fig. 6 Results of submit.js

In queryTest.js, the date and hour are used as parameters, and all data for the specified period is successfully returned. The Wallet path in both figures is the location where the user and RPi store their identity files. The identity files include certificates and private keys. In submit.js, PRi reads the latest data from the local database and submits the data as an argument to the API.

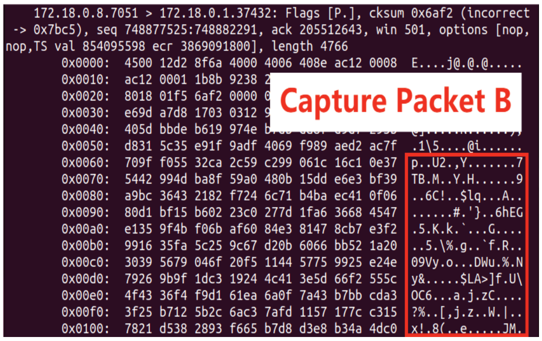
In this system, ECC is used to encrypt the data so that cyber attackers cannot access the data during transmission. This ensures the security of the data transmission. This experiment uses the 'tcpdump' packet capture tool to capture the data submitted by the client to Peer0 in Fig. 2 and compare the results obtained by transmitting the data without ECC encryption. Fig. 7 is an example of data packet and Fig. 8 shows an example of data packet after ECC encryption.



```
    19172.18.0.8.7051 > 172.18.0.1.37432: Flags [P.], cksum 0x8d63 (incorr
ect -> 0xa24e), seq 2720336531:2720336903, ack 2635588276, win 510, option
s [nop,nop,TS val 3753808995 ecr 2548910030]
        0x0000:  4500 01a8 62b3 4000 4006 4a
        0x0010:  ac12 0008 1a0a 8f35 a225 0a
        0x0020:  8018 01fe 8d63 0000 0101 080a dfbe 9463  .....c.........c
        0x0030:  97ed 47ce 7b22 5469 6d65 223a 2232 3032  ..G.{"Time":"202
        0x0040:  322d 3130 2d32 3520 3138 3a32 353a 3036  2-10-25.18:25:06
        0x0050:  222c 2252 6563 6f72 6422 3a7b 2264 6576  ","Record":{"dev
        0x0060:  6963 6522 3a22 7261 7370 6265 7272 795f  ice":"raspberry_
        0x0070:  3122 2c22 646f 6354 7970 6522 3a22 6c61  1","docType":"la
        0x0080:  6234 3036 222c 2268 756d 6964 6974 7922  b406","humidity"
        0x0090:  3a22 3736 2e30 2522 2c22 7465 6d70 6572  :"76.0%","temper
        0x00a0:  6174 7572 6522 3a22 3235 2e30 2a22 2c22  ature":"25.0*","
        0x00b0:  6761 7322 3a22 4e6f 7420 6c65 616b 6167  gas":"Not.leakag
        0x00c0:  6522 2c22 6d6f 7469 6f6e 223a 2230 222c  e","motion":"0",
        0x00d0:  2275 726c 223a 2231 3932 2e31 3638 2e33  "url":"192.168.3
        0x00e0:  312e 3230 393a 3830 3831 227d 7d2c 7b22  1.209:8081"}},{"
        0x00f0:  5469 6d65 223a 2232 3032 322d 3130 2d32  Time":"2022-10-2
        0x0100:  3520 3138 3a33 353a 3130 222c 2252 6563  5.18:35:10","Rec
```

그림 7. 데이터 패킷 예제
Fig. 7 An example of data packet

Capture Packet A is a data packet transmitted without encryption, and the data is transmitted in plaintext over the network so the home data be easily viewed.



```
    172.18.0.8.7051 > 172.18.0.1.37432: Flags [P.], cksum 0x6af2 (incorrect
 -> 0x7bc5), seq 748877525:748882291, ack 205512643, win 501, options [nop,
nop,TS val 854095598 ecr 3869091800], length 4766
        0x0000:  4500 12d2 8f6a 4000 4006 408e ac12 0008  E....j@.@.@.....
        0x0010:  ac12 0001 1b8b 9238 2
        0x0020:  8018 01f5 6af2 0000 0
        0x0030:  e69d a7d8 1703 0312 9
        0x0040:  405d bbde b619 974e
        0x0050:  d831 5c35 e91f 9adf 4069 f989 aed2 ac7f  .1\5....@i......
        0x0060:  709f f055 32ca 2c59 c299 061c 16c1 0e37  p..U2.,Y.......7
        0x0070:  5442 994d ba8f 59a0 480b 15dd e6e3 bf39  TB.M..Y.H......9
        0x0080:  a9bc 3643 2182 f724 6c71 b4ba ec41 0f06  ..6C!..$lq...A..
        0x0090:  80d1 bf15 b602 23c0 277d 1fa6 3668 4547  ......#.'}..6hEG
        0x00a0:  e135 9f4b f06b af60 84e3 8147 8cb7 e3f2  .5.K.k.`...G....
        0x00b0:  9916 35fa 5c25 9c67 d20b 6066 bb52 1a20  ..5.\%.g..`f.R..
        0x00c0:  3039 5679 046f 20f5 1144 5775 9925 e24e  09Vy.o...DWu.%.N
        0x00d0:  7926 9b9f 1dc3 1924 4c41 3e5d 66f2 555c  y&.....$LA>]f.U\
        0x00e0:  4f43 36f4 f9d1 61ea 6a0f 7a43 b7bb cda3  OC6...a.j.zC....
        0x00f0:  3f25 b712 5b2c 6ac3 7afd 1157 177c c315  ?%..[,j.z..W.|..
        0x0100:  7821 d538 2893 f665 b7d8 d3e8 b34a 4dc0  x!.8(..e.....JM.
```

그림 8. ECC 암호화 후 데이터 패킷 예제
Fig. 8 An example of data packet after ECC encryption

Capture Packet B is a packet transmitted after ECC encryption, and the data is transmitted in the garbled form on the network. The receiver must have the private key corresponding to the public key to decode and view the data. In addition, every message transmission is encrypted in the proposed blockchain system. ECC encryption effectively addresses the security of data transmission and ensures the privacy of users and devices accessing the network.

In this paper, the performance of the proposed system architecture is tested by simulating concurrent accesses of multiple threads. The time required by the smart contract to process different numbers of concurrent requests is calculated. In addition, 10, 25, 50, 100, 200, and 300 virtual client requests were set up to access the network concurrently. The statistical results are shown in Fig. 9 and 10.

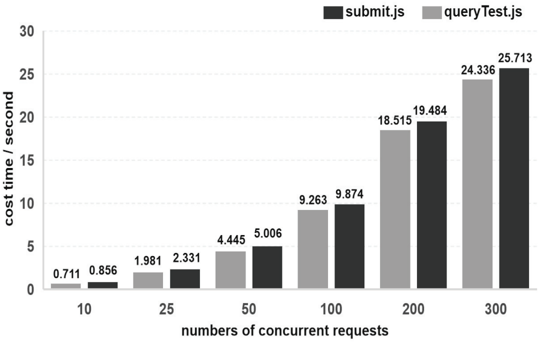


그림 9. 동시 요청에 대한 시간 비용
Fig. 9 Time cost for concurrent requests

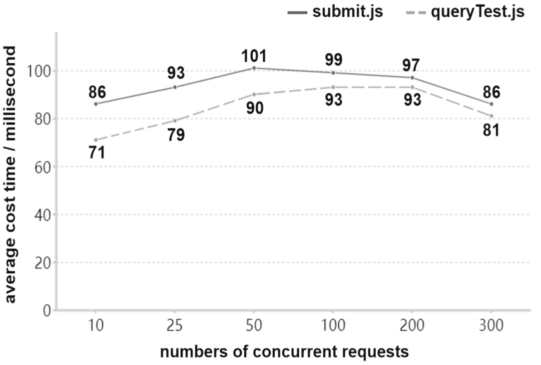


그림 10. 동시 요청에 대한 스마트 계약의 평균 비용 시간
Fig. 10 The average cost time of smart contracts for concurrent requests

From the data in these two graphs, the average time for queryTest.js is 84.5 [ms], and the average time for submit.js is 93.67 [ms]. The throughput of the system increases as the number of requests increases, while the average cost time remains stable, indicating that the system maintains high throughput for larger requests. The proposed system with millisecond access latency effectively improves the device access efficiency and the high throughput allows the system to access more homes without worrying about network congestion.

Ⅳ. Conclusion

In this paper, a blockchain-based smart home system is implemented. The smart contract is written in JavaScript and allows users or devices to access the blockchain through direct calls from a Node.js-based client. The smart contract has millisecond access latency, so the client queries and submits data faster. The Orderer service ensures the consistency of the ledger. The messages in the network are encrypted and sent through the ECC algorithm, and attackers cannot access the transmitted home data in the packet capture. Meanwhile, the smart contract detects the legitimacy of the identity, and there is no problem with false identity access, so the proposed data transmission scheme is feasible. The experimental results prove that the system ensures data security and high data throughput.

References

[1] I. Yaqoob, E. Ahmed, I. A. T. Hashem, A. I. A. Ahmed, A. Gani, M. Imran, and M. Guizani, "Internet of things architecture: Recent advances, taxonomy, requirements, and open challenges," *IEEE wireless communications,* vol. 24, no. 3, June 2017, pp. 10-16.

[2] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE communications surveys & tutorials,* vol. 17, no. 4, June 2015, pp. 2347-2376.

[3] T. CHOI, D. Ryu, "Development of Portable IoT Device for Lifesaving," *J. of The Korea Institute of Electronic Communication Sciences,* vol. 17, no. 5, Oct. 2022, pp. 883~888.

[4] M. AbuNaser and A. A. Alkhatib, "Advanced survey of blockchain for the internet of things smart home," *2019 IEEE Jordan Int. joint Conf. on electrical engineering and information technology (JEEIT),* Amman, Jordan, May 2019, pp. 58-62.

[5] M. Lee and K. Jung, "Design of IoT-based Energy Monitoring System for Residential Building," *J. of The Korea Institute of Electronic Communication Sciences,* vol. 16, no. 6, Dec. 2021, pp. 1223-1230.

[6] Y. Lee, S. Rathore, J. H. Park, and J. H. Park, "A blockchain-based smart home gateway architecture for preventing data forgery," *Human-centric Computing and Information Sciences,* vol. 10, no. 1, Mar. 2020, pp. 1-14.

[7] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in Internet of Things: The road ahead," *Computer networks,* vol. 76, Jan. 2015, pp. 146-164.

[8] L. Lao, Z. Li, S. Hou, B. Xiao, S. Guo, and Y. Yang, "A survey of IoT applications in blockchain systems: Architecture, consensus, and traffic modeling," *ACM Computing Surveys (CSUR),* vol. 53, no. 1, Feb. 2020, pp. 1-32.

[9] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Business Review,* vol. 21260, Aug. 2019.

[10] X. Yuan, J. Chen, N. Zhang, X. Fang, and D. Liu, "A federated bidirectional connection broad learning scheme for secure data sharing in Internet of Vehicles," *China Communications,* vol. 18, no. 7, July 2021, pp. 117-133.

[11] R. Madani, B. Alturki, S. Reiff-Marganiec, and W. Alsafery, "My smart remote: A smart home management solution for children," *2018 1st Int. Conf. on Computer Applications & Information Security (ICCAIS),* Riyadh, Saudi Arabia, Apr.

2018, pp. 1-8.

[12] D. S. Namdeo and V. R. Pawar, "A review: IoT based power & security management for smart home system," *2017 Int. Conf. of Electronics, Communication and Aerospace Technology (ICECA),* Coimbatore, India, Apr. 2017, pp. 552-556.

[13] C. Lin, D. He, N. Kumar, X. Huang, P. Vijayakumar, and K. K. R. Choo, "HomeChain: A blockchain-based secure mutual authentication system for smart homes," *IEEE Internet of Things J.,* vol. 7, no. 2, Sept. 2019, pp. 818-829.

[14] M. Ma, Z. He, Q. Xu, and X. J. Li, "Design and development of smart home sensing supported by blockchain technology," In *Proc. the 2019 7th Int. Conf. on Information Technology: IoT and Smart City,* Shanghai, China, Dec. 2019, pp. 525-530.

[15] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, and J. Yellick, "Hyperledger fabric: a distributed operating system for permissioned blockchains," In *Proc. the thirteenth EuroSys Conf.,* Porto, Portugal, Apr. 2018, pp. 1-15.

[16] C. Boettiger, "An introduction to Docker for reproducible research," *ACM SIGOPS Operating Systems Review,* vol. 49, no. 1, Jan. 2015, pp. 71-79.

[17] D. Ongaro and J. Ousterhout, "The raft consensus algorithm," *Lecture Notes CS,* Oct. 2015.

[18] H. Javaid, C. Hu, and G. Brebner, "Optimizing validation phase of hyperledger fabric," 2019 *IEEE 27th Int. Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS),* Rennes, France, Oct. 2019, pp. 269-275.

[19] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ECDSA)," *Int. J. of information security,* vol. 1, no. 1, July 2001, pp. 36-63.

[20] G. Saldamli, C. Upadhyay, D. Jadhav, R. Shrishrimal, B. Patil, and L. A. Tawalbeh, "Improved gossip protocol for blockchain applications," *Cluster Computing,* vol. 25, no. 3, Jan. 2022, pp. 1915-1926.

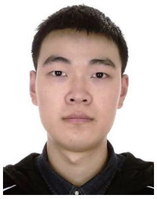
[21] R. C. Pandey, M. Verma, L. K. Sahu, and S. Deshmukh, "Internet of things (IOT) based gas leakage monitoring and alerting system with MQ-2 sensor," *Int. J. of Engineering Development and Research,* vol. 5, no. 2, Jan. 2017, pp. 2135-2137.

[22] M. A. Pagnutti, R. E. Ryan, M. J. Gold, R. Harlan, E. Leggett, and J. F. Pagnutti, "Laying the foundation to use Raspberry Pi 3 V2 camera module imagery for scientific and engineering purposes," *J. of Electronic Imaging,* vol. 26, no. 1, Feb. 2017, pp. 1-13.

[23] R. Wahyuni, A. Rickyta, U. Rahmalisa, and Y. Irawan, "Home security alarm using Wemos D1 and HC-SR501 sensor based telegram notification," *J. of Robotics and Control (JRC),* vol. 2, no. 3, May 2021, pp. 200-204.

[24] L. Junyan, X. Shiguo, and L. Yijie, "Application research of embedded database SQLite," *2009 Int. Forum on Information Technology and Applications,* Chengdu, China, May 2009, pp. 539-543.

저자 소개



아창위(Chang-Yu Ao)

2020년 Beijing Institute of Petrochemical Technology, Communication Engineering 졸업(공학사)
2021년 ~현재 전남대학교 대학원 컴퓨터공학과 재학(공학석사)
※ 관심분야 : Blockchain, IoT

김강철(Kang-Chul Kim)

1981년 서강대학교 전자공학과 학사
1983년 서강대학교 전자공학과 석사
1996년 경상대학교 전자공학과 박사
현재 전남대학교 전기컴퓨터공학부 교수
※ 관심분야 : 임베디드시스템, NoC, IoT
Pattern Recognition