J Inf Process Syst, Vol.19, No.2, pp.240~257, April 2023
https://doi.org/10.3745/JIPS.03.0184

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

A Novel Smart Contract based Optimized Cloud Selection Framework for Efficient Multi-Party Computation

Haotian Chen, Abir EL Azzaoui, Sekione Reward Jeremiah, and Jong Hyuk Park*

Abstract

The industrial Internet of Things (IIoT) is characterized by intelligent connection, real-time data processing, collaborative monitoring, and automatic information processing. The heterogeneous IIoT devices require a high data rate, high reliability, high coverage, and low delay, thus posing a significant challenge to information security. High-performance edge and cloud servers are a good backup solution for IIoT devices with limited capabilities. However, privacy leakage and network attack cases may occur in heterogeneous IIoT environments. Cloud-based multi-party computing is a reliable privacy-protecting technology that encourages multi-party participation in joint computing without privacy disclosure. However, the default cloud selection method does not meet the heterogeneous IIoT requirements. The server can be dishonest, significantly increasing the probability of multi-party computation failure or inefficiency. This paper proposes a blockchain and smart contract-based optimized cloud node selection framework. Different participants choose the best server that meets their performance demands, considering the communication delay. Smart contracts provide a progressive request mechanism to increase participation. The simulation results show that our framework improves overall multi-party computing efficiency by up to 44.73%.

Keywords

Blockchain, Cloud Computing, Industrial Internet of Things, Multi-party Computing, Smart Contract

1. Introduction

Cyber risks are numerous, and the impact of the coronavirus disease 2019 (COVID-19) pandemic in 2019 exacerbated these challenges. The COVID-19 pandemic has affected both large and small enterprises economically, spawning many online business models and prompting many companies to explore online markets, which on the other side, increases network and information security challenges. Alliance Virtual Offices (AVO) analyzed hundreds of studies and millions of data points to determine the impact and costs of continuing to work remotely or hybrid in businesses after the pandemic. One key finding from the AVO report was that cybercriminals' target on remote workers increased, and there was a 238% increase in cyber-attacks on remote server access due to working from home [1]. In that regard, network security is increasingly valued by enterprises and users. One of the significant challenges facing network security today is the urgent need to address privacy and trust among heterogeneous devices. In

Manuscript received November 22, 2022; first revision January 26, 2023; second revision March 2, 2023; accepted March 2, 2023.

* **Corresponding Author:** Jong Hyuk Park (jhpark1@seoultech.ac.kr)
Dept. of Computer Science and Engineering, Seoul National University of Science and Technology, Seoul, Korea (chahot@seoultech.ac.kr, abir.el@seoultech.ac.kr, reward@seoultech.ac.kr, jhpark1@seoultech.ac.kr)

particular, dishonest third-party entities in centralized data centers may sell large amounts of private data for financial gain [2]. With the continuous development of wireless communication and information processing technology, many intelligent devices are needed to process real-time data in heterogeneous IIoT environment. However, they face significant challenges since they dynamically communicate and provide processed results to achieve industrial intelligence [3].

The industrial Internet of Things (IIoT) collects data through heterogeneous intelligent devices and transmits collected data to servers, a classic application scenario of a distributed network. IIoT leverages deep learning, natural language processing, computer vision, and robotic process automation to build an intelligent industrial system. Driven by the IIoT, smart devices are overgrowing, and the volume of data is increasing. In addition, heterogeneous devices' have different computing power, storage capacity and security mechanism [4], causing them security risks. For example, in multiparty computing, the heterogeneity of the IIoT could be a starting point for cyber-attacks. Attackers can access networks and sensitive data by hacking IIoT devices with lower performance than ordinary computers to learn enterprise entities' latest business strategies or objectives [5]. In some cases, business partnerships require sensitive data exchange to accomplish work collaboration, which many do not desire. This is because the privately collected data can be interpreted through data mining and big data analysis techniques to understand an entity's business strategies and objectives. Such undertakings can negatively affect business initiatives and plans [6].

Studies show that blockchain is a reliable technology to solve the problems of privacy disclosure and trust [7]. Blockchain can solve the problem of source trust and result verification, but it lacks the protection function for the privacy of computing input. Still, multi-party computation (MPC) can fully protect this input. In the meantime, MPC has the characteristics of privacy of information, the correctness of calculation and decentralization, and maintaining data confidentiality to solve the problems of private data sharing, data analysis and mining [8]. In a distributed network, multiple participating entities hold their private input data, and all parties hope to complete the calculation of a specific function jointly. Still, each participating entity must obtain no input information from other participating entities except the calculation results. MPC guarantees that neither party shall acquire any information other than its processed output and correctness of the received data. Moreover, MPC prevents dishonest participants denial-of-service attacks interruptions [9].

However, the efficiency of MPC in IIoT has been a high-profile challenge. Since most of the devices in IIoT don't have enough computing power to do MPC, they often require a robust cloud server to do the calculations, and the IIoT devices are only responsible for transferring their data. Second, the input to the MPC may be illegal, and performing calculations with invalid inputs may result in wasted time and computing resources. In contrast, anyone, by default, can join multiparty calculations. In addition, the default first-come, first-served service mode leads to strong randomness in the selection of cloud servers. Tasks requiring complex computing power may be assigned to servers with relatively low computing power, thus slowing down the progress of the overall IIoT intelligence work. Finally, when the user is unsatisfied with the cloud service, there is no way to provide feedback to the cloud server.

In this paper, we propose a smart contract-based MPC cloud server selection framework to ensure the effectiveness and efficiency of all MPC participants. We aim to develop a high-efficiency framework for valid identity authentication under the IIoT environment. Our framework divides the IIoT system into three layers, with blockchain deployed at the edge layer to provide verification. Smart contracts deployed at the cloud layer ensures efficient operation of MPC at the cloud layer. The framework solves the following problems: (1) overall efficiency defects of cloud selection-based MPC; (2) lack of incentive mechanism between an efficient cloud server and participating edge nodes; (3) unknown node participating in MPC; and (4) cloud servers' repudiation of their calculation results.

The main contributions of this paper are as follows:

- We propose an efficient server selection method that enables each device to select the most suitable server for computing its task and dramatically reducing computing latency.
- Blockchain provides authentication and identification functions. It ensures that all nodes joining the MPC are authenticated, and unauthenticated nodes cannot access network resources in the framework. Only devices registered in the Blockchain can participate in cloud computing.
- The blockchain network ensures the witness and accountability characteristics of the computational results.
- Smart contracts help the cloud to achieve a highly differentiated server hierarchy so that users with different needs can choose the most suitable server according to their needs.
- Smart contracts marginalize untrusted cloud nodes by assessing their reputation level in the cloud environment. Furthermore, it prevents malicious cloud nodes from controlling the MPC process.

The rest of the paper is organized as follows: Section 2 presents a general overview of other existing research. Section 3 proposes a smart contract-based MPC cloud server selection framework for IIoT in the cloud environment. We further provide an overview of the architecture and explain its process flows. Section 4 evaluates the proposed framework with extensive simulations analysis and presents our findings. Finally, in Section 5, we conclude this paper.

2. Related Work

The IIoT paradigm has developed rapidly, and the security mechanism is constantly improving. In this section, we discuss the key considerations and existing researches attempting to improve the efficiency of multiparty computing in IIoT networks.

2.1 Key Considerations

An IIoT-based cloud environment has four critical requirements for efficient multiparty computation.

Efficiency: Some subindustries in the IIoT require low-latency data processing, while others require high-bandwidth data processing. However, regardless of the utilized model, the computational efficiency should be as high as possible under the appropriate model. This paper provides an MPC model based on optimized cloud selection with higher computing efficiency and lower communication delay.

Scalability: IIoT has so many subindustries that it is difficult to have a single model that perfectly fits all industrial needs. But it should be compatible as much as possible or as easily adjustable to match the diverse technology ecosystem of the future. In this paper, all edge nodes can choose the server that suits them according to their own needs and thus perform well even in a heterogeneous IIoT environment.

Availability: Many IIoT devices are based on real-time communication connections and operations. In distributed networks, if one node (server or device) fails, processes on other nodes should be able to continue. In this paper, smart contract has independent control over edge nodes, and each edge node is independently subject to the control of a smart contract. Therefore, even if some edge or cloud nodes have problems, the complete network service will not be affected.

Non-repudiation: Cloud servers' service quality and attitude differ in the mixed cloud environment. There may be cases where the cloud server makes a fatal error but denies it. In this paper, blockchain technology is used. The certified nodes store the results of cloud selection and the values of the calculated results in the blockchain; no one can modify these contents. This ensures that the cloud server cannot deny its behavior and faults.

2.2 Existing Researches

IIoT is a globally distributed system composed of intelligent objects, physical information assets, related general information technology, and auxiliary computing platforms such as cloud or edge computing [10]. For IIoT, performance needs to be evaluated from multiple perspectives, using techniques appropriate to its characteristics. Table 1 summarizes some studies on networks efficiency [11–15].

Table 1. Summary of existing research

Study	Year	Technique	Contributions	Limitation
Zhou et al. [11]	2020	Blockchain, Spectrum sharing	Solve the problem of insufficient spectrum resources in the IoT era.	The distribution information based on income is uncertain
Tian et al. [12]	2022	Blockchain, Deep learning, Edge computing	Built a new type of smart contract to encourage multiple parties to participate in edge services and improve data processing efficiency.	High communication overhead
Bugshan et al. [13]	2023	Deep learning, Edge computing	Balancing privacy protection and edge network model performance.	High communication overhead
Kumari et al. [14]	2021	Blockchain, Big data analysis	Used blockchain to improve the potential of industrial IIoT services, provide privacy protection, and prevent single points of failure.	Communication and scalability are lacking
Li et al. [15]	2022	Homomorphic encryption, MPC	A homomorphic algorithm with high computational efficiency is proposed to ensure the privacy protection of data users.	Slightly lower communication efficiency
Our work	2022	Smart contract, Blockchain	Provides an efficient MPC computing framework using blockchain and smart contracts.	High energy costs

Zhou et al. [11] pointed out that the increase of IoT devices will lead to the problem of insufficient spectrum resources of the 5G network on which the IoT depends and the solution of allocating unused spectrum of human-to-human network to machine-to-machine network. An asymmetric framework with privacy protection, incentive compatibility, and efficient spectrum allocation is designed from a realistic perspective. This research considers secure resource allocation over physical channels but does not specify a secure design for edge services. Tian et al. [12] expounded on the problems of privacy disclosure and precision insufficiency of edge computing with the increasing number of intelligent devices and proposed a learning framework suitable for the IIoT environment. Their proposed framework improves multiparty learning models' accuracy and data processing efficiency on edge services. The size weighted aggregation strategy was presented to verify and integrate model parameters and improve the model's accuracy. Theoretical analysis and simulation show that this method protects data security and has low overhead for the edge service of IIoT. Bugshan et al. [13] developed a distributed privacy protection technology based on microservices, using differential privacy and radial basis function network to balance the model performance of privacy protection and edge network. Kumari et al. [14] emphasized the data transmission problem in the industrial Internet environment. They proposed a Blockchain-based decentralized chemical IIoT model to meet the security and privacy of mass data. A simulation comparison is made from load balance, energy management cost, and transmission delay, and the results show that the delay is lower than in the traditional Blockchain system.

It was an exciting finding that many researchers used security and efficiency as essential indicators, especially for network research. Li et al. [15] classified three types of privacy among data owners, third-party cloud servers, and data users. A lightweight privacy protection protocol for the data owner server data user model is proposed based on homomorphic encryption. Simulation results show that the scheme effectively prevents privacy leakage in the IoT. The cost of evaluating computing moves from resource-constrained IIoT devices to powerful third-party servers without compromising security and privacy. This approach allows remote, untrusted cloud computing to perform complex computations on encrypted data. It also allows data owners and users to verify decryption accuracy. However, there are apparent communication efficiency problems.

3. Smart Contract-based Optimal Node Selection Framework

This section presents a hierarchical description of the proposal's architecture, and we discuss the process flow of data at different levels and explain the workings of the smart contract. Finally, from a general perspective, the whole data processing stages is presented. As shown in Fig. 1, the proposed MPC framework based on smart contracts is divided into three main layers: the IIoT device layer, the edge layer, and the cloud layer. Data is transmitted bottom-up from the device layer to the cloud. In this section, we will discuss the role of each layer and the details of how they work.

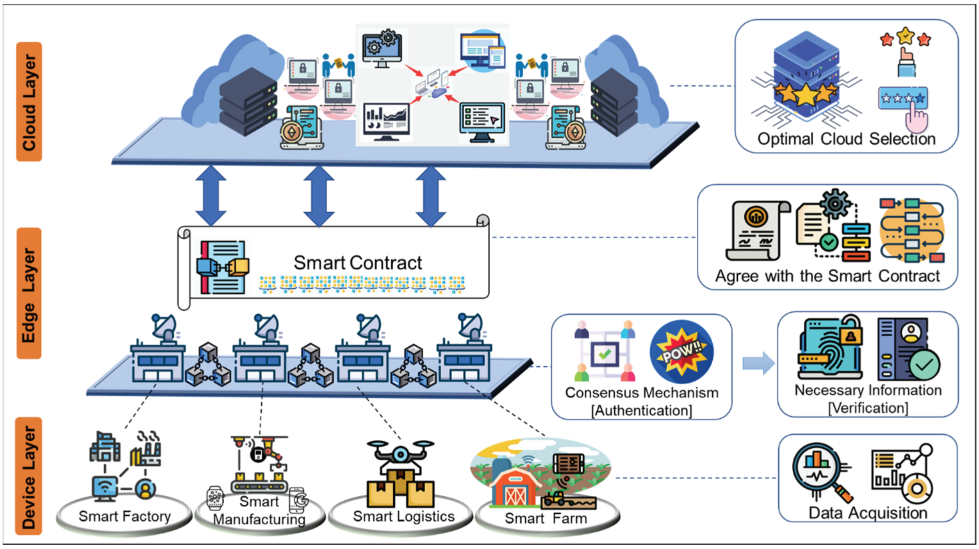

Fig. 1. Proposed smart contract-based optimal node selection framework.

Device layer: At this layer, sensors collect various industrial data and perform limited data operations on local smart devices before forwarding the data to the upper layer, the edge layer. The computational performance of smart devices is much lower than that of conventional computers, so their data operations cannot be done locally. High-performance servers in the cloud are the desired target for device layer data.

Edge layer: A public Blockchain network is provide on this layer where devices are authenticated. Here, other nodes in the network need to recognize all participants through the consensus mechanism. At the same time, as the manager of the blockchain, the base station must carry out essential identification and legitimacy authentication for all data collected by the device layer. In this layer, the users are now termed edge node.

Cloud layer: Each edge node selects one server from many cloud servers and uses it for work. When the work is finished, the corresponding edge nodes can give evaluation feedback to the server, which will affect the comprehensive performance evaluation of the server in the cloud. The input confidentiality of MPC ensures that the attacker is unaware of other edge nodes' input information during the whole process from the perspective of the mathematical theory of discrete algebra. Suppose the encrypted data cannot be decrypted in real time, or the cost required to decrypt it is much higher than the value of the data itself. In that case, we consider this method of encryption mathematically secure. Because of smart contracts' the speed of server selection that meets edge node indicators is significantly increased, and the computing efficiency is also greatly improved.

3.1 Workflow of the Proposed Framework

Fig. 2 illustrates the workflow of the device and edge layers for the proposed framework to illustrate a more accurate data flow and network structure.

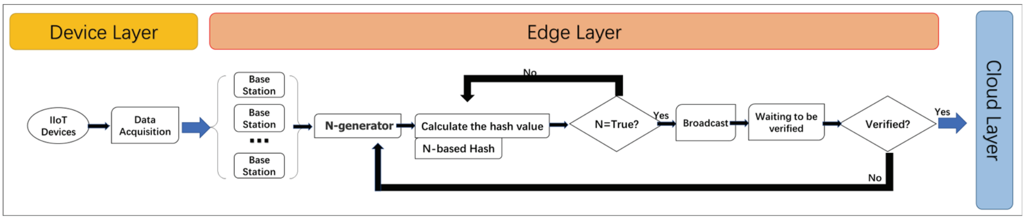

Fig. 2. Workflow of proposed framework about device and edge layer.

The sensor collects data and uploads it to the edge layer, where authentication is required for security. We use the public blockchain here, allowing all nodes to join the blockchain network and using proof-of-work (PoW) for authentication. All edge nodes that want to join multiparty computing at the edge layer must register with the Blockchain system. They need to go through the consensus mechanism when they want to transfer data to the upper layer. The entity target of authentication is the device itself. To save energy, the device can continuously send transmitted data after being authenticated once rather than requiring authentication each time. This can significantly save computing resources.

As shown in Algorithm 1, edge nodes will compute a hash value with the first *N* terms being 0. This is a random problem, and the device that wants to be authenticated needs to compete with other nodes to get this number quickly, so the process is fair for all nodes involved in the calculation. The number of bits of *N* depends on the consensus algorithm. A new *N* of similar complexity is generated when an *N* is computed.

Algorithm 1. Blockchain authentication & verification

```
Input: Device ID and the request message to join the Cloud
Output: Decision and final consensus result (If the device can join and participate in the cloud or not)
Process:
$CI_k$.Send(<$C_{id}$, Msg, t>, request, BS);    //with $C_{id}$: client identify, t: timestamp, BS: Base station
B.S. Verify (<$C_{id}$, Msg, t>, C);
$C_0$.Prepare(<$C_{id}$, Msg, t>, nonce);    //nonce: authentication for PoW
    $C_i$.nonce = HashComputation();
    while(!isValidHashDifficulty($C_i$.nonce)) {
        if (!OthersDoneReceviced($C_x$.S)    //Other miners solve the cryptographic puzzle
                break;
        else:
            nonce = nonce + 1;
            input = previous hash + timestamp + data + nonce;
            hash = CryptoJS.SHA256(input)    }
$C_0$.Broadcast(<$C_{id}$, Msg, t>,$pk_i$, hash, S);    // $pk_i$: a public key for verification, S: the digest signature of $pk_i$
$C_0$.WaitForHashCheck();
if ($C_0$.Recevice(Response) > $\frac{n}{2}$ )
    add $C_0$ToBlockhain (B.C., $C_{id}$, 0, $pk_i$ );
    connectToSmartContarct($C_{id}$)
else
    redo;
```

For all nodes in the network that want to participate in calculating a standard password puzzle, the first edge node to calculate the result will be published in the network. Other nodes in the network will witness it, and only when the result of more than half of the agreement will it be considered to reach a consensus. After the consensus mechanism is passed, the data will be ready to be sent to the cloud under the constraints of the smart contract, and a block will also be generated. When the consensus is reached, the public key of the corresponding edge node will also be transmitted to other nodes. At this time, the identity information of the data source can be confirmed through a digital signature, and it is non-repudiation. This allows the edge node (and only the edge node) to sign a signature that can be verified against a public key stored in the blockchain. This identity of the edge node can serve as a decentralized source of verification. Upon completion of the consensus, the client sends the data to the cloud on the premise that the client agrees to the smart contract.

3.2 Server Selection Process

Sensors in smart devices collect data in different industrial environments. At the edge layer, the base station acts as the manager of the blockchain, registering node registration information of all data sources. The edge node then sends the data to the cloud and, after confirming that they agree to the constraints of the smart contract, starts to choose the server. After confirmation, each device is registered with the blockchain network and is termed an Authenticator in the blockchain network.

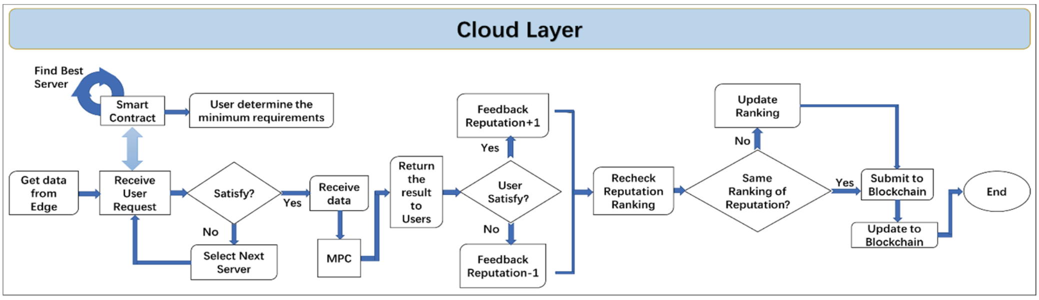

Fig. 3. Workflow of smart contract in cloud layer.

In Algorithm 2, we first consider server performance. As shown in Fig. 3, the edge nodes send a computation request to the cloud, requiring it to return its performance metrics. Only the cloud nodes that satisfy the minimum performance metrics needed for the computation operations can participate in this request. The second essential requirement is the estimated time to operate on the data held by the edge node, which will be the maximum. This measures the efficiency of a calculation, requiring a maximum number of seconds to complete the calculation. In the search results of the two items, select the intersection, which is the theoretical alternative cloud server. Performance is the priority if there is no intersection between two search results. If the intersection of the search results for the two required conditions is not unique, the edge node will use another extension condition to determine the option. At this point, the edge node will issue another query to the servers in the candidate area, asking them who can do the task faster. Then choose a better one or several of them. If we get to this point and there are still multiple servers to choose from, we choose the one that is physically closest to us, bringing lower latency to our overall process. After the server is selected, multiparty computation begins.

Algorithm 2. Server selection

```
Process:
SendRequest(MinReputation);
SendRequest(MaxCostTime);
if no one response
    skip;
else if one response
    AskForLowestRequirement();
    if yes
        Choose();
    else
        skip;
else if two response
    SendExtraRequest(performance);
    if someone is better than others
        Choose();
    else
        SendExtraRequest(distance);
        ChooseCloest(Response.distance);
```

The evaluation criterion of the edge node is the edge node's satisfaction degree to the cloud node's computing results, that is, the allowable delay of returning the computing results. After the calculation, the edge node will also evaluate the data and, based on their satisfaction with the data, send feedback and an evaluation message to the corresponding server, which will affect the performance options in the cloud server in the first step. The higher the performance and lower the delay of the server, the higher the reputation. Each time the server gets feedback, the smart contract forces it to change its reputation value, directly affecting whether other edge nodes select it and how often it is selected. When cloud servers get lousy feedback, their reputation suffers (and vice versa). Because reputation selection is the top priority for other edge nodes when choosing a server. Reputation changes will be updated to the blockchain, meaning no one can modify or change it. Any new node added to the blockchain can more easily see a cloud server's historical and current performance. All operations will be performed automatically when the preset instructions are met. Based on this feature, we store the results in the blockchain, ensuring the process is transparent, open, and traceable.

3.3 Smart Contract Design

A smart contract is a trusted global execution machine that temporarily escrows assets and executes a set of predefined instructions [16]. As shown in Algorithm 3, the smart contract initiates when all edge nodes grant control to the smart contract and start selecting the best server they want. The predefined instructions specify the conditions that must be met for the computation to proceed. Based on this feature, we store the results in the blockchain, ensuring the process is transparent, open, and traceable. As shown in Fig. 4, the first two requests are made, one for reputation and the other for minimum computation time. After the first request is sent, you must receive a waiting list, but sometimes the waiting list is empty (null value). If neither requirement is met, the participant will not participate in MPC and will try sending the request again after a specific time. If only one condition is met, ask the edge node whether to continue. If the edge node originates from MPC with limited reception performance, it can participate in this calculation. Otherwise, it will quit and try cloud selection again after another period. If there are multiple servers to choose from, in this case, there will be two alternate lists, and you can get the best one by taking their intersection, or additional requirements are sent to those servers, which can provide higher performance. Then choose the best one. If multiple options still exist, select the one that is physically closest to you, with the lowest latency and highest efficiency from the data communication point of view.

After the server selection is complete, MPC computation is performed. After the MPC operation, the edge node takes the data back, executes it, and then makes a feedback report according to the satisfaction degree of delay. If the delay is satisfactory to the edge node, it will give positive feedback. Otherwise, it will be negative. The value of this evaluation will affect the reputation of a cloud server. The cloud server performs reputation ranking when the edge node reputation evaluation is completed.

The latest ranking and previous ranking of the cloud server and which cloud service did this calculation will be updated to the blockchain, which means that the change in the reputation ranking is not modifiable and can be verified jointly by anyone.

There are many types of servers in the cloud, and their performance varies. The performance of a cloud server determines whether the server can handle edge node requirements. Related to this, this framework provides an incentive mechanism between cloud servers and nodes participating in MPC. Cloud servers must provide honest and efficient services to have a higher probability of being selected by most users and generate revenue, while poorly performing cloud servers will be marginalized due to fewer selection opportunities. Users can get high efficiency by choosing a high-performance server and get better economic benefits.

Algorithm 3. Smart contract for reputation control

```
Input: Authenticated and verified data from the device
Process:
RecevieSmartContract();
ChooseBestServer();                    // Process in Algorithm2
MPC();
WaitForDataStaisfy();
    if (!Satisfy($C_{id}$))            // Compare real cost-time and Estimated cost-time
        SendFeedbackForCloud(-1);
    else
        SendFeedbackForCloud(1);
RemakeServerReputation($S_{id}$);      //S for Cloud Server
UpdateToBlock($C_{id}$);
```

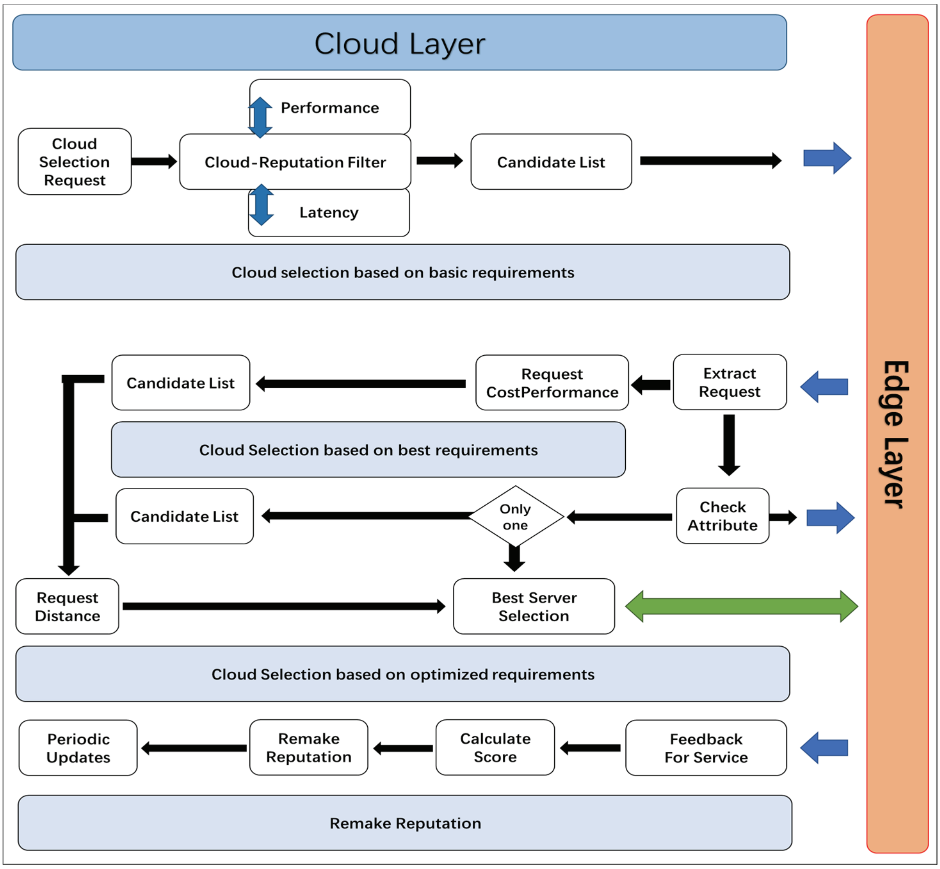

Fig. 4. Smart contract method for optimal cloud selection.

4. Evaluation and Performance

For our research, the most important thing is to evaluate the accuracy of analytical data and the difference in computation time with or without a smart contract. The second is to assess the delay gap caused by selecting the server with the closest physical distance after selecting a plurality of high-performance servers in the cloud selection stage.

4.1 Experiment Setup

We intend to use Hyperledger Fabric to build a blockchain network. Fabric is a framework that provides a modular distributed ledger solution with features such as confidentiality, scalability, flexibility, and extensibility. Fabric is also a blockchain project within the Hyperledger, containing a ledger, uses smart contracts, and is a system for managing transactions through all participants. Fabric allows the creation of a channel, allowing participants to create a separate ledger for a transaction. Only participants in the same track have the ledger in the channel, while other participants not in the channel cannot see the ledger. The channel's isolation technology brings higher security. It is also an essential Fabric feature.

Table 2. Information on Hyperledger Fabric attributes

Environment	Language	Version	Description	Nodes
Ubuntu 18.4, i7 processor, 32 GB RAM	Go, Java, JavaScript	1.1	Hyperledger Fabric 1.1 can optionally use this service to generate certificates and key materials to configure and manage identities in the Blockchain network	Edge 2, Cloud 5
	Go, Java, JavaScript, Node.js, Python	1.2	Fabric 1.2 upgrades to private data sets, service discovery, access control, and verifiable performance	Edge 2, Cloud 5
	Go, Java, JavaScript, Node.js, Python, C#, Ruby	1.3	Fabric 1.3 is a framework that provides a modular distributed ledger solution with independent, functionally different modules that can be adapted to various complex scenarios in an economic society.	Edge 2, Cloud 5

We analyze the proposed MPC framework using Ubuntu 18.04 and i7 processor with 32 GB RAM. As shown in Table 2, the information required for the experimental environment is given. We started with the initial version of Fabric and summarized the iterations and upgrades. Fabric 1.3 was adequate for our study; although Fabric 1.3 is not the latest version, it is still suitable for our research. Therefore, the Blockchain network on the edge layer is built utilizing the Hyperledger Fabric version 1.3. Many languages are supported for this release, such as Java, Go, Node.js, and Python, but we chose Python. Since Fabric 1.3 is only compatible with Python3.6, we use Python3.6 to create the smart contract. Virtual Cloud and Edge nodes are generated using VMware 14 with a set of 2 edge nodes (e_1, e_2), and 5 Cloud nodes $(c_1, c_2, c_3, c_4, c_5)$. c_1 and c_2 have a ranking reputation of 3, computing power of 1 GHz, and memory of 256 MB. c_3 and c_4 have a ranking reputation of 1, computing power of 600 MHz, and memory of 128 MB. c_5 has a reputation of 0 with 300 MHz computing power and a memory of 64 MB, indicating poor performance probability. A total of 10 tasks $(t_1, t_2, t_3 \dots, t_{10})$ of 15 MB each are measured that are selected as delay-intolerant tasks. The performance evaluation of task computation is based on the optimal selection of cloud operators using the baseline parameters set by the smart contract, which includes task completion accuracy, the proximity of the cloud node with the edge node, and the time required to complete the task.

4.2 Performance Evaluation

Each task is set with a delay tolerance of 3 ms to measure the effectiveness of the framework's support for computation accuracy. Of the two edge nodes in total, e_1, and e_2 have 10 tasks each. For comparison with the baseline model, we assume that e_1 implements the proposed MPC framework's optimal cloud node selection process. The second edge node e_1, e_2, selects cloud nodes based on a first-come-first-serve (FCFS) basis. Furthermore, at the edge layer, e_1 and e_2 are part of the blockchain network and maintain access control measures preventing unknown devices from uploading data for computation operations.

Fig. 5 illustrates the computation efficiency between e_1 and e_2 based on two different data computation operation methods selected. The blockchain at the edge node (e_1 and e_2) represents devices pre-authorized and validated to transmit computation requests at cloud nodes.

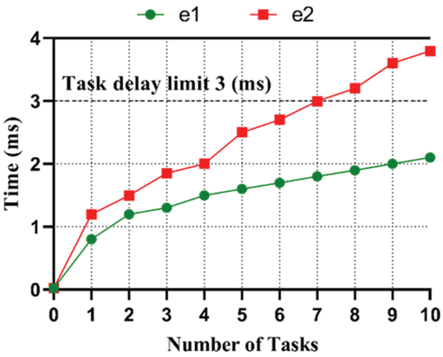

Fig. 5. Comparison of computation efficiency between the proposed method and FCFS method.

Both e_1 and e_2 selects cloud operators from nodes c_1, c_2, c_3, c_4,and c_5. We pre-assign each cloud node with three pre-existing computation tasks to increase the computation time required for future tasks. Additionally, cloud nodes c_1, c_3, c_4,and c_5 are placed in proximity to e_1. We observe that e_1 selects c_1, c_2,and c_3 due to their higher reputation ranking and lower time required to complete the assigned tasks. Although c_1 and c_2 are each running a single task, they have higher computing power than the remaining cloud nodes. The second edge node, e_2 selects c_3, c_4, and c_5 cloud nodes based on which the cloud node communicates with the edge node requesting the computation task. The overall time required by cloud nodes selected by e_1. They are much lower and complete all 10 tasks in less than 3 ms. The optimal selection of cloud nodes shows that the proposed framework makes high-performance computing of tasks possible. The baseline model using the FCFS method (e_2) took 1.7 ms more than the proposed framework, making it unsuitable for low delay-tolerant computing operations. Overall, efficiency increased by 44.73%. The minimum required for task completion is 3 ms, and Tasks 8–10 take more than 3 ms for task completion due to the unoptimized selection of cloud nodes using the FCFS model.

In Fig. 6, we observe the effect of selecting a cloud node c_1 by e_1 and compare it with the task computations performance of other nodes c_2 and c_3. Cloud nodes c_1 and c_2 share a similar reputation ranking of 3, the amount of memory, and the processing power to perform high-performance computations. In this test, we removed all previous running tasks from all three cloud nodes to accurately measure the task completion time. However, due to the proximity of node c_1 with edge node e_1, the network selects c_1 for computation operations. We observe that each task of 15 MB, between c_1, c_2, and c_3, c_1 and c_2 have nearly identical performances with total task completion times of 1.8 and 1.9 milliseconds (ms). Both nodes complete tasks more quickly due to their higher computation power and memory. However, the proximity of c_1 to e_1 makes it a better selection due to its reduced time to receive, complete the computation task and return results to the edge node e_1. An additional observation is the performance between c_1 and c_3, where there is a noticeable delay of 1 ms by c_3 to complete all tasks. The increased latency is despite both c_1 and c_3 being in proximity to e_1. c_3 has lower computation power of 600 Mhz and a memory of 128 MB, resulting in only holding a maximum of 8 tasks of 15 MB each. c_3 is required to wait for task completion to load the final two tasks in its memory.

The proposed Smart Contract based optimal selection of cloud nodes has superior computation performance for IIoT devices than the FCFS-based baseline model. Efficient computation of tasks is essential for IIoT networks. The Smart Contract based selection of reputation ranking and task completion time, along with the penalty incurred due to poor performance, ensures that only ideal clouds are selected in the task assignment process.

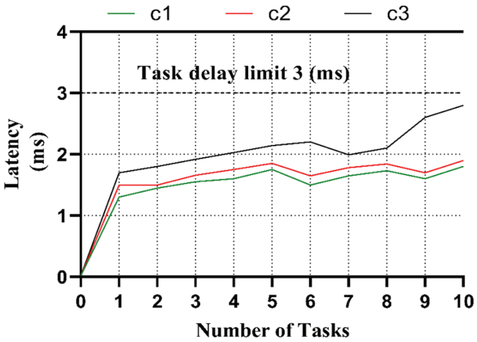

Fig. 6. Comparison of latency between the cloud nodes c_1, c_2, and c_3.

4.3 Performance Analysis

This section analyzes similar studies based on proposed requirements for efficient MPC. As shown in Table 2, we compare recent studies on cloud service selection and MPC, then discuss them regarding the efficient MPC considerations we provide in Section 2.1.

Kumar et al. [17] proposed a new framework named optimal service selection and ranking of cloud computing services (CCS-OSSR). Users who want to choose a cloud server evaluate and compare existing services based on QoS standards and choose a better one. This method has better consistency and fewer comparison times from a high-performance cloud server selection perspective. However, this framework does not address a cloud environment's continuous and frequently changing requirements and is, therefore, unsuitable for heterogeneous networks such as IIoT. The framework also ignores the verification of the results of dishonest cloud servers, each of which can try not to take responsibility for a problem by denying it. Jatoth et al. [18] proposed the multi-criteria decision-making model (MCDM), in which QoS parameters are first quantified, and different QoS-based hierarchies are allocated to cloud servers considering user preferences. This method effectively resolves the uncertainty of service selection while maintaining the efficiency and scalability of cloud services. While this model performs well in terms of efficiency, it also does not consider the non-repudiation required in computing like MPC. Kumar et al. [19] also used AHP and TOPSIS to determine the most suitable candidate cloud services. Under this framework, cloud selection data is processed with high efficiency and data correctness is ensured through methods that verify its accuracy. As a result, the server cannot deny its calculation results. Sun et al. [20] broke new ground and questioned the independent evaluation criteria for cloud service selection technology. A framework of cloud service selection with criteria interactions (CSSCI) is proposed to solve the service selection problem without historical information to determine the standard relationships and weights. Their research can always choose the best server in the ranking, and more realistic indicators are discussed. We hold reservations about the expansibility of this method. The more interactions between standards are considered, the more communication overhead will exist in the complex and heterogeneous IIoT environment. Therefore, with the addition of heterogeneous data, the efficiency of the network may be reduced. In summary, the above literature discusses cloud computing efficiency but ignores non-repudiation.

Aside from discussing the studies of cloud selection, it is also valuable to discuss research on MPC that guarantees non-repudiation. Several studies [21-23] are a collection of similar studies that propose a secure MPC framework and use blockchain technology to assist or support MPC. However, they also have a similar disadvantage: their efficiency is limited. An MPC technology based on the Hyperledger Fabric framework is created in [21], which can protect the private data of all participating users, but the efficiency of information processing is insufficient. A study [22] proposed a reputed mutual incentive blockchain-based MPC scheme (BFR-MPC), which has the characteristics of high fairness and stability but cannot adapt to a wide range of efficient deployment requirements. Guan et al. [23] proposed a blockchain-based dual-side privacy-preserving multiparty computation scheme for an edge-enabled smart grid (BPM4SG). This approach uses consortium blockchains and smart contracts to further enhance the system's security and avoid dependence on trusted third parties. In addition, a data obfuscation method based on ring signatures and a new one-time address scheme is proposed to protect the data owner and receiver's privacy. Furthermore, the above three types of literature use the characteristics of blockchain to emphasize the trustworthiness and non-repudiation of the MPC process, but they do not consider efficiency and scalability in their frameworks. Still, their frameworks do not take efficiency and scalability into consideration.

Table 3. Key consideration comparisons with other related works

Study	Mechanism	Efficiency	Scalability	Availability	Non-repudiation
Kumar et al. [17]	CCS-OSSR	○	×	○	×
Jatoth et al. [18]	Hybrid MCDM	○	○	○	×
Kumar et al. [19]	Hybrid MCDM	○	○	○	△
Sun et al. [20]	CSSCI	○	△	○	×
Benhamouda et al. [21]	Hyperledger fabric with MPC	×	×	△	○
Gao et al. [22]	BFR-MPC	×	△	○	○
Guan et al. [23]	BPM4SG	△	×	○	○
Our work	Smart contract-based optimal cloud selection	○	○	○	○

○=exist, △=limited, ×=not-exist.

By comparing the results of all the works in Table 3, it is clear that non-repudiation is almost ignored by all the research schemes considering efficient cloud selection in recent years. At the same time, the latest research on MPC often focuses on security and trust. In contrast, research on Blockchain as a solution often lacks the consideration of efficiency and expansibility. The proposed cloud selection framework ensures computing efficiency by selecting the best cloud server and communication efficiency by physical distance. Meanwhile, all nodes certified by blockchain are valid data providers, and their interactions with cloud nodes will be recorded in blockchain to ensure that no one can tamper with the results. All cloud nodes will not be able to deny their behavior. At the same time, no matter how many edge nodes try to join the MPC, they have different needs for different tasks. They can allocate resources according to their needs, reducing the pressure on a single cloud node and thus providing scalability in the entire cloud environment. Blockchain and cloud environments are distributed networks characterized by high availability. Meanwhile, the characteristics of MPC also include the tolerance of specific malicious nodes. Therefore, the overall availability of this framework is beyond doubt. Of course, there are weaknesses in our proposed model, which we will discuss in the next section.

4.4 Discussion

This section discusses open research challenges. This paper proposes an efficient cloud selection framework based on smart contract, which reasonably solves the heterogeneous computation task requirements in the IIoT environment. Each user chooses the best cloud server to meet their needs according to their business requirements, thus ensuring overall computing efficiency. Meanwhile, based on the blockchain configured in the edge layer, all nodes attempting to participate in MPC must be authenticated and confirm their information sources, thus ensuring information validity for each edge node participating in cloud MPC. For MPC, the fact that each input is validated increases the accuracy and computational efficiency of the whole process because the probability of malicious information is lower. In addition, according to the smart contract, the return delay of MPC calculation results can be verified, thus reducing the probability of malicious cloud nodes tampering with data and the problem of insecure clouds being selected. With the operation of smart contracts, incentives constantly change the reputation ranking of cloud nodes, and malicious edge nodes will be gradually marginalized. From the whole process, the cloud node and edge node motivate each other; the edge node can give the cloud node a higher reputation value to increase its frequency of use. The cloud node feeds back the edge node with efficient and correct calculation results. Supposing fewer and fewer users select a malicious cloud node, its maintenance cost will be progressively higher, thus maintaining the overall stability of the network. At the same time, high-performance and high-reputation cloud nodes are increasingly selected, and their revenue will also increase.

Our research also found weaknesses in the proposed framework. Several open research challenges for future studies have been previously discussed. Our proposed framework is still not secure enough. Attackers can steal data from the edge layer or cloud. Although the mechanism of MPC itself is secure, there is a risk of data theft before and after MPC participation. This means that various challenges in actual applications still face the framework. Future research will address this issue to ensure efficient cloud-secure multiparty computing in IIoT.

5. Conclusion

This paper proposes a network architecture based on the blockchain and smart contract to provide comprehensive performance assurance for the IIoT network. Blockchain-based authentication and identification ensure data validity for every user participating in MPC. The smart contract-based cloud server selection strategy ensures that each participating node receives the best service that suits its requirements or exceeds expectations, with lower latency. The cloud server gets a reputation ranking for computing performance based on aggregated feedback from all users it serves. The cloud nodes' computing records and reputation ranking values are recorded in the blockchain. This ensures non-repudiation and public witnessing of cloud nodes' computing results and reputation rankings. The reputation level improves the enthusiasm of other users in the network to choose the server, forming a benign incentive mechanism to promote the interaction between nodes and the server. Deductive results show that the framework we provided in this study has excellent performance and low latency. Compared with the baseline method, the efficiency improved by 44.73%. We plan to design a more comprehensive and secure MPC algorithm to fit this efficient framework in future work.

Acknowledgement

This study was supported by the Research Program funded by SeoulTech (Seoul National University of Science and Technology).

References

[1] Alliance Virtual Offices, “Working from home increases cyberattack frequency by 238%, new study by alliance virtual offices finds,” 2022 [Online]. Available: https://www.globenewswire.com/en/news-release/2022/03/15/2403837/0/en/Working-From-Home-Increases-Cyberattack-Frequency-by-238-New-Study-by-Alliance-Virtual-Offices-Finds.html.

[2] A. El Azzaoui, M. Y. Choi, C. H. Lee, and J. H. Park, “Scalable lightweight blockchain-based authentication mechanism for secure VoIP communication,” *Human-centric Computing and Information Sciences*, vol. 12, article no. 8, 2022. https://doi.org/10.22967/HCIS.2022.12.008

[3] J. S. Park and J. H. Park, “Future trends of IoT, 5G mobile networks, and AI: challenges, opportunities, and solutions,” *Journal of Information Processing Systems*, vol. 16, no. 4, pp. 743-749, 2020.

[4] I. Sitton-Candanedo, R. S. Alonso, O. Garcia, L. Munoz, and S. Rodriguez-Gonzalez, “Edge computing, IoT and social computing in smart energy scenarios,” *Sensors*, vol. 19, no. 15, article no. 3353, 2019. https://doi.org/10.3390/s19153353

[5] M. M. Salim, I. Kim, U. Doniyor, C. Lee, and J. H. Park, “Homomorphic encryption based privacy-preservation for IoMT,” *Applied Sciences*, vol. 11, no. 18, article no. 8757, 2021. https://doi.org/10.3390/app11188757

[6] R. J. Hassan, S. R. Zeebaree, S. Y. Ameen, S. F. Kak, M. A. Sadeeq, Z. S. Ageed, A. Al-Zebari, and A. A. Salih, “State of art survey for IoT effects on smart city technology: challenges, opportunities, and solutions,” *Asian Journal of Research in Computer Science*, vol. 8, no. 3, pp. 32-48, 2021.

[7] S. K. Singh, A. E. Azzaoui, T. W. Kim, Y. Pan, and J. H. Park, “DeepBlockScheme: a deep learning-based blockchain driven scheme for secure smart city,” *Human-centric Computing and Information Sciences*, vol. 11, article no. 12, 2021. https://doi.org/10.22967/HCIS.2021.11.012

[8] N. Volgushev, M. Schwarzkopf, B. Getchell, M. Varia, A. Lapets, and A. Bestavros, “Conclave: secure multi-party computation on big data,” in *Proceedings of the 14th EuroSys Conference 2019*, Dresden, Germany, 2019, pp. 1-18.

[9] H. Boyes, B. Hallaq, J. Cunningham, and T. Watson, “The industrial internet of things (IIoT): an analysis framework,” *Computers in Industry*, vol. 101, pp. 1-12, 2018.

[10] J. Cheng, W. Chen, F. Tao, and C. L. Lin, “Industrial IoT in 5G environment towards smart manufacturing,” *Journal of Industrial Information Integration*, vol. 10, pp. 10-19, 2018.

[11] Z. Zhou, X. Chen, Y. Zhang, and S. Mumtaz, “Blockchain-empowered secure spectrum sharing for 5G heterogeneous networks,” *IEEE Network*, vol. 34, no. 1, pp. 24-31, 2020.

[12] Y. Tian, T. Li, J. Xiong, M. Z. A. Bhuiyan, J. Ma, and C. Peng, “A blockchain-based machine learning framework for edge services in IIoT,” *IEEE Transactions on Industrial Informatics*, vol. 18, no. 3, pp. 1918-1929, 2022.

[13] N. Bugshan, I. Khalil, N. Moustafa, and M. S. Rahman, “Privacy-preserving microservices in industrial Internet of Things driven smart applications,” *IEEE Internet of Things Journal*, vol. 10, no. 4, pp. 2821-2831, 2023.

[14] A. Kumari, S. Tanwar, S. Tyagi, and N. Kumar, “Blockchain-based massive data dissemination handling in IIoT environment,” *IEEE Network*, vol. 35, no. 1, pp. 318-325, 2021.

[15] S. Li, S. Zhao, G. Min, L. Qi, and G. Liu, “Lightweight privacy-preserving scheme using homomorphic encryption in industrial Internet of Things,” *IEEE Internet of Things Journal*, vol. 9, no. 16, pp. 14542-14550, 2022.

[16] Y. Wang, D. K. Kim, and D. Jeong, “A survey of the application of blockchain in multiple fields of financial services,” *Journal of Information Processing Systems*, vol. 16, no. 4, pp. 935-958, 2020.

[17] R. R. Kumar, B. Kumari, and C. Kumar, “CCS-OSSR: a framework based on hybrid MCDM for optimal service selection and ranking of cloud computing services” *Cluster Computing*, vol. 24, no. 2, pp. 867-883, 2021.

[18] C. Jatoth, G. R. Gangadharan, U. Fiore, and R. Buyya, “SELCLOUD: a hybrid multi-criteria decision-making model for selection of cloud services,” *Soft Computing*, vol. 23, pp. 4701-4715, 2019.

[19] R. R. Kumar, S. Mishra, and C. Kumar, “A novel framework for cloud service evaluation and selection using hybrid MCDM methods,” *Arabian Journal for Science and Engineering*, vol. 43, pp. 7015-7030, 2018.

[20] L. Sun, H. Dong, O. K. Hussain, F. K. Hussain, and A. X. Liu, “A framework of cloud service selection with criteria interactions,” *Future Generation Computer Systems*, vol. 94, pp. 749-764, 2019.

[21] F. Benhamouda, S. Halevi, and T. Halevi, “Supporting private data on Hyperledger fabric with secure multiparty computation,” *IBM Journal of Research and Development*, vol. 63, no. 2/3, article no. 3, 2019. https://doi.org/10.1147/JRD.2019.2913621

[22] H. Gao, Z. Ma, S. Luo, and Z. Wang, “BFR-MPC: a blockchain-based fair and robust multi-party computation scheme,” *IEEE Access*, vol. 7, pp. 110439-110450, 2019.

[23] Z. Guan, X. Zhou, P. Liu, L. Wu, and W. Yang, “A blockchain-based dual-side privacy-preserving multiparty computation scheme for edge-enabled sm

Haotian Chen **https://orcid.org/0000-0002-6056-1497**

He received a B.S. degree in computer science and engineering from the Seoul National University of Science and Technology, Seoul, South Korea. He is currently pursuing the M.S. combined PhD in computer science and engineering with the Ubiquitous Computing Security (UCS) Laboratory from the Seoul National University of Science and Technology, Seoul, South Korea, under the supervision of Prof. Jong Hyuk Park. His research interests include applied quantum information, focusing on Blockchain and Internet of Things security.

Abir EL Azzaoui **https://orcid.org/0000-0002-9406-8932**

She received the M.S. degree from the Seoul National University of Science and Technology, Seoul, South Korea, and a B.S. degree in computer science from the University of Picardie Jules Verne, Amiens, France. She graduated from the National School of Higher Education Hassan II in the Development of Information Systems, Marrakech, Morocco. She is currently pursuing a PhD in computer science and engineering with the Ubiquitous Computing Security (UCS) Laboratory at Seoul National University of Science and Technology, Seoul, South Korea, under the supervision of Prof. Jong Hyuk Park. Her current research interests include Blockchain, Internet-of-Things (IoT) security, and post-quantum cryptography. She is also a reviewer of IEEE Access. She has received the Quarterly Franklin Membership from the London Journal of Engineering Research (LJER), London, UK.

Sekione Reward Jeremiah **https://orcid.org/0000-0002-9396-2270**

He received MSc. degree in Computer Science from the University of Dar es salaam (Tanzania) and a BSc. ICT from Mzumbe University in 2018 and 2015, respectively. From 2018 to 2019, he was an Assistant lecturer with the Department of Informatics at MJNUAT. He is currently a Doctoral student in Electrical and Information Engineering at SeoulTech. His research interests include reinforcement learning, IoT security, Green IoV, Digital Twins, unmanned aerial vehicles, and wireless networks.

Jong Hyuk (James J.) Park **https://orcid.org/0000-0003-1831-0309**

He received Ph.D. degrees in the Graduate School of Information Security from Korea University, Korea. He is a professor at the Department of Computer Science and Engineering and the Department of Interdisciplinary Bio I.T. Materials, Seoul National University of Science and Technology (SeoulTech), Korea. He has published about 300 research papers in international journals and conferences. His research interests include the IoT, human-centric ubiquitous computing, information security, digital forensics, vehicular cloud computing, and multimedia computing. He is a member of the IEEE Computer Society, KIPS, and KMMS. He got the best paper awards from ISA-2008 and ITCS-2011 conferences and outstanding leadership awards from IEEE HPCC-2009, ICA3PP-2010, IEE ISPA-2011, PDCAT-2011, and IEEE AINA-2015. Furthermore, he received outstanding research awards from SeoulTech, in 2014. He has been serving as the Chair, the Program Committee, or the Organizing Committee Chair for many international conferences and workshops. He is also the Steering Chair of international conferences–MUE, FutureTech, CSA, CUTE, UCAWSN, and World IT Congress-Jeju. He is Editor-in-Chief of Human-centric Computing and Information Sciences (HCIS) by KIPS, The Journal of Information Processing Systems (JIPS) by KIPS, and the Journal of Convergence (JoC) by KIPS CSWRG. In addition, he has been serving as a Guest Editor for international journals by some publishers: Springer, Elsevier, John Wiley, Oxford University Press, Emerald, Inderscience, and MDPI.