IEICE TRANSACTIONS

on Communications

DOI:10.23919/transcom.2023EBP3156

A PUBLICATION OF THE COMMUNICATIONS SOCIETY

The Institute of Electronics, Information and Communication Engineers
Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3chome, Minato-ku, TOKYO, 105-0011 JAPAN

PAPER

A Distributed Efficient Blockchain Oracle Scheme for Internet of Things*

Youquan XIAN[†,††a)], Lianghaojie ZHOU[†,††b)], Jianyong JIANG[†,††c)], Boyi WANG[†,††d)], Hao HUO[†,††e)], *and* Peng LIU[†,††f)], *Nonmembers*

SUMMARY In recent years, blockchain has been widely applied in the Internet of Things (IoT). Blockchain oracle, as a bridge for data communication between blockchain and off-chain, has also received significant attention. However, the numerous and heterogeneous devices in the IoT pose great challenges to the efficiency and security of data acquisition for oracles. We find that the matching relationship between data sources and oracle nodes greatly affects the efficiency and service quality of the entire oracle system. To address these issues, this paper proposes a distributed and efficient oracle solution tailored for the IoT, enabling fast acquisition of real-time off-chain data. Specifically, we first design a distributed oracle architecture that combines both Trusted Execution Environment (TEE) devices and ordinary devices to improve system scalability, considering the heterogeneity of IoT devices. Secondly, based on the trusted node information provided by TEE, we determine the matching relationship between nodes and data sources, assigning appropriate nodes for tasks to enhance system efficiency. Through simulation experiments, our proposed solution has been shown to effectively improve the efficiency and service quality of the system, reducing the average response time by approximately 9.92% compared to conventional approaches.

key words: *blockchain, oracle, Internet of Things(IoT), Trusted Execution Environment(TEE), threshold signature*

1. Introduction

With the rapid development of the Internet of Things, the traditional centralized network structure faces many challenges, such as vulnerability to hacker attacks and paralysis caused by central failures. Blockchain, as a decentralized distributed network ledger technology, has advantages such as security, transparency, and traceability, which can well meet the needs of the Internet of Things [1].

Blockchain can provide a secure and reliable underlying infrastructure for the Internet of Things. By using blockchain technology, the authenticity and uniqueness of IoT devices and data can be ensured, and data tampering can be prevented [2]. However, blockchains and smart contracts have no access to the external systems (i.e., off-chain), nor use IoT devices to perform some computing tasks. Therefore, blockchain oracle is proposed to solve the above problems [3]–[5].

TEE is a security solution based on trusted hardware, which guarantees the security of the runtime environment by trusted hardware [6]. Town Crier [7] is the first oracle scheme to combine Intel SGX with Ethereum smart contracts. Smart contracts can perform operations such as authentication and data acquisition processing in TEE, but Town Crier has a clear risk of single point of failure. Based on Intel SGX, Woo et al. [8] proposed a decentralized oracle scheme for IoT applications. They are mainly used to collect data from IoT devices and improve the availability of oracle networks in the event of a single point of failure or a malicious node through the Byzantine Fault Tolerance (BFT) consensus mechanism. However, a key problem is that TEE devices are still relatively scarce in the Internet of Things, and a small number of devices are difficult to support the heavy computing and data acquisition tasks in the Internet of Things [6]. In addition, we also analyze the aggregation strategy of threshold signature as an example. The conclusion shows that in a certain time range, the more nodes involved in obtaining data, the greater the probability of reaching a consensus. Therefore, it is necessary to improve the scalability of the oracle while ensuring security.

On the other hand, from Bitcoin to Ethereum to the proposal of sharding blockchain, researchers are gradually improving the response speed of blockchain [9]. Therefore, reducing the response time of oracle is not only to meet some delay-sensitive blockchain applications, such as obtaining real-time sensor data, but also to match the response speed of the blockchain. In particular, Wang et al. [10] and Zhang et al. [11] both consider the node communication delay in the sharding process to improve the efficiency of the blockchain. Moreover, we also find that the response time of the data obtained by the oracle is related to the selection of the oracle node, and there is an implicit connection between the oracle node and the data source (Section. 3.3.2).

[†]The author is with the College of Computer Science and Information Technology, Guangxi Normal University, Guangxi, China.

[††]The author is with the Key Lab of Education Blockchain and Intelligent Technology, Ministry of Education, Guangxi Normal University, Guangxi, China.

[*]The research was supported in part by the Guangxi Science and Technology Major Project (No. AA22068070), the National Natural Science Foundation of China (Nos. 62166004,U21A20474), the Key Lab of Education Blockchain and Intelligent Technology, the Center for Applied Mathematics of Guangxi, the Guangxi "Bagui Scholar" Teams for Innovation and Research Project, the Guangxi Talent Highland Project of Big Data Intelligence and Application, the Guangxi Collaborative Center of Multisource Information Integration and Intelligent Processing.

a) E-mail: xianyouquan@stu.gxnu.edu.cn
b) E-mail: zhoulianghaojie@stu.gxnu.edu.cn
c) E-mail: jiangyong@stu.gxnu.edu.cn
d) E-mail: wangboyi@stu.gxnu.edu.cn
e) E-mail: huohao@stu.gxnu.edu.cn
f) E-mail: liupeng@gxnu.edu.cn(Corresponding author)

Although there are many studies to improve the security and efficiency of the oracle. For example, based on the game theory to constrain the behavior of the oracle nodes [12], [13], or through threshold signature and other cryptographic methods to ensure the credibility of the obtained data and encourage users to respond quickly [14], [15]. However, how to improve the scalability while ensuring the security and credibility of the oracle system, and according to the inherent relationship between the node and the data source, select the appropriate node for the task to improve the efficiency of the oracle, is still an urgent problem to be solved.

In this paper, we design a distributed and efficient blockchain oracle solution tailored for IoT scenarios, enabling the blockchain to acquire complex data or execute computational tasks from the IoT. Firstly, we design a novel distributed blockchain oracle architecture that incorporates hybrid TEE devices. By categorizing oracle nodes into ordinary nodes and TEE nodes, we ensure system security and trustworthiness while improving system scalability. Secondly, we define a new Quality of Service (QoS) standard for oracles to quantify the matching relationship between oracle nodes and data sources. We employ a weighted random algorithm to select more suitable nodes for each task, enhancing the execution efficiency of every task.

The contributions of this paper are summarized as follows :

- In order to improve the scalability of the oracle system while ensuring security and credibility, we designed a distributed multi-data source oracle architecture that includes TEE and ordinary devices. It uses a committee composed of TEE to improve the credibility and aggregation efficiency of off-chain data, and combines ordinary nodes to increase the scalability of the system.
- Since the selection of oracle nodes will affect the efficiency of data collection, we redesign a new Quality of Service (QoS) metric. It considers the response time and result accuracy of nodes obtaining data from the data source. Then, we use an efficient weighted random node selection algorithm to select task nodes in a non-replacement weighted extraction way, which fully improves the service efficiency of the system.
- Through experimental verification, our scheme can effectively improve the scalability and efficiency of the system. Compared with the ordinary scheme, the response time of each task is reduced by about 9.92%.

The rest of this paper is organized as follows. Section 2 introduces some work related to this study. Section 3 introduces oracle, threshold signature, and current problem. Section 4 introduces our proposed oracle framework. Section 5 shows the simulation results and analysis. Section 6 gives the conclusion of this paper.

2. Related Work

The purpose of the blockchain oracle is to help the blockchain obtain reliable off-chain data [4]. The data received by the oracle nodes may be inconsistent, and the oracle needs to ensure the authenticity and integrity of the acquired data. We divide the literature review into three parts: voting-based oracles, hardware-based oracles, and distributed oracles.

2.1 Voting-based Oracles

Augur is the first prediction market platform for decentralized oracle proposed by Ethereum. It hands data request events to all participants in the market. If the data provided by one node is inconsistent with the consensus results of other nodes, the pledged tokens will be confiscated by the system and redistributed to other nodes, forcing the nodes to remain honest [12]. Adler et al. [13] designed a decentralised oracle scheme based on a voting game that divides participants into three categories: submitters, voters or verifiers. Voters and validators play games with each other. Eventually, the Nash equilibrium is reached where all rational participants are forced to act honestly. Cai et al. [16] thought that the linear growth of participants' equity may lead to herd behavior, so they propose a nonlinear equity scaling rule to prevent Sybil attacks. However, the voting-based oracle requires a definite answer to the proposition, and the user's voting is inefficient.

2.2 Hardware-based Oracles

Trusted Execution Environment (TEE) can protect the core code of the program from interference by other malicious programs, while avoiding data leakage [6]. Zhang et al. [7] proposed a hardware-based oracle system, which uses Intel SGX to ensure the security and credibility of computing. However, the above system has the possibility of a single point of failure, so Woo et al. [8], [17] proposed a distributed oracle using Intel SGX to improve the availability of the oracle network in the event of a single point of failure or malicious nodes through the BFT consensus mechanism. Liu et al. [18] using TEE to extend blockchain trust from the blockchain to the off-chain, continuously perceive the inside of the vaccine box through trusted sensors and generate anti-counterfeiting data to obtain reliable vaccine tracking data. However, hardware-based oracles require specific hardware support, so their scalability is poor and it is difficult to take advantage of the large number of IoT devices without TEE.

2.3 Distributed Oracles

DECO is a distributed oracle scheme that helps users prove that the data they access through TLS is from a specific website without the need for trusted hardware or server-side modifications [19]. He et al. [20] proposed a novel scalable data feed service SDFS, which ensures the rationality of the data acquisition process and the scalability of the system services. In addition, a node reputation evaluation strategy is designed to judge whether each node is a malicious node and update the reputation value of each node. Chainlink [14] proposes a decentralized oracle network to achieve reliable data input and connection between smart contracts and external data sources. It uses reputation systems and cryptographic algorithms to ensure reliable operation of the system. DOS Network [15] provides off-chain computing in a decentralized manner, and uses m-out-n multi-signature transactions to achieve consensus. Similarly, Manoj et al. [21] obtained trusted agricultural data for the blockchain using a threshold secret-sharing scheme. Lin et al. [22] proposed a novel IIoT-oriented digital twin architecture by combining oracle with decentralized learning. Gigli et al. [23] proposed a robust oracle architecture with a reputation algorithm for selecting trusted data sources. Even if there are multiple malicious data sources, it can consistently provide high-quality data.

The above solutions are all focused on improving the accuracy, reliability, and completeness of oracle data. However, these solutions do not consider the impact of the matching relationship between oracle nodes and data sources on the performance of the oracle system. Therefore, this paper designs a high-performance distributed multi-data-source oracle system with a hybrid TEE device for IoT scenarios. By combining the advantages of TEE and distributed oracle systems, the system selects efficient and trustworthy oracle nodes through weighted random selection, thereby improving the access efficiency of the oracle system.

3. Preliminaries and Current Problems

3.1 Oracle

The word 'Oracle' comes from Greek mythology and refers to people who can directly communicate with God and foresee the future. In a blockchain environment, oracle is a system that provides blockchain with information from the real world [24].

The main steps are shown in Fig. 1, 1) The user contract calls the oracle contract to obtain the data of the corresponding data source; 2) After the oracle node listens to the event of the oracle contract, it obtains data from the corresponding data source; 3) The data source returns the corresponding data; 4) Oracle nodes verify and sign the data and return the oracle contract; 5) The oracle contract returns data to the user contract.

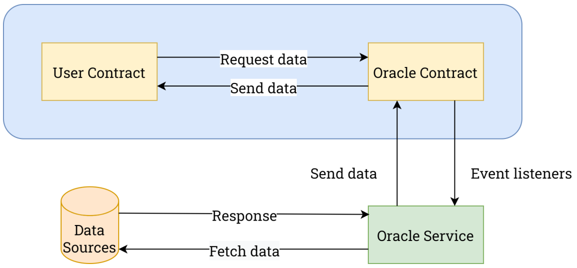


Fig. 1 Overview of oracle system.

3.2 Threshold Signature

Threshold signature is a group of n members, the group has a pair of group signature public key and private key [25]. The group's signature private key is shared by n members in a threshold manner. Only the number of honest members in the group greater than or equal to the threshold value t can sign on behalf of the group with the group signature private key. And it is easy for anyone to verify the signature through the group public key. These characteristics of threshold signature make it can be used in the process of oracle node aggregation data to prevent the problem of freeloading [14]. The process of threshold signature used in this paper is as follows:

1) Generate public key and group member private key: input the security parameter s, the threshold value t and the number of members n, and return the private key $sk_i(i = 1, 2, ..., n)$ for each member's partial signature. Each member saves a private key fragment to generate a partial threshold signature. pk is used to verify the threshold signature, as shown in the equation (1), where $L = sk_1, sk_2, ..., sk_n$.

$$pk, L = Generate(s, t, n) \tag{1}$$

2) Generate partial threshold signature: input the message msg and the member private key sk_i to generate a partial signature $Sign_i$, as shown in the equation (2):

$$Sign_i = Sign(msg, sk_i) \tag{2}$$

3) Synthetic threshold signature: if the valid partial signatures $Sign_i$ collected by a member reach the threshold value t, these $Sign_i$ can be used to synthesize the threshold signature $Sign^*$.

4) Verify threshold signature: input message msg, public key pk and threshold signature $Sign^*$, return judgment result $true$ and $false$, where $true$ means authentication passed and $false$ means not authenticated, as shown in equation (3):

$$Verify(msg, pk, Sign^*) == true \tag{3}$$

3.3 Analysis of Existing Problems

3.3.1 Impact of number of nodes on aggregation success

Facing the real-time data source in the Internet of Things, how to improve the probability of meeting the threshold t in a certain time range is the core issue that we need to pay attention to. In this regard, we make a detailed analysis in Appendix A. The conclusion shows that: 1) In order to get m identical request results returned, the larger the number of selected nodes n, the higher the probability of reaching the threshold m; 2) The smaller the variation period T of the data source, the more nodes n are needed.

3.3.2 The problem of matching data sources to oracles nodes

In order to improve the robustness and security of the system, the current oracle scheme uses multiple oracle nodes and multiple data source nodes to reduce the impact of single point of failure and malicious [23]. The user smart contract specifies multiple data sources to obtain data, and matches different oracle nodes to perform tasks. However, there is an implicit matching relationship between the oracle node and the data source, and the network connectivity will greatly affect the speed of data acquisition. For example, due to network and geographical differences, the response time of the German oracle node A to obtain the data source D_1 in Germany is often faster than the data source D_2 in China. For example, the network connectivity will greatly affect the speed of data acquisition. As shown in Fig. 2, region A and B are distributed with some data sources and oracle nodes, and cross-domain requests will experience a longer network routing wait. Therefore, how to select appropriate nodes to participate in the task according to the implied matching relationship between data sources and oracle nodes is an effective means to improve the efficiency of the system.

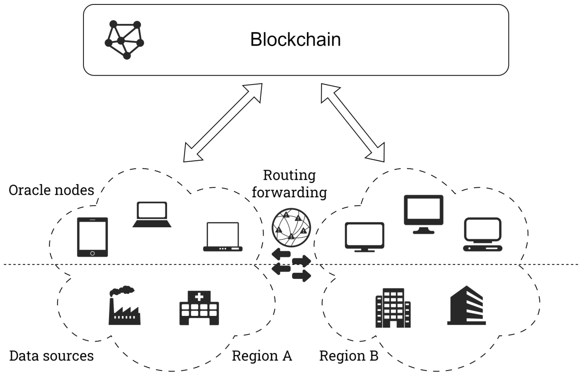


Fig. 2 The process of obtaining data from data sources in regions A and B.

4. System Design

4.1 Overview

In this Section, we present the oracle scheme proposed in this paper. We design a high-performance distributed oracle weighted matching scheme with multiple data sources. The system is mainly divided into on-chain contracts and off-chain nodes, where off-chain nodes are further divided into ordinary oracle nodes and TEE nodes. Ordinary oracles account for the majority of industrial Internet devices, providing scalability for the oracle system. A few nodes with TEE can provide trusted support for the system and improve system trustworthiness. The system overview is shown in Fig. 3.

The main steps include the following:

① Firstly, the user contract initiates a data acquisition request to the oracle contract.

② The oracle contract splits multiple data source requests into subtasks of a single data source.

③ The oracle contract takes the QoS of each oracle node to the requested data source as the weight of node selection, and uses the A-EepJ weighted random sampling algorithm to select the set of ordinary oracle nodes to participate in subsequent tasks.

④ The off-chain oracle node uses a one-time non-interactive Distributed Key Generation (DKG) protocol to generate a distributed public and private key after listening to the task.

⑤ According to the user task, the oracle node off-chain obtains data from the specified data source and signs it with its own private key fragment sk_i to obtain the signature fragment $Sign_i$.

⑥ The TEE node is responsible for jointly verifying the signature fragment $Sign_i$ uploaded by the ordinary node, recording its response time t_d^i, and finally returning the Leader, which is aggregated by the Leader to obtain the group signature $Sign^*$

⑦ After aggregating the group signature, the Leader of the TEE node committee packages the group signature $Sign^*$, the original data and the node response time t_d^i, and then signs and uploads the oracle contract.

⑧ After the signature of the oracle contract verification is correct, the original data is obtained, and the behavior information of the oracle node and the corresponding QoS record are updated.

⑨ The final aggregation result is aggregated by oracle contract according to the user-defined multi-data source result aggregation strategy.

⑩ Finally, the oracle contract returns the final result to the user contract.

4.2 Node Selection

Firstly, the oracle contract processes data requests from user contracts in the form of a four-tuple $Request(m, n, D, r)$, where m represents the number of nodes required for aggregate signature, n represents the total number of nodes required for this task, D represents the array of data sources, and r represents the random number generated by the hash of the previous block. The oracle contract needs to split the multi-data source request task into multiple subtasks $Request(m, n, d, r)$ and record, $d \in D$, d represents a single data source.

$$Request(m, n, d, r) \leftarrow Request(m, n, D, r) \tag{4}$$

For a single subtask $Request(m, n, d, r)$, the oracle contract needs to select n ordinary oracle nodes to obtain data d according to the requirements of the subtask. To select efficient

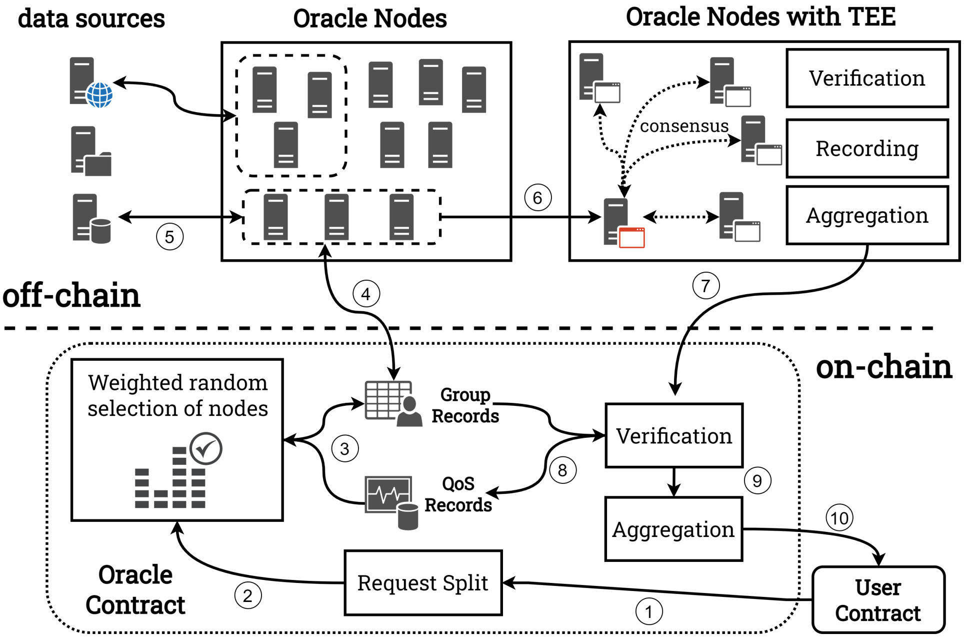


Fig. 3 Overview of our oracle system.

nodes while ensuring randomness, we use the oracle node as a weight for the data source d service quality QoS_d, and randomly select nodes. In order to reduce the space and time complexity of the oracle contract, we use the A-ExpJ algorithm [26] to randomly select n nodes from N to participate in the data acquisition task, N represents the set of all ordinary oracle nodes. For each node i, a random number $u_i = rand(0, 1)$ that obeys uniform distribution is selected, $k_i = u_i^{1/w_i}$ is used as the key value of sampling, and m samples with the largest key value are selected as the sampling results. This method effectively reduces the time and space complexity of sampling compared with the cardinal method [27]. In addition, the non-replacement selection can reduce the repeated selection of some oracle nodes and reduce the response time.

$$R = AExpJ(N, QoS_d, n) \tag{5}$$

R denotes the set of n ordinary oracle nodes selected from N by the A-ExpJ algorithm. We give a detailed description in Algorithm 1.

Algorithm 1: Weight node selection (AExpJ)

```
Input: Set $V$, Weight of set elements $W$, Number of samples $m$
Output: Selected Node Collection $R$
The first $m$ items of $V$ are inserted into $R$
for $w_i \in R$ do
    $k_i = u_i^{1/w_i}$
    $u_i = random(0, 1)$
end
Threshold $T_w$ is the minimum weight of $R$
while $v_c \in V$ and $v_c$ is not been visited before do
    $r = random(0, 1)$
    $X_w = \frac{log(r)}{log(T_w)}$
    From the current item $v_c$ skip items until item $v_i$, such that:
    $w_c + w_{c+1} + \ldots + w_{i-1} < X_w \leq w_c + w_{c+1} + \ldots + w_{i-1} + w_i$
    The element in $R$ with the minimum weight is replaced by
      item $v_i$
    $t_w = T_w^{w_i}$
    $r_2 = random(t_w, 1)$
    $k_i = (r_2)^{1/w_i}$
    Threshold $T_w$ is the minimum weight of $R$
end
```

4.3 Data Request and Aggregation

The oracle nodes are divided into two categories according to whether they have TEE: ordinary nodes and TEE nodes. TEE nodes together form a consensus committee to record, verify and aggregate data signatures. The ordinary oracle node generates a distributed public and private key based on the contract task, performs computational logic such as obtaining data or performing computational tasks, and signs the result.

Firstly, the selected oracle node set R generates n private key fragments sk_i and public key pk with a threshold of m according to the $Request(m, n, d, r)$ one-time non-interactive distributed key generation (DKG) protocol [28] with a threshold of m (It is worth noting that this phase can be paralleled with the data acquisition process).

$$pk, sk_i = Generate(r, m, n) \tag{6}$$

Then, the node i obtains data from the data source d, and uses its own private key fragment sk_i to sign the obtained data msg to obtain the signature fragment $Sign_i$.

$$Sign_i = Sign(msg, sk_i) \tag{7}$$

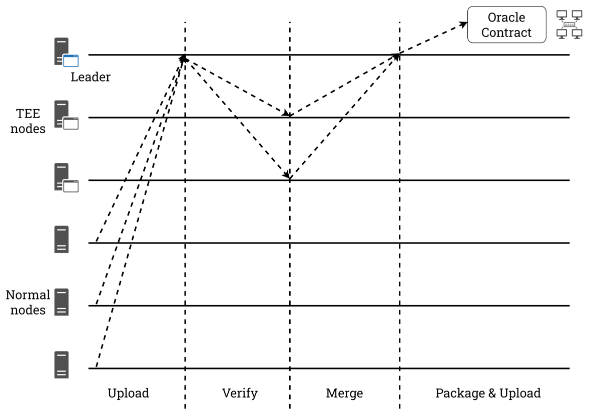


Fig. 4 Data flow of TEE node committee.

As shown in Fig. 4, the oracle node i submits the signature fragment $Sign_i$ to the consensus committee formed by the TEE node. Their node executes the program in the TEE environment [29], [30] to ensure the security and credibility of the program. The consensus committee verifies the signature fragment $Sign_i$ based on the Raft protocol [31] and aggregates it to form a group signature $Sign^*$. In addition, the committee will also record the time when node i uploads the signature as its response time t_i^d. Finally, the Leader node will package the consensus group signature $Sign^*$, the original data msg and the successfully aggregated node response time t^d as S, and use the Leader's private key sk_{leader} to sign to get $sign_S$, upload the oracle contract.

$$Sign_S = Sign(S, sk_{leader}) \tag{8}$$

The Raft-based TEE committee not only avoids the single point of failure issue of TEE nodes but also enhances resistance against Sybil attacks. Additionally, by using the Raft protocol for aggregated signatures, the number of broadcasts for threshold signature data is reduced, thus improving system efficiency. Furthermore, the TEE-based execution environment ensures the trustworthiness of response time records for regular nodes, providing a reliable data foundation for subsequent weighted random selection of nodes.

4.4 Data Validation and QoS Update

After receiving the data $Sign_S$ and S from the TEE committee, the oracle contract uses Leader's pk_{leader} to verify $Verify(S, Sign_S, pk_{leader})$, ensuring that the data has been verified by the TEE committee.

Then, the oracle contract uses the generated threshold signature public key pk to verify the group signature $Sign^*$ formed by aggregation $Verify(msg, Sign^*, pk)$.

Finally, when the verification is passed, the oracle contract update the number of times h and the response time t_d^i that the node i in $Sign^*$ successfully completes the task from the data source d.

In order to improve the quality of service of the oracle, we define a quality of service index of the oracle, which includes the response time and service accuracy of the oracle node i to the request of the data source d. According to the current task results, we divide the device into nodes that complete the task and those that do not complete the task, and take the longest time $max(t_d)$ of the aggregated nodes as the total time t_d^{total} of this task. At the same time, we use the timeout ratio β of the device multiplied by the total time $max(t_d)$ of this task as the estimated time t_d^{delay} of other unfinished tasks.

$$t_d^{delay} = \beta \times t_d^{total} \tag{9}$$

We use the proportion of the response time t_d^i of device i to the longest time t_d^{delay} of this task as its relative response time $T_i^d \in [0, 1]$.

$$T_d^i = 1 - \frac{t_d^i}{t_d^{delay}} \tag{10}$$

We use the number of times that the node i successfully participates in the aggregation h_d^i and the total number of times selected c_d^i as the service accuracy A_d^i of the node to the data source d.

$$A_d^i = \frac{h_d^i}{c_d^i} \tag{11}$$

We set a hyper-parameter α to determine the proportion of response time and accuracy in the QoS of the node to the data source d. $\alpha > 0.5$ indicates that we pay more attention to the response time of the node, while $\alpha \leq 0.5$ indicates that we pay more attention to the service accuracy of the node.

$$Qos_d^i = (\alpha * T_d^i) + ((1 - \alpha) * A_d^i) \tag{12}$$

When we get the result returned by $Request(m, n, D, r)$'s $|D|$ subtask $Request(m, n, d, r)$, we try to aggregate it to get the final result $Result(s_1, ..., s_{|D|})$ and return the user contract. It can be found that the user contract expects a consistent result for the $|D|$ results, but when the returned results are different in $s_1, ..., s_{|D|}$ and it is difficult to reach an agreement, the user can set different aggregation strategies for multi-data source data according to their own needs.

5. Experimental

In this Section, we implemented a prototype of the system and conducted simulation experiments to validate the improvements in system efficiency and scalability offered by the proposed approach.

5.1 Experiment setting

The oracle system designed in this paper consists of on-chain contracts, off-chain oracle nodes, and TEE committee. The smart contract is written by Golang and deployed on the chainmaker blockchain [32], the on-chain and off-chain establish communication with the on-chain contract through the ChainMaker SDK. The total number of nodes N is 10, the number of nodes n selected for each task is 5, and the threshold m required for each task is 3.

5.2 Evaluation

In order to validate the effectiveness of the proposed scheme, we will analyze how the proposed scheme improves the efficiency of the system and reduces the response time. Then, we also experimentally verify the impact of scalability on the success rate of data aggregation.

5.2.1 Response time

Fig.5 compares the average response time of our scheme with the other three selection strategies (random selection, only selecting the n nodes with the best QoS, and the n nodes with the worst QoS) in 100 tasks. Obviously, the proposed scheme has a significant performance improvement compared with the traditional random node selection scheme. The average response time per task is reduced by 9.92%, and the standard deviation of response time is reduced by 2.8%. Although the proposed node selection scheme cannot achieve the optimal performance, the weighted random method can maintain a certain randomness to prevent specific attacks against high-reputation nodes. It is consistent with our original intention to improve the performance of the oracle system while maintaining security. The above results also show that the proposed framework and QoS standard can effectively record and describe the behavior of nodes, and provide data support for upper-layer strategies such as node selection.

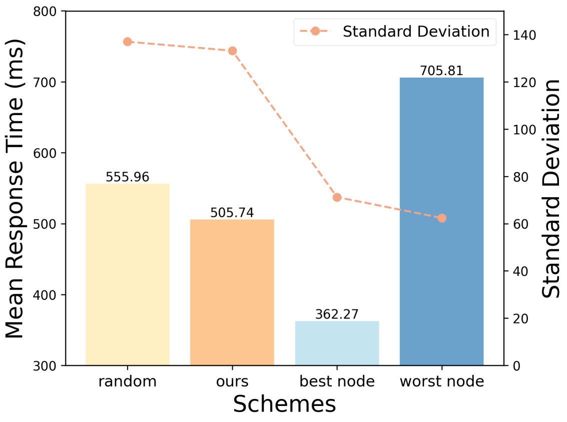


Fig. 5 Response time of different strategies in 100 request tasks.

In addition, as shown in Fig.6, we also calculate the proportion of response time in oracle task. It can be found that the response time of the nodes off-chain accounts for more than 50% of the whole task time. Therefore, reducing the response time is of great significance to improve the execution efficiency of the oracle task. Furthermore, the time spent on an oracle task is also closely related to the efficiency of the blockchain used. However, it is foreseeable that the gradual improvement of blockchain consensus efficiency and the more complex network conditions in practical applications will further increase this proportion, and the advantages of the proposed scheme will be further highlighted.

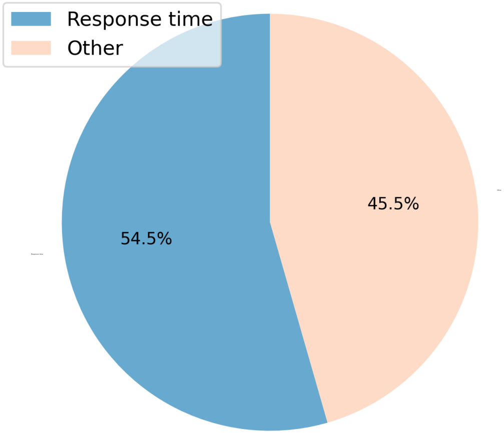


Fig. 6 The proportion of response time in a task.

5.2.2 Effectiveness of QoS

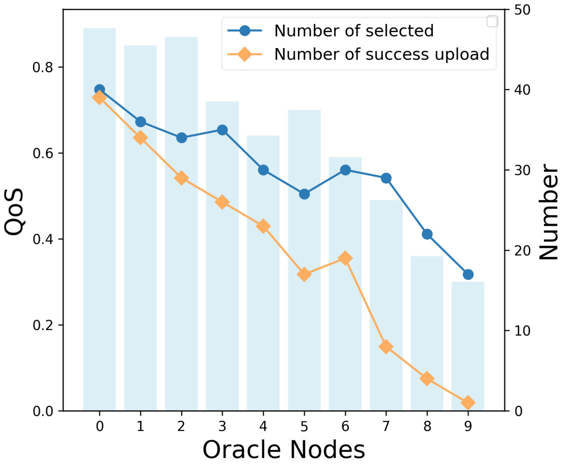


Fig. 7 Effectiveness analysis of QoS computing.

Fig.7 shows the change of metrics after our scheme performs 100 request tasks on a data source on 10 oracle nodes. Obviously, it can be observed that there is a clear correlation between the node QoS and the successful completion of the task. This shows that the proposed scheme can identify nodes with slow response and reduce their QoS according to the actual operation status of nodes. This is consistent with our expectations. The higher the QoS of the high-performance nodes that can complete the task in time, the greater the probability of their selection.

5.2.3 Node selection and QoS

In addition, we also analyze the relationship between QoS and the number of nodes selected more intuitively through the Fig. 8. We can find that with the decrease of QoS, the number of node selection is also decreasing. This shows that the proposed node selection scheme is effective.

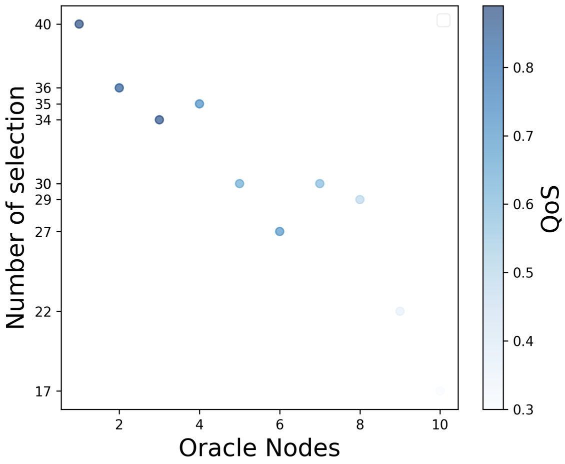


Fig. 8 The Relationship between node selection and QoS.

5.2.4 Expandability

Fig. 9 shows the impact of system scalability on the success rate of data aggregation in real-time data acquisition tasks in the Internet of Things. We observe the effect of choosing a different number of nodes m per round on the success rate of oracle aggregation with a threshold n of 2. The result shows that selecting more nodes per round of tasks is more likely to result in successful aggregation under the same threshold. This is consistent with the original intention of our design, and it is also the reason why we design the hybrid TEE oracle architecture. Massive ordinary nodes are responsible for data acquisition tasks to improve system performance. A small number of secure TEE nodes form a fault-tolerant recording and verification committee, which is responsible for ensuring the security and credibility of data.

6. Conclusions

This paper proposes a distributed efficient blockchain oracle scheme for the Internet of Things, which aims to obtain IoT data efficiently and credibly for the blockchain. In order to improve the success rate of aggregation data when the oracle obtains real-time data, we divide the nodes into two categories, and combine the advantages of ordinary nodes and TEE nodes to improve the scalability and aggregation success rate. Then we propose a new oracle quality of service

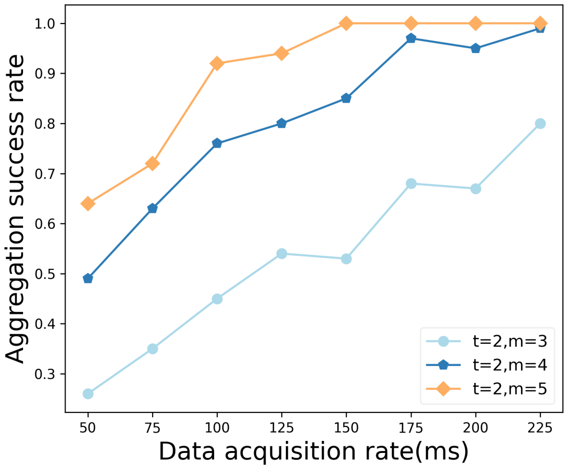


Fig. 9 The relationship between aggregation success rate and the number of participating nodes.

standard to measure the relationship between nodes and data sources, select more suitable nodes for tasks, and improve system performance. Experiments show that the scheme can effectively improve the access efficiency and scalability of the system, so that the blockchain can obtain IoT data more efficiently.

Appendix A: Modeling of the problem

A.1 Hypothesis

We assume that n selected oracle nodes are given the data request task at the same time, their response times satisfy the (μ, σ) normal distribution, and the collection of data from IoT devices satisfies a periodic variation with a period of T [33]. As shown in the Fig. A· 1.

A.2 Question

The problem is defined as: in the normal distribution curve in the interval of $[0, +\infty]$ and x axis surrounded by the area of S_{total}, Randomly scattered n points, the probability that the number of scattered points m in the $[x, x+T]$ interval area S_t is greater than or equal to the threshold value t of the threshold signature.

A.3 The relationship between threshold requirement and n

The total probability interval area S_{total} is expressed as:

$$S_{total} = \int_0^{+\infty} N(\mu, \sigma^2)dx \tag{A· 1}$$

Because of $\mu \geq 0$, so $\frac{1}{2} \leq S_{total} \leq 1$.
The selected probability interval S_t in T time is:

$$S_t = \int_t^{t+T} N(\mu, \sigma^2)dx \tag{A· 2}$$

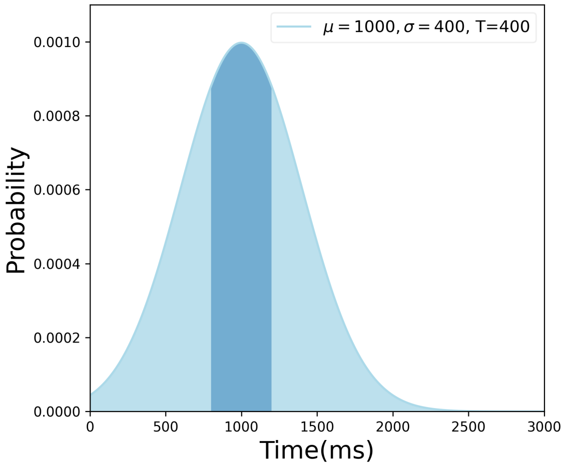


Fig. A· 1 The Relationship between Node Selection and QoS.

Therefore, when test or select of 1 node, the probability of occurrence of event A is P $(0 < p < 1)$:

$$P = \frac{S_t}{S_{total}} \tag{A· 3}$$

In n independent experiments, the probability that event A happens to occur m times is:

$$P_n(m) = C_n^m P^m (1-P)^{n-m} \tag{A· 4}$$

We transform the problem into an N-fold Bernoulli experiment, conduct n experiments or select n nodes, and the probability of successfully aggregating group signatures varies with n as follows($P_n(m \geq t)$ is the maximum probability that the threshold t can be met):

$$P_n(m \geq t) = \sum_{i=t}^{n} C_n^i P^i (1-P)^{n-i} \tag{A· 5}$$

Since both S_t and S_{total} are independent of n, P is also independent of n.

Also, $C_n^i = \binom{n}{i} = \frac{n!}{i!(n-i)!}$is a positive correlation with n.

And $P_n(m \geq t)$ is an additive term. When n increases, $P_n(m \geq t)$ increases monotonically.

Therefore, n is positively correlated with P, and when n increases, P also increases.

A.4 Conclusion

Therefore, we can get a conclusion that when the number of nodes n selected for each task increases, the probability P of successful data aggregation obtained in changing data such as the Internet of Things also increases.

References

[1] Q. Wang, X. Zhu, Y. Ni, L. Gu, and H. Zhu, “Blockchain for the iot and industrial iot: A review,” Internet of Things, vol.10, p.100081, 2020.

[2] M.S. Rahman, M. Chamikara, I. Khalil, and A. Bouras, “Blockchain-of-blockchains: An interoperable blockchain platform for ensuring iot data integrity in smart city,” Journal of Industrial Information Integration, vol.30, p.100408, 2022.

[3] A. Pasdar, Y.C. Lee, and Z. Dong, “Connect api with blockchain: A survey on blockchain oracle implementation,” ACM Computing Surveys, vol.55, no.10, pp.1–39, 2023.

[4] S.K. Ezzat, Y.N. Saleh, and A.A. Abdel-Hamid, “Blockchain oracles: State-of-the-art and research directions,” IEEE Access, 2022.

[5] U. Bodkhe, S. Tanwar, K. Parekh, P. Khanpara, S. Tyagi, N. Kumar, and M. Alazab, “Blockchain for industry 4.0: A comprehensive review,” IEEE Access, vol.8, pp.79764–79800, 2020.

[6] P. Jauernig, A.R. Sadeghi, and E. Stapf, “Trusted execution environments: properties, applications, and challenges,” IEEE Security & Privacy, vol.18, no.2, pp.56–60, 2020.

[7] F. Zhang, E. Cecchetti, K. Croman, A. Juels, and E. Shi, “Town crier: An authenticated data feed for smart contracts,” Proceedings of the 2016 aCM sIGSAC conference on computer and communications security, pp.270–282, 2016.

[8] S. Woo, J. Song, and S. Park, “A distributed oracle using intel sgx for blockchain-based iot applications,” Sensors, vol.20, no.9, p.2725, 2020.

[9] Y. Liu, J. Liu, M.A.V. Salles, Z. Zhang, T. Li, B. Hu, F. Henglein, and R. Lu, “Building blocks of sharding blockchain systems: Concepts, approaches, and open problems,” Computer Science Review, vol.46, p.100513, 2022.

[10] J. Wang, Y. Zhou, X. Li, T. Xu, and T. Qiu, “A node rating based sharding scheme for blockchain,” 2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS), pp.302–309, IEEE, 2019.

[11] P. Zhang, W. Guo, Z. Liu, M. Zhou, B. Huang, and K. Sedraoui, “Optimized blockchain sharding model based on node trust and allocation,” IEEE Transactions on Network and Service Management, 2023.

[12] J. Peterson, J. Krug, M. Zoltu, A.K. Williams, and S. Alexander, “Augur: a decentralized oracle and prediction market platform,” arXiv preprint arXiv:1501.01042, 2015.

[13] J. Adler, R. Berryhill, A. Veneris, Z. Poulos, N. Veira, and A. Kastania, “Astraea: A decentralized blockchain oracle,” 2018 IEEE international conference on internet of things (iThings) and IEEE green computing and communications (GreenCom) and IEEE cyber, physical and social computing (CPSCom) and IEEE smart data (SmartData), pp.1145–1152, IEEE, 2018.

[14] S.N. Steve Ellis, Ari Juels, “Chainlink a decentralized oracle network.” https://chain.link, 2017.

[15] D. Network, “A decentralized oracle service network to boost blockchain usability with real world data and computation power.” https://dos.network, 2019.

[16] Y. Cai, G. Fragkos, E.E. Tsiropoulou, and A. Veneris, “A truth-inducing sybil resistant decentralized blockchain oracle,” 2020 2nd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS), pp.128–135, IEEE, 2020.

[17] L. Chen, R. Yuan, and Y. Xia, “Tora: A trusted blockchain oracle based on a decentralized tee network,” 2021 IEEE International Conference on Joint Cloud Computing (JCC), pp.28–33, IEEE, 2021.

[18] C. Liu, H. Guo, M. Xu, S. Wang, D. Yu, J. Yu, and X. Cheng, “Extending on-chain trust to off-chain–trustworthy blockchain data collection using trusted execution environment (tee),” IEEE Transactions on Computers, 2022.

[19] F. Zhang, D. Maram, H. Malvai, S. Goldfeder, and A. Juels, “Deco: Liberating web data using decentralized oracles for tls,” Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security, pp.1919–1938, 2020.

[20] J. He, R. Wang, W.T. Tsai, and E. Deng, “Sdfs: a scalable data feed service for smart contracts,” 2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS), pp.581–585, IEEE, 2019.

[21] T. Manoj, K. Makkithaya, and V. Narendra, "A trusted iot data sharing and secure oracle based access for agricultural production risk management," Computers and Electronics in Agriculture, vol.204, p.107544, 2023.

[22] Y. Lin, Z. Gao, W. Shi, Q. Wang, H. Li, M. Wang, Y. Yang, and L. Rui, "A novel architecture combining oracle with decentralized learning for iiot," IEEE Internet of Things Journal, 2022.

[23] L. Gigli, I. Zyrianoff, F. Montori, C. Aguzzi, L. Roffia, and M. Di Felice, "A decentralized oracle architecture for a blockchain-based iot global market," IEEE Communications Magazine, vol.61, no.8, pp.86–92, 2023.

[24] G. Caldarelli, "Understanding the blockchain oracle problem: A call for action," Information, vol.11, no.11, p.509, 2020.

[25] H. Rong, J. Mo, B. Chang, G. Sun, and F. Long, "Key distribution and recovery algorithm based on shamir's secret sharing," Journal on Communications, vol.36, no.3, pp.60–69, 2015.

[26] P.S. Efraimidis and P.G. Spirakis, "Weighted random sampling with a reservoir," Information processing letters, vol.97, no.5, pp.181–185, 2006.

[27] P.S. Efraimidis, "Weighted random sampling over data streams," in Algorithms, Probability, Networks, and Games, pp.183–195, Springer, 2015.

[28] T. Hanke, M. Movahedi, and D. Williams, "Dfinity technology overview series, consensus system," arXiv preprint arXiv:1805.04548, 2018.

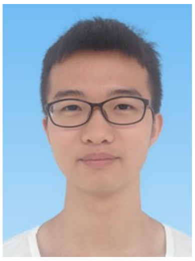
[29] V. Costan and S. Devadas, "Intel sgx explained," Cryptology ePrint Archive, 2016.

[30] Z. Hua, J. Gu, Y. Xia, H. Chen, B. Zang, and H. Guan, "vtz: Virtualizing arm trustzone," 26th USENIX Security Symposium (USENIX Security 17), pp.541–556, 2017.

[31] D. Ongaro and J. Ousterhout, "In search of an understandable consensus algorithm," 2014 USENIX Annual Technical Conference (Usenix ATC 14), pp.305–319, 2014.

[32] ChainMaker, "Chainmaker." `https://chainmaker.org.cn/home`, 2017.

[33] C. Fraleigh, F. Tobagi, and C. Diot, "Provisioning ip backbone networks to support latency sensitive traffic," IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428), pp.375–385, IEEE, 2003.



Youquan Xian received the BEdegree from the BeiBu Gulf University, in 2021. He is currently working toward the master's degree at Guangxi Normal University. His research interests include blockchain, edge computing, and federated learning.

Lianghaojie Zhou received the BEdegree from the HuaQiao University, in 2019. He is currently working toward the master's degree at Guangxi Normal University. His research interests include blockchain and energy trading.

Jianyong Jiang is currently pursuing a master's degree at Guangxi Normal University. His research encompasses various aspects, including distributed ledger technology, consensus algorithms, and smart contracts.

Boyi Wang received the BEdegree from the Chongqing Jiaotong University, in 2020. He is currently working toward the master's degree at Guangxi Normal University. His main interests are blockchain and privacy computing.

Hao Huo obtained a bachelor's degree in 2021 from Shanxi Agricultural University. He is currently studying for a master's degree from Guangxi Normal University. His research interests include blockchain and cryptography, and he is committed to the privacy protection of blockchain.

Peng Liu received his Ph.D. degree in 2017 from Beihang University, China. He began his academic career as an assistant professor at Guangxi Normal University in 2007 and was promoted to full professor in 2022. His current research interests are focused on federated learning and blockchain.