Received: 20 October 2024 | Revised: 3 December 2024 | Accepted: 27 February 2025 | Published online: 27 March 2025

Journal of Data Science and Intelligent Systems
2025, Vol. 00(00) 1–9
DOI: 10.47852/bonviewJDSIS52024592

BON VIEW PUBLISHING

RESEARCH ARTICLE

Notary Evaluation Algorithm Adaptable to Node State Changes for Cross-Chain Notaries

Chenhong Xie[1], **Xiaoming Hu**[1,*], **Shuangjie Bai**[1] **and Yan Liu**[1]

[1]*School of Computer and Information Engineering, Shanghai Polytechnic University, China*

Abstract: Cross-chain technology has emerged as a current research hotspot in the field of blockchain. Scholars have proposed various cross-chain technologies to address the issue of "data islands". Among these technologies, the notary mechanism is one of the principal approaches. However, existing research on notary evaluation tends to favor older nodes and struggles to adapt to changes in node states. To address this challenge, this paper proposes a notary evaluation algorithm capable of adapting to node state changes. The algorithm divides the node reputation value into transaction scores and trust scores. The reputation value is then calculated based on the actual transaction cycle, thereby bringing the node reputation value closer to the node's current performance level. The experimental results show that the algorithm can effectively cope with node state changes and more accurately identify malicious nodes and trustworthy nodes than similar schemes, preventing the efficiency and success rate of the cross-chain system from decreasing due to the inability of the algorithm to adapt to node state changes. At the same time, the problem of low reputation value of new nodes due to the lack of trust relationship is alleviated, which is more conducive to mobilizing nodes to join the notary group. Through these improvements, the reliability and operational efficiency of the cross-chain system can be effectively improved.

Keywords: cross-chain, blockchain, notary mechanism, reputation value, PageRank

1. Introduction

In 2008, Satoshi Nakamoto first introduced the Bitcoin system [1], which brought blockchain technology into the spotlight and gained widespread attention. Due to its characteristics of openness, transparency [2], decentralization [3], and immutability [4], blockchain technology has been widely applied in various industries, including but not limited to supply chain [5], healthcare [6], transportation [7], and others. With the development of blockchain technology, different blockchains have emerged. However, these blockchains operate independently and lack interconnectivity, leading to the emergence of value islands [8]. Achieving interconnection and interoperability between blockchains has become an unavoidable issue [9]. The development of cross-chain technology is crucial in addressing the problem of data islands and promoting further advancements and applications of cross-chain technology [10].

Cross-chain technology was initially proposed by Ripple [11]. Currently, the main cross-chain mechanisms include the notary mechanism, sidechain/relaying mechanism [12], hash locking mechanism [13], and distributed private key control [14]. Among them, the notary mechanism is relatively simple and easy to implement compared to other cross-chain mechanisms. The working mode of the notary mechanism is similar to that of traditional exchanges [15] and is more compatible with existing frameworks [16]. However, the reliability and efficiency of this cross-chain mechanism depend on the selected notaries. If unreliable notaries are elected, it may lead to transaction failures and low transaction volumes and undermine the reliability and credibility of the cross-chain system. Therefore, an effective notary evaluation algorithm that can assess notary nodes and distinguish malicious nodes is of paramount importance.

This paper presents a notary evaluation algorithm that can adapt to changes in node states, dividing the notary's reputation value into transaction scores and trust scores. The transaction score is calculated based on the data generated when the node undertakes cross-chain tasks. The evaluation time of the node is used to determine the transaction cycle, and the transaction score is computed by considering both the historical scores and the scores generated from the transaction performance within the current cycle, ensuring that the final result closely reflects the node's current state. The trust score is determined through trust voting among nodes and is jointly influenced by the trust relationships among nodes and their long-term performance. Prior to node evaluation, trust relationships among nodes are collected, and an improved PageRank algorithm is used for iterative computation to calculate the trust score of each node. Then, the weights for transaction scores and trust scores are calculated based on the actual number of transaction cycles in which the node participates. These weights are used to compute the node's final reputation value.

The main contributions of this paper are as follows:

1) In order to improve the accuracy and fairness of node evaluation, this paper proposes the concept of an effective transaction cycle, which divides different cycles by the time point of node evaluation. According to whether the node undertakes cross-chain tasks during the transaction cycle, the transaction cycle is divided into the node's effective transaction cycle and

*Corresponding author: Xiaoming Hu, School of Computer and Information Engineering, Shanghai Polytechnic University. Email: xmhu@sspu.edu.cn

ineffective transaction cycle. The calculation of reputation value is aided by the number of effective transaction cycles of a node, which can distinguish between old and new nodes, and also takes into account the nodes that have joined the notary group for a longer time but have fewer cross-chain tasks.

2) A new method for calculating reputation value is proposed. The method will evaluate the reputation value of nodes from two dimensions: transaction performance and trust relationship. The scheme fully considers the variability of node state, utilizes the effective transaction period to distinguish the transaction data in different periods, increases the influence of recent data on the reputation value, and improves the adaptability of the algorithm to the change of node state.
3) In this paper, several experiments are designed from the aspects of identification of node nature, evaluation of new nodes with different natures, reputation value and ranking of trusted new nodes, decline in node's transaction performance, and transformation of reliable nodes into malicious nodes and are compared with several representative schemes. It is verified that this scheme can effectively cope with the change of node status as well as give reasonable evaluation according to the nature of old and new nodes while recognizing malicious nodes, which proves the effectiveness of the scheme.

The structure of this paper is as follows: Section 1, Introduction, introduces the background information of blockchain cross-chain technology. Section 2, Related work, discusses the schemes of notary election in the cross-chain field in the past few years. Section 3, Relevant knowledge, introduces the notary public mechanism and PageRank ranking algorithm. Section 4, Theoretical framework, introduces the theory and implementation steps of the algorithm. Section 5, Theoretical and experimental analysis, through the experiment to verify the effectiveness of the algorithm. Section 6, Conclusion.

2. Related Work

The improved PageRank algorithm was utilized in Dai et al. [17] for calculating the reputation value of notaries. By considering the transaction information of notary nodes and the trust relationships among nodes, the algorithm calculated the reputation value of each node, serving as the basis for electing high-quality notary nodes. However, the algorithm focused on trust relationships and lacked differentiation in assessing the inherent value differences among nodes.

In Jiang et al. [18], a time factor was introduced into the calculation of the damping coefficient. The evaluation considered both the inherent value of the notary node and the value obtained through undertaking notary tasks, thereby alleviating the bias toward older nodes in the election algorithm. Cao and Yang [19] proposed a two-stage election algorithm that used the improved PageRank algorithm to evaluate nodes and employed a verifiable random function for node selection. This approach increased the randomness and unpredictability of notary elections while mitigating the Matthew effect and alleviating the centralization issue. Although Jiang et al. [18] as well as Cao and Yang [19] have made improvements to the algorithm in different ways, they suffer from the issue of a single indicator when calculating the inherent value of nodes, which prevents a comprehensive evaluation of the nodes.

Chen et al. [20], based on the original evaluation system, introduces factors such as the historical transaction records of nodes, message response time, and collateral deposits, thus improving the evaluation system for validators. Chen et al. [21] propose a multiple indicator credit ranking scheme based on a notary mechanism, addressing the problem of a single evaluation indicator for the intrinsic value of certification nodes through multiple indicator evaluation and the entropy weighting method. This scheme provides a more comprehensive evaluation of the value of nodes by incorporating indicators such as transaction volume, user feedback, and success rate when assessing the nodes' intrinsic value. Zhao and Cao [22] combine trust voting scores between nodes with transaction performance to calculate the evaluation value between nodes. It uses the average value of the evaluation as the basic reputation value of nodes, thereby reducing the initial value of malicious nodes and the probability of them entering the certification group. The three aforementioned articles have employed different methods to increase the evaluation indicators for nodes, making the algorithm's evaluation of nodes more comprehensive. However, using the historical average value as the basis for evaluating nodes can lead to evaluation results that do not align with the current state of the nodes. Xiong et al. [23] propose a notary node election scheme based on a reputation mechanism. It adjusts the reputation value of nodes based on transaction completion status and reports of malicious behavior and removes malicious nodes based on their reputation value. This algorithm can effectively elect reliable nodes but also limits the election results to well-performing old nodes, making it difficult for new nodes to obtain opportunities to undertake cross-chain tasks. Cao et al. [24] calculate the reputation value of nodes based on the voting results between nodes, transaction processing efficiency, and transaction processing success rate. The algorithm effectively distinguishes nodes with different characteristics. However, the method of calculating voting scores based on the proportion of votes received by nodes in the total votes makes it difficult to prevent collusion among malicious nodes. In the aforementioned studies, certain approaches fail to differentiate between newly joined nodes and old nodes when calculating reputation values, resulting in a disadvantage for new nodes during node evaluation due to a lack of trust relationships. Although the literature [18–20] differentiates between new and old nodes, they rely on the time of joining the notary group as the basis for controlling the weight of trust relationships and transaction performance when calculating reputation values. This criterion lacks fairness for nodes that have been in the group for a longer duration but have undertaken fewer cross-chain tasks.

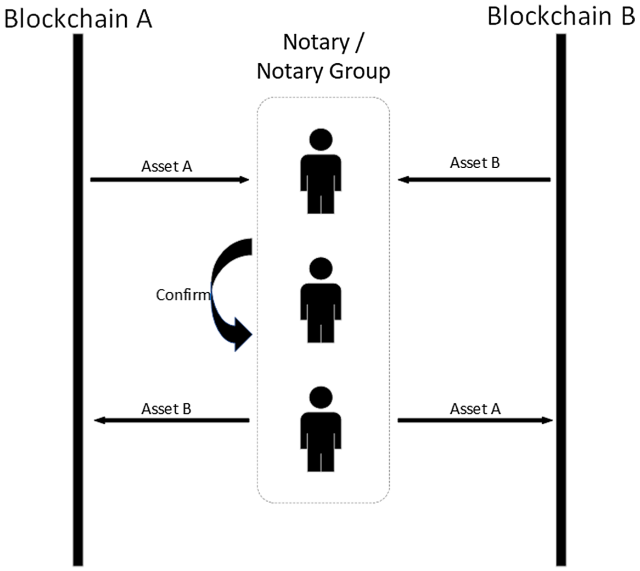
Through the analysis of the aforementioned literature, there are still several shortcomings in the current algorithms for notary node evaluation. First, when nodes engage in trust voting, they tend to trust nodes with outstanding performance and more trust relationships. However, nodes with average performance struggle to establish trust relationships. In such cases, if malicious nodes collude and engage in mutual voting, it may lead to the ranking of malicious nodes surpassing that of ordinary nodes. Second, existing research attempts to adjust the damping factor by using the time of nodes joining the certification group, aiming to reduce the disadvantage faced by new nodes due to a lack of trust relationships. However, this approach lacks fairness for nodes that have been in the group for a longer duration but have undertaken fewer cross-chain tasks. These nodes, due to their limited involvement in cross-chain tasks, face difficulties in gaining trust from other nodes based on transaction performance. Their characteristics are more aligned with those of new nodes; however, the node evaluation still adopts the reputation calculation method of old nodes. Third, changes in node status cannot be promptly reflected in reputation values. Existing research calculates the inherent value based on transaction performance using historical average values, which lack consideration for the temporal aspect and cannot accurately reflect the current state of the nodes.

3. Relevant Knowledge

3.1. Notary mechanism

The working principle of the notary mechanism is similar to that of traditional exchanges, making it easier to implement and deploy compared to other solutions. This mechanism does not require focusing on the structure and consensus mechanism of the blockchain but rather focuses on the cross-chain tasks themselves. The notary mechanism works by electing trustworthy nodes or notary groups to undertake cross-chain tasks. However, not all notary nodes can be considered reliable, as there is a risk of malicious behavior by certain nodes [25]. Depending on the method of signature, the notary mechanism can be categorized into single-signature notary mechanisms and multi-signature notary mechanisms. The basic workflow of the notary mechanism is shown in Figure 1, where notary nodes or notary groups act as intermediaries to facilitate the circulation of assets between different blockchains.

Figure 1
The workflow of the notary public mechanism



3.2. PageRank

The PageRank algorithm [26] was proposed by Google and is used to evaluate and rank the importance of web pages. The core idea is to give the same initial value to the pages and then use the linking relationship between the pages to transfer the value, iteratively calculating the value of the page until the value of all the pages is stabilized.

There are two important assumptions for this algorithm:

1) Quantitative assumption: the more links a page has to this page, the more important the page is. In the notary evaluation algorithm, the more a node is trusted by more nodes, the more trustworthy the node is.
2) Quality assumption: the importance of each page varies, and pages linked to by high-importance pages have higher importance. In the notary evaluation algorithm, the trustworthiness of each node varies, the node trusted by the high trustworthiness node has higher trustworthiness.

The iterative formula for the PageRank algorithm is as follows:

$$\mathrm{PR(u)} = (1-d) + d \times \sum_{v \in \mathrm{B(u)}} \frac{PR(v)}{L(v)} \quad (1)$$

where $PR(u)$ represents the PR value (PageRank value) of web page u, indicating the importance of web page u, d is the damping coefficient usually taken as 0.85, $B(u)$ represents the set of all web pages linking out to u, and $L(v)$ represents the number of links out to page v.

4. Theoretical Framework

Based on the work presented in Zhao and Cao [22], this paper proposes an adaptive notary evaluation algorithm that can accommodate changes in node states. In this scheme, regular computations and rankings of node reputation values are conducted, and the time interval for reputation value calculations can be freely determined based on the number of transactions. The period between two consecutive reputation value calculations is referred to as a transaction cycle, and if a node undertakes cross-chain tasks during this cycle, it is considered an effective transaction cycle for that node. Prior to each round of node evaluation, trust relationships between nodes and the transaction records of each node during the current transaction cycle are collected.

Each notary node sets expected weights for transaction success rate and transaction efficiency when joining the notary group, representing the node's expectations for the transaction performance of other nodes. Subsequently, the node's score for the transaction performance of other nodes will be calculated based on these weights. Meanwhile, the sum of the two weights should be a fixed value, with a minimum value also being present.

In this paper, the reputation value of a node is divided into transaction score and trust score. The transaction score is calculated based on the node's transaction performance. For new nodes that have not undertaken any cross-chain tasks, the transaction data generated when using the node as a cross-chain service user is used for the calculation.

The success rate is an important criterion for assessing the credibility of notary nodes, and it is calculated using the following formula (1). In the formula, $V_{succ}(i)$ represents the success rate score of node i during the current transaction cycle, $Succ_{cur}(i)$ represents the number of successful transactions by node i in the current transaction cycle, and $Numcur(i)$ represents the total number of transactions by node i in the current transaction cycle.

$$\mathrm{V_{succ}(i)} = \frac{\mathrm{Succ_{cur}(i)}}{\mathrm{Num_{cur}(i)}} \quad (2)$$

The efficiency in processing transactions is an important criterion for measuring the value of a node. The transaction efficiency score for node i is calculated using the following formula:

$$\mathrm{T_{tra}(i)} = \frac{\mathrm{Time_{cur}(i)}}{\mathrm{Succ_{cur}(i)}} \quad (3)$$

$$\mathrm{V_{effi}(i)} = 1 - \frac{T_{tra}(i) - Min(Ttra)}{Max(Ttra) - Min(Ttra)} \times 0.9 \quad (4)$$

$T_{tra}(i)$ represents the average duration for node i to process a cross-chain transaction in the current cycle, while $Time_{cur}(i)$ represents the total transaction time for node i in the current cycle. Since transaction efficiency can be influenced by different cross-chain protocols and its value range cannot be unified with the success rate, it is necessary to normalize the average transaction duration $T_{tra}(i)$ for each node. The node with the shortest average duration (highest efficiency) is assigned a score of 1, while the node with the longest average duration (lowest efficiency) is assigned a score of 0.1. Other nodes are assigned scores based on their performance. $V_{effi}(i)$ represents the transaction efficiency score of node i in the current transaction cycle, T_{tra} represents the collection of average transaction duration for all nodes in the current transaction cycle, $Max(T_{tra})$ represents the longest average transaction duration, and $Min(T_{tra})$ represents the shortest average transaction duration. For nodes with a transaction success rate of 0%, since it is not possible to calculate the average duration for individual transactions, their efficiency score is fixed at 0.05. The calculation of the transaction score is defined by the following formula (5).

TransactionValue(i) represents the transaction value for node i in the current reputation evaluation, which will be used as the historical score in the next round of node evaluation. $V_{his}(i)$ represents the historical score of node i. W_h and W_c represent the weights for the historical score and current cycle score, respectively. To ensure that the score reflects the node's current performance, it is suggested in this paper that W_h should be smaller than W_c. W_e and W_s represent the weights for the efficiency score and success rate score when calculating the current cycle score. Considering the higher impact of transaction success rate during the transaction process compared to transaction efficiency, it is recommended to set W_e smaller than Ws. For nodes that have not participated in any cross-chain tasks for the first time, the average of the historical scores of the previous 2/3 nodes is used as their own historical score. For nodes that have not undertaken any transaction tasks in the current transaction cycle, the transaction score obtained from the previous node evaluation is used as the current score.

$$\begin{aligned}\mathrm{Transaction\ Value(i)} = {} & \mathrm{W_h \times V_{his}(i) + W_c} \\ & \times (\mathrm{W_e \times V_{effi}(i) + W_s \times V_{succ}(i)})\end{aligned} \tag{5}$$

When calculating the trust score of a node, the evaluation of this node toward other nodes is first calculated using the transaction success rate weight and transaction volume weight pre-set for the node. The calculation formula is as follows:

$$\mathrm{Eva_{i\to j}} = \mathrm{W_{succ}(i)} \times VH_{succ}(j) + W_{effi}(i) \times \mathrm{VH}_{effi}(j) \tag{6}$$

where $Eva_{i\to j}$ represents the rating of node i toward node j; $W_{succ}(i)$ and $W_{effi}(i)$ represent the weights for success rate and efficiency set by node i upon joining the notary group, representing node i's expectations for the transaction performance of other nodes; and $VH_{succ}(j)$ and $VH_{effi}(j)$ represent the success rate score and transaction efficiency score of node j. Considering that the establishment of trust is the result of long-term observations of the transaction performance of the trusted node, $VH_{succ}(j)$ and $VH_{effi}(j)$ are calculated using transaction data from all periods, including the current transaction cycle, for node j. The calculation method is the same as the calculation method for V_{succ} and V_{effi} in the transaction score.

In this paper, an improved PageRank algorithm is employed to calculate the trust value $PR(i)$ of node i. The formula is as follows:

$$\mathrm{PR}(i) = \frac{\sum_{j\in A} Eva_{j\to i}}{N-1} + d \times \sum_{j\in G}\left(PR(j) \times \frac{Eva_{j\to i}}{\sum_{k\in V} Eva_{j\to k}}\right) \tag{7}$$

In the formula, d represents the damping factor, which the value of d here is 0.85 according to the setting of the original algorithm. A represents the set of all notary nodes excluding node i, and N represents the total number of notary nodes. $Eva_{j\to i}$ represents the evaluation score of node j toward node i, V represents the set of all nodes trusted by node j, and G represents the set of all nodes that trust i.

During the iteration process, the algorithm assigns a portion of the node's own PR value to the nodes it trusts. However, as different nodes have varying transaction performances, the degree of compliance with one's expectations can be determined through the evaluation scores. Nodes that receive higher scores, indicating a closer alignment with their expectations, should be allocated a larger share of the PR value. By proportionally distributing the node's PR value based on the evaluation scores provided by the node, this approach allows nodes that meet the expectations of other nodes to obtain more PR value. This further distinguishes the value of different nodes.

To facilitate the computation of the final reputation value, the trust value $PR(i)$ and transaction score $TransactionValue(i)$ are normalized as follows. $V_{PR}(i)$ represents the trust score of node i, and V_{TV} (i) represents the transaction score of node i. PR denotes the set of PR values for all nodes, and TransactionValue denotes the set of transaction values for all nodes.

$$\mathrm{V_{PR}(i)} = 10 + \frac{\mathrm{PR}(i) - Min(PR)}{Max(PR) - Min(PR)} \times 90 \tag{8}$$

$$\mathrm{V_{TV}(i)} = 10 + \frac{\mathrm{TransactionValue(i)} - Min(\mathrm{TransactionValue})}{Max(\mathrm{TransactionValue}) - Min(\mathrm{TransactionValue})} \times 90 \tag{9}$$

After calculating the transaction score and trust score of a node, the final score of the node is computed using the following formula, where $CS(i)$ represents the reputation value of node i:

$$\mathrm{CS}(i) = (2 - dr(i)) \times \mathrm{V_{TV}(i)} + \mathrm{dr(i)} \times \mathrm{V_{PR}}(i) \tag{10}$$

When evaluating nodes, older nodes that have served as notaries for a longer period of time have had more time to gain the trust of other nodes compared to newer nodes, giving them a clear advantage. Therefore, when calculating the final score of a new node, more emphasis should be placed on the transaction score. Considering that there is no absolute correlation between the duration of a node's membership in the notary group and the number of transaction tasks it undertakes, this algorithm assigns weights for the transaction score and trust score based on the number of effective transaction cycles for a node. The calculation of $dr(i)$ is determined by the following formula (11), where T can be adjusted according to the specific circumstances, and t represents the number of effective transaction cycles for node i.

$$\mathrm{dr}(i) = 1 - \mathrm{e}^{-\frac{t}{T}} \tag{11}$$

The pseudo-code is shown in algorithm 1.

Algorithm 1 Reputation Value Calculation Algorithm

```
Input: N: Number of nodes
       TR[]: Node trust relationship table;
       REC[]: Node transaction record table;
       EW[]: Expected weight of node table;
Output: SC[]: Expected weight of node table;
for i in N do
    $V_{succ}$ [i] Calculate_success_rate(REC[i].this_cycle)
    /* Calculate the success rate of this trading cycle.*/
    $T_{tra}$[i] Calculate_average_time(REC[i].this_cycle)
    /* Calculates the average transaction time during the
       trading cycle.*/
end for
for i in N do
    $V_{effi}$[i] Calculate_efficiency_score(Tra;i)
    TransactionValue[i] Calculate_transaction value()
end for
Record(TransactionValue[])
  /* The recorded score is used as the historical score
     in the next calculation. */
for i in N do
    for j in N-1 do
        $VH_{succ}$(i) Calculate_success_rate(REC[i])
        /* Calculate the success rate of trades for all cycles.*/
        $VH_{effi}$(i) Calculate_average_time(REC[i])
        /* Calculate the average transaction time for all cycles.*/
        $Eva_{i\to j}$ Calculate_rating_inter_node()
    end for
end for
for i in N do
    PR[i] Improved_PageRank()
end for
for i in N do
    $V_{PR}$[i] Normalized ($V_{PR}$)
    $V_{TV}$[i] Normalized ($V_{TV}$)
end for
for i in N do
    dr(i)  Calculate_weight(TR[i])
    SC[i]  Calculate_Reputation_Value(TR[],i)
end for
return SC[]
```

5. Theoretical and Experimental Analysis

5.1. Theoretical analysis

This scheme comprehensively considers factors such as a node's transaction performance, trust relationships, state changes, and the distinction between new and old nodes when calculating the node's reputation value. Compared to older nodes, new nodes have a shorter time of joining the notary group and undertake fewer cross-chain transaction tasks, making it difficult for them to establish trust relationships that align with their performance. This results in a significant disadvantage for new nodes during node evaluation. However, in this paper, the use of effective transaction cycles to determine the weight between transaction score and trust score helps mitigate the disadvantage faced by new nodes. Existing research places greater emphasis on the duration of a node's membership in the notary group as a basis for improving the weight of transaction performance in evaluating new nodes. This approach lacks flexibility and fairness, particularly for nodes that have been in the group for a long time but undertake fewer cross-chain transaction tasks. Although these nodes have been members of the notary group for a longer duration, their limited involvement in cross-chain transaction tasks makes it difficult for them to establish trust relationships with other nodes, resembling the characteristics of new nodes. In contrast, this scheme's use of effective transaction cycles to enhance the impact of transaction performance on reputation value is more flexible and can accommodate nodes with a longer membership duration but fewer cross-chain transaction tasks. In the calculation of the transaction score, dividing the score into the portion generated based on the current transaction cycle and the historical score helps reduce the influence of early data on node evaluation. This approach makes the node's reputation value more closely aligned with its current state, effectively adapting to the node's own state changes. When a node's transaction performance decreases due to network or other objective reasons, the evaluation algorithm can quickly adjust its reputation value accordingly. In the calculation of the trust score, evaluating the relationships between nodes through setting expected weights promotes more objective and accurate evaluations among nodes.

5.2. Experimental analysis

To verify the effectiveness of the algorithm, this paper conducted three sets of experiments using a personal computer equipped with an Intel(R) Core i5-7300HQ CPU @ 2.50GHz, 16GB of RAM, and running a 64-bit Windows 10 operating system. The simulations were performed using the Java language on the VScode platform. In the experiments, the value of T was set to 5, and $W_{succ}+W_{effi}=5$, with $W_{succ}\geq 0.5$ and $W_{effi}\geq 0.5$. Additionally, the weights W_h, W_c, W_e, and W_s were set to 0.4, 0.6, 0.3, and 0.7, respectively. All malicious nodes were assigned efficiency weights and success rate weights of 4.5 and 0.5, respectively, to increase their initial scores when calculating trust scores for other malicious nodes.

Experiment 1: This experiment is conducted based on the trust relationships between nodes and their transaction performance to calculate the node rankings for the proposed scheme, the traditional PageRank algorithm [26], the Jiang scheme [18], and the Cao scheme [24]. A total of 40 notary nodes were deployed for the experiment, numbered from 1 to 40. Among them, there were 6 unreliable nodes with the numbers 2, 10, 19, 26, 31, and 36 and 8 reliable nodes with the numbers 1, 3, 5, 11, 12, 17, 25, and 33. No new nodes were added. The malicious nodes had normal transaction efficiency but extremely low success rates. They established trust relationships with each other. High-trust nodes demonstrated outstanding performance and had a large number of trust relationships, while ordinary nodes had fewer trust relationships. For the algorithm proposed in Cao and Zhao [24], this paper assumes that ordinary nodes would vote for nodes with excellent performance, while malicious nodes would collude to vote for Node 2. The results are shown in Table 1.

Table 1 presents the rankings of malicious nodes after a reputation evaluation. In the traditional PageRank algorithm, the mutual voting among malicious nodes resulted in a rapid increase in their trust scores, surpassing the majority of nodes. However, the remaining schemes, including the algorithm proposed in this paper, consider the transaction performance and trust relationships of nodes in different ways. Therefore, these schemes effectively identify malicious nodes and reliable nodes based on their scores.

Experiment 2: This experiment aims to simulate the scenario of new nodes joining in order to demonstrate that the proposed solution can prevent new nodes from having low reputation due to a lack of

Table 1
Ranking of nodes

	Traditional PageRank [26]	Jiang scheme [18]	Scheme of this paper	Cao scheme [24]
Node 1	2	3	2	4
Node 3	3	4	3	6
Node 5	1	5	1	5
Node 11	7	8	8	8
Node 12	6	1	6	7
Node 17	5	6	4	1
Node 25	4	7	5	2
Node 33	8	2	7	3
Node 2	9	36	36	35
Node 10	10	39	38	38
Node 19	11	35	37	36
Node 26	12	40	35	39
Node 31	13	37	39	37
Node 36	14	38	40	40

trust relationships and promptly identify malicious nodes. The experiment initially deployed 34 nodes, including 5 malicious nodes. During the reputation calculation, three new nodes, namely, Node 35, Node 36, and Node 37, were added, and the following specifications were defined: Node 35 is a malicious node but exhibits excellent transaction performance as a user of the cross-chain system. It directly possesses trust from all malicious nodes and attempts to obtain a high reputation score based on this. Node 36 is a malicious node with poor transaction performance as a user of the cross-chain system, and it also possesses trust from all malicious nodes. Node 37 is an ordinary node with excellent transaction performance as a user, but as a new node, it has no existing trust relationships. For comparison purposes, the experiment also included eight high-trust nodes with excellent transaction performance and numerous trust relationships, labeled Node 1, Node 3, Node 5, Node 11, Node 12, Node 17, Node 25, and Node 33. The experimental results are presented in Table 2.

According to Table 2, it can be observed that Node 35 and Node 37 achieved favorable rankings in the initial reputation calculation due to their excellent transaction performance as users of the cross-chain functionality. Their rankings were similar to those of highly trusted nodes. Although Node 35 is a malicious node, its nature cannot be determined at present. However, due to its previous exceptional transaction performance, it obtained an extremely high score. Conversely, Node 36 obtained a lower ranking due to its poor transaction performance. In the subsequent first transaction cycle, Node 35 began engaging in malicious behavior, resulting in a rapid decline in its ranking to the 31st position. Despite immediately gaining trust from all malicious nodes upon joining the notary group, as a new node, its reputation value primarily relies on transaction performance, leading to a swift decline in ranking. On the other hand, Node 37 maintained excellent transaction performance and secured a relatively high ranking, despite lacking trust from other nodes. It relied on its outstanding transaction performance to attain a favorable position and buy time to gain the trust of other nodes. In this experiment, the algorithm effectively identified malicious nodes among the new nodes. It could promptly adjust the reputation values of nodes whose nature could not be directly determined, once they began engaging in malicious behavior. Additionally, for new nodes with outstanding performance, the algorithm enabled them to maintain a higher reputation value despite the lack of trust relationships, allowing them to gain time to further earn the trust of other nodes.

Table 2
Ranking of nodes after new nodes join

	First round	Second round	Third round
Node 35	1	31	34
Node 36	32	34	36
Node 37	2	3	10
Node 1	4	2	3
Node 3	7	7	4
Node 5	3	1	1
Node 11	12	11	9
Node 12	10	6	2
Node 17	9	4	5
Node 25	6	5	7
Node 33	5	9	6

To further demonstrate the effectiveness of the algorithm, let's assume that all nodes maintain their transaction performance as observed during the first transaction cycle after new nodes join the notary group, and Node 37 fails to gain the trust of other nodes. The changes in rankings are shown in Figure 2, while the changes in reputation values are presented in Figure 3. For Node 37, under the assumption of unchanged transaction performance, its reputation score and ranking gradually decline at a slow pace, providing time for establishing trust relationships with other nodes. As for malicious nodes, if they exhibit excellent transaction performance as users of the cross-chain service, they will receive high scores in the initial evaluation. However, once they start engaging in malicious behavior, their rankings and reputation values rapidly decline. If they act as requesters of cross-chain services and have poor transaction performance, their rankings will be low from the first evaluation. Due to the mutual trust relationships among malicious nodes, their reputation values increase to some extent as the proportion of trust scores in the total score rises and gradually stabilizes. After calculations based on the assumption of maintaining the transaction performance of all nodes, Node 35 and Node 36 rank 33rd and 34th, respectively, after 50 transaction cycles. Therefore, the upward trend in reputation values can be neglected.

This experiment demonstrates that the algorithm not only promptly and effectively identifies malicious nodes but also successfully evaluates the reputation of new nodes that lack trust relationships, thereby alleviating the issue of algorithmic bias toward old nodes during reputation assessment.

Experiment 3: In this experiment, the rankings of nodes are calculated based on the trust relationships among nodes and their transaction performance. The rankings of our proposed approach, the traditional PageRank algorithm [26], the Jiang scheme [18], and the Cao scheme [24] are compared. The aim is to verify the adaptability of the algorithm to changes in node states through the changes in node rankings. There are a total of 40 notary nodes in this experiment, with 8 of them being malicious nodes. In the experiment, Node 3, initially considered a highly trusted node, accumulates a significant amount of outstanding transaction data in previous transactions. However, its state undergoes a transformation, and ordinary nodes gradually withdraw their trust in Node 3. Two scenarios are considered in this context: Scenario 1: Node 3 is not a malicious node. After the transformation in its state, Node 3 maintains the trust relationships it had before and does not gain the trust of malicious nodes. The ranking changes are depicted in Figure 4. Scenario 2: Node 3 is a malicious node.

Figure 2
Changes in node ranking

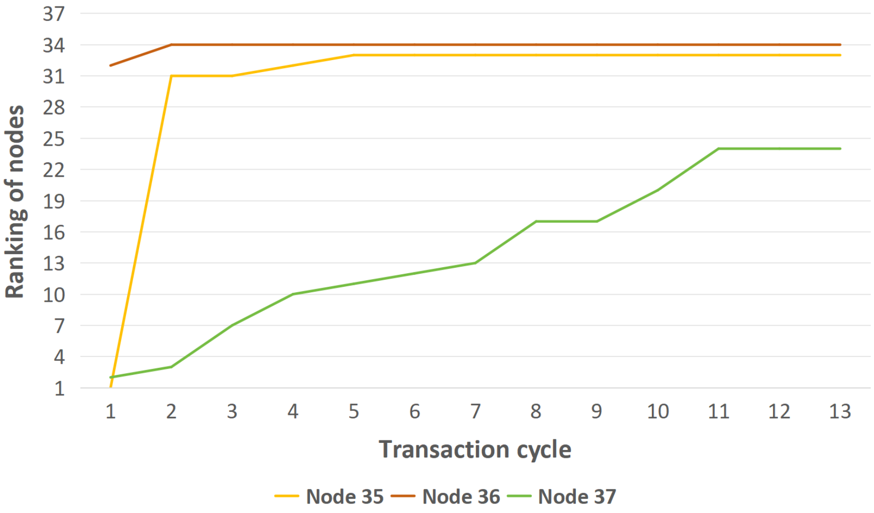
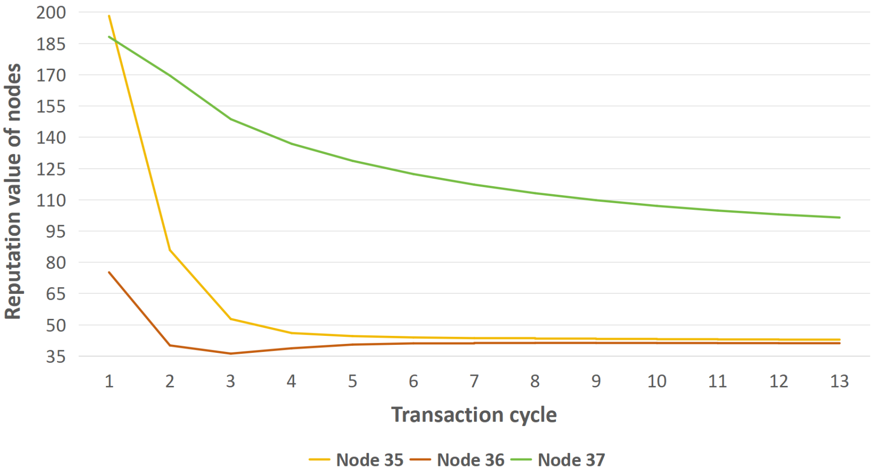


Figure 3
Changes in node reputation values



After the transformation in its state, Node 3 cancels the trust relationships it had with other nodes and establishes trust relationships with malicious nodes. At an appropriate time, other malicious nodes start trusting Node 3, allowing it to obtain a higher reputation value. The ranking changes are illustrated in Figure 5.

During the experiment, the nature of Node 3 had not changed during the first node evaluation. However, between the first and second node evaluations, during the transaction period, Node 3 underwent a transformation in its nature. By the time of the sixth node evaluation, Node 3 had lost the trust of all ordinary nodes.

As shown in Figure 4, the reputation value in the traditional PageRank algorithm is determined by the trust relationships among nodes. Therefore, the reputation value gradually decreases as the node loses trust relationships, but the rate of decrease is influenced by the speed of trust relationship reduction. The Jiang scheme [18] is an improvement upon PageRank, where the reputation value of a node is influenced by both transaction performance and trust

Figure 4
When Node 3 is a malicious node, the ranking of the node changes

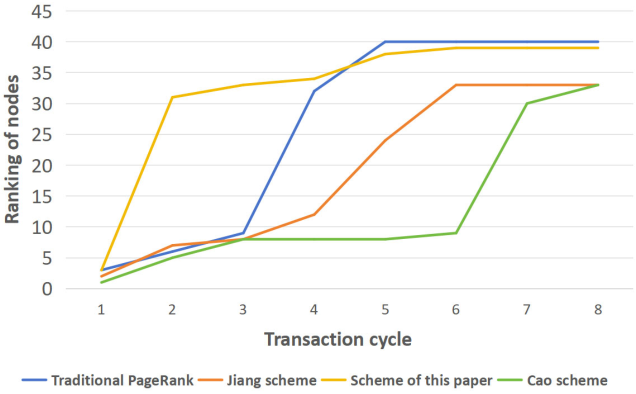
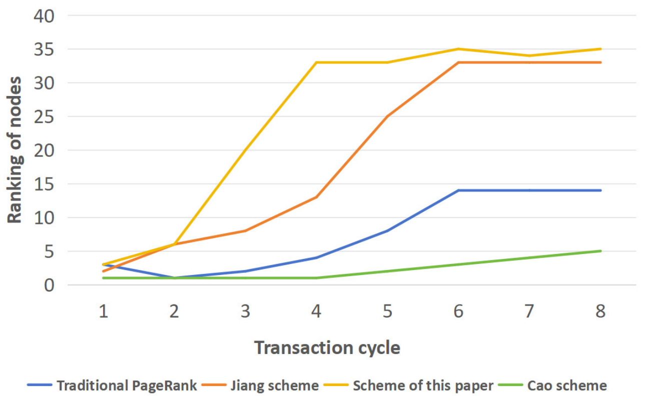


Figure 5
When Node 3 is not a malicious node, the ranking of the node changes



relationships. However, due to the significant accumulation of outstanding transaction data in the early stages, the rate of decrease in reputation is relatively slow. In the Cao scheme [24], trust scores and transaction scores are calculated independently. Although the trust score of Node 3 gradually decreases as trust relationships diminish, its ranking decreases at a slower rate due to the accumulated transaction data. In our proposed approach, after a change in node nature, the trust score gradually decreases as trust relationships diminish. However, the transaction score experiences a significant decline in the current transaction period, allowing the overall score of the node to rapidly adapt to its transaction performance.

In Figure 5 in the traditional PageRank algorithm, node rankings are determined by the trust relationships among nodes. Therefore, even when all ordinary nodes cancel their trust in Node 3, Node 3 can still obtain a favorable ranking by relying on other malicious nodes. In the Cao scheme [24], the decrease in Node 3's ranking is not significant. In terms of trust scores, due to the collusion of malicious nodes, Node 3 has a significant advantage in the voting score, as the malicious nodes unanimously vote in favor of Node 3. On the other hand, ordinary nodes tend to vote for nodes with outstanding performance. However, when there are more reliable nodes, the votes get distributed, making it difficult to resist the coordinated malicious behavior of the nodes. In terms of transaction performance scores, the previous outstanding transaction data accumulated by Node 3 makes it difficult for short-term malicious behavior to have an immediate impact on its score. As a result, Node 3 ends up with a higher ranking than most nodes. In the Jiang scheme [18], after multiple cycles, the node's ranking decreases to a reasonable range. However, the early accumulation of transaction data by Node 3 slows down the decrease in its transaction score, resulting in an overall slower decrease in ranking. In the proposed approach in this paper, the speed at which the ranking decreases after a node starts engaging in malicious behavior is significantly higher compared to other approaches. Our approach considers the trust relationships, historical performance, and recent performance of nodes in the calculation of transaction scores, allowing for a more rapid reflection of node performance in the rankings.

Compared to other approaches, our proposed solution not only adapts more quickly to node changes but also effectively addresses coordinated malicious behavior among nodes. In cases where there are changes in node nature or coordinated malicious actions by nodes in the pool of candidates, our approach can provide more valuable reputation rankings for the selection of notaries.

6. Conclusion

This paper proposes a notary evaluation algorithm that can adapt to node state changes in order to address the issues of existing algorithms, which struggle to cope with node state changes and exhibit a bias toward old nodes. Existing solutions often overlook node state changes, allowing malicious nodes to evade algorithm screening by accumulating transaction data before engaging in malicious activities. In this paper, we divide the reputation value into a trust score and a transaction score, which are calculated independently. When calculating the transaction score, we introduce the concept of effective transaction cycles and differentiate between historical and recent data. This approach enables the node's score to reflect its current performance more accurately. Regarding the bias toward old nodes, although previous research has proposed some solutions, they lack flexibility by solely relying on the duration of a node's participation in the notary group to differentiate between new and old nodes. Additionally, these solutions overlook the issue of nodes with fewer cross-chain tasks that have a nature more similar to new nodes. In this paper, we address this problem by using the effective transaction value period to distinguish between new and old nodes, providing a more accurate and flexible distinction.

The experimental results show that compared to existing solutions, our proposed solution effectively solves the problem of malicious nodes manipulating their scores by accumulating transaction data in advance. If malicious nodes start to do evil after accumulating a large amount of excellent transaction data in the early stage, the proposed solution in this article can adjust node scores and rankings more quickly compared to existing solutions to screen out malicious nodes. Even if malicious nodes collaborate to cause harm by establishing trust relationships with each other, the proposed solution in this article can still provide effective evaluations for each node. At the same time, for new nodes, by increasing the proportion of transaction scores in reputation value calculation, trusted nodes can obtain better rankings and win the trust of other nodes through excellent transaction performance after joining the notary group, and malicious nodes among them will be quickly identified after starting to do evil. This can effectively reduce the risk of malicious nodes doing evil and improve the reliability of notary nodes, thereby enhancing the reliability of the entire cross-chain system.

Although the notary public mechanism can rely on the notary evaluation algorithm to increase the reliability of the elected notary public, its essence is to rely on some reliable nodes to assist in the completion of cross-chain tasks, which cannot solve the problem of notary public mechanism centralization. How to reduce the impact of centralization on cross-chain transactions will be the next research goal. We also hope that the development of blockchain technology can bring new directions for the development of cross-chain technology.

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

The data that support the findings of this study are not openly available at this time as the data is also form part of an ongoing study.

Author Contribution Statement

Chenhong Xie: Conceptualization, Methodology, Software, Validation, Formal analysis, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Xiaoming Hu:** Conceptualization, Validation, Investigation, Resources, Writing – review and editing, Supervision, Project administration. **Shuangjie Bai:** Investigation, Data curation, Writing – review & editing, Visualization, Supervision. **Yan Liu:** Resources, Writing – review & editing, Visualization, Supervision.

References

[1] Nakamoto, S. (2008). *Bitcoin: A peer-to-peer electronic cash system*. Retrieved from: https://assets.pubpub.org/d8wct41f/31611263538139.pdf

[2] Mahapatro, R. K., Ali, A., & Ramakrishnan, N. (2023). Blockchain segmentation: A storage optimization technique for large data. In *2023 8th International Conference on Communication and Electronics Systems*, 499–504.

[3] Mao, H., Nie, T., Sun, H., Shen, D., & Yu, G. (2022). A survey on cross-chain technology: Challenges, development, and prospect. *IEEE Access*, *11*, 45527–45546.

[4] Chen, J., Yang, H., He, K., Li, K., Jia, M., & Du, R. Y. (2023). Current situation and prospect of blockchain scaling technology [qū kuài liàn kuò zhǎn jì shù xiàn zhuàng yǔ zhǎn wàng]. *Journal of Software*, *35*(2), 828–851.

[5] Ye, J., Pang, C., Li, X., Zhang, X., & Liu, L. (2022). Blockchain-based supply chain data hierarchical access control mechanism [jī yú qū kuài liàn de gōng yìng liàn shù jù fēn jí fǎng wèn kòng zhì jī zhì]. *Journal of University of Electronic Science and Technology of China*, *51*, 408–415.

[6] De Aguiar, E. J., Faiçal, B. S., Krishnamachari, B., & Ueyama, J. (2020). A survey of blockchain-based strategies for healthcare. *ACM Computing Surveys*, *53*(2), 1–27.

[7] Yuan, Y., & Wang, F. Y. (2016). Towards blockchain-based intelligent transportation systems. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems*, 2663–2668.

[8] Han, P., Yan, Z., Ding, W., Fei, S., & Wan, Z. (2023). A survey on cross-chain technologies. *Distributed Ledger Technologies: Research and Practice*, *2*(2), 1–30.

[9] Li, F., Li, Z. R., & Zhao, H. (2019). Research on the progress in cross-chain technology of blockchains. *Journal of Software*, *30*(6), 1649–1660.

[10] Li, G. Z., Li, L. X., & Gao, H. Y. (2024). Cross-chain technology development and application research [kuà liàn jì shù fā zhǎn yǔ yìng yòng yán jiū jìn zhǎn]. *Journal of Computer Engineering & Applications*, *60*(2), 32–45.

[11] Hope-Bailie, A., & Thomas, S. (2016). Interledger: Creating a standard for payments. In *Proceedings of the 25th International Conference Companion on World Wide Web*, 281–282.

[12] Yin, L., Xu, J., & Tang, Q. (2021). Sidechains with fast cross-chain transfers. *IEEE Transactions on Dependable and Secure Computing*, *19*(6), 3925–3940.

[13] Zabka, P., Foerster, K. T., Schmid, S., & Decker, C. (2022). Empirical evaluation of nodes and channels of the lightning network. *Pervasive and Mobile Computing*, *83*, 101584.

[14] Sun, H., Mao, H. Y., Zhang, Y. F., Yu, G., Xu, S. C., & He, G. Y. (2022). Development and application of blockchain cross-chain technology [qū kuài liàn kuà liàn jì shù fā zhǎn jí yìng yòng]. *Computer Science*, *49*(5), 287–295.

[15] Ou, W., Huang, S., Zheng, J., Zhang, Q., Zeng, G., & Han, W. (2022). An overview on cross-chain: Mechanism, platforms, challenges and advances. *Computer Networks*, *218*, 109378.

[16] Ye, X. H., Liu, X. Y., Wang, B. H., & Xing, S. S. (2022). Distributed notary cross-chain model for consortium chain [miàn xiàng lián méng liàn de fēn bù shì gōng zhèng rén kuà liàn mó xíng]. *Journal of Applied Sciences*, *40*(4), 567–582.

[17] Dai, B. R., Jiang, S. M., Li, D. W., & Li, C. (2021). Evaluation model of cross-chain notary mechanism based on improved pagerank algorithm [jī yú gǎi jìn PageRank suàn fǎ de kuà liàn gōng zhèng rén jī zhì píng jià mó xíng]. *Computer Engineering*, *47*(2), 26–31.

[18] Jiang, C. Y., Fang, L. X., Zhang, N., & Zhu, J. M. (2022). Cross-chain interaction safety model based on notary group [jī yú gōng zhèng rén zǔ de kuà liàn jiāo hù ān quán mó xíng]. *Journal of Computer Applications*, *42*(1), 3438–3443.

[19] Cao, L., & Yang, H. (2023). Two-phase election algorithm for cross-chain notaries. In *2023 3rd International Conference on Computer Science and Blockchain*, 53–58.

[20] Chen, L., Yao, Z., Si, X., & Zhang, Q. (2023). Three-stage cross-chain protocol based on notary group. *Electronics*, *12*(13), 2804.

[21] Chen, L., Chen, Y., Tan, C., Dou, H., Luo, Y., & Chen, P. (2023). A multiple indicator credit ranking scheme based on notary mechanism. In *2023 IEEE International Conference on Dependable, Autonomic and Secure Computing, International Conference on Pervasive Intelligence and Computing, International Conference on Cloud and Big Data Computing, International Conference on Cyber Science and Technology Congress*, 0492–0498.

[22] Zhao, S., & Cao, L. (2022). Dynamic notary group election algorithm based on reputation value. In *2022 International Conference on Bigdata Blockchain and Economy Management*, 903–915.

[23] Xiong, A., Liu, G., Zhu, Q., Jing, A., & Loke, S. W. (2022). A notary group-based cross-chain mechanism. *Digital Communications and Networks*, *8*(6), 1059–1067.

[24] Cao, L., Zhao, S., Gao, Z., & Du, X. (2023). Cross-chain data traceability mechanism for cross-domain access. *The Journal of Supercomputing*, *79*(5), 4944–4961.

[25] Duan, L., Sun, Y., Ni, W., Ding, W., Liu, J., & Wang, W. (2023). Attacks against cross-chain systems and defense approaches: A contemporary survey. *IEEE/CAA Journal of Automatica Sinica*, *10*(8), 1647–1667.

[26] Gleich, D. F. (2015). PageRank beyond the web. *SIAM Review*, *57*(3), 321–363.

How to Cite: Xie, C., Hu, X., Bai, S., & Liu, Y. (2025). Notary Evaluation Algorithm Adaptable to Node State Changes for Cross-Chain Notaries. *Journal of Data Science and Intelligent Systems*. https://doi.org/10.47852/bonviewJDSIS52024592