A Novel Cross-Chain Relay Method Based on Node Trust Evaluation

Yafeng Li[1], Wantao Tuo[2], Qiaozu Hu[2], and Lichuan Ma[2(✉)]

[1] China CETC Key Laboratory of Technology on Data Link, Xi'an, China
xxddxdd@yeah.net

[2] Xidian University and Shaanxi Key Laboratory of Blockchain and Secure Computing, Xi'an, China
23151214211@stu.xidian.edu.cn, lcma@xidian.edu.cn

Abstract. With the increasing complexity of blockchain network business requirements, there is a growing demand for interconnection mechanisms among different blockchains, leading to the emergence of cross-chain technology. This article discusses three mainstream cross-chain technologies: the notary mechanism, hash time lock, and side chain/relay mode, and highlights their limitations. In order to address issues such as low efficiency in relay chain reorganization and instability in cross-chain systems within the relay mode involving node turnover, this paper proposes a node trust-based cross-chain relay scheme. The scheme includes the construction of a trust model for blockchain nodes, the design of a weighted random election algorithm for relay nodes, and the development of a complete cross-chain transaction process. Simulation experiments are conducted to demonstrate the performance of the proposed scheme.

Keywords: blockchain · cross-chain relay · node trust evaluation

1 Introduction

Although blockchain technology has made significant advancements in recent years and some blockchain projects have gained traction, the overall development of current blockchain technology reveals that each blockchain remains a closed and independent system. This characteristic makes individual blockchains susceptible to forming value islands [1]. With the increasing application of blockchain technology across various fields, the business requirements of blockchain networks in different scenarios have become more complex. This complexity has driven a growing need for interconnection mechanisms between diverse blockchains, thereby promoting the emergence of cross-chain technology.

Vitalik Buterin, the founder of Ethereum, has provided a significant summary of cross-chain technology [2]. He also analyzed the development of cross-chain solutions and proposed that mainstream cross-chain technology can be primarily categorized into three types based on implementation principles: notary mechanism, hash time lock, and side chain/relay mode. However, upon examining

J. Liu et al. (Eds.): TridentCom 2023, LNICST 523, pp. 3–20, 2024.
https://doi.org/10.1007/978-3-031-51399-2_1

the current research landscape, it becomes evident that these three technologies each have their inherent limitations.

Aiming to address the issues of low efficiency in relay chain reorganization and the instability of cross-chain systems stemming from node turnover and reorganization, this paper proposes a node trust-based periodic turnover relay cross-chain scheme. The scheme establishes a blockchain node trust model to calculate the trust value of each node in the cross-chain system. This trust value is utilized to design the election algorithm for the relay chain, ensuring the security and efficiency of the algorithm within the cross-chain system. Utilizing the relay model, this scheme attains cross-chain functionality and encompasses various design aspects. The main contributions of this paper can be summarized as follows:

1. The trust model of blockchain nodes in the cross-chain system is constructed, and the trustworthiness of the nodes is derived through quantitative analysis of their various behaviors. Based on this model, a classification standard for node types is formulated. The trust model assists in completing the formation and governance cycle of the relay chain, effectively constraining the relay nodes, and significantly improving the stability of the cross-chain system.
2. Based on the trust model, a weighted random election algorithm for relay nodes has been designed. The algorithm incorporates cryptographic techniques, including encryption, signature, zero-knowledge proof, etc. It utilizes the trust value of the nodes as the basis for the election process. By integrating verifiable random functions, nodes with higher trust values have an increased probability of being randomly selected. This algorithm ensures both randomness and protection against various attacks.
3. A comprehensive cross-chain transaction process has been developed, which encompasses a shared cross-chain message transmission protocol, a cross-chain transaction processing flow, and a mechanism for handling exceptions. The integrity and atomicity of cross-chain transactions are ensured through the careful design of the transmission protocol and the implementation of process controls.
4. Extensive experiments are undertaken to validate the effectiveness and efficiency of the proposed method.

The rest of this paper is organized as follows. Related work is summarized in Sect. 2. The system model is presented in Sect. 3. In Sect. 4, the novel cross-chain relay method based on node trust evaluation is derived. Experimental results are provided in Sect. 5 to show the superior performance of the proposed method and Sect. 6 concludes the paper.

2 Related Works

So far, various schemes have been proposed to support interactions among different blockchains. In [3], the Tendermint team proposed Cosmos Network to achieve was proposed to transferring assets between different blockchains, with

cross-chain consensus powered by the Byzantine Consensus [4]. The blockchains in the Cosmos Network communicate with each other using the Inter-Blockchain Communication Protocol (IBC), which is a set of rules and standards that allow for the transfer of value and data between these separate blockchains. The IBC also ensures security during the transfer of assets. Cosmos serves as a relay chain to facilitate interactions between blockchains by exchanging information, and its emergence has inspired many developers, leading to the design of cross-chain platforms

After the emergence of Cosmos Network, the Polkado whitepaper was released in [5] at the end of 2016. Unlike Cosmos, Polkadot is suitable for cross-chain operations in a broader range of scenarios. Polkadot acts as a relay chain to facilitate transaction transfer and consensus among parallel chains. Polkadot's governance is based on the Proof of Stake protocol, with the main goal of ensuring that a majority of the stakes can always control the network [6].

In [7], sidechain technology and hash-locking are combined to build a new blockchain as a third-party transaction platform. This platform can be a public or private chain used to record transaction credentials, and the scheme ensures the transfer of trust between different blockchains. The authors of [8] propose a new protocol for atomic cross-chain exchanges. This protocol extends previous results in cross-chain interaction to support atomic exchanges to blockchains without hash time-locking functionality, enabling asset transactions between blockchains with only multi-signature functionality. In [9], a cross-chain transaction model based on the combination of notary and hash locking is proposed to address security issues in traditional hash locking. This scheme can prevent malicious participants from creating large traffic that blocks the channel based on the key of the unlocking condition. Additionally, a notary multi-signature scheme is designed to solve the problem of trust in the traditional model.

The authors of [10] design an addressable storage model based on the relay model, reducing the operational cost of existing relay solutions and outlining how relays can be utilized to achieve blockchain interoperability. In [11], a message relay scheme for heterogeneous blockchains is put forward via the periodic turnover of cross-chain nodes. This scheme helps in delivering messages between different blockchains by periodically forming a cross-chain relay chain committee. Recently, a relay system for cross-chain energy trading has been designed in [12]. This system addresses the trading problem of the power system and enables the trading of the same kind of energy within the chain and heterogeneous energies across different chains. a cross-blockchain asset transfer protocol is designed in [13] to support the transfer of arbitrary assets while adhering to the global consistency requirement and improving flexibility in asset transfer. In [14], the authors put forward EOVPC, a cross-chain transaction processing algorithm. This algorithm achieves the atomicity of cross-chain transactions with an optimistic cross-chain consensus method. A comprehensive cross-chain interaction system is proposed in [15] to allow assets on different chains to be mapped to corresponding assets on proxy chains for cross-chain asset transactions. After this work, a decentralized cross-chain data integrity verification scheme is designed

by the authors of [16] to achieve secure and accurate cross-chain data sharing between different blockchains. The focus of this scheme is to address the data integrity verification problem in cross-chain interaction.

3 System Model

In this section, the underlying system architecture is first given where multiple blockchains coexist and interactions frequently occur among them. Given this architecture, useful parameters and definitions are then offered.

3.1 System Architecture

The overall architecture is shown in Fig. 1. Here, there exist multiple blockchains that are maintained via different nodes. These blockchains interact with each other via the newly introduced component Relay Chain.

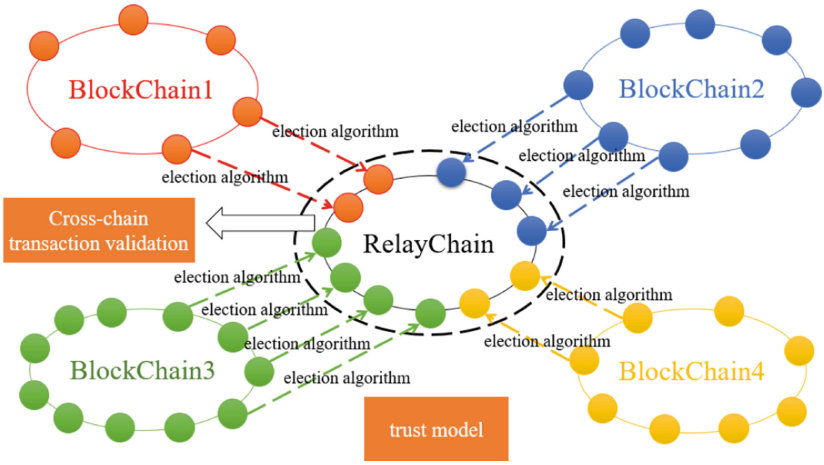

Fig. 1. The system architecture

The whole system works as follows. For each complete node of these different blockchains, a trust value for it is obtained based on its behavior. This value helps to construct the relay chain. After node trust values are obtained, a relay node election algorithm is designed to choose the proper nodes that maintain the relay chain. Once the construction of the relay chain is completed, the rule for cross-chain transactions is put forward to support interoperability and interaction among different blockchains.

3.2 Parameters

In the proposed architecture, we use N^* to denote the set of all nodes in a blockchain denoted as $*$. Each node within the blockchain $*$ is represented as n_i^*, where i refers to the node's position in the blockchain sequence. For example, the set of nodes $\{n_1^*, n_2^*, n_3^*, \dots\}$ is connected through P2P communication, forming a blockchain network that jointly maintains the same blockchain ledger. The notation $N^* = \{n_1^*, n_2^*, n_3^*, \dots\}$ is used to express this relationship. Within the blockchain N^*, multiple transaction information exists. These transactions are defined as $Tx^*(i, j)$ in this paper, representing the j-th transaction data of the i-th block generated by the blockchain denoted as $*$. The cross-chain scheme proposed in this paper not only requires the establishment of a cross-chain blockchain collection, but also involves the division of trust values among all blockchain nodes. Based on their respective trust values, these nodes are assigned to different trust domains. Hence, we assume a collection of trust domains with varying levels of trust, denoted as $\{T_1, T_2, T_3, \cdots\}$. Each blockchain node is assigned to its corresponding trust domain based on its own trust value. Within the cross-chain system, nodes belonging to different trust domains possess different privileges.

Meanwhile, in the cross-chain system, there exist multiple collections of blockchains. In order to facilitate cross-chain transactions between different blockchains, a relay chain is required to verify such transactions. In this paper, the relay chain is denoted as R. According to the cross-chain model proposed in this paper, the relay chain is generated by the nodes within each blockchain utilizing an election algorithm. Therefore, this paper defines the equation as follows (3-1):

$$N^1 \ominus N^2 \ominus N^3 \ominus \cdots = R \tag{3-1}$$

where $\ominus$ represents the selection of nodes for the relay chain, and Eq. (3-1) denotes the set of nodes that are jointly elected among the participating blockchains in the cross-chain process to produce the relay blockchain R. Additionally, the relay chain R needs to run its own cross-chain consensus in order to facilitate the verification of cross-chain transactions.

Using the above parameters, the interaction between any two nodes from the same blockchain or different blockchains can be denoted in a more strict manner. Suppose that nodes n_1^1 and n_2^2 belong to different blockchains, i.e., $n_1^1 \in N_1$, $n_2^2 \in N_2$. If n_1^1 can interact with n_2^2 via R to modify the ledger data of n_2^2 and generate transaction $Tx^2(2, j)$, then $n_1^1 \rightarrow n_2^2$ is used to indicate the interaction. Similarly, if n_2^2 can interact with n_1^1 via R to modify the ledger data of n_1^1 and generate transaction $Tx^1(1, j)$, then $n_2^2 \rightarrow n_1^1$ is used to indicate the interaction. Therefore, blockchain cross-chain interactions are denoted as $n_1^1 \leftrightarrow n_2^2$, and the relay reward is allocated after the completion of the cross-chain transaction, thus completing the entire cross-chain operation. Hence, the blockchain cross-chain interaction is represented as $\Psi(i)$, and the system incentive is allocated accordingly once the cross-chain transaction is finalized, thereby completing the entire cross-chain operation.

4 The Proposed Cross-Chain Relay Method

In this section, the details of the proposed cross-chain relay method are presented. This method is composed of three main components: cross-chain node trust model, weighted randomized relay node election, and cross-chain transaction construction. At first, the trust values of enrolling relay nodes are evaluated via the trust model, where the factors of communication reliability, service time degree, transaction credibility, and historical trust values are considered. After the relay node trust values are computed, a weighted randomized relay node election algorithm is put forward to balance randomness, fairness, security, and efficiency. Subsequently, how cross-chain transactions are completed is illustrated. Moreover, some special cases, like system initialization, transaction exception, and blockchain joining and exiting, are taken into consideration.

4.1 Cross-Chain Node Trust Model

At a high level, the node trust model is composed of four factors, namely communication reliability, service time, transaction credibility, and historical trust values.

(1) **Communication reliability**

Ideally, the system aims for the nodes to maintain a continuous state of normal communication. To quantify the communication reliability mathematically, Eq. (4-1) is designed:

$$T_{com} = \frac{t_{normal}}{t_{total}} \tag{4-1}$$

where T_{com} represents the quantitative result of a node's communication reliability, t_{normal} denotes the time when the node is able to communicate normally with other nodes in the cross-chain system throughout the entire cycle, and t_{total} represents the total cycle time.

However, node communication is affected by multiple factors, such as fluctuations in the transmission channel during communication and potential interference from noise or other uncontrollable situations. In this manuscript, we aim to enhance the fault tolerance of nodes by implementing threshold control on communication reliability. Specifically, we set the upper limit of the node communication threshold as θ_1 and the lower limit as θ_2. If the communication reliability of a node exceeds θ_1, it indicates that the node's communication is problem-free. Conversely, if it falls below θ_2, it suggests that the node's communication cannot be stable and thus this node is not reliable when fulfilling cross-chain relay tasks. Equation (4-2) provides a detailed description of this concept.

$$T_{com} = \begin{cases} 1 & \frac{t_{normal}}{t_t otal} \geq \theta_1 \\ \frac{t_{normal}}{t_{total}} & \theta_2 < \frac{t_{normal}}{t_{total}} < \theta_1 \\ 0 & \frac{t_{normal}}{t_{total}} \leq \theta_2 \end{cases} \tag{4-2}$$

(2) **Service time degree**

The second factor that affects the trust value of a relay node is the service time that undertaking relay tasks. Let T_{time} denote the influence of this factor. It can be computed via Eq. (4-3):

$$T_{time} = e^{-\frac{a}{t}} \tag{4-3}$$

Here, t represents the time experienced by the node after joining the cross-chain system. The parameter $a(a > 0)$ is the time regulation factor, which can regulate the growth rate of T_{time} by adjusting the value of a. The larger the value of a, the slower T_{time} grows. Overall, the value of T_{time} increases with the value of a and tends to be 1. This means that as more nodes join the cross-chain system, their service time degree gradually approaches 1, in line with the design expectations.

(3) **Transaction credibility**

In a blockchain system, the transactions of a node serve as an important indicator of the integrity of the node's behavior. All the data within the blockchain is generated through a series of transactions after consensus is reached. Hence, the performance of each node in the transaction process plays a crucial role. This paper introduces the concept of transaction credibility for nodes and provides a mathematical quantification. The transaction credibility is defined as shown in Eq. (4-4):

$$T_{trans} = \rho \cdot b \cdot \left(\frac{tx}{tx_{total}} + \frac{con}{con_{total}} \right) \tag{4-4}$$

where T_{trans} is the quantitative result of node transaction credibility. b is the correlation coefficient and is set to 0.5 to ensure that the value of transaction credibility is in the range of 0 to 1. tx is the number of packaged transactions of the node after joining the system, and tx_{total} is the total number of transactions determined by the whole system. con is the number of times nodes participate in the consensus verification, while con_{total} is the number of times that the system as a whole has been verified by the consensus. ρ is the control factor $(0 \leq \rho \leq 1)$. When nodes participate in transactions to reach a certain number, ρ can become 1. The growth rate of T_{trans} can be controlled by regulating ρ. From Eq. (4-4), it can be seen that nodes actively participating in transaction packaging and consensus will have higher transaction credibility. There is a control factor ρ to ensure that its value does not grow rapidly when the nodes only participate in a small number of transactions after joining the cross-chain system.

(4) **Historical trust values**

When there is a lack of prior knowledge, establishing a trust relationship requires gradual accumulation over time. Therefore, the trust model should include the influence of historical trust values to better reflect the evolution of trust. To achieve this, this manuscript introduces the following design: it records the list of trust values of the node as $\{h_1, h_2, h_3, \cdots\}$, representing the results of

trust value calculations by the node using the trust model in previous instances. To better capture trustworthiness fluctuations, this paper defines D as the trust fluctuation value, with specific calculations shown in Eq. (4-5):

$$D = \sqrt{\frac{\sum_{1}^{n}(\bar{h} - h_i)^2}{n}} \tag{4-5}$$

In this formula, n represents the number of recorded node trust values, $\bar{h}$ represents the average value of n trust values and h_i represents the i-th trust value of the node. The formula reflects the fluctuation of the node's trust value, where a larger value of D indicates more drastic changes in the node's trust value.

The trust value may fluctuate greatly during the process of change, either due to malicious nodes or node downtime. However, the proposed method aims to maintain the historical trust value of the node within a certain fluctuation range to ensure node stability and reliability. Consequently, this paper introduces a parameter ξ as a threshold to constrain D. Building upon this, the historical trustworthiness is defined as shown in Eq. (4-6):

$$T_{history} = \begin{cases} \bar{h} & D \leq \xi \\ 0 & D < \xi \end{cases} \tag{4-6}$$

As a result, the current trust value of a node is defined as $T_{current}$, which represents the combined trustworthiness of the node across multiple dimensions. The specific calculation method is illustrated in Eq. (4-7).

$$T_{current} = \eta(t) \times (w_1 T_{com} + w_2 T_{time} + w_3 T_{trans} + w_4 T_{history}) \tag{4-7}$$

Here, w_i represents the weights assigned to each factor, where the trustworthiness of each dimension should be assigned different weights based on the specific requirements. ξ is the control factor of $T_{current}$. By regulating $\xi(t)$, the growth rate of $T_{current}$ can be controlled, ensuring that the trust value of nodes does not increase too quickly and thereby threatens the security of the entire system. The system calculates the trust value of each node by monitoring every node and disseminates it to all member nodes of the chain through consensus uplinking, after which it is regularly updated and maintained.

Subsequently, we introduce an update cycle denoted as C. When each time a cycle C is completed, the trust value of a node is updated based on its historical trust value and the model computation. This update aims to facilitate the subsequent update of the relay chain nodes. Additionally, since the trust value is utilized for the relay node election process, it is updated accordingly before the turnover of the relay chain cycle. In cases where a node exhibits malicious behavior within the system, its trust value is instantly reset to zero during the next trust value update, and it will not be incremented thereafter.

The trust value of a node, calculated by the trust model presented in this paper, ranges between 0 and 1. To enhance node categorization, this paper

Table 1. Node Trust Domain Distribution Table

Node Trust Domain	Distribution of confidence values
High-trust nodes	$0.8 \leq T_{current} \leq 1$
common node	$0.6 \leq T_{current} < 0.8$
low-trust node	$0 < T_{current} < 0.6$
evil nexus	$T_{current} = 0$

divides the nodes into various trust domains based on their trust values. Please refer to Table 1 for further details.

Nodes are assigned to different trust domains based on their trust values, which are categorized as high-trust nodes, ordinary nodes, low-trust nodes, and evil nodes. These trust values directly influence the subsequent election of relay chain nodes. To ensure the overall system security, this paper imposes restrictions on the privileges of nodes within different trust domains. Evil nodes are prohibited from participating in the election process of the relay chain and are only allowed to exist in the source blockchain. Low-trust nodes have a low probability of being elected as relay chain nodes, and even if elected, they can only serve as observer nodes within the relay chain. They are not allowed to participate in the voting session of cross-chain consensus and cannot be elected as master nodes of cross-chain consensus. Both ordinary nodes and high-trust nodes have the ability to participate in the cross-chain consensus within the relay chain. Therefore, all nodes are expected to maintain positive behaviors and strive to improve their trust values.

4.2 Weighted Randomized Relay Node Election Algorithm

Here, we propose a new election concept: the probability of a node becoming a relay node is directly proportional to its trust value. Building upon this concept, we outline the following objectives for the election algorithm to ensure its effectiveness:

1. Randomness: the outcome of a single election cannot be predicted;
2. Fairness: in the long run, the probability of a node being selected should be consistent with the proportion of trust value it has. The higher the trust value, the higher the probability of being selected;
   The above are the hard metrics that need to be met, and here are the performance metrics:
3. Security: Security means that the system can operate normally in the presence of malicious nodes and withstand external attacks;
4. Efficiency: Efficiency means that the algorithm calculates the final result quickly. In other words, the algorithm can swiftly select the relay chain nodes without wasting computational power.

Then, the system determines a threshold value τ, which determines the total number of nodes expected to be elected by the algorithm in this round.

A network-wide recognized random number s is generated and broadcasted throughout the cross-chain network. The node generates the VRF function proof based on the broadcasted random number s and its own private key sk as shown in Eq. (4-8), which is implemented by performing the signature operation on the combined number generated by s and sk through the ECDSA algorithm:

$$proof = VRF\_Proof(sk, s) \tag{4-8}$$

The node will generate the proof of the VRF function through the hash function for mapping calculation, the output has a random nature represented by the value $random$, as shown in Eq. (4-9). Specifically, in this paper, the hash function SHA256 is used for implementation:

$$random = Hash(proof) \tag{4-9}$$

The node normalizes the generated random number $random$ to generate a random number n in the range of 0 to 1 as shown in Eq. (4-10):

$$n = \frac{random}{\alpha} \tag{4-10}$$

where α is 2^{256}, which is also the maximum value of the hash map.

The node calculates the probability of not being selected in this round based on the trust value as shown in Eq. (4-11):

$$p = B(k, T, \omega) = C_T^k \cdot \omega^k \cdot (1 - \omega)^{T-k} \tag{4-11}$$

Here, $B(\cdot)$ is the binomial distribution formula, C_T^k is the combination coefficient, T represents the node trust value (rounded to facilitate calculation, after uniformly expanding it by 100), ω denotes the weight, given by $\omega = \tau/T_1$, where τ is the threshold value determined by the system in step 1, and T_1 is the sum of trust values of all nodes. Since the calculation assesses the probability of not being selected, k is set to 0. In the formula, as the node's trust value increases, the value of p decreases, resulting in a higher probability of being selected.

Each node compares the random numbers above to determine whether it can be elected as a relay node for the next round, as shown in Eq. (4-12):

$$\begin{cases} n \in [p, 1] & electionsuccess \\ n \in [0, p) & electionfailure \end{cases} \tag{4-12}$$

Finally, the node that determines that its election is successful broadcasts the result of the computation across the network, containing $proof$, $random$, and n. The other nodes validate the received message to verify whether the random number generated by the successfully elected node is legitimate or not, in order to determine whether the node is selected or not. This process is depicted in Eqs. (4-13) and (4-14):

$$result\_verify = VRF\_P2H(proof) \tag{4-13}$$

$$verify = VRF\_Verify(proof, random, n, pk) \tag{4-14}$$

The above description presents a single election process. However, based on the previous analysis, it can be inferred that if the nodes of the relay chain remain fixed, it is challenging to eliminate both the dormant nodes and the malicious nodes. Additionally, restraining the power of the relay chain nodes becomes difficult. Consequently, if this state persists for an extended period, the entire cross-chain system will encounter significant security issues. To address this, the relay chain in this paper will undergo a re-election process based on specific rules to eliminate abnormal nodes from the relay chain.

The re-election of the relay chain can guarantee the security of the chain. However, if re-elections occur too frequently, it can create substantial overhead, even with an efficient election algorithm. Therefore, it is necessary to specify the rules for re-election. This paper defines the specific rules for the reorganization of the relay chain:

1. The system sets the time period λ and reorganizes the cycle every time λ passes;
2. Whenever a new blockchain is added to the cross-chain system, reorganize;
3. Reorganize whenever the cross-chain system suffers a serious security incident.

4.3 Cross-Chain Transactions

After the relay chain is formed during the election, it can execute cross-chain operations. To ensure the smooth functioning of transactions, it is essential to

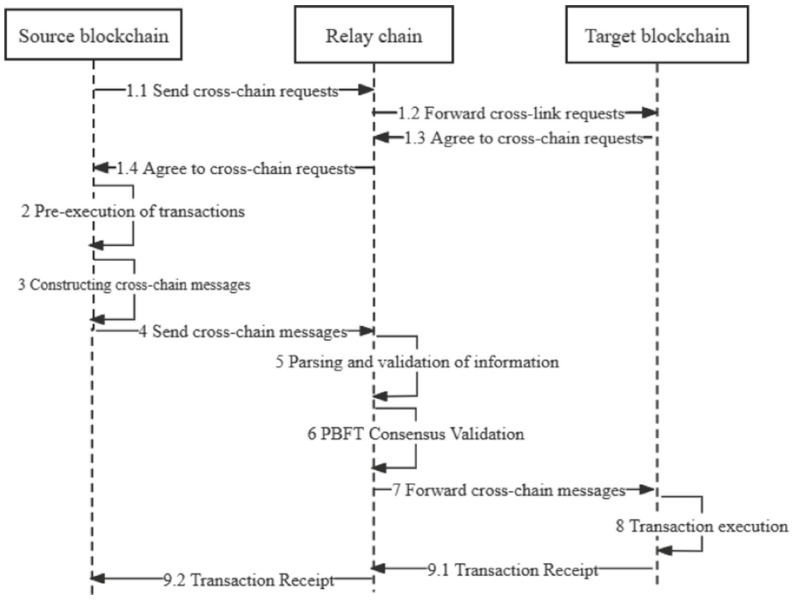

Fig. 2. Cross-chain transaction process

specify the transaction process in detail. The cross-chain transaction process in this program is as follows, Fig. 2:

Step 1: The source blockchain initiates a cross-chain request, and the relay and target blockchains respond to the request;

Step 2: The source blockchain pre-executes the cross-chain transaction and generates the corresponding transaction block;

Step 3: The source blockchain generates the corresponding transmission message according to the cross-chain information transmission protocol and signs the message using the request node in the source blockchain;

Step 4: The source blockchain sends the signed transmission message to the transaction pool of the relay chain and waits for the relay chain to perform the relevant verification process;

Step 5: The relay chain takes the message from the source blockchain out of its own transaction pool, parses it accordingly, and verifies the signature;

Step 6: The relay chain takes out the content of the parsed transaction, performs the PBFT consensus of the relay chain on it, and verifies the existence of the transaction through SPV technology;

Step 7: If the validation passes, it indicates that the transaction in the source blockchain has indeed been executed, then the transmission message is submitted to the target blockchain with the signature of the relay node;

Step 8: After receiving the message from the relay chain, the target blockchain parses and checks the message, and completes the corresponding transaction through its own internal consensus;

Step 9: After the transaction is completed, the corresponding receipt is sent and the whole cross-chain operation is completed.

Following the procedures **Step 1**–**9**, interactions among different blockchains are achieved, and the rewards for nodes fulfilling relay tasks are allocated automatically.

4.4 Countermeasures for Some Special Cases

This section primarily discusses the handling solutions when the system encounters special situations. It is divided into three main parts: how to construct the relay chain during system initialization, how to ensure the atomicity of transactions when exceptions occur, and how to handle the joining or exiting of blockchains in the cross-chain system.

System Initialization. By using the election algorithm, the nodes of the relay chain can be efficiently replaced and reorganized. However, during the initial construction of the relay chain, the system has not yet established connections between the blockchain nodes, and the system has not started running yet. Therefore, it is necessary to specify the system initialization.

Initialization is mainly responsible for creating the first relay chain. It is stipulated that initial participating blockchains each select n nodes to jointly form the initial relay chain. The value of n can be determined based on the

number of participating blockchains and the required number of relay nodes. After the initial relay chain is constructed, connections are established between the blockchains, and the various modules of the system can run. Subsequently, the relay chain is periodically reorganized and replaced according to the relay node election algorithm.

After completing the creation of the first relay chain, the relay chain nodes will synchronously collect block header data from the participating blockchains, facilitating subsequent relay chains to verify cross-chain transactions through SPV technology. The main purpose of the first relay chain is to establish the initial connection between the blockchains and does not engage in cross-chain consensus. After this, the periodic construction of the relay chain begins, and the system operates normally.

Handling Transaction Exceptions. Transaction exceptions can be categorized into transactions not executed on the source chain and transactions not executed on the target chain.

The relay chain can verify the existence of transactions on the source blockchain through PBFT consensus, the Proof field in cross-chain messages, and SPV verification. If the verification fails, it means that the source blockchain did not execute its own transactions. In this case, the relay chain will reject the request from the source blockchain, and the entire cross-chain transaction is terminated. If the source blockchain wishes to continue the transaction, it needs to initiate a new cross-chain transaction and send the new cross-chain transaction information to the relay chain for verification based on the cross-chain message transmission protocol.

If the relay chain verifies the transactions on the source blockchain and the target blockchain does not execute the final cross-chain transaction within a certain period, the relay chain will notify the source blockchain to cancel the cross-chain transaction. The source blockchain will roll back the pre-executed transactions and execute the transactions in reverse order to ensure the atomicity of the cross-chain transaction. As a result, the entire cross-chain transaction is terminated.

Blockchain Joining and Exiting. When a blockchain wants to join the cross-chain system, it needs to submit a cross-chain joining request. Then, it must undergo consensus among the current relay chain, and only when consensus is reached, it will be eligible to join the cross-chain system. After the joining process is completed, a new round of relay chain election will commence for the entire cross-chain system.

Similarly, as the cross-chain system operates, some blockchains may no longer have the need for cross-chain transactions and want to exit the cross-chain system. In that case, they can submit an exit request to the relay chain. After consensus among the relay chain, the blockchain will be excluded in the next relay chain reorganization. However, if a blockchain encounters significant security issues, such as generating denial of service or malicious behavior, the relay

chain will immediately perform a reorganization and exclude the blockchain from the entire cross-chain system to ensure system security.

5 Experiments

This experiment evaluates the performance of the scheme by creating multiple processes, each simulating a node of the blockchain. Experimental settings are listed in the following table (Table 2).

Table 2. Experimental Settings

Setting	Parameters
System	Windows 10 and Ubuntu 16.04
CPU	Intel Core i7-10750H
GPU	GTX 1650
RAM	16 GB
Tool	IntelliJ IDEA(2020.2.3)
Language	Java (JDK 8)

5.1 Trust Model Testing

In order to facilitate a more accurate comparison, the trust value changes of three types of nodes are tested: normal nodes, downtime nodes, and evil nodes. Normal nodes actively participate in cross-chain related operations upon joining the cross-chain system. They remain connected to other nodes and do not engage in any malicious behavior. Downtime nodes, on the other hand, exhibit periods of inactivity after joining the cross-chain system, ceasing their interactions with the cross-chain system. Evil nodes, after a certain period of time within the cross-chain system, deliberately send incorrect information and engage in malicious behavior. The variation of trust values for these three types of nodes is depicted in Fig. 3.

As observed in the figure, the trust value of a normal node steadily increases upon joining the cross-chain system through active participation in normal cross-chain behavior. Over time, it transforms into a high-trust node. Conversely, when a downtime node occurs, its trust value gradually declines and does not recover within a given period. Eventually, it becomes a low-trust node. As for the evil node, its trust value drops to 0 following the engagement of malicious behavior, and it remains stagnant without any further increase in trust value.

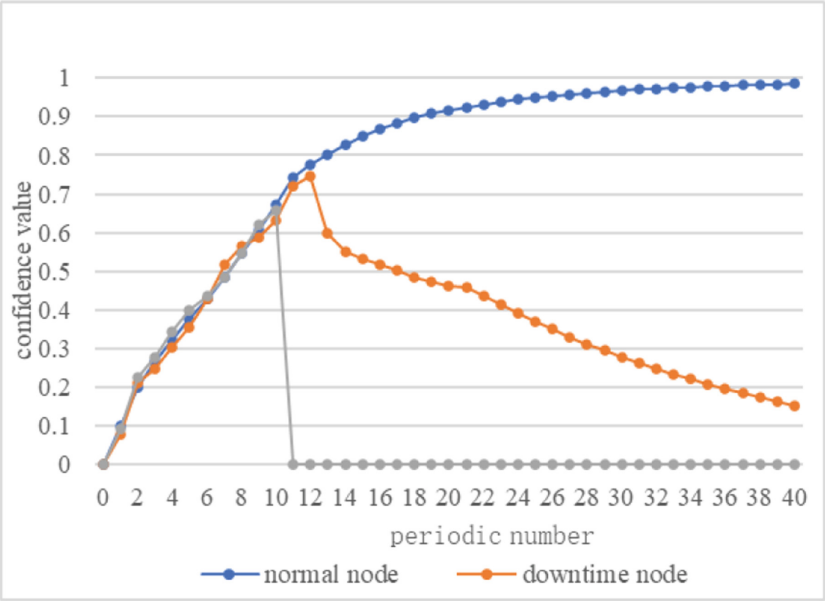

Fig. 3. Node trust value changes

5.2 Algorithm Testing

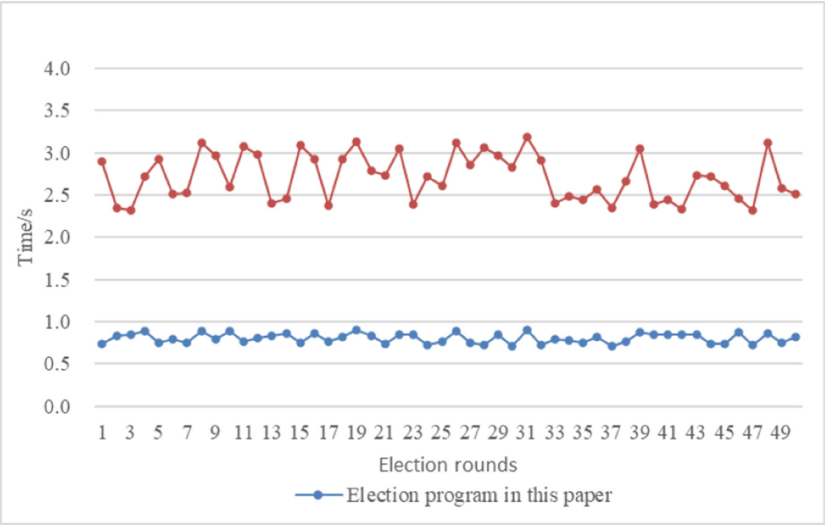

Fig. 4. Algorithm comparison

The election algorithm presented in this paper is efficient because it eliminates the need for arithmetic competition among nodes. In this section, we conduct a time test that compares multiple rounds of the election with the relay chain node election algorithm proposed by Wu et al. [17]. The results of the comparison are illustrated in Fig. 4.

From Fig. 4, it is evident that the election algorithm utilized in this scheme has an average round consumption of approximately 0.8 s. In contrast, the election algorithm proposed by Wu et al. relies on the arithmetic competition of PoW and has an average round consumption of approximately 2.7 s. The experiments demonstrate that the election algorithm employed in this scheme significantly reduces arithmetic consumption, shortens the time required for electing the relay chain, and facilitates the formation of a fast-cycle relay chain. Overall, the algorithm exhibits high efficiency.

5.3 Cross-Chain Transaction Testing

In this subsection, cross-chain transactions are tested. During the testing process, the number of nodes in the relay chain varies, and three types of node numbers 8, 16, and 32 are used to test the consensus performance of cross-chain transactions. The main focus is on testing the time taken for a transaction to be successfully submitted to the relay chain and then verified by cross-chain consensus. The experimental results are shown in Fig. 5. From Fig. 5, it can be observed that the time delay increases as the number of relay chain nodes increases. This is a normal phenomenon due to message broadcasting between the PBFT con-

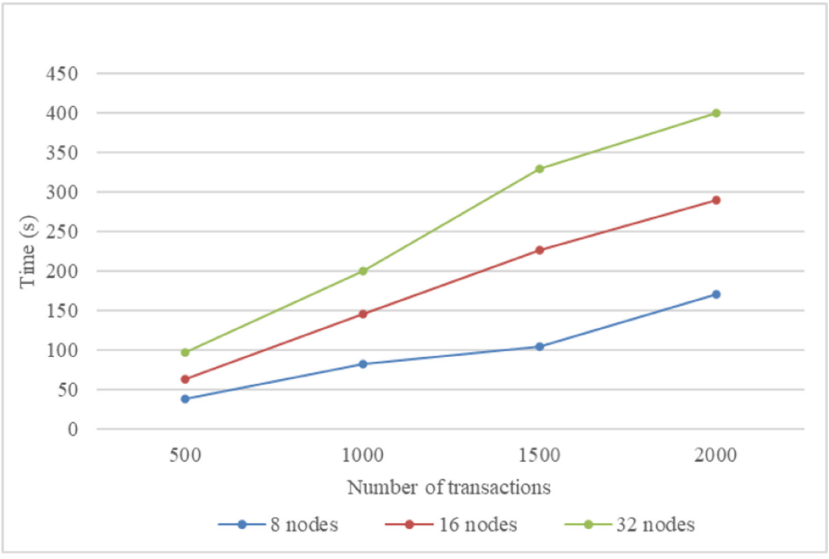

Fig. 5. Cross-chain transaction testing

sensus nodes. Additionally, when the number of nodes is fixed, the transaction processing time increases with a higher number of cross-chain transactions.

6 Conclusion

In this paper, a node trust-based cycle turnover relay cross-chain scheme is proposed to support cross-chain interaction between different blockchains. The trust model of blockchain nodes in the cross-chain system is first constructed, and the trust value of each node is calculated by evaluating multiple dimensions. Based on this evaluation, nodes are categorized into evil nodes, low-trust nodes, ordinary nodes, and high-trust nodes. This categorization effectively constrains node behaviors in the system and improves the stability of the cross-chain system. Furthermore, to tackle the challenging problem of relay chain formation, the paper introduces a weighted random election algorithm based on node trust. This algorithm, which incorporates VRF technology, enables safe and efficient turnover and reorganization of the relay chain. It ensures randomness while also offering resistance against various attacks. Additionally, the paper specifies the process of cross-chain transactions and guarantees the atomicity of such transactions through SPV verification technology. Finally, the proposed scheme is extensively evaluated through numerous simulation experiments, and the experimental results demonstrate the good performance of the various modules in the scheme.

Acknowledgement. This work is supported by the Key Research and Development Programs of Shaanxi under Grants 2021ZDLGY06-03 and the National Natural Science Foundation of China under Grant 62132013 and 61902292.

References

1. Zhang, J., Liu, Y., Zhang, Z.: Research on cross-chain technology architecture system based on blockchain. In: Liang, Q., Wang, W., Liu, X., Na, Z., Jia, M., Zhang, B. (eds.) CSPS 2019. LNEE, vol. 571, pp. 2609–2617. Springer, Singapore (2020). https://doi.org/10.1007/978-981-13-9409-6_318
2. Buterin, V.: Chain interoperability. R3 Res. Pap. **9**, 1–25 (2016)
3. Kwon, J., Buchman, E.: Cosmos whitepaper. http://cosmos.network/resources/whitepaper (2019)
4. Assiri, B., Khan, W.: Enhanced and lock-free tendermint blockchain protocol. In: IEEE International Conference on Smart Internet of Things, pp. 220–226 (2019)
5. Polkadot, W.G.: Vision for a heterogeneous multi-chain framework. White Pap. (2016)
6. Abbas, H., Caprolu, M., Di Pietro, R.: Analysis of polkadot: architecture, internals, and contradictions. IEEE Int. Conf. Blockchain **2022**, 61–70 (2022)
7. Deng, L., Chen, H., Zeng, J., Zhang, L.-J.: Research on cross-chain technology based on sidechain and hash-locking. In: Liu, S., Tekinerdogan, B., Aoyama, M., Zhang, L.-J. (eds.) EDGE 2018. LNCS, vol. 10973, pp. 144–151. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-94340-4_12

8. Zie, J.-Y., Deneuville, J.-C., Briffaut, J., Nguyen, B.: Extending atomic cross-chain swaps. In: Pérez-Solà, C., Navarro-Arribas, G., Biryukov, A., Garcia-Alfaro, J. (eds.) DPM/CBT -2019. LNCS, vol. 11737, pp. 219–229. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-31500-9_14
9. Dai, B., Jiang, S., Zhu, M., et al.: Research and implementation of cross-chain transaction model based on improved hash-locking. In: International Conference on Blockchain and Trustworthy Systems, pp. 218–230 (2020)
10. Frauenthaler, P., Sigwart, M., Spanring, C., et al.: Leveraging blockchain relays for cross-chain token transfers. Gas **300**, 600 (2020)
11. Wu, Z., Xiao, Y., Zhou, E., et al.: A solution to data accessibility across heterogeneous blockchains. In: 2020 IEEE 26th International Conference on Parallel and Distributed Systems (ICPADS), Hong Kong, pp. 414–421 (2020)
12. Shaomin, Z., Cong, H.: Model of decentralized cross-chain energy trading for power systems. Global Energy Interconnection **4**(3), 324–334 (2021)
13. Sigwart, M., Frauenthaler, P., Spanring, C., Sober, M., Schulte, S.: Decentralized cross-blockchain asset transfers. In: 2021 Third International Conference on Blockchain Computing and Applications (BCCA), Tartu, Estonia, pp. 34–41 (2021)
14. Wang, W., Zhang, Z., Wang, G., Yuan, Y.: Efficient cross-chain transaction processing on blockchains. Appl. Sci. **12**(9), 4434 (2022)
15. Yiming, H., Dawei, L., Chi, Z., et al.: Practical agentchain: a compatible cross-chain exchange system. Future Gener. Comput. Syst. **130**, 207–218 (2022)
16. Jiang, J., Zhang, Y., Zhu, Y., et al.: DCIV: decentralized cross-chain data integrity verification with blockchain. J. King Saud Univ.-Comput. Inf. Sci. **34**(10), 7988–7999 (2022)
17. Wu, Z, Xiao, Y., Zhou, E., et al.: A solution to data accessibility across heterogeneous blockchains. In: 2020 IEEE 26th International Conference on Parallel and Distributed Systems (ICPADS), Hong Kong, pp. 414–421 (2020)