Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.059497

ARTICLE

Blockchain-Based Trust Model for Inter-Domain Routing

Qiong Yang[1], **Li Ma**[1,2,*], **Sami Ullah**[3], **Shanshan Tu**[1], **Hisham Alasmary**[4] and **Muhammad Waqas**[5,6]

[1]Faculty of Information Technology, Beijing University of Technology, Beijing, 100124, China
[2]School of Information Science and Technology, North China University of Technology, Beijing, 100144, China
[3]Department of Computer Science, Shaheed Benazir Bhutto University, Sheringal, 18050, Pakistan
[4]Department of Computer Science, King Khalid University, Abha, 61421, Saudi Arabia
[5]School of Computing and Mathematical Sciences, Faculty of Engineering and Science, University of Greenwich, London, SE10 9LS, UK
[6]School of Engineering, Edith Cowan University, Perth, WA 6027, Australia

*Corresponding Author: Li Ma. Email: mali@ncut.edu.cn

Received: 09 October 2024; Accepted: 11 December 2024; Published: 06 March 2025

ABSTRACT: Border Gateway Protocol (BGP), as the standard inter-domain routing protocol, is a distance-vector dynamic routing protocol used for exchanging routing information between distributed Autonomous Systems (AS). BGP nodes, communicating in a distributed dynamic environment, face several security challenges, with trust being one of the most important issues in inter-domain routing. Existing research, which performs trust evaluation when exchanging routing information to suppress malicious routing behavior, cannot meet the scalability requirements of BGP nodes. In this paper, we propose a blockchain-based trust model for inter-domain routing. Our model achieves scalability by allowing the master node of an AS alliance to transmit the trust evaluation data of its member nodes to the blockchain. The BGP nodes can expedite the trust evaluation process by accessing a global view of other BGP nodes through the master node of their respective alliance. We incorporate security service evaluation before direct evaluation and indirect recommendations to assess the security services that BGP nodes provide for themselves and prioritize to guarantee their security of routing service. We forward the trust evaluation for neighbor discovery and prioritize the nodes with high trust as neighbor nodes to reduce the malicious exchange routing behavior. We use simulation software to simulate a real BGP environments and employ a comparative experimental research approach to demonstrate the performance evaluation of our trust model. Compared with the classical trust model, our trust model not only saves more storage overhead, but also provides higher security, especially reducing the impact of collusion attacks.

KEYWORDS: Inter-domain routing; BGP security; blockchain; trust model; trust mechanisms; trust evaluation

1 Introduction

BGP is the standard inter-domain routing protocol for the Internet, which connects many ASes, conveys Network Layer reachability information, and establishes routes to different destinations [1,2]. Due to the importance of BGP on the Internet, ensuring its security is essential for the safe and reliable operation of the Internet [3]. Consequently, this unconditional trust mechanism exposes BGP routes to malicious attacks or misconfigurations, triggering security threats such as prefix hijacking, path forging, and route leakage, which can lead to traffic hijacking, traffic redirection, and network disruptions that affect Internet connectivity [4,5]. BGP nodes can be viewed as distributed dynamic network nodes that offer services, where each BGP node can both request services from and provide services to other BGP nodes. BGP nodes face trust challenges in a distributed dynamic environment. Due to the lack of a trust evaluation mechanism for inter-domain routing, any AS experiencing abnormal errors or malicious attacks can influence the behavior and routing decisions of other ASes via BGP. This situation leads to a more significant security threat to the inter-domain routing system [6]. Therefore, a trust evaluation mechanism is needed to assess the trust level of inter-domain routing. Trust evaluation mechanisms can motivate distributed network nodes and inhibit their untrustworthy interactions [7]. Trust evaluation mechanisms have been extensively studied in wireless communications and the Internet of Things [8–11]. Trust is a subjective concept that describes the level of trust a node has in another node [12]. Trust evaluation is a prerequisite for establishing cooperation between nodes [13]. Due to the distributed autonomy inherent in inter-domain routing, researchers have begun to apply trust evaluation mechanisms to constrain the malicious behavior of ASes and improve the security of inter-domain routing.

Existing studies evaluate trust when exchanging routing information to suppress malicious routing behavior. However, these studies are typically based on direct observations and indirect suggestions received by BGP nodes from other BGP nodes, which are poorly scalable and unable to meet the dynamic needs of BGP nodes. A new BGP node may be malicious without prior interaction, yet it is unreasonable for existing studies to assume that a new BGP node has a high initial trust value. Trust needs to be evaluated quickly among BGP nodes, but existing studies cannot share trust data to speed up the trust evaluation process effectively. In addition, existing studies do not consider collusion attacks. Therefore, we face an essential question: how can we realize a scalable trust evaluation solution considering collusion attacks while at the same time achieving fast trust evaluation with no pre-trusted BGP nodes? Our goal is to enhance the security of inter-domain routing, establish a transferable trust relationship, and inhibit the malicious behavior of ASes. The combination of BGP and blockchain utilizes the unique attributes of blockchain's decentralization, tamper-proofing, and traceability to establish a transferable trust relationship between ASes, provide technical support for collaborative work, and provide a new solution to the trust problem of inter-domain routing.

Considering the above facts, this paper proposes a new solution, the Blockchain-Based Trust Model for Inter-Domain Routing (BTMIR). We adopt a distributed AS alliance architecture that utilizes blockchain technology to enable scalable trust evaluation and sharing of trust data without requiring prior trust. The main contributions of this paper are as follows:

1. We propose a new trust evaluation solution called BTMIR, which meets the dynamic demands of BGP nodes and achieves scalability.
2. The BTMIR can form a global view of trust evaluation data, which is propagated and stored in a blockchain, maintained by a distributed AS alliance master node, and accessible from anywhere.
3. We forward trust evaluation for neighbor discovery and prioritize nodes with high trust as neighbor nodes to reduce malicious exchange routing behavior.
4. We incorporate security service evaluation before direct evaluation and indirect recommendations to assess the security services that BGP nodes provide for themselves and prioritize their security of routing service. We also use a random function to randomly select BGP nodes using their trust value weights as indexes to reduce the impact of collusive bad-mouthing attacks and ballot-stuffing attacks.

The rest of the paper is organized as follows. We present related work on inter-domain routing trust evaluation in Section 2. In Section 3, we describe our system architecture. We discuss our trust evaluation scheme in Section 4. In Section 5, we present our experiments. We perform a security analysis in Section 6. Section 7 summarizes the paper and outlines future work.

2 Related Work

This section reviews the inter-domain routing trust evaluation mechanisms relevant to our proposed work. Hu et al. [14] proposed a reputation mechanism for evaluating AS routing behavior based on the effectiveness of the historical routing behavior of an AS. The mechanism consists of multiple ASes collaborating to complete the reputation calculation of the target AS, which can suppress the malicious routing behavior of ASes. However, they base the reputation calculation solely on the counts of affirmative and adverse events from the routing detection results, which do not reflect the behavioral details of the ASes. In addition, they use cluster analysis, which results in nodes far away from the target AS being unable to obtain a comprehensive evaluation. Wang et al. [15] proposed a reputation model for evaluating an AS's trust in source-initiated routing. This reputation model enables ASes to prioritize the originating route announcements of ASes with high reputation values at the source end, which can help suppress prefix hijacking. Nevertheless, the proposed reputation model, which categorizes prefix route announcements into only two types, i.e., legal and illegal, does not accurately reflect routing behavior.

Xia et al. [16] introduced an inter-domain routing trust model for trusted evaluation of the behavior of route announcements. They introduced a trust recommendation mechanism to promote AS participation in trust recommendations and to suppress false prefix route announcements and their propagation. However, they used the number of route announcements that satisfy real prefixes as the basis for calculating the behavioral value of these announcements, which is not practical. Similarly, Chen et al. [17] proposed an inter-domain routing reputation model based on AS collaboration. Their reputation model synthesizes the target AS's current and historical reputation evaluations to dynamically update the reputation evaluation while penalizing continuous anomalous behaviors through a time decay function. This approach can help suppress anomalous route propagation. However, the proposed reputation model has not been evaluated for collusion attacks.

Zhao et al. [18] proposed a reputation-based solution for inter-domain routing. Their solution consists of a reputation evaluation and a reputation-based routing algorithm. The reputation evaluation delineates the behavior of routing nodes in detail. It incorporates feedback mechanisms to reflect both the strengths and weaknesses of the routing nodes, as well as their ability to combat malicious attacks. The routing algorithm, designed based on reputation evaluation, can suppress the propagation of malicious routes. However, the proposed solution has not been evaluated for collusion attacks.

Literature [19] proposed InBlock, a trust model for verifying BGP route source authorization. The scheme uses a blockchain to store address assignments and route origin authorizations, and a third party verifies routes by accessing the blockchain. The scheme requires consensus among all parties to modify the existing prefix address assignment information. Saad et al. [20] proposed a two-layer blockchain model RouteChain. The scheme utilizes blockchain to reach consensus on prefix announcements, enhancing trust among ASes and enabling traceable routing paths. The scheme suppresses false prefixes maintains a consistent view of routing paths, and consensus can be reached quickly among ASes. Li et al. [21] proposed a trust consortium model DeBGP. The scheme divides the ASes into consortiums, and each consortium maintains a local blockchain that accomplishes intra-consortium validation of BGP update messages. Two neighboring consortiums maintain the collaborative blockchain and complete the BGP update message validation between the consortiums. The scheme utilizes the local and collaborative blockchain to transfer trust between ASes.

The above study mainly focuses on BGP neighbor nodes exchanging routing information, specifically by introducing a trust evaluation mechanism in the route announcement or forwarding process. However, it does not address how to select BGP neighbor nodes. The overview of related works is in Table 1. Compared with existing solutions, our BTMIR is mainly used to select BGP neighbors. BGP neighbors play an important role in BGP. Due to the large size of the Internet, traffic from ASes needs to rely on BGP neighbor forwarding to reach the destination network. If an AS forms a neighbor relationship with a malicious AS, it cannot forward AS traffic efficiently, affecting network performance. Therefore, choosing BGP neighbors is important for BGP security.

Table 1: Overview of related works

Proposal	Trust mechanism	Trust approach	Trust computing
[14]	Reputation mechanism	Evaluate historical routing behavior	√
[15]	Reputation model	Evaluate the trust of originated routes	√
[16]	Trust model	Evaluates route announcement behavior	√
[17]	Reputation model	Evaluate the trust of routing behavior	√
[18]	Reputation evaluation	Evaluate routing behavior	√
[19]	Leveraging Blockchain	Storages routing origin authorization	×
[20]	Two-layer blockchain model	Consensus on prefix announcement	×
[21]	Trust consortium model	Validates update messages with blockchain	×

3 System Architecture

This section describes our proposed system architecture's components, threat models and assumptions, which can be outlined below.

3.1 Main Components

We consider a decentralized trust evaluation architecture, as shown in Fig. 1, and contains the following components:

1) AS Alliance (AA)

Distributed trust evaluation can be managed through an AS alliance, where a single AS node must maintain a large amount of evaluation information. This situation increases storage and communication overhead due to repeated computations. Each AS alliance includes a master node that calculates and stores the trust values of its member nodes and is responsible for inter-alliance communication. In this paper, blockchain provides a transferable trust base for trust evaluation, where a master node uploads the trust values of nodes in its alliance to the blockchain. The master node acts as a full node of the blockchain without increasing the computational overhead of the blockchain and can improve scalability. Each BGP node can obtain the trust values of other BGP nodes through the master node of its alliance and request services from them. In Fig. 1, three alliances, AA1, AA2, and AA3, are shown, with AS1, AS2, and AS3 serving as the master nodes of these alliances, respectively.

2) New Node

Routers that can perform BGP operations are called BGP speakers. First, BGP speakers need to establish a peering relationship with their neighboring routers to exchange routing information, which involves discovering neighbors. Routing information will be exchanged once BGP peers are formed by establishing BGP connections. In our architecture, a new node must evaluate its neighboring nodes before establishing peer-to-peer relationships and requesting their services. In Fig. 1, the new node AS10 proposes to request a service from AS8. AS10 first evaluates the trust level of AS8 by requesting AS1 (the master node of AA1) to query and calculate AS8's trust value. If AS8's trust value meets the requirements, AS10 can establish a neighboring relationship with AS8 and request its services.

3) Blockchain

BGP and blockchain are combined to create transferable trust relationships between ASes by leveraging the unique attributes of blockchain, which are decentralized, tamper-proof, and traceable [22]. We utilize blockchain to store trust values related to BGP nodes. The blockchain is maintained by the master node of the AS alliance, which reliably manages the trust values of the BGP nodes. The master node provides the BGP node requesting the service with a global view of the trust evaluations of other BGP nodes. Our architecture combines blockchain technology and the AS alliance to design a scalable and secure trust model that employs Practical Byzantine Fault Tolerance and Proof of Stake consensus mechanisms.

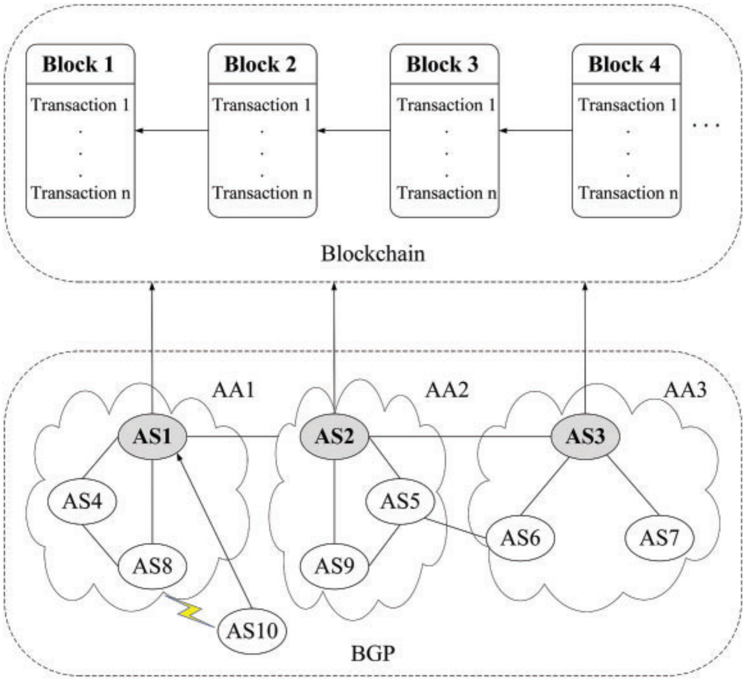


Figure 1: System architecture

3.2 *Threat Model and Assumptions*

Our trust model is built on the blockchain. We assume that the blockchain is secure and do not consider possible attacks on it; we ignore security threats to the blockchain. Since the blockchain is maintained by the master node of the AS alliance, we assume that the master node of the AS alliance is secure. For BGP nodes other than the master node of the AS alliance, we assume that there are dishonest BGP nodes who may maliciously attack other BGP nodes for their benefit so that other BGP nodes select them to provide services for them. Trust evaluation is at risk of dishonest evaluation or malicious attacks [23]. We consider bad-mouthing attacks and ballot-stuffing attacks, which are two forms of collusive attacks that undermine the trustworthiness of good nodes and boost the trustworthiness of malicious nodes [24]. A bad-mouthing attack occurs when a malicious BGP node gives bad advice to trusted BGP nodes, undermining their trust and reducing their chances of being selected to provide services. A ballot-stuffing attack happens when a malicious BGP node offers positive suggestions to untrustworthy BGP nodes, boosting their trust and increasing their chances of being selected for service provision.

4 Our Trust Evaluation Scheme

In this section, we define the main steps of the trust model, and we allow BGP nodes in the architecture to evaluate the trustworthiness of other BGP nodes.

4.1 Our Trust Model

We define BGP as an undirected, weighted, and acyclic-connected graph, denoted as $G = < V, E, W >$, where V represents a BGP node, E represents an edge connecting two BGP nodes, and W represents the weight of the edge connecting the two BGP nodes. W consists of two parts, the trust and time weights, denoted as $W = < W_T, W_t >$, where W_T represents the trust weight and W_t represents the time weight. $W_T = < W_B, W_D, W_R >$, where W_B represents the node's own trust weight, W_D is the node's direct evaluation trust weight, and W_R is the node's indirect recommendation trust weight. The main notations used in this paper are shown in Table 2.

Table 2: List of notations

Notation	Description
W_B	Own trust weight
W_D	Direct evaluation trust weight
W_R	Indirect recommendation trust weight
C_s	Number of successful service processing
C_f	Number of service processing failures
${T_i}_B^{S_B}$	Value of trust in own security service
${D_i}_D^{S_D}$	Trust value of direct evaluation
${R_i}_R^{S_R}$	Trust value of indirect recommendation
T_i	Trust value of a node
μ	Trust factor of historical records
Δt	Time interval
$TS\left(S_m\right)$	Trust value of a service
τ	Trust factor
${S_i}_j^{m_1}$	Evaluation of S_{m1} service of V_i by node V_j
${D_i}_j^{S_{D1}}$	The evaluated value of all S_{m1} services of node V_j to V_i
${S_i}_k^{m_2}$	Recommendation of node V_k to node V_j about S_{m2} service of node V_i
${R_i}_k^{S_{R1}}$	Recommendation value of node V_k for all S_{m2} Services of node V_j about node V_i

We abstractly define each interaction of the BGP protocol as an event, and the content of the event is shown in Table 3. An interaction is an event (which can also be denoted as an interaction context) and is described by the five-tuple of scenario, initiator, receiver, service, and result as follows:

$$C = \{S_c, V_s, V_d, S_r, R_s\}, \tag{1}$$

where C stands for a protocol interaction abstraction (context), S_c is the working scenario of a BGP node, V_s is the initiating requester of an interaction, V_d is the receiver of an interaction, S_r is the service an interaction provides, i.e., request and response. Here, R_s is the processing result of an interaction, i.e., success and failure. In addition, the usual appellations used in trust models are defined as follows:

The trust value of a node T_i: is a real number in the range of $[0, 1]$, representing the trust level of a BGP node V_j with respect to the service provided by node V_i at time t. A maximum value of 1 indicates that the BGP node V_i is fully trusted with respect to node V_j, and 0 indicates that BGP node V_i is a malicious or bad node. Own security service trust value ${T_i}_B^{S_B}$: is a real number in the range of $[0, 1]$, representing the trust level of the service provided by the BGP node V_i itself.

Indirect recommended trust value ${R_i}_R^{S_R}$: is a real number in the range [0, 1], computed by the alliance master node based on the trust values reported by other BGP nodes involving BGP node V_i. This value is sent to the BGP node V_j during the latest time interval Δt.

Direct evaluation trust value ${D_i}_D^{S_D}$: is a real number in the range of $[0, 1]$, representing the satisfaction level of the service provided by node V_i during the interaction between the BGP node V_j and node V_i.

Table 3: Table of events

Message type	Services	Initial values
Open	Establish a session connection	0.2
Update	Route update or revocation	0.25
Keep-alive	Keep-alive	0.2
Route-Refresh	Receive route refresh	0.15
Notification	Error reporting and connection termination	0.2

4.2 Trust Calculations

When a BGP node V_j wants to request S_m service from the node V_i, the choice of node V_i is based on the trust level of that node, i.e., the trust of node V_i. Node V_j evaluates the trust of V_i as follows:

$$T_i(\tau) = \begin{cases} W_B * {T_i}_B^{S_B} + W_D * {D_i}_D^{S_D} + W_R * {R_i}_R^{S_R}, & if\, P(j, i) \\ W_B * {T_i}_B^{S_B} + W_R * {R_i}_R^{S_R}, & otherwise \end{cases}, \tag{2}$$

where $T_i(\tau)$ represents the trust value based on trust weight, ${T_i}_B^{S_B}$ is the trust value of own security service, ${D_i}_D^{S_D}$ is the trust value of direct evaluation, ${R_i}_R^{S_R}$ is the trust value of indirect recommendation, and τ is the trust factor. W_B, W_D, and W_R represent the trust weight of own trust, trust the weight of direct evaluation, and indirect recommendation trust weight, respectively, while $0 \leq W_B, W_D, W_R \leq 1$ and $W_B + W_D + W_R = 1$.

In Eq. (2), $P(j, i)$ represents a judgment: if a node V_j has previously interacted with node V_i, then $P(j, i)$ is true; otherwise, $P(j, i)$ is false. Based on the interaction experience between nodes V_j and V_i, there are two cases as follows:

1. If node V_j and node V_i have interacted previously, evaluate the trust level of the node V_i based on ${T_i}_B^{S_B}$, ${D_i}_D^{S_D}$, and ${R_i}_R^{S_R}$.
2. If node V_j and node V_i have not interacted with each other previously and there is no ${D_i}_D^{S_D}$, the trust of node V_i can be evaluated based on ${T_i}_B^{S_B}$ and ${R_i}_R^{S_R}$ of node V_i.

Algorithm 1 summarizes the different steps of trust computation performed by BGP nodes. As shown in Algorithm 1, the node's security service trust value is first computed and uploaded to the blockchain (see steps 4 to 14 in Algorithm 1). Second, the node's trust value is calculated by combining direct evaluations and indirect recommendations (see steps 15 to 17 in Algorithm 1). Finally, the node's trust value is updated in real-time by combining the node's historical trust value with the current trust value, which is then uploaded to the blockchain (see steps 18 to 20 in Algorithm 1).

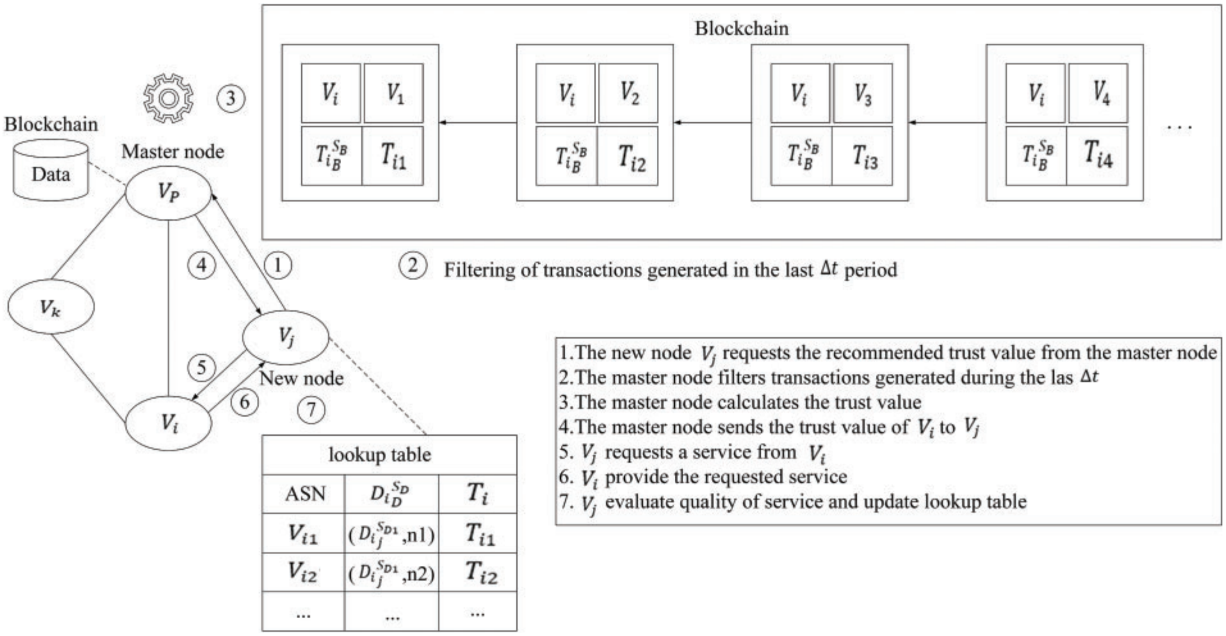
Algorithm 1: Compute node V trust value

```
Input: $N$: node sets, $V_i$: specified node, $V_P$: gateway nodes
Output: $T_i(t)$
1:  procedure Compute Node Trust Value
2:  $S_M \leftarrow \{S_B, S_D, S_R\}; N = \{V_1, V_i \ldots, V_n\}; N_M \leftarrow \{V_j, \ldots, V_n\}; N_P \leftarrow \{V_p\}; {T_i}_B^{S_B}; {D_i}_D^{S_D}; {R_i}_R^{S_R};$
3:  $t_x \leftarrow \{t_1, t_2, \ldots, t_N\}; C_{ms} \leftarrow \{\}; C_{mf} \leftarrow \{\}; W_B; W_D; W_R; \mu;$
4:  ${T_i}_B^{S_B}$ = Compute Base Trust Value()
5:  Send the trust value${T_i}_B^{S_B}$ to the gateway node $V_p$ and transmit it on the blockchain
6:  Obtain the trust value of neighboring nodes $V_j$ on the blockchain through gateway nodes $V_p$
7:  for $N_j \in N_M$ do
8:       Send $S_M$ message to node $V_j$
9:       Record $C = \{S_c, V_s, V_d, S_r, R_s\} \leftarrow \{\text{BGP}, V_i, V_j, S_M, \text{Result}\}$
10:      if $Result = succ$ then $C_{ms} \leftarrow C_{ms} + 1$
11:      else
12:           $C_{mf} \leftarrow C_{mf} + 1$
13:      end if
14: end for
15: ${D_i}_D^{S_D} \leftarrow$ Compute Direct Trust Value()
16: ${R_i}_R^{S_R} \leftarrow$ Compute Indirect Trust Value()
17: $T_i(\tau) \leftarrow W_B * {T_i}_B^{S_B} + W_D * {D_i}_D^{S_D} + W_R * {R_i}_R^{S_R}$
18: $T_i(t) \leftarrow \mu * T_i(\tau)(t - \Delta t) + (1 - \mu) * T_i(\tau)$
19: Send $C$ and $T_i(t)$ as transactions to the gateway node $V_p$ and transmit it on the blockchain
20: return $T_i(t)$
21: end procedure
```

The trust of a node comes from the level of trust in the services provided by the node, i.e., the degree of trust in the node's services. Since a new BGP node has no interaction history, the initial trust value needs to be set to facilitate interaction with other nodes. In this paper, we consider the existence of dishonest BGP nodes, and to reasonably reflect the level of trust in the services provided by the nodes, we set the initial value of the node's service trust level to 0.5. As the quality of node service changes, we use the number of successes and failures of service provided by interacting nodes $\sum_{t_x \in \{t_1,\ldots,t_n\}} \frac{C_s}{C_s + C_f} * (t_x)$ as the evaluation metric to update the trust value of node services. The trust value for the node service is shown below:

$$TS(S_m) = \begin{cases} 0.5, C = \varnothing \\ TS(S_m) * \sum_{t_x \in \{t_1,\ldots,t_n\}} \frac{C_s}{C_s + C_f} * (t_x), C \neq \varnothing \end{cases}, \tag{3}$$

where $TS(S_m)$ represents the trust value of a service, C_s is the number of times the service has been processed successfully, C_f is the number of times the service has been processed unsuccessfully, and t_x is the time series. The detailed steps of our trust model are shown in Fig. 2.


Blockchain
Data
Master node
3
V_P
V_k
V_j
New node
V_i
1
4
5
6
7
Blockchain
V_i V_1 $T_{i_B}^{S_B}$ T_{i1}
V_i V_2 $T_{i_B}^{S_B}$ T_{i2}
V_i V_3 $T_{i_B}^{S_B}$ T_{i3}
V_i V_4 $T_{i_B}^{S_B}$ T_{i4}
...
2 Filtering of transactions generated in the last Δt period

lookup table		
ASN	$D_{i_D}^{S_D}$	T_i
V_{i1}	$(D_{i_j}^{S_{D1}}$,n1)	T_{i1}
V_{i2}	$(D_{i_j}^{S_{D1}}$,n2)	T_{i2}
...	...	...

1.The new node V_j requests the recommended trust value from the master node
2.The master node filters transactions generated during the las Δt
3.The master node calculates the trust value
4.The master node sends the trust value of V_i to V_j
5. V_j requests a service from V_i
6. V_i provide the requested service
7. V_j evaluate quality of service and update lookup table

Figure 2: Main steps of the trust model

4.2.1 Evaluation of Security Service

Existing research implicitly assumes that routing services are secure. However, as a network infrastructure, routing cannot guarantee the routing protocol's secure operation once the routing service is attacked. Therefore, the security of routing services is a prerequisite. In this paper, we consider the issue of the routing node's security service and comprehensively evaluate the security service that the routing node itself possesses. Table 4 shows BGP nodes' security services.

Table 4: Security services

Security service	Service contents	Initial values
Authenticate	Digital signature	0.05
Cryptography	Symmetric encryption, asymmetric encryption	0.1
Access control	Mandatory access control, autonomous access control	0.1
Intrusion detection	Intrusion detection system	0.1
Security reinforcement	Trusted computing	0.15

The own security service trust value is the sum of the trust values of the node's security services, as follows:

$$T_{i_B}^{S_B} = \sum_{b=1}^{B} TS\left(S_b\right), \tag{4}$$

where $T_{i_B}^{S_B}$ represents the trust value of the node's security service and $TS\left(S_b\right)$ is the trust value of the node's security service.

4.2.2 Direct Evaluation

In our trust model, when node V_j requests S_{m1} service from node V_i, node V_j measures the quality of the service provided by node V_i, i.e., node V_j directly evaluates the service provided by node V_i. The different steps involved in direct evaluation are detailed below.

In the first step, a randomization function randomly selects the direct evaluation nodes to defend against bad-mouthing attacks and ballot-stuffing attacks colluded by multiple nodes. We randomly select multiple direct evaluation nodes by using the trust value weights of the direct evaluation nodes as indexes, which are calculated as follows:

$$N_{D1} = WRI(N_D), \tag{5}$$

where N_{D1} represents the number of randomly selected direct evaluation nodes, N_D is the total number of direct evaluation nodes, and $WRI(N_D)$ is the trust value weight of direct evaluation nodes.

In the second step, upon calculating the evaluation value of the randomly selected direct evaluation nodes, the trust value weights of the randomly selected direct evaluation nodes need to be considered comprehensively. We take the ratio of the trust value of a randomly selected direct evaluation node to the sum of the trust values of all direct evaluation nodes as the trust value weight. This ratio is used to adjust the evaluation values of different direct evaluation nodes, which are calculated as follows:

$$D_{i\,j}^{\;S_{D1}} = \sum\nolimits_{m_1=1}^{M} TS\left(S_{i\,j}^{\;m_1}\right) * \frac{T_j}{\sum\nolimits_{n_1=1}^{N_{D1}} T_{n_1}}, \tag{6}$$

where $D_{i\,j}^{\;S_{D1}}$ represents the evaluated value of all S_{m1} services of node V_j to node V_i, $S_{i\,j}^{\;m_1}$ is the evaluated value of the S_{m1} services of node V_j to node V_i, and $\frac{T_j}{\sum_{n_1=1}^{N_{D1}} T_{n_1}}$ is the randomly selected weight of trust value of directly evaluated nodes.

In the third step, when calculating the direct evaluation value, the ratio of randomly selected direct evaluation nodes to the total number of nodes is used to adjust the sum of the evaluation values for all selected nodes, ensuring the reasonableness of the evaluation results. The calculation is as follows:

$$D_{i\,D}^{\;S_D} = \frac{N_{D1}}{N} * \sum\nolimits_{j\in N_{D1}} D_{i\,j}^{\;S_{D1}}, \tag{7}$$

where $D_{i\,D}^{\;S_D}$ represents the direct evaluation trust value and $\frac{N_{D1}}{N}$ is the ratio of the number of all randomly selected direct evaluation nodes to the total number of nodes.

4.2.3 Indirect Recommendation

In our trust model, if node V_j and node V_i have no prior interaction history, when node V_j wants to request service S_{m2} from node V_i, node V_j needs to evaluate the trustworthiness of node V_i before making the service request. Therefore, node V_j first obtains the evaluation value of the node V_i from other BGP nodes that have interaction with node V_i, and then combines the recommendation suggestions of other BGP nodes to measure the trust of node V_i. The different steps involved in indirect recommendations are detailed below.

In the first step, the indirect recommendation nodes are randomly selected by a random function to defend against the bad-mouthing attack, and ballot-stuffing attack colluded by multiple nodes. We randomly select multiple indirect recommendation nodes by using the trust value weights of the indirect recommendation nodes as indexes, which are calculated as follows:

$$N_{R1} = WRI(N_R), \tag{8}$$

where N_{R1} represents the number of randomly selected indirectly recommended nodes, N_R is the total number of indirectly recommended nodes, and $WRI(N_R)$ is the trust value weight of indirectly recommended nodes.

In the second step, when calculating the recommendation value of the randomly selected indirect recommendation node, two parts of the trust value weights need to be considered comprehensively. One part is the proportion of the trust value of a randomly selected indirect recommender node to the sum of the trust values of all randomly selected indirect recommenders, which serves as the trust value weight of that node. The other part is the proportion of the trust value of the randomly selected indirect recommender node to the trust value of the recommended party's node, representing the weight of the trust value of the recommender with respect to the recommended party. We use the trust value weights of these two parts to adjust the recommendation values of different indirect recommendation nodes, which are calculated as follows:

$$R_{i_k}^{S_{R1}} = \sum_{m_2=1}^{M} TS\left(S_{i_k}^{m_2}\right) * \frac{T_k}{\sum_{n_2=1}^{N_{R1}} T_{n_2}} * \frac{T_k}{T_j + T_k}, \tag{9}$$

where $R_{i_k}^{S_{R1}}$ represents the recommendation value of the node V_k to node V_j about all S_{m2} services of node V_i. Here, $S_{i_k}^{m_2}$ is the recommendation of node V_k to node V_j about the S_{m2} services of node V_i, $\frac{T_k}{\sum_{n_2=1}^{N_{R1}} T_{n_2}}$ is the trust value weight of randomly selected trust value weights of the indirect recommender nodes, and $\frac{T_k}{T_j+T_k}$ is the trust value weights of the randomly selected indirect recommender nodes with respect to the recommended parties.

In the third step, when calculating the indirect recommendation value, the ratio of randomly selected indirect recommender nodes to the total number of nodes is used to adjust the sum of the recommendation values of all selected nodes to equalize the recommendation results. The value can be calculated as:

$$R_{i_R}^{S_R} = \frac{N_{R1}}{N} * \sum_{k \in N_R} R_{i_k}^{S_{R1}}, \tag{10}$$

where $R_{i_R}^{S_R}$ represents the indirect recommendation trust value and $\frac{N_{R1}}{N}$ is the ratio of the number of all randomly selected indirectly recommended nodes to the total number of nodes.

4.3 Trust Update

The trust level of a node changes over time, and the trust value of a node needs to be updated in real time to reflect the latest level of trust of the node. In this paper, we consider the time factor and combine the node's historical trust value and current trust value for the weight calculation to update the node's trust value, which can be computed as:

$$T_i = T_i(t), \tag{11}$$

$$T_i(t) = \mu * T_i(\tau)(t - \Delta t) + (1 - \mu) * T_i(\tau), \tag{12}$$

where T_i represents the trust value of the node, $T_i(t)$ is the time weight-based trust value, i.e., the trust value of the node V_i at time t, $T_i(\tau)(t - \Delta t)$ is the most recent trust value of the node V_i, and μ is the trust factor of the history.

5 Performance Evaluation

In this section, we experimentally evaluate the effectiveness of our proposed BTMIR trust model. The main parameters used in our experiments are shown in Table 5.

Table 5: Main parameters

Parameters	Values
W_B	0.2
W_D	0.4
W_R	0.4
μ	0.02

5.1 Experimental Environment and Experimental Design

To validate the effectiveness of our proposed solution, we compare our proposed BTMIR trust model with EigenTrust [25]. EigenTrust is a classical reputation-based trust model used in peer-to-peer networks to enable participants to establish a trust relationship by using eigenvectors to convey trust. EigenTrust can perform trust in large-scale network computation, but it does not work reliably when participants collude with each other.

To evaluate the advantages of our trust model against malicious node attacks, we build a test environment for the BGP node trust model. The test environment is an Intel(R) Core(TM) i5-7200U CPU @ 2.50 GHz with 4 GB of RAM and the Ubuntu 20.04 operating system. We chose the Tendermint framework for the blockchain, which consists of tools that can execute smart contracts to reach consensus and create blocks in a distributed network. We built the application using the blockchain application programming interface provided by Tendermint, which provides access to trusted values. We use Go as the programming language. We use Network Simulator version 2.34 to simulate a real BGP environment and employ a comparative experimental research approach to demonstrate the performance evaluation of our trust model BTMIR. Furthermore, to realize the reasonable distribution of BGP nodes, we select four groups of nodes for testing, and the total number of nodes in each group is 10, 20, 50, and 100, respectively. We divide each group of nodes into three types of nodes: hardened nodes, malicious nodes, and ordinary nodes. Moreover, the number of these three types of nodes in each group accounts for 20%, 20%, and 60%, respectively. We evenly distribute the hardened nodes and malicious nodes into different alliances to reflect the real rationality of the nodes between alliances as well as within alliances. The distribution of nodes in each group is shown in Table 6.

We also tested EigenTrust according to our total number of nodes per group, and the distribution of malicious nodes per group remained consistent with our test. Unlike our tests, EigenTrust has no hardened nodes, only malicious nodes and ordinary nodes in each group, and their share of the number in each group is 20% and 80%, respectively. We focus on testing the proportion and number of malicious nodes that are selected as trustworthy nodes due to their trust changes when malicious attacks occur. For malicious attacks, we focus on bad-mouthing attacks and ballot-stuffing attacks. First, we compare our solution and EigenTrust in terms of effectiveness against collusion attacks. Second, we compare our solution and EigenTrust in terms of the impact on malicious nodes when their trust weights are changed. Next, storage and computation overheads are compared under different network scales. Finally, the robustness of trust models is compared under different attack intensities.

Table 6: Distribution of nodes by group

Group	Group 1	Group 2	Group 3	Group 4
Number of nodes	10	20	50	100
Hardened node No.	1,3	1–2,5–6	1–5,11–15	1–10,21–30
Malicious node No.	2,4	3–4,7–8	6–10,16–20	11–20,31–40
Ordinary node No.	5–10	9–20	21–50	41–100

5.2 Effectiveness Against Collusion Attacks

We focus on two types of collusion attacks: collusive bad-mouthing attacks and collusive ballot-stuffing attacks. In our simulation scenario, a malicious node performs both a collusive bad-mouthing attack and a collusive ballot-stuffing attack.

We conducted 4 sets of comparison experiments according to the previous grouping. For our BTMIR trust model, each group of nodes is a unit. In the first step, the trust values of the nodes are calculated and sorted in descending order. In the second step, the sorted node serial numbers are selected using the random values generated by the random function. In the third step, the neighbor nodes are selected based on the node serial number from the second step. EigenTrust does not have a random selection process. For comparison, the trust values of the nodes in each group are also computed and sorted in descending order, and the number of nodes selected as neighbors is the same as the number of nodes we have randomly selected.

Fig. 3 shows the proportion of malicious nodes that may be selected as neighboring nodes under collusion attacks for our BTMIR trust models and EigenTrust. BTMIR reduces the impact of collusion attacks compared to EigenTrust The percentage of malicious nodes is lower in all four comparison experiment groups compared to EigenTrust, with a significant decrease of 66.67% in group 3 using our solution. This is because a random function is utilized to randomly select nodes in our solution. Thus, our solution reduces the multi-node collusion, bad-mouthing attacks, and ballot-stuffing attacks.

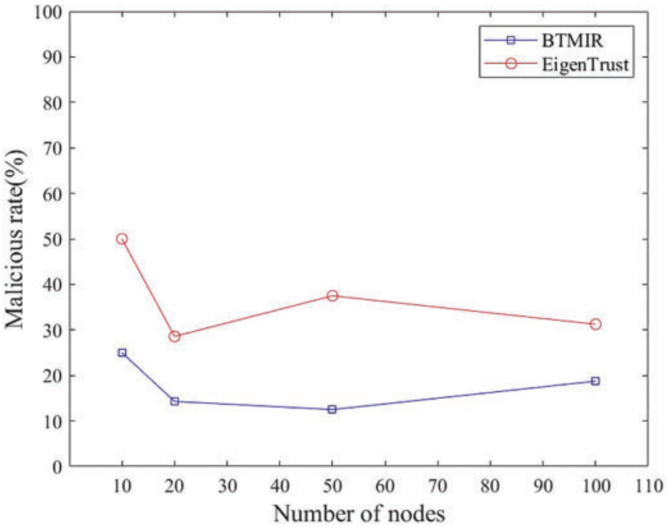


Figure 3: Proportion of malicious nodes selected as neighbor nodes under collusion attack

Fig. 4 shows the number of malicious nodes that may be selected as neighbor nodes in BTMIR and EigenTrust under collusive attacks. We note that the number of malicious nodes appearing in BTMIR has also been reduced compared to EigenTrust. Due to the addition of hardened nodes in BTMIR, the impact caused by some of the malicious nodes is somewhat suppressed.

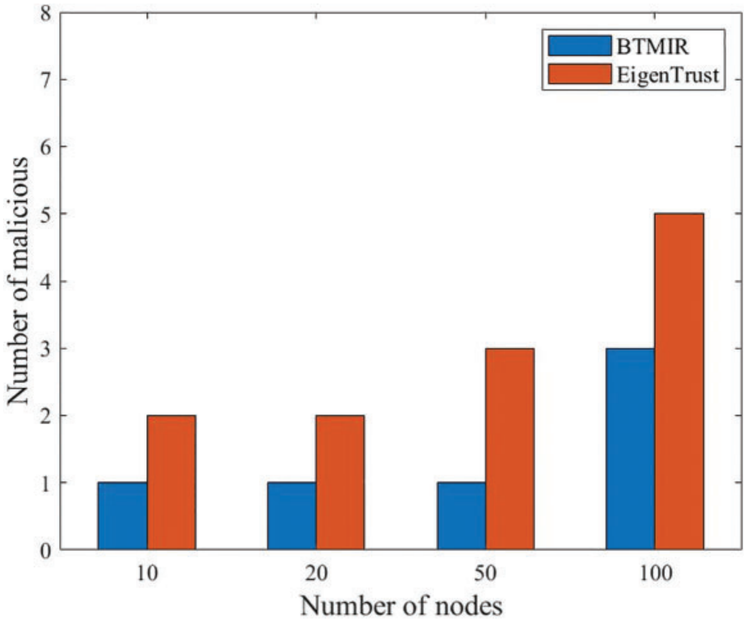


Figure 4: Number of malicious nodes selected as neighbor nodes under collusion attack

5.3 *Effect of Changing Trust Weights on Malicious Nodes*

We vary the own trust weights W_B based on the experiments in the previous section. We tested 100 nodes for the effect of malicious nodes on neighboring nodes under different W_B. We set W_B to 0.1, 0.2, 0.3, 0.4 and 0.5.

Fig. 5 shows the proportion of malicious nodes that may be selected as neighbor nodes for the proposed BTMIR and EigenTrust with different own trust weights W_B. BTMIR reduces the proportion of malicious nodes that could be neighbor nodes compared to EigenTrust. For both trust weights $W_B = 0.3$ and $W_B = 0.4$, our solution achieves a 60% reduction compared to EigenTrust. Fig. 6 shows the number of malicious nodes that may be selected as neighbor nodes for BTMIR and EigenTrust with different trust weights W_B. BTMIR reduces the number of malicious nodes that may become neighbor nodes compared to EigenTrust. Thus, our solution can somewhat suppress malicious nodes' influence on neighboring nodes.

5.4 *Storage and Computation Overheads for Different Network Scales*

The storage and computation overheads of our trust model BTMIR and EigenTrust under different network scales are shown in Table 7. Our trust model produces lower storage overhead at different network scales than EigenTrust. As the network scales become more considerable, our trust model saves more storage overhead and is suitable for BGP environments. In addition, although the computational overhead of our trust model is slightly higher than that of EigenTrust, our solution provides higher security. The time complexity of our algorithm is $O(\log n)$.

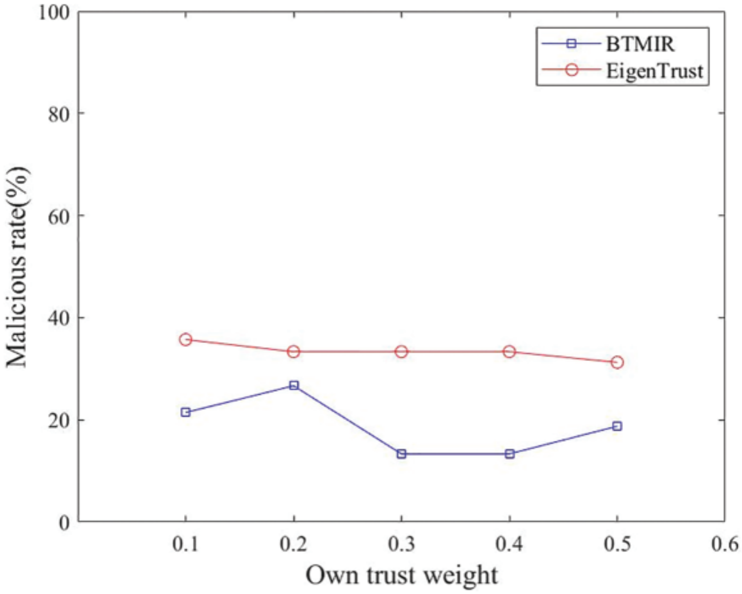


Figure 5: Proportion of malicious nodes selected as neighbor nodes under changing own trust weights

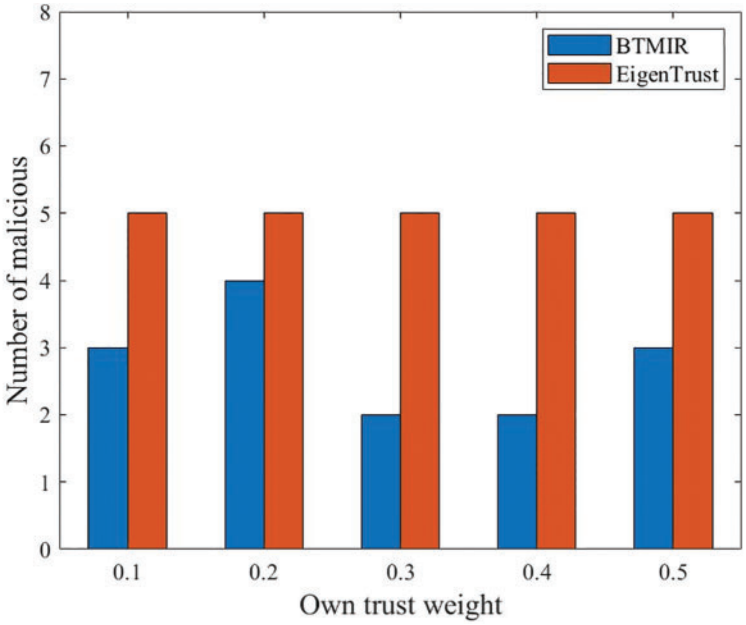


Figure 6: Number of malicious nodes selected as neighbor nodes under changing their trust weights

Table 7: Storage and computation overhead for different network scales

Number of	Storage overhead		Computational overhead	
network nodes	EigenTrust	BTMIR	EigenTrust	BTMIR
10	222B	98B	0.495s	0.792s
20	479B	180B	0.813s	1.285s

(Continued)

Table 7 (continued)

Number of	Storage overhead		Computational overhead	
network nodes	EigenTrust	BTMIR	EigenTrust	BTMIR
50	1.2KB	213B	1.808s	3.031s
100	2.4 KB	430B	3.739s	5.769s

5.5 Robustness of the Trust Model under Different Attack Intensities

We conduct a comparison experiment with 100 nodes, varying the attack intensity and setting the percentage of malicious nodes to 15%, 20%, and 25%, respectively. Fig. 7 shows the robustness of the trust model under different attack intensities. We observe that even by increasing the attack intensity, the number of malicious nodes selected as neighbor nodes in our trust model BTMIR is lower than EigenTrust, reflecting the excellent robustness of our trust model.

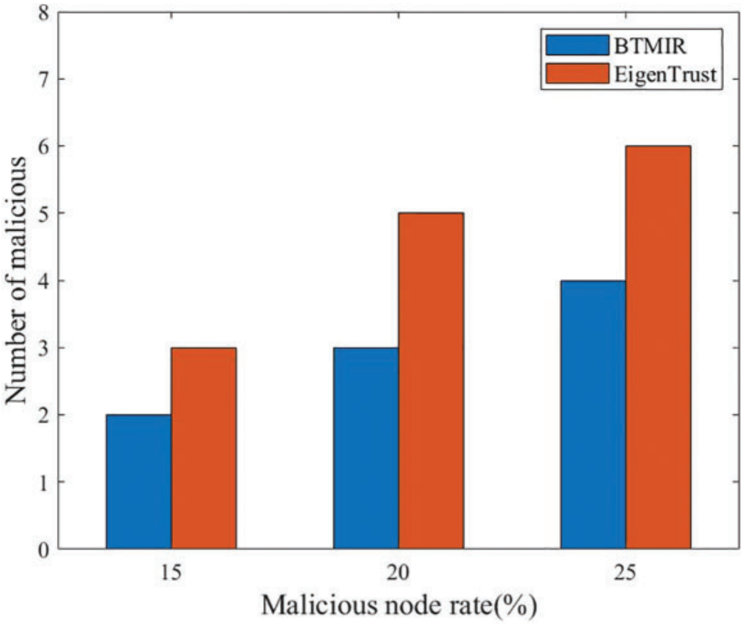


Figure 7: Robustness of the trust model under different attack intensities

6 Security Analysis

In this section, we evaluate the security of our proposed BTMIR trust model, which contains the following main aspects:

1. Credibility: our trust model is combined with blockchain. Blockchain can guarantee trustworthy transactions in a P2P network environment by transmitting information peer-to-peer and utilizing cryptography technology and consensus mechanisms [26]. We conduct trust evaluations with the help of blockchain, which can guarantee the credibility of the evaluation.
2. Integrity: our trust model consists of a master node uploading the trust values of the BGP nodes within the AS alliance to the blockchain. The data in the blockchain is stored in blocks, and each block contains

the previous block's hash value except the genesis block, which realizes the non-tampering of the data. Storing the trust values of BGP nodes by the blockchain ensures the integrity of the trust evaluation.

3. Traceability: as the AS alliance master node that maintains the blockchain, it can view the trust values of other BGP nodes. The blockchain uses an unforgeable chain structure to store data, forming chains in chronological order to ensure data traceability. Our trust model utilizes blockchain technology to enable traceability of trust evaluation.
4. Availability: we adopt the AS alliance structure, where a master node calculates and stores the trust values of the members in the alliance and communicates with other alliances, which can reduce the storage and communication overhead. The master node acts as the full node of the blockchain and does not increase the computational overhead of the blockchain. Therefore, our solution is usable.
5. Defend against DDoS attacks: blockchain does not have a central control node, each node is of equal status, and the topology of nodes is flat, which gives rise to the decentralized nature of blockchain. Each node stores a copy of the data locally, and the nodes back up each other so that it does not affect the global picture even if a node's data is destroyed. We utilize this distributed architecture of blockchain to defend against DDoS attacks.
6. Defend against replay attacks: A replay attack refers to an attacker intercepting legitimate data and then resending the data to the receiver in its original form, deceiving the receiver into thinking it is legitimate data. We utilize timestamps in the blockchain in the trust evaluation process to record the data's generation time to ensure the data's freshness and defend against replay attacks.
7. Defend against bad-mouthing and ballot-stuffing attacks: Malicious nodes may attack other BGP nodes to reduce their trust by executing bad-mouthing attacks on honest nodes and boost their trust by executing ballot-stuffing attacks on their malicious accomplices. We introduce a random function to randomly select BGP nodes using trust value weights as indexes, which can defend against bad-mouthing and ballot-stuffing attacks.
8. Defend against collusive attacks: malicious nodes may collude to destroy the trust level of honest nodes and enhance the trust level of malicious nodes to execute collusion attacks. We reduce the impact generated by collusion attacks by increasing the number of hardened nodes and introducing a randomization function even if the malicious nodes perform collusion attacks.

7 Conclusion

This paper proposes a new inter-domain routing trust evaluation solution called the BTMIR trust model. Unlike existing studies, we move trust evaluation forward, and our trust model is used to discover neighbors and prioritize nodes with high trust as neighbor nodes before requesting their services. We constructed a decentralized trust evaluation architecture where each BGP node in the architecture can evaluate the trust of other BGP nodes. Based on blockchain, our trust model provides a global view of BGP nodes and can speed up the trust evaluation process. We comprehensively evaluated the trust levels of BGP nodes based on our security service evaluation, direct evaluation, and indirect recommendation. We updated these evaluations in real time as conditions changed. Experimental results show that our solution can reduce the impact of collusive attacks, especially collusive bad-mouthing attacks and ballot-stuffing attacks. In the future, we plan to further investigate the proposed trust models to select secure routes.

Acknowledgment: The authors thank the anonymous reviewers and the editorial team for their valuable feedback and suggestions.

Funding Statement: This research was funded by the National Natural Science Foundation of China, grant numbers (62272007, 62001007) and the Natural Science Foundation of Beijing, grant numbers (4234083, 4212018). The authors also extend their appreciation to King Khalid University for funding this work through the Large Group Project under grant number RGP.2/373/45.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Qiong Yang, Li Ma; data collection: Qiong Yang; analysis and interpretation of results: Shanshan Tu, Sami Ullah, Muhammad Waqas; draft manuscript preparation: Qiong Yang; funding, review and editing, Hisham Alasmary. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Not applicable.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Mastilak L, Helebrandt P, Galinski M, Kotuliak I. Secure inter-domain routing based on blockchain: a comprehensive survey. Sensors. 2022;22(4):1–26.
2. Chen D, Qiu H, Zhu JH, Wang QX, Fan SW. Blockchain-based validation method for inter-domain routing policy compliance. J Softw. 2023;34(9):4336–50.
3. Yang Q, Ma L, Tu S, Ullah S, Waqas M, Alasmary H. Towards blockchain-based secure BGP routing, challenges and future research directions. Comput Mater Contin. 2024;79(2):2035–62. doi:10.32604/cmc.2024.049970.
4. Li J, Xu M, Cao J, Meng Z, Zhang G. Decentralized Internet number resource management system based on blockchain technology. J Tsinghua Univ (Sci & Technol). 2023;63(9):1366–79.
5. Xu K, Fu ST, Li Q, Liu BY, Jiang WY, Wu B, et al. The research progress on intrinsic internet security architecture. Chin J Comput. 2021;44(11):2149–72 (In Chinese).
6. Xu K, Ling ST, Li Q, Wu B, Shen M, Zhang ZC, et al. Research progress of network security architecture and key technologies based on blockchain. Chin J Comput. 2021;44(1):55–83 (In Chinese).
7. Putra GD, Dedeoglu V, Kanhere SS, Jurdak R, Ignjatovic A. Trust-based blockchain authorization for IoT. IEEE Trans Netw Serv Manag. 2021;18(2):1646–58. doi: 10.1109/TNSM.2021.3077276.
8. Zhao J, Huang F, Liao L, Zhang Q. Blockchain-based trust management model for vehicular ad hoc networks. IEEE Internet Things J. 2024;11(5):8118–32. doi: 10.1109/JIOT.2023.3318597.
9. Fortino G, Messina F, Rosaci D, Sarnè GML. Using trust measures to optimize neighbor selection for smart blockchain networks in IoT. IEEE Internet Things J. 2023;10(24):21168–75. doi: 10.1109/JIOT.2023.3263582.
10. Tong F, Chen X, Huang C, Zhang Y, Shen X. Blockchain-assisted secure intra/inter-domain authorization and authentication for internet of things. IEEE Internet Things J. 2023;10(9):7761–73. doi: 10.1109/JIOT.2022.3229676.
11. Kouicem DE, Imine Y, Bouabdallah A, Lakhlef H. Decentralized blockchain-based trust management protocol for the internet of things. IEEE Trans Dependable Secur Comput. 2022;19(2):1292–306.
12. Fan X, Liu L, Zhang R, Jing Q, Bi J. Decentralized trust management: risk analysis and trust aggregation. ACM Comput Surv. 2020;53(1):1–33.
13. Ren Y, Li X, Liu H, Cheng Q, Ma J. Blockchain-based trust management framework for distributed internet of things. J Comput Res Dev. 2018;55(7):1462–78.
14. Hu N, Zou P, Zhu PD. Reputation-based collaborative management method for inter-domain routing security. J Softw. 2010;21(3):505–15. doi: 10.3724/SP.J.1001.2010.03479.
15. Wang N, Wang BQ. A reputation-based method to secure inter-domain routing. In: 2013 IEEE 10th International Conference on High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing; 2013; Zhangjiajie, China: IEEE. p. 1424–9.
16. Xia N, Li W, Lu Y, Jiang J, Shan F, Luo J. A trust model for the inter-domain routing system. J Comput Res Dev. 2016;53(4):845–60.
17. Chen D, Qiu H, Zhu K, Wang Q, Zhu J. An inter-domain routing reputation model based on autonomous domain collaboration. Sci Sin Inform. 2021;51(9):1540–88. doi: 10.1360/SSI-2020-0215.

18. Zhao S, Huang X, Zhong Z. Research and implementation of reputation-based inter-domain routing selection mechanism. J Commun. 2023;44(6):47–56.
19. Garcia-Martinez A, Angieri S, Liu BY, Yang F, Bagnulo M. Design and implementation of InBlock—A distributed IP address registration system. IEEE Syst J. 2021;15(3):3528–39. doi: 10.1109/JSYST.2020.3003526.
20. Saad M, Anwar A, Ahmad A, Alasmary H, Yuksel M, Mohaisen D. RouteChain: towards Blockchain-based secure and efficient BGP Routing. Comput Netw. 2022;217(4):1–10.
21. Li J, Cao J, Yang Y, Liu Z, Li Q, Wang Y, et al. DeBGP: decentralized and efficient BGP hijacking prevention system. In: Proceedings of International Conference on Computer Communications and Networks (ICCCN); 2023 Jul 24–26; Waikiki Beach, Honolulu, HI, USA. p. 1–10.
22. Chen D, Qiu H, Zhu JH, Wang QX. Research on blockchain-based interdomain security solutions. J Softw. 2020;31(1):208–27.
23. Fan X, Liu L, Li M, Su Z. GroupTrust: dependable trust management. IEEE Trans Parallel Distrib Syst. 2017;28(4):1076–90. doi: 10.1109/TPDS.2016.2611660.
24. Chen IR, Bao F, Guo J. Trust-based service management for social internet of things systems. IEEE Trans Dependable Secur Comput. 2016;13(6):684–96. doi: 10.1109/TDSC.2015.2420552.
25. Kamvar SD, Schlosser MT, Garcia-Molina H. The EigenTrust algorithm for reputation management in P2P networks. In: Proceedings of the 12th International Conference on World Wide Web; 2003 May 20–24; Budapest, Hungary. p. 640–51.
26. Chen P, Bai F, Shen T, Gong B, Zhang L, Huang L, et al. SCCA: a slicing-and coding-based consensus algorithm for optimizing storage in blockchain-based IoT data sharing. Peer-to-Peer Netw Appl. 2022;15:1964–78. doi: 10.1007/s12083-022-01335-2.