Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.040776
Article

Cross-Domain Data Traceability Mechanism Based on Blockchain

Shoucai Zhao, Lifeng Cao*, Jinhui Li, Jiling Wan and Jinlong Bai

He'nan Province Key Laboratory of Information Security, Zhengzhou, 450000, China
*Corresponding Author: Lifeng Cao. Email: caolf302@sina.com
Received: 30 March 2023; Accepted: 09 June 2023; Published: 30 August 2023

Abstract: With the application and development of blockchain technology, many problems faced by blockchain traceability are gradually exposed. Such as cross-chain information collaboration, data separation and storage, multi-system, multi-security domain collaboration, etc. To solve these problems, it is proposed to construct trust domains based on federated chains. The public chain is used as the authorization chain to build a cross-domain data traceability mechanism applicable to multi-domain collaboration. First, the architecture of the blockchain cross-domain model is designed. Combined with the data access strategy and the decision mechanism, the open and transparent judgment of cross-domain permission and cross-domain identity authentication is realized. And the public chain consensus node election mechanism is realized based on PageRank. Then, according to the characteristics of a nonsingle chain structure in the process of data flow, a data retrieval mechanism based on a Bloom filter is designed, and the cross-domain traceability algorithm is given. Finally, the safety and effectiveness of the traceability mechanism are verified by security evaluation and performance analysis.

Keywords: Cross-domain; data traceability; blockchain; bloom filter

1 Introduction

With the advent of the era of big data, the speed of data generation and flow has achieved unprecedented growth. Interconnection and data sharing between different companies and departments have become a trend. However, data flow between different trust domains causes security problems such as easy data transmission leakage and difficult detection of illegal user access. The privacy and security of data have become the new development bottleneck in the process of data cross-domain sharing. In the process of cross-domain data flow, the traceability of data is the key to ensuring the security and credibility of data, and also the basis of cross-domain data availability. Therefore, it is of great significance to ensure the traceability of the cross-domain data flow. However, there are still some problems in the current data traceability, such as data tampering is not easy to find, and the centralized traceability mechanism is not reliable.

To realize the credibility of flow data, the protection of data must be realized from two aspects. First, the correctness of flow data can be effectively verified, and illegal data. Such as data tampering can be found in time. Second, it can trace the source of the data transfer process and quickly discover unauthorized access from illegal users, which is easy to cause the risk of data leakage [1]. The emergence of blockchain technology provides a new solution to protect data security. Blockchain ensures that blockchain data cannot be tampered with by creating a decentralized distributed chain ledger in an environment of mistrust.

Satoshi Nakamoto first proposes the concept of Bitcoin [2], marking the emergence of blockchain technology. Due to its unique characteristics of decentralization and tamperproof, blockchain technology is widely used in the Internet of Things, cloud computing, and big data scenarios [3–6]. However, in the current application of blockchain-based data traceability, most data is only stored on a single chain. Due to the inherent property of blockchain, it is bound to cause blockchain data inflation [7], affecting the efficiency of data traceability query. In addition, at present, most blockchain traceability applications are limited to a single blockchain or a single trust domain, and such traceability applications are difficult to solve the problems of multisystem and multi-security domain collaboration such as cross-chain information collaboration and data separation storage. Due to the lack of integrity, the credibility of traceability data in a single-link single-domain environment is greatly compromised.

Given the above problems, this paper proposes building a trusted domain with the alliance chain and an authorization chain with the public chain to build a cross-domain cross-chain data traceability mechanism. Details are as follows:

(1) According to the requirements of the cross-domain traceability scenario, the cross-domain model architecture of blockchain is designed. The security trust domain is constructed with the alliance chain, and the public chain is used as the cross-domain authorization chain. The data access strategy and decision-making mechanism are combined to realize the open and transparent judgment of cross-domain authority. And the public chain consensus node election mechanism is realized based on PageRank. At the same time, based on the Public Key Infrastructure (PKI) authentication technology, the cross-domain authentication architecture and process are realized, and the identity consistency of users in different trust domains is realized.

(2) According to the cross-domain model, a detailed cross-domain access process is designed. Considering that in the process of data flow and evolution, multiple different data can be derived from the same data at the same time, the traceable data are no longer a single-chain structure. Because of such data characteristics, a data retrieval mechanism based on the Bloom filter is designed and the cross-domain traceable algorithm is given.

2 Related Research

With the application of blockchain technology, many scholars have studied the application of blockchain technology in the field of data traceability.

To solve the problem of secure storage and sharing of dynamic data generated by a large number of IOT devices, Qiao et al. [1] designs an optimization scheme of the dynamic data traceability mechanism based on the alliance chain. And establishes a secure storage mechanism of dynamic Internet of Things (IoT) data through the dual alliance chain structure of multidimensional authorization of entities and dynamic data storage. ENRICO [8] elaborates on the shortcomings of the blockchain storage mechanism and analyzes the problems of simple query function and low query performance faced by

the existing storage mechanism. Li et al. [9] designs a compound chain structure by combining a private chain and an alliance chain and proposes a compound blockchain-associated event traceability method for financial activities based on the Apriori algorithm. Zhang et al. [10] realizes product traceability in the supply chain through smart contracts and develops a side-chain-based supply chain traceability system, which realizes goods management and information sharing in the commodity supply chain and improves the credibility of the supply chain commodities. Tian [11] introduces the concept of BigchainDB in the blockchain, aiming at the scalability problem caused by the need to store a large amount of data, and designs a traceability system for agricultural products based on Radio Frequency Identification (RFID) technology and blockchain technology.

Lin et al. [12] develops a food safety traceability system based on the Ethereum public blockchain platform to meet traceability requirements in the field of food safety and realizes the recording, sharing, and tracking of data information in the food supply chain. Li et al. [13] designs a hierarchical graph blockchain pharmaceutical traceability model by introducing graph blockchain into the pharmaceutical field to solve the problems of low throughput and high storage overhead of the traditional single-chain blockchain pharmaceutical traceability system. Ruan et al. [14] stores traceability information during the execution of smart contracts in the Merkel tree using a skip list, designs a fine-grained, safe, and efficient traceability system, and improves the efficiency of the blockchain data traceability query.

Through the analysis of existing studies, it is found that most existing blockchain data traceability mechanisms are only carried out in a single blockchain or a single trust domain. However, with the flow and sharing of data among multiple systems and departments, a single data traceability mechanism can no longer meet the requirements of data integrity and credibility. Therefore, cross-domain cross-chain traceability of data in complex scenarios has become the key to data integrity and credibility.

3 Consensus Mechanism

Given the consistency requirements of blockchain global data information in the cross-domain process, this section designs a consensus mechanism applicable to cross-domain scenarios based on the improved PageRank algorithm.

The PageRank algorithm [15], also known as the Page ranking algorithm, is a link analysis algorithm proposed by Larry Page and Sergey Brin when building the search system prototype. PageRank is based on two assumptions: quantity and quality. The initial value of the importance of a web page is calculated based on the number of incoming links and the quality of the incoming pages. The initial value of credibility is determined by the node's cross-domain transaction participation rate, processing efficiency, and other historical performance. The final stable importance value is calculated recursively based on the initial value of the importance of the web page. The recursive calculation process is shown in formula (1).

$$PR(j) = 1 - d + d \times \sum_{i \in P_j} \frac{PR(i)}{L(i)} \tag{1}$$

where, d represents the damping coefficient, which is usually 0.85; P_j represents a collection of pages linked to pages; $PR(i)$ represents PR values representing the importance of the page; $L(i)$ indicates the number of outgoing pages.

The traditional PageRank algorithm is mainly used to rank the importance of web pages, but there are some problems with applying it directly to the ranking of blockchain nodes. Specific performance

is as follows: (a) The algorithm evenly allocates the PR value of nodes to other nodes, which cannot reflect the quality difference between nodes and affects the quality of node ranking; (b) In the algorithm, the importance of nodes is calculated only through the voting of other nodes, and the initial score of nodes is not fully considered.

Given the problems existing in traditional PageRank, the following improvements are made to fully apply to the cross-domain node sorting of the blockchain: (a) To further distinguish the granularity of the voting intention of nodes, voting weight parameters are set for all nodes, and voting nodes can vote for other nodes freely in proportion; (b) To make full use of the initial node score, the initial node score is added to the iterative calculation formula. Calculating the credit value of the node across the domain after improvement is shown in formula (2).

$$PR(i) = PR_0(i) + d \times \sum_{j \in Q, j \neq i} \left(PR(j) \times w_{j,i} \right) \tag{2}$$

where $PR(j)$ represents the PR values of node j; Q represents the cross-domain consensus node set; $PR_0(i)$ indicates the initial trust value of node i in the domain. $w_{j,i}$ indicates that node j is the vote weight of node i. For honest nodes, the vote weight parameter is positive; for malicious nodes, the vote weight parameter can be negative. The voting weight is shown in formula (3).

$$w = \begin{pmatrix} 0 & w_{1,2} & \cdots & w_{1,n} \\ w_{2,1} & 0 & \cdots & w_{2,n} \\ \cdots & \cdots & 0 & \cdots \\ w_{n,1} & w_{n,2} & \cdots & 0 \end{pmatrix}, \left(\sum_{j=1}^{n} w_{i,j} = 1 \right) \tag{3}$$

At the same time, considering the existence of malicious nodes in the voting process of nodes in the same domain. malicious nodes may vote for a node through collusion attacks, and the malicious node eventually becomes the cross-domain consensus master node with a higher value of PR. By introducing the initial value of node $PR_0(i)$ in the iterative calculation process, the cost of a malicious node collusion attack is increased. At the same time, to further ensure that malicious nodes cannot launch collusion attacks, the voting weight of nodes is restricted here. We stipulate that in the weight value matrix, a single weight value meets $|w_{i,j}| < 0.33$.

$$|PR_{r-1}(i) - PR_r(i)| < \frac{PR_r(i)}{100} \tag{4}$$

As shown in Eq. (4), The iteration ends when the difference between two adjacent iterations of the reputation values of all nodes is less than one percent of their reputation values. Several nodes with the highest ranking of the final credit values are selected to form the notary node. Meanwhile, the node with the highest credit value acts as the main node of the cross-domain transaction consensus and finally completes the cross-domain transaction consensus.

4 Cross-Domain Model

4.1 Cross-Domain Architecture

The architecture of the cross-domain model is shown in Fig. 1, including the trust domain composed of different alliance chains, the decision of the cross-domain cross-chain based on the notary group, the authorization of the cross-domain, and other parts.

Each trust domain consists of an alliance network. Different blockchain nodes in the same trust domain jointly maintain data security in the alliance network. According to the demand for data object sharing between different blockchains in the domain, data object owners can define their access

policies. At the same time, different blockchains within each trust domain can be not only the access chain that initiates cross-domain traceability behavior but also the destination chain of data storage when users trace the source on other blockchains.

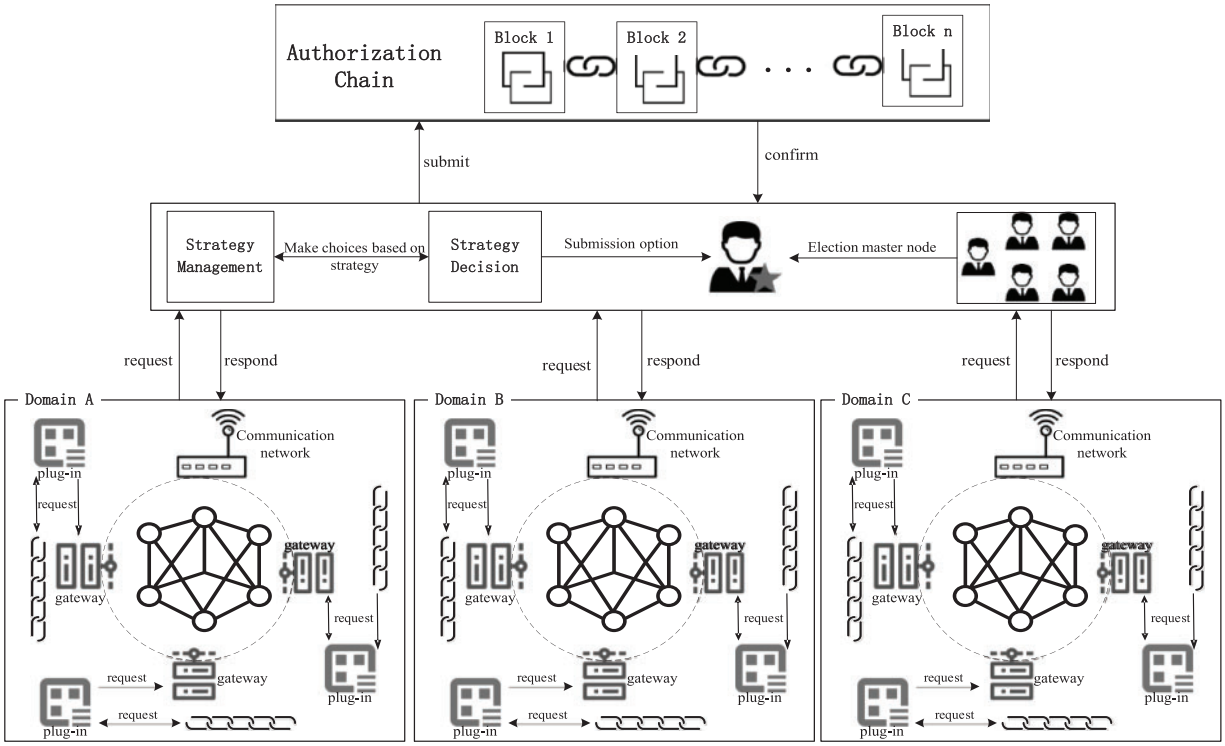

Figure 1: Cross-link data trace architecture for cross-domain access

The cross-domain cross-chain management decision based on the notary group includes data object policy management, policy decision, and notary group. The policy management module is responsible for storing access policies of cross-domain data. and the owner of the data object makes access control rules according to the data nature, privacy requirements, and other factors. The policy decision module decides whether to allow the cross-domain traceability access behavior according to the permission information of the cross-domain request user subject and the access policy of the interviewed data and submits the decision to the master node. The notary group is composed of some trusting nodes in different trust domains. According to the transaction processing efficiency, success rate, and activity degree of entrusting nodes in historical transactions, the credit value of each entrusting node is calculated, and the credit value is used as the voting weight. The improved PageRank algorithm is used to calculate the final credit value of each node. The node with the highest credit value is selected as the main node of this round of cross-domain transactions.

Cross-domain authorization data are recorded by the public chain. After the master node is packaged and verifies the authorization data, it is uploaded to the authorization chain. Data access policies and authorization information are stored on the blockchain in an open and transparent form, completing a decentralized transparent judgment without the need for a trusted third-party endorsement.

4.2 Cross-Domain Authentication

Cross-domain authentication refers to the process of users completing consistent authentication in different security domains [16]. In the cross-domain traceability process of users, reliable identity authentication is the key to ensuring the data security of the traceability system. Only based on user identity authentication can the target data of traceability be transferred safely and effectively between different trust domains. When data flows between multiple different trust domains, data traceability must be expanded to different trust domains. Therefore, for users with cross-domain traceability and cross-domain access requirements, designing a globally identical cross-domain identity authentication mechanism is the key to improving the efficiency of cross-domain access traceability.

According to the different cryptographic systems used in cross-domain authentication, cross-domain authentication can be divided into symmetric key technology authentication architecture and asymmetric key technology authentication architecture. Symmetric key authentication has the advantages of fast running speed and high authentication efficiency. But this authentication mechanism also has the obvious risk of key disclosure and is not suitable for the security requirements of identity authentication in the process of cross-domain traceability. Asymmetric Key authentication architecture can be divided into Public Key Infrastructure (PKI) Based Identity authentication technology and Identity-Based Cryptography (IBC) based identity authentication technology. Among them, the PKI-based identity authentication technology has excellent scalability and flexibility, but inherent computational complexity, overhead redundancy, and other problems reduce its performance in the application process [17]. In IBC-based authentication technology, the public key is bound to the user information, which effectively simplifies the management process of the cross-domain authentication key. However, in IBC-based identity authentication technology, the user's private key is obtained through centralized calculation by the key generation center, which is consistent with the idea of decentralization of the blockchain. It can only be used in the scope of a small trust domain and cannot be well applied to the cross-domain traceability field of blockchain.

In this section, based on existing PKI identity authentication technology and through the public chain consensus mechanism, an identity authentication model suitable for the cross-domain traceability scenario is designed, as shown in Fig. 2.

Cross-domain authentication consists of an intra-domain certificate management layer and an out-of-domain authentication layer. The certificate management layer consists of different trust domains. Each trust and trust include a common user node N from different blockchains in the domain and the certificate management server CA responsible for certificate management in the domain. The authentication layer mainly includes authentication servers AS from different security domains.

The certificate management layer in the domain contains the security trust domain composed of different alliance chains. The domain contains the user nodes on different blockchains and the certificate management server in the domain. Before joining the security trust zone, all nodes in the zone must pass CA for identity verification and intra-zone certificate issuance. Nodes can be divided into N cross-domain nodes and N^* non-cross-domain nodes based on whether the user node has cross-domain access requirements. Non-cross-domain nodes only need to be accessed within the same trust domain. Therefore, the cross-domain authentication server AS is not required for authentication. When joining a cross-domain node, it first applies to CA for joining the node, and then CA submits the identity information of the node to AS to complete the cross-domain node information registration.

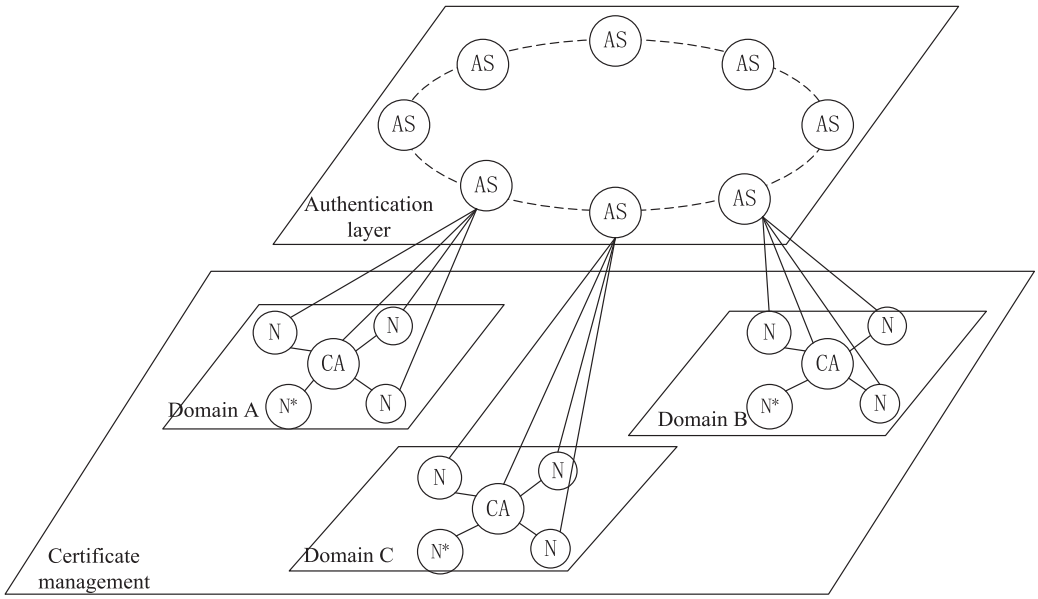

Figure 2: Diagram of the cross-domain authentication model

The authentication server *AS* is used for certificate storage and authentication of cross-domain required nodes in the domain and is responsible for the interaction of identity authentication information between the trust domain and other trust domains. The public chain consensus node determines whether the cross-domain behavior is authorized or not according to the data access policy and the authentication result of cross-domain user identity information, sends the authorized user information to the target domain *AS*, and issues the certificate of the target domain.

In the process of cross-domain traceability, the user *N* first initiates a cross-domain request, and the master node of the public chain sends a random number to the user *N* for identity authentication. Based on the verification results, the public chain master node makes a consensus decision on the access permission of the cross-domain request node. If the request passes, it applies to the cross-domain certificate of the node from the target domain *AS*. At the same time, the target domain *AS* implements two-domain authentication on the user *N* through the domain *AS* and *CA* where the user *N* resides. After authentication, the target domain *AS* applies for the cross-domain access certificate of user *N* from the target domain *CA*. Finally, the cross-domain authentication result is sent to *AS* of the domain where the user *N* resides through the public chain, the authorization certificate is linked to the public chain, and the authentication result is sent to the user *N*. Fig. 3 shows the cross-domain authentication process of nodes.

The cross-domain authentication phase verifies whether the requested user is a legitimate cross-domain user. After authentication, the user obtains the preliminary access qualification of the target domain. When traceability access to specific data in the target domain is required, users still need to apply for access authorization. Consensus nodes on the public chain decide whether to authorize or not by consensus according to data access policies.

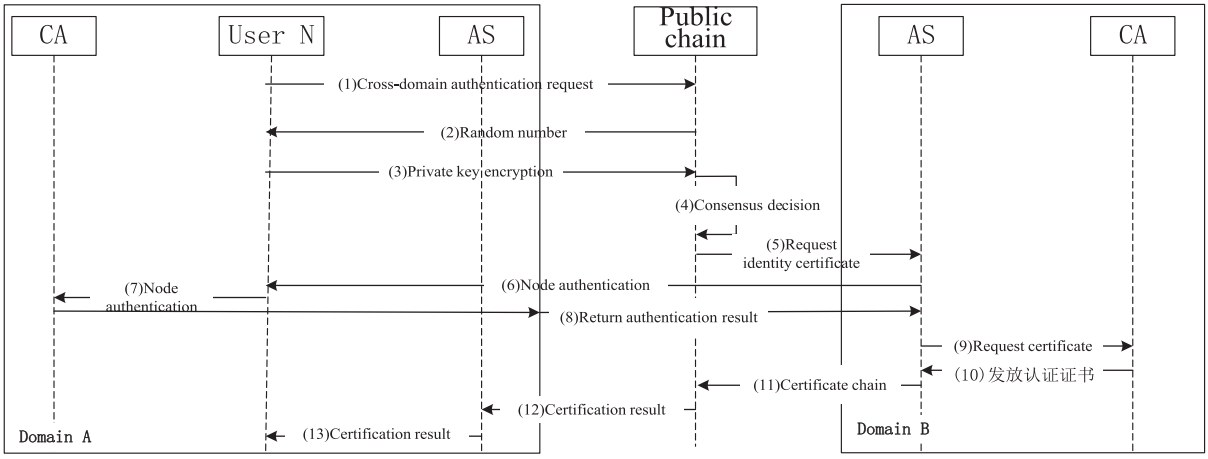

Figure 3: Flow chart of cross-domain identity authentication and certificate issuance

5 Cross-Domain Traceability

5.1 Cross-Domain Access Flow

The data traceability structure across the security trust domains is shown in Fig. 4. According to the logical sequence, it can be divided into three stages: data object upload, cross-domain access authorization, and cross-domain data access. The steps in each stage are as follows:

The first stage is the data object upload stage. When uploading the data object, users can upload the data only to their blockchain. If the user is on several different blockchains at the same time, they can freely choose a certain blockchain to upload the data.

Step 1: When the data owner uploads the data to the blockchain database, the message format is $< upData, DID, BID, data, H(data), NID >, \sigma_{NID} >$, in which *upData* represents the data upload mark, *DID* represents the domain number of the blockchain, *BID* represents the blockchain number, *data* represents the uploaded data object, $H(data)$ represents the summary information of the uploaded data, *NID* represents the identity information of the data owner, and σ_{NID} represents the signature of the data owner. In the process of data upload, the consensus node in the chain records the uploaded data as an initial transaction on the blockchain.

Step 2: After the data object is connected, the data owner will upload the data storage location information index and data access strategy to the public chain. The message format is $< upStrategy, H(data), Strategy, DID, BID, NID >, \sigma_{NID} >$. Where *Strategy* represents the access policy of data $H(data)$. The access policy can specify requirements such as the level of the user and the security domain where the user resides. By recording on the public chain, any user can query the access policy of the data, ensuring the transparency of the decentralized decision.

The second phase is the authorization phase for cross-domain access. In this phase, the user obtains the identity authentication of the destination domain and obtains permission to access the destination data.

Step 3: Users with cross-chain requirements in the domain send cross-chain traceability requests, and the traceability requests message format is $< request, TDID, DID, BID, H(data), NID >, \sigma_{NID} >$. Among them, *request* is the initiating mark of cross-chain traceability request, *TDID* is the security

trust domain number of the target data H (*data*), *DID* is the security trust domain number of the source requesting user, *BID* is the blockchain number of the target data, *NID* is the identity information of the cross-domain access user, and 88 is the signature of the cross-chain traceability requesting user.

Step 4: After monitoring the cross-chain request of users in the domain, the monitoring module analyzes the request message to get the target domain *TDID*. If the target domain number is the same as the local area number, the cross-chain traceability request message will be forwarded to the intradomain traceability query. Otherwise, the traceability request message is forwarded to the consensus node in the domain.

Step 5: The consensus node within the domain determines whether the node has a malicious tendency according to the user's performance in the transaction within the domain, to decide whether to allow cross-domain access. For cross-domain requests of well-behaved nodes, the message is decapsulated by the master node in the domain and sent to the public chain.

Step 6: After receiving the cross-domain access request from the user, the public chain analyzes the request message, obtains the identification information of the requester *NID* and the hash value of the access data information H (*data*), and finds the data access policy uploaded by the data owner according to the hash value of the data to be accessed.

Step 7: The policy decision module makes a transparent judgment on the cross-domain access request according to the identity information and data management strategy of the cross-domain access request user and sends the result to the master node of the public chain.

Step 8: After the master node receives the cross-domain decision, it sends the decision message to the public chain consensus node-set and the consensus node agrees on the decision result. Additionally, cross-domain access authorization is recorded in the form of logs on the public chain to ensure that the authorization result is accessible and verifiable.

Step 9: The user authorized by cross-domain will be notified by the public chain master node to the consensus node of the domain where the cross-domain request user resides. The notification message format is $< result, DID, NID, H(data), Cert_{DID \to TDID} >, \sigma_{MNID} >$, *result* indicates the cross-domain request message return flag, *DID* indicates the security trust domain number of cross-domain request user *NID*, $Cert(NID)_{DID \to TDID}$ indicates the cross-domain authorization certificate of cross-domain request user *NID*, *TDID* indicates the security trust domain number of target data H (*data*), and σ_{MNID} indicates the signature of the master node of the public chain.

Step 10: After obtaining the cross-domain license, the consensus node in the domain issues the cross-domain license certificate to the certificate management server in the domain and informs the cross-domain requesting user. Then cross-domain request users with cross-domain certificate cross-domain traceability access.

The third stage is the data access stage cross-domain. This phase mainly solves the problem of traceability of cross-domain access when users have obtained cross-domain access authorization.

Step 11: The authorized cross-domain access user sends the cross-domain access message to the target domain through the communication network of the domain where the user resides. The message format is $< aRequest, TDID, DID, BID, NID, H(data), Cert(NID)_{DID \to TDID} >, \sigma_{NID} >$. Where, *aRequest* indicates the cross-domain access flag, *TDID* indicates the number of the security trust domain where target data H (*data*) resides, *DID* indicates the number of the domain where the cross-domain requestor resides, *BID* indicates the number of the blockchain where the target data resides, *NID* indicates the number of the cross-domain access user, $Cert(NID)_{DID \to TDID}$ indicates the cross-domain access certificate, and σ_{NID} indicates the signature of the cross-domain access user.

Step 12: After the target domain communication network obtains the cross-domain access message, analyze and obtain the identity information and cross-domain access certificate of the user. The user's identity information is compared with the identity authentication records issued by the target domain *CA*. At the same time, the public chain authorization logs are used to verify the authenticity of the cross-domain access authorization certificate of the user *NID*. User *NID* is allowed to access the target data in the target domain *TDID* only when both the identity authentication and the cross-domain access authorization certificate are authenticated.

Step 13: Distribute the authenticated user access request to the target blockchain through the network for data access query. Meanwhile, record the cross-domain access behavior of the user through the access log, to conduct subsequent joint analysis according to the user's access behavior and authorized behavior, and timely discover the risk of data leakage.

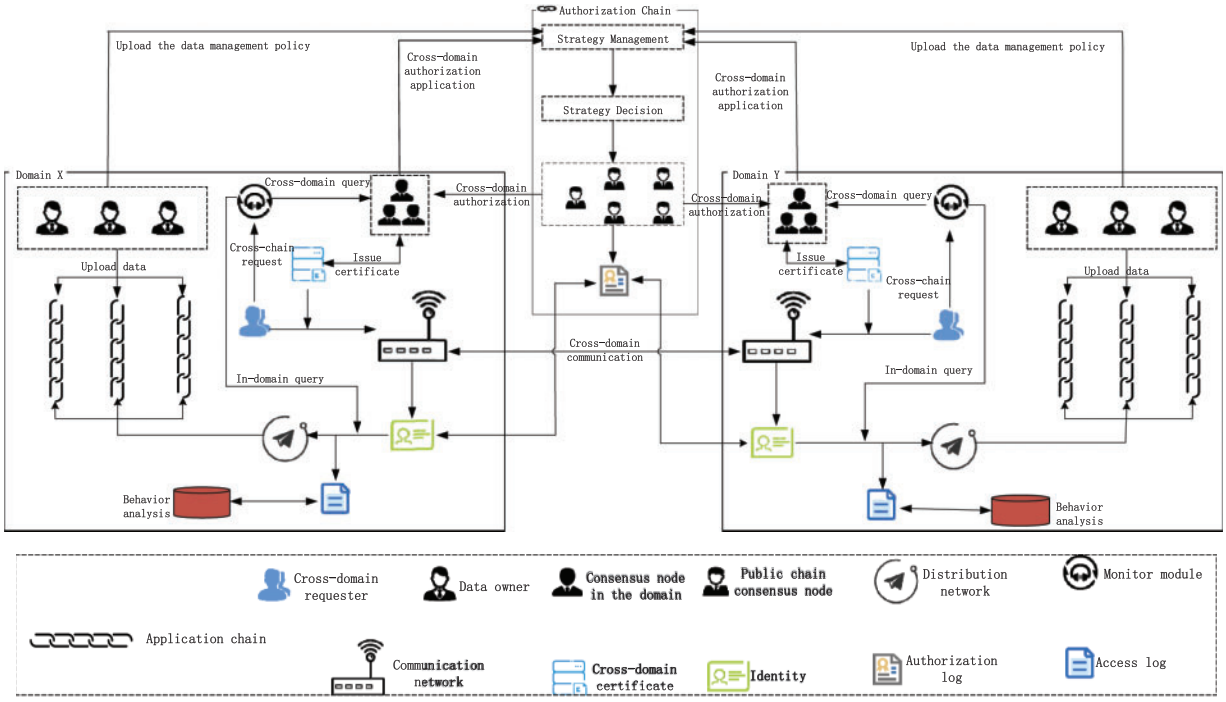

Figure 4: Cross-domain traceability structure diagram

5.2 Data Retrieval Mechanism Based on Bloom Filter

In the traditional blockchain data query, each piece of data points to the previous piece of data, and after the output of the previous piece of data transaction, it cannot be output again; otherwise, it will cause a double flower attack. However, in the process of data traceability, the same data may undergo different types of change by multiple users and eventually evolve into multiple derived data at the same time. This form of data evolution is no longer a single-chain structure. In this case, if the existing blockchain traceability query is still adopted, only a single chain of data evolution can be obtained, rather than complete data traceability information, and it is not easy to find the risk of data loss or leakage. To solve this problem, this section proposes a fast-tracing mechanism based on the Bloom Filter.

Bloom proposed Bloom Filter for the first time in 1970 [18]. Bloom Filter is a probabilistic data structure composed of a certain length binary array and a group of random and mutually independent hash functions, which have the advantages of less space and high query efficiency [19].

By sacrificing query accuracy to query time efficiency, the Bloom filter has the characteristics of one-way misjudgment. For a large data set $D = \{d_1, d_2, \ldots, d_n\}$, whether an element d belongs to set D ($d \in D$?) or not can be judged in a very short time by the Bloom filter. If element d is in set D, it must return the correct result, which is $d \in D$. When the query result returns an element that is not in the set, the element is not in the set. However, when $d \notin D$, the query result may return 88. This one-way misjudgment is also called false positives. Table 1 shows the actual situation and possible query results.

Table 1: The query result corresponds to the actual situation

Actual situation	Query information
$d \in D$	$d \in D$
$d \notin D$	$d \notin D$ or $d \in D$

According to its working process, Bloom Filter can be divided into two stages: element insertion and element search. Initially, all m bits of the Bloom filter binary vector is set to 0, each element in the set to be inserted is indexed by k hash functions, and the corresponding k bits in the binary vector are set to 1. In this section, the hash function *MurmurHash*3, which is fast to compute, is used as the seed function, and k hash functions are derived from $H_k = MurmurHash3\,(d + k)$. When the length of the binary vector is 16 and the number of hash functions is 4, Bloom Filter is constructed during the process of adding elements to the set, as shown in Fig. 5. Fig. 5a shows the process of initializing the Bloom filter. In Fig. 5b, the index value $H_i\,(d)$ of element d to be added is calculated according to the hash function, where $i \in [1, k]$ is included. For example, $H_1\,(d) = 5$ sets the fifth bit of the binary vector to 1.

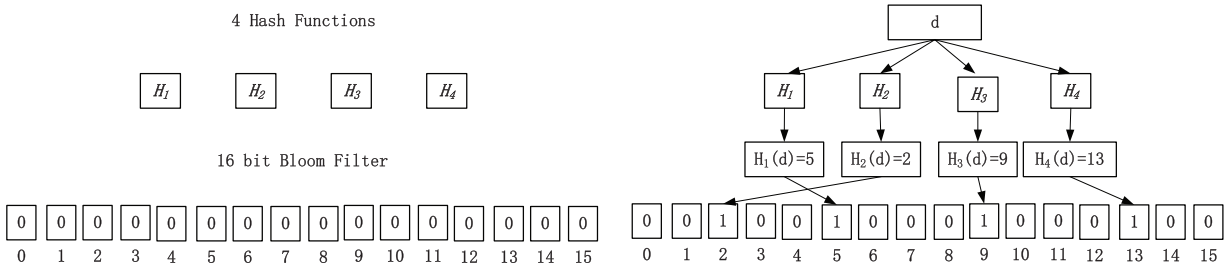

Figure 5: Construction of the Bloom Filter

In the element search stage, for a given Blum filter binary vector table, the element search is carried out by first calculating k index values $H_i\,(d)$ of the element d to be searched, and then comparing the bits on the corresponding binary bits one by one according to the index values. As long as one bit of the binary vector bit corresponding to the index value is 0, it means that the element must not be in the set. If all are 1, the element is probably in the set. The element search process is shown in Fig. 6. In Fig. 6a, element d_i may exist in the set because all corresponding positions of hash indexes are equal to 1; in Fig. 6b, element d_j must not exist in the query set because corresponding positions of hash indexes are 0.

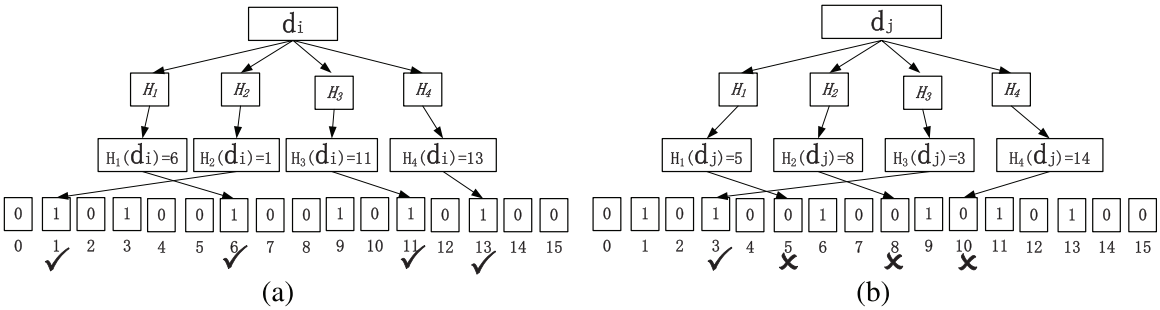

Figure 6: Bloom filter query

According to the above quick query principle, the owner of the original data will upload its hash value index $H\ (data_0)$ when uploading the original data $data_0$, and take it as the index value of the subsequent evolution of data $data_0$. In other words, the index value of data $data_0$ will not change in the process of being modified and transferred by different users on the blockchain. In the process of cross-domain traceability, the user can conduct global traceability queries, a global traceability query on the evolution data of $data_0$ flow between different domains according to the index value, to realize the rapid, accurate, and complete construction of the data traceability graph.

5.3 Cross-Domain Traceability Algorithm

According to the cross-domain access architecture and access process proposed in this paper, users' cross-domain and cross-chain data traceability requirements are realized, and the data traceability results are visualized in the form of directed acyclic graph $G = (V, E)$, where V represents the data state and E represents the data flow direction.

The display form of traceability results is shown in Fig. 7. Each traceable data contains data information and attribute information. The data information contains the data key index $key = H\ (data_0)$, the hash value $H\ (data_i)$ of the current data $data_i$, and the parent hash $PreH$ of the current data. The Data attribute information contains the following: Attribution of authority (AOA), Grant authority (GA), Change of authority (COA), Access strategy (AS), and Access record (AR).

In attribute AOA, it contains data owner information, data domain, and blockchain number information; GA Records the existing authorization of the current data owner for this data, including adding, deleting, modifying, checking, etc. COA Indicates the operation to change the permission of the authorized user on the data. AS refers to the data access policy formulated by the data owner when the data is uploaded; AR Records the user information that accesses the current data object.

To further clarify the flow status, the following provisions are made:

Rule 1: For the original data, both user A and user B have permission to modify them. After user A modifies the data, the data change form is $data \xrightarrow{A} data'$, while after user B modifies the data, the data change form is $data \xrightarrow{B} data'$. After modification by Party A and Party B, although the data change result is the same, the subject of the data changes after modification by different subjects. The ownership of the former data belongs to Party A, while the ownership of the latter data belongs to Party B. Therefore, we believe that the data generated after modification by Party A and Party B are in different states.

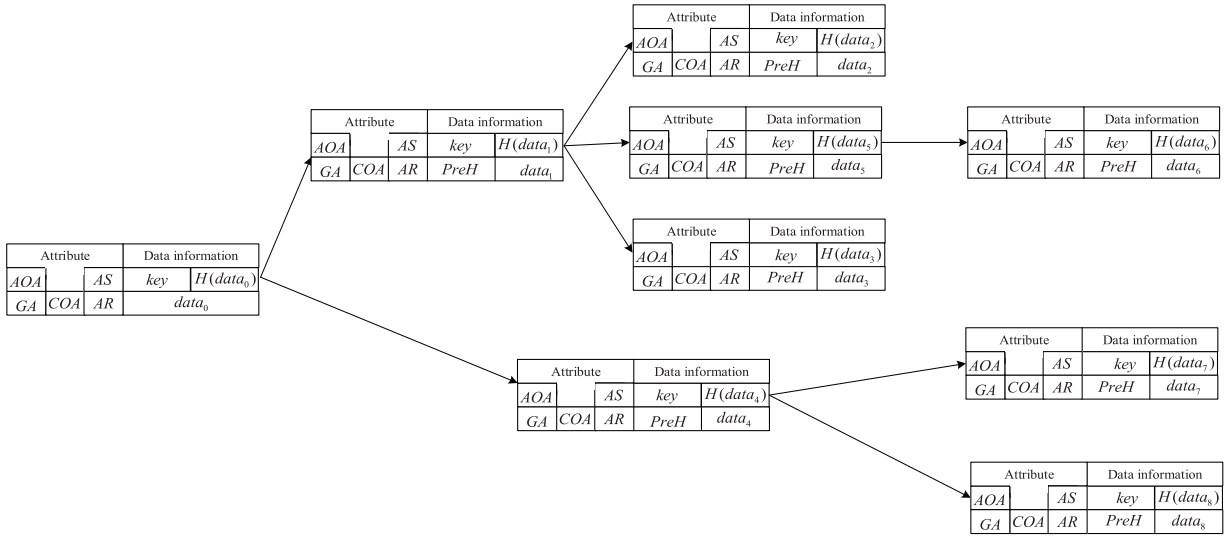

Figure 7: Tracing results show the structure diagram

Rule 2: If the owner of the data object *data* is judged as a malicious user by the system, all downstream data evolved through *data* will be judged as illegal data, and authorization operations of all downstream data related to *data* will be set as invalid authorization.

According to existing rules and traceability data display structure, the cross-domain data traceability algorithm is designed as follows:

Algorithm 1: Cross-domain traceability algorithm

Input: Traceable data index $key = H(data_0)$
Output: Data tracing result graph $G = (V, E)$
Phase 1 // Look for permission information on the public chain
if (Query (*key*) $= \varnothing$) // Query *key* on the public chain
  return Failure
else
  $H \leftarrow Query(key)$ // The authorization information related to the *key* is placed in the set H
  Foreach $H(data_i)$ *in* H // Iterate through each authorization data in the collection
    if $(H(data_i) \notin V)$
      Queue.enqueue $(H(data_i))$ // Add $H(data_i)$ to the queue
      add $H(data_i)$ *as* V_i *into* V // Add $H(data_i)$ to the traceability graph vertex
      if $(V_i.PreH \neq \varnothing)$
        add $(V_i.PreH, V_i)$ *into* E // Pointing data to the previous state
      add DID & NID & BID into $V_i.AOA$ // Add data attribute information
      add strategy into $V_i.AS$ // Adding data access policies
      if operation = authorization // Add authorization action to the property
        add authorization into $V_i.GA$
      if operation = changeAuthority // Adding permission change operations to properties

(Continued)

Algorithm 1 (continued)

add changeAuthority into $V_i.COA$
else // The data has been added to a vertex and there are multiple authorizations or permission changes
if operation = authorization // Add authorization action to the property
add authorization into $V_i.GA$
if operation = changeAuthority // Adding permission change operations to properties
add changeAuthority into $V_i.COA$

Phase 2 // Find data information in different domains

while (*Queue* $\neq \varnothing$)
$V_i \leftarrow$ *Queue.dequeue* () // Get the data already in the authorization chain from the queue
Alog $\leftarrow$ *from DID & BID Access log about* V_i // Get the access record of V_i
add Alog into $V_i.AR$ // Adding data access records to the access log
foreach Au in V_i*.authorization*
if ((*Query*(*key*) *from Au.TDID & BID*) $\neq \varnothing$) // The presence of post-authorization modified data on the authorization target chain
$V_k \leftarrow$ *Query* (*key*) *from TDID & BID*
add V_k *into V*
add DID & NID & BID into $V_i.AOA$ // Add data attribute information
add(V_i, V_k) *into E* // Pointing data to the previous state
Queue.enqueue (V_k)
end

According to the results of the traceability query, in data security analysis, according to the authorization record and the data interview record of the user subject under a certain data state, the user behavior of unauthorized access can be obtained through joint analysis.

6 Experimental Analysis

6.1 Theoretical Analysis

6.1.1 Credibility

In the process of access to cross-domain traceability of users, the security trust domain is constructed using the alliance chain. Different blockchains within each alliance chain serve as the access chain, and the cross-chain authorization chain is constructed by the public chain to realize open and searchable cross-domain authorization operations. Meanwhile, in the public chain with the highest degree of decentralization, it is more difficult for malicious nodes to launch attacks to forge cross-domain authorization logs, so that the cross-domain authorization operation is reliable and searchable. After data object owners upload data access policies to the public chain, consensus nodes on the public chain can make decisions on granting permissions to users requesting cross-domain access according to the access policies of the data object, realizing open and transparent judgment of cross-domain authorization and further improving the credibility of cross-domain authorization.

In the process of data flow and evolution, multiple users may have permission to modify or access certain data at the same time. When multiple users the permission to modify the same data at the same time, multiple sub-data will be generated, that is, one data state can evolve into multiple data states at the same time. This form of data evolution is no longer a single chain structure and is no longer suitable for point-by-point traceability queries in a blockchain chain structure. With the introduction

of the Bloom filter, complete traceability of data in the form of undirected graph flow is realized, ensuring the integrity of blockchain data flow traceability. The Blum filter has false positives with unidirectional misjudgment. For m vector bits, n elements are inserted and k function mappings are performed, respectively. The false positive misjudgment rate is

$$\left(1-\left(1-\frac{1}{m}\right)^{kn}\right)^{k} \approx \left(1-e^{-\frac{kn}{m}}\right)^{k} \tag{5}$$

When the relation between variables meets $k = (m/n)\,\mathrm{Ln}2$, the false positive probability reaches the minimum of $(0.5)^k$. Although the Bloom filter has false positive misjudgment, the hash value $key = H\,(data_0)$ of the original data is used as the index in the data. Through the query of the key value of the data in the false positive position, the false positive data can be easily judged, avoiding the false positive misjudgment of tracing results caused by the Bloom filter, and making full use of the advantage of the extremely short constant query time of the Bloom filter.

6.1.2 Security

In the process of cross-domain access tracing, this paper adopts a dual verification mechanism of cross-domain authorization and identity authentication to separate cross-domain authorization and identity authentication and further improve the security of cross-domain access tracing. The public chain consensus node grants permission to the user to request cross-domain access according to the access policy of the data object owner and the user identity level. Only authorized users can perform cross-domain access request operations. When the cross-domain access request is made, the public chain and the target domain verify the cross-domain authorization certificate and the user identity again. After the cross-domain authorization certificate and identity authentication are successful, the target domain certificate management server issues certificates of the corresponding level in the target domain to the cross-domain user for access to the cross-domain user. Permission authentication and identity authentication ensure that the permission and identity of the cross-domain user correspond, preventing fraudulent attacks by malicious users.

According to Rule 2, in the data flow authorization access operation, once malicious users are found, their data will be set as illegal data, and downstream data evolved from the data flow will also be set as invalid. To further restrict the flow of invalid data, authorized operations related to invalid data and its downstream data are set as invalid, to prevent its circulation in the traceability system and ensure the security of legitimate users and data flow.

6.2 Performance Evaluation

In this paper, we use the trunk chain to build the public authorization chain. Nodes in each security trust domain join the public chain to form the public chain notary node, and the notary node makes a consensus on cross-domain authorization, access, and other affairs. An Ethereum test chain is built to test the rationality of notary node selection. The parameter settings of each node are shown in Table 2.

To visualize the distribution of reputation values of different nodes, we set the initial value parameters for different nodes. According to the PageRank algorithm proposed in this paper, the distribution of different node reputation values is obtained as shown in Fig. 8.

The results show a clear distribution range of reputation values for different types of nodes. The credit values of malicious nodes are significantly lower than those of ordinary and trusted nodes, indicating that the improved PageRank algorithm is effectively used for the identification of malicious nodes.

Table 2: Node configuration parameters

Node type	Number	Average processing time	Transaction success rate	Service participation rate
Trusted nodes	5	[10 ms, 20 ms]	95%	95%
General node	15	(20 ms, 40 ms]	80%	80%
Malicious nodes	5	(40 ms, -)	[0%–60%]	[0%–60%]

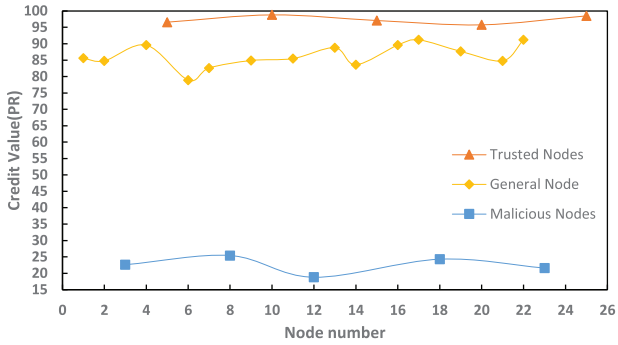

Figure 8: Node credit value distribution

The main difference between trusted nodes and ordinary nodes is that the average transaction processing time and communication delay between nodes are different, while malicious nodes only vote for themselves or try to collude attacks to vote for a malicious node in the voting process. In the initial stage, since the quality of each node is unknown, we set the initial credit value of all nodes as 50. The initial reputation value of a non-newly joined node is determined by the node's average online time, the participation rate in historical transactions, and cross-domain transaction processing efficiency. For the convenience of statistics, the node credit value is assigned to the percentage system by formula (6).

$$PR(i) = \frac{PR(i) - MinPR}{MaxPR - MinPR} \times 100 \tag{6}$$

According to the configuration of existing parameters, after testing different numbers of consensus transactions, the average credit value distribution of the nodes is shown in Fig. 9.

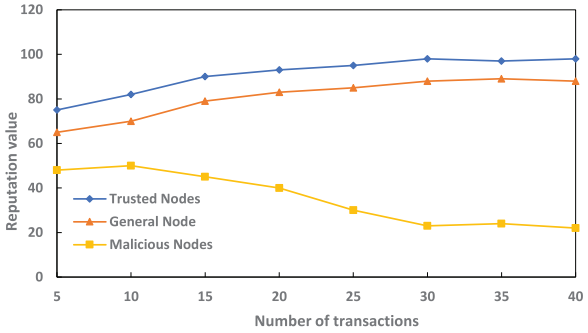

Figure 9: Distribution of the average credit values of different nodes

The results show that for different types of nodes, there is a significant difference in the trend of credit value change as the number of participating in cross-domain consensus transactions increases. When the nodes participate in a certain number of cross-domain transactions, other nodes can easily judge the goodness of the nodes. Thus, the risk of malicious nodes attacking during cross-domain authorization access is reduced.

To test cross-domain transaction processing throughput on the public chain, we assume that the consensus node can only respond to a single cross-domain transaction each time and inject multiple cross-domain transaction requests at the same time. Under different requests, the average time delay of each round of request processing is shown in Fig. 10.

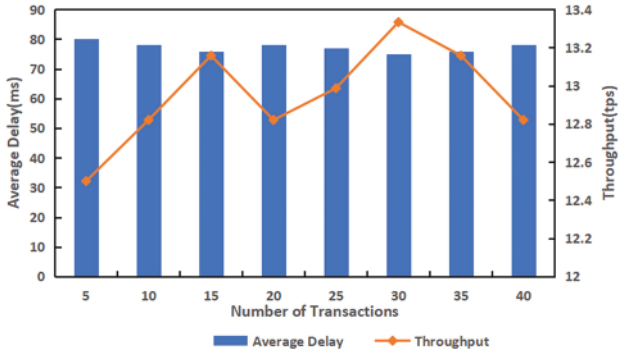

Figure 10: Average duration and throughput of the nodes

As can be seen from the figure, with the increase in the number of cross-domain request transactions, the average round of cross-domain transaction processing delay tends to be stable, about 77 ms. According to the calculation method of the throughput rate $v = 1/t$, the throughput rate is about 12.98 tps, while the current Ethereum public network throughput rate is about 7–11 tps. Therefore, this paper can meet throughput requirements.

In this paper, in the nonchain structure of the data flow scenario, to speed up the efficiency of data tracing in each domain, the Bloom filter search mechanism is introduced. Under different conditions of m/n and k, the comparison of traceability search efficiency is shown in Fig. 11. Where m/n is the ratio of the number of bits m to the number of inserted elements n, and k is the number of mapping functions.

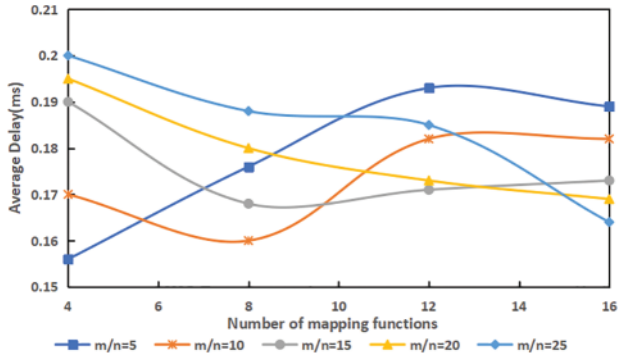

Figure 11: Average traceability search delay

The results show that the search delay for the data tracing varies with the number of mapping functions. When the relation is close to satisfying $k = (m/n)$ Ln2, the tracing delay is the lowest, that is, the corresponding tracing query efficiency is the highest, which satisfies the theoretical analysis results. When the Bloom filter is used for traceability query, the intra-domain search is first conducted according to the key value. When all mapping functions are displayed as 1, it indicates that there may be target traceability data in the domain. However, in the process of Blum filter search, due to the existence of a misjudgment rate, it may enter the domain without target data for a query, resulting in extra time cost and an increase of traceability delay. As the ratio of m/n gradually increases, when it approaches $k = (m/n)$ Ln2, the Blum filter has the lowest misjudgment rate, that is, the lowest tracing delay and the highest efficiency. As the ratio of m/n continues to increase, $k < (m/n)$ Ln2 appears, thus reducing the traceability efficiency.

7 Conclusion

In this paper, A cross-domain data traceability mechanism is proposed for the current problems in the field of blockchain traceability, such as cross-chain information collaboration and cross-domain access authentication. Firstly, we use the improved PageRank algorithm to design a consensus mechanism suitable for cross-domain scenarios. On this basis, we design a cross-domain architecture model and give the cross-domain identity authentication mechanism. According to the cross-domain architecture, we design a detailed cross-domain access process and combine the Bloom filter to give the traceability algorithm under the cross-domain scenario. Finally, through safety analysis and experimental evaluation, the safety and effectiveness of the traceability mechanism are verified.

Funding Statement: The authors received no specific funding for this study.

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

References

[1] R. Qiao, Y. Cao and Q. X. Wang, "IOT dynamic data based on the alliance chain traceability system," *Journal of Software*, vol. 30, no. 6, pp. 1614–1631, 2019.

[2] Z. Zhang, G. Wang, J. Xu and Y. Du, "Survey on data management in blockchain systems," *Journal of Software*, vol. 31, no. 9, pp. 2903–2925, 2020.

[3] O. Novo, "Blockchain meets IoT: An architecture for scalable access management in IoT," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1184–1195, 2018.

[4] I. Sukhodolskiy and S. Zapechnikov, "A blockchain-based access control system for cloud storage," in *Proc. of the 2018 IEEE Conf. of Russian Young Researchers in Electrical and Electronic Engineering*, New Jersey, NJ, USA, pp. 1184–1195, 2018.

[5] Y. Zhu, Y. Qin, G. H. Gao, Y. Shuai, W. Chen *et al.,* "TBAC: Transaction-based access control on blockchain for resource sharing with cryptographically decentralized authorization," in *Proc. of the 2018 IEEE 42nd Annual Computer Software and Applications Conf.*, New Jersey, NJ, USA, pp. 535–544, 2018.

[6] H. Es-Samaali, A. Outchakoucht and P. J. Leroy, "A blockchain-based access control for big data," *Journal of Computer Networks and Communications*, vol. 5, no. 7, pp. 137–147, 2017.

[7] M. Bartoletti, A. Bracciali, S. Lande and L. Pompianu, "A general framework for blockchain analytics," in *Proc. of the 1st Workshop on Scalable and Resilient Infrastructures for Distributed Ledgers*, Las Vegas, Nevada, USA, pp. 11–15, 2017.

[8] C. Enrico, N. Serena, N. Antinino, U. Domenico and V. Luca, "A two-tier Blockchain framework to increase protection and autonomy of smart objects in the IoT," *Computer Communications*, vol. 181, pp. 338–356, 2022.
[9] S. Li, B. Y. Song, D. Li and J. L. Wang, "A composite blockchain related event traceability method for financial activities," *Computer Science*, vol. 49, no. 3, pp. 346–353, 2022.
[10] C. D. Zhang, B. S. Wang and W. P. Deng, "Based on the technology of side chain supply chain traceability system design," *Computer Engineering*, vol. 45, no. 11, pp. 1–8, 2019.
[11] F. Tian, "A supply chain traceability system for food safety based on HACCP, blockchain & Internet of Things," in *Int. Conf. on Service Systems and Service Management (ICSSSM)*, Dalian, China, pp. 1–6, 2017.
[12] Q. Lin, H. Wang, X. Pei and J. Wang, "Food safety traceability system based on blockchain and EPCIS," *IEEE Access*, vol. 7, pp. 20698–20707, 2019.
[13] L. Li, Z. K. Yang, C. Zhang, Y. Wu and Y. Chen, "Pattern classification of block chain medicine traceability model design," *Computer Engineering and Design*, vol. 42, no. 12, pp. 3555–3562, 2022.
[14] P. Ruan, G. Chen, T. A. Dinh, Q. Lin and M. Zhang, "Fine-grained, secure and efficient data provenance on blockchain systems," *Proceedings of the VLDB Endowment*, vol. 12, no. 9, pp. 975–988, 2019.
[15] L. Page, S. Brin, R. Motwani and T. Winograd, "The PageRank citation ranking: Bringing order to the web," *Stanford Digital Libraries WorkingPaper*, vol. 9, no. 1, pp. 1–14, 1998.
[16] S. J. Wei, S. S. Li and J. H. Wang, "Cross-domain authentication protocol based on identity cryptosystem and blockchain," *Chinese Journal of Computers*, vol. 44, no. 5, pp. 908–920, 2021.
[17] R. Canetti, D. Shahaf and M. Vald, "Universally composable authentication and key-exchange with global PKI," in *Proc. of the IACR Int. Workshop on Public Key Cryptography (PKC)*, New York, USA, pp. 265–296, 2016.
[18] B. H. Bloom, "Space/time trade-offs in Hash coding with allowable errors," *Communications of the ACM*, vol. 13, no. 7, pp. 422–426, 1970.
[19] X. Fan and B. N. Niu, "New Blockchain Bloom filter based on blockchain application," *Journal of Computer Science and Exploration*, vol. 15, no. 10, pp. 1921–1929, 2021.