Hindawi
Security and Communication Networks
Volume 2022, Article ID 5804109, 19 pages
https://doi.org/10.1155/2022/5804109

Research Article

Antitampering Scheme of Evidence Transfer Information in Judicial System Based on Blockchain

Jingjing Guo[1], **Xuliang Wei**[1], **Yuling Zhang**[1], **Jianfeng Ma**[1], **Huamin Gao**[1], **Libo Wang**[2] and **Zhiquan Liu**[2]

[1]*School of Cyber Engineering, Xidian University, Xi'an710071, China*
[2]*College of Cyber Security, Jinan University, Guangzhou 510632, China*

Correspondence should be addressed to Jingjing Guo; jjguo@xidian.edu.cn

Received 19 August 2021; Revised 23 November 2021; Accepted 27 December 2021; Published 29 January 2022

Academic Editor: Kuo-Hui Yeh

In the process of handling criminal cases, it is crucial to avoid evidence tampering and ensure the integrity, consistency, and nonrepudiation of evidence transfer records, which is highly related to the fairness and credibility of the judiciary. To address this problem, we propose a consortium blockchain network to record evidence transfer events among different departments of China's judicial system. We design the format of a transaction and a block. In addition, the smart contracts for three types of transactions are also proposed. The Raft consensus algorithm is adopted to accomplish the consensus process. A security analysis shows that the proposed scheme can achieve the design goal (the integrity, consistency, and nonrepudiation of evidence transfer records stored in blockchain). Furthermore, a set of experiments were conducted to analyse the performance of the proposed scheme. The experiments results show that the throughput of the system is proportional to the send rate within a certain threshold. The latency decreases with increasing send rate if the send rate is within a certain threshold. Peer nodes in the system consume the most storage and communication cost. The values of block size and block generation interval time have a slight influence on the performance of the system.

1. Introduction

Evidence is the cornerstone of criminal proceedings and guarantees the quality of case handling. Recently, a number of criminal unjust and wrongly decided cases have been exposed in China; consequently, the prevention and correction of unjust, false, and wrongly decided cases has once again become a hot topic of public concern under the background of establishing judicial credibility. Therefore, China's central Political and Legal Commission requires judges, prosecutors, and police to be responsible for life for the quality of case handling within their responsibilities. An investigation shows that there are many reasons for unjust and false cases, but most of these cases have had serious evidence problems.

In the process of handling criminal cases, the relevant evidence needs to be transferred among multiple departments of the judicial system. In a traditional trial, the evidence transfer process is managed entirely manually, with the involved entities being required to fill in and sign paper documents that accompany the evidence. The rapid development of electronic information technology in recent years has promoted the modernization of trial procedure in China. In 2016, the Supreme People's court began to deploy and promote the in-depth application of electronic files in courts across the country. As of 2019, 3363 courts in China have built an electronic file generation system, and 67% of the cases in the country have generated electronic files along with the cases and transferred them for application [1]. In Western countries, with the facilitation of information system and Internet technologies, the judicial system has also been able to combine the information systems of various departments, institutions, and units to create a new approach for handling cases online.

In China, the general framework of the electronic case handling network includes a central node (the Political and

Legal Committee) and four platform nodes (the public security bureau, the procuratorate, the court, and the judiciary). For supervision and inspection, the information transferred among different platforms should be transmitted through the central nodes. In the system, each node records the evidence transfer event into a log. In the case of a log record inconsistency between parties, the log record of the Political and Legal Commission shall prevail.

For the above system to be used in the trial process of criminal cases, the standardized use and circulation of evidence within the judicial system must be ensured. Specifically, the following security requirements need to be met: (1) (*consistency*) the evidence transfer events are recorded and stored accurately, and the records are consistent among departments; (2) (*integrity*) the evidence transfer records cannot be tampered with artificially; and (3) (*nonrepudiation*) the recorded events cannot be denied by the involved entities.

The network architecture described above cannot ensure the security of evidence transfer records, since the nodes in the network cannot guarantee the integrity of their log records. When the evidence transfer records stored among different nodes are inconsistent, the correct transfer scenario cannot be identified, accountability cannot be ensured, and the unforgeability and the nonrepudiation of the evidence transfer event cannot be guaranteed.

Based on the above analysis, we need an effective approach for guaranteeing the consistency, integrity, and nonrepudiation of the evidence transfer recorded among departments within the judicial system. In recent years, as an emerging information technology, blockchain has attracted increasing attention. According to Wikipedia, blockchain is a distributed database technology. By maintaining the chain structure of data blocks, this technology can maintain the continuous growth of data records that cannot be tampered with. That feature makes blockchain a promising technology in multiple domains, including the judicial system, credit inquiries, and so on. In judicial system, distinctive features of the blockchain, such as traceability, post-auditing, and data-tampering prevention, make it a promising tool which can significantly improve the credibility and authenticity of e-evidence.

The established blockchain systems applied in the judicial domain are used mainly to solve the trust problem between the judicial system and the external world. However, to the best of our knowledge, the trust problem among departments within the judicial system has not been considered in the Internet judiciary system. To solve this problem, in this paper, we propose a blockchain-based evidence transfer management model for criminal cases in China's judicial system. In the proposed framework, the evidence transfer records of criminal cases are stored in a judicial system consortium blockchain to ensure that the records cannot be tampered with and that the copies of evidence transfer records maintained by all departments are consistent. In this way, various problems existing in the established criminal case handling system can be solved, namely, the vulnerability to tampering of evidence transfer records, the inconsistency of evidence transfer records after being tampered with, and the difficulty of determining liability when the records are inconsistent.

In summary, this paper makes the following contributions:

(1) In this paper, according to the transfer rules of criminal evidence in the judicial system of China, we established a blockchain-based criminal case evidence transfer record management system. The proposed system does not rely on a trusted third party to ensure that the evidence transfer records are consistency, integrity, and nonrepudiation. Raft consensus algorithm is adopted to record the verified transactions in the blockchain.

(2) We design the transaction structure for storing the evidence transfer events. In addition, we design the endorsement policy and smart contract for the considered three types of transactions (evidence sending, evidence revocation, and evidence transfer record query) to guarantee the consistency, integrity, and nonrepudiation of evidence transfer events.

(3) The security of the proposed scheme is verified in terms of integrity, consistency, and nonrepudiation of the evidence transfer records stored in the blockchain. The experimental prototype is implemented and the performance of the proposed system is measured in terms of throughput, latency, and resource overhead.

The remainder of this paper is organized as follows. Section 2 reviews the related works, followed by the background knowledge about the proposed system in Section 3. The proposed scheme is presented in Section 4, and a security analysis and experimental performance evaluation are presented in Sections 5 and 6, respectively. Finally, Section 7 concludes this paper.

2. Related Works

Advancements in information technology in recent years have enabled an increasing number of case handling processes, including evidence transfer, to be conducted online. As the most crucial basis for judgment, the complete lifecycle of the evidence should be strictly recorded to guarantee its integrity, authenticity, and auditability. Great efforts have been made to enhance the integrity, authenticity, and auditability of the evidence in both judicial circles and informatics.

Richter et al. pointed out that digital evidence is considered admissible in the court of law if it meets the following criteria: authentic, complete, reliable, and believable [2]. Cosic et al. proposed a digital evidence management framework (DEMF) that could improve the chain of custody of digital evidence [3]. The authors used a hash function to generate a digital fingerprint of evidence. Then, the digital fingerprint and biometric authentication are used to identify the persons who handled the evidence. They also suggested the addition of timestamps to guarantee the integrity of the evidence and chain of custody [4]. Prayudi et al. introduced a

model of digital evidence cabinets (DECs) to implement the digital evidence handling and chain of custody. This framework consists of three main components: a digital evidence management framework for handling the interactions of investigators, a tag cabinet to represent the digital evidence cabinet, and access control and secure communication to support trust-based computing [5]. The authors also proposed a digital evidence cabinet-based framework to support digital evidence handling according to the regulations in Indonesia [6]. However, in these studies, the authors only provided solutions for digital evidence handling on an abstract level. They did not describe the proposed frameworks in detail. Furthermore, these frameworks do not consider the integrity or nonrepudiation of the digital evidence handling records. If some false digital evidence handling records are added to the logging system or recorded events are deleted from the logging system, the chain of custody will be broken. This will pose a hidden danger regarding the occurrence of unjust and false cases, thereby damaging the judicial credibility of the country.

With the emerging and rapid development of blockchain technology, the properties of blockchain render it suitable for the formation of a digital evidence chain of custody. Researchers have proposed blockchain-based schemes for recording digital evidence that render it almost impossible to change the digital evidence recorded on the blockchain. Fisher Justin et al. proposed an authentication and verification method of digital data using blockchain technology [7]. In this method, digital data are hashed to produce a hash fingerprint/signature and submitted to the Bitcoin blockchain so that the digital data can be verified without the involvement of a third party. This system can only guarantee the integrity of the evidence but cannot record or verify the compliance of operations conducted on the evidence. Moreover, since the Bitcoin blockchain is a permissionless blockchain, anyone can join the blockchain network and submit digital data to the blockchain; furthermore, its anonymity renders it unsuitable for the judicial system. Wenqi Yan et al. proposed a blockchain-based digital evidence chain of custody [8]. The evidence related to a certain case should be signed by 50% of the people involved in the case before it can be submitted to the blockchain. The evidence transfer event will be stored on the blockchain by recording the transfer time and the new owner of the evidence. The authors regarded the evidence handling by different entities as the same, which was inconsistent with reality. Furthermore, digital evidence handled online includes increasingly many video files, which occupy a substantial amount of storage space; hence, the storage of such evidence on the blockchain is infeasible. Auqib Hamid Lone et al. proposed forensic-chain, a blockchain-based digital forensics chain of custody with PoC in Hyperledger Compose [9]. As in reference [8], the authors did not consider the differences in evidence handling among different parties. For example, they assumed that evidence could be transferred to any entity in the blockchain network.

Based on the above analysis, most studies used blockchain to record the evidence and its state changes (such as the owner and transfer time), but did not consider the related regulations governing changes in the state of evidence. Hence, there is almost no access control on state changes of the evidence. Furthermore, the ever-increasing amount of video evidence requires massive space and is not suitable to be recorded on the blockchain. To prevent irregularities within the judicial system and to ensure that problems and responsible parties can be easily identified when disputes occur, we propose a permissioned blockchain system for recording criminal evidence transfer events within the judicial system to decrease the occurrence of unjust and false cases due to the unfair operation of the judicial system.

3. Preliminaries

This section provides the necessary background knowledge for this paper, which includes regulations for criminal evidence transfer in China's judicial system and blockchain technology.

3.1. Criminal Evidence Transfer Process in Chinese Judicial System. In China, the processes of handling criminal cases mainly include investigation, prosecution, and trial. The involved organizations within the judicial system mainly include the Public Security Bureau, the procuratorate, and the court, and the main supervision organization is the Political and Legal Commission. The general handling process of a criminal case is as follows. First, the public security organ investigates the case. After the investigation is completed, the file materials and evidence are transferred to the procuratorate for examination. Then, the procuratorate examines the case and checks the relevant materials of the case that have been sent by the public security organ. If any material is deficient, it will be returned to the public security organ for supplementation. If the evidence is accurate and sufficient, and the procuratorate considers it is necessary to investigate the criminal responsibility, the procuratorate will put forward a public prosecution to the court. During this process, the relevant evidence and materials of the case should be submitted to the court for review. The court will decide whether to hold a court session after examination. To ensure the efficiency of case handling, the material and evidence transfer needs to be handled by the relevant personnel with strict time regulations. As a supervision and guidance unit, the Political and Legal Commission can supervise and inspect the judicial system's case handling process, including the evidence transfer.

3.2. Blockchain. The blockchain is a computational paradigm that emerged with the Bitcoin protocol in 2008 [10]. It is essentially a dependable distributed ledger that stores transactions in a chain of chronological blocks that are linked via hash values [11–19]. The ledger is enforced with cryptography and carried out collectively in a peer-to-peer network. Generally, blockchains can be classified into two types, namely, permissionless blockchain (e.g., Bitcoin [10] and Ethereum [20]) and permissioned blockchain. Permissioned blockchains, which consist of consortium

blockchains (e.g., Hyperledger Fabric [21]) and private blockchains, are only available to members who have been granted authorization.

By deploying smart contracts on a blockchain network, the viewable contract obligation codes are triggerable by granted members, and the trustworthiness of execution results appended to the blockchain will be ensured by the consensus algorithm [22]. The typical consensus mechanisms used in permissioned blockchains include BPFT [23], Raft [24], and so on.

As previously discussed, we need to establish a blockchain system to record the evidence transfer events of criminal cases among different departments within the judicial system. Hence, a consortium blockchain is more appropriate for our system because it only allows the authorized entities within the judicial system to join the system and be involved in the transactions. Moreover, all transactions can only be checked within the justice system, in accordance with the criminal case handling regulations.

4. Proposed Scheme

This section presents a blockchain-based system for the secure recording of evidence transfer events among different departments of the judicial system. First, we will present the threat assumption and design goals of the system. Then, we will present the complete framework of the proposed system. Afterward, we will describe each component of the blockchain system, including the block format, identity management scheme, smart contracts, and consensus algorithm.

4.1. Adversary Model and Design Goals. In this paper, we assume the following: (1) all nodes in the system may be subjected to external or internal attacks, such as forgery, denial, or modification of evidence transfer records by the sender or receiver of evidence, and ultra vires operations (such as ultra vires revocation of evidence). (2) All nodes in the system may fail (e.g., by failure to process tasks and provide a response), but less than half of the nodes may fail simultaneously. (3) The nodes in the system will not collude with each other.

To eliminate the security vulnerabilities of the current centralized online case handling system and to build trustworthy evidence transfer records among departments of the judicial system, our system has the following design goals:

(i) *Full decentralization*. No single party can control the system.

(ii) *Tamper-proofing*. No single party can tamper with the evidence transfer record stored on the blockchain.

(iii) *Consistency*. The system can guarantee the consistency of the evidence transfer records among the departments within the judicial system.

(iv) *High robustness*. The system can still perform well under the failure of a certain number of nodes in the blockchain network.

4.2. System Architecture. The architecture of the proposed system is illustrated in Figure 1. The blockchain network comprises four organizations, namely, the Public Security Bureau, the procuratorate, the court, and the Political and Legal Commission, which are denoted as Org1, Org2, Org3, and Org4, respectively. Each organization has two peer nodes, either of which can act as an endorser or a master node, and each of which is a committer. Applications (clients) are deployed within each organization. These applications can communicate with the peer nodes of the organization to which they belong. They can send transaction proposals, which are processed by the blockchain network to generate new blocks, and receive the information returned by the blockchain network (such as the evidence transfer record stored on the blockchain). The evidence is transferred over the Internet rather than via the blockchain. Here, compared with the traditional electronic case handling network mentioned in Section 1, the judiciary is not involved in the blockchain network according to the criminal evidence transfer process in the Chinese judicial system. In addition, the number of peer nodes within each organization can also be set to one or more than two.

Figure 2 illustrates the operational flow of the proposed blockchain network. We can see that there are four organizations participating in the network, namely, the Public Security Bureau, the procuratorate, the court, and the Political and Legal Commission. The clients of these four participants are able to carry out three kinds of applications, namely, sending, revocation, and query. Figure 2 shows that the Public Security Bureau and procuratorate can initiate a sending application, the procuratorate, and the court are allowed to carry out a revocation application, and all clients of the four participants are able to conduct the query application. All entities of these four participants are managed by an identity management module. Entities with different identities are assigned different permissions to participant in the blockchain network.

There are four processes from a client initiating a certain type of transaction (application) to the transaction being written into the blockchain. The four processes are 1. initiating a transaction proposal, 2. Endorsement, 3. Consensus, and 4. Adding blocks to the blockchain, corresponding to the digital label next to the dotted arrow in Figure 2. Each process is described in detail as follows:

(1) *Initiating a Transaction Proposal*. When a client finishes a business related to the evidence transfer, it needs to record this evidence transfer event to the blockchain. To do this, the information about this event needs to be constructed as a transaction proposal and sent to the endorsement nodes for endorsement.

(2) *Endorsement*. When endorsement nodes receive a transaction proposal from a client, the corresponding smart contract will be executed to get the transaction simulation result. Based on the endorsement logic, if an endorser decides to support the transaction proposal, it will sign the read-write set generated by the simulated transaction and send it back to the client after signing.

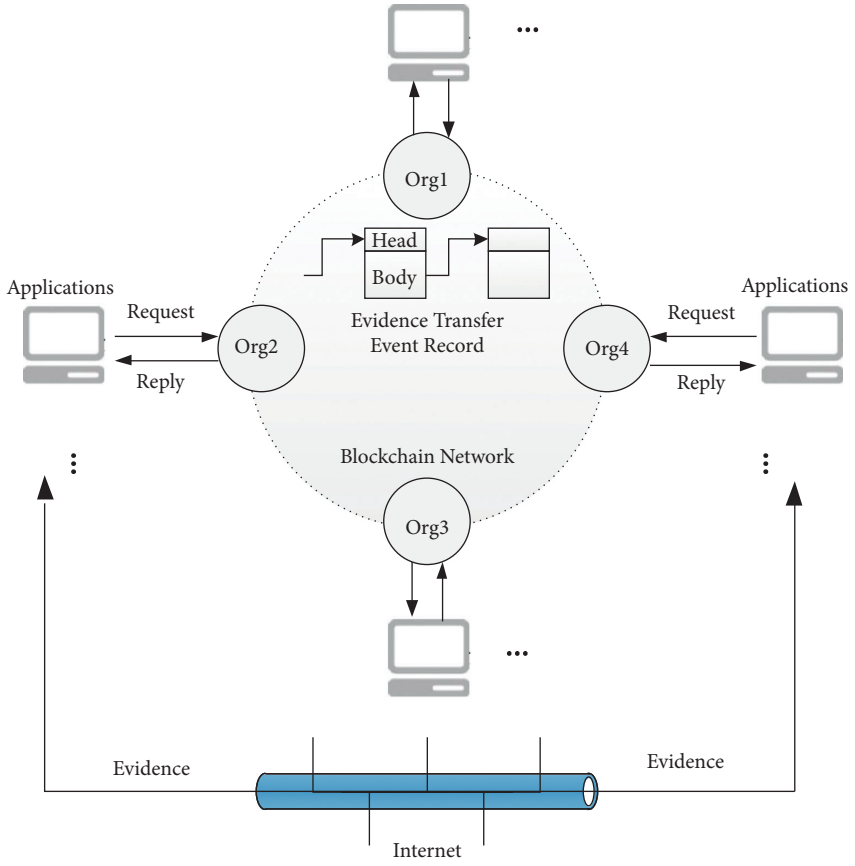

Figure 1: The architecture of the proposed evidence transfer management framework.

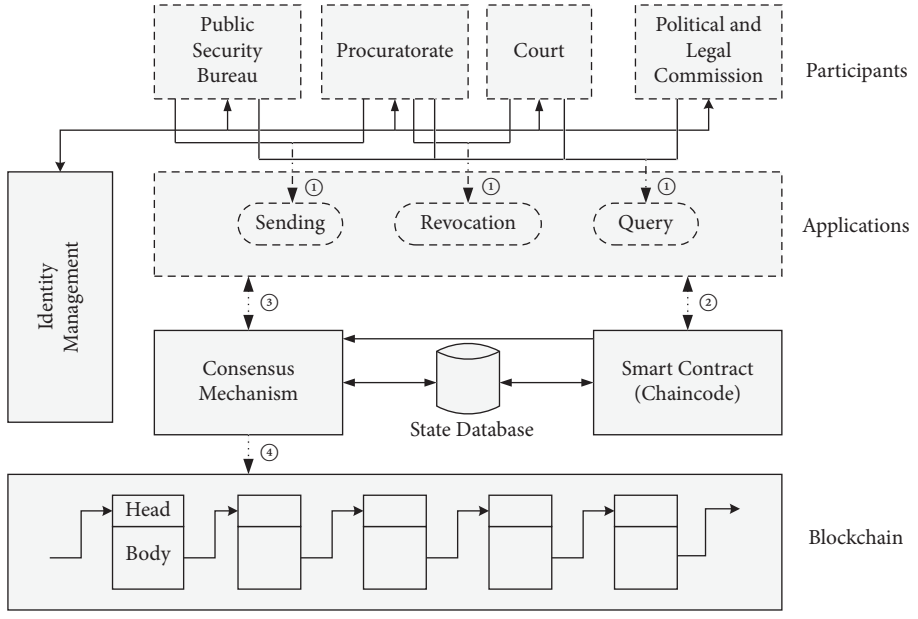

Figure 2: The operational flow of the proposed blockchain network.

(3) *Consensus*. When a transaction proposal has passed the endorsement strategy, the client that initiated the proposal would generate the corresponding transaction and send it to the ordering service to store the transaction on the blockchain. During this process, we adopt the Raft consensus algorithm to achieve the consensus and ensure the consistency of the ledger among all nodes.

(4) *Adding Blocks to the Blockchain*. After transactions sorted by the ordering service, they will be generated as a block. The leader node is in charge of the commitment of the new generated block. The committed block will be added to the blockchain replica of all peer nodes. In addition, the state database in Figure 2 is also a part of the ledger. In the implementation of the smart contract and the consensus mechanism, nodes in the blockchain network need to access the state database for the corresponding read-write operations to finish their tasks.

4.3. Block Format. In this section, we will present the formats of a transaction and a block on the blockchain. The format of a transaction proposal is presented in Table 1, where *type* represents the type of the transaction; *sending*, *revocation*, and *query* are represented as 1, 2, and 3, respectively; *timestamp* is the beginning time of an evidence transfer event (e.g., the evidence sending time and the evidence revocation time); *recID* is the identity of the receiver in an evidence transfer event; and *txID* represents the transaction ID, which is composed of the evidence ID, the sender's ID, and the transaction reference number. Moreover, the *prooflist* stores evidence IDs and their corresponding hash values, where the evidence ID is composed of the case ID and the evidence serial number; the variable *ack* is the acknowledgement message from the receiver of an evidence transfer event; and the field *note* can store supplemental information about the transaction.

Table 2 presents the format of a block. A block consists of three main parts, namely, the block header, block data, and block metadata. The block data part stores several transactions as a Merkel tree. The hash value of the Merkel tree's root is stored in the block header part. The hash value of the previous block is also stored there. These hash values form a chain to connect the adjacent blocks and can be used to verify the integrity of the blocks.

4.4. Identity Management and Access Control. In the proposed framework, all participants in the blockchain network should obtain entry permission. We deployed an identity management module for key generation and certificate management for all participants.

The identity management scheme is illustrated in Figure 3. A CA (Certificate Authority) is deployed in each organization to generate certificates for all the members of the organization. All members generate their own public and private key pair, register with the administrator, and obtain a certificate from the corresponding CA. Members of the same organization have the same MSP (Membership Service Provider), so they have the same root certificate and administrator for authenticating each other and share data. According to the architecture of the proposed system and Hyperledger Fabric, each organization will create a certificate for one administrator, two peer nodes, one orderer, and several clients. Meanwhile, we establish a channel MSP, which includes the information of each organization's MSP,

Table 1: Data structure of a transaction proposal.

```
typedef tx struct{
    int type;
    int timestamp;
    int recID;
    int txID;
    map string I D, string hash prooflist;
    string ack;
    string note;
}
```

Table 2: Format of a block.

```
Block Header
{
    Block Number
    Current Block Hash//Hash of the Merkle Tree's Root
    Previous Block Hash
}
Block Data
    a Set of Transactions//Stored as a Merkle Tree
Block Metadata
{
    Block Creation Time
    Writer's Certificate and Signature
}
```

so that the peer nodes and orderers of the four organizations can share data in this channel and authenticate the channel participants.

To achieve access control, we set corresponding roles and attributes to different types of entities in the network, which can be reflected in their certificates. As mentioned above, there are four kinds of roles in the system, namely, administrator, peer node, orderer, and client. The authority of the administrator is to register members that are willing to enter the corresponding MSP. The peer node can submit transactions to the blockchain network and authenticate the identity of entities in the network. The orderer's authority is to sort the unpackaged transactions to generate a block. The client can only submit transaction proposals to the endorser and broadcast the endorsed transaction to the orderer. In addition, to facilitate the implementation of the chaincode, which will be described in the following sections, we set priorities that differ among organizations. The priorities of Org1, Org2, Org3, and Org4 (Figure 1) are incremental and are represented by the numbers 1, 2, 3, and 4, respectively.

4.5. Endorsement. In the endorsement stage, the endorsing peer checks the validity of the transaction proposal sent by a client, then simulates the execution chaincode (smart contract) to obtain the transaction result, and finally judges whether to support the transaction proposal according to the endorsement logic. If it is decided to support the transaction proposal according to the endorsement logic, the endorsing peer will sign the read-write set generated by the simulated transaction and send it back to the client after signing.

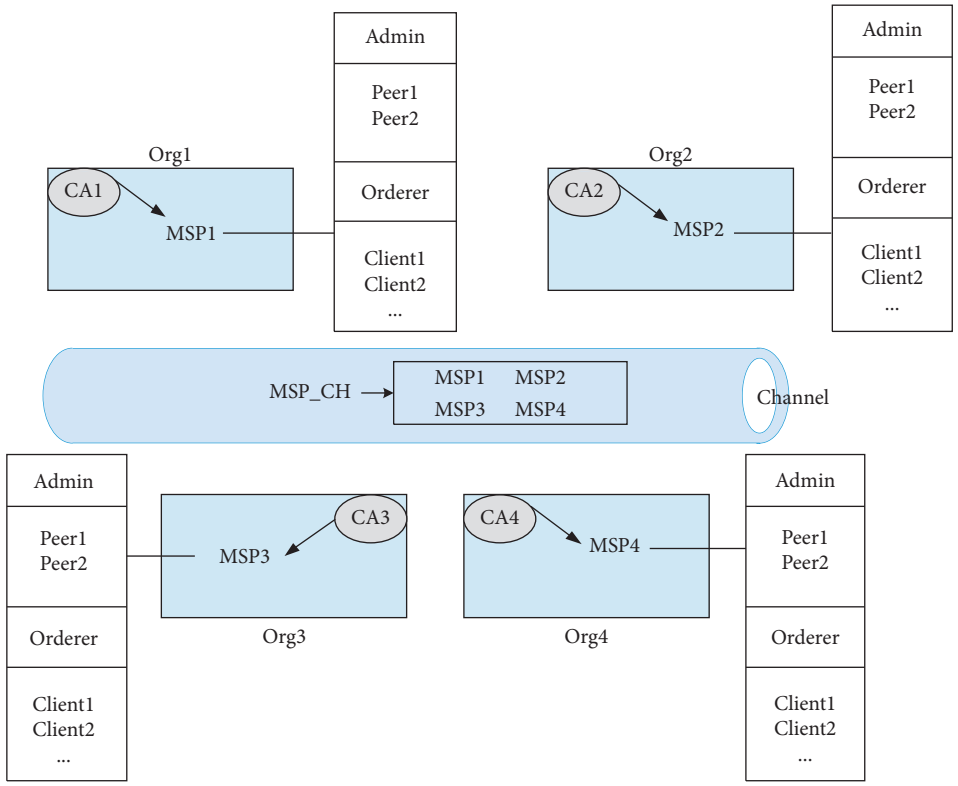

Figure 3: Identity management scheme.

Before we introduce the chaincode of the proposed system, we define the content of the state database that maintains the status of the ledger to facilitate the fast query of the application. The structure of the state database, which is a key-value-based database, is presented in Table 3. Each entry contains two fields, namely, key and value, which represent certain information related to an item of evidence. For a specified evidence item, the field "key" contains the evidence ID, and the field "value" contains the hash value of the evidence, the values of three indicators (TBC, valid, and flag), and the list of clients that possess this evidence. In the following, we will use symbol $SD_{ev}.f$ to denote the value of field f for evidence ev and symbol EV_{SD} to denote the set of evidence stored in the state database.

As we discussed in Section 3, the events related to evidence transfer within the judicial system mainly include evidence submission, evidence rejection, and evidence query, which correspond to three types of transactions in the blockchain system, namely, evidence sending, evidence revocation, and evidence transfer record query, respectively. The concrete business is processed through the Internet, which does not require the participation of the blockchain. When the business has been finished, the initiator will submit the corresponding transaction to record this event in the blockchain. In the following, we introduce the three types of transactions and present their smart contract (chaincode) in the form of algorithm pseudocode.

(i) *Evidence Sending.* The evidence can only be sent from the Public Security Bureau to the procuratorate or from the procuratorate to the court; hence, the sending transaction proposal can only be submitted by the clients that belong to the Public Security Bureau and procuratorate. In addition, to prevent the forgery of transactions, the originator of the transaction proposal must have the acceptance confirmation of the recipient.

(ii) *Evidence Revocation.* If the received evidence is considered defective by the recipient, it will be revoked and required to be supplemented by the sender. Thus, this transaction can only be initiated by a client of the procuratorate or court. Moreover, the client who initiates the transaction could only revoke the evidence from its sender. If a revocation transaction is already stored on the blockchain, then the evidence included in this transaction is considered invalid.

(iii) *Evidence Transfer Record Query.* This transaction can be submitted by all clients of the system to gain access to information related to a specified evidence item or client.

For the evidence sending transaction, the process includes three phases: (1) sending evidence data off-chain, (2) generating the transaction proposal, and (3) endorsement. Figure 4 illustrates the process of the first phase. The symbols in Figure 4 are defined in Table 4. For the content of P in Table 4, we have $P = Ev|\text{transaction } ID|\text{receiver } ID$, where Ev is the evidence set included in this transaction and transaction ID and receiver ID are the ID of this transaction and the evidence receiver, respectively. In addition, we define $Ev = \langle Ev_i \rangle \, || i \in [1, n]$, and $Ev_i = \langle$evidence I

Table 3: Structure of the state database.

Field		Meaning
Key		Evidence ID
Value $e \in \{\text{Org1}, \text{Org2}, \text{Org2}\}$	Hash	The evidence's hash value
	TBC	1: the evidence has been sent by the procuratorate and the confirmation of the court is awaited; 0: the evidence is in other states.
	$valid_e$	1: the evidence is currently in the valid state in organization e; 0: the evidence is currently in the invalid state in organization e.
	$flag_e$	1: e owns the evidence; 0: e does not own the evidence.
	Owner list	owner1′$sI\,D$, owner2′$sI\,D$, ...

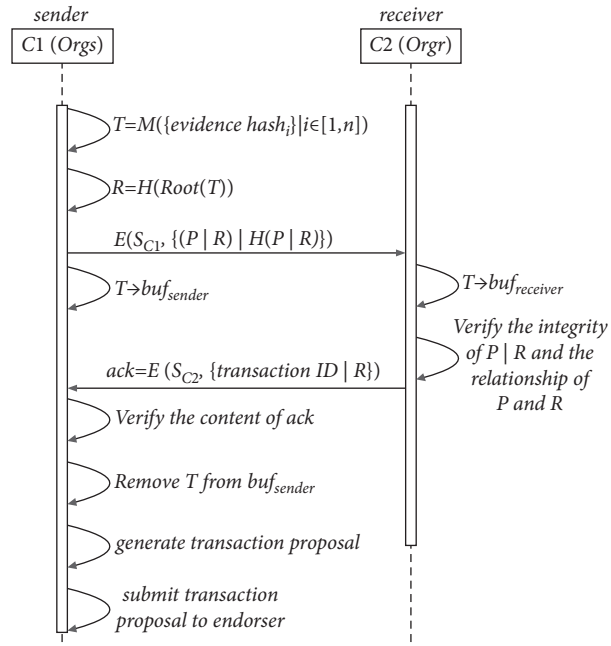

Figure 4: The process of sending evidence data off-chain between client *C1* and *C2*.

Table 4: Meaning of symbols in our scheme.

C1	The sender of the evidence
C2	The receiver of the evidence
evidence hash$_i$	The hash value of the *ith* evidence included in the transaction
N	The number of evidences in the transaction
M (X)	A function that generates a Merkel tree for data set *X*
T	A symbol representing a Merkel tree
Root (T)	A function returning the root of Merkel tree *T*
H	A hash function
P	The data of this transaction.
Transaction ID	The ID of this transaction
R	A symbol whose value is equal to *H (Root (T))*
E (r1, r2)	An encryption function that uses key *r1* to encrypt plaintext *r2*
S_C	The private key of client *C*
buf$_{sender}$	A buffer storing the information that a client sends out but has not confirmed by the receiver.
buf$_{receiver}$	A buffer storing the information that a client receives but has not recorded on the blockchain.
ACK	The acknowledgement message of a transaction.

D|evidence hash⟩, evidence*I D* is the *ID* of this evidence, which is the same as the evidence ID stored in *tx*ID shown in Table 1, and evidence hash is the hash value of evidence data related the evidence *ID*. When client *C1* (which belongs to *Orgs*) wants to send a set of evidence *Ev* to client *C2* (which belongs to *Orgr*), it will send the

evidence data and wait for a response from the receiver. After receiving the *ack* message from the receiver, it will generate the transaction proposal (the format is presented in Table 1) and submit it to the endorsement peers.

When an endorser receives a sending transaction proposal, it will verify related constraints. If the transaction proposal is verified, it will execute the transaction, generate the read-write set, and sign the endorsement file. The chaincode of this process is shown in Algorithm 1. Symbol "D" appearing in line 2 means the decryption function which is used to decrypt the *ack* message by the receiver.

For the evidence revocation transaction, the chaincode is presented as Algorithm 2. The code from line 3 to line 12 corresponds to the validity checking phase of the endorser. In line 3, the symbol "$\text{rer.buf}_{\text{sender}}$" stores the evidences that have been sent to the corresponding receiver by the revoker, while they have not been confirmed by the receiver. In this case, the revoker cannot revoke these evidences directly, because these evidences have been sent to departments with higher priority and they should be revoked first by these departments. If the transaction proposal is proved valid, the code from line 21 to line 25 will complete the endorsement operation. The endorser will simulate the execution of the transaction. During this process, the endorser will access the state database to obtain the data that it needs (the read set), and after the simulation, it will obtain the updated state database entry (the write set). Then, the endorser will sign the transaction and read-write set as its response to the client.

Algorithm 3 describes the endorsement process for evidence transfer query transaction. Only the information of evidence that has a corresponding entry in the state database can be found. If the queried information is stored in a field of entry ID_{inq}, the endorser will directly return *inf* as its response; otherwise, transaction information related to ID_{inq} and that is stored on the local ledger will be returned to the inquirer.

For the transaction proposal of evidence sending and evidence revocation, the client must receive signed responses from all endorsers before it generates a transaction and sends it to the ordering service.

4.6. Consensus Algorithm. When a transaction proposal has passed the endorsement strategy, the client initiating the proposal will generate the corresponding transaction and send it to the ordering service to store the transaction on the blockchain. During this process, we adopt the Raft consensus algorithm [24] to achieve the consensus and ensure the consistency of the ledger among all nodes. All peer nodes in the blockchain network participate in the Raft consensus process. The whole process is composed of three phases: leader node election, block broadcasting, and ledger maintenance.

For the leader node election, each peer node can be in one of three states: follower, candidate, or leader. Initially, each node is in the follower state and has a random timer. When the timer times out, the node reaches the candidate state. At this time, the node sends voting requests to other nodes. When a candidate node obtains support from more than half of the nodes in the network (including itself), it is elected as the leader node. If more than one candidate receives the same number of votes simultaneously, they will make a new round of election requests. When a candidate node finds that a leader node has been generated, it will return to the follower state. The details of the leader election process can be found in [24].

As discussed above, a transaction that satisfies the endorsement strategy will be sent by the client to the ordering service. Figure 5 illustrates the complete procedure from the submission of a transaction by a client to the storage of the transaction on the blockchain. In Figure 5, we have a client, a leader node (A), and two follower nodes (B and C). In fact, there can be more follower nodes, but for simplicity, we only set two follower nodes here. The leader node and follower nodes constitute the whole Raft network which will finish the consensus process and the ledger maintenance.

From Figure 5, we can see all nodes in the Raft network have a "Validity check" module, a "Log" module, a "State machine" module, and a "Ledger." Furthermore, the leader node has an "Ordering" module and a "Block generating" module additionally. In general, the received transactions should be sorted by the leader node first and then packaged into a block. Then, the newly generated block will go through four stages, namely, verified by the "Validity check" module, stored in the "Log" module, executed by the "State machine" module, and finally added to the blockchain. The details of each step in this procedure are described as follows with the step numbers corresponding to the numbers in Figure 5:

(1) A client sends a request *R (t, ef)* to the Raft network, where t is a transaction and *ef* is the endorsement file related to t. Request R will be addressed initially by the leader node. If the receiver of R is not the leader, it will also be routed to the leader node.

(2) Received transactions are sorted in chronological order by the "Ordering" module of the leader node. When sufficiently many transactions have been received for the generation of a block, the "Block generating" module of the leader node will generate a block.

(3) The newly generated block will be sent to the "Validity check" module which will check the validity of each transaction in the block.

(4) If the block is valid, it will be stored in the log of the leader node as an uncommitted block.

(5) The newly generated block is also sent to all follower nodes. The "Validity check" module of the follower nodes will check the validity of the received block.

(6) If the block is verified to be valid, it will be stored in the logs of the follower nodes.

(7) After a block has been stored in the log of a follower node, the node will send a response message back to the leader node.

(8) When the leader node receives responses from all follower nodes, the uncommitted block stored in its log will be executed by the state machine.

Input: transaction proposal TP;
Output: endorsement result;
(1) **if** ($C1 \in$ Org1&&$C2 \in$ Org2) ||($C1 \in$ Org1&&$C2 \in$ Org3) **then**
(2) ACK = $D(P_{C2}, TP.\text{ack})$//decrypt ACK with receiver's public key
(3) **if** TP.ack is signed by TP.recID **then**
(4) **if** $ACK.transactionID == TP.txID$ &&
(5) $ACK.R == H$ ($Root$ ({$M(TP.prooflist.hash)$})) **then**
(6) simulate the transaction;
(7) calculate the read-write set;
(8) sign the endorsement file;
(9) **return** signed endorsement file;
(10) **end if**
(11) end if
(12) else
(13) **return** false**;**
(14) end if

Algorithm 1: Chaincode of evidence sending.

Input: $EV = \{ev1, ev2, \ldots, ev_n\}$;//the set of IDs of the evidence to be revoked
rer,//revoker's ID
org_r,//the organization of the revoker
org_e,//the organization of the revocation recipient
Output: the endorsement result;
(1) int $\text{check}_{ev}[n+1] = 0$**;**//state array for EV
(2) **for** $i = 1 \sim n$ do
(3) **if** $ev_i \in \text{rer.buf}_{\text{sender}}$ **then**
(4) **return** false;
(5) **else**
(6) **if** $ev_i \in \{SD_{ev}.\text{key}|ev \in EV_{SD}\}$ **then**
(7) **if** $SD_{\text{evi}}.\text{flag}_{\text{org}_r} == 1$ **then**
(8) **for** all organization org that has higher priority than org_r **do**
(9) **if** $SD_{\text{evi}}.\text{valid}_{\text{org}} == 0$ **then**
(10) **if** $SD_{\text{evi}}.\text{flag}_{\text{org}_e} == 1$&& $SD_{\text{evi}}.\text{valid}_{\text{org}_e} == 1$ **then**
(11) **if** rer $\in SD_{\text{evi}}$.ownerlist **then**
(12) $\text{check}_{ev}[i] = 1$;
(13) **end if**
(14) end if
(15) end if
(16) end for
(17) end if
(18) end if
(19) end if
(20) end for
(21) **if** $\sum_{i=1}^{n} \text{check}_{ev}[i] == n$ **then**
(22) simulate the execution of the transaction;
(23) generate the read-write set;
(24) sign the proposal and the read-write set;
(25) **return** signed endorsement file;
(26) else
(27) **return** false**;**
(28) end if

Algorithm 2: Chaincode of evidence revocation.

Input: ID_{inq} //ID of the evidence being queried
inf//the concrete information of ID_{inq} that the querier wants to query
Output: the queried information or false
(1) **if** $ID_{\text{inq}} \in \{SD_{ev}.\text{key} | ev \in EV_{SD}\}$ **then**
(2) **if** inf $\in SD_{ID_{\text{inq}}}$'s field **then**
(3) **return** $SD_{ID_{\text{inq}}}$.inf
(4) **else**
(5) search the local ledger with ID_{inq};
(6) **return** information *inf* related to evidence ID_{inq};
(7) **end if**
(8) else
(9) **return** No information was found.
(10) end if

Algorithm 3: Chaincode of evidence transfer record query.

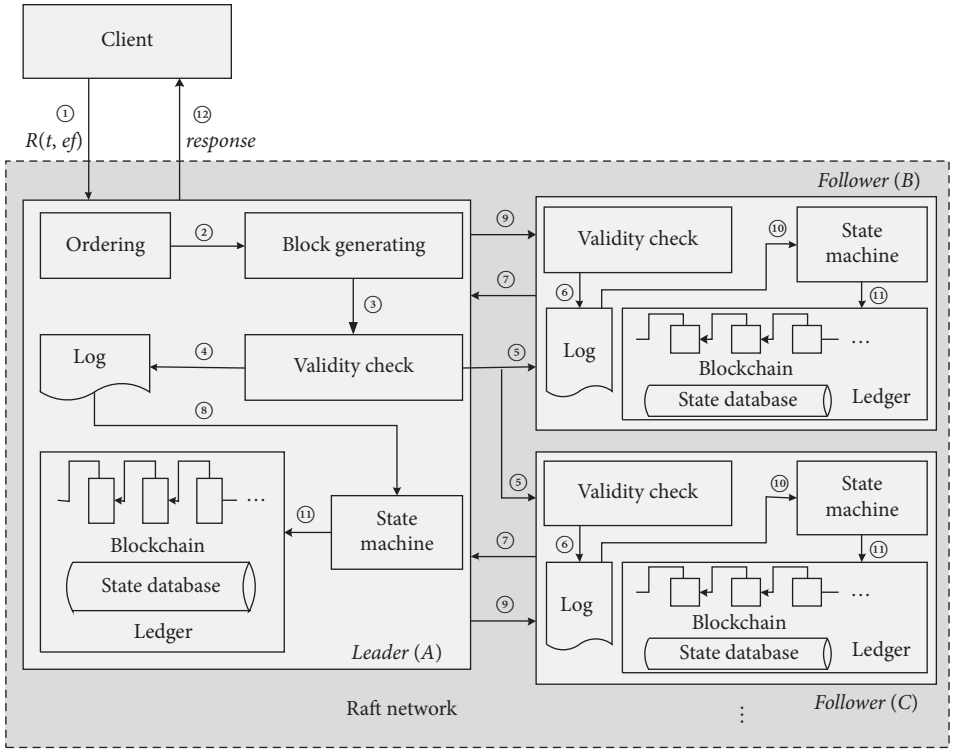

Figure 5: The process of consensus.

(9) After the leader commits the uncommitted block, it will send a message, along with its heartbeat RPC (Remote Procedure Call), to instruct the followers to commit the block.

(10) Each follower node will commit the uncommitted block to its state machine.

(11) The execution process of a follower's state machine is the same as that of the leader's state machine. During this procedure, the new block will be added to the blockchain. Moreover, for each valid transaction, its write set will be committed to the state database for update.

(12) The leader will return the result of the request *R (t, ef)* to the client.

5. Security Analysis

In this section, we will analyse the security of the proposed scheme in terms of integrity, consistency, and nonrepudiation.

5.1. Integrity

Lemma 1. *An evidence item involved in a transaction that is recorded on the blockchain cannot be tampered with during the transfer process.*

Analysis. For an evidence item included in a sending transaction, the sender of the evidence will encrypt the evidence data and the corresponding hash value with its

private key. If the evidence has been tampered with, the receiver can easily detect it by decrypting the received message and verifying the relationship between the evidence data and the corresponding hash value, as illustrated in Figure 4. If the verification result demonstrates that the evidence has been tampered with, the sender will not receive an *ack* message from the receiver, so the transaction will not satisfy the endorsement strategy.

For an evidence item included in a revocation transaction, the evidence data will not be transferred; thus, there is nothing that can be tampered with.

Lemma 2. *No transactions recorded on the blockchain can be tampered with.*

Analysis. All transactions stored in a block are organized as a Merkle tree, the root of which is calculated based on all the transactions. If any transaction is tampered with, the value of the Merkle tree's root will also be changed, which can be easily detected. Furthermore, this can result in a change in the hash value of the block, which is recorded in the next block. Thus, it is also easy to detect this change unless all blocks of the blockchain can be changed by the attacker, and the success rate of tampering with the whole blockchain is negligible.

5.2. Consistency

Lemma 3. *The local ledger stored in each node is consistent.*

Analysis. According to the consensus process, a newly generated block will be recorded in the log of each node before it is executed by the state machine. The difference in the log of each node caused by the change of leader node can be resolved by the Raft algorithm. Thus, all consensus nodes' logs must store the same blocks. The state machine of each node has the same execution process, so the execution result will lead to the same update to the state database and the blockchain.

5.3. Nonrepudiation

Lemma 4. *The validity of the evidence transfer records recorded on the blockchain cannot be denied by the involved clients.*

Analysis. For evidence sending events, the sender of the evidence will sign the evidence that it sends and can be verified by the receiver. Furthermore, the response of the receiver *ack* in Figure 4 is also signed with the receiver's private key. When the related transaction proposal is submitted to the endorser, the endorser will verify the identities of the sender and receiver. For evidence revocation events, the endorser will also check the identity of the revoker and the revocation recipient in the validity check phase.

Only when a transaction proposal is determined to be valid will the endorser sign the endorsement file for it. Then, it can be submitted to the consensus network for storage on the blockchain. For all the transfer events stored on the blockchain, the identities of the involved clients are verified by the endorser and recorded in the transaction; hence, the validity of the record cannot be denied.

5.4. Controllability and Auditability

Lemma 5. *The proposed scheme can implement fine-grained management during the evidence transfer process, which includes management of the permission of each client and the record rules of evidence transfer events.*

Analysis. For the permission of each client, each client that registers in its MSP can be authenticated when it communicates with other entities. If necessary, the certificate of a client can be revoked so that it cannot be authenticated. During the endorsement process, which includes checking whether the related client has permission to conduct a transaction, the endorsement strategy of each transaction also defines the rules.

If the record rules of evidence transfer events are changed, the format of a transaction, the structure of the state database, and the logic of the chaincode for each transaction can be redesigned to satisfy the new requirements.

Lemma 6. *All activities related to evidence transfer can be audited.*

Analysis. The clients that belong to the Political and Legal Commission can access the whole ledger to supervise the compliance of all recorded transactions.

6. Experiment

This section presents the implementation and evaluation findings of the proposed scheme.

6.1. Implementation of the Scheme. We implement the proposed scheme on Hyperledger Fabric, which is an open-source permissioned blockchain platform that supports the *Go* language for writing chaincode. We use Hyperledger Caliper to conduct performance tests on the proposed blockchain network. Table 5 presents the experimental platform information.

6.2. Performance Evaluation and Analysis. We deployed our prototype with 4 organizations. Each organization is composed of 2 peers and 1 client. There is only one channel in the blockchain network. In the following, we evaluate the throughput and latency of the proposed system. Then, we analyse the impacts of the block capacity (*BC*) and the block generation interval time (*BGT*) on the performance of the network (including the storage and communication overheads). Here, the block capacity is defined as the maximum number of transactions that can be included in a block. Table 6 presents the parameter settings of *BC* and *BGT* in the experiment.

Figure 6 shows the throughput of each transaction as a function of the send rate. According to Figure 6(a), for all

Table 5: Experimental platform information.

Operating system	Ubuntu 18.04.3
CPU	2.6 GHz AMD EPYC Rome
Memory	8 GB
Blockchain platform	Hyperledger Fabric 1.4.4

Table 6: Parameter settings of *BC* and *BGT*.

BGT (s)	1	2	3	4	5
BC	20	40	60	80	100

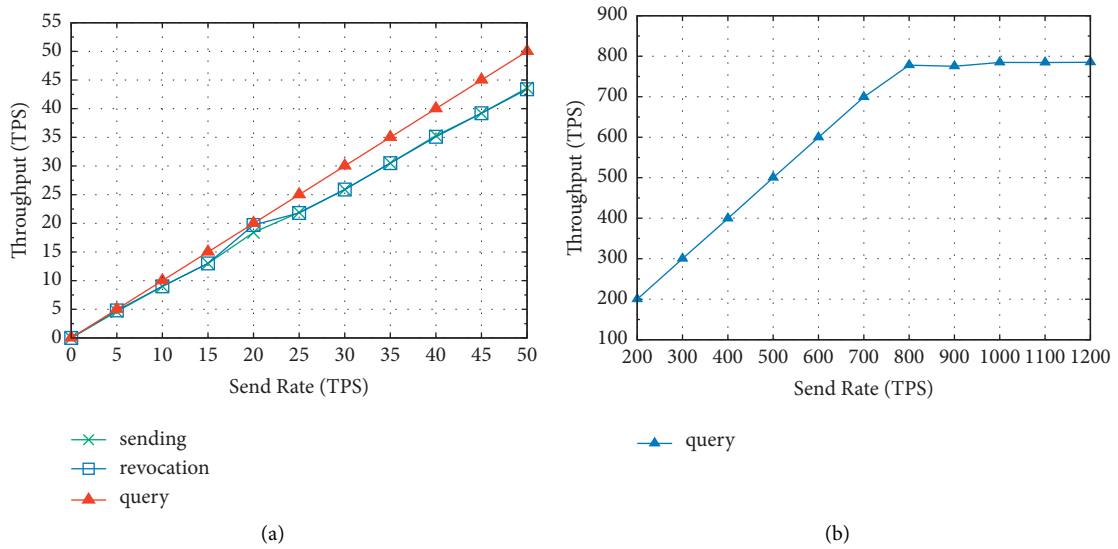

Figure 6: Throughput of each transaction with BC = 20 and BGT = 2 s. (a) Throughput of each transaction with send rate <50. (b) Throughput of query transaction with send rate >50.

three types of transactions, the throughput increases with the send rate. Under the same send rate, the query transaction has the maximum throughput. According to Figure 6(b), if the send rate exceeds 800 TPS, the throughput of the query transaction will be stable between 790 TPS and 800 TPS.

Figure 7 presents the latency of each transaction with different send rates. From Figures 7(a) and 7(b), we can see that the query transaction has almost no latency until the send rate exceeds 700 TPS. For the sending and revocation transactions, the latency decreases with increasing send rate, and the latency reaches its minimum value when the send rate is 35 TPS. When the send rate exceeds 35 TPS, the latency of the revocation transaction will increase with increasing send rate, whereas that of the sending transaction will not decrease anymore.

Figure 8 presents the storage and communication overheads of various entities in the proposed network for the sending transaction. According to Figure 8(a), the storage communication of CA and the client node is almost 0 MB regardless of the send rate. For the orderer node, state machine, and peer node, the storage overhead is proportional to the send rate. Furthermore, the peer node requires the most storage space.

From Figure 8(b), it is obvious that the higher the send rate, the larger the communication overheads of the peer node and orderer, while for the client and CA, the communication consumption is almost 0 MB. The communication overhead of the state machine is between that of the CA and the peer node, which increases slowly as the send rate increases.

In Figure 9, we present the storage and communication overheads of different entities in the proposed network for the revocation transaction. From Figure 9(a), we can see that the size relationship of the storage overhead of different kinds of entities in the revocation transaction is the same as that in the sending transaction; that is, the CA and client consume almost 0 MB memory, and the peer node consumes the maximum storage space. By comparison, at the same send rate, the revocation transaction consumes more memory than the sending transaction.

Figure 9(b) shows the communication overhead of the revocation transaction. We can see that the CA traffic is almost 0 MB. The other entities' communication overheads are proportional to the send rate, while the orderer and peer node have more traffic than the client and state machine. Compared with the sending transaction, the entities have slightly higher communication overheads in the revocation transaction.

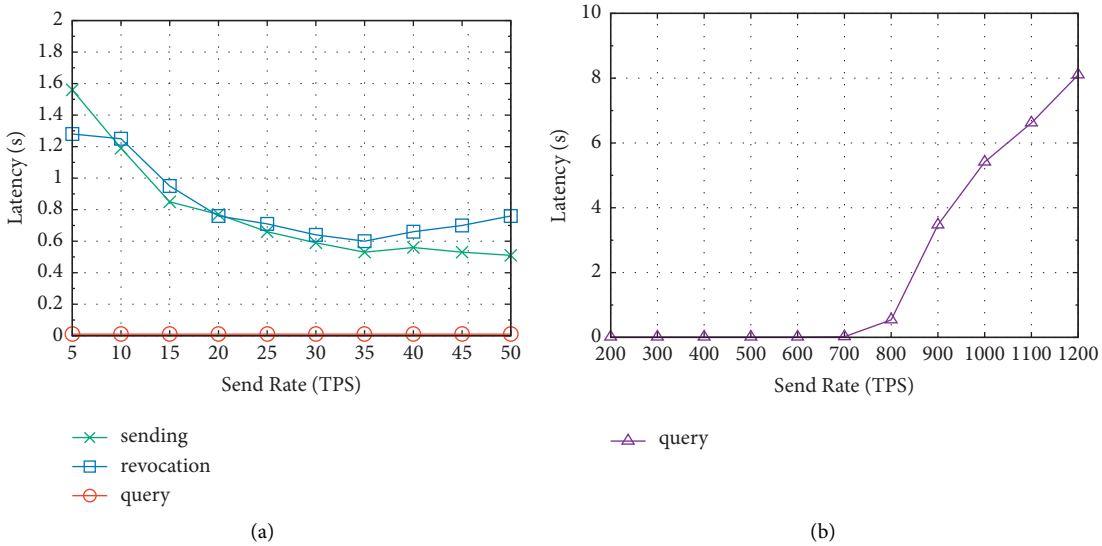

FIGURE 7: Latency of each transaction with $BC = 20$ and $BGT = 2$ s. (a) Latency of each transaction with send rate <50. (b) Latency of sending transaction with send rate >50.

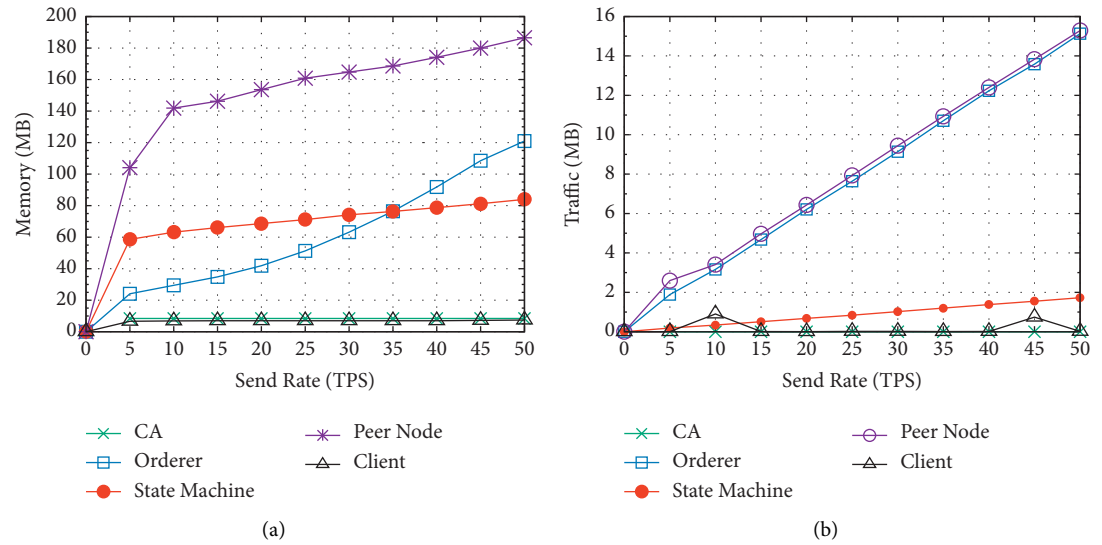

FIGURE 8: Storage and communication overhead of the sending transaction with $BC = 20$ and $BGT = 2$ s. (a) Storage overhead of sending transaction. (b) Communication overhead of sending transaction.

In Figure 10, we present the storage and communication overheads of various entities in the proposed network for the query transaction. The graph in Figure 10(a) is very similar to that in Figure 9(a), except for the trends of the storage space at the peer node, orderer, and state machine. For the peer node and orderer, the memory consumption is stable as the send rate varies. The state machine requires less memory as the send rate increases when the send rate exceeds 25 TPS. The reason is that the query transaction controls the access to the ledger, so the consumed memory is not influenced by the send rate.

From Figure 10(b), we can see that the CA also generates almost 0 KB traffic, and the peer node has the highest communication overhead. The communication overheads of the peer node, orderer, and state machine are proportional to the send rate with various ratios. The communication overheads of the entities in the query transaction are much lower than those in the sending and revocation transactions. When the send rate is 50 TPS, the peer node only generates less than 1 MB (800 KB) in traffic.

Overall, for each kind of transaction, the storage and communication overheads are acceptable for evidence transfer within China's judicial system.

In the following, we will analyse the influence of the *BC* and *BGT* on the performance of our system.

Figure 11 shows the impact of the block capacity on the throughput of the network. Comparing Figures 11(a) and 11(b), the throughput of the sending transaction is highly

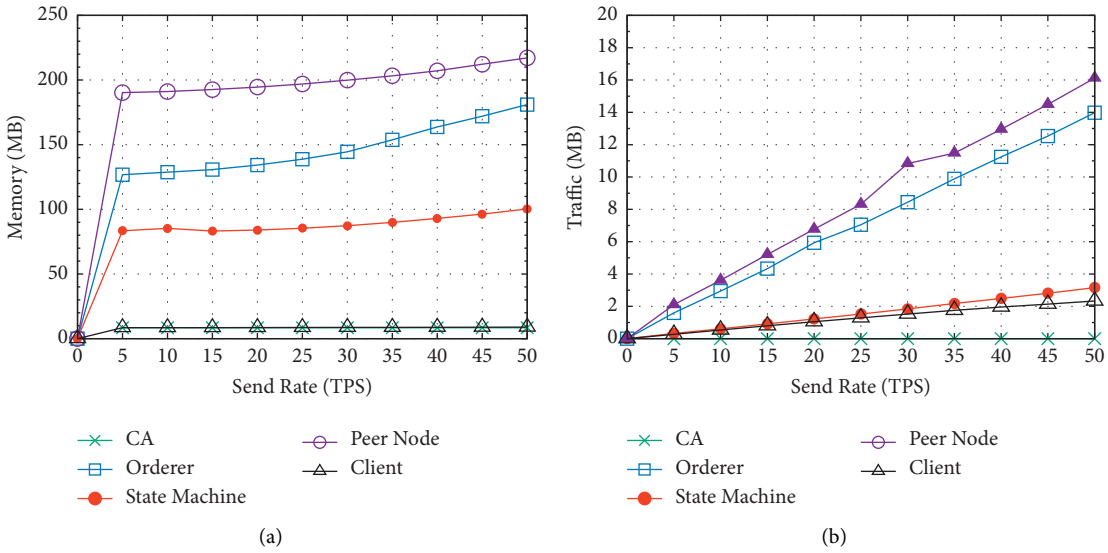

FIGURE 9: Storage and communication overhead of the revocation transaction with $BC = 20$ and $BGT = 2$ s. (a) Storage overhead of revocation transaction. (b) Communication overhead of revocation transaction.

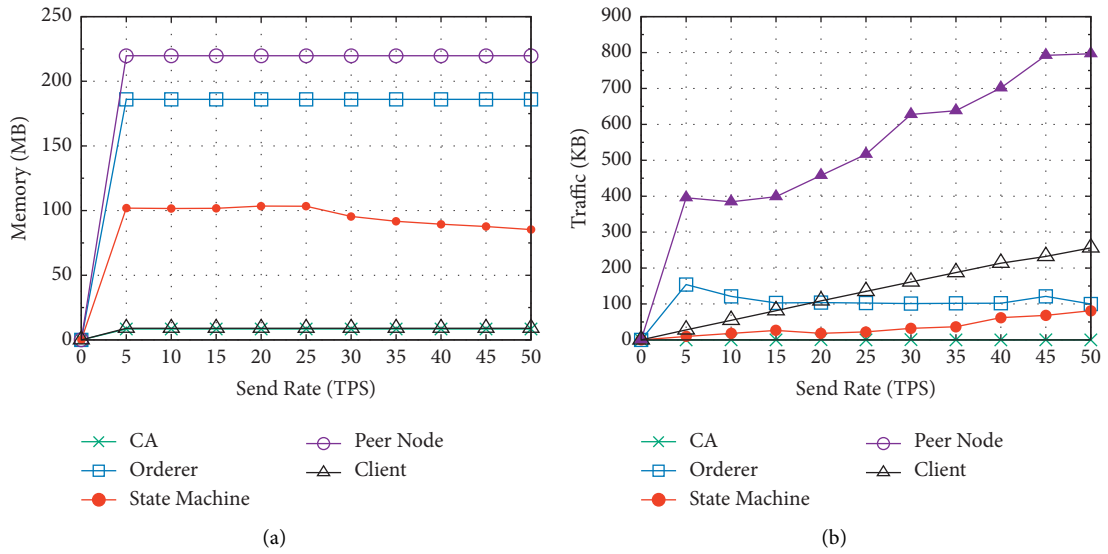

FIGURE 10: Storage and communication overhead of the query transaction with $BC = 20$ and $BGT = 2$ s. (a) Storage overhead of query transaction. (b) Communication overhead of query transaction.

similar to that of the revocation transaction. When $BGT = 1$ s and 2 s, the throughput of the network remains stable regardless of the value of BC. If $BGT > 2$ *s*, the throughput of the network will increase from approximately 40 TPS to approximately 50 TPS. Finally, the throughput difference will be very small in the cases in which $BC > 80$.

For the query transaction, we know from Figure 6 that the value of send rate at which the highest throughput is attained is 800 TPS. When the send rate exceeds 800 TPS, the throughput of the query transaction will maintain between 750 TPS and 800 TPS. According to Figure 11(c), when the send rate is 50 TPS (less than 800 TPS), the throughput of the network will remain at 50 TPS regardless of the values of BC and BGT. In contrast, Figure 11(d) shows when the send rate is 1000 TPS (greater than 800 TPS), the throughput will fluctuate between 700 TPS and 800 TPS regardless of the value of BC and BGT. So, we can see that the throughput is mainly influenced by the send rate of the transaction rather than the BC and BGT.

The influence of BC on the storage and communication overhead is shown in Figure 12. We can see from Figure 12(a) that, given the value of BC, there is little difference in the memory cost under different BGT values, which is generally within 200 MB. In addition, with the increase of BC, the overall storage consumption will increase slightly. That is because the throughput is almost unrelated with the value of BC and BGT, so the ledger's memory consumption is also not influenced by BC and BGT. The

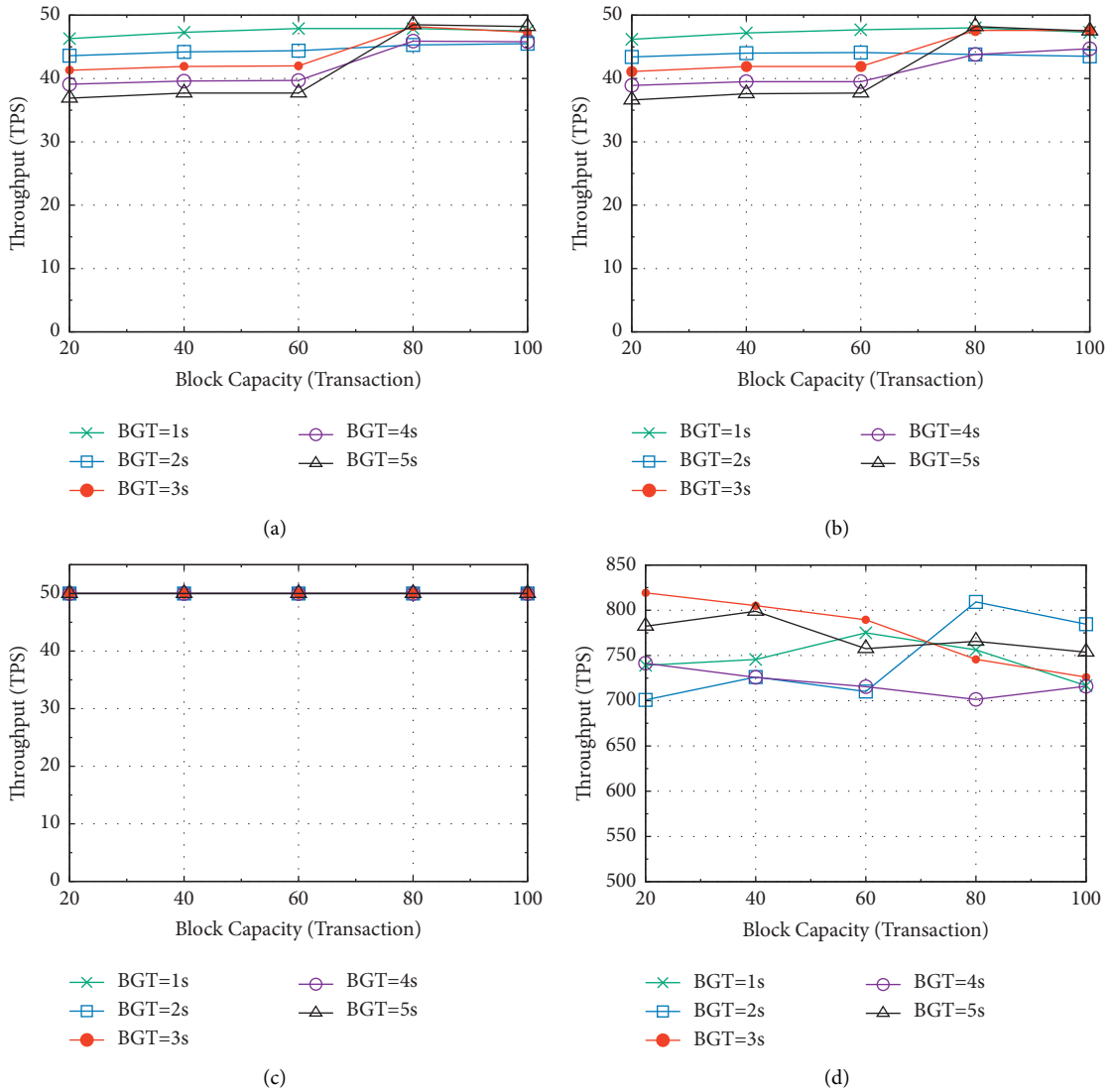

Figure 11: Influence of *BC* on the throughput of each transaction. (a) Throughput of sending transaction with send rate = 50 TPS. (b) Throughput of revocation transaction with send rate = 50 TPS. (c) Throughput of query transaction with send rate = 50 TPS. (d) Throughput of query transaction with send rate = 1000 TPS.

component that has the greatest impact on storage overhead is the sorting module of the leader node, because the bigger the block capacity, the more the transactions and corresponding endorsement files will be stored before they are packaged and generated as a block, while this memory consumption is only a small fraction of the total blockchain network storage space.

Figure 12(b) shows the relationship of the value of *BC* and *BGT* with the communication overhead of the system. It is obvious that when send rate equals 50 TPS, the communication overhead of sending transaction is around 200 MB regardless of the value of *BC* and *BGT*. This is because the communication activities of the network mainly include the communication between the client and endorsement node in endorsement process, and step 5, 7, and 9 of the consensus process. Given the send rate, the communication between the client and endorsement node will be maintained at a stable level. The same is true for the communication overhead of the consensus process.

The influence of *BGT* on the throughput of each transaction is presented in Figure 13. Similar to the scenario in Figure 11, the value of *BGT* has similar influences on the throughputs of the sending transaction and the revocation transaction. When $BC > 80$, the throughputs of the sending transaction and the revocation transaction are not substantially affected by the value of *BGT*, while if $BC < 80$, the throughput is inversely proportional to *BGT*.

Figures 13(c) and 13(d) show the throughput of the query transaction. The same as Figure 11, we take the send rate of the system into account when analysing the influence of *BC* and *BGT* on the throughput. In general, the throughput of the query transaction is basically not influenced by the value of *BC* or *BGT*. If the send rate is 50 TPS (less than the threshold 800 TPS), the throughput will

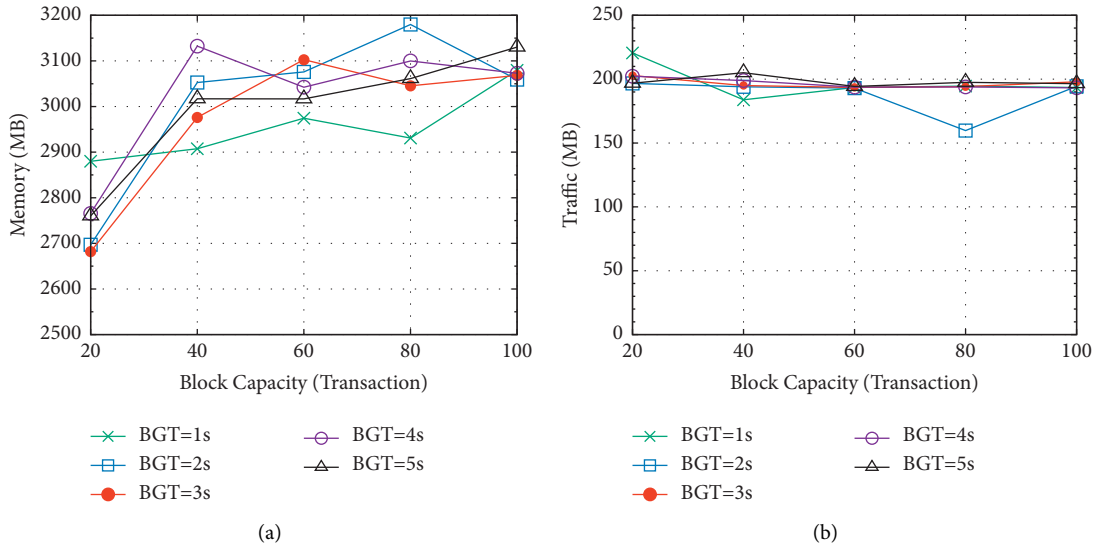

FIGURE 12: Influence of *BC* on the storage and communication overhead with send rate = 50 TPS. (a) Storage overhead of sending transaction. (b) Communication overhead of sending transaction.

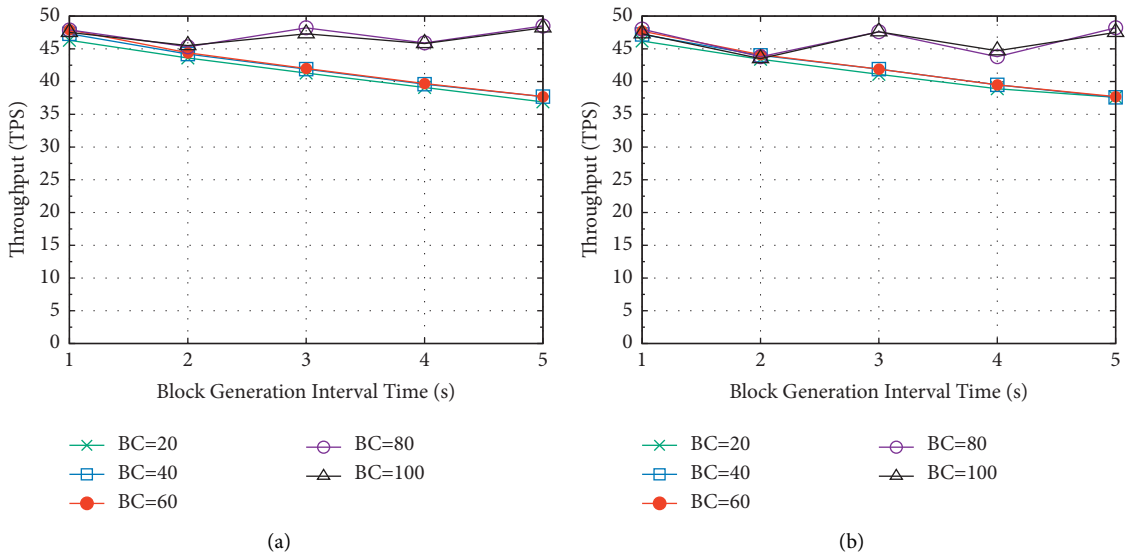

FIGURE 13: Continued.

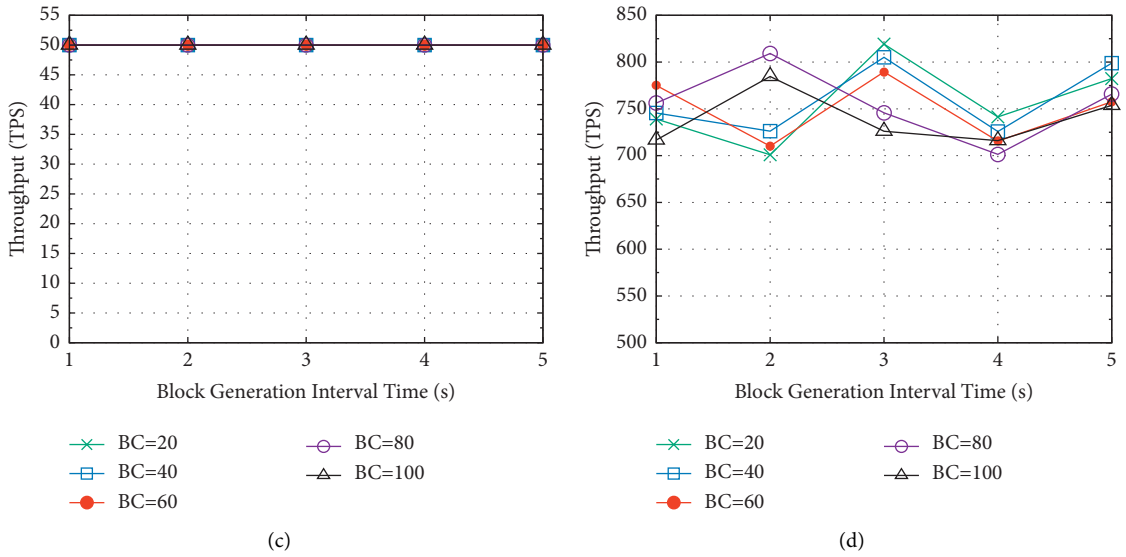

Figure 13: Influence of *BGT* on the throughput of each transaction. (a) Throughput of sending transaction with send rate = 50 TPS. (b) Throughput of revocation transaction with send rate = 50 TPS. (c) Throughput of query transaction with send rate = 50 TPS. (d) Throughput of query transaction with send rate = 1000 TPS.

remain stable at a value equal to the send rate. If the send rate exceeds the threshold (equal to 1000 TPS), the throughput of the query transaction will fluctuate over a small interval (between 700 TPS and 800 TPS). That is consistent with Figures 6 and 11.

In conclusion, we can see that both the CA and the client require minimal storage and communication resources, and the orderer consumes the most memory and traffic in the network. Given a certain send rate, the best performance can be realized with various values of *BC* and *BGT*. Thus, we can select suitable devices and parameters for the deployment of the proposed system according to the volume of evidence transfer in different regions' judicial systems.

6.3. Discussion. As presented in Table 5, our prototype is deployed with Hyperledger Fabric 1.4.4. The latest version of Fabric is v2.0, which shows a substantial performance improvement compared with the previous version. According to reports of systems that have been implemented based on v2.0, the performance improvement of v2.0 over the previous versions is approximately 3 times, although the performance differences depend on the operations. Thus, our proposed scheme can achieve greater performance with the development of the blockchain platform.

Within the judicial system, other functions need to be addressed except for evidence transfer. In this case, we can expand the proposed network to complete more functions via multichannel and interchain technologies, among other technologies.

Because of the consensus algorithm that we adopted, the proposed scheme may be vulnerable to Byzantine attacks. Moreover, the maximum number of fault tolerant nodes supported is $(N-1)/2$, where N is the total number of nodes in the blockchain network. To address this problem, the operational network within the judicial system is classified and isolated from the Internet, so the probability of suffering from a Byzantine attack is low. In addition, we can adopt the PBFT consensus mechanism to easily defend against Byzantine attacks.

Furthermore, our proposed scheme may suffer from denial of service attacks. In our scheme, we can introduce an anomaly detection mechanism or limit the transaction proposal submission rate to reduce the rate of successful attacks.

7. Conclusions

In this paper, we propose a blockchain network for recording evidence transfer events among different departments of the judicial system to ensure the non-tampering and consistency of the evidence transfer record. We design the form of the transaction and the content of a block to properly store the necessary information. Moreover, we construct three smart contracts for evidence sending, invocation, and query. The Raft consensus mechanism is adopted to finish the consensus process of a transaction. The security analysis proves that our scheme can achieve the design goal. Moreover, we have implemented the system on a test chain and conducted a set of experiments to evaluate the performance of the system. The results show that the performance of the proposed scheme can satisfy the requirements of China's judicial system. According to the performance analysis results, we can choose a suitable device and set suitable parameters to deploy the proposed system in the judicial system.

In the future, we will deploy the proposed scheme in part of the judicial system in one province of China. Part of the evidence transfer record management of criminal cases will be completed by using the deployed system. On this basis, we

will seek its wider use. Specifically speaking, we will consider using blockchain technology to complete more business within the judicial system.

Data Availability

The data used to support the findings of this study are unavailable.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this study.

Acknowledgments

This work was supported by a grant from the National Key R&D Program of China (no. 2017YFB0801805), the Key Research and Development Plan of Xinjiang Production and Construction Corps (no. 2019AB001), the National Natural Science Foundation of China (nos. 61872283, 62032025, 62102167, 61962022, 61672415, and 61872088), and the Fundamental Research Funds for the Central Universities (nos. JBX181504 and 21618332). This work was also supported by the 111 project (Grant B16037).

References

[1] S. P. C. of, "The peoples republic of china, chinese courts and internet judiciary," 2019, http://wlf.court.gov.cn/upload/file/2019/12/03/11/40/20191203114024_87277.pdf.

[2] J. Richter, N. Kuntze, and C. Rudolph, "Security digital evidence," in *Proceedings of the 2010 Fifth IEEE International Workshop on Systematic Approaches to Digital Forensic Engineering*, Article ID 119C130, Oakland, CA, USA, May 2010.

[3] J. Cosic and M. Baca, "A framework to (im) prove chain of custody in digital investigation process," in *Proceedings of the 21st Central European Conference on Information and Intelligent Systems*, Article ID 435C438, Varazdin, Croatia, September2010.

[4] J. Cosic and M. Baca, "(im)proving chain of custody and digital evidence integrity with time stamp," in *Proceedings of the 33rd International Convention MiPRO*, Article ID 1226C1230, Opatija, Croatia, May 2010.

[5] Y. Prayudi, A. Ashari, and T. K. Priyambodo, "Digital evidence cabinets: a proposed framework for handling digital chain of custody," *International Journal of Computer Application*, vol. 107, no. 9, Article ID 975C8887, 2014.

[6] Y. Prayudi, A. Ashari, and T. K. Priyambodo, "The framework to support the digital evidence handling," *Journal of Cases on Information Technology*, vol. 22, no. 3, pp. 51–71, 2020.

[7] J. Fisher and M. H. Sanchez, "Authentication and verification of digital data utilizing blockchain technology," Patent US20160283920A1, 2016.

[8] W. Yan, J. Shen, Z. Cao, and X. Dong, "Blockchain based digital evidence chain of custody," in *Proceedings of the 2020 the 2nd International Conference on Blockchain Technology, ICBCT20*, pp. 19–23, Association for Computing Machinery, New York, NY, USA, March 2020.

[9] A. H. Lone and R. N. Mir, "Forensic-chain: blockchain based digital forensics chain of custody with poc in hyperledger composer," *Digital Investigation*, vol. 28, pp. 44–55, 2019.

[10] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008, https://bitcoin.org/bitcoin.pdf.

[11] H. D. Z. Zheng, S. Xie, and H. Wang, "Blockchain challenges and opportunities: a survey," *International Journal of Web and Grid Services*, vol. 14, no. 4, Article ID 352C375, 2018.

[12] N. R. D. J. Yaga, P. M. Mell, and K. Scarfone, "Blockchain technology overview," Tech. Rep. 8202, National Institute Standards Technol, Gaithersburg, MD, USA, 2018.

[13] N. Teslya and I. Ryabchikov, "Blockchain platforms overview for industrial IoT purposes," in *Proceedings of the 22st Conference of Open Innovations Association FRUCT, FRUCT Oy*, Article ID 250C256, Helsinki, Finaland, April 2018.

[14] Z. Lu, Q. Wang, G. Qu, and Z. Liu, "Bars: a blockchain-based anonymous reputation system for trust management in VANETs," in *Proceedings of theseveenth IEEE International Conference on Trust, Security and Privacy in Computing and Communications/12th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE)*, Article ID 98C103, New York, NY, USA, August2018.

[15] Y. Xu, J. Ren, G. Wang, C. Zhang, J. Yang, and Y. Zhang, "A blockchain-based nonrepudiation network computing service scheme for industrial IoT," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3632–3641, 2019.

[16] N. Kabra, P. Bhattacharya, S. Tanwar, and S. Tyagi, "Mudrachain: blockchain-based framework for automated cheque clearance in financial institutions," *Future Generation Computer Systems*, vol. 102, pp. 574–587, 2020.

[17] D. Li, Y. Hu, and M. Lan, "IoT device location information storage system based on blockchain," *Future Generation Computer Systems*, vol. 109, pp. 95–102, 2020.

[18] G. He, W. Su, S. Gao, J. Yue, and Td-Root, "TD-Root: a trustworthy decentralized DNS root management architecture based on permissioned blockchain," *Future Generation Computer Systems*, vol. 102, pp. 912–924, 2020.

[19] C. Lin, D. He, X. Huang, X. Xie, and K.-K. R. Choo, "Blockchain-based system for secure outsourcing of bilinear pairings," *Information Sciences*, vol. 527, pp. 590–601, 2020.

[20] G. Wood, *Ethereum: A Secure Decentralised Generalised Transaction Ledger*, 2014, http://paper.gavwood.com.

[21] Hyperledger-Fabricdocs Documentation, 2021, https://hyperledger-fabric.readthedocs.io/_/downloads/en/release-1.4/pdf/.

[22] N. Szaho, "Smart contracts: building blocks for digital markets," *Extrory: Journal Transhumanist Thought*, vol. 16, pp. 1–11, 1996.

[23] M. Castro and B. Liskov, "Practical byzantine fault tolerance and proactive recovery," *ACM Transactions on Computer Systems*, vol. 20, no. 4, pp. 398–461, 2002.

[24] D. Ongaro and J. Ousterhout, "In search of an understandable consensus algorithm," in *Proceedings of the 2014 USENIX Conference on USENIX Annual Technical Conference*, pp. 305–320, USENIX ATC14, USENIX Association, Philadelphia, PA, USA, June 2014.