Information Sciences 491 (2019) 151–165

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Block-DEF: A secure digital evidence framework using blockchain

Zhihong Tian[a], Mohan Li[a], Meikang Qiu[b,*], Yanbin Sun[a,*], Shen Su[a]

[a] Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou 510006, China
[b] Department of Electrical Engineering, Columbia University, New York City, NY 10027, USA

ARTICLE INFO

Article history:
Received 30 September 2018
Revised 30 March 2019
Accepted 1 April 2019
Available online 3 April 2019

Keywords:
Digital evidence
Blockchain
Scalability
Privacy
Traceability

ABSTRACT

A secure digital evidence system should ensure that evidence cannot be tampered with and that private information cannot be leaked. Blockchain, a distributed tamper-resistant and privacy-preserving ledger, provides a promising solution for decentralized secure digital evidence systems. However, due to the huge number of digital evidences and the contradiction between the traceability and the privacy of evidence, blockchain faces big data and privacy challenges. To solve the above issues, we propose a secure digital evidence framework using blockchain (Block-DEF) with a loose coupling structure in which the evidence and the evidence information are maintained separately. Only the evidence information is stored in the blockchain, and the evidence is stored on a trusted storage platform. To avoid blockchain bloat, a lightweight blockchain combining a mixed block structure with an optimized name-based practical byzantine fault tolerance consensus mechanism is proposed. To support the traceability and the privacy of evidence, the multi-signature technique is adopted for evidence submission and retrieval. The analytical and experimental results show that Block-DEF is a scalable framework, it guarantees the integrity and validity of evidence, and balances privacy and traceability well.

1. Introduction

The application of the Internet has changed from host-centric to content-centric. Publishing and retrieving contents is becoming the main requirement of Internet users [45]. The content may be a file or a piece of file that is transported across the Internet, such as web pages, images, audios or videos. Accordingly, content security becomes an important part of cyber security, and it mainly focuses on three security properties: privacy, integrity and non-repudiation. Unfortunately, not all contents are properly protected. Many content objects (files) are tampered with due to network attacks or other reasons. Such file tampering has many negative effects. For example, web page tampering can be used to craft phishing attacks or broadcast illegal information. Executable files can be injected with malicious codes by the attacker to monitor user behavior or illegally access private data. For technical, economic and legal reasons, it is often necessary to investigate the corresponding digital evidence for file tampering.

The process of digging and collecting digital evidence has attracted considerable attention [32,34]. When digital evidence is obtained, it is always transmitted directly to the third party management or stored in local devices. Evidence storage,

[*] Corresponding authors.
E-mail addresses: tianzhihong@gzhu.edu.cn (Z. Tian), limohan@gzhu.edu.cn (M. Li), qiumeikang@yahoo.com (M. Qiu), sunyanbin@gzhu.edu.cn (Y. Sun), sushen@gzhu.edu.cn (S. Su).

https://doi.org/10.1016/j.ins.2019.04.011
0020-0255/© 2019 Elsevier Inc. All rights reserved.

management and transmission are all based on these systems. However, the security of digital evidence systems has been ignored. Digital evidence system may harbor vulnerabilities. These vulnerabilities can be exploited by attackers, resulting in (1) evidence tampering, in which the evidence may be maliciously modified, removed or untraceable, and (2) privacy leaks. Private information, such as evidence content, evidence providers, and other information, may be leaked. How to maintain the security of digital evidence is worth studying.

Existing secure digital evidence systems such as those described in [1,9,27] mostly adopt centralized designs. They provide tamper-resistant mechanisms on a single device or a centralized system via secure software, secure hardware, physical separation or hybrid strategies. The centralized design faces the following challenges: (1) the single point of failure, which may invalidate the system; (2) the scalability issue, which arises if the amount of evidence is too large to store.

The blockchain, which is widely used in cyptocurrency systems [26,42], is a promising technology that can be used to overcome the foregoing challenges due to its distributed, tamper-resistant and private nature. However, blockchain also faces a scalability issue: blockchain bloat. In a blockchain, each node stores all blocks. As the length of the blockchain increases, the storage requirement for each node also increases. Thus, a lightweight blockchain is demanded for a secure digital evidence system. At the same time, to guarantee the availability and legitimacy of the evidence, the evidence should be traceable. How to track the evidence while ensuring privacy is also another of problems associated with the use of a blockchain.

In this paper, we choose file tampering as a case study and propose a lightweight, scalable secure digital evidence framework using blockchain (Block-DEF). The main contributions of Block-DEF are as follows. First, Block-DEF adopts a loose coupling design. Only the evidence information is stored in the blockchain, and the evidence is stored on a trusted storage platform. Thus, in Block-DEF, the storage pressure is significantly reduced. Second, two multi-signature schemes for evidence submission and retrieval are proposed, such that the traceability and the privacy of evidence are balanced. Third, to avoid blockchain bloat, a lightweight blockchain with a mixed block structure and an optimized name-based practical byzantine fault tolerance (PBFT) consensus mechanism is proposed. Each node only needs to store all the block headers and a part of the block bodies. The results of analyses and experiments show that Block-DEF effectively supports scalability, integrity, validity, privacy and traceability.

The remainders of this paper are organized as follows. Section 2 discusses related work. Section 3 presents the architecture of Block-DEF and Section 4 details the design of Block-DEF. In Section 5, we analyze and evaluate the performance of Block-DEF. We then conclude the paper in Section 6.

2. Related work

A blockchain can be viewed as a public ledger in which each node in the blockchain network stores the same ledger. The ledger consists of a sequence of blocks in which all committed transactions are stored. With the exception of the genesis block, each block has a hash pointer to the previous block. Block miners can generate new blocks and append the block to the end of the chain via various consensus mechanisms, such as proof of work (PoW) [26], proof of stake (PoS) [39], PBFT [7], and others. Any modification to a block will break the hash pointer to the block. Thus, transactions cannot be maliciously modified once they are stored in the blockchain. In addition, all users participating in the blockchain use multiple generated addresses rather than their real identities via cryptographic hash and digital signatures. The privacy of the users can thus be guaranteed to a certain degree.

The blockchain was first used as the core technology of cyptocurrencies, such as bitcoin [26], the first decentralized digital cyptocurrency. Due to its immutability and privacy, bitcoin is widely deployed. Following bitcoin, a great number of alternative currencies were proposed, e.g., Etherenum, Litcoin, Ripple, Zcash, and others.

With the development of blockchain technology, the blockchain has attracted considerable attention. It can be divided into three categories: public blockchain, consortium blockchain and private blockchain. In addition to cryptocurrency, researchers have begun to explore approaches to solving security issues using blockchain. To date, the blockchain has been widely used in various fields, including blockchain-based DNS [23], security service for cloud [21,46,48], IoT security [18], and other fields. In the field of digital forensics, the blockchain is also a promising approach for evidence verification and management, and it is widely studied.

Digital forensics is an important part of cyberspace security. With the development of the Internet, wireless sensor networks, and IoTs, cloud computing and data-centric application scenarios such as data-driven routing [38], IPTV services [44], the smart grid [47] and the edge computing [37] are widely deployed, including in the financial industry [15]. However, during data transmission and management, data may be stolen, replaced and tampered with due to various attacks [10,22,36]. To overcome these security issues, security techniques, such as fully homomorphic encryption [14,20], differential privacy [47], key management schemes [11–13,31], and attribute-based semantic access controls [30] have been proposed. These techniques provide effective solutions that prevent potential security issues. For security issues that have already occurred, investigation, collection and management of the evidence of attacks are needed. Thus, the security of evidence becomes an important requirement.

Previous evidence management focuses on the secure storage of evidence, and tamper-resistant storage mechanisms constitute its main focus. The evidence here is mostly the audit log. It is maintained by both the end-user and the evidence server. Schneier and Kelsey [33] proposed a secure audit logging storage protocol that combines hash chains and evolving cryptographic systems. Based on Schneier and Kelsey's protocol, Chong et al. [9] proposed a hardware-based tamper-resistant system for end-users in which the protocol is embedded in tamper-resistant hardware. However, the server is assumed to be trusted. To support secure servers, SBBox [1], a tamper-resistant system for forensic data collection and storage, was proposed. The security is achieved by a black-box-like hardware that is connected via a PCIe bus. Nieto et al. [27] proposed a digital witness definition that safeguards digital evidence in personal devices through collaboration. Evidence obtained by personal devices is finally sent to an official collection point. The evidence management systems based on the above schemes are all centralized, a characteristic that may cause a single point of failure and scalability issues.

In recent years, some blockchain-based secure digital evidence schemes have been proposed. Zyskind et al. [49] proposed a personal data management platform based on blockchain that is similar to the evidence management systems. Both the data pointer and the access control list are stored in the blockchain. Thus, only permitted services can access the corresponding data. This platform guarantees the privacy of personal data without the use of a trusted third-party, but the data cannot be tracked. Bonomi et al. [5] proposed a blockchain-based chain of custody (B-CoC). B-CoC uses blockchain to track the process of evidence investigation and ensure that evidence is not altered during the investigation. It guarantees the integrity and traceability of evidence. However, because at each time point the evidence is assumed to have a single owner, the application scenario may be limited. Nieto et al. [3] analyzed the privacy requirements of digital witnessing, which transmits digital evidence from a personal device. They proposed a blockchain-based solution that balances the property of the digital witness and the privacy requirement by using HawK [19]; the privacy of transaction is the main goal of this work. Blockchain can also be used in forensics applications of vehicular networks. Cebe et al. [8] proposed a blockchain-based vehicular forensics system (B4F). B4F uses VPKI and fragmented ledger to address storage overhead and membership management. Evidence is stored or distributed in the blockchain; this system is suitable for vehicular networking scenarios, but its applicability to other scenarios may be limited. Unlike Block-DEF, these schemes only partially satisfy the need for privacy, traceability and scalability. Block-DEF can satisfy all of these properties.

3. Architecture of Block-DEF

In this section, we first list the requirements that Block-DEF should satisfy, and then present the architecture of Block-DEF.

3.1. Requirements of Block-DEF

Block-DEF is built by one or multiple authority organizations and provides services for the public. It supports evidence collection, storage, verification and retrieval. The evidence collection function collects evidences from the evidence provider, the evidence storage function stores the collected evidences, the evidence verification function verifies the evidence and records the verified evidence in the blockchain, and the evidence retrieval function provides evidences or evidence information to authorized requesters, e.g., police officers, lawyers, judges, etc. However, the blockchain is not a “one size fits all” solution. Although the blockchain is tamper-resistant and preserves the privacy of the evidence, Block-DEF should still consider the following requirements.

1. Scalability. Due to the large number of tampered files and the occurrence of blockchain bloat [41], it is difficult to store all evidences in a blockchain. The storage capability of a node in the blockchain network may face challenges.
2. Integrity. The evidence should not be modified after it is submitted to Block-DEF, i.e., the system should ensure that no one can modify the evidence without being detected.
3. Validity. The evidence should be consistent with its corresponding file, i.e., the system should ensure that the file has indeed been tampered with as the evidence described.
4. Privacy. Private information about the evidence (e.g., the evidence content, the evidence provider and requester, etc.) cannot be obtained by unauthorized requesters.
5. Traceability. The source of evidence (e.g., the evidence provider) should be tracked by Block-DEF. At the same time, the chain of evidence, which lists the evidences of a file in the order of evidence submission, should be recorded, and it reveals the relationship between these evidences.

3.2. Three-layer architecture

As shown in Fig. 1, the architecture of Block-DEF is logically divided into three layers: a service layer, a blockchain layer, and a network layer. Each layer provides support for the upper layer to achieve the requirements of Block-DEF.

The service layer focuses on how to support specific evidence-related services by using the blockchain. In this layer, multiple signature schemes are employed to ensure integrity, verifiability, privacy, and other necessary characteristics. To satisfy the scalability requirements and simplify the design of Block-DEF, Block-DEF adopts a loose coupling structure with two modules, a storage module and a blockchain module, to separate evidence storage and management. The storage module is used to support evidence storage, and the blockchain module participates in evidence collection, verification, and retrieval.

The blockchain layer is responsible for constructing a blockchain for each node in the underlying network. Considering the sensitivity of the digital evidence, Block-DEF can be built either within or between authority organizations. The network environment is relatively safe, and the block mining does not rely on an economic incentive. Thus, Block-DEF can adopt a

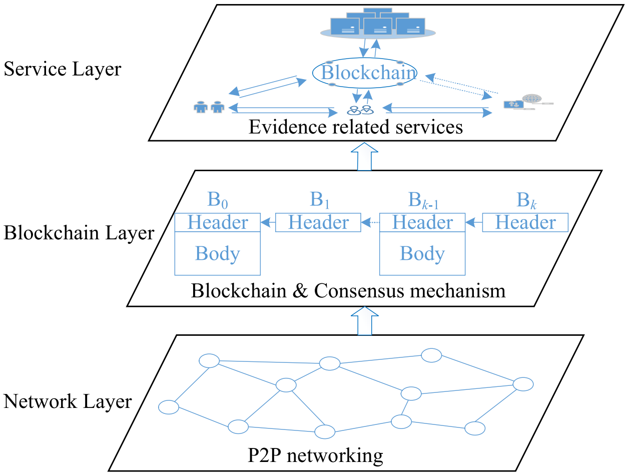

Fig. 1. Hierarchical architecture of Block-DEF with three layers.

private or a consortium blockchain. The main challenge in this layer is the blockchain bloat. To overcome this challenges, we propose a mixed blockchain rather than a full blockchain, as well as a corresponding name-based consensus mechanism. The block bodies are distributed to different nodes.

The network layer is used to construct the network topology and support communication for above two layers. Most blockchain schemes use the peer-to-peer (P2P) network as the blockchain network. Block-DEF also adopts the P2P network to organize nodes and provides P2P routing and encryption communication to support the specific tasks of the two layers, for example, broadcast transactions, encrypted transmission of evidence information, maintain the consensus of the blockchain. Existing P2P network schemes can be used directly or improved for blockchain, e.g., etherenum uses Kademlia [24] to construct its network. Even the redesigned P2P networks of some blockchain schemes follow the basic principles of P2P networks. For Block-DEF, an existing P2P network scheme can also be used. According to the blockchain structure and the name-based consensus mechanism used in the blockchain layer, the network topology of Block-DEF can be divided into multiple groups that form a hierarchical structure according to the node names. Correspondingly, a hierarchical P2P network scheme can be used for Block-DEF. For the inter-group scheme, existing schemes such as Chord [35] or Kademlia [24] can be adopted. For the intra-group scheme, flooding may be suitable. Since Block-DEF devotes more attention to the above two layers, the P2P network scheme of the network layer will not be further studied, and it will be left for further research. Thus, we simplify the blockchain network and assume that each node is known to each other, i.e., the blockchain network is a fully-connected logical topology.

4. Details of Block-DEF

Inspired by information-centric networking [40,43], in Block-DEF, both files and evidences are assigned names for addressing. Thus, in this section, a naming scheme for files and evidences is first presented, and an evidence service model for evidence submission and retrieval is then described. In the end, a blockchain model with a mixed blockchain structure and a consensus mechanism is detailed.

4.1. Naming scheme

According to existing naming schemes [16], there are two ways to classify names: the location-based approach and the structure-based approach. In the location-based approach, names are divided into location-dependent names and location-independent names. In the structure-based approach, the names are divided into hierarchical names and flat names. For Block-DEF, both the files and the evidences should be identified. Naming schemes for the file and evidence are proposed based on the above divisions.

Block-DEF adopts a location-dependent hierarchical naming scheme for the files. In Block-DEF, each file and its replicas are considered different files. Since replicas may suffer from different forms of tampering, the evidence for each replica should be recorded separately and should belong to a different evidence chain. The location-dependent property is used to distinguish these files. The hierarchical structure is inspired by the URI, and it provides readable and understandable names. The structure of the file name f_n is $/Protocol/IP\_address/Port/File\_path/File\_name$. The file name reveals two types of key information: the provider of the file (by the IP address) and the access path to the file. Both types of information are conducive to evidence traceability and validity verification.

A digital evidence is determined by its corresponding file and by the file content. The content of a file may be maliciously modified at different times. Each modification may correspond to a new evidence. All the evidences for a file sequentially

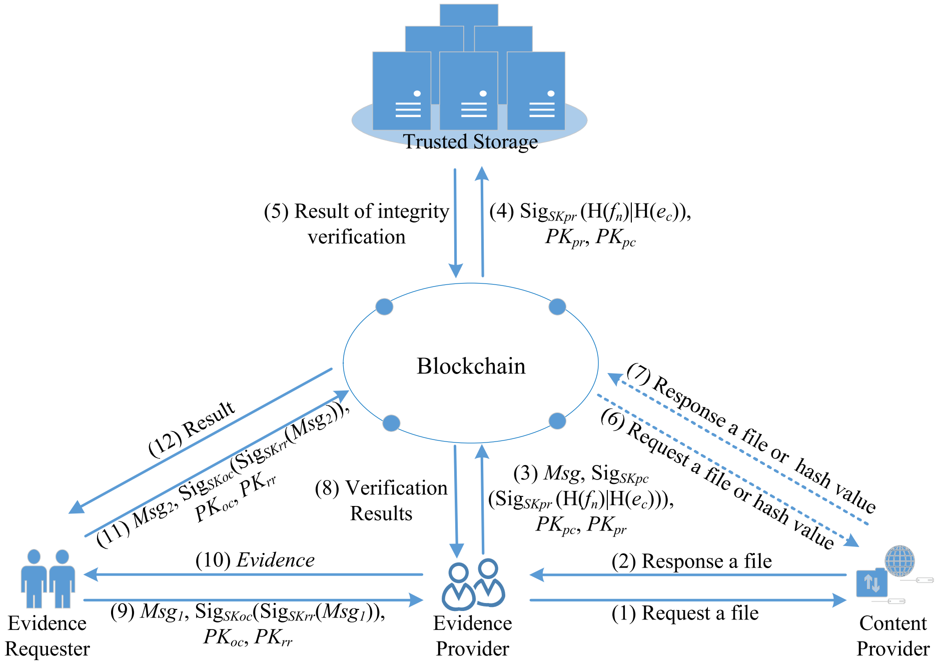
Trusted Storage
(5) Result of integrity verification
(4) $\mathrm{Sig}_{SK_{pr}}(\mathrm{H}(f_n)|\mathrm{H}(e_c))$, PK_{pr}, PK_{pc}
Blockchain
(12) Result
(11) Msg_2, $\mathrm{Sig}_{SK_{oc}}(\mathrm{Sig}_{SK_{rr}}(Msg_2))$, PK_{oc}, PK_{rr}
(8) Verification Results
(3) Msg, $\mathrm{Sig}_{SK_{pc}}(\mathrm{Sig}_{SK_{pr}}(\mathrm{H}(f_n)|\mathrm{H}(e_c)))$, PK_{pc}, PK_{pr}
(7) Response a file or hash value
(6) Request a file or hash value
(10) Evidence
(9) Msg_1, $\mathrm{Sig}_{SK_{oc}}(\mathrm{Sig}_{SK_{rr}}(Msg_1))$, PK_{oc}, PK_{rr}
(2) Response a file
(1) Request a file
Evidence Requester
Evidence Provider
Content Provider

Fig. 2. Evidence service model of Block-DEF for evidence submission and retrieval.

form an evidence chain. To distinguish different evidences for the same file, the evidence in Block-DEF is assigned a flat name in which the hash value of the file name $H(f_n)$ is combined with the hash value of the evidence content $H(e_c)$ using SHA256, and the evidence name N_E is expressed as $H(f_n)|H(e_c)$. $H(f_n)$ identifies the evidence chain, and $H(e_c)$ distinguishes evidences in a single evidence chain. The two components of the evidence name can also be used to verify the integrity of both the file name and the evidence.

4.2. Evidence service model

As shown in Fig. 2, Block-DEF consists of two modules: the blockchain module and the storage module. The storage module is under the control of Block-DEF, and it can directly adopt a trusted cloud storage system or a trusted file storage blockchain system. Both systems should avoid the malicious modification of evidence. Therefore, the cloud storage system should support immutable storage. When the evidence has been submitted, it cannot be modified but can only be updated by submitting the latest evidence. Multiple security schemes such as those mentioned in [17] can be applied to the system. The file storage blockchain can be designed as a new blockchain. This provides decentralized file storage service for Block-DEF. Both IPFS [4] and filecoin[1] provide good references for the file storage blockchain. Since the blockchain module is the core of Block-DEF, in this paper, we do not focus on the details of the storage module.

Although Block-DEF adopts a private or consortium blockchain, it provides services to the public. Block-DEF can collect and store evidences, verify the integrity and validity of evidence, and respond to the retrieval of evidence. These services can be divided into two processes: evidence submission and evidence retrieval. There are three types of participants in Block-DEF.

1. Content provider (CP). The CP provides files that it owns to the Internet, such as the website, the email sender or the file server.
2. Evidence provider (EP). The EP collects evidences of file tampering and submits the evidence to Block-DEF, such as the file consumer, the forensic investigator. A CP can also be an EP, it submits the original file to prove that it is attacked.
3. Evidence requester (ER). The ER requests digital evidences from the EP or Block-DEF and queries the evidence information to Block-DEF, such as the authorized evidence investigators or analysts.

4.2.1. Evidence submission

The process of evidence submission is divided into the following steps.

In steps 1 and 2, an EP requests a file from a CP. When the EP receives the file, it may find that the file is invalid, i.e., the file may have been tampered with. If the EP ensures that the file has been tampered with, it submits the digital evidence

[1] https://www.filecoin.io/.

(mostly the tampered file), an evidence name, a random public key of the EP and a corresponding signature to the storage module. The storage module temporarily stores the evidence, the evidence name and the public key before the integrity of the evidence is verified.

In step 3, after the evidence is stored in the storage module, the evidence information and the evidence signature are submitted to the blockchain for the verification of evidence. According to the requirements of Block-DEF, the privacy and traceability of EP should be guaranteed. That is, the EP cannot be tracked except through the Block-DEF.

To fulfill the above requirements, Block-DEF adopts a modified 2-of-2 multi-signature scheme inspired by multi-signature [28]. Each EP has two types of key pairs, the key pair (PK_{pc}, SK_{pc}), which is authenticated by the certificate authority, and the key pair (PK_{pr}, SK_{pr}), which is randomly generated by the EP itself. The two types of key pairs can be viewed as belonging to two entities in the multi-signature scheme. The first key pair identifies the EP based on the certification of the certificate authority, and the last key pair hides the relationship between the EP and the evidence. In our multi-signature scheme, a data d is signed by SK_{pr} and SK_{pc} in order and a signature $Sig_{SK_{pc}}(Sig_{SK_{pr}}(d))$ is obtained. The signature reveals a mapping relation between the two key pairs. According to the signature, Block-DEF obtains the mapping relation by storing the two public keys PK_{pr} and PK_{pc}. Thus, given any signature signed by the last key pair, the source of evidence is tracked by Block-DEF according to the first key pair.

In addition to the multi-signature scheme, the group signature scheme [6] is also a promising approach. The group member can sign messages on behalf of the group. The signature is verified by the group public key, which avoids leaking the identity of the signer. The signer is tracked by the group manager. The blockchain nodes and EPs can form a group, and the entire blockchain is viewed as a group manager. Due to the large number of EPs, the size of the group may pose a challenge.

For the specific process, the EP sends data to the blockchain including a message Msg (f_n, $h(f_n)|h(e_c)$, [$path$]), two public keys (PK_{pc}, PK_{pr}), and a signature $Sig_{SK_{pc}}(Sig_{SK_{pr}}(h(f_n)|h(e_c)))$. The Msg contains the evidence information. The $path$ in Msg is used to access the corresponding file or the hash value of the file such that the validity of the evidence can be verified in steps 6 and 7, and it is optional according to the choice of EP. The signature is obtained by the multi-signature scheme and it is used in steps 4 and 5.

In steps 4 and 5, when the data is received by a node of the blockchain, the muti-signature is first verified in the reverse order of signing. The node then communicates with the storage module and verifies the integrity of the evidence based on two criteria. First, the system determines whether the EP has submitted the evidence to the storage module correctly, i.e., whether the received public key PK_{pr} is the same as the PK_{pr} stored in the storage module. Second, the system determines whether the evidence has been modified, i.e., whether the two evidence names are the same and the hash value of the evidence is $h(e_c)$. If the two aspects are both successfully verified, the evidence and corresponding information are permanently stored in the storage module. Especially for the two public keys PK_{pc} and PK_{pr}, they are also stored in the module such that the relationship between the two keys is maintained. After that, the result of integrity verification is delivered to the node.

In steps 6 and 7, a set of nodes in the blockchain is selected and used to verify the evidence validity. This process can be skipped if the EP does not wish to verify the validity for some reasons, such as privacy leakage or to avoid a cumbersome process. Validity verification may be a time-consuming process, and it some times requires human intervention. Since we only focus on one case, i.e., file tampering, the result can be obtained automatically and quickly by comparison of the file hash values. If the two hash values are equal, the validity is verified. For other cases, future study is required.

In the last step, after the integrity and validity have been verified, Block-DEF records the evidence information in a transaction and sends the verification results back to the EP.

4.2.2. Evidence retrieval

The participants in evidence retrieval are mostly evidence investigators. During evidence investigation, the investigator obtains evidences in three ways: it may be (1) provided by the evidence provider, (2) requested from Block-DEF, or (3) transferred from other investigators. Each time the investigator obtains an evidence, the integrity and validity of the evidence should be verified.

The processes used to obtain evidences in the above three ways are similar. Therefore, we present the first way, which is shown in Fig. 2.

Due to the privacy and sensitivity of evidence, an ER should first prove that it is authorized by an authority organization to retrieve the evidence. To satisfy this requirement, a multi-signature scheme is adopted. Request messages are signed by both the ER and an authority organization. A request message Msg is expressed as (OP, $h(f_n)|h(e_c)$), where OP is an operation on the evidence, e.g., download the evidence, verify the integrity or validity of the evidence, track the providers of the evidence, and so on. The message is first signed by a random key pair (PK_{rr}, SK_{rr}) and a certified key pair (PK_{rc}, SK_{rc}) of the ER. Then, the message and the signature $Sig_{SK_{rc}}(Sig_{SK_{rr}}(Msg))$, as well as the two public keys PK_{rr} and PK_{rc} are sent to the corresponding authority organization. After the message is verified, the signature $Sig_{SK_{rr}}(Msg)$ is re-signed by the organization using its certified key pair (PK_{oc}, SK_{oc}). When the multi-signed message is received by the ER, the message and its signature $Sig_{SK_{oc}}(Sig_{SK_{rr}}(Msg))$ are directly sent to Block-DES or to the EPs. Anyone who receives the message will verify whether the message has been authorized according to the digital certificate of the organization.

In steps 9 and 10, after the ER receives a message Msg_1 (OP_1, $h(f_n)|h(e_c)$) and its signature $Sig_{SK_{oc}}(Sig_{SK_{rr}}(Msg_1))$ from the authority organization, it requests a file tampering evidence from an EP using the message and the signature. The EP verifies

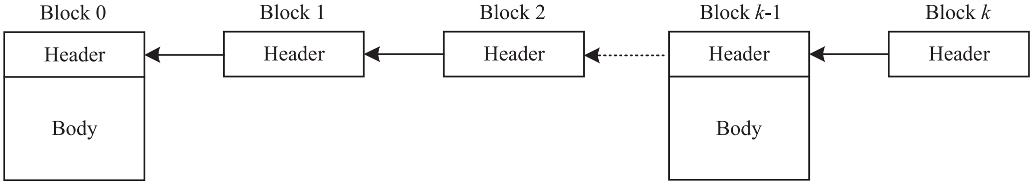

Fig. 3. Mixed blockchain structure.

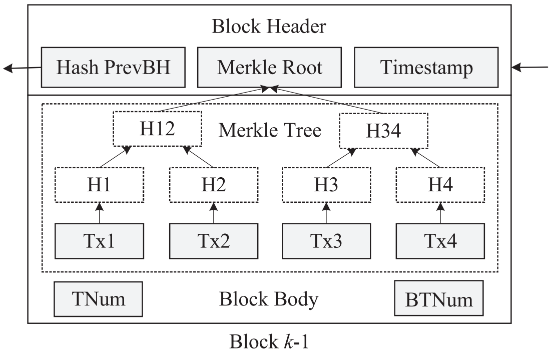

Fig. 4. Redesigned block structure based on Bitcoin.

that the message has been authorized by the corresponding organization, and it then transmits the evidence to the ER. Note that, the public key PK_{pr} should be the same as the random public key that is used in evidence submission.

In steps 11 and 12, when the evidence is received by the ER, the integrity and validity of the evidence should be verified. Thus, a verification request Msg_2 $(OP_2, h(f_n)|h(e_c))$, its signature $Sig_{SK_{oc}}(Sig_{SK_{rr}}(Msg_2))$ and the corresponding keys are sent to Block-DEF from the ER. Block-DEF verifies the message, queries the corresponding records stored in the blockchain or storage module, and sends the result back to the ER.

The multi-signature scheme is an effective approach for Block-DEF. On the one hand, the identity of ER is hidden. On the other hand, the authentication and access control of the ER are made by the authority organization, which effectively reduces the workload on Block-DEF.

4.3. Blockchain model

A blockchain is a distributed ledger in which multiple transactions are maintained by trustless nodes in a P2P network. It focuses on two main problems. (1) How to record the transactions in the blockchain, i.e., the structure of the blockchain. (2) How to construct the blockchain, i.e., the consensus mechanism. In Block-DEF, the information of evidence is recorded in a transaction, and multiple transactions are stored in a block. Block-DEF not only focuses on the above two problems, but also considers the requirement of scalability.

4.3.1. Structure of the blockchain

The main challenge of blockchain is the scalability issue [2]. The number of blocks increases linearly with time. All nodes participating in the network need to maintain a full copy of the blockchain which requires a huge storage space. The huge storage requirement also causes an endless ledger problem, i.e., the new node joining the blockchain requires too much time to download and verify the blockchain.

To overcome the scalability issue, Block-DEF adopts a mixed blockchain structure. Generally, a block consists of two parts: the block header and the block body. The body is considerably larger in size than the header. Most blockchains (such as cryptocurrency) require integrated blocks because verification of the transaction relies on other transactions that were recorded in previous blocks. In Block-DEF, transaction verification is independent of other transactions. A node does not have to frequently query other blocks when verifying a transaction. Thus, the blocks of the blockchain do not need to be integrated in Block-DEF. Each node in the blockchain network stores all the block headers, and the block bodies are stored in the network in a distributed manner. As shown in Fig. 3, only some of the blocks are integrated, the remaining blocks are all block headers. Which node stores the block body will be discussed in the next subsection.

Based on the mixed blockchain, the block structure of Block-DEF is redesigned according to that of Bitcoin [26]. As shown in Fig. 4, the block header consists of three parts: *Hash PrevBH, Timestamp* and *MerkleRoot. Hash PrevBH* is the hash value of the previous block header. It points to the previous block header and ensures that the previous block header is tamper-resistant. *Timestamp* is the time at which the block was created. *Merkle Root* is the root of merkle tree [25], and it is viewed as the hash value of all transactions in a block. The merkle tree uses the hash value of the transaction as the leaf node. For

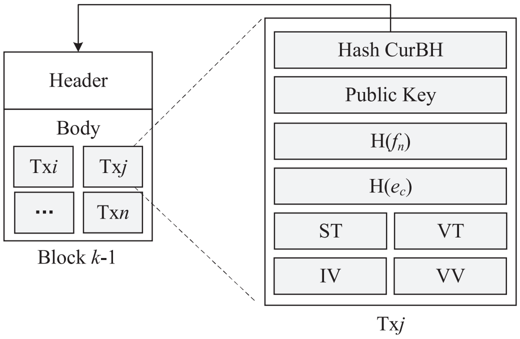

Fig. 5. Transaction structure of verified evidence.

the merkle tree, only the root node and transactions are stored in the block. The approach used by *Merkle Root* to verify a transaction is similar to that used by Bitcoin. All transactions (*Tx*1, *Tx*2...) are stored in the block body. *TNum* is the number of transactions represented by *BTNum* bytes.

A transaction is used to record a verified evidence. Fig. 5 shows the structure of a transaction. *Hash CurBH* is the hash value of the current block header. According to the hash value, a node can determine which block the transaction belongs to. This item is used for the evidence chain. *Public Key* is the public key PK_{pr} of the first EP of the evidence. An evidence may correspond to multiple EPs. Since the transaction cannot store the dynamically added public keys of these EPs, only the first public key is recorded in the transaction. The remaining public keys can be stored in the storage module or in a separate blockchain. $H(f_n)$ and $H(e_c)$ can uniquely identify the evidence. They are also used to prevent evidence tampering and track the corresponding source file. The transaction records two timestamps of the evidence, *ST* (submit timestamp) and *VT* (verify timestamp). *ST* is the time at which the evidence was submitted to Block-DEF, and *VT* is the time at which the evidence was verified. The verification results are recorded by *IV* (integrity verification) and *VV* (validity verification).

All transactions of the same file make up an evidence chain. A file may suffer from tampering at different times, and this may result in multiple evidences. To correlate these individual evidences, transactions with the same $h(f_n)$ can be stored in a set of nodes as a chain according to the name hash in the order of submission. The block corresponding to a specific evidence in the evidence chain can be found by *Hash CurBH*.

The mixed blockchain structure can significantly reduce the amount of storage space needed, but there is no full node in the network. Node failures may break the blockchain. To solve this problem Block-DEF can adopt two strategies: each block body can be distributed to multiple nodes in the blockchain network; the storage module can be treated as a full node, and the complete blockchain can be stored in this module.

4.3.2. Consensus mechanism

The consensus mechanism of blockchain chooses a node to create and broadcast the blockchain's next block and guarantees that the blockchain stored in each node is consistent. Nodes in public blockchains always use a costly mining approach with an economic incentive such as PoW [26] or PoS [39] to reach consensus. These consensus mechanisms can effectively guard against malicious nodes and failed nodes.

Unlike the public blockchain, Block-DEF is private or consortium, and it does not care about the economic incentive. Thus, each node is a block miner and does not need to compete for creating the next block. At the same time, the network of Block-DEF is controlled, so Block-DEF only needs to focus on failure nodes.

Block-DEF adopts a name-based consensus mechanism based on PBFT [7] (NPBFT). Most of the time, RAFT [29] is more suitable for the private blockchain. Since Block-DEF may use the consortium blockchain in some scenarios, PBFT is suitable for both the private blockchain and the consortium blockchain.

To support NPBFT, a name-based topology is constructed. In the topology, each node is assigned an arbitrary name. The node that creates the next block, is determined by the node name. Since each node stores the block header, when a new block i is created and broadcasted, each node can obtain the hash value of the block header $H(b_i)$. Thus, a node v with a name hash value $H(n_v)$ is chosen to generate the next block $i+1$ if $H(n_v)$ is the nearest to $H(b_i)$, i.e.,

$$v = \underset{u \in V}{\operatorname{argmin}}((H(n_u) \oplus H(b_i))) \tag{1}$$

where V is the node set of the blockchain network.

Since each new transaction will be broadcasted in the blockchain network, each node stores all unprocessed transactions. When node v finds that it is the generator of the next block, it counts the number of unprocessed transactions. If the number reaches a threshold, the node generates a new block that contains a list of the unprocessed transactions.

According to PBFT, NPBFT is divided into three steps, as shown in Fig. 6(a). First (Pre-prepare process), when the node v generates a new block, the block (B) is broadcasted to other nodes. Second (Prepare process), when each node receives the block, it verifies all transactions in the block by comparing the *merkle root* in the block with the *merkle root* the node

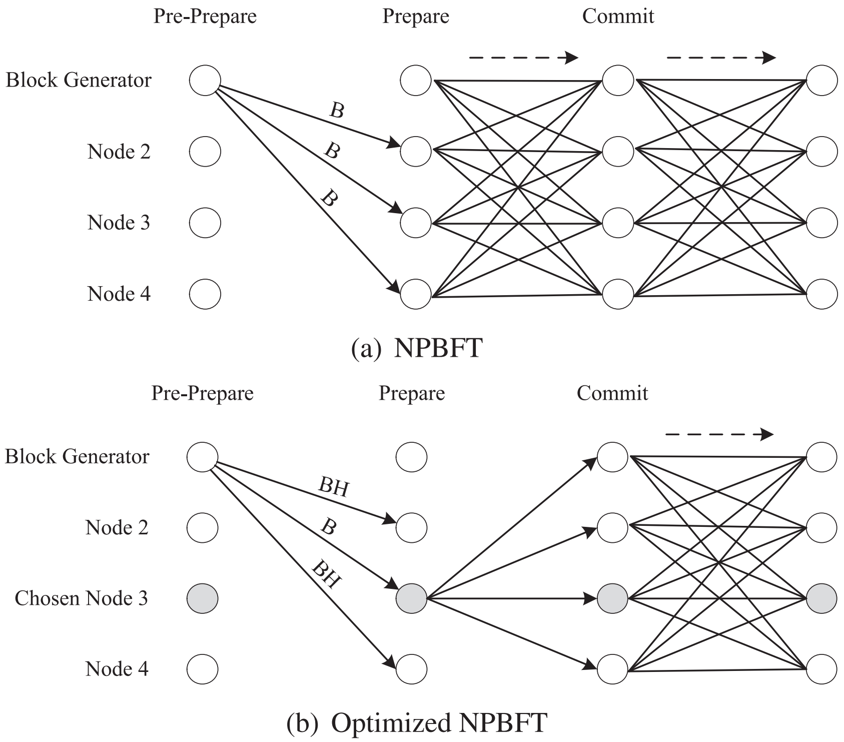

Fig. 6. NPBFT and Optimized NPBFT.

generated according to the transactions it stores. If the two values are equal, the hash value of the block header is calculated and broadcasted to other nodes. Meanwhile, each node can receive the hash value of the block header from other nodes. Third (Commit process), assume that the node number is n and $f = (n-1)/3$. For a node, if there are $2f+1$ received hash values that are equal to the hash value that the node calculates, the node broadcasts a commit message. Meanwhile, if the node receives $2f+1$ commit messages from other nodes, the block is stored in the blockchain.

To enhance the robustness of the blockchain, we divide the nodes into multiple groups, and all nodes in a group stores the same block bodies. According to the hashed node names, the nodes with the same first k bits are in the same group. Let n denote the size of the network, $k = \lceil \log(n) \rceil - h$, where 2^h is the size of the group. The first k bits can be viewed as the group ID. Thus, a block body is stored in a group if the group ID is the same as the first k bits of *hashBH* of the block body.

Due to the structure of the blockchain in Block-DEF, NPBFT can be optimized. Since the network environment is safe and most nodes probably only need to store the block header, there is no need for all nodes to participate in the block verification. As shown in Fig. 6(b), a set of nodes (the node number can be adjusted) are chosen to verify the block and are sent a complete block. The remaining nodes are sent block headers. In the prepare process, these chosen nodes verify the block and broadcast the verified result (the hashed block header) to other nodes. The verified result is then compared with the hash value of the block header each node received. If a node receives results from multiple chosen nodes and the number of correct comparison results is greater than half of the total result number, the node broadcasts a commit message. The remainder of the process is the same as for NPBFT. The optimized NPBFT not only decreases the communication overhead, but also provides different levels of security by dynamically adjusting the number of chosen nodes.

Since the size of the blockchain network is much smaller than that of the namespace, the distribution of the block number each node generates and the number of block bodies each node stores may be unbalanced. To solve this problem, virtual node, which is always used in P2P networks for load balance [35], can also be applied to Block-DEF. In Block-DEF, each node is assigned k names, and each name participates in the creation and storage process of the blockchain. Thus, $k-1$ names can be viewed as $k-1$ virtual nodes.

5. Analysis and evaluation

In this section, we first analyze the properties of Block-DEF and discuss whether Block-DEF can fulfill the requirements proposed in Section 3.1, and then evaluate the performance of Block-DEF through some simulation experiments.

5.1. Property analysis

Block-DEF focuses primarily on five properties: scalability, integrity, validity, privacy and traceability. Here, we adopt either a quantitative or a qualitative analysis method.

5.1.1. Scalability

Block-DEF provides a scalable blockchain. In Block-DEF, the scalability of the blockchain includes two aspects: the size of the blockchain and the communication overhead of the consensus mechanism.

For a network with n nodes, the average size of the group is g when the number of node names is 1. Let S_t denote the size of a transaction and N_t denote the number of transactions in a block. For a blockchain with l blocks, the size of blockchain S is expressed as follows,

$$\begin{aligned} S &= l \cdot \frac{g}{n}(S_h + N_t \cdot S_t) + l \cdot \left(1 - \frac{g}{n}\right) \cdot S_h \\ &= l \cdot \left(S_h + N_t \cdot S_t \cdot \frac{g}{n}\right) \end{aligned} \tag{2}$$

If each node has N_k names, the node belongs to N_k groups. Since each group stores the same block body, the size of blockchain S can be expressed as follows,

$$S = l \cdot \left(S_h + N_t \cdot S_t \cdot \frac{N_k \cdot g}{n}\right) \tag{3}$$

Obviously, the blockchain size S is much smaller than the size of the blockchain with complete blocks.

The communication overhead of the consensus mechanism consists of three parts: messages associated with the pre-prepare precess, messages associated with the prepare process and messages associated with the commit process. According to NPBFT and Optimized NPBFT (O-NPBFT), the communication overheads (measured by the number of messages) of the two mechanisms are expressed as follows,

$$M_{NPBFT} = 2n^2 + n - 1 \tag{4}$$

$$M_{O-NPBFT} = n^2 + (N_h + 1) \cdot n - 1 \tag{5}$$

where n is the network size and N_h is the number of nodes chosen to verify the block in O-NPBFT.

Assume that S_h is the size of the block header, S_b is the size of the block and S_c denotes the size of the commit message. If we take the message size into account, then the communication overheads (measured by message overhead) of the two mechanisms are expressed as follows,

$$M'_{NPBFT} = S_b \cdot (n^2 + n - 1) + S_c \cdot n^2 \tag{6}$$

$$M'_{O-NPBFT} = S_b \cdot N_h \cdot (n + 1) + S_h \cdot (n - N_h - 1) + S_c \cdot n^2 \tag{7}$$

The communication complexities of both NPBPF and O-NPBPF are $O(n^2)$, and they are the same as the communication complexity of PBFT. However, for O-NPBPF, both the message number and the message overhead are significantly reduced, which is also verified by the experimental results.

5.1.2. Integrity and validity

Block-DEF not only guarantees that the submitted evidence is not modified, but also determines whether the original data has been tampered with as the evidence described. The integrity is verified by comparison of the hash value of the evidence and $H(e_c)$ in the evidence name. Since the evidence name is stored in the evidence transaction, the integrity of the evidence can always be verified regardless of where the evidence came from.

The validity is based on the integrity. Only when the integrity is successfully verified can the validity be verified. Since both properties (*IV* and *VV*) of the evidence are recorded in the transaction, evidences obtained from other sources must first be assessed to determine whether they are the same as the evidence in the storage module, i.e., its integrity must be evaluated, and then we obtain their validity results from the corresponding transactions.

5.1.3. Privacy and traceability

Block-DEF solves the conflict between the privacy and the traceability of evidence by using multi-signature schemes with a two-layer signature. For the two-layer signature, the first layer signature is signed by a random key pair, and the second signature is signed by a traceable key pair.

Privacy is naturally supported by the blockchain technology. Each participant in Block-DEF (EP or ER) can randomly generate multiple key pairs that can be used to hide the identity of the participant. Thus, any signature signed by these random key pairs cannot be tracked. The traceability is supported by a traceable key pair. The traceable key pair may be certified by a certificate authority or belong to an authority organization. Given a signature signed by the traceable key pair,

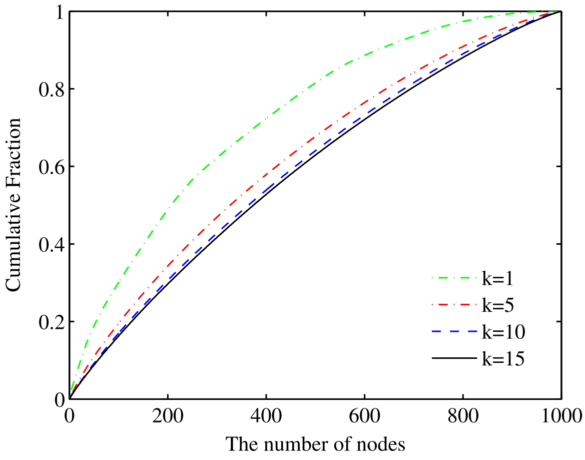

Fig. 7. The cumulative distribution of generated blocks.

the EP or ER can be identified by querying the certificate authority or the authority organization. At the same time, the authority organization can also provide control access for evidence, only the authorized operation can be issued and signed by the key pair of the authority organization.

Only Block-DEF stores both the random public key and the traceable public key of a participant. The relationship between the two keys is unknown to others. Thus, no one other than the Block-DEF who obtains information from the blockchain can track the evidence.

5.2. Evaluation

We evaluated Block-DEF by simulation on full-connected overlay topologies, i.e., each node is one hop to the others. To simulate the virtual nodes, each node is assigned k names. The bit length of the group ID used for blockchain storage is set to $\lceil \log(n) \rceil - h$. We considered three parameters for our experiments: the load of the block generator, the size of the blockchain and the communication overhead of the consensus mechanism.

5.2.1. The load of the block generator

The load of the block generator measures the distribution of blocks generated by each node. It reveals whether each node has an equal opportunity to generate blocks in NPBFT. We adopted a topology with 1000 nodes and sequentially generated 10^5 blocks, then counted the number of blocks each node generated.

Fig. 7 shows the cumulative fraction of generated block number with x nodes. The variable k is the number of node names. The nodes on the x-axis are ranked by the number of generated blocks in descending order. For a point (x, y), y is expressed as $y = \sum_{i=1}^{i=x}(N_i)/\sum_{j=1}^{j=n}(N_j)$, where N_i is the block number of the ith node. The more uniformly the load is distributed, the more likely it is that the line will be straight. When $k = 1$, the curve shows a steep increase. By increasing the number of node names, the load of the generator is balanced uniformly. The more node names there are, the closer the increase is to linear. However, as the number of node names increases, the growth effect of load balancing gradually decreases. Overall, these block generators can effectively achieve better load balancing by adding a small number of node names.

Fig. 8 shows the distribution of generated block numbers when the number of node names k is 5. In Fig. 8, the x-axis is the block number, and the y-axis is the proportion of node numbers. The mean number of blocks generated by each node is 100. According to the distribution of generated block numbers in the figure, the proportion of nodes with block numbers between 50 and 130 is approximately 70%. The number of generated blocks is concentrated around the mean. Overall, when $k = 5$, the effect of load balancing is acceptable for the block generator.

5.2.2. The size of the blockchain

In this experiment, we focus on the distribution of complete blocks (with block headers and block bodies) and on the size of the blockchain.

The distribution of complete blocks that each node stores reflects the load balancing of blockchain storage. A topology with 1000 nodes is still adopted, and 10^5 blocks are broadcasted. The number of node names is set to 1. We repeat the experiment three times. Each time, we adopted different group sizes by adjusting the variable h, then counted the number of complete blocks each node stores.

Fig. 9 shows the cumulative fraction of block numbers with x nodes. This figure is similar to Fig. 7. Obviously, the distribution of complete blocks each node stores is balanced, because all three curves are close to linear. According to the inset

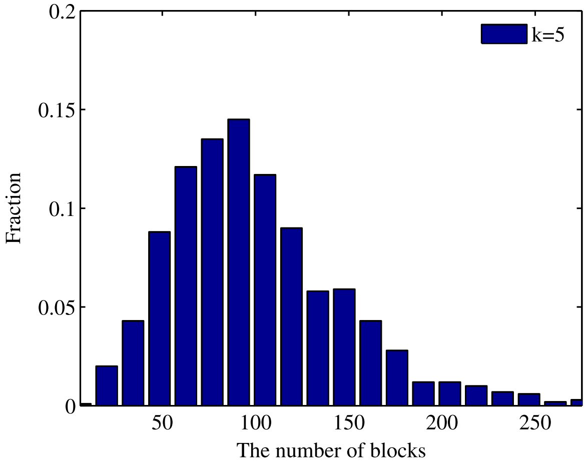

Fig. 8. The distribution of generated blocks when k=5.

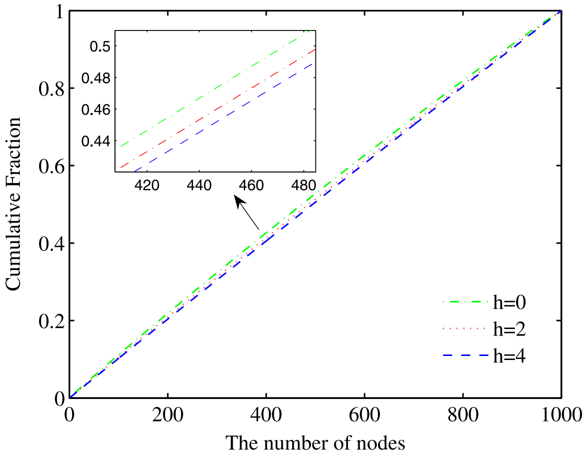

Fig. 9. The cumulative distribution of complete blocks.

in the figure, the effect is slightly better for the experiment with larger group size. The main difference in these three experiments is the mean number of complete blocks, and the mean complete block numbers are 1.9, 7.8 and 31.2, respectively. Overall, Block-DEF shows a good performance on load balancing of blockchain storage, and the group size has little effect on the result.

To evaluate the size of the blockchain, we produced different numbers of blocks on a topology with 1000 nodes. For the topology, the name number is set to 1 and the variable h for group size is set to 3 bits. The number of transactions in a block is set to 2000. Then, we measured the storage space of blockchain on each node.

Fig. 10 shows the blockchain size with the growth of block number. The max, mean and min sizes of the blockchain are all measured based on the mixed blockchain, and the full size of blockchain is based on a general scenario in which all nodes store the complete blockchain. It is clear that the size of the mixed blockchain is much smaller than that of the general blockchain. For all four types of result, the size of the blockchain increases linearly with the increase in block number, which is in accordance with the theoretical analysis.

5.2.3. The communication overhead of the consensus mechanism

Since the topology is a simplified overlay topology regardless of the underlying topology, it is difficult to measure the network load, the communication latency, and other parameters. Instead, in this experiment, we measured the message number and the message overhead (the number of bytes of all messages) as a basis for evaluating the consensus mechanism with different sizes of topologies. For each topology, both NPBFT and O-NPBFT are measured. The number of nodes used to verify the block in O-NPBFT is denoted by m. Here, we set m to 1, 50 and 100, respectively.

Figs. 11 and 12 show the message number and the message overhead change as the size of the topology increases, respectively. The results of O-NPBFT are lower than those of NPBFT for both message number and message overhead. The message number of O-NPBFT is approximately half that of NPBFT. This is because only m nodes send messages to others

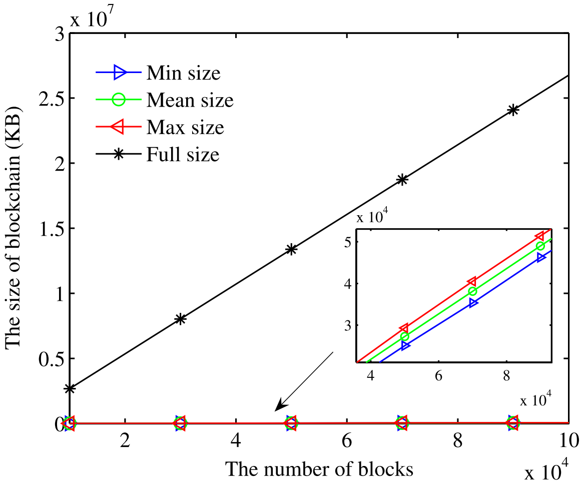

Fig. 10. The size of blockchain.

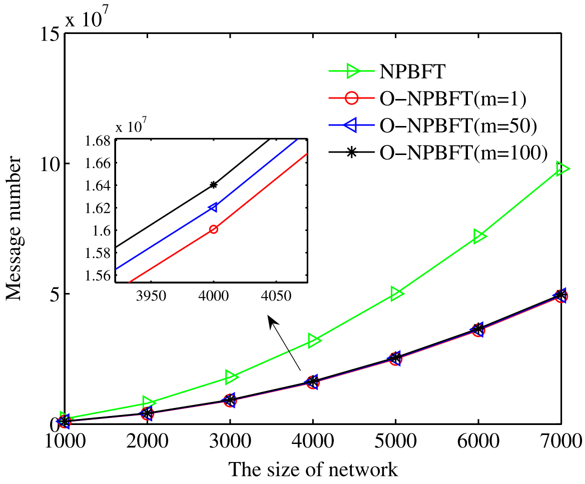

Fig. 11. The message numbers of NPBFT and O-NPBFT.

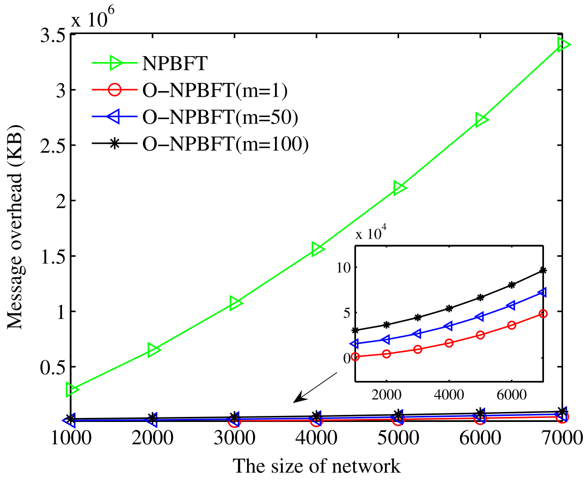

Fig. 12. The message overheads of NPBFT and O-NPBFT.

during the prepare process of O-NPBFT, while all nodes send messages to others for NPBFT. For message overhead, there is a much larger difference in the number of bytes between O-NPBFT and NPBFT. The main reason for this is that the short messages in the commit process make up most of the O-NPBFT's messages. In conclusion, O-PBFT can adjust *m* according to different security requirements and effectively reduce the communication overhead.

6. Conclusion

This paper focuses on the security of digital evidence for file tampering. Based on the blockchain technology, we proposed a secure digital evidence framework, Block-DEF. Block-DEF adopts a loose coupling design by combining a redesigned scalable blockchain module with an existing storage module, and it provides integrity and validity verification for evidences. The scalable blockchain module adopts a mixed blockchain structure and an optimized name-based PBFT. Meanwhile, to compatible for privacy and traceability, multi-signature schemes with random and certificated key pairs are adopted. Our analyses and experiments demonstrate that Block-DEF can well satisfy the requirements for scalability, integrity, validity, privacy and traceability.

Acknowledgment

This work is funded by the National Natural Science Foundation of China (No. 61871140, 61702223, 61702220, 61572153, U1636215) and the National Key Research and Development Plan (Grant No. 2018YFB0803504).

References

[1] M. Alam, Z.C. Lee, C. Nicopoulos, K.H. Lee, J. Kim, J. Lee, SBBox: A tamper-resistant digital archiving system, Int. J. Cyber-Secur. Digit. Forensics 5 (3) (2016) 122–131.

[2] M. Ali, J.C. Nelson, R. Shea, M.J. Freedman, Blockstack: A global naming and storage system secured by blockchains., in: Proceedings of the USENIX Annual Technical Conference, 2016, pp. 181–194.

[3] R.R. Ana Nieto, J. Lopez, Digital witness and privacy in iot: Anonymous witnessing approach, in: Proceedings of the IEEE Conference on Trustcom/BigDataSE/ICESS, IEEE, 2017, pp. 1–8.

[4] J. Benet, IPFS-content addressed, versioned, p2p file system (DRAFT 3), arXiv:1407.3561 (2014), https://arxiv.org/pdf/1407.3561.pdf.

[5] S. Bonomi, M. Casini, C. Ciccotelli, B-CoC: A blockchain-based chain of custody for evidences management in digital forensics, arXiv:1807.10359 (2018), https://arxiv.org/pdf/1807.10359.pdf.

[6] J. Camenisch, M. Stadler, Efficient group signature schemes for large groups, in: Proceedings of the Annual International Cryptology Conference, Springer, 1997, pp. 410–424.

[7] M. Castro, B. Liskov, et al., Practical byzantine fault tolerance, in: Proceedings of the OSDI, 99, 1999, pp. 173–186.

[8] M. Cebe, E. Erdin, K. Akkaya, H. Aksu, S. Uluagac, Block4forensic: An integrated lightweight blockchain framework for forensics applications of connected vehicles, IEEE Commun. Mag. 56 (10) (2018) 50–57.

[9] C.N. Chong, Z. Peng, P.H. Hartel, Secure audit logging with tamper-resistant hardware, in: Proceedings of the IFIP International Information Security Conference, Springer, 2003, pp. 73–84.

[10] M. Dong, K. Ota, L.T. Yang, A. Liu, M. Guo, LSCD: A low-storage clone detection protocol for cyber-physical systems, IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. 35 (5) (2016) 712–723.

[11] X. Du, H.H. Chen, Security in wireless sensor networks, Wirel. Commun. IEEE 15 (4) (2008) 60–66.

[12] X. Du, M. Guizani, Y. Xiao, H.H. Chen, Transactions papers a routing-driven elliptic curve cryptography based key management scheme for heterogeneous sensor networks, IEEE Trans. Wirel. Commun. 02 (05) (2011) 1223–1229.

[13] X. Du, Y. Xiao, M. Guizani, H.H. Chen, An effective key management scheme for heterogeneous sensor networks, Ad Hoc Netw. 5 (1) (2007) 24–34.

[14] K. Gai, M. Qiu, Blend arithmetic operations on tensor-based fully homomorphic encryption over real numbers, IEEE Trans. Ind. Inf. 14 (8) (2018) 3590–3598.

[15] K. Gai, M. Qiu, X. Sun, A survey on FinTech, J. Netw. Comput. Appl. 103 (2018) 262–273.

[16] A. Ghodsi, T. Koponen, J. Rajahalme, P. Sarolahti, S. Shenker, Naming in content-oriented architectures, in: Proceedings of the ACM SIGCOMM Workshop on Information-centric Networking, ACM, 2011, pp. 1–6.

[17] F. Hu, M. Qiu, J. Li, T. Grant, D. Taylor, S. McCaleb, L. Butler, R. Hamner, A review on cloud computing: design challenges in architecture and security, J. Comput. Inf. Technol. 19 (1) (2011) 25–55.

[18] M.A. Khan, K. Salah, IoT security: review, blockchain solutions, and open challenges, Futur. Gener. Comput. Syst. 82 (2018) 395–411.

[19] A. Kosba, A. Miller, E. Shi, Z. Wen, C. Papamanthou, Hawk: The blockchain model of cryptography and privacy-preserving smart contracts, in: Proceedings of the IEEE Symposium on Security and Privacy (SP), IEEE, 2016, pp. 839–858.

[20] L. Kuang, L. Yang, J. Feng, M. Dong, Secure tensor decomposition using fully homomorphic encryption scheme, IEEE Trans. Cloud Comput. 6 (3) (2018) 868–878.

[21] J. Li, J. Wu, L. Chen, Block-secure: blockchain based scheme for secure p2p cloud storage, Inf. Sci. 465 (2018) 219–231.

[22] Y. Liu, M. Dong, K. Ota, A. Liu, Activetrust: secure and trustable routing in wireless sensor networks, IEEE Trans. Inf. Forensics Secur. 11 (9) (2016) 2013–2027.

[23] A. Loibl, J. Naab, Namecoin, namecoin. info (2014).

[24] P. Maymounkov, D. Mazieres, Kademlia: A peer-to-peer information system based on the XOR metric, in: Proceedings of the International Workshop on Peer-to-Peer Systems, Springer, 2002, pp. 53–65.

[25] R.C. Merkle, A digital signature based on a conventional encryption function, in: Proceedings of the Conference on the Theory and Application of Cryptographic Techniques, Springer, 1987, pp. 369–378.

[26] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system (2008).

[27] A. Nieto, R. Roman, J. Lopez, Digital witness: safeguarding digital evidence by using secure architectures in personal devices, IEEE Netw. 30 (6) (2016) 34–41.

[28] K. Ohta, T. Okamoto, A digital multisignature scheme based on the Fiat–Shamir scheme, in: Proceedings of the International Conference on the Theory and Application of Cryptology, Springer, 1991, pp. 139–148.

[29] D. Ongaro, J.K. Ousterhout, In search of an understandable consensus algorithm., in: Proceedings of the USENIX Annual Technical Conference, 2014, pp. 305–319.

[30] M. Qiu, K. Gai, B. Thuraisingham, L. Tao, H. Zhao, Proactive user-centric secure data scheme using attribute-based semantic access controls for mobile clouds in financial industry, Futur. Gener. Comput. Syst. 80 (2018) 421–429.

[31] V.K. Rayi, Y. Xiao, B. Sun, X.J. Du, F. Hu, A survey of key management schemes in wireless sensor networks, Comput. Commun. 30 (11) (2007) 2314–2341.

[32] A. Saberi, M. Vahidi, B.M. Bidgoli, Learn to detect phishing scams using learning and ensemble? Methods, in: Proceedings of the IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology-Workshops, IEEE Computer Society, 2007, pp. 311–314.

[33] B. Schneier, J. Kelsey, Secure audit logs to support computer forensics, ACM Trans. Inf. Syst. Secur. 2 (2) (1999) 159–176.

[34] K. Sitara, B.M. Mehtre, Digital video tampering detection: an overview of passive techniques, Digit. Investig. 18 (2016) 8–22.

[35] I. Stoica, R. Morris, D. Karger, M.F. Kaashoek, H. Balakrishnan, Chord: A scalable peer-to-peer lookup service for internet applications, ACM SIGCOMM Comput. Commun. Rev. 31 (4) (2001) 149–160.

[36] Q. Tan, Y. Gao, J. Shi, X. Wang, B. Fang, Z.H. Tian, Towards a comprehensive insight into the eclipse attacks of Tor hidden services, IEEE Internet Things J. (2018), doi:10.1109/JIOT.2018.2846624.

[37] Z. Tian, W. Shi, Y. Wang, C. Zhu, X. Du, S. Su, Y. Sun, N. Guizani, Real time lateral movement detection based on evidence reasoning network for edge computing environment, IEEE Trans. Ind. Inf. (2019), doi:10.1109/TII.2019.2907754.

[38] Z. Tian, S. Su, W. Shi, X. Du, M. Guizani, X. Yu, A data-driven method for future internet route decision modeling, Futur. Gener. Comput. Syst. 95 (2019) 212–220.

[39] P. Vasin, Blackcoins proof-of-stake protocol v2, https://blackcoin.co/blackcoin-pos-protocol-v2-whitepaper.pdf (2014).

[40] L. Wang, Z. Zhang, M. Dong, L. Wang, Z. Cao, Y. Yang, Securing named data networking: attribute-based encryption and beyond, IEEE Commun. Mag. 56 (11) (2018) 76–81.

[41] S. Wilkinson, J. Lowry, T. Boshevski, Metadisk a blockchain-based decentralized file storage application, Technical Report, 2014. Technical Report.

[42] G. Wood, Ethereum: A secure decentralised generalised transaction ledger, Ethereum project yellow paper 151 (2014) 1–32.

[43] G. Xylomenos, C.N. Ververidis, V.A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K.V. Katsaros, G.C. Polyzos, et al., A survey of information-centric networking research., IEEE Commun. Surv. Tutor. 16 (2) (2014) 1024–1049.

[44] X. Yang, X. Du, J. Zhang, H. Fei, S. Guizani, Internet protocol television (IPTV): the killer application for the next-generation internet, IEEE Commun. Mag. 45 (11) (2007) 126–134.

[45] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J.D. Thornton, D.K. Smetters, B. Zhang, G. Tsudik, D. Massey, C. Papadopoulos, et al., Named Data Networking (NDN) Project, Relatório Técnico NDN-0001, Xerox Palo Alto Research Center-PARC, 2010.

[46] Y. Zhang, R.H. Deng, X. Liu, D. Zheng, Blockchain based efficient and robust fair payment for outsourcing services in cloud computing, Inf. Sci. 462 (2018) 262–277.

[47] Z. Zhang, W. Cao, Z. Qin, L. Zhu, Z. Yu, K. Ren, When privacy meets economics: enabling differentially-private battery-supported meter reporting in smart grid, in: Proceedings of the IEEE/ACM 25th International Symposium on Quality of Service (IWQoS), IEEE, 2017, pp. 1–9.

[48] L. Zhu, Y. Wu, K. Gai, K.-K. R. Choo, Controllable and trustworthy blockchain-based cloud data management, Futur. Gener. Comput. Syst. 91 (2019) 527–535.

[49] G. Zyskind, O. Nathan, et al., Decentralizing privacy: using blockchain to protect personal data, in: Proceedings of the IEEE Security and Privacy Workshops (SPW), IEEE, 2015, pp. 180–184.