A Trustworthy Safety Inspection Framework Using Performance-Security Balanced Blockchain

Weishan Zhang*, Gang Sun*, Liang Xu‡, Qinghua Lu†, Huansheng Ning‡,
Peiying Zhang*, Su Yang§
*College of Computer Science and Technology, China University of Petroleum (East China), China
†Data61, CSIRO, Australia
‡College of Computer & Communication Engineering, Beijing University of Science and Technology, China
§College of Computer Science, Fudan University, China

***Abstract*—Regular safety inspection is critical to reduce safety risk in industry. Applying consortium blockchain technology to safety inspection can ensure the effectiveness of inspection process and tracing of problems. However, there are two major issues when using conventional consortium blockchain. It is challenging to guarantee the authenticity of the retrieved data source, and meanwhile, achieving a balance between performance and security is not easy. Hence, this paper proposes a blockchain based performance-security balanced safety inspection framework (PSB-SIF), in which a safety inspection box is designed to ensure the authenticity of the inspector's identity while inspection logic is executed automatically via smart contracts. In addition, this paper also proposes a novel credit scoring based Byzantine fault tolerant (BFT) consensus algorithm, named Safety Inspection Byzantine Fault Tolerance consensus algorithm (SIBFT), which is used to balance the performance and security of consensus network in safety inspection. We evaluate the proposed approach by comparing with the solutions using RAFT, Practical Byzantine Fault Tolerance (PBFT), and SIBFT consensus algorithms in terms of throughput, transaction latency, scalability and security of PSB-SIF. The evaluation results show that PSB-SIF is efficient for all these quality metrics.**

***Index Terms*—Blockchain, safety, performance-security balance, authenticity, consensus.**

I. INTRODUCTION

EFFECTIVE safety inspection [1] is important as accidents can bring huge losses to the society. Blockchain technology has been applied in many areas, such as cryptocurrency [2], financial services [3], and supply chain [4]. One promising application area is credible safety inspection, as blockchain can ensure the authenticity of inspection data and improve the effectiveness of the inspection process. Given the privacy requirement of inspection data, it is necessary to adopt consortium blockchain technology [5]. However, there are two major issues when applying consortium blockchain for safety inspection.

- **The authenticity of data sources is difficult to control.** Although the data stored on blockchain is difficult to tamper, it is difficult to guarantee the authenticity of the data source before the data is stored on blockchain [6]. Some safety inspectors consider the inspection process is a dull routine and upload fake inspection records to the blockchain without reaching the inspection location, resulting in fake data stored on the blockchain at the beginning.
- **Performance and security are hard to be balanced using classical consortium blockchain.** This has been a dilemma for using blockchains to develop real applications with good performance.

Weishan Zhang and Peiying Zhang are the corresponding authors. Email: zhangws@upc.edu.cn, zhangpeiying@upc.edu.cn

In general, there are two types of blockchain, permissionless blockchain and permissioned blockchain [7]. Permissionless blockchains, such as Bitcoin [8] and Ethereum [9], usually use Proof of Work (PoW) consensus algorithm [10] to ensure security because of less verification of participant's identity. The PoW consensus algorithm wastes resources with low efficiency. Therefore, it is not appropriate for real-world safety inspection applications. The permissioned blockchain usually uses consensus algorithms with higher consensus efficiency and lower security [11], such as kafka [12] and RAFT [13], due to the addition of identity authentication mechanism. Hyperledger Fabric (HLF) is a representative permissioned and consortium blockchain that uses Certification Authority (CA) to manage members. Joining members will get public-private key pairs representing their identities. Once the public-private key pair is leaked, the entire blockchain network will face the problem of being unable to handle malicious node attacks [14]. The consortium blockchain sacrifices too much security when pursuing performance, and cannot achieve a balance between performance and security. The PBFT consensus algorithm can tolerant a certain number of malicious nodes and seems to be a good choice. However, the performance of PBFT degrades dramatically as the number of consensus nodes increases [15]. Thus, to adopt consortium blockchain for safety inspection applications, we need to design a performance and security balanced consensus algorithm.

To balance security and performance in consortium blockchain, Wang et al. [16] proposed a method of leader node election based on HLF. Dynamic elections are carried out based on the ratings of consensus nodes by users in the consortium blockchain, which improves the stability and security of consensus reached by consortium blockchain network. However, this method is only suitable for dynamic election of peer leader nodes. In Proof of Vote consensus algorithm (PoV) [17], different types of nodes undertake different roles,

and only the elected commissioner nodes need to participate in consensus. This inspires us to design a new consensus algorithm using leader node election mechanism, which elects some leader nodes rather than all nodes to participate in consensus, thereby reducing the consensus complexity of network communication. In order to enhance the security of the consensus process, Dual Vote Confirmation based Consensus (DVCC) [15] proposes a credit evaluation mechanism for determining the voting rights of consensus and the probability of a node being selected as a miner. However, the credit evaluation mechanism cannot detect and punish malicious nodes.

Fingerprint is a technology used to verify an entity's identity information by means of fingerprint recognition [18]. Connecting to blockchain through fingerprint recognition can effectively guarantee the authenticity of the identity source. Similar to human fingerprints, chip fingerprints are the unique identification of the chips. Each chip has a unique chip fingerprint and cannot be copied. The chip fingerprint information is stored on the blockchain to ensure that cannot be tampered. At the same time, the chip fingerprint anchors the unique chip in the physical world, connects the blockchain and the physical world, and ensures the authenticity of the data source on the blockchain, so that it cannot be recycled, relabeled or cloned.

In this paper, we propose a novel safety inspection framework using Performance-Security Balanced blockchain (PSB-SIF) for the safety inspection of petroleum factories, power grid, etc. In PSB-SIF, a blockchain safety inspection box is designed to ensure the authenticity of inspector's identity source and prevent identity fraud in the process of inspection. On the other hand, consortium blockchain framework HLF is improved for better fitting safety inspection scenarios, where a new type of consortium blockchain consensus algorithm SIBFT is designed based on the distribution and characteristics of members in safety inspection scenarios. We compare the scalability, security, throughput and transaction latency of PSB-SIF using RAFT, PBFT, and SIBFT consensus algorithms. The contributions of this paper include:

- A safety inspection box composed of NFC and Raspberry Pi devices is designed to ensure the authenticity of the source of inspector's identity. The inspector's identity is obtained through safety inspection box and stored on the blockchain. When performing security inspection tasks, the unique safety inspection smart contract installed on the blockchain performs identity authentication and authorization to ensure the authenticity of inspection process.
- A unique SIBFT consensus algorithm is designed for safety inspection to balance performance and security of consortium blockchain network. Meanwhile, a novel credit scoring mechanism in SIBFT is designed for the detection of malicious nodes and the switching of leader nodes.
- Comprehensive evaluations of PSB-SIF are made using three different consensus algorithms, and evaluations show that the performance and security of PSB-SIF using SIBFT consensus algorithm are effectively improved for safety inspection scenarios.

The rest of this paper is organized as follows. Section II gives a detailed description of PSB-SIF. Section III evaluates the PSB-SIF constructed by three different consensus algorithms. Section IV presents related work. Section V concludes this paper and outlines future work.

II. PSB-SIF OVERVIEW

PSB-SIF aims to build a general safety inspection framework based on HLF blockchain infrastructure, which can be applied to the safety inspection of fire, petroleum factories, electric power factories, etc. The PSB-SIF system architecture is shown in Fig. 1.

The software architecture of PSB-SIF is divided into three main layers as shown in Fig. 2. They are application layer, blockchain layer, and physical layer. The physical layer uses inspection equipment and blockchain safety inspection box to connect an inspector and blockchain network to ensure the authenticity of inspector's identity. The blockchain layer is composed of smart contract, private data, and consensus. The smart contract is a piece of code installed on the blockchain, which realizes six functions of "Inspection record", "Inspector identity", "Record query", "Identity query", "Task arrangement" and "Certificate issuance". We use channel isolation and private data collection to protect the privacy of inspectors, and only expose the identity of inspectors to supervision. Based on the HLF consensus algorithm interface, we design and implement SIBFT consensus algorithm to meet the needs of actual safety inspection scenarios. The application layer is responsible for interacting directly with users, which includes three main functions, namely "Task arrangement", "Personnel authorization" and "Safety certificate issuance".

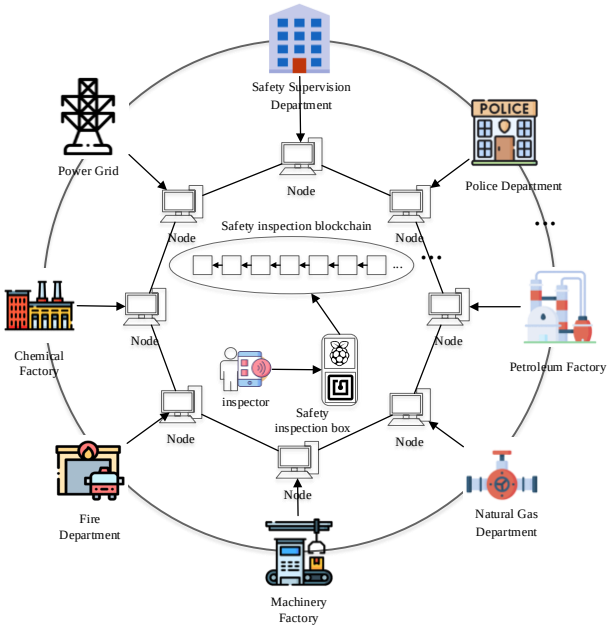

Fig. 1: System architecture

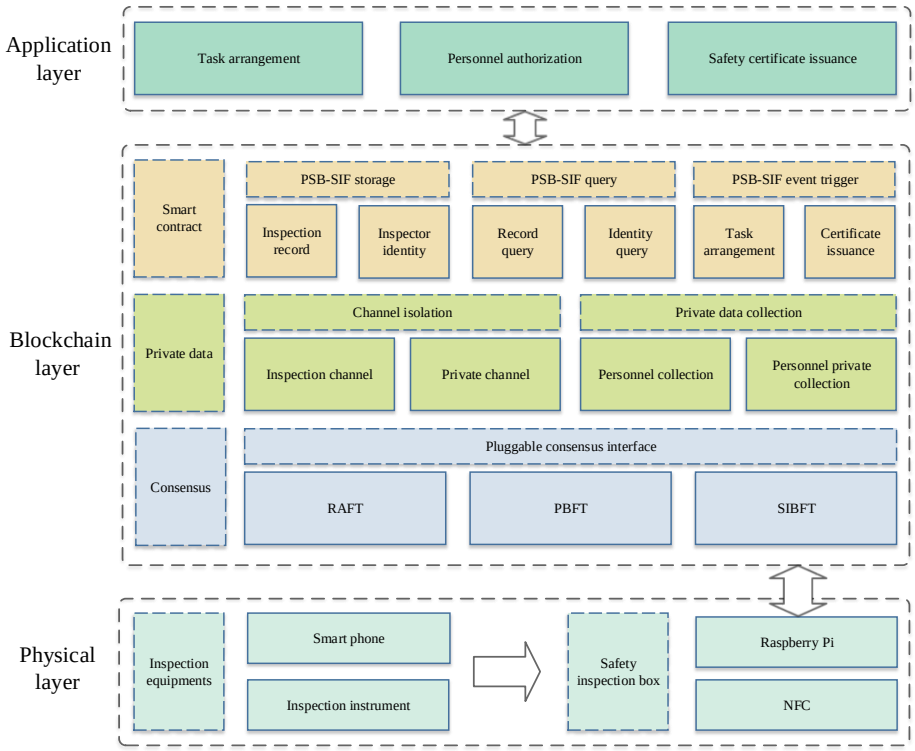

Fig. 2: Architecture design of the PSB-SIF

A. *Blockchain safety inspection box*

Although the data stored on the blockchain is difficult to be tampered, it is hard to guarantee the authenticity of the data source. Therefore, we design the blockchain based safety inspection box and placed it in the safety cabinet at the inspection location. It is used to identify and authorize inspectors when performing inspection tasks.

The blockchain safety inspection box is composed of Near Field Communication (NFC) identification equipment and a Raspberry Pi. The NFC identification equipment is used to obtain the identity information of the safety inspectors and the fingerprint of the NFC chip. The chip fingerprint is unique and cannot be copied. This ensures that the information of the inspection personnel on the blockchain corresponds to the actual safety inspector in a one-to-one correspondence manner, which is irreplaceable. The chip fingerprint information is obtained by actively monitoring the NFC chip through the inspection box, which can prevent acquiring fake chip fingerprint data during the passive acceptance process.

Since NFC devices cannot send requests to the blockchain network directly, we use a Raspberry Pi as a terminal to accept the identity of the inspector and the chip fingerprint information is sent by NFC identification device. As this process is completed in the safety inspection box, the data can not be tampered during the transmission. Then, an inspector can use the identity information to initiate the inspection authorization request to the blockchain. Only inspectors authorized by blockchain can perform inspection tasks.

TABLE I: Orderer type and functions

Orderer type	Functions
leader	1. Receive messages sent by client 2. Send the messages to SL nodes
SL	1. Receive messages sent by leader node 2. Send messages to replica in subnet
SCL	1. Replace SL when it fails
replica	1. Store replica of blockchain

B. *SIBFT consensus algorithm for PSB-SIF*

In order to balance performance and security of HLF consensus algorithm, for safety inspection scenarios, we designed a novel consensus algorithm, SIBFT, based on the pluggable consensus interface of HLF.

The SIBFT's Application Programming Interface (API) and design are shown in Fig. 3. An interface called Chain is used to start, stop and configure the blockchain, which is implemented by the Chain class in the *sibft* package. The Chain class is associated with a Node class. The Sequence, SharedConfig, HttpServer, Buffer, CreditScore classes, and NodeType enumeration are aggregated into the Node class. The CreditScore class is used to calculate and process credit scores. The NodeType enumeration is used to mark the type of nodes.

In SIBFT, the types of consensus nodes are divided into four types, leader, sub leader (SL), sub candidate leader (SCL) and replica. Their roles in the consensus are shown in Table I.

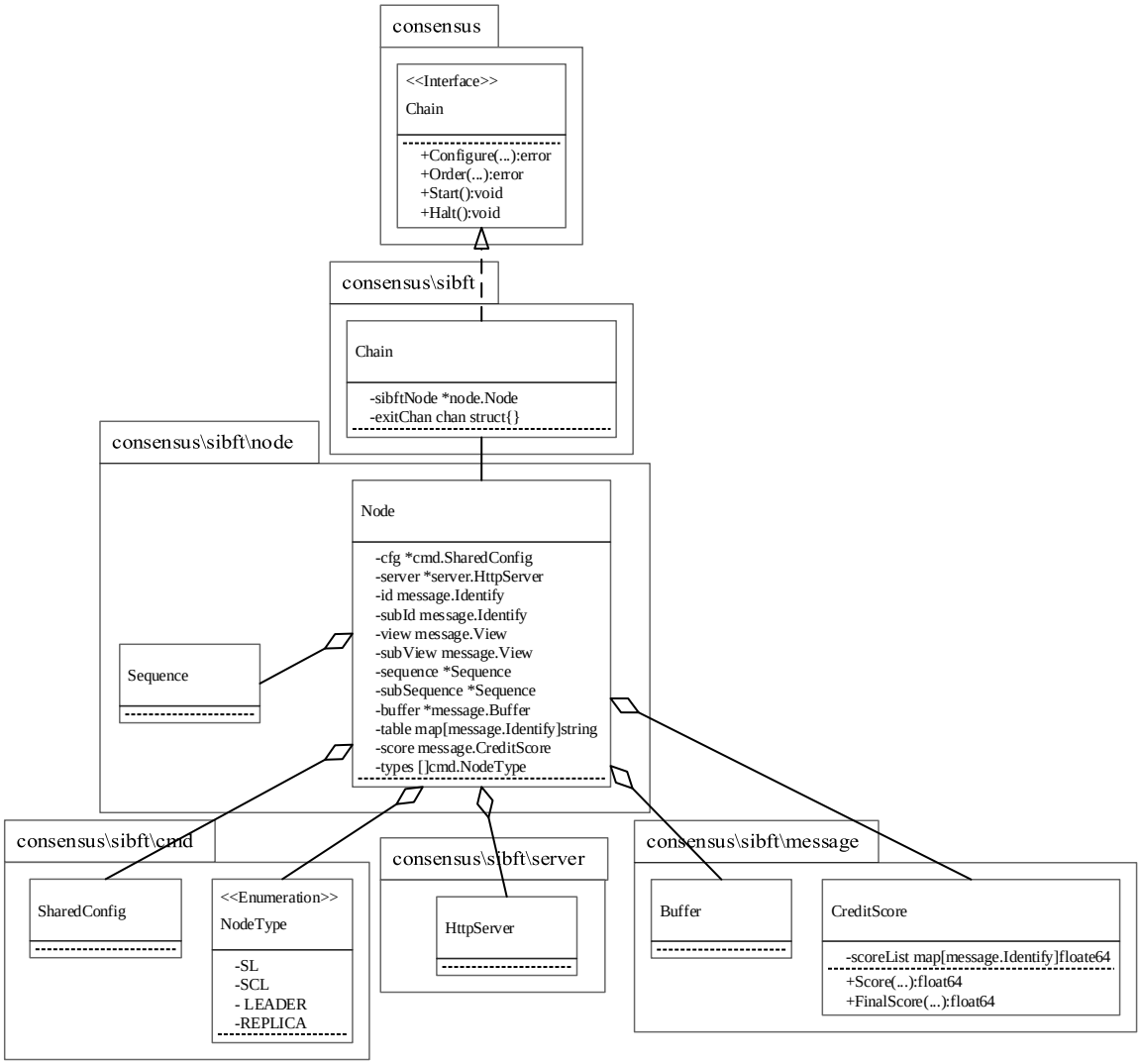

Fig. 3: Design of SIBFT

1) ***SL and SCL election****:* In PBFT consensus algorithm, the consensus node needs to send prepare and commit messages to all other consensus nodes in prepare and commit consensus phase. The complexity of its network communication is $O(n^2)$ [19]. As the number of consensus nodes increases, the complexity of consensus network communication will show a parabolic growth. In order to reduce the complexity of consensus communication in PBFT, SIBFT splits the consensus network and reduces the complexity of network communication to $O(nlogn)$. That is to say, the consortium blockchain network is divided into multiple subnets according to the industry sectors in which the organization is located. Then a SL is elected from the subnet to participate in the consensus of mainnet. The Nodes in the subnet can have three states, namely *candidate*, *leader* and *follower*.

As shown in algorithm 1, first, the nodes in the subnet are initialized to the *follower* state, and each node has only one election vote. After a random delay, if any one of the nodes does not receive the election request from other nodes, it will change state to *candidate*, and initiate a SL election request to other nodes in the subnet and vote for itself immediately. If one SL node receives multiple SL elections at the same time, the SL node will randomly select one to vote. In the case that no node obtains more than half of the votes, all consensus nodes will refresh the random delay and votes, and execute the above mentioned SL election process cyclically until a node obtains more than half of the votes to become SL node.

Taking a three nodes subnet as an example to illustrate SL election process, where the three nodes are called node1, node2, and node3 respectively. After a random delay, node1 receives no election requests from other nodes, and it first sends a SL election request to node2 and node3. Node1 then receives the votes of node2 and node3, and changes its state to *leader* participating in the consensus of mainnet as SL. Node2 and node3 are transformed into the state of *followers*. After the election is completed, node1 synchronizes and maintains

the state of the subnet by sending heartbeats regularly.

In order to eliminate the transaction latency caused by the re-election in subnet when an SL node fails, we set an SCL node in the subnet. The SCL is specified by the SL, and replaces SL node when SL node encounters a network or physical failure. After the SL election, the SL node broadcasts SCL election request in subnet. The first node that responds to the SCL election request will become SCL node. If the SCL node crashed, the SL will re-initiate a request for the SCL election.

Algorithm 1 SL election process

Require:
state: Node state in subnet
total: Total number of nodes in subnet
voteCount: Number of votes obtained
hasVoted: Has it voted
hasLeader: Is there a leader node in subnet
delay: Random delay
electionRequstList: Election request queue from other nodes
nodeNum: The number of node
Begin:
while $hasLeader \neq true$ **do**
  $state \leftarrow follower$
  $hasVoted \leftarrow false$
  $delay \leftarrow random()$
  while $reachDelay() \neq true$ **do**
    $electionRequstList \leftarrow recieveElectionRequst()$
    if $len(electionRequstList) > 0$ **and** $hasVoted \neq true$ **then**
      $voteForRandom(electionRequstList)$
      $hasVoted \leftarrow true$
      $clear(electionRequstList)$
    end if
    if $leaderNodeNum \leftarrow recieveHeartbeat().nodeNum$ **then**
      $hasLeader \leftarrow true$
      return $leaderNodeNum$
    end if
  end while
  $state \leftarrow candidate$
  $votesCount \leftarrow 1$
  $hasVoted \leftarrow true$
  $sendSLElectionRequest()$
  $delay \leftarrow random()$
  while $reachDelay() \neq true$ **and** $recieveVoteRes()$ **do**
    $votesCount++$
    if $voteCount > \frac{total}{2}$ **then**
      $state \leftarrow leader$
      $hasLeader \leftarrow true$
      $sendHeartbeat()$
      return $nodeNum$
    end if
  end while
end while

*2) **Consensus process**:* The consensus process is shown in Fig. 4, where the PSB-SIF has a consortium blockchain composed of 8 organizations. The corresponding types of consensus nodes in Fig. 4 are shown in Table II.

The consensus process consists of five phases. We assume that the total of all SL nodes is *total*. The maximum number of malicious SL nodes that can be tolerated in SIBFT is *f*, which is calculated by equation 1. That is to say, when the number of malicious SL nodes in PSB-SIF exceeds *f*, the consensus will not be reached [20]. The message is *m*. The message digest is *d*. The time leader node broadcasting *pre-prepare* message is *t*. The main network view is *v*. The subnet view is *v-sub*. The main network message sequence number is *n*. The subnet message sequence number is *n-sub*. The mainnet node sequence number is *i*. The subnet node sequence number is *i-sub*. We take Fig. 4 as an example to explain the consensus process.

TABLE II: Corresponding types

Name	Orderer types	
	Main net	Sub net
PoliceOrderer	leader, SL	leader, replica
SupervisionOrderer	SL	leader, replica
FireOrderer1		
ElectricOrderer1		
FireOrderer2		SCL, replica
ElectricOrderer3		
FireOrderer3		replica
ElectricOrderer2		

$$f = \lfloor \frac{total - 1}{3} \rfloor \tag{1}$$

Pre-prepare phase: As the leader, PoliceOrderer forwards the message requested by the client to all SLs, and sends a message <*pre-prepare,v,n,t,d,m*> to request consensus.

Distribute request phase: SL packages the pre-prepare message as <*distribute,*<*pre-prepare,v,n,t,d,m*>*,v-sub,n-sub*> and sends it to replicas in the subnet. After receiving the message, the replica returns a response <*recieved,pre-prepare,v-sub,n-sub,i-sub*> to SL.

Prepare phase: After the SL receives the response from more than half replicas in the subnet, it enters into prepare phase. At this phase, the SL sends a message <*prepare,v,n,i, m*> to other SLs to indicate that it and the replicas in the subnet have received *m*.

Commit phase: When the SL receives $2f$ prepare messages, it enters into commit phase, and then sends a message <*commit,v,n,i*> to other SLs.

Distribute commit phase: When the SL receives $2f + 1$ commit messages, it sends messages <*distribute,*<*commit,v,n, i*>*,v-sub,n-sub*> to replicas in the subnet. Replicas execute *m* after received the message, and returns a response <*received,commit,v-sub,n-sub,i-sub*> to the SL. When the SL receives responses from more than half of the replicas in the subnet, it will reply to the client.

*3) **Credit score mechanism**:* In order to encourage nodes to actively participate in consensus, we design a credit score mechanism in SIBFT. The evaluation of credit score is related to the speed of node *prepare* response and node stability. In order to reduce the consumption of computing resources caused by calculations, the credit score assessment is conducted every 100 rounds of consensus.

First, after each node receives *prepare* message from other nodes, it will calculate the delay $Delay_{i,j}^{r}$ for other nodes in the *r* round. $Delay_{i,j}^{r}$ is the time difference between a node leader in mainnet sending <*pre-prepare,v,r,t,d,m*> message and node *j* receiving <*prepare,v,r,i,m*>. If node *j* does not receive <*prepare,v,r,i,m*> from node *i* before entering the

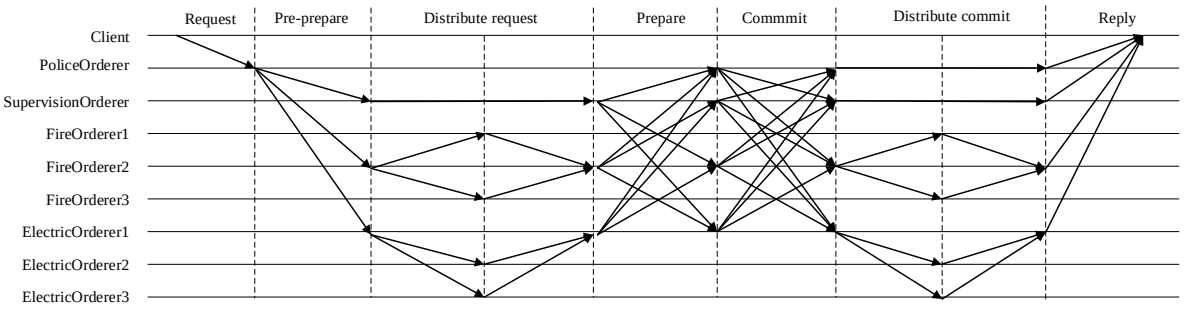

Fig. 4: SIBFT working process

commit phase, set $Delay^r_{i,j}$ to 100*ms*. The definitions of *k* and *r* is as shown in equation 2. The calculations of $Delay^r_{i,j}$ and $DelayMean^k_{i,j}$ is shown in equation 3 and equation 4. $DelayMean^k_{i,j}$ is the average delay for node *j* receiving <*prepare,v,r,i,m*> in the last 100 rounds of consensus.

$$k \in (\mathbb{N}+) \times 100;\ r \in \mathbb{N} \tag{2}$$

$$Delay^r_{i,j} = receivedTime^r_{i,j} - sendTime^r \tag{3}$$

$$DelayMean^k_{i,j} = \frac{\sum_{r=k-100}^{k-1} Delay^r_{i,j}}{100} \tag{4}$$

Secondly, the stability of the node is also an important part of the evaluation. We use $\mu^k_{i,j}$ as a weight to measure the stability of the node. The calculation is shown in equation 5.

$$\mu^k_{i,j} = \frac{1}{1+\sigma^k_{i,j}} \tag{5}$$

where $\sigma^k_{i,j}$ calculated by equation 6 is the standard deviation of $Delay_{i,j}$ in the last 100 rounds.

$$\sigma^k_{i,j} = \sqrt{\frac{\sum_{r=k-100}^{k-1} \left(Delay^r_{i,j} - DelayMean^r_{i,j}\right)^2}{100}} \tag{6}$$

Finally, we comprehensively evaluate the node's credit score through the delay average and standard deviation. If <*prepare,v,n,i,m*> received by node *j* is different from consistent *prepare* messages, node *j* will set $Score^k_{i,j}$ to 0. The evaluation equation 7 is used to calculate the credit score $Score^k_{i,j}$ of node *j* to node *i* in the *k* round.

$$Score^k_{i,j} = \frac{\mu^k_{i,j}}{DelayMean^k_{i,j}} \tag{7}$$

In round *k*, all SL nodes will wrap scores of other nodes in a *prepare* message and send them to leader node. Leader node calculates $FinalScore^k_i$ of each node, then wraps it in a *commit* message and sends it to all SL nodes. The calculation is shown in equation 8.

$$FinalScore^k_i = \sum_{j=1,j\neq i}^{total} Score^k_{i,j} \tag{8}$$

We assume that there are *n* SL nodes in PSB-SIF. If a leader node receives $\lceil \frac{n}{3} \rceil$ messages with a score of 0 for node *j*, then node *j* is judged to be a malicious node. The SCL will periodically inquire other SL nodes, and if it finds that the currently connected SL is a malicious node, it will discard the connection with the current SL node. Then the SCL is upgraded to SL, and other replica nodes in subnet will connect to this new SL.

In addition, if a node is found to be a malicious or faulty node, the credit score of this node will be cleared and excluded from the consensus network. Its scores on other nodes will also be invalid, and we use the method of subtracting malicious scores to express this. Algorithm 2 demonstrates the detailed credit score mechanism.

In PSB-SIF, there can be two authoritative organizations, for examples, PoliceOrg and SupervisionOrg. We assume that these two organizations must be honest, so when there are PoliceOrderer and SupervisionOrderer in the consensus network, these two nodes are preferred as leaders. If both nodes are faulty, we choose the SL with the highest credit score and the highest node number as the leader.

When the leader in the subnet, that is, the SL fails, the SCL replaces it and becomes new leader. Then the new leader will re-initiate a round of SCL campaign request.

C. *Safety inspection smart contract*

Smart contract enables blockchain to handle more complex business processes and save more complex data structures. Smart contract is also called chaincode in HLF. In PSB-SIF, we use chaincode to code business process of safety inspection to replace traditional back-end business procedures. Next, we will first design the overall business process of PSB-SIF, and then take community fire safety inspection as an example to introduce a practical application scenarios.

1) ***PSB-SIF overall execution process****:* A complete business process in PSB-SIF is roughly divided into twelve steps. As shown in Fig. 5, the life cycle of entire business process is from the regular issuance of inspection tasks to the automatic issuance of security certificates.

1) The terminal of the SupervisionOrg in PSB-SIF regularly releases a safety inspection task, and sends a chaincode event "taskIssued" when the chaincode is executed.

Algorithm 2 Credit score mechanism

Require:
round: Consensus round
i,j: Node number
$sendTime^r$: Time when node leader broadcast *pre-prepare* message
$receivedTime^r_{i,j}$: Node *j* receives the *prepare* message time of node *i*
$prepare^r_{i,j}$: The prepare message sent by node *i* to node *j*
$consist^r_j$: Prepare message agreed by node *j*
$total$: Total number of SL
$Score^k_{i,j}$: Node *j*'s score on node *i*
$FinalScore^k_i$: Node *i*'s final score
$maliciousMark_i$: Mark the malicious node, the initial value is 0
$isMalicious_i$: Is node *i* a malicious node
Begin:
/*Node *j*'s evaluation of other nodes*/
while $round \in k$ **do**
  for $i \leftarrow 1$ **to** $total$ **do**
    for $r \leftarrow round - 100$ **to** $round - 1$ **do**
      if $prepare^r_{i,j}$ **then**
        if $prepare^r_{i,j} == consist^r_j$ **then**
          $DelayTime^r_{i,j} \leftarrow Delay()$
        else
          $Score^{round}_{i,j} \leftarrow 0$
          break
        end if
      else
        $DelayTime^r_{i,j} \leftarrow 100$
      end if
    end for
    $Score^{round}_{i,j} \leftarrow Score()$
  end for
end while

/*Leader node*/
while $round \in k$ **do**
  for $i \leftarrow 1$ **to** $total$ **do**
    for $j \leftarrow 1$ **to** $total$ **do**
      if $Score^{round}_{i,j} == 0$ **then**
        $maliciousMark_i++$
      end if
      if $maliciousMark_i \geq \lceil \frac{total}{3} \rceil$ **then**
        $isMalicious_i \leftarrow true$
        $FinalScore^{round}_i \leftarrow 0$
        delte *i* **from total list**
        break
      end if
    end for
    $FinalScore^{round}_i \leftarrow FinalScore()$
  end for
end while
return $FinalScore\ list$

2) After the chaincode event "taskIssued" is monitored by the terminal of the organization in the PSB-SIF, the safety inspection task is pushed to the smart phone of inspectors.
3) The inspector receives inspection task, arrives at the inspection location, and uses the NFC function of the smart phone to authorize the inspection task.
4) The Raspberry Pi and NFC device installed in the safety inspection box obtain the identity information of the inspector.
5) The Raspberry Pi interacts with the chaincode via program built by Software Development Kit (SDK) provided by HLF.
6) Peer nodes return the endorsement results of chaincode to Raspberry Pi.
7) The Raspberry Pi sends the endorsement results to orderer node.
8) Peer nodes pull blocks from orderer nodes for synchronization.
9) The safety inspector checks whether he has been authorized for inspection.
10) After obtaining authorization, the safety inspector fills in inspection records.
11) The safety inspector uploads inspection records by NFC device of smart phone. Then the Raspberry Pi device in the safety inspection box requests the "inspectionRecordStore" method in the chaincode. After waiting for the blockchain network to reach a consensus, the inspection records will be stored on the blockchain.
12) A chaincode event "taskComplete" will be triggered in the chaincode call of step 11). After the event monitoring terminal in SupervisionOrg of PSB-SIF listened "taskComplete" event, it will call the "issueacertification" method in the chaincode to issue a safety certificate to the inspection location.

*2) **Application scenarios**:* As Fig. 6 shows, we take community fire safety inspection as an example to show a practical application scenario of PSB-SIF. The inspection process is divided into six steps, from issuance of inspection task by SupervisionOrg to issuance of safety certificate by SupervisionOrg.

1. The SupervisionOrg issues a task for regular safety inspections of the Foo community.
2. The inspector Bob who received the inspection task carries his smart phone to the 1st floor of Unit 1, Building 1, Foo Community, to perform the inspection task.
3. After arriving at the inspection location, Bob uses a key to open the inspection cabinet installed in the inspection location, and uses the NFC function of his smart phone to interact with the safety inspection box installed in the inspection cabinet to trigger the smart contract installed on the blockchain, so as to obtain an inspection authorization.
4. After obtaining the inspection authorization, Bob starts to check the safety equipment, fills in the inspection record form with the smart phone, adds the damaged fire blanket and fire extinguisher to the damaged equipment record list, and adds the lost fire hammer to the lost equipment list.
5. After the completion of the report, Bob uses the NFC function of the smart phone to interact with the safety inspection box again to upload the inspection record to the blockchain, and at the same time this triggers the chaincode event of "taskComplete". For the lost fire hammer, the "lost" chaincode event is automatically triggered, and the PoliceOrg automatically establishs a theft case after receiving the "lost" chaincode event.
6. After the the SupervisionOrg receives the chaincode event of "taskComplete", it will automatically issue a safety certificate to the inspection location where there is no equipment damage or loss.

III. EVALUATION

In this section, we will evaluate the performance of PSB-SIF using three different consensus algorithms, RAFT, PBFT and

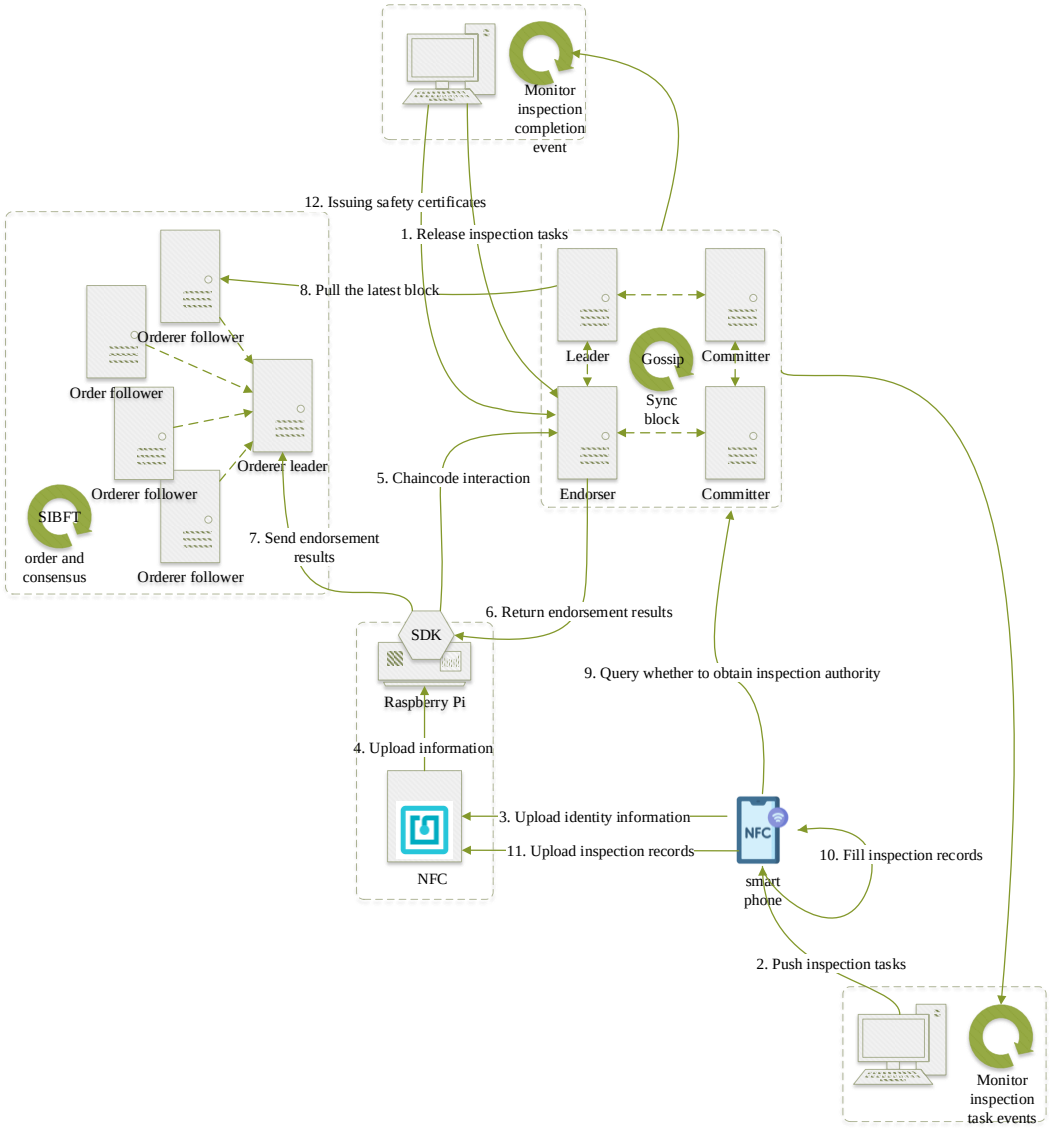

Fig. 5: The overall process of PSB-SIF

SIBFT respectively. We mainly focus on the impact of these three different consensus algorithms on throughput, transaction latency, scalability and security. The evaluation tool used to calculate transaction latency and throughput in this paper is hyperledger-caliper. The experimental environment is shown in Table III.

TABLE III: Running environment

Software	Version
golang	1.13.5
HLF	1.4.4
hyperledger-caliper	0.3.0
docker	18.06.0-ce
docker-compose	1.23.2

In the experiment, caliper is used to create three clients to initiate concurrent requests to the blockchain, and to initiate a call to the ”personnelRegister” method in the safety inspection smart contract for performance testing. Each consensus algorithm is tested with 10 orderer nodes. We conduct 10 experiments for these three consensus algorithms, and each experiment uses a different request sending frequency to test the changes of throughput and transaction latency [21].

A. *Throughput and Transaction Latency*

We built three different blockchain networks based on HLF, using RAFT consensus algorithm, PBFT consensus algorithm and SIBFT consensus algorithm respectively. In order to exclude the influence of other factors on the experiment, we install the same chaincode on these three blockchain networks. The number of peer nodes is 20. Since the research focus of this experiment is comparisons of three different consensus algorithms, we focus on the test of orderer nodes and ignore

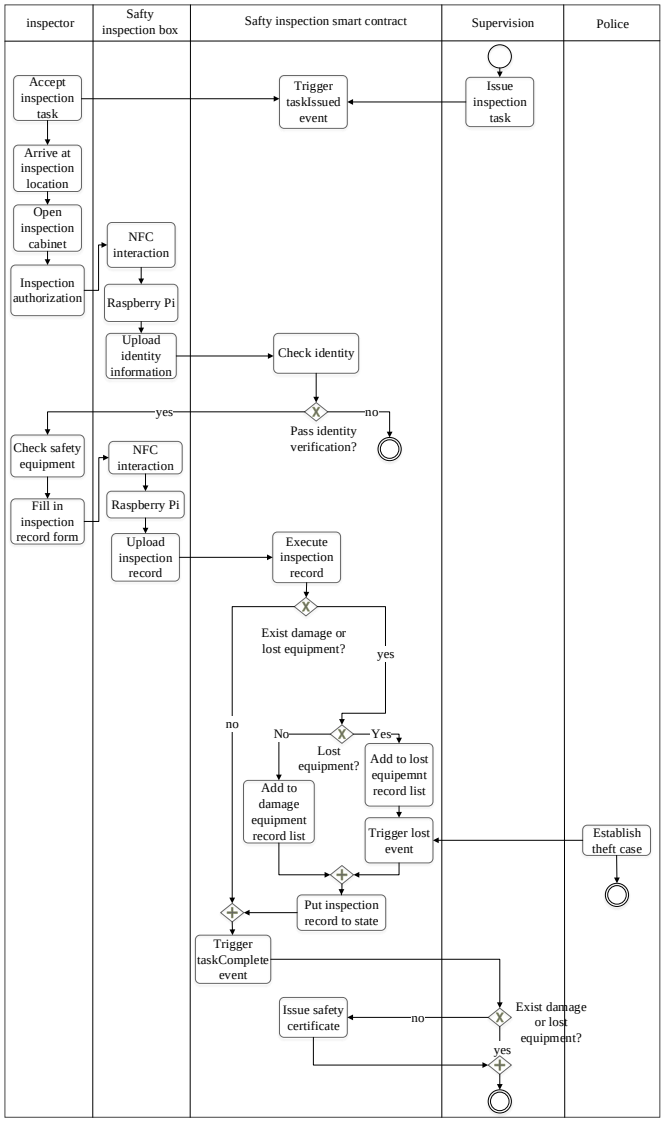

Fig. 6: Application scenario process

the test of the peer nodes. Because only orderer nodes are responsible for consensus in HLF [22].

The results are shown in Fig. 7 and Fig. 8. From Fig. 7, we can see that the throughput of the blockchain network using these three consensus algorithms will first increase as the frequency of transaction sending increases. However, when the maximum throughput is reached, as the frequency of transaction sending increases, its throughput will not change significantly, but fluctuates around the highest throughput, tending to a stable value. The maximum throughput of the blockchain network using the SIBFT consensus algorithm is 18.1tx/s. Although this is slightly less than the maximum throughput of the blockchain network using the RAFT consensus algorithm (20.6tx/s), it is much greater than the maximum throughput of the blockchain network using the PBFT consensus algorithm (12.4tx/s).

The transaction latency is also one of the important indicators for evaluating the performance of blockchain consensus

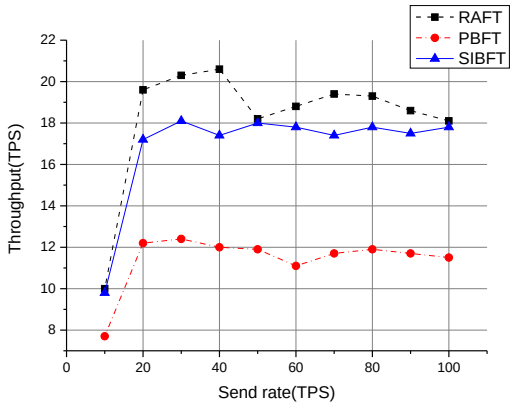

Fig. 7: Throughput of three kinds of consensus algorithm

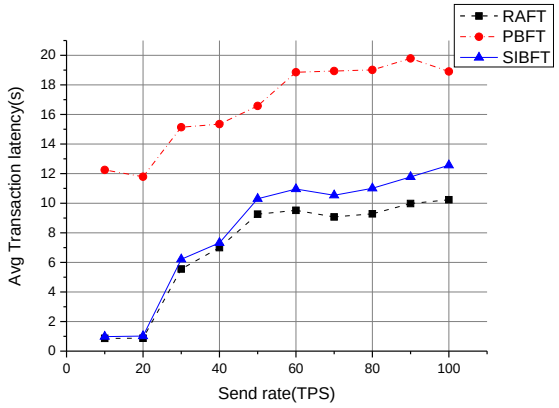

Fig. 8: Avg Transaction Latency of three kinds of consensus algorithm

algorithms. As shown in Fig. 8, the average transaction latency of blockchain network using these three consensus algorithms shows an overall upward trend as the frequency of transaction sending increases. Especially when the transaction request frequency reaches maximum throughput, the average transaction latency will increase significantly. This is because when request frequency exceeds the maximum throughput, it will lead to the accumulation of requests. The average transaction latency of the blockchain network using the SIBFT consensus algorithm is slightly higher than using the RAFT consensus algorithm, but its transaction latency is much smaller than that using the PBFT consensus algorithm.

B. Scalability test

Scalability in this paper refers to the impact of increasing number of consensus nodes on the throughput of blockchain network. That is to say, the smaller the impact of increase in the number of consensus nodes on the throughput of consensus network, the consensus algorithm is considered to have satisfactory scalability. In order to eliminate the influence of peer nodes on the test, we only set up a peer node for the entrance of the throughput test. The test results are shown in Table IV. We test from the initial consensus network of 4

consensus nodes, adding 3 consensus nodes each time until 30 consensus nodes are reached.

Since different consensus algorithms have different classifications of consensus nodes, the way to add consensus nodes is also different. For the RAFT and PBFT consensus algorithms, we set one node as the leader node, and subsequent nodes added are all follower nodes. For the SIBFT consensus algorithm, the initial four consensus nodes are all set as SL consensus nodes, and two of them are special nodes, namely "PoliceOrderer" and "SupervisionOrderer". In order to reduce the impact on the consensus nodes in subnet when SL node fails or becomes a malicious node, we set the number of consensus nodes in subnet to at most 3. The detailed adding method is shown in Table IV.

The detailed results are shown in Fig. 9. With the increase in the number of consensus nodes, due to the increase in network communication, the maximum throughput of blockchain network using these three consensus algorithms shows a downward trend. When the number of consensus nodes increases to 16, the maximum throughput of blockchain network using the PBFT consensus algorithm has dropped significantly. At this time, failed transactions begin to appear. When the consensus node reaches 25, due to the complexity of blockchain network increases rapidly, the maximum throughput is close to 0, and the network is in a state of paralysis. The blockchain network using the SIBFT consensus algorithm is similar to the blockchain network using the RAFT consensus algorithm. As the number of consensus nodes increases, the maximum throughput of the blockchain network shows a slow and steady downward trend.

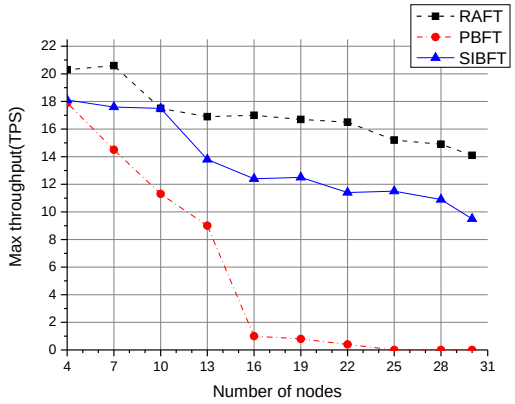

Fig. 9: Scalability test

C. *Credit scoring mechanism*

We use PSB-SIF of 20 consensus nodes to test credit scoring mechanism, in which the number of SL nodes is set to 10, and 1000 rounds of consensus will be conducted. According to equation 1, the maximum number of malicious SL nodes that can be tolerated in SIBFT in this experiment is 3. SL7, SL8, SL9, and SL10 are set as malicious nodes in the experiment. They launched attacks (sending wrong messages) on other nodes in different consensus rounds. The specific attack process is as follows. In the 150th round, SL8 launches malicious attacks on SL2 and SL5; SL9 launches malicious attacks on SL2 and SL3; SL10 launches a malicious attack on SL3. In the 250th round, SL10 launches a malicious attack on SL2. In the 350th round, SL8 launches malicious attacks on SL3 and SL6; SL9 launches a malicious attack on SL5; SL10 launches a malicious attack on SL4. In the 450th round, SL7 launches malicious attacks on SL2, SL3, and SL4; SL9 launches a malicious attacke on SL6; SL10 launches a malicious attack on SL5. In the 550th round, SL7 launches a malicious attack on SL5. The changes of credit score are shown in Fig. 10.

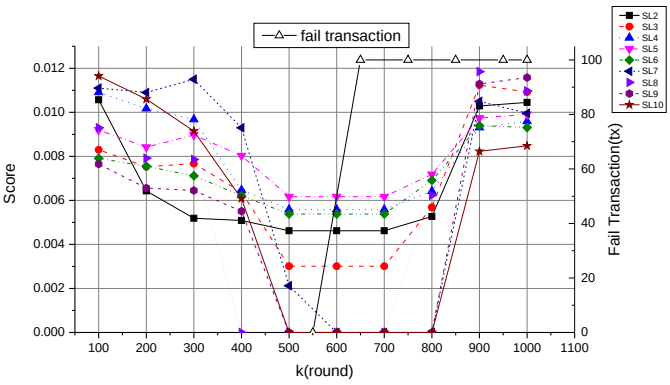

Fig. 10: Credit score comparison

From Fig. 10, we can see that when a node is maliciously attacked, its credit score will show a downward trend. This is because the malicious node makes a dishonest evaluation of it. SL8's credit score drop to 0 in the 400th round. This is because SL8 has already attacked SL5, SL2, SL3, and SL6 in the 350th round and is judged to be a malicious node. Similarly, in the 450th round, SL9 and SL10 have already attacked 4 nodes, so they are also judged as malicious nodes. Their credit scores are reduced to 0 in the 500th round.

The credit scores of the 600th and the 700th rounds are the same as the 500th round. This is because in the 550th round, SL7 has attacked 4 SL nodes and is judged as malicious nodes. At this time, there are 4 malicious SL nodes in the consensus network, namely SL7, SL8, SL9, and SL10, which have exceeded the number of malicious SL nodes that SIBFT can tolerate. As a result, consensus cannot be reached and failed transactions started.

In the 650th round, SCL8 is the first to discover that SL8 is a malicious node, and it abandoned connection with SL8, upgraded to SL8 to participate in the consensus of main network. At this time, the number of malicious SL nodes in the consensus network is reduced to 3, and consensus process is restored.

In the 750th round, SCL7, SCL9, and SCL10 successively discover that SL7, SL9, and SL10 are malicious nodes. They abandoned connection with SL7, SL9 and SL10, upgraded to SL7, SL9, SL10 to participate in the consensus of main network.

In the 800th round, except for the SL7, SL9 and SL10 nodes, the credit scores of other nodes all increase to a certain

TABLE IV: Scalability test of three kinds of consensus algorithm

Number of Nodes	RAFT			PBFT			SIBFT		
	Leader	Follower	Max Throughput(TPS)	Leader	Follower	Max Throughput(TPS)	SL	Replica-SL	Max Throughput(TPS)
4	1	3	20.3	1	3	17.9	4	0	18.1
7	1	6	20.6	1	6	14.5	4	3	17.6
10	1	9	17.5	1	9	11.3	4	6	17.5
13	1	12	16.9	1	12	9	7	6	13.8
16	1	15	17	1	15	1	7	9	12.4
19	1	18	16.7	1	18	0.8	7	12	12.5
22	1	21	16.5	1	21	0.4	10	12	11.4
25	1	24	15.2	1	24	0	10	15	11.5
28	1	27	14.9	1	27	0	10	18	10.9
30	1	29	14.1	1	29	0	12	18	9.5

extent. This is because SL8 participate in the credit score statistics from the 700th round consensus. In the 900th and 1000th rounds, the credit scores of all nodes returned to normal levels. This is because SL7, SL9 and SL10 restarted credit statistics from the 800th round of consensus.

D. *Discussion*

In the experiment to measure influence of consensus algorithm on throughput and transaction latency of blockchain network, we built three different HLF block'chain networks based on RAFT, PBFT, and SIBFT consensus algorithm. Experimental results show that the performance of SIBFT has been significantly improved compared to PBFT. The throughput is increased by about 47%, and the average transaction latency is reduced by about 53%. Although it is still slightly slower than RAFT consensus algorithm, it has higher security than the RAFT consensus algorithm.

One of the important factors affecting blockchain consensus algorithm is the complexity of the consensus network. The network complexity of PBFT consensus algorithm is $O(n^2)$. The SIBFT consensus algorithm is based on PBFT consensus algorithm to optimize the complexity of the network and reduces it to $O\left(nlogn\right)$. SIBFT is a consensus algorithm specially designed for safety inspection scenarios. If it is applied to a production environment in a city, the number of consensus nodes will not exceed 30. With the increase of consensus nodes, the complexity of the blockchain network using SIBFT consensus algorithm will not increase dramatically.

In the scalability evaluation experiment, we found that the scalability of blockchain network using PBFT consensus algorithm is poor. This is because that with the increase of consensus nodes, the network traffic will greatly increases. It is difficult to meet the demand for increasing consensus nodes as the business grows. The blockchain network using SIBFT consensus algorithm has good scalability, similar with the RAFT consensus algorithm.

Common attack methods in blockchain include double-spending attack [23] and sybil attack [24]. Because PSB-SIF is a blockchain framework without tokens, there is no double-spending attack problem with SIBFT. HLF adopts a permission control method for the joining of consensus nodes. The identity of the node needs to be verified when making consensus, so it cannot pretend to be other nodes to carry out sybil attack. So this paper only discusses the case of malicious nodes sending error messages [25]. The discovery of malicious nodes is very important for the security of consensus network. The credit scoring mechanism proposed by SIBFT consensus algorithm in this paper not only reduces the time consumed when switching views in blockchain networks, but also has a positive effect on the discovery of malicious nodes. Because the consensus nodes in the subnet do not directly participate in the consensus of mainnet, the SCL in subnet needs to periodically inquire other SL nodes in the mainnet to discover malicious SLs.

IV. RELATED WORK

A. *Existing safety inspection solutions*

An important reason for safety accidents is that people do not pay enough attention to safety as the impact of safety on life is ambiguous in people's cognition [26]. Currently, deep learning based approach can be used to detect safety issues, e.g. KHAN et al. [27] proposed a cost-effective fire video detection architecture. However, this method has false reports on fire accidents. Han et al. [28] used Building Information Modeling (BIM) to enable fire inspection personnel to quickly obtain the required information and store inspection data in a cloud database to facilitate equipment inspection and maintenance. Ensuring the correctness of data of all these solutions is a problem.

Khana et al. [29] used optical character recognition technology to monitor fire equipment and convert image information into texts. Then it use the sha256 encryption method to hash the text information, and store the generated text hash value in a blockchain. In this paper, the details of all inspection information is stored in the blockchain for easy supervision and query, and we also improve the blockchain to cater for safety inspection scenarios.

Collecting data from IoT devices and ensuring the quality of the data has always been a challenging problem. Ozyilmaz KR [30] et al. designed a blockchain-based IoT infrastructure. Smart contracts of Ethereum are used to replace traditional back-end programs. This paper uses the chaincode in HLF to

replace the traditional back-end program, and stores the latest state of the safety inspection data in the distributed Couchdb database. Consortium blockchain technology is used to share between consortiums only, which ensure data privacy in a more robust way, using the improved security mechanism as proposed.

B. Blockchain data source

To make sure that data send to block is authentic. Kamilaris et al. [31] proposed a three-tier food supply chain system based on blockchain technology, which is composed of logistics layer, digital flow layer and blockchain network layer. NFC is used to obtain each historical data generated in the food supply chain. And these data is recorded in blockchain network to ensure the data can not be tampered. Using NFC as an intermediate device to connect network world and physical world is a convenient and efficient method, which can be used for anti-counterfeiting and other applications [32].

There are two problems in using NFC related technology to prevent counterfeit products. The first one is to use modified physical tag, the other one is to reuse the tag for different target. Alzahrani et al. [33] proposed a robust and anti-counterfeiting supply chain by combining blockchain technology with NFC technology, that can effectively prevent malicious tampering and reuse of tags. The chip fingerprint of the NFC card proposed in this paper has a one-to-one correspondence with the identity of the inspector, forming a unique identification of the person's identity. The use of blockchain for distributed storage also achieves the anti-counterfeiting information from inspectors.

C. Performance and security balance for blockchain

One of the most prominent problems of the blockchain is the balance between performance and security [34]. In order to solve the problems of security and performance limitations in computationally scalable Byzantine consensus protocol (SCP), Huang et al. [35] improved the SCP consensus algorithm with a multi-partition consensus model and proposed an improved SCP consensus algorithm (ISCP). At the same time, an algorithm for consensus within the committee is proposed. Experiments prove that ISCP effectively improves throughput and security. In SIBFT, we make use the idea of using proxies, which is similar to the committee rules in ISCP. The difference is that SIBFT is a consensus algorithm designed for consortium blockchain.

Reducing the risk of the credibility of blockchain consensus algorithm is important. Assigning a certain reputation value to the node can effectively reduce the cost of consensus and improve the efficiency of consensus [36]. Eric et al [37] proposed a Proof of X-repute (PoX) consensus algorithm based on node reputations. The reputation value of a node is determined by evaluating the behavior of that node. A node with a higher reputation value is more likely to produce a block. Experiments show that this method can make the blockchain network faster and safer to reach a consensus. This method is similar to our credit scoring mechanism. The difference is that in SIBFT, the node with the higher credit score is more likely to become the leader node in the consensus network.

V. CONCLUSIONS AND FUTURE WORK

Consortium blockchain can be beneficial for safety inspection to ensure data security. However, the current consortium blockchain technology is difficult to balance performance and security, and the authenticity of the data source is difficult to be guaranteed. This paper proposes a performance security balanced approach, and design a safety inspection framework PSB-SIF to solve these issues. In order to ensure the authenticity of the identity of the safety inspector, a blockchain safety inspection box composed of NFC identification and a Raspberry Pi is designed. We also design a consensus algorithm SIBFT for PSB-SIF catering for safety inspection scenarios.

We have evaluated PSB-SIF using the RAFT, PBFT and SIBFT consensus algorithms. The evaluation includes throughput, transaction latency, scalability and security. Experiments show that the blockchain network using the SIBFT consensus algorithm has higher performance than PBFT and higher security than RAFT. Balancing performance and security to cater for safety inspection scenarios, the proposed solution can meet higher security without losing too much performance. The credit scoring mechanism also has a significant effect on the detection of malicious nodes.

In the future, we will design more advanced algorithms to realize dynamic switching between consensus node types, and dynamically adjust the complexity of consensus network according to actual security and performance requirements.

VI. ACKNOWLEDGMENTS

The research is supported by the National Natural Science Foundation of China (62072469), National Key R&D Program (2018yfe0116700), Shandong Natural Science Foundation (ZR2019MF049,ZR2020MF006), basic research fund of Central University (2015020031), West Coast artificial intelligence technology innovation center (2019-1-5, 2019-1-6), and the Opening Project of Shanghai Trusted Industrial Control Platform (TICPSH202003015-ZC).

REFERENCES

[1] S. Nanda, H. Joshi, and S. Khairnar, "An iot based smart system for accident prevention and detection," in *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*. IEEE, 2018, pp. 1–6.

[2] Z. Li, Q. Lu, S. Chen, Y. Liu, and X. Xu, "A landscape of cryptocurrencies," in *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 2019, pp. 165–166.

[3] Q. Lu, X. Xu, H. M. N. D. Bandara, S. Chen, and L. Zhu, "Design patterns for blockchain-based payment applications," 2021.

[4] Q. Lu and X. Xu, "Adaptable blockchain-based systems: A case study for product traceability," *IEEE Software*, vol. 34, no. 6, pp. 21–27, 2017.

[5] M. Du, Q. Chen, and X. Ma, "Mbft: A new consensus algorithm for consortium blockchain," *IEEE Access*, vol. 8, pp. 87 665–87 675, 2020.

[6] S. Hasavari and Y. T. Song, "A secure and scalable data source for emergency medical care using blockchain technology," in *2019 IEEE 17th International Conference on Software Engineering Research, Management and Applications (SERA)*. IEEE, 2019, pp. 71–75.

[7] M. Liu, K. Wu, and J. J. Xu, "How will blockchain technology impact auditing and accounting: Permissionless versus permissioned blockchain," *Current Issues in Auditing*, vol. 13, no. 2, pp. A19–A29, 2019.

[8] B. Cao, Z. Zhang, D. Feng, S. Zhang, L. Zhang, M. Peng, and Y. Li, "Performance analysis and comparison of pow, pos and dag based blockchains," *Digital Communications and Networks*, vol. 6, no. 4, pp. 480–485, 2020.

[9] Y. Li, B. Cao, M. Peng, L. Zhang, L. Zhang, D. Feng, and J. Yu, "Direct acyclic graph-based ledger for internet of things: Performance and security analysis," *IEEE/ACM Transactions on Networking*, vol. 28, no. 4, pp. 1643–1656, 2020.

[10] B. Cao, Y. Li, L. Zhang, L. Zhang, S. Mumtaz, Z. Zhou, and M. Peng, "When internet of things meets blockchain: Challenges in distributed consensus," *IEEE Network*, vol. 33, no. 6, pp. 133–139, 2019.

[11] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *2017 IEEE international congress on big data (BigData congress)*. IEEE, 2017, pp. 557–564.

[12] S. Biswas, K. Sharif, F. Li, S. Maharjan, S. P. Mohanty, and Y. Wang, "Pobt: A lightweight consensus algorithm for scalable iot business blockchain," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 2343–2355, 2019.

[13] E. Iosif, K. Christodoulou, M. Touloupou, and A. Inglezakis, "Leadership uniformity in raft consensus algorithm," in *European, Mediterranean, and Middle Eastern Conference on Information Systems*. Springer, 2020, pp. 125–136.

[14] S. Brotsis, N. Kolokotronis, K. Limniotis, G. Bendiab, and S. Shiaeles, "On the security and privacy of hyperledger fabric: Challenges and open issues," in *2020 IEEE World Congress on Services (SERVICES)*. IEEE, 2020, pp. 197–204.

[15] Z. Qiu, J. Hao, Y. Guo, and Y. Zhang, "Dual vote confirmation based consensus design for blockchain integrated iot," in *NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2020, pp. 1–7.

[16] X. Wang, Q. Feng, and J. Chai, "The research of consortium blockchain dynamic consensus based on data transaction evaluation," in *2018 11th International Symposium on Computational Intelligence and Design (ISCID)*, vol. 2. IEEE, 2018, pp. 214–217.

[17] K. Li, H. Li, H. Hou, K. Li, and Y. Chen, "Proof of vote: A high-performance consensus protocol based on vote mechanism & consortium blockchain," in *2017 IEEE 19th International Conference on High Performance Computing and Communications; IEEE 15th International Conference on Smart City; IEEE 3rd International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*. IEEE, 2017, pp. 466–473.

[18] J.-H. Huh and K. Seo, "Blockchain-based mobile fingerprint verification and automatic log-in platform for future computing," *The Journal of Supercomputing*, vol. 75, no. 6, pp. 3123–3139, 2019.

[19] V. Gramoli, "From blockchain consensus back to byzantine consensus," *Future Generation Computer Systems*, vol. 107, pp. 760–769, 2020.

[20] M. Castro, B. Liskov *et al.*, "Practical byzantine fault tolerance," in *OSDI*, vol. 99, no. 1999, 1999, pp. 173–186.

[21] M. Kuzlu, M. Pipattanasomporn, L. Gurses, and S. Rahman, "Performance analysis of a hyperledger fabric blockchain framework: throughput, latency and scalability," in *2019 IEEE international conference on blockchain (Blockchain)*. IEEE, 2019, pp. 536–540.

[22] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich *et al.*, "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proceedings of the thirteenth EuroSys conference*, 2018, pp. 1–15.

[23] J. Jang and H.-N. Lee, "Profitable double-spending attacks," *Applied Sciences*, vol. 10, no. 23, p. 8477, 2020.

[24] S. Zhang and J.-H. Lee, "Double-spending with a sybil attack in the bitcoin decentralized network," *IEEE transactions on Industrial Informatics*, vol. 15, no. 10, pp. 5715–5722, 2019.

[25] W. She, Q. Liu, Z. Tian, J.-S. Chen, B. Wang, and W. Liu, "Blockchain trust model for malicious node detection in wireless sensor networks," *IEEE Access*, vol. 7, pp. 38 947–38 956, 2019.

[26] J. G. Friesinger, A. Topor, T. D. Bøe, and I. B. Larsen, "The ambiguous influences of fire safety on people with mental health problems in supported housing," *Palgrave Communications*, vol. 5, no. 1, pp. 1–9, 2019.

[27] K. Muhammad, J. Ahmad, I. Mehmood, S. Rho, and S. W. Baik, "Convolutional neural networks based fire detection in surveillance videos," *IEEE Access*, vol. 6, pp. 18 174–18 183, 2018.

[28] H.-y. Han, Z.-g. Hua, and X.-x. Ding, "Design of simulation training system for fire safety inspection based on computer simulation technology," *Procedia engineering*, vol. 211, pp. 199–204, 2018.

[29] N. Khana, D. Leea, A. K. Alia, and C. Parka, "Artificial intelligence and blockchain-based inspection data recording system for portable firefighting equipment," in *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, vol. 37. IAARC Publications, 2020, pp. 941–947.

[30] K. R. Ozyilmaz and A. Yurdakul, "Designing a blockchain-based iot with ethereum, swarm, and lora: the software solution to create high availability with minimal security risks," *IEEE Consumer Electronics Magazine*, vol. 8, no. 2, pp. 28–34, 2019.

[31] A. Kamilaris, A. Fonts, and F. X. Prenafeta-Boldύ, "The rise of blockchain technology in agriculture and food supply chains," *Trends in Food Science & Technology*, vol. 91, pp. 640–652, 2019.

[32] N. K. Singh, "Near-field communication (nfc)," *Information Technology and Libraries*, vol. 39, no. 2, 2020.

[33] N. N. Ahamed, P. Karthikeyan, S. Anandaraj, and R. Vignesh, "Sea food supply chain management using blockchain," in *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*. IEEE, 2020, pp. 473–476.

[34] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, "A survey on the security of blockchain systems," *Future Generation Computer Systems*, vol. 107, pp. 841–853, 2020.

[35] Z.-C. Li, J.-H. Huang, D.-Q. Gao, Y.-H. Jiang, and L. Fan, "Iscp: An improved blockchain consensus protocol." *IJ Network Security*, vol. 21, no. 3, pp. 359–367, 2019.

[36] S. Guo, Y. Qi, Y. Jin, W. Li, X. Qiu, and L. Meng, "Endogenous trusted drl-based service function chain orchestration for iot," *IEEE Transactions on Computers*, 2021.

[37] E. K. Wang, R. Sun, C.-M. Chen, Z. Liang, S. Kumari, and M. K. Khan, "Proof of x-repute blockchain consensus protocol for iot systems," *Computers & Security*, vol. 95, p. 101871, 2020.