

Blockchain and Smart Contracts Towards Decentralized Applications in Industry 4.0

Lead Guest Editor: Jiewu Leng

Guest Editors: Shaokun Fan and J. Leon Zhao





Blockchain and Smart Contracts Towards Decentralized Applications in Industry 4.0

Security and Communication Networks

**Blockchain and Smart Contracts
Towards Decentralized Applications in
Industry 4.0**

Lead Guest Editor: Jiewu Leng

Guest Editors: Shaokun Fan and J. Leon Zhao







Copyright © 2022 Hindawi Limited. All rights reserved.

This is a special issue published in "Security and Communication Networks." All articles are open access articles distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Chief Editor

Roberto Di Pietro, Saudi Arabia

Associate Editors

Jiankun Hu , Australia
Emanuele Maiorana , Italy
David Megias , Spain
Zheng Yan , China

Academic Editors

Saed Saleh Al Rabae , United Arab Emirates
Shadab Alam, Saudi Arabia
Goutham Reddy Alavalapati , USA
Jehad Ali , Republic of Korea
Jehad Ali, Saint Vincent and the Grenadines
Benjamin Aziz , United Kingdom
Taimur Bakhshi , United Kingdom
Spiridon Bakiras , Qatar
Musa Balta, Turkey
Jin Wook Byun , Republic of Korea
Bruno Carpentieri , Italy
Luigi Catuogno , Italy
Ricardo Chaves , Portugal
Chien-Ming Chen , China
Tom Chen , United Kingdom
Stelvio Cimato , Italy
Vincenzo Conti , Italy
Luigi Coppolino , Italy
Salvatore D'Antonio , Italy
Juhriyansyah Dalle, Indonesia
Alfredo De Santis, Italy
Angel M. Del Rey , Spain
Roberto Di Pietro , France
Wenxiu Ding , China
Nicola Dragoni , Denmark
Wei Feng , China
Carmen Fernandez-Gago, Spain
AnMin Fu , China
Clemente Galdi , Italy
Dimitrios Geneiatakis , Italy
Muhammad A. Gondal , Oman
Francesco Gringoli , Italy
Biao Han , China
Jinguang Han , China
Khizar Hayat, Oman
Azeem Irshad, Pakistan





M.A. Jabbar , India
Minho Jo , Republic of Korea
Arijit Karati , Taiwan
ASM Kayes , Australia
Farrukh Aslam Khan , Saudi Arabia
Fazlullah Khan , Pakistan
Kiseon Kim , Republic of Korea
Mehmet Zeki Konyar, Turkey
Sanjeev Kumar, USA
Hyun Kwon, Republic of Korea
Maryline Laurent , France
Jegatha Deborah Lazarus , India
Huaizhi Li , USA
Jiguo Li , China
Xueqin Liang, Finland
Zhe Liu, Canada
Guangchi Liu , USA
Flavio Lombardi , Italy
Yang Lu, China
Vincente Martin, Spain
Weizhi Meng , Denmark
Andrea Michienzi , Italy
Laura Mongioi , Italy
Raul Monroy , Mexico
Naghme Moradpoor , United Kingdom
Leonardo Mostarda , Italy
Mohamed Nassar , Lebanon
Qiang Ni, United Kingdom
Mahmood Niazi , Saudi Arabia
Vincent O. Nyangaresi, Kenya
Lu Ou , China
Hyun-A Park, Republic of Korea
A. Peinado , Spain
Gerardo Pelosi , Italy
Gregorio Martinez Perez , Spain
Pedro Peris-Lopez , Spain
Carla Ràfols, Germany
Francesco Regazzoni, Switzerland
Abdalhossein Rezai , Iran
Helena Rifà-Pous , Spain
Arun Kumar Sangaiah, India
Nadeem Sarwar, Pakistan
Neetesh Saxena, United Kingdom
Savio Sciancalepore , The Netherlands

De Rosal Ignatius Moses Setiadi ,
Indonesia
Wenbo Shi, China
Ghanshyam Singh , South Africa
Vasco Soares, Portugal
Salvatore Sorce , Italy
Abdulhamit Subasi, Saudi Arabia
Zhiyuan Tan , United Kingdom
Keke Tang , China
Je Sen Teh , Australia
Bohui Wang, China
Guojun Wang, China
Jinwei Wang , China
Qichun Wang , China
Hu Xiong , China
Chang Xu , China
Xuehu Yan , China
Anjia Yang , China
Jiachen Yang , China
Yu Yao , China
Yinghui Ye, China
Kuo-Hui Yeh , Taiwan
Yong Yu , China
Xiaohui Yuan , USA
Sherali Zeadally, USA
Leo Y. Zhang, Australia
Tao Zhang, China
Youwen Zhu , China
Zhengyu Zhu , China

Contents

Blockchain-Aided Searchable Encryption-Based Two-Way Attribute Access Control Research
Zhigang Xu , Shiguang Zhang, Hongmu Han , Xinhua Dong, Zhiqiang Zheng, Haitao Wang, and Wenlong Tian 

Research Article (13 pages), Article ID 2410455, Volume 2022 (2022)

Service-Oriented Modeling for Blockchain-Enabled Supply Chain Quality Information Systems
Yani Shi , Jiji Ying , Dongying Shi , and Jiaqi Yan 




Research Article (16 pages), Article ID 1987933, Volume 2022 (2022)

GPBFT: A Practical Byzantine Fault-Tolerant Consensus Algorithm Based on Dual Administrator Short Group Signatures

Xiaosheng Yu , Jie Qin , and Peng Chen 



Research Article (11 pages), Article ID 8311821, Volume 2022 (2022)

CR-BA: Public Key Infrastructure Certificate Revocation Scheme Based on Blockchain and Accumulator

Jingxue Xie , Xinghong Tan , and Liang Tan 





Research Article (15 pages), Article ID 2069195, Volume 2022 (2022)

A Blockchain-Based Framework for Developing Traceability Applications towards Sustainable Agriculture in Vietnam

Duc-Hiep Nguyen, Nguyen Huynh-Tuong , and Hoang-Anh Pham 



Research Article (10 pages), Article ID 1834873, Volume 2022 (2022)

Selection Strategy of Mining Pool under Various Different Payment Mechanisms

Tan Xing-Hong , Fu Lu-Xia , Zhang Zhuang , and Tan Liang 







Research Article (10 pages), Article ID 8265140, Volume 2022 (2022)

Design of a Blockchain-Based Traceability System with a Privacy-Preserving Scheme of Zero-Knowledge Proof

Yudai Xue  and Jinsong Wang 






Research Article (12 pages), Article ID 5842371, Volume 2022 (2022)

BCFDPS: A Blockchain-Based Click Fraud Detection and Prevention Scheme for Online Advertising

Qiuyun Lyu , Hao Li , Renjie Zhou , Jilin Zhang , Nailiang Zhao , and Yan Liu 





Research Article (20 pages), Article ID 3043489, Volume 2022 (2022)

A Novel Semifragile Consensus Algorithm Based on Credit Space for Consortium Blockchain

Xiaohong Deng , Zhiqiong Luo , Yijie Zou , Kangting Li , and Huiwen Liu 

Research Article (13 pages), Article ID 1955141, Volume 2022 (2022)

A Novel Consensus Algorithm Based on Segmented DAG and BP Neural Network for Consortium Blockchain

Xiaohong Deng , Kangting Li , Zhiqiang Wang , and Huiwen Liu 

Research Article (16 pages), Article ID 1060765, Volume 2022 (2022)

Improved PBFT Algorithm Based on Vague Sets

Guangxia Xu  and Yishuai Wang

Research Article (7 pages), Article ID 6144664, Volume 2022 (2022)

Based on Consortium Blockchain to Design a Credit Verifiable Cross University Course Learning System

Chin-Ling Chen , Tianyi Wang , Woei-Jiunn Tsaur , Wei Weng , Yong-Yuan Deng , and Jianfeng Cui 

Research Article (18 pages), Article ID 8241801, Volume 2021 (2021)

Research Article

Blockchain-Aided Searchable Encryption-Based Two-Way Attribute Access Control Research

Zhigang Xu ¹, Shiguang Zhang,¹ Hongmu Han ¹, Xinhua Dong,¹ Zhiqiang Zheng,² Haitao Wang,² and Wenlong Tian ³

¹School of Computer Science, Hubei University of Hubei University of Technology, Wuhan 430068, China

²Narcotics Control Bureau of Department of Public Security of Guangdong Province, Guangzhou 510050, China

³School of Computer Science and Technology, University of South China, Hengyang 421001, China

Correspondence should be addressed to Zhigang Xu; 20181100@hbut.edu.cn

Received 15 April 2022; Accepted 25 August 2022; Published 29 September 2022

Academic Editor: Jiewu Leng

Copyright © 2022 Zhigang Xu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In the Internet of Things (IoT), data sharing security is important to social security. It is a huge challenge to enable more accurate and secure access to data by authorized users. Blockchain access control schemes are mostly one-way access control, which cannot meet the need for ciphertext search, two-way confirmation of users and data, and secure data transmission. Thus, this paper proposes a blockchain-aided searchable encryption-based two-way attribute access control scheme (STW-ABE). The scheme combines ciphertext attribute access control, key attribute access control, and ciphertext search. In particular, two-way access control meets the requirement of mutual confirmation between users and data. The ciphertext search avoids information leakage during transmission, thus improving overall efficiency and security during data sharing. Moreover, user keys are generated by the coalition blockchain. Besides, the ciphertext search and pre-decryption are outsourced to cloud servers, reducing the computing pressure on users and adapting to the needs of lightweight users in the IoT. Security analysis proves that our scheme is secure under a chosen-plaintext attack and a chosen keyword attack. Simulations show that the cost of encryption and decryption, keyword token generation, and ciphertext search of our scheme are preferable.

1. Introduction

In Industry 4.0, the IoT is commonly used in industrial environments and often requires processing large amounts of data. Due to the limited resources of IoT devices, we often store large amounts of data from IoT devices on cloud servers. However, this outsourced storage approach may cause many privacy and security problems, such as identity leakage, illegal access to private data, and data tampering. The solution to these problems is to store the ciphertext in the cloud server. Symmetric encryption can guarantee data confidentiality but cannot achieve fine-grained access control and secure data sharing.

Attribute access control is an access control mechanism proposed by Sahai and Waters [1] to ensure effective and secure data sharing and fine-grained access. Technically, attribute access control is mainly divided into two types:

ciphertext-policy attribute-based encryption (CP-ABE) [2] and key-policy attribute-based encryption (KP-ABE) [3]. In the CP-ABE scheme, each data user obtains the corresponding attribute secret key from the authorization agency according to their attributes, and the access structure of the file is determined by the data owner. Only when the attribute set in the secret attribute key of the data acquirer meets the access structure of the file can the file be viewed correctly. In the KP-ABE, by contrast, files can only be viewed when the access structure of the identity key satisfies the ciphertext properties. However, these two methods of attribute access control are only a single method of authentication. They address the need for one-way control of data sharing but do not meet the need for two-way confirmation of users and data. For this reason, Attrapadung and Imai [4] proposed a two-policy attribute access control scheme whose core idea is to combine ciphertext access control with key access

control. On the one hand, the ciphertext is obtained by associating the plaintext with the corresponding user access structure and plaintext attributes. On the other hand, the user's private key is computed by associating its attribute set with the ciphertext access control structure. The plaintext can only be decrypted if both the ciphertext access control and the key access control match. However, this solution is a centrally authorized agent prone to a single point of failure. Han et al. [5] proposed a distributed bidirectional attribute access control strategy. However, the scheme does not consider users' security requirements for personal data queries and transmissions in the IoT environment.

Blockchain is increasingly used in non-transactional scenarios such as supply chains, the IoT, smart healthcare, and public security, where data often contain users' private data. The data cannot be fully disclosed to everyone as a transaction and can only be shared to a limited extent. Through blockchain research, the use of blockchain to manage users' keys ensures secure data sharing for the development of the Industrial Internet of Things (IIoT).

In this paper, we propose a blockchain-aided searchable encryption-based two-way attribute access control scheme (STW-ABE) to manage massive IoT data and meet people's demand for data access control of private data. The main contributions of our scheme can be summarized as follows:

- (1) *Blockchain-Aided Key Generation.* Blockchain consensus nodes jointly execute the DKG to generate the secret key. It avoids the problem of secret key leakage caused by a single point of failure.
- (2) *Blockchain-Cloud-Aided Keyword Search.* The combination of attribute encryption technology and searchable encryption achieves fine-grained two-way access control of transaction ciphertexts in the blockchain. The blockchain sends a token containing a single keyword to the CS. The CS uses the token to perform a ciphertext search to avoid leakage of private data during transmission.
- (3) *Cloud-Aided Pre-Decryption.* The CS provides the pre-decryption service for users with access permission, and the user only needs to perform one exponential operation to decrypt the ciphertext. It reduces computational pressure for users and meets the needs of resource-constrained IoT devices.

The rest of this article is organized as follows. Section 2 reports the most related work. Section 3 introduces relevant knowledge, including linear secret-sharing schemes, distributed key generation protocols, searchable encryption, and blockchain technology. Section 4 presents the system definition, including the system model, the STW-ABE scheme, and the security model. In Section 5, we reveal the detailed construction of the STW-ABE scheme. Section 6 analyzes the security of our scheme and compares the time cost with other schemes in encryption, decryption, and ciphertext search. Finally, we conclude in Section 7.

2. Related Work

In Industry 4.0, access control technology is essential to build trust and sustainability in a distributed context of the IoT. Leng et al. [6] proposed a blockchain model with chemical signature access under a distributed context, Makerchain, which binds unique signature data to the blockchain and automatically executes smart contracts set between manufacturers to achieve service trust between manufacturers. Rahman et al. [7] proposed a distributed multi-signature technology based on blockchain to realize multi-party identity authentication and guarantee the trust between multiple parties in the Industry 4.0 system. However, it does not consider the resource limitation of IoT terminal devices. Most data encryption techniques in use today are based on bilinear mapping encryption, which means that the computational cost of decryption is high. Most lightweight devices do not adapt to attribute-based access control. Therefore, many attribute access control schemes propose the method of outsourcing decryption. Li et al. [8] proposed an outsourcing ABE scheme search based on keyword search. However, the search method used in this scheme is a common public key encryption of keywords, which cannot achieve a fine-grained searchable encryption scheme. Ziegler et al. [9] proposed an outsourcing decryption scheme based on a prime order group to bridge the gap between the highly dynamic IIoT environment and resource-constrained devices. The IoT includes a core network and an edge network, and data security problems will be encountered in data sharing. Liu et al. [10] proposed a privacy-protecting multi-keyword searchable encryption scheme in a distributed system. Through a multi-server architecture, authorized servers can jointly search whether the token matches the ciphertext, thus improving the search efficiency. Miao et al. [11] put forward a multi-keyword search scheme based on attributes and transformed attributes into 0 and 1 codes for attribute judgment comparison, thus improving the efficiency of strategy judgment.

In the IIoT, blockchain is a new generation of security technology with immutability and traceability characteristics. Leng et al. [12] discussed how blockchain promotes the sustainable development of manufacturing and product life management in Industry 4.0. Mehta et al. [13] proposed a blockchain-based copyright contract transaction scheme for the Industry 4.0 supply chain, which ensured the security of copyright transactions for different stakeholders in the industry. But the blockchain has its potential security problems. Leng et al. [14] proposed the PDI model and divided blockchain security issues into process level, data level, and infrastructure level. This paper mainly studies data access control to solve the data-level security sharing problem to improve blockchain systems' data security. In Industry 4.0, blockchain provides key technology for the secure intelligent manufacturing of IIoT, but distributed Industry 4.0 needs to realize collaborative trust. Leng et al. [15] put forward eight network security obstacles in the intelligent manufacturing

of blockchain. The cybersecurity barriers include device deception, false authentication, and trust in data sharing among participants. Therefore, implementing blockchain identity authentication in the IIoT is of great significance for the multi-party trust and sustainability of Industry 4.0. Li et al. [16] proposed a multi-keyword encrypted search scheme applicable for blockchain, which implements ciphertext information search and data access control through smart contracts. Before ciphertext search, the smart contract automatically determines data access permission to enhance trust among IoT users. Feng et al. [17] proposed a data privacy protection scheme based on blockchain searchable attribute access control. The user's permission authentication is implemented by the user's local server, avoiding the security risk of submitting the user's private key and access structure to the blockchain network. Gao et al. [18] proposed a trusted secure ciphertext policy and attributed a hiding access control scheme based on blockchain. The scheme hides the ciphertext policy and attribute information and reduces accidental leakage of data information. Therefore, Liu et al. [19] proposed a searchable attribute-based encryption scheme in which a coalition blockchain replaces the traditional centralized server to be responsible for the generation and storage. Qin et al. [20] proposed a lightweight IoT access control scheme based on attribute encryption and blockchain to verify the accuracy of outsourced decrypted data in IoT through a smart contract. In addition, some schemes use the distributed feature of blockchain to distribute secret keys as the authority. Lewko and Waters [21] proposed a multi-authority attribute-based encryption scheme. In this scheme, the secret user key consists of multiple components, each from a different organization, to prevent collusion attacks among users. Qin et al. [22] proposed a blockchain multi-attribute access control scheme for cloud data sharing. Smart contracts on blockchain manage attribute tokens across domains to solve the trust problem between multiple users. Shi et al. [23] proposed a blockchain-based distributed access control scheme for IoT. The solution uses blockchain nodes as the addresses of IoT devices. It uses blockchain to complete the data authorization, cancellation, access control, and auditing process to ensure data security in the distributed IoT system.

The access control scheme mentioned above compensates for the deficiency of the blockchain access control mechanism in the IIoT environment. However, combining the existing access control scheme with blockchain is not enough to meet users' demand for secure sharing access control of private data. Currently, most blockchain access control solutions only implement user access policy settings for the data and do not address the need for two-way policy confirmation between the user and the data. Furthermore, the security of the data during sharing and the usability of users of lightweight devices were not considered. Therefore, this paper proposes a blockchain-aided searchable encryption-based two-way attribute access control scheme (STW-ABE).

3. Preliminaries

3.1. Linear Secret-Sharing Schemes (LSSS). The linear secret-sharing scheme [24] is defined as follows.

Definition 1. Let P be a set of parties. Let M be a $l \times k$ matrix. Let $\rho: \{1, \dots, l\} \rightarrow P$ be a function that maps a row to a party for labelling. Let (M, ρ) represent a linear secret-sharing scheme with access structure A , which usually consists of two polynomial-time algorithms:

- (1) For $x = 1, \dots, l$, the x -th row of matrix M is labeled by a party $\rho(i)$, where $\rho: \{1, \dots, l\} \rightarrow P$ is a function that maps a row to a party for labelling. The algorithm takes as input the secret value $s \in \mathbb{Z}_p$ that is shared. $y_2, \dots, y_k \in \mathbb{Z}_p$ are randomly chosen, and $\vec{v} = (s, y_2, \dots, y_k)^T$. The share $\lambda_{\rho(i)} = M_i \cdot \vec{v}$ belongs to party $\rho(i)$.
- (2) Let $S \in A$ be input. Let $I = \{i | \rho(i) \in S\}$, and randomly select $c_i \in \mathbb{Z}_p^*$. The output is a constant with linear reconstruction characteristics: $\sum_{i \in I} c_i \sigma_i = s$.

3.2. Distributed Key Generation (DKG) Protocol. Traditional key generation is performed by the central server. This centralized management approach is prone to a single point of failure problem. To solve this problem, researchers proposed the distributed key generation (DKG) protocol [25]. In the DKG protocol, the generation of secret values is done by multiple parties, not by an authoritative center, and does not rely on trusted third parties. Multiple nodes jointly generate a secret value α , and each node has a corresponding secret value α to share. The secret value can be restored only when the sharing rule (t, n) is met, where t is the number of nodes authorized by participants and n is the threshold. The secret value generated by t nodes must be shared by at least n participants to complete the sharing of secret value α .

3.3. Searchable Encryption. Song et al. [26] proposed the practical technology of encrypted data search. In this technique, the scheme for searching the encrypted data is described, and the security of the generated encryption system is proved. The third-party server can only obtain the matching ciphertext results if only the ciphertext data are provided. Nevertheless, it cannot obtain the data information in plaintext, which implements query isolation. In addition, a hidden query is supported. Data users only need to send the search token containing the query keyword to the third-party server for ciphertext search without disclosing the detailed information of the keyword to the server.

3.4. Blockchain Technology. Generally, there are three types of blockchain: public blockchain, private blockchain, and coalition blockchain. A public blockchain allows any node to

generate transaction information and view all information in the block. In a private blockchain, all nodes on the network are controlled by a single organization, and only a small number of authorized nodes have access to the data information. In the coalition blockchain, authorized nodes can join the blockchain network and participate in transactions and information synchronization with strong controllability and high privacy.

This paper uses a coalition blockchain. The blockchain is controlled by a group of trusted nodes that control the consensus protocol. Other authorized nodes can generate data and send them to the blockchain for storage. Then, the consensus node runs the consensus protocol to complete the ledger update in the coalition blockchain so that all nodes keep the whole state consistent. In this paper, the specific functions of the coalition blockchain are as follows. (1) The consensus node in the blockchain initializes system parameters using the distributed secret key generation protocol. (2) The consensus node is responsible for generating, storing, and distributing global public keys, public and private key pairs of users, and user identity keys. (3) The consensus node responds to the keyword searched by the user, generates a ciphertext index through the blockchain, and sends it to the cloud server.

4. System Definition

4.1. System Model. The STW-ABE scheme contains four participants presented as follows. The detailed structural components of the scheme are shown in Figure 1.

- (1) *Data Publisher (DP).* Any IoT device can generate data. The plaintext data containing ciphertext attributes and user access structure are encrypted on the local service. Then, the ciphertext and ciphertext index are uploaded to the cloud server. Data publishers can be people and any IoT device.
- (2) *Data Acquirer (DA).* The data acquirer receives the user identity key from the blockchain, which contains the user attributes and the ciphertext access structure. The DA can only capture the ciphertext if the DA attribute meets the user access structure of the DP and the ciphertext attribute meets the ciphertext access structure of the DA. The DA obtains the ciphertext that meets the individual's conditions and decrypts it with its user identity key.
- (3) *Blockchain (BC).* A coalition blockchain comprises trusted consensus nodes. The blockchain is responsible for initializing the global public key and generating users' public and private key pairs, user identity secret keys, and tokens.
- (4) *Cloud Server (CS).* Cloud servers are used to store large amounts of ciphertext and ciphertext indexes that are uploaded by DP. In addition, CS responds to users' search requests, verifies access control permission, provides pre-decryption services for DA who meets the permission, and returns the pre-decrypted intermediate ciphertext to the DA.

The STW-ABE scheme is divided into three parts. The first part is encryption. First, the DP obtains the global public key and users' public and private key pairs from the blockchain. Then, the DP encrypts the plaintext data through the ciphertext attribute set and user access structure. The DP then sends the ciphertext and ciphertext index to CS. The second part is the ciphertext search. DA searches for ciphertext information by keyword. First, DA sends a keyword to the blockchain network. Second, the blockchain network encrypts a keyword into a token and sends the token to CS, which conducts a ciphertext search through the ciphertext index and search tokens. Finally, the retrieved ciphertext is stored. The third part is decryption. The CS verifies the access control permissions of the set of users, that is, whether the user attributes meets the user access structure and whether the ciphertext attribute set meets the ciphertext access structures. The CS provides a pre-decryption service to generate intermediate ciphertext for the DA, satisfying the two-way access structure. When the DA receives the intermediate ciphertext from CS, the DA uses the user identity key to decrypt the intermediate ciphertext into plaintext.

4.2. System Procedure. The composition of the STW-ABE scheme is as follows.

4.2.1. Initialization. $\text{Setup}(\lambda) \rightarrow \text{GP}$. Setup: the process runs on blockchain consensus nodes participating in authorization and outputs global public key GP.

Authority Setup (GP) $\rightarrow \text{SK, PK}$. User public key and private key generation: the process runs in the blockchain consensus nodes, with global public key GP as input, and outputs user public key PK and user private key SK.

4.2.2. User Identity Key Generation. $\text{KeyGen}(\text{GP, SK, PK, UID, K, } (P, \eta)) \rightarrow \text{UK}_{\text{UID}}$. User identity key generation: the process is run consensus nodes in blockchain that execute the distributed key generation protocol, taking the global public key GP, the user public key PK, the user private key SK, the user attributes set K, the ciphertext access structure (P, η) , and user's identity UID as input, and outputs the user identity key UK_{UID} .

4.2.3. Encryption. $\text{Encrypt}(\text{GP, SK, PK, UID, } (F, \rho), \Lambda, M) \rightarrow D, \text{KW}$. Encryption: this process is run by the DP, taking the global public key GP, the user public key PK, the user private key SK, the user's identity UID, the user access structure (F, ρ) , ciphertext attribute set Λ , and plaintext M as input, and outputs ciphertext D and keywords of ciphertext KW.

4.2.4. Index Generation. $\text{IndexGen}(\text{GP, PK, KW}) \rightarrow \text{Index}$. Index generation: this process is run by the DP, with the global public key GP, the user public key PK, and the keywords of ciphertext KW as input, and outputs the ciphertext index Index.

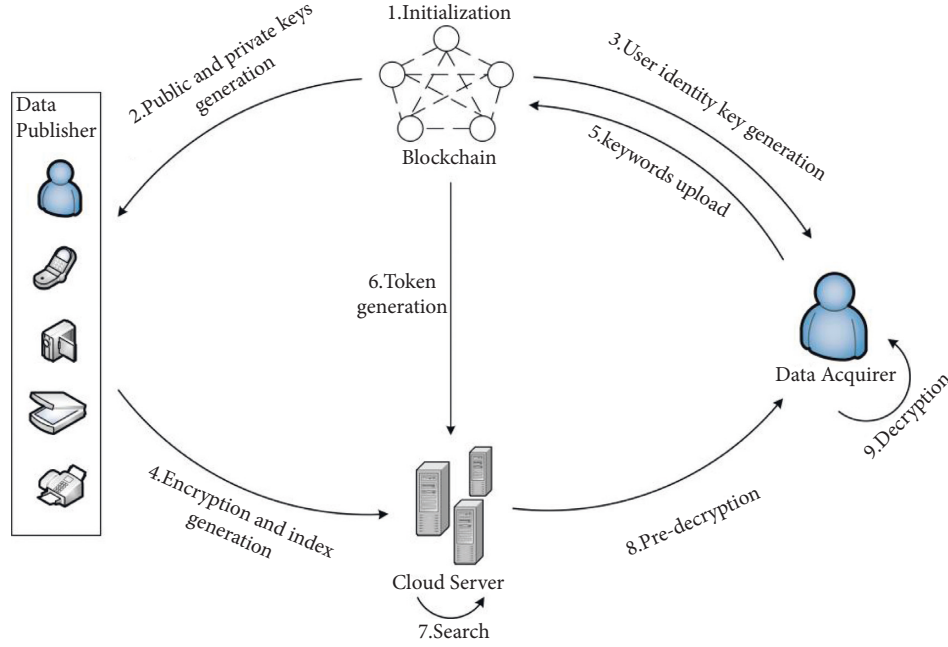


FIGURE 1: Scheme structure. The structure contains a specific implementation process for access control.

4.2.5. Token Generation. $\text{TokenGen}(GP, UK_{\text{UID}}, kw) \rightarrow \text{Tok}$. Token generation: this process is run by the blockchain consensus nodes, with the global public key GP , the user identity key UK_{UID} , and the keywords of the data user kw as input, and outputs user search token Tok .

4.2.6. Search. $\text{CipherTextSearch}(GP, \text{Tok}, \text{Index}) \rightarrow D$. Search: this process is run by the CS, taking the global public key GP , the user search token Tok , and the ciphertext index Index as input to output the matching ciphertext D .

4.2.7. Decryption. $\text{ProxyDecrypt}(GP, D, UK_{\text{UID}}) \rightarrow D'$. Proxy decryption: this process is run by the CS, taking the global public key GP , the ciphertext D , and the user identity key UK_{UID} as input. If the ciphertext attribute set Λ satisfies the ciphertext access structure (P, η) and the user attribute set K satisfies the user access structure (F, ρ) , the ciphertext is pre-decrypted and sends the intermediate ciphertext D' returned to the DA.

$\text{userDecrypt}(GP, D, D', UK_{\text{UID}}) \rightarrow M$. User decryption: this process is run by the DA, taking the global public key GP , the ciphertext D , the intermediate ciphertext D' , and the user identity key UK_{UID} as input, and outputs plaintext M .

The notations used in our scheme are summarized in Table 1.

4.3. Security Model

4.3.1. Ciphertext Indistinguishability. The indistinguishability security under chosen-plaintext attack (IND-CPA) of an STW-ABE scheme is defined by the following game between a challenger C and a probabilistic polynomial-time

(PPT) adversary A . Let A_u be the authority universe of size t . We define adversary A as a (t, n) adversary who can compromise at most $t - 1$ authority. This security model adopts the (t, n) key generation protocol. The description of the game is as follows:

- (1) Initialization: C runs the *Initialization* of STW-ABE and returns the global public key GP , user public key PK , and user private key SK to A .
- (2) Query phase I: adversary A queries the following oracles adaptively.
 - (a) *User Identity Key Oracle.* A submits an identity UID to C . C runs the $\text{KeyGen}(GP, SK, PK, \text{UID}, K, (P, \eta)) \rightarrow UK_{\text{UID}}$. Finally, it returns UK_{UID} to A .
 - (b) *Encryption Oracle.* A sends $((F, \rho), \Lambda, M)$ to C . C runs the $\text{Encrypt}(GP, SK, PK, (F, \rho), \Lambda, M) \rightarrow D, KW$ to generate the ciphertext D . Notice that the user access structure (F, ρ) does not satisfy the challenge user attribute set K , and the ciphertext attribute set Λ does not satisfy the challenge ciphertext access structure (P, η) .
- (3) Challenge: A submits two plaintexts of equal length M_0, M_1 and sends them to C . C selects a random number $\partial \in \{0, 1\}$ and encrypts the selected plaintext with user access structure (F^*, ρ^*) and ciphertext attribute set Λ^* . The final ciphertext will be generated (D^*) and sent to A .
- (4) Query phase II: A still can make queries adaptively as in *Query Phase I*.
- (5) Guess: A outputs a guess b' for b .

TABLE 1: Notations.

| Notation | Meaning |
|-------------|---|
| λ | A security parameter |
| GP | The global public key |
| SK | User's private key |
| PK | User's public key |
| UID | User's identity |
| K | User's attribute set |
| (P, η) | The ciphertext access structure |
| UK_{UID} | User's identity key |
| (F, ρ) | The user access structure |
| Λ | The ciphertext attribute set |
| KW | The keywords of ciphertext |
| Index | The ciphertext index |
| kw | User search keywords |
| Tok | Search token |
| D/D' | Ciphertext of the data/pre-decrypted ciphertext |

The advantage of A in this game is defined as follows:

$$\text{Adv}_A^D = \left| \Pr[b' = b] - \frac{1}{2} \right|. \quad (1)$$

Definition 2. An STW-ABE scheme is IND-CPA secure if the advantage defined above for any (t, n) PPT adversary A is negligible.

4.3.2. Index Indistinguishability. Index indistinguishable security (IND-CKA) under chosen access structure and chosen keyword attack is defined as the security game of challenger C and a probabilistic polynomial-time (PPT) adversary A for the STW-ABE scheme. In this scheme, only single keyword ciphertext retrieval is considered. The description of the game is as follows:

- (1) Initialization: A defines a user access structure (F^*, ρ^*) and ciphertext attribute set Λ^* .
- (2) Setup: C runs the *Initialization* of STW-ABE and returns the global public key GP , user public key PK , and user private key SK to A .
- (3) Query phase I: adversary A queries the following oracles adaptively.
 - (a) *User Identity Key Oracle.* A submits an identity UID to C . C runs the $\text{KeyGen}(GP, SK, PK, UID, \theta, (P, \eta)) \rightarrow UK_{UID}$. Finally, it returns UK_{UID} to A .
 - (b) *Token Oracle.* A send (kw) to C . C runs the $\text{TokenGen}(GP, SK, kw) \rightarrow \text{Tok}$ to generate the token Tok. Notice that the user access structure (F, ρ) does not satisfy the challenge user attribute set K , and ciphertext attribute set Λ does not satisfy the challenge ciphertext access structure (P, η) . We assume that all query results (Tok) have at least one matched index that can be searched out.
 - (c) *Index Oracle.* A submits $(UK_{UID}, \{KW\})$ to C , and C runs the $\text{IndexGen}(GP, PK, UK_{UID}, KW) \rightarrow \text{Index}$ to generate the index.

- (4) Challenge: A submits two keywords of equal length kw_0 , and kw_1 to C . C chooses randomly number $b \in \{0, 1\}$ and runs the $\text{IndexGen}(GP, PK, UK_{UID}, KW) \rightarrow \text{Index}$ with the challenge user access structure (F^*, ρ^*) and ciphertext attribute set Λ^* to return Index^* to A .
- (5) Query phase II: A still can make queries adaptively as in *Query Phase I* after receiving the challenge index. Similarly, A cannot query on the user access structure, which satisfies the challenge user attribute set, and ciphertext attribute set, which satisfies the ciphertext access structure.
- (6) Guess: A outputs a guess b' for b .

The advantage of A in this game is defined as follows:

$$\text{Adv}_A^{kw} = \left| \Pr[b' = b] - \frac{1}{2} \right|. \quad (2)$$

Definition 3. An STW-ABE scheme is IND-CKA secure if the advantage defined above for any PPT adversary A is negligible.

5. Construction

This section presents a detailed construction of our STW-ABE scheme, including initialization, user identity key generation, encryption, decryption, token generation, and search.

5.1. Initialization. This stage is divided into two parts. First, the blockchain consensus node executes the distributed key generation protocol to generate the global public key. Then, the blockchain consensus nodes generate user public and private keys.

Part One. $\text{Setup}(\lambda) \rightarrow GP$. First, the q -order bilinear group \mathbb{G}_0 with generator g and bilinear mapping $\mathbb{G}_0 \times \mathbb{G}_0 \rightarrow \mathbb{G}_T$ is selected in the setup. In addition, the description of a hash function $H: \{0, 1\}^* \rightarrow \mathbb{G}_0$ that maps

user identity UID to elements of \mathbb{G}_0 is published. Finally, the global public key is generated.

$$GP = \{g, H\}. \quad (3)$$

Part Two. Authority Setup (GP) \rightarrow SK, PK. The authorization center $CN_i (i = 1, 2, \dots, n)$ manages the set of user attributes \tilde{U}_i and ciphertext attributes \hat{U}_i of all users. CN_i random selection of parameters $\alpha_q \in \mathbb{Z}_p^* (q \in \overline{U}_i)$, $\beta_d \in \mathbb{Z}_p^* (d \in \hat{U}_i)$ according to the attribute set. Then, blockchain consensus nodes generate user public key $PK = \{e(g, g)^{\alpha_i}, g^{\beta_i}\}$ and user private key $SK = \{\alpha_i, \beta_i\}$.

5.2. User Identity Key Generation. $\text{KeyGen}(GP, SK, PK, \text{UID}, K, (P, \eta)) \rightarrow UK_{\text{UID}}$. This KeyGen is run by the consensus nodes that execute the distributed key generation protocol, taking the global public key GP , the user public key PK , the user private key SK , the user attributes set K , the ciphertext access structure (P, η) , and user's identity UID as inputs to output the user identity key UK_{UID} .

- (1) Let P be a $l_o \times k_o$ matrix. The process randomly selects $\varphi \in \mathbb{Z}_p^*$, $y_i \in \mathbb{Z}_p^* (i = 2, \dots, k_o)$ and constructs the vector $\vec{v}_{k_o} = (\varphi, y_2, \dots, y_{k_o})^T$ and vector $\vec{w}_{k_o} = (0, y_2, \dots, y_{k_o})^T$. φ is the secret value to be shared.
- (2) Let P_x be the x -th row of the matrix P , and calculate $\sigma_x = P_x \cdot \vec{v}_{k_o}$, $\tau_x = P_x \cdot \vec{w}_{k_o}$.
- (3) Select $\mu_x \in \mathbb{Z}_p^*$, $x = 1, 2, \dots, l_o$ for each P_x to calculate the following equation:

$$U = e(g, g)^\varphi, \\ U_{x,1} = e(g, g)^{\sigma_x} e(g, g)^{\alpha_{\eta(x)} \mu_x}, U_{x,2} = g^{\mu_x}, U_{x,3} = g^{\beta_{\eta(x)} \mu_x} g^{\tau_x}. \quad (4)$$

- (4) Create a key that belongs to a primary attribute $t (t \in K)$ for the user identity UID and do the following calculation: $U_t = g^{\alpha_t} H(\text{UID})^{\beta_t}$.
- (5) Finally, the user identity key is generated ($UK_{\text{UID}} = \{(P, \eta), U, \{U_{x,1}, U_{x,2}, U_{x,3}\}_{x=1,2,\dots,l_o}, \{U_t\}_{t \in K}\}$) and sent to the DA.

5.3. Encryption. The encryption consists of two processes, namely, encryption $\text{Encrypt}(GP, SK, PK, \text{UID}, (F, \rho), \Lambda, M) \rightarrow D, KW$ and the index generation $\text{IndexGen}(GP, SK, PK, KW) \rightarrow \text{Index}$.

$$\text{Encrypt}(GP, SK, PK, \text{UID}, (F, \rho), \Lambda, M) \rightarrow D, KW.$$

This process is run by the DP , taking the global public key GP , the user public key PK , the user private key SK , the user identity UID, the user access structure (F, ρ) , ciphertext attribute set Λ , and plaintext M as input.

- (1) Let F be a $l_e \times k_e$ matrix. The process first randomly selects $s \in \mathbb{Z}_p^*$, $t_j \in \mathbb{Z}_p^* (j = 2, \dots, k_e)$. Let vector

$\vec{v}_{k_e} = (s, t_2, \dots, t_{k_e})^T$, $\vec{w}_{k_e} = (0, t_2, \dots, t_{k_e})^T$, and s be the secret value to be shared.

- (2) Let F_x be the x -th row of the matrix F , and $\lambda_x = F_x \cdot \vec{v}_{k_e}$, $\mu_x = F_x \cdot \vec{w}_{k_e}$.
- (3) Select $r_x \in \mathbb{Z}_p^*$, $x = 1, 2, \dots, l_e$ for each F_x to calculate the following equation:

$$D = Me(g, g)^s, \\ D_{x,1} = e(g, g)^{\lambda_x} e(g, g)^{\alpha_{\rho(x)} r_x}, D_{x,2} = g^{r_x}, D_{x,3} = g^{\beta_{\rho(x)} r_x} g^{\mu_x}. \quad (5)$$

- (4) Create a key that belongs to the corresponding subattribute $i (i \in \Lambda)$ for the encrypted file, and the following calculation is performed:

$$D_k = g^{\alpha_k} H(\text{UID})^{\beta_k}. \quad (6)$$

- (5) Finally, the ciphertext is generated:

$$D = \{(F, \rho), D, \{D_{x,1}, D_{x,2}, D_{x,3}\}_{x=1,2,\dots,l_e}, \{D_k\}_{k \in \Lambda}\}, \quad (7)$$

and sent to the CS.

$\text{IndexGen}(GP, PK, KW) \rightarrow \text{Index}$. This process was conducted by DP on local devices, with the global public key GP , the user public key PK , and the keywords of ciphertext KW as inputs. U_w is the number of data keywords. The following calculations are performed to encrypt each keyword into a ciphertext index.

$$\text{Idx}_{1,\omega} = e(g, g)^{\alpha_\omega \cdot \varphi \cdot H(kw_\omega)}, \text{Idx}_{2,i} = g^{\beta_i \cdot \varphi}. \quad (8)$$

Finally, the ciphertext index $\text{Index} = \{\{\text{Idx}_{1,\omega}\}_{\omega \in U_w}, \text{Idx}_{2,i}\}$ is obtained and sent to the CS.

5.4. Token Generation. $\text{TokenGen}(GP, UK_{\text{UID}}, kw) \rightarrow \text{Tok}$. This process is run by the consensus nodes that execute the distributed key generation protocol, with the global public key GP , the user identity key UK_{UID} , and the keywords of the data users kw as input. The following calculations are performed.

$$\text{tok}_i = \left(\frac{\alpha_i}{g^{\beta_i}} \right)^{H(kw)}. \quad (9)$$

Finally, the user tokens are generated ($\text{Tok} = \{\text{tok}_i\}$) and sent to the CS.

5.5. Search. $\text{CipherTextSearch}(GP, \text{Tok}, \text{Index}) \rightarrow D$. The search is conducted by the CS. This process takes the global public key GP , the user search token Tok , and the ciphertext index Index as input. Suppose the ciphertext search is successful, output the ciphertext. Otherwise, the process is terminated.

(1) Judge if the following equation holds:

$$\text{Idx}_{1,\omega} = \prod_{i \in U_i} e(\text{Idx}_{2,i}, \text{tok}_i). \quad (10)$$

(2) If yes, output the storage ciphertext D ; else, abort.

5.6. Decryption. The decryption consists of two processes, namely, the proxy decryption process $\text{ProxyDecrypt}(GP, D, UK_{\text{UID}}) \rightarrow D'$ and the user decryption process $\text{userDecrypt}(GP, D, D', UK_{\text{UID}}) \rightarrow M$.

$\text{ProxyDecrypt}(GP, D, UK_{\text{UID}}) \rightarrow D'$. Proxy decryption is run by the CS, taking the global public key GP , the ciphertext D , and the user identity key UK_{UID} as input. Determine whether the user attributes satisfy the file access permission, whether the ciphertext attribute set Λ satisfies the ciphertext access structure (P, η) , and whether the user attribute set K satisfies the user access structure (F, ρ) .

Verify that the user attribute set satisfies the user access structure; randomly selected $c_x \in \mathbb{Z}_p^*$ makes $\sum_{x \in \theta} c_x \lambda_x = s$, $\sum_{x \in \theta} c_x \mu_x = 0$. Similarly, verify that the ciphertext attribute set satisfies the ciphertext access structure; randomly selected $d_y \in \mathbb{Z}_p^*$ makes $\sum_{y \in \omega} d_y \sigma_y = \varphi$, $\sum_{y \in \omega} d_y \tau_y = 0$. If the authentication succeeds, perform the following calculation for the ciphertext pre-decryption.

Pre-decryption equation:

$$\begin{aligned} D' &= \frac{\prod_{x \in K} (D_{x,1} \cdot e(H(\text{UID}), D_{x,3}) / e(\check{K}_{\rho(x)}, D_{x,2}))^{c_x}}{\prod_{y \in \Lambda} (U_{y,1} \cdot e(H(\text{UID}), U_{y,3}) / e(\check{C}_{\eta(y)}, U_{y,2}))^{d_y}} \\ &= \frac{\prod_{x \in K} (e(g, g)^{\sigma_x} e(H(\text{UID}), g)^{\tau_x})^{c_x}}{\prod_{y \in \Lambda} (e(g, g)^{\lambda_y} e(H(\text{UID}), g)^{\mu_y})^{d_y}} \\ &= \frac{e(g, g)^s}{e(g, g)^\varphi}. \end{aligned} \quad (11)$$

$\text{userDecrypt}(GP, D, D', UK_{\text{UID}}) \rightarrow M$. The user decryption is run by DA, taking the global public key GP , the ciphertext D , the intermediate ciphertext D' , and the user identity key UK_{UID} as input.

Decryption equation:

$$M = \frac{D}{D' \cdot U}. \quad (12)$$

6. Security and Performance Analysis

6.1. Security Analysis. The STW-ABE simplifies the security problem to a decisional bilinear Diffie–Hellman (DBDH) problem.

Theorem 1. *The STW-ABE scheme is IND-CPA secure if the decisional bilinear Diffie–Hellman (DBDH) problem is hard.*

Proof. If adversary A can break the STW-ABE scheme with a non-negligible advantage, adversary A can solve the

DBDH problem with a non-negligible advantage. A_q -order bilinear group \mathbb{G}_0 with generator g and bilinear mapping $\mathbb{G}_0 \times \mathbb{G}_0 \rightarrow \mathbb{G}_T$ exists. C plays as the challenger in the following steps. Given an instance of the DBDH problem (g, g^a, g^b, g^c, Z) , where $a, b, c, z \in \mathbb{Z}_q$ are randomly selected. $\rho \in \{0, 1\}$; when $\rho = 0$, $Z = e(g, g)^{abc}$; when $\rho = 1$, $Z = e(g, g)^z$.

Initialization. C runs the *Initialization* of STW-ABE and returns the global public key GP , user public key PK , and user private key SK to A .

Query Phase I. Adversary A queries the following oracles adaptively.

User Identity Key Oracle. A submits an identity UID to C . C runs the $\text{KeyGen}(GP, SK, PK, \text{UID}, K, (P, \eta)) \rightarrow UK_{\text{UID}}$. Finally, it returns UK_{UID} to A .

Encryption Oracle. A sends $((F, \rho), \Lambda, M)$ to C . C runs the $\text{Encrypt}(GP, SK, PK, (F, \rho), \Lambda, M)$ to generate the ciphertext D . Notice that the primary access structure (F, ρ) does not satisfy the challenge primary attribute set K , and ciphertext attribute set Λ does not satisfy the challenge secondary access structure (P, η) .

Challenge. A submits two plaintexts of equal length M_0, M_1 and sends them to C . C selects a random number $\partial \in \{0, 1\}$ and then encrypts the selected plaintext with user access structure (F^*, ρ^*) and ciphertext attribute set Λ^* . The final ciphertext D^* will be generated and sent to A .

Query Phase II. A still can make queries adaptively as in *Query Phase I* after receiving the challenge ciphertext D . Similarly, A cannot query the user access structure that satisfies the challenge user attribute set and the ciphertext attribute set that satisfies the ciphertext access structure.

Guess. A outputs a guess ∂' for ∂ . If $\partial' = \partial$, C outputs $\rho' = 0$, and C receives a tuple $(g, g^a, g^b, g^c, e(g, g)^{abc})$. Otherwise, C outputs $\rho' = 1$, and C receives a tuple $(g, g^a, g^b, g^c, e(g, g)^z)$. The advantage of A is analyzed as follows.

When $\rho = 1$, $Z = e(g, g)^z$, A cannot obtain the information of D . Thus, $\Pr[\partial' \neq \partial | \partial = 1] = 1/2$. When $\rho' = 1$, $\Pr[\rho' = \rho | \rho = 1] = 1/2$.

When $\rho = 0$, $Z = e(g, g)^{abc}$, A obtains ciphertext D . Thus, $\Pr[\partial' = \partial | \partial = 0] = 1/2 + \epsilon$. When $\rho' = 0$, $\Pr[\rho' = \rho | \rho = 0] = 1/2 + \epsilon$.

Thus, C guesses $\rho' = \rho$, and the correct advantage is

$$\begin{aligned} \text{Adv} &= \Pr[\rho' = \rho] - \frac{1}{2} = \frac{1}{2} \Pr[\rho' = \rho | \rho = 1] \\ &\quad + \frac{1}{2} \Pr[\rho' = \rho | \rho = 0] - \frac{1}{2} = \frac{\epsilon}{2}. \end{aligned} \quad (13)$$

In summary, if adversary A can break the proposed scheme with a non-negligible advantage in polynomial time, a scheme that can solve the DBDH problem with a non-negligible advantage $\epsilon/2$ in polynomial time exists. However, the DBDH problem is difficult, so the STW-ABE scheme is IND-CPA secure. \square

Theorem 2. *The STW-ABE scheme is IND-CKA secure if the decisional bilinear Diffie-Hellman (DBDH) problem is hard.*

Proof. Assume that there is a PPT adversary A who can win the index indistinguishability security game defined in Section 4.3.2 with non-negligible advantage ϵ . Then, we can construct a C to solve the DBDH problem with a non-negligible advantage $(\epsilon/2)$. C plays as the challenger in the following steps. Given an instance of the DBDH problem (g, g^a, g^b, g^c, Z) , where $a, b, c, z \in \mathbb{Z}_q$ are randomly selected. $\rho \in \{0, 1\}$; when $\rho = 0$, $Z = e(g, g)^{abc}$; when $\rho = 1$, $Z = e(g, g)^z$.

Initialization. A defines a user access structure (F^*, ρ^*) and ciphertext attribute set Λ^* .

Setup. C runs the *Initialization* of STW-ABE and returns the global public key GP , user public key PK , and user private key SK to A .

Query Phase I. Adversary A queries the following oracles adaptively.

User Identity Key Oracle. A submits an identity UID to C . C runs the $KeyGen(GP, SK, PK, UID, K, (P, \eta)) \rightarrow UK_{UID}$. Finally, it returns UK_{UID} to A .

Token Oracle. A sends (kw) to C . C runs the $TokenGen(GP, SK, kw) \rightarrow Tok$ to generate the token Tok . Notice that the user access structure (F, ρ) does not satisfy the challenge user attribute set K , and the ciphertext attribute set Λ does not satisfy the challenge ciphertext access structure (P, η) . We assume that all query results (Tok) have at least one matched index that can be searched out.

Index Oracle. A submits $(UK_{UID}, \{KW\})$ to C , and C runs the $IndexGen(GP, PK, UK_{UID}, KW)$ to generate the index.

Challenge. A submits two keywords of equal length kw_0 and kw_1 to C . C chooses number $b \in \{0, 1\}$ randomly and runs the $IndexGen(GP, PK, UK_{UID}, KW) \rightarrow Index$ with the challenge user access structure (F^*, ρ^*) and ciphertext attribute set Λ^* to return Idx^* to A .

$$Idx_{1, kw_b}^* = Z^H(kw_b), Idx_2^* = A^b. \quad (14)$$

The advantage of A is analyzed as follows.

When $Z = e(g, g)^{abc}$, we set $s = a, \alpha = bc$; then, the index presented as follows is identical to an actual index:

$$\begin{aligned} Idx_{1, kw_b}^* &= (e(g, g)^{abc})^{H(kw_b)} \\ &= e(g, g)^{\alpha \cdot \varphi \cdot H(kw_b)} \\ Idx_2 &= g^{b \cdot \varphi}. \end{aligned} \quad (15)$$

When $Z = e(g, g)^z$, due to the randomness of z , this index is random to the adversary and contains no information about b .

Query Phase II. A still can make queries adaptively as in *Query Phase I* after receiving the challenge *Index*. Similarly, A cannot query the user access structure that satisfies the challenge user attribute set and the ciphertext attribute set that satisfies the ciphertext access structure.

Guess. A outputs a guess b' for b . If $Z = e(g, g)^{abc}$, the probability of A outputs $b' = b$ is $1/2 + \epsilon$. If $Z = e(g, g)^z$, the probability of A outputs $b' = b$ is $1/2$. Thus, the advantage of C solving the DBDH problem is

$$\begin{aligned} Adv &= \left| \frac{1}{2} \Pr[b' = b | Z = e(g, g)^{abc}] \right. \\ &\quad \left. + \frac{1}{2} \Pr[b' = b | Z = e(g, g)^z] - \frac{1}{2} \right| \\ &= \left| \left[\frac{1}{2} \left(\frac{1}{2} + \epsilon \right) + \frac{1}{2} \cdot \frac{1}{2} \right] - \frac{1}{2} \right| \\ &= \frac{\epsilon}{2}. \end{aligned} \quad (16)$$

Because the DBDH problem is hard, we can get that $\epsilon/2$ is negligible. In other words, the advantage of A breaking our scheme is negligible, and our scheme achieves chosen keyword security. \square

6.2. Performance Analysis. In this section, we analyze the performance and computational efficiency of STW-ABE. We compare the performance of STW-ABE with other schemes in Table 2, where “ \sqrt ” indicates that the solution supports this method. “ \times ” indicates that the solution does not support this method. In Table 3, we compare the computational efficiency of STW-ABE with other schemes, in which E represents an exponential operation, P represents a pairing operation, H represents a hash operation, i represents the number of attributes in the authorized institution, n represents the number of keywords in each document, m is the number of keywords searched by the user, M_i is the number of the ciphertext attributes, and M_e is the number of the user attributes.

As seen in Table 2, our scheme not only realizes two-way access control of ciphertext search but also uses CS to provide outsourced decryption service, reducing the computational pressure on users.

Table 3 compares the computational efficiency of encryption, decryption, index generation, token generation,

TABLE 2: Comparison of functions.

| Scheme | DP-ABE | PAB-MSK | D-ABE | BC-SABE | STW-ABE |
|-----------------------|--------|---------|-------|---------|---------|
| Two-way | √ | × | × | × | √ |
| Searchable encryption | × | √ | × | √ | √ |
| Proxy encryption | √ | √ | √ | √ | √ |

TABLE 3: Comparison of computational methods.

| Scheme | DP-ABE | PAB-MSK | D-ABE | BC-SABE | STW-ABE |
|------------------|--|-----------------|-------------|--------------------------------------|------------------------------------|
| Encryption | $(6M_i + 5)E + H$ | $(3n + 3)E + H$ | $(5i + 1)E$ | $(4i + 3)E$ | $(3M_i + 2)E + H$ |
| Index generation | — | $(3n + 3)E$ | — | $(3n + 1)E + H$ | $(2i + n)E + H$ |
| Token generation | — | $(2m + 3)E$ | — | $(2m + 2)E + H$ | $(i + m)E + H$ |
| Search | — | $E + (2n + 4)P$ | — | $iE + (2i + 1)P$ | $M_iE + (i + m)P$ |
| Decryption | User: $2E$ Cloud: $2E + ((M_i + M_e)P)$ | $iE + 2iP$ | $iE + 2iP$ | User: E Cloud: $iE + (3i + 2)P$ | User: E Cloud: $(M_i + M_e)P$ |

and ciphertext search. In our scheme, first, the user needs to perform a $(3M_i + 2)$ exponential operation and a hash operation to encrypt the data, in which only one exponential operation is required for each ciphertext attribute. The user performs a hash operation on each ciphertext keyword and $(2i + n)$ exponential operation to generate a ciphertext index. Secondly, the cloud server performs a hash operation and $(i + m)$ exponential operation for each keyword to be searched to generate a token for data users. Then, the cloud server performs a ciphertext search by an exponential operation of M_i and pairing operation time of $(i + m)$. In decryption, the scheme divides the decryption cost into the user part (denoted as User) and the cloud server part (denoted as Cloud). The cloud server performs the pairing operation $(M_i + M_e)$. The user then only needs to perform the exponential operation once to decrypt the ciphertext into plaintext. Furthermore, the STW-ABE scheme is compared with two multi-permission ABE schemes, DP-ABE [5] and D-ABE [21], and two searchable encryption schemes, PAB-MSK [11] and BC-SABE [19], in Table 3. The cost of linear secret-sharing protocol is ignored in efficiency analysis.

Figures 2 and 3 contain the simulation results of the five processes. We simulated this on an Ubuntu 16 desktop system. The system has an Intel Core i7-8700 CPU and 4GB RAM. All programs were developed using Charm (version 0.50) [27], a rapid prototyping framework based on the Python encryption scheme.

Figure 2(a) shows the encryption time cost of three ABE schemes with multiple authority agencies. As seen in the figure, the time cost of all the resulting schemes has a linear relationship with the number of attributes contained in the encrypted access structure. Figure 2(b) shows the decryption time cost of schemes D-ABE, BC-SABE, and STW-ABE. As seen in Figure 2, the time cost of cloud decryption of D-ABE, BC-SABE, and STW-ABE has a linear relationship with the number of attributes, and the user decryption cost in STW-ABE is independent of the number of attributes. Since STW-ABE outsources most of the decryption work to cloud servers, the computing pressure on users is greatly reduced. This scheme is more suitable for using lightweight devices in the IoT environment.

Figure 3(a) shows the time cost of the generated ciphertext index. It can be seen from the figure that STW-ABE has better computing performance than scheme PAB-MSK and scheme BC-SABE. Figure 3(b) shows the simulation results of the time cost required for the generation of the Token. The time cost of the schemes is linearly related to the number of attributes, but it can be seen that the time cost of STW-ABE is much shorter than that of scheme BC-SABE. Figure 3(c) shows the time cost of the search process under simulation. In this scheme, the file index and the number of files are fixed as simulated constants. The results of the search process represent only the performance of the search process and do not include the time cost of searching the actual database. As seen in Figure 3(c), the time cost of the STW-ABE search process is similar to that of the search process in the PAB-MSK scheme and the BC-SABE scheme. Moreover, they are all linearly related to the number of attributes.

Discussion. There are two concerns when designing searchable encryption access control schemes. (1) *Security.* In Section 4.1, the two-way access control scheme based on searchable encryption has been proven to be IND-CPA security and IND-CKA security, which is also achieved by most searchable encryption schemes. In this paper, we use the distributed feature of blockchain and change the central authorization model in traditional access control to a blockchain consensus node with DKG that generates the relevant secret key. Where DKG follows the (t, n) ($n \leq t$)-sharing principle, t is the total number of blockchain consensus nodes involved in key generation, and n is the minimum number of consensus nodes involved in key generation. The secret key sharing must be participated by more than n consensus nodes, thus improving the robustness of the scheme. At the same time, a blockchain is a distributed ledger that ensures data integrity, immutability, and traceability of the information stored in it such as global public keys and user keys. (2) *Efficiency.* In this paper, the simulation experiment simulated the efficiency of the scheme and compared it with other schemes. It can be found that this scheme has certain advantages in implementation

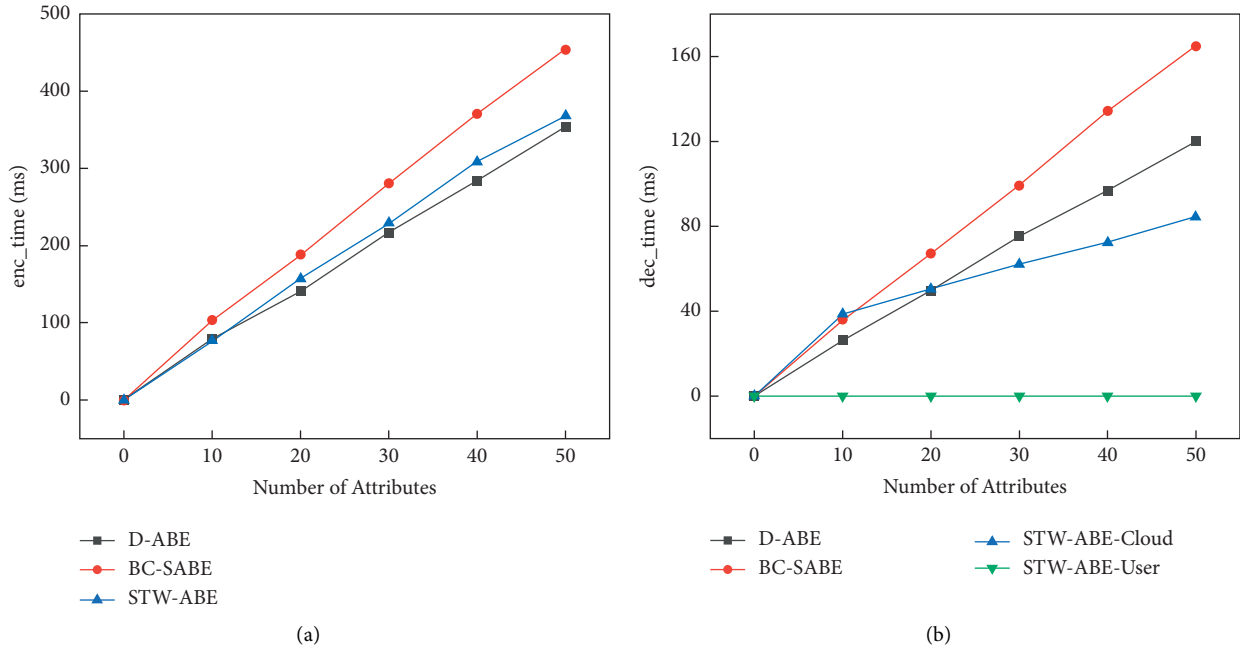


FIGURE 2: Time cost of encryption and decryption. (a) Encryption time. (b) Decryption time.

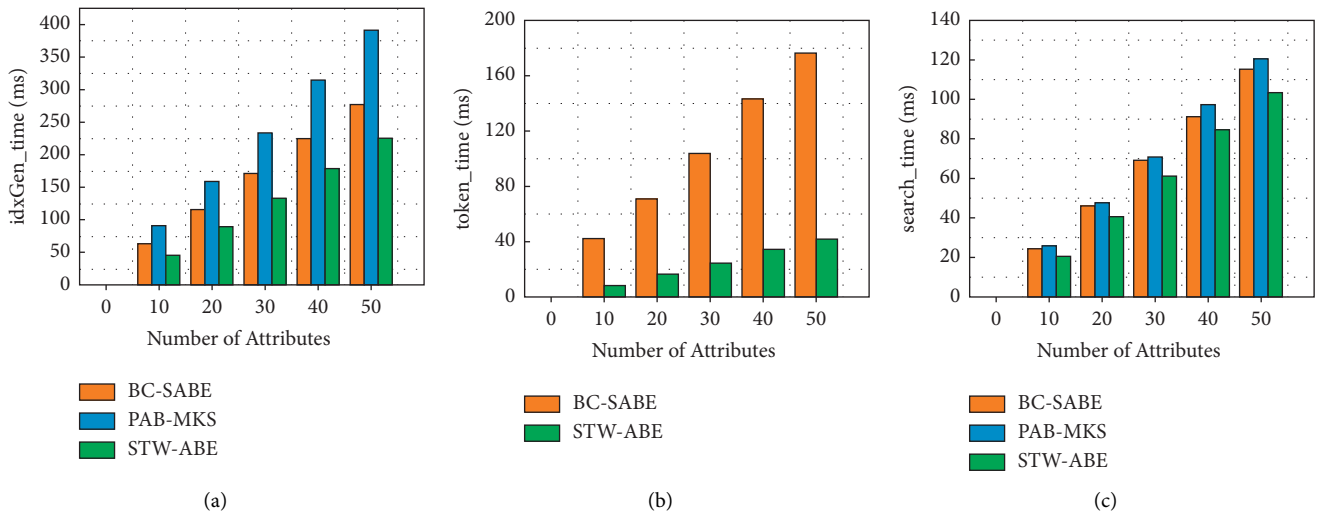


FIGURE 3: Time costs of ciphertext search. (a) Ciphertext index generation time. (b) Token generation time. (c) Ciphertext search time.

efficiency. In the decryption process, considering that most IoT devices have limited resources and cannot perform efficient decryption calculations, this scheme uses cloud servers to assist users in decryption. A large amount of decryption computation is outsourced to CS, thus reducing the computational pressure on users. This scheme adopts a distributed key generation protocol, and multiple blockchain consensus nodes participate in generating users' public and private key pairs, which will not affect the security and robustness of the scheme. Meanwhile, the secret keys generated by blockchain nodes do not need to respond to user requests in real time, so the time cost of secret key generation is not simulated in this paper. Among them, in the BC-SABE scheme, the cloud server is used to complete

the generation of tokens with the user jointly, and the user does not need to perform the calculation related to the number of attributes. In the BC-SABE scheme, the token generation time by the blockchain consensus node is not given, so there is no comparison between them in Figure 3(b). Similarly, the generation of a token in STW-ABE is completed by the blockchain consensus node. The user does not need to calculate the consumption in the generation of tokens.

7. Conclusions

This paper proposes a distributed STW-ABE scheme using coalition blockchain and cloud servers to assist users with

accurate and secure data search and sharing. Our solution not only enables two-way confirmation between users and data but also enables the fine-grained search of ciphertext and lightweight decryption for users. In addition, our scheme utilizes a coalition blockchain to replace the centralized key management server. The consensus nodes jointly generate key parameters through the DKG, improving the security of the IoT system. Then, the blockchain is responsible for generating public and private keys, user identity keys, and keyword tokens. Due to the limited resources of IoT devices and the massive pairing operations required for the search and decryption process, we delegate a large amount of computation to CS during the search and decryption process. The user only needs one exponential operation to complete the decryption process from ciphertext to plaintext. The present security and efficiency analysis shows that the scheme has good safety and practicality.

Our ultimate aim is to design a secure and efficient data sharing system for the IIoT. The possible further research direction is to implement dynamic updating of access policies based on the current work.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the National Natural Science Foundation of China (grant no. 61772180), the Key-Area Research and Development Program of Guangdong Province (2020B1111420002), the Key-Area Research and Development Program of Hubei Province (2022BAA040), the Science and Technology Project of Department of Transport of Hubei Province (2022-11-4-3), and the Innovation Fund of Hubei University of Technology (BSQD2019027, BSQD2019020, and BSQD2016019).

References

- [1] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proceedings of the International Conference on Theory & Applications of Cryptographic Techniques*, Springer, Berlin, Heidelberg, 2005.
- [2] B. Waters, "Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization," in *Proceedings of the International Workshop on Public Key Cryptography*, Springer, Berlin Heidelberg, 2008.
- [3] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM Conference on Computer and Communications Security*, Alexandria, VA, USA, 2006.
- [4] N. Attrapadung and H. Imai, "Dual-policy attribute based encryption," in *Proceedings of the 7th International Conference on Applied Cryptography and Network Security*, Springer, Berlin, Heidelberg, 2009.
- [5] D. Han, J. Chen, L. Zhang, Y. Shen, and Y. Gao, "Access control of blockchain based on dual-policy attribute-based encryption," in *Proceedings of the 2020 IEEE 22nd International Conference on High Performance Computing and Communications; IEEE 18th International Conference on Smart City; IEEE 6th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, Yanuca Island, Cuvu, Fiji, December 2020.
- [6] J. Leng, M. Zhou, J. L. Zhao, Y. Huang, and Y. Bian, "Makerchain: a blockchain with chemical signature for self-organizing process in social manufacturing," *Journal of Cleaner Production*, vol. 234, pp. 767–778, 2019.
- [7] Z. Rahman, I. Khalil, X. Yi, and M. Atiquzzaman, "Blockchain-based security framework for a critical industry 4.0 cyber-physical system," *IEEE Communications Magazine*, vol. 59, no. 5, pp. 128–134, 2021.
- [8] J. Li, X. Lin, Y. Zhang, and J. Han, "KSF-OABE: outsourced attribute-based encryption with keyword search function for cloud storage," *IEEE Transactions on Services Computing*, vol. 10, no. 5, pp. 715–725, 2017.
- [9] D. Ziegler and A. Marsalek, "Efficient Revocable Attribute-Based Encryption with Hidden Policies," in *Proceedings of the 2020 IEEE 19th International Conference On Trust, Security And Privacy In Computing And Communications (TrustCom)*, Guangzhou, China, 2020.
- [10] X. Liu, G. Yang, W. Susilo, J. Tonien, X. Liu, and J. Shen, "Privacy-preserving multi-keyword searchable encryption for distributed systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 3, pp. 561–574, 2021.
- [11] Y. Miao, J. Ma, X. Liu, X. Li, Z. Liu, and H. Li, "Practical attribute-based multi-keyword search scheme in mobile crowdsourcing," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 3008–3018, 2018.
- [12] J. Leng, G. Ruan, P. Jiang et al., "Blockchain-empowered sustainable manufacturing and product lifecycle management in industry 4.0: a survey," *Renewable and Sustainable Energy Reviews*, vol. 132, Article ID 110112, 2020.
- [13] D. Mehta, T. Sudeep, B. Umesh, and S. Arpit, "Blockchain-based royalty contract transactions scheme for industry 4.0 supply-chain management-," *Sciencedirect. Information Processing & Management*, vol. 58, no. 4.
- [14] J. Leng, M. Zhou, J. L. Zhao, Y. Huang, and Y. Bian, "Blockchain Security: A Survey of Techniques and Research Directions," *IEEE Transactions on Services Computing*, vol. 15, 2020.
- [15] J. Leng, S. Ye, M. Zhou et al., "Blockchain-secured smart manufacturing in industry 4.0: a survey," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 1, pp. 237–252, 2021.
- [16] M. Li, C. Jia, and W. Shao, "Blockchain based multi-keyword similarity search scheme over encrypted data Security and Privacy in Communication Networks," in *Security and Privacy in Communication Networks. SecureComm 2020. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, Vol. 336, Springer, Cham, Switzerland, 2020.
- [17] T. Feng, H. Pei, R. Ma, Y. Tian, and X. Feng, "Blockchain data privacy access control based on searchable attribute encryption," *Computers, Materials & Continua*, vol. 66, no. 1, pp. 871–890, 2020.
- [18] S. Gao, G. Piao, J. Zhu, X. Ma, and J. Ma, "Trustaccess: a trustworthy secure ciphertext-policy and attribute hiding

- access control scheme based on blockchain,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 5784–5798, 2020.
- [19] S. Liu, J. Yu, Y. Xiao, Z. Wan, S. Wang, and B. Yan, “Bc-sabe: blockchain-aided searchable attribute-based encryption for cloud-iot,” *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 7851–7867, 2020.
- [20] X. Qin, Y. Huang, Z. Yang, and X. Li, “Lbac: a lightweight blockchain-based access control scheme for the internet of things -,” *Information Sciences*, vol. 554, pp. 222–235, 2021.
- [21] A. Lewko and B. Waters, “Decentralizing attribute-based encryption,” *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, Berlin, Heidelberg, 2011.
- [22] X. Qin, Y. Huang, Z. Yang, and X. Li, “A blockchain-based access control scheme with multiple attribute authorities for secure cloud data sharing,” *Journal of Systems Architecture*, vol. 112, no. 11, Article ID 101854, 2021.
- [23] N. Shi, L. Tan, C. Yang, C. He, and H. Xu, “Bacs: a blockchain-based access control scheme in distributed internet of things,” *Peer-to-Peer Networking and Applications*, vol. 14, no. 6, 2020.
- [24] A. Beimel, *Secure Schemes for Secret Sharing and Key Distribution*, Israel Institute of Technology-Technion, Haifa, Israel, 1996.
- [25] T. P. Pedersen, “A threshold cryptosystem without a trusted party (extended abstract),” in *Proceedings of the Advances in Cryptology - EUROCRYPT '91, Workshop on the Theory and Application of of Cryptographic Techniques*, Brighton, UK, 1991.
- [26] D. X. Song, D. Wagner, and A. Perrig, “Practical techniques for searches on encrypted data,” in *Proceeding of the 2000 IEEE Symposium on Security and Privacy. Science Progress 2000*, pp. 44–55, Berkeley, CA, USA, May 2000.
- [27] J. A. Akinyele, C. Garman, I. Miers et al., “Charm: a framework for rapidly prototyping cryptosystems,” *Journal of Cryptographic Engineering*, vol. 3, no. 2, pp. 111–128, 2013.

Research Article

Service-Oriented Modeling for Blockchain-Enabled Supply Chain Quality Information Systems

Yani Shi ¹, Jiji Ying ², Dongying Shi ² and Jiaqi Yan ²

¹School of Economics and Management, Southeast University, Nanjing, China

²School of Information Management, Nanjing University, Nanjing, China

Correspondence should be addressed to Jiaqi Yan; jiaqian@nju.edu.cn

Received 19 April 2022; Accepted 16 July 2022; Published 9 August 2022

Academic Editor: Jiewu Leng

Copyright © 2022 Yani Shi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Quality management is one of the most critical issues in supply chain management. The rapid growth of information technologies, such as blockchain technology, has facilitated effective information systems development to support supply chain quality management. However, a significant challenge in developing blockchain-enabled supply chain quality information systems is how to deal with information asymmetry and the conflicting interests of supply chain partners. Taking a service-dominant view, this research proposes a Blockchain-Oriented Service Modeling (BOSM) approach for blockchain-enabled supply chain quality information systems. We provide a visual language for modeling the coordination and integration of business processes and domain knowledge at the knowledge level to facilitate the alignment of blockchain technology with supply chain quality management. The proposed approach bridges operational service computing with strategic service management in blockchain-enabled supply chain quality management and facilitates the communication between business people in supply chain management and software professionals in blockchain-based service computing. A case study on a dairy supply chain is presented to show advantages of the modeling framework under the service-dominant view, separating the cause of quality from the carrier of quality in the design of blockchain-enabled supply chain quality information systems.

1. Introduction

Managing quality is one of the most important factors in supply chains that involve many organizations collaborating to provide products or services. If the quality of materials from suppliers is not appropriately controlled, it may affect the end product's quality and lead to serious outcomes. The systematic collaboration between supply chain organizations in producing products makes it important to conduct quality management at a supply chain level. Robinson and Malhotra [1] reviewed the literature on quality management and supply chain management and argued that quality practice must advance from traditional firm-centric and product-based mindsets to an interorganizational supply chain orientation involving customers, suppliers, and other partners. Supply Chain Quality Management (SCQM) is defined as a system-based approach for performance

improvement that leverages the opportunities created by upstream and downstream linkages with suppliers and customers [2, 3].

Quality management at a supply chain level faces many challenges. Supply chain partner enterprises are usually geographically diverse and belong to organizations with different interests. There is no perfect inspection technology to accurately measure product quality. Thus, as a result of information asymmetry on product quality, the moral hazard effect exists in supply chain quality inspection, which may cause an inefficient supply chain. To tackle the SCQM problem, one approach is to leverage advanced information technology to build quality information systems. Among other solutions, blockchain has emerged as a leading technology since it provides secure traceability and control, immutability, and trust creation among stakeholders in a low-cost IT solution [4]. Many recent studies have discussed on how to improve SCQM by adopting blockchain

technology [5–8], finding that “trackability” and “traceability” are considered as the prime success factors of a blockchain-based supply chain [9].

However, there are few studies exploring blockchain-oriented software engineering [10], in particular, the requirement modeling methods for blockchain-enabled supply chain quality management. This can be attributed to the lack of an appropriate modeling perspective that synthesizes the nature of supply chain quality management with the characteristics of blockchain technology. SCQM needs to consider quality initiatives along supply chains, including upstream and downstream parties; thus, an appropriate modeling framework for blockchain-enabled Supply Chain Quality Information Systems (SCQIS) should consider both decentralized and network features. As blockchain technology is centered around a peer-to-peer network, enabling collaboration between different parties, it becomes an enabler of service systems [11]. In the service-dominant (S-D) logic perspective, service refers to the application of specialized competences (knowledge and skills) through deeds, processes, and performances for the benefit of another entity or the entity itself. While a service system is such a configuration of different entities or resources that relies on trusted and shared information [12], blockchain provides a platform in which interacting supply chain parties can transparently and precisely interact with each other (i.e., through the definition of coded contracts), facilitating the formation and coordination of service systems.

In this paper, we take a service-oriented perspective and propose a Blockchain-Oriented Service Modeling (BOSM) approach to facilitate the design and development of blockchain-enabled SCQIS. Our approach presents a visual language for knowledge-level modeling. This approach provides a foundation for the encapsulation, coordination, and integration of services in supply chains to measure, analyze, and continually improve the quality of products, services, and processes. We conduct a case study in a dairy supply chain context to illustrate how the proposed modeling approach can be applied to real-world situations to direct the construction of blockchain-enabled SCQIS.

The major contributions of this research are as follows. (1) We propose a service-oriented modeling approach to support quality inspection in blockchain-enabled supply chain quality management, which brings operational service computing to strategic service management in the SCQM domain. It considers the strategic goals and intentions of partner enterprises. (2) The proposed modeling approach bridges the gap between business services and software services in the context of quality management applications. It enables communication between business people in supply chain management and software professionals in service computing. (3) We extend the service-dominant view and reconceptualize the supply chain as a network of service systems. We classify the enterprises’ resources into operant resources and operand resources, which separates the causes of quality from the carriers of quality to facilitate the analysis and design of blockchain-enabled quality information

systems. (4) We investigate the application of the proposed modeling approach in a dairy supply chain environment, which has significant practical implications.

2. Literature Review

2.1. Supply Chain Quality Management. Quality management in supply chains is widely covered in the operations management and information systems literature. From a supply chain perspective, previous studies often focus on how the contract should be set up to mitigate the moral hazard problem and control supply quality. The supply chain contract mechanisms complementing or supplanting quality inspection often include appraisal, certification, and warranty contracts. For example, Hwang et al. [13] compared the inspection strategy with the certification regime in supply chain quality management. Because an inspection method provides noisy information on a supplier’s quality management efforts, the supplier can be induced to perform unwanted/preemptive inspection. Balachandran and Radhakrishnan [14] examined a warranty/penalty contract between the buyer and the supplier based on information from inspections and external failures. The relationships between product architecture, supply chain performance metrics, and supply chain efficiency are also discussed to address the incentive contracting issue in supply chains [15].

From a manufacturing perspective, quality inspection policies are another important aspect in the quality management literature. Inspections are carried out to measure the goods provided by suppliers based on technical requirements. If the goods meet the technical requirements, they can be put into further steps of processing. In the food safety domain, Starbird and Amanor-Boadu [16] found that the effectiveness of supply chain inspection contracts and traceability depends on the accuracy of the inspection, the cost of failing to inspect, the cost of causing a food-borne illness, and the proportion of these costs paid by the supplier. Note that excessive inspection can lead to incurring higher costs than competitors, whereas inadequate inspection can lead to significant inspection errors and failure in quality assurance. Thus, the research on inspection policy is often framed as a mathematical optimization problem to allocate the inspection resources (testing methods) to different stations in production [17].

From an information systems perspective, acquiring upstream and downstream information is also critical for SCQM since quality decision-making needs to be conducted in the scope of the entire supply chain. Zhu et al. [18] considered the quality improvement decisions in a co-operative supply chain and showed that the buyer’s involvement can have a significant impact on the profits of both parties. Mayer et al. [19] examined the relationship between product inspection and supplier plant inspection and suggested that a buyer’s ability to commit to the intensity of supply inspection is the key to analyzing whether product and plant inspections complement or supplant each other. The rationale is that if the process lies comfortably within the specification limits, most of the product output will conform

to the quality standard. Thus, how to leverage the information in supply chains to develop information systems and support quality inspection is an important direction in supply chain quality management.

Information systems have been used in quality management to support decision-making by collecting and analyzing quality information such as customer requirements, quality goal, product/service design, material inspection, process control, storage, shipment, packaging, and delivery [20, 21]. Naveh and Halevy [22] proposed a framework with three levels for handling quality information, with the aim of improving quality and productivity in an organization: control of the process, evaluation of the process, and organizational assessment. Yeung et al. [23] investigated the existence of different patterns of quality information systems and the relationship between such patterns and organizational performance, identifying four patterns of quality information systems: undeveloped, frame, accommodating, and strategic. McMeekin et al. [24] provided a state-of-the-art review of the information systems applied in food safety management, which showed tremendous research and application opportunities for information systems in quality management.

2.2. Blockchain-Enabled SCQIS. There are two major challenges in designing effective SCQIS, namely, information asymmetry in production processes and measuring product quality, for which blockchain technology provides possible solutions [5]. First, information asymmetry is an important obstacle that hinders the development of SCQIS. Wankhade and Dabade [25] analyzed and validated the existence of quality uncertainty against the backdrop of information asymmetry and found that it is important to measure the quality uncertainty due to both information asymmetry and commensurate revenue loss of the company. Hobbs [26] identified three functions of SCQIS, including ex post reactive systems that allow the trace back of affected products in the event of a contamination problem to minimize social costs, ex post systems that facilitate the allocation of liability, and information systems that provide ex ante quality verification. Although information technologies have reduced information asymmetry, Longo et al. [27] conducted an experimental study showing that the companies participating in a supply chain are less inclined to share data when information is sensitive and partners cannot be fully trusted, while blockchain technology can minimize the negative consequences of information asymmetry over the echelons of a supply chain and discourage companies from any misconduct (e.g., counterfeiting data or low data accuracy). Many recent studies suggest that blockchain technology facilitates companies to directly share data with supply chain partners and thereby reduce information asymmetry [28–30]. Moreover, blockchain can effectively guarantee the security and verifiability of information and provides a solution when the supply chain is under attack [31]. Nevertheless, Chen et al. [6] found that the complexity of information systems integration remains one of the major challenges for current blockchain adoption. In other words,

although the blockchain technology could enable supply chain transparency to reduce information asymmetry, it is a significant undertaking to integrate multiple datasets and platforms from all supply chain partners into the conceptual modeling of blockchain-based systems.

Measuring product quality is complex, as it requires sufficiently validated scales. Quality inspection is a widely adopted practice in SCQIS to ensure that suppliers provide goods of sufficient quality. Decision-making on quality inspection is a knowledge reasoning process that relates to domain-specific knowledge of product and inspection technologies. How to represent and leverage domain knowledge and information is a major challenge in building SCQIS. Traditional modeling methodologies in management science and operations management mainly focus on mathematical modeling and analysis of conflicting goals between supply chain partners, which lack an effective representation mechanism to model the domain-specific knowledge. To fill this gap, Kim [32] proposed measurement ontology and traceability ontology to represent and reason about quality based on enterprise models. He also introduced measurement ontology for semantic web applications, which represents not only units of measurement and quantities but also measurement concepts such as sampling, mean values, and evaluation of quality [33]. Tan et al. [21] proposed a quality information system structure within the WWW-based intranet infrastructure and discussed the role of quality information systems in the e-commerce integrated environment. However, as indicated by Lau et al. [34], there is a shortage of literature on intelligent systems for quality inspection, including the shortage of system infrastructure models synthesizing the nature of quality measurement. As suppliers may update their defrauding methods daily, the inspection capability, inspection errors, and other related parameters are always dynamically changing. The SCQIS, including the knowledge it captured, needs to evolve according to the dynamic and uncertain world. Blockchain brings a new hope for SCQIS that ensures traceability right across nodes to the involved stakeholders in the value chain and ensures product quality to consumers through a specified measurement of product quality. George et al. [35] proposed a restaurant prototype using blockchain that captures data from various stakeholders across the food supply chain, segregates it, and applies the Food Quality Index (FQI) algorithm to measure product quality. The challenges and difficulties of modeling quality inspection in SCQM require a modeling approach that will overcome the limitations of traditional modeling methodologies and can connect knowledge representation with reasoning mechanisms for decision-making.

2.3. Contemporary Modeling Languages and Techniques for SCQM. Business process modeling and service modeling play a central role in SCQM, and many modeling languages and techniques have been proposed. Essentially, a model is a simplified abstract view of a complex reality, and thus the objective of modeling languages and techniques is to have a representation of some phenomenon to interpret the reality.

Typically, only some aspects of the reality are referred to as a model, and two models of the same phenomenon may be essentially different. This may be due to the differing requirements of the model's end users or due to the modelers' conceptual or esthetic differences and decisions made during the modeling process.

Van der Aalst [36] reviewed business process modeling languages and classified them into three classes: formal languages, conceptual languages, and execution languages. Formal languages, such as Petri Net, are languages with unambiguous semantics and allow for analysis. Conceptual languages are typically informal, do not have well-defined semantics, and do not allow for analysis. Examples of conceptual languages for business process modeling include BPMN (Business Process Modeling Notation), EPCs (event-driven process chains), and UML activity diagrams. Execution languages, such as BPEL (Business Process Execution Language), are concerned with implementation details and are executable for specifying actions within a business process.

Due to the rigorous semantics (making it impossible to leave things intentionally vague) and low-level nature, business users in practice often have problems using formal languages or execution languages and, therefore, typically prefer to use higher-level languages, that is, conceptual languages [36]. BPMN which is commonly used as a representative conceptual language for business process modeling is considered the state-of-the-art in the field and is an industry standard maintained by Object Management Group (see <https://www.omg.org>). BPMN is commonly used as the basis for business process representation, simulation, and automation, which are important in the contemporary service-oriented architectures common in information technology. The BPMN diagram has been designed for ease of use and understanding, offering a very complex expressive model of business processes. BPMN is a complex language that undergoes constant revisions and extensions. It contains a larger set of constructs in contrast to competing languages and offers a multitude of options for conceptual modeling.

Goal-oriented business process modeling was identified as one of the most important issues in driving business processes towards their goals [37]. It aims to extend traditional business process modeling that addresses the "how" of the business process, which is concerned with efficient execution, to also include the "why" to ensure the effectiveness of business processes [37]. Goal orientation is often regarded as an aspect of an individual's motivation that describes the goals they choose and the methods used to pursue those goals. The goal-oriented view of business process engineering dictates that business goals are the driving force for structuring and evaluating business processes [37]. The i^* framework [38], originating in the field of requirements engineering, provides the best compromise in the field of goal-oriented process modeling [37] as it allows for complex goal classification structures according to goal types and facilitates the modeling of logical, causal, and influencing relationships between goals and business processes.

Nowadays, ontologies and semantic web have been widely adopted to represent services and business processes [39]. A form of ontology represents a common understanding of a domain or domains, including a shared vocabulary and the types or concepts of objects and their attributes and relationships existing in specific fields [40]. In the definition of service-oriented modeling, several existing international standards define ontologies, models, and metamodels to describe evaluated services, including service-oriented architecture modeling language (SoaML), SOA Reference Model (SOA RM), SOA Reference Architecture (SOA RA), SOA Ontology (SOAO), and Web Services Architecture (WSA). Based on ontology representation, a semantic web is not an independent web, but rather it is an extension of the current web, in which information is given a clear meaning so that computers and people can work together better [41]. Based on ontologies, it can understand words and concepts but also the logical relationship between them, which can make communication more efficient and valuable. The main goals of semantic web can be summarized as follows: allowing software agents to automatically obtain information, integrating content from different sources, optimizing search, and realizing trust on the web. Using a semantic web means adopting a brand-new data description and retrieval paradigm [42]. The semantic web concept introduces the use of ontology to construct information in machine-readable format, and it also improves the clarity of understanding difference information [43]. Now there are many languages that can realize semantic description, such as RDF (Resource Description Framework), RDFS (RDF Schema), OWL (Web Ontology Language), and WSMO (Web Service Modeling Ontology) [44].

There are three major problems in applying the existing modeling languages: (1) the complexity of the modeling languages, which makes it costly to teach business users the existing model notations to deal with a particular business scenario [45]; (2) the ontological deficiencies of the modeling languages, which include construct deficit, construct redundancy, construct overload, and construct excess [46]; and (3) the conceptual mismatch between the design and the execution of modeling languages. There is a lack of semantics in conceptual modeling languages, making it impossible to directly execute them [36]. On the other hand, there is a conceptual mismatch between the mapping of conceptual languages and execution languages [47]. These three problems also pose challenges in applying modeling languages and techniques to supply chain quality management, which motivates us to propose a modeling approach for the service-oriented analysis and design of supply chain quality information systems. The modeling approach is based on the extension and simplification of the aforementioned modeling languages and techniques that is simple enough for business users to easily understand while expressive enough to represent and solve the supply chain quality inspection problem and, furthermore, executable to easily implement the quality information system.

3. Motivational Context: A Dairy Supply Chain

To facilitate the discussion in the paper, we put this research in the context of a food supply chain, specifically a dairy product supply chain. Food production is an application domain with high quality requirements, which fits the purpose of our proposed approach.

As shown in Figure 1, the stakeholders along the dairy supply chain include raw milk suppliers, a dairy firm, and end consumers. In dairy product production, the dairy firm often uses HACCP (hazard analysis critical control point) systems to control food safety. The HACCP system is implemented within the dairy firm to test products at critical control points, such as the reception of raw milk, storage in silo tanks, clarification, separation, standardization, pasteurization, and homogenization. Blockchain technology provides an efficient way to track items throughout the supply chain. However, the raw milk suppliers, who control raw milk production, may have different interests from dairy firms. Business process modeling is needed to leverage different stakeholders' available information for quality control. In this paper, we propose a modeling framework that can support the analysis and design of such blockchain-enabled SCQIS.

4. A Service-Oriented Modeling Framework

We propose that effective information system building for blockchain-enabled SCQM should incorporate institutional analysis and adopt a service-dominant business strategy to guide the service-oriented IT modeling. The service-dominant logic offers a different view from the traditional good-dominant logic to model blockchain-enabled SCQIS [48]. Prior studies have suggested that blockchain technology enables the formation and coordination of a service system, particularly in a supply chain context [10, 49]. In this section, we propose a Blockchain-Oriented Service Modeling (BOSM) approach for blockchain-enabled SCQIS.

4.1. Modeling Guidelines. The concept of service and service-oriented modeling has shifted since Lusch and Vargo [48] introduced service-dominant logic. They defined a service as “the application of specialized competences (knowledge and skills) through deeds, processes, and performances for the benefit of another entity or the entity itself.” A service-dominant view is inherently a resource-based view of the firm that emphasizes the strategic value of a firm's skill and cultural competencies [50], and it extends the resource-based view by further differentiating operand resources (those on which an act or operation is performed) and operant resources (those that act on other resources) [51]. It shifted the thinking of value from operand resources—usually tangible, static resources—to operant resources—usually intangible, dynamic resources. It is also aligned with the service-oriented architecture developed in information technology [52].

Based on the service-dominant view, we re-conceptualize the supply chain as a network of service systems, each

representing a role with distinct resources. Supply chain partners exchange operand resources to acquire services of operant resources, and the blockchain records the exchange of operand resources. Each service has an effect that will lead to the achievement of a goal. In other words, each service exchange takes place because one entity relies on another entity's service to achieve their goal. Operant resources, such as manufacturing skills and knowledge [53], are the focus for service. Technology, including SCQIS and blockchain, can be conceptualized as operant resources that are capable of acting on other resources to create values [54]. The application of operant resources in providing services is associated with several operand resources that can be tangibly recorded in SCQIS, including tangible products (raw materials, prototypes), procedure specifications of service execution, inputs or outputs of the service, the plant, and conditions of service provision. The applications of knowledge and skills in providing service may have their constraints. For example, specific manufacturing plants and conditions may be required to accomplish the provision of a specific service. The constraints of service provision should be modeled as an operand resource. Overall, we derive four design guidelines for blockchain-enabled SCQIS following a service-dominant view (Figure 2).

4.2. A Service Model for SCQIS Requirement Modeling. In this study, we develop a conceptual model to represent the service-oriented modeling in a blockchain-enabled supply chain context. Figure 3 shows the visual representations we give to these concepts and relations. In our modeling framework, a service is built on four classes of concepts, actor, goal, resources, and tasks, and the relationship between the concepts. Figure 4 shows a portion of a simplified ontology for service provision in a supply chain. Because of the complexity of this figure, many links, such as Part_of, Instance_of, Object_property and Datatype_property, have been omitted. The ontology is produced at three levels: metaclass level, domain level, and instance level. The entities at the instance level correspond to the instances of domain classes, while the domain classes inherit attributes from the metaclass level. As OWL has flexible modeling ability and powerful knowledge reasoning ability, it will work well in our context involving many supply chain participants with varied knowledge and can be used as our ontology implementation language.

In light of the service-dominant view and guidelines we discussed above, we differentiate resources to operant resource and operand resource in our visual language. Furthermore, a blockchain-enabled SCQIS may be concerned with functional requirements (specific functions or services of the service) and nonfunctional requirements (criteria or quality attributes of the service). Since nonfunctional requirements are usually stated informally and may have conflicts, Mylopoulos et al. [55] proposed the concept of the soft goal for modeling and analyzing nonfunctional requirements. In this research, we also differentiate soft goal and hard goal in our visual language. The service components in our visual language are as follows:

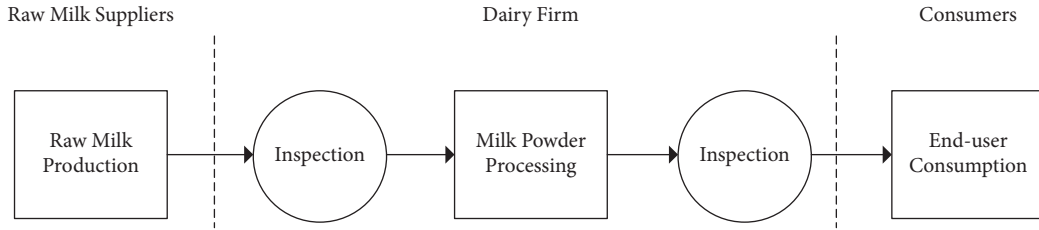


FIGURE 1: Case illustration.

Guideline 1: Supply chain entities should be modeled as roles of a service provider possessing operant resources and operand resources. Blockchain-enabled SCQIS should be modeled as entities providing services to fulfill SCQM goals. Entities provide their service in exchange for other entities' service to fulfill their goals.

Guideline 2: Service is the application of operant resources to fulfill an achievable goal. Supply chain entities' skills and knowledge, such as suppliers' supply, manufacturers' production and inspection, and consumers' product review, should be modeled as operant resources. Blockchain-enabled SCQIS includes several operant resources, such as distributed ledger and inspection service.

Guideline 3: Operant resources are associated with several operand resources. Goods and production materials, such as inputs and outputs in a manufacturing process, procedure specifications, plants and conditions, should be modeled as operand resources. Operand resources can be recorded in a distributed ledger to track and trace the application of operant resources.

Guideline 4: The exchange goals can be packaged into smart contracts. The fulfillment of decomposed goals can be used to measure service quality.

FIGURE 2: Design guidelines according to the service-dominant view.

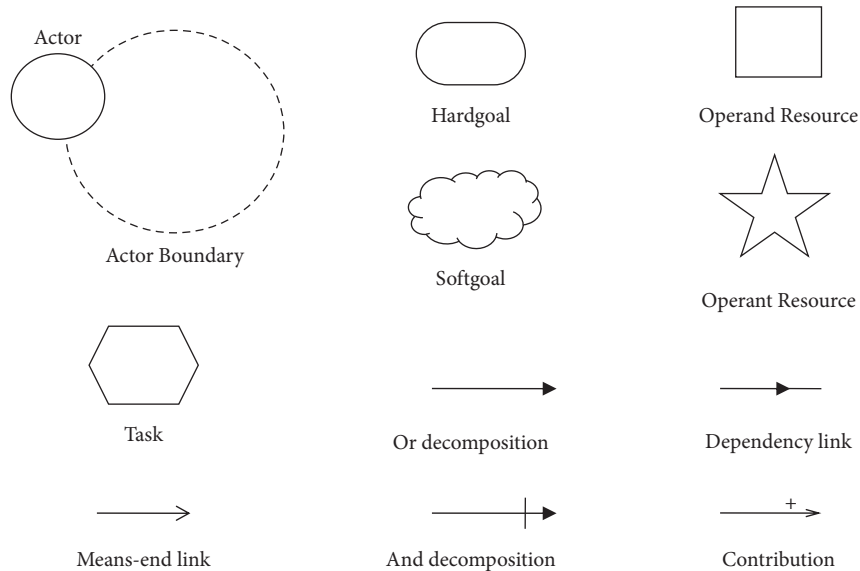


FIGURE 3: Legend for knowledge-level modeling language.

- (1) Actor models a service provider that has strategic goals, possesses resources, and intentionally acts within the service setting. An actor can be a physical, social, or software agent that provides a type of service. In our dairy supply chain example, the actors may include supplier, manufacturer, retailer, consumer, and blockchain-enabled SCQIS.
- (2) Goal represents an actor's strategic interests. One actor may rely on another actor to fulfill its goal. For example, a manufacturer relies on suppliers for a

good raw material supply. Goal is classified into two categories: hard goal and soft goal. The hard goals can be checked through verification techniques. Soft goals have no clear-cut criteria to check whether they are satisfied or not.

- (3) Resources represent the belongings an actor possesses. Resources are further classified into operant resources and operand resources. Operant resources can act on or in concert with other resources to create value, such as manufacturing skills. Operand

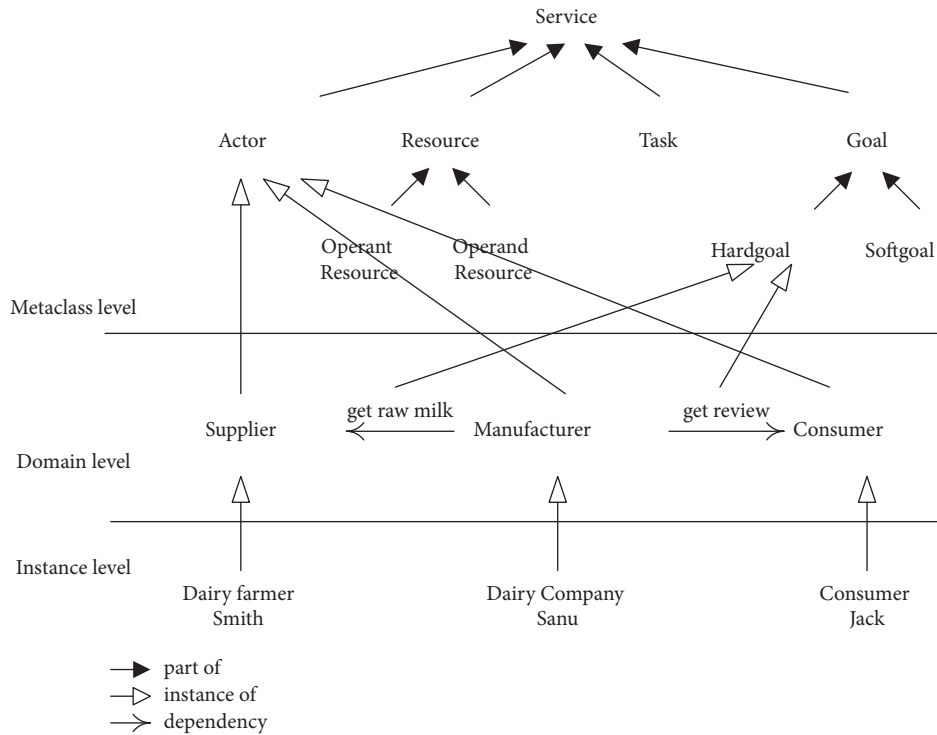


FIGURE 4: A simplified ontology for SCQM.

resources are resources on which an operation or act is performed to produce an effect, such as goods at different production stages, including the raw material and final product.

- (4) Task is an activity that needs to be performed by the actor. The execution of a task can be a means to satisfy a hard goal. A task may be carried out under some constraints. In supply chains, the payment, production, and delivery activities can be modeled as a task. A task may be decomposed into subtasks.

The relationships between the service components consist of traditional association relationships and strategic relationships. Traditional relationships include the Is-part-of association, Is-A association, and AND-OR decomposition. The strategic relationships are specifically adopted from i^* modeling [56], as follows:

- (1) Contribution relationship describes how one goal (soft goal or hard goal) contributes to the achievement of another goal. Contributions can be either negative or positive. A positive (negative) contribution means that a goal is helpful (harmful) to the achievement of another goal.
- (2) Means-ends relationship shows how the goal (i.e., end) can be fulfilled by the series of tasks (i.e., means) through the manipulation of resources. A goal may be satisfied in several possible ways (means).

- (3) Dependency relationship, between two actors, or actors and goals, indicates that one actor depends on the other in order to attain some hard goal. The former actor is called the depender, while the latter is called the dependee.

- (4) Configuration relationship, between an operant resource and operand resources, represents how an operant resource is configured by some operand resources as inputs, outputs, procedure, and constraints.

To further explain the service components and their relationships defined in our visual language, we illustrate them in Figure 5. As shown in Figure 5, actors play a central role in our modeling framework. The goal and its subclasses, soft goal and hard goal, are desired by actors. Actors are connected to each other through the dependency relationship, which is a quaternary relationship involving depender, dependee, and dependum (i.e., a hard goal). Actors process the resources to conduct the tasks. Goals (of the actors) can be analyzed to clarify their related decomposition, contribution, and means-ends relations. Contribution is a ternary relationship between an actor and two goals, which identifies that one goal can contribute positively or negatively towards the fulfillment of another goal. Means-ends relation is a ternary relationship defined among an operant resource (the constraints), a goal (the end), and tasks (the means),

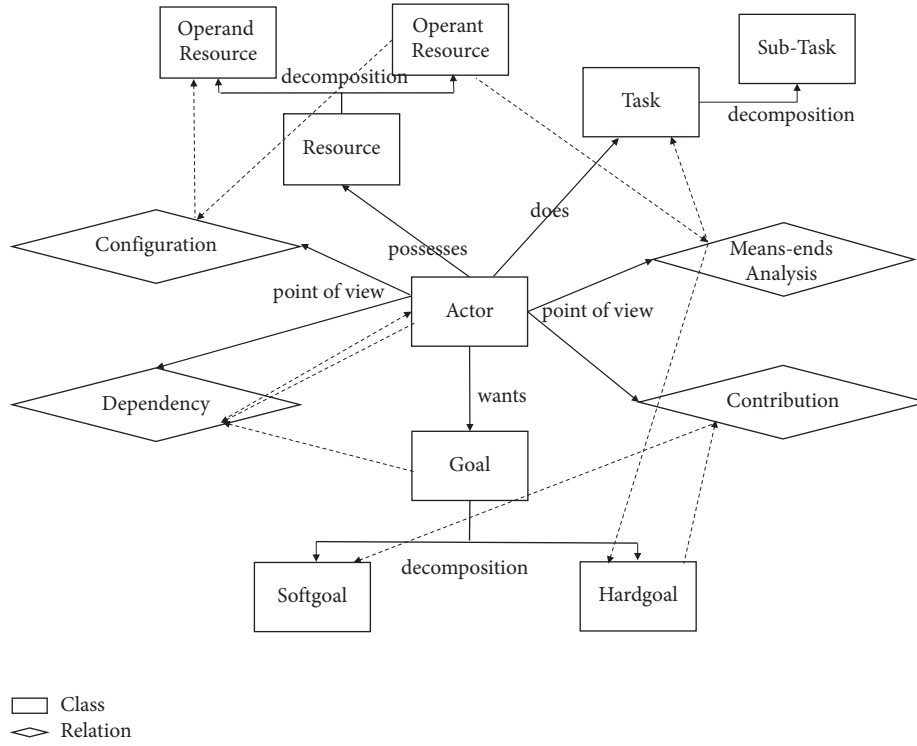


FIGURE 5: Relationships between the service components.

showing that the actors can conduct tasks with some resources to attain the goal. Configuration is a ternary relationship between an actor, an operant resource, and some operand resources, showing that the operand resources needed to apply a specific operant resource.

4.3. Conceptual Modeling for Blockchain-Enabled SCQIS. With the defined service model, we are able to conceptually model the facts and relationships between service providers in a blockchain-enabled supply chain. The major procedures using our proposed service modeling for knowledge-level modeling contain four steps: (1) actor and goal modeling, (2) service and resource modeling, (3) goal and resource dependence modeling, and (4) blockchain-enabled SCQIS modeling.

First, one needs to identify all the entities participating in SCQM as actors and elaborate each actor's goals. Figure 6 shows examples of a manufacturer and its corresponding goals, in which the dashed circle shows the boundary of each actor. As we can see, the general purpose (soft goal) of manufacturers is to get a qualified supply, which can be decomposed into two soft goals: "trust in the production process" and "trust in the quality inspection."

Definition 1. Actor

An actor is a 5-tuple $\langle a_id, G, S, R, T \rangle$, in which a_id is the unique identifier of the service provider, $G = \{g | g \text{ is a goal in the scenario}\}$, $S = \{s | s \text{ is a service in the scenario}\}$, $R = \{r | r \text{ is a resource in the scenario}\}$, and $T = \{t | t \text{ is a task in the scenario}\}$.

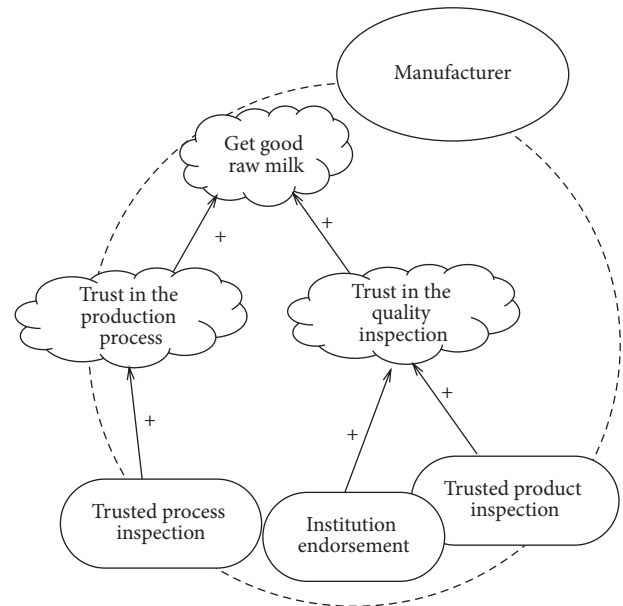


FIGURE 6: Actors and goals.

According to references [56–58], blockchain-enabled SCQIS can be modeled as a set of actors possessing various goals to fulfill. As the goals for blockchain-enabled SCQIS are ambiguous, we only model the supply chain enterprises in this first step. Blockchain-enabled SCQIS will be modeled after the goal exchange phase.

In the goal modeling phrase, we need to detail the decomposition and contribution relationships among goals. Figure 7 further illustrates such analysis, in which the soft

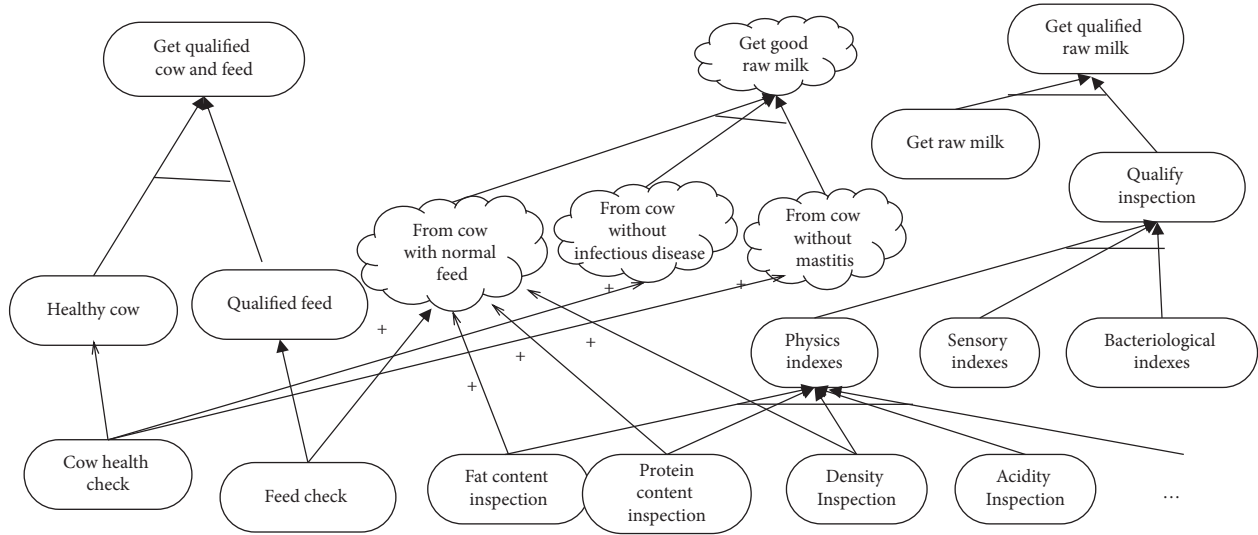


FIGURE 7: An example of goal modeling.

goal of “get good raw milk” is decomposed to the AND-OR soft goals “from cow with good feed,” “from cow without infectious disease,” and “from cow without mastitis.” The soft goals can be further transformed to explicit and achievable goals (hard goal). In Figure 7, the soft goal “from cow with good feed” can be attributed to the contribution of the hard goals “fat inspection,” “protein inspection,” and “density inspection.” The hard goals of “cow health check” and “feed check” can contribute to the soft goal of “get good raw milk.” As such, the hard goal of g_1 “quality inspection” can be decomposed into g_2 “physics indexes,” g_3 “sensory indexes,” and g_4 “bacteriological indexes.” Such an AND composition of hard goal g_1 can be represented as a constraint $c_1(c_1: g_1 \Rightarrow g_2 \wedge g_3)$ meaning that if g_1 exists, then both g_2 and g_3 exist. Here, “ c_1 ” is the unique identifier of this constraint.

Second, in service and resource modeling, we need to depict the resources possessed by each entity, including operant resources and their related operand resources. A service is the application of an operant resource, while operand resources are explicitly documented or tangible and need to be associated with at least one operant resource. Figure 8 shows an example of the operant resource (a service of raw milk supply) associated with four operand resources (milking procedure, cows, feed, and raw milk).

Definition 2. Service

A service is a 5-tuple $s = \langle I, O, C, P, T \rangle$. I and O represent the input and output elements (operand resources or other operant resources) accepted by a particular operation and made available after the operation, respectively. C is the set of conditions (including the availability of operand resources or other operant resources) to invoke the operation. P is the description of the operant resource’s status, state, operation procedures, or other explicit features. T is the set of tasks carried out to provide the service.

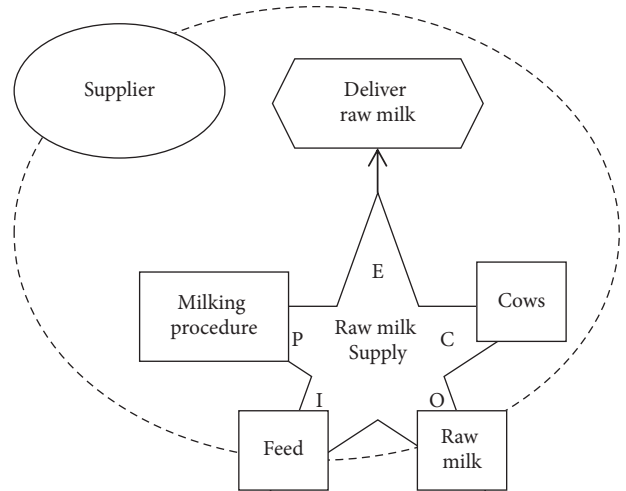


FIGURE 8: Resource modeling.

After identifying an actor’s resources, we are able to model the individual actor’s goal exchange and fulfillment. From a service perspective, multiple actors in a supply chain will exchange services to fulfill those goals. As some goals cannot be fulfilled by the actor, we need to connect different actors’ goals through service exchange modeling. As we can see in Figure 9, the service of raw milk supply can fulfill the goal of “get raw milk,” which should be a manufacturer’s goal. In our framework, we allow actors to exchange hard goals fulfilled by others. Figure 9 shows an example in which the manufacturer exchanges her hard goal “get raw milk” for the supplier’s hard goal “get paid.” The exchange of goals may not be limited to one-to-one relationships. In this step, we began to define entities of blockchain-enabled SCQIS to fulfill the goals from the manufacturer and the supplier. As an example, in Figure 9, the inspection service is defined to fulfill the hard goals of “get cow health check” and “get protein inspection,” and the distributed ledger is defined to “get consensus on cow data.”

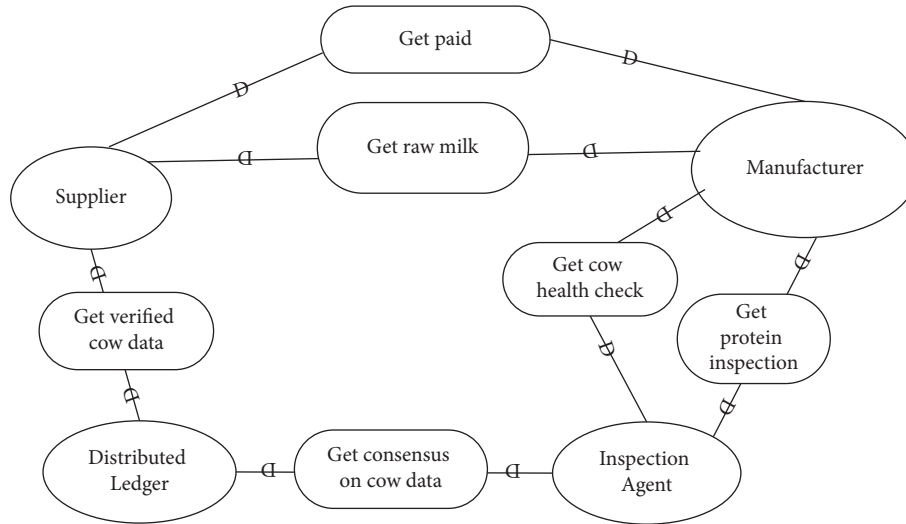


FIGURE 9: Service exchange modeling.

Up to this point, we need to identify tasks to satisfy the hard goal through the application of blockchain-enabled SCQIS. Now, we can use blockchain-enabled SCQIS to get a full conceptual model to depict the requirements and dependencies of supply chain partners with blockchain-enabled SCQIS. Figure 10 shows a part of conceptual modeling of the distributed ledger and inspection service. For instance, the distributed ledger offers consensus service to get consensus on cow data. Specifically, the consensus service takes cow data from the supplier and verification data from other nodes as inputs, and then, using its consensus algorithm (e.g., PBFT) as the processing procedure and peers' endorsement as constraints, the service will offer mutual agreement on cow data as outputs. It is worth noticing that an actor may offer several services in parallel. For example, the inspection agent can offer different services for different inspection requirements, including protein content inspection and fat content inspection. In Figure 10, we show an example of the Kjeldahl method to provide protein content inspection. It takes raw milk samples as inputs and shows nitrogen percentage as outputs, including a processing procedure of digestion, distillation, and titration. The Kjeldahl method measures nitrogen as a proxy of protein in milk, fulfilling the goal of "get protein inspection."

5. Case Study on Modeling a Blockchain-Enabled SCQIS

To illustrate the feasibility of applying our proposed approach, we use it to build a prototype for a dairy supply chain. We develop a conceptual model for the regular product tracing and quality inspection process in the supply chain, which is partially shown in Figure 11. In this figure, we identify that the quality of the milk is related to the milk production process, that is, cows and the feeding process. Previous literature suggested that the inspection of supplies and the inspection of supplier facilities complement each other [19]. Thus, the quality inspection

system needs to identify and record both types of information for decision-making. So when building the quality inspection system, examining the raw milk supply service from the raw milk supplier focuses on the intangible operant resource of "supply" capability, associated with tangible operand resources *I* as feed, *P* as milking procedure, *C* as cows, and *O* as raw milk.

To ensure the quality of the final product, the dairy product manufacturer needs to check the quality of shipped raw milk. The quality inspection includes various examination indexes such as protein content, fat content, and density, all of which are evaluated by testing methods decided by testing policy. With different levels of inspection technologies and capabilities, the raw milk suppliers could have different potential deception intentions to manipulate the product and dupe some examination attributes. For example, adding melamine can dupe the Kjeldahl method for protein content detection. However, it is not feasible for the dairy product manufacturer (i.e., the buyer) to apply every inspection technology to eliminate the deception due to cost. So the manufacturer needs to decide its testing policy that can discourage a supplier's deception while keeping cost manageable. We build a blockchain-enabled SCQIS to facilitate the manufacturer's decision.

The designed blockchain-enabled SCQIS has enabled a flexible inspection in SCQM. As shown in Figure 12, services are captured by the proposed service modeling framework. The effect of flexible inspection is achieved by the service of contract execution, which depends on a service composition of data collection, data recording, data consensus, and quality inspection. A supply contract between the manufacturer and the supplier defines the flexible testing policy to be executed depending on the data stored in the distributed ledger. For instance, when the distributed ledger gets a record of cow data that indicates it is an unhealthy cow, a protein inspection will be carried out. To reach mutual agreement on such data of an unhealthy cow, the distributed ledger service provider provides a service of data recording via peer-to-peer recording in permitted nodes and a service

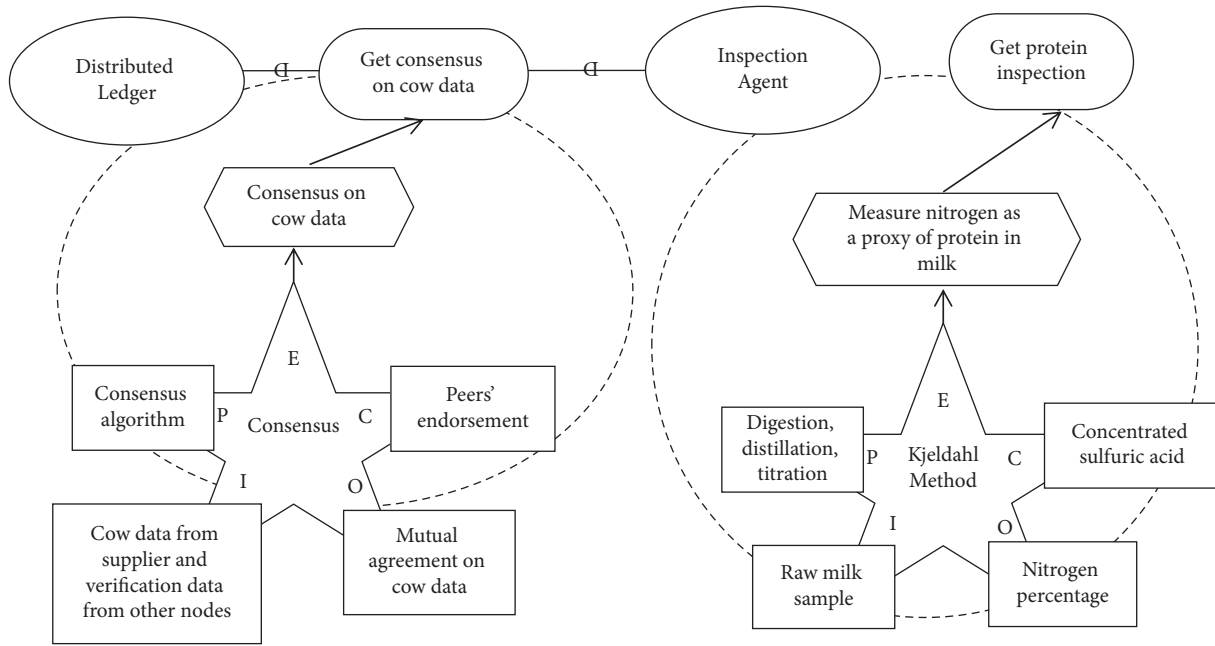


FIGURE 10: Blockchain-enabled SCQIS modeling.

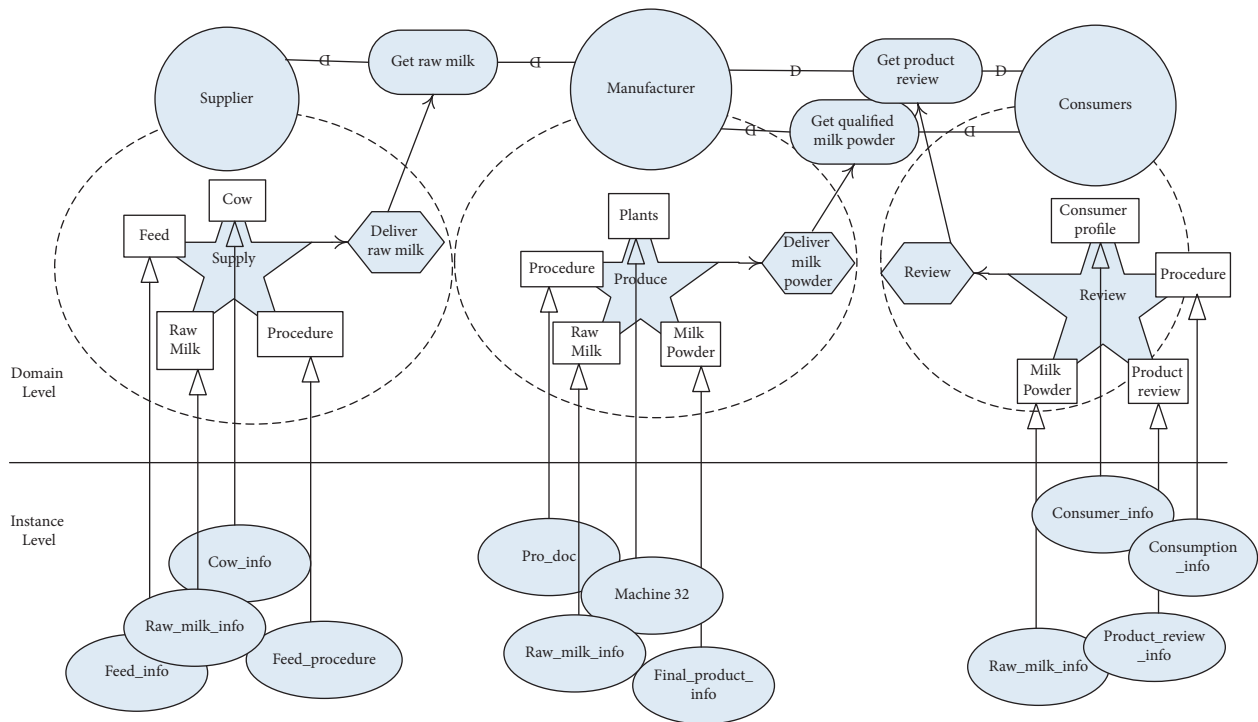


FIGURE 11: Product and process inspection.

of data consensus by the peer endorsements. The service of data recording, in turn, receives data from the IoT data collecting service.

6. Discussion

6.1. *Transforming BOSM into BPMN.* As business process models are important for information systems design, we discuss how to transform the proposed model into business

process models in this section. Transforming to business process models can help to quickly develop the business process of SCQIS applications and provide a lens through which we can examine the practical significance and feasibility of the proposed model.

First, in the BOSM, participants, tasks, and other elements can be mapped to BPMN. The participants defined in the supply chain, including suppliers, manufacturers, and consumers, can be mapped into actors (represented as pools)

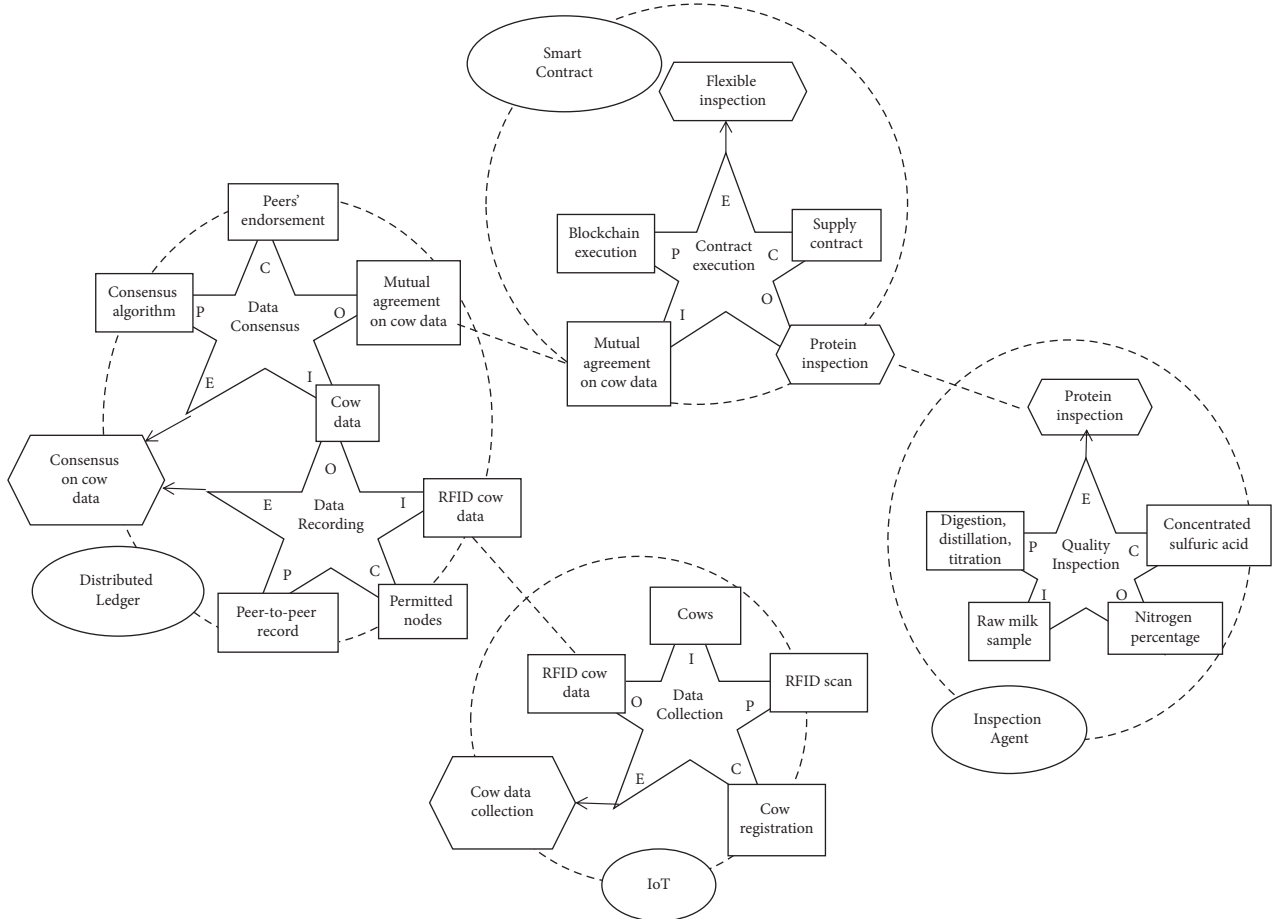


FIGURE 12: Services in a blockchain-enabled SCQIS.

in the BPMN diagram. However, the pools in BPMN will not automatically become participants, and they all are needed to be verified in the blockchain before they can become nodes in it. Then the behaviors in the BOSM can be mapped into tasks in the BPMN, such as feeding cows and producing milk. All these behaviors will be migrated to a blockchain platform through a consensus mechanism, and relevant information will be shared by all nodes. In addition, the quality inspection in the BOSM can be mapped into a gateway in the BPMN; only qualified products can enter the next round of the supply chain.

In Figure 13, we transform the service of dairy product supply in the BOSM to a BPMN model. As we can see from this BPMN model, all behaviors of the supplier are recorded and uploaded to a blockchain, including cow information, feeding information, production information, and transportation information. When the raw milk is delivered to the manufacturer, the manufacturer can obtain all the information of the raw milk production process from the blockchain. In addition to the inspection of raw milk products themselves, other operand resources can be inspected. This method can not only better detect product problems but also effectively discover the causes of product problems.

In the proposed BOSM approach, we also regard the experiences and opinions of consumers as important. As

shown in Figure 14, transforming to the BPMN model shows that after a consumer buys a product, he can get all the product information through the blockchain, ensuring that the finally obtained information on dairy products is authentic and reliable. In addition, consumers can upload their feedback on products to the supply chain for manufacturers' reference, which will enable manufacturers to improve their products and services.

6.2. Comparing Blockchain-Enabled SCQIS with Traditional SCQIS. By transforming from BOSM to BPMN to illustrate the business processes of blockchain-enabled SCQIS, we can find the differences between blockchain-enabled SCQIS and traditional SCQIS in terms of business process implementation. Traditional SCQIS builds an internal information tracing system according to requirements of a central enterprise, mostly with traditional tracing technologies such as bar code, two-dimensional code, or radio frequency identification (RFID), and uploads the tracing data into enterprise data systems [59]. Each enterprise along supply chains has its own database. In blockchain-enabled SCQIS, all enterprises jointly use blockchain as a platform for data sharing and update process data in supply chains in real time through other collaborative technologies such as the Internet of Things, leading to collaboration among enterprises.

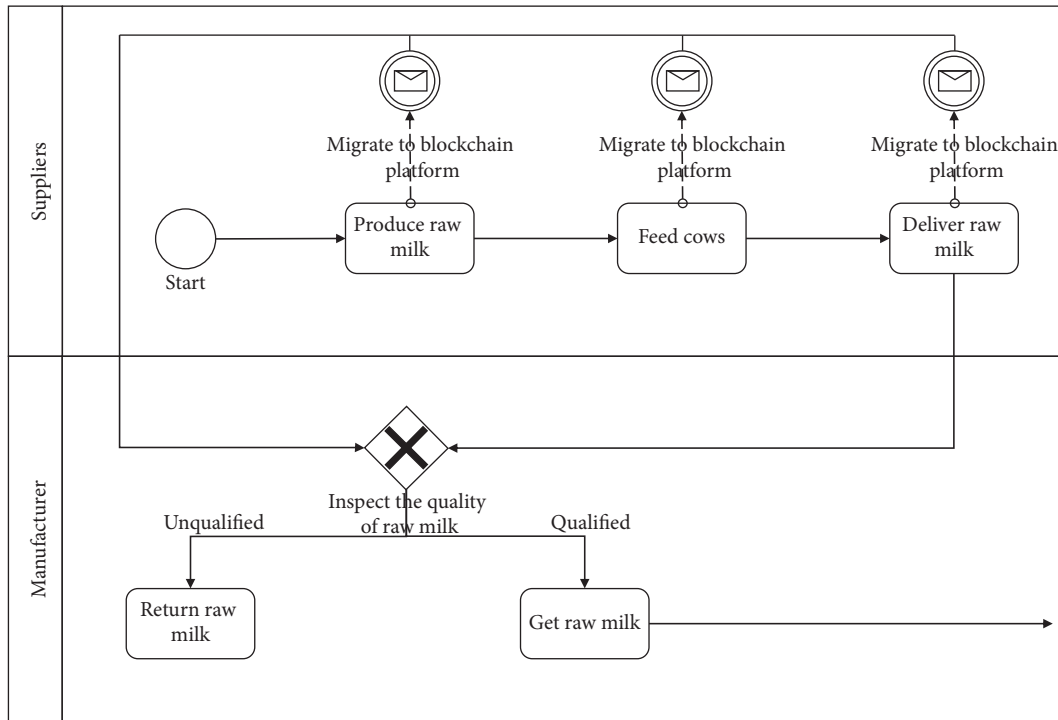


FIGURE 13: Dairy supply business process model.

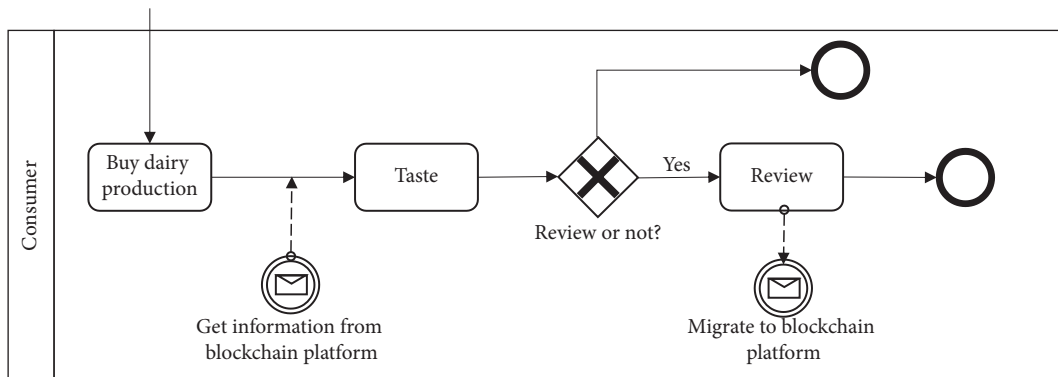


FIGURE 14: Consumers in the business process model.

Blockchain-enabled SCQIS can effectively improve the traceability of SCQIS and provide better visibility and higher efficiency by creating records in the supply chain grids [60]. Taking dairy supply chains as an example, the interests and needs of participants in the supply chain are different, and some participants may modify their process data privately, resulting in data tempering problems in supply chain management. For example, raw milk suppliers may modify the production time of raw milk in order to sell expired milk. In contrast, with the introduction of blockchain technology, the decentralization of data recording can improve trust-building among raw milk suppliers, dairy companies, and consumers, can minimize negative consequences of information asymmetry along supply chains, and can prevent improper behaviors of various stakeholders (such as falsifying product quality data). Therefore, blockchain can avoid

the vulnerability of centralized nodes in establishing trust [61]. In addition, blockchain technology can bring certain security to SCQIS. According to a survey report [62], small organizations are often targeted by network attacks because of their size. Traditional SCQIS relies on communication and coordination at the same time, which may easily attract network attacks on the SCQIS, leading to a fragile situation [59]. For example, SCQIS may face the risk of counterfeit tag attacks and counterfeit product attacks. In contrast, with the introduction of blockchain technology, the blockchain-enabled SCQIS can have certain resistance capability in the face of such attacks [31].

Blockchain-enabled SCQIS has its limitations. Blockchain requires considerable computing power [63]. A blockchain-enabled SCQIS uses a lot of computer energy because it is necessary to keep all nodes updated from time to

time to ensure the consensus of traceability. During the transaction process, every transaction needs to be signed by a cryptographic scheme, which will also bring high energy consumption. Without enough computing power, ordinary users may not be able to participate in the blockchain network, which further affects the application of blockchain in SCQIS. In addition, the integration of blockchain with existing systems may bring great challenges to actual business, as not all SCQIS can perfectly adapt to blockchain [64].

6.3. Comparing the BOSM Approach with Traditional Service-Oriented Modeling Approaches. The BOSM approach aims to model services in blockchain-enabled SCQIS, while traditional service-oriented modeling approaches do not take the context of supply chains and features of blockchain into consideration. In comparison with commonly used methods in service-oriented modeling approaches, such as UML [65], the proposed BOSM approach has advantages in the following aspects.

From a semantic perspective, BOSM distinguishes operand resources from operand resources, while traditional service-oriented modeling approaches, such as UML, do not possess such capabilities. Compared with UML, the BOSM method enables us to have a clearer representation of what the SCQIS offers and the conditions to achieve goals. In the knowledge-level modeling, we adopted the goal-oriented modeling technique to focus on the self-interest characteristics of supply chain participants and studied different behaviors under different knowledge and goals. This modeling method allows us to focus not only on the product itself but also on product manufacturing processes and the motivation of major participants, with better explanations for their behaviors.

From a grammar perspective, the BOSM approach reduces complexity of the modeling language, makes it easier for business users to understand, and can effectively reduce the cost of communication between business users to deal with specific business scenarios. In contrast, UML lacks grammatical elements, and its sentences are not coherent [66]. The BOSM approach provides a series of models and operations with graphical explanations, simplifies the modeling language at the knowledge level, and reduces business user's learning costs.

From an implementation perspective, in the process of UML modeling, there are repetitive and useless model elements in SCQM scenarios, which will cause ontology defects in the process of ontology building, including structural defects, structural redundancy, structural overload, and structural excess [46]. In the process of ontology construction, BOSM builds services on four kinds of concepts, that is, participants, goals, resources and tasks, and the relationship between concepts. This method ensures integrity and practicability in the process of ontology construction in SCQM scenarios.

7. Conclusion

SCQM faces several challenges due to the self-interested and distributed nature of supply chains, such as the information asymmetry that exists in the production process and the

difficulty in quality measurement. Blockchain-enabled SCQIS holds the potential to alleviate such concerns for SCQM. However, the modeling techniques for developing blockchain-enabled SCQIS have not been fully investigated in literature. This research provides a novel service-oriented modeling framework to fill this gap.

Our proposed BOSM modeling approach enables us to model what the system does and how it does it from both a service management perspective and a service computing perspective. In the knowledge-level modeling process, we follow a service-dominant view and develop a visual language to capture the possible activities of partners. We conducted a case study and developed a prototype in the context of quality inspection in a dairy supply chain to illustrate how the proposed modeling approach is applied to real-world situations.

Our proposed modeling framework under the service-dominant view has several advantages as compared with other modeling perspectives. First, the modeling of operand resources separates the manufacturing process—the cause of quality—from products—the carrier of quality. This separation facilitates the detection of product defects and the inspection of the reason for these defects. Second, the modeling of services' interaction and cocreation of value with supply chain partners encapsulates the system-based view of the blockchain-enabled supply chain. The modeling approach characterizes the supply chain entities with different motivations or interests in acquiring the benefits of specialized competences of others. This perspective offers an instrument to analyze the different interests of supply chain partners as well as the competences they can offer, which is a key element for coordination in a supply chain.

Data Availability

This paper includes a case study in the manuscript.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

National Science Foundation of China (No. 72171115).

References

- [1] C. J. Robinson and M. K. Malhotra, "Defining the concept of supply chain quality management and its relevance to academic and industrial practice," *International Journal of Production Economics*, vol. 96, no. 3, pp. 315–337, 2005.
- [2] S. T. Foster, C. Wallin, and J. Ogden, "Towards a better understanding of supply chain quality management practices," *International Journal of Production Research*, vol. 49, no. 8, pp. 2285–2300, 2011.
- [3] V. H. Lee, P. Y. Foo, G. W. H. Tan, K. B. Ooi, and A. Sohal, "Supply chain quality management for product innovation performance: insights from small and medium-sized manufacturing enterprises," *Industrial Management & Data Systems*, vol. 121, no. 10, pp. 2118–2142, 2021.

- [4] J. Li, A. Maiti, M. Springer, and T. Gray, "Blockchain for supply chain quality management: challenges and opportunities in context of open manufacturing and industrial internet of things," *International Journal of Computer Integrated Manufacturing*, vol. 33, no. 12, pp. 1321–1355, 2020.
- [5] S. Chen, R. Shi, Z. Ren, J. Yan, Y. Shi, and J. Zhang, "A blockchain-based supply chain quality management framework," in *Proceedings of the 2017 IEEE 14th International Conference on E-Business Engineering (ICEBE)*, Shanghai, China, November 2017.
- [6] S. Chen, X. Liu, J. Yan, G. Hu, and Y. Shi, "Processes, benefits, and challenges for adoption of blockchain technologies in food supply chains: a thematic analysis," *Information Systems and e-Business Management*, vol. 19, no. 3, pp. 909–935, 2021.
- [7] P. Helo and Y. Hao, "Blockchains in operations and supply chains: a model and reference implementation," *Computers & Industrial Engineering*, vol. 136, pp. 242–251, 2019.
- [8] Y. Wang, J. H. Han, and P. Beynon-Davies, "Understanding blockchain technology for future supply chains: a systematic literature review and research agenda," *Supply Chain Management: International Journal*, vol. 24, no. 1, pp. 62–84, 2019.
- [9] M. Shoaib, M. K. Lim, and C. Wang, "An integrated framework to prioritize blockchain-based supply chain success factors," *Industrial Management & Data Systems*, vol. 120, no. 11, pp. 2103–2131, 2020.
- [10] S. Porru, A. Pinna, M. Marchesi, and R. Tonelli, "Blockchain-oriented software engineering: challenges and new directions," in *Proceedings of the 2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)*, Buenos Aires, Argentina, May 2017.
- [11] S. Seebacher and R. Schüritz, "Blockchain technology as an enabler of service systems: a structured literature review," *Exploring Services Science Springer International Publishing*, vol. 279, pp. 12–23, 2017.
- [12] J. Spohrer, P. P. Maglio, J. Bailey, and D. Gruhl, "Steps toward a science of service systems," *Computer*, vol. 40, no. 1, pp. 71–77, 2007.
- [13] I. Hwang, S. Radhakrishnan, and L. N. Su, "Vendor certification and appraisal: implications for supplier quality," *Management Science*, vol. 52, no. 10, pp. 1472–1482, 2006.
- [14] K. R. Balachandran and S. Radhakrishnan, "Quality implications of warranties in a supply chain," *Management Science*, vol. 51, no. 8, pp. 1266–1277, 2005.
- [15] S. Baiman, P. E. Fischer, and M. V. Rajan, "Information, contracting, and quality costs," *Management Science*, vol. 46, no. 6, pp. 776–789, 2000.
- [16] S. A. Starbird and V. Amanor-Boadu, "Do inspection and traceability provide incentives for food safety?" *Journal of Agricultural and Resource Economic*, vol. 1, 2006.
- [17] P. F. Zantek, G. P. Wright, and R. D. Plante, "Process and product improvement in manufacturing systems with correlated stages," *Management Science*, vol. 48, no. 5, pp. 591–606, 2002.
- [18] K. Zhu, R. Q. Zhang, and F. Tsung, "Pushing quality improvement along supply chains," *Management Science*, vol. 53, no. 3, pp. 421–436, 2007.
- [19] K. J. Mayer, J. A. Nickerson, H. Owan, J. A. Nickerson, and J. M. Olin, "Are supply and plant inspections complements or substitutes? A strategic and operational assessment of inspection practices in biotechnology," *Management Science*, vol. 50, no. 8, pp. 1064–1081, 2004.
- [20] M. M. Siddh, G. Soni, R. Jain, M. K. Sharma, and V. Yadav, "Agri-fresh food supply chain quality (AFSCQ): a literature review," *Industrial Management & Data Systems*, vol. 117, no. 9, pp. 2015–2044, 2017.
- [21] B. Tan, C. Lin, and H. C. Hung, "An ISO 9001:2000 quality information system in e-commerce environment," *Industrial Management & Data Systems*, vol. 103, no. 9, pp. 666–676, 2003.
- [22] E. Naveh and A. Halevy, "A hierarchical framework for a quality information system," *Total Quality Management*, vol. 11, no. 1, pp. 87–111, 2000.
- [23] A. C. L. Yeung, L. Y. Chan, and T. S. Lee, "An empirical taxonomy for quality management systems: a study of the Hong Kong electronics industry," *Journal of Operations Management*, vol. 21, no. 1, pp. 45–62, 2003.
- [24] T. A. McMeekin, J. Baranyi, J. Bowman et al., "Information systems in food safety management," *International Journal of Food Microbiology*, vol. 112, no. 3, pp. 181–194, 2006.
- [25] L. Wankhade and B. M. Dabade, "Analysis of quality uncertainty due to information asymmetry," *International Journal of Quality & Reliability Management*, vol. 23, no. 2, pp. 230–241, 2006.
- [26] J. E. Hobbs, "Information asymmetry and the role of traceability systems," *Agribusiness*, vol. 20, no. 4, pp. 397–415, 2004.
- [27] F. Longo, L. Nicoletti, A. Padovano, G. d'Atri, and M. Forte, "Blockchain-enabled supply chain: an experimental study," *Computers & Industrial Engineering*, vol. 136, pp. 57–69, 2019.
- [28] S. van Engelenburg, M. Janssen, and B. Klievink, "A blockchain architecture for reducing the bullwhip effect," *Lecture Notes in Business Information Processing*, vol. 319, pp. 69–82, 2018.
- [29] P. K. Wan, L. Huang, and H. Holtskog, "Blockchain-enabled information sharing within a supply chain: a systematic literature review," *IEEE Access*, vol. 8, pp. 49645–49656, 2020.
- [30] Z. Wang, T. Wang, H. Hu, J. Gong, X. Ren, and Q. Xiao, "Blockchain-based framework for improving supply chain traceability and information sharing in precast construction," *Automation in Construction*, vol. 111, p. 103063, 2020.
- [31] Z. Liu and Z. Li, "A blockchain-based framework of cross-border e-commerce supply chain," *International Journal of Information Management*, vol. 52, p. 102059, 2020.
- [32] H. M. Kim, "Representing and reasoning about quality using enterprise models," in *Bibliothèque Nationale Du Canada*, Canada, 1999.
- [33] H. M. Kim, A. Sengupta, M. S. Fox, and M. Dalkilic, "A measurement ontology generalizable for emerging domain applications on the semantic web," *Journal of Database Management*, vol. 18, no. 1, pp. 20–42, 2007.
- [34] H. C. W. Lau, G. T. S. Ho, K. F. Chu, W. Ho, and C. K. M. Lee, "Development of an intelligent quality management system using fuzzy association rules," *Expert Systems with Applications*, vol. 36, no. 2, pp. 1801–1815, 2009.
- [35] R. V. George, H. O. Harsh, P. Ray, and A. K. Babu, "Food quality traceability prototype for restaurants using blockchain and food quality data index," *Journal of Cleaner Production*, vol. 240, p. 118021, 2019.
- [36] W. M. P. van der Aalst, "Business Process Management: A Comprehensive Survey," *ISRN Software Engineering*, vol. 2013, pp. 1–37, Article ID 507984, 2013.
- [37] D. Neiger and L. Churilov, "Goal-oriented business process modeling with EPCs and value-focused thinking," *Lecture Notes in Computer Science*, vol. 3080, pp. 98–115, 2004.
- [38] E. Yu and J. Mylopoulos, "Using goals, rules, and methods to support reasoning in business process reengineering," in *Proceedings of the Twenty-Seventh Hawaii International*

- Conference on System Sciences*, Wailea, HI, USA, January 1994.
- [39] L. F. Lin, W. Y. Zhang, Y. C. Lou, C. Y. Chu, and M. Cai, "Developing manufacturing ontologies for knowledge reuse in distributed manufacturing environment," *International Journal of Production Research*, vol. 49, no. 2, pp. 343–359, 2011.
- [40] R. Studer, V. R. Benjamins, and D. Fensel, "Knowledge engineering: principles and methods," *Data & Knowledge Engineering*, vol. 25, no. 1-2, pp. 161–197, 1998.
- [41] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," *Scientific American*, vol. 284, no. 5, pp. 34–43, 2001.
- [42] I. Torre, "Adaptive systems in the era of the semantic and social web, a survey," *User Modeling and User-Adapted Interaction*, vol. 19, no. 5, pp. 433–486, 2009.
- [43] S. Singh, S. Ghosh, J. Jayaram, and M. K. Tiwari, "Enhancing supply chain resilience using ontology-based decision support system," *International Journal of Computer Integrated Manufacturing*, vol. 32, no. 7, pp. 642–657, 2019.
- [44] H. Nacer and D. Aissani, "Semantic web services: standards, applications, challenges and solutions," *Journal of Network and Computer Applications*, vol. 44, pp. 134–151, 2014.
- [45] M. zur Muehlen and J. Recker, "How much language is enough? Theoretical and practical use of the business process modeling notation," *Seminal Contributions to Information Systems Engineering*, vol. 1, pp. 429–443, 2013.
- [46] J. Recker, M. Rosemann, P. Green, and M. Indulska, "Do ontological deficiencies in modeling grammars matter?" *MIS Quarterly*, vol. 35, no. 1, p. 57, 2011.
- [47] M. Weidlich, G. Decker, A. Großkopf, and M. Weske, "BPEL to BPMN: the myth of a straight-forward mapping," *On the Move to Meaningful Internet Systems: OTM 2008*, vol. 5331, pp. 265–282, 2008.
- [48] R. F. Lusch and S. L. Vargo, "Evolving to a new dominant logic for marketing," *The Service-Dominant Logic of Marketing*, vol. 21 p. 46 2021.
- [49] G. R. Online, I. Weber, X. Xu et al., *Untrusted Business Process Monitoring and Execution Using Blockchain*, Springer, Salmon Tower, 2016.
- [50] E. J. Arnould, "Service-dominant logic and resource theory," *Journal of the Academy of Marketing Science*, vol. 36, no. 1, pp. 21–24, 2008.
- [51] S. Madhavaram and S. D. Hunt, "The service-dominant logic and a hierarchy of operant resources: developing masterful operant resources and implications for marketing strategy," *Journal of the Academy of Marketing Science*, vol. 36, no. 1, pp. 67–82, 2008.
- [52] J. Yan, K. Ye, H. Wang, and Z. Hua, "Ontology of collaborative manufacturing: alignment of service-oriented framework with service-dominant logic," *Expert Systems with Applications*, vol. 37, no. 3, pp. 2222–2231, 2010.
- [53] S. L. Vargo and R. F. Lusch, "The four service marketing myths: remnants of a goods-based, manufacturing model," *Journal of Service Research*, vol. 6, no. 4, pp. 324–335, 2004.
- [54] M. A. Akaka and S. L. Vargo, "Technology as an operant resource in service (eco)systems," *Information Systems and e-Business Management*, vol. 12, no. 3, pp. 367–384, 2014.
- [55] J. Mylopoulos, L. Chung, and E. Yu, "From object-oriented to goal-oriented requirements analysis," *Communications of the ACM*, vol. 42, no. 1, pp. 31–37, 1999.
- [56] E. Yu, P. Giorgini, N. Maiden, and J. Mylopoulos, "Modeling strategic relationships for process reengineering," in *Social Modeling for Requirements Engineering*, The MIT Press, Cambridge, MA, 2010.
- [57] A. S. Vingerhoets, S. Heng, and Y. Wautelet, "Using i* and UML for blockchain oriented software engineering: strengths, weaknesses, lacks and complementarity," *Complex Systems Informatics and Modeling Quarterly*, vol. 26, pp. 26–45, 2021.
- [58] A. S. Vingerhouts, S. Heng, and Y. Wautelet, "Organizational Modeling for Blockchain Oriented Software Engineering with Extended-I* and UML," *PoEM Workshops*, vol. 2749, 2020.
- [59] Z. P. Fan, X. Y. Wu, and B. B. Cao, "Considering the traceability awareness of consumers: should the supply chain adopt the blockchain technology?" *Annals of Operations Research*, vol. 309, no. 2, pp. 837–860, 2022.
- [60] J. Leng, G. Ruan, P. Jiang et al., "Blockchain-empowered sustainable manufacturing and product lifecycle management in industry 4.0: a survey," *Renewable and Sustainable Energy Reviews*, vol. 132, p. 110112, 2020.
- [61] J. Leng, P. Jiang, K. Xu et al., "Makerchain: a blockchain with chemical signature for self-organizing process in social manufacturing," *Journal of Cleaner Production*, vol. 234, pp. 767–778, 2019.
- [62] S. Boyson, "Cyber supply chain risk management: revolutionizing the strategic control of critical IT systems," *Technovation*, vol. 34, no. 7, pp. 342–353, 2014.
- [63] J. Golosova and A. Romanovs, "The advantages and disadvantages of the blockchain technology," in *Proceedings of the 2018 IEEE 6th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE)*, Vilnius, Lithuania, November 2018.
- [64] J. Leng, M. Zhou, J. L. Zhao, Y. Huang, and Y. Bian, "Blockchain security: a survey of techniques and research directions," *IEEE Transactions on Services Computing*, vol. 1, 2022.
- [65] I. Todoran, Z. Hussain, and N. Gromov, "SOA integration modeling: an evaluation of how SoaML completes UML modeling," in *Proceedings of the 2011 IEEE 15th International Enterprise Distributed Object Computing Conference Workshops*, pp. 57–66, Helsinki, Finland, September 2011.
- [66] T. Zhang, S. Ying, S. Cao, and X. Jia, "A modeling framework for service-oriented architecture," in *Proceedings of the 2006 Sixth International Conference on Quality Software (QSIC'06)*, pp. 219–226, Beijing, China, December 2006.

Research Article

GPBFT: A Practical Byzantine Fault-Tolerant Consensus Algorithm Based on Dual Administrator Short Group Signatures

Xiaosheng Yu ^{1,2}, Jie Qin ^{1,2} and Peng Chen ^{1,2}

¹Hubei Province Engineering Technology Research Center for Construction Quality Testing Equipments, China Three Gorges University, Yichang 443002, China

²College of Computer and Information Technology, China Three Gorges University, Yichang 443002, China

Correspondence should be addressed to Xiaosheng Yu; yuxiaosheng@ctgu.edu.cn

Received 13 April 2022; Accepted 14 July 2022; Published 5 August 2022

Academic Editor: Jiewu Leng

Copyright © 2022 Xiaosheng Yu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The practical Byzantine fault-tolerant consensus algorithm reduces the operational complexity of Byzantine protocols from an exponential level to a polynomial level, which makes it possible to apply Byzantine protocols in distributed systems. However, it still has some problems, such as high communication overhead, low security, poor scalability, and difficulty in tracking. In this article, we propose a Byzantine fault-tolerant consensus algorithm based on dual administrator short group signatures (GPBFT). Firstly, the certification authority chooses the master node and group administrators based on the credit value. The group administrators organize the nodes into a group, and the members generate the signatures by applying the short group signatures scheme, in which any group member can represent the group during the GroupSign phase. Additionally, the GPBFT algorithm adds the Trace phase. According to member and client authentication information, the group administrator can track the true identity of the malicious node, identify the malicious node, and revoke it. The experimental results show that compared with the PBFT algorithm, the GPBFT algorithm can reduce the network communication overhead, reduce the consensus delay, and greatly improve the security and stability of the system. The algorithm can effectively manage member nodes and enable the tracking of identified malicious nodes while maintaining anonymity in terms of node tracking.

1. Introduction

In the practical applications of blockchain, storage scalability and security are the major problems that researchers confront in the existing field of sustainable manufacturing. The primary security problems include the generation and protection of private keys, vulnerabilities of the signature algorithm, the centralization of the consensus process, vulnerabilities of smart contracts, and vulnerabilities of decentralized applications. It is challenging to design scalable and highly secure consensus algorithms to assist self-adaptive coordination effectively in each sustainable manufacturing system [1, 2]. The consensus algorithm is the core mechanism in the blockchain system, and it aims at solving the problem of data consistency across distributed nodes in the system [3–5]. The Byzantine fault-tolerant algorithm (BFT) is a fault-tolerant algorithm based on the

Byzantine problem, which addresses how to reach consensus with reliable communication but the possibility of node failure [6]. However, the algorithm's exponential operational complexity makes it difficult to implement in practice. In Ref. [7], Castro and Liskov proposed the PBFT algorithm, an improved algorithm of BFT, which reduces the operational complexity of Byzantine protocols from the exponential level to the polynomial level, allowing Byzantine protocols to be used in distributed systems. The Hyperledger Fabric project was the first to use the PBFT algorithm in the consortium blockchain [8–10]. The Tendermint algorithm of the Cosmos blockchain combines the PBFT and the PoS algorithm and uses a token mortgage to select some consensus nodes for BFT consensus. It weakens the asynchronous assumption and incorporates the concept of lock based on the PBFT algorithm, allowing consensus nodes to reach consensus through two-stage communication in a partially

synchronous network [11, 12]. Based on Tendermint, the Hotstuff algorithm integrates the blockchain's chained-block structure with each phase of BFT, where the signatures confirmation of the previous block and the construction of a new block are performed simultaneously between nodes at each phase, simplifying the algorithm's implementation [13–15]. The MBFT algorithm combines hierarchical and slicing technology. The former can reduce the load of individual nodes and effectively improve consensus efficiency. The latter can assign transactions to different node groups to improve the processing power and decrease delay [16]. By grouping the network nodes, RBFT adopts the improved RAFT to participate in the consensus within the group. The leaders generated by the RAFT algorithm form a new group, and the PBFT consensus mechanism is adopted among the new groups. The algorithm solves the problem that some traditional PBFTs cannot support low delay, high throughput, and high security in large-scale networks [17]. The above consensus algorithms primarily use a subset of nodes to replace the entire network, which can reduce the traffic and improve the algorithm efficiency. However, in large-scale networks, only a few nodes participating in the consensus will have an impact on the system's degree of centralization, and the scalability is limited. On the other hand, many improvements of the algorithm are based on grouping or hierarchical thinking. Although the traffic of the PBFT algorithm can be reduced by dividing the consensus process into multiple levels, the algorithm still maintains a high complexity. Furthermore, the existing improved algorithms do not put the security of the PBFT algorithm in the first place or improve the handling of malicious nodes.

In this article, we propose a practical Byzantine fault-tolerant consensus algorithm based on dual administrator short group signatures, in which the client initiates a request to start the consensus process after selecting the master node and the group administrator with short group signatures. Compared with the PBFT algorithm, the GPBFT algorithm adds the GroupSign phase and Trace phase. In the GroupSign phase, the group administrator organizes the replica nodes into a group, in which one group member can represent the group as long as it completes the consensus process, which will reduce the communication overhead and decrease the number of communications. In the Trace phase, the group tracking administrator can trace the specific identity information of the node whose authenticator failed to verify and then revoke the member to ensure the security and stability of the system.

2. Related Work

Chaum and van Heyst introduced the concept of group signatures in 1991 [18]. Camenish et al. later modified and refined the concept [19, 20]. Group signatures are widely used in management, military, political, and economic aspects. Group signatures, like other digital signatures, can be verified publicly and only with a single group public key. The group administrator in a group signature ensures that the signature is secure and traceable, in addition to basic

anonymity. The group administrator can search for the real signer by opening the group signatures [21].

2.1. The Foundation of Short Group Signatures. Boneh, a professor at Stanford University, proposed short group signatures for the first time at the International Conference on Cryptography in 2004 [22]. The security of this signatures scheme is based on the strong Diffie–Hellman (SDH) and linear Diffie–Hellman (LDH) assumptions in cryptography. The signatures use bilinear mapping $e: G_1 \times G_2 \rightarrow G_T$, which guarantees the length of the signature while satisfying the characteristics of the group signatures and meets the security criteria.

2.1.1. Bilinear Mapping, and SDH and LDH Assumptions

Bilinear mapping: Let G_1 , G_2 , and G_T be three multiplicative cyclic groups of prime order n , and the generating element of G_n is g_n . A bilinear mapping is a mapping relation $e: G_1 \times G_2 \rightarrow G_T$ defined on these three groups, satisfying bilinearity, nondegeneracy, and computability.

q-Strong Diffie–Hellman (q -S DH): Given $(q+2)$ tuples $(g_1, g_2, g_2^y, g_2^{y^2}, \dots, g_2^{y^q})$ as input and a pair of $(g_1^{1/(y+x)}, x)$, $x \in Z_p^*$ as output.

Linear Diffie–Hellman (LDH): Decision linear problem in G_1 : $u, v, u^a, v^b, h^c \in G_1$ are given as input, and the output is Yes if $a + b = c$; otherwise, the output is No.

2.1.2. Short Group Signatures Technology. In a short group signature scheme, any member of a group can sign messages anonymously on behalf of the entire group. Short group signatures, like all other digital signatures, are publicly verifiable and can be verified with just one group public key, as shown in Figure 1.

2.1.3. Short Group Signatures Security Standards. Assuming that communication between the group members and administrators is confidential, a short group signature scheme should ensure that the signature system is both effective and long-lasting. The properties it needs to meet are shown in Table 1.

2.2. The PBFT Consensus Algorithm. The practical Byzantine fault-tolerant consensus algorithm is a distributed consistency algorithm based on state machine replication. It requires each node to sign when sending messages, and other nodes cannot modify other nodes' messages. After receiving a client request, the next request will be sent for execution only after the completion of the previous request by network-wide broadcast.

In the PBFT algorithm, all nodes operate in the same configuration, where there is only one master node and the other nodes act as replica nodes. The master node is responsible for sorting the requests from the clients and sending them to the replica nodes in order. The basic process of the whole algorithm is shown in Figure 2.

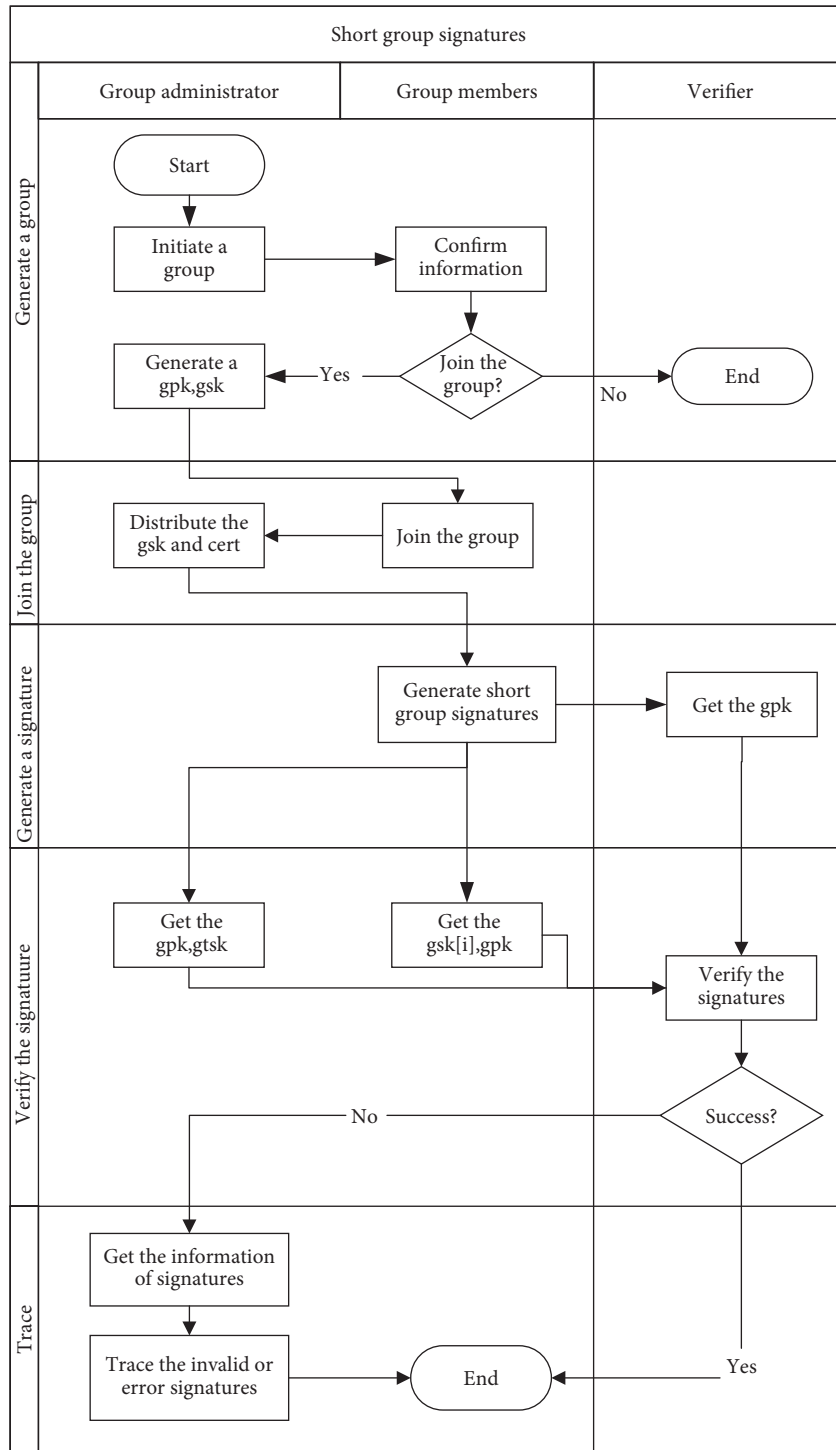


FIGURE 1: The short group signatures swimlane flowcharts.

There are three core phases in the process of the PBFT algorithm: the Pre-prepare phase, the Prepare phase, and the Commit phase. At first, a client sends a request to the master node. Then, the master node N_0 will send a Pre-prepare message to the other nodes after receiving the client request. Other nodes start the core three-phase consensus process after receiving the Pre-prepare message. The details are as follows:

- (a) Pre-prepare Phase: The node decides whether to agree to the request based on the message content or the request number order after receiving the Pre-prepare message.
- (b) Prepare Phase: After agreeing to the request, the node sends a prepare message to other nodes. If more than $2f$ (f denotes the maximum number

TABLE 1: Short group signatures security standards.

| Security standards | Explanation |
|--------------------|---|
| Correctness | Legal group members' signature is properly verified and that the group signature can be traced back to the original signer |
| Unforgeability | A legal group signature can only be generated by members who have obtained a group membership certificate and a signing key |
| Anonymity | The user who receives the signature can only verify the signature's legality, not the identity of the group member who generated it, or even the identity of the other members in the group |
| Traceability | Only the administrator can open a signature and find the identity of a signed group member |
| Unlinkability | It is computationally impossible to determine whether two signatures are signed by the same group member for unopened signatures |
| Irreplaceability | No member of the group can generate a signature on behalf of other users |
| Anti-joint attack | Even if some group members are federated, they cannot produce a valid group signature that can be tracked by the group administrator |

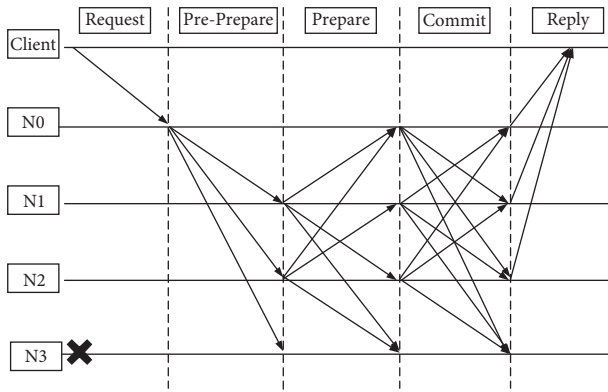


FIGURE 2: The PBFT consensus process.

of fault-tolerant malicious nodes) different nodes receive a prepare message within a certain time, the Prepare phase is complete.

- (c) Commit Phase: Broadcast commit messages to other nodes. When $2f + 1$ commit messages are received (including its own), most of the nodes have entered the Commit phase, and consensus has been reached in this phase, so the node executes the request and writes the data.

The node sends a message to the client when the process is finished.

3. The Dual Administrator Short Group Signatures Scheme

When designing a group signature scheme, the length of a group signature has always been an important factor. When the network bandwidth is limited, short group signatures are commonly used. The short group signatures can guarantee the group member's privacy, and one of the main advantages of this scheme is that the signature is short. For example, when the elements in G_1 are 171-bit strings, the signature's length is only 192 bytes, which can reduce the system's communication load. Moreover, the security is approximately the same as that of the RSA

signature algorithm with a signature length of 1024 bits. To ensure the stability and security of the signature algorithm, the dual administrator short group signatures scheme in this article must satisfy the following three conditions:

- A group signature scheme may require the membership revocation to simplify the membership management.
- Given the limited storage resources of the blockchain, the signature data cannot be too large.
- The group membership administrator initiates requests to establish groups, select users with high reputation as the group members, and select the group tracking administrator who is responsible for determining the members, opening the group signatures, and tracking the malicious users.

The dual administrator short group signatures scheme is shown in Figure 3.

The process is as follows:

- Initialize

Initialize (n): Initialize algorithm, that is, two group administrators establish a group according to the relevant parameters. The input parameter is n , where n is the number of the group members (including the administrators). The outputs are the group public key gpk , the group tracking private key $gtsk$, and the group member's private key $gsk[i]$.
- Join

Join (x_i): Join algorithm, that is, the process of user i ($1 \leq i \leq n$) applying to join the group. The input parameter is x_i . The group member i randomly selects $x_i \in Z_p^*$ as the user's private key. The output is the group member's private key $gsk[i]$ of user i .
- Sign

Sign ($gpk, gsk[i], M$): The signature algorithm, that is, the process of group signatures of the message M by the group members. The inputs are the group

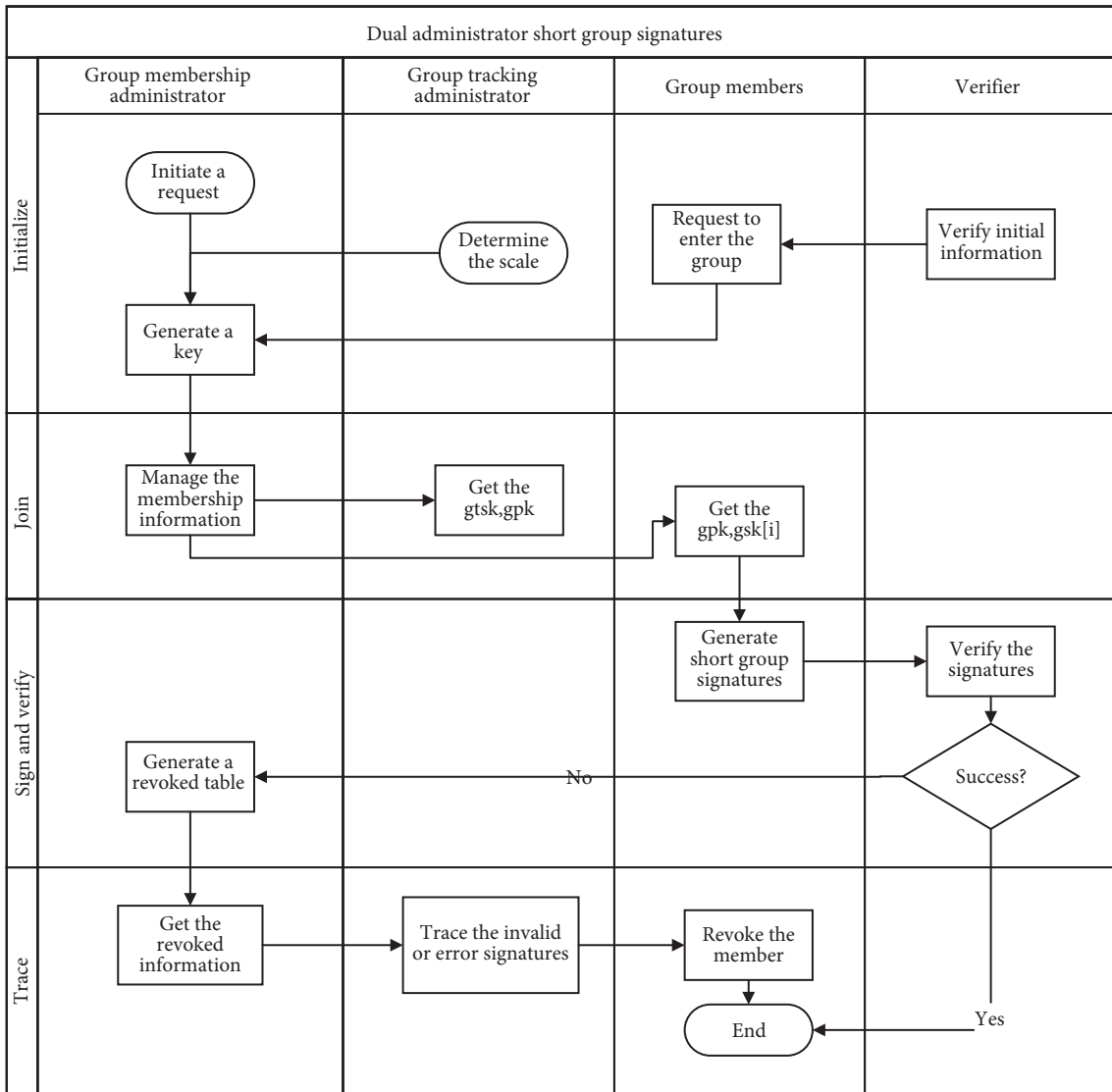


FIGURE 3: The dual administrator short group signatures swimlane flowcharts.

public key gpk , the group member’s private key $gsk[i]$, and the message M . The output is the signature σ .

(d) Verify

Verify (gpk, M, σ): The verification algorithm, that is, the procedure used by the verifier to determine whether σ is a valid signature. The inputs are the group public key gpk , the message M , and the group signature σ . The output is the verification result, which is a Boolean type yes or no.

(e) Trace

Trace ($gpk, gtsk, M, \sigma$): The tracing algorithm, that is, the signatory member can be traced according to the signature. The inputs are the group public key gpk , the tracing private key $gtsk$, the message M , and the signature σ . The output is the information of the parameters in the group member’s private key.

Finally, the group member is revoked based on the group membership information.

3.1. Dual Administrator Short Group Signatures Security Analysis. In addition to the basic security standards for the group signature, the short group signatures scheme in the article focuses on the following:

- (a) Correctness: Verifying a signature is a process of verifying that a data record is correct. A short group signature σ is a data record in the SDH hypothetical protocol. In this article, the signature σ generated by the short group signatures scheme must be verified by the Verify algorithm.
- (b) Anonymity: The verifier of short group signatures verifies the signature using the group public key, so it is impossible to determine which group member

signed the signature, thereby ensuring the group members' anonymity.

- (c) Traceability: The group tracking administrator has the key to open the signature at any time to obtain the identity of the group members in the event of a verification failure.
- (d) Irreplaceability: Each group member has its own tuple (A_i, x_i, y_i) , with the exception of a few public parameters, and the value of y_i can be kept secret by the group members. As a result, no member of the group can generate a signature on behalf of other members, including the group administrator.

4. A Practical Byzantine Fault-Tolerant Consensus Algorithm Based on Dual Administrator Short Group Signatures

There are some issues in the PBFT algorithm, such as high communication overhead, low security, poor scalability, and traceability. To address these issues, the GPBFT algorithm, a practical Byzantine fault-tolerant consensus algorithm based on dual administrator short group signatures, is proposed in this article. The consensus algorithm mainly includes five phases: Request, Pre-prepare, Prepare, GroupSign, and Trace. Firstly, the client initiates a request to the master node in the Request phase. The request is then processed in the Pre-prepare and Prepare phases. The dual administrator short group signatures scheme is proposed in the GroupSign phase of the algorithm, which chooses the group membership administrator as the algorithm's master node. It can reduce the possibility that the master node is a Byzantine node and speed up the view changing and "three-phase consensus" process. Finally, with the involvement of the supervisor, the group tracking administrator can track the identity of the signer by obtaining the signer certificate signer's certificate and the signature information in the Trace phase. The flow chart is shown in Figure 4.

The flow of the GPBFT consensus algorithm is as follows:

4.1. Preparation. In the GPBFT algorithm, the master node N_0 , that is, the group membership administrator, should be the first administrator in the short group signatures, who is responsible for the joining and revocation of nodes, and the second node N_1 , that is, the group tracking administrator, is responsible for tracking malicious nodes. The selection is based on certification from a reputable CA organization. Therefore, the probability of Byzantine error in the master node is low, which greatly avoids the number of view switches and reduces the cost and communication overhead. In addition, client c acts as the verifier of the short group signatures.

4.2. Request Phase. The client c sends a request $\langle \text{Request}[M, d(M)], o, t, cli \rangle$ to the master node N_0 , where M is the message content of the request entity, $d(M)$ is the digest of the message M , o is the operation requested by the client, t is the timestamp, and cli is the client's identifier.

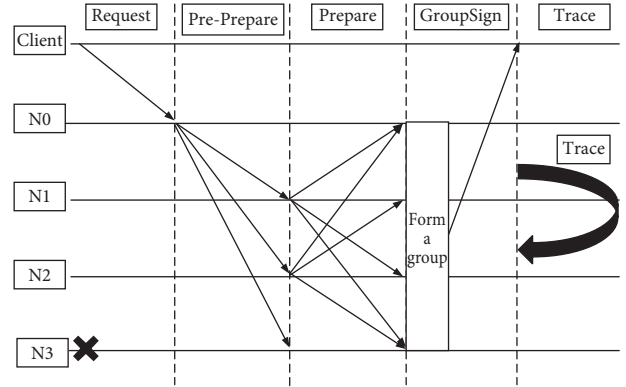


FIGURE 4: The GPBFT consensus algorithm.

4.3. Pre-prepare Phase. When the master node N_0 receives a request from a client, the message serial number n is first added to the message. Then, the message digest is obtained and signed to generate the signature M . The information is then spliced together and broadcast to the remaining replica nodes.

4.4. Prepare Phase. After receiving the Pre-prepare message from the master node, each replica node verifies the message digest d , message sequence number n , and message signature M in the Pre-prepare message. d must match the message digest of M in the Pre-prepare, the message sequence number n must be in the same view v , and the number is also n . If any of them fail, the Prepare broadcast will be rejected.

The illegal request is discarded. If the request is correct, the replica node i signs the message with its own private key and then sends a Prepare message to other nodes, including the master node.

The entire consensus process must be under the same view in the Pre-prepare and Prepare phases. The non-malicious nodes perform a consistent ordering of the message M , denoted by $\text{Order}(v, M, n)$, where v is the view ordinal number, M is the message, and n is an ordinal number that is confirmed for the message M when $\text{View} = v$.

4.5. GroupSign Phase. The messages in the preparation phase are verified by the master and replica nodes when the Prepare message is received. It mainly verifies whether the messages received by each node are in the same view v , whether the message digest d and the message sequence number n are both consistent, and whether the Prepare message of the replica node is correct. The illegal message requests are discarded after verification, and the legal request messages are carried out in the next steps. The group administrator initiates the request to build a group, and each group member joins to form a group, accepts the message from the Prepare phase, and performs the short group signatures operation. The message serial number remains n . Because of the reauthentication of the group signature, the likelihood of view change is greatly reduced. Even if view change does occur, node revocation can be performed reliably.

The group membership administrator N_0 and group tracking administrator N_1 will be combined with other replica nodes to form a group and generate a short group signature σ through the short group signatures scheme given in the article. Here, all group members can send messages on behalf of the group, and any member of the group who is the first to receive and verify the request message can immediately send the contents of the GroupSign message to the client for verification. The message content is $\langle \text{GroupSign}, d, gpk, \sigma, M \rangle$, where gpk is the group public key and σ is the message signatures generated for the member. After signing with a short group signature, the message sequence number remains n . Because of the revalidation of the new signatures scheme, node revocation can be performed stably even if view change occurs.

When a GroupSign message is received from a group member, the verifier firstly verifies the message digest d and message content M . Then, in the verification algorithm of short group signatures, the message content M , the group public key gpk , and the message signature σ are used as input parameters to verify whether the signature information is correct. If the message is correct, the corresponding information is sent to the group administrator to complete the basic consensus phase. If the message verification is incorrect, the message $\langle \text{GroupSign}, d, M, \sigma \rangle$ is sent back to the group membership administrator for information management updating, and the group tracking administrator checks it and enters the Trace phase.

4.6. Trace Phase. The group tracking administrator checks whether the verified messages M and d are correct and whether σ is a valid signature on M after receiving the error information feedback from the verified client. Then, the group members are tracked according to the feedback information. The Trace algorithm takes the group tracking private key $gtsk$, the signature σ in the feedback information, and the group public key gpk as input and returns the identity of node i in the consensus stage, as well as the identity and information of abnormal nodes.

5. Experiment

To evaluate the performance of the GPBFT algorithm, the Go language is used to simulate the flow of the GPBFT and PBFT algorithm. The experimental environment is an AMD Ryzen 7 4800h with a Radeon Graphics CPU, 16 GB memory, and 6 GB video memory. The operating system is Ubuntu 64 bit, and the go language version is GO1.15.6. The content of consensus transmission information is set to 48 bytes and 384 bits. The experimental results were processed by Python.

To demonstrate the accuracy and reliability of the GPBFT algorithm, this article compares the GPBFT and PBFT algorithm from five aspects: communication overhead, consensus delay, tracking efficiency, a signature generation time changes with the number of nodes, and basic algorithm security.

5.1. Communication Overhead Analysis. Pre-prepare, Prepare, and Commit are the three main phases of the PBFT algorithm. Because the communication times of the three phases are, $n - 1$, $n * (n - 1)$, and $n * (n - 1)$, the total communication times are $(n - 1) + (n^2 - n) + (n^2 - n)$ or $2n^2 - n - 1$, and the algorithm complexity is $O(n^2)$.

The Pre-prepare, Prepare, and GroupSign are the main phases of the GPBFT algorithm. In the Pre-prepare phase, the master node broadcasts the Pre-prepare message to other nodes, so communication times are $n - 1$. In the Prepare phase, after each node agrees to the request, it broadcasts the Prepare message to other nodes, with the communication times of $n * (n - 1)$, that is, $n^2 - n$. In the GroupSign phase, after the group administrator and other nodes form a group, only one node that received the Prepare message needs to sign the short group signatures and then respond to the client. The total communication times in the GroupSign phase are n . Therefore, the communication times are $(n - 1) + (n^2 - n) + n$, that is, $n^2 + n - 1$, and the algorithm complexity is also $O(n^2)$. The communication overhead between them is shown in Figure 5.

5.2. Consensus Delay. Consensus delay refers to the time difference between the initiation and completion of a request. It is an important indicator of the speed of the consensus algorithm. A lower consensus delay allows requests to be confirmed more quickly and makes blockchains more secure and practical. The consensus delay for the test is the time it takes to complete a consensus process, as defined in the following formula:

$$\text{DelayTime} = T_{\text{complete}} - T_{\text{submit}} \quad (1)$$

where T_{complete} is the time when the client confirmation is completed and T_{submit} is the time when the request starts.

The consensus delay of the PBFT and GPBFT algorithm is investigated by using 4, 25, 50, 75, and 100 nodes, respectively. The results of each group of experiments are the average of 30 different experiments. The consensus delay in the PBFT algorithm includes the time of the Request, Pre-prepare, Prepare, Commit, and Reply phases, plus the time taken by RSA to generate a signature for each node. The consensus delay in the GPBFT algorithm is the total time of the Request, Pre-prepare, Prepare, GroupSign, Trace phases, plus the generation time of dual administrator short group signatures. Because of the characteristics of short group signatures, any member node in the GPBFT group can sign the message anonymously on behalf of the entire group and then feed back to the client. The total time delay of this algorithm should consider the time difference between the first node sending the message and the feedback. If the client verifies correctly, the consensus is complete. If the verification is incorrect, the Trace tracing phase starts. The group tracking administrator opens its group signature and announces the malicious node, and the group membership administrator realizes the cancellation of the member. The consensus time delay between GPBFT and PBFT is shown in Figure 6.

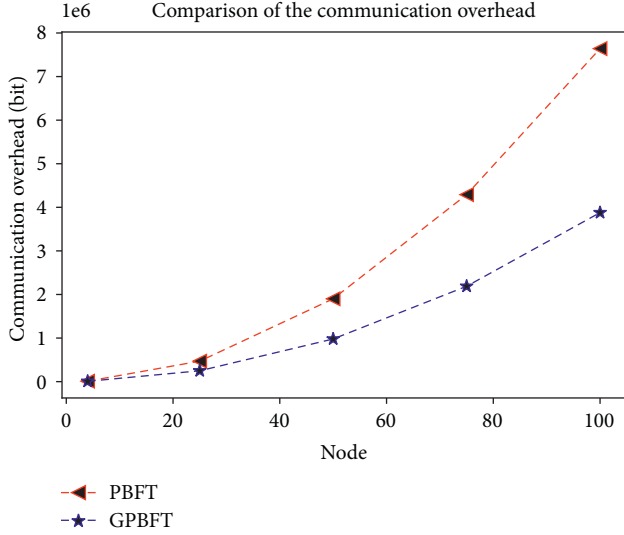


FIGURE 5: Comparison of communication overhead between the PBFT and the GPBFT algorithm.

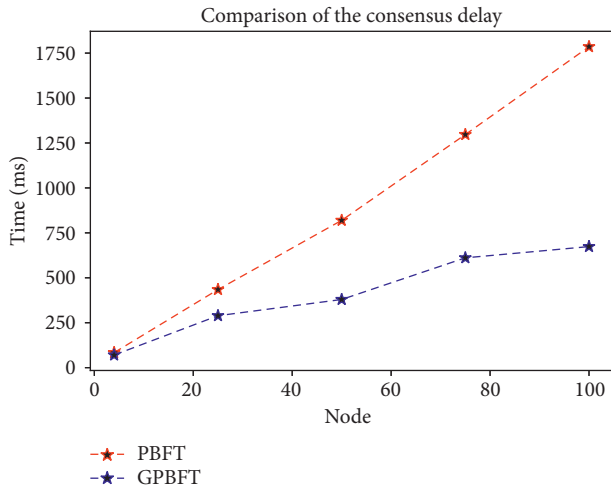


FIGURE 6: Comparison of the PBFT and GPBFT consensus delay.

5.3. Algorithm Tracking Efficiency. We test the time to trace the group signature taken by GPBFT when the number of nodes is 4, 15, 25, 35, 45, 55, 65, 75, 85, 95, and 105. The experimental results are the average of 50 experiments for each group, in which the abnormal data are excluded. The experimental results show that the tracking time of most nodes is between 1.5 ms and 2.5 ms when the number of nodes is small. The tracking time becomes longer as the node grows, but a few of them remain between 1.5 ms and 2.5 ms. When the number of nodes reaches 75, more than one-third of the nodes' tracking time increases to 70–85 milliseconds; when the number of nodes reaches more than 100, at least half of the nodes' tracking time exceeds 100 milliseconds. The group tracking administrator tracking efficiency is shown in Figure 7.

5.4. The Signature Generation Time of Different Numbers of Nodes in the GPBFT Algorithm. With the growth of nodes, the time for the signature algorithm to generate public and

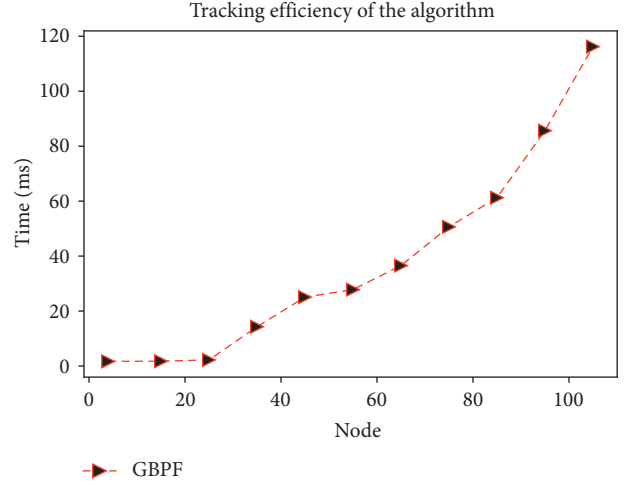


FIGURE 7: Tracking efficiency of the GPBFT algorithm.

private keys for nodes in GPBFT will also change. We compare the key generation time with different signature schemes between the GPBFT and the PBFT algorithm. This shows the superiority of the short group signatures algorithm in this scheme.

The experimental results show that the GPBFT's short group signatures scheme has the advantage of generation time. When the number of nodes reaches more than 75, this advantage becomes apparent. The results are shown in Figure 8.

5.5. Algorithm Security. Regarding security issues, the PDI framework proposed in Ref. [23] reviews blockchain security research from three aspects: the process level, the data level, and the infrastructure level. Starting from the data level, this article takes a consensus algorithm, authentication, signature scheme, and other aspects as a breakthrough to solve the security problem of blockchain. Data-level security is flanked by process and infrastructure, so the optimized algorithm in this article can also bring beneficial changes to the other two levels. For the consensus algorithm, the following aspects are considered.

5.5.1. Number of Malicious Nodes. As the number of malicious nodes in the simulated network changes, we test whether consensus is reached in the GPBFT algorithm. If there is a single malicious node in the GPBFT, consensus can be reached. If the number of malicious nodes reaches the maximum limit of malicious nodes, consensus cannot be reached. In GPBFT, the Pre-prepare and Prepare phases must still satisfy the Byzantine rules; that is, at least $2f + 1$ messages must be received. The experiments show that the fault tolerance rate of the GPBFT algorithm is consistent with that of the PBFT algorithm. It has normal fault tolerance.

5.5.2. The Master Node Problem. In the PBFT algorithm, the master node is generated by random selection, which has a high level of uncertainty. In contrast, in the GPBFT

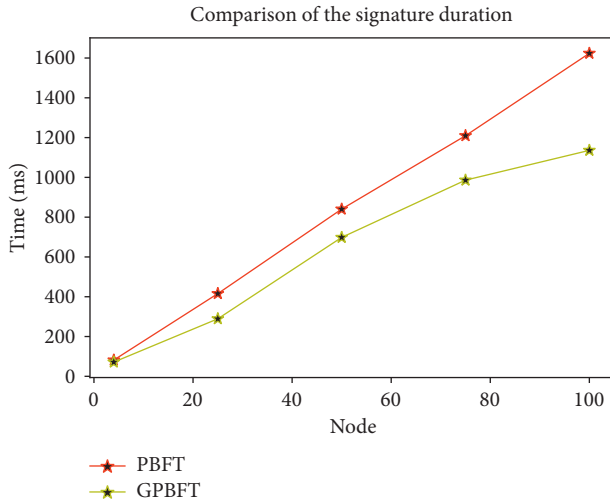


FIGURE 8: Temporal comparison between the GPBFT's short group signatures and RSA signature.

algorithm, it (i.e., group membership administrator) is generated by the CA authentication institution, and the likelihood of Byzantine error is greatly reduced.

When the master node becomes a malicious node or fails in the GPBFT algorithm, it can be effectively identified and responded to. If the malicious request is propagated by the master node, the consensus system will fail to recognize the request information and will be unable to reach the consensus. As a result, the consensus system will send a view switching request to revoke the malicious master node and then reselect the master node (i.e., the group administrator). Furthermore, when the master node is down, the consensus cannot be reached in the GPBFT.

The tests show that it has good resistance and response ability for the master node problem.

5.5.3. Sybil Attack. In a P2P network without trusted node identity authentication institutions, it is difficult to guarantee that multiple backup nodes are different entities. By deploying only one entity that broadcasts multiple identity IDs to the network, an attacker can act as multiple distinct nodes, and these forged identities are commonly referred to as Sybil nodes. Because there is no God perspective in an entirely decentralized system, no single node will naturally know the exact number of nodes involved. They can only judge the overall situation by the data they received. As a result of this property, the attacking node disguises itself as multiple nodes and broadcasts in the P2P network. The number of Byzantine nodes that can be resisted in PBFT is $N \geq 3f + 1$ (f is the number of malicious nodes).

A dual administrator short group signatures mechanism is introduced in the GPBFT algorithm, in which each node only needs to sign by itself and then feedback the message when it enters the GroupSign phase. Although each node is a group member, each node is relatively independent, and the consensus is reached when the client verifier receives a group member's message. If the attacker

who used the Sybil Attack disguises the node, the node will be tracked and revoked.

The consensus can be reached if there is a disguised malicious node in the consensus node. However, the consensus cannot be reached if the total number of nodes remains unchanged and the number of disguised nodes exceeds the maximum number of malicious nodes that the system can bear. When the attacker impersonates the master node and spreads malicious requests, the system will not be able to complete the consensus. At the same time, the system will change its view to overthrow the master node with malicious behavior and then reselect a new master node.

5.5.4. Fault Tolerance Analysis. The maximum number of fault-tolerant nodes of the PBFT algorithm is $f_1 \leq N - 1/3$.

For the GPBFT algorithm, in addition to supporting fault nodes, it also needs to support error nodes. Assume the number of cluster nodes is N and the problematic node is f . Among the problematic nodes, it can be either fault or error or fault and error. There are two extremes.

In the first case, f problematic nodes are both fault and error. According to the features of group signature, at least one node completes consensus in the GroupSign phase. Then, the cluster can reach consensus. This means that the maximum number of fault-tolerant nodes is $(N - 1)$ in this case.

In the second case, the fault nodes and the error nodes are both different nodes. Hence, there will be f fault nodes and f error nodes. When a node is found to be a fault node, it will be excluded by the cluster, and f error nodes remain. Then, according to the features of the group signature, the number of normal nodes in the cluster is at least one. Therefore, there is one correct node, f fault nodes, and f error nodes in all nodes, that is, $2f + 1 = N$. Therefore, the maximum number of fault-tolerant nodes is $f_2 \leq N - 1/2$ in this case.

Due to $f_1 < f_2$, the GPBFT algorithm in this article has a higher fault tolerance than the PBFT algorithm.

5.6. Comparison with Other Literature Studies. For the existing consensus algorithm optimization scheme, we can evaluate it from four dimensions in the actual design, namely, decentralization, efficiency, security, and fault tolerance. The idea of optimization is generally divided into the following aspects: optimizing the consensus process, selecting the primary node, and selecting the appropriate signature algorithm or the underlying communication mode

Buchman [11] replaced messages changing in the PBFT view with variables and deleted the garbage collection mechanism. The simplified Tendermint algorithm only has three stages, which is more concise and understandable than PBFT.

dBFT [24] and Tendermint select nodes based on PoS. dBFT is an algorithm proposed by the AntChain (Neo), which combines PoS with the PBFT mechanism. Although it can improve performance, the election process is static. Because the electoral scheme and results are entirely determined by the project side, the NEO has also been overcentralized. RBFT uses the hash algorithm to group

nodes firstly, and the Raft mechanism is used to elect leaders among groups. Then, the leaders are assembled to run the PBFT algorithm [17].

RBFT, SBFT [25], and HotStuff [13] reduce the system communication complexity to $O(n)$ by introducing threshold signature. Jalalzai et al. proposed the Fast-Hot-Stuff algorithm by using aggregate signature in the NewView phase of consensus, which improves the efficiency of consensus.

The GBC consensus algorithm [26], which is based on the Gossip protocol, improves the fault tolerance of the system from $1/3$ to $1/2$.

Compared with the above literature, the short group signatures are used as the underlying signature algorithm in this article, which can maintain a certain anonymity, track malicious members in malicious groups quickly and effectively, and increase the stability and security of the algorithm. On the other hand, GPBFT simplifies the algorithm process and reduces the communication overhead of the algorithm, and the introduced GroupSign phase can enter the tracking phase when the consensus fails. The certification authority is used to select the group master node in the article, which reduces the probability of Byzantine errors on the nodes and improves the scalability of the algorithm. In addition, the algorithm is more secure and stable against Sybil attacks with fault tolerance $N - 1/2$.

6. Conclusions

In this article, we proposed the PBFT consensus algorithm (GPBFT) based on dual administrator short group signatures, which combines the advantages of short group signatures with short length, high applicability, low algorithm complexity, and traceable nodes for administrators. Experimental results show that it can not only reduce communication overhead, greatly reduce network traffic, and improve communication efficiency but also be applied in consensus schemes with more nodes, higher scalability, and lower consensus delay. The tracing and dual administrator mechanism in the GPBFT algorithm can make the algorithm more secure and have a greater control rate on malicious nodes. The GPBFT is a weakly centralized algorithm that can be used in the practical applications, including e-commerce, e-banking, e-voting, and e-auction. The selection of the master node and group administrators is an important focus for future work, as it will make the selected administrators more suitable and authoritative and make them more applicable to the alliance chain. In addition, with the rapid development of quantum computing, current blockchain platforms that rely on group signature and hash algorithms are vulnerable to quantum attacks. We can use multichain synchronization and optimized group signature to solve this problem, such as side chain technology or group signatures schemes on lattices [23].

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors thank the Big Data team of the Computer and Information Institute of China Three Gorges University for providing the environment required for the experiment. This research was supported by the Hubei Science and Technology Major Project (Grant no. 2020AEA012) and the National Key Research and Development Program of China (Grant no. 2016YFC0802500).

References

- [1] J. Leng, G. Ruan, P. Jiang et al., "Blockchain-empowered sustainable manufacturing and product lifecycle management in industry 4.0: a survey," *Renewable and Sustainable Energy Reviews*, vol. 132, Article ID 110112, 2020.
- [2] J. Leng, M. Zhou, J. L. Zhao, Y. Huang, and Y. Bian, "Blockchain security: a survey of techniques and research directions," *IEEE Transactions on Services Computing*, vol. 11, 2021.
- [3] Y. Yuan, X. C. Ni, and S. Zeng, "Blockchain consensus algorithms: the state of the art and future trends," *Acta Automatica Sinica*, vol. 44, no. 11, pp. 2011–2022, 2018.
- [4] S. M. H. Bamakan, A. Motavali, and A. Babaei Bondarti, "A survey of blockchain consensus algorithms performance evaluation criteria," *Expert Systems with Applications*, vol. 154, Article ID 113385, 2020.
- [5] X. Cai, Y. Deng, L. Zhang, and Jiuchen Shi, "Blockchain principle and its core technology," *Journal of Computer Science*, vol. 44, no. 1, pp. 84–131, 2021.
- [6] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," *ACM Transactions on Programming Languages and Systems*, vol. 4, no. 3, pp. 382–401, 1982.
- [7] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," in *Proceedings of the Appears in the Proceedings of the Third Symposium on Operating Systems Design and Implementation*, New Orleans, LA, USA, February 1999.
- [8] 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.
- [9] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, and K. Christidis, "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proceedings of the 13th EuroSys Conference*, pp. 1–15, ACM Press, New York, NY, USA, April 2018.
- [10] H. Sukhwani, N. Wang, K. S. Trivedi, and A. Rindos, "Performance modeling of hyperledger fabric (permissioned blockchain network)," in *Proceedings of the 2018 IEEE 17th International Symposium on Network Computing and Applications*, pp. 1–8, IEEE Press, Cambridge, MA, USA, November 2018.
- [11] E. Buchman, *Tendermint: Byzantine Fault Tolerance in the Age of Blockchains*, University of Guelph, Guelph Canada, 2016.
- [12] F. Saleh, "Blockchain without waste: proof-of-stake," *Review of Financial Studies*, vol. 34, no. 3, pp. 1156–1190, 2021.
- [13] M. Yin, D. Malkhi, M. K. Reiter, G. G. Gueta, and I. Abraham, "HotStuff: BFT consensus in the lens of blockchain," 2018, <https://arxiv.org/abs/1803.05069>.

- [14] M. Yin, D. Malkhi, M. K. Reiter, G. G. Gueta, and I. Abraham, "Hotstuff: bft consensus with linearity and responsiveness," in *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing*, Toronto, Canada, July 2019.
- [15] Y. Desmedt and Y. Frankel, "Shared generation of authenticators and signatures," in *Proceedings of the Advances in Cryptology — CRYPTO '91*, Springer, Santa Barbara, CA, USA, August 1991.
- [16] M. Du, Q. Chen, and X. Ma, "MBFT: a new consensus algorithm for Consortium blockchain," *IEEE Access*, vol. 8, Article ID 87665, 2020.
- [17] D. Huang, Li Lang, B. Chen, and Bo Wang, "Rbft: Byzantine fault tolerant consensus mechanism based on raft cluster," *Journal of Communications*, vol. 42, no. 3, pp. 209–219, 2021.
- [18] D. Chaum and E. van Heyst, "Group signatures," in *Proceedings of Eurocrypt of LNCS*, D. W. Davies, Ed., Vol. 547, Springer, Berlin, Germany, 1991.
- [19] J. Camenisch and M. Michels, "A group signature scheme based on an RSA-variant," *Basic Research in Computer Science*, vol. 5, no. 27, 1998.
- [20] G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik, "A practical and provably secure coalition-resistant group signature scheme," in *Proceedings of Crypto1880 of LNCS*, M. Bellare, Ed., Springer, Berlin, Germany, 2000.
- [21] X. G. Cheng, J. Wang, and J. X. Du, "Survey on group signature," *Application Research of Computers*, vol. 30, no. 10, pp. 2881–2886, 2013.
- [22] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," *An extended abstract of this paper is to appear in Advances in Cryptology—CRYPTO 2004*, Springer, Berlin, Germany, 2004.
- [23] J. Leng, S. Ye, M. Zhou et al., "Blockchain-secured smart manufacturing in industry 4.0: a survey," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 1, pp. 237–252, 2021.
- [24] Q. Wang, J. Yu, Z. Peng, V. C. Buli, S. Chen, and Y. Ding, "Security Analysis on dBFT protocol of NEO," in *Proceedings of the Financial Cryptography and Data Security*, Kota Kinabalu, Malaysia, February, 2020.
- [25] G. G. Gueta, I. Abraham, S. Grossman, D. Malkhi, and B. Pinkas, "SBFT: a scalable and decentralized trust infrastructure," in *Proceedings of the 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, Portland, OR, USA, June 2018.
- [26] Q. W. Zhang, Z. Q. Wang, and Y. Q. Zhang, "Research on trust collection consensus algorithm based on Gossip protocol," *Computer Science*, vol. 47, no. S1, pp. 391–394, 2020.

Research Article

CR-BA: Public Key Infrastructure Certificate Revocation Scheme Based on Blockchain and Accumulator

Jingxue Xie ¹, Xinghong Tan ¹, and Liang Tan ^{1,2}

¹School of Computer Science, Sichuan Normal University, Chengdu 610000, Sichuan, China

²Institute of Computer Science, Chinese Academy of Sciences, Beijing 100000, China

Correspondence should be addressed to Liang Tan; jkxy_tl@sicnu.edu.cn

Received 29 April 2022; Accepted 11 June 2022; Published 31 July 2022

Academic Editor: Jiewu Leng

Copyright © 2022 Jingxue Xie et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the development of blockchain, many studies apply blockchain to certificate revocation. However, existing blockchain-based certificate revocation schemes have two shortcomings. First, the storage overhead on the blockchain is relatively large. Second, as the number of revoked certificates increases, the misjudgment rate of certificate status will increase accordingly, so a public key infrastructure implementation certificate revocation scheme based on blockchain and accumulators, called CR-BA, is proposed. First, CR-BA expands the certificate structure, adding a revocation factor and a smart contract account for accessing the blockchain in the certificate extension, which is filled by the CA when the certificate is generated. Then, when the certificate is to be revoked, CA generates the revocation fingerprint through the revocation factor and publishes it to the blockchain. Finally, when the user needs to verify the status of the certificate, CA calculates the revocation fingerprint according to the revocation factor on the certificate, then compares it with the existing revocation fingerprint on the blockchain, and returns the comparison result to the user. The experimental results show that this scheme can effectively overcome the storage and misjudgment problems caused by existing blockchain-based certificate revocation schemes and improve the query efficiency of certificate revocation information.

1. Introduction

Public key infrastructure (PKI) collects hardware, software, people, policies, and procedures. It can realize the generation, management, storage, distribution, and revocation of keys and certificates based on public key cryptosystems [1], which is the foundation and core of network security construction. It is now widely used in secure e-mail, virtual private networks, e-commerce, and e-government and is the basis for achieving network security [2]. Certificate revocation is one of the core functions of PKI, which indicates the end of certificate life. When personal identity information changes or the private key of the certificate is leaked, or the fraudulent behavior of the certificate owner, the certificate user should promptly submit a certificate revocation request to CA. CA should also put certificates into the publicly released Certificate Revocation Lists (CRLs) in time. The traditional certificate revocation method is to store revoked certificate number in the LDAP directory server,

and the user is informed of the certificate revocation information by querying LDAP. This centralized query approach suffers from the trustworthiness of the LDAP administrators, and the LDAP directory server can become a performance bottleneck as the number of accesses increases.

Blockchain [3, 4] has been developed relatively quickly in recent years. In essence, it is a shared database, a distributed ledger technology based on the point-to-point network, providing a set of distributed data structures, interaction mechanisms, and computing paradigms [5], with decentralized storage, decentralization, tamper-proof and traceable characteristics [6–8]. Blockchain has laid a solid foundation of “trust.” It is widely used in data security [9], becoming a better solution to the problem of traditional certificate revocation due to its superiority in transparency, traceability, and security [10]. There are many research results. Fromknecht et al. [11] designed a fully decentralized PKI using consistency provided by the Namecoin blockchain. Kubilay et al. [12] proposed a new blockchain-based

PKI architecture for certificate transparency. In particular, Rabieh et al. [13] used Bloom filters to reduce the size of CRL, and Medury et al. [14] and Huang et al. [15] used cuckoo filters to quickly verify revoked certificates and use blockchain publishing filters. However, existing blockchain-based certificate revocation schemes also have two problems: first, the storage overhead on the blockchain is relatively large. For example [13] uses the insertion of certificate fingerprints in Bloom filters. When a new certificate revocation transaction is generated, it is necessary to publish a new complete filter to the blockchain, with each block storing a complete array. Second, as the number of revoked certificates increases, the false-positive rate of certificate status increases accordingly. The Bloom filter used in [13] is a probabilistic data structure designed by multiple hash function algorithms. The principle is to calculate the hash value by hash function and then map this value to an array set to 1. However, different values may generate the same address, resulting in a hash collision, causing the problem that filter query results do not match actual data. The cuckoo filter used in [15] is a probabilistic data structure designed based on the Cuckoo Hashing algorithm. The principle is to calculate the fingerprint and hash value of the data and calculate another hash value from the fingerprint and hash value. It maps two hash values to two locations. If the insertion fails at both positions, one fingerprint is randomly squeezed out, and a new position is found for that fingerprint again. This method may cause a hash collision on fingerprint information in extreme cases, leading to the misjudgment that elements outside the set exist in the set. Benaloh and Mare [16] first proposed accumulators [17]. It can hash a large set of inputs into a short value. Moreover, given an accumulator, an element, and a membership witness, it can verify the existence of the element in the cumulative set [11]. Member witnesses are generated when relevant elements are added to the accumulator and are usually updated when the collection is changed. Member witnesses that are not elements of the accumulator are difficult to find computationally. The feature of the accumulator is that when an element is added or removed, accumulated value and membership proofs can be effectively updated, and it supports member proofs and nonmember proofs. Accumulator uses the strong RSA assumption in cryptography to ensure security and zero-knowledge proofs. It ensured that users do not reveal information about themselves when proving their legitimacy to the verifier.

Therefore, this article proposes a public key infrastructure certificate revocation scheme based on blockchain and accumulator. First, we expand the certificate structure and add a new revocation factor and blockchain access information to the certificate extension, which is populated by CA invoking smart contracts when generating certificates. Then, when a certificate is to be revoked, CA generates a revocation fingerprint through the revocation factor and accumulator and publishes it to the blockchain. Finally, when verifying certificate status, CA verifies the validity of the certificate according to the revocation factor and revocation fingerprint of the blockchain. The experimental results show that this scheme can effectively overcome

storage and misjudgment problems caused by the previous certificate revocation scheme and improve the query efficiency of certificate revocation information. The rest of this article is organized as follows: Section 2 describes current research work on certificate revocation. Section 3 describes the essential concepts of this system. Section 4 describes the system design. Then, Section 5 presents a feature analysis of this article. Section 6 describes experiments and gives a performance evaluation of the proposed method. Section 7 concludes the whole article and presents future research prospects.

2. Related Work

For certificate revocation of PKI, many research results have been achieved. The following will analyze and summarize the current primary certificate revocation mechanisms and blockchain-based certificate revocation mechanisms.

2.1. Main Certificate Revocation Schemes. Certificate Revocation List (CRL) [18, 19] is a time-stamped list in which all certificate information that has been revoked or hung is listed, issued by the certification authority CA and published periodically. CRL contains two fields: the current update date and the next update date. Users can determine whether the current CRL is the latest from two date information, and CRL contains the signature of CA. So CRL can be stored in any node on the network. To check the validity of a certificate, the verifier initiates a request to the LDAP directory server hosting the corresponding CRL with the CA identifier parameters that issued the certificate. Then it receives the latest CRL generated by CA and checks the CRL signature and its validity. Finally, the certificate is searched in CRL to determine whether the certificate and key pair are trusted. The advantages of the traditional CRL approach are simplicity, information richness, and low risk. The disadvantages are high bandwidth cost, low query efficiency, and long delay time. The size of CRL is its main drawback. The amount of communication between user and directory server is heavy. Each verification of the public key certificate requires downloading the entire CRL, which requires high bandwidth for verification and update. When the scale of CA becomes larger and larger and users use certificate information more and more frequently, a large number of users download new CRLs on LDAP. CAs have to keep publishing new CRLs to LDAP, which at this time tends to cause congestion within CRL requests. The feature greatly limits the scalability of this method. At present, many improved CRL schemes have been proposed, and some well-known ones are described below. Incremental distribution (Delta-CRL) [20] provides a more efficient way to distribute certificate status information. Instead of generating a complete and potentially growing CRL every time a certificate is revoked, the list only records all the unexpired certificates that have been revoked since the last CRL was issued. Clients do not have to download the entire CRL but only maintain their own CRL database and keep it updated with a Delta-CRL that is much smaller than the size of the entire CRL, saving

communication bandwidth and time. Delta-CRL aims to solve the scalability problem of downloading CRLs. However, Delta-CRL only represents a part of CRL, and revocation information can only be used after it is associated with the main CRL. That is, any request issued at a certain point in time requires a complete CRL. This scheme cannot solve the problems of verification time and computational complexity of revoked certificates. CRL distribution points (CRL-DP) [21] is a way for CA to address scalability by partitioning CRL using CRL distribution points on a compromise and routine revocation basis. The main idea is that system divides the entire authentication space into small fragments according to some classification. Each fragment is associated with a particular CRL distribution point, which can be located on a different host or on a different directory on the same host. Clients checking certificate status can access the CRL distribution point specified in the certificate instead of the CRL distribution point in the main CRL. Therefore, the CRL distribution point reduces the length of CRL downloaded by the user, which is more advantageous than complete CRL in balancing network load and improving authentication efficiency. However, this method does not reduce peak requests and increases users' average request rate and waiting time when they need to query multiple segments [1]. Moreover, since the location of CRL distribution points is fixed throughout the life of the certificate, CA must know in advance how to segment the CRL information and fix the location of CRL distribution points, which is also a problem of the method. Redirect certificate revocation list (RCRL) [22] can solve the problem that the location in the CRL release point cannot be changed. In this mechanism, a new critical CRL extension is defined. This extension consists of a range that covers authenticated certificates and a pointer to the new CRL location of the problematic certificate. Even though redirected CRL solves the problem of fixed distribution point locations. It still causes an increase in the average CRL request rate and longer user wait times as the number of CRL segments increases. Indirect CRL [22] enables the publication of revocation information from multiple CAs in a single CRL. That is, multiple CAs can use the same CRL distribution point. The use of indirect CRLs reduces the total number of CRLs that users need to retrieve during the certificate validation process, reducing traffic load and cost. However, since revocation information comes from different places, it is necessary to determine the CA of each item in the certificate revocation list. Therefore, a certificate issuer field needs to be set in each item. The distribution point is maintained by another trusted third party, increasing the difficulty of maintaining a single distribution point. Another alternative to RL is the Certificate Revocation Status (CRS) [23, 24], which is an authentication dictionary data structure with evidence having the characteristic of being delivered through unscientific third parties [22]. CRS was designed in accordance with the following principles: increasing the amount of communication between CA and directory during the update of revocation information and being able to minimize the length of evidence obtained when a user queries the status of the certificate from a directory (this

contains all the information of revoked certificate in CRL). CA sends a signed statement to the CRS directory every day stating the status of individual issued certificates, and each unexpired certificate has a signed statement. When a user queries for certificate revocation status, CRS Directory replies with information that the user can use to verify the requested status. CRS reduces the communication load between server and end entity, achieving an overall performance improvement compared to the CRL method. However, it greatly increases the communication load between the server and CA.

In addition to this, an alternative to the CRL scheme is Certificate Revocation Tree (CRT) [25]. It is usually a Merkle hash tree representing all certificate revocation information for a given PKI domain, providing a set of statements about the certificate sequence numbers in leaves. The main advantage of this approach is that we do not need a complete CRL to provide certificate validation. However, its main disadvantage is updating, since any change in the revocation certificate set may cause the entire list to be recomputed, resulting in a continuous workload [26].

Online Certificate Status Protocol (OCSP) [27] is an online revocation system that relies on a request/response mechanism. It acts between the client and the server and provides a way for applications to obtain certificate status online. The client, called an OCSP requester, generates an OCSP request to send to the server if it wants to verify the status of one or more certificates. The server, called an OCSP responder, first verifies the request's syntax and semantics after receiving the client's request and then constructs an OCSP response to return to the requester. Revocation information is obtained at the server of the OCSP responder, which receives it directly from CA. In fact, the CA does not sign the OCSP response, so the revocation server must be trusted by the CA. OCSP approach solves low timeliness and revocation information update problems. However, this method has some drawbacks, mainly (1) since this method is centralized, the OCSP server represents a single point of failure [28]. (2) OCSP responds to verify the certificate's revocation status without checking the validity sequence number. A malicious user can use the validation flood server to request a certificate not belonging to CA. This makes the server work concentrated will lead to denial of service. (3) OCSP lookup has a high overhead [29, 30]. (4) OCSP is an ineffective online scheme for offline systems [30]. (5) OCSP can provide real-time responses to revocation queries, but it is unclear whether these responses contain updated revocation information. (6) OCSP approaches introduce privacy risks. OCSP responders know which certificates end users are verifying, so they can track which sites users are visiting [30].

2.2. Blockchain-Based Revocation Schemes. In recent years, blockchain has become popular in certificate revocation research. Blockchain-based technologies are appealing because they allow for secure, robust, and trustworthy solutions and bring improvements compared to current

technologies or management systems in terms of transparency and traceability. It is the ideal technology for PKI design and deployment [31]. The following describes several blockchain-based PKI methods, focusing on their certificate revocation management component.

Fromknecht et al. [11] proposed a fully decentralized PKI that leverages the consistency provided by the Namecoin blockchain to provide strong identity retention guarantees, which [11] has five functions: register, update, find, verify, and withdraw. Although Fromknecht et al. [11] solved some problems, the method still has many shortcomings. As in the high cost of mining and public key lookup and verification, there is no actual verification of the linkability of ID links to registered public keys. Moreover, during the revocation process of Fromknecht et al. [11], the owner of the identity ID can revoke its public key only by publishing a transaction to the blockchain. The entire revocation process is completely handled by the owner himself, which will cause many problems; for example, (1) handling revocation by the user himself is a difficult task as it requires some expertise. Furthermore, a user cannot know if the key has been compromised. (2) Malicious users will not revoke their keys. (3) To verify the certificate's status, the scheme must first verify that a revoked certificate is published in the blockchain. It is all about browsing the blockchain to ensure that the certificate has not been revoked. However, censoring search content in blockchain can take much time. Hu et al. [32] proposed Certificate Revocation Guard (CRG), which intercepts all TLS communications from entities such as organizational gateways using an intermediate box that performs OCSP requests to check certificate revocation status. If a revoked certificate is detected, a malformed certificate is returned to the client, effectively blocking the connection. The policy does not require any modification by clients to participate. However, mobile clients such as laptops and smartphones will lose protection when they leave the network due to using intermediaries. Hewa et al. [33] proposed the application of an Elliptic Curve Qu-Vanstone (ECQV) certificate, which is lightweight for resource-constrained IoT devices. Additionally, they integrate blockchain-based smart contracts to handle certificate-related operations. They apply smart contracts to certificate issuance and develop a smart contract-based threat scoring mechanism to revoke certificates automatically. The lightweight nature of ECQV certificates enables distributed ledgers to store, renew, and revoke certificates. Kubilay et al. [12] proposed a new blockchain-based certificate transparency PKI architecture, called CertLedger, and provided an ideal certificate revocation transparency. The revocation status of all TLS certificates, the entire revocation process, and trusted CA management is carried out in CertLedger. BARS [34] is a blockchain-based anonymous reputation system to break the linkability between real identities and public keys to preserve privacy. BARS has two main contributions: first, they exploit the features of blockchain to extend conventional public key infrastructure with an effective privacy-preserving authentication mechanism. The linkability

between the public key and the real identity of a vehicle is eliminated when a certificate authority (CA) operates the certificate issuance and revocation. Second, the algorithm evaluates the trustworthiness of each vehicle according to the authenticity of broadcasted messages and opinions from other vehicles. All the messages are recorded on the blockchain. The reputation score provides an incentive for internal vehicles to prevent misbehavior and mitigate forged messages' distribution. In [35], Malik et al. proposed a framework for transaction authentication and revocation that authenticates vehicles and speedily updates revoked vehicles' status in the shared blockchain ledger with the PoA mechanism. This method reduces the dependency on CA in the validation process. Feng et al. [36] presented an efficient privacy-preserving authentication model called EPAM that shortens the time of checking Certificate Revocation Lists (CRLs), alleviates the presentation problem during mutual authentication, and achieves privacy properties such as anonymity and unlinkability. Wang et al. [37] utilized a smart contract as a transparent agent to manage the revocations. The user sends a revocation request to the smart contract, and the smart contract periodically transmits valid requests to CA. That scheme directly displays the revocation identity to a smart contract, which may violate the user's privacy and face scalability issues. Lin et al. [38] proposed a novel BCPPA protocol. The Elliptic Curve Digital Signature Algorithm (ECDSA) based on PKI is used in this scheme. The algorithm is based on a public blockchain (Ethereum) for secure communication. Participating vehicles do not need to store "private keys," further reducing verification time and costs. Yao et al. [39] proposed a privacy-preserving blockchain-based certificate status validation scheme called PBCert. The scheme is designed to store all revoked certificates in the OCSP server, and only the minimal control information (namely, certificate hashes and related operation block height) is stored in the blockchain. The scheme uses bloom filters to improve the efficiency of client-side status validation. Rabieh et al. [13], for the scalability of Advanced Metering Infrastructure (AMI) networks, partitioned the network into clusters of SMs. However, there is a trade-off between the overhead of a certificate authority (CA) and the overhead of a cluster. Bloom filters are used to reduce the size of CRL. However, bloom filters will give false positives. The additional distribution of the list of certificates that trigger false positives through the gateway and CA identifies and eliminates false positives, but this adds overhead. Medury et al. [14] and Huang et al. [15] used the cuckoo filter to verify revoked certificates quickly, and they stored certificate information and cuckoo filter coefficients in the blockchain. This method can reduce the cost of certificate storage, but there is a problem of false-positive rate caused by the filter.

To sum up, although many research results have been achieved in certificate revocation. However, there are still two problems in the combination of blockchain and certificate revocation: storage overhead on the blockchain is relatively large. As the number of revoked certificates

increases, the rate of misclassification of certificate status will increase accordingly. These two issues are still not better addressed.

3. Related Algorithms of the Accumulator

In this article, the accumulator is used for revocation factors and status verification. We introduce the related algorithms of cryptographic accumulators as follows:

- (1) $\text{KeyGen}(k, M)$ is a probabilistic algorithm that is executed in order to instantiate the scheme. It takes as input a security parameter 1^k and the upper bound M on the number of accumulated elements and returns an accumulator parameter $P = (P_u, P_r)$, where P_u is a public key and P_r is a private key.
- (2) $\text{AccVal}(L, P)$ is a probabilistic algorithm that computes an accumulated value. It takes as input a set of elements $L = \{C_1, C_2, C_3, \dots, C_m\}$ ($1 < m \leq M$) and returns an accumulated value v , along with some additional information a_c and A_l .
- (3) $\text{WitGen}(a_c, A_l, P_u)$ is a probabilistic algorithm that creates the witness for every element. It takes as input the auxiliary information a_c and A_l and the parameter P and returns a witness W_i for each C_i ($i = 1, 2, \dots, m$).
- (4) $\text{Verify}(c, W, v, P_u)$ is a deterministic algorithm that verifies that a given element is accumulated in the value v . It takes as input an element c , its witness W , the accumulated value v , and the public key P_u and returns YES if the witness W constitutes a valid proof that c has been accumulated in v , or NO otherwise.
- (5) $\text{AddEle}(L^-, a_c, v, P)$ is a probabilistic algorithm that adds new elements to the accumulator and generates a new accumulated value. It takes as input a set of new elements $L^+ = \{c_1^+, c_2^+, \dots, c_k^+\}$ ($L^+ \subset C, 1 \leq i \leq M - m$), auxiliary information a_c , the accumulated value v , and the parameter P , returns a new accumulated value v' corresponding to the set $L^+ \cup L$, witnesses $\{W_1^+, L, W_k^+\}$ for the newly inserted elements $\{c_1^+, c_2^+, \dots, c_k^+\}$, along with new auxiliary information a_c and a_u .
- (6) $\text{DelEle}(L^-, a_c, v, P)$ is a probabilistic algorithm that deletes some elements from the accumulated value. It takes as input a set of elements $L^- = \{c_1^-, c_2^-, \dots, c_k^-\}$ ($L^- \subset L, 1 \leq k \leq m$) that are to be deleted, the auxiliary information a_c , the accumulated value v , and the parameter P and returns a new accumulated value v' corresponding to the set L/L^- , along with new auxiliary information a_c and a_u .
- (7) $\text{UpdateWit}(W_i, a_u, P_u)$ is a deterministic algorithm that updates witness for the elements that have been accumulated in v and v' after adding or deleting operations to the set L . It takes as input the witness W_i , the auxiliary information a_u , and the public key P_u and returns an updated witness W'_i , proving that the element c_i is accumulated in the new value v' .

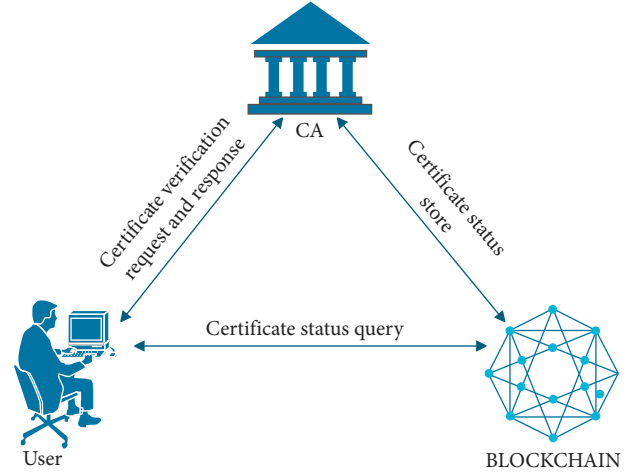


FIGURE 1: System structure.

- (8) Input a_c , A_l , and parameter P and output the witness W_i ($i = 1, 2, \dots, m$) for each C_i ($i = 1, 2, \dots, m$).

4. Certificate Revocation Scheme Based on Blockchain and Accumulator

Verification of certificate revocation status is a critical link in the reliability of public key infrastructure systems. This article's method aims for the reliable distribution and storage of certificate revocation information and designs a public key infrastructure certificate revocation scheme based on blockchain and accumulator. The core is to introduce a field to expand the X.509 certificate structure, namely, the revocation factor. The revocation factor is issued by the blockchain and embedded in the revocation certificate by CA. Blockchain stores the accumulator value. Whenever CA revokes a certificate, it recalculates the corresponding accumulator value and provides a new transaction to put it stored in the blockchain. Since this system only needs to detect revocation information, it only uses the accumulator's accumulation and nonmember certification functions. Newly generated certificates do not need to broadcast new accumulator values. Only each revocation needs to broadcast accumulator values, thus detecting whether certificates have been revoked. When the user checks whether a certificate is revoked, the smart contract is used to verify whether the revocation factor is the factor of the blockchain accumulator value. Finally, get the certificate status.

The whole system includes three entities: certificate authority, blockchain, and user, as shown in Figure 1:

- (1) Certificate Authority (CA): CA is an entity that revokes a certificate. CA responds to the user's certificate request and performs certificate issuance. CA sends a new transaction to the blockchain for each certificate revocation to share the information.
- (2) Blockchain: A distributed ledger stores revocation information, storing accumulator values on the blockchain.

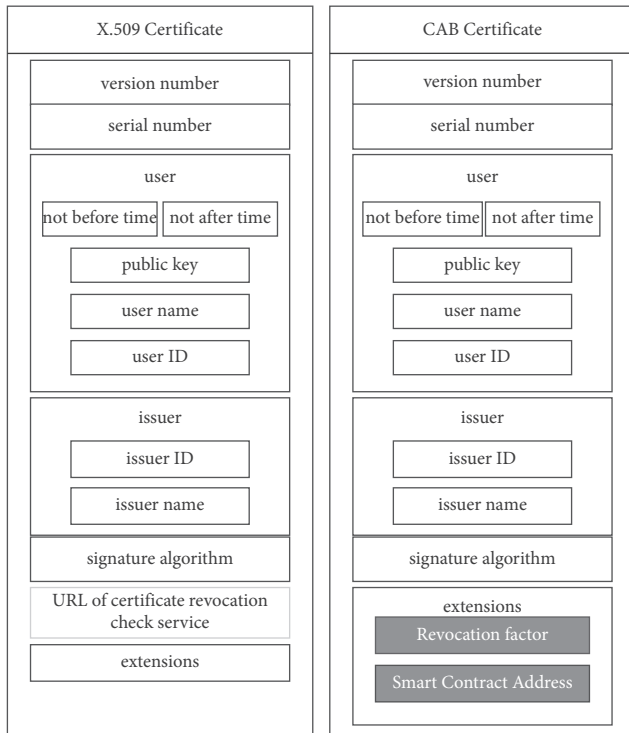


FIGURE 2: X.509 certificate and CAB certificate.

- (3) User: An entity that requests and receives certificates issued by CA. The user submits identity information in the registration stage, and the communication requires identity verification using a digital certificate. The authentication step includes revocation status verification to ensure a valid certificate status before establishing a communication connection.

4.1. Certificate Structure Design. This article designs a new certificate based on the X.509 certificate. X.509 certificate and CAB certificate are shown in Figure 2.

Compared with the traditional X.509 certificate, the main improvements are as follows:

- (1) This article adds a revocation factor to the extension. When CA generates a certificate for the user, the smart contract invoking the licensed blockchain uses the accumulator to generate a revocation factor and an accumulator value. The revocation factor is returned to CA, the certificate is inserted, and the accumulator value is written into the blockchain. CA uses the revocation factor to verify that the certificate is in the accumulator when the client verifies certificate status.
- (2) The certificate designed in this article changes the URL of the certificate revocation check service to the smart contract address. When traditional PKI queries whether a certificate is revoked, it finds the location of the CRL distribution point according to

the URL of the certificate revocation check service. It downloads the CRL list to check the certificate serial number in it to check the certificate status. The scheme changes the URL module to the address of the smart contract. When a client needs to query the certificate status, it only needs to verify whether the unique value contained in the certificate is included in the accumulator value according to the revocation factor provided by the user.

When the customer applies for a certificate, the information is passed to the verification center in this system. After the verification center verifies the customer information, CA issues a certificate as follows:

```

Certificate := SEQUENCE {
    tbsCertificate TBSCertificate, signatureAlgorithm AlgorithmIdentifier,
    signatureValue BITSTRING
}

TBSCertificate := SEQUENCE {
    version v3, -- Certificate version number
    serialNumber CertificateSerialNumber
    default; -- Serial number
    signatureAlgorithmIdentifier default, -- Signature algorithm identification
    issuerName default, -- issuer name
    validity default, -- Certificate validity period
    subjectName CAB-Certification, -- Certificate subject Name
    subjectPublicKeyInfo, SubjectPublicKeyInfo, -- Certificate public key
    issuerUniqueID default, -- Certificate issuer ID
    subjectUniqueID default, -- Certificate subject ID
    extensions Extension -- Extension
}

Extension := SEQUENCE {
    extnID OBJECT IDENTIFIER,
    critical Boolean DEFAULT FALSE,
    extnValue OCTET STRING,
    UndoRF default, -- Revocation search factor
    CPSDistributionPoints default, -- Revocation check address
}

```

The certificate is designed based on X.509 certificate. The certificate carries out regular authentication but differs from the CRL mechanism in the state check part. It provides a witness of the unique value contained in the certificate to make the client believe that the certificate is still valid. Therefore, the revocation factor provided by the blockchain is added to the extension.

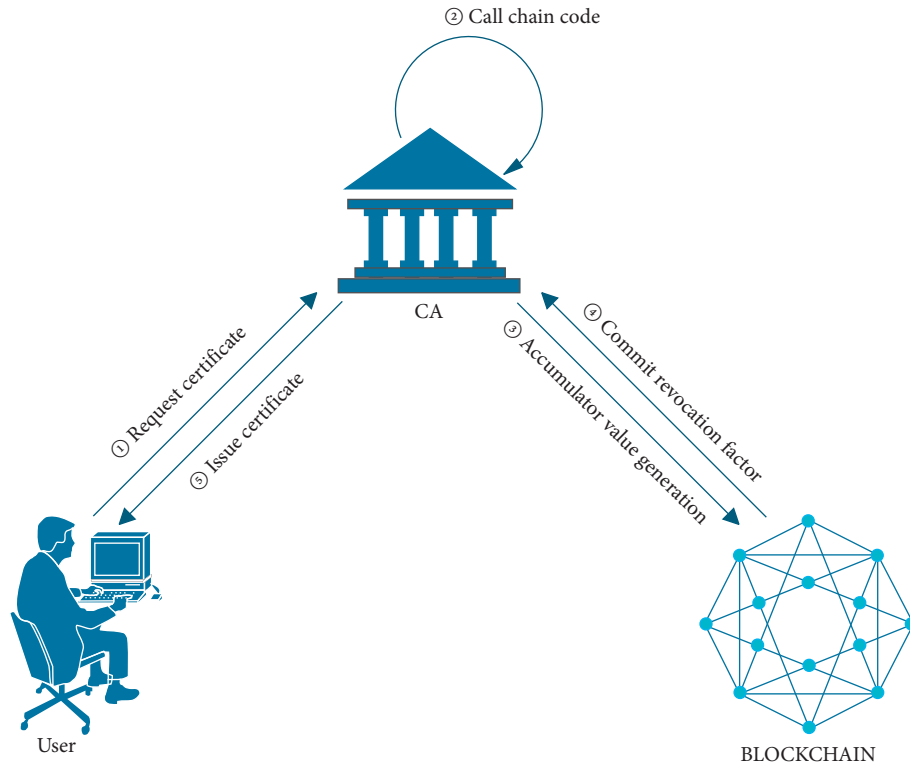


FIGURE 3: Revocation factor generation.

4.2. Accumulator-Based Revocation Factor Generation.

The user initiates a certificate request to CA. CA makes the user information, the public key obtained by the KMC, and the revocation factor generated by calling the smart contract into a certificate. The revocation factor is generated by the accumulator distributed in the blockchain. Accumulator performs the following steps to calculate the accumulated value and the revocation factor: first, it concatenates the certificate's serial number with its issuer's public key to obtain a unique string that prevents problems caused by having the same serial number from different CAs. It computes a relative prime number from a string. Then, add this prime number to the accumulation list through `proveMembership`, and calculate a newly accumulated value `accValue` and the corresponding revocation factor witness, as shown in Figure 3.

A user initiates a certificate revocation request to the CA, which issues a certificate to the user containing a revocation factor. The revocation factor is generated by the blockchain distribution accumulator, which aims to use blockchain to improve the availability of revocation information and reduce the risk of insider threats caused by compromised nodes in the distribution system. The certificate revocation process is shown in Figure 4.

(1) User \rightarrow CA: `req ()`

The user initiates a certificate request to CA.

(2) CA: `verify (certID, keypub)`

After receiving the request, CA assigns a unique certificate to the current operation initiator. The

process of calling the accumulator in the smart contract and generating the accumulator value and revocation factor is as follows.

- (a) `Add (certID)`: Generate a new accumulator value (`accValue'`) by passing in the object and the current certificate accumulator value (`accvalue`).
- (b) `WitCreate (certID, accValue')`: The corresponding revocation factor (witness) is generated through a public key (`keypub`), accumulated value (`accValue`), and element (`certID`).
- (c) `updateAcc (data)`: Write the updated accumulator value to the blockchain.

(3) CA \rightarrow User: `res (certID)`

The system returns the certificate to the user. The specific algorithm is Algorithm 1.

The input parameters of Algorithm 1: `accValue` is the certificate accumulator value on the blockchain, `member` is the certificate member to be added to the accumulator, and `key` is the user key. The output parameters: `accValue'` is the updated accumulator value, and `witness` is the revocation factor, which is the generated member witness. The function of lines 1 to 3 is to get the accumulator value on the blockchain corresponding to the issuer. The function of line 4 is to use `verify ()` function to verify that the member is in that accumulator value. The function of line 5 is to add this member to the accumulator if it is not in this accumulator. The function of line 6 is to generate a new accumulator value `accValue'` and a new revocation factor witness. The function of line 7 is to store the accumulator

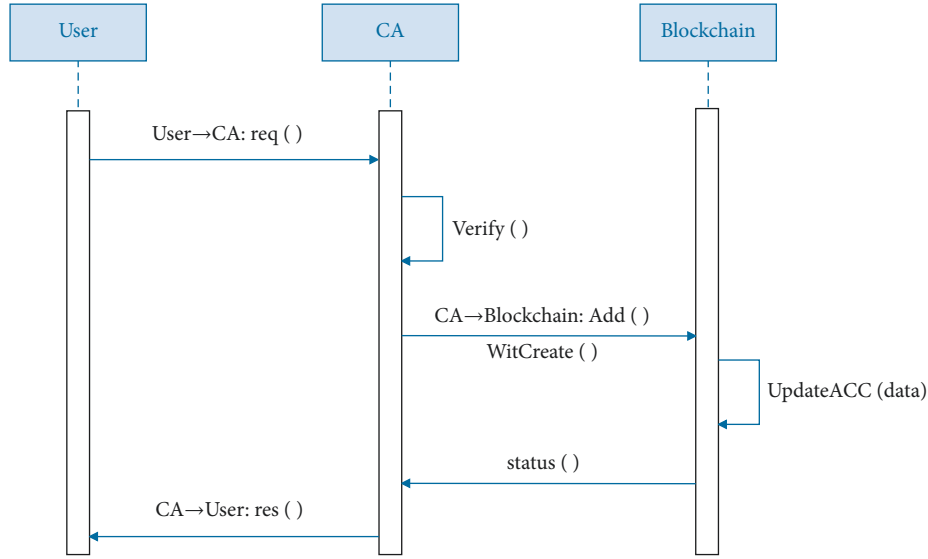


FIGURE 4: Certificate revocation process.

value in the blockchain. The function of line 8 is to return the revocation factor to CA. Instead, tell CA that this member already exists.

4.3. Accumulator-Based Revocation Factor Update. When authenticating, users use revocation factors to prove the validity of their identity. The revocation list is compressed into a short value using an accumulator to verify the certificate's validity. This short value can be easily updated and distributed on a properly instantiated and managed blockchain network. When a certificate is added to the revocation accumulator, both the accumulator value and revocation factor are updated. The process for updating the revocation factor is shown in Figure 5.

- (1) User \rightarrow CA: req(certID, witness, sign(certID, witness)).

As shown in Figure 6, the user first initiates a request to revoke the certificate to CA, where certID is the unique identity certificate of this user, witness is the revocation factor issued by blockchain for this certificate, and sign(certID, witness) represents the user's signature value for their account and revocation factor.

- (2) CA: verify(certID, witness, sign(certID, witness))

After CA receives the request, it performs the verification signature operation to ensure that the certificate corresponding to the current operation initiator belongs to the current user. After verification is passed, the system calls the revocation certificate function in the smart contract. The revocation protocol process is as follows.

- (a) query(certID): the deserialized accumulator object verifies if the certificate is in the blockchain by passing in the revocation factor and current certificate accumulator value.

- (b) revokeFromAcc(certID): call revoke certificate interface of the accumulator to remove the member from the accumulator and recalculate the accumulator value.

- (c) updateAcc(data): write updated accumulator value to the blockchain.

- (3) CA \rightarrow User: res(certID, status)

The system returns the certificate certID and status of the revoked certificate to the user, where status contains revocation success or revocation failure. The algorithm is shown as follows.

When verification credential needs to be regenerated, execute MemWitUp algorithm to update accumulator value and revocation factor. The specific algorithm is Algorithm 2.

The description of Algorithm 2 is as follows. The function of lines 1 to 3 is to get the accumulator value on the blockchain corresponding to the issuer. The function of lines 4 to 6 is to use verify() function to verify that the member is in that accumulator value. The function of lines 7 to 8 is to delete the member using the delete() function if it is included in the accumulator value. The function of line 9 is to generate a new accumulator value and revocation factor using the proveMembership() function. The function of lines 10 to 11 stores the accumulator value in the blockchain and returns the revocation factor to CA. The function of lines 12 to 13 is that the member is not in the accumulator and cannot be updated.

4.4. Accumulator-Based Certificate Status Verification. The user applies for certificate status query operation and submits the user's certificate ID, revocation factor, and signature parameters. After CA receives the request, it performs the verification signature operation to ensure that the certificate corresponding to the current operation initiator belongs to the current user. After

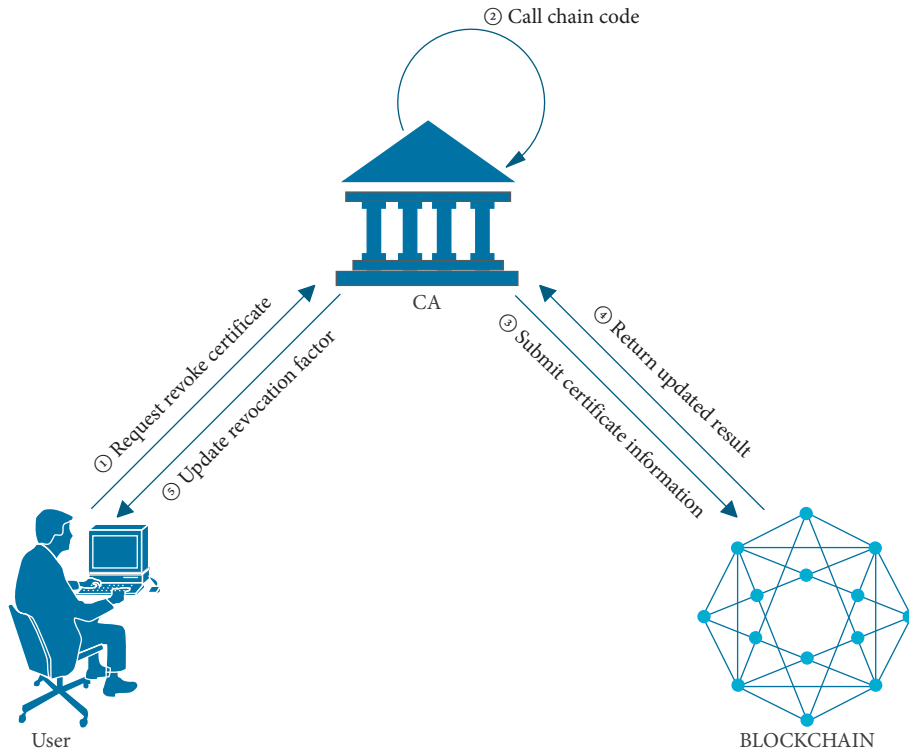


FIGURE 5: Updating revocation factor process.

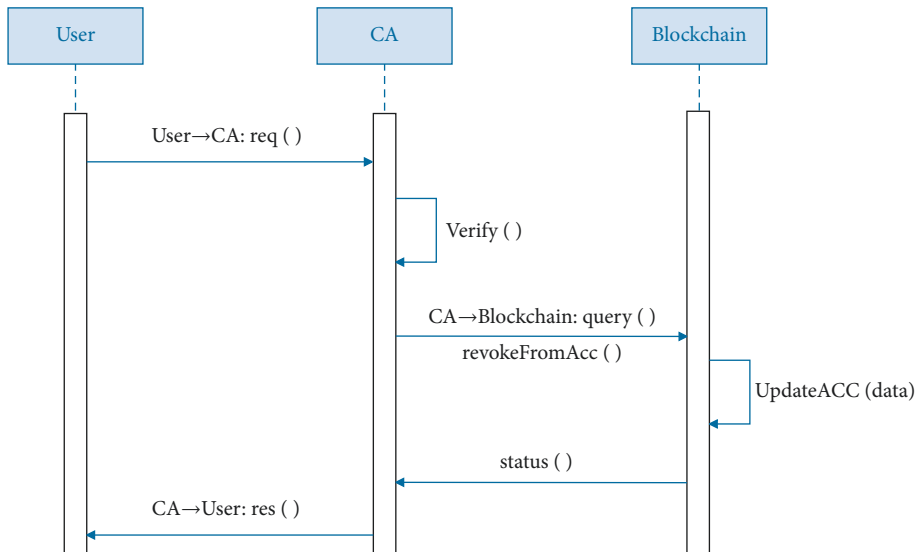


FIGURE 6: Updating accumulator value process.

verification is passed, the revocation status check function in the smart contract is called, and the queried certificate status is returned to the user by executing the member verification algorithm. The specific process is shown in Figure 7.

A certificate query refers to querying the corresponding certificate status from the blockchain through the information given by the user. Protocol design for querying certificate revocation status is shown in Figure 8.

- (1) User \rightarrow CA: queryCert (certID, witness, sign ())
The user applies for certificate status query operation, and submitted parameters represent the user's certID, the revocation factor, and the signature value.
- (2) CA: verify (certID, witness, sign ())
After CA receives the request, it performs the verification signature operation to ensure that the

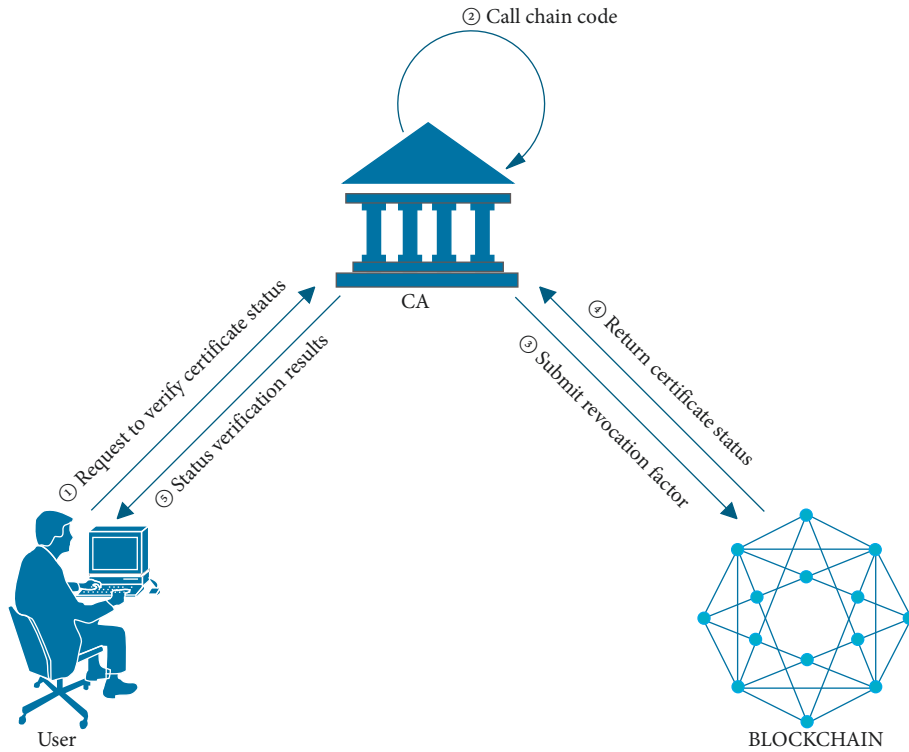


FIGURE 7: Revocation status verification.

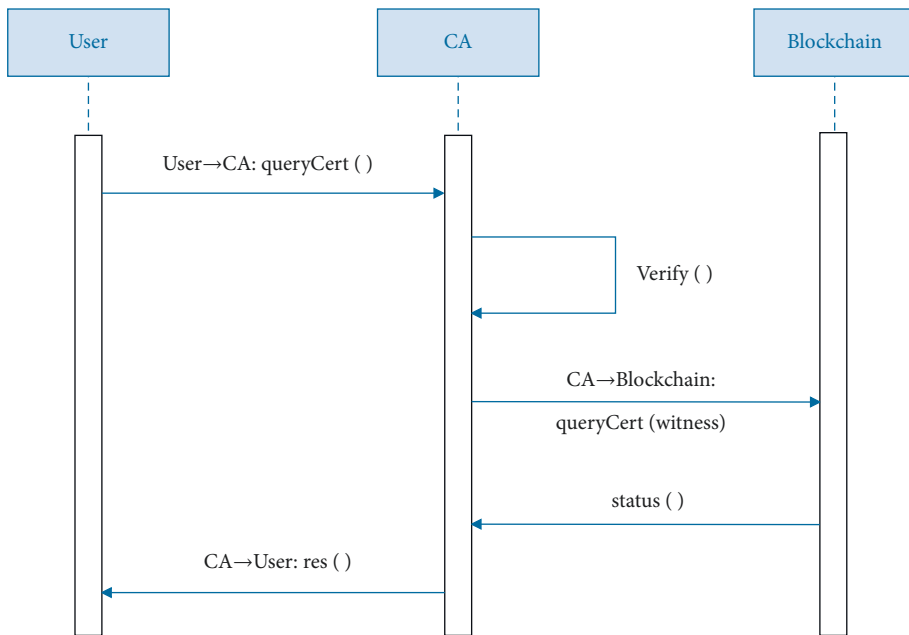


FIGURE 8: Query certificate status process.

certificate corresponding to the current operation initiator belongs to the current user. After the verification is passed, the revocation status check function in the smart contract is called. The protocol flow is as follows.

(a) `queryCert (certID)`: By executing the member verification algorithm in the accumulator, verify

$witness_{c_i}^{c_i} \bmod N = acc_{cert}$ correctness and return the queried certificate status to the user.

(b) `CA → User: {data}` returns the certificate status verified from the blockchain.

The specific algorithm is Algorithm 3.

The description of Algorithm 3 is as follows. The function of lines 1 to 4 is to get the accumulator value on the

```

Input: accValue, member, key
Output: accValue', witness
(1) ChaincodeStub stub = ctx.getStub ();
(2) byte[] objectBytes = stub.getState (Accumulator.class.getSimpleName ());
(3) Accumulator accValue = deserialize(objectBytes); //deserialize the accumulator object
(4) result ← Acc.verify (member); //verify that current certificate exists
(5) accValue ← Acc.add(member); //add member to accumulator
(6) witness = acc.proveMembership (sha256 (cert, n)); //compute new accumulator value and new revocation factor
(7) SendBlockchainTransaction (accValue'); //update accValue to the blockchain
(8) return accValue', witness; //return revocation factor, and the accumulator value is saved to the blockchain

```

ALGORITHM 1: Generative algorithm, provemembership.

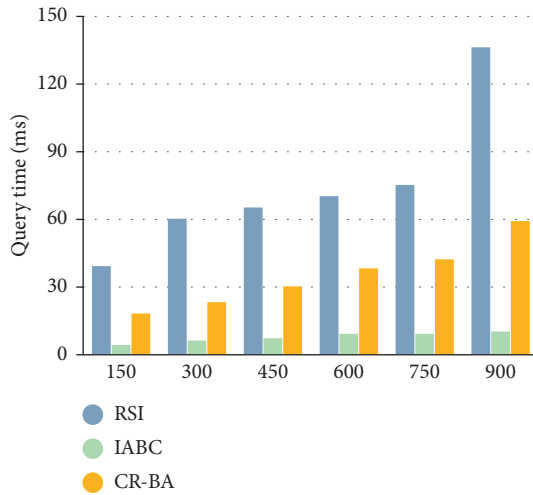


FIGURE 9: Query time of the revoked certificate.

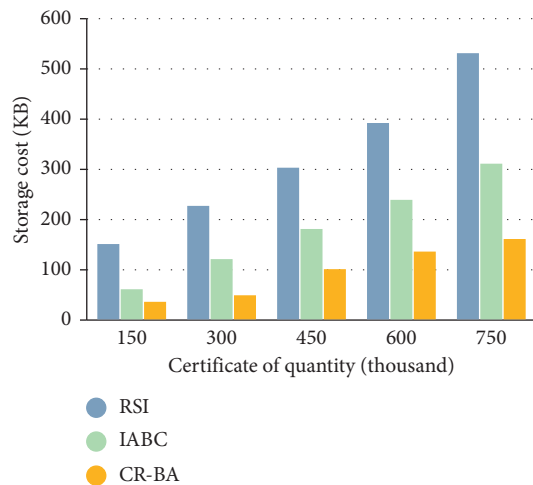


FIGURE 10: Certificate storage cost comparison.

blockchain corresponding to the issuer. The function of line 5 is to use the verify () function to verify that the member is in that accumulator value. The function of lines 6 to 7 is to inform CA that this certificate is still valid if this member is in the accumulator. The function of lines 7 to 8 is to return the certificate status to CA that it has been revoked.

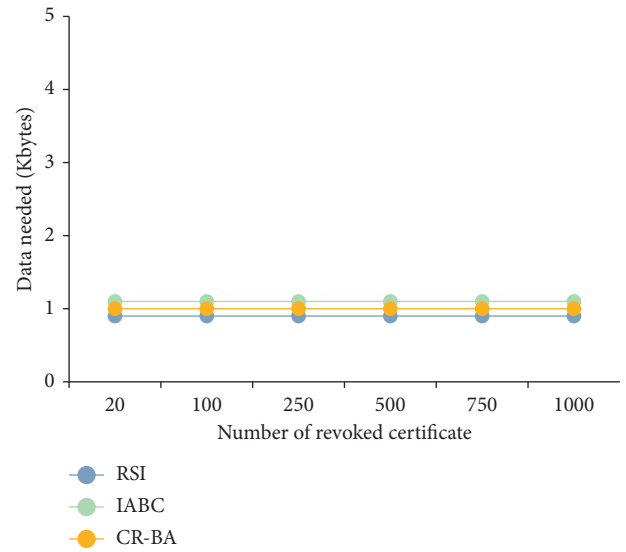


FIGURE 11: Amount of data needed to exchange to provide a response on the revocation status of a nonrevoked certificate.

As a security property of the accumulators, the probability of finding a nonmember witness for an element in accumulating set is negligible. Therefore, in the case of certificate revocation, the server cannot update its witness and prove it is not on the revocation list.

5. Feature Analysis

5.1. Security. The system uses accumulators and blockchain. We use an accumulator to compare the revocation list into a digest, which is updated and distributed through a properly instantiated and managed blockchain network. This small digest allows us to easily distribute validation data, reduce communication overhead and improve system scalability. The accumulator in this article is a secure accumulator based on a strong RSA assumption. Under this assumption, the problem of finding $f(w, m) = w^m \bmod n$ that satisfies the condition is polynomials hard to solve in a short time. Given v and m , finding a w such that $v = f(x, y)$ is difficult, so the accumulator $f(w, m) = w^m \bmod n$ is secure. In this article, we use blockchain to distribute accumulator values and blockchain to improve the availability of revocation information and reduce the risk of

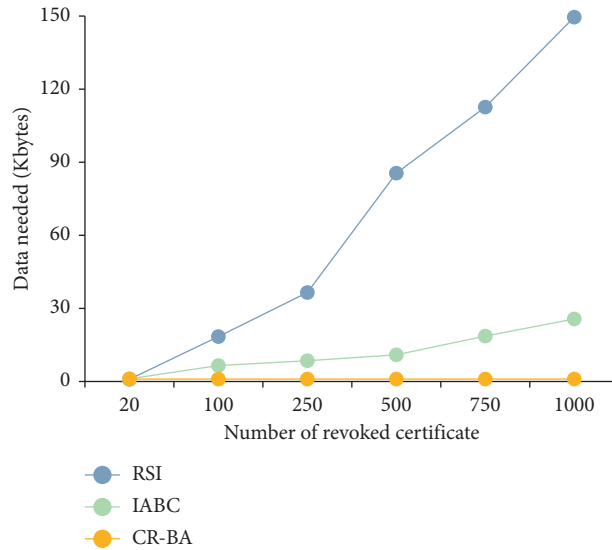


FIGURE 12: Amount of data needed to exchange to provide a response on the revocation status of a revoked certificate.

```

Input: accValue, member, witness, key
Output: f_result or false//validation response
(1) ChaincodeStub stub = ctx.getStub ();
(2) byte[] ojectBytes = stub.getState (Accumulator.class.getSimpleName ());
(3) Accumulator accValue = deserialize (ojectBytes); //deserialize the accumulator object
(4) for member in MEMBER//member are in MEMBER
(5) Boolean verify = acc.verifyMembership (accValue, member, witness, acc.getN ()); //Verify that the current certificate exists
(6) result ← Acc.verify (member); //get the verification result
(7) if (result == 0)
(8)     Acc ← Acc.Delete(member); //delete the certificate
(9)     witness = acc.proveMembership (sha256 (cert, n)); //compute new accumulator value and new revocation factor
(10)    f_result ← provemembership (sha256 (cert, n)); //f_result is the result
(11)    Return f_result; //return update result
(12) else
(13) Return false;

```

ALGORITHM 2: Update algorithm, MemWitUp.

```

Input: accValue, member, witness, key
Output: true or false//validation response
(1) ChaincodeStub stub = ctx.getStub ();
(2) byte [] ojectBytes = stub.getState(Accumulator.class.getSimpleName ());
(3) Accumulator accValue = deserialize (ojectBytes); //deserialize the accumulator object
(4) Acc ← FindBlockchainContract; //get the accumulator value
(5) result ← Acc.verify(member);//verify member is in the accumulator
(6) if (result == 0) {
(7)     return true; //verify successfully
(8) else
(9) return false;

```

ALGORITHM 3: Verification algorithm, verify.

TABLE 1: Comparison of advantages and disadvantages of different methods.

| Methods | Certificate management | False-positive rate | Certificate change |
|--------------------|------------------------|---------------------|--------------------|
| Reference [8] IABC | Exist | Exist | Exist |
| Reference [9] RSI | Not exist | Exist | Not exist |
| Reference CAB | Exist | Not exist | Exist |

possible insider threats caused by compromised nodes in the distribution system.

5.2. Lower Revocation Storage Costs. This solution reduces the cost of storing certificates after introducing the accumulator to the certificate storage. CA acts as an accumulator administrator, aggregating certificates into accumulator values. Accumulator represents the entire set of elements with a single value, and the accumulator value and witness are the sizes of the RSA modulus. Accumulator allows the witness to prove whether the element is in the set, independent of the number of elements in the element.

5.3. No False Positives. The verification algorithm will always return 1 for all honestly generated keys, all honestly calculated cumulative values, and evidence. It is difficult to find membership evidence for elements that do not belong to the set, and it is also challenging to find evidence of non-membership, which is collision-free. Accumulator has undeniability, indicating that computing two conflicting pieces of evidence for elements $x \in X$ or $x \notin X$ is computationally infeasible.

6. Experiment

6.1. Experimental Environment. The experimental model is deployed on a PC with the following configuration: Intel(R) Core(TM) i7-10750H CPU, 16 GB RAM. Ubuntu 18.04 OS, Hyperledger Fabric v1.4.2, chain code using Golang 1.14.12, Docker version number 19.03.2. The chain code uses Golang 1.14.12 and Docker version number 19.03.2.

6.2. Results and Discussion. Before the experimental test, 5000 digital certificates are created in batches. To avoid the contingency of experimental results, repeat five times to calculate the average value. Moreover, it compares certificate revocation methods using bloom and cuckoo filters. Table 1 indicates the comparison of the advantages and disadvantages of different methods. Figure 9 compares the average query time of the revoked certificate. Figure 10 represents the cost comparison of using different blockchain certificate storage methods. Figure 11 represents the amount of data required to respond to an unrevoked certificate's revocation status. Figure 12 represents the amount of data required to respond to a revoked certificate's revocation status.

Figure 9 shows the results of the time required when querying the status of a revoked certificate. Query time of revocation certificate for CR-BA is 35.01 ms on average, and it is also about 1 second to query 10,000 certificates. RSI takes more time than IABC and CR-BA. Because additional

verification is required when querying the status of a revoked certificate. As the number of certificates increases, so does the additional validation required. IABC has the most efficient query but suffers from false positives (providing a positive response while the certificate is still not revoked). Our approach has no false positives, and the response time is within reasonable limits. Figure 10 shows the comparison of certificate storage costs. RSI stores a complete Bloom filter on the blockchain, and IABC stores a cuckoo filter on the blockchain, both as a complete array. CR-BA stores value, which is a small summary as mentioned before. As seen from the figure, the storage consumption of certificate storage in our approach is about half of the other methods compared to others.

Figure 11 shows the amount of data needed to exchange in response to the revocation status of a nonrevoked certificate. All three approaches relied on a simple request and verified response that does not change much when the number of certificates increases. Figure 12 shows the amount of data needed to exchange to respond to a revoked certificate's revocation status. RSI achieves the worst performance. Since RSI needs to download the RSI structure, after the filter provides a positive response, it must ensure no false-positive response. It needs to download all LRSI structures. There are as many LRSI as revoked certificates, so it will take more time and data volume. IABC requires a similar amount of data as our method, which does not change much as the certificate increases. However, in this validation scenario, IABC has the problem of false positives.

It can be seen from the above performance tests that the certificate query time of [9] increases linearly with the increase of test set size. The time consumption of this article fluctuates very little with the increase in the number of certificates, but it lags behind the query speed of reference [8]. References [8, 9] suffer from the probability of misjudgment, but this article does not have this problem. Moreover, compared with other methods, certificate storage in this article consumes less storage. With the increase of blockchain data, the cost of blockchain certificate data storage can be reduced, and it has certain validity and feasibility.

7. Conclusion

This article first analyzes the current certificate revocation mechanism's shortcomings and expounds relevant knowledge of blockchain and accumulators. A public key infrastructure certificate revocation scheme based on blockchain and accumulator is proposed to address problems existing in the current certificate status query method. It take advantages of the efficient and verifiable features of the accumulators and features that support dynamic addition and removal of member elements. It builds a certificate containing the revocation factor by generating a revocation

accumulator in the smart contract. The certificate's fingerprint is written into the accumulator as a member value, which improves query efficiency when the data on the chain is huge and reduces certificate storage overhead.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China, under Grant no. 61373162, and Sichuan Provincial Science and Technology Department Project, under Grant no. 2022YFG0161.

References

- [1] J. P. Monteuiis, B. Hammi, and E. Salles, "Securing pki requests for c-its systems," in *Proceedings of the 2017 26th International Conference on Computer Communication and Networks (ICCCN)*, July 2017.
- [2] R. Wang, J. He, and C. Liu, "A privacy-aware PKI system based on permissioned blockchains," in *Proceedings of the 2018 IEEE 9th international conference on software engineering and service science (ICSESS)*, November 2018.
- [3] J. Leng, M. Zhou, and J. L. Zhao, "Blockchain security: a survey of techniques and research directions," *IEEE Transactions on Services Computing*, vol. 1, p. 1, 2020.
- [4] J. Leng, G. Ruan, P. Jiang et al., "Blockchain-empowered sustainable manufacturing and product lifecycle management in industry 4.0: a survey," *Renewable and Sustainable Energy Reviews*, vol. 132, Article ID 110112, 2020.
- [5] J. Leng, S. Ye, M. Zhou et al., "Blockchain-Secured smart manufacturing in industry 4.0: a survey," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 1, pp. 237–252, 2021.
- [6] S. Jinshan and L. Ru, "Survey of blockchain access control in internet of things [J]," *Journal of Software*, vol. 30, no. 06, pp. 1632–1648, 2019.
- [7] L. Tan, H. Xiao, K. Yu, M. Aloqaily, and Y. Jararweh, "A blockchain-empowered crowdsourcing system for 5G-enabled smart cities," *Computer Standards & Interfaces*, vol. 76, Article ID 103517, 2021.
- [8] L. Tan, H. Xiao, and X. Shang, "A blockchain-based trusted service mechanism for crowdsourcing system," in *Proceedings of the 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, May 2020.
- [9] L. Tan, N. Shi, C. Yang, and K. Yu, "A blockchain-based access control framework for cyber-physical-social system big data," *IEEE Access*, vol. 8, Article ID 77215, 2020.
- [10] J. Leng, D. Yan, Q. Liu et al., "ManuChain: combining permissioned blockchain with a holistic optimization model as Bi-level intelligence for smart manufacturing," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 1, pp. 182–192, 2020.
- [11] C. Fromknecht, D. Velicanu, and S. Yakoubov, "A decentralized public key infrastructure with identity retention," *Cryptology ePrint Archive*, 2014.
- [12] M. Y. Kubilay, M. S. Kiraz, and H. A. Mantar, "CertLedger: a new PKI model with Certificate Transparency based on blockchain," *Computers & Security*, vol. 85, pp. 333–352, 2019.
- [13] K. Rabieh, M. M. Mahmoud, K. Akkaya, and S. Tonyali, "Scalable certificate revocation schemes for smart grid AMI networks using bloom filters," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 4, pp. 420–432, 2017.
- [14] S. Medury, A. Skjellum, and R. R. Brooks, "Scaaps: X. 509 certificate revocation using the blockchain-based scribe secure provenance system," in *Proceedings of the 2018 13th International Conference on Malicious and Unwanted Software (MALWARE)*, IEEE, Nantucket, MA, USA, October 2018.
- [15] S. Huang, L. Jian, and F. A. N. Bingbing, "IABC: a cross-domain authentication method based on blockchain and cuckoo filter," *Journal of Chinese Computer Systems*, vol. 41, no. 12, p. 6, 2020.
- [16] J. Benaloh and M. D. Mare, "One-way accumulators: a decentralized alternative to digital signatures," in *Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques*, Springer, New York, NY, USA, May 1993.
- [17] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proceedings of the Workshop on the theory and application of cryptographic techniques*, Springer, New York, NY, USA, June 1984.
- [18] R. Perlman, "An overview of PKI trust models," *IEEE network*, vol. 13, no. 6, pp. 38–43, 1999.
- [19] S. Tuecke, V. Welch, and D. Engert, *Internet X. 509 Public Key Infrastructure (PKI) Proxy Certificate Profile*, Proposed Standard, 2004, <https://www.ietf.org/rfc/rfc3820.txt>.
- [20] R. Hunt, "PKI and digital certification infrastructure," in *Proceedings of the Proceedings Ninth IEEE International Conference on Networks, ICON 2001*, IEEE, Bangkok, Thailand, October 2001.
- [21] L. Harn and J. Ren, "Generalized digital certificate for user authentication and key establishment for secure communications," *IEEE Transactions on Wireless Communications*, vol. 10, no. 7, pp. 2372–2379, 2011.
- [22] D. Cooper, S. Santesson, and S. Farrell, *Internet X. 509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*, 2008, <https://rfc-editor.org/rfc/rfc5280.txt>.
- [23] R. Housley, W. Polk, and W. Ford, *Internet X. 509 public key infrastructure certificate and certificate revocation list (CRL) profile*, NIST: National Institute of Standards and Technology: Gaithersburg, MD, USA, 2002.
- [24] D. A. Cooper, "A more efficient use of delta-CRLs," in *Proceedings of the 2000 IEEE Symposium on Security and Privacy. S&P 2000*, pp. 190–202, IEEE, Berkeley, CA, USA, May 2000.
- [25] S. Boeyen, T. Howes, P. Richard, and X. Internet, "509 public key infrastructure LDAPv2 schema," *IETF Request For Comments*, vol. 2587, pp. 99–104, 1999.
- [26] J. I.-W. U. Jing and L. I. N. Jing-Qiang, *FENG DENG-GUO Technologies on Public Key Infrastructure*, Science Press, Beijing China, 2008.
- [27] Y. U. A. N. Xue, *Research on Certificate Revocation Mechanisms*, Institute of Software Chinese Academy of Sciences, Beijing China, 2004.

- [28] S. Micali, *Efficient Certificate Revocation*, Technical Memo MIT/LCS/TM-542b, Massachusetts Institute of Technology, Laboratory for Computer Science, MA, USA, 1996.
- [29] M. Naor and K. Nissim, "Certificate revocation and certificate update," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 4, pp. 561–570, 2000.
- [30] P. C. Kocher, *On Certificate Revocation and Validation*-Springer, Heidelberg, Germany, 1998.
- [31] J. Leng, P. Jiang, K. Xu et al., "Makerchain: a blockchain with chemical signature for self-organizing process in social manufacturing," *Journal of Cleaner Production*, vol. 234, pp. 767–778, 2019.
- [32] Q. Hu, M. R. Asghar, and N. Brownlee, "Certificate Revocation Guard (CRG): an efficient mechanism for checking certificate revocation," in *Proceedings of the IEEE 41st Conference on Local Computer Networks (LCN)*. IEEE, Dubai, UAE, November 2016.
- [33] T. Hewa, A. Braeken, M. Ylianttila, and L. Madhusanka, "Multi-access edge computing and blockchain-based secure telehealth system connected with 5G and IoT," in *Proceedings of the IEEE Global Communications Conference (Globecom)*, Taipei, Taiwan, December 2020.
- [34] Z. Lu, Q. Wang, and G. Qu, "BARS: a blockchain-based anonymous reputation system for trust management in VANETs," in *Proceedings of the 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, August 2018.
- [35] N. Malik, P. Nanda, and A. Arora, "Blockchain based secured identity authentication and expeditious revocation framework for vehicular networks," in *Proceedings of the 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, August 2018.
- [36] X. Feng, Q. Shi, Q. Xie, and L. Liu, "An efficient privacy-preserving authentication model based on blockchain for VANETs," *Journal of Systems Architecture*, vol. 117, Article ID 102158, 117 pages, 2021.
- [37] Q. Wang, R. Li, and Q. Wang, "Poster: transparent certificate revocation for CBE based on blockchain," in *Proceedings of the Poster Session of 41st IEEE Symposium on Security and Privacy (SP)*, IEEE, Guangdong, China, June 2020.
- [38] C. Lin, D. He, X. Huang, N. Kumar, and K. KR. Choo, "BCPPA: a blockchain-based conditional privacy-preserving authentication protocol for vehicular ad hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 12, pp. 7408–7420, 2021.
- [39] S. Yao, J. Chen, K. He, R. Du, T. Zhu, and X. Chen, "PBCert: privacy-preserving blockchain-based certificate status validation toward mass storage management," *IEEE Access*, vol. 7, pp. 6117–6128, 2019.

Research Article

A Blockchain-Based Framework for Developing Traceability Applications towards Sustainable Agriculture in Vietnam

Duc-Hiep Nguyen,^{1,2,3} Nguyen Huynh-Tuong ,^{1,2} and Hoang-Anh Pham ^{1,2}

¹Ho Chi Minh City University of Technology (HCMUT), 268 Ly Thuong Kiet Street, District 10, Ho Chi Minh City, Vietnam

²Vietnam National University Ho Chi Minh City (VNU-HCM), Linh Trung Ward, Thu Duc City, Ho Chi Minh City, Vietnam

³Vietnam Blockchain Corporation, District 11, Ho Chi Minh City, Vietnam

Correspondence should be addressed to Hoang-Anh Pham; anhpham@hcmut.edu.vn

Received 30 April 2022; Accepted 28 June 2022; Published 14 July 2022

Academic Editor: Jiewu Leng

Copyright © 2022 Duc-Hiep Nguyen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Recently, many governments in the world have been focusing on building sustainable agriculture to improve the life quality of farmers and significantly increase their income. In Vietnam, however, the farmers still face the problems of “good harvest–low prices, and vice versa” and lack capital for scaling or transforming the production model. One of the main reasons for this phenomenon is that the price of agricultural products does not depend on farmers’ efforts but is based on the purchase price of the trader or the market price. Besides, the farmers also maintain farming habits based on regional culture or follow trendy and profitable agricultural products. Those production strategies make this type of product oversupplied, leading to a down in price shortly, so the farmers’ income will decrease. The above problems stem from the lack of information and communication tools between actors in the agricultural value chain, especially between cooperatives, farmers, and consumers. This paper presents a Blockchain-based framework for developing a traceability solution as an effective method of communication between actors in the agricultural value chain toward a sustainable agricultural model. The proposed approach helps to fully convey the production and distribution of agricultural products and the ability to verify traceability information, thereby helping to increase prices and protect the brand of agricultural products.

1. Introduction

Vietnam is an agricultural country with a rich and diverse product range and many regional specialties. Besides, Vietnam is one of the countries with great potential for agricultural development and leading agricultural export globally. Although the number of agricultural, forestry, and fishery production organizations in Vietnam has increased, their small scale and low investment make production and business efficiency not high. The lack of product consumption or intense consumption fluctuation over time, the phenomenon of “good harvest - low price,” makes most agricultural firms barely cover costs. Therefore, reinvesting is difficult, leading to capital deficiency to expand the production scale and improve business efficiency.

Vietnamese farmers are disproportionately paid for their efforts in small and medium-scale businesses. When they want to improve this, they do not have enough funding or access to capital to expand the production scale. Meanwhile, many countries are moving towards establishing sustainable agriculture to improve the lives of farmers. According to the United States Department of Agriculture (USDA) [1], sustainable agriculture is an agricultural production process that protects the environment, expands the Earth’s natural resources, and improves soil fertility. In particular, sustainable agriculture aims to increase income for farms, promote environmental protection and production management, improve farmers’ life quality, and satisfy human food and fiber needs.

There have been many approaches to solve the above problems of farmers. Each method has different advantages

and disadvantages and is suitable for various goals of sustainable agriculture. For instance, the methods proposed in [2, 3] aim to improve transparency in the supply chain, while the methods in [4, 5] aim to increase the sales volume and enhance trade compliance. Our study will focus on the most practical purposes that directly affect the lives of farmers, which are increasing income for farmers and improving the life quality of farming families and communities. Specifically, building a sustainable agricultural model solves two topical issues of Vietnam's agricultural industry, such as (1) the problem of good harvest, low price - bad harvest, high price; and (2) the problem of capital deficiency for expanding business. These two problems generally stem from the lack of information and exchange methods between actors in the agricultural value chain, especially between cooperatives, farmers, and consumers.

Currently, cooperatives and farmers do not directly interact with consumers but mainly deal with traders. Therefore, they cannot grasp consumers' needs to adjust production plans or improve quality accordingly. More importantly, the lack of information leaves them with no basis to set prices for their agricultural products. The inability to communicate with consumers makes it impossible for them to prove the product quality is commensurate with the cost. In other words, the selling price of the product does not reflect the farmers' effort but is based on the purchase price of the trader or the market price. Depending on traders and having to sell copper at market value makes farmers only earn enough to cover expenses, leading to no motivation to make more efforts to improve product quality.

Besides, the current farming habits of cooperatives and farmers in Vietnam are still based on regional practices or chasing trendy agricultural products that bring high profits. Therefore, it leads to an increase in supply, suddenly exceeding the market's consumption capability, leading to a sharp drop in selling prices. Meanwhile, consumers get difficulty buying high-quality agricultural products amid growing concerns about food safety. According to IBM's report, 71% of consumers are willing to pay 37% more for products with traceability and transparent information [6]. Additionally, the Covid-19 epidemic also changes users' behavior from "in person" to "online" shopping, making it even more difficult for consumers to choose safe and clean agricultural products. Thus, it can be seen that there is a vast gap between supply (farmers) and demand (consumers) in terms of information.

Consequently, it is necessary to have a tool to support bidirectional communication between farmers and consumers to solve the two aforementioned problems for sustainable agriculture in Vietnam. This tool can provide consumers with information about products, cultivation, and distribution processes and send consumers' feedback to the producer. We find that an electronic traceability solution is an appropriate approach. With a traceability system, cooperatives and farmers provide consumers with transparent information about products and quality certification to create a competitive advantage, affirm the product quality commensurate with the price, and build a trusted brand. On the other hand, consumers have enough information to

choose and buy products having transparent information and origin.

With technological advancements, digital systems are being developed with transformative technologies to improve food traceability's speed, accuracy, and effectiveness. One of the most significant limitations is that the current solutions do not demonstrate complete transparency and ensure user accountability when recording traceability information [7]. Meanwhile, Blockchain has been receiving increased interest due to its success in the financial sector and its ability to prohibit data alterations from even the internal system. Technically, Blockchain is a public ledger that records the whole transaction history on a peer-to-peer computer network of the time. All collaborative entities within an ecosystem will share a common ledger that provides data immutability and indisputable accountability for boosting data transparency. Consequently, applying Blockchain technology in agriculture will improve the current traceability process [8–14].

Many existing studies investigate the challenges and benefits of adopting Blockchain. Among them, two survey studies [15, 16] are the most outstanding ones in the smart manufacturing sector. The authors presented twelve valuable metrics ($M1$ to $M12$) that help analyze the differences between various studies on Blockchain adoption. Our study focuses on two metrics, $M6$ and $M12$, while other metrics such as $M1$, $M5$, $M7$, and $M8$ can be achieved based on inherent dominant features of Blockchain and smart contracts. This paper proposes a Blockchain-based framework for developing a digital traceability solution as a transparent and reliable communication between actors in an agricultural value chain toward building sustainable agriculture in Vietnam. Another contribution is to propose an enterprise Blockchain platform to build a traceability software solution. This means that there will be no relation to cryptocurrency, leading to not being limited by legal constraints in Vietnam. The experimental results also indicate that enterprise Blockchain platforms have suitable properties for deploying Blockchain-based applications in the agricultural sector.

2. The Proposed Framework

Storing data on the Blockchain will be executed by sending an interactive transaction to the smart contract. In the agricultural product traceability context, each task in the production farming process will be recorded and stored on Blockchain, leading to massive transactions proportional to the number of users. The proposed framework is designed according to the following objectives.

- (1) **Improving transaction processing capability:** Due to the limitation of the Blockchain platform in terms of the maximum number of transactions processed at a time and the processing time of a block of data [17], the proposed system will be designed to combine similar data into the same transaction or minimize the number of transactions sent to the Blockchain for ensuring the processing performance and accommodating a large number of users.

- (2) **Ensuring data transparency and privacy:** By leveraging the transparent property of Blockchain technology, all traceable data will be stored so that all participants can trace and authenticate on the Blockchain [18]. However, some parts of the data will be encrypted to ensure privacy, especially business-related confidential data.

Figure 1 depicts the overall architecture of the proposed framework based on a 4-layer model that is a revision of our previous work [19]. This framework enables us to develop a traceability software suite including various modules (e.g., administration, data collection, and traceability portal) according to different users' roles via core services at the application layer. These modules will directly interact with each other and revolve around a Blockchain platform. Regarding the three remaining layers, the Blockchain data processing layer combined with the core services module at the application layer will act as a bridge between software applications and the smart contract layer. Meanwhile, the smart contract layer will handle the business logic, and the data will be stored at the Blockchain network layer.

2.1. Application Layer. The application layer consists of a software suite and a group of core services. The software suite includes mobile apps and web-based applications, which enable business owners (i.e., producers or manufacturers) to preconfigure farm descriptions such as crop information, production processes, and raw materials. In addition, the business owner can describe the number of employees in the business, employee identification information, and a separate action account for each employee. Meanwhile, employees (i.e., farmers) use their activated accounts to record daily production activities based on information preconfigured by the business owner. Moreover, a traceability portal will display the traceable information according to the QR code on the product scanned by end-users.

Each product will be identified with a unique code represented in a QR code printed into a stamp and then affixed to the physical product. This method has the advantage of being low cost, suitable for the vast majority of products, and accessible to people who are not very familiar with high technologies (such as farmers). However, this cheap method cannot completely solve the anti-counterfeiting of things because QR codes can be easily copied and pasted on poor-quality products, which means multiple products have the same QR code (i.e., same identifier). To overcome this problem, when scanning the QR code, users will know the genuine distribution locations of the product, along with information on whether the product has been sold or not? Then, users will rely on the difference in the place of purchase (not in the official list that the QR code gives) or the product's status (sold or unsold) to avoid buying the counterfeiting products. Additionally, as an innovative method proposed by Leng et al. [20], composing biological features or edible chemical signatures (besides the physical QR, RFID, and NFC) may be helpful for things counterfeiting in a distributed agriculture context.

Each core service is a collection of related APIs and shares some common tasks. Designing core services can take advantage of inheritance, reduce programming effort, and ensure the consistency of the software system. These core services will communicate directly with the Blockchain data processing layer, send transactions to the Blockchain network for storing data, and interact with smart contract entities.

- (i) **Account Allocation service** provides APIs so that other software modules can create digital objects (e.g., user accounts or production objects) in the database and smart contract entities (e.g., for storing digital identifiers).
- (ii) **ID Allocation service** is trusted for other software modules to request for assigning identifiers to objects. The processing requests from the software will be asynchronous, leading to there can be many requests to generate new identifiers at the same time. Besides, the difference in processing time between the software application and the Blockchain network is also why the identifiers may overlap. Therefore, this service must ensure uniqueness, structure, and a secure coding system.
- (iii) **Traceability service** provides APIs related to the traceability business, such as APIs for managing production areas, production objects, and production logging, or APIs for other tasks related to QR code stamps management.
- (iv) **Preorder service** provides APIs that allow actors in the agricultural value chain to preorder agricultural products.

2.2. Blockchain Data Processing Layer. Conventionally, data will be confirmed almost instantaneously in traditional software systems, while Blockchain transactions will have a certain delay depending on how long a data block is created and confirmed on the network via a consensus mechanism, leading to challenges in data synchronization and performance guarantee. Besides, it is not easy to create transactions, addresses, or interactions directly on the Blockchain due to demands on technical skills. Therefore, we design the Blockchain data processing layer as a communication bridge for processing data to avoid data conflicts arising when users perform relevant functions on the Blockchain network. As shown in Figure 1, this layer provides three groups of functions developed in the form of APIs that interact with smart contract entities deployed on Blockchain networks.

- (i) **Transaction Processing and Management** module provides APIs to perform transaction information retrieval, transaction initialization, block information retrieval, and other related information. These APIs help users without much knowledge of Blockchain technology but still interact with the Blockchain network.
- (ii) **Data Query** module provides APIs to perform data retrieval (e.g., user information, Blockchain address,

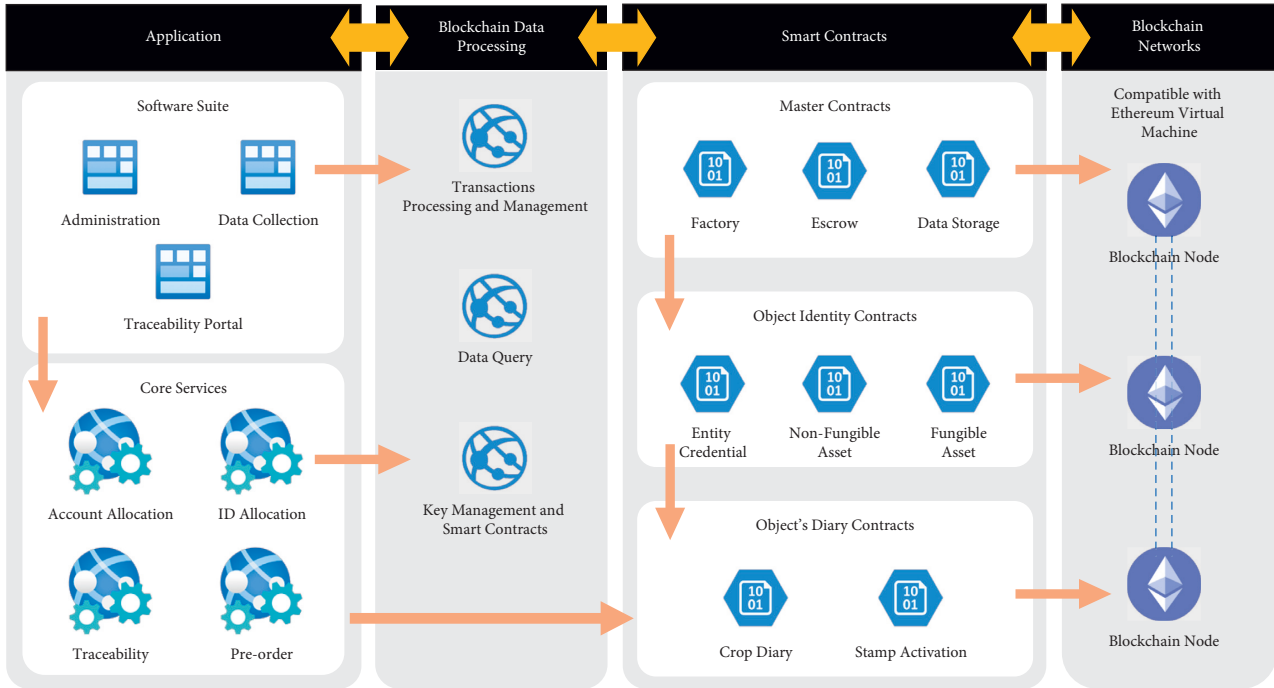


FIGURE 1: The overall architecture of the proposed framework.

or events generated by smart contracts) on the Blockchain network. These APIs help some actors participating in the ecosystem not necessarily become a node in the Blockchain system, which helps to eliminate redundant data and reduces the workload on database synchronization.

- (iii) **Key Management and Smart Contracts** module provides APIs to help manage the secret keys and smart contract entities of whole accounts in the system. The most challenging issue is to provide a simple, transparent, and reliable mechanism to manage the secret keys for low-tech users. It must ensure that only authentic users can know and use their secret keys while the key's manager cannot impersonate and manipulate them.

2.3. Smart Contract Layer. Smart contracts are used to describe the business processes and digitize objects participating in the value chain. Each object or group of objects will be digitized by a smart contract and interact with others. Each smart contract will be assigned a unique address for deploying on the Blockchain network. Transactions will be generated and sent to the corresponding contract address for recording or retrieving the object's identification, description, and related information. Our proposed model organizes eight smart contracts into three groups, including Master, Object Identity, and Object Diary Contracts, as depicted in Figure 2.

First, the contracts in the **Master Contracts** group play a general executive role for the entire Blockchain system architecture.

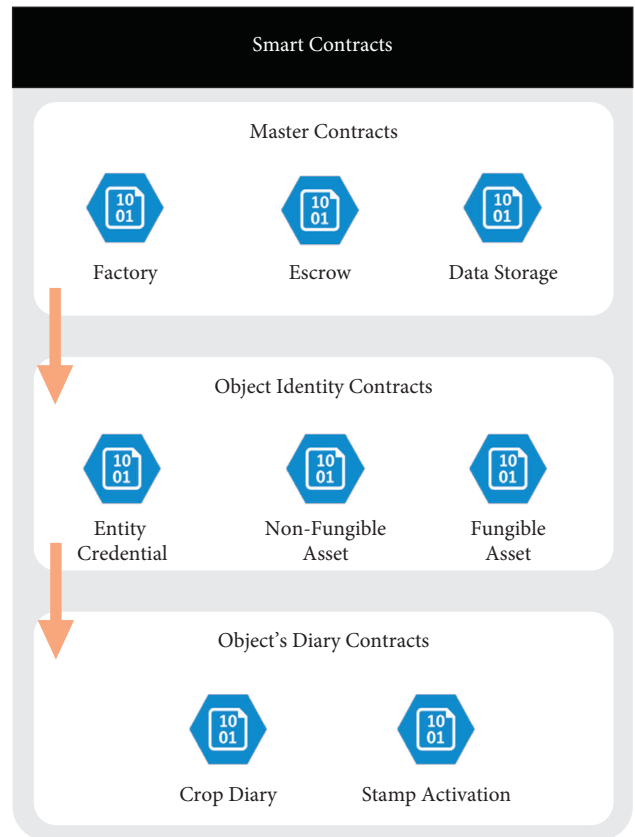


FIGURE 2: The architecture of Smart Contract layer in the proposed framework.

- (i) **Factory** contract will create an instance of the corresponding object identity contract depending on the information provided by the user.

- (ii) **Escrow** contract describes the conditions between the two parties when making a preorder. Once a purchase is made in advance, all transaction information will be modeled into an instance based on this contract.
- (iii) **Data Storage** contract stores important information about the Blockchain system, which keeps the Blockchain platform from being dependent on a centralized system and minimizes the risk of data loss.

Second, the contracts in the **Object Identity Contracts** group are used to map physical objects to digital ones in the system on the Blockchain platform. Three contract types with different variable descriptions and functions represent three different types of objects in the real world.

- (i) **Entity Credential** contract is used to digitalize the identities of real-world users, including farmers, producers, and consumers. Each user will have a corresponding instance of the contract and store his/her identity information. Each instance is identified by a Blockchain address, registered, and stored in the **Data Storage** contract.
- (ii) **Nonfungible Asset** contract is used to supplement the **Fungible Asset** contract for describing the quantities of products having similar characteristics and relationships.
- (iii) **Fungible Asset** contract is used to digitalize physical products such as agricultural products in our case study. These agricultural products will generate new contracts if they have different characteristics.

Third, the contracts in the **Object Diary Contracts** group are responsible for creating and keeping operations related to objects on the Blockchain platform.

- (i) **Crop Diary** contract represents a production crop, including farming activities for an agricultural product. This contract will be tied to a **Fungible Asset** contract.
- (ii) **Stamp Activation** contract records the timestamps of activating the QR code stamp for harvested agricultural products and the Blockchain address of the next recipient in the ecosystem.

2.4. Blockchain Network Layer. There are currently numerous Blockchain networks, among which the well-known ones are Bitcoin, Ethereum, Binance Smart Chain, and Cardano. Each Blockchain network will solve a specific problem, but the most current ones are for finance and payment. A Blockchain network suitable for developing decentralized software applications (dApp) must support programming via smart contracts to support developing decentralized software applications. According to CoinMarketCap's recent statistics, more than one hundred Blockchain projects currently support smart contracts in various programming languages. However, most Blockchain

networks will be designed to be compatible with the Ethereum virtual machine (EVM) due to the completeness and efficiency of the Ethereum network.

It should be noted that public Blockchain networks will require cryptocurrency as a transaction processing fee to maintain the network. For example, it takes about \$0.05 for a simple cryptocurrency transaction (e.g., the transaction for recording farming diaries) in the Polygon Blockchain network, even though it is one of the cheapest transaction fees. Thus, public Blockchain networks are inappropriate choices for implementing traceability solutions. Instead, we will choose an enterprise Blockchain network [21, 22], in which Blockchain nodes will be deployed and operated by an organization. The primary goal is to store data on the Blockchain network transparently without using cryptocurrencies as transaction fees.

3. Implementation

We adopt JavaScript language with the Nodejs Framework to implement modular software on the server side. Meanwhile, desktop applications are implemented using JavaScript's ReactJS framework, compatible with the server side, and can speed up the response to user requests. Besides, we utilize MongoDB as the database because MongoDB is a NoSQL database management system appropriate for storing and querying large volumes of data with high access speed.

3.1. Blockchain Network Selection. By investigating several suitable Blockchain platforms, we choose VBChain since it supports various EVM-compatible Blockchain networks and famous open-source codes such as Open Ethereum or Hyperledger Besu. In this study's scope, we deploy our application software modules on a preconfigured Blockchain network with the setting parameters summarized in Table 1.

3.2. Smart Contracts' Implementation. Smart contracts are implemented based on common standards of the Ethereum community called Ethereum Request for Comment (ERC) to ensure the system's compatibility with other decentralized applications. As depicted in Figure 3, all smart contracts in the proposed framework (see Figure 1) inherit a common smart contract entity according to the ERC-165 standard. In detail, the master contracts, including Factory, Escrow, and Data Storage, will be first deployed on the Blockchain network to operate the whole system. Then, the object identity contracts will be created on the Blockchain network once an object (e.g., a user account, product type, crop diary, or stamp activation) is created in the software application. The object identifier contracts, including Entity Credential, Nonfungible Asset, and Fungible Asset, are described as follows:

- (i) **Entity Credential** contract utilizes ERC-735 for structured storage and verifying claims about that user (such as identifiers). Meanwhile, it adopts

TABLE 1: VBChain’s configuration for deploying our application software module.

| VBChain* | Description |
|------------------------------------|---|
| Blockchain Node’s source code | Open Ethereum (https://openethereum.github.io/) |
| Consensus | Proof of authority (PoA) [23] |
| #Validator node | 3 |
| The min. Processing time per block | 15 (seconds) |
| The max. Gas per block | 240.000.000 |

*<https://vietnamblockchain.asia/>.

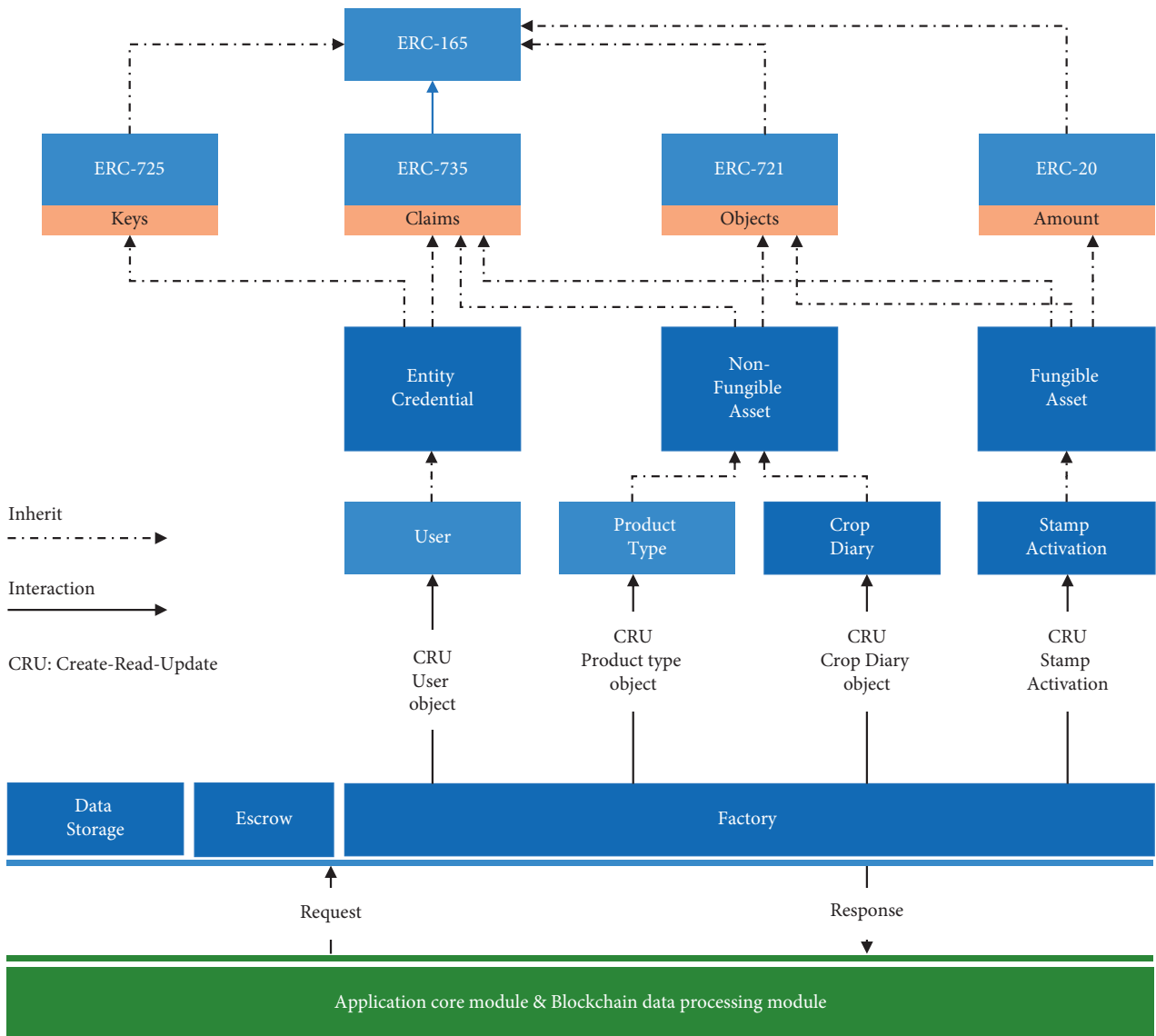


FIGURE 3: The diagram of smart contract implementation.

ERC-725 to manage user-related access keys and smart contracts.

- (ii) **Nonfungible Asset** contract uses ERC-721 to digitalize a real-world object into a digital one on the Blockchain network. Each group of objects (with the same description) will be digitized as an entity of an ERC-721-based smart contract, and the objects in

the same group are distinguished by a unique identifier.

- (iii) **Fungible Asset** contract uses ERC-20 to describe the number of agricultural products on the Blockchain network. An entity of this contract will be attached to an instance of a Nonfungible Asset contract on the Blockchain network.

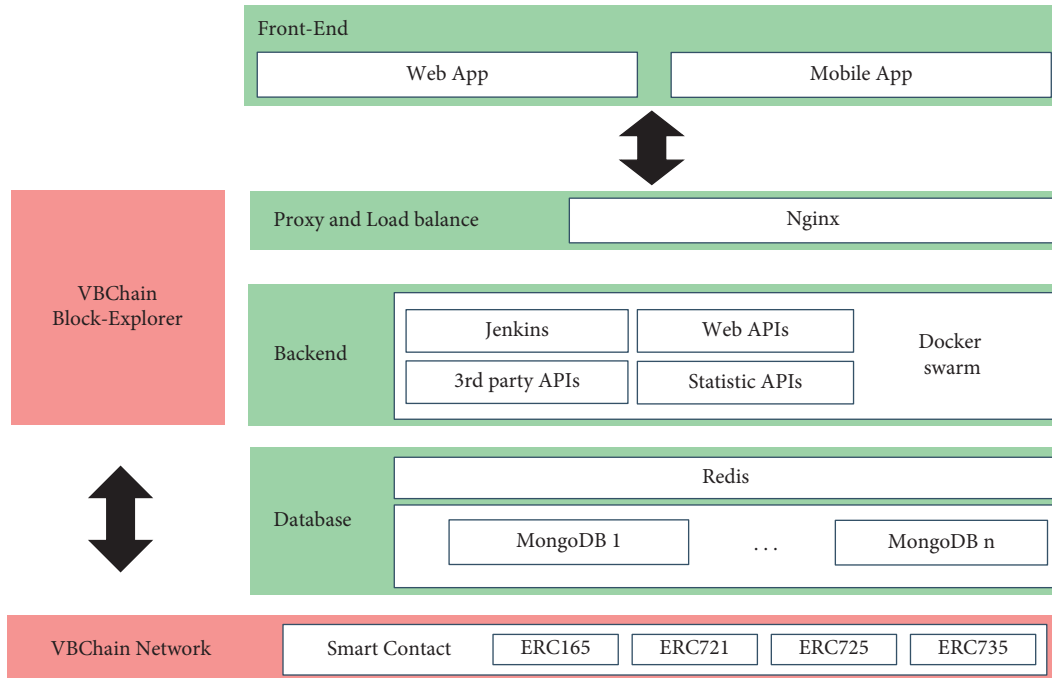


FIGURE 4: The architecture of a prototype of traceability software solution based on the proposed framework.

4. Results and Discussion

4.1. Pilot Deployment. We develop a prototype of traceability software, as shown in Figure 4, to demonstrate and evaluate the proposed framework. Our traceability software solution has been deployed at the Kata Farm Group, located in Dak Lak province, Vietnam, for six months with the recorded information as follows.

- (i) Standardize and digitize 291 production objects
- (ii) Record 3746 production logs
- (iii) Perform 92 stamp activations with a total of 5823 stamps affixed to the final product.
- (iv) Generate a total of 4131 transactions on the Blockchain network

4.2. Transaction Fee Evaluation. Table 2 summarizes a sample process, including the main steps to create products with activated stamps so that users can trace the origin information. As seen, it takes seven transactions to complete the stamp activation step; the total GAS fee for processing these seven transactions is 10,967,832; and the consuming storage space is 3,793 Bytes (i.e., 3.7 KB). Then, we conduct a comparison with several EVM-compatible Blockchain platforms to evaluate the transaction fee.

As shown in Table 3, the cheapest transaction fee (on Polygon Network) is about \$0.68, while the most expensive cost (on the public Ethereum) is much higher, about \$2,926 for a sample case study in Table 2. However, in practice, users will create a massive number of transactions to record farming diaries of various products and seasons. Consequently, the transaction fee will be a significant barrier for users intending to adopt traceability software on the

Blockchain. Thus, we suggest adopting an Enterprise Blockchain platform to deploy the traceability software without any transaction fee.

4.3. Processing Performance and Storage Usage Evaluation. The transaction processing time (txps) [24] can be considered the Blockchain networks' performance, calculated as the number of transactions in a block divided by a block's processing time. Table 4 summarizes the comparison results of several popular EVM-compatible Blockchain networks. Binance Smart Chain has the best performance with 23 txps, which is better than the currently configured VBChain with 14 txps. However, the theoretical processing speed of enterprise Blockchain networks should be much faster than public Blockchain networks because it adopts a smaller number of nodes and the consensus rules with some centralized factors rather than fully decentralization [25]. Therefore, we can investigate further to find the best configuration of VBChain with better performance.

In addition to transaction costs and processing time, storage usage should be considered when applying Blockchain technology [11]. Since all data recorded on the Blockchain network will grow larger and larger over time. Table 5 summarizes the daily storage usage inferred from the processing capability in Table 4, assuming that the system operates at 100% capacity. Accordingly, VBChain can process 1232064 transactions per day and consume 1411 MB of storage space. This result is reasonable because the faster the processing speed is, the more storage usage is.

4.4. The Data Security Issues in the Off-Chain Information Flow. Most conventional software solutions deployed across multiple enterprises will store customer data in the same

TABLE 2: A breakdown of transaction processing costs for a sample product.

| | Tasks | | | | | Total |
|-----------------|------------------|----------------------|-------------------|----------------------|----------------------|-------------------|
| | (*) registration | (1) product creation | (2) crop creation | (3) production diary | (*) stamp activation | |
| #Transactions | 1 | 1 | 1 | 3 | 1 | 7 |
| GAS fee | 4,820,522 | 4,629,425 | 142,216 | 1,116,464 | 259,205 | 10,967,832 |
| Storage (bytes) | 1,679 | 336 | 206 | 1,201 | 371 | 3,793 |

TABLE 3: Transaction fee comparison.

| Criteria | Platform | | | |
|---|---------------------------------|--------------------|---------------------|----------------------|
| | Enterprise blockchain (VBChain) | Ethereum | Binance smart chain | Polygon network |
| Native token | No | ETH | BNB | MATIC |
| Native token price* | No | ~ \$2,900 | ~\$401 | ~\$1.24 |
| Standard GAS price | 0 | $92 * 10^{-9}$ ETH | $5 * 10^{-9}$ BNB | $50 * 10^{-9}$ MATIC |
| Total transaction fee for 10,967,832 GAS | 0 | ~\$2,926 | ~\$22 | \$0.68 |

*Reference price at April 28th, 2022 from CoinMarketCap.

TABLE 4: Processing performance comparison.

| Criteria | Platform | | | |
|---------------------------------|---------------------------------|------------|---------------------|-----------------|
| | Enterprise blockchain (VBChain) | Ethereum | Binance smart chain | Polygon network |
| Block time (second) | 15 | 14 | 3 | 2 |
| GAS limit per block | 240,000,000 | 30,000,000 | 80,000,000 | 20,000,000 |
| GAS used per sample transaction | | 1,116,464 | | |
| Maximum transaction per block | 214 | 26 | 71 | 17 |
| Transaction per second | 14.26 | 1.85 | 23.66 | 8.5 |

TABLE 5: Storage usage comparison.

| Criteria | Platform | | | |
|----------------------------------|---------------------------------|----------|---------------------|-----------------|
| | Enterprise blockchain (VBChain) | Ethereum | Binance smart chain | Polygon network |
| Transaction per second (Table 4) | 14.26 | 1.85 | 23.66 | 8.5 |
| Transaction per day | 1,232,064 | 159,840 | 2,044,244 | 734,400 |
| Data size per sample transaction | 1201 bytes | | | |
| Maximum data storage per day | 1,411 MB | 183 MB | 2,341 MB | 841 MB |

centralized database. This forces enterprises to share data with at least the software vendor, and the security of the data will depend on the software provider's capability. Meanwhile, in our proposed approach, each enterprise will have a private database storing only its data. However, before storing data in the enterprise's database, this data will be hashed and stored on the Blockchain network (shared by all businesses) via smart contracts. This approach helps stakeholders proactively choose and adopt appropriate methods to ensure data security while still retaining the ability to verify the correctness of data by using uneditable hashes stored on the Blockchain network. Additionally, the proposed approach can avoid single-point database failure, which means that other enterprises' data will still be safe when one's database is hacked or exploited.

5. Conclusions

Thanks to Blockchain technology, the proposed framework can provide transparent information helping actors in the

agricultural value chain have a reliable communication method in the digital environment, leading to mutual benefits for all parties toward sustainable agriculture in Vietnam. Applying Blockchain technology in traceability will help protect related stakeholders when something goes wrong. For example, farmers can prove the product's quality or government agencies can handle wrongdoing with reliable and undeniable evidence. The traceability information stored in the Blockchain will be a reliable reconciliation since this information is immutable and transparent without being manipulated by any individual or organization. Any actor can conduct a verification process in the system.

We have also analyzed, evaluated, and compared EVM-compatible Blockchain platforms in terms of technology application. The results indicate that an enterprise Blockchain platform is suitable for practical application in Vietnam because it does not use cryptocurrency to pay transaction fees, so it is not limited by legal constraints in Vietnam.

In the future, we will study incorporating IoT devices [26] to help manufacturers collect data automatically, save human resources, or combine with AI solutions to support information standardization and detect “scans” at suspicious times and places to warn consumers promptly. Additionally, as proposed and demonstrated in [27], the optimization and self-learning ability of Blockchain applications are critical for realizing system sustainability, which is an essential metric of Blockchain technology adoption in the agriculture sector. We will digitalize and integrate the farming process dedicated to each product type to optimize farming activities by adopting AI algorithms in data analytics.

Although most existing approaches adopting Blockchain technology aim to enhance data security and transparency to support traceability. However, Blockchain itself also has security issues elaborated and presented systematically in [28], which give some considerable directions in our further studies.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

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

Acknowledgments

This research was funded by the Ho Chi Minh City University of Technology (HCMUT), VNU-HCM (Grant no. HCMUT-002603-2022). We acknowledge Ho Chi Minh City University of Technology (HCMUT), VNU-HCM for supporting this study.

References

- [1] National Institute of Food and Agriculture, “Sustainable agriculture programs,” *U.S. Code Title*, vol. 7.
- [2] C. Bai, M. Quayson, and J. Sarkis, “Analysis of Blockchain’s enablers for improving sustainable supply chain transparency in Africa cocoa industry,” *Journal of Cleaner Production*, vol. 358, Article ID 131896, 2022.
- [3] M. H. Ali, L. Chung, A. Kumar, S. Zailani, and K. H. Tan, “A sustainable Blockchain framework for the halal food supply chain: lessons from Malaysia,” *Technological Forecasting and Social Change*, vol. 170, Article ID 120870, 2021.
- [4] X. Li, D. Wang, and M. Li, “Convenience analysis of sustainable E-agriculture based on blockchain technology,” *Journal of Cleaner Production*, vol. 271, Article ID 122503, 2020.
- [5] S. Saurabh and K. Dey, “Blockchain technology adoption, architecture, and sustainable agri-food supply chains,” *Journal of Cleaner Production*, vol. 284, Article ID 124731, 2021.
- [6] L. Park, “IBM Study: Purpose and Provenance Drive Bigger Profits for Consumer Goods In 2020,” Media Report, IBM Institute for Business Value, USA, 2020.
- [7] J. Sunny, N. Undralla, and V. Madhusudanan Pillai, “Supply chain transparency through blockchain-based traceability: an overview with demonstration,” *Computers & Industrial Engineering*, vol. 12, Article ID 106895, 2020.
- [8] R. Azzi, R. K. Chamoun, and M. Sokhn, “The power of a blockchain-based supply chain,” *Computers & Industrial Engineering*, vol. 135, pp. 582–592, 2019.
- [9] A. Jeppsson and O. Olsson, *Blockchains as a Solution for Traceability and Transparency*, Thesis, Lund University, Sweden, 2017.
- [10] R. Kamath, “Food traceability on blockchain: walmart’s pork and mango pilots with IBM,” *The Journal of The British Blockchain Association*, vol. 112 pages, 2018.
- [11] K. Behnke and M. Janssen, “Boundary conditions for traceability in food supply chains using blockchain technology,” *International Journal of Information Management*, vol. 52, p. 06, Article ID 101969, 2020.
- [12] P. Jahanbin, S. Wingreen, and R. Sharma, “A blockchain traceability information system for trust improvement in agricultural supply chain,” in *Proceedings of the 27th European Conference on Information Systems (ECIS)*, Stockholm-Uppsala, Sweden, January 2020.
- [13] C. Sekhar Bhusal, “Blockchain technology in agriculture: a case study of blockchain start-up companies,” *International Journal of Computer Science and Information Technology*, vol. 10, pp. 31–48, 2021.
- [14] H. N. Nguyen, M. T. Le, D. H. Nguyen, T. V. Le, N. Huynh-Tuong, and H.-A. Pham, “Towards a blockchain-based framework for traceability in compliance with GS1,” *Science & Technology Development Journal - Engineering and Technology*, vol. 3, no. S11, p. S110, 2020.
- [15] J. Leng, G. Ruan, P. Jiang et al., “Blockchain-empowered sustainable manufacturing and product lifecycle management in industry 4.0: a survey,” *Renewable and Sustainable Energy Reviews*, vol. 132, Article ID 110112, 2020.
- [16] J. Leng, S. Ye, M. Zhou et al., “Blockchain-secured smart manufacturing in industry 4.0: a survey,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 1, pp. 237–252, 2021.
- [17] H. Wang, Z. Zheng, S. Xie, H. N. Dai, and X. Chen, “Blockchain challenges and opportunities: a survey,” *International Journal of Web and Grid Services*, vol. 14, no. 4, p. 352, 2018.
- [18] B. Sezer, S. Topal, and U. Nuriyev, “An Auditability, Transparent, and Privacy-Preserving for Supply Chain Traceability Based on Blockchain,” 2021, <https://arxiv.org/abs/2103.10519>.
- [19] D. H. Nguyen, T. N. Huynh, and H. A. Pham, “Blockchain-based farming activities tracker for enhancing trust in the community supported agriculture model,” in *Proceedings of 11th International Conference on Information and Communication Technology Convergence*, pp. 737–740, ICTC 2020, Jeju, Korea (South), October 2020.
- [20] J. Leng, P. Jiang, K. Xu et al., “Makerchain: a blockchain with chemical signature for self-organizing process in social manufacturing,” *Journal of Cleaner Production*, vol. 234, pp. 767–778, 2019.
- [21] A. Karamchandani, S. K. Srivastava, and A. Srivastava, “Enterprise blockchain: an applications-based comprehensive literature review,” *International Journal of Technology Intelligence and Planning*, vol. 13, no. 1, p. 1, 2021.
- [22] E. Ben, K. L. Brousicmiche, H. Levard, and E. Thea, “Blockchain for enterprise: overview, opportunities and challenges,” in *proceedings of the 13th International Conference on Wireless and Mobile Communications*, vol. 07, ICWMC 2017, Nice, France, June 2017.

- [23] K. Toyoda, K. Machi, Y. Ohtake, and A. N. Zhang, "Function-level bottleneck analysis of private proof-of-authority Ethereum blockchain," *IEEE Access*, vol. 8, pp. 141611–141621, 2020.
- [24] A. Baliga, I. Subhod, P. Kamat, and S. Chatterjee, "Performance Evaluation of the Quorum Blockchain Platform," 2018, <https://arxiv.org/abs/1809.03421>.
- [25] C. Arslan, S. Sipahioğlu, E. Şafak, M. Gözütok, and T. Köprülü, "Comparative analysis and modern applications of PoW, PoS, PPoS blockchain consensus mechanisms and new distributed ledger technologies," *Advances in Science, Technology and Engineering Systems Journal*, vol. 6, no. 5, pp. 279–290, 2021.
- [26] K. Dey and U. Shekhawat, "Blockchain for sustainable e-agriculture: literature review, architecture for data management, and implications," *Journal of Cleaner Production*, vol. 316, Article ID 128254, 2021.
- [27] J. Leng, D. Yan, Q. Liu et al., "ManuChain: combining permissioned blockchain with a holistic optimization model as Bi-level intelligence for smart manufacturing," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 1, pp. 182–192, 2020.
- [28] J. Leng, M. Zhou, J. L. Zhao, Y. Huang, and Y. Bian, "Blockchain security: a survey of techniques and research directions," *IEEE Transactions on Services Computing*, p. 1, 2022.

Research Article

Selection Strategy of Mining Pool under Various Different Payment Mechanisms

Tan Xing-Hong ¹, Fu Lu-Xia ¹, Zhang Zhuang ¹, and Tan Liang ^{1,2}

¹College of Computer Science, Sichuan Normal University, Chengdu, Sichuan 610101, China

²Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China

Correspondence should be addressed to Tan Liang; jkxy_tl@sicnu.edu.cn

Received 6 May 2022; Accepted 7 June 2022; Published 6 July 2022

Academic Editor: Jiewu Leng

Copyright © 2022 Tan Xing-Hong et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Blockchain is becoming increasingly popular and has received extensive attention in various fields. In a proof-of-work-based blockchain, miners usually choose to join a mining pool for mining to gain revenue. Different mining pools may use other payment mechanisms, and each miner can earn different revenues in different pools. There are currently four common payment mechanisms used by mining pools to distribute mining revenue, namely, PPS, PPLNS, PPS+, and FPPS, and there are no relevant research results on the selection strategies of these four payment mechanisms. To this end, this paper models the pool selection problem as a risky decision problem and proposes the selection strategies of these four mining pool payment mechanisms. Firstly, miners' income under the four payment mechanisms is given; then, a mining pool selection strategy based on the change of computing power is constructed based on the Laplace criterion; finally, the proposed strategy is verified and analyzed by simulation. The experiments show that the proposed mining pool selection strategy is effective. The results of this paper can provide an essential reference for miners when making pool selection decisions.

1. Introduction

In 2008, Satoshi Nakamoto published a white paper on Bitcoin [1]. Following this, blockchain technology is becoming increasingly popular and attracting wide attention in various fields [2–6]. Due to its decentralized, de-trusted, collectively maintained, and tamper-evident properties [7], blockchain technology has been used in several areas such as medical information security management [8], smart city [9, 10], smart manufacturing [11, 12], access control framework system [13], and trusted service mechanism [14]. Bitcoin is a typical application of blockchain technology. Bitcoin mining is the process of obtaining Bitcoins, which is based on the principle of using computer computing power to solve cryptographic puzzles and use this to generate blocks that are eventually rewarded without blocks of Bitcoins. In a proof-of-work-based blockchain network, miners participate in solving a mathematical puzzle by contributing their computing power. If they are able to arrive at a solution

that satisfies the practical block difficulty of the blockchain network, they are considered to have found a new block. They are rewarded for their contribution of their computing power [15]. We call the solution to a mathematical puzzle that satisfies the difficulty of generating a new block a full workload proof. For miners, solving a full workload proof alone is very difficult due to the sheer amount of computing power on the blockchain network, and to improve the efficiency of the solution, miners usually join the mining pool and contribute their computing power as a way to improve the stability of their revenue. The pool administrator, on the other hand, usually divides the mathematical puzzle that satisfies the difficulty of generating a new block into multiple less difficult mathematical problems and asks the miners in the pool to submit a solution to this less difficult mathematical problem, which is usually referred to as a partial workload proof [16]. When a miner in the pool obtains and submits a solution as a full workload proof during the calculation process, the pool is considered to have mined a

new block, and the block reward will be settled to the corresponding miner in the pool in accordance with the pool's payment mechanism. In practice, the issue of pool selection is actually the first problem miners face in pool mining, and for miners in a pool, the payment mechanism used by the pool has a significant impact on their earnings. For example, in the article [17], during the selection process of two payment mechanism mining pools, PPS and PPLNS, the PPLNS pool is settled according to the last N partial workload proofs submitted by miners, and the change of N leads to the shift of miners for the pool selection.

We perform a revenue analysis of four common mechanisms commonly used by mining pools in practice, PPS, PPLNS, PPS+, and FPPS [18–22]. We refer to the revenue generated by the packaging transaction when a block is generated as the miner's fee, and the PPS mining pool deducts the theoretical income from the mining fee and proceeds the income settlement according to the proportion of miners' calculation power in the mining pool. PPLNS mine pool will make a settlement with the miners who have recently submitted N partial proof of work after deducting the mining fee from the aggregate of the actual block production reward of the mining pool after several rounds. The PPS+ mining pool mechanism combines PPS and PPLNS payment modes. The block payment reward is settled according to PPS and the number of blocks produced by the mining pool theory. The miner's fee is settled based on the actual mining fee produced by the mining pool and N partial workload proofs submitted by miners recently. The FPPS mining pool mechanism is also known as full PPS, where the pool's block rewards and miner fees are settled according to the PPS model. There are already research results comparing the PPS and PPLNS, PROP, and PPLNS payment mechanisms in the context of independent mining pools adopting different reward distribution systems. The PROP mechanism is based on the principle that when a mining pool finds a new block, it allocates a corresponding amount of revenue to the miner according to the amount of computing power he has contributed to the pool. The only difference between the principle of the PPS mechanism and PROP is that in PPS, the payout is distributed according to the size of the miner's contribution, regardless of whether or not the current pool has found a new block. Article [17] models the pool selection problem faced by miners in the case of two payment mechanisms, PPS and PPLNS, as a risky decision problem based on the maximum likelihood criterion, based on the phenomenon that miners have different returns for selecting pools with varying mechanisms of reward, and investigates the impact of the variation of N in the PPLNS mechanism on miners' optimal pool selection decision, with N referring to the number of partial workload proofs taken at the end of each settlement in the PPLNS mechanism. The article [16], on the other hand, addresses the pool selection problem faced by miners under the competitive relationship between two payment mechanisms, PROP and PPLNS, and builds a pool selection model based on the risk decision criterion, calculating the

miners' returns in different pools. And the article derives the optimal selection strategy using the maximum likelihood criterion and the expected value criterion, respectively. It investigates the influence of pool computing power and reward mechanism on the miners' optimal selection strategy.

This paper aims to conduct a comparative analysis of four payment mechanisms for Bitcoin mining, model the problem of choosing a mining pool under the four payment mechanisms faced by miners as a risky decision problem, and construct a pool selection model based on Laplace's criterion. This study focuses on the pool selection of four common mining pool payment mechanisms, PPS, PPLNS, PPS+, and FPPS, in the same blockchain network competing for computing power resources, highlighting the impact of computing power allocation on pool selection. Unfortunately, the starting point of this study is different from other mine pool selection strategy papers, so it is not compared with the results of the papers with other selection strategies. The relevant selection strategy research article [16, 17] focuses on selecting the mine pool strategy under N change in the PPLNS mechanism. Still, our study focuses more on the influence of computing power distribution change. We regard the change of N as an equal possibility value. The miners who chose the PPLNS payment mechanism submitted only two results: the proof of work falling into the value range of N or not falling into the value range of N . We highlight the influence of the computing power distribution change on the choice of the four mine pool strategies. The results of this paper can provide an essential reference for miners when making pool selection decisions. Using computational experiments, we validate the effectiveness of our proposed mining pool selection strategy.

2. Related Work

The purpose of the study of mining pool strategies is to improve miners' profitability. Currently, there are several mining pools in the market. Different pools have the different computing powers and may adopt different payment mechanisms, which leads to the fact that miners cannot get the same profit from different pools. Research on mining pool strategies has produced several results in mining pool incentive strategies, mining pool attack-defense strategies, and mining pool selection strategies.

First, in terms of incentive strategies for mining pools, articles [15, 23, 24] aim to encourage honest mining, reduce the waste of computational resources due to miners' malicious behavior, and address the problems of inefficient mining and unfair returns by providing miners with a game to think about to encourage honest and cooperative mining to improve returns. In paper [25, 26], mining strategies that can improve the profits and reduce dishonest miners' profits are proposed to enhance the overall mining profits of the mining pool.

Secondly, in terms of mining pool attack and defense strategies, the article [27, 28] defines and analyzes the game theory problem of the prisoner's dilemma arising from a mining pool attack, uses the results of the congestion game to establish a pure Nash equilibrium, gives an efficient

algorithm for finding such an equilibrium, and calculates the miner's gain in the case of a mining pool attack. Articles [29–31] address the problem that miners will choose to attack each other due to the pursuit of superior strategies and high returns. By building a model of mining pool defense strategy and comparing the expected cooperative returns with attack returns, they reduce the wastage of computing power and the phenomenon of driving down mining returns caused by miners when conducting attacks, promote the cooperation of miners, and ensure stable returns of mining pools. The article [32] designs a new blockchain and provides a trust model for it to address the problem of internal attacks on mining pools. Articles [33–35] then provide mining pool attack strategies to show the vulnerability of existing mining pool structures, with the intention of deciphering the problem of the miner's prisoner's dilemma and providing advice to miners when choosing to attack or cooperate.

Finally, in terms of pool selection strategies, articles [16, 17, 36, 37] investigate how miners choose pools in blockchain networks under the influence of different block mining strategies, reward allocation mechanisms, computing power, and latency, and use pool selection strategies to obtain the best returns.

There has been a lot of research and research findings on mining pool incentive strategies, attack and defense strategies, and selection strategies. However, there is a lack of research in the literature on pool selection strategies for multiple different payment mechanisms, so there is some value in this study [38–41].

3. Mining Pool Selection Strategies

3.1. Four Mining Pool Payment Mechanisms and Miner's Earnings. A comparison of the four mining pool payment mechanisms is shown in Table 1.

The mining fee is the fee charged by a mining pool for conducting mining, usually expressed as δ . The miner's fee is the transaction fee for all transactions obtained by packing this block, in addition to the block reward. The lucky value is the ratio of the pool's actual block yield to the theoretical block yield.

$$\text{Lucky Valu} = \frac{\text{Actual Benefits}}{\text{Threoretical Benefits}} \times 100\%. \quad (1)$$

For the purposes of this paper, we assume that all four pools receive a fixed miner's fee of ϕ per block packed. In this paper, we assume that the lucky value of mining is 100%; i.e., the expected revenue on blocks is equal to the actual return on blocks. Suppose there are a total of four mining pools V_1 , V_2 , V_3 , and V_4 on the blockchain network, taking PPS, PPLNS, PPS+, PPS+, and FPPS payment mechanisms, respectively, with each payment mechanism accounting for e_1 , e_2 , e_3 , and e_4 of the blockchain network's computing power, respectively, then $e_1 + e_2 + e_3 + e_4 = 1$. Assume that the blockchain network is a round from the start of mining new blocks to the end of blocking, and that each round of blocking lasts for a fixed time T , for a total of K rounds. The blockchain network has a fixed total block reward of R for each round. Total mining revenue per mining pool for round

TABLE 1: Comparison of four common mining pool payment mechanisms.

| Payment mechanisms | Block rewards | Miners' fees | Supported mining pools |
|--------------------|-------------------|-------------------|------------------------|
| PPS | Theoretical value | No distribution | ViaBTC |
| PPLNS | Actual value | Actual value | AntPool, ViaBTC |
| PPS+ | Theoretical value | Actual value | AntPool, F2Pool |
| FPPS | Theoretical value | Theoretical value | BTC.com |

K is $S = e_i KR$, $i = \{1, 2, 3, 4\}$. Assume that each pool miner fee K round of total revenue is $S_c = K\phi$. Assume that each pool has exactly M partial workload proofs per round, and for ease of calculation, assume that each miner provides only one partial workload proof per round, and that the position of the partial workload proof submitted by the miner among the M partial workload proofs is random and this position is the same in K rounds, and let the probability that the partial workload proof submitted by the miner in each round is at position i be p_i , $p_i = (1/M)$.

3.1.1. PPS (Pay per Share). PPS payment mechanism of the pool is based on the miner's computing power in the pool; an estimate of the daily output can be obtained in the pool, not to allocate the miner's fee; the pool will retain δ percentage of mining fees.

The mining pool that chooses the PPS payment mechanism for mining is defined as event V_1 . The mining revenue for miners in event V_1 is

$$v_1 = \frac{1}{M} e_1 (1 - \delta) KR. \quad (2)$$

Since the model does not calculate miner's fees, the miner's gain in miner's fees in event V_1 is

$$S_c(v_1) = 0. \quad (3)$$

3.1.2. PPLNS (Pay per Last N Shares). The PPLNS payment mechanism mining pool will settle with the miner who submitted the last N partial workload proofs after several rounds by adding up the actual block bonus of this pool with the actual miner's fee, minus the mining fees. The revenue will be allocated to the miner who submits the last N partial workload proofs, as shown in the literature [17], $N = (k - 1)M + j$, $k \in [1, K]$, $j \in [1, M]$, where k refers to the number of rounds N contains and j refers to the number of partial workload proofs that N contains when a full round is not included.

The mining pool that chooses the PPLNS payment mechanism for mining is defined as event V_2 , in which miners have two revenue states a and b .

In state a , the probability that some of the workload proofs submitted by miners at the settlement of each reward do not all fall within the last N candidates is

$$p_{2,a} = \frac{M-j}{M}. \quad (4)$$

The miner's mining revenue in state a of event V_2 is

$$v_{2,a} = \frac{k-1}{N} e_2 (1-\delta) KR. \quad (5)$$

The miner's gain in miner's fee for state a of event V_2 is

$$S_c(v_{2,a}) = \frac{k-1}{N} (1-\delta) K\varphi. \quad (6)$$

In state b , some of the workload proofs submitted by miners at each settlement of the reward fall within the last N candidates with the probability are

$$p_{2,b} = \frac{j}{M}. \quad (7)$$

The mining revenue of the miner in state b of event V_2 is

$$v_{2,b} = \frac{k}{N} e_2 (1-\delta) KR. \quad (8)$$

The miner's gain in miner's fee for state b of event V_2 is

$$S_c(v_{2,b}) = \frac{k}{N} (1-\delta) K\varphi. \quad (9)$$

3.1.3. PPS+ (Pay per Shares plus). The PPS+ payment mechanism pool combines both PPS and PPLNS models, with PPS settling the pool's theoretical block, payout rewards, and PPLNS determining the miners' fees generated by the pool's actual block payouts.

The mining pool that chooses the PPS+ payment mechanism for mining is defined as event V_3 , and from Sections 3.1.2 and 3.1.3, the mining returns of miners in event V_3 are

$$\begin{aligned} v_{3,a} &= \frac{1}{M} e_3 (1-\delta) KR, \quad p_{3,a} = \frac{M-j}{M}, \\ v_{3,b} &= \frac{1}{M} e_3 (1-\delta) KR, \quad p_{3,b} = \frac{j}{M}. \end{aligned} \quad (10)$$

The miner's gains from the miner's fees in event V_3 were

$$\begin{aligned} S_c(v_{3,a}) &= \frac{k-1}{N} (1-\delta) K\varphi, \quad p_{3,a} = \frac{M-j}{M}, \\ S_c(v_{3,b}) &= \frac{k}{N} (1-\delta) K\varphi, \quad p_{3,b} = \frac{j}{M}. \end{aligned} \quad (11)$$

3.1.4. FPPS (Full Pay per Shares). FPPS payment mechanism mining pools, also known as full PPS mining pools, are settled according to theoretical earnings after deducting mining fees for block rewards and miner fees.

The mining pool that chooses the FPPS payment mechanism for mining is defined as event V_4 , and from Sections 3.1.2 and 3.1.3, it follows that the miner's mining revenue in event V_4 is

$$v_4 = \frac{1}{M} e_4 (1-\delta) KR. \quad (12)$$

The miner's gain from the miner's fee in event V_4 is

$$S_c(v_4) = \frac{1}{M} (1-\delta) K\varphi. \quad (13)$$

3.2. Mining Pool Selection Strategy under the Laplace Criterion. Laplace's criterion: Laplace's criterion, also known as the equal likelihood criterion, is based on the assumption that multiple states $C_j, j = \{1, 2, 3, \dots, n\}$ of event $V_i, i = \{1, 2, 3, \dots, m\}$ have the same probability $P(C_j)$ of occurring, i.e., $P(C_j) = (1/n), j = \{1, 2, 3, \dots, n\}$, the total payoff of the event in each state is denoted as v_j^* , and then the expected payoff of the event is $E(V_i) = \sum_{j=1}^n P(C_j) * v_j^*$, $j = \{1, 2, \dots, n\}$, where m refers to the number of possible events and N refers to the number of states in which the event may occur. The optimal choice of Laplace's criterion should satisfy $E_{V_i}^* = \max_{V_i=\{V_1, V_2, \dots, V_m\}} E(V_i)$ in the expected revenue value of each event.

In Laplace's criterion, when the decision-maker cannot determine which event is easy to occur in the decision-making process, he has to think that the opportunity of various events is equal; that is, the probability of occurrence is equal. In other studies of mining pool selection strategies [16, 17], the variation of N in the PPLNS mechanism is used as the primary variable. This study did not focus on N change to highlight the effect of computing power allocation change on choosing four mine pool strategies. We believe that the change in N will only lead to two outcomes, and the probability of the two outcomes is equal: the partial proof of workload submitted by miners falls into the value range of N or not into the value range of N . This understanding is usually more in line with ordinary miners' knowledge of the various mining mechanisms when choosing a pool since not all miners are experts in understanding the mining pool. Our study controls for the results caused by changes in N . This control is more consistent with the requirements of Laplace decision-making, so it is reasonable to choose Laplace decision research in this paper.

Based on the Laplace criterion, the benefits of each of the four mining pool mechanisms can be summarized in conjunction with Section 3.1 as shown in Table 2.

Using the Laplace criterion principle, we assume that in a mining pool that includes a PPLNS payment mechanism, there are only two possibilities for partial workload proofs submitted by miners: falling into the range of values of N and not falling into the range of values of N . Under this criterion, the partial workload proofs submitted by miners in events V_2 and V_3 have the same probability of falling into both states a and b , i.e., $(M-j/M) = (j/M), N = (k-1)M + (1/2)M = (k-1/2)M, k \in [1, K]$. Let the expected revenue of the four mining pool mechanisms be $E(V_1), E(V_2), E(V_3)$, and $E(V_4)$. The following conclusions can be drawn from Table 2.

TABLE 2: Benefits of each of the four mining pool mechanisms under the Laplace criterion.

| Payment mechanisms | Events V_i | Status C_j | Probability $P(C_j)$ | Total revenue v_j^* |
|--------------------|--------------|--------------|----------------------|--|
| PPS | V_1 | — | — | $v_1^* = (1/M)e_1(1 - \delta)KR$ |
| PPLNS | V_2 | a b | $M - j/M$ j/M | $v_{2,a}^* = (k - 1/N)e_2(1 - \delta)KR + (k - 1/N)(1 - \delta)K\varphi$ $v_{2,b}^* = (k/N)e_2(1 - \delta)KR + (k/N)(1 - \delta)K\varphi$ |
| PPS+ | V_3 | a b | $M - j/M$ j/M | $v_{3,a}^* = (1/M)e_3(1 - \delta)KR + (k - 1/N)(1 - \delta)K\varphi$ $v_{3,b}^* = (1/M)e_3(1 - \delta)KR + (k/N)(1 - \delta)K\varphi$ |
| FPPS | V_4 | — | — | $v_4^* = (1/M)e_4(1 - \delta)KR + (1/M)(1 - \delta)K\varphi$ |

TABLE 3: Matrix of values for $R_E(i, j)$.

| $R_E(i, j)$ | $E(V_1)$ | $E(V_2)$ | $E(V_3)$ | $E(V_4)$ |
|-------------|---------------------------------|---|---|---|
| $E(V_1)$ | 0 | $((e_1 - e_2)R - \varphi/e_2R + \varphi)$ | $((e_1 - e_3)R - \varphi/e_3R + \varphi)$ | $((e_1 - e_4)R - \varphi/e_4R + \varphi)$ |
| $E(V_2)$ | $((e_2 - e_1)R + \varphi/e_1R)$ | 0 | $((e_2 - e_3)R/e_3R + \varphi)$ | $((e_2 - e_4)R/e_4R + \varphi)$ |
| $E(V_3)$ | $((e_3 - e_1)R + \varphi/e_1R)$ | $((e_3 - e_2)R/e_2R + \varphi)$ | 0 | $((e_3 - e_4)R/e_4R + \varphi)$ |
| $E(V_4)$ | $((e_4 - e_1)R + \varphi/e_1R)$ | $((e_4 - e_2)R/e_2R + \varphi)$ | $((e_4 - e_3)R/e_3R + \varphi)$ | 0 |

$$\begin{aligned}
E(V_1) &= \frac{1}{M}e_1(1 - \delta)KR, \\
E(V_2) &= \frac{1}{M}e_2(1 - \delta)KR + \frac{1}{M}(1 - \delta)K\varphi, \\
E(V_3) &= \frac{1}{M}e_3(1 - \delta)KR + \frac{1}{M}(1 - \delta)K\varphi, \\
E(V_4) &= \frac{1}{M}e_4(1 - \delta)KR + \frac{1}{M}(1 - \delta)K\varphi.
\end{aligned} \tag{14}$$

Let $R_E(i, j) = (E(V_i)/E(V_j)) - 1$ and obtain $R_E(i, j)$ taking matrix as shown in Table 3.

As can be seen from Table 3, for events V_1 , when $e_1 - e_2 < (\varphi/R)$, $e_1 - e_3 < (\varphi/R)$, $e_1 - e_4 < (\varphi/R)$, and $e_1 < (3\varphi + R/4R)$, there is $E(V_1) < \min_{i=\{2,3,4\}}E(V_i)$, and the mining pool with the PPS payment mechanism is not selected at this time. When $e_1 - e_2 > (\varphi/R)$, $e_1 - e_3 > (\varphi/R)$, $e_1 - e_4 > (\varphi/R)$, and $e_1 > (3\varphi + R/4R)$, there is $E(V_1) > \max_{i=\{2,3,4\}}E(V_i)$, and the mining pool with the PPS payment mechanism is the optimal choice at this point.

For event V_2 , when $e_1 - e_2 > (\varphi/R)$, $E(V_1) > E(V_2)$, and a mining pool with a PPLNS payment mechanism is not selected. When $e_1 - e_2 < (\varphi/R)$, $E(V_1) < E(V_2)$, at which point if $e_2 - e_3 < 0$ or $e_2 - e_4 < 0$, there is $E(V_2) < \max_{i=\{3,4\}}E(V_i)$, the mining pool that does not select the PPLNS payment mechanism. If $e_2 - e_3 > 0$, $e_2 - e_4 > 0$, and $e_2 > (R - \varphi/4R)$, there is $E(V_2) > \max_{i=\{3,4\}}E(V_i)$ and at this point the mining pool with the PPLNS payment mechanism is optimal choice.

For event V_3 , when $e_1 - e_3 > (\varphi/R)$ and $e_2 - e_3 > 0$, there is $E(V_3) < \max_{i=\{1,2\}}E(V_i)$, and the mining pool with PPS+ payment mechanism is not selected. When $e_1 - e_3 < (\varphi/R)$ and $e_2 - e_3 < 0$, there is $E(V_3) > \max_{i=\{1,2\}}E(V_i)$, at which point if $e_3 - e_4 < 0$ and $e_3 < (R - \varphi/4R)$, there is $E(V_3) < E(V_4)$, and the mining pool with the PPS+ payment mechanism is not selected. If $e_3 - e_4 > 0$ and $e_3 > (R - \varphi/4R)$, there is $E(V_3) > \max_{i=\{1,2,4\}}E(V_i)$, and the mining pool with PPS+ payment mechanism is the optimal choice.

For event V_4 , when $e_1 - e_4 > (\varphi/R)$, $e_2 - e_4 > 0$, and $e_3 - e_4 > 0$, there is $E(V_4) < \min_{i=\{1,2,3\}}E(V_i)$, and the mining pool with FPPS payment mechanism is not selected. When $e_1 - e_4 < (\varphi/R)$, $e_2 - e_4 < 0$, $e_3 - e_4 < 0$, and $e_4 > (R - \varphi/4R)$, there is $E(V_4) > \max_{i=\{1,2,3\}}E(V_i)$, and at this point the mining pool with FPPS payment mechanism is the optimal choice.

In particular, when $e_1 - e_2 = (\varphi/R)$, $e_1 - e_3 = (\varphi/R)$, $e_1 - e_4 = (\varphi/R)$, $e_2 - e_3 = 0$, and $e_3 - e_4 = 0$, i.e., when $e_1 = (3\varphi + R/4R)$ and $e_2 = e_3 = e_4 = (R - \varphi/4R)$, all have $E(V_1) = E(V_2) = E(V_3) = E(V_4)$, and the choice of any payment mechanism mining pool is optimal.

4. Simulation and Analysis

This chapter evaluates the mining pool selection strategies under the Laplace criterion. We refer to the strategy that always selects one of the PPS, PPLNS, PPS+, and FPPS mining pools as strategy C_1 , C_2 , C_3 , and C_4 , and the strategy that uses the scheme proposed in this paper as strategy C_5 . There are four events under strategy C_5 , namely, V_1 , V_2 , V_3 , and V_4 , representing the selection of the PPS, PPLNS, and PPS+ FPPS mining pools as the optimal choice for each case. In the discussion in Chapter 4, we analyzed the relationship between the mining pool selection strategy and the proportion of computing power allocation under the Laplace criterion. In this experiment, we will verify the effectiveness of the strategies proposed in this paper by comparing the total returns of each strategy under several different combinations of computing power allocation. To be able to verify more intuitively the validity of the strategies derived in this paper using the Laplace criterion, the following experiments were carried out.

4.1. Experimental Scenario. The experiments were implemented on a 64-bit Windows 10 system, with the Python 3 programming tool, using the NumPy scientific computing library for the simulations and the Matplotlib plotting library for the graphical presentation of the simulation results.

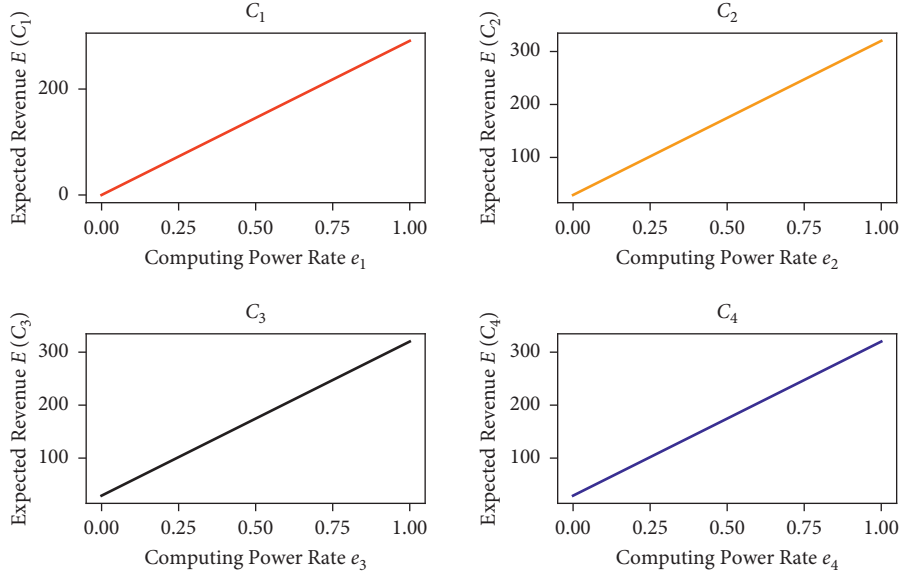


FIGURE 1: Expected revenue of strategy C_1 , C_2 , C_3 , and C_4 under different computing power allocations.

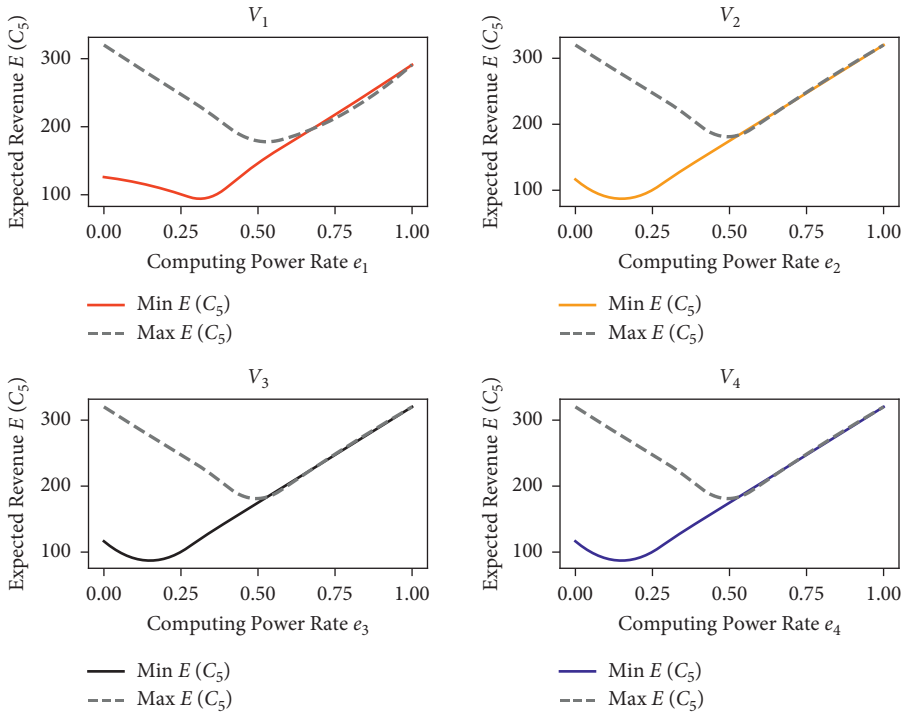


FIGURE 2: Expected revenue of strategy C_5 for each event under change in computing power.

4.2. Experiment Content. In this experiment, four mining pools are set up in the blockchain network, using the PPS, PPLNS, PPS+, and FPPS payment mechanisms, respectively, and each of them is governed by a computing power ratio of e_1 , e_2 , e_3 , and e_4 . All four pools will produce blocks in each round and receive block rewards and miner's fees corresponding to the computing power ratio, but the rewards are distributed according to their respective payment mechanisms, and each miner submits only partial proof of workload in each round. Assuming $M = 5$, $K = 15$, $\delta = 0.03$,

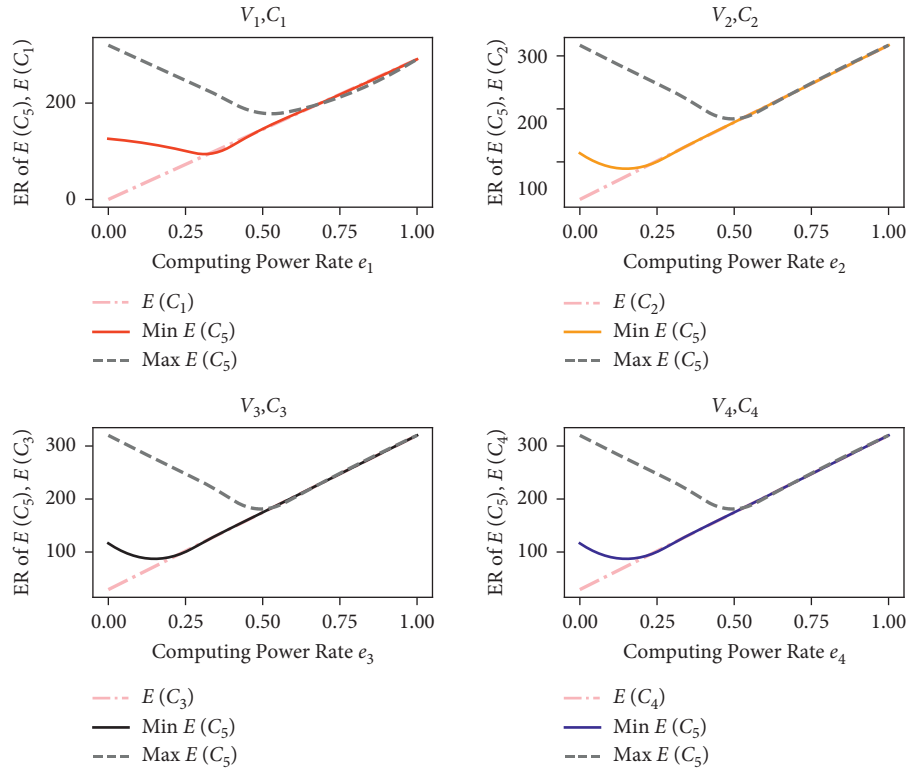
$R = 100$, and $\varphi = 10$, the data with computing power e_1 , e_2 , e_3 , and e_4 all in the range of $[0, 1]$ and $e_1 + e_2 + e_3 + e_4 = 1$ are recorded as a combination, and the experiment gives the optimal mining pool selection strategy under a variety of different combinations of computing power taking values.

5. Results and Analysis

For strategies C_1 , C_2 , C_3 , and C_4 , experiments are conducted according to the possible values of their computing power e_1 ,

TABLE 4: Selected values for strategy C_5

| Percentage of computing power | | 0 | 0.225 | 0.325 | 0.475 | 0.675 | 1 |
|-------------------------------|--------------|---------|---------|---------|---------|---------|-------|
| e1 | Min $E(C_5)$ | 126.003 | 104.275 | 94.575 | 138.225 | 196.425 | 294 |
| | Max $E(C_5)$ | 320.1 | 254.625 | 225.525 | 181.875 | 196.425 | 291 |
| e2 | Min $E(C_5)$ | 116.4 | 94.575 | 123.675 | 167.325 | 225.525 | 320.1 |
| | Max $E(C_5)$ | 320.1 | 254.625 | 225.525 | 181.875 | 225.525 | 320.1 |
| e3 | Min $E(C_5)$ | 116.4 | 94.575 | 123.675 | 167.325 | 225.525 | 320.1 |
| | Max $E(C_5)$ | 320.1 | 254.625 | 225.525 | 181.875 | 225.525 | 320.1 |
| e4 | Min $E(C_5)$ | 116.4 | 94.575 | 123.675 | 167.325 | 225.525 | 320.1 |
| | Max $E(C_5)$ | 320.1 | 254.625 | 225.525 | 181.875 | 225.525 | 320.1 |


 FIGURE 3: Expected revenue for strategy C_1 , C_2 , C_3 , and C_4 versus strategy C_5 under different event choices.

e_2 , e_3 , and e_4 , respectively. Figure 1 represents the expected revenue values of the strategy C_1 , C_2 , C_3 , and C_4 , when the computing power e_1 , e_2 , e_3 , and e_4 takes values in the range $[0, 1]$. Figure 1 represents the relationship between the expected revenue $E(C_i)$ of the strategy C_i using a separate mine pool payment mechanism and the computing power e_i assigned to this strategy. As can be seen from Figure 1, the expected payoff of the C_1 , C_2 , and C_3 , strategy is proportional to the computing power it is assigned to.

Figure 2 represents the expected revenue that can be fetched by strategy C_5 for each event computing power variation, and Table 4 shows the values taken for some of the points in strategy C_5 . The dashed line represents the maximum expected revenue of strategy C_5 for each event computing power variation, and the realization represents the minimum expected revenue of strategy C_5 for each event computing power variation. In Figure 2, we show the relationship between the expected revenue of the strategy C_5

and the corresponding computing power e_i in each of the four events, V_1 , V_2 , V_3 , and V_4 , respectively. For any one event, the minimum expected gain is the greater of the gain of that event at computing power e and the gain of the other events that have equally divided the remaining computing power $(1 - e)$ other than the computing power of that event; the maximum expected gain arises when the other events have concentrated the remaining computing power $(1 - e)$ in one of the other events. Taking event V_1 as an example, the computing power of event V_1 is e_1 , the gain is $E(V_1)$, and the gain of other events V_2 and V_3 is $E(V_2)$, $E(V_3)$, and $E(V_4)$. When $e_2 = e_3 = e_4 = (1 - e_1)/3$, there is $E(V_2) = E(V_3) = E(V_4)$ and $\text{Min } E(C_5) = \text{Min}\{E(V_1), E(V_i)\}$, $i = 2, 3, 4$; when $e_2 = 1 - e_1$, $e_3 = e_4 = 0$, there is $\text{Max } E(C_5) = \text{Max}\{E(V_1), E(V_2)\}$. When $E(V_1) = E(V_2) = E(V_3) = E(V_4)$, take the theoretical minimum, the solid line turning point in the diagram; when $E(V_1) = E(V_2)$, $\text{Max } E(C_5)$ coincides with the value of $\text{Min } E(C_5)$, both the

points in the diagram where the realized and dashed lines intersect.

As can be seen from Figure 2, the optimal choice of strategy events changes depending on the amount of computing power allocated to e_1 , e_2 , e_3 , and e_4 ; when $e_1 = 0.325$, and $e_2 = e_3 = e_4 = 0.225$, i.e., when $e_1 = (3\varphi + R/4R)$ and $e_2 = e_3 = e_4 = (R - \varphi/4R)$, strategy C_5 takes the minimum expected revenue of $\text{Min} E(C_5) = E(V_1) = E(V_2) = E(V_3) = E(V_4)$, at which point the choice of any mechanism of the mining pool is optimal.

Figure 3 shows the expected revenue of strategy C_1 , C_2 , C_3 , and C_4 compared to strategy C_5 under different event choices depending on the change in computing power, where the dotted line represents the expected revenue of strategy C_1 , C_2 , C_3 , and C_4 . In this chapter, the computing power of the four events of policy C_5 corresponds to that of strategies C_1 , C_2 , C_3 , and C_4 , respectively. In Figure 3, we group [event V_i , strategy C_i] and compare the relationship between the expected revenues of strategy C_5 and strategy C_i in each group, using the computing power e_i as the variable. Combined with Table 2, it can be seen that in event V_1 , when $e_1 < (3\varphi + R/4R)$, strategy C_5 does not select a mining pool with a PPS payment mechanism, and when $e_1 > (3\varphi + R/4R)$, strategy C_5 will select a payment mechanism based on a ratio of e_1 , e_2 , e_3 , and e_4 , at which point the mining pool that selects a PPS payment mechanism will receive the minimum expected revenue $\text{Min} E(V_1)$, at which point the maximum expected revenue $\text{Max} E(V_1)$ is provided by a mining pool with another payment mechanism. Once the e_1 ratio of computing power grows to meet $\text{Min} E(V_1) = \text{Max} E(V_1)$, the mining pool with the PPS payment mechanism becomes the optimal choice. Similarly in events V_2 , V_3 , and V_4 , when $e_2 < (R - \varphi/4R)$, strategy C_5 does not select the pool with the PPLNS payment mechanism, and when $e_2 > (R - \varphi/4R)$, the pool with the PPLNS payment mechanism will obtain the minimum expected revenue $\text{Min} E(V_2)$, and the pool with the PPLNS payment mechanism becomes the optimal choice after the proportion of computing power e_2 grows to satisfy $\text{Min} E(V_2) = \text{Max} E(V_2)$. When $e_3 < (R - \varphi/4R)$, strategy C_5 does not select the pool with the PPS+ payment mechanism, and when $e_3 > (R - \varphi/4R)$, the pool with the PPS+ payment mechanism will obtain the minimum expected revenue $\text{Min} E(V_3)$, and the pool with the PPS+ payment mechanism becomes the optimal choice after the computing power e_3 grows proportionally to satisfy $\text{Min} E(V_3) = \text{Max} E(V_3)$. When $e_4 < (R - \varphi/4R)$, strategy C_5 does not select the pool with FPPS payment mechanism; when $e_4 > (R - \varphi/4R)$, the pool with FPPS payment mechanism will get the minimum expected revenue $\text{Min} E(V_4)$, and the pool with FPPS payment mechanism becomes the optimal choice after the proportion of computing power e_4 grows to satisfy $\text{Min} E(V_4) = \text{Max} E(V_4)$.

As can be seen from Figure 3, when the proportion of computing power represented by strategy C_1 , C_2 , C_3 , and C_4 is high, the expected revenue obtained by strategy C_5 is equal to the expected revenue obtained by strategy C_1 , C_2 , C_3 , and C_4 ; when the proportion of computing power represented by strategy C_1 , C_2 , C_3 , and C_4 is low, the expected revenue

obtained by strategy C_5 is higher than the expected revenue obtained by strategy C_1 , C_2 , C_3 , and C_4 . This indicates that strategy C_5 is superior to strategy C_1 , C_2 , C_3 , and C_4 .

6. Conclusion

This paper examines the problem of pool selection faced by miners when mining in blockchain networks. Consider the impact on the revenue of miners choosing a pool with a different payment mechanism when the four common pool payment mechanisms compete for blockchain network computing power. We adopt Laplace's criterion for the optimal selection strategy for four mining pools with different computing power and design corresponding experiments to evaluate the proposed pool selection strategy, and the experimental results verify the effectiveness of this pool selection strategy. This paper has shortcomings in the following questions:

RQ1: How to implement a selection strategy for multiple payment mechanism pools when miners submit multiple partial workload certificates in a single round.

RQ2: How to implement a selection strategy for multiple payment mechanism pools in the case of changing computing power allocation.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

No potential conflicts of interest were reported by the authors.

Acknowledgments

This work was supported by the National Natural Science Foundation of China, under Grant no. 61373162, and the Sichuan Provincial Science and Technology Department Project, under Grant no. 2022YFG0161.

References

- [1] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system [EB/OL]," 2008, <https://bitcoin.org/bitcoin.pdf>.
- [2] Li Dong and W. Jinwu, "Theory, application fields and challenge of the blockchain technology[J]," *Telecommunications Science*, vol. 32, no. 12, pp. 20–25, 2016.
- [3] A. Liu, X. Du, Na Wang, and S. Z Li, "Research progress of blockchain technology and its application in information security[J]," *Journal of Software*, vol. 29, no. 7, pp. 2092–2115, 2018.
- [4] J. Leng, M. Zhou, J. L. Zhao, Y. Huang, and Y. Bian, "Blockchain Security: A Survey of Techniques and Research Directions," *IEEE Transactions on Services Computing*, vol. 11 page, 2022.
- [5] J. Leng, G. Ruan, P. Jiang et al., "Blockchain-empowered sustainable manufacturing and product lifecycle management in Industry 4.0: a survey," *Renewable and Sustainable Energy Reviews*, vol. 132, Article ID 110112, 2020.

- [6] J. Leng, S. Ye, M. Zhou et al., "Blockchain-Secured Smart Manufacturing in Industry 4.0: A Survey," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 1, pp. 237–252, 2021.
- [7] Y. Yuan and F. Wang, "Blockchain: the state of the art and future trends[J]," *Acta Automatica Sinica*, vol. 42, no. 04, pp. 481–494, 2016.
- [8] L. Tan, K. Yu, N. Shi, C. Yang, W. Wei, and H. Lu, "Towards secure and privacy-preserving data sharing for COVID-19 medical records: a blockchain-empowered approach," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 1, pp. 271–281, 2022.
- [9] L. Tan, H. Xiao, K. Yu, M. Aloqaily, and Y. Jararweh, "A blockchain-empowered crowdsourcing system for 5G-enabled smart cities," *Computer Standards & Interfaces*, vol. 76, p. 103517, 2021.
- [10] L. Tan, K. Yu, C. Yang, and A. K. Bashir, "A blockchain-based shamir's threshold cryptography for data protection in industrial internet of things of smart city," in *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking (MobiCom 2021), Virtual Conference*, New Orleans, Louisiana, October 2021.
- [11] J. Leng, P. Jiang, K. Xu et al., "Makerchain: a blockchain with chemical signature for self-organizing process in social manufacturing," *Journal of Cleaner Production*, vol. 234, pp. 767–778, 2019.
- [12] J. Leng, D. Yan, Q. Liu et al., "ManuChain: combining permissioned blockchain with a holistic optimization model as Bi-level intelligence for smart manufacturing," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 1, pp. 182–192, 2020.
- [13] L. Tan, N. Shi, C. Yang, and K. Yu, "A blockchain-based access control framework for cyber-physical-social system big data," *IEEE Access*, vol. 8, pp. 77215–77226, 2020.
- [14] L. Tan, H. Xiao, X. Shang, Y. Wang, F. Ding, and W. Li, "A blockchain-based trusted service mechanism for crowdsourcing system," *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, in *Proceedings of the IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, pp. 1–6, Antwerp, Belgium, May 2020.
- [15] C. Tang, C. Li, X. Yu, Z. Zheng, and Z. Chen, "Cooperative mining in blockchain networks with zero-determinant strategies," *IEEE Transactions on Cybernetics*, vol. 50, no. 10, pp. 4544–4549, 2020.
- [16] Di Jian and W. Lin, "Research and analysis of mining pool selection strategy in blockchain[J]," *Computer Application Research*, vol. 37, no. 06, pp. 1804–1807, 2020.
- [17] R. Qin, Y. Yuan, and F.-Y. Wang, "Research on the selection strategies of blockchain mining pools," *IEEE Transactions on Computational Social Systems*, vol. 5, no. 3, pp. 748–757, Sept. 2018.
- [18] M. Rosenfeld, "Analysis of Bitcoin pooled mining reward systems," 2011, <https://arxiv.org/abs/1112.4980>.
- [19] M. Skorjanc, "How mining pools distribute rewards? PPS vs FPPS vs PPLNS [EB/OL]," 2019, <https://www.nicehash.com/blog/post/how-mining-pools-distribute-rewards-pps-vs-fpps-vs-pplns>.
- [20] L. Tech, "Different bitcoin mining pool payment methods (PPS vs FPPS vs PPLNS vs PPS+) [EB/OL]," 2018, <https://medium.com/luxor/mining-pool-payment-methods-pps-vs-pplns-ac699f44149f>.
- [21] T. MineBest, "Different mining pool payouts explained: PPS vs. FPPS vs. PPLNS vs. PPS+ [EB/OL]," 2021, <https://minebest.com/blog/pps-vs-fpps-vs-pplns-vs-pps-mining-pool-payouts-explained>.
- [22] Dr Haribo, "Comparison of mining pools [EB/OL]," 2022, https://en.bitcoin.it/wiki/Comparison_of_mining_pools.
- [23] O. Schrijvers, J. Bonneau, D. Boneh, and T. Roughgarden, "Incentive compatibility of bitcoin mining pool reward functions," *Financial Cryptography and Data Security*, in *Financial Cryptography and Data Security. FC 2016. Lecture Notes in Computer Science*, J. Grossklags and B. Preneel, Eds., vol. 9603, pp. 477–498, Springer, Berlin, Heidelberg, 2017.
- [24] Y. Zolotavkin, J. García, and C. Rudolph, "Incentive compatibility of Pay per last N Shares in bitcoin mining pools," *Lecture Notes in Computer Science*, in *Decision and Game Theory for Security. GameSec 2017. Lecture Notes in Computer Science*, S. Rass, B. An, C. Kiekintveld, F. Fang, and S. Schauer, Eds., vol. 10575, pp. 21–39, Springer, Cham, 2017.
- [25] Y. Liu, X. Chen, L. Zhang, C. Tang, and H. Kang, "An intelligent strategy to gain profit for bitcoin mining pools," *2017 10th International Symposium on Computational Intelligence and Design (ISCID)*, in *Proceedings of the 2017 10th International Symposium on Computational Intelligence and Design (ISCID)*, pp. 427–430, Hangzhou, China, December 2017.
- [26] R. Zhang and B. Preneel, "Publish or perish: a backward-compatible defense against selfish mining in bitcoin," *Topics in Cryptology - CT-RSA 2017*, in *Topics in Cryptology - CT-RSA 2017. Lecture Notes in Computer Science*, H. Handschuh, Ed., vol. 10159, pp. 277–292, Springer, Cham, 2017.
- [27] I. Eyal, "The miner's dilemma," *2015 IEEE Symposium on Security and Privacy*, in *Proceedings of the 2015 IEEE Symposium on Security and Privacy*, pp. 89–103, San Jose, CA, USA, May 2015.
- [28] E. Altman, D. Menasché, A. Reiffers-Masson et al., "Blockchain competition between miners: a game theoretic perspective," *Frontiers in Blockchain*, vol. 2, p. 26, 2020.
- [29] S. Singh, M. Salim, M. Cho, J. Cha, Y. Pan, and J. Park, "Smart contract-based pool hopping attack prevention for blockchain networks," *Symmetry*, vol. 11, no. 7, p. 941, 2019.
- [30] T. Yang and Z. Xue, "The game problem and optimization among mining pools in blockchain systems[J]," *Communications Technology*, vol. 52, no. 05, pp. 1189–1195, 2019.
- [31] L. Fan, H. Zheng, J. Huang, Z. Li, and Y. Jiang, "A cooperative evolutionary approach for blockchain mining pools based on adaptive zero determinant strategy," *Computer Applications*, vol. 39, no. 03, pp. 918–923, 2019.
- [32] A. Kaci and A. Rachedi, "PoolCoin: toward a distributed trust model for miners' reputation management in blockchain," *2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC)*, in *Proceedings of the 2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC)*, pp. 1–6, Las Vegas, NV, USA, January 2020.
- [33] Y. Velner, J. Teutsch, and L. Luu, "Smart contracts make bitcoin mining pools vulnerable," *Financial Cryptography and Data Security*, in *Financial Cryptography and Data Security. FC 2017. Lecture Notes in Computer Science*, vol. 10323, pp. 298–316, Springer, Cham, 2017.
- [34] Na Ruan, H. Liu, and S. Xueming, "The 'catfish effect' among mining attackers in blockchain with proof-of-work consensus mechanism," *Journal of Computer Science*, vol. 44, no. 01, pp. 177–192, 2021.
- [35] Y. Wang, C. Tang, F. Lin, Z. Zheng, and Z. Chen, "Pool strategies selection in PoW-based blockchain networks:

- game-theoretic analysis,” *IEEE Access*, vol. 7, pp. 8427–8436, 2019.
- [36] X. Liu, W. Wang, D. Niyato, N. Zhao, and P. Wang, “Evolutionary game for mining pool selection in blockchain networks,” *IEEE Wireless Communications Letters*, vol. 7, no. 5, pp. 760–763, 2018.
- [37] C. Xu, K. Zhu, R. Wang, and Y. Xu, “Dynamic selection of mining pool with different reward sharing strategy in blockchain networks,” *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, in *Proceedings of the ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, pp. 1–6, Dublin, Ireland, June 2020.
- [38] Team F2pool, “Starting amounts and fees for each currency [EB/OL],” 2018, <https://blog.f2pool.com/zh/faq/threshold>.
- [39] Team Viabtc, “Tariff rates[EB/OL],” 2019, <https://www.viabtc.com/pricing>.
- [40] Team Antpool, “Miner configurations and rates [EB/OL],” 2022, <https://antpoolhelp.zendesk.com/hc/zh-cn/articles/900001014643>.
- [41] Team BTC.com, “BTC.com pool’s rates, settlement methods and starting amounts [EB/OL],” 2022, <https://help.pool.btc.com/hc/zh-cn/articles/900001116943-BTC-com>.

Research Article

Design of a Blockchain-Based Traceability System with a Privacy-Preserving Scheme of Zero-Knowledge Proof

Yudai Xue  and Jinsong Wang 

School of Computer Science and Engineering, Tianjin University of Technology, Tianjin, China

Correspondence should be addressed to Jinsong Wang; jswang@tjut.edu.cn

Received 6 March 2022; Accepted 3 June 2022; Published 29 June 2022

Academic Editor: Jiewu Leng

Copyright © 2022 Yudai Xue and Jinsong Wang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the fast development of the industrial Internet, its interconnectivity poses new challenges for the cooperation of industrial entities. Cooperation among these entities is built on trust, and trust is based on high-quality industrial products at reasonable prices. A traceability system can play an essential role in objectively reflecting the production process and promoting this trust. However, traditional traceability systems often have data privacy issues. Because traceability data are collected or generated during the production process (namely, production-related data), they could be considered privacy data. Several researchers have introduced privacy protection schemes into the traceability system, such as authentication or encryption. Nevertheless, when a privacy protection scheme is established, the original data are disclosed to the legal user of the system, but the data may still be leaked intentionally or unintentionally. Except for data privacy issues, a traditional traceability system can be vulnerable to network attacks, data unavailability, and reliability issues. The authors conducted a study to overcome these shortcomings, and this paper reports the results. We built a traceability prototype system using a blockchain protocol and a zero-knowledge proof method. First, we built a blockchain to record key production process data, aiming to maintain data reliability and availability. Second, through an analysis of traceability purpose using production knowledge, the traceability purpose could be divided into multiple provable statements. By introducing privacy protection through a zero-knowledge proof, the traceability process was converted to proving relative statements. Finally, the statements were validated by a smart contract that provided openness and reliability during the traceability process. Analysis has shown that our approach could meet the requirements for high security and privacy. In addition, the paper also discusses the calculation cost of the traceability process to show our work's viability. The traceability system described in this paper creates new possibilities for constructing a healthy and reliable trust relationship between production entities to provide further support in the development of the industrial Internet.

1. Introduction

With the application of information technology (IT) to traditional industrial production, production power has significantly increased, and senior automation and information technology have optimized the production process. However, as the scale of production expands, the industrial production model may be overwhelmed by high production levels, the bullwhip effect, or biased pricing [1]. These issues impair the trust and cooperation between industry entities [2]. To break the production limit, Industry 4.0 [3] and the industrial Internet [4] were introduced by Germany in 2010 and General Electric in 2012. The primary purpose of this structure is to

connect people, data, and machines with an open and globalized network and to achieve a high degree of integration of industrial systems with computing, analysis, and IT systems [5]. In particular, the industrial Internet may create a data corridor between each element in production scenarios. By integrating the traditional manufacturing technology with big data analysis, artificial intelligence, and advanced semiconductor technology, industrial Internet could reform the entire production and cooperation model [6].

In traditional industrial production, cooperation among entities depends on the supply chain [7], and the traceability data are one of the key parts of the supply chain system to ensure the high efficiency and stable operation. Traceability

data are also an important constituent for improving the quality of industrial products [8] and optimizing the supply chain and the production process [9]. As an important part of the supply chain, a traceability system can significantly affect product quality control, order management, and production service. However, with the continuous expansion of the supply chain and development of industrial Internet, the traceability system may face challenges due to the requirement of interconnection and open cooperation between industrial entities [10]. Current traceability system-related data are not sensitive. However, some field data related to the production process can be unsafe for a company to publish due to industrial confidentiality. Against an industrial background, sensitive production data can objectively reflect the quality and even advanced technology of industrial products, so traceability requirements for these types of sensitive data do exist. Furthermore, tracing these production data can promote and encourage interindustry cooperation and interactions. This new type of cooperation and production may transform the dominant industrial model.

For regular scenarios, traceability can be treated as a process that satisfies a particular purpose related to demand. In this situation, the initiator of traceability should have a clear purpose, such as locating quality problems in the production process [11] or verifying whether a particular process in the production process satisfies required standards or specifications. No matter the purpose of the traceability, it is ultimately up to the traceability initializer to judge the relevant data obtained from a traceability system. However, acquirers of traceability data might lack knowledge about production in an actual traceability process. After the related data are received, a third party is needed to interpret the result using traceability data, which may eventually cause production data to leak. Therefore, the traceability process for sensitive production-related data should avoid data transmission.

To achieve the above, this study adopts a privacy protection mechanism based on a zero-knowledge proof and realizes the traceability of the target with no need for the traceability data owner to provide any original data. In addition, the traceability system still needs to solve the problem of original data availability and reliability, and the proof should be open and without the possibility of repudiation. To satisfy the above requirements, this paper introduces blockchain technology into the design of traceability system, which could provide features that are tamper-resistant, unable to be repudiated, and open to supervision [12]. To design a complete system, this study made the following assumptions:

- (1) Raw industrial production data are stored in the respective production domains, while the related data digest is stored in the blockchain.
- (2) The traceability process was initialized by the traceability data acquirer with a clear purpose or an expected traceability result.
- (3) There is a correlation between the traceability purpose and the industrial traceable production data.

Under the above premise, this paper introduces a zero-knowledge proof for raw data production. Through a

purpose analysis of traceability data acquirer, the production process was converted into multiple statements and adopted a zero-knowledge proof engine generating a validator to prove those statements. Finally, through publishing a smart contract, the traceability process is completed in an open and fair manner. The innovations of this paper are as follows:

- (1) An abstraction of the industrial production process into several traceability features according to their traceability is developed.
- (2) A privacy-preserving traceability system architecture for production data is constructed and the algorithm flow involved in the architecture is explained.
- (3) Availability and security issues are discussed through a comparative analysis.

2. Background Knowledge

2.1. Blockchain and Distributed Ledgers. A blockchain is a type of distributed ledger system designed on a cryptography algorithm, peer-to-peer (P2P) network, and distributed consensus algorithm [13]. Blockchain has attracted much attention since its creation. Many researchers have shown increasing interest in the application of blockchain. Generally speaking, blockchain can be divided into two categories: unauthorized blockchain and authorized blockchain [14]. The former category is usually used to build payment systems instead of centralized banks, such as Bitcoin or Ethereum [15, 16], and the latter is designed for specific application scenarios such as medical, agrifood, or other fields. No matter what type of blockchain is being used, it can help to build trust between participants in an exchange. Authorized blockchain supported with smart contracts could be applied to financial, medical, and logistics scenarios [17]. Moreover, its features of tamper-resistance and non-repudiation may provide highly reliable data that could be the basis for industrial entities creating cooperative relationships. Especially in the research area of combining blockchain with industrial Internet or industry 4.0, the security and trust features of blockchain [18] may help industrial entities create healthy cooperative relationships and promote the production level.

The main idea of the traceability system designed in this paper is to trace the privacy data generated in the production process. In industrial production scenarios, such as the industrial Internet of things or the industrial Internet, the data generated from billions of sensors, controllers, and data collectors makes it possible for entities to make production more intelligent, optimize production plans, and realize cooperative production [19]. To create a reliable traceability system, the first step is to guarantee the reliability and availability of production data [20]. In traditional centralized traceability approaches, there are many potential information security issues, including denial-of-service attacks, spoofing attacks, and data leaking and tampering, while a blockchain-based approach may be immune to the security issues above and make the data both tamper-resistant and highly available [21]. In this situation, the circulation of data is treated as a transaction, and data digests can be recorded

in a transaction for confirmation by all participants. Supported by blockchain, a traceability system able to fully record industrial production could be created.

Nevertheless, the openness of blockchain may also create privacy issues [22]. Production-related data may be tightly bonded with industrial secrets and sensitive data, making it impossible for entities to share their production data for traceability. Therefore, a privacy-preserving scheme should be deployed.

2.2. Zero-Knowledge Proofs. Zero-knowledge proofs can enable one subject to verify the correctness of a statement put forward by another subject without involving any raw data or relying on a third party [23]. Therefore, zero-knowledge proofs can be used as effective privacy protection mechanisms. There are two leading roles involved in the zero-knowledge proof process. The first is the prover, who declares a statement and generates a proof with raw data. The second role is the verifier, or the proof receiver, who has the ability to verify the proof. Zero-knowledge proofs are widely used in privacy-preserving schemes due to their completeness, soundness, and zero-knowledge [24]. Completeness means the statement can be verified by the prover and convince the verifier of its veracity. Its soundness provides an environment in which the prover cannot cheat the verifier with a false proof. Zero-knowledge ensures that the raw data is never revealed to the public. The prover can always maintain their ownership of the raw data during the proving and verifying processes to protect their privacy.

In the real-world usage of zero-knowledge proof schemes, a toolkit based on zero-knowledge succinct non-interactive arguments of knowledge (zkSNARK) is introduced to build corresponding systems [25]. zkSNARK allows the prover to prove its statement with low process complexity using a simple message. Therefore, the toolkit based on zkSNARK is widely used in the design of blockchain-based applications [26]. According to the application scenario, a zkSNARK-based toolkit offers a flexible and effective way to create a smart contract for automatic verification and generate the proof with raw data.

In this situation, the application scenarios of zero-knowledge proof are significantly expanded and provide the possibility for the implementation of traceability in this paper.

2.3. Related Work. Research on the combination of blockchain and traceability systems has shown that the data recorded in a blockchain can provide reliable support for data traceability so long as production data can be stored with high reliability. Previous work has successfully connected the traceability process with the data produced during production [27]. The traceability of production data is beneficial for tracking production drawbacks and raising production quality [28], and, in the research area of traceability, Chen et al. [29] demonstrated the relationship between quality control and traceability and then designed a quality control model based on traceability. On this basis, Tsai and Wang et al. [30] then designed a cooperative

production method based on the production data to improve and optimize the production process. By importing blockchain into traceability system design, more researchers have concentrated on the design of blockchain-based decentralized traceability systems. Helo et al. [31] designed a high-performance traceability model for the supply chains based on blockchain, Radio Frequency Identification (RFID), and Internet of Things (IoT) technology. Zhu et al. [32] optimized the supply chain by using blockchain-based traceability system to trace production processes and coordination. Xiao et al. [33] designed a traceability model for the agrifood industry, providing a safe and traceable environment for food production quality control and anti-counterfeiting. Tarun [34], based on blockchain, constructed a traceability system for textile manufacturing and improved production efficiency. Uddin [35] built a blockchain-based traceability framework for the pharmacy industry that provided reliability verification for the circulation of medicine. Patelli et al. [36] built a traceability system for the supply chain management of food industry based on blockchain, which is immune to several network attacks compared to the traditional traceability mechanism. The above studies have proven that blockchain technology can be effectively integrated with traceability mechanisms to improve production efficiency and product quality to ensure data reliability. Except for studies on the reliability of traceability data, privacy-preserving schemes for traceability data were also discussed by researchers. Yang [37] used RFID encryption to provide production data privacy during data collection. Wang [38] treated the traceability process as transactions in the blockchain and divided the traceability process into three actions: demanding, pricing, and trading.

Privacy-preserving mechanisms have also been introduced to avoid privacy leakage issues formed by the openness of blockchain. The above research shows that privacy preservation methods mainly depend on access control, identity authorization, or data encryption. Although the traceability process is protected, raw traceability data can still be leaked by the traceability data receiver.

In this paper, a zero-knowledge proof is used to protect the raw traceability data. Zero-knowledge proofs can provide proof for satisfying specified conditions in a specific scene without disclosing any private information. Currently, zero-knowledge proofs are widely used in digital currency. Zcash realized a privacy-protected digital currency system by applying a zero-knowledge proof. In addition, Eberhardt [39] combined a zero-knowledge proof with an Ethereum smart contract by constructing ZoKrates, realizing a zero-knowledge proof mode of offline computing and online verification, thus expanding the possible applications of zero-knowledge proofs. Based on this, Westerkamp [40] designed a side chain proof mechanism using ZoKrates. In addition to the field of digital currency, Ibrahim [41] applied zero-knowledge proofs and homomorphic encryption to an anonymous voting system. Rasheed et al. [42] realized an anonymity authentication method based on the premise of protecting user privacy by applying a zero-knowledge proof in the area of Internet of vehicle. Jeong et al. [43] applied smart contracts and zero-knowledge proofs in online real

estate transactions to provide a transaction process with a privacy protection mechanism. Qi et al. [44] applied a zero-knowledge proof to the auto insurance industry, achieving an effective insurance evaluation method based on usage habits while preserving privacy. Umar et al. [45] combined zero-knowledge proofs with wireless body sensors to effectively avoid privacy leakage caused by malicious attackers monitoring communication channels. Huang et al. [46] proposed an auditable information-sharing mechanism for the industrial Internet based on a zero-knowledge proof mechanism, which effectively avoided the leakage of sensitive information into the industrial environment. The above research results show that zero-knowledge proofs can provide a proof process with a privacy protection mechanism for different fields according to their needs, and so it is feasible and beneficial to attempt to apply it to the privacy protection process of production data traceability.

3. System Architecture

3.1. Architecture Review. Based on industrial production data, as industrial production entities record production data digests in the blockchain, these production data can objectively describe the production process and reflect the technologies of production and the flow of the production process. Traceability processes are often used to show the high quality or advanced technology of production [47]. In this situation, the traceability process needs to be open and fair. However, the traceability data acquirers may lack production-related knowledge and cannot judge whether acquired data could reach their purpose. Therefore, the current solution relies on a trusted third party, which means that the traceability data acquirers need to inform their purposes to a trusted third party, entrust it as an agent to receive the traceability data, and make a final judgement [48]. However, this solution still exposes the production data to a third-party verifier. At the same time, the verification process would not be open and transparent. To establish an open, fair, and privacy-preserving traceability mechanism, this paper designed a new traceability system based on a zero-knowledge proof combined with a smart contract and blockchain, which is called a zero-knowledge-based traceability system (ZKTS), to create a privacy data traceability method independent from any third party in order to protect the privacy of producers. The architecture diagram of the traceability system is shown in Figure 1.

The traceability system in this paper contains three layers. The first is the physical data layer, which contains the data produced by each independent entity according to the actual production, the physical data generated by the transportation and sales process, and the raw data saved in the database and constructed by an independent production entity. At the same time, the digest of raw data was published in an authorized blockchain system called a data chain. Data chain is open for all certificated users to access the data digest of raw data. The second is the data privacy layer, composed of a zero-knowledge engine and its related external interfaces. The data privacy layer is mainly responsible for receiving the traceability features and the related data digest

from the upper layer, building the smart contract, and interacting with the data provider to generate the proof for traceability using a zero-knowledge proof engine. The data privacy layer is responsible for the core function of the traceability system, privacy preservation. The third layer is the application layer; in this paper, we define the purpose of the traceability data acquirer as the traceability purpose, and the application layer maintains the authentication process and the traceability purpose analysis process or the traceability feature-generating algorithm. This algorithm is used to analyze the primary purpose of traceability and generate related features and could satisfy the traceability purpose. As shown in Figure 1, the entire system contains two relatively independent blockchains. One is used to ensure the integrity and reliability of relevant production data, and the other is used for traceability verification in public scenarios. The two blockchains cannot interact directly but can be accessed through authentication with supervision. The specific functions of each layer in the traceability model are as follows.

3.1.1. Physical Data Layer. The physical data layer includes sensors, controllers, data collection devices, and other production-related devices. In the physical data layer, production data can be mapped to devices using a unique label through RFID or other technologies. The physical data layer collects the data generated by those production devices, and these collected data can be used to generate traceability features. In this study, we assumed that no fake data had been created in the physical data layer and that the data digest would be published in a particular blockchain system (data chain).

3.1.2. Data Privacy Layer. The data privacy layer is the key component ensuring the protection of privacy in the traceability system. The data privacy layer has a data extraction and privacy processing engine. First, the data privacy layer collects the data digest of production from the data chain. According to the analysis of the purpose of the traceability data acquirer, the privacy processing engine then generates a smart contract with need-to-proof issues. A witness is then generated to interact with the owner of the raw production data to generate the proof.

The processing of the data privacy layer involves the privacy information of production, so the process is offline. After the proof is generated, the smart contract, related proof, and traceability features are disclosed to verify whether traceability is achieved.

3.1.3. Traceability Application Layer. The traceability application layer directly interacts with the traceability data acquirer and production data owner. The primary function of the application layer is to provide a traceability interface for both sides with an authentication mechanism and also to provide a platform for mutual traceability negotiation. The traceability application layer was established by industrial entities and traceability-related individuals or entities. The traceability application was developed with a smart contract

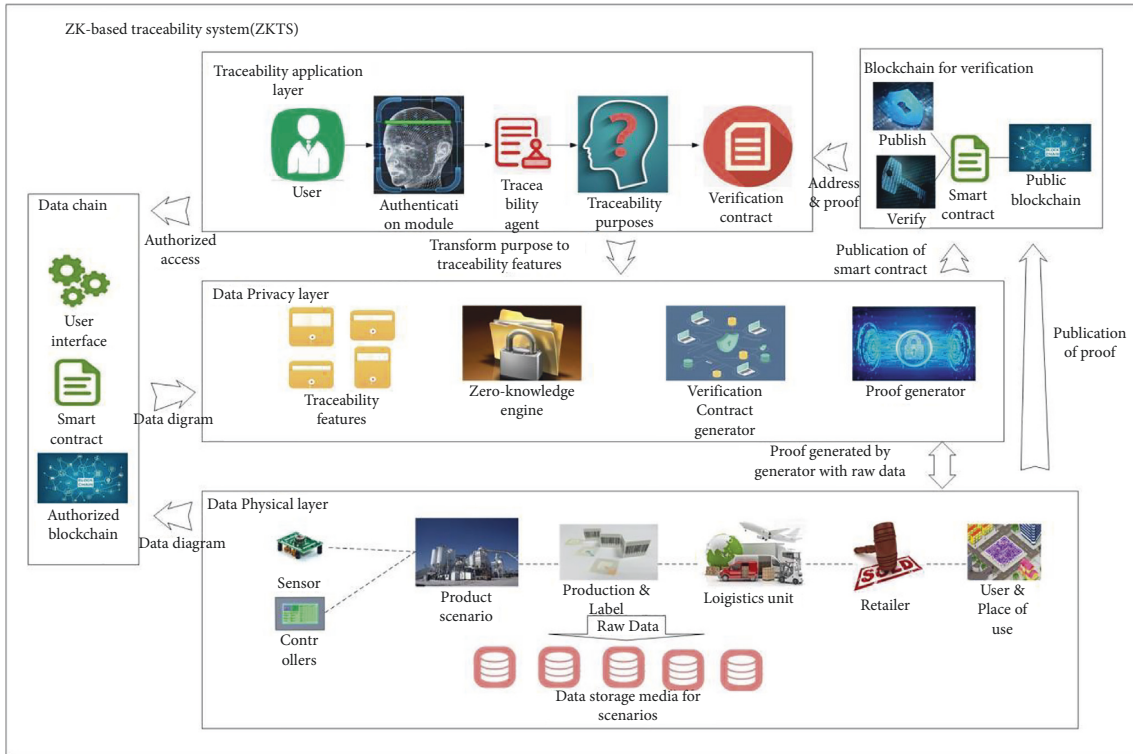


FIGURE 1: Overview of the ZKTS system.

on a public or light-authenticated blockchain system. The traceability application layer can publish the smart contract generated by the data privacy layer and provide a verification interface for the proof. A blockchain-based application layer is an essential part of the traceability system, allowing the system to be both fair and transparent.

3.2. Participant Design. In the traceability system we designed, given the ownership of the traceability data, there were three roles: the initializer of the traceability process, called the traceability data acquirer (TDA); the traceability data owner, called the traceability data provider (TDP); and a third party, called the traceability agent (TA).

The TDP was the original holder and provider of the production data to be traced. In production-related traceability, the data has production technology privacy that needs to be protected during the production and circulation of the relevant products. The TDA was the initializer of the tracing process and could be a partner or a consumer with either a cooperative or a transaction-based relationship with the TDP. The TDA initiated the tracing process with a clear purpose such as judgement of production quality, making it a sign of traceability completion. The TA introduced in this paper is a new role. The TA held the production knowledge of the TDP, which means that the TA could analyze the primary purposes of the TDA. According to the features generated by the TA, smart contracts with related need-to-proof issues were created and transferred to TDP. According to those need-to-proof issues, the TDP generates the proof with raw production data and publishes it to the public.

Since the behavior of the TA did not involve the privacy of both sides of the traceability process, the whole traceability process can be conducted transparently to ensure the openness and effectiveness.

3.3. Traceability Process Design. The traceability process designed in this study contains four phases: the authentication phase, preprocessing phase, construction phase, and verification phase. In different stages, different roles may execute corresponding workflows, and a complete process of our approach together with the relationship between roles and workflows is shown in Figure 2.

A typical traceability process is shown in Figure 2. In the authentication phase, an authentication scheme is implemented based on the data recorded in the data chain. A traceability request is then sent from the TDA to the TA. In the preprocessing phase, the TA extracts the tracing proposal from the request, generates the traceability features, and transfers them to the TDP. In the construction phase, the TDP receives the traceability features and generates the proof using the raw production data. Meanwhile, a smart contract is published by the TA for verification. Finally, in the verification phase, the TA receives the generated proof and passes it to the TDA. After the complete process, the TDA determines whether traceability has been achieved. A more detailed explanation of the algorithm and its process is fully explained in what follows.

3.3.1. Authentication Phase. In this phase, due to the openness of the traceability system, the TDA and TDP both

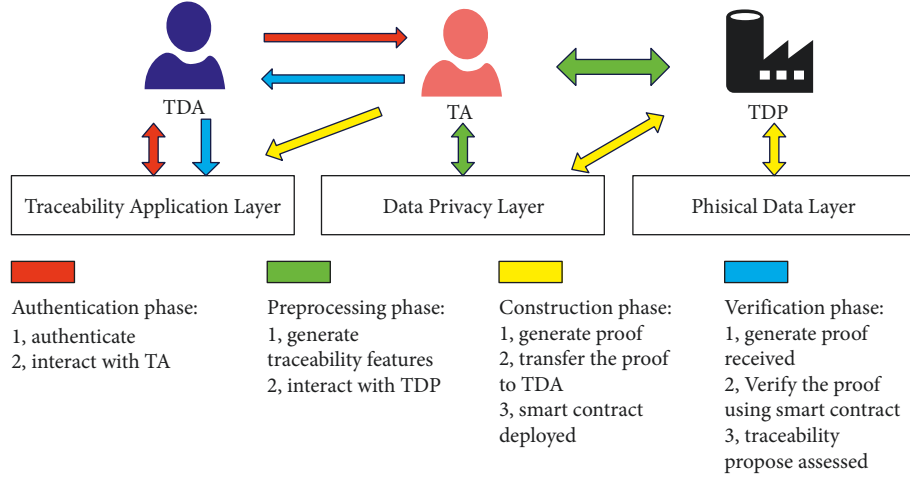


FIGURE 2: Overview of the entire traceability process.

have the privilege of accessing the data digest stored in the data chain, which means the TDA and TDP each have a public and private key pair: PK_a/SK_a and PK_p/SK_p , respectively. Furthermore, in the process of production trading, a relationship between the TDA and TDP is established and the related transaction data are also recorded in the data chain; these data could be used to generate ProductData (PD) using the following formula:

$$\text{ProductData} = \text{PID}|\text{TxTS}|\text{TxHash}|\text{random}(r). \quad (1)$$

The ProductData contains relevant information about production trading between the TDA and TDP, including product identification (PID), transaction timestamp (TxTS), transaction digest (TxHash), and a random number (r). The ProductData are generated in the process of product circulation. According to the relevant data, SK_p and PK_a are used to encrypt and sign PD, while ProductSign (PS) generated using formula (2) is used for authority authentication of TDA:

$$\text{ProductSign} = \text{sign}(\text{sign}(\text{ProductData}, SK_p), PK_a)|\text{PID}. \quad (2)$$

The PS is then passed to the TDA, who can verify the product according to SK_a and PK_p . After PD is obtained, the TDA calculates the PD digest, simultaneously generates an authentication request, and then submits the request to the TDP to prove the traceability permission of the corresponding product. This completes the authentication process between the TDA and TDP. The authentication algorithm is shown in Algorithm 1.

3.3.2. Preprocessing Phase. After the authentication phase, the system workflow enters the preprocessing phase. In this phase, the traceability purpose of the TDA is analyzed by the traceability feature generating algorithm. The core of the algorithm is based on public industrial production standards or key technical knowledge. According to these standards, knowledge, and related production data, traceability features can be generated.

As the production process progressed, data corresponding to the production process is generated in industrial production. Here, we define all the production processes as P , and we obtain the following equation:

$$P = P_1 \cup P_2 \cup P_3 \cup \dots \cup P_n, \quad (3)$$

where P_i represents one of the production steps in industrial production process, and, for any I , we use the following formula:

$$\{D_1, D_2, \dots, D_{k-1}, D_k\} \longrightarrow P_i, \quad (4)$$

D_i Refers to the relevant data generated in a single production process; that is, there is a mapping relationship between the production process and the data, and each data point generated in the production process can be defined by the following formula:

$$D_j \in \{\text{bool}, \text{num}, \text{hash}\}. \quad (5)$$

In other words, the production process data can be defined as a Boolean (bool), numeric (num), or hash type in a production record. Boolean data show the state of industrial devices, such as controllers or switches. Numeric data are used to record the data generated by the production devices like sensors. Hash data are used in industrial production to record relevant signatures. In this situation, a production process can be defined by the following equation:

$$P_i = D_1^i \cup D_2^i \cup \dots \cup D_k^i, \quad (6)$$

where D_k^i is defined as the k th data generated in the i th production process. Therefore, for a single production product, its entire production process can be defined by the following equation:

$$\begin{aligned} \text{Product} = & \{D_1^1 \cup D_2^1 \cup \dots \cup D_m^1\} \cup \{D_1^2 \cup D_2^2 \cup \dots \cup D_n^2\} \\ & \cup \dots \cup \{D_1^k \cup D_2^k \cup \dots \cup D_p^k\}. \end{aligned} \quad (7)$$

On this basis, each industrial production process can be mapped to a set of data that holds all the data generated by

```

Input: ID, PKt, SKte, Tx
Output: AuthResult
(1) ProductData = PID|TxTS|TxHash|random(r)
(2) ProductSign = sign(sign(ProductData,SKp),PKa)|TxHash
(3) TDP.send(ProductSign)
(4) recvData = TDA.recv()
(5) ProductData = decrypt(recvData,SKa).decrypt(recvData,PKp)
(6) if verify(PID):
(7)   authToken = hash(ProductData)
(8) else:
(9)   deny()
(10) TDA.send(authToken|PID)
(11) authToken = TDP.recv()
(12) if authToken == hash(ProductData)
(13)   IdentityConfirmed()
(14) Else
(15)   deny

```

ALGORITHM 1: Authentication algorithm.

the production process. Considering the production data and the correlation of the traceability process, a function, `dataParser`, was introduced to filter the key data out of the set. The key data are then generated using the following equation:

$$\text{KeyData} = \text{dataParser}(\text{Product}). \quad (8)$$

KeyData obtained through this process can be further processed and combined with relevant knowledge of the production field to generate traceability features. In this situation, the traceability purpose is defined as TP, and the traceability feature can be generated using the following equation:

$$\text{TraceabilityFeature} = \text{parse}(\text{KeyData}, \text{knowledge}). \quad (9)$$

In the above equation, knowledge means the production knowledge that can be obtained by the relevant production experts or by a public production standard. TraceabilityFeature is generated as a producer of zero-knowledge proof. On this basis, the TA will use TraceabilityFeature to generate a smart contract and also transfer the TraceabilityFeature to TDP for generating the proof with raw data.

In addition to traceability features generated in the preprocessing phase, high-availability production data are provided by the TDP. Sensors generate raw production data during production, transportation, trading, and other activities, among which different production entities are distributed. These data can be marked as `rawData`, and the `rawData` digest is marked as `HashData`. `HashData` and the identity of the data source form a transaction record, `Tx`. Finally, `Tx` is published to the data chain, which creates a relationship between production data and transactions. The on-chain data digest ensures the tamper-resistant and antirepudiation characteristics of the production data. The specific algorithm of data preprocessing is shown in Algorithm 2.

The data are first collected by a data collection device in the data physical layer and are used to extract the traceability

```

Input: Data
Output: Tx
(1) Data = [sensor, collector,controller, etc].collect()
(2) ID = [sensor, collector,controller, etc].PID
(3) Hashdata = hash(Data)
(4) Tx = TXgenerator(Hashdata|ID)
(5) Tx.submit()

```

ALGORITHM 2: Data preprocessing algorithm.

features with production knowledge. The blockchain-based data record model has been previously discussed in the literature [34, 37]. In our study, we only refer to those conclusions. The high-availability record of the production data provides a basis for the privacy protection approach to be adopted in the later construction phase.

3.3.3. Construction Phase. After the two phases above are finished, the traceability features are provided with the available production data. The construction phase may import the zero-knowledge engine to generate related smart contracts and the proof. In the construction phase, the TA uses the traceability features generated by the preprocessing phase to create need-to-proof issues, which is called the witness, and then transfers it to the TDP. The TDP receives and generates the proof using a witness and the related raw data. After the proof is generated, the TDP submits the proof to the TA or the public, and then the TDA receives the data and verifies the proof.

At first, the generation of witness should be discussed. We define the witness as `verifyKey`, and three main issues should be proven. First, the available data must prove that the data that the TDP used to generate the proof are the same as those recorded in the data chain. Second, those data must be from the production being traced. Third, the traceability feature proof must show that the proof of data is satisfied

Input: PID, TraceabilityFeature(TF), offlineVolume
Output: witness

- (1) Tx = Datachain.search(PID)
- (2) Hash = Tx.DataHash
- (3) Data = offlineVolume. Find (“PID”)
- (4) For each in TF
- (5) verifyKey.append(Hash(Data) == Tx.hashData)
- (6) verifyKey.append(PID == Tx.PID)
- (7) verifyKey.append(Data.satisfy(TF))
- (8) return verifyKey

ALGORITHM 3: VerifyKey generation algorithm.

among the traceability features. This ensures the correctness of the traceability process. The verifyKey generation algorithm is shown in Algorithm 3.

After generating verifyKey, a zero-knowledge engine (ZKe) is introduced to the phase. The TA first uses ZKe to compile and set up with verifyKey to generate specific keys and transfer them to the TDP for creating a witness and proof. Simultaneously, the TA generates the smart contract using the genContract() function to provide an interaction interface for the traceability application layer. The TDP receives the keys to generating the witness using the related raw production data and then creates a proof that can be published to the public. The proof and contract generation algorithm is shown in Algorithm 4.

In the above algorithm, the TA publishes a smart contract address to the public to verify the proof. Through the TA, the TDP can first receive the traceability features and verification keys and then generate and publish the proof using raw production data. Meanwhile, the TDA can verify the proof through the smart contract in an open manner and finally reach traceability purpose in the verification phase.

3.3.4. Verification Phase. After the completion of the above phases, the TA and TDP can provide the contract address, proof, and traceability features. The TDA can analyze the traceability features with common knowledge and judge the expected result. If the TDP fails to generate a related proof, the traceability failed, and the TDA and TA might negotiate new traceability features and restart the traceability process.

It was worth noting that traceability is based on traceability features that are generated with expert or public knowledge. Simultaneously, the related smart contract was permanently deployed in the blockchain, which means that the smart contract with the related proof could be reused to simplify the traceability process. The algorithm designed in the verification stage is shown in Algorithm 5.

In the verification phase, according to the judgement of traceability features, the TDA, TDP, and TA may complete the traceability process in an open and privacy-preserving way. The traceability system for this paper could be effectively applied to typical traceability scenarios such as anti-counterfeiting verification [49], standard execution proofs [50], or abnormal investigations [51]. Furthermore, in the

Input: verifyKey, PID, TF
Output: Proof, contractAddr

- (1) TA.compile(verifyKey)
- (2) TA.setup()
- (3) SC = TA.genContract()
- (4) verifyKey,TF,PID = TDP.recv()
- (5) rawData = TDP.getRawDataByID()
- (6) witness = ZK.genWitness(rawData, verificationKey)
- (7) proof = Verify.genProof(Witness)
- (8) TDP.send(proof,TA)
- (9) TA.publish(proof|contractAddr)

ALGORITHM 4: Proof and contract generation algorithm.

Input: proof, TF

Output: TP, result

- (1) TDA.subscribe(TA,TDP)
- (2) TraceabilityResult = TDA.verify(ProofAddr, Proof|ProofFeature)
- (3) If TraceabilityResult == satisfied:
- (4) Finish tracing
- (5) If TraceabilityResult == not satisfied
- (6) Restart traceability process()
- (7) Return tracing failed

ALGORITHM 5: Verification algorithm.

above relevant traceability process, the TA could be industrial entities or even production experts. Those TA may create a competitive environment for traceability feature generation. As more new technologies have been imported into the production process, a competitive relationship between TAs may be beneficial in raising the availability of the traceability system.

4. Analysis with Discussion

4.1. Security Analysis. Security issues in network attacks and privacy protection are discussed in this section. For network attacks, in the traditional traceability system, malicious intruders may launch Distributed Denial of Service (DDoS) attack or Advanced Persistent Threat (APT) attacks in the centralized server [52]. Furthermore, the traceability data may be tampered with to destroy the traceability system's authority. However, the blockchain-based traceability system designed in this paper is immune to DDoS attacks and APT attacks [53]. All data digests stored in the blockchain for privacy protection are mapped with raw production data. With the importing of the zero-knowledge proof scheme, the traceability process can be transferred into a proof of related traceability features. The system described in this paper can avoid transferring the raw production data between any entities or individuals and so finally realize the traceability of privacy data. With the traceability system based on smart contracts, the entire traceability process could be monitored by the public, enhancing the traceability compliance and openness of the process. The system designed in this paper

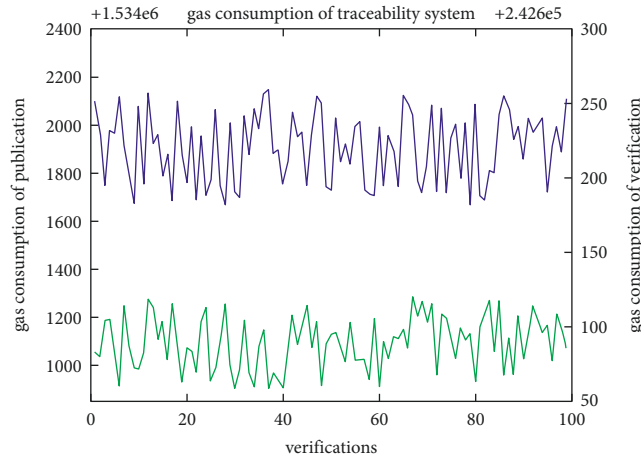


FIGURE 3: Gas consumption of publication and verification.

can be used to promote the industrial Internet, help it achieve a broader range of joint production, and provide an essential basis for trust [54].

Despite the network attack launched by malicious intruders, the blockchain itself may also have security issues such as transaction-related attacks (e.g., double spending attack), consensus failure, and smart contract-oriented attacks. In our work, we deployed the whole application on Ethereum environment to realize the availability of traceability process and assumed that the consensus process, smart contract execution, or other application process logic are operating without malicious behavior. Further research was deeply discussed in related works [55]; we just make the assumption above in order to reach the availability of traceability system.

4.2. Available Analysis. In this paper, zkSNARK toolkit was introduced to form a privacy-preserving scheme. According to the generated traceability features, a smart contract could be published in the Ethereum blockchain. Our experimental environment included an i7-6700 CPU and 8 GB of memory; considering the highly available and widely used blockchain system, a widely used IDE with Ethereum testnet [56] was also used. We designed the test with 1 to 100 traceability features and generated a related proof and smart contract. However, the calculation of Ethereum is not free; the verification and the publication of a smart contract could require a computing fee (gas). In this situation, gas could be treated as a cost of traceability process, and the gas consumption of contract publication and verification is shown in Figure 3.

As shown in the figure above, the gas consumption does not significantly increase as the number of traceability features increases. The gas consumptions of smart contract publication and validation are approximately 1535000 and 242800. With the current price of the Ethereum, the costs are approximately 0.15 ETH and 0.02 ETH. As shown above, the publication and verification may increase the cost of traceability. In contrast, in traceability scenarios, the gas consumption could be optimized, such as through the production

of a continuous linear process that could be implemented in a contract for multiple data validations and could finally reduce the gas consumption of publication and verification. At the same time, the system designed in this paper is built in the Ethereum testnet environment, and the traceability system in this paper can also be deployed into mainstream authorized blockchain systems, such as Hyperledger. Through corresponding smart contracts, the extra cost of publication and verification may effectively be avoided.

4.3. Comparative Analysis. In the comparative analysis, this section compares our traceability system with the traceability systems constructed in related studies to demonstrate the advantages of our approach. In this paper, we choose three typical models of traceability systems: a traceability system designed without blockchain [11] (Centralized), a blockchain-based traceability system without a privacy-preserving scheme [30–35], and a blockchain-based traceability system with a privacy-preserving scheme [36, 37]. Although the traceability systems we choose to compare were designed for different purposes and industrial environments, the issues in each typical model are common; therefore, these traceability system models were selected for comparative analysis of data reliability, traceability target, privacy protection scheme, attack resistance, and cost of traceability. The analysis results are shown in Table 1.

Unlike other related traceability systems, this paper constructed a traceability system using a decentralized architecture based on blockchain; on this basis, unlike a centralized traceability system, data reliability and availability could be fully guaranteed. At the same time, our approach can defend against DDoS attacks, and, due to the privacy protection scheme of zero-knowledge proofs together with authentication, our approach realizes a traceability of privacy data in industrial production that is superior to other decentralized traceability systems and could effectively build trust relationships between industrial entities. For the comparison of traceability cost, a centralized traceability system needs to spend much to build and maintain a traceability system, that is, the traceability system

TABLE 1: Comparative of traceability systems.

| | Traceability target | Attack resistance | Privacy protection | Data reliability | Traceability cost |
|--|----------------------------------|-------------------|--|------------------|---|
| Our approach | Privacy data/ nonprivacy data | √ | Certification + zero- knowledge proof | √ | According to the contract consumption of calculation |
| Centralized | Nonprivacy data | x | Certification | x | Determined by the traceability system creator |
| Blockchain-based without privacy-preserving | Nonprivacy data | √ | None | √ | According to the contract consumption of calculation |
| Blockchain-based with privacy-preserving | Privacy data | √ | Certification | √ | According to the contract consumption of calculation |

manager may have the right to fix the price of traceability service. It may influence the cost of traceability system users. In contrast, in our approach, pricing was determined by the calculation cost of the smart contract. The TA was introduced to make the price of traceability more flexible and open. The TDA and TDP would also benefit from the traceability process.

5. Conclusions and Future Work

With the development of industrial production, the industrial Internet may create many more opportunities for industrial entities of all sizes. As a key part of building industrial cooperation between entities, the traceability system plays a significant role in the development of industrial Internet. The traceability system designed in this paper achieves traceability for privacy production data, which makes it possible to objectively judge the quality of products in order to raise production quality or optimize supply chain structure. On the other hand, the privacy-preserving scheme designed in this paper raises the willingness of parties to share data, which may take good effect to raise the production capacity or reduce the resource consumption.

In order to make a stronger trust relationship between industrial entities in industrial Internet, our work could be combined with anticounterfeiting system [57] to reach a higher data privacy level or applied in production lifecycle management to reach sustainable manufacturing [58] with privacy. We would research the feasibility of those application scenarios with our work in the future and our work could make contribution to meeting the privacy demand in industrial Internet to some extent.

It is worth mentioning that our work still needs to be improved. Firstly, in this paper, the traceability feature generation process still depends on traceability knowledge held by a third party or public production standard [59]. Future studies should focus on areas such as artificial intelligence or industrial big data. Traceability features will be an important research direction in our future work. Secondly, security issues also need to be further researched, especially the security-oriented in blockchain system and smart contract. Our work is based on the premise of non-malicious blockchain nodes; security protection schemes should be further discussed in order to provide more secure environment for blockchain [12].

Data Availability

The gas consumption data used to support the findings of this study are included in the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Key R&D Program of China under Grant no. 2021YFB3300900, the National Natural Science Foundation of China (no. 62072336), and the New Generation Artificial Intelligence Technology Major Project of Tianjin (no. 19ZXZNGX00080).

References

- [1] Y. Wang and M. Singgih, J. Wang, M. Rit, "Making sense of blockchain technology: how will it transform supply chains?" *International Journal of Production Economics*, vol. 211, pp. 221–236, 2019.
- [2] M. Balog and L. Knapíková, "Advances of intelligent techniques used in Industry 4.0: proposals and testing," *Wireless Networks*, vol. 27, 2019.
- [3] R. Y. Zhong, X. Xu, E. Klotz, and S. T. Newman, "Intelligent manufacturing in the context of industry 4.0: a review," *Engineering*, vol. 3, no. 5, pp. 616–630, 2017.
- [4] D. Li, E. L. Xu, and L. Li, "Industry 4.0: state of the art and future trends," *International Journal of Production Research*, vol. 56, no. 8, pp. 2941–2962, 2018.
- [5] S. Cisneros-Cabrera, G. Pishchulov, P. Sampaio, N. Mehandjiev, Z. Liu, and S. Kununka, "An approach and decision support tool for forming Industry 4.0 supply chain collaborations," *Computers in Industry*, vol. 125, Article ID 103391, 2021.
- [6] F. Kerschbaum, A. Schroepfer, A. Zilli et al., "Secure collaborative supply-chain management," *Computer*, vol. 44, no. 9, pp. 38–43, 2011.
- [7] G. B. Zhang, Y. Ran, and X. L. Ren, "Study on product quality tracing technology in supply chain," *Computers & Industrial Engineering*, vol. 60, no. 4, pp. 863–871, 2011.
- [8] xxxx.
- [9] R. Naderi, M. Shafiei Nikabadi, A. Alem Tabriz, and M. S. Pishvae, "Supply chain sustainability improvement using exergy analysis," *Computers & Industrial Engineering*, vol. 154, no. 1, Article ID 107142, 2021.

- [10] J. Q. Li, F. R. Yu, G. Deng, C. Luo, Z. Ming, and Q. Yan, "Industrial internet: a survey on the enabling technologies, applications, and challenges," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1504–1526, 2017.
- [11] S. Ammendrup and L. O. Barcos, "Aplicación de los sistemas de trazabilidad," *Revue Scientifique et Technique de l'OIE*, vol. 25, no. 2, pp. 763–773, 2006.
- [12] H. N. Dai, Z. Zheng, and Y. Zhang, "Blockchain for internet of things: a survey," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8076–8094, 2019.
- [13] 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, 2018.
- [14] S. Johar, N. Ahmad, W. Asher, H. Cruickshank, and A. Durrani, "Research and applied perspective to blockchain technology: a comprehensive survey," *Applied Sciences*, vol. 11, no. 14, p. 6252, 2021.
- [15] A. Manimuthu, R. Sreedharan, and D. Marwaha, "A literature review on bitcoin: transformation of crypto currency Into a global phenomenon," *IEEE Engineering Management Review*, vol. 47, no. 1, pp. 28–35, 2019.
- [16] S. S. Kushwaha, S. Joshi, D. Singh, M. Kaur, and H.-N. Lee, "Systematic review of security vulnerabilities in ethereum blockchain smart contract," *IEEE Access*, vol. 10, pp. 6605–6621, 2022.
- [17] S. Wang, L. Ouyang, Y. Yuan, X. Ni, X. Han, and F. Y. Wang, "Blockchain-Enabled smart contracts: architecture, applications, and future trends," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 11, pp. 2266–2277, 2019.
- [18] J. Leng, S. Ye, M. Zhou et al., "Blockchain-secured smart manufacturing in industry 4.0: a survey," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 1, pp. 237–252, 2021.
- [19] A. Bedin, M. Capretz, and S. Mir, "Blockchain for Collaborative Businesses," *Mobile Networks and Applications*, vol. 26, pp. 1–8, 2020.
- [20] K. Demestichas, N. Peppas, T. Alexakis, and E. Adamopoulou, "Blockchain in agriculture traceability systems: a review," *Applied Sciences*, vol. 10, no. 12, p. 4113, 2020.
- [21] J. F. Galvez, J. C. Mejuto, and J. Simal-Gandara, "Future challenges on the use of blockchain for food traceability analysis," *TRAC Trends in Analytical Chemistry*, vol. 107, pp. 222–232, 2018.
- [22] S. Soni and B. Bhushan, "A Comprehensive survey on Blockchain: working, security analysis, privacy threats and potential applications," in *Proceedings of the 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICT)*, July 2019.
- [23] J. Brandt, I. Damgard, P. Landrock, and T. Pedersen, "Zero-knowledge authentication scheme with secret key exchange," *Journal of Cryptology*, vol. 11, no. 3, pp. 147–159, 1998.
- [24] D. Catalano and I. Visconti, "Hybrid commitments and their applications to zero-knowledge proof systems," *Theoretical Computer Science*, vol. 374, no. 1-3, pp. 229–260, 2007.
- [25] J. Kim, J. Lee, and H. Oh, "Simulation-extractable zk-SNARK with a single verification," *IEEE Access*, vol. 8, Article ID 156569, 2020.
- [26] Y. Zhang, Y. Long, Z. Liu, Z. Liu, and D. Gu, "Z-channel: Scalable and Efficient Scheme in Zerocash," *Information Security and Privacy*, Springer, Cham, Switzerland, 2018.
- [27] I. Weber, X. Xu, R. Riveret, G. Governatori, A. Ponomarev, and J. Mendling, "Untrusted Business Process Monitoring and Execution Using Blockchain," *Business Process Management*, Springer International Publishing, Berlin, Germany, 2016.
- [28] D. Berdik, S. Otoum, N. Schmidt, D. Porter, and Y. Jararweh, "A survey on blockchain for information systems management and security - ScienceDirect," *Information Processing & Management*, vol. 58, no. 1.
- [29] X. C. Chen, D. Peng, Y. Nai-qing, and M.-X. Bi, "Study on discrete manufacturing quality control technology based on big data and pattern recognition," *Mathematical Problems in Engineering*, vol. 2021, Article ID 8847094, 10 pages, 2021.
- [30] T. P. Tsai and F. C. Wang, "Improving supply chain management: a model for collaborative quality control advanced semiconductor manufacturing," in *Proceedings of the 2004. ASMC '04. IEEE Conference and Workshop IEEE*, Boston, MA, USA, May 2004.
- [31] P. Helo and A. Shamsuzzoha, "Real-time supply chain—a blockchain architecture for project deliveries," *Robotics and Computer-Integrated Manufacturing*, vol. 63, Article ID 101909, 2020.
- [32] X. N. Zhu, G. Peko, D. Sundaram, and S. Piramuthu, "Blockchain-based agile supply chain framework with IoT," *Information Systems Frontiers*, pp. 1–16.
- [33] G. Zhao, S. Liu, C. Lopez, H. Lu, S. Elgueta, and B. M. Boshkoska, "Blockchain Technology in Agri-Food Value Chain Management: A Synthesis of Applications, Challenges and Future Research Directions - ScienceDirect," *Computers in Industry*, vol. 109, 2019.
- [34] T. K. Agrawal, V. Kumar, R. Pal, L. Wang, and Y. Chen, "Blockchain-based framework for supply chain traceability: a case example of textile and clothing industry," *Computers & Industrial Engineering*, vol. 154, Article ID 107130, 2021.
- [35] M. Uddin, "Blockchain Medledger: hyperledger fabric enabled drug traceability system for counterfeit drugs in pharmaceutical industry," *International Journal of Pharmaceutics*, vol. 597, Article ID 120235, 2021.
- [36] M. P. Caro, M. S. Ali, M. Vecchio, and R. Giuffreda, "Blockchain-based Traceability in Agri-Food Supply Chain Management: A Practical Implementation," in *Proceedings of the 2018 IoT Vertical and Topical Summit on Agriculture - Tuscany (IOT Tuscany)*, pp. 1–4, Tuscany, Italy, May 2018.
- [37] K. Yang, D. Forte, and M. Tehranipoor, "ReSC: an RFID-enabled solution for defending IoT supply chain," *ACM Transactions on Design Automation of Electronic Systems*, vol. 23, 2018.
- [38] Z. Wang, Z. Zheng, W. Jiang, and S. Tang, "Blockchain-Enabled data sharing in supply chains: model, operationalization, and tutorial," *Production and Operations Management*, vol. 30, no. 7, pp. 1965–1985, 2021.
- [39] J. Eberhardt and S. Tai, "ZoKrates - scalable privacy-preserving off-chain computations," in *Proceedings of the 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data*, July 2018.
- [40] M. Westerkamp and J. Eberhardt, "zkRelay: facilitating Sidechains using zkSNARK-based Chain-Relays," in *Proceedings of the IEEE SECURITY & PRIVACY ON THE BLOCKCHAIN (IEEE S&B 2020)*, September 2020.
- [41] M. K. Ibrahim, *Robust Electronic Voting System Using Homomorphic Encryption Protocol and Zero-Knowledge Proof*, vol. 5, no. 1, 2016.
- [42] A. A. Rasheed, R. N. Mahapatra, and F. H. Lup, "Adaptive Group-Based Zero Knowledge Proof-Authentication Protocol (AGZKP-AP) in Vehicular Ad-Hoc Networks," *IEEE*

- Transactions on Intelligent Transportation Systems*, vol. 21, 2019.
- [43] S. H. Jeong and B. Ahn, "Implementation of real estate contract system using zero knowledge proof algorithm based blockchain," *The Journal of Supercomputing*, vol. 77, no. 10, Article ID 11881, 2021.
- [44] H. Qi, Z. Wan, Z. Guan, and X. Cheng, "Scalable decentralized privacy-preserving usage-based insurance for vehicles," *IEEE Internet of Things Journal*, vol. 8, p. 1, 2020.
- [45] M. Umar, Z. Wu, and X. Liao, "Channel characteristics aware zero knowledge proof based authentication scheme in body area networks," *Ad Hoc Networks*, vol. 112, no. 9, Article ID 102374, 2021.
- [46] C. Huang, D. Liu, J. Ni, R. Lu, and X. Shen, "Achieving accountable and efficient data sharing in industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 1416–1427, 2021.
- [47] X. Yang, "Design and application of safe production and quality traceability system for vegetable," *Transactions of the Chinese Society of Agricultural Engineering*, vol. 24, no. 3, pp. 162–166, 2008.
- [48] M. Thakur and C. R. Hurburgh, "Framework for implementing traceability system in the bulk grain supply chain," *Journal of Food Engineering*, vol. 95, no. 4, pp. 617–626, 2009.
- [49] Y. Lu, P. Li, and H. Xu, "A Food anti-counterfeiting traceability system based on Blockchain and Internet of Things," *Procedia Computer Science*, vol. 199, pp. 629–636, 2022.
- [50] Z. Ge, P. Li, W. Ren, S. Hu, Q. Yin, and Q. Zhang, "Quantity traceability method of 1000 kV standard voltage transformers," in *Proceedings of the 2020 IEEE International Conference on High Voltage Engineering and Application (ICHVE)*, September 2020.
- [51] Y. Cai, X. Li, M. Li et al., "Traceability and quality control in traditional Chinese medicine: from chemical fingerprint to two-dimensional barcode," *Evidence-based Complementary and Alternative Medicine*, vol. 2015, Article ID 251304, 6 pages, 2014.
- [52] R. Mahmoud, T. Yousuf, F. Aloul, and I. Zualkernan, "Internet of Things (IoT) Security: Current Status, Challenges and Countermeasures," in *Proceedings of the 2015 10th International Conference for Internet Technology and Secured Transactions (ICITST)*, London, UK, December 2015.
- [53] C. S. Kouzinopoulos, G. Spathoulas, K. M. Giannoutakis et al., "Using Blockchains to Strengthen the Security of Internet of Things," *Security in Computer and Information Sciences*, Springer, Cham, Switzerland, 2018.
- [54] D. Li, C. Li, and R. Gu, "Evolutionary game analysis of promoting industrial internet platforms to empower manufacturing SMEs through value cocreation cooperation," *Discrete Dynamics in Nature and Society*, vol. 2021, Article ID 4706719, 14 pages, 2021.
- [55] J. Leng, M. Zhou, J. L. Zhao, Y. Huang, and Y. Bian, "Blockchain Security: A Survey of Techniques and Research Directions," *IEEE Transactions on Services Computing*, vol. 99, 2020.
- [56] D. Vashisth, P. Khandelwal, R. Johari, and V. Gaur, "Blockchain technology based smart contract agreement on REMIX IDE," in *Proceedings of the 8th International Conference on Signal Processing and Integrated Networks (SPIN 2021)*, Noida, India, August 2021.
- [57] J. Leng, P. Jiang, K. Xu et al., "Makerchain: a blockchain with chemical signature for self-organizing process in social manufacturing," *Journal of Cleaner Production*, vol. 234, pp. 767–778, 2019.
- [58] J. Leng, G. Ruan, P. Jiang et al., "Blockchain-empowered sustainable manufacturing and product lifecycle management in industry 4.0: A survey," *Renewable and Sustainable Energy Reviews*, vol. 132, 2020.
- [59] L. U. Tie, "On technical standardization and industrial standard strategy," *China Industrial Economy*, vol. 7, pp. 43–49, 2005.

Research Article

BCFDPS: A Blockchain-Based Click Fraud Detection and Prevention Scheme for Online Advertising

Qiuyun Lyu ¹, Hao Li ¹, Renjie Zhou ^{2,3}, Jilin Zhang ^{2,3}, Nailiang Zhao ²,
and Yan Liu ⁴

¹School of Cyberspace, Hangzhou Dianzi University, Hangzhou 310018, China

²School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China

³Key Laboratory of Complex Systems Modeling and Simulation of the Ministry of Education, Hangzhou Dianzi University, Hangzhou 310018, China

⁴Zhejiang Panshi Information Technology Co., Ltd., Hangzhou 310015, China

Correspondence should be addressed to Renjie Zhou; rjzhou@hdu.edu.cn

Received 25 November 2021; Revised 10 March 2022; Accepted 21 March 2022; Published 29 April 2022

Academic Editor: Jiewu Leng

Copyright © 2022 Qiuyun Lyu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Online advertising, which depends on consumers' click, creates revenue for media sites, publishers, and advertisers. However, click fraud by criminals, i.e., the ad is clicked either by malicious machines or hiring people, threatens this advertising system. To solve the problem, many schemes are proposed which are mainly based on machine learning or statistical analysis. Although these schemes mitigate the problem of click fraud, several problems still exist. For example, some fraudulent clicks are still in the wild since their schemes only discover the fraudulent clicks with a probability approaching but not 100%. Also, the process of detecting a click fraud is executed by a single publisher, which makes a chance for the publisher to obtain illegal income by deceiving advertisers and media sites. Besides, the identity privacy of consumers is also exposed because the schemes deal with the plain text of consumers' real identity. Therefore, in this paper, a blockchain-based click fraud detection and prevention scheme (BCFDPS) for online advertising is proposed to deal with the above problems. Specifically, the BCFDPS mainly introduces bilinear pairing to implicitly verify whether a consumer's real digital identity is contained in a click message to significantly avoid click fraud and employs a consortium blockchain to ensure the transparency of the detection and prevention process. In our scheme, the clicks by machines or fraud ones by a human can be accurately detected and prevented by media sites, publishers, and advertisers. Furthermore, ciphertext-policy attribute-based encryption is adopted to protect the identity privacy of consumers. The implementation and evaluation results show that compared with the existing click fraud detection and prevention schemes based on machine learning and statistical analysis, BCFDPS achieves detection of each fraudulent click with a probability of 100% and consumes lower computation cost; furthermore, BCFDPS adds functions of consumers' privacy protection and click fraud detection and prevention, compared to the existing blockchain-based online advertising scheme, by introducing limited communication cost (4,984 bytes) at lower storage cost.

1. Introduction

Nowadays, cost-per-click (CPC) is by far the most popular model used in online advertising [1]. An online advertising system mainly includes four entities [2–4], namely, consumers (Us), advertisers (ADEs), publishers (PUBs), and media sites (MSs). An ad promotion process includes seven steps such as publishing, clicking, paying, and so on, which is shown in Figure 1. The ADE's ad is published by PUB to U

on the website of MS, as shown in steps 1–3 in Figure 1. A *click* is counted when a *U* clicks on the ad, as shown in steps 4–5 in Figure 1. Then, ADE needs to pay advertising promotion fees to PUB because of these *clicks*, and PUB also pays advertising *click* fees to MS, as shown in steps 6–7 in Figure 1. There are mainly two types of implementations of online advertising system to publish ads. The first type is the traditional online advertising systems (Google, Twitter, etc.) which mainly rely on centralized servers. Also, inspired by

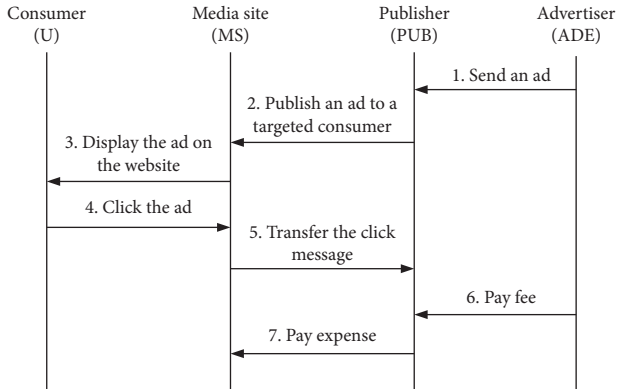


FIGURE 1: Process of online advertising system.

the tamper-proof and decentralized characteristics of blockchain, the other one is blockchain-based advertising systems [5, 6] which are implemented to achieve the transparency of an advertising business.

For higher revenue in the two types of implementations of online advertising systems, an ad will be published to a targeted U who is the potential consumer, which is called ad precision targeting. As we know, two main types of click fraud methods are designed by the attackers in the above two systems to gain extra illegal revenue: *the first type is to generate the repeated click messages by machines*. In detail, malicious advertisers use web crawler, botnet and proxy server, etc. to click an ad by machines [7–9] for exhausting his competitors’ budgets. *The second one is click fraud by a human*. Specifically, the malicious publishers and media sites recruit many people to click the same ad frame [10] or abuse the click history of legitimate users [11] to charge advertisers more ad promotion fees [4, 12]. Thus, these fake clicks generate additional budgets for advertisers, but do not create any revenue [13, 14], which undoubtedly disrupts the order of the advertising system.

Therefore, many click fraud detection and prevention schemes have been proposed to predict the authenticity of each click and to maintain the stability of the advertising system. According to the technology adopted, these schemes may be classified into two categories: *machine learning-based scheme* and *statistical analysis-based scheme*. *Machine learning-based schemes* [4, 7–9, 13, 15–22] utilize machine learning algorithms to train models that can judge whether a new click is fraudulent from massive click traffic. For example, in the scheme of [7], a machine learning algorithm based on convolutional neural networks and decision tree is designed to construct a classifier that distinguishes whether a click message is generated by machines or human beings according to the sensors of mobile device. However, the dataset used for training is easy to be mixed with fraudulent clicks in these schemes, causing the process of training a model to be susceptible to adversarial attack [23]. Thence, *statistical analysis-based schemes* [1, 10, 24–32] aim to mitigate the adversarial attacks. For instance, the schemes in [10, 25] predict malicious crowdsourcing platforms by clustering algorithms. Xu and Li [25] used the DP-means

clustering method to predict malicious groups, while Tian et al. [10], inspired by the DP-means clustering method, proposed a non-parametric method to solve the problem of malicious coalition fraud. Although they prevent the fraud of short-term malicious crowdsourcing platforms, their approaches are not enough for multiple traffics with long fraud intervals.

Apart from this, in the above two categories of schemes, the click fraud is still in the wild since they predict the click fraud only with a probability which is less than 100%. Also, the transparency of the click fraud detection and prevention process is not achieved, since these fraud detection and prevention algorithms are only implemented within a single central agency (publisher). That is, the publisher could gain illegal income from misreporting the number of the real clicks. Moreover, U ’s privacy is also leaked since these schemes analyze U ’s some original identity information such as the username and phone number.

Recently, as a tamper-proof and distributed technology, blockchain has attracted the attention of online advertising systems to significantly increase the trust between consumers and advertisers without additional costs and intermediaries. Specifically, by using a distributed ledger, the data related to the delivery of ads, clicks, and the analysis result of the real click number are all stored in the blockchain, which can be audited and verified by everyone [33]. On the other hand, people’s activities in physical space have been transferred to cyberspace increasingly. To build a better cyberspace, a digital identity that maps one-to-one with a physical identity in cyberspace is becoming the focus of the future. To this end, many digital identity infrastructures [34–38] have emerged to better manage user behavior in cyberspace.

Therefore, taking the above merits and problems into account, we introduce the blockchain and the existing digital identity infrastructures to detect and prevent fraudulent clicks for an online advertising system. The main contributions of this paper are summarized as follows.

- (1) We propose a blockchain-based click fraud detection and prevention scheme (BCFDPS) for online advertising, which significantly avoids clicking by machines and increases the cost of fraud ones by a human.
- (2) Whether a click is fraudulent can be confirmed directly in our scheme rather than predicted with a probability less than 100%. A consumer’s digital identity that is one and only mapping to a person in the physical world is embedded in a click message. That is, a fraudulent click does not contain a legitimate digital identity, and many duplicate clicks contain only the same digital identity.
- (3) When negotiating the ad billing fee between entities, the problem of tampering with the real number of clicks by media sites and publishers is solved because the transparency of the number in the click fraud detection and prevention process is realized through introducing a consortium blockchain. Specifically, the analysis result of the clicks is periodically recorded by publishers. The media sites and the

advertisers can also verify the result independently through the data in the blockchain.

- (4) The risk of leaking consumers' identity privacy from adversaries is alleviated by the bilinear pairing and ciphertext-policy attribute-based encryption without excessively affecting the publisher's accurate target of ads to consumers.

The rest of this article is organized as follows. Related works are discussed in Section 2. Section 3 reviews some preliminaries. Section 4 formulates the problem being addressed. Section 5 describes the proposed BCFDPS in detail. Security is analyzed in Section 6, and an experiment is designed and implemented in Section 7, followed by discussion and conclusion in Sections 8 and 9, respectively.

2. Related Work

2.1. Machine Learning-Based Schemes. Machine learning-based schemes are widely used in advertising fraud detection scenarios with large amounts of click data. User's click features are first extracted, then these features are used to train a model with the training dataset, and further the trained model is evaluated with the test dataset [13]. Oentaryo et al. [15] and Kanei et al. [4] mainly utilized random forest to detect click fraud in online advertising systems. But they are unable to catch coalition attacks involving multiple fraudulent approaches. Then, Wang et al. [16] presented CLUE in 2017, a novel recurrent neural network (RNN)-based online e-commerce transaction fraud detection system, and they deployed the CLUE on JD.com, serving over 220 million active users, to achieve real-time detection of fraudulent transactions. However, the CLUE in [16] will face gradient vanishing or gradient exploding problems when the click traffic is too complicated, leading to a poor fraud detection model. In 2018, Haider et al. [18] used two ensemble learning techniques, bagging and boosting algorithms, to train a model to detect and prevent click fraud. In 2019, support vector machine (SVM), K-nearest neighbor (KNN), AdaBoost, decision tree, and bagging were evaluated to detect a click by Almahmoud et al. [8]. In 2020, gradient tree boosting (GTB) algorithm was used to address the challenges encountered in effectively classifying fraudulent publishers [19]. Nevertheless, the user's identity privacy is exposed in [8, 18, 19] since they used the original identity information of consumers, such as the real username and the address of the visitor, to train models. Then, in 2021, two XGBoost-based schemes [21, 22] were proposed for click fraud detection, but both of them require manual classification of a large amount of click traffic in advance, which is time consuming. Apart from the problems mentioned above, according to the paper of Mikhailov and Trusov [23], all the schemes in this category are prone to adversarial attacks, for which they need a large number of samples as input to train the model. Also, these machine learning-based schemes can only determine whether the click is fraudulent with a probability approaching 100%.

2.2. Statistical Analysis-Based Schemes. Graph-based propagation approaches were first proposed in [24, 27, 29] to analyze the advertising traffic. The main idea of Stitelman et al. [24] is to use the co-visitation network between websites to identify media sites with a large amount of fraudulent traffic, but this approach relies on the fact that the experts have informed views about which websites look reasonable and which do not. Since it is difficult to collect all the users' data in a co-visitation network, Hu et al. [27] analyzed the behavior characteristic of individual mobile advertising user and then reduced malicious user clicks. As a specific deployment of the idea in [27], Dong et al. [29] proposed FraudDroid, a novel hybrid approach, to detect ad frauds in mobile Android apps. It dynamically analyzes applications to build UI state transition graphs, collects their associated runtime network traffic, and then uses it to identify advertising fraud.

Additionally, a pattern-based click fraud detection scheme for mobile applications [32] was designed, and it mainly has two components: *offline pattern extractor* and *online fraud detector*. The *extractor* is responsible for extracting traffic patterns for ad and non-ad traffics, and the *detector* is in charge of monitoring network traffic and detecting click fraud with the traffic patterns. But the schemes in [29, 32] may fail to handle subsequent variant click fraud [39].

Different from the above graph-based and pattern-based analysis schemes, three contextual-based attributes concerning interarrival time (IAT), diurnal activity (DA), and eigenscore (ES) were analyzed in comparison-shopping services to calculate a click's credible score for detecting whether it is fake in [26]. Moreover, Meghanath et al. [28] proposed a new contextual outlier detection technology (ConOut) and applied it to the advertising domain to identify fraudulent publishers. Besides, the work in [30] presents Bag-of-Words algorithm to assess clicks in online advertising system, which is based on the concept of text search methods. However, the outlier detection technology in [28] and the Bag-of-Words algorithm in [30] involve users' real username and address of the visitor, which reveals user's identity privacy.

In 2019, a novel inference technique (Clicktok) was developed to isolate click fraud attacks in [31]. Clicktok analyzes the traffic matrix, including matrix decomposition and construction, to propose two defenses, mimicry and bait-click. The mimicry isolates click spam by observing the reuse pattern of legitimate click traffic, and the ad network uses bait-clicks to watermark the channel periodically, which sets off watermark detectors when an attacker harvests and reuses a legitimate clickstream in the channel. But the Clicktok does not have a good ability to prevent the new types of click fraud whose traffic matrix is similar to the one of a normal click. On the other hand, these statistical analysis-based methods are deployed in publishers' devices and they do not achieve the transparency of the click fraud detection and prevention process for each entity in the advertising system. In addition, the statistical analysis-based schemes leak user's identity privacy since they analyze the

original data which includes user's real username, address of the user, etc.

2.3. Blockchain-Based Advertising System. Recently, blockchain has been widely adopted in many disciplines owing to its trusted computing model and open nature. Therefore, blockchain-based advertising systems [5, 6] are proposed to provide trust between entities in advertising business. The scheme of Liu et al. [5] develops transparent and accountable vehicular local advertising (TAVLA) by utilizing the message digest, multi-party verification, and smart contract of blockchain. Specifically, the hash of the advertising information database is stored in the blockchain, and the code of the advertising query functions is stored off-chain. A vehicle user first requests the ad off-chain, and then the off-chain query result will be verified and assembled in the blockchain smart contract, and finally, the smart contract sends the result to the user. However, the communication data in this scheme is in plaintext, which is not secure for online advertising systems. Moreover, to improve the low trust caused by the click fraud in the online advertising system, Ding et al. [6] designed and implemented a blockchain-based digital advertising media system (B2DAM) and deployed the business logic based on smart contracts and Hyperledger SDK. But the communication cost between multiple blockchains in their scheme is expensive, as they read and write messages too many times on the blockchains.

All in all, in the above schemes, all real-time interactions of entities are settled in the blockchain, and each ad delivery and click behavior are recorded in the blockchain, so that the throughput is hard to meet the high concurrency in the advertising system. As a result, we periodically record the analysis results of the clicks on the blockchain in our scheme.

3. Preliminaries

3.1. Blockchain. Blockchain records all the transactions which are generated in a peer-to-peer network, and it is actually a decentralized ledger system. In the system, all the blocks include the hash of the previous block; in this way, they are linked together by the hash, and a blockchain is formed. According to the decreasing order of decentralization, the blockchain consists of three categories: public blockchain, consortium blockchain, and private blockchain [40, 41]. The public blockchain is open to all nodes, and everyone can read and write data on it. The consortium blockchain is partially open since it is managed by several organizations and only the authenticated members can access and record data on it. Also, the private blockchain is considered to be centralized as it is fully controlled by a single enterprise or organization. Considering that several enterprises and organizations are included in the advertising system, the consortium blockchain is adopted in our scheme.

3.2. Shamir (t, n) Threshold Secret Sharing Algorithm. A threshold secret sharing algorithm [42] was proposed by Shamir in 1979 to share a master secret in a safe way. In the

literature, a trusted center (TC) splits the master secret K into n sub-secrets ($K_1, K_2, K_3, \dots, K_n$) and then distributes them to n participants ($U_1, U_2, U_3, \dots, U_n$). The master secret K cannot be reconstructed with fewer than t sub-secrets and the specific steps are described as follows. Firstly, a random $(t-1)$ -th degree polynomial as $f(x) = a_0 + a_1x + \dots + a_{t-1}x^{t-1}$ is generated by the TC, in which the master secret $K = f(0) = a_0$. Then, the TC calculates n sub-secrets $K_i = f(x_i)$, $i = 1, 2, \dots, n$ and allocates K_i to U_i secretly. Next, when U_i receives K_i , he saves it safely. Finally, the master key K can be recovered by the TC using the *Lagrange interpolation formula*: $f(x) = \sum_{i=1}^t K_i \sum_{j=1, j \neq i}^t \frac{x - x_j}{x_i - x_j}$, and the master key $K = f(0)$.

3.3. Bilinear Mapping. Assume G_1, G_2 , and G_T denote three additive and multiplicative cyclic groups of the same order p , where p is a large prime and g is the generator of G_1, G_2 . Besides, $\psi: G_2 \rightarrow G_1$ is an isomorphism, and G_1, G_2, G_T are equipped with pairing. The bilinear pairing mapping $e: G_1 \times G_2 \rightarrow G_T$ satisfies the following properties [43, 44].

- (1) *Bilinear*: $\forall P \in G_1, Q \in G_2$ and $a, b \in \mathbb{Z}_p^*$, where $\mathbb{Z}_p^* = [1, 2, \dots, p-1]$; if $e(aP, bQ) = e(P, Q)^{ab}$, the mapping $e: G_1 \times G_2 \rightarrow G_T$ is said to be bilinear.
- (2) *Non-degenerate*: there exists $P \in G_1, Q \in G_2$ such that $e(P, Q) \neq 1_{G_T}$.
- (3) *Computability*: $\forall P \in G_1, Q \in G_2$, there is an efficient algorithm to compute $e(P, Q)$.

The group G_T that possesses such a map e is called a bilinear group.

3.4. Ciphertext-Policy Attribute-Based Encryption (CP-ABE). CP-ABE schemes [45, 46] are designed to realize complex access control on encrypted data. In CP-ABE, a party wishing to encrypt a message M specifies a policy by an access tree T , and the private key must meet the policy to decrypt it, where the access tree is constructed by the party and the private key is generated by a set of descriptive attributes S of the decryptors. In the access tree T , each non-leaf node represents a threshold gate, described by its children and a threshold value, and each leaf node x of the tree is described by an attribute y and a threshold value t_x . To facilitate working with the access trees, the parent of the node x is described by $\text{parent}(x)$, and the function $\text{att}(x)$ is defined only if x is a leaf node and denotes the attribute associated with the leaf node x . The access tree T also defines an ordering between the children of every node, that is, the children of a node are numbered from 1 to num . The function $\text{index}(x)$ returns such a number associated with the node x .

According to [47], the four algorithms of the bilinear mapping-based CP-ABE scheme are as follows:

- (1) *Setup*: this algorithm gives the public parameters PK and master key MK. It chooses a bilinear group G_1 of prime order p with generator g . Next, it chooses two random exponents $\alpha, \beta \in \mathbb{Z}_p$. Then, the public key is published as

$$PK = \{G_1, g, h, f, l\}, \quad (1)$$

where $h = g^\beta$, $f = g^{1/\beta}$, $l = e(g, g)^\alpha$, and the master key MK is (β, g^α) .

- (2) *Encrypt* (PK, M, T): this algorithm encrypts message M to get ciphertext CT using the public parameters PK and the access tree T . In detail, it first chooses a polynomial p_x for each node x (including the leaves) in the tree T , in which the degree d_x of the polynomial p_x is one less than the threshold value t_x , that is, $d_x = t_x - 1$. Starting with the root node R in T , the algorithm chooses a random $s \in \mathbb{Z}_p$ and sets $p_R(0) = s$. Then, it sets $p_x(0) = p_{\text{parent}(x)}(\text{index}(x))$. Finally, it lets Y be the set of leaf nodes in T , and the ciphertext CT is

$$CT = (T, \tilde{C} = M \cdot l^s, C = h^s, \forall y \in Y: C_y = g^{p_y(0)}, C'_y = (h(\text{att}(y)))^{p_y(0)}). \quad (2)$$

- (3) *KeyGen* (MK, S): the KeyGen algorithm outputs the private key SK using the master key MK and the attribute set S . Firstly, it chooses $r \in \mathbb{Z}_p$ and $r_j \in \mathbb{Z}_p$ for each attribute $j \in S$. Then, it computes the private key SK as

$$SK = (D = g^{(\alpha+r)/\beta}, \forall j \in S: D_j = g^r \cdot h(j)^{r_j}, D'_j = g^{r_j}). \quad (3)$$

- (4) *Decrypt* (CT, SK): this algorithm decrypts the ciphertext CT with the private key SK for people who satisfy the attribute set S . The decryption procedure is a recursive algorithm in which

$$\begin{aligned} \text{DecryptNode}(CT, SK, x) &= \frac{e(D'_x, C_x)}{e(D'_x, C'_x)} \\ &= e(g, g)^{r p_x(0)}. \end{aligned} \quad (4)$$

When x is the root node R in the tree T , it can be concluded that $\text{DecryptNode}(CT, SK, R) = e(g, g)^{r p_R(0)} = A$. Then, the message M can be computed by

$$\begin{aligned} \frac{\tilde{C}}{(e(C, D)/A)} &= \frac{\tilde{C}}{(e(h^s, g^{(\alpha+r)/\beta})/e(g, g)^{rs})} \\ &= M. \end{aligned} \quad (5)$$

4. Problem Statement

Many digital identity infrastructures [34–38] have emerged to better manage user's behavior in cyberspace, among which an identity management agency is responsible for generating and maintaining one-to-one mappings between digital identities and physical identities. The one-to-one mapping of digital identity infrastructures can prevent identity-based attacks (Sybil, whitewashing, etc.) in

cyberspace. Therefore, based on the existing infrastructures, we designed our blockchain-based click fraud detection and prevention scheme (BCFDPS) for online advertising system. To elaborate our scheme clearly, the main entities and procedures of existing digital identity infrastructures are also included.

4.1. System Model. The proposed BCFDPS consists of seven entities: identity management agency (IMA), entity identity blockchain (EIB), consumer (U), access behavior blockchain (ABB), media site (MS), publisher (PUB), and advertiser (ADE), where the IMA and EIB are the entities of the existing digital identity infrastructures, as shown in Figure 2.

- (i) IMA generates identities for U, PUB, and ADE, issues their identity licenses, and provides U with a signature on U's masked identity during the registration phase. IMA records the real identities of U, PUB, and ADE in EIB. Note: the IMA belongs to the existing digital identity infrastructures.
- (ii) EIB is responsible for recording the hash of digital identity in the advertising system other than MS. Also, it is a consortium blockchain maintained by several IMAs. Note: the EIB belongs to the existing digital identity infrastructures.
- (iii) U sends the encrypted masked identity and ad click messages to MS whenever he visits MS's website and clicks the ad.
- (iv) ABB is in charge of recording the information of PUB's advertising bidding, MS's forwarding results of U's click message, and PUB's analysis result of U's access behavior. The ABB is a consortium blockchain, which is controlled by many MSs and PUBs.
- (v) MS represents the media site between U and PUB. It is responsible for displaying PUB's ad for U and forwarding all the click messages of U for PUB. MS summarizes the result of ad bidding and the forwarding information and periodically records them in the ABB. MS can verify the click number independently for detecting and preventing click fraud.
- (vi) PUB publishes ADE's ad, joins the ad bidding process of MS, analyzes the effective ad clicks generated by U, and records the analysis results in the ABB. PUB can verify the click number independently for detecting and preventing click fraud.
- (vii) ADE sends an ad to PUB for publishing, and he can also detect and prevent click fraud alone through verifying the number of clicks in the ABB.

4.2. Security Model. In BCFDPS, we have the following security assumptions.

- (i) IMA and PUB are semi-honest and they will strictly follow the protocol but are curious about the information.

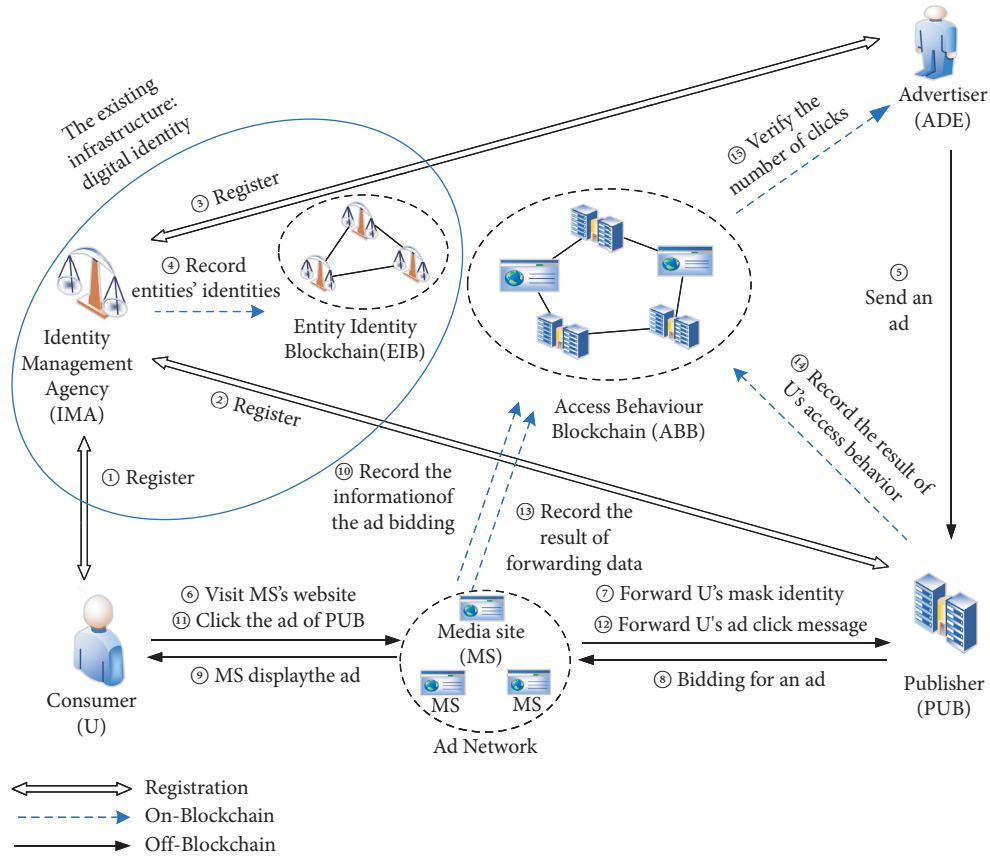


FIGURE 2: System model.

- (ii) U is considered as a malicious entity and he will intentionally click on the same ad many times out of profit or curiosity.
- (iii) MS is regarded as dishonest. It may deploy click fraud methods and even directly tamper with the statistical results of clicks to try to obtain extra illegal revenue from the PUB.
- (iv) ADE is also seen as a malicious entity. He may attempt to deliberately falsify his statistics to reduce the ad expenses from the PUB.
- (v) Two consortium blockchains, maintained by multiple IMAs, MSs, and PUBs, respectively, are fast and secure enough in recording transactions. Also, we assume that the standard cryptographic algorithms used in our scheme are secure and unbreakable.
- (vi) It is built upon the Canetti–Krawczyk (CK) threat model [48], in which any two parties could communicate via an unauthenticated network. Specifically, an adversary can fully control the communication in a probabilistic polynomial time and try to reveal, track, or even imitate U through sniffing and tampering with messages between U and MS.

- (vii) Corresponding to the physical identity, a person in cyberspace has his one and only digital identity. The U 's digital identity in each ad click message is protected by a masked identity and the random numbers, and the click message is generated by U 's browser plugin, where the plugin is assumed to be integrated in the browser in advance to protect U 's privacy.

4.3. *Design Goals.* According to the above system model and security model, the design goals of our scheme are as follows.

- (1) *No impact on ad precision targeting:* a PUB can still accurately target an ad to a U although the U 's digital identity is masked. In other words, only the PUB can link the U 's masked identity from different click messages.
- (2) *Acceptability of ad response speed:* the ad response speed in our scheme is acceptable for a U , even though the cryptographic algorithms are used to protect the U 's identity privacy in the process of publishing an ad.
- (3) *Transparency of ad billing fee:* MS, PUB, and ADE can count the real number of clicks on the same ad in

an independent way. That is, the process of verifying the ad billing fee is transparent between MS, PUB, and ADE.

5. Proposed BCFDPS

The BCFDPS is proposed to detect and prevent click fraud, and it is mainly divided into three phases, as shown in Figure 2. The first phase allows U , PUB, and ADE to register with the IMA and obtain their digital identities and identity licenses. Meanwhile, IMA stores the hash value of their identities on the EIB, as shown in steps ①–④ (note: the four steps belong to the existing digital identity infrastructure). The second phase permits MS and PUB to work together to recommend ADE's ad to U . U clicks the ad that he is interested in and MS records the hash value of the data that it forwarded in the ABB, as shown in steps ⑤–⑧. The last phase lets both PUB and ADE detect and prevent click fraud independently using the data in the ABB, which is shown in steps ⑨ and ⑩.

The detailed process of BCFDPS includes four phases: initialization, registration, ad publishing, and click fraud detection and prevention. To elaborate our scheme clearly, the notations and descriptions of BCFDPS are shown in Table 1.

5.1. Initialization. The digital identity is the cornerstone of cyberspace which is provided and validated whenever a user accesses the network services. The initialization of this section is not exclusive to our scheme. In other words, in order to describe our scheme clearly, the pivotal initialization of the existing digital identity infrastructure is described in this section. Specifically, the IMA performs initialization to generate its public and private keys, and the EIB generates the system public parameters PP. In addition, U , PUB, and ADE generate their public and private keys.

5.1.1. IMA Initialization. IMA initializes its public and private keys $PK_{\text{IMA}}/SK_{\text{IMA}}$. Then, IMA publishes PK_{IMA} in the system. In addition, IMA defines PUB's attribute set including but not limited to these attributes $S = \{\{\text{PUB}\}, \{\text{Hat} \cup \text{Pants} \cup \text{Shoes} \cup \dots\}\}$.

5.1.2. EIB Initialization. The EIB performs initialization to generate the system public parameters PP. Firstly, it selects a large prime p , an elliptic curve $E_p(a, b)$, and a base point P with order n under the finite field F_p . Then, it chooses a bilinear group G with generator g and two random numbers $\alpha, \beta \in \mathbb{Z}_p$. Next, it calculates parameters: $h = g^\beta$, $f = g^{1/\beta}$, $l = e(g, g)^\alpha$, and $MK = (\beta, g^\alpha)$, and publishes $PP = \{E_p(a, b), P, G, g, hf, l, MK\}$ in the system so that IMA can get PP. Lastly, EIB generates a shared private key x denoting the master key described in Section 3.2, and it uses the Shamir (t, n) threshold secret sharing algorithm [42] to distribute the sub-secrets of x to each IMA.

5.1.3. U , PUB, and ADE Initialization. U , PUB, and ADE also generate their own public and private keys, referred to as PK_U/SK_U , $PK_{\text{PUB}}/SK_{\text{PUB}}$, $PK_{\text{ADE}}/SK_{\text{ADE}}$.

TABLE 1: Notations and descriptions.

| Notation | Description |
|-------------------|--|
| PP | Public parameter generated by EIB |
| UID | U 's real digital identity |
| MUID | U 's masked identity |
| MS | The identity of media site |
| PUBID | The identity of publisher |
| ADEID | The identity of advertiser |
| ID_{ad} | The identity of an ad |
| x | The shared private key of EIB |
| S | The publisher's attribute set |
| U_{Sig} | The signature of U from IMA |
| PK_e | The public key of entity e |
| SK_e | The private key of entity e |
| IL_e | The identity license of entity e |
| AuS_e | The authentication symbol for entity e |
| SA_e | The secure authentication for entity e |
| ts, ts_1, ts_2 | The timestamp |
| $h(\cdot)$ | hash function: $\{0, 1\}^* \rightarrow \{0, 1\}^n$ |
| $U \parallel V$ | Concatenate operation between U and V |
| $E_a(b)/D_a(b)$ | Encrypt/decrypt message b with key a |
| $\text{Sig}_a(b)$ | The signature on message b with key a |
| M | Message generated after U clicks an ad |

5.2. Registration. Similar to Section 5.1, the registrations of U , PUB, and ADE are not exclusive to our scheme. In other words, in order to describe our scheme clearly, the pivotal registration of the existing digital identity infrastructure is described in this section. Specifically, U registers with IMA to obtain his real identity UID, identity license IL_U , and the signature U_{Sig} . Similar to U , PUB registers with IMA to get its digital identity PUBID, identity license IL_{PUB} , and an attribute set S as an ad publisher. Also, ADE receives his digital identity ADEID and identity license IL_{ADE} from IMA.

5.2.1. U Registration (UR)

STEP UR1. IMA collects U 's biometric data, e.g., fingerprint, digitalizes the fingerprint to obtain the digitized data, and selects and assembles a set of unique code segments from the code library according to the data, and at the same time, the hash value of the code segments is calculated. The hash value is U 's real identity UID. Note that if a U is disguised by a machine or has already registered, the IMA would not generate an identity for the U . In order to issue an identity license IL_U to U , the IMA gathers other's sub-secrets and uses the Shamir (t, n) threshold secret recovering algorithm [42] to recover the shared private key x for calculating $IL_U = \text{UID} \cdot h(x) \cdot P$ and U 's masked identity $\text{MUID} = \text{UID} \cdot P$. Then, IMA sends $E_{PK_U}(\text{UID} \parallel IL_U \parallel U_{\text{Sig}} \parallel PP)$ to U 's browser plugin, where U_{Sig} is the signature of U from IMA, shown in (6), and PP refer to the public parameters in EIB. This process is shown in step ① in Figure 2.

$$U_{\text{Sig}} = \text{Sig}_{SK_{\text{IMA}}}(h(IL_U) \parallel \text{MUID}). \quad (6)$$

STEP UR2. At last, IMA records $h(\text{UID})$ in EIB, which is used for accountability when a click fraud happens. This process is shown in step ④ in Figure 2.

5.2.2. PUB Registration (PUBR)

STEP PUBR1. IMA generates PUB's identity PUBID and an attribute set $S = \{\{\text{PUB}\}, \{\text{Hat} \cup \text{Food} \cup \dots\}\}$ according to the business scope of PUB. Hereafter, similar to *STEP URI1*, IMA generates PUB's identity license $\text{IL}_{\text{PUB}} = \text{PUBID} \cdot h(x) \cdot P$ and sends $E_{\text{PK}_{\text{PUB}}}(\text{PUBID} \parallel \text{IL}_{\text{PUB}} \parallel S \parallel \text{PP})$ to PUB. This process is shown in step ② in Figure 2.

STEP PUBR2. At last, IMA records the $h(\text{PUBID})$ in EIB for supervision when a click fraud appears. This process is shown in step ④ in Figure 2.

5.2.3. ADE Registration (ADER)

STEP ADER1. IMA generates ADE's identity ADEID and identity license $\text{IL}_{\text{ADE}} = \text{ADEID} \cdot h(x) \cdot P$ and sends $E_{\text{PK}_{\text{ADE}}}(\text{ADEID} \parallel \text{IL}_{\text{ADE}})$ to ADE. This process is shown in step ③ in Figure 2.

STEP ADER2. At last, IMA records $h(\text{ADEID})$ in EIB to supervise when a click fraud arises. This process is shown in step ④ in Figure 2.

5.3. Ad Publishing. To obtain higher revenue, PUB often publishes ads to a targeted U through MS's ad bidding. Then, U clicks the ad that he is interested in.

5.3.1. Publisher Publishes an Ad (PPA). This phase deals with the process that a browser plugin sends U 's masked identity in ciphertext to PUB and PUB displays the related ads to the targeted U , as shown in steps ①–⑥ in Figure 3.

STEP PPA1. ADE sends an ad to PUB for publication. Then, ADE and PUB reach a consensus on ADE's ad and create the ad's identity ID_{ad} , as shown in step ① in Figure 3.

STEP PPA2. U sends his masked identity in ciphertext (instead of a real identity in the real world) to PUB for getting the ad that he is interested in, protecting his privacy. Firstly, U visits MS's website, and U 's web browser plugin encrypts the secret U_{Sig} through the CP-ABE algorithm to prevent entities other than the collection of PUBs from obtaining U 's identity privacy. Then, U sends the ciphertext CT to the MS. After that, the MS directly broadcasts the CT to the PUBs cooperating with the MS. Here is the specific process. U uses the public parameters PP and defines an access tree T according to the attributes of ads that he is interested in. The format of T is shown in Figure 4, where "1/2" means that PUB must satisfy at least one of the two attributes $\{\text{Hat} \cup \text{Shoes}\}$. Then, U encrypts the secret U_{Sig} to get the ciphertext CT.

$$\begin{aligned} \text{CT} &= (T, \tilde{C} = (U_{\text{Sig}} \parallel ts_1) \cdot l^s, C = h^s, \forall y \in Y: \\ &C_y = g^{p_y(0)}, C'_y = (h(\text{att}(y)))^{p_y(0)}), \end{aligned} \quad (7)$$

where ts_1 is the timestamp, $l = e(g, g)^\alpha$, $h \in \text{PP}$, $s \in \mathbb{Z}_p$ is a random number, p_y is a polynomial for each node y in T , and $p_y(0) = s$ and $\text{att}(y)$ are the attributes associated with the leaf node y .

After U visits MS's website, U 's browser plugin sends CT to MS, which is then directly broadcast to different PUBs by MS. This process is as in steps ② and ③ in Figure 3.

STEP PPA3. Next, PUB decrypts \tilde{C} in CT to get the secret U_{Sig} . Further, PUB gets U 's masked identity MUID from the U_{Sig} and decides whether to bid for an ad according to the MUID, as shown in step ④ in Figure 3. The specific process of getting the U_{Sig} is as follows.

According to the attribute set S obtained as described in Section 5.2.2, PUB uses the master key MK in the public parameters PP to generate the decryption key SK according to

$$\text{SK} = (D = g^{(\alpha+r)/\beta}, \forall j \in S: D_j = g^r \cdot h(j)^{r_j}, D'_j = g^{r_j}), \quad (8)$$

where $r, r_j \in \mathbb{Z}_p$ are random numbers, $j \in S$ is an attribute, and α, β belong to the public parameter PP.

Then, PUB uses (9) to decrypt the leaf nodes of the access tree T with SK and CT when PUB's $S = \{\{\text{PUB}\}, \{\text{Hat} \cup \text{Food} \cup \dots\}\}$ satisfies the attributes which U requires.

$$\text{DecryptNode}(\text{CT}, \text{SK}, x) = \frac{e(D'_x, C_x)}{e(D_x, C'_x)} = e(g, g)^{r p_x(0)}, \quad (9)$$

where x is a leaf node in T . After PUB obtains all leaf nodes, it uses the Lagrangian interpolation formula to obtain the parent node, and this process is recursive until T 's root node is obtained. T 's root node is $A = e(g, g)^{r \cdot P_R(0)}$.

Next, PUB can get the secret U_{Sig} by

$$\begin{aligned} \frac{\tilde{C}}{(e(C, D)/A)} &= \frac{\tilde{C}}{(e(h^s, g^{(\alpha+r)/\beta})/e(g, g)^{rs})} \\ &= (U_{\text{Sig}} \parallel ts_1). \end{aligned} \quad (10)$$

At last, PUB decrypts U_{Sig} with IMA's public key PK_{IMA} and obtains the masked identity MUID of U . PUB searches its local database to get the portrait of MUID and decides whether to bid for the ad. If a PUB joins in the bidding process, it sends the ID_{ad} and the price fee to the MS. This bidding process will be executed by many PUBs. *STEP PPA4.* MS displays the ad of the bid winner and sends the PUBID, ADEID, PK_{PUB} , PK_{ADE} , and ad frame to the U , as shown in step ⑤ in Figure 3.

STEP PPA5. MS periodically (e.g., once a day) records the results of the ad bidding in the ABB sorted by periods and ID_{ad} s. The format of the results is

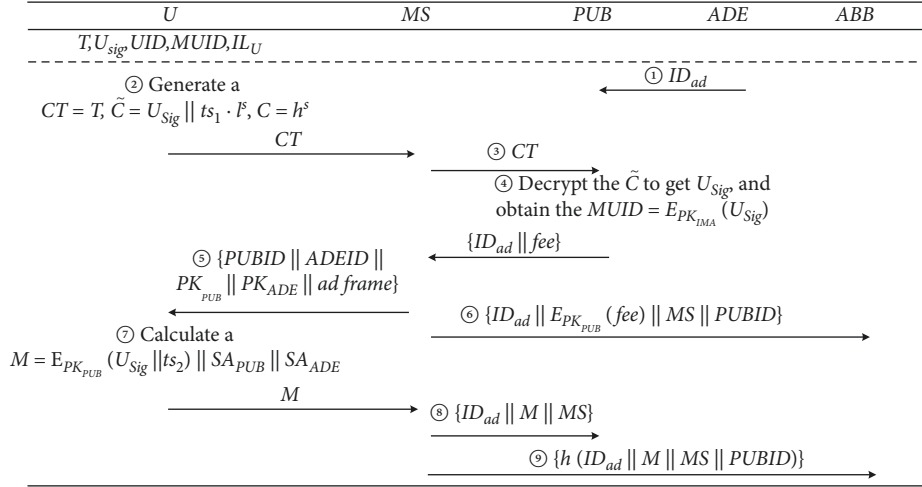
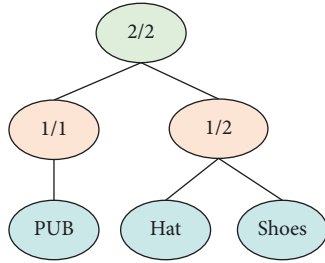


FIGURE 3: Process of publishing an ad to a targeted consumer.


 FIGURE 4: An example of U 's access tree T .

$\{ID_{ad} || E_{PK_{PUB}}(fee) || MS || PUBID\}$, where “fee” is the price that PUB should pay to MS after an ad is clicked once by U , as shown in step ⑥ in Figure 3.

5.3.2. U Clicks the Ad (UCA). U clicks the ad that he is interested in after MS displays it on the website. This section is shown in steps ⑦–⑨ in Figure 3.

STEP UCA1. U 's browser plugin gets PUBID, ADEID, PK_{PUB} , and PK_{ADE} from the ad frame and calculates $AuS_{PUB} = e(IL_U \cdot PUBID, P)^{h(IL_U)}$ and $AuS_{ADE} = e(IL_U \cdot ADEID, P)^{h(IL_U)}$. It then embeds the timestamp ts_2 to calculate $SA_{PUB} = E_{PK_{PUB}}(AuS_{PUB} || ts_2)$ and $SA_{ADE} = E_{PK_{ADE}}(AuS_{ADE} || ts_2)$. Next, the plugin sends the click message M about the ad to MS. The click message is shown as in equation 6 and as in step ⑦ in Figure 3.

$$M = E_{PK_{PUB}}(U_{Sig} || ts_2) || SA_{PUB} || SA_{ADE}. \quad (11)$$

STEP UCA2. MS forwards $\{ID_{ad} || M || MS\}$ to the PUB who won the bidding and stores $\{ID_{ad} || M || MS || PUBID\}$ in its local database, as shown in step ⑧ in Figure 3.

STEP UCA3. Finally, the data $\{h(ID_{ad} || M || MS || PUBID)\}$ are classified by periods and ID_{ad} s and periodically (e.g., once a day) recorded in the ABB by the MS, as shown in step ⑨ in Figure 3.

5.4. Click Fraud Detection and Prevention. This phase achieves click fraud detection and prevention between entities in an advertising system based on ABB.

5.4.1. PUB Detects and Prevents Click Fraud (PUBD). To prevent MS from forging the data and ensure the transparency of this ad click analysis process, PUB can detect and prevent fraudulent click, and it is shown in Figure 5.

STEP PUBD1. PUB uses its private key SK_{PUB} to decrypt M from MS to obtain the secret U_{Sig} and $\{AuS_{PUB} || ts_2\}$ from SA_{PUB} . The PUB verifies the timeliness of the ts_2 to prevent the replay attacks, as shown in step ① in Figure 5.

STEP PUBD2. PUB uses IMA's public key PK_{IMA} to restore $h(IL_U)$ and MUID from the U_{Sig} and then calculates AuS_{PUB} by (12), as shown in step ② in Figure 5:

$$\begin{aligned} e(IL_{PUB}, MUID)^{h(IL_U)} &= e(IL_{PUB}, UID \cdot P)^{h(IL_U)} = e(IL_{PUB}, P)^{UID \cdot h(IL_U)} \\ &= e(IL_U \cdot PUBID, P)^{h(IL_U)} \\ &= AuS'_{PUB}. \end{aligned} \quad (12)$$

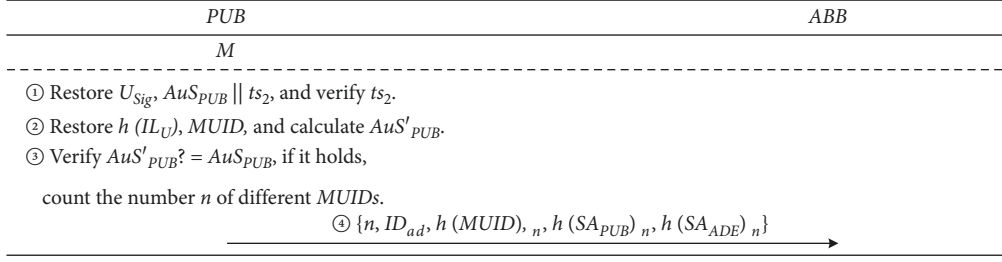


FIGURE 5: Process of PUB verifying the effective clicks.

STEP PUBD3. If (12) holds, PUB counts the number of different MUID s in AuS_{PUB} s , denoted as n , which is the number of valid advertising clicks in a certain period (e.g., one day). This means that in this period, PUB only pays for n clicks to MS. In this way, the PUB can detect all the fraudulent clicks in the message forwarded by MS. Also, the PUB pays nothing to MS for the repeated MUIDs, so the click fraud by a malicious MS can be prevented. Simultaneously, PUB records U 's access behavior information like $\{ID_{ad} || M || MUID || U_{Sig} || SA_{PUB} || ts_2 || SA_{ADE} || MS || fee || behaviour\}$ locally, as shown in step ③ in Figure 5.

STEP PUBD4. Finally, PUB periodically (e.g., once a day) records the result $\{n, ID_{ad}, h(MUID)_n, h(SA_{PUB})_n, h(SA_{ADE})_n\}$ in ABB sorted by periods and ID_{ad} s, as shown in step in ④ in Figure 5.

5.4.2. ADE Detects and Prevents Click Fraud (ADED). Similarly, ADE also verifies the results recorded by PUB in the ABB to detect and prevent click fraud, and this section is shown in Figure 6.

STEP ADED1. ADE communicates with PUB to obtain the original access information $\{ID_{ad} || U_{Sig} || MS || fee || SA_{ADE}\}$ of U in the PUB local database. ADE then uses PK_{PUB} to encrypt the fee and compares it with the data on the ABB to prevent PUB's cheating, as shown in step ① in Figure 6.

STEP ADED2. ADE decrypts SA_{ADE} with private key SK_{ADE} , obtains $\{AuS_{ADE} || ts_2\}$, and verifies the timeliness of ts_2 to prevent replay attacks, as shown in step ② in Figure 6.

STEP ADED3. Similar to *STEP PUBD2*, ADE also restores $h(IL_U)$ and MUID from U_{Sig} , and then calculates AuS'_{ADE} by

$$\begin{aligned}
 e(IL_{ADE}, MUID)^{h(IL_U)} &= e(IL_{ADE}, UID \cdot P)^{h(IL_U)} \\
 &= e(IL_{ADE}, P)^{UID \cdot h(IL_U)} \\
 &= e(IL_U \cdot ADEID, P)^{h(IL_U)} \\
 &= AuS'_{ADE}.
 \end{aligned} \tag{13}$$

If (13) holds, ADE also counts the number n' of different MUIDs in AuS_{ADE} s in a certain period (e.g., one day), as shown in step ③ in Figure 6.

STEP ADED4. ADE reads the data n' recorded in ABB by PUB and compares n' with the n . If the equation $n' = n$ holds, ADE pays fee to PUB according to the n , as shown in step ④ in Figure 6. Therefore, the ADE can detect all the fraudulent clicks in the original access information from PUB. Also, the ADE pays nothing to PUB for the repeated MUIDs, so the fraudulent click by a malicious PUB can be prevented.

5.4.3. MS Detects and Prevents Click Fraud (MSD). MS obtains all the $\{AuS_{PUB} || ts_2\}$ from PUB and uses PK_{PUB} to encrypt them successively to get the encrypted result $SA'_{PUB} = E_{PK_{PUB}}(AuS_{PUB} || ts_2)$. Then, MS compares the SA'_{PUB} with the SA_{PUB} in M from MS's local database one by one; if it holds, the data from the PUB are valid. Finally, MS counts the number n'' of the different AuS_{PUB} and verifies if $n'' = n$ holds. If it holds, MS charges PUB fees according to the n'' . As a result, the MS can detect all the fraudulent clicks in the data from PUB. Also, MS cannot charge more PUB for the repeated AuS_{PUB} s, so the fraudulent click by a malicious MS can be prevented.

6. Security Analysis

In this section, we first analyze the security of our scheme from three levels: the processing level, the data level, and the infrastructure level, which can be called PDI model-based security [49–52]. Then, we give the informal analysis of security under the security assumptions in Section 4.2. Lastly, we demonstrate that the BCFDPS scheme is provably secure.

6.1. PDI Model-Based Security Analysis. As the one of the latest and most mature blockchain security analysis frameworks for Industry 4.0, the PDI model [49] conducts a comprehensive and detailed analysis of security issues. In the PDI model, the blockchain security is divided into three levels, which are the process level, the data level, and the infrastructure level [51]. Similarly, we also analyze the security of our blockchain-based click fraud detection and prevention scheme according to the three aspects.

6.1.1. The Process Security

- (1) *Off-blockchain data processing security:* a large number of data processing operations are run off-

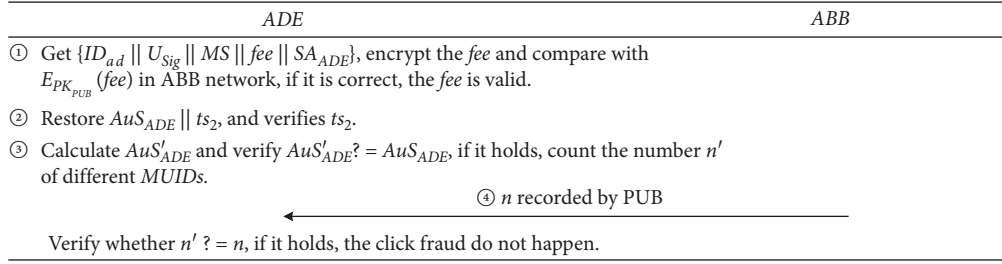


FIGURE 6: Process of ADE verifying the effective clicks.

blockchain in our scheme, since the data statistical analysis ability of the existing blockchain applications is weak [50]. In our scheme, a U 's masked identity MUID and his ad click message M are encrypted (denoted as mm) and sent to the MS by U 's browser plugin locally. Then, mm is forwarded to a PUB by a MS off-blockchain. Next, the MS, PUB, and ADE can independently count the real click number from mm with ECC and bilinear pairing algorithms. Also, since mm is ciphertext and being processed off-blockchain, it is difficult for an attacker to gather, crack, and modify it. That is, the data processing security off-blockchain is guaranteed in these entities.

- (2) *Data processing security in the blockchain*: to implement our scheme in a real-time online advertising scenario, the data processing in the blockchain of our scheme is to periodically read and write content in the access behavior blockchain (ABB) through smart contracts. The ABB is a consortium blockchain that only allows authorized MSs and PUBs to write data, which avoids the unauthorized access. Also, the consensus protocol in ABB guarantees the correctness and consistency of the data when it is written to the ABB, largely eliminating exceptions in data processing and ensuring the security of data processing in the blockchain.

6.1.2. The Data Security

- (1) *Data tamper-proof*: in our scheme, all original business data are stored in the local servers of MS and PUB, and the aggregated results of the original data are regularly recorded in the consortium blockchain as the form of hash values. In this way, even if attackers obtain the data in the blockchain, they cannot get the original data in the local servers of MS and PUB, so they cannot view or tamper with the original data. On the other hand, blockchain can ensure data consistency in distributed ledgers. Therefore, business data security is achieved whether the data are in the blockchain or not.
- (2) *Consumer's identity privacy*: similar to the digital twin in [53–55], a U in our scheme can only obtain his unique digital identity to visit MS's websites and click on PUB's ads. Also, a masked identity MUID,

CP-ABE algorithm, and ECC algorithm are utilized by the U to hide his identity, while preserving the ad precision targeting. In addition, nobody except the PUB can mark the U , and no one can reveal U 's real digital identity UID. That is, our scheme protects the privacy of consumer's identity.

6.1.3. The Infrastructure Security

- (1) *System structure security*: the two-level mutual verification between MS, PUB, and ADE maintains the stability of our system structure. For one thing, PUB counts the real and effective clicks from a large number of users' ad click messages M s which are forwarded by the MS. Once the M s are tampered with or forged by the MS, they cannot pass the verification of PUB. At the same time, the MS can use the ECC algorithm to count the real clicks from the data stored in local database to prevent PUB from forging the amount of the clicks. For the other thing, since the raw data are generated by U , the ADE can find anomalies once the PUB adds entries in the raw data. Thus, our scheme has system structure security.
- (2) *Cryptographic facilities security*: we use the standard cryptographic facilities to build our system. Specifically, CP-ABE algorithm, bilinear pairing algorithm, and ECC algorithm are used by a U to protect his identity. The bilinear pairing algorithm and ECC algorithm are adopted by a PUB and an ADE to detect the fraudulent click, while a MS utilizes the ECC algorithm to detect a click fraud. The security of our scheme relies on these standard cryptographic facilities and we assume that the standard cryptographic facilities used in our scheme are secure and unbreakable.

6.2. Informal Analysis of Security. In this section, we analyze the security of our scheme under the security assumptions in Section 4.2 in an informal way.

6.2.1. Prevention of a False MUID. In Section 5.2.1, a machine cannot obtain a valid MUID since it has no way to pass the IMA biometric authentication. Even if it forges a false MUID, it still cannot generate a valid $U_{Sig} = \text{Sig}_{\text{SK}_{\text{IMA}}}(h(\text{IL}_U) || \text{MUID})$ without IMA's private key

SK_{IMA} . That is, the click message, containing an invalid MUID, generated by the machine in phase 5.3.2 will be discarded. Therefore, in our scheme, the number of false MUIDs is not included in the number of valid clicks.

6.2.2. Transparency of Clicks between Entities. PUB decrypts the SA_{PUB} in M to get AuS_{PUB} using SK_{PUB} , then verifies the AuS_{PUB} , and counts the number n of different AuS_{PUB} . Similarly, ADE restores the AuS_{ADE} in SA_{ADE} from PUB's local database using SK_{ADE} to verify the authenticity of AuS_{ADE} , and then ADE counts the number n' of different AuS_{ADE} . Although SA_{ADE} of M comes from PUB, AuS_{ADE} in SA_{ADE} is encrypted by PK_{ADE} , and only SK_{ADE} can decrypt it. Therefore, PUB cannot tamper with SA_{ADE} ; furthermore, ADE ensures the validity of n' . MS encrypts the original data $\{AuS_{PUB} || ts_2\}$ from PUB using PK_{PUB} and compares the encrypted result $SA'_{PUB} = E_{PK_{PUB}}(AuS_{PUB} || ts_2)$ with SA_{PUB} in M from MS's local database to verify whether the PUB is honest. In this way, PUB, ADE, and MS can verify the number of clicks about the same ad in an independent way.

6.2.3. U 's Conditional Unlinkability. First of all, U sends his masked identity in ciphertext CT to MS and MS broadcasts it to PUBs in phase 5.3.1. Then, only PUBs can decrypt U 's U_{Sig} from CT since CT is calculated using the CP-ABE algorithm and only attributes S owned by PUBs can generate a decryption key SK. Secondly, U sends his click message M to MS and MS forwards it to PUB in phase 5.3.2. Next, only PUB can reveal U 's masked identity MUID from M using its private key for advertising precision marketing. In the entire communication of U , neither the attacker in the channel nor the MS can directly link U 's masked identity MUID because both CT and M are encrypted by CP-ABE algorithm or asymmetric cryptographic algorithm and only PUBs can decrypt them. However, even PUB cannot link U 's masked identity MUID to U 's real identity UID in ABB since PUB does not have the right to write and read in EIB. Thence, the scheme achieves U 's conditional unlinkability.

6.2.4. Data Security and Integrity. Firstly, in this scheme, all the commercial contract data, e.g., fee, are encrypted and only the data owner PUB and MS can decrypt these ciphertexts. In addition, all the commercial contract data and the hash value of click result are recorded in the ABB (a consortium blockchain) which is shown in steps ⑩, ⑬, and ⑭ and any adversary cannot tamper with these data in the consortium blockchain.

6.2.5. Resistance to Replay Attacks. In phases 5.3.1 and 5.3.2, the timestamp ts_1 and ts_2 are included in the message CT and M , and PUB first checks their timeliness to avoid replay attacks. Further, in phases 5.4.2 and 5.4.3, ADE and MS can avoid replay attacks by the timestamp ts_2 . Consequently, our scheme is resistant to replay attacks in a great probability.

6.2.6. Resistance to Forgery. For one thing, in phase 5.3.1, MS stores U 's CT while it has no ability to construct U 's click message M in phase 5.3.2 without a IL_U . For another thing, in phase 5.4.2, PUB records SA_{PUB} and SA_{ADE} , but it cannot forge a SA_{ADE} because it also does not have a IL_U . In other words, the click message M containing MUID can only be generated by U . That is, the BCFDPS can resist forgery attacks.

6.3. Provable Security. The proposed scheme is based on bilinear pairing cryptosystem on elliptic curves (denoted as BPCEC), ciphertext-policy attribute-based encryption (denoted as CP-ABE), and elliptic curve cryptography (denoted as ECC). According to the security characteristics of each module, we show that our scheme meets click fraud detection and prevention and U 's conditional unlinkability.

6.3.1. Theorem 1. If the BPCEC, CP-ABE, and ECC algorithms satisfy the basic security properties, then the scheme in this paper can detect and prevent click fraud.

Proof. Define A_{BPCEC} as an adversary who attacks the security of BPCEC algorithm, A_{CP-ABE} as an adversary attacking the security of CP-ABE algorithm, and A_{ECC} as an opponent attacking the security of ECC algorithm. Assuming A_{CF} clicks fraud successfully, a polynomial time algorithm $A_\theta \in (A_{BPCEC}, A_{CP-ABE}, A_{ECC})$ is defined, which has the ability to attack the algorithms of BPCEC, CP-ABE, and ECC. Through the query of A_{CF} and the A_θ 's interaction in the click fraud game, A_θ is optimized repeatedly to successfully attack the BPCEC, CP-ABE, and ECC algorithms. That is, if the adversary A_{CF} clicks fraud successfully in the scheme, it means A_θ successfully attacks the security of algorithms of BPCEC, CP-ABE, and ECC with a certain probability.

According to the steps defined above, here are the interactions between algorithm A_θ and the adversary A_{CF} :

STEP 1. Registration phase: through the identity generated in U 's registration phase, algorithm A_θ obtains U 's digital identity and receives U 's identity UID, U 's masked identity MUID, U 's identity license IL_U , and IMA's signature U_{Sig} . At last, A_θ sends $\{UID, MUID, IL_U, U_{Sig}\}$ to A_{CF} .

STEP 2. Inquiry phase: the adversary A_{CF} can query the algorithm A_θ for polynomial time:

- (1) *Generate the ciphertext CT:* A_{CF} visits MS's website, generates the ciphertext CT by the CP-ABE algorithm, and sends CT to MS.
- (2) *Generate the click message M :* MA_{CF} generates the click message M which contains a $b \in \{0, 1\}$ randomly selected by A_{CF} through BPCEC and ECC algorithm and A_{CF} then clicks PUB's ad to send M .

STEP 3. Verification phase: PUB verifies U 's click message M and outputs b' using the ECC and BPCEC

algorithms. If $b' = b$ exists, it indicates that the adversary A_{CF} successfully carried out the click fraud attack. The success probability of the adversary A_{CF} is

$$\begin{aligned}
\text{Adv}_{A_{CF}}(k) &= \Pr[\text{Exp}_{A_{CF}}(k) = 1] \\
&= \Pr[A_{CF}(\text{verify}) = 1|b = 1] \cdot \Pr[b = 1] + \Pr[A_{CF}(\text{verify}) = 0|b = 0] \cdot \Pr[b = 0] \\
&= \frac{1}{2} \Pr \left[\begin{array}{l} A_{BPCEC}(\text{verify}) = 1, \\ A_{CP-ABE}(\text{verify}) = 1, |b = 1 \\ A_{ECC}(\text{verify}) = 1, \end{array} \right] + \frac{1}{2} \Pr \left[\begin{array}{l} A_{BPCEC}(\text{verify}) = 0, \\ A_{CP-ABE}(\text{verify}) = 0, |b = 0 \\ A_{ECC}(\text{verify}) = 0, \end{array} \right] \\
&< \frac{1}{2} (\Pr[A_{BPCEC}(\text{verify}) = 1|b = 1] + \Pr[A_{CP-ABE}(\text{verify}) = 1|b = 1] + \Pr[A_{ECC}(\text{verify}) = 1|b = 1]) \\
&+ \frac{1}{2} (\Pr[A_{BPCEC}(\text{verify}) = 0|b = 0] + \Pr[A_{CP-ABE}(\text{verify}) = 0|b = 0] + \Pr[A_{ECC}(\text{verify}) = 0|b = 0]) \\
&= \Pr[\text{Exp}_{A_{BPCEC}}(k) = 1] + \Pr[\text{Exp}_{A_{CP-ABE}}(k) = 1] + \Pr[\text{Exp}_{A_{ECC}}(k) = 1] \\
&= \text{Adv}_{A_{BPCEC}}(k) + \text{Adv}_{A_{CP-ABE}}(k) + \text{Adv}_{A_{ECC}}(k).
\end{aligned} \tag{14}$$

If an attacker A_{BPCEC} successfully attacks the BPCEC algorithm, an attacker A_{CP-ABE} successfully attacks the CP-ABE algorithm, and an attacker A_{ECC} can successfully attack ECC algorithm, A_{CF} can carry out the click fraud attack successfully. However, the probability of A_{BPCEC} , A_{CP-ABE} , and A_{ECC} successfully attacking the BPCEC, CP-ABE, and ECC algorithms is almost $1/n$, respectively; then, A_{CF} wins in the click fraud attack game of BCFDPS scheme with a probability of $3/n$. But, according to the assumptions that BPCEC, CP-ABE, and ECC algorithms satisfy the basic security properties, it is concluded that the probability of A_{CF} successfully attacking can be ignored, so the scheme can detect and prevent click fraud. \square

6.3.2. Theorem 2. If all the crypto-algorithms such as BPCEC, CP-ABE, and ECC satisfy the basic security

features, then U 's conditional unlinkability can be achieved in the BCFDPS.

Proof. Define A_{BPCEC} as an adversary who attacks the linkability of MUID of BPCEC algorithm, A_{CP-ABE} as an adversary attacking the linkability of U_{Sig} of CP-ABE algorithm, and A_{ECC} as an opponent attacking the linkability of U_{Sig} of ECC algorithm. Assuming A_{CP} (except PUB) links U 's masked identity MUID successfully, a polynomial time algorithm $A_{\tau} \in (A_{BPCEC}, A_{CP-ABE}, A_{ECC})$ is defined, which has the ability to attack the algorithms of BPCEC, CP-ABE, and ECC. During the communication process of U , MS, and PUB, two messages CT and M are encrypted by the algorithms BPCEC, CP-ABE, and ECC. Therefore, for the adversary A_{CP} , the probability of successfully linking many different messages to the same U is

$$\begin{aligned}
\text{Adv}_{A_{CP}}(k) &= \Pr[\text{Exp}_{A_{CP}}(k) = 1] \\
&= \Pr[\text{Ver}(CT) = 1] \cdot \Pr[\text{Ver}(U_{\text{Sig}}) = 1] \cdot \Pr[\text{Ver}(AuS_{\text{PUB}}) = 1] \\
&= \Pr[\text{Exp}_{A_{CP-ABE}}(k) = 1] \cdot \Pr[\text{Exp}_{A_{ECC}}(k) = 1] \cdot \Pr[\text{Exp}_{A_{BPCEC}}(k) = 1] \\
&= \text{Adv}_{A_{CP-ABE}}(k) \cdot \text{Adv}_{A_{ECC}}(k) \cdot \text{Adv}_{A_{BPCEC}}(k).
\end{aligned} \tag{15}$$

Therefore, if the attacker A_{BPCEC} successfully attacks the BPCEC algorithm, the attacker A_{CP-ABE} successfully attacks the CP-ABE algorithm, and the attacker A_{ECC} successfully

attacks the ECC algorithm, then A_{CP} wins in the conditional unlinkability simulation attack game. However, according to the assumptions about these security features, the

probability of A_{CP} successfully attacking can be ignored. As a result, the scheme accomplishes U 's conditional unlinkability. \square

7. Implementation and Evaluation

We evaluate our scheme in terms of computation, communication, storage, and Ethereum gas cost based on JPBC library [56] and Ethereum.

In the proposed scheme, four phases of initialization, registration, ad publishing, and click fraud detection and prevention are involved. Because the first two phases happen rarely, they are not implemented in this section and we mainly focus on the phases of ad publishing and click fraud detection and prevention in which an ad is published and the click fraud is detected and prevented.

7.1. Computation Cost

7.1.1. Evaluation of Our Scheme. We mainly focus on the phases of the ad publishing and click fraud detection and prevention in this section. We execute evaluation tests to get the time cost of meta-operations and the evaluation test is based on a PC (Intel Core i5-9400F CPU @ 2.90 GHz, 16 GB RAM @ 2667 MHz and Windows 10 \times 64). We use JDK 1.8, JPBC library [56], to support efficient bilinear pairing operations.

To achieve persuasive expression of computation comparison, the symbols and parameters are introduced: T_{C_Enc} is the encryption algorithm in CP-ABE scheme, T_{C_KG} denotes the key generation algorithm in CP-ABE scheme, T_{C_Dec} means the decryption algorithm in CP-ABE scheme, T_{E_Enc} expresses the encryption algorithm in ECC, T_{E_Dec} signifies the decryption algorithm in ECC, and T_{bp} represents the bilinear pairing operation. Their time cost is as follows: $T_{C_Enc} = 146.41$ ms, $T_{C_KG} = 118.80$ ms, $T_{C_Dec} = 33.12$ ms, $T_{E_Enc} = 3.17$ ms, $T_{E_Dec} = 0.36$ ms, and $T_{bp} = 6.79$ ms. In addition, the time cost of hash function and concatenate operation is small, and we do not take this into account in computation cost. The detailed computation costs for each phase are illustrated in Table 2.

In phase 5.3.1, U is required to perform one encryption algorithm in CP-ABE scheme and PUB needs to execute one key generation algorithm in CP-ABE scheme, one decryption algorithm in CP-ABE scheme, and one decryption algorithm in ECC, that is, the running time is $T_{C_Enc} + T_{C_KG} + T_{C_Dec} + T_{E_Dec} = 298.69$ ms. According to Ma et al. [57], the response speed of publishing an ad in our scheme is in the acceptable threshold (150 ~ 600 ms) and is lower than the one in [6] which closes to 400 ms. In phase 5.3.2, U computes three encryption algorithms in ECC and two bilinear pairing operations. Therefore, the execution time to generate a click message is $3T_{E_Enc} + 2T_{bp} = 23.09$ ms, and it has no effect on the user experience. Further, in phase 5.4.1, PUB is required to run three decryption algorithms in ECC and one bilinear pairing operation, that is $3T_{E_Dec} + T_{bp} = 7.87$ ms. In summary, the computation cost from publishing an ad for U (phase 5.3.1)

to verifying the effective clicks by PUB (phase 5.4.1) is $298.69 + 23.09 + 7.87 = 329.65$ ms, where the time cost of one click fraud detection and prevention is only 7.87 ms. After PUB counts the effective clicks, ADE and MS will also verify the clicks to ensure their profit. Similar to PUB, ADE performs three decryption algorithms in ECC and one bilinear pairing operation, that is, $3T_{E_Dec} + T_{bp} = 7.87$ ms. For the MS, it executes one encryption algorithm in ECC to detect a click fraud, which is $T_{E_Enc} = 3.17$ ms. From Table 2, it can be seen that the CP-ABE algorithm increases the run time in phase 5.3.1, but it protects U 's privacy from MS and the sniffer of a channel. In addition, it should be noted that the computation overhead of PUB in phases 5.3.1 and 5.4.1 can be improved at the publisher with powerful computing clusters. Moreover, distributed and parallel optimization techniques for verifiable computations can also be adopted to further enhance publisher's performance in publishing the ad to a U who is the potential consumer of the ad.

On the other hand, blockchain is introduced in our scheme; in order to demonstrate the practical performance of our blockchain-based scheme, we evaluate the execution cost of our smart contract based on a public Ethereum testnet (Rinkeby). We used Chrome v89.0 explorer with the plugin MetaMask and Remix which is a browser-based IDE to connect the contract between Ethereum and the program simulated. Rinkeby testnet was started by the Ethereum team in April 2017 and it uses Clique PoA (Proof of Authority) consensus protocol. Importantly, it is immune to spam attacks, as Ether supply is controlled by several trusted parties and only they can write transactions in the blockchain, which makes it like a consortium blockchain; thence, the waiting time for transaction confirmation is relatively short to be ignored.

We deploy smart contracts on Rinkeby to record the transaction data and count the gas cost of smart contracts on deployment and recall. The gas cost of our scheme is shown in Table 2. In our scheme, a smart contract is only deployed once in phase 5.3.1 and the gas cost of deploying the contract is 89,003. Additionally, in phases 5.3.1, 5.3.2, and 5.4.1, the cost of recalling the contract to write 256, 32, and 128 bytes of analysis result on blockchain is 27,054, 23,470, and 25,006 gas, respectively. All in all, judging from the evaluation results, our scheme is feasible in practice.

7.1.2. Comparison of the Computation Cost in Click Fraud Detection and Prevention Process. As far as we know, the click fraud detection and prevention schemes that use blockchain are hardly found. Therefore, we choose the click fraud detection schemes [8, 16, 18, 29, 31] which do not use blockchain and compare the computation costs with them in publisher's click fraud detection and prevention process (phase 5.4.1), and the comparison result is shown in Table 3.

It can be seen from Table 3 that Almahmoud et al. [8] utilized SVM, KNN, etc. to detect a fraudulent click by machines, and the time taken to build the model of the generated 500 instances is 10 ms, while the time taken to classify a single instance whether legitimate or illegitimate is 50 ms with a precision of 95.10%. The scheme in [16] uses

TABLE 2: Computation cost of our scheme in ad publishing and click fraud detection and prevention.

| Phases | Time (ms) | | | | Gas on contracts | | | |
|----------------|---------------------------------|---------------------|--|-------------------------------|------------------|--------|--------|---------|
| | U | MS | PUB | ADE | Total | Deploy | Call | Total |
| 5.3.1: PPA | $T_{C_Enc} = 146.41$ | 0 | $T_{C_KG} + T_{C_Dec} + T_{E_Dec} = 152.28$ | 0 | 298.69 | 89,003 | 27,054 | 116,057 |
| 5.3.2: UCA | $3T_{E_Enc} + 2T_{bp} = 23.09$ | 0 | 0 | 0 | 23.09 | 0 | 23,470 | 23,470 |
| 5.4.1: PUBD | 0 | 0 | $3T_{E_Dec} + T_{bp} = 7.87$ | 0 | 7.87 | 0 | 25,006 | 25,006 |
| 5.4.2: ADED | 0 | 0 | 0 | $3T_{E_Dec} + T_{bp} = 7.87$ | 7.87 | 0 | 0 | 0 |
| 5.4.3: MSD | 0 | $T_{E_Enc} = 3.17$ | 0 | 0 | 3.17 | 0 | 0 | 0 |

TABLE 3: Comparison of computation cost in click fraud detection and prevention.

| Schemes | Methods | Precision (%) | Preparation time | Verification time (ms) |
|---------|--|---------------|---------------------------|------------------------|
| [8] | Machine learning (SVM, KNN, etc.) | 95.10 | 10 ms | 50 |
| [16] | Machine learning (RNN) | 33.80 | 12 h (roughly 6–8 epochs) | — |
| [18] | Machine learning (bagging and boosting) | 96.29 | ≈ 800 s | — |
| [29] | Statistical analysis (UI transition graphs) | ≈ 93 | 216.7 s | 400 |
| [31] | Statistical analysis (traffic matrix analysis) | 89.34 | — | — |
| Ours | Blockchain (identity authentication) | 100 | 0 | 7.87 |

Method refers to the algorithms or ideas used in these schemes. Precision indicates the credibility of a click traffic detection result. Preparation time denotes the time cost to train a model or analyze a pattern of a click fraud. Verification time describes the time cost to detect a click fraud using the model or pattern.

recurrent neural network to train a model with more than 1.6 million sessions so that the typical training duration is 12 hours (roughly 6–8 epochs), but the precision is 33.80%. The dataset of the scheme in [18] contains 393,708 deliveries (243,650 ok deliveries and 150,058 fraud deliveries), and the time required to train classifier with 10 features is about 800 seconds with a precision rate of 96.29%. Dong et al. [29] utilized 12,000 ad-supported apps, and an average of 216.7 seconds was spent to construct the UI transition graphs and an average of 400 ms was spent to detect the ad frauds. The dataset in [31] is from a university campus network between June 2015 and November 2017 with total of 217,334,190 unique clicks. After training, the precision is 89.34%. Table 3 shows that the preparation times of schemes in [16, 18, 29] are longer than ours because their schemes are based on machine learning and statistical analysis, and they need to spend more time training machine models and analyzing the pattern of the click traffic, while the preparation time is not included in our scheme. The verification time of a click fraud in the schemes in [16, 18, 31] is not explained, but in the scheme in [29], it is 400 ms, which is obviously higher than ours. In summary, our scheme is the best one for publishers to detect and prevent a click fraud.

7.2. Communication and Storage Cost

7.2.1. Evaluation of Our Scheme. Our scheme is embedded in the advertising system and many entities in the system need to send data to publish an ad and store data as evidences to pay for fees. To evaluate the feasibility of our scheme in practice, we simulate the scheme in terms of ad

publishing and click fraud detection and prevention, and the results of communication and storage cost are shown in Table 4. Specifically, we assume that the output size of the general hash function (h) is 256 bits, the size of an elements in the elliptic curve is 256 bits, the size of an element in a bilinear group is 1,024 bits, the length of identities is 256 bits, and the timestamp size is 112 bits.

In phase 5.3.1, an ADE first sends an ad's identity ID_{ad} to a PUB, then a U transmits a ciphertext CT containing his own MUID to a MS, the MS further forwards the ciphertext CT to the PUB, and after the PUB decrypts and obtains the MUID, the PUB sends the ID_{ad} and bidding fee to the MS; next, the MS displays the ad frame for the U. The communication cost of U, MS, PUB, and ADE is $CT = 1,508$, $CT + ad$ frame = 2,508, $ID_{ad} + fee = 33$, and $ID_{ad} = 32$ bytes. Also, U stores 259-byte parameters $\{T, U_{Sig}, MUID, IL_U\}$ to compute ad click messages M faster. To make it easier to publish the ad, the MS stores the $\{ID_{ad}, fee, PUBID\}$ that are 65 bytes, the PUB reserves $\{CT, MUID, ID_{ad}, fee, MS\}$, which are 1,637 bytes, and the ADE keeps his 32 bytes $\{ID_{ad}\}$. Similarly, in phase 5.3.2, the contents of the communication of U, MS, PUB and ADE are $M = 484$ bytes, $ID_{ad} + M + MS = 548$ bytes, 0, and 0, respectively. The storage cost of them is 0, $\{M\} = 484$ bytes, $\{M\} = 484$ bytes, and 0 separately. The click fraud is detected and prevented by the MS, PUB, and ADE in an independent way in phases 5.4.1, 5.4.2, and 5.4.3, and the processed results are also stored. Specifically, the PUB writes a total of $n + ID_{ad} + Info + ts = 143$ bytes of data in the ABB and it consumes 175 bytes to store $\{n, ID_{ad}, MS, Info, ts\}$. The ADE receives $SA_{ADE} + ts + U_{Sig} = 255$ bytes to verify the click messages, and the ADE stores $\{n, ID_{ad}, PUBID, ts\} = 79$

TABLE 4: Communication and storage cost of our scheme in ad publishing and click fraud detection and prevention.

| Phases | Communication cost (bytes) | | | | | Storage cost (bytes) | | | | |
|-------------|----------------------------|-------|-----|-----|-------|----------------------|-----|-------|-----|-------|
| | U | MS | PUB | ADE | Total | U | MS | PUB | ADE | Total |
| 5.3.1: PPA | 1,508 | 2,508 | 33 | 32 | 4,081 | 259 | 65 | 1,637 | 32 | 1,993 |
| 5.3.2: UCA | 484 | 548 | 0 | 0 | 1,032 | 0 | 484 | 484 | 0 | 968 |
| 5.4.1: PUBD | 0 | 0 | 143 | 0 | 143 | 0 | 0 | 175 | 0 | 175 |
| 5.4.2: ADED | 0 | 0 | 0 | 255 | 255 | 0 | 0 | 0 | 79 | 79 |
| 5.4.3: MSD | 0 | 142 | 0 | 0 | 142 | 0 | 207 | 0 | 0 | 207 |

bytes of data. Moreover, $AUS_{PUB} + ts = 142$ bytes of message are obtained by the MS to detect the click fraud, and it stores $\{n, ID_{ad}, PUBID, AUS_{PUB}, ts\} = 207$ bytes of result.

For the data presented in Table 4, the communication and storage cost in our scheme is mainly consumed in phases 5.3.1 and 5.3.2. A total of about 8,000 bytes are used, which is negligible in today's common online advertising systems.

7.2.2. Comparison of the Communication Cost in Publishing and Clicking an Ad. We did our best to search for current blockchain-based online advertising click fraud detection and prevention schemes but only found two blockchain-based online advertising schemes [5, 6] which do not realize the detection and prevention of click fraud. Additionally, Ding et al. [6] were mainly concerned about the throughput of the blockchain transactions, and they did not give details of sending the advertising messages. Therefore, from the perspective of scheme similarity, we only make a comparison in the processes of "Publisher publishes an ad (phase 5.3.1)" and " U clicks the ad (phase 5.3.2)" with a vehicular local advertising system of Liu et al. [5]. Table 5 visually describes the communication cost in the processes of publishing an ad and clicking an ad.

In the scheme of Liu et al. [5], a PUB directly sends an ad to a U , and the U then clicks on the ad. ADE and MS are not included in the process of publishing and clicking on the ad, so the cost of ADE and MS is 0. To obtain an ad in the scheme of Liu et al. [5], a U needs to send his local position of $2 \times 8 = 16$ bytes, five attributes of $5 \times 10 = 50$ bytes, and a number of 1 byte to a PUB, in which the communication of a U is $16 + 50 + 1 = 67$ bytes. Also, the PUB returns two positions of $2 \times 8 = 16$ bytes and forty attributes of $40 \times 10 = 400$ bytes to the U , in which the communication of a PUB is $2 \times 16 + 400 = 432$ bytes. However, in their scheme, the click fraud still exists since they did not verify the authenticity of the click. Also, the privacy of U 's locations and interests is leaked to the sniffer in the channel because the communication data are in plaintext. In our scheme, we are able to detect and prevent click fraud while protecting the identity privacy of the U . Specifically, an ADE first sends an ad ID_{ad} to a PUB, a U sends a ciphertext CT to the MS, and the MS forwards the CT to the PUB for getting an ad. Then, the PUB sends a ID_{ad} and a price fee to the MS, and the MS displays the ad to the U . Next, the U clicks on an ad and sends a click message M to the MS, and the M is forwarded to the PUB. In these steps, the U 's communication cost includes a CT and a M , which is $1,508 + 484 = 1,992$ bytes, the

TABLE 5: Comparison of communication cost (bytes) in publishing and clicking an ad.

| | [5] | Ours |
|------------------|-----|-----------------------|
| Consumer (U) | 67 | $1,508 + 484 = 1,992$ |
| Media site (MS) | 0 | $2,508 + 548 = 3,056$ |
| Publisher (PUB) | 432 | 33 |
| Advertiser (ADE) | 0 | 32 |
| Total | 499 | 5,113 |

communication cost of the MS contains a CT , an ad frame, a ID_{ad} , an identity MS, and a M , which is $1,508 + 1,000 + 32 + 32 + 484 = 3,056$ bytes, the PUB's communication cost consists of a ID_{ad} and a fee, which is $32 + 1 = 33$ bytes, and the ADE only sends 32 bytes of ID_{ad} . The communication cost of ours is higher than that of Liu et al. [5] since we add some additional authenticity information in the click message to detect and prevent click fraud. Moreover, the communication data are encrypted by the CP-ABE algorithm to protect U 's privacy from the transmission medium.

When we place our scheme and Liu et al.'s scheme [5] with the same level of U 's privacy protection and without regarding to click fraud detection and prevention, the ad publishing steps in our scheme can be modified as follows: a U needs to send UID to the MS, then the MS forwards UID to the PUB, next, the PUB sends ID_{ad} and fee to the MS, and finally, the MS displays ID_{ad} for the U . As a result, during these steps, the total communication content within the system is $\{UID, UID, ID_{ad}, fee, ID_{ad}\} = 32 + 32 + 32 + 1 + 32 = 129$ bytes, which is significantly lower than 499 bytes of Liu et al.'s scheme. That is, we add $5,113 - 129 = 4,984$ bytes of communication overhead for U 's privacy protection and click fraud detection and prevention. Also, the overhead (4,984 bytes) added to our scheme is acceptable in the background that the mainstream network bandwidth is above 3 MB/s (the average bandwidth of a 4G network is 3 MB/s).

7.2.3. Comparison of the Storage Cost in Publishing and Clicking an Ad. Besides, the comparison of storage cost when a publisher publishes an ad and a consumer clicks the ad is also shown in Figure 7.

In the processes of publishing and clicking an ad, Liu et al.'s scheme [5] does not involve the advertiser and the media site, that is, the storage cost of them is 0. Also, to request an ad faster, the U stores his ad query in advance, in which the storage cost of U is 67 bytes. After publishing an ad to the U , the PUB records the result of the ad query, and according to the experimental

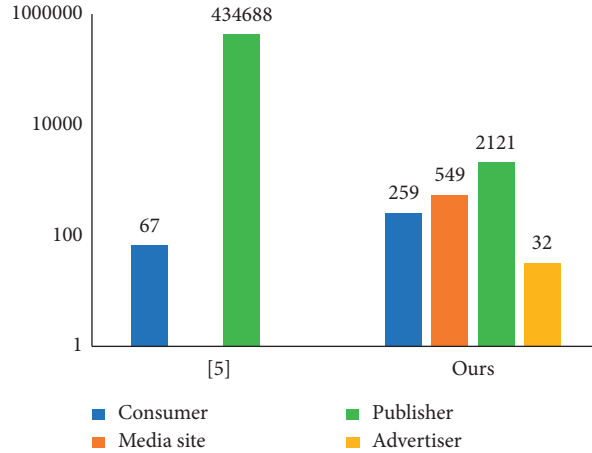


FIGURE 7: Comparison of storage cost (bytes) when a publisher publishes an ad and a consumer clicks the ad.

result, the total length is $12 \times 8 \times 128 + 3 \times 22 \times 50 \times 128 = 434,688$ bytes. For our scheme, the access tree (T), the signature from IMA (U_{Sig}), the masked identity (MUID), and the identity license (IL_U) of U are stored in U 's browser plugin in advance, which is a total of $60 + 71 + 64 + 64 = 259$ bytes. The MS is responsible for forwarding messages and retaining the forwarding results $\{ID_{a_d}, \text{fee}, \text{PUBID}, M\}$, so its storage cost is $32 + 1 + 32 + 484 = 549$ bytes. Additionally, the PUB stores $\{CT, \text{MUID}, ID_{a_d}, \text{fee}, MS, M\}$ forwarded by the MS, which is a total of $1,508 + 64 + 32 + 1 + 32 + 484 = 2,121$ bytes. Also, the ADE only stores 32 bytes of ad information $\{ID_{a_d}\}$. In a word, the total storage cost of our scheme is significantly lower than that of Liu et al. [5] because they need to store all the similarity results between multiple ads and one consumer.

8. Discussion

Our scheme addresses the challenging problems encountered in online advertising click fraud detection and prevention, namely, incompletely reliable detection results, tampering with the number of real clicks by the PUB itself (the PUB can count the real click number), and leakage of consumer's identity privacy. However, it still has some shortcomings that need to be solved.

First of all, although an entity identity blockchain (EIB) exists in our scheme, fraudulent adversaries have not been held accountable in our current scheme. Specifically, the EIB is designed as a consortium blockchain that records the digital identity hash of entities which can serve as evidence to hold malicious entities accountable when a fraudulent click fraud occurs. To restrain the malicious entities, an accountability system needs to be designed in the future.

Secondly, the time spent by MS to detect and prevent click fraud is slightly higher. In detail, when a MS detects click fraud, it needs to use the PK_{PUB} to encrypt the

$\{AuS_{\text{PUB}} \| ts_2\}$ successively and then compare the encrypted result with the SA_{PUB} in its local database one by one. As a result, to reach an agreement with PUB on ad billing fees, the time cost for MS to detect real clicks may be high in a certain period. Therefore, our future research will focus on reducing the time cost of MS in its detection process.

Lastly, the problem of consumers' partial data loss may still exist. In our scheme, we assume that the parameters obtained by registration such as the user's identity license IL_U are secretly stored in his browser plugin, so how to prevent the leakage of parameters from the plugin also needs to be further studied.

9. Conclusion

In this paper, we proposed a blockchain-based click fraud detection and prevention scheme (BCFDPS) for online advertising to avoid clicking by machines and increases the cost of fraud ones by a human. Specifically, a click fraud by a malicious machine is significantly avoided since a consumer's immutable digital identity is embedded in the click message with the bilinear pairing algorithm and the machine does not have a digital identity to generate a valid click message. Also, the cost of click fraud by a human increases because many valid clicks by the same recruited person can only be counted once. Additionally, the introduced consortium blockchain maintains all the hash values of analysis result of consumers' click messages to achieve the transparency of the click fraud detection and prevention process for each entity in the advertising system. Further, the identity privacy of consumers is protected from media sites, advertisers, and the sniffers in the channel by ciphertext-policy attribute-based encryption. Our implementation and evaluation demonstrate the advantages of BCFDPS in computation and storage cost, and the Ethereum gas cost

is limited. Additionally, to protect the user's identity privacy, the communication cost is moderately increased.

Data Availability

Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Qiuyun Lyu was responsible for conceptualization, methodology, analysis, and writing. Hao Li was responsible for conceptualization, methodology, software, and writing. Renjie Zhou was responsible for revision and funding acquisition. Jilin Zhang was responsible for methodology and validation. Nailiang Zhao was responsible for validation and analysis. Yan Liu was responsible for review and editing.

Acknowledgments

This study was partially supported by the Zhejiang Provincial Key Technology Research and Development Program under grant no. 2019C03134 and National Key Technology Research and Development Program of China under grant no. 2019YFB2102100.

References

- [1] M. Gabryel, "Data analysis algorithm for click fraud recognition," in *Proceedings of the International Conference on Information and Software Technologies*, pp. 437–446, Springer, Vilnius, Lithuania, October 2018.
- [2] J. Estrada-Jiménez, J. Parra-Arnau, A. Rodríguez-Hoyos, and J. Forné, "On the regulation of personal data distribution in online advertising platforms," *Engineering Applications of Artificial Intelligence*, vol. 82, pp. 13–29, 2019.
- [3] Z. Gharibshah and X. Zhu, "User response prediction in online advertising," *ACM Computing Surveys*, vol. 54, no. 3, pp. 1–43, 2021.
- [4] F. Kanei, D. Chiba, K. Hato, and M. Akiyama, "Precise and robust detection of advertising fraud," in *Proceedings of the 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, pp. 776–785, IEEE, Milwaukee, WI, USA, July 2019.
- [5] D. Liu, J. Ni, X. Lin, and X. Shen, "Transparent and accountable vehicular local advertising with practical blockchain designs," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 15694–15705, 2020.
- [6] Y. Ding, D. Luo, H. Xiang, W. Liu, and Y. Wang, "Design and implementation of blockchain-based digital advertising media promotion system," *Peer-to-Peer Networking and Applications*, vol. 14, no. 2, pp. 482–496, 2021.
- [7] C. Shi, R. Song, X. Qi, Y. Song, B. Xiao, and S. L. ClickGuard, "Exposing hidden click fraud via mobile sensor side-channel analysis," in *Proceedings of the ICC 2020-2020 IEEE International Conference on Communications (ICC)*, July 2020.
- [8] S. Almahmoud, B. Hammo, and B. Al-Shboul, "Exploring non-human traffic in online digital advertisements: analysis and prediction," *Computational Collective Intelligence*, Springer, in *Proceedings of the International Conference on Computational Collective Intelligence*, pp. 663–675, March 2019.
- [9] G. Thejas, K. Boroojeni, K. Chandna, I. Bhatia, S. Iyengar, and N. Sunitha, "Deep learning-based model to fight against ad click fraud," in *Proceedings of the 2019 ACM Southeast Conference*, pp. 176–181, Switzerland, May 2019.
- [10] T. Tian, J. Zhu, F. Xia, X. Zhuang, and T. Zhang, "Crowd fraud detection in internet advertising," in *Proceedings of the 24th International Conference on World Wide Web*, Switzerland, May 2015.
- [11] B. Stone-Gross, R. Stevens, A. Zarras, R. Kemmerer, C. Kruegel, and G. Vigna, "Understanding fraudulent activities in online ad exchanges," in *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, pp. 279–294, Berlin, Germany, November 2011.
- [12] J. Hu, T. Li, Y. Zhuang, S. Huang, and S. Dong, "A C Approach for Fraud Detection in mobile Advertising," *Security and Communication Networks*, vol. 2020, Article ID 1656460, 2020.
- [13] F. Kanei, D. Chiba, K. Hato, K. Yoshioka, T. Matsumoto, and M. Akiyama, "Detecting and understanding online advertising fraud in the wild," *IEICE - Transactions on Info and Systems*, vol. 103, no. 7, pp. 1512–1523, 2020.
- [14] IAB US benchmarking study, "What Is an Untrustworthy Supply Chain Costing: The U.S. Digital Advertising Industry?," 2015, https://www.iab.com/wp-content/uploads/2015/11/IAB_EY_Report.pdf.
- [15] R. Oentaryo, E. Lim, M. Finegold et al., "Detecting click fraud in online advertising: a data mining approach," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 99–140, 2014.
- [16] S. Wang, C. Liu, X. Gao, H. Qu, and W. Xu, "Session-based fraud detection in online e-commerce transactions using recurrent neural networks," *Machine Learning and Knowledge Discovery in Databases*, Springer, in *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 241–252, September 2017.
- [17] R. Mouawi, M. Awad, A. Chehab, H. Imad, H. El, and A. Kayssi, "Towards a machine learning approach for detecting click fraud in mobile advertising," in *Proceedings of the International Conference on Innovations in Information Technology (IIT)*, pp. 88–92, IEEE, Al Ain, United Arab Emirates, November 2018.
- [18] C. M. R. Haider, A. Iqbal, A. H. Rahman, and M. S. Rahman, "An ensemble learning based approach for impression fraud detection in mobile advertising," *Journal of Network and Computer Applications*, vol. 112, pp. 126–141, 2018.
- [19] D. Sisodia and D. Sisodia, *Gradient Boosting Learning for Fraudulent Publisher Detection in Online Advertising*, Data Technologies and Applications, United Kingdom, 2020.
- [20] J.-A. Choi and K. Lim, "Identifying machine learning techniques for classification of target advertising," *ICT Express*, vol. 6, no. 3, pp. 175–180, 2020.
- [21] N. P. Gohil and A. D. Meniya, "Click ad fraud detection using XGBoost gradient boosting algorithm," in *Proceedings of the International Conference on Computing Science, Communication and Security*, pp. 67–81, Springer, Gujarat, India, February 2021.
- [22] G. Thejas, S. Dheeshjith, S. Iyengar, N. Sunitha, and P. Badrinath, "A hybrid and effective learning approach for click fraud detection," *Machine Learning with Applications*, vol. 3, Article ID 100016, 2021.

- [23] E. Mikhailov and R. Trusov, "How Adversarial Attacks Work," 2017, <https://blog.ycombinator.com/how-adversarial-attacks-work/>.
- [24] O. Stitelman, C. Perlich, B. Dalessandro, R. Hook, T. Raeder, and F. Provost, "Using co-visitation networks for detecting large scale online display advertising exchange fraud," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1240–1248, Illinois, Chicago, USA, August 2013.
- [25] J. Xu and C. Li, "Detecting crowdsourcing click fraud in search advertising based on clustering analysis," in *Proceedings of the 2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing*, pp. 894–900, IEEE, Beijing, China, August 2015.
- [26] S.-C. Lee, C. Faloutsos, D.-K. Chae, and S.-W. Kim, "Fraud detection in comparison-shopping services: patterns and anomalies in user click behaviors," *IEICE - Transactions on Info and Systems*, vol. 100, no. 10, pp. 2659–2663, 2017.
- [27] J. Hu, J. Liang, and S. D. iBGP, "A bipartite graph propagation approach for mobile advertising fraud detection," *Mobile Information Systems*, vol. 2017, Article ID 7602384, 12 pages, 2017.
- [28] M. Meghanath, D. Pai, and L. A. ConOut, "Contextual outlier detection with multiple contexts: application to ad fraud," in *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 139–156, Springer, France, September 2018.
- [29] F. Dong, H. Wang, L. Li et al., "Automated ad fraud detection for android apps," in *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pp. 257–268, November 2018, <https://doi.org/10.1145/3236024.3236045>.
- [30] M. Gabryel and K. Przybyszewski, "The dynamically modified BoW algorithm used in assessing clicks in online ads," *Artificial Intelligence and Soft Computing*, Springer, in *Proceedings of the International Conference on Artificial Intelligence and Soft Computing*, pp. 350–360, April 2019.
- [31] S. Nagaraja and R. S. Clicktok, "Click fraud detection using traffic analysis," in *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks*, Florida, Miami, May 2019.
- [32] C. Cao, Y. Gao, Y. Luo et al., "Efficient and deployable click fraud detection for mobile applications," *IEEE Transactions on Mobile Computing*, vol. 20, no. 4, pp. 1285–1297, 2020.
- [33] N. Kshetri and J. Voas, "Online advertising fraud," *Computer*, vol. 52, no. 1, pp. 58–61, 2019.
- [34] L. Lv Z. Wang and M. Yang, "Research on trusted identity architecture in cyberspace based on eid," *Netinfo Security*, vol. 9, pp. 97–100, 2015.
- [35] P. Tammppuu and A. Masso, "Transnational digital identity as an instrument for global digital citizenship: the case of Estonia's E-residency," *Information Systems Frontiers*, vol. 21, no. 3, pp. 621–634, 2019.
- [36] A. Abraham, K. Theuermann, and E. Kirchengast, "Qualified eID derivation into a distributed ledger based IdM system," in *Proceedings of the 2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/12th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE)*, pp. 1406–1412, IEEE, New York, NY, USA, August 2018.
- [37] G. Shahaf, E. Shapiro, and N. Talmon, "Genuine personal identifiers and mutual sureties for sybil-resilient community growth," in *Proceedings of the International Conference on Social Informatics*, pp. 320–332, Springer, Doha, Qatar, November 2020.
- [38] G. Shahaf, E. Shapiro, and N. Talmon, "Genuine Personal Identifiers and Mutual Sureties for Sybil-Resilient Community Formation," 2019, <https://arxiv.org/abs/1904.09630>.
- [39] T. Zhu, Y. Meng, H. Hu, X. Zhang, M. Xue, and H. Zhu, "Dissecting Click Fraud Autonomy in the Wild," 2021, <https://arxiv.org/abs/2105.11103>.
- [40] Y. Wang, A. Zhang, P. Zhang, and H. Wang, "Cloud-assisted EHR sharing with security and privacy preservation via consortium blockchain," *IEEE Access*, vol. 7, Article ID 136719, 2019.
- [41] M. Shen, J. Zhang, L. Zhu, K. Xu, and X. Tang, "Secure SVM training over vertically-partitioned datasets using consortium blockchain for vehicular social networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 5773–5783, 2019.
- [42] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.
- [43] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," *Advances in Cryptology - CRYPTO 2001*, Springer, in *Proceedings of the Annual International Cryptology Conference*, pp. 213–229, August 2001.
- [44] D. Galindo, "Boneh-Franklin identity based encryption revisited," in *Proceedings of the International Colloquium on Automata, Languages, and Programming*, pp. 791–802, Springer, Portugal, August 2005.
- [45] V. Goyal, A. Jain, O. Pandey, and A. Sahai, "Bounded ciphertext policy attribute based encryption," in *Proceedings of the International Colloquium on Automata, Languages, and Programming*, pp. 579–591, Springer, Iceland, July 2008.
- [46] B. Waters, "Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization," in *Proceedings of the International Workshop on Public Key Cryptography*, pp. 53–70, Springer, Italy, March 2011.
- [47] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proceedings of the 2007 IEEE Symposium on Security and Privacy (SP'07)*, pp. 321–334, IEEE, Berkeley, CA, USA, May 2007.
- [48] R. Canetti and H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels," *Lecture Notes in Computer Science*, Springer, in *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 453–474, May 2001.
- [49] J. Leng, M. Zhou, J. Zhao, Y. Huang, and Y. Bian, "Blockchain Security: A Survey of Techniques and Research Directions," *IEEE Transactions on Services Computing*, Article ID 9271868, 2020.
- [50] J. Leng, G. Ruan, P. Jiang et al., "Blockchain-empowered sustainable manufacturing and product lifecycle management in industry 4.0: a survey," *Renewable and Sustainable Energy Reviews*, vol. 132, Article ID 110112, 2020.
- [51] J. Leng, S. Ye, M. Zhou et al., "Blockchain-secured smart manufacturing in industry 4.0: a survey," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 1, pp. 237–252, 2020.
- [52] J. Leng, P. Jiang, K. Xu et al., "Makerchain: a blockchain with chemical signature for self-organizing process in social manufacturing," *Journal of Cleaner Production*, vol. 234, pp. 767–778, 2019.
- [53] Q. Liu, J. Leng, D. Yan et al., "Digital twin-based designing of the configuration, motion, control, and optimization model of a flow-type smart manufacturing system," *Journal of Manufacturing Systems*, vol. 58, pp. 52–64, 2021.

- [54] Q. Liu, H. Zhang, J. Leng, and X. Chen, "Digital twin-driven rapid individualised designing of automated flow-shop manufacturing system," *International Journal of Production Research*, vol. 57, no. 12, pp. 3903–3919, 2019.
- [55] J. Leng, H. Zhang, D. Yan, Q. Liu, X. Chen, and D. Zhang, "Digital twin-driven manufacturing cyber-physical system for parallel controlling of smart workshop," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 3, pp. 1155–1166, 2019.
- [56] V. Boyko, P. MacKenzie, and S. Patel, "Provably secure password-authenticated key exchange using Diffie-Hellman," *Advances in Cryptology - EUROCRYPT 2000*, Springer, in *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 156–171, May 2000.
- [57] Y. Ma, Y. Liu, B. Xu, and J. Zhi, "Impacts of Waiting on mobile Users – Case Study of Digital Novels," *Data Analysis And Knowledge Discovery*, vol. 2, no. 8, 2018.

Research Article

A Novel Semifragile Consensus Algorithm Based on Credit Space for Consortium Blockchain

Xiaohong Deng ^{1,2,3}, Zhiqiong Luo ², Yijie Zou ², Kangting Li ², and Huiwen Liu ¹

¹School of Electronics and Information Engineering, Gannan University of Science and Technology, Ganzhou 341000, China

²School of Information Engineering, Jiangxi University of Science and Technology, Ganzhou 341000, China

³Key Laboratory of Cloud Computing and Big Data, Ganzhou 341000, China

Correspondence should be addressed to Zhiqiong Luo; 6720200799@mail.jxust.edu.cn

Received 24 January 2022; Accepted 28 March 2022; Published 18 April 2022

Academic Editor: Jiewu Leng

Copyright © 2022 Xiaohong Deng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Nowadays, blockchain is known as a new generation of secure information technologies for realizing business and industrial sustainability, and consensus algorithm is the key technology of blockchain. In order to solve the problem of “oligarchy” nodes and excessive punishment for nodes in existing credit consensus algorithms, a novel semifragile consensus algorithm based on the credit space for consortium blockchain is proposed in this paper. Firstly, the accounting node selection mechanism based on credit space is proposed. The credit value of the node is calculated according to a novel credit evaluation model, and then the credit space of the node is allocated according to the size of the credit value. Afterward, a random algorithm is used to select the accounting node in the credit space. This mechanism effectively inhibits the generation of “oligarchy” nodes and maintains the enthusiasm of nodes. Secondly, this paper proposes a semifragile hierarchical punishment mechanism, which punishes the malicious nodes with severe measures and gives the nonmalicious nodes the opportunity to continue participating in the consensus. So, this semifragile punishment mechanism solves the problem of excessive punishment of nodes. Experimental simulation results demonstrate that the proposed consensus algorithm has randomness while maintaining the credit incentive among nodes. In addition, the node’s punishment mechanism is more reasonable. This algorithm has better security and can be well applied to consortium blockchain scenarios.

1. Introduction

In 2008, Satoshi Nakamoto publicly published Bitcoin [1]. Afterward, with the crazy of Bitcoin, blockchain as a core technology of Bitcoin has received extensive research attention [2]. Blockchain has the characteristics of decentralization, hard tamperability, traceability, and transparency, which solves the data monopoly and security problems current in the existing centralized platform [3]. At the same time, many studies have found that blockchain has many innovative applications in the field of IoT and sensor networks. For instance, Satapathy et al. [4] proposed a secure architecture based on open blockchain, which can solve some of the challenges in IoT applications, like issues with confidentiality and privacy of data; Mrinal et al. [5] proposed

a blockchain-based wireless sensor network for secure vehicle tracking, reducing the need for an Internet connection and eliminating the use of continuous GPS tracking; that is, it can effectively protect the privacy of commuters and the security of collected data. Therefore, blockchain is known as a new generation of secure information technology. As the core part of the blockchain system, the consensus algorithm is the mechanism for each node of the blockchain to reach consensus on the block information of the whole network [6]. More precisely, it can ensure whether the latest block is correctly added to the blockchain. It is worth mentioning that the performance efficiency and security of the entire blockchain system will be affected by the merits of the consensus algorithm [7]. Similarly, consensus algorithm has always been the key technology of decentralized system,

which is widely used in resource-constrained edge computing fields. For instance, Zeng et al. [8] proposed a scheme by utilizing the idle resources in volunteer vehicles to handle the overloaded issues in VEC servers; the scheme can reduce the offloading cost of vehicles and improve the utility of VEC servers; Zeng et al. [9] proposed a new vehicle edge computing framework based on software-defined networks, which introduces the reputation to measure the contribution of each vehicle. The proposed scheme not only brings more benefits to the edge server side but also reduces the average delay a lot.

Generally, different blockchain frameworks use different consensus algorithms. In summary, a common classification divides blockchain into three categories, including public blockchain, consortium blockchain, and private blockchain [10]. The number of nodes in the public blockchain is large, so the transaction speed will be slower. On the contrary, there are fewer nodes in private blockchain and consortium blockchain than in public blockchain, and the transaction speed will be faster [11]. However, the permissions in the private blockchain are controlled by a few nodes, which deviates from the original intention of decentralization [12]. Compared with the private blockchain, the permission design requirements in the consortium blockchain are more complicated and more credible. Now, relevant researches show that the consortium blockchain has more practical value in the fields of IoT applications and medical scenarios. For example, Thomas et al. [13] proposed an anonymous identity and access control system based on consortium blockchain, which improves the security of cross-domain identity authentication in the Internet of Things; Huang et al. [14] proposed a medical data privacy protection and safe sharing scheme based on consortium blockchain, which can effectively ensure the safety of patients' medical information and can safely share information.

At present, the consensus algorithm of consortium blockchain is mainly represented by the Practical Byzantine Fault Tolerance (PBFT) protocol [15]. PBFT has a high transaction speed; however, with the number of nodes increasing, the network overhead of PBFT will increase rapidly, and the consumption of computing power will be high [16]. Moreover, PBFT selects the leader node according to the continuous switching of view number, which may select malicious nodes as the leader node, resulting in poor system security [17]. As such, in order to solve the problem of malicious nodes becoming accounting nodes, researchers have proposed a credit mechanism to generate accounting nodes. The credit value was calculated on the basis of the node's performance in the system, and the node with a higher credit value preferentially became the accounting node [18]. For instance, Li et al. [19] proposed a consortium consensus algorithm based on credit (CCAC), which calculated the credit value of nodes by the contribution of node participation consensus, and selected a node to become an accounting node in turn according to the size of node credit. Notably, a consensus mechanism based on credit reduced the consumption of algorithm computing power and improved the efficiency of consensus. However, this is not effective for the node with a

small credit value and easily leads to low enthusiasm of nodes. Wang et al. [20] proposed a proof of work algorithm based on credit model (CPoW) and designed a node credit model based on BP neural network, which effectively reduced the huge resource consumption of repeated calculation in the production process of new blocks. Unfortunately, generating new blocks according to the order of credit value was easy to produce "oligarchy" nodes. Li et al. [21] proposed a dynamic hierarchical Byzantine fault-tolerant consensus mechanism based on credit (DHBFT). The presented reward and punishment plan could effectively reduce the possibility of malicious nodes becoming the leader node, but it could easily cause node with high credit values to be selected as the master node, which lacks fairness and easily causes other nodes to be less motivated. Liu et al. [22] proposed a master-slave multichain blockchain consensus mechanism based on reputation, which introduced credit value evaluation into the consensus mechanism based on proof of stake. In addition, it designed a joint consensus mechanism that integrates multiple consensus mechanisms, which improved the throughput of the transaction and ensured the consistency and nontamperability of the data. However, the punishment for all malicious nodes was too heavy, resulting in nodes being unable to normally participate in the consensus for a long time. Bugday et al. [23] proposed a reputation-based consensus group learning model to calculate the credit value based on the weight value of all nodes in the trust committee, which could effectively avoid malicious nodes, but the weight value of malicious nodes is large. Once a node had malicious acts, the credit value of this node would fall to a very low level, and it was difficult to continue to join the consensus. Huang et al. [24] proposed a credit-based proof of work mechanism for IoT devices, which improved security and enhanced transaction efficiency. Similarly, the punishment for malicious nodes was to reduce the credit value directly to a negative value, which made it difficult for nodes to participate in normal consensus.

To sum up, although the existing consensus algorithm based on credit has improved the efficiency and security of consensus, there are still problems that it is easy to generate "oligarchy" nodes and the punishment for nonmalicious nodes is too large. In order to solve the above problems, this paper proposes a semifragile consortium blockchain consensus algorithm based on credit space. The main contributions of this paper are as follows:

- (1) An accounting node selection mechanism based on credit space is proposed. A credit evaluation model is formulated to calculate the credit value of the node, and the credit space of the node is allocated based on the credit value. Based on the credit space, an algorithm for randomly selecting accounting nodes is designed. The nodes with large credit space have a high probability of becoming accounting nodes. At the same time, a threshold equation for the number of accounting nodes is set for the problem of "oligarchy" nodes so that the number of times of becoming accounting nodes is limited.
- (2) A semifragile hierarchical punishment mechanism is designed. Nodes with good working conditions are

in the *normal* layer, and the nodes with malicious behaviours are placed in the *prison* layer for “custody.” Furthermore, we judge whether the node is malicious or nonmalicious; for malicious nodes, the “custody” time will be longer, and for nonmalicious nodes, they can be returned to the *normal* layer beyond the “custody” time. Therefore, the nonmalicious nodes have the opportunity to participate in the following consensus, and this mechanism can reduce the existence rate of malicious nodes.

2. Problem Statement

2.1. Problem of “Oligarchy” Nodes. Among the existing consensus algorithms based on credit, most of the accounting nodes are selected according to the size of the credit value, which is easy to produce “oligarchy” nodes, and the incentive degree for nodes with small credit value is not enough, such as CCAC algorithm [19]. The credit value of each node is calculated after the credit evaluation of the node, then the credit value is sorted from largest to smallest, and an accounting node is selected in this order, which can easily lead to the production of “oligarchy” nodes and cause other nodes to be less motivated. In this paper, we test the proportion of “oligarchy” nodes as accounting nodes in the total consensus times for CCAC to verify the adverse effects of “oligarchy” nodes on the network, and the results are shown in Table 1.

It can be seen from Table 1 that, with the number of consensus increasing, the number of “oligarchy” nodes becoming accounting nodes also accounts for an increasing proportion, which can easily cause other nodes to be less motivated to work. Therefore, this paper proposes a mechanism for selecting accounting nodes based on credit space, which can effectively inhibit the generation of “oligarchy” nodes and increase the enthusiasm of nodes.

2.2. Problem of Node’s Excessive Punishment. In view of the existing consortium blockchain consensus algorithm based on credit, the punishment for malicious nodes is too severe. More precisely, they do not judge whether the malicious behaviour of a node is deliberate or not, and the credit value of the nodes is always severely reduced so that these nodes cannot continue to participate in the consensus, typically such as the consensus algorithm in [22]. A PoS consensus mechanism based on credit value is proposed, and a credit value evaluation method is designed. The punishment equation for the credit value of malicious nodes is as follows:

$$\text{trust}_h^i = -\text{trust}_{h-1}^i, \quad (1)$$

where trust_h^i represents the credit value of node i at the end of the h th cycle and trust_{h-1}^i represents the credit value of node i at the end of the $h-1$ th cycle. It can be seen from equation (1) that the credit value of the malicious node will be directly reduced to a negative value, making it difficult for the node to continue to participate in the following consensus. Besides, references [23, 24] mentioned in the Introduction also have the same punishment

for malicious nodes. Both have too harsh punishments for malicious nodes, and normal consensus cannot be carried out for a long time. In this paper, we compare these algorithms to test the change in credit value of nodes with malicious behaviours, and the experimental results are shown in Figure 1.

It can be seen from Figure 1 that the credit value of malicious nodes in [22] will rapidly decrease from positive value to negative value, which is difficult to continue to participate in consensus for a long time. Although the algorithm in [23] did not reduce to a negative value, the credit value is very close to 0 and cannot compete with the credit value of normal nodes. In [24], the credit value of the malicious node is always below 0, and it is difficult to continue the normal consensus. By comparing the changes in the credit value of nodes with malicious behaviour in these three algorithms, it can be seen that they cannot participate in normal consensus for a long time for nodes with malicious behaviour. Therefore, this paper proposes a semifragile hierarchical punishment mechanism. This mechanism can make it difficult for malicious nodes to participate in consensus again, but nonmalicious nodes can continue to participate in consensus within a short amount of time.

3. Proposed Algorithm

3.1. Credit Evaluation Model. The credit of nodes represents the working performance of nodes in the process of participating in consensus [18]. The credit evaluation model proposed by the CCAC only considers the number of valid and invalid blocks generated by the accounting node and the time required to add on the chain [19]. However, it does not consider the time when the node is passive and offline from the block. In this paper, the credit evaluation of nodes will be carried out according to the four indicators of the number of transactions in the valid block, the time of the chain, the off-chain time, and the generation of invalid blocks. The credit evaluation indicators are given in Table 2.

Combined with these credit indicators, the data will be standardized so that the data can be calculated uniformly. In this paper, the minimum-maximum planning method is used to standardize the data. This method is the linear transformation of the original data, and the maximum \max and minimum \min will be set. After calculation by the standardized equation, the data range will fall between $[0, 1]$, and then the following credit value is calculated. The computation equation is as follows:

$$i' = \frac{i - \min}{\max - \min}, \quad (2)$$

where \max and \min are obtained by preprocessing. In particular, the result of preprocessing is based on 100 consensus experiments in this paper. In the process of consensus experiments, the data of these indicators will be obtained, and \max and \min are the maximum and minimum values of data in each indicator. When an indicator in a node needs to be measured, it is only necessary to put the data into

TABLE 1: Percentage statistics of “oligarchy” nodes become accounting nodes.

| Total consensus number | The proportion of times that nodes become accounting nodes (%) |
|------------------------|--|
| 400 | 52.4 |
| 600 | 58.3 |
| 800 | 62.8 |
| 1000 | 78.1 |

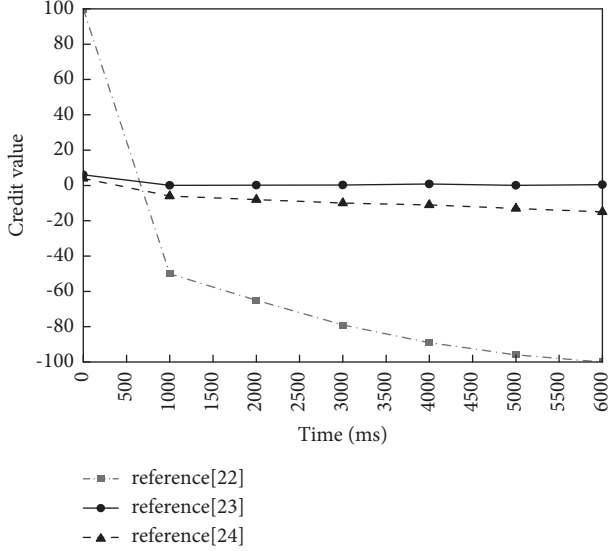


FIGURE 1: The credit value change diagram of the malicious nodes of each algorithm, recorded in a 6000 ms period. These dots represent the credit value of the malicious node taken every 500 ms.

the (2) to obtain the standardized value: i'_{num} , i'_{time} , $i'_{off-time}$, $i'_{invalid}$.

After getting the standardized data, some data may be positive or negative. Then, these data are added together, and finally, a value x that reflects the quality of the credit value is obtained, as shown in (3). Subsequently, the credit value $C(x_i)$ is to be accumulated or deleted by node i by (4).

$$x_i = i'_{num} + i'_{time} + i'_{off-time} + i'_{invalid}, \quad (3)$$

$$C(x_i) = \int_{-\infty}^x \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/2\sigma^2} dx, \quad -\infty < x < \infty, \quad (4)$$

where x_i represents the number after processing of the standardized data mentioned above. $C(x_i)$ represents the credit value of node i . Besides, μ is the mean value calculated from the data obtained in the preprocessing, and σ is the variance calculated after preprocessing.

Similarly, the credit value of the accounting node that works hard will be accumulated, as shown in (5). The initial credit value of each node is 1. When node i becomes an accounting node for the first time, its credit value is equal to the initial credit value plus the $C(x_i)$ calculated by (4). Moreover, when node i is selected as the accounting node again, its credit value is the sum of the newly calculated credit value and the previously obtained. The equation for calculating the credit value of node i as an accounting node for the n th time is as follows:

$$Credit_i^n = \begin{cases} 1 + C(x_i) & n = 1, \\ Credit_i^{n-1} + C(x_i) & n \neq 1. \end{cases} \quad (5)$$

In contrast, for nodes with malicious behaviour, it will be subtracted after calculating the corresponding credit value, as shown in (6). For the initial malicious nodes, its credit value is the initial credit value minus $C(x_i)$ calculated by (4), that is, the credit value obtained after work. But for the malicious nodes in the consensus process, the credit value subtracts the new credit value from the previous credit value. The calculation equation of the m th malicious credit value of node i is as follows:

$$Credit_i^m = \begin{cases} 1 - C(x_i) & m = 1, \\ Credit_i^{m-1} - C(x_i) & m \neq 1. \end{cases} \quad (6)$$

Through the credit evaluation model for node credit evaluation, the nodes in working well condition can get a larger credit value. That is, their opportunity to become an accounting node will be greater, which creates a benign network environment for nodes actively participating in consensus.

3.2. Credit Space. In this paper, the credit space is used as the basis for selecting an accounting node. After a node obtains its credit value, its corresponding credit space is allocated according to the proportion of the credit value in the entire space. That is, the greater the credit value of the node, the greater the allocated space, and the greater the probability of becoming accounting node. For this reason, this method can better motivate nodes to work. Furthermore, the random algorithm also ensures the randomness of the algorithm, and it does not mean that nodes with larger credit space will certainly become accounting nodes. The credit space of node i can be computed by

$$C_Space_i = \frac{Credit_i}{\sum_{i=1}^n Credit_i} \times L, \quad (7)$$

where $Credit_i$ represents the credit value of node i and $\sum_{i=1}^n Credit_i$ is the sum of the credit values of each node in this round. L is the total length of the credit space. Figure 2 is a graph of the change of credit space when a node is selected as an accounting node. Figure 2(a) shows the distribution of credit space of a certain round of nodes. Assuming that the pointer is randomly selected to node 3, the credit value of node 3 becomes larger after being selected as an accounting node and packaged block successfully. Obviously, according to the calculation equation of credit space, its credit space will also become larger. So node 3 has a greater probability in the next selection of accounting node, which is like playing a

TABLE 2: Credit evaluation indicators.

| Indicator name | Explanation |
|----------------|--|
| i_{time} | The time when the node generates a block |
| $i_{off-time}$ | The time the node leaves the blockchain |
| i_{num} | The number of transactions in a valid block |
| $i_{invalid}$ | The number of invalid blocks generated by nodes in consensus |

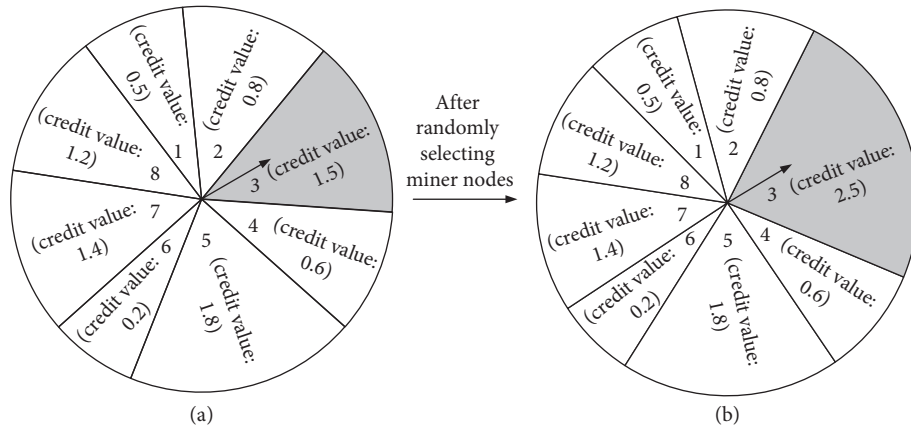


FIGURE 2: Credit space change diagram. Each sector in the figure represents the credit value of each node, and the pointer is like a turntable to represent random selection. (a) Distribution of credit space of each node in a certain round. (b) Distribution of credit space after selecting miner nodes.

roulette game. The greater the credit space of node 3, the greater the probability of the pointer pointing to node 3. Since the size of the whole credit space is fixed, each node's credit space is calculated according to the proportion of credit value, so the credit space of other nodes will be proportionally reduced.

What is more, after the node is selected as an accounting node, the corresponding packaging work must be completed. The packaging work involves the block structure, which is used to store and verify the credit value. The block structure includes the following:

- (1) Blockhead: block version number, hash value of the previous block, timestamp, and random number.
- (2) Blockchain time: record the time accounting nodes successfully package block into chain, which helps to verify whether the credit value is accumulated correctly.
- (3) The hash value of the block's transaction data: record the transaction data generated by the accounting node of the block.
- (4) Credit array in block: record the credit value obtained by nodes on blocks.
- (5) Counting array in block: record the number of times a node becomes an accounting node.

Once completed the packaging work, the credit value will be calculated and accumulated in the original credit value. When the next accounting node selection begins, the credit space will be allocated according to the size of the credit value. However, there is a problem at present. Nodes with larger credit values may always be selected as

accounting nodes, which leads to the generation of "oligarchy" nodes. Therefore, a threshold for becoming an accounting node is set. When it exceeds the current number threshold, it cannot continue to be selected as an accounting node. The threshold equation is as follows:

$$\tau = \frac{\sum_{i=1}^{Num} num_i}{Num} + t, \quad (8)$$

where τ is a threshold, Num represents the number of nodes, num_i denotes the number of node i becoming an accounting node, and $\sum_{i=1}^{Num} num_i$ represents the total number of times that all nodes become accounting nodes. The threshold calculated by the equation will change with the number of nodes becoming accounting nodes in the whole consensus network. When the threshold of this round increases, nodes may still be selected as accounting nodes. The constant t in the equation will be obtained through experiments, and the specific value is explained in the subsequent experimental part.

3.3. Semifragile Hierarchical Punishment Mechanism. Generally, the punishment of malicious nodes in the existing credit mechanism is too severe, which directly reduces the credit value of malicious nodes and makes it too difficult to continue to participate in consensus. Consequently, this paper proposes a semifragile hierarchical punishment mechanism. Semifragile refers to the ability to distinguish whether a node with malicious acts is deliberate or non-deliberate. In our algorithm, the nodes judged as non-deliberate are given the opportunity to reparticipate in consensus.

In order to determine whether the malicious node is deliberate or nondeliberate, this paper judges the node by the number of malicious acts. When the number of malicious acts of a node is less than m , it is judged as a nondeliberate node, and vice versa. With respect to the value of m , this paper counts the number of malicious nodes through many experiments, and the results are shown in Table 3.

It can be seen from Table 3 that the nodes with the number of malicious acts less than or equal to 2 account for 98.53%, basically covering most of the nodes. Therefore, this paper selects 2 as the value of m , which is determined as the critical value of nonmalicious nodes.

The specific process of the semifragile hierarchical punishment mechanism is as follows; the general process is shown in Figure 3. First, all nodes will be placed in the *normal* layer, and then the credit space of nodes is calculated to select the accounting node. Moreover, the credit evaluation of the accounting node will be carried out. If the node has malicious behaviour, the node will be placed in the *prison* layer after calculating the credit value. It is worth mentioning that the nodes in the *prison* layer have no chance to be selected as accounting nodes and only the nodes in the *normal* layer have the chance to allocate the credit space to be selected as accounting nodes. The nodes in the *prison* layer will allocate the “custody” time according to the number of malicious acts. During this period, the nodes still need to participate in the data synchronization of the cluster. In particular, if the node is found to have malicious behaviour such as not performing block data synchronization or not working, it will continue to increase the “custody” time. After the time has passed, it is determined whether the node is deliberate or nondeliberate. If the node is a non-deliberate node, it will return to the *normal* layer and give it the opportunity to be selected as an accounting node again. Otherwise, the malicious node will continue to be punished.

About the node’s “custody” time, when the number of nodes performing malicious acts increases, the time will increase obviously with the number of times. According to this characteristic, this paper uses the following function:

$$T = e^x. \quad (9)$$

The function is monotonically increasing, where T is the “custody” time and x is the number of malicious acts. It can be seen from (9) that T is monotonically increasing, that is, when x increases, that is, when the number of malicious acts increases, the time increases exponentially. In contrast, for nonmalicious nodes, only one or two malicious activities are performed, and the “custody” time is relatively appropriate. It conforms to the principle of the proposed punishment mechanism, gives a good buffer to the nodes that do not deliberately perform malicious acts, and then gives the opportunity to participate in the consensus.

4. Algorithm Design

Firstly, the credit value of all nodes is initialized to 1. In the beginning, each node is placed in the *normal* layer, and each participating node is numbered. Moreover, the

TABLE 3: Statistics of the number of malicious acts. The proportion indicates the proportion of nodes with different times of malicious acts in the total nodes.

| Number of malicious acts | Proportion of the number of nodes (%) |
|--------------------------|---------------------------------------|
| ≤ 0 | 96.25 |
| ≤ 1 | 97.84 |
| ≤ 2 | 98.53 |
| > 2 | 1.47 |

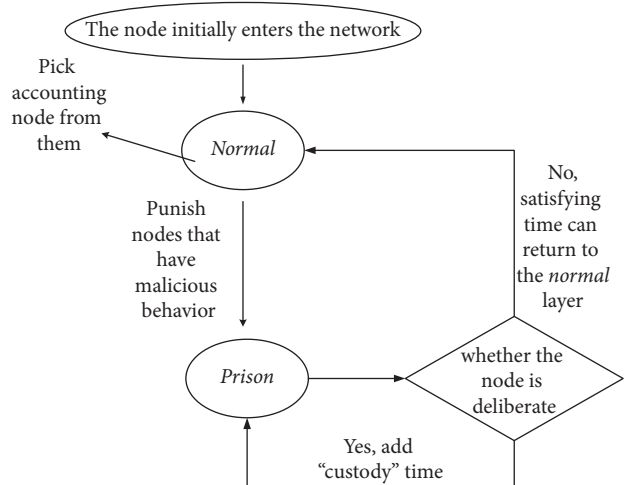


FIGURE 3: Semifragile hierarchical punishment mechanism. Malicious nodes will judge whether the node is deliberate and take corresponding measures.

corresponding credit space is allocated according to the credit value of each node. Obviously, the size of the space allocated by each node is the same, and the total space is unchanged. Then the credit array Cn and count array Cc are constructed. Cn is used to store the credit value of the node, and Cc is the number of times the storage node has become an accounting node. Thereafter, begin the cycle of selecting the accounting nodes. The process of credit consensus is shown in Figure 4. As shown in Figure 4, the whole process can be divided into four steps: initialization stage, cyclic selection of accounting node stage, constructing block stage, and checking the new block stage.

4.1. Initialization Stage. In the initial stage, the initial credit value of each participating node in the *normal* layer is set to 1, and the total credit space length is set to 100. The credit space of each node is calculated by equation (7), and the number of participating nodes is assigned. Thereafter, the credit array Cn and the count array Cc are constructed to store the credit value of the node and the number of nodes becoming accounting nodes, respectively. Algorithm 1 shows how to allocate the node’s credit space.

4.2. Cyclic Selection of Accounting Node Stage. Through Algorithm 1, we have obtained the credit space of each node. Then, the algorithm randomly selects the accounting node.

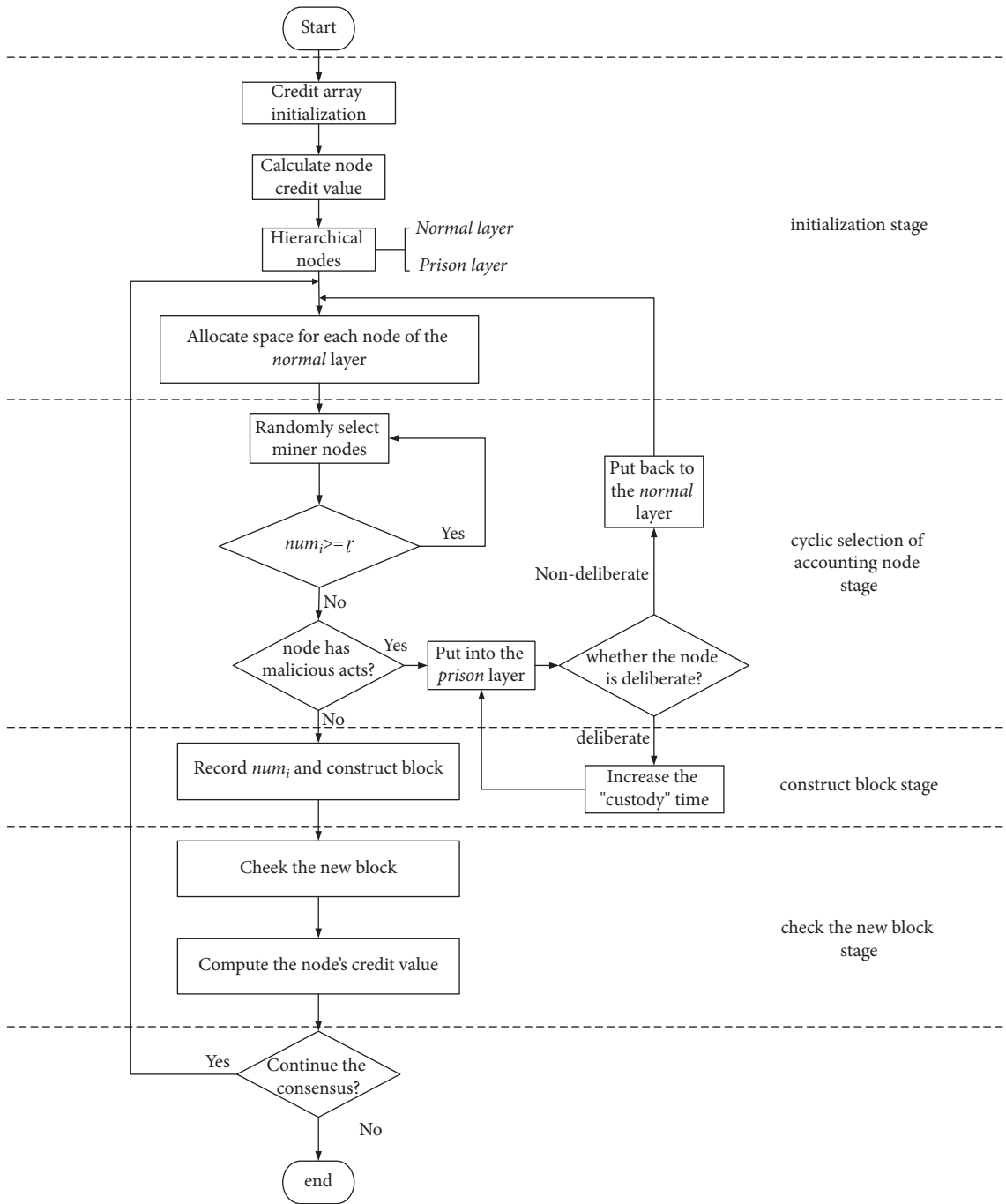


FIGURE 4: Credit consensus process. This flowchart describes how nodes select accounting nodes and how to punish malicious nodes.

More precisely, each interval represents each node's credit space, and the algorithm selects accounting node by setting a random number and judging which interval the random number falls into. As a result, the node represented by this interval is selected as accounting node. Subsequently, the accounting node will complete the corresponding work and obtain the corresponding credit value. Generally, if an accounting node does not work, the node will be punished beyond the given time and enter the next accounting node's selection. When the next accounting node selection is conducted, the corresponding space will be allocated according to the credit value. If the credit value is larger, it is

easier to obtain the packaging right. Since it is randomly selected, there will be nodes with low credit values that get the right to package. In order to avoid the generation of "oligarchy" node, τ is set. If num_i exceeds τ , node i cannot be selected as an accounting node. But it does not mean that i cannot be selected as an accounting node anymore because τ will change with $\sum_{i=1}^{Num} num_i$ in the whole consensus network. When τ becomes larger, node i may still be selected as the accounting node. Algorithm 2 gives the process of randomly selecting an accounting node.

If the node has malicious acts, the node will be placed in the *prison* layer for "custody." "Custody" time will be

```

Input:  $C_n$  (node's credit value array)
Output: spaceArray (node's credit space array)
(1) spaceArray[] = {0}; //Initialization of Credit Space Array
(2) for  $i = 1$  to  $n$  do //Traversing all nodes
(3)   if JudgePrsion ( $C_n[i]$ ) //Judge if the node is in the prison layer, if it is, do not allocate
(4) space
(5)   continue;
(6)   end if
(7)   if  $i = 0$  //When  $i$  is the first node in space
(8)     spaceArray[i] = ( $C_n[i]$ /countSum ( $C_n$ )) * spaceLength; //Calculating the length of credit
(9) space
(10)  continue;
(11)  end if
(12)  spaceArray[i] = ( $C_n[i]$ /countSum ( $C_n$ )) * spaceLength + spaceArray[i - 1]; //The length of
(13) credit space after becoming an accounting node,  $C_n$  is an array of normal layers
(14) end for

```

ALGORITHM 1: Node layering and credit space allocation algorithm.

calculated according to (9). Then, it will determine whether the time has expired. If it has expired, determine whether the node is a malicious node. If it is not a malicious node, it will be released back to the *normal* layer. If it is a malicious node, continue to stay in the *prison* layer for “custody.” If it has not expired, it will continue to stay in the *prison* layer. Algorithm 3 gives the penalty mechanism of malicious nodes.

4.3. Construct Block Stage. Once the accounting node is selected, the accounting node will broadcast the constructed block to all adjacent nodes. Afterward, adjacent nodes will receive the new block and broadcast it to the whole network after successful verification. When the block is verified, the block will be added to each node's blockchain copy. After all nodes have received and verified the block, the work of the next block construction will proceed.

4.4. Check the New Block Stage. After selecting the accounting node and completing the related transactions on the block, the node will broadcast the generated block to the whole network and then verify the credit value. Once the verification is correct, the node will obtain the corresponding credit reward. On the contrary, if the node has malicious behaviour, the behaviour will be recorded in the block, and the corresponding credit punishment will be carried out.

5. Experimental Results and Analysis

In our experiments, we use Golang programming language and JetBrainsGoLand 2020.3.4 for the simulation test. First, we use the Go language to write a single-machine multinode platform to simulate the consensus process. Then, we compare the performance of the consensus algorithm proposed in this paper with CCAC algorithm [19], CPoW algorithm [20], and master-slave multichain algorithm [22], and test the number of malicious nodes, punishment mechanism, and the consensus delay of nodes. Finally, the

images are drawn according to the experimental data for comparative analysis.

5.1. Threshold Equation Constant Experiment. This experiment is to analyze the value of threshold equation constant. We selected 40 nodes for 600 consensuses and tested the average time consumption to select accounting nodes under different threshold equation constants.

It can be seen from Figure 5 that the average time consumption of selecting accounting nodes with a constant 3 is the least, while the average time consumption of other nodes is relatively high. Therefore, we choose constant 3 as the value of t in the threshold equation.

5.2. Statistics of Accounting Node Number. In this experiment, we test the number of times nodes become accounting nodes to verify the credit evaluation model and the mechanism of selecting accounting nodes. First, set 20 nodes, conduct consensus on them 600 times, and select accounting nodes. The experimental results are shown in Figure 6. As can be seen from the data in Figure 6, each node can become an accounting node in the consensus process. Some nodes have become accounting nodes only 5 or 6 times, and some nodes have become accounting nodes 18 or 20 times. This shows that the proposed algorithm can reflect the role of credit value and ensure the randomness of selecting accounting nodes through credit space.

In order to test whether the threshold τ can better limit the “oligarchy” node, this paper tests 20 nodes, carries out 1500 consensuses on them, selects the accounting node, and then records the number of rounds of threshold change and the highest number of accounting node in this round. The experimental results are shown in Figure 7. As shown in Figure 7, the threshold τ will change with the number of nodes becoming accounting nodes in the whole network, and the number of accounting nodes is also limited to the threshold τ . The number of accounting nodes increases more and more slowly and requires a longer consensus time. This

```

Input: spaceArray (Node's Credit Space Array)
Output:  $i$  (accounting node's serial number)
(1) while (nodeSelect) //nodeSelect is whether to select the miner to complete the identifier,
(2) the initial value is true
(3)   rand.Seed (time.Now().Unix()); //Set random number time seed
(4)   randomSize = randomFloat (0, spaceLeangth); //Random number selected in space
(5)   node = judgeSelect (spaceArray, randomSize); //Determine which node is selected
(6)   if CoutArray [node] ≤ Exceeded //The requirement cannot exceed the threshold
(7)     nodeSelect = false; //The selection is complete, jump out of the loop, otherwise
(8)   continue to choose
(9)     Cc[i]++; //Count value plus 1
(10)    return  $i$ ;
(11)  end if
(12)  end while

```

ALGORITHM 2: Random selection accounting node algorithm.

```

Input: U (the set of malicious nodes)
Output: prisonArray (prison layer array)
(1) prisonArray[] = {}; //Initialize the prison layer
(2) while (node in U)
(3)   if JudgeMalicious (node) //Determine whether the node is malicious
(4)     time = pow (e, x); //Calculate penalty time
(5)     insert (node, prisonArray, time); //Put the node in jail and record the punishment time
(6)     node++; //Pointer moved to the next malicious node
(7)     continue;
(8)   end if
(9)   if JudgeTimeOut (node) //Determines whether the node penalty time expires
(10)    if (maliciousCount ≤ 2) //Determines whether the node is a malicious node
(11)      remove (node, prisonArray); //Remove the node
(12)      node++; //Pointer moved to the next malicious node
(13)      continue;
(14)    else
(15)      stayPrison (node); //Leave the node in the prison layer
(16)      node++; //Pointer moved to the next malicious node
(17)      continue;
(18)    end if
(19)  end if
(20)  end while

```

ALGORITHM 3: Punishment algorithm for malicious nodes.

shows that the proposed threshold mechanism can effectively restrain the emergence of “oligarchy” nodes.

5.3. Semifragile Hierarchical Punishment Mechanism Experiment. In order to prove that the punishment mechanism proposed in this paper can effectively avoid malicious nodes destroying the consensus process, we do an experiment to test the number of malicious nodes in different algorithm. The experimental results are given in Figure 8. At first, 1000 nodes are set in the system, and 273 malicious nodes are set and labelled artificially in these nodes. With the increase of consensus times, it can be found that the number of labelled malicious nodes in each algorithm is gradually decreasing, but it should be noted that the number of malicious nodes in the proposed algorithm in this paper has a more obvious decline.

From Figure 8, it can be seen that when the 70th consensus is carried out, the number of labelled malicious nodes in the algorithm proposed in this paper is reduced to 32, and the number of malicious nodes in other algorithms is more than that of this algorithm. This shows that the credit evaluation model proposed in this paper will gradually reduce the credit value of the malicious nodes. At the same time, the hierarchical punishment mechanism will also punish malicious nodes, which further restrains malicious nodes from doing evil. With the increase in the number of consensus, the probability of selecting the malicious nodes as accounting nodes will be greatly reduced; as such, it will make the blockchain system more safe and reliable.

In order to further test the performance of the proposed semifragile hierarchical punishment mechanism, this paper does an experimental test to determine the malicious

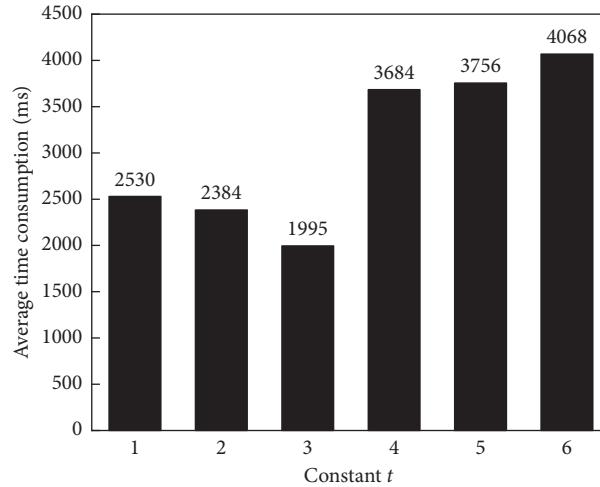


FIGURE 5: The experimental statistics of the constant t . The average time consumption of each constant in 600 consensuses is recorded.

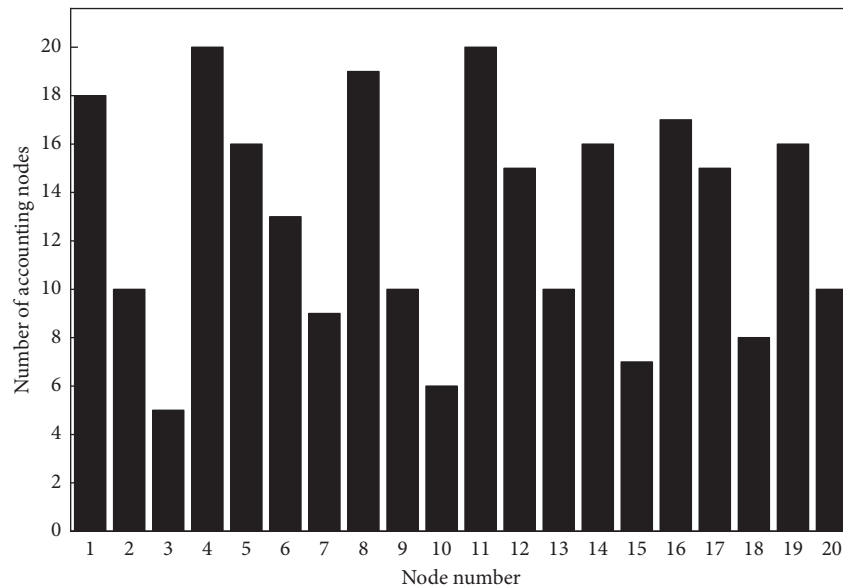


FIGURE 6: Statistics of the number of accounting nodes. The number of times each node becomes an accounting node in 600 consensuses is recorded.

behaviours of nodes. Firstly, a deliberate node and a non-deliberate node are marked, respectively, and they are placed in the *prison* layer. According to the proposed mechanism, nondeliberate nodes will be put back to the *normal* layer over time to continue to join the consensus, we record their credit values to observe the work of the node, and the results are shown in Figure 9.

It can be seen from Figure 9 that the credit value of a nondeliberate node decreases after malicious acts. After putting it into the *prison* layer, the credit value remains unchanged. If it is put back to the *normal* layer after exceeding the “custody” time, it can normally participate in the consensus. However, the credit value of the deliberate node declines after committing malicious acts. It is worth mentioning that if the deliberate node continues to commit malicious acts in the *prison* layer, the “custody” time will be double. It shows that the mechanism gives an opportunity to

nondeliberate nodes and does not reduce its credit value to the point of being unable to participate in the consensus. That is, it makes nondeliberate nodes become normal nodes, while deliberate nodes are punished accordingly.

5.4. Consensus Delay. The consensus delay comparison results are shown in Figure 10. As can be seen from Figure 10, with the increase of consensus times, consensus delay increases gradually. The consensus delay of the CCAC algorithm is the lowest, the consensus delay of the proposed algorithm is only higher than that of CCAC, and the consensus delay of the CPoW algorithm is the highest. This is because the proposed algorithm in this paper selects accounting nodes based on the credit space and introduces the hierarchical punishment mechanism, which results in higher delay than CCAC. However, the consensus delay of the algorithm is within an acceptable range, and it does not

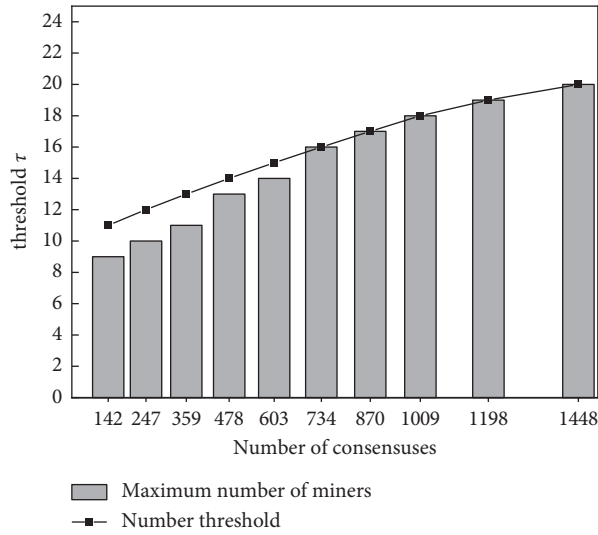


FIGURE 7: The change of the threshold τ . Each interval of the abscissa represents the current number of rounds when the number of thresholds has changed.

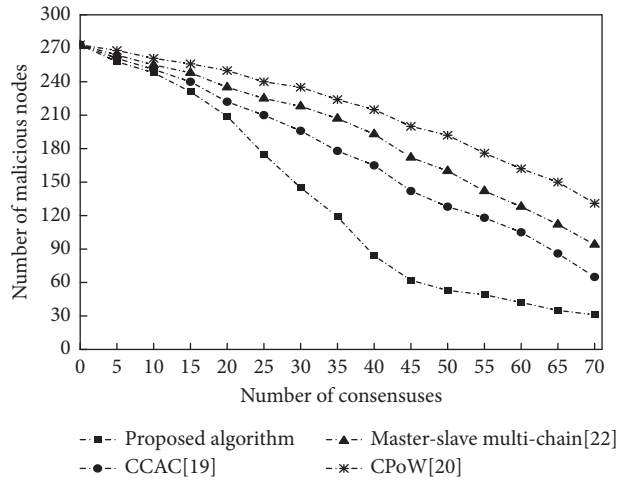


FIGURE 8: The number change of malicious nodes, recorded in 70 consensus.

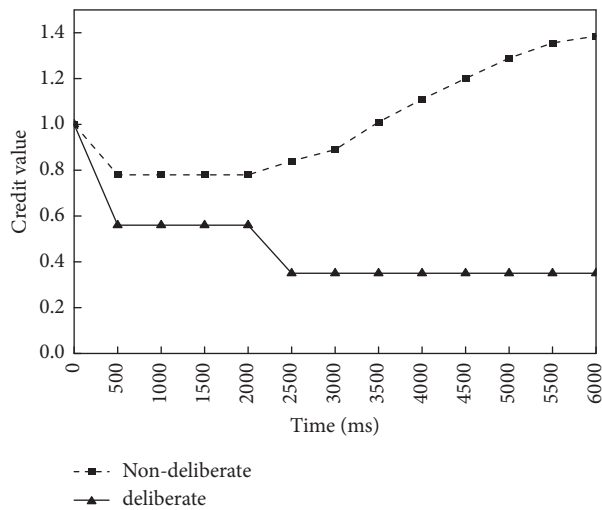


FIGURE 9: The credit value change of malicious nodes, recorded in a 6000 ms period.

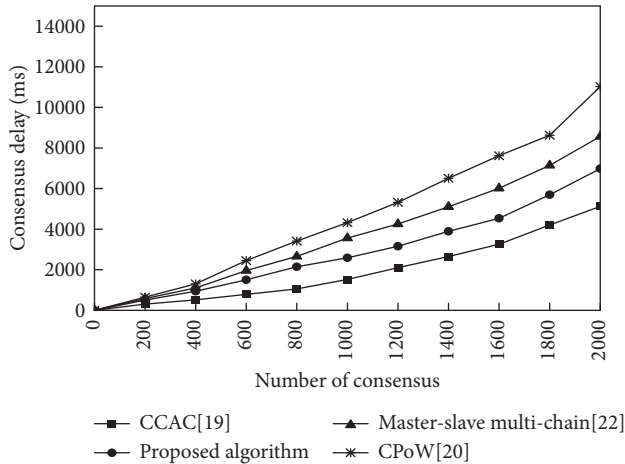


FIGURE 10: Comparison of consensus delay, recorded in 2000 consensuses.

affect the normal operation of the entire blockchain system. Compared with the proposed algorithm in this paper, CPoW is more difficult to solve the hash problem with the increase of blockchain length. Therefore, CPoW will consume a lot of computing power and have a high consensus delay. Because the master-slave multichain algorithm is based on the PoS algorithm, compared with CPoW, it saves a lot of energy consumption without mining. However, compared with the algorithm proposed in this paper, its consensus process is more complex and prone to bifurcation. Thus, the consensus delay is higher than that of the proposed algorithm in this paper.

5.5. Limitation. It can be seen from the results of the above experiments that the algorithm proposed in this paper can suppress the “oligarchy” nodes and deal with the deliberate nodes very well, but there are still some limitations. In this part of the credit evaluation model, the evaluation indicators set are not complete enough, so the evaluation of the nodes may not be comprehensive enough. This is a relatively limited point, and there is room for improvement in the future.

6. Conclusion

This paper proposed a semifragile consortium blockchain consensus algorithm based on credit space. According to the working situation of the node, we designed a credit evaluation model to calculate the credit value of the node and allocated the credit space. Besides, we proposed a randomly select mechanism for the accounting node based on the credit space, which solved the problem of insufficient incentive in the consensus algorithm and ensured the randomness of the node to become an accounting node. The experimental results show that the consensus mechanism in this paper has randomness while ensuring credit incentive; it enhances the security of the algorithm. In addition, it is more reasonable for the node penalty mechanism and has better performance in consensus efficiency, which is suitable for

consensus in the consortium blockchain. Nonetheless, the algorithm still has shortcomings in the determination of malicious nodes and the design of the “custody” time equation of the semifragile hierarchical punishment mechanism. The next step will continue to conduct in-depth research on these two aspects.

Data Availability

The coding data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

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

Acknowledgments

This work was supported by the National Natural Science Foundation of China (nos. 61762046 and 62166019), the Science and Technology Research Project of Education Department of Jiangxi Province (no. GJJ209412), and the National Innovation and Entrepreneurship Training Program for College Students (no. 201913434005).





References

- [1] S. Nakamoto, “Bitcoin: a peer-to-peer electronic cash system,” 2008, <https://bitcoin.org/bitcoin.pdf>.
- [2] S. Zhang and .-H. Lee, “Analysis of the main consensus protocols of blockchain,” *ICT express*, vol. 6, no. 2, pp. 93–97, 2020.
- [3] M. X. Du, X. F. Ma, Z. Zhang, W. Xiangwei, and C. Qijun, “A Review on Consensus Algorithm of blockchain,” in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 2567–2572, Banff, AB, Canada, October 2017.
- [4] U. Satapathy, B. K. Mohanta, S. S. Panda, S. Sobhanayak, and D. Jena, “A secure framework for communication in internet of things application using hyperledger based blockchain,” in *Proceedings of the 2019 10th international conference on computing, communication and networking technologies (ICCCNT)*, pp. 1–7, Kanpur, India, July 2019.
- [5] M. Mrinal, A. Garg, V. D. Maikandavel, and A. Panja, “Blockchain secured vehicle tracking using wireless sensor network,” *Ilkogretim Online*, vol. 20, no. 1, pp. 2472–2480, 2021.
- [6] G. T. Nguyen and K. Kyungbaek, “A survey about consensus algorithms used in blockchain,” *Journal of Information processing systems*, vol. 14, no. 1, pp. 101–128, 2018.
- [7] Y. A. Min, “A study on performance evaluation factors of permissioned blockchain consensus algorithm,” *Jouranl of Information and Security*, vol. 20, no. 1, pp. 3–8, 2020.
- [8] F. Zeng, Q. Chen, L. Meng, and J. Wu, “Volunteer assisted collaborative offloading and resource allocation in vehicular edge computing,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3247–3257, 2021.
- [9] F. Zeng, Y. Chen, L. Yao, and J. Wu, “A novel reputation incentive mechanism and game theory analysis for service caching in software-defined vehicle edge computing,” *Peer-to-Peer Networking and Applications*, vol. 14, no. 2, pp. 467–481, 2021.

- [10] S. M. H. Bamakan, A. Motavali, and A. Babaei Bondarti, "A survey of blockchain consensus algorithms performance evaluation criteria," *Expert Systems with Applications*, vol. 154, no. 9, Article ID 113385, 2020.
- [11] S. Pahlajani, A. Kshirsagar, and V. Pachghare, "Survey on Private Blockchain Consensus Algorithms," in *Proceedings of the International Conference on Innovations in Information and Communication Technology (ICIICT)*, pp. 1–6, Chennai, India, April 2019.
- [12] A. Zhang and X. Lin, "Towards secure and privacy-preserving data sharing in e-health systems via consortium blockchain," *Journal of Medical Systems*, vol. 42, no. 8, pp. 1–18, 2018.
- [13] H. Thomas and P. Alex, "Verifiable Anonymous Identities and Access Control in Permissioned Blockchains," Massachusetts Institute of Technology, 2016, <http://arxiv.org/abs/1903.04584>.
- [14] H. Huang, P. Zhu, F. Xiao, X. Sun, and Q. Huang, "A blockchain-based scheme for privacy-preserving and secure sharing of medical data," *Computers & Security*, vol. 99, no. 12, pp. 102010–102023, 2020.
- [15] S. J. Alsunaidi and F. A. Alhaidari, "A Survey of Consensus Algorithms for Blockchain Technology," in *Proceedings of the 2019 International Conference On Computer And Information Sciences (ICCIS)*, pp. 1–6, Sakaka, Saudi Arabia, April 2019.
- [16] Y. Wu, P. Song, and F. Wang, "Hybrid consensus algorithm optimization: a mathematical method based on POS and PBFT and its application in blockchain," *Mathematical Problems in Engineering*, vol. 2020, 2020.
- [17] X. Zheng, W. Feng, M. Huang, and S. Feng, "Optimization of PBFT algorithm based on improved C4. 5," *Mathematical Problems in Engineering*, vol. 2021, 2021.
- [18] D. Wang, C. Jin, H. Li, and M. Perkowski, "Proof of activity consensus algorithm based on credit reward mechanism," in *Proceedings of the International Conference on Web Information Systems and Applications*, pp. 618–628, Guangzhou, China, September 2020.
- [19] S. Z. Li, L. Huang, X. H. Deng, Z. Q. Wang, and H. W. Liu, "Consortium chain consensus algorithm based on credit," *Application Research of Computers*, vol. 38, no. 8, pp. 2284–2287, 2021.
- [20] Z. Wang, Y. L. Tian, Q. X. Li, and X. YANG, "Proof of work algorithm based on credit model," *Journal on Communications*, vol. 39, no. 8, pp. 185–198, 2018.
- [21] F. Li, K. Liu, J. Liu, Y. Fan, and S. Wang, "DHBFT: dynamic hierarchical Byzantine fault-tolerant consensus mechanism based on credit," in *Proceedings of the Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference On Web And Big Data*, pp. 3–17, Tianjin, China, August 2020.
- [22] H. Z. Liu, S. S. Li, W. L. Lv, and S. J. Wei, "Master-slave multiple-blockchain consensus based on credibility," *Journal of Nanjing University of Science and Technology*, vol. 44, no. 3, pp. 325–331, 2020.
- [23] A. Bugday, A. Ozsoy, S. M. Öztaner, and H. Sever, "Creating consensus group using online learning based reputation in blockchain networks," *Pervasive and Mobile Computing*, vol. 59, no. 10, pp. 101056–101070, 2019.
- [24] J. Huang, L. Kong, G. Chen, L. Cheng, K. Wu, and X. Liu, "B-IoT: Blockchain Driven Internet of Things with Credit-Based Consensus Mechanism" in *Proceedings of the 2019 IEEE 39th International Conference On Distributed Computing Systems (ICDCS)*, pp. 1348–1357, Dallas, TX, USA, October 2019.

Research Article

A Novel Consensus Algorithm Based on Segmented DAG and BP Neural Network for Consortium Blockchain

Xiaohong Deng ^{1,2,3}, Kangting Li ², Zhiqiang Wang ² and Huiwen Liu ¹

¹School of Electronics and Information Engineering, Gannan University of Science and Technology, Ganzhou 341000, China

²School of Information Engineering, Jiangxi University of Science and Technology, Ganzhou 341000, China

³Key Laboratory of Cloud Computing and Big Data, Ganzhou 341000, China

Correspondence should be addressed to Huiwen Liu; 9320070286@jxust.edu.cn

Received 13 February 2022; Accepted 21 March 2022; Published 8 April 2022

Academic Editor: Jiewu Leng

Copyright © 2022 Xiaohong Deng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Currently, because of the excellent properties of decentralization, hard tamperability, and traceability, blockchain is widely used in WSN and IoT applications. In particular, consortium blockchain plays a fundamental role in the practical application environment, but consensus algorithm is always a key constraint. Over the past decade, we have been witnessing the obvious growth in blockchain consensus algorithms. However, in the existing consortium blockchain consensus algorithms, there is a limited characteristic of scalability, concurrency, and security. To address this problem, this work introduces a new consensus algorithm that is derived from a directed acyclic graph and backpropagation neural network. First, we propose a partitioned structure and segmented directed acyclic graph as data storage structure, which allows us to improve scalability, throughput, and fine-grained granularity of transaction data. Furthermore, in order to provide the accuracy of node credit evaluation and reduce the possibility of Byzantine nodes, we introduce a novel credit evaluation mechanism based on a backpropagation neural network. Finally, we design a resistant double-spending mechanism based on MapReduce, which ensures the transaction data are globally unique and ordered. Experimental results and security analysis demonstrate that the proposed algorithm has advantages in throughput. Compared with the existing methods, it has higher security and scalability.

1. Introduction

In 2008, Satoshi Nakamoto proposed Bitcoin for the first time, and then the digital currency represented by Bitcoin has developed rapidly and became an integral part of the digital finance field. Blockchain as a core technology of Bitcoin has received extensive research attention, expanding to other fields besides finance; for example, in Wireless Sensor Network (WSN) [1–3], almost all the scenarios of WSNs require an efficient and accurate localization process. However, the main disadvantage of the existing frameworks and algorithms is that they are not so much significant with the trust of the beacon nodes, which are an integral part of WSNs. This must be ensured for localization. At the same time, it is obvious to have the malicious nodes in the hostile environment of WSN operations. Literature [3] provides a

secure localization scheme based on trust assessment for WSNs using blockchain technology. While addressing this challenge, it also provides a trust-based framework for secure localization. In the field of the Industrial Internet of Things (IIoT), blockchain technology has become a new way to solve cooperative trust issues. Using blockchain, a tamper-proof system can be built, which can be used as an audit tool for hardware products of the industrial Internet of Things from chip to whole equipment. Blockchain improves the productivity and operational efficiency of IIoT through a smart contract, allowing machines to manage themselves. To apply blockchain technology in IIoT, one of the main issues is to solve the security and efficiency problems of consensus protocols. Literature [4] proposes a reputation-based incentive module that can be implemented on state-of-the-art PoX protocols and can make the PoX protocols achieve

better consensus states. This scheme can effectively encourage the cooperative behavior of the nodes in IIoT, which can benefit the network. At the same time, in the field of vehicle edge computing [5, 6], due to the tamper-proof, traceable, and distributed storage properties of blockchain, it can provide a reliable storage environment for its data. However, what affects the blockchain's real entry into practical application is the blockchain's infrastructure. Currently, a general classification divides blockchain into three categories, including public blockchain, consortium blockchain, and private blockchain. Among them, the consortium blockchain is widely welcomed due to the features of controlled access node identity and decentralized storage. However, the existing consortium blockchain architecture still suffers from low performance of consensus mechanism, which leads to low operational efficiency of the whole blockchain system.

Consensus algorithm is the core technology in blockchain, used to solve how to reach an agreement between distributed nodes [7]. Most of the existing mainstream consortium blockchain consensus algorithms are based on Byzantine Fault Tolerance [7] (BFT). Although the BFT algorithm can solve the "Byzantine General" problem, it also brings new problems. For example, as the number of nodes increases, the communication complexity of PBFT [8] will increase rapidly, which will directly lead to a decrease in system throughput. In addition, the PBFT elects the leader node by trying to serially switch the number, which will greatly increase the possibility of a malicious node becoming a leader node and lead to poor system security. For this reason, HotStuff [9] uses threshold signature and star communication topology to solve the problem of high communication complexity of PBFT, but there is the possibility of malicious nodes becoming leader nodes, and the HotStuff algorithm adopts a star topology structure, so its algorithm performance is limited by leader node's hardware resource. Besides, Raft [10] uses the communication structure of master-slave nodes to make the system throughput higher under small-scale nodes, but there are problems such as non-Byzantine resistance and throughput limited by the hardware resources of the leader node.

In order to solve the problem of malicious nodes doing evil, researchers proposed to introduce a reputation mechanism into BFT algorithms. For example, Alex et al. [11] proposed a reputation consensus algorithm against "Sybil" attacks, which effectively reduces the risk of malicious nodes becoming leader nodes. However, the algorithm still does not solve the problem that the larger the size of the node, the lower the throughput, and the calculation of the reputation model is relatively fixed, so the scalability is poor. DE Oliverira et al. [12] proposed an adaptive hedging algorithm to change the calculation of the reputation model in a dynamic way. However, the algorithm has poor activity; that is, when the leader node goes down, the algorithm will run abnormally and it is difficult to restore to the normal operating state. In addition, the underlying data structure used by the above consensus algorithm is the traditional chain structure, which will limit the throughput of the system. Directed acyclic graph [13] (DAG) has the

characteristics of adapting to high concurrency, which can better solve this problem and greatly increase the throughput of the system. Although traditional DAG has better performance in throughput, it also has problems such as double-spending and high retrieval complexity. In summary, the existing mainstream consortium blockchain consensus algorithms have problems such as poor throughput, poor scalability, and security risks.

In response to the above problems, we propose a consensus algorithm for consortium blockchain with low communication resource consumption, reliable performance, and easy scalability. In the following, the main contributions of this paper are mentioned:

- (1) Propose a node reputation evaluation mechanism based on BP neural network, which can measure the node's credit value more accurately. And use the reputation value to select the accounting node to reduce the risk of malicious nodes becoming accounting nodes. In addition, select multiple nodes with higher reputation value to enter the committee to verify the accuracy of transaction messages to improve security.
- (2) Design a partition structure for nodes to join and exit freely, which is used to solve the problem of poor scalability. Introduce a segmented DAG as the data storage structure solves the problem of poor throughput while reducing the complexity of retrieval in traditional directed acyclic graphs and improves the fine-grained nature of data operations by using transactions as the basic storage unit.
- (3) A resistant double-spending mechanism based on MapReduce [14] is proposed to ensure that the data is globally unique. At the same time, it solves the poor scalability of the BFT consensus algorithm and the double-spending problem in DAG [15].

This paper is organized as follows: In Section 2, we briefly summarize the related knowledge, including the existing mainstream consensus algorithms, BP algorithm, MapReduce, and DAG. In Section 3, we describe the consensus algorithm proposed in this paper in detail, including the credibility evaluation model, the underlying topology, the election of the organizing committee members, and the consensus process. The experimental results and analysis are described in Section 4, which introduces the experimental environment, performance test results, security analysis, and comparison with other pieces of literature. Finally, in Section 5, we outline conclusions and future research directions.

2. Related Knowledge

2.1. Consensus Algorithm. Generally, a good consensus algorithm can greatly save the time required for the synchronization of the ledger data of the blockchain network nodes, thereby improving the operating efficiency of the entire blockchain system. At present, the consensus algorithms used in the blockchain framework can be roughly divided into three categories: the first is based on the

attribute value proof of the node itself, typical representative algorithms such as proof of work [16] (PoW) of the Bitcoin system, Proof of Stake [17] (PoS) of Nextcoin, and the Proof of Delegated [18] (DPoS) of EOS v1.0 [19]; the second is the node voting system, typical representative algorithms such as PBFT of Fabric v0.6 [20]; the third is the Paxos-like consensus algorithm, typical representative consensus algorithms such as Paxos [21–23] and Raft of Fabric v1.4.4.

In particular, Paxos is the origin of traditional distributed algorithms. Many consensus algorithms are based on their evolution and development, and Raft also evolved from this idea. However, both Paxos and Raft do not have anti-Byzantine characteristics, so they are not suitable for the blockchain environment. The PoW algorithm proposed by Satoshi Nakamoto solves the Byzantine problem but uses the method of solving the hash problem to select the accounting nodes, which has the problem of wasting computing power and lower throughput. Afterward, PoS proposes solutions to the problems of excessive waste of PoW resources and slow block generation time, but there are problems such as harmless attacks and long-range attacks. Moreover, DPoS introduces a proxy mechanism, and token holders can elect supernodes as accounting representatives to solve the problem of oligarchy. But when abnormal super nodes appear, the election system cannot solve the problems caused by abnormal and malicious nodes in time.

PBFT is a method of state machine copy replication to solve the BFT problem. In PBFT, all replica states are converted in the view, and the leader node selection method is the master node view number modulo the number of nodes. That is, a round of consensus is to take a view as a cycle and switch views when the consensus is completed. Although PBFT reduces the communication complexity in the BFT problem from exponential to polynomial, the nodes need to continuously broadcast message; as the scale becomes larger, the network performance requirements are higher, and the efficiency becomes slower and slower. More precisely, when the leader node in PBFT switches frequently, the complexity will reach $O(n^3)$, so this method is only suitable for consortium blockchain. As such, for the problem of high communication complexity, HotStuff adopts the way that all messages are received and distributed by the leader node, which reduces the average communication complexity of PBFT from $O(n^2)$ to $O(n)$. In addition, the view switching and consensus process in PBFT are executed separately. If the views are frequently switched, the communication complexity is as high as $O(n^3)$. However, HotStuff relies on a synchronized clock to integrate switching views and the consensus process. When the verifier raises an objection during the consensus process, that is, there is a problem in the authentication procedure, the view will be switched after the time expires. Both of the above BFT algorithms' leader nodes are elected according to the view number order for switching. Unfortunately, this way will have the problem of poor security.

Interestingly, the emergence of reputation-based consensus algorithms has solved the problem of leader node election. Literature [12] proposed a new model to replace the proof of work to form a consensus group. The proposed model uses an adaptive hedging method to calculate

reputation values for nodes that want to participate in the consensus committee and select nodes with higher reputation values for the consensus committee to reduce the chance of evil nodes. But this method does not take into account the problem of algorithm activity.

Besides, Liu et al. [24] proposed a consensus mechanism of reputation proof, which solves the problem that the verification node in the blockchain is vulnerable to attack and loses the ability to distinguish honest nodes. By constructing all nodes into a directed weighted graph, the largest weakly connected branch is taken as the set of verification nodes with the highest positivity. Moreover, the "Leader-Rank" algorithm [25] is used to calculate the contribution degree of the verification node according to the out-degree and in-degree of the node. Afterward, it calculates the reliability of the number of valid blocks, valid votes, invalid blocks, and invalid votes created by the verification node and finally calculates the weighted sum of the contribution and reliability to obtain the final comprehensive reputation. Based on the comprehensive reputation ranking, the leader in the current round of BFT is selected. This reputation proof mechanism can effectively solve the problem of verifying nodes being manipulated by attacks, but it has the problem of unbalanced weight distribution between contribution and reliability and potential reputation oligarchs. Among them, literature [26] proposed a PoS consensus mechanism based on reputation. Aiming at the problems of low performance and low security of existing blockchain, a master-slave multichain structure is designed to ensure that the block information cannot be tampered with through the anchoring of the master-slave chain. At the same time, a joint consensus mechanism for the main chain is proposed, which uses multiple consensus mechanisms to calculate together in the main-slave chain. However, the use of different algorithms on the master-slave chain will produce a barrel effect, which leads to the problem of high concurrency difficulty.

In summary, from the above consensus algorithm, we can see that the election methods of accounting nodes are mainly randomly selected, fixed election, or election based on some attribute values. In particular, a good election method of accounting nodes can increase the security of the whole system. Therefore, the election method of the accounting node becomes particularly critical. Generally, the election method not only considers the weight distribution of attributes but also needs to take into account the performance of the entire network.

2.2. Backpropagation Algorithm (BP) [27]. As the core of deep learning [28–30], the BP algorithm's function is to calculate the error based on the forward output and then conduct backpropagation to adjust the weights in the neural network based on the error. In brief, the core idea of the BP algorithm is to use gradient descent to find the most suitable weights and bias values so that the fit of the function is optimal. The BP neural network model is shown in Figure 1.

As can be seen, the model is divided into an input layer, a hidden layer, and an output layer. The connection of neurons between layers is the weight w , and the target of

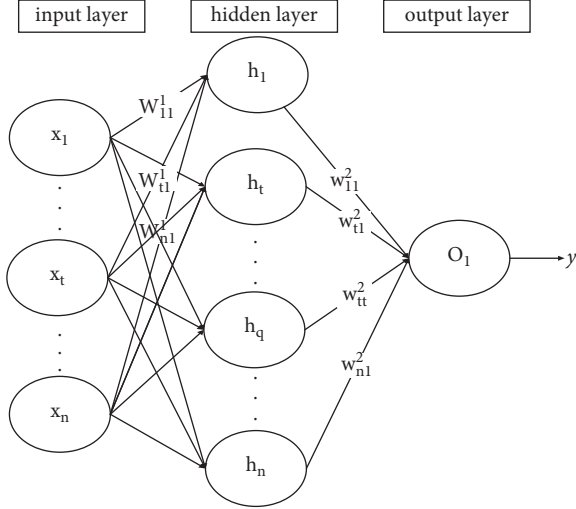


FIGURE 1: Model of BP neural network.

network training is to adjust w to the optimal value. More precisely, we take the adjustment of the first weight w_{11}^1 of the first layer as an example. First, the output of O_1 needs to be calculated forward, as shown in formula (1).

$$y = \sum_{i=1}^n h_i \times w_{i1}^2. \quad (1)$$

In formula (1), h_i is the neural unit of the hidden layer, and the calculation formula for h_1 is shown as follows:

$$y = \sigma \left(\sum_{j=1}^n X_j \times w_{j1}^1 \right). \quad (2)$$

Among them, σ is the activation function, and the common activation functions are ReLU, tanh, sigmoid, and so on. The activation function makes the neural network have a nonlinear fitting ability. X_j is the input value.

And then, the calculation formula of the loss value is shown as follows:

$$\text{loss} = (y - \hat{y})^2. \quad (3)$$

Among them, the meaning of loss is to measure the difference between the predicted value y and the true value \hat{y} . The adjustment method of w_{11}^1 is shown in formula (4).

$$w_{11(2)}^1 = w_{11(1)}^1 - lr \times \Delta w_{11}^1, \quad (4)$$

where lr is a real number between 0 and 1 and $w_{11(2)}^1$ represents the second-round adjustment value of w_{11}^1 . The result is the first round w_{11}^1 minus the gradient multiplied by lr , and the calculation method is shown in formula (5).

$$\Delta w_{11}^1 = \frac{\partial \text{loss}}{\partial O_1} \frac{\partial O_1}{\partial h_1} \frac{\partial h_1}{\partial w_{11}^1}. \quad (5)$$

Finally, repeat formula (4), and after multiple iterations of updating, w_{11}^1 completes the adjustment. Because the neural network can independently adjust the advantages of characteristic nodes, we use it to predict the reputation value of each node.

2.3. MapReduce. Undoubtedly, when an information system has a huge amount of data, the data needs to be divided and processed separately. MapReduce is a computing architecture that uses functional programming ideas to divide a calculation into two calculation processes, Map and Reduce. More precisely, MapReduce can divide a large computing task into multiple small computing tasks and then assign each small computing task to the corresponding computing node in the cluster and always track the progress of each computing node to decide whether to reexecute the task. Finally, the calculation results on each node are collected and output. Its working principle is shown in Figure 2. In the chaotic and disorderly color data, it first performs the Map operation on the color data, then splits the key-value data structure, and sends it to different computing units performing the Reduce operation, which mainly counts and sorts the number and types of colors. Finally, the results of the types and quantities of colors are summarized. Since MapReduce has the characteristics of multinode collaboration and deduplication of data, this paper will use this architecture to solve the poor scalability of consensus algorithms and the double-spending problem in DAG.

2.4. Directed Acyclic Graph. Particularly, the emergence of DAG has transformed the ledger form from a single chain to a directed acyclic graph pattern, avoiding the limitations of serialized writes that exist in single chains and allowing the ledger to support high concurrency. In fact, in the blockchain represented by Bitcoin, except for the genesis block, each block has one and only one predecessor block and one successor block, and the blocks form a single chain. Conversely, if two blocks are reserved at the same time, it will cause the blockchain to fork. According to the longest chain principle, only one block will be retained on the main chain, and the other will be discarded. However, in a distributed ledger based on DAG, as shown in Figure 3, the basic unit of each ledger can reference one or more predecessor units and can be referenced by one or more subsequent units at the same time. As such, this structural difference enables DAG-based ledgers to support concurrent operations, and multiple nodes can add transactions or block units to the ledgers at the same time, thereby greatly improving system throughput. However, although the traditional DAG has better performance in throughput, there are other problems. For example, using the Iota [31] framework with DAG as the bottom layer, transactions in this framework require a large number of Markov Monte Carlo random walks and a small amount of proof of work to add to the ledger. This method is too complicated, and the transactions in Iota are not globally ordered, so it cannot completely resist the double-spending problem, and the retrieval time is long. In addition, another HashGraph [32] framework that uses a parachain DAG uses a gossip algorithm and virtual voting to confirm that the entire transaction is globally ordered in an asynchronous environment, and there will be a long voting process in the virtual voting stage. For this reason, it will result in more rounds of voting to confirm that the transaction is valid and reliable. In summary, the existing DAG framework has

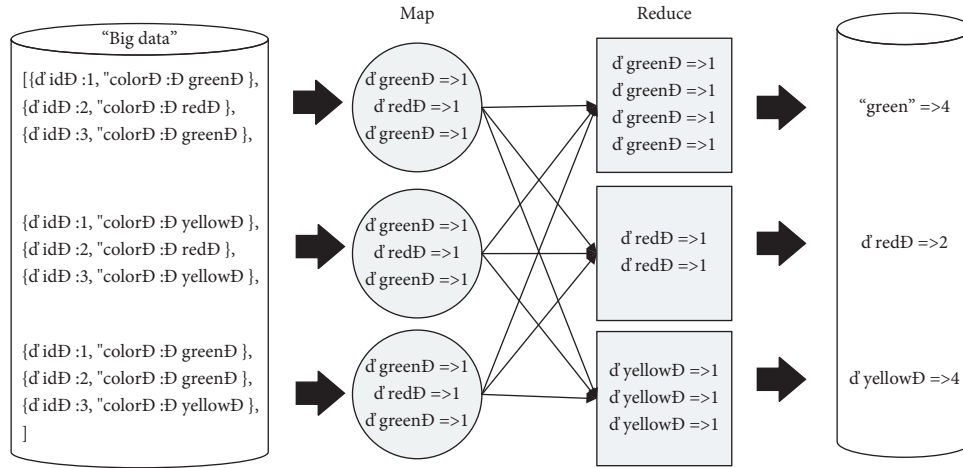


FIGURE 2: Working principle of MapReduce.

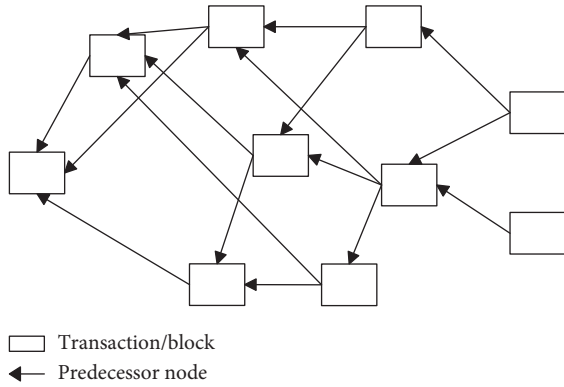


FIGURE 3: Architecture of directed acyclic graph.

problems such as double-spending, high search complexity, and long time to add to the ledger. We propose a segmented DAG to solve the above problems, the details of which will be introduced in subsequent chapters.

2.5. Attack Model. The blockchain system is a network composed of cooperation between nodes. In order to subdivide the functions of the blockchain system, the blockchain is usually divided into a six-layer structure, which includes data layer, network communication layer, consensus layer, incentive layer, contract layer, and application layer. The proposed algorithm in this paper mainly involves the data layer, network communication layer, and consensus layer. The data layer is mainly based on a certain data structure to store data, the network communication layer is responsible for broadcasting and verifying transactions, and the purpose of the consensus layer is to allow nodes to coordinate and cooperate to achieve a consensus on data consistency. This paper mainly discusses the attack methods involved in these three layers. Common attack methods [33] are as follows:

- (1) Double-spending attacks: the main situations of common double-spending attacks are as follows:
 - (1) When a new transaction enters the block to

obtain a sufficient number of confirmations, and the length of the attacker's side chain exceeds the main chain, the attacker's side chain becomes the main chain. So, the first transaction initiated by the attacker was determined to be invalid, and the double-spending attack was successful. (2) For the Naive DAG, when the transaction enters the ledger, it relies on the PoW algorithm to calculate the weight to choose to eliminate the double-spending transaction, but there is a situation that is not timely.

- (2) 51% attack: for this paper, an attacker who has more than half of the reputation value is 51% attack.
- (3) Solar eclipse attack: an attacker tries to isolate a group or one node, isolate it from communication with other nodes, and prevent it from obtaining the latest world state.
- (4) Denial of service attack: the node deliberately does not actively participate in the calculation process. In this paper, it can be considered that the calculation process forwards heartbeat packets too much, does not respond or repeatedly sends double-spending transactions, and almost does not send normal transactions. In brief, the proportion of normal transactions that is less than 50% is considered a denial of attack.

This paper applies the above attack model to the security considerations of the proposed algorithm.

3. Proposed Algorithms

3.1. Reputation Evaluation Model. In the traditional consortium blockchain consensus algorithm, the nodes participating in the consensus generally switch sequentially or randomly, which easily leads to malicious nodes deliberately doing evil. In the following research, the reputation consensus algorithm has been proposed, node's reputation is measured by the behavior of nodes participating in the consensus, and the accounting nodes are selected in turn by the size of the reputation value. In this way, it better solves the problem of

nodes doing evil. However, most of them use simple linear formulas to evaluate the reputation of nodes. Unfortunately, linear algorithms cannot effectively extract the behavioral characteristics of nodes, so they cannot make full use of the characteristics and assign corresponding weights. For this reason, we propose a reputation evaluation model based on the BP algorithm. Since the neural network can approximate the properties of any function from arbitrary precision [34], it has strong feature extraction capabilities. We use some attributes of the node as feature vectors to consider the reputation value of the node. The reputation value is used to quantify the possibility that the node is a Byzantine node. The node attributes are shown in Table 1.

The characteristics of nodes in the blockchain mainly include two aspects. (1) Security features: they contain the number of maliciously sending false transaction messages E_Tx , the node's historical reputation value H_Rep , and the node's online time On_Time . (2) Performance characteristics: they contain the throughput TPS of the node, the number of effective forwarding transactions C_ETx of the node, and the average delay forwarding time D_AFT , where E_Tx , C_FTx , and C_ETx are discrete values, and the rest are continuous values.

In this paper, we constructed a four-layer BP neural network, in which the input layer contains six neurons, corresponding to six features, and the two-layer hidden layer contains 1,000 neurons. The activation function uses the ReLU function, and the last output layer uses the sigmoid function to map the reputation value between 0 and 1. Construct reputation evaluation as shown in formula (6).

$$rep_j^i = \begin{cases} F(X_j^{i-1})rep_j^{i-1} \geq 0 \text{ and evil} = 0, \\ -\sum_{i=0}^{\text{latest}} rep_j^i rep_j^{i-1} \geq 0 \text{ and evil} = 1, \\ rep_j^{i-1} + \alpha e^{-\text{count}} rep_j^{i-1} < 0, \end{cases} \quad (6)$$

where $evil=0$ means that the node is not doing evil and $evil=1$ means that the node is doing evil. rep_j^i is the reputation value of the j node in round i , and X_j^{i-1} is the feature vector of the j node in round $i-1$. $X_j = [E_Tx, H_Rep, On_Time, TPS, C_ETx, D_AFT]$; X_j needs to undergo dimensionless processing. $F(x)$ is the neural network model, and when rep_j^{i-1} is greater than or equal to 0 and the node is not doing evil, use $F(x)$ to predict the reputation value of the node. However, when rep_j^{i-1} is greater than or equal to 0 and the node has malicious behavior, the node's reputation is the negative number of the node's accumulated reputations from the first time to the latest; in contrast, when rep_j^{i-1} is less than 0, the node's reputation is calculated by adding an exponential function related to the number of evils and the reputation of the previous round, where $\alpha \in (0, 1)$, and count is the number of evils.

3.2. Underlying Topology. Compared with the serial processing of chain structure, DAG is more suitable for natural high concurrency. We propose a segmented DAG to solve the problem of high retrieval complexity and long time for transactions to be added to the ledger. As shown in Figure 4,

the shaded block is the organizing committee block, which contains the organizing committee's group signature, timestamp, reputation record, and hash pointer group. The white blocks represent ordinary nodes, which contain the signatures, timestamps, transaction information, and hash pointer groups of ordinary nodes. The numbers in the grey and white blocks represent the sequence. The black block is a fast index block array, which contains a timestamp, a hash pointer, and the block hash of its own block.

The genesis block 0 is fixedly generated as organizing committee block, and subsequent blocks are connected to it by hash pointers. The connection method is to randomly select the nearest n timestamp transaction data. A fixed organizing committee block is generated every fixed time or the corresponding number of blocks. The transaction information between two organizing committee blocks is verified and deduplicated by the former. In addition, in terms of retrieval, compared to the retrieval of all transactions in the original DAG, we divide the DAG according to the time dimension, only relying on the black block to quickly index according to the timestamp. As such, the search complexity is greatly reduced.

DAG has two forms in physical structure: one is an adjacency matrix, and the other is an adjacency list. Since the adjacency matrix is a sparse matrix, it will waste a lot of space, so the storage form of the adjacency list is adopted. As shown in Figure 5, the leftmost is an array of fast index blocks, which contains timestamps and hash pointers, the grey part in the middle is the committee block, and the white part is the original block. Both the committee block and the white block contain transaction information, hash value, and hash pointer and are connected to the corresponding transaction information block.

Aiming at the problem of double-spending that is difficult to eliminate in DAGs, we propose a resistant double-spending mechanism based on MapReduce, as shown in Figure 6. First, use n organizing committee nodes to accept the client's transaction operations, then divide all ordinary nodes into several partitions, and select a number of ordinary nodes with higher reputation values or organizing committee nodes in different partitions for transaction message statistics. Furthermore, the ordinary node sends the transaction message to the organizing committee node. In addition to verifying the correctness of the transaction, the organizing committee node not only verifies the transaction's correctness but also removes the double-spending transaction message according to the reputation value of the node. Finally, the results are summarized to the leader node for verification. Afterward, the leader packs the transaction message, sends it to the remaining nodes of the organizing committee, and sends it to the other ordinary nodes through the gossip protocol [35].

3.3. Consensus Algorithm Process

3.3.1. Algorithm for Election of Organizing Committee Members. The method for electing members of the organizing committee is shown in Algorithm 1. The members

TABLE 1: Characteristic attributes of node.

| Characteristic symbol | Explanation | Value range |
|-----------------------|---|----------------|
| E_Tx | Number of maliciously sending false transaction messages | $[0, +\infty]$ |
| H_Rep | Node's historical reputation | $[-\infty, 1]$ |
| On_Time | Node's online time | $[0, +\infty]$ |
| C_ETx | The number of effective forwarding transactions by the node | $[0, +\infty]$ |
| D_AFT | Average delay forwarding time | $[0, +\infty]$ |
| TPS | Node's throughput | $[0, +\infty]$ |

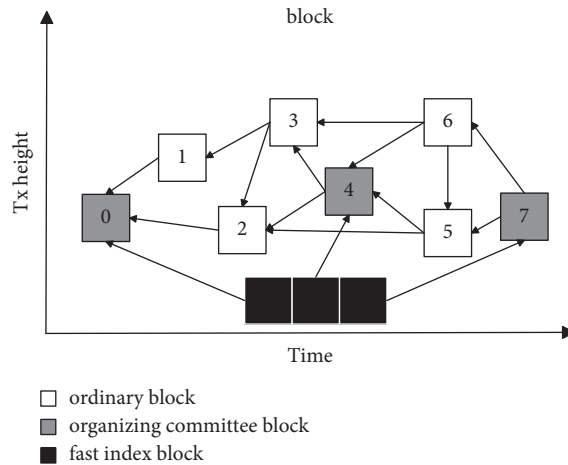


FIGURE 4: Topology diagram of segmented DAG.

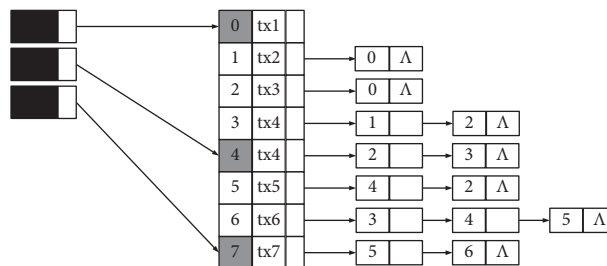


FIGURE 5: The storage structure of the segmented DAG.

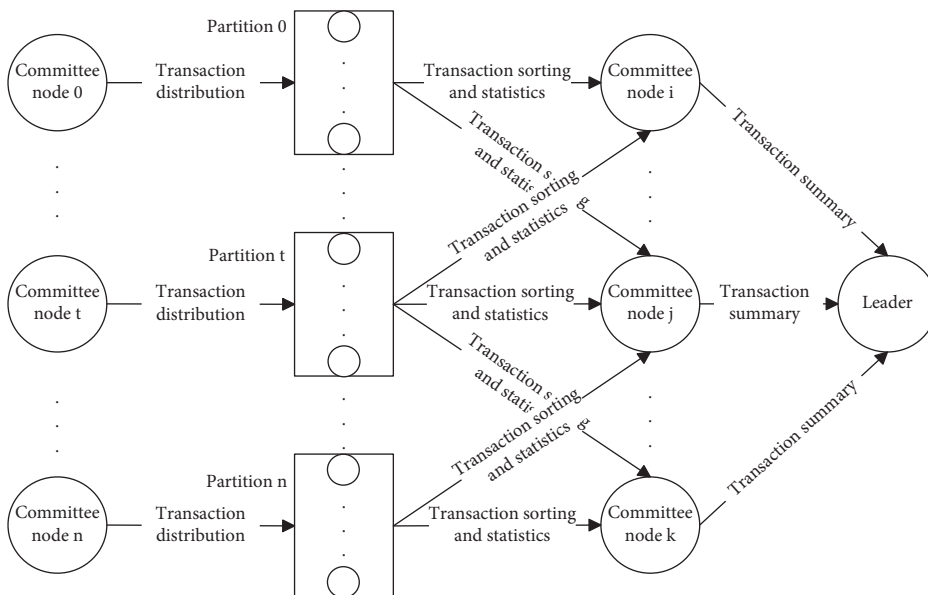


FIGURE 6: Resistant double-spending mechanism based on MapReduce.

who enter the committee in the first round are fixedly selected. Besides, the subsequent election of the organizing committee can be described as follows. First, input the characteristics into formula (6), and obtain the reputation of each round of nodes; second, rank the reputation, and the leader of the organizing committee uses a verifiable random function [36] (verifiable random functions (VRF)) to generate verifiable random numbers, which are used to randomly select nodes with higher reputation values. Finally, the nodes in the group verify whether the random number and the binomial distribution pass. If the verification of the VRF function fails or overtime, the organizing committee is requested to perform random sampling again and regenerate the random number through VRF; if it passes, the organizing committee will synchronize the data of random numbers and select several nodes to enter the organizing committee.

3.3.2. Consensus Process. In this paper, nodes are divided into ordinary nodes and organizing committee nodes, and the reputation value of the node is obtained by the reputation model. The overall consensus process is shown in Algorithm 2. First, the organizing committee randomly elects a leader node and preselects a backup leader node. Generally, the organizing committee selects the backup leader with the highest reputation value. The backup leader node is normally responsible for collecting transaction messages and monitoring the status of the leader and follower nodes. Once the leader node goes down or acts maliciously, the backup node starts to take over. The remaining nodes are follower nodes, which are responsible for collecting and verifying transaction data and sorting them. After a period of time, the results are returned to the leader node. Second, according to the results, the leader node removes the double-spending data according to the entry degree and distributes the relevant information to the organizing committee nodes to wait for a reply. Last, if more than 1/2 of the reputation node replies are received (the reputation value of 1/2 is the maximum error tolerance threshold of this algorithm, which will be proved by subsequent experiments), then gossip protocol broadcasts to other ordinary nodes to achieve global unification. In contrast, if not received or timed out, immediately switch the leader to enter the next round. Or when the term of the committee is reached, all the members of the organizing committee will be replaced; else go directly to the next round.

3.3.3. Resistant Double-Spending Mechanism Based on MapReduce. The mechanism used in this paper to remove double-spending transactions based on MapReduce is shown in Algorithm 3. First, n organizing committee nodes monitor transaction messages, and the backup leader section group is responsible for monitoring the status of the organizing committee nodes, scheduling, and removing some malicious nodes. Second, each organizing committee node will divide the transaction message into m parts and then send the m parts to m organizing committee nodes with higher reputation value for map. The map operation will

traverse the transaction message and return the data in k-v format. The key is the hash of the transaction message, and the value is the reputation value of the node that sent the transaction. These m organizing committee nodes do the MD5 operation of the key and modulate r and then send the k-v to the corresponding r organizing committee nodes. Third, after these r organizing committee nodes receive the corresponding transaction message, they will merge the messages, shuffle the messages according to the size of the value, remove the double-spending operation, and then return the deduplicated transaction message to leader. Finally, the leader node receives the message of the organizing committee node, performs verification, packs, and returns the DAG structure transaction message.

4. Experimental Results and Analysis

4.1. Experimental Environment. In order to test the performance of the consensus algorithm proposed in this paper, we designed several simulation experiments. The operating system selected for the experimental environment is centos, using Python and PyTorch to write consensus algorithms, using flask to write web program interfaces, and using docker containers to load web programs to simulate nodes. In addition, we used Alibaba Cloud server to simulate the experiment of multimachine multinode stress test, used siege to carry out stress test and throughput, and set the concurrency to 280 transactions/s. It should be noted that, without a special statement, we set the number of organizing committee nodes to one-half of the number of summary points, set the appointment period of the organizing committee to 2 min, and set the delay waiting threshold to be random between 3 s and 4 s. In order to better carry out quantitative experiments, we used transactions with tags. The transaction tags are normal, empty, malicious, and double spend, which represent normal transactions, empty transactions (heartbeat packets), malicious transactions, and double spend transactions, respectively. The purpose of the experiment is to test the performance and security of the consensus algorithm in an ideal environment or a Byzantine environment. The performance includes the error of the neural network, the ability to process transactions in the consensus process, and the delay in completing the consensus. On the other hand, the purpose of security is to test the system operation under the condition of a hypothetical adversary attack.

4.2. Performance Test

4.2.1. Regressor Performance. To test the accuracy of the regressor, 500 node's index data and their reputation evaluation results are selected. According to the method of the reputation evaluation model, the selected index data is first standardized to facilitate the neural network processing. More precisely, the data of 400 nodes are used as training data each time, and the data of the remaining 100 nodes are used as verification data. Besides, the learning rate is set to 0.001 in the experiment.


```

Input:  $X$  (the feature vector of the node)
Output: flag1 (whether the committee members are successfully elected)
(1)  $Vec = \text{formula}(X)$  # Get the reputation value of the node in each round
(2)  $Sorted\_Vec = \text{Sort}(Vec)$  # Reputation ranking
(3) flag1 = false
(4)  $Random\_number, proof = \text{VRF}(seed)$  # VRF function to generate random numbers and evidence
(5) if( $\text{Verify}(Random) == \text{True} \&\& \text{Verify}(proof) == \text{True} \&\& \text{Time} < \text{Congfig\_time}$ ): #Verify that the random number and evidence
    are correct at the specified time
(6) flag1 =  $\text{Choose}(Random\_number)$  # Select nodes to enter the committee and set flag to True
(7) else:
(8)     Go To Step4
(9) end else
(10) end if
(11) return flag1

```

ALGORITHM 1: Election of organizing committee members.

```

Input: Random seed
Output: Consensus result flag2
(1)  $(leader, leader\_backs, follower) = \text{VRF}(seed)$  # Select leader node, backup node, and follower node
(2)  $Tx\_sorted = \text{Sort}(\text{gather\_follower}(TX))$  # Collect verification transactions and sort them
(3)  $Block = \text{MapReduce}(\text{gather\_follower}(Tx\_sorted))$  # Remove duplicate transactions and pack
(4) flag2 = false
(5) if( $\text{Collect}(Block) \geq 1/2 \text{reputation} \&\& \text{Time} < \text{Congfig time}$ ) # Collect more than half of the reputation value at a fixed time
(6)     flag2 =  $\text{Broadcast}(Block)$  # Broadcast block success is True
(7)     if  $\text{Check}(\text{Term of Service}) == \text{True}$  # Is it in the service cycle
(8)         Go to step1 # Repeat step 1
(9)     end if
(10)    else
(11)         $\text{Clean}(\text{committee})$  # Clearance Committee
(12)
(13)    end else
(14) else:
(15)     $\text{Change}(leader)$  # Switch leader node
(16)    Go to Step1 # Go back to step 1 and restart transaction collection
(17) end else
(18) end if
(19) return flag 2

```

ALGORITHM 2: Consensus process.

```

Input: transaction message msg
Output: the transaction set block and malicious node number after deduplication
(1)  $msg\_i = \text{follower}_i.\text{watch}(msg)$   $i$  in  $\text{range}(0, n-1)$  # There are  $n$  organizing committee nodes to monitor transaction messages
(2)  $msg\_ij = \text{Devide}(msg\_i)$   $j$  in  $\text{range}(0, m-1)$  # Divide the transaction message  $msg\_i$  of each node into  $m$  parts
(3)  $key, value = \text{node}_j.\text{map}(msg\_ij)$   $j$  in  $\text{range}(0, m-1)$  # Map the transaction message, use the transaction hash as the key, and the
    value is the reputation value of the node that sent the transaction
(4)  $key\_temp = \text{md5}(key) \bmod r$  # Do the md5 operation of the key and modulate  $r$ , and send the key and value to the corresponding
    follower $_t$ 
(5)  $msg\_t = \text{follower}_t.\text{reduce}(key\_temp, value)$   $t$  in  $\text{range}(0, r-1)$  #The follower node receives the corresponding transaction
    message, merges the messages, shuffles according to the size of the value, removes the double-spending operation, and
    returns the deduplicated transaction message
(6) if  $msg\_t$  is  $db\_tx$  or  $error\_msg$ :
(7)      $nodeid = t$ 
(8) end if
(9)  $block = \text{leader}.\text{collect}(msg\_t)$  # The leader node accepts the follower node message and performs verification and packaging
(10) return block, nodeid # Return DAG transaction message

```

ALGORITHM 3: Resistant double-spending mechanism based on MapReduce.

The change of loss function is shown in Figure 7. As can be seen from the data in Figure 7, as the number of pieces of training increases, the loss function has a sharp downward trend and finally tends to a stable value of about 0.1. However, the loss on the verification set is stable at around 0.067. These results provide substantial evidence for the original assumptions that the neural network model has learned the corresponding features and the error with the true value is small.

4.2.2. Throughput Test. In order to test the throughput of different consensus algorithms in a multimachine multinode environment, four hosts are configured with 25, 50, 75, 100, 125, 150, 175, and 200 docker simulation nodes. Each consensus algorithm takes the average value of the transactions per second (TPS) of 100 rounds of consensus. That is, the total transaction volume in 100 rounds divides the time taken for 100 rounds of consensus.

The test results are shown in Figure 8. The results reveal that the four algorithms all increase the TPS when the node size is less than 100. The proposed algorithm is slightly worse than HotStuff and Raft, but better than PBFT. However, when the node size is greater than 100, the TPS of PBFT decreases rapidly, while the TPS of Raft and HotStuff grows more slowly, and the proposed algorithm grows approximately linearly and is better than the other three algorithms. This is because, in the case of a smaller scale, the star topology used by Raft and HotStuff is faster. However, as the scale becomes larger, the performance of PBFT becomes lower and lower as the communication becomes more complicated. Since HotStuff and Raft use a star topology communication method, the throughput of the entire system is limited by the IO device of the master node. In particular, when the number of nodes increases, once the traffic exceeds or reaches the maximum processing capacity of the main node's IO, TPS will begin to decrease. In contrast, the performance of the proposed consensus algorithm will increase linearly with the increase in the number of nodes. This is because the MapReduce architecture is used to process tasks on multiple nodes, which greatly weakens the hardware performance limitations of a single node.

4.2.3. Response Time. Generally, the delay of the blockchain system can be defined as the time difference between the client submitting the transaction request and the client receiving the response result. In the experiment, we tested the delays of PBFT, the proposed algorithm, Raft, and HotStuff, when the number of nodes in the whole network is 25, 50, 75, 100, 125, 150, 175, and 200, respectively. Each algorithm tests the average delay of 100 rounds of consensus results.

The result is shown in Figure 9. Figure 9 illustrates that the delay of PBFT increases sharply with the increase of the number of nodes, while the proposed algorithms, Raft, and HotStuff keep the delay low. The reason for the above phenomenon is that the PBFT communication time is $O(n^2)$, so as the number of nodes increases, the access delay will increase greatly. Since both HotStuff and Raft use a star

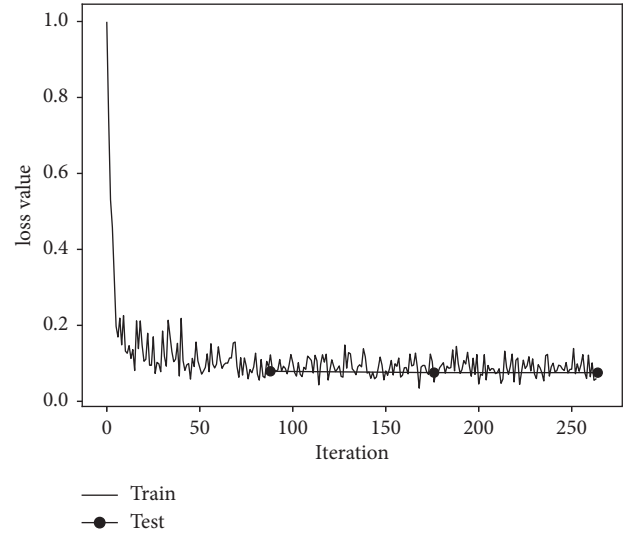


FIGURE 7: Change curve of loss function.

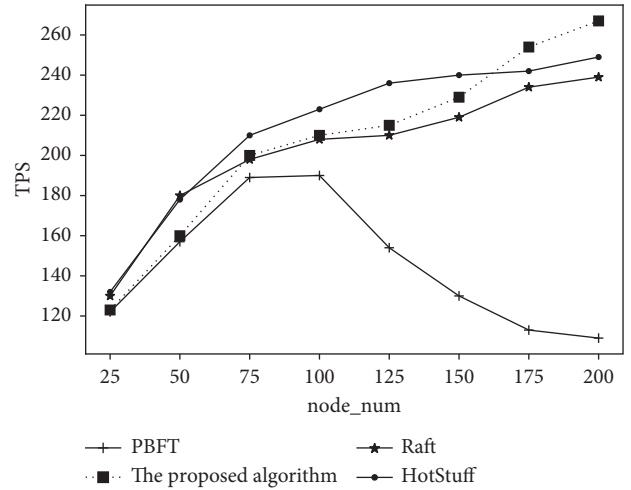


FIGURE 8: Throughput comparison.

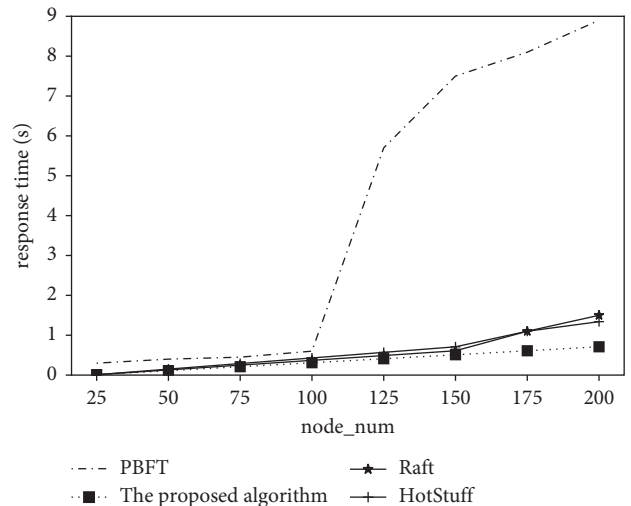


FIGURE 9: Algorithm delay comparison.

topology for communication, the client and the leader directly perform read and write operations, this method will enter a bottleneck period after being limited by hardware resources, and the delay will slowly increase in the later period. However, the algorithm in this paper has a small workload of the leader node due to the simultaneous write mechanism of the partitions. Therefore, it has lower latency.

4.2.4. Comparison of Retrieval Speed of Different Storage Structures. In this paper, the retrieval transaction scale is set between 0 and 20,000 transactions, and the purpose is to compare the retrieval speed in different storage structure scenarios of chain structure, simple DAG, and segmented DAG structure. The experimental results are shown in Figure 10.

As can be seen from Figure 10, the retrieval speed of traditional chained and naive DAGs increases linearly with the number of transactions. The segmented DAG retrieval does not follow the trend of linear growth in the number of transactions. The reason for the above phenomenon is that the chained data structure is searched in order. Even if the tree search method is used, it is limited to within the block. When searching for a transaction, the external memory needs to be transferred into the memory, so it takes longer. Simple DAG requires a BFS or DFS search method, so with the increase of transaction messages, the retrieval time will also increase accordingly. However, the segmented DAG used in this paper is stored in the form of a hash table and can be retrieved in chronological order, so the retrieval time is greatly reduced.

4.3. Security Analysis

4.3.1. Attack Model Analysis

- (1) Double spend attack. As mentioned in Section 2.5, in view of the first case, the underlying DAG topology we proposed will not have chain bifurcation, and there will be no two segmented DAGs at the same time. Only when a solar eclipse attack occurs, will the network splits make two segmented DAGs. In this case, you need to master the 50% reputation value; however, it is almost nonexistent in the subsequent experimental verification. For the second case, we designed the MapReduce architecture to ensure that the transactions in the DAG will be deduplicated and then sorted. In addition, the leader node will perform deduplication again, so the second case can also be avoided.
- (2) 51% attack and eclipse attack (reputation cumulative split attack).

The segmented DAG structure we used is still updated based on the longest DAG structure. Assuming that the solar eclipse attack is successful, there will be multiple DAGs in an asynchronous environment. This phenomenon shows the Poisson distribution [16]. At a specific time, something will happen randomly at any time. More precisely, when this

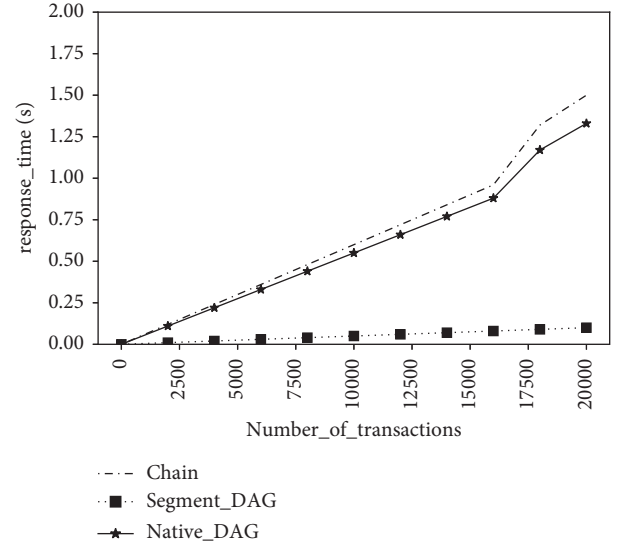


FIGURE 10: Comparison of retrieval speed of different storage structures.

time period is divided into very small time slices, it can be considered that, within each time slice, the event may or may not happen. However, it is almost impossible to consider situations that occur more than once because the time slice can be divided into small enough time slices. The Poisson distribution formula is as follows:

$$P(X = i) = \binom{n}{i} \left(\frac{\lambda}{n}\right)^i \left(1 - \frac{\lambda}{n}\right)^{n-i}. \quad (7)$$

In formula (7), when $n \rightarrow \infty$, $\binom{n}{i} / n^i \rightarrow 4/i!$, $(1 - (\lambda/n)) \rightarrow e^{-\lambda}$, it can be derived as formula (8).

$$P(X = i) = \binom{n}{i} \left(\frac{\lambda}{n}\right)^i \left(1 - \frac{\lambda}{n}\right)^{n-i}, \quad (8)$$

$$= \frac{e^{-\lambda} \lambda^i}{i!}.$$

Formula (8) represents the probability that the attacker succeeds in the i block, where $i = z$, $\lambda = q_z$, and the calculation method of q_z is shown in formula (9). Among them, q is the probability of an attacker's successful attack, and p is the probability of an honest node that normally generates a transaction block.

$$q_z = \begin{cases} 1, & \text{If the attacker is the longest chain,} \\ \left(\frac{q}{p}\right)^z, & \text{If the attacker is behind by } z \text{ blocks.} \end{cases} \quad (9)$$

We set the organizing committee node groups with 10%, 33%, 49%, and 50% reputation values to exist in an asynchronous network environment and set z from 0 to 50 and then try to attack the segmented DAG in the state of the real environment.

The results of the experiment are shown in Figure 11. As can be seen, the node group with 10% and 30% reputation values may have a larger drop and approximate to 0 as more blocks fall behind; the attacker with 50% reputation value will have a successful attack. In addition, the attacker with 49% reputation value tends to 0 as more blocks are created. As such, we set the threshold of reputation value to 50% in Algorithm 2 because mastering 51% reputation value will attack successfully and split the network. However, this is almost impossible. First of all, in the procedure of the election of the group committee, any normal node can enter the group committee, and controlling the group committee is almost to control all the nodes. Secondly, the communication between the nodes is local P2P, so the link is multichannel, and the possibility of splitting the network with the increase of the group committee nodes is almost 0, so this situation is almost impossible to exist.

(3) Denial of service attack.

In the experiment, we tested the reputation value changes of four nodes, node 1 to node 4, with different attribute values, and the results are shown in Figure 12.

More specifically, set the adjustable parameter α to 1, and set node 1 and node 2 to maintain good and medium characteristic attribute values, respectively. Besides, node 3 transforms from relatively medium attribute values (approximates attribute values of node 2) to better attribute values (approximates attribute values of node 1), and node 4 is set as a malicious node and begins to deny service in the fourth round. As the results of Figure 12 show, the reputation value of node 1 and node 2 has no obvious change, while the reputation value of node 3 is slowly increasing. In the fourth round, node 4 sends malicious information, and its reputation value changes to the inverse of the sum of its historical reputation values. The reason for the stable changes in the reputation value is because the neural network predicts the reputation value with high accuracy, and the reputation value results brought by similar feature attribute values are all approximately the same. Particularly, nodes with good characteristic attribute values will be given corresponding good scores, so the increase or decrease of reputation value will not be obvious, so it has good stability. In contrast, for the calculation of the reputation value of a malicious node, its reputation value changes directly to the inverse of the sum of its historical reputation values. In addition, as the number of malicious actions increases, an exponential function is used to calculate the follow-up reputation value, and the follow-up reputation growth rate is approximately zero. Undoubtedly, nodes with a reputation value less than 0 will not have the right to participate in the organizing committee and sort and count transactions.

4.3.2. Continuous Switching of Malicious Leader Nodes.

In the experiment, 25 nodes were set up to test the stability of the throughput of each consensus algorithm under the condition of continuous switching of the leader node.

As shown in Figure 13, the throughput of the PBFT and Raft fluctuates drastically, and the lowest is only 91TPS. In

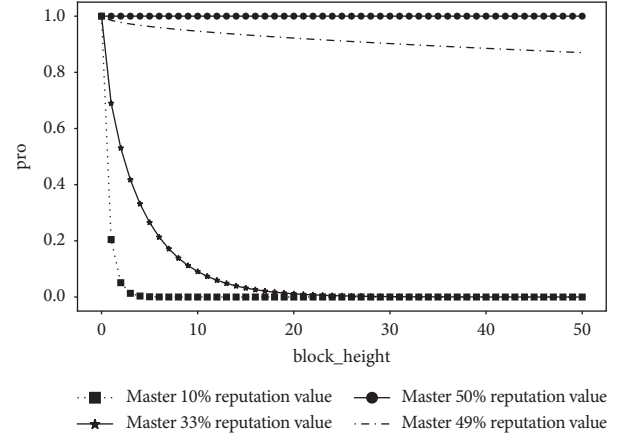


FIGURE 11: Cumulative reputation attacks.

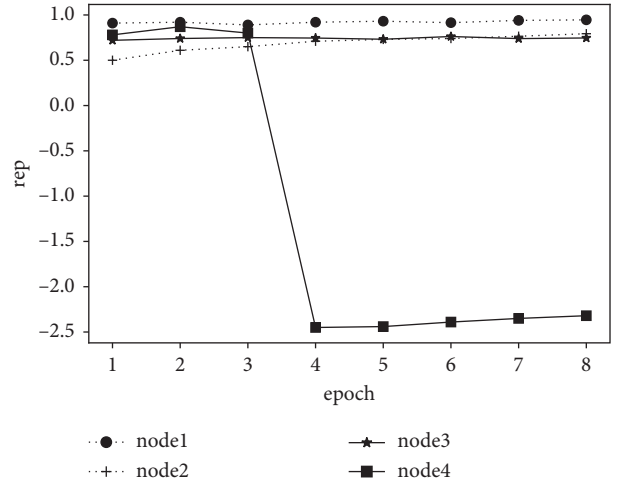


FIGURE 12: Change curve of node reputation.

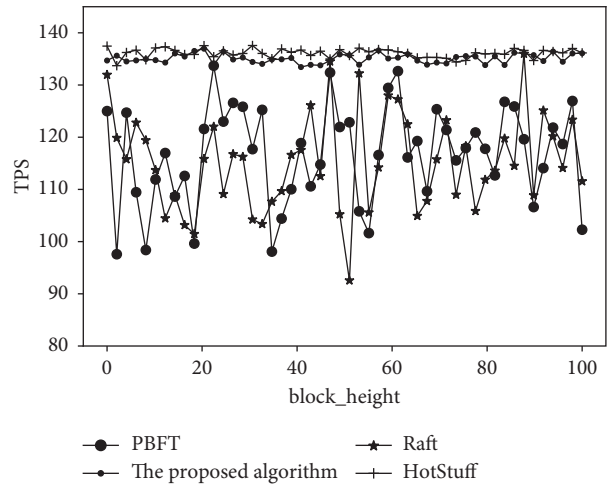


FIGURE 13: Continuous leader switching.

contrast, the throughput of HotStuff and the proposed algorithm has good stability and has been maintained at around 130TPS. This is because the traditional PBFT

algorithm switches the view after the voting is completed. Due to the multistage point-to-point communication, the complexity is $O(n^2)$. In particular, if the leader switches continuously, the communication complexity will be as high as $O(n^3)$. As Raft, it switches the leader node based on the heartbeat packet, frequent switching will lead to a continuous election process. However, HotStuff integrates view switch and transaction broadcast communication and uses the pipeline block topology, so it has a better stability. In this paper, the direct switching mechanism between the backup node and the organizing committee node is used to reduce the risk of malicious downtime and switching of the leader node, and the node with a high reputation value is selected as the organizing committee node to further ensure stability.

4.3.3. Expulsion of Malicious Nodes. In the experiment, 100 nodes were set up to test the rate of the proposed consensus algorithm to eliminate malicious nodes, and the running time of the system was set to ten minutes. Malicious nodes are randomly distributed in the network, and the ratio is set to 10%, 20%, 30%, 40%, and 50% of the network nodes. Moreover, they sent double-spending and error messages with a probability of 25%, 50%, 75%, and 100%, respectively. The experimental results are shown in Figure 14. The results indicate that no matter what the ratio of malicious nodes is, the eviction rate will rise sharply from 0.2 to 0.98 before the ratio of malicious messages reaches 0.75, until 100% eviction, which has nothing to do with the ratio of malicious nodes. The reason can be further described in Figure 15. With the proportion of malicious message sending being 0.25, the proportion of transaction messages is only about 0.42 if the proportion of valid transactions with double-spending is more than 0.5. This is because, in this paper, we consider that as long as the percentage of valid transactions is more than 0.5, even if a double spend transaction is sent, it is considered normal, so the eviction rate is low. Once a malicious transaction is detected, the node will be directly eliminated, and the eviction rate will increase as the proportion of malicious messages increases, so the final approach is approximately 100%.

The results of the expulsion velocity experiment are shown in Figure 16. As can be seen that when the malicious message ratio is 0.25, the node expulsion rate is about 13 s. It should be noted that the larger the scale of the malicious node is, the longer the time it takes. When the malicious message ratio is 0.5, the expulsion rate is about 30 s, and when the ratio of malicious transactions is 0.75 and 1, the time consumption will decrease. The main reason is that when the proportion of malicious messages is 0.25, 42% of nodes will send double-spending messages. However, the inspection mechanism needs to wait for the effective ratio to be lower than 0.5 before removing them, so waiting time is required. When the proportion of malicious messages is 0.5, the nodes have some randomness, and the possibility of malicious nodes sending double-spending transactions is stronger, which results in higher time-consuming. In particular, when the proportion of malicious messages is 0.75 and 1, malicious nodes are more likely to send malicious

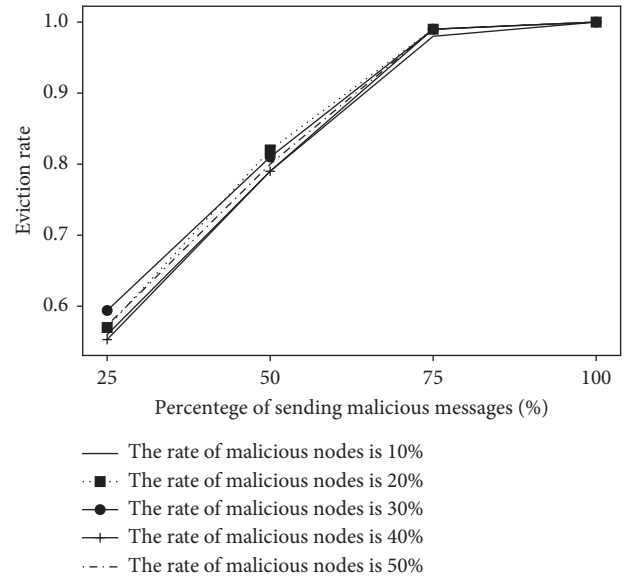


FIGURE 14: Rate of exclusion of malicious nodes.

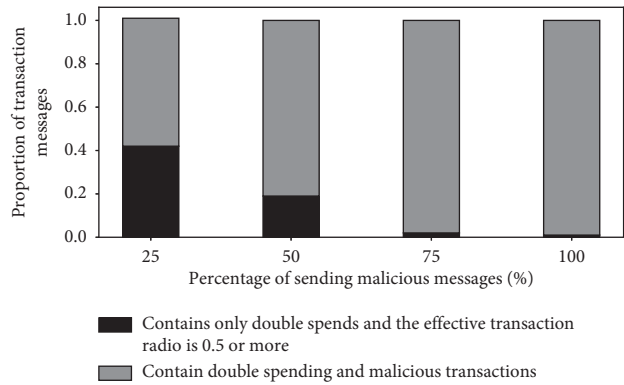


FIGURE 15: Malicious message ratio.

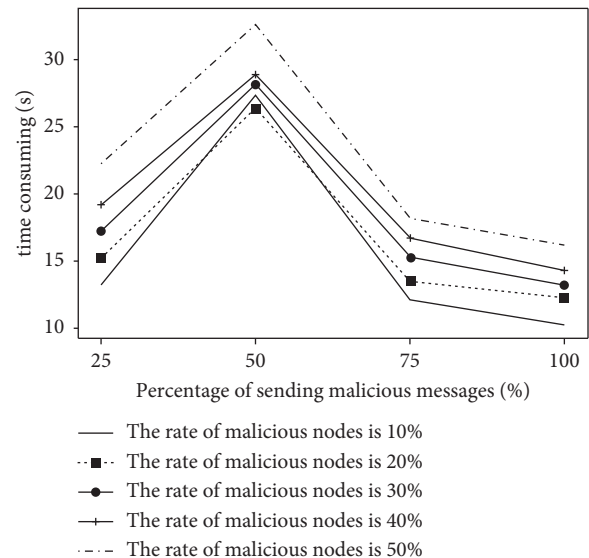


FIGURE 16: Average velocity of eviction of malicious nodes.

TABLE 2: Comparison results with existing literature.

| | Scalability | Fault tolerance (%) | Underlying topology | Active | High concurrency | Resistant double-spending | Communication complexity | Fine-grained |
|------------------------|-------------|---------------------|---------------------|--------|------------------|---------------------------|--------------------------|--------------|
| Literature [12] | Low | 49 | Chain | No | No | High | $O(n^2)$ | Block |
| Literature [9] | High | 33 | Chain | Yes | Yes | High | $O(n)$ | Block |
| Literature [26] | High | 49 | Chain | No | No | High | $O(n^2)$ | Block |
| Literature [24] | Low | 33 | Chain | Yes | No | High | $O(n^2)$ | Block |
| The proposed algorithm | High | 49 | DAG | Yes | Yes | High | $O(n^2)$ | Transaction |

messages directly. As long as the system detects malicious messages, it will directly eliminate them, so the time is shorter.

4.3.4. Comparison with Other Pieces of Literature.

Generally, consensus algorithms are usually compared from three aspects, namely, the degree of decentralization, security, and performance. Among them, the degree of decentralization is the scale of nodes participating in the consensus, the security is mainly anti-Byzantine ability and resistant double-spending, and the performance mainly considers factors such as algorithm activity, throughput, communication complexity, and scalability. The proposed algorithm is compared with existing similar literature, and the comparison results are shown in Table 2. Literature [9], literature [12], literature [26], and literature [24] all use the traditional chained bottom topology, and the smallest unit of operation is a block, so their algorithms have good resistant double-spending ability. In terms of scalability, literature [9] adopts a pipelined block topology structure, and there is no mandatory sequence relationship between the generation of blocks, which enhances the scalability. In literature [26], multiple algorithms coexist, making nodes increase and exit free. However, literature [12] and literature [24] are both based on traditional BFT, so there is no improvement in scalability. In terms of fault tolerance, literature [9] and literature [24] are based on the maximum fault tolerance of BFT, that is, 33%. Particularly, the difference is that the threshold of literature [24] is 33% of the total reputation value; literature [12] uses a new examiner mechanism to control the threshold at 49%. Besides, the maximum threshold of literature [26] for a mixture of multiple proof algorithms is 49%. Regarding the complexity of communication, literature [12], literature [26], and literature [24] all use point-to-point propagation, so the communication complexity is $O(n^2)$. Literature [9] uses a star topology, so the communication complexity is $O(n)$. In terms of concurrency, literature [12], literature [33], and literature [24] all use the traditional chain structure, so the concurrency is not high. In contrast, literature [9] adopts the pipeline mechanism, and there is no necessary time limit for the generation of the front and back blocks, so the concurrency may be higher. Compared with the above literature, we use DAG as the underlying topology, which naturally supports high concurrency and uses the MapReduce architecture to split tasks into multiple small tasks and send them to other nodes for sorting and deduplication, increasing the

availability of ordinary nodes. Therefore, the scalability and resistant double-spending ability of the proposed algorithm is relatively high. Although the communication complexity of this paper is $O(n^2)$, the scale is just between the organizing committee nodes, the broadcast adopts the gossip protocol, and the communication complexity is $O(n)$. In addition, this paper utilizes the similar functions of pacemakers to make the algorithm active and support semiasynchronous.

5. Conclusion

In summary, the consortium blockchain architecture has become the first choice for blockchain applications. However, limited by the traditional chain structure, the throughput of the blockchain has been greatly affected. Although the appearance of DAG increases the system throughput in a concurrent manner, it brings new problems of high algorithm complexity and double-spending. For this reason, we propose a high concurrency and scalable consortium blockchain consensus algorithm, which designs a segmented DAG structure to increase system throughput while reducing the time complexity of global retrieval. The resistant double-spending mechanism based on the MapReduce architecture effectively ensures the global uniqueness of the transaction. The consensus algorithm proposed in this paper is suitable for the parallel and collaborative computing of large-scale sensors in the Internet of Things, which improves the security of computing and the scalability of device clusters. In addition, the credible nodes are elected through the reputation model based on the BP neural network, which reduces the risk of malicious nodes doing evil. The simulation experiment results also prove that the algorithm in this paper has better performance. However, detailed theoretical proof was not obtained, and the following three aspects are worthy of in-depth study: (1) The underlying topology: the existing DAG has high concurrency and parallel characteristics better than the chain structure, but the security is poor, and the double-spending problem is difficult to solve. (2) Use the idea of division and autonomy to fragment the blockchain network, thereby reducing the communication scale of the network and increasing the speed of consensus. (3) Most of the existing consensus algorithms tend to adopt hybrid consensus algorithms, which will be a trend. Maybe, the integration of proof-like algorithms and BFT technology is a very meaningful research direction.

Data Availability

The coding data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

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

Acknowledgments

This work was supported by the National Natural Science Foundation of China (nos. 61762046 and 62166019) and the Science and technology research project of Education Department of Jiangxi Province (no. GJJ209412).

References

- [1] J. Leng, G. Ruan, P. Jiang et al., "Blockchain-empowered sustainable manufacturing and product lifecycle management in industry 4.0: a survey," *Renewable and Sustainable Energy Reviews*, vol. 132, Article ID 110112, 2020.
- [2] J. Leng, P. Jiang, K. Xu et al., "Makerchain: a blockchain with chemical signature for self-organizing process in social manufacturing," *Journal of Cleaner Production*, vol. 234, pp. 767–778, 2019.
- [3] G. Rekha, K. Gulshan, and A. Mamoun, "A secure localization scheme based on trust assessment for WSNs using blockchain technology," *Future Generation Computer Systems*, vol. 125, no. 11, pp. 221–231, 2021.
- [4] E. K. Wang, Z. Liang, C.-M. Chen, S. Kumari, and M. K. Khan, "PoRX: a reputation incentive scheme for blockchain consensus of IIoT," *Future Generation Computer Systems*, vol. 102, no. 1, pp. 140–151, 2020.
- [5] F. Zeng, Q. Chen, L. Meng, and J. Wu, "Volunteer assisted collaborative offloading and resource allocation in vehicular edge computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3247–3257, 2021.
- [6] F. Zeng, Y. Chen, L. Yao, and J. Wu, "A novel reputation incentive mechanism and game theory analysis for service caching in software-defined vehicle edge computing," *Peer-to-Peer Networking and Applications*, vol. 14, no. 2, pp. 467–481, 2021.
- [7] M. Seyed, M. Amirhossein, and B. Alireza, "A survey of blockchain consensus algorithms performance evaluation criteria," *Expert Systems with Applications*, vol. 154, no. 16, pp. 113385–113406, 2020.
- [8] H. Sukhwani, J. M. Martinez, and X. Chang, "Performance modeling of PBFT consensus process for permissioned blockchain network (hyperledger fabric)," in *Proceedings of the 36th Symposium on Reliable Distributed Systems (SRDS)*, Hong Kong, China, September 2017.
- [9] I. Abraham, D. Malkhi, and K. Nayak, "Sync HotStuff: simple and practical synchronous state machine replication," in *Proceedings of the 41th Symposium on Security and Privacy*, Piscataway, May 2020.
- [10] S. Rahul, W. Mohammad, and G. Prosanta, "A blockchain based secure communication framework for community interaction," *Journal of Information Security and Applications*, vol. 58, no. 3, pp. 102790–102803, 2021.
- [11] A. Biryukov and F. D. ReCon, "sybil-resistant consensus from reputation," *Pervasive and Mobile Computing*, vol. 61, no. 1, pp. 1574–1192, 2020.
- [12] M. T. d. Oliveira, L. H. A. Reis, D. S. V. Medeiros, R. C. Carrano, S. D. Olabarriaga, and D. M. F. Mattos, "Blockchain reputation-based consensus: a scalable and resilient mechanism for distributed mistrusting applications," *Computer Networks*, vol. 179, no. 10, pp. 107367–107380, 2020.
- [13] A. Bakhtiar, F. Syahirul, and D. Thanh, "Big data directed acyclic graph model for real-time COVID-19 twitter stream detection," *Pattern Recognition*, vol. 123, no. 3, pp. 108404–108416, 2022.
- [14] J. Dean and S. Ghemawat, "MapReduce," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.
- [15] K. Lyudmila, K. Dmytro, and N. Andrii, "Decreasing security threshold against double spend attack in networks with slow synchronization," *Computer Communications*, vol. 154, no. 6, pp. 75–81, 2020.
- [16] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system," 2022, <https://bitcoin.org/bitcoin.pdf>.
- [17] NXT, "Nxt Whitepaper," 2022, <https://nxtwiki.org/wiki/Whitepaper:Nxt>.
- [18] BitShares, "Delegated proof of stake," 2022, <http://docs.bitshares.org/bitshares/dpo8.html>.
- [19] E. WhitePaper, "A next-generation smart contract and decentralized application platform," 2022, <https://github.com/ethereum/wiki/wiki/WhitePaper>.
- [20] S. Singh, "Hyperledger Fabric WhitePaper," 2022, <https://github.com/hyperledger/fabric>.
- [21] L. Lamport, "The part-time parliament," *ACM Transactions on Computer Systems*, vol. 16, no. 2, pp. 33–169, 1998.
- [22] L. Lamport, "Paxos made simple," *ACM SIGACT News*, vol. 32, no. 1, pp. 18–25, 2001.
- [23] L. Lamport and M. Massa, "Cheap Paxos," in *Proceedings of the 2004 International Conference on Dependable Systems & Networks*, Florence, Italy, June 2004.
- [24] N. A. Lin, Z. H. Chen, and G. K. Liu, "Mechanism for proof-of-reputation consensus for blockchain validator nodes," *Journal of Xidian University*, vol. 47, no. 5, pp. 61–66, 2020.
- [25] Q. Li, T. Zhou, L. Lü, and D. Chen, "Identifying influential spreaders by weighted LeaderRank," *Physica A: Statistical Mechanics and Its Applications*, vol. 404, no. 2, pp. 47–55, 2014.
- [26] H. Liu, S. Li, and W. Lv, "Master-slave multiple-blockchain consensus based on credibility," *Journal of Nanjing University of Science and Technology*, vol. 44, no. 3, pp. 325–331, 2020.
- [27] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [28] X. Zhu, Y. Luo, A. Liu, M. Z. A. Bhuiyan, and S. Zhang, "Multiagent deep reinforcement learning for vehicular computation offloading in IoT," *IEEE Internet of Things Journal*, vol. 8, no. 12, pp. 9763–9773, 2021.
- [29] M. Chen, T. Wang, S. Zhang, and A. Liu, "Deep reinforcement learning for computation offloading in mobile edge computing environment," *Computer Communications*, vol. 175, no. 11, pp. 1–12, 2021.
- [30] Y. Liu, Y. X. Lan, and B. Y. Li, "Proof of Learning (PoLe): empowering neural network training with consensus building on blockchains," *Computer Networks*, vol. 2011, no. 2, pp. 108594–108603, 2021.
- [31] W. F. Silvano and R. Marcelino, "Iota Tangle: a cryptocurrency to communicate Internet-of-Things data," *Future*

- Generation Computer Systems*, vol. 112, no. 12, pp. 307–319, 2020.
- [32] Hedera, “The swirls hashgraph consensus algorithm: fair, fast, byzantine fault tolerance,” 2022, <https://docs.hedera.com/guides/core-concepts/hashgraph-consensus-algorithms>.
 - [33] Y. Wen, F. Lu, Y. Liu, and X. Huang, “Attacks and countermeasures on blockchains: a survey from layering perspective,” *Computer Networks*, vol. 191, no. 8, pp. 107978–107994, 2021.
 - [34] K. Hornik, M. Stinchcombe, and H. White, “Multilayer feedforward networks are universal approximators,” *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.
 - [35] T. Yu and J. Xiong, “Distributed consensus-based estimation and control of large-scale systems under gossip communication protocol,” *Journal of the Franklin Institute*, vol. 357, no. 14, pp. 10010–10026, 2020.
 - [36] S. Micali, M. Rabin, and S. Vadhan, “Verifiable random functions,” in *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, New York, USA, October 1999.

Research Article

Improved PBFT Algorithm Based on Vague Sets

Guangxia Xu  and Yishuai Wang

School of Software Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

Correspondence should be addressed to Guangxia Xu; xugx@cqupt.edu.cn

Received 28 October 2021; Revised 12 February 2022; Accepted 10 March 2022; Published 29 March 2022

Academic Editor: Jiewu Leng

Copyright © 2022 Guangxia Xu and Yishuai Wang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The traditional PBFT consensus algorithm has several limitations in the consortium blockchain environment, such as unclear selection of primary node, excessive communication times, etc. To solve these limitations, an improved consensus algorithm VS-PBFT based on vague sets was proposed. VS-PBFT has three phases: node partition, primary node selection, and global consensus. Firstly, the nodes of the whole network are partitioned using the consistent hashing-like consensus algorithm, and then the local primary node is selected by the primary node selection algorithm in each partition. The local primary nodes run the four-phase PBFT consensus algorithm to complete the global consensus. The analysis of the VS-PBFT consistency algorithm shows that the algorithm can improve the fault-tolerant rate and reduce communication complexity, and the algorithm is dynamic; that is, node can join and quit adaptively.

1. Introduction

In recent years, the concept of blockchain has become very common. The definition of a blockchain is that blockchain is a mixed use of the chain structure, consensus algorithms, cryptography techniques, distributed data storage, peer-to-peer transmission, and automated smart contracts [1]. It is essentially a decentralized database. The data or information stored in the blockchain have the characteristics of decentralization, tamper resistance, traceability, collective maintenance, openness, and transparency [2]. In 2008, a person with the anonym Nakamoto published “Bitcoin: a peer-to-peer electronic cash system” [3]. Bitcoin was first applied to the financial industry as a grassroots electronic money. From this time on, the blockchain has gradually become a new idea for everyone to solve limitations.

The blockchain has four core technologies: distributed ledger [4], asymmetric cryptography algorithm [5], smart contract [6], and consensus mechanism [7]. Distributed ledger refers to the fact that transaction accounting is completed by multiple nodes distributed in different places, and each node records a complete account, so they can participate in the supervision of the legality of the transaction, and they can also testify for it together [8].

Asymmetric cryptographic algorithms are used to ensure the security of the data on the blockchain and the privacy of individuals. Each node on the blockchain has a pair of public key and private key. The public key is disclosed to the nodes of the whole network. On the contrary, the private key is private to the nodes of the whole network. One node signs the transaction with the private key, and the other node verifies the signature with the public key. As early as 1994, Szabo put forward the concept of smart contracts. Szabo described smart contracts as “a series of commitments specified in digital form, including agreements between parties to fulfill these commitments” [9]. In 2013, Dickerson et al. applied smart contracts to the real system for the first time [10]. Consensus mechanism is one of the core technologies of the blockchain, which is mainly used to make the nodes of the whole network reach consensus in a distributed environment. It needs to satisfy two properties:

Consistency: All the non-Byzantine nodes in the whole network store the same data.

Validity: All the information published by the non-Byzantine nodes will eventually be recorded in their ledger by all other non-Byzantine nodes [11].

Blockchains are usually divided into three types: Public chains, Consortium chains, and Private chains. The current mainstream consensus algorithms in Public chains include Proof of Work (PoW) [12], Proof of Stake (PoS) [13], and Delegated Proof of Stake (DPoS) [14]. The most popular consensus algorithm in the Consortium chains is the PBFT (Practical Byzantine Fault Tolerance) [15] and its derivatives. There are many restrictions in the Private chains, so they generally consider the use of a strong consensus protocol Raft [16] to achieve consensus under non-Byzantine failures.

In this article, Section 2 introduces several existing consensus algorithms. Section 3 discusses the preliminary knowledge of VS-PBFT, including the basic knowledge of vague sets and the traditional PBFT consensus algorithm. Section 4 describes the VS-PBFT algorithm, including node partition, primary node selection, and global consensus, in detail. Section 5 evaluates the VS-PBFT algorithm and its advantages are pointed out. Section 6 summarizes the whole article.

2. Related Work

The consensus mechanism of the blockchain can be classified according to the type of blockchain. The most well-known consensus algorithm in Public chains is PoW. In 1998, Dwork and Naor proposed the PoW algorithm for the first time [17]. In 2008, PoW was first applied to bitcoin. The main idea of the PoW consensus algorithm is to select the node responsible for generating blocks by finding the fastest node to calculate the difficulty value. The work of PoW is the certain amount of calculations that every node needs to perform; it will take a certain amount of time to find the hash result out. The node solving the hash equation in less time can get the right to generate blocks. We need to calculate the hash value of this block through the hash value of the previous block and the random value nonce, which is the solution to the hash equation. Once the node finds the nonce, it can calculate the solution of the hash equation. There is no fixed solution for this hash equation, so we can only find the final solution through constant trial and error. This method is also called hash collision. Hash collision is a probabilistic event. The more the attempts, the faster the calculation time and the greater the collision probability.

Consortium chains meet the requirements of the enterprise level, and its application is more targeted and efficient. Therefore, Consortium chains are becoming a hot topic of the blockchain in the future. The algorithm proposed in this article is also applicable to Consortium chains. The main consensus algorithm in Consortium chains is the PBFT. In 1999, Miguel and Barbara proposed the PBFT consensus algorithm [15], which reduces the complexity of the Byzantine protocol to a polynomial level, so that the Byzantine protocol can be used in distributed systems. In Section 3, we will introduce the detailed flow of the PBFT algorithm. In the blockchain project Hyperledger, the PBFT consensus algorithm was publicly implemented for the first time [18]. Although the PBFT consensus algorithm has $3f+1$ fault tolerance and can guarantee a certain performance at

the same time, it has many limitations, such as excessively high communication times, low scalability, and unclear primary node selection. In an unstable network, the system delay of PBFT is very high. At present, the improved algorithm mainly aimed at these points. In 2009, Clement et al. proposed a new method, Aardvark, to establish a Byzantine fault-tolerant replication system [19]. When f servers and any number of clients have Byzantine failures, this method enables the system to achieve a high Byzantine failure, peak performance, and high throughput, but this method does not solve the problem of excessive communication times of the PBFT algorithm, and its system scalability is not enough. GG Gueta et al. proposed the SBFT algorithm in 2019 [20], which addressed the problem of excessive communication times in the PBFT algorithm. To reduce the communication times to a linear level, a method, which used a fast path to reduce client communication, was proposed using collectors and threshold signatures. However, the selection of the primary node is still fuzzy.

3. Preliminaries

The materials and methods section should contain sufficient details so that all procedures can be repeated. If several methods are described, it can be divided into heading sections.

3.1. Overview of PBFT. The origin of the PBFT consensus algorithm can be attributed to the Byzantine failures. The efficiency of solving Byzantine fault is improved, and the complexity of the algorithm is reduced from exponential level to polynomial level, which makes Byzantine fault-tolerant algorithm feasible in practical system applications. The PBFT consensus algorithm is the first practical algorithm in the BFT class to work under a weakly synchronous network. It has three roles: client, primary node, and replica node. After the client puts forward the transaction request, it will be immediately sent to the primary node, and the primary node initiate the transaction voting in the global network, and then the replica node and the primary node will jointly maintain the fairness of the transaction voting. When the primary node fails, the view change program will be triggered to elect a new primary node.

We will briefly introduce the overall process of the PBFT algorithm. As shown in Supplementary Figure 1, it is the process of the PBFT algorithm, and replica node 3 is a Byzantine node.

- (a) Request: At this phase, the client node sends a transaction request to the primary node.
- (b) Pre-prepare: After the primary node receives the transaction request, it will verify the request and send a pre-prepared message to the replica node if the transaction is legal, otherwise the transaction is invalid and it will be discarded.
- (c) Prepare: The replica node verifies the validity of the pre-prepared message. Once it is legal, the replica node will send the prepared message to other nodes

in the whole network and receive the prepared message from other nodes at the same time. After the node receives the prepared message, it will verify the legitimacy of the prepared message as soon as possible.

- (d) Commit: When the node receives $2f+1$ legal prepared messages, the node enters the commit phase, where f refers to the number of Byzantine nodes in the system. During the commit phase, the node will send a commit message to other nodes in the whole network.
- (e) Reply: When the node receives $2f+1$ commit messages, it will send a reply message to the client node. After the client node received $f+1$ identical reply messages, the whole consensus is completed.

The garbage collection mechanism and view change program are not the focus of this article. For more details, please read the reference [15].

3.2. Vague Sets. Most of the existing voting-based blockchain consensus algorithms only consider agreement and disagreement, but it is obviously not enough in practical application. In 1965, Zadeh proposed the concept of fuzzy sets [21]. Fuzzy sets give us a neutral option to vote, and we can use fuzzy sets to optimize the voting process in the blockchain to make it more in line with human thinking.

Fuzzy set refers to a given domain U , then a mapping from U to the unit interval $[0, 1]$ is called a fuzzy set on U or a fuzzy subset of U . The fuzzy set can be denoted as S . The mapping (function) $\mu_S(\cdot)$ or $S(\cdot)$ is called the membership function of the fuzzy set S . For each $x \in U$, $\mu_S(x)$ is called the membership degree of element x to fuzzy set S .

Gau and Buehrer further improved the theory of vague sets in 1993 [22]; they pointed out that the members of vague sets are three subintervals between $[0, 1]$. The three subintervals correspond to three kinds of information of the element ($u \in U$): favor, against, and abstentions. We can use two functions to represent a vague set S in the domain U . $t_S(u)$ is usually used to represent a truth membership function, and $t_S(u)$ is a lower bound on the grade of membership of u derived from the evidence for u . $f_S(u)$ is usually used to represent a false membership function, and $f_S(u)$ is a lower bound on the negation of u derived from the evidence against u . Both $t_S(u)$ and $f_S(u)$ are a certain number between $[0, 1]$, where $t_S(u) + f_S(u) \leq 1$.

When U is continuous, a vague set S can be denoted as

$$S = \frac{\int_U [t_S(u), 1 - f_S(u)]}{u} \quad (1)$$

When U is discrete, a vague set S can be denoted as

$$S = \frac{\sum_{i=1}^n [t_S(u_i), 1 - f_S(u_i)]}{u_i} \quad (2)$$

In general, the value of a vague set of an element can be denoted as

$$[t_S(u), 1 - f_S(u)]. \quad (3)$$

The concept of vague set mentioned above can be seen as a voting model. Assuming that S is the vague set of element u in U , the value of S is $[0.3, 0.7]$; from (3), we can calculate $t_S(u) = 0.3$, $f_S(u) = 1 - 0.7 = 0.3$. It means that the degree that u belongs to S is 0.3 and the degree that u does not belong to S is 0.3. If the total number of votes is 10, $(0.3, 0.7)$, it means that the number of votes favor is 3, the number of votes against is 3, and the number of votes abstention is 4.

Yong et al. proposed a general formula in 2008 to convert the vague sets into the final fuzzy score [23]:

$$\mu_S F = t_S(u) + \frac{1}{2} \left[1 + \frac{t_S(u) - f_S(u)}{t_S(u) + f_S(v) + 2\lambda} \right] \cdot [1 - t_S(u) - f_S(u)], \quad \lambda > 0. \quad (4)$$

In this article, we use this final score to select the primary node. We set to choose the highest final score; when the highest scores are equal, we randomly select a node as the primary node.

4. VS-PBFT Algorithm

4.1. Algorithm Overview. Blockchain technology has gained a lot of preference in the world due to its own anonymity and decentralization, and the research on its main core technology consensus mechanism is one of the most significant parts. The PBFT algorithm is widely used in Consortium chains with its own advantages. However, if the number of nodes in the used system increases, specifically, after reaching 100, the communication times of the system will rise sharply, and the selection of the primary node that plays a key role in the algorithm is unclear. Therefore, a new improved PBFT consensus algorithm based on vague sets was proposed. The overall algorithm flowchart is shown in Supplementary Figure 2.

The algorithm we proposed has three phases: node partition, primary node selection, and global consensus. Firstly, we will partition the nodes in the whole network, then we will select the primary node in each partition. The selection method of the primary node is based on the idea of vague set, so the selection of the primary node is more in line with people's thinking compared to ordinary voting-based scheme, and each partition selects one local primary node.

In the global consensus phase, we will select a global primary node among all the local primary nodes. This global primary node undertakes the similar task to the primary node in the PBFT consensus algorithm. We will reduce communication times by reducing the consensus phase. The effect of communication times makes the whole consensus process more efficient. After the consensus is completed, the new block will be added to the whole blockchain, and then the next round of consensus will begin.

4.2. Node Partition. The communication complexity of the PBFT algorithm running under the consortium blockchain condition is $O(N^2)$. In the case of large-scale nodes, the

number of communication times will increase exponentially. In a distributed system, partition is mainly for improving the scalability and availability of the system. In 1997, Karger and others of the Massachusetts Institute of Technology proposed consistent hashing algorithm. This article will use its idea to partition nodes.

Through node partition, The N nodes of the whole network are divided into k groups. Each group is represented by n_k and k is the group number. We use the hash value of IP corresponding to each node as the unique identifier of each node.

Firstly, we create a hash function H with a value space $[0, 2^{32} - 1]$. We organize the whole hash value space into a virtual circle, and the whole space is organized in a clockwise direction, that is, 0 and $(2^{32} - 1)$ coincide at zero. Thereafter, we randomly generate k points on the hash ring, and we need to continue to randomly generate k points until k mutually exclusive points are generated, which means that we randomly divide the hash ring into k areas. We name these k areas $1, 2, \dots, i, \dots, k$, respectively. In the next step, the IP corresponding to each node uses the same hash function H to calculate the hash value and determine the location of this node on the ring. Assuming that this position is in the i^{th} area, then this node is divided into the i^{th} area. If the calculated hash value is equal to the value of a certain boundary, we put it in the smaller area.

We assume that N is 4 and k is 3, which means that we need to generate three mutually exclusive hash values and divide the hash ring into three areas; and we calculate the hashes of N node IP and divide them into corresponding area. As shown in Supplementary Figure 3, three areas: area1, area2, and area3 are generated. IP0, IP2, and IP3 are divided into area2, area3, and area1, respectively. Since the hash value of IP1 is the same as that of random hash 2, IP1 is assigned to area2.

This situation occurs during the partition process, where there are too many nodes in one area, and too few nodes in the other. In those circumstances, the local primary node selected by the partition with uneven node distribution cannot represent the node of the whole network. In this situation, we introduce a virtual node mechanism, that is, calculate multiple hashes for each certain random hash, and each calculated hash is used as a new random hash point, called a virtual node. This can be achieved by adding a number at the back of the IP. According to the regional partition formed between virtual nodes, the number of virtual nodes is usually set to 32 or even larger, so the nodes can be relatively evenly distributed.

4.3. Primary Node Selection. When the partition is completed, it means that nodes with similar IP hash values have been allocated to the same area. Next, we need to run the primary node selection to elect the local primary node. All nodes in one area vote for the most suitable node to be the local primary node. We added the option of abstention in the voting process, and used the general model transformed vague set to obtain a comprehensive score, and the highest comprehensive score became the local primary node. The

flowchart of primary node selection is shown in Supplementary Figure 4.

The primary node selection algorithm is as follows:

- (1) All nodes in the same area will vote for other nodes in the same area within the specified time. There are three choices of favor, against, and abstention, and the three votes of each node are counted.
- (2) Calculate the vague set value of each node by formula (3) according to the number of votes counted in step (1).
- (3) Calculate the comprehensive score according to formula (4).
- (4) Sort the comprehensive scores and use the node own the highest comprehensive score as the local primary node. If there are multiple same highest scores, one is randomly selected as the local primary node.

The local primary node of an area is equivalent to the leader of all nodes in the area, delegating other nodes to complete the consensus, and the state of the node in this area is consistent with the state of the local primary node. When an error occurs in the local primary node, a new node can be voted again to replace the old local primary node. Because of the access rules of Consortium chains, the probability of this circumstances happen is very small, and we can almost ignore it.

4.4. Global Consensus. Each area conducted a primary node selection, and selected k local primary nodes to participate in the global consensus. Supplementary Figure 5 is a network topology diagram after the primary node selection. Nodes 0, 1, and 2 choose node 2 as the local primary node; nodes 3, 4, and 5 choose node 4 as the local primary node; nodes 6, 7, and 8 choose node 6 as the local primary node; and nodes 2, 4, and 6 participate in the global consensus.

Global consensus also needs to run a primary node selection to select the primary node. We will not repeat this process here, we use * to represent node 2 as the global primary node. The global primary node plays the same role as the primary node in the PBFT consensus algorithm. In the traditional PBFT consensus algorithm, with the increase in the number of nodes, the communication times of the algorithm will increase dramatically. The main function of the pre-preparation phase of the PBFT algorithm is to ensure that in the case of network disconnection or link disconnection, the nodes will reach agreement too, but this situation is almost impossible to occur in today's era of highly developed networks, so we reduce the number of communications by subtracting the pre-preparation phase in this article. Supplementary Figure 6 is a flowchart of the PBFT algorithm with the pre-preparation phase cut. The global consensus also follows this process.

The simplified version of the PBFT consensus process is as follows:

- (1) Request: The client sends a transaction request m to the global primary node. In Supplementary Figure 6,

client C sends a transaction request m to global primary node 2^* .

- (2) Prepare: After the global primary node received the clients' transaction request m , it will broadcast the preparation message to the whole network nodes immediately. The scheme of the prepared message is $\langle \text{PREPARED}, m, v, n, d, t, n_i, i, Q_i \rangle$, where m is the original text of the request message, v is the current view number, n is the message sequence number of m , and d is the hash value of message m . t is the timestamp of message m , n_i is the partition number, i is the current node number, and Q_i is the digital signature of node i . After receiving the prepared message, the node will verify the message. After verification, the node will enter the prepared state and send a commit message $\langle \text{COMMIT}, m, v, n, d, t, n_i, i, Q_i \rangle$ to other nodes in whole network.
- (3) Commit: After the node received the commit message, it will verify the message as in the prepared phase. When the same message sent by $2f + 1$ different nodes is verified, the node will send a reply message to the client.
- (4) Reply: When the client receives $f + 1$ identical reply messages $\langle \text{REPLY}, m, v, n, d, t, n_i, i, Q_i \rangle$, the consensus is completed, the transaction is added into the blockchain.

5. Evaluation

In this part, we prove the superiority of the VS-PBFT algorithm through theoretical analysis.

5.1. Dynamic Analysis. The nodes in the blockchain are dynamic, and at any time, there may be nodes joining or exiting the blockchain. The traditional PBFT algorithm cannot detect the joining or exiting of nodes in time and dynamically adapt to the network environment.

The VS-PBFT algorithm proposed in this article uses the idea of hash consensus algorithm to place N nodes in the whole network into k areas, and each area selects a local primary node to participate in the global consensus. When a new node joins the blockchain network, it will run the hash algorithm to calculate the area that the node belongs to and divide it into the designated partition directly. When a node in one area exits, other nodes in the area can still vote for local primary nodes to participate in the global consensus. The algorithm is dynamic.

5.2. Communication Times Analysis. The traditional PBFT algorithm has five phases, and each phase needs to send a message for communication. First, the client sends a transaction request to the master node in the request phase, and the number of communications is 1. Thereafter, the primary node sends a pre-prepared message to other replica nodes, and the number of communications is $(N - 1)$. The prepared message is sent from the node to other nodes in the

whole network in the preparation phase, and the number of communications is $(N - 1)^2$. All nodes send commit messages to other nodes in the commit phase, and the number of communications is $N(N - 1)$. All nodes send a completion message to the client in the reply phase, and the number of communications is N . Adding the communication times of the above five phases to get a consensus, the communication times T_1 of the traditional PBFT algorithm is

$$T_1 = 2N^2 - N + 1. \quad (5)$$

In our proposed VS-PBFT, we need to divide the N nodes of the whole network into k areas. We know that the number of nodes in the PBFT algorithm must not be less than 3, so the number of nodes participating in the global consensus must not be less than 3, thus $k \geq 3$. Since the number of nodes in each area is at least 1, $N \geq 3$.

In the VS-PBFT algorithm, one round primary node selection needs to send $(N/k - 1)N/k$ messages to select a local primary node. There are k areas, so k primary node selections are required; in addition, global consensus need one round primary node selection. For one round consensus, $(k + 1)$ primary node selections are required in total, and the number of communications is $(N/k - 1)(k + 1)N/k$. In the four-phase consensus, the number of communications in the request phase, preparation phase, confirmation phase, and reply phase is 1, $(k - 1)$, $k(k - 1)$, and k , respectively. Therefore, the total number of communications T_2 of VS-PBFT is

$$T_2 = \left(\left(\frac{N}{k} - 1 \right) \frac{N}{k} + k \right) (k + 1). \quad (6)$$

As $N \geq 3$ and $k \geq 3$, $T_2 < T_1$. Therefore, the communication times of our proposed VS-PBFT algorithm are better than that of the traditional PBFT algorithm.

5.3. Fault Tolerance Rate Analysis. We all know that the maximum number of fault-tolerant nodes of the traditional PBFT algorithm is f_1 :

$$f_1 = \frac{N - 1}{3}. \quad (7)$$

In the VS-PBFT algorithm, the total number of nodes in the whole network is N , and the nodes in the whole network are divided into k areas. We assume that the number of nodes in each area is equal, and the number of nodes in each area is N_k . For each area, in theory, as long as the number of Byzantine nodes is less than the number of normal nodes, the most suitable node can be selected as the local primary node, so the maximum number of fault-tolerant nodes in each area is $N/2k$. Therefore, the maximum number of fault-tolerant nodes of the VS-PBFT algorithm is f_2 :

$$f_2 = \frac{N}{2}. \quad (8)$$

As $N \geq 3$, $f_2 > f_1$. Therefore, the fault tolerance rate of our proposed VS-PBFT algorithm is higher than that of the traditional PBFT algorithm.

6. Conclusion

In this article, we proposed an improved PBFT algorithm based on vague sets, named VS-PBFT. Above all, we partition the nodes and vote based on vague sets within the partitions. Each partition selects a local primary node with the highest score to participate in the four-phase consensus, so as to achieve global consensus. Theoretical analysis shows that our VS-PBFT algorithm is superior to the PBFT algorithm in fault tolerance and communication times, our algorithm is dynamic, and it can adapt to the joining and exiting of nodes at any time.

VS-PBFT has only been proved to be effective in theory, but there will be many limitations in practical applications, and the effect needs to be verified in practical environment.

In this period, blockchains are facing many limitations, one of which is that network isolation makes it extremely difficult to coordinate actions among different blockchains. Cross-chain technology is a good solution to this problem [24], but due to the lack of a consensus mechanism suitable for cross-chain technology, the development of cross-chain technology is considerably slow. Therefore, how to design a dynamic and adaptive cross-chain consensus mechanism will be the future research direction.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation (Grant nos. 61772099, 61772098, and 61802039); the Science and Technology Innovation Leadership Support Program of Chongqing (Grant no. CSTCCXLJRC201917); the Innovation and Entrepreneurship Demonstration Team Cultivation Plan of Chongqing (Grant no. CSTC2017kjrc-cxscytd0063); and Chongqing Research Program of Basic Research and Frontier Technology (Grant no. cstc2018jcyjAX0617).

Supplementary Materials

Supplementary Figure 1: The process of PBFT algorithm; it is a process of the traditional PBFT algorithm, which contains five steps: request, pre-prepare, prepare, commit, and reply. Supplementary Figure 2: The process of VS-PBFT; it is a process of the proposed algorithm, which contains three steps: node partition, primary node selection, and global consensus. After the three steps, the transactions are updated to the blockchain. Supplementary Figure 3: Partition diagram; we divide the entire circular network into three areas, and obtain the area where the node is located by calculating the node IP. Supplementary Figure 4: The process of primary node selection, after this primary node selection, we will get

k local primary node. Supplementary Figure 5: Network topology diagram; it is the network topology after the primary node selection, the three local primary node will run global consensus to find a global primary node. Supplementary Figure 6: The process of simplified PBFT; we cut the pre-prepare step to reduce the communication times. . (Supplementary Materials)

References

- [1] J. Leng, S. Ye, M. Zhou, J. Leon Zhao, Q. L. Wei, and L. fu, "Blockchain-secured smart manufacturing in industry 4.0: a survey," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 1, pp. 237–252, 2021.
- [2] V. B. Thurner, "Blockchain and the future of energy," *Technology in Society*, vol. 57, pp. 38–45, 2019.
- [3] S. Nakamoto, "A peer-to-peer electronic cashsystem," 2008, <https://bitcoins.info/bitcoin.pdf>.
- [4] M. U. Hassan, M. H. Rehmani, and J. Chen, "Deal: differentially private auction for blockchain-based microgrids energy trading," *IEEE Transactions on Services Computing*, vol. 13, pp. 263–275, 2020.
- [5] W. Z. Feng, "A hybrid cryptography scheme for nilm data security," *Electronics*, vol. 9, pp. 11–28, 2020.
- [6] J. Leng, P. Jiang, K. Xu et al., "Makerchain: a blockchain with chemical signature for self-organizing process in social manufacturing," *Journal of Cleaner Production*, vol. 234, pp. 767–778, 2019.
- [7] X. Fu, H. Wang, and P. Shi, "A survey of blockchain consensus algorithms: mechanism, design and applications," *Science China Information Sciences*, vol. 64, no. 2, pp. 1–15, 2021.
- [8] C. Qin, B. Guo, Y. Shen, T. Li, Y. Zhang, and Z. Zhang, "A Secure and Effective Construction Scheme for Blockchain Networks," *Security and Communication Networks*, vol. 2020, Article ID 8881881, 20 pages, 2020.
- [9] N. Szabo, "Smart contracts:building blocks for digital markets," *Entropy:The Journal of Trahumanist Thought*, vol. 56, p. 16, 1994.
- [10] T. Dickerson, P. Gazzillo, M. Herlihy, and E. Koskinen, "Adding concurrency to smart contracts," *Distributed Computing*, vol. 33, no. 3, pp. 209–225, 2020.
- [11] A. B. Alberto, L. Alfio, M. Giacomo, and Q. Salvatore, "On the use of blockchain technologies in wifi networks," *Computer Networks*, vol. 162106855 pages, 2019.
- [12] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. M. Leung, "Blockchain-based decentralized trust management in vehicular networks," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1495–1505, 2018.
- [13] S. Kudva, S. Badsha, S. Sengupta, I. Khalil, and A. Zomaya, "Towards secure and practical consensus for blockchain based VANET," *Information Sciences*, vol. 545, pp. 170–187, 2021.
- [14] X. Guangxia, L. P. Yong, and K. Waqas, "Improvement of the dpos consensus mechanism in blockchain based on vague sets," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4252–4259, 2020.
- [15] C. Miguel and L. Barbara, "Practical byzantine fault tolerance and proactive recovery," *ACM Transactions on Computer Systems*, vol. 20, no. 4, pp. 398–461, 2002.
- [16] D Ongaro and J Ousterhout, "In search of an understandable consensus algorithm," in *Proceedings of the 2014 USENIX Annual Technical Conference (Usenix ATC 14)*, pp. 305–319, Philadelphia, PA, USA, 2014.

- [17] C. Dwork and M. Naor, "Pricing via processing or combatting junk mail," in *Advances in Cryptology—CRYPTO' 92. CRYPTO 1992*, E. F. Brickell, Ed., Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, 1993.
- [18] E. Androulaki, A. Barger, V. Bortnikov et al., "Hyperledger Fabric: A Distributed Operating System for Permissioned blockchains," in *Proceedings of the Thirteenth EuroSys Conference*, pp. 1–15, New York, NY, USA, 2018.
- [19] A. Clement, E. L. Wong, and L. Alvisiet al, "Making byzantine fault tolerant systems tolerate byzantine faults," *6th USENIX Symposium on Networked Systems Design and Implementation*, vol. 18, pp. 153–168, 2009.
- [20] G. G. A. G. M. Pinkas, "A hybrid cryptography scheme for nilm data security, 2019 49th," *Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, vol. 69, pp. 568–580, 2019.
- [21] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.
- [22] W.-L. Gau and D. J. Buehrer, "Vague sets," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 2, pp. 610–614, 1993.
- [23] L. Yong, G. Wang, and F. Lin, "A general model for transforming vague sets into fuzzy sets," *Transactions on computational science II*, vol. 1, Article ID 133144, 2008.
- [24] J. Leng, M. Zhou, J. L. Zhao, Y. Huang, and Y. Bian, "Blockchain security: a survey of techniques and research directions," *IEEE Transactions on Services Computing*, vol. 11 page, 2021.

Research Article

Based on Consortium Blockchain to Design a Credit Verifiable Cross University Course Learning System

Chin-Ling Chen ^{1,2,3} **Tianyi Wang** ¹ **Woei-Jiunn Tsaur** ⁴ **Wei Weng** ¹
Yong-Yuan Deng ³ and **Jianfeng Cui** ⁵

¹School of Computer and Information Engineering, Xiamen University of Technology, Xiamen 361024, China

²School of Information Engineering, Changchun Sci-Tech University, Changchun 130600, China

³Department of Computer Science and Information Engineering, Chaoyang University of Technology, Taichung 41349, Taiwan

⁴Computer Center, National Taipei University, New Taipei 237303, Taiwan

⁵School of Software Engineering, Xiamen University of Technology, Xiamen 361024, China

Correspondence should be addressed to Tianyi Wang; 2022031496@stu.xmut.edu.cn and Jianfeng Cui; jfcui@xmut.edu.cn

Received 1 October 2021; Revised 7 November 2021; Accepted 16 November 2021; Published 16 December 2021

Academic Editor: Jiewu Leng

Copyright © 2021 Chin-Ling Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In recent years, the attention of online cross-university courses has been increasing, and students in universities want to increase their knowledge and professional skills by taking online courses from different universities, which raises the issue of course credit verification. In the past, the credits obtained by students in online courses lack endorsement from the education department, and the students' learning process could not be verified. Therefore, the credits of online courses in one university could not be recognized by other universities. The education departments of some countries and regions implement credit conversion rules to convert the credits obtained by students in online courses into university credits or certificates endorsed by the education department. However, these schemes rely too much on the authority of the education department, and the process of students obtaining credits cannot be verified. In addition, the centralized storage method makes the data of education departments at risk of leakage or tampering. With the emergence of blockchain technology, some researchers have proposed the use of blockchain to store students' credits, making it possible to reach consensus among multiple parties on the blockchain while ensuring that credits are not tampered with, but these schemes cannot test the learning process of students and the recognition of credits still relies on the authority of the education department. To solve the above problems, this paper proposes a cross-university course learning system with verifiable credits based on Hyperledger Fabric consortium blockchain technology, and the consortium includes many universities. The credits obtained by students in the course and the hash value of the learning records are stored on the blockchain, and the data on the blockchain is jointly maintained by the universities in the system. One university can verify the homework and final examination of students to check the real ability of students, thus recognizing the credits from other universities, and at the same time, to protect the privacy of students, the important data of students are encrypted for transmission.

1. Introduction

1.1. Background. Education is a way for students to improve themselves, and higher education is directly related to students' development direction and employment prospects. Students want to learn more courses in their universities, or even take courses across universities, to expand their

knowledge and skills. Currently, cross-university online courses are popular among students, and students can take courses from different universities online through the Internet. Especially in 2020, when the outbreak of COVID-19 spread worldwide and almost all universities around the world stopped offline courses, online courses ensured the sustainability of education [1]. By April 2021, Udemy, an

American online learning platform, has more than 40 million learners [2]. From September 2020 to July 2021, learners in just one country completed more than 140,000 courses on the Coursera platform of the United States [3]. However, there are limitations to cross-university online courses. Students can acquire more knowledge and skills through cross-university courses, but they lack simple and efficient ways to prove their ability. For a university who offers courses to students from other universities, it promotes the dissemination of professional knowledge and the reputation of the university, and the grades of students' online courses can be used as part of the entrance examination scores. Unfortunately, when it comes to the online courses from other universities, it is difficult for the university to recognize the grades even if students perform well. For teachers, by offering courses for online learning, they can show their research directions and attract more students to engage in research in related fields, but it becomes a challenge to check students' course learning outcomes because teachers do not know the capabilities of these students.

With the increase in the number of cross-university course learners, more and more students want to obtain credits for online courses as their learning credentials for further education or employment, and receiving credits or certificates means that the grades of online courses are recognized by the universities to which the courses belong. Indian Institute of Management (IIM-Kozhikode) announces partnership with Coursera to launch course certificates in business, strategy, marketing, and product management [4]. However, there exists a recognition problem with the credits obtained by taking courses across universities. The credits of online courses are jointly awarded by the learning platform and the university, lacking the verification of authoritative institutions. Although online courses are created by universities, it is difficult for the two universities to reach a consensus on the course content and learners' ability. Therefore, after completing the online course, the credits obtained by students in one university cannot be recognized by other universities.

To solve the problems of credit recognition, the European Credit Transfer and Accumulation System (ECTS) [5] helps students transfer credits between universities. In this system, the credits obtained by students in their universities can be converted into a certain number of ECTS credits, and if a student wants to transfer to another university, the ECTS credits of the student will be converted into the credits of the target university. Moreover, the Credit Bank System (CBS) [6] in Korea allows students to convert their learning achievements into credits and then convert credits into higher education degrees. However, the above schemes have some problems: Firstly, they rely on the management of the central educational institution, and it is because of the authority of the central educational institution that the credits in the system can be recognized. Secondly, the credit conversion rules lack verification for the learning process of students. Although the credit is recognized by all universities in the system, it does not fully reflect the student's ability. Finally, since the system data are stored in a centralized way, there is a risk of data loss or tampering.

The emergence of blockchain technology [7] has provided a new way to solve the credit verification problem [8]. The blockchain can be regarded as a distributed database where the data is tamper-evident, traceable [9–11], maintained by multiple parties, and can reach a consensus among multiple parties [12]. This paper proposes a decentralized cross-university course learning system based on consortium blockchain technology, where the consortium includes many universities. For students, they can take courses from any university in the system and get credits. The hash values of students' homework and final examinations are stored on the blockchain, and the data on the blockchain is maintained by all universities in the system. Since each university manages its teachers and students, the data on the blockchain can reach a consensus among all universities, teachers, and students without relying on a central institution, and one university recognizes course credits obtained by students from other universities by verifying students' homework and final examination.

1.2. Related Works. Currently, research in the education field focuses on storing students' credits, certificates, and learning records via blockchain technology and using distributed storage technology and cryptography to ensure that the data on the blockchain is tamper-evident, thus facilitating the sharing of data among multiple parties. The related works are listed in Table 1.

In 2016, Sharples and Domingue [13] proposed to use blockchain to store students' learning processes and achievements, thus enabling distributed storage of student-related data, but this paper did not introduce the system architecture.

In 2018, Turkanović et al. [14] proposed a blockchain education credit platform named EduCTX where educational institutions can award students credits that can be checked by third parties and students can transfer credits between different educational institutions. Unfortunately, this system can only check whether a student has obtained credits and has no way to verify the student's learning process.

In 2020, Zhao et al. [15] proposed a student portfolio management system that stores students' learning records and teachers' evaluations of students through blockchain technology. However, teachers' evaluations are subjective and cannot objectively reflect students' abilities. In addition, the article failed to provide a method to protect the privacy of students' learning records.

In 2021, Mishra et al. [16] proposed an Ethereum-based student credential sharing system, where universities encrypt the students' credentials before uploading them to the blockchain, and if a third party wants to check students' credentials, the students encrypt credentials before sending them to protect their privacy. Considering that students may not be able to afford the gas in Ethereum, the system has set up a fund organization to provide financial support for system members.

Based on the above research, it can be found that the recognition of students' credits still relies on authority and

TABLE 1: The related works survey.

| Authors | Year | Objective | Technologies | Merits | Demerits |
|------------------------|------|---|-----------------------|--|---|
| Sharples et al. [13] | 2016 | A distributed system for the educational record, reputation, and reward | Blockchain | Realize distributed storage of student-related data | No system architecture is proposed |
| Turkanović et al. [14] | 2018 | Higher education credit platform | Public blockchain | Student credits can be awarded and transferred | Students' learning process cannot be checked |
| Zhao et al. [15] | 2020 | System for student e-portfolio assessment | Consortium blockchain | Realize storage of students' learning records and teachers' evaluation | Security analysis is not sufficient |
| Mishra et al. [16] | 2021 | System for sharing students' credentials | Ethereum | Student certificates are encrypted before being uploaded to the system | Consumption of tokens is inevitable |
| Jeong et al. [17] | 2021 | The multilateral personal portfolio authentication system | Hyperledger fabric | Detailed system implementation based on hyperledger fabric | Unable to protect the privacy of learning records |

the process of obtaining credits cannot be verified. Therefore, this paper proposed a cross-university course learning system with credits verifiable, where a student needs to complete homework and final examination to obtain credit from the university, and other universities can verify the student's homework and final examination to recognize the credit. This system encrypts students' homework and final examinations, thus effectively protecting students' privacy. Moreover, the system adopts consortium blockchain architecture with no token consumption, which improves operational efficiency compared to public blockchain architecture.

The remainder of this paper is organized as follows: Section 2 briefly introduces the preliminary. Section 3 shows the proposed system structure and an application scenario. The paper gives the security and feature analysis in Section 4 and presents the discussion in Section 5. Finally, Section 6 concludes this paper.

2. Preliminary

2.1. Elliptic Curve Cryptography ECDSA. In the blockchain, elliptic curve cryptography [18] is used for digital signature. If a member A in the system wants to send a message M to a member B , the member A needs to digitally sign the message. The process of the signature algorithm is as follows.

2.1.1. Determine Parameters and Generate Keys. The system will first determine the parameters a and b of the elliptic curve $y^2 = (x^3 + ax + b) \bmod p$, the base p , and the origin G , and then the system will generate private keys d_A and d_B for members A and B , respectively, and generate the public key $Q_A = d_A G$ for the member A .

2.1.2. Generate Signature. Member A selects a random number k , then calculates $H = \text{hash}(M)$, $(x, y) = kG$, $r = x \bmod p$, $s = k^{-1}(H + rd_A) \bmod p$, and sends the ECDSA signature pair (r, s) together with the message M to the member B .

2.1.3. Verify Signature. After receiving the signature pair (r, s) and the message M , the member B calculates $H' = \text{hash}(M)$, $u = H's^{-1} \bmod p$, $v = rs^{-1} \bmod p$, $(x', y') = uG + vd_A G$. If $x' \bmod p = r$, member B confirms that the signature pair (r, s) and the message M sent by the member A are correct.

2.2. Smart Contract. In 1996, Nick Szabo first proposed the concept of the smart contract [19], which digitized contracts in the real world. In a smart contract, both parties will agree on the content of the contract in advance, and the contract will be executed automatically when the conditions are met, without the need for supervision by a third party. Blockchain has the characteristics of nontampering of data and decentralization. The emergence of blockchain technology provides a platform to support the execution of smart contracts, which are jointly executed by the nodes of the whole blockchain network, and the results of their execution become impossible to tamper with after the whole network reaches consensus. Blockchain and smart contract technologies provide new solutions to existing problems in many industries [20], and this paper aims to solve the problem of credit verification across universities.

2.3. Hyperledger Fabric. Hyperledger Fabric was proposed by IBM in 2018 [21], which is an open-source blockchain platform and one of the most popular consortium blockchains so far [22]. Unlike public blockchain systems such as Bitcoin and Ethereum, Hyperledger Fabric uses consortium blockchain technology where system members reach a consensus on transactions without consuming tokens. Different from Bitcoin and Ethereum where the data is publicly accessible [23], all members who join the Hyperledger Fabric network need to be authenticated to ensure that unrelated people cannot join the network and get the data on the blockchain. Since the identities of all members in the network are known, the nodes of Hyperledger Fabric do not need to reach a consensus through Proof of Work (POW) [24], and the number of transactions generated in

Hyperledger Fabric can reach 3500 per second, which is much higher than 3.5 of Bitcoin and 5.4 of Ethereum [25].

In addition to Hyperledger Fabric, there are currently many consortium blockchain platforms in the market, such as Ethereum [26], Corda [27], Quorum [28], and Multi-Chain [29]. Compared with the above platforms, Hyperledger Fabric has higher throughput and shorter latency [30], and it has wide interest and application in many industries (including finance, IoT, supply chain, manufacturing, and technology) [31]. Therefore, this paper chooses to design a credit verifiable cross-university course learning system based on Hyperledger Fabric.

There are mainly the following components in the Hyperledger Fabric network:

- (1) Certificate authority (CA): certificate authority provides an identity authentication mechanism for system users. Before a user can interact with the blockchain network, he or she needs to connect to a CA server, which provides the user with identity information as well as the public and private keys.
- (2) Orderer: an orderer node collects the endorsed transactions sent by users from client nodes and packages them into blocks, then sends these blocks to the peer nodes.
- (3) Peer: peer nodes are divided into endorse peer, leader peer, and anchor peer, where the endorse peer calls chaincode to simulate the execution of a transaction and endorse the transaction, the leader peer broadcasts the block received from the orderer node to all the peer nodes in the organization, and the anchor peer exchange data with other anchor peers between different organizations. All the peer nodes can be considered as committer peers who check each transaction in the received block and update the ledger after the check is completed, and the ledger consists of a blockchain that stores all the transactions and the World State that stores the state data of all the members in the system.
- (4) Client: the client node is operated by a system member, which must be connected to one of the peer nodes or orderer nodes to communicate with the blockchain network. Firstly, the client node sends a transaction proposal to the peer in the organization for endorsement. Once the client node has received a sufficient number of signed proposal responses from endorse peers, the client node sends the transaction containing endorsed transaction proposal responses to the orderer node, and the orderer node orders the transactions into blocks.
- (5) Channel: channel can achieve isolation of different services and there is only one blockchain in a channel. Users need to get certificates from the CA node firstly, then they can communicate with the peer node or orderer node through the channel.
- (6) Organization: organizations represent entities such as enterprises and institutions in the blockchain network. Each organization contains endorse peer,

leader peer, and anchor peer that store the ledger, and each member in the system belongs to an organization.

- (7) Chaincode: chaincode can be regarded as the smart contract in Hyperledger Fabric, which is written in some language and is deployed on every peer node, and users can achieve query and modification of data on the blockchain by invoking chaincode.

3. System Model

3.1. System Architecture. This research proposed a cross-university course learning system based on Hyperledger Fabric where the consortium includes many universities. The main members of the system include university administrators and users, and users include teachers and students. Teachers, students, and university administrators in the same university form an organization, and each university has administrators to manage its users. The system architecture is shown in Figure 1.

The members of the system are described as follows.

- (1) Certificate authority (CA): CA nodes provide certificates, public and private keys for users who want to join the system, and each university has its own CA node.
- (2) University: a university is managed by university administrators, and there are many teachers and students and some administrators in each university.
- (3) University administrator: a university administrator creates the channel, joins the channel with the peers in his or her organization, installs the chaincode on each peer node and initializes the chaincode, reviews the identity of users, and creates system accounts for them, reviews courses created by the teachers and checks students' grades. By invoking chaincode, administrators add course information to teachers' accounts, award credits to students, and verify students' homework and final examinations.
- (4) Teacher: teachers apply to their universities for teaching courses and grade students' homework and final examinations. By invoking chaincode, teachers add course information and grades to students' accounts.
- (5) Students apply to the teachers for learning courses, submit homework and final examinations to the teachers for grades and apply to their universities for course credits. By invoking chaincode, students add the hash values of their homework and final examinations to their accounts.

3.2. Application Scenario. Figure 2 shows the application scenario where Student A of University A wants to learn Course B of Teacher B who comes from University B, and University C wants to verify the credit obtained by Student A.

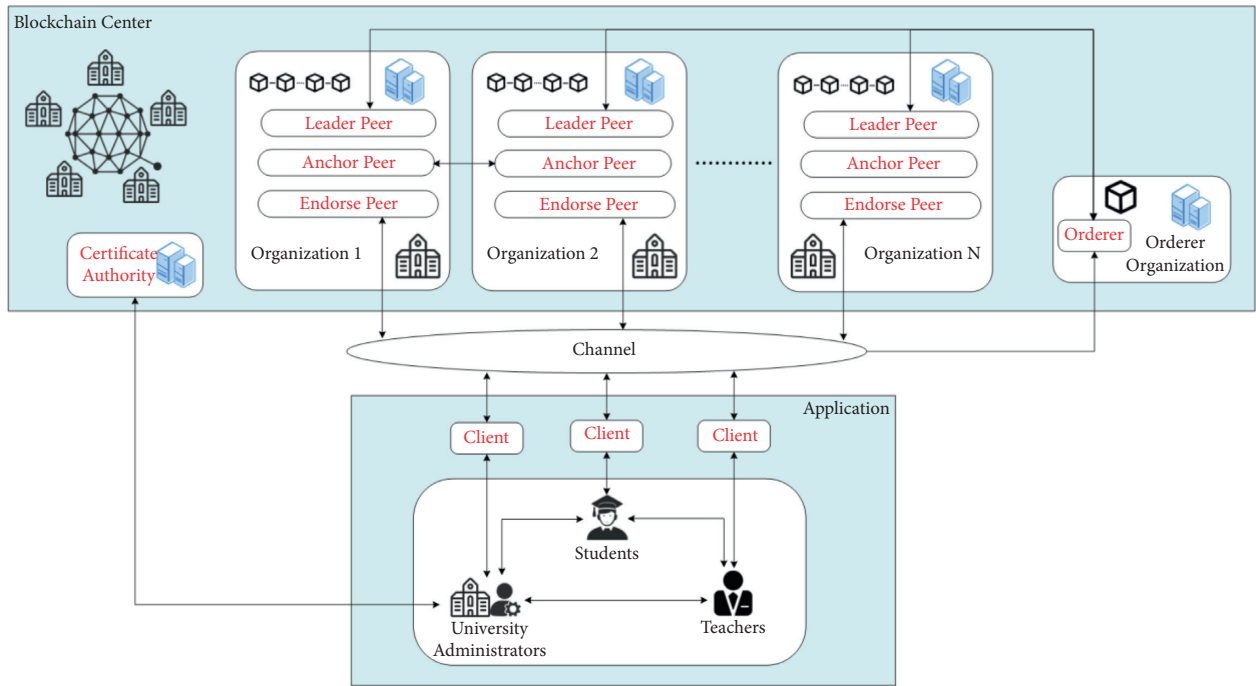


FIGURE 1: System structure.

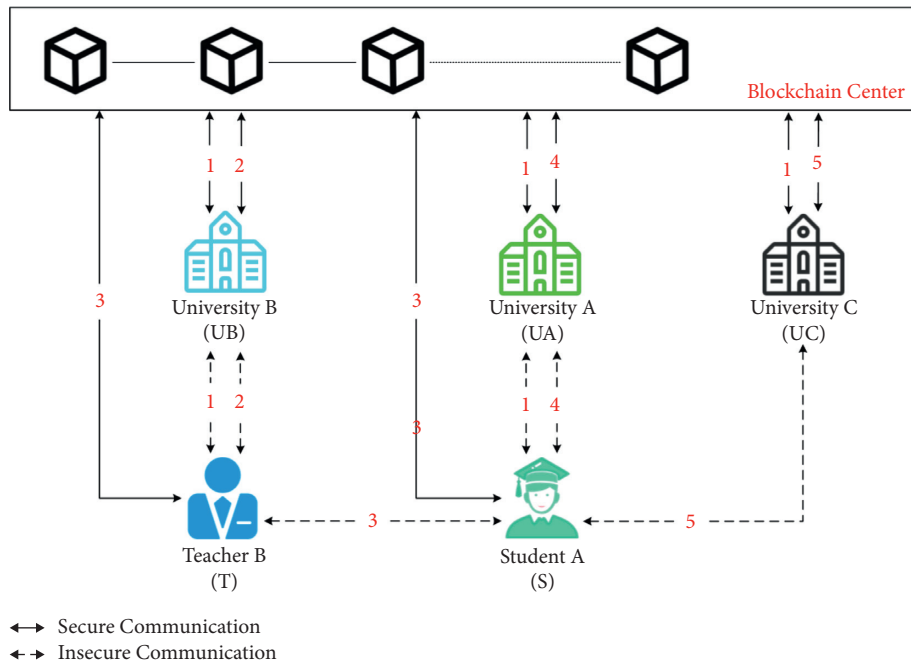


FIGURE 2: Application scenario.

Step 1: the administrator of a university first creates a channel, then the administrator of each university in the system joins the channel with the peers in his or her organization, installs the chaincode on each peer node, and initializes the chaincode. All users (including teachers and students) need to apply for registration with the administrators of their universities first to obtain a system account.

Step 2: Teacher B applies to University B for teaching Course B, which needs to be reviewed by his university. After the course is approved, University B adds the course information to the account of Teacher B by invoking chaincode.

Step 3: Student A applies to Teacher B for learning Course B, and Teacher B adds the course information to the account of Student A by invoking chaincode.

During the course learning period, Student A needs to upload the hash value of homework and final examination and send them to Teacher B for grading, then Teacher B will upload the grade of Student A by invoking chaincode.

Step 4: Student A applies to his university for obtaining the credit of Course B after receiving the grade from Teacher B. By invoking chaincode, University A checks the grade of Student A and awards credit if the grade is qualified.

Step 5: Student A wants to transfer from University A to University C, and University C and, therefore, wants to verify the credits of Student A. University C submits the credit verification request to Student A, and Student A sends the homework and final examination to University C. Before reviewing the content of homework and final examination, University C needs to calculate the hash value of the homework and final examination, then compares it with the hash value uploaded by Student A to ensure that the homework and final examination have never been tampered with.

3.3. Initial Phase. In the initial phase, the university administrator creates a channel, then installs, and initializes the chaincode for each peer node in the organization.

Step 1: the university administrator logs in to the system through an application program, then starts a client node and creates a channel.

Step 2: the university administrator connects to each peer node in the organization in turn through the channel, then installs the chaincode on each peer node and initializes the chaincode. The chaincode is shown in Algorithm 1.

3.4. User Registration Phase. In this phase, User X submits registration application and identity information to his or her university. After verifying the identity, the university administrator connects to the CA node to generate the certificate, the public key, and the private key of User X . Figure 3 shows the flowchart of the user registration phase.

Step 1: User X generates the registration application $M_{Request}$ and identity information $M_{Identity}$, then transmits $M_{Request}$, $M_{Identity}$ to his or her university

Step 2: University administrators verify $M_{Identity}$, then transmit $M_{Identity}$ to the CA node if $M_{Identity}$ is valid. CA generates the private key d_X , the public key Q_X , and the certificate $Cert_X$ of User X based on $M_{Identity}$

Step 3: The application program generates the system account and ID_X of User X based on d_X , Q_X , $Cert_X$ and $M_{Identity}$, then sends (d_X, Q_X, ID_X) to User X

3.5. Course Registration Phase. In this phase, Teacher B sends Course B to his university for review. If Course B is valid, the administrator of University B adds the course information to

the account of Teacher B by invoking chaincode. Figure 4 shows the flowchart of the course registration phase.

Step 1: Teacher B wants to add Course B to the system, first he generates M_{Course} and the course teaching request $M_{Request}$, then chooses a random number k_1 , calculates $H_1 = \text{hash}(M_{Course}, ID_T, M_{Request}, TS_T)$, $(x_1, y_1) = k_1G$, $r_1 = x_1 \bmod p$, $s_1 = k_1^{-1}(H_1 + r_1d_T) \bmod p$, and sends $M_{Request}, ID_T, M_{Course}, TS_T, Cert_T, (r_1, s_1)$ to University B.

Step 2: after receiving $M_{Request}$ from Teacher B, the administrator of University B first uses $TS_{NOW} - TS_T \leq \Delta T$ to confirm whether the timestamp is valid, then searches for the public key Q_T of teacher B by ID_T , verifies $Cert_T$ by Q_T , and calculates $H_2 = \text{hash}(M_{Course}, ID_T, M_{Request}, TS_T)$, $u_1 = H_2s_1^{-1} \bmod p$, $v_1 = r_1s_1^{-1} \bmod p$, $(x_2, y_2) = u_1G + v_1Q_T$, check $x_2 \bmod p \stackrel{?}{=} r_1$. If the signature verification is passed, the university administrator reviews Course B, generates the result M_{Result} , and invokes the chaincode `CheckCourse`. The chaincode is shown in Algorithm 2.

3.6. Course Learning Phase. If Student A wants to learn Course B from University B, he will first apply for course learning to Teacher B and Teacher B adds the course information to his account by invoking chaincode. During the period of learning Course B, Student A uploads the hash value of his homework and final examination by invoking chaincode and sends them to Teacher B for grading, then Teacher B uploads the grade of Student A by invoking chaincode. Figure 5 shows the flowchart of the course learning phase.

Step 1: Student A selects Course B that he wants to learn, then generates the course learning application $M_{Request}$, chooses a random number k_3 , calculates $H_5 = \text{hash}(M_{Request}, ID_C, ID_S, TS_S)$, $(x_5, y_5) = k_3G$, $r_3 = x_5 \bmod p$, $s_3 = k_3^{-1}(H_5 + r_3d_S) \bmod p$, and sends $M_{Request}, ID_C, ID_S, TS_S, Cert_S, (r_3, s_3)$ to Teacher B.

Step 2: after receiving $M_{Request}$ from Student A, Teacher B first uses $TS_{NOW} - TS_S \leq \Delta T$ to confirm whether the timestamp is valid, then searches for the public key Q_S of Student A by ID_S , verifies $Cert_S$ by Q_S , and calculates $H_6 = \text{hash}(M_{Request}, ID_C, ID_S, TS_S)$, $u_3 = H_6s_3^{-1} \bmod p$, $v_3 = r_3s_3^{-1} \bmod p$, $(x_6, y_6) = u_3G + v_3Q_S$, $x_6 \bmod p \stackrel{?}{=} r_3$. If the signature verification is passed, Teacher B invokes the chaincode `LearnCourse`. The chaincode is shown in Algorithm 3.

Then, Student A chooses a random number k_4 , calculates $H_8 = \text{hash}(M_{Work}, ID_C, ID_S, TS_S)$, $(x_7, y_7) = k_4G$, $r_4 = x_7 \bmod p$, $s_4 = k_4^{-1}(H_8 + r_4d_S) \bmod p$, and sends $C_{Work}, ID_C, ID_S, TS_S, Cert_S, (r_4, s_4)$ to Teacher B.

Step 4: after receiving C_{Work} from Student A, Teacher B first uses $TS_{NOW} - TS_S \leq \Delta T$ to confirm whether the timestamp is valid, then verifies $Cert_S$ by Q_S , decrypts C_{Work} with his private key d_T , generates $M_{Work} = D_{d_T}(C_{Work})$ and calculates $H_9 = \text{hash}(M_{Work},$

```

(1) type Chaincode struct {
(2) }
(3) type Teacher struct {
(4)   Name string 'json:"name"'
(5)   University string 'json:"university"'
(6)   Course string 'json:"course"'
(7) }
(8) type Student struct {
(9)   Name string 'json:"name"'
(10)  University string 'json:"university"'
(11)  Hash string 'json:"hash"'
(12)  Grade int 'json:"grade"'
(13)  Credit int 'json:"credit"'
(14) }
(15) func (t *Chaincode) Init(stub shim.ChaincodeStubInterface) peer.Response {
(16)   return shim.Success(nil)
(17) }
(18) func (t *Chaincode) Invoke(stub shim.ChaincodeStubInterface) peer.Response {
(19)   function, args: = stub.GetFunctionAndParameters()
(20)   if function == "CheckCourse" {
(21)     return t.CheckCourse(stub, args)
(22) } else if function == "LearnCourse" {
(23)   return t.LearnCourse(stub, args)
(24) } else if function == "WorkUpload" {
(25)   return t.WorkUpload(stub, args)
(26) } else if function == "AddGrade" {
(27)   return t.AddGrade(stub, args)
(28) } else if function == "CheckStudent" {
(29)   return t.CheckStudent(stub, args)
(30) } else if function == "AwardCredit" {
(31)   return t.AwardCredit(stub, args)
(32) }
(33) return shim.Error("Invalid Smart Contract function name. ")
(34) }

```

ALGORITHM 1: Chaincode used for initialization.

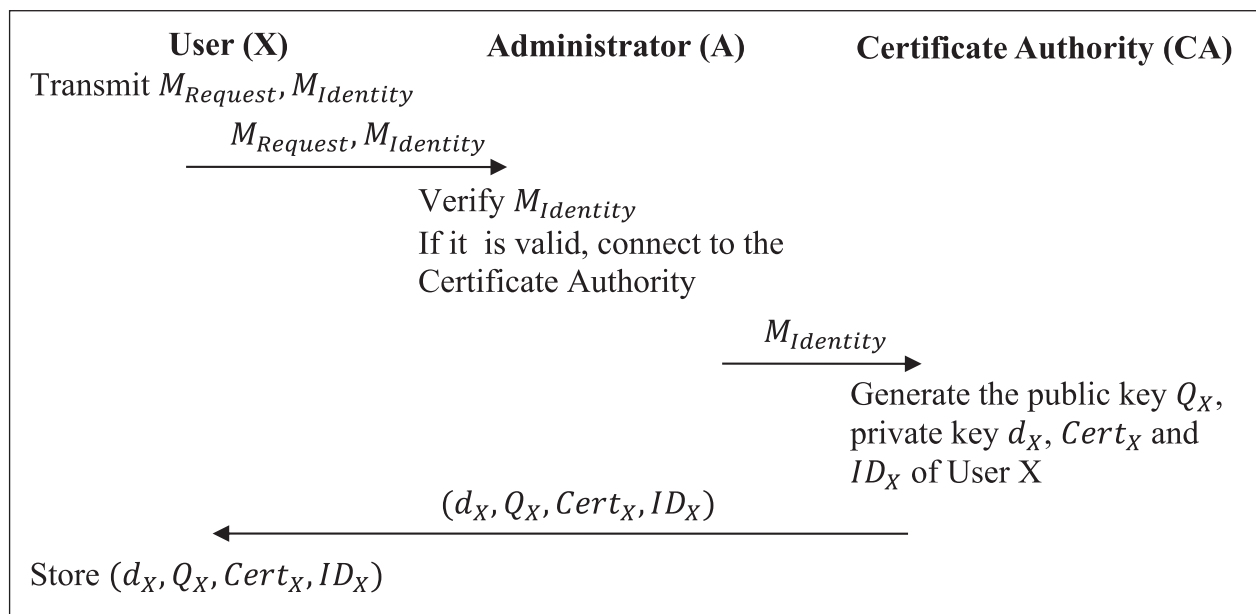


FIGURE 3: User registration phase.

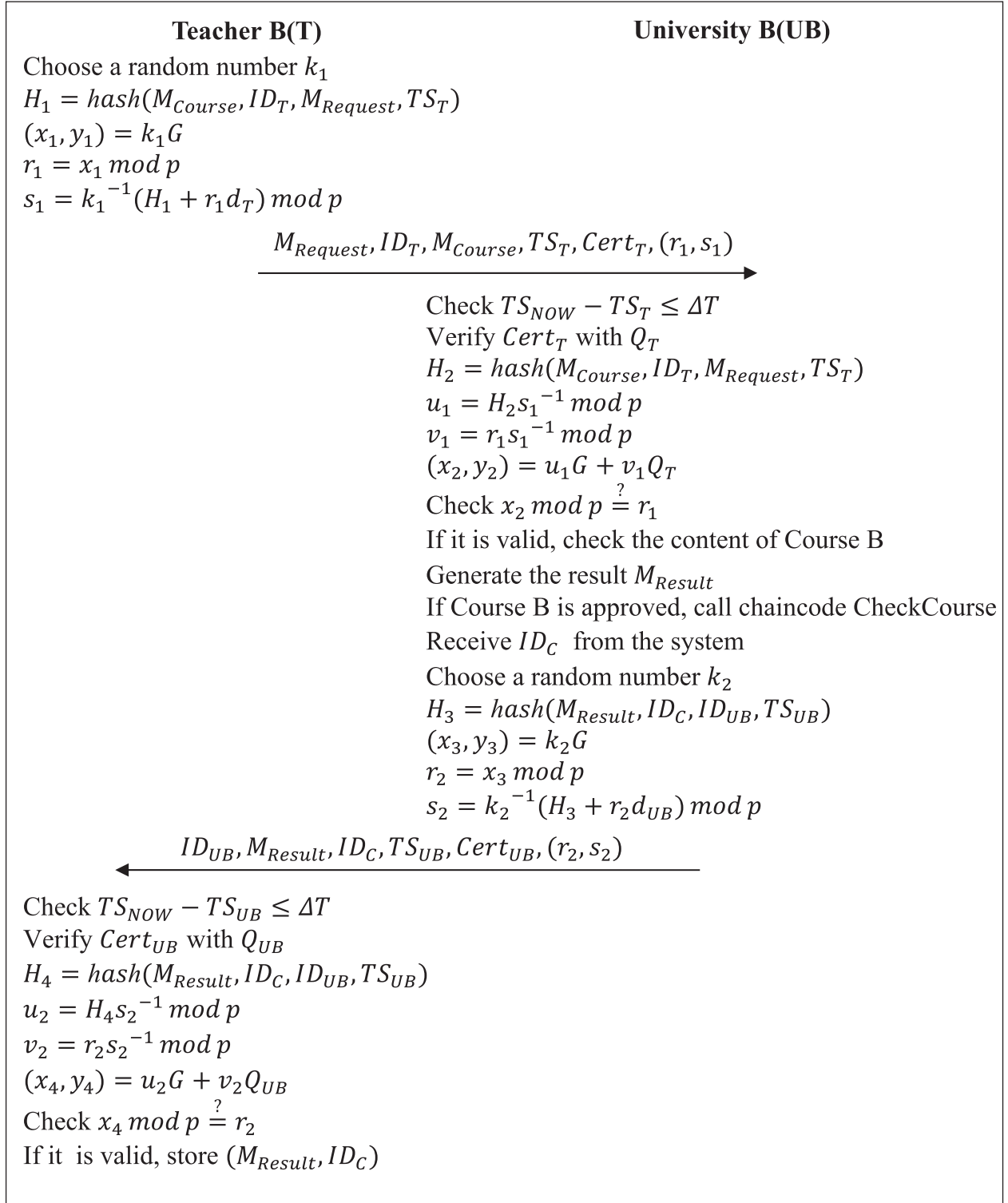


FIGURE 4: Course registration phase.

ID_C, ID_S, TS_S , $u_4 = H_9 s_4^{-1} \bmod p$, $v_4 = r_4 s_5^{-1} \bmod p$, $(x_8, y_8) = u_4 G + v_4 Q_S$, $x_8 \bmod p = r_4$. If the signature verification is passed, Teacher B reviews M_{Work} and generates the grade g , then he uploads the grade by invoking the chaincode AddGrade. The chaincode is shown in Algorithm 5.

3.7. *Credit Application Phase.* In this phase, Student A applies to his university for the credit of Course B. If the grade of Student A is qualified, the administrator of University A awards him the credit by invoking the chaincode. Figure 6 shows the flowchart of the credit application phase.

```

(1) func (t *Chaincode) CheckCourse(APIStub shim.ChaincodeStubInterface, args []string) peer.Response {
(2) if len(args) != 4 {
(3)   return shim.Error("nu Incorrect mber of arguments. Expecting 4")
(4) }
(5) var teacher = Teacher{Name: args [1], University: args [2], Course: args [3]}
(6) teacherAsBytes, _ := json.Marshal(teacher)
(7) APIStub.PutState(args[0], teacherAsBytes)
(8) return shim.Success(nil)
(9) }

```

Step 3: after receiving $ID_{UB}, M_{Result}, ID_C, TS_{UB}, Cert_{UB}, (r_2, s_2)$ from University B, Teacher B first uses $TS_{NOW} - TS_{UB} \leq \Delta T$ to confirm whether the timestamp is valid, then searches for the public key Q_{UB} of University B by ID_{UB} , verifies $Cert_{UB}$ by Q_{UB} , and calculates $H_4 = \text{hash}(M_{Result}, ID_C, ID_{UB}, TS_{UB})$, $u_2 = H_4 s_2^{-1} \bmod p$, $v_2 = r_2 s_2^{-1} \bmod p$, $(x_4, y_4) = u_2 G + v_2 Q_{UB}$, $x_4 \bmod p = r_2$.

ALGORITHM 2: Chaincode for the university to check courses.

Step 1: Student A generates the credit application $M_{Request}$ and sends $M_{Request}, ID_C, ID_S$ to University A.

Step 2: after receiving $M_{Request}, ID_C, ID_S$ from Student A, the administrator of University A invokes the chaincode `CheckStudent` to check the grade g of Student A. The chaincode is shown in Algorithm 6. If the grade g is qualified, the administrator adds the credit to the account of Student A by invoking the chaincode `AwardCredit`. The chaincode is shown in Algorithm 7.

3.8. Credit Verification Phase. Student A wants to transfer from University A to University C, and thus University C wants to verify the credits of Student A. University C first applies for credit verification, then Student A sends his homework and final examination which are encrypted to University C. After decrypting the message and ensuring that the data is not tampered with, the administrator of University C reviews the content of the homework and final examination. Figure 7 shows the flowchart of the credit verification phase.

Step 1: University C generates the credit verification application $M_{Request}$, chooses a random number k_5 , calculates $H_{10} = \text{hash}(M_{Request}, ID_C, ID_{UC}, TS_{UC})$, $(x_9, y_9) = k_5 G$, $r_5 = x_9 \bmod p$, $s_5 = k_5^{-1}(H_{10} + r_5 d_{UC}) \bmod p$, and sends $M_{Request}, ID_C, ID_{UC}, TS_{UC}, Cert_{UC}, (r_5, s_5)$ to Student A.

Step 2: after receiving $M_{Request}$ from University C, Student A first uses $TS_{NOW} - TS_{UC} \leq \Delta T$ to confirm whether the timestamp is valid, then searches for the public key Q_{UC} of University C by ID_{UC} , verifies $Cert_{UC}$ by Q_{UC} , and calculates $H_{11} = \text{hash}(M_{Request}, ID_C, ID_{UC}, TS_{UC})$, $u_5 = H_{11} s_5^{-1} \bmod p$, $v_5 = r_5 s_5^{-1} \bmod p$, $(x_{10}, y_{10}) = u_5 G + v_5 Q_{UC}$, $x_{10} \bmod p = r_5$. If the signature verification is passed, Student A generates M_{Result} , encrypts M_{Work} with the public key Q_{UC} , generates $C_{Work} = E_{Q_{UC}}(M_{Work})$, chooses a random number k_6 , calculates $H_{12} = \text{hash}(M_{Result}, M_{Work}, ID_C, ID_S, TS_S)$, $(x_{11}, y_{11}) = k_6 G$, $r_6 = x_{11} \bmod p$, $s_6 = k_6^{-1}(H_{12} + r_6 d_S) \bmod p$, and sends $M_{Result}, C_{Work}, ID_C, ID_S, TS_S, Cert_S, (r_6, s_6)$ to Student A.

Step 3: after receiving M_{Result} from Student A, the administrator of University C first uses $TS_{NOW} - TS_S \leq \Delta T$ to confirm whether the timestamp is valid, then searches for the public key Q_S of Student A by ID_S , verifies $Cert_S$ by Q_S , decrypts C_{Work} with the private key d_{UC} , generates $M_{Work} = D_{d_{UC}}(C_{Work})$ and calculates $H_{13} = \text{hash}(M_{Result}, M_{Work}, ID_C, ID_S, TS_S)$, $u_6 = H_{13} s_6^{-1} \bmod p$, $v_6 = r_6 s_6^{-1} \bmod p$, $(x_{12}, y_{12}) = u_6 G + v_6 Q_S$, $x_{12} \bmod p = r_6$. If the signature verification is passed, the administrator checks H_7 which was uploaded by Student A by invoking the chaincode `CheckStudent` and calculates $H_{14} = \text{hash}(M_{Work})$. If $H_{14} = H_7$, it means that M_{Work} sent by Student A to University C are the same as the homework and final examination which were submitted by Student A in the course learning phase. Having ensured that the homework and final examination are not tampered with, the administrator of University C reviews the content of M_{Work} .

4. Security and Feature Analysis

4.1. Mutual Authentication. In this paper, BAN logic [32] was used for identity authentication. The notation of BAN logic is described as below.

- (1) $\langle X \rangle_K$ The message X is combined with a key K
- (2) $P| \equiv XP$ believes X
- (3) $P \stackrel{K}{\leftrightarrow} QP$ and Q use a shared key K to communicate
- (4) $\#(X)$ The message X is fresh
- (5) $P| \Rightarrow XP$ has jurisdiction over X
- (6) $P \triangleleft XP$ sees X
- (7) $P| \sim XP$ once said X

The main goals of the scheme are to authenticate the identity between User X and User Y .

- (1) G1: $X| \equiv X \stackrel{X \leftrightarrow Y}{\leftrightarrow} Y$
- (2) G2: $X| \equiv Y| \equiv X \stackrel{X \leftrightarrow Y}{\leftrightarrow} Y$
- (3) G3: $Y| \equiv X \stackrel{X \leftrightarrow Y}{\leftrightarrow} Y$
- (4) G4: $Y| \equiv X| \equiv X \stackrel{X \leftrightarrow Y}{\leftrightarrow} Y$
- (5) G5: $X| \equiv ID_Y$

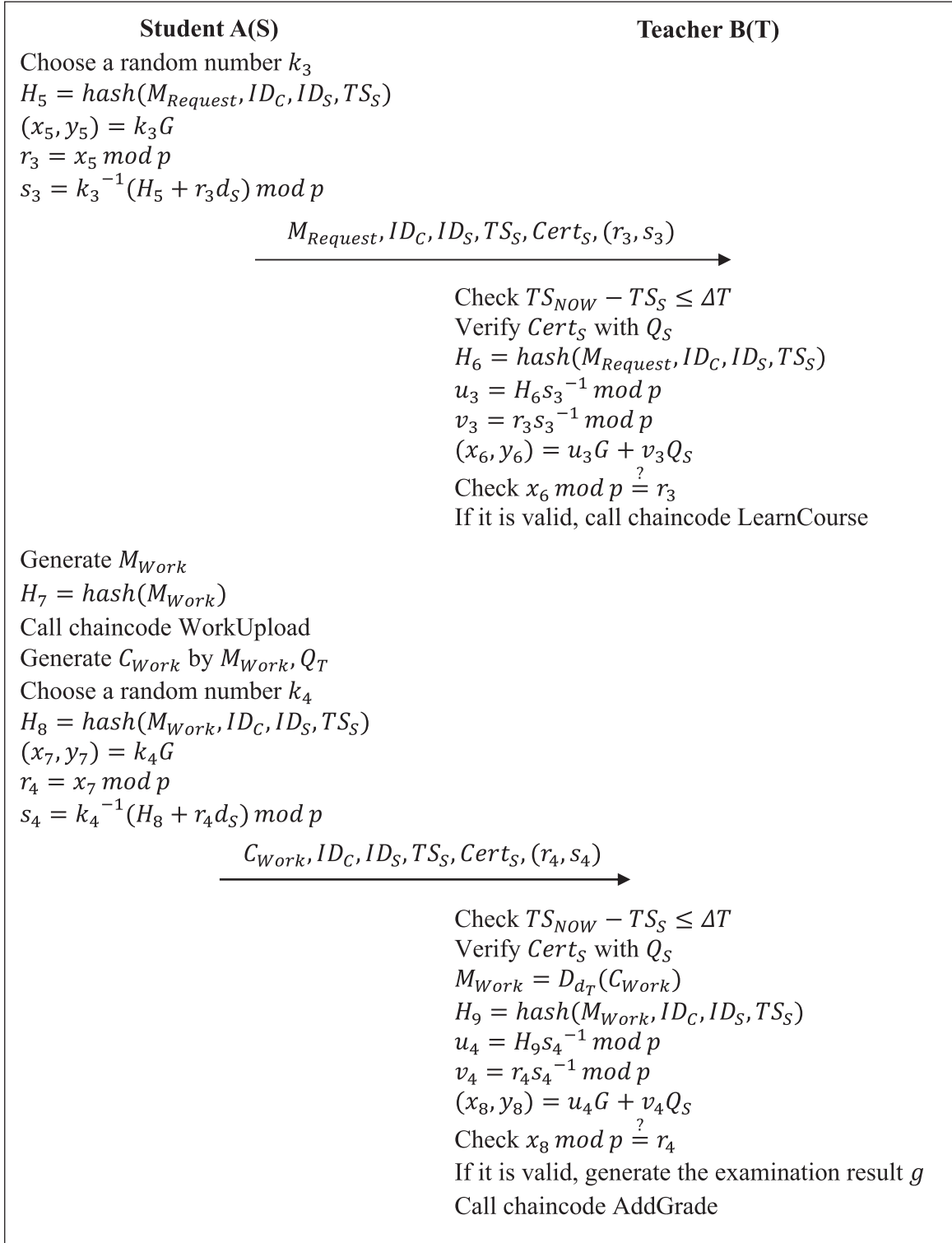


FIGURE 5: Course learning phase.

(6) G6: $X| \equiv Y| \equiv ID_Y$

(7) G7: $Y| \equiv ID_X$

(8) G8: $Y| \equiv X| \equiv ID_X$

BAN logic is used for producing an idealized form as follows:

(1) M1: $(\langle \text{hash}(ID_X, M_X, TS_X) \rangle_{x_{X-Y}})$

(2) M2: $(\langle \text{hash}(ID_Y, M_Y, TS_Y) \rangle_{x_{Y-X}})$

It is necessary to make the following assumptions before analyzing the proposed scheme:

(1) A1: $X| \equiv \#(TS_X)$

```

(1) func (t *Chaincode) LearnCourse(APIStub shim.ChaincodeStubInterface, args []string) peer.Response {
(2)   if len(args) != 6 {
(3)     return shim.Error("Incorrect number of arguments. Expecting 6")
(4)   }
(5)   var student = Student{Name: args [1], University: args [2], Hash: args [3], Grade: args [2], Credit: args [3]}
(6)   studentAsBytes, _ := json.Marshal(student)
(7)   APIStub.PutState(args[0], studentAsBytes)
(8)   return shim.Success(nil)
(9) }

```

Step 3: When Student A finishes his homework and final examination M_{Work} , he searches for the public key Q_T of Teacher B by ID_T , encrypts M_{Work} with the public key Q_T , generates $C_{\text{Work}} = E_{Q_T}(M_{\text{Work}})$, calculates $H_7 = \text{hash}(M_{\text{Work}})$, and uploads H_7 by invoking the chaincode WorkUpload. The chaincode is shown in Algorithm 4.

ALGORITHM 3: Chaincode for the teacher to approve students to join the course.

```

(1) func (t *Chaincode) WorkUpload(APIStub shim.ChaincodeStubInterface, args []string) peer.Response {
(2)   if len(args) != 2 {
(3)     return shim.Error("Incorrect number of arguments. Expecting 2")
(4)   }
(5)   studentAsBytes, _ := APIStub.GetState(args[0])
(6)   student := Student{}
(7)   json.Unmarshal(studentAsBytes, &student)
(8)   student.Hash = args[3].
(9)   studentAsBytes, _ = json.Marshal(student)
(10)  APIStub.PutState(args[0], studentAsBytes)
(11)  return shim.Success(nil)
(12) }

```

ALGORITHM 4: Chaincode for the student to upload the hash value of his homework and examination.

```

(1) func (t *Chaincode) AddGrade(APIStub shim.ChaincodeStubInterface, args []string) peer.Response {
(2)   if len(args) != 2 {
(3)     return shim.Error("Incorrect number of arguments. Expecting 2")
(4)   }
(5)   studentAsBytes, _ := APIStub.GetState(args[0])
(6)   student := Student{}
(7)   json.Unmarshal(studentAsBytes, &student)
(8)   student.Grade = args[4].
(9)   studentAsBytes, _ = json.Marshal(student)
(10)  APIStub.PutState(args[0], studentAsBytes)
(11)  return shim.Success(nil)
(12) }

```

ALGORITHM 5: Chaincode for the teacher to add course grades to the student account.

- (2) A2: $Y | \equiv \#(TS_X)$
- (3) A3: $X | \equiv \#(TS_Y)$
- (4) A4: $Y | \equiv \#(TS_Y)$
- (5) A5: $X | \equiv Y | \Rightarrow X \stackrel{x_{Y-X}}{\leftrightarrow} Y$
- (6) A6: $Y | \equiv X | \Rightarrow X \stackrel{x_{X-Y}}{\leftrightarrow} Y$
- (7) A7: $X | \equiv Y | \Rightarrow ID_Y$
- (8) A8: $Y | \equiv X | \Rightarrow ID_X$

According to these assumptions, the main proof of the authentication is as follows:

4.1.1. *User Y Authenticates User X.* By M1 and the seeing rule, derive

$$Y \triangleleft \langle \text{hash}(ID_X, M_X, TS_X) \rangle_{x_{X-Y}}. \quad (1)$$

By A2 and the freshness rule, derive

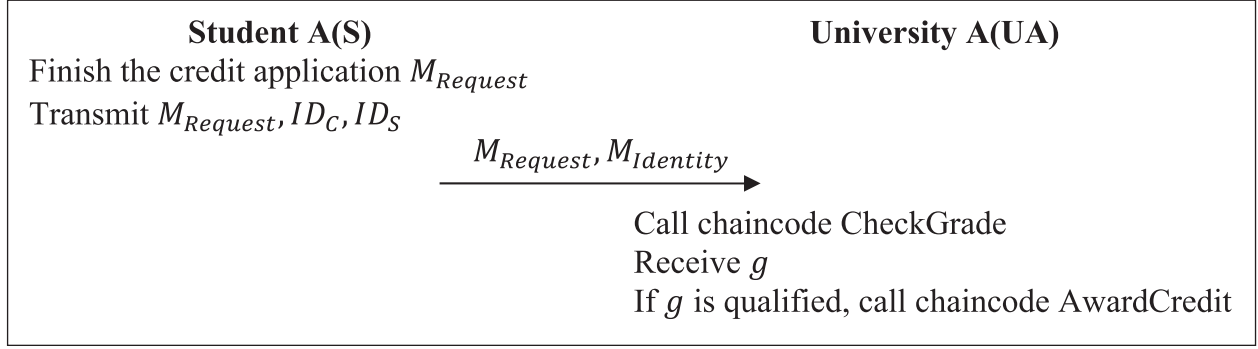


FIGURE 6: Credit application phase.

```

(1) func (t *Chaincode) CheckStudent(APIstub shim.ChaincodeStubInterface, args []string) peer.Response {
(2)   if len(args) != 1 {
(3)     return shim.Error("Incorrect number of arguments. Expecting 1")
(4)   }
(5)   studentAsBytes, _ := APIstub.GetState(args[0])
(6)   return shim.Success(studentAsBytes)
(7) }

```

ALGORITHM 6: Chaincode for university and teachers to check student's information on the blockchain.

```

(1) func (t *Chaincode) AwardCredit(APIstub shim.ChaincodeStubInterface, args []string) peer.Response {
(2)   if len(args) != 2 {
(3)     return shim.Error("Incorrect number of arguments. Expecting 2")
(4)   }
(5)   studentAsBytes, _ := APIstub.GetState(args[0])
(6)   student := Student{}
(7)   json.Unmarshal(studentAsBytes, &student)
(8)   student.Credit = args[5].
(9)   studentAsBytes, _ = json.Marshal(student)
(10)  APIstub.PutState(args[0], studentAsBytes)
(11)  return shim.Success(nil)
(12) }

```

ALGORITHM 7: Chaincode for the university to award student credit.

$$Y| \equiv \#(\langle \text{hash}(ID_X, M_X, TS_X) \rangle_{x_{X-Y}}). \quad (2)$$

By A6, statement 1, and the message meaning rule, derive

$$Y| \equiv X| \sim (\langle \text{hash}(ID_X, M_X, TS_X) \rangle_{x_{X-Y}}). \quad (3)$$

By statement 2, statement 3, and the nonce verification rule, derive

$$Y| \equiv X| \equiv (\langle \text{hash}(ID_X, M_X, TS_X) \rangle_{x_{X-Y}}). \quad (4)$$

By statement 4 and the belief rule, derive

$$Y| \equiv X| \equiv X \stackrel{x_{X-Y}}{\leftrightarrow} Y. \quad (5)$$

By A6, statement 5, and the jurisdiction rule, derive

$$Y| \equiv X \stackrel{x_{X-Y}}{\leftrightarrow} Y. \quad (6)$$

By statement 6 and the belief rule, derive

$$Y| \equiv X| \equiv ID_X. \quad (7)$$

By A8, statement 7, and the jurisdiction rule, derive

$$Y| \equiv ID_X. \quad (8)$$

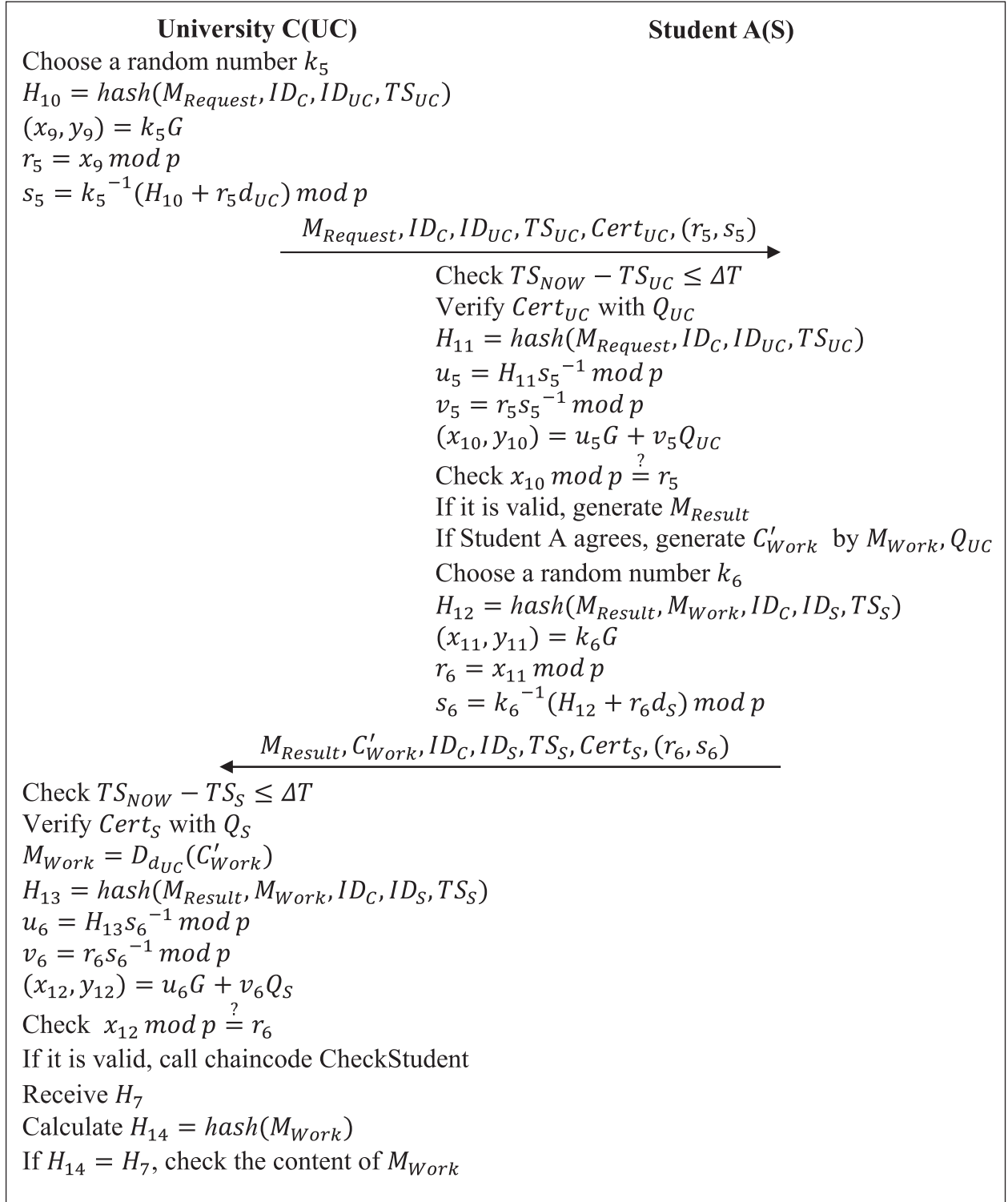


FIGURE 7: Credit verification phase.

4.1.2. *User X Authenticates User Y.* By M2 and the seeing rule, derive

$$X \triangleleft (\langle \text{hash}(ID_Y, M_Y, TS_Y) \rangle_{x_Y-X}). \quad (9)$$

By A3 and the freshness rule, derive

$$X | \equiv \# (\langle \text{hash}(ID_Y, M_Y, TS_Y) \rangle_{x_Y-X}). \quad (10)$$

By A5, statement 9, and the message meaning rule, derive

$$X | \equiv Y | \sim (\langle \text{hash}(ID_Y, M_Y, TS_Y) \rangle_{x_Y-X}). \quad (11)$$

By statement 10, statement 11, and the nonce verification rule, derive

$$X| \equiv Y| \equiv \left(\langle \text{hash}(\text{ID}_Y, M_Y, \text{TS}_Y) \rangle_{x_{Y-X}} \right). \quad (12)$$

By statement 12 and the belief rule, derive

$$X| \equiv Y| \equiv X \stackrel{x_{Y-X}}{\leftrightarrow} Y. \quad (13)$$

By A5, statement 13, and the jurisdiction rule, derive

$$X| \equiv X \stackrel{x_{Y-X}}{\leftrightarrow} Y. \quad (14)$$

By statement 14 and the belief rule, derive

$$X| \equiv Y| \equiv \text{ID}_Y. \quad (15)$$

By A7, statement 15, and the jurisdiction rule, derive

$$X| \equiv \text{ID}_Y. \quad (16)$$

By statement 5, statement 6, statement 7, statement 8, statement 13, statement 14, statement 15, and statement 16, it can be proven that User X and User Y authenticate each other.

4.2. Data Integrity. In this paper, Elliptic Curve Digital Signature Algorithm (ECDSA) is used to ensure data integrity, and any important data transmitted in the system need to be signed by the sender. In the process of data transmission, the sender first calculates the hash value of the data and then generates the digital signature by the hash value and his private key. After receiving the data and the digital signature from the sender, the receiver also calculates the hash value of the data and then searches for the public key of the sender. If the sender's signature can be generated by the receiver's hash value and the sender's public key, then it means that the data transmitted in the system has not been tampered with or lost.

Scenario: for example, Student A wants to send the course learning application M_{Request} of Course B to Teacher B. During data transmission, the data may be lost, or someone wants to maliciously tamper with the content of the application, then Teacher B receives M'_{Request} which is different from M_{Request} .

Analysis: after receiving M'_{Request} , ID_C , ID_S , TS_S , Cert_S , (r_3, s_3) from Student A and verifying the timestamp and Cert_S , Teacher B calculates $H' = \text{hash}(M'_{\text{Request}}, \text{ID}_C, \text{ID}_S, \text{TS}_S)$, $u' = H' s_3^{-1} \bmod p$, $v_3 = r_3 s_3^{-1} \bmod p$, $(x', y') = uG + v_3 Q_S$, if $x' \bmod p \neq r_3$ and then realizes that M'_{Request} was not generated by Student A.

4.3. Privacy Protection. In order to protect the students' privacy, the homework and final examinations of students in the system will be encrypted before transmission. When sending homework and final examination message, the student will encrypt the message with the receiver's public key, and the receiver decrypts the message with his or her private key before reviewing the content of the homework and final examination.

Scenario: University C wants to verify the credit of Student A for Course B, then Student A sends C_{Work} which is generated by his homework and final examination M_{Work} and the public key Q_{UC} to University C. Supposing that an attacker wants to obtain the homework and final examination without the permission of Student A, attacker intercepts the data when Student A transmits the message.

Analysis: by intercepting the data, the attacker obtains C_{Work} which is displayed as a meaningless string and the attacker cannot get any valid information from C_{Work} . If the attacker wants to decrypt C_{Work} , he needs to use the private key of University C to calculate $M_{\text{Work}} = D_{d_{\text{UC}}}(C_{\text{Work}})$. However, d_{UC} is known only to University C and is not accessible to the attacker, and thus the attacker cannot get the content of the homework and final examination.

4.4. Decentralization and Distribution. The blockchain system has the characteristics of decentralization and distribution, and the data on the blockchain are not managed by a central organization. In the proposed system, universities jointly maintain the data on the blockchain, the entire ledger is stored on the peer nodes of each organization, and any user's operation on the blockchain in the system is executed synchronously on each peer node. Therefore, the data will remain consistent across the organization, and any two members of the system can reach a consensus on the data on the blockchain such as students' credits, course grades, and the hash values that students upload.

Scenario: a Peer node in the system fails and the data on the node is lost.

Analysis: the remaining peer nodes in the system can still operate normally for transaction endorsement and ledger updates. Since the entire ledger is stored on each peer node, the loss of one copy of the ledger will not affect the operation of the system.

4.5. Traceability. All transactions in a blockchain system are packaged into blocks and arranged in chronological order, so all operations of the system users on the data on the blockchain can be traced, and cheating can be eliminated by using the traceability of blockchain.

Scenario: supposing that Student A cannot finish Course B and fails the final examination, Teacher B wants to help Student A cheat by calling the chaincode AddGrade to give him a high score g' .

Analysis: University A wants to verify the credit of Student A for Course B, and gets the grade g' of Student A by invoking the chaincode CheckStudent. The administrator of University A will first review the content of the homework and final examination, finding that M_{Work} is unqualified and cannot meet g' , then complain to the ministry of education. The education department queries all transactions related to Teacher B and Student A on the blockchain to find the block where Teacher B invoked the chaincode AddGrade, which contains the irregular operation of Teacher B and the timestamp, and can be used as evidence for prosecution.

TABLE 2: The computation cost analysis of the proposed scheme.

| Phase role | User registration phase | Course registration phase | Course learning phase | Credit application phase | Credit verification phase |
|---------------|-------------------------|--|--|-------------------------------------|--|
| Administrator | $1T_{\text{Cmp}}$ | N/A | N/A | N/A | N/A |
| CA | $1T_{\text{Mul}}$ | N/A | N/A | N/A | N/A |
| University A | N/A | N/A | N/A | $1T_{\text{Cmp}} + 2T_{\text{Sig}}$ | N/A |
| University B | N/A | $7T_{\text{Mul}} + 2T_H + 3T_{\text{Cmp}} + 2T_{\text{Sig}}$ | N/A | N/A | N/A |
| University C | N/A | N/A | N/A | N/A | $7T_{\text{Mul}} + 3T_H + 4T_{\text{Cmp}} + 1T_{\text{Enc}} + 2T_{\text{Sig}}$ |
| Teacher B | N/A | $7T_{\text{Mul}} + 2T_H + 3T_{\text{Cmp}} + 1T_{\text{Sig}}$ | $8T_{\text{Mul}} + 2T_H + 6T_{\text{Cmp}} + 1T_{\text{Enc}} + 2T_{\text{Sig}}$ | N/A | N/A |
| Student A | N/A | N/A | $6T_{\text{Mul}} + 3T_H + 1T_{\text{Enc}} + 3T_{\text{Sig}}$ | N/A | $7T_{\text{Mul}} + 2T_H + 3T_{\text{Cmp}} + 1T_{\text{Enc}} + 1T_{\text{Sig}}$ |

T_{Mul} : multiplication operation T_{Enc} : asymmetric encryption. T_H : Hash function operation T_{Sig} : signature operation. T_{Cmp} : comparison operation.

TABLE 3: Communication cost analysis of the proposed scheme.

| Item Phase | Message length (bits) | Round | 3.5 G (14 Mbps) | 4 (ms) G (100 Mbps) | 5 (ms) G (20 Gbps) |
|---------------------------|-----------------------|-------|-----------------|---------------------|--------------------|
| User registration phase | 640 | 3 | 0.046 | 0.006 | 0.032 us |
| Course registration phase | 4272 | 2 | 0.305 | 0.043 | 0.214 us |
| Course learning phase | 6320 | 2 | 0.451 | 0.063 | 0.316 us |
| Credit application phase | 240 | 1 | 0.017 | 0.002 | 0.012 us |
| Credit verification phase | 6400 | 2 | 0.457 | 0.064 | 0.320 us |

TABLE 4: The comparison of the previous schemes and the proposed scheme.

| Authors | Year | Objective | 1 | 2 | 3 | 4 | 5 |
|-----------------------|------|--|---|---|---|---|---|
| Turkanović et al. [9] | 2018 | Higher education credit platform | Y | N | N | N | N |
| Zhao et al. [10] | 2020 | System for student e-portfolio assessment | Y | N | Y | N | N |
| Mishra et al. [11] | 2021 | System for sharing students' credentials | Y | N | N | Y | Y |
| Jeong et al. [12] | 2021 | Multilateral personal portfolio authentication system | Y | N | Y | N | N |
| The proposed scheme | 2021 | A university course learning system with credit verifiable | Y | Y | Y | Y | Y |

Notes: 1—propose an architecture or framework, 2—verifiable learning process, 3—no token consumption, 4—encrypt private information, 5—security analysis, Y—yes, N—no.

4.6. Credit Is Verifiable. In the proposed system, a university can verify the credits of students from other universities to determine whether the students' abilities meet the university's requirements. If student's homework and final examination are sent to the university, then it means that all teachers and administrators in the university can review the content of the homework and final examination.

Scenario: Student A obtains the credit for Course B and wants to transfer from University A to University C. To verify the learning situation of Student A on Course B, University C needs to review content of the homework and final examination.

Analysis: Student A sends C_{Work} which was generated by his homework and final examination M_{Work} and the public key Q_{UC} to University C, and University C decrypt C_{Work} with the private key d_{UC} to generate M_{Work} . For ensuring that M_{Work} generated by University C are the same as the homework and final examination submitted by Student A in the course learning phase, the administrator of University C invokes the chaincode CheckStudent to get the hash value H_{Work} of the homework and final examination which was uploaded by Student A and calculates $H'_{\text{Work}} = \text{hash}(M_{\text{Work}})$. If $H'_{\text{Work}} = H_{\text{Work}}$, then it means that M_{Work} sent by Student A were not tampered with, and the administrator reviews the content of the homework and final examination.

5. Discussion

5.1. Computation Cost Analysis. The computation cost analysis of the proposed scheme is shown in Table 2, and the highest computation cost is found in the course learning phase. Teacher B requires 8 multiplication operations, 2 hash function operations, 6 comparison operations, 1 asymmetric encryption operation, and 2 signature operations. Student A requires 6 multiplication operations, 3 hash function operations, 1 asymmetric encryption operation, and 3 signature operations. Thus, the proposed scheme has a good computational cost.

5.2. Communication Cost Analysis. Table 3 shows the communication efficiency of the proposed system. It is assumed that the ECDSA key and signature require 160 bits, course, homework, and final examination message and certificate require 1024 bits, and encrypted homework and exam message require 3072 bits, while other messages, like timestamp, identity information, request message, and result from the message, require 80 bits. Taking the credit verification phase, for example, it requires four ECDSA signatures, two certificates, an encrypted homework and exam message, and eight other messages. Thus, it requires $160 \times 4 + 1024 \times 2 + 3072 \times 1 + 80 \times 8 = 6400$ bits in total, which takes 0.457 ms under 3.5 G (14 Mbps) communication environment, 0.064 ms under 4G (100 Mbps) communication environment, and 0.320 us under 5G (20 Mbps) communication environment.

5.3. Comparison. Table 4 shows the comparison of the previous schemes and the proposed scheme. Compared to the related works, the proposed scheme focuses on proposing a university course learning system which has the advantages of verifiable learning process, no token consumption, protection of students' privacy and complete security analysis.

6. Conclusions

With the increase in the number of university online courses and students, the problem of credit verification becomes inevitable. Previously, credits were managed by each university and each online education platform alone, which led to the fact that credits earned by students in one university could not be recognized by other universities. This paper proposes a cross-university course learning system based on Hyperledger Fabric, which stores students' credits and hash values of the homework and final examinations on the blockchain, and the data on the blockchain are jointly maintained by all universities. For universities, they can

verify students' credits and review the content of students' homework and final examinations by invoking the chaincode, to determine whether students' abilities meet their requirements and then recognize the credits.

This paper shows a complete system architecture, details the application scenario, and provides the chaincode. To improve the security of the system, students' homework and final examinations are encrypted before transmission, thus effectively protecting students' privacy. At the same time, the Elliptic Curve Digital Signature Algorithm in Hyperledger Fabric can ensure data integrity during communication. The security analysis using BAN logic shows that our proposed system enables mutual authentication of the system members. Compared with previous systems based on blockchain technology, the universities in the proposed system recognize credits by reviewing students' homework and final examinations, rather than relying on the authority of central educational institutions. The proposed scheme uses consortium blockchain architecture, which improves system operating efficiency and saves the money needed for mining compared to public blockchain architecture. The final analysis shows that the proposed system also performs well in terms of computational cost and communication cost.

To sum up, this research achieved the following contributions:

- (1) Proposes a cross-university course learning system based on Hyperledger Fabric where universities can review students' homework and final examinations to recognize students' credits.
- (2) Proposes a complete system architecture, details the application scenario, and provides the chaincode.
- (3) Uses Elliptic Curve Digital Signature Algorithm to ensure data integrity during communication and asymmetric encryption algorithm to protect students' privacy.
- (4) Uses consortium blockchain architecture to improve system operating efficiency and save the money needed for mining compared to public blockchain architecture
- (5) Presents security analysis through BAN logic to ensure mutual authentication of the system members

In the future, the research will consider adding the role of enterprises in the system to realize the verification of students' credits and learning records by enterprises, thus facilitating enterprises to understand the ability of students and select the students they need. At the same time, as the number of system members increases, how to ensure the privacy of students and system operating efficiency also need to be considered.

Notations

User X: Any university or teacher or student in the system
 E: The elliptic curve

G : The origin generated on the elliptic curve E
 Cert $_X$: A digital certificate of User X conforms to the X.509 standard
 d_X : The private key of User X based on Elliptic Curve Cryptography
 Q_X : The public key of User X based on Elliptic Curve Cryptography
 ID_C : The identity of a course
 ID_X : The identity of User X
 TS_X : Timestamp of User X
 $M_{Request}$: Request message sent by a user
 M_{Result} : The resulting message of the request
 $M_{Identity}$: Identity information message of a user
 M_{Course} : Course message of a teacher
 M_{Work} : Homework and final examination message of a student
 g : The grade of a student for a course
 k_i : The i th random number generated by the user
 (r_i, s_i) : The i th digital signature generated by the user
 hash(\cdot): One way hash function
 H_i : The i th hash value generated by the user
 $E_{Q_X}(M)$: Asymmetrically encrypt the message M with the public key Q_X
 $D_{d_X}(M)$: Asymmetrically decrypt the message M with the private key d_X
 C_{Work} : The cyphertext of homework and final examination message generated by asymmetric encryption
 $A \stackrel{?}{=} B$: Verify whether A is equal to B .

Data Availability

The data supporting this study are available within the article.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work was supported in part by the Ministry of Science and Technology, Taiwan, R.O.C., under contract MOST 110-2218-E-305-001-MBK and MOST 110-2410-H-324 -004 -MY2, and the Education and Teaching Reform Project of the Xiamen University of Technology (no. JG2021007).

References

- [1] Q. Yang and Y. C. Lee, "The critical factors of student performance in MOOCs for sustainable education: a case of Chinese universities," *Sustainability*, vol. 13, no. 8089, 2021.
- [2] "Udemy vs Coursera - which learning app is better," <https://www.mobileappdaily.com/udemy-vs-coursera/amp>.
- [3] "Filipino Coursera learners complete over 140K courses in 10 months," 2021, <https://mb.com.ph/2021/07/09/filipino-coursera-learners-complete-over-140k-courses-in-10-months-dost-chief/>.

- [4] "IIM-Kozhikode partners with Coursera to launch certificate programs," 2021, <https://www.thehindu.com/education/colleges/iim-kozhikode-partners-with-coursera-to-launch-certificate-programmes/article35039579.ece>.
- [5] "European credit transfer and accumulation system (ECTS)," 2021, https://ec.europa.eu/education/resources-and-tools/european-credit-transfer-and-accumulation-system-ects_en.
- [6] Credit Bank System(CBS), <http://www.cb.or.kr/>, 2021.
- [7] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system," 2021, <https://bitcoin.org/bitcoin.pdf>.
- [8] A. Alammary, S. Alhazmi, M. Almasri, and S. Gillani, "Blockchain-based applications in education: a systematic review," *Applied Sciences*, vol. 92400 pages, 2019.
- [9] C. L. Chen, M. L. Chiang, Y. Y. Deng, W. Weng, K. Wang, and C.-C. Liu, "A traceable firearm management system based on blockchain and IoT technology," *Symmetry*, vol. 13, no. 439, 2021.
- [10] C. L. Chen, Y. Y. Deng, W. J. Tsaur, C.-Ta Li, C.-C. Lee, and C.-M. Wu, "A traceable online insurance claims system based on blockchain and smart contract technology," *Sustainability*, vol. 13, no. 9386, 2021.
- [11] Y. C. Wang, C. L. Chen, and Y. Y. Deng, "Museum-authorization of digital rights: a sustainable and traceable cultural relics exhibition mechanism," *Sustainability*, vol. 13, no. 2046, 2021.
- [12] C. L. Chen, C. Y. Lin, M. L. Chiang, Y.-Y. Deng, P. Chen, and Yi-J. Chiu, "A traceable online will system based on blockchain and smart contract technology," *Symmetry*, vol. 13, no. 6, 2021.
- [13] M. Sharples and J. Domingue, "The blockchain and kudos: a distributed system for educational record, reputation and reward," in *Proceedings of the European Conference on Technology Enhanced Learning*, pp. 490–496, Springer, Cham, Midtown Manhattan, New York City, September 2016.
- [14] M. Turkanovic, M. Holbl, K. Kopic, M. Hericko, and A. Kamisalic, "EduCTX: a blockchain-based higher education credit platform," *IEEE Access*, vol. 6, pp. 5112–5127, 2018.
- [15] G. Zhao, H. He, B. Bi, Q. Xia, and Z. Fu, "A blockchain-based system for student e-portfolio assessment using smart contract," in *Proceedings of the 2020 4th International Conference on Computer Science and Artificial Intelligence*, pp. 34–40, ACM, Zhuhai China, 11 December 2020.
- [16] R. A. Mishra, A. Kalla, A. Braeken, and M. Liyanage, "Privacy protected blockchain based architecture and implementation for sharing of students' credentials," *Information Processing & Management*, vol. 58102512 pages, 2021.
- [17] J. Jeong, D. Kim, S.-Y. Ihm, Y. Lee, and Y. Son, "Multilateral personal portfolio authentication system based on hyperledger fabric," *ACM Transactions on Internet Technology*, vol. 21, no. 1, pp. 1–17, 2021.
- [18] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ECDSA)," *International Journal of Information Security*, vol. 1, no. 1, pp. 36–63, 2001.
- [19] N. Szabo, "Smart contracts: building blocks for digital markets," *EXTROPY J. Transhumanist Thought*, vol. 18, no. 16, 1996.
- [20] Y. C. Wang, C. L. Chen, and Y. Y. Deng, "Authorization mechanism based on blockchain technology for protecting museum-digital property rights," *Applied Sciences*, vol. 111085 pages, 2021.
- [21] E. Androulaki, A. Barger, V. Bortnikov et al., "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proceedings of the EuroSys '18: Proceedings of the Thirteenth EuroSys Conference*, pp. 1–15, ACM, Porto Portugal, 23 April 2018.
- [22] N. Lu, Y. Zhang, W. Shi, S. Kumari, and K. K. R. Choo, "A secure and scalable data integrity auditing scheme based on hyperledger fabric," *Computers & Security*, vol. 92, Article ID 101741, 2020.
- [23] M. Macdonald, L. Liu-Thorold, and R. Julien, "The blockchain: a comparison of platforms and their uses beyond Bitcoin," *COMS4507 - Advanced Computer and Network Security*, vol. 54, 2017.
- [24] P. Yabo, "Key metrics of blockchain platforms," <https://docs.google.com/spreadsheets/d/1DQ770nGnHfjOoRSqTLmIkhuVK5CABsFgqb6UoGMfVM/edit%23gid=0>.
- [25] T.-T. Kuo, H. Zavaleta Rojas, and L. Ohno-Machado, "Comparison of blockchain platforms: a systematic review and healthcare examples," *Journal of the American Medical Informatics Association*, vol. 26, no. 5, pp. 462–478, 2019.
- [26] K. Toyoda, K. Machi, Y. Ohtake, and A. N. Zhang, "Function-level bottleneck analysis of private proof-of-authority ethereum blockchain," *IEEE Access*, vol. 8, pp. 141611–141621, 2020.
- [27] "Open source, but private," <https://www.corda.net/blog/open-source-but-private-a-case-for-private-decentralized-ledger-tech>.
- [28] "Build on Quorum, the complete open source blockchain platform for business," <https://consensys.net/quorum>.
- [29] MultiChain for Developers: <https://www.multichain.com/developers/>.
- [30] J. Polge, J. Robert, and Y. Le Traon, "Permissioned blockchain frameworks in the industry: a comparison," *ICT Express*, vol. 7, no. 2, pp. 229–233, 2021.
- [31] "Blockchain still not enterprise ready, but the Hyperledger Fabric 1.0 release can show the way," https://www.hfsresearch.com/blockchain/blockchain-not-ready-but-hyperledger-fabric-release-show-the-way_071217.
- [32] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," *ACM Transactions on Computer Systems*, vol. 8, no. 1, pp. 18–36, 1990.